Effective March 2013
Supersedes December 2010

Eaton Logic Controller

Programming Manual

Powering Business Worldwide™

ELC Programming Manual

Publication History

Release	Description of Changes	Date
First Edition	The first edition is issued.	2010-12-09
Second Edition	The descriptions of ELC-PB/PC/PH are deleted, and the descriptions of ELC2 series are added to the manual.	2014-05-02

ELC SERIES PLCS

Programming Manual

Table of Contents

Chapter 1 – ELC Concepts

Chapter 2 – Programming Concepts

Chapter 3 – Instruction Set

Chapter 4 – Sequential Function Chart

Chapter 5 – Setting and Using an Ethernet ELC/Module

Appendix A – Communications

Appendix B – Troubleshooting

Appendix C – Installing a USB Driver in the ELC

ELC Concepts

This chapter introduces basic and advanced concepts of ladder logic, which is the most used programming language with the ELC. Users familiar with the ELC concepts can move to the next chapter for further programming concepts. Users not familiar with the operating principles of the ELC, should read this chapter to get a full understanding of these concepts.

This Chapter Contains

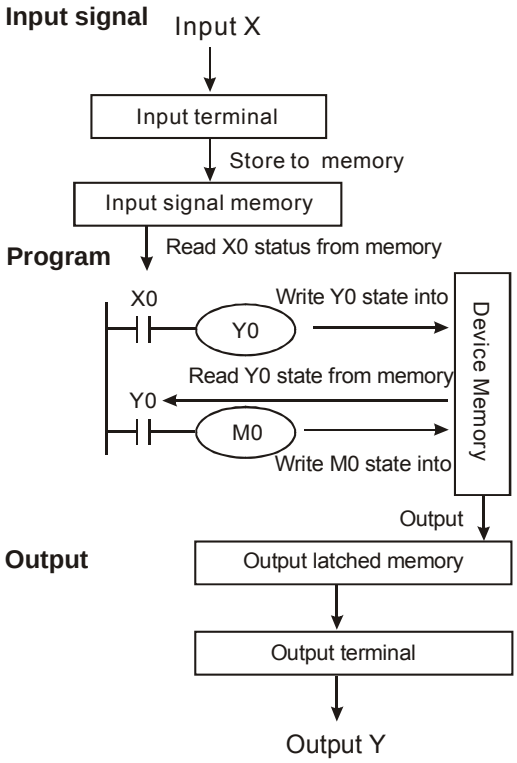
1 ELC Concepts

1.1 ELC Scan Method

ELC utilizes a standard scan method when evaluating the user program.

Scanning process:

Scan input status	Read the physical input status and store the data in internal memory.
Evaluate user program	Evaluate the user program with data stored in internal memory. Program scanning starts from top to bottom and left to right until reaching the end of the program.
Refresh the outputs	Write the evaluated data to the physical outputs



Input signal:
The ELC reads the ON/OFF status of each input and stores the status into memory before evaluating the user program.
Once the external input status is stored into internal memory, any change at the external inputs will not be updated until next scan cycle starts.

Program:
The ELC executes instructions in the user program from top to down and left to right then stores the evaluated data into internal memory. Some of this memory is latched.

Output:
When END command is reached the program evaluation is complete. The output memory is transferred to the external physical outputs.

Scan time
The duration of the full scan cycle (read, evaluate, write) is called the “scan time.” With more I/O or a longer program, the scan time becomes longer.

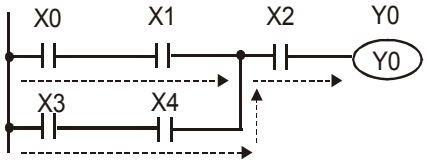
Read scan time	The ELC measures its own scan time and stores the value (0.1ms) in register D1010, the minimum scan time in register D1011, and the maximum scan time in register D1012.
Measure scan time	Scan time can also be measured by toggling an output every scan and then measuring the pulse width on the output being toggled.
Calculate scan time	Scan time can be calculated by adding the known time required for each instruction in the user program. For scan time information for each instruction, please refer to Ch3 in this manual.

Scan time exception
The ELC can process certain items faster than the scan time. Program interrupts halt the scan time to process the interrupt subroutine program. A direct I/O refresh instruction REF allows the ELC to access I/O immediately during user program evaluation instead of waiting until the next scan

cycle.

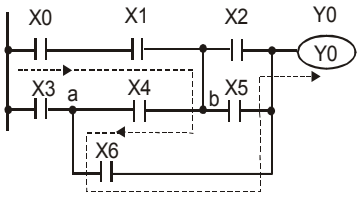
1.2 Current Flow

Ladder logic follows a left to right principle. In the example below, the current flows through paths started from either X0 or X3.

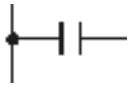
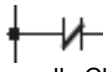


Reverse Current

When a current flows from right to left, which makes a reverse current logic, an error will be detected when compiling the program. The example below shows the reverse current flow.



1.3 NO Contact, NC Contact

NO contact	 Normally Open Contact, A contact
NC Contact	 Normally Closed Contact, B contact

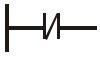
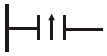
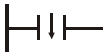
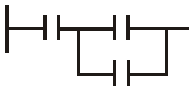
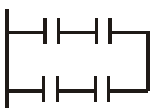
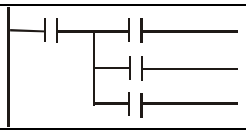
1.4 ELC Registers and Relays

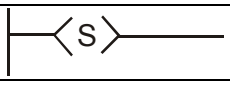
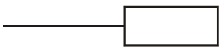
Introduction to the basic internal devices in an ELC

X (Input Relay)	Bit memory which represents the physical input points and receives external input signals. ■ Represented as **X** and numbered in octal, e.g. X0~X7, X10~X17…
Y (Output Relay)	Bit memory which represents the physical output points and saves the status to be refreshed to physical output devices. ■ Represented as **Y** and numbered in octal, e.g. Y0~Y7, Y10~Y17...
M (Internal Relay)	Bit memory indicates ELC status. ■ Internal bit memory: represented as **M** and numbered in decimal, e.g. M0, M1, M2…
S (Step Relay)	Bit memory indicates ELC status in Step Function Control (SFC) mode. If no STL instruction is applied in program, step point S can be used as an internal relay M as well as an annunciator. ■ Internal bit memory: represented as **S** and numbered in decimal, e.g. S0, S1, S2…
T (Timer) (Relay) (Word)	Bit, word or double word memory used for timing. When its coil is ON and the set time is reached, the associated contact will be energized. Every timer has its resolution (unit: 1ms/10ms/100ms). ■ Represented as **T** and numbered in decimal, e.g. T0, T1, T2…
C (Counter) (Relay) (Word) (Dword)	Bit, word or double word memory used for counting. The counter counts once (1 pulse) when the coil goes from OFF to ON. When the predefined counter value is reached, the associated contact will be energized. There are 16-bit and 32-bit high-speed counters available for users. ■ Represented as **C** and numbered in decimal, e.g. C0, C1, C2…
D (Data register) (Word)	Word memory stores values and parameters for data operations. Every register is able to store a word (16-bit binary value). A double word will occupy 2 consecutive data registers. ■ Represented as **D** and numbered in decimal, e.g. D0, D1, D2…
E, F (Index register) (Word)	Word memory used as a modifier to indicate a specified device (word and double word) by defining an offset. Index registers not used as a modifier can be used as general purpose registers. ■ Represented as E0 ~ E7 and F0 ~ F7.

1.5 Ladder Logic Symbols

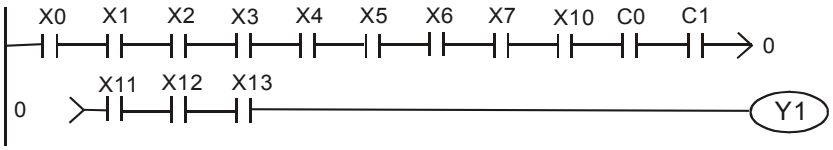
The following table displays the list of ELCSoft program symbols, along with their description, command, and memory registers that use them.

Ladder Diagram Structure	Explanation	Instruction	Available Devices
	NO (Normally Open) contact / A contact	LD	X, Y, M, S, T, C
	NC (Normally Closed) contact / B contact	LDI	X, Y, M, S, T, C
	NO contact in series	AND	X, Y, M, S, T, C
	NC contact in series	ANI	X, Y, M, S, T, C
	NO contact in parallel	OR	X, Y, M, S, T, C
	NC contact in parallel	ORI	X, Y, M, S, T, C
	Rising-edge trigger switch	LDP	X, Y, M, S, T, C
	Falling-edge trigger switch	LDF	X, Y, M, S, T, C
	Rising-edge trigger in series	ANDP	X, Y, M, S, T, C
	Falling-edge trigger in series	ANDF	X, Y, M, S, T, C
	Rising-edge trigger in parallel	ORP	X, Y, M, S, T, C
	Falling-edge trigger in parallel	ORF	X, Y, M, S, T, C
	Block in series	ANB	None
	Block in parallel	ORB	None
	Multiple output branches	MPS MRD MPP	None

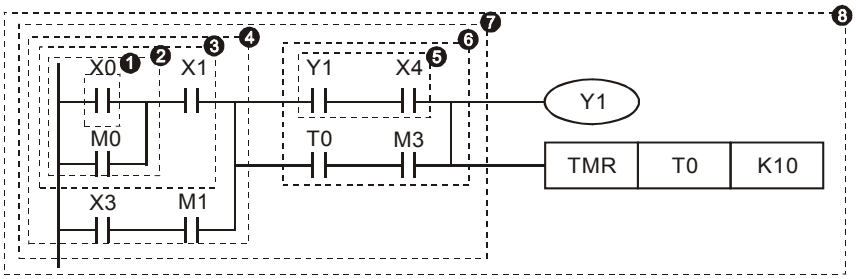
Ladder Diagram Structure	Explanation	Instruction	Available Devices
	Output coil	OUT	Y, M, S
	Step ladder	STL	S
	Basic / Application instruction	-	Basic instructions and API instructions. Please refer to chapter 3 Instruction Set
	Inverse logic	INV	None

1.5.1 CREATING A ELC LADDER PROGRAM

Editing a program should start from the left side bus line to the right side bus line, and from top to bottom. However, the right side bus line is omitted when editing in ELCSoft. A single row can have a maximum of 11 contacts. If more than 11 contacts are connected, a continuous symbol “0” will be generated automatically and the 12th contact will be placed at the start of next row. The same input points can be used repeatedly. See the figure below:



When evaluating the user program, the ELC scan starts from left to right and proceeds to the next row down until the ELC reaches END instruction. Output coils and basic / application instructions are output instructions and are placed at the right of ladder diagram. The sample program below explains the execution order of a ladder diagram. The numbers in the black circles indicate the execution order.

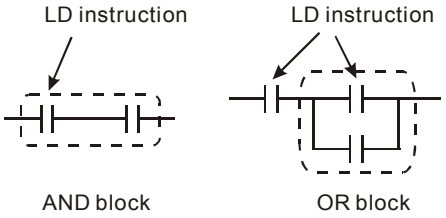


Execution order of the sample program:

1	LD	X0	
2	OR	M0	
3	AND	X1	
4	LD	X3	
	AND	M1	
	ORB		
5	LD	Y1	
	AND	X4	
6	LD	T0	
	AND	M3	
	ORB		
7	ANB		
8	OUT	Y1	
	TMR	T0	K10

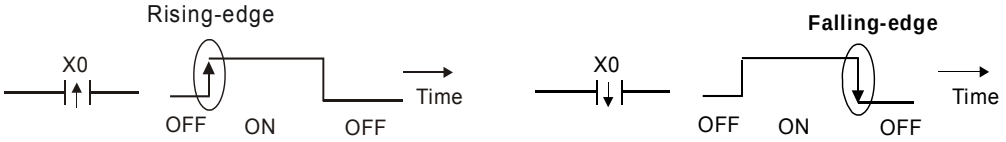
1.5.2 LD / LDI (LOAD NO CONTACT / LOAD NC CONTACT)

LD or LDI starts a row or block



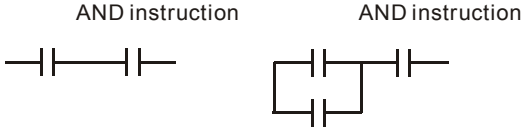
1.5.3 LDP / LDF (LOAD RISING EDGE TRIGGER/ LOAD FALLING EDGE TRIGGER)

Similar to the LD instruction, LDP and LDF instructions act on the rising edge or falling edge when the contact is ON, as shown in the figure below.



1.5.4 AND / ANI (CONNECT NO CONTACTS IN SERIES / CONNECT NC CONTACTS IN SERIES)

AND (ANI) instruction connects a NO (NC) contact in series with another device or block.

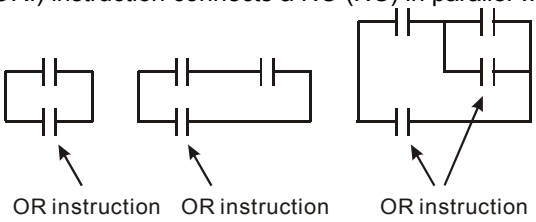


1.5.5 ANDP / ANDF (CONNECT RISING EDGE IN SERIES/ CONNECT FALLING EDGE IN SERIES)

Similar to AND instruction, ANDP (ANDF) instruction connects rising (falling) edge triggers in series with another device or block.

1.5.6 OR / ORI (CONNECT NO CONTACT IN PARALLEL / CONNECT NC CONTACT IN PARALLEL)

OR (ORI) instruction connects a NO (NC) in parallel with another device or block.

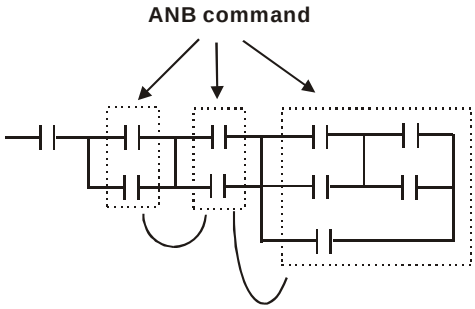


1.5.7 ORP / ORF (CONNECT RISING EDGE IN PARALLEL/ CONNECT FALLING EDGE IN PARALLEL)

Similar to OR instruction, ORP (ORF) instruction connects rising (falling) edge triggers in parallel with another device or block

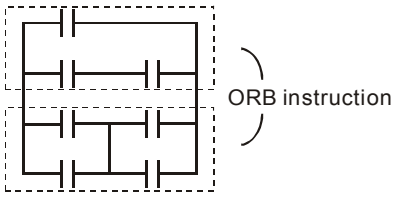
1.5.8 ANB (CONNECT BLOCK IN SERIES)

ANB instruction connects a block in series with another block



1.5.9 ORB (CONNECT BLOCK IN PARALLEL)

ORB instruction connects a block in parallel with another block

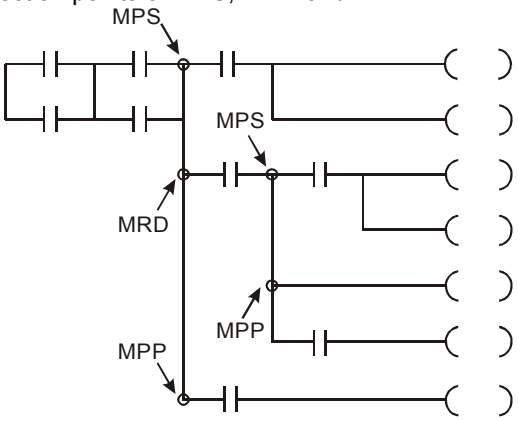


1.5.10 MPS / MRD / MPP (BRANCH INSTRUCTIONS)

These instructions provide a method to create multiplexed output branches based on the current result stored by the MPS instruction.

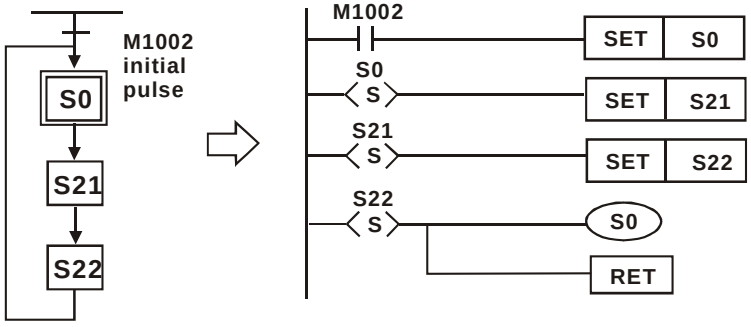
Branch instruction	Branch Symbol	Description
MPS	┬	Start of branches. Stores the current result of the program evaluation. Max. 8 MPS-MPP pairs can be applied
MRD	├	Reads the stored current result from the previous MPS
MPP	└	End of branches. Pops (reads then resets) the stored result in previous MPS

Note: When compiling ladder programs with ELCSoft, MPS, MRD and MPP could be automatically added to the compiled results in instruction format. However, sometimes the branch instructions are ignored by ELCSoft if not necessary. Users programming in instruction format can enter branch instructions as required.
Connection points of MPS, MRD and MPP:



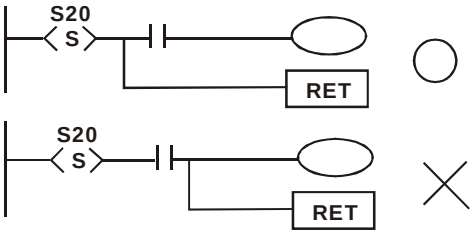
1.5.11 STL (STEP LADDER PROGRAMMING)

STL programming uses step points, e.g. S0 S21, S22, which allow users to program in a clear and understandable way like drawing a flow chart. The program will proceed to the next step only if the previous step is completed, therefore it forms a sequential control process similar to SFCs (Sequential Function Charts). The STL sequence can be converted into an ELC ladder diagram which is called “step ladder diagram” as below.



1.5.12 RET (RETURN)

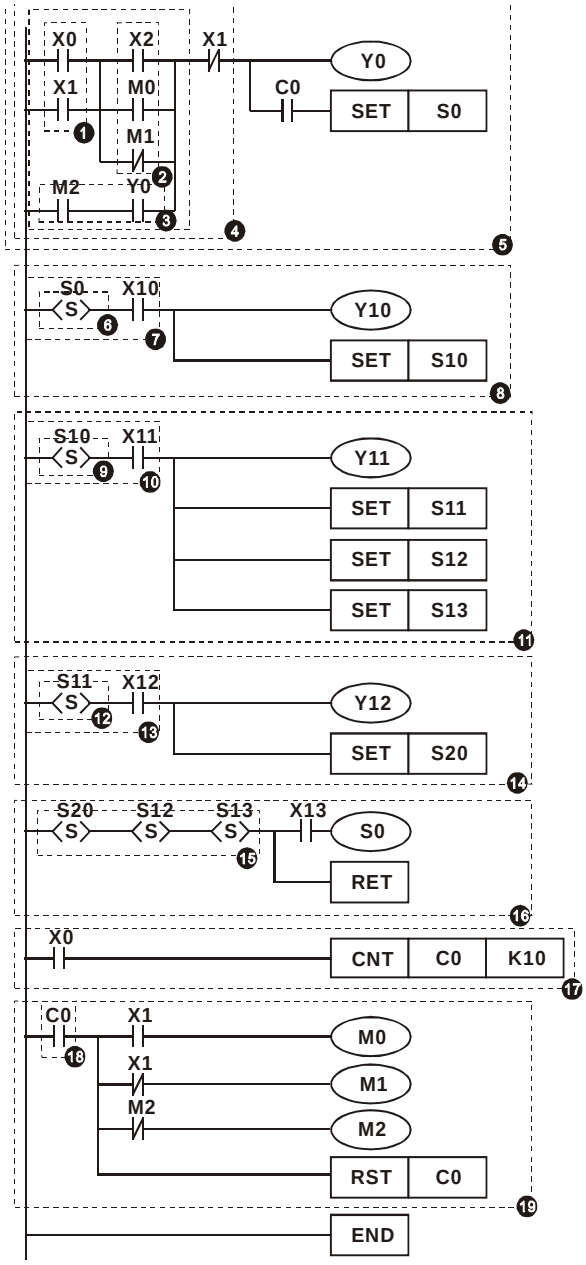
A RET instruction must be placed at the end of a sequential control process to indicate the completion of STL flow.



Note: Always connect the RET instruction immediately after the last step point indicated in the above diagram or a program error may occur.

1.6 Conversion between Ladder Diagram and Instruction List Mode

Ladder Diagram



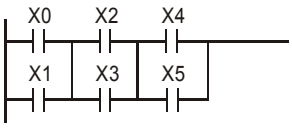
Instruction

```
LD    X0      ❶ OR block
OR    X1
LD    X2      ❷ OR block
OR    M0
ORI   M1
ANB   ←  Block in series
LD    M2      ❸ AND block
AND   Y0
ORB   ←  Block in parallel
AN I  X1      ❹ ANI
OUT   Y0      ⎫
AND   C0      ⎬ Multiple outputs
SET   S0      ⎭
                ❺ The output continues based on status of ❹
STL   S0      Start of step ladder ❻
LD    X10     S0 status operates with X10 ❼ ❽
OUT   Y10     ⎫ Output Y10 and
SET   S10     ⎭ transfer of step point
STL   S10     ❾ Read S10 status
LD    X11     ❿ S10 operates with X11 ⓫
OUT   Y11     ⎫
SET   S11     ⎬ Output Y11 and
SET   S12     ⎭ transfer of step points
SET   S13
STL   S11     ⓬ Read S11 status
LD    X12     ⓭ S11 operates with X12 ⓮
OUT   Y12     Output Y12 and
SET   S20     transfer of step points
STL   S20
STL   S12     ⎫ ⓯ Convergence of
STL   S13     ⎭ multiple status
LD    X13     ⎫ Read X13 status and
OUT   S0      ⎭ transfer of step point
                ⓰ End of step ladder
RET           Return
LD    X0
CNT   C0 K10  ⓱
LD    C0      ⓲ Read C0
MPS           ⎫
AND   X1      ⎪
OUT   M0      ⎪
MRD           ⎪
AN I  X1      ⎬ Multiple outputs
OUT   M1      ⎪
MPP           ⎪
AN I  M2      ⎪
OUT   M2      ⎪
RST   C0      ⎭
                ⓳
END           End of program
```

Fuzzy Syntax

Generally, the ladder diagram programming is scanned “top to bottom and left to right”. However, some programming methods do not follow this principle and still deliver the same control results. Here are some examples explaining this kind of “fuzzy syntax.”

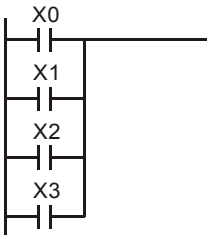
Example 1:



Better method		OK method	
LD	X0	LD	X0
OR	X1	OR	X1
LD	X2	LD	X2
OR	X3	OR	X3
ANB		LD	X4
LD	X4	OR	X5
OR	X5	ANB	
ANB		ANB	

The two instruction programs can be converted into the same ladder diagram. The difference between the Better and the OK method is the ANB operation. The ANB instruction cannot be used continuously more than 8 times. If more than 8 ANB instructions are used continuously, a program error will occur. Therefore, applying the ANB instruction after a block is made is the better method to prevent possible errors. In addition, it’s also the more logical and clearer programming method for general users.

Example 2:

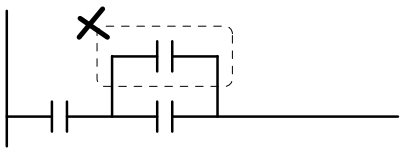


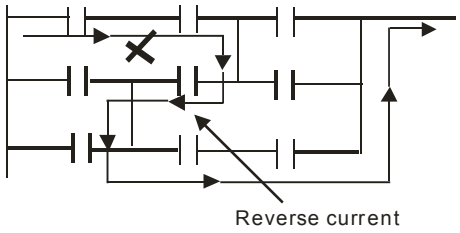
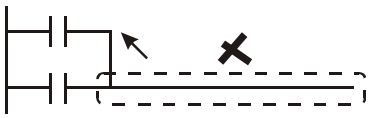
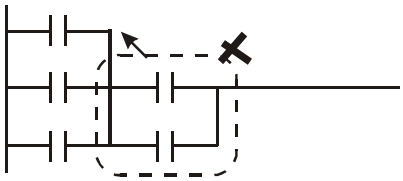
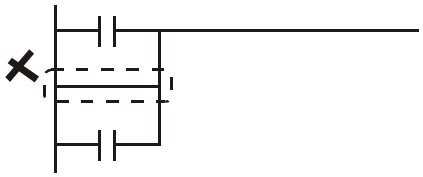
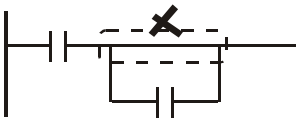
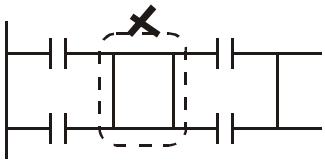
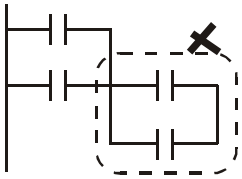
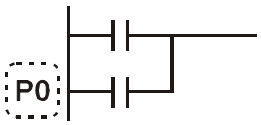
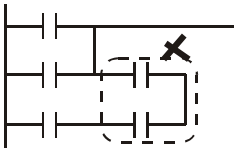
Good method		Bad method	
LD	X0	LD	X0
OR	X1	LD	X1
OR	X2	LD	X2
OR	X3	LD	X3
		ORB	
		ORB	
		ORB	

The difference between the Good and the Bad method is very clear. With longer program code, the required memory increases in the Bad method. Following the general principle and applying good / better methods when editing programs prevents possible errors and improves program execution speed as well.

Common Programming Errors

The ELC processes the diagram program from top to bottom and left to right. When editing ladder diagrams users should adopt this principle as well or an error would be detected by ELCSoft when compiling the user program. Common program errors are listed below:

	OR operation upward is not allowed.

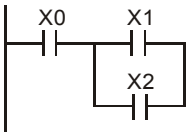
	“Reverse current” flow is not allowed.
	Output should be connected on top of the circuit.
	Block combination should be made on top of the circuit.
	Parallel connection without an instruction is not allowed.
	Parallel connection without an instruction is not allowed.
	No instruction in the middle block is not allowed.
	Instructions and blocks in series should be horizontally aligned
	Label P0 should be at the first row of the complete network.
	“Reverse current” exists

1.7 Correcting Ladder Diagram

There are many ways to accomplish your ladder logic. The list below displays methods for creating ladder logic. Some methods will not work and others could be better. For each method that will not work or could be better, there is a suggested improvement. Review the instructions for each

method. The improved method will shorten the number of instructions, saving memory and improving the scan time.

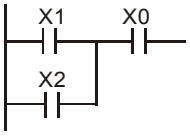
Example 1:
Connect the block to the front to eliminate the ANB instruction. This simplifies the program and improves processing speed



Instruction List

LD	X0
LD	X1
OR	X2
ANB	

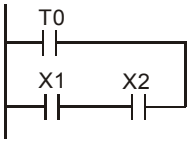
⇩



Instruction List

LD	X1
OR	X2
AND	X0

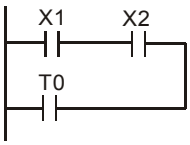
Example 2:
When an instruction is to be connected to a block, connect the instruction to the lower row to omit the ORB instruction.



Instruction List

LD	T0
LD	X1
AND	X2
ORB	

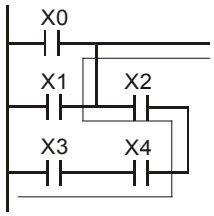
⇩



Instruction List

LD	X1
AND	X2
OR	T0

Example 3:
"Reverse current" shown in diagram (a) is not allowed by the ELC.

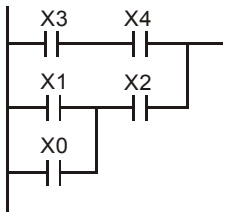


(a)

Instruction List

LD	X0
OR	X1
AND	X2
LD	X3
AND	X4
ORB	

⇩

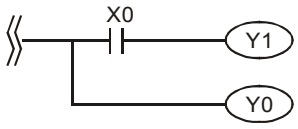


(b)

Instruction List

LD	X3
AND	X4
LD	X1
OR	X0
AND	X2
ORB	

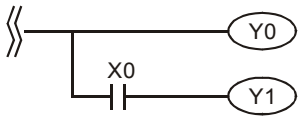
Example 4:
For multiple outputs, connect the output without additional input devices to the top of the circuit to omit the MPS and MPP instructions.



Instruction List

MPS	
AND	X0
OUT	Y1
MPP	
OUT	Y0

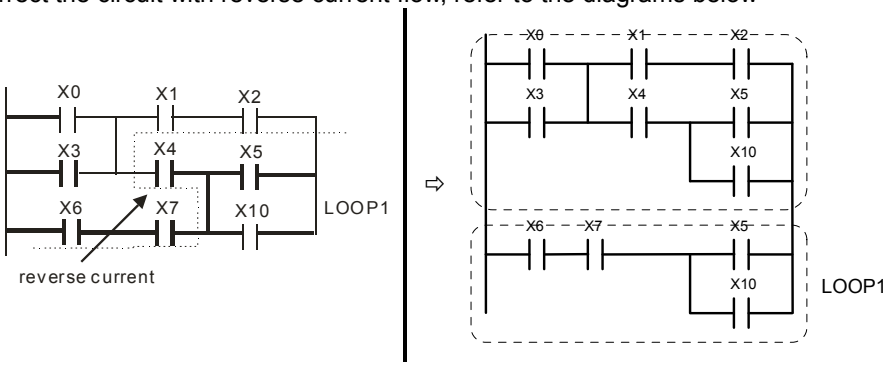
⇩



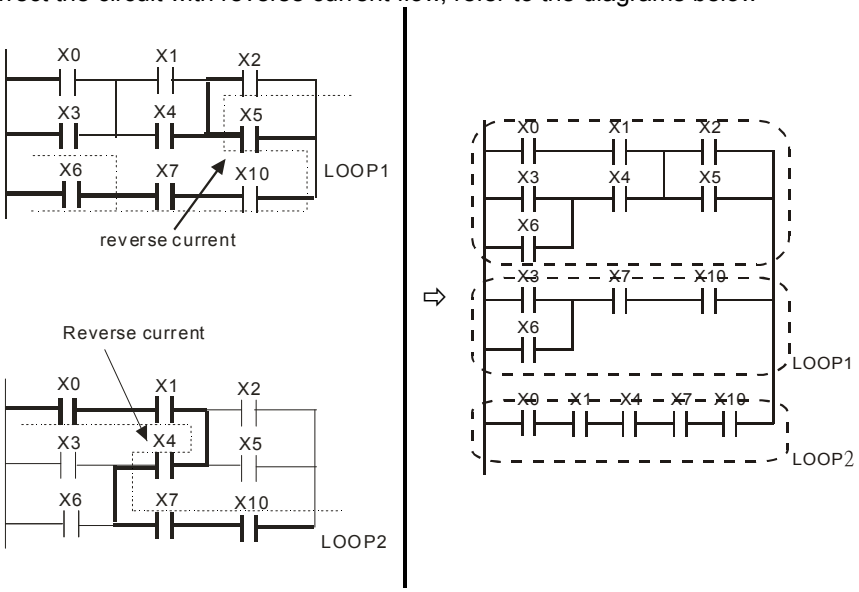
Instruction List

OUT	Y0
AND	X0
OUT	Y1

Example 5:
To correct the circuit with reverse current flow, refer to the diagrams below



Example 6:
To correct the circuit with reverse current flow, refer to the diagrams below

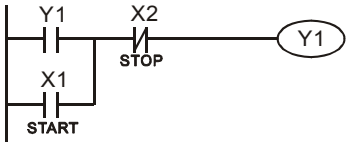


1.8 Basic Program Design Examples

The examples that follow illustrate how common functions can be programmed.

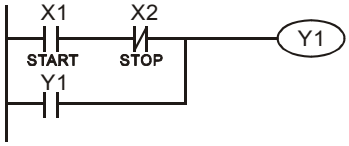
Example 1 - Stop First latched circuit

When X1 (START) = ON and X2 (STOP) = OFF, Y1 will be ON. If X2 is turned on, Y1 will be OFF. This is a Stop First circuit because the STOP button has the control priority over the START button.



Example 2 - Start First latched circuit

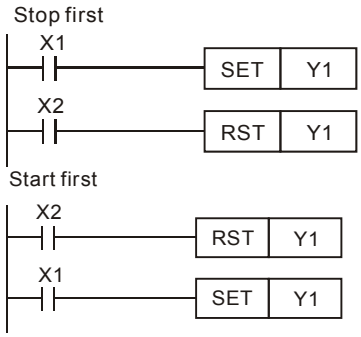
When X1 (START) = ON and X2 (STOP) = OFF, Y1 will be ON and latched. If X2 is turned ON, Y1 remains ON. This is a Start First circuit because the START button has the control priority over the STOP button.



Example 3 - Latched circuit using SET and RST

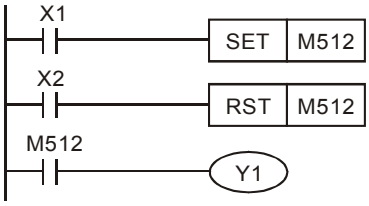
The diagrams are latched circuits using the RST and SET instructions.

The instruction encountered last in a program will determine the final state of Y1. Therefore, if both X1 and X2 are ON and the RST instruction is after the SET instruction, this forms a Stop First circuit. Conversely, if the SET instruction is after the RST instruction, this forms a Start First circuit.

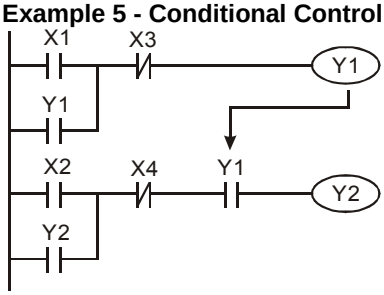


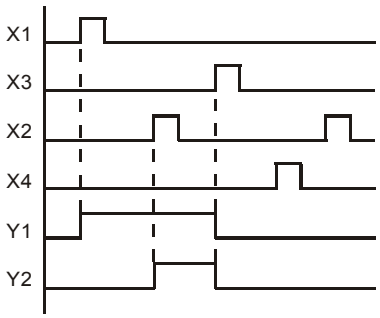
Example 4 - Power down latched circuit

The auxiliary relay M512 is a latched relay. Once X1 is ON, Y1 retains its status before power down and resumes after power up.



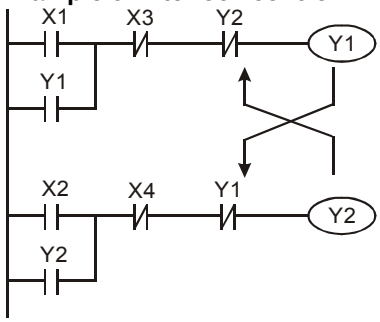
Example 5 - Conditional Control

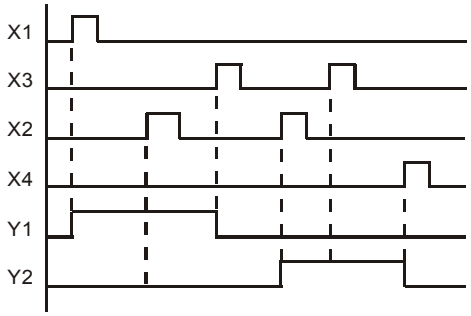




Because NO contact Y1 is connected to the circuit of Y2 output, Y1 becomes one of the conditions for enabling Y2, i.e. for turning on Y2, Y1 must be ON

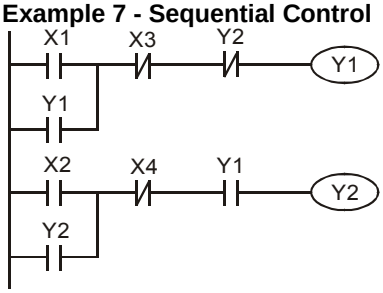
Example 6- Interlock control





NC contact Y1 is connected to the Y2 output circuit and the NC contact Y2 is connected to the Y1 output circuit. If Y1 is ON, Y2 will be OFF and vice versa. This forms an Interlock circuit which prevents both outputs from being ON at the same time. Even if both X1 and X2 are ON, in this case only Y1 will be enabled.

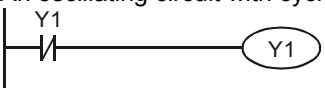
Example 7 - Sequential Control

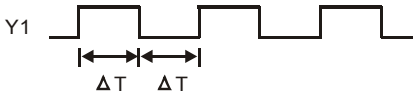


Connect the NC contact Y2 to the Y1 output circuit and the NO contact Y1 to the Y2 output circuit. Y1 becomes one of the conditions to turn on Y2. In addition, Y1 will be OFF when Y2 is ON, which forms a sequential control process.

Example 8 - Oscillating Circuit

An oscillating circuit with cycle time ΔT+ΔT

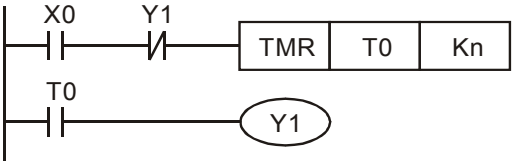


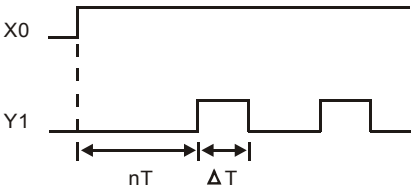


In the first scan, Y1 turns on. In the second scan, Y1 turns off due to the reversed state of contact Y1. Y1 output status changes in every scan and forms an oscillating circuit with output cycleΔT(ON)+ΔT(OFF)

Example 9 – Oscillating Circuit with Timer

An oscillating circuit with cycle nT+△T

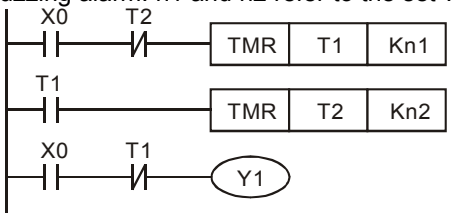


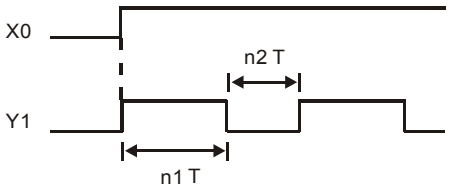


When X0 = ON, T0 starts timing (nT). Once the set time is reached, contact T0 = ON to enable Y1(△T). In next scan, Timer T0 is reset due to the reversed status of contact Y1. Therefore contact T0 is reset and Y1 = OFF. In next scan, T0 starts timing again. The process forms an oscillating circuit with output cycle nT+△T.

Example 10 - Flashing Circuit

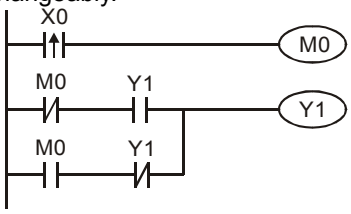
The ladder diagram uses two timers to form an oscillating circuit which enables a flashing indicator or a buzzing alarm. n1 and n2 refer to the set values in T1 and T2 and T refers to timer resolution.

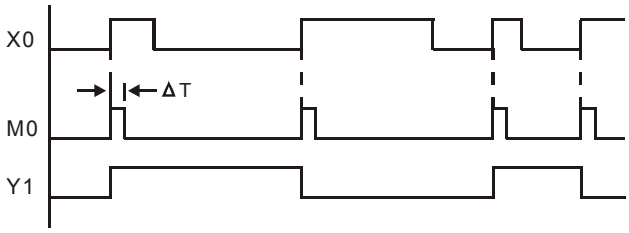




Example 11 - Trigger Circuit

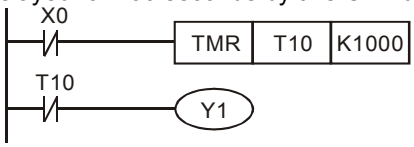
In this diagram, rising-edge contact X0 generates trigger pulses to control two actions executing interchangeably.



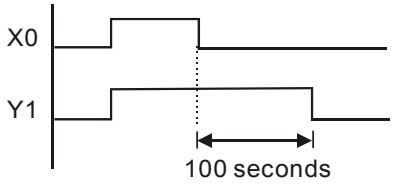


Example 12 – OFF Delay Circuit

If X0 = ON, timer T10 is not energized but coil Y1 is ON. When X0 is OFF, T10 is activated. After 100 seconds (K1000 × 0.1 sec = 100 sec), NC contact T10 is ON to turn off Y1. Turn-off action is delayed for 100 seconds by this OFF delay circuit.

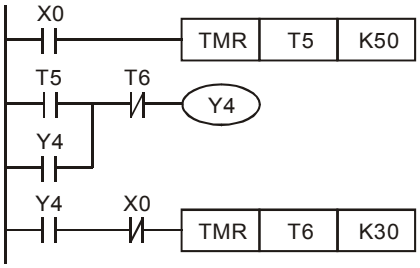


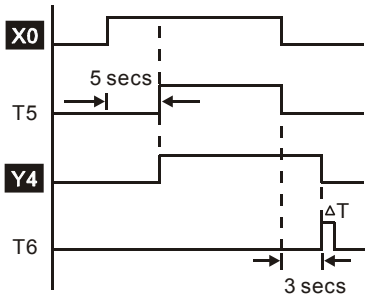
Timer Resolution: 0.1 sec



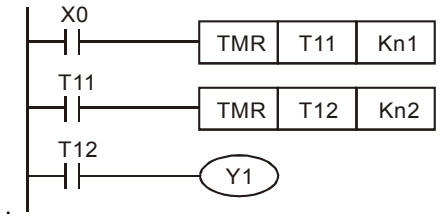
Example 13 - Output delay circuit

The output delay circuit is composed of two timers. If input X0 is ON or OFF, output Y4 will be delayed.



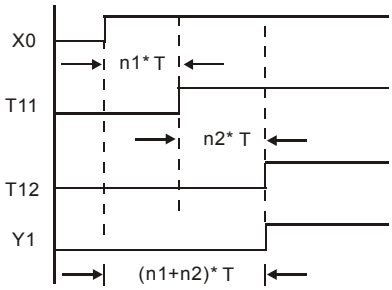


Example 14 -_Timing extension circuit

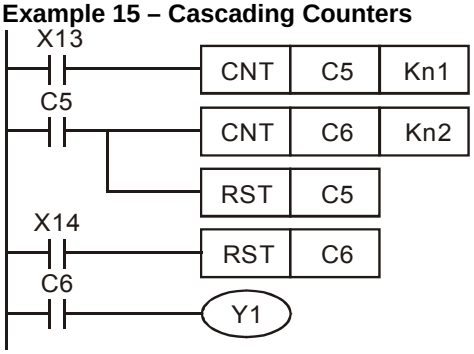


Timer = T11, T12
Timer resolution: T

The total delay time: (n1+n2)* T. T refers to the timer resolution.



Example 15 – Cascading Counters

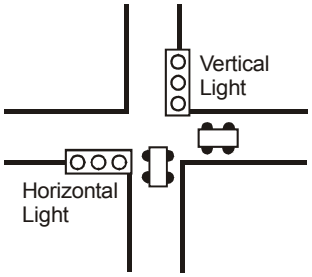


The counting range of a 16-bit counter is 0 ~ 32,767. The circuit on the left uses two counters to increase the counting range to n1*n2. When the value in counter C6 reaches n2, The pulses counted from X13 will be n1*n2.

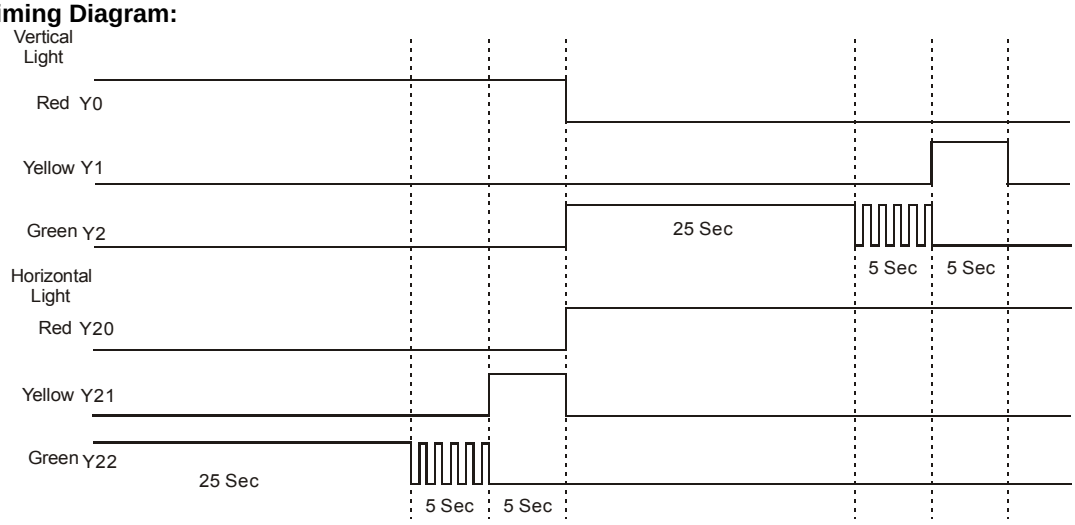
Example 16 - Traffic light control (Step Ladder Logic)

Traffic light control

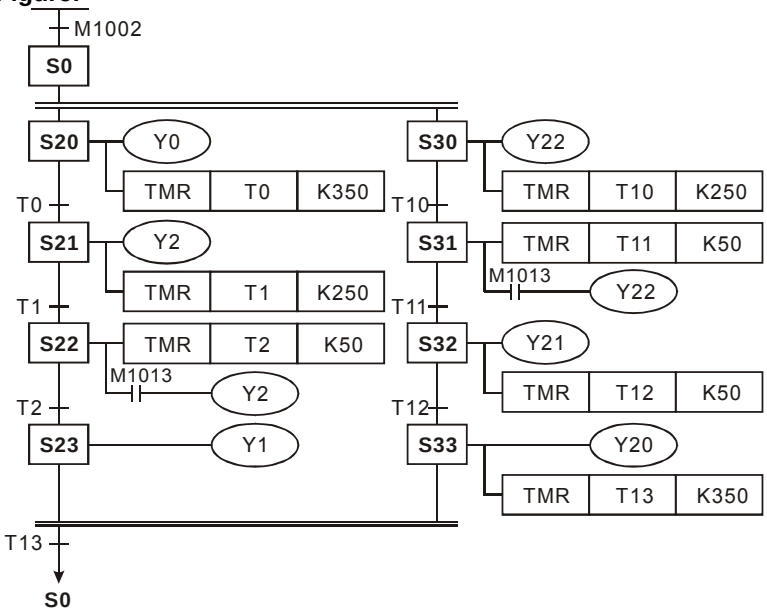
	Red light	Yellow light	Green light	Green light blinking
Vertical light	Y0	Y1	Y2	Y2
Horizontal light	Y20	Y21	Y22	Y22
Light Time	35 Sec	5 Sec	25 Sec	5 Sec



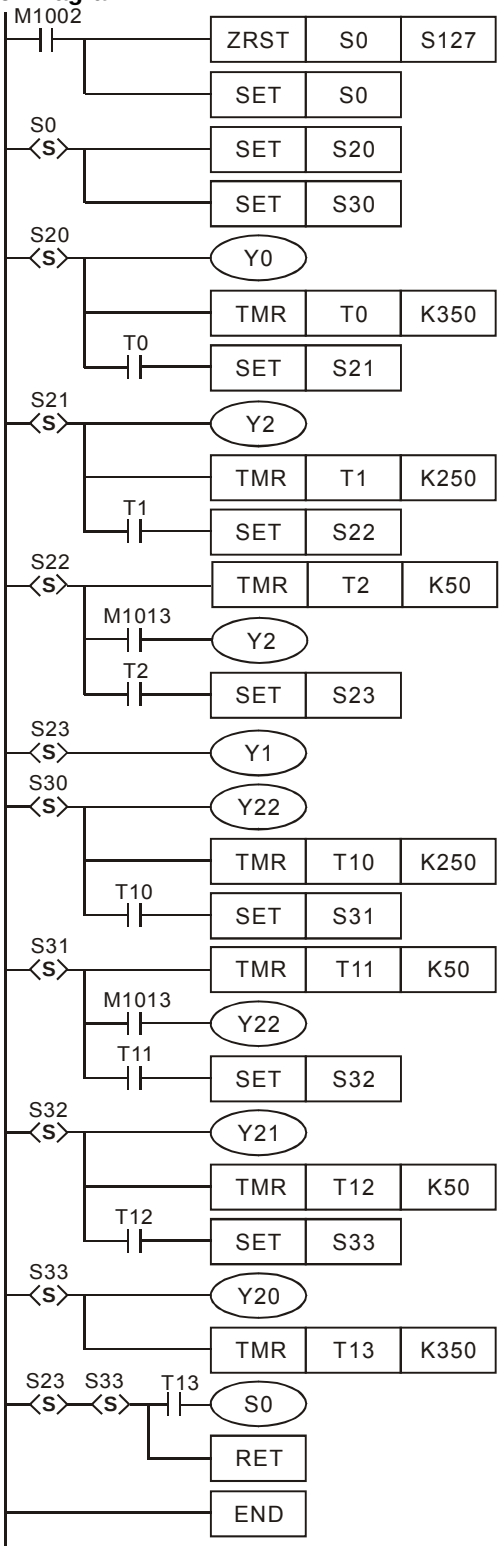
Timing Diagram:



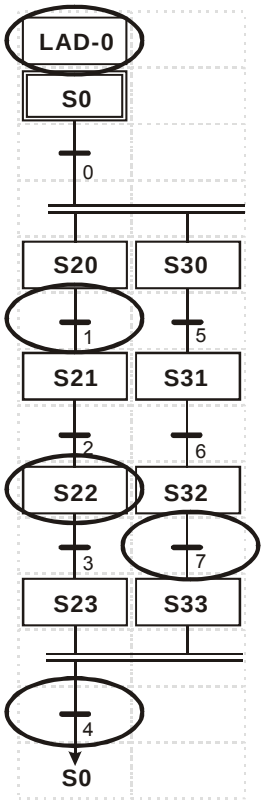
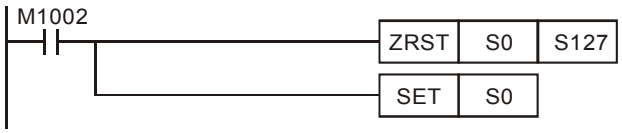
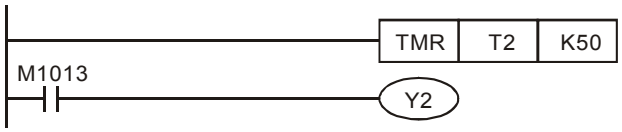
SFC Figure:



Ladder Diagram:



ELCSoft programming (SFC mode)

SFC logic	Internal Ladder Logic
	**LAD-0**  **Transfer condition 1**  **S22**  **Transfer condition 4**  **Transfer condition 7** 

Programming Concepts

The Eaton Logic Controller (ELC) is a programmable logic controller spanning an I/O range of 10 to 256 I/O points. ELC processors are so versatile they range from nano to small application size without ever needing to change processors. The ELC can control a wide variety of devices to solve your automation needs. The ELC monitors inputs and manipulate outputs as controlled by the user program. The user program provides features like boolean logic, counting, timing, complex math operations, and communication with other products.

This Chapter Contains

2 Programming Concepts

2.1 ELC Memory Map for ELCB-PB controllers

Items				Specifications		Remarks
Control Method				Stored program, cyclic scan system		
I/O Processing Method				Batch processing method (when END instruction is executed)		Fast I/O refresh instruction can override batch update
Execution Speed				Basic instructions – 3.5μ seconds minimum		Application instructions varies per instruction
Program language				Instructions + Ladder Logic + SFC		
Program Capacity				3792 Steps		Built-in EEPROM
Instructions				32 Basic instructions, Application instructions: 109		
Bit Contacts	X	External inputs		X0~X177, octal number system, 128 points max.	Total 256 I/O	Physical input points
	Y	External outputs		Y0~Y177, octal number system, 128 points max.		Physical output points
	M	Auxiliary relay	General	M0~M511, M768~M999 744 points Note 1	Total 1280 bits	Main internal relay area for general use.
			Latched	M512~M767, 256 points Note 3		
			Special	M1000~M1279, 280 points, some are latched		
	T	Timer	100ms	T0~T63, 64 points	Total 128 bits	Contact = ON when timer reaches preset value.
			10ms (M1028=ON)	T64~T126, 63 points		
			1ms	T127 1 points		
	C	Counter	16-bit count up	C0~C111, Note 1	Total 141 bits	Contact = ON when counter reaches preset value.
				C112~C127, Note 3		
			32bit high-speed count up/down	C235~C238, C241, C242, C244, 1 phase 1 input, 7 points Note 4		
				C246, C247, C249, 1 phase 2 input, 3 points Note 4		
				C251, C252, C254, 2 phase 2 input, 3 points Note 4		
	S	Step point	Initial step point	S0~S9, 10 points, Note 4	Total 128 bits	SFC usage S10~S19 is used with IST instruction
			Zero return	S10~S19, 10 points, Note 4		
			Latched	S20~S127, Note 4		

Items				Specifications		Remarks
Word Register	T	Current value		T0~T127, 128 words		
	C	Current value		C0~C127, 16-bit counter,		
				C235~C254, 32-bit counter		
	D	Data register	General	D0~D407, Note 1	Total 912 words	General storage for word length data.
			Latched	D408~D599, Note 3		
			Special	D1000~D1311, 312 words		
			Index	E=D1028, F=D1029, Note 1		
Pointer	N	Master control loop		N0~N7, 8 points		
	P	Pointer		P0~P63, 64 points		Subroutines pointer
	I	Interrupt Service	External interrupt	I001 (X0), I101 (X1), I201 (X2), I301 (X3); 4 points (all are rising-edge trigger)		Address for interrupt subroutines
			Time interrupt	I610~I699, 1 points (Timer resolution = 1ms)		
			Communication	I150, 1 point		
Constant	K	Decimal		K-32,768 ~ K32,767 (16-bit operation) K-2,147,483,648 ~ K2,147,483,647 (32-bit operation)		
	H	Hexadecimal		H0000 ~ HFFFF (16-bit operation), H00000000 ~ HFFFFFFFF (32-bit operation)		
Serial ports				COM1: RS-232 (Slave), COM2: RS-485 (Master/Slave), Both can be used at the same time.		
Clock/Calendar (RTC)				None		
Special Expansion Modules				Attach up to 8 modules of any type analog I/O expansion modules		

Notes:
1. Data area is non-latched.
2. Default is non-latched, optionally can be set to latched.
3. Default is latched, optionally can be set to non-latched.
4. Data area is latched.

2.2 ELC Memory Map for ELC-PA controllers

Items				Specifications		Remarks
Control Method				Stored program, cyclic scan system		
I/O Processing Method				Batch processing method (when END instruction is executed)		Fast I/O refresh instruction can override batch update
Execution Speed				Basic instructions – 2.5μ seconds minimum		Application instructions varies per instruction
Program language				Instructions + Ladder Logic + SFC		
Program Capacity				7920 STEPS		SRAM + Battery
Instructions				32 Basic instructions		178 Application instructions
Bit Contacts	X	External inputs		X0~X177, octal number system, 128 points max.	Total 256 I/O	Physical input points
	Y	External outputs		Y0~Y177, octal number system, 128 points max.		Physical output points
	M	Auxiliary relay	General	M0~M511, Note 1	Total 4096 bits	Main internal relay area for general use.
			Latched	M512~M999, Note 3		
				M2000~M4095, Note 3		
			Special	M1000~M1999 some are latched		
	T	Timer	100ms	T0~T199, Note 1	Total 256 bits	Contact = ON when timer reaches preset value.
				T192~T199 for Subroutine		
				T250~T255(accumulative), 6 points Note 4		
			10ms	T200~T239, Note 2		
				T240~T245(accumulative), 6 points, Note 4		
			1ms	T246~T249(accumulative), 4 points, Note 4		
	C	Counter	16-bit count up	C0~C95, Note 1	Total 235 bits	Contact = ON when counter reaches preset value.
				C96~C199, Note 3		
			32-bit count up/down	C200~C215, Note 1		
				C216~C234, Note 3		
			32bit high-speed count up/down	C235~C244, 1 phase 1 input, 9 points, Note 3	Total 16 bits	
				C246, C247, C249, 1 phase 2 input, 3 points, Note 1		
				C251, C252, C253, C254, 2 phase 2 input, 4 points, Note 3		

Items				Specifications		Remarks
Bit Contacts	S	Step point	Initial step point	S0~S9, 10 points Note 1	Total 1024 bits	Sequential Function Chart (SFC) usage
			Zero point return	S10~S19, 10 points (use with IST instruction) Note 1		
			General	S20~S511, 492 points Note 1		
			Latched	S512~S895, 384 points Note 3		
			Alarm	S896~S1023, 128 points Note 3		
Word Register	T	Current value		T0~T255, 256 words		
	C	Current value		C0~C199, 16-bit counter, 200 words		
				C200~C254, 32-bit counter		
	D	Data register	General	D0~D199, Note 1	Total 5000 words	General storage for word length data
			Latched	D200~D999, Note 3		
				D2000~D4999, Note 3		
			Special	D1000~D1999, 1000 words		
			Index	E0~E3, F0~F3, Note 1		
	None	File register		0~1599, 1600 words Note 4		Additional storage area to be used
Pointer	N	Master control loop		N0~N7, 8 points		Master control nested loop
	P	Pointer		P0~P255, 256 points		Subroutine pointer
	I	Interrupt Service	External interrupt	I001 (X0), I101 (X1), I201 (X2), I301 (X3), I401 (X4), I501 (X5); 6 points (all are rising-edge trigger)		Address for interrupt subroutines
			Time interrupt	I601~I699, I701~I799, 2 points (Timer resolution = 1ms)		
			Hi-speed counter	I010, I020, I030, I040, I050, I060; 6 points		
			Communication	I150, 1 point		

Items			Specifications	Remarks
Constant	K	Decimal	K-32,768 ~ K32,767 (16-bit operation), K-2,147,483,648 ~ K2,147,483,647 (32-bit operation)	
	H	Hexadecimal	H0000 ~ HFFFF (16-bit operation), H00000000 ~ HFFFFFFFF (32-bit operation)	
Serial ports			COM1: RS-232 (Slave), COM2: RS-485 (Master/Slave) Both can be used at the same time. COM1 is typically the programming port.	
Clock/Calendar (RTC)			Year, Month, Day, Week, Hours, Minutes, Seconds	Keep by battery
Special Expansion Modules			Attach up to 8 modules of any type analog I/O expansion modules	

Notes:
1. Data area is non-latched.
2. Default is non-latched, optionally can be set to latched.
3. Default is latched, optionally can be set to non-latched.
4. Data area is latched.

2.3 ELC Memory Map for ELC-PV and ELC2-PV controllers

Items				Specifications		Remarks
Control Method				Stored program, cyclic scan system		
I/O Processing Method				Batch processing method (when END instruction is executed)		Fast I/O refresh instruction can override batch update
Execution Speed				Basic instructions – 0.24μ seconds minimum		Application instructions varies per instruction
Program language				Instructions + Ladder Logic + SFC		
Program Capacity	ELC-PV			15872 STEPS		SRAM + Battery
	ELC2-PV			30000 STEPS		Flash ROM
Instructions				32 Basic instructions		197 Application instructions
Bit Contacts	X	External inputs		X0~X377, octal number system, 256 points max.	Total 512 I/O	Physical input points
	Y	External outputs		Y0~Y377, octal number system, 256 points max.		Physical output points
	M	Auxiliary relay	General	M0~M511, Note 2	Total 4096 bits	Main internal relay area for general use.
			Latched	M512~M999, Note 3		
				M2000~M4095, Note 3		
			Special	M1000~M1999 some are latched		
	T	Timer	100ms	T0~T199, Note 2	Total 256 bits	Contact = ON when timer reaches preset value.
				T192~T199 for Subroutine		
				T250~T255(accumulative), 6 points Note 4		
			10ms	T200~T239, Note 2		
				T240~T245(accumulative), 6 points, Note 4		
			1ms	T246~T249(accumulative), 4 points, Note 4		
	C	Counter	16-bit count up	C0~C99, Note 2	Total 235 bits	Contact = ON when counter reaches preset value.
				C100~C199, Note 3		
			32-bit count up/down	C200~C219, Note 2		
				C220~C234, Note 3		
			32bit high-speed count up/down	C235~C244, 1 phase 1 input, 10 points, Note 3	Total 18 bits	
				C246~C249, 1 phase 2 input, 4 points, Note 3		
				C251~C254, 2 phase 2 input, 4 points, Note 3		

Items				Specifications		Remarks
Bit Contacts	S	Step point	Initial step point	S0~S9, 10 points Note 2	Total 1024 bits	Sequential Function Chart (SFC) usage
			Zero point return	S10~S19, 10 points (use with IST instruction) Note 2		
			General	S20~S499, 480 points Note 2		
			Latched	S500~S899, 400 points Note 3		
			Alarm	S900~S1023, 124 points Note 3		
Word Register	T	Current value		T0~T255, 256 words		
	C	Current value		C0~C199, 16-bit counter, 200 words		
				C200~C254, 32-bit counter, 53 words		
	D	Data register	General	D0~D199, Note 2	Total 10000 words	For ELC-PV, General storage for word length data
			Latched	D200~D999, Note 3		
				D2000~D9999, Note 3		
			Special	D1000~D1999, 1000 words		
			Index	E0~E7, F0~F7, Note 1		
	D	Data register	General	D0~D199, Note 2	Total 12000 words	For ELC2-PV, General storage for word length data
			Latched	D200~D999, Note 3		
				D2000~D11999, Note 3		
			Special	D1000~D1999, 1000 words		
			Index	E0~E7, F0~F7, Note 1		
	None	File register		0~9999, 10000 words, Note 4		Additional storage area to be used
Pointer	N	Master control loop		N0~N7, 8 points		Master control nested loop
	P	Pointer		P0~P255, 256 points		Subroutine pointer
	I	Interrupt Service	External interrupt	I000/I001(X0), I100/I101 (X1), I200/I201 (X2), I300/I301 (X3), I400/I401 (X4), I500/I501 (X5), 6 points (01, rising-edge trigger ⌐, 00, falling-edge trigger ¬)		Address for interrupt subroutines
			Time interrupt	I601~I699, I701~I799, 2 points (Timer resolution = 1ms) I801~I899, 1 points (Timer resolution = 0.1ms)		
			Interruption inserted when high-speed counter reaches target	I010, I020, I030, I040, I050, I060, 6 points		
			Pulse interruption	I110, I120, I130, I140, 4 points		
			Hi-speed counter	I010, I020, I030, I040, I050, I060; 6 points		
			Communication	I150, I160, I170, 3 points		

Items			Specifications	Remarks
Constant	K	Decimal	K-32,768 ~ K32,767 (16-bit operation), K-2,147,483,648 ~ K2,147,483,647 (32-bit operation)	
	H	Hexadecimal	H0000 ~ HFFFF (16-bit operation), H00000000 ~ HFFFFFFFF (32-bit operation)	
Serial ports			COM1: RS-232 (Slave), COM2: RS-485 (Master/Slave) Both can be used at the same time. COM1 is typically the programming port.	

Clock/Calendar (RTC)	Year, Month, Day, Week, Hours, Minutes, Seconds	Keep by battery
Analog Volume dial	2	
Special Expansion Modules	Attach up to 8 modules of any type analog I/O expansion modules	
High Speed Expansion Modules	Left side expansion port: Attach up to 8 high speed modules	

Notes:

1. Data area is non-latched.
2. Default is non-latched, optionally can be set to latched.
3. Default is latched, optionally can be set to non-latched.

2.4 ELC Memory Map for ELCM-PH/PA controllers

Items				Specifications		Remarks
Control Method				Stored program, cyclic scan system		
I/O Processing Method				Batch processing method (when END instruction is executed)		Immediate I/O refresh instruction can override batch update
Execution Speed				Basic instructions – 0.54μs		MOV instruction – 3.4μs
Program language				Instructions + Ladder Logic + SFC		
Program Capacity				15872 STEPS		Flash-ROM
Instructions				32 Basic instructions		200 Application instructions
Bit Contacts	X	External inputs		X0~X377, octal number system, 256 points max, Note 4	Total 256+ 16 I/O	Physical input points
	Y	External outputs		Y0~Y377, octal number system, 256 points max, Note 4		Physical output points
	M	Auxiliary relay	General	M0~M511, 512 points, Note 1 M768~M999, 232 points, Note 1 M2000~M2047, 48 points, Note 1	Total 4096 points	Main internal relay area for general use.
			Latched	M512~M767, 256 points, Note 2 M2048~M4095, 2048 points, Note 2		
			Special	M1000~M1999, 1000 points, some are latched		
	T	Timer	100ms (M1028=ON, T64~T126: 10ms)	T0~T126, 127 points, Note 1 T128~T183, Note 1	Total 256 points	Contact = ON when timer reaches preset value.
				T184~T199 for Subroutines, 16 points, Note 1		
				T250~T255(accumulative), 6 points Note 1		
			10ms (M1038=ON, T200~T245: 1ms)	T200~T239, 40 points, Note 1		
				T240~T245(accumulative), 6 points, Note 1		
			1ms	T127, 1 points, Note 1 T246~T249(accumulative), 4 points, Note 1		

Items					Specifications		Remarks
Bit Contacts	C	Counter	16-bit count up		C0~C111, 112 points, Note 1 C128~C199,72 points, Note 1	Total 232 points	Contact = ON when counter reaches preset value.
					C112~C127,16 points, Note 2		
			32-bit count up/down		C200~C223, 24 points, Note 1		
					C224~C231, 8 points, Note 2		
			32bit high-speed count up/down	Soft-ware	C235~C242, 1 phase 1 input, 8 points, Note 2	Total 23 points	
					C232~C234, 2 phase 2 input, 3 points, Note 2		
				Hard-ware	C243~C244, 1 phase 1 input, 2 points, Note 2		
					C245~C250, 1 phase 2 input, 6 points, Note 2		
					C251~C254 2 phase 2 input, 4 points, Note 2		
	S	Step point	Initial step point		S0~S9, 10 points, Note 2	Total 1024 points	Sequential Function Chart (SFC) usage
			Zero point return		S10~S19, 10 points (use with IST instruction), Note 2		
			Latched		S20~S127, 108 points, Note 2		
			General		S128~S911, 784 points, Note 1		
			Alarm		S912~S1023, 112 points, Note 2		
Word Register	T	Current value			T0~T255, 256 words		
	C	Current value			C0~C199, 16-bit counter, 200 words		
					C200~C254, 32-bit counter, 55 words		
	D	Data register	General		D0~D407, 408 words, Note 1 D600~D999, 400 words, Note 1 D3920~D9999, 6080 words, Note 1	Total 10000 points	General storage for word length data
			Latched		D408~D599, 192 words, Note 2 D2000~D3919, 1920 words, Note 2		
Word Register	D	Data register	Special		D1000~D1999, 1000 words, some are latched		General storage for word length data
			For AIO modules		D9900~D9999，100 words, Note 1, Note 5		
			Index		E0~E7, F0~F7, 16 words, Note 1		

Items				Specifications	Remarks
Pointer	N	Master control loop		N0~N7, 8 points	Master control nested loop
	P	Pointer		P0~P255, 256 points	The location point of CJ, CALL
	I	Interrupt Service	External interrupt	I000/I001(X0), I100/I101(X1), I200/I201(X2), I300/I301(X3), I400/I401(X4), I500/I501(X5), I600/I601(X6), I700/I701(X7), 8 points (01, rising-edge trigger ⌐↑, 00, falling-edge trigger ↓⌐)	Address for interrupt subroutines
			Timer interrupt	I602~I699, I702~I799, 2 points (Timer resolution = 1ms)	
			High-speed counter interrupt	I010, I020, I030, I040, I050, I060, I070, I080,8 points	
			Communication interrupt	I140(COM1), I150(COM2), I160(COM3), 3 points, Note 3	
Constant	K	Decimal		K-32,768 ~ K32,767 (16-bit operation), K-2,147,483,648 ~ K2,147,483,647 (32-bit operation)	
	H	Hexadecimal		H0000 ~ HFFFF (16-bit operation), H00000000 ~HFFFFFFFF (32-bit operation)	
Serial ports				COM1: built-in RS-232 ((Master/Slave), COM2: built-in RS-485 (Master/Slave), COM3: built-in RS-485 (Master/Slave), COM1 is typically the programming port.	
Clock/Calendar (RTC)				Year, Month, Day, Week, Hour, Minute, Second	Can't keep at version 1.00. Keep 1~2 week at version 2.00.
Special Modules				Up to 8 AIO modules can be connected	

Notes:

1. Data area is non-latched.
2. Data area is latched.
3. COM1: built-in RS232 port. COM2: built-in RS485 port. COM3: built-in RS485 port.
4. When input points(X) are expanded to 256 points, only 16 output points(Y) are applicable. Also, when output points(Y) are expanded to 256 points, only 16 input points(X) are applicable.
5. This area is applicable only when the MPU is connected with AIO modules. Every AIO module occupies 10 points.

2.5 ELC Memory Map for ELC2-PB controllers

Specifications						
Control Method					Stored program, cyclic scan system	
I/O Processing Method					Batch processing method (when END instruction is executed)	
Execution Speed					LD instructions – 0.54μs, MOV instructions – 3.4μs	
Program language					Instruction List + Ladder + SFC	
Program Capacity					7920 steps	
Bit Contacts	X	External inputs			X0~X377, octal number system, 256 points max.	Total 480+14 I/O(*4)
	Y	External outputs			Y0~Y377, octal number system, 256 points max.	
	M	Auxiliary relay	General		M0~M511, 512 points, (*1) M768~M999, 232 points, (*1) M2000~M2047, 48 points, (*1)	Total 4096 points
			Latched		M512~M767, 256 points, (*2) M2048~M4095, 2048 points, (*2)	
			Special		M1000~M1999, 1000 points, some are latched	
	T	Timer	100ms (M1028=ON, T64~T126: 10ms)		T0~T126, 127 points, (*1) T128~T183, 56 points, (*1)	Total 256 points
					T184~T199 for Subroutines, 16 points, (*1)	
					T250~T255(accumulative), 6 points (*1)	
			10ms (M1038=ON, T200~T245: 1ms)		T200~T239, 40 points, (*1)	
					T240~T245(accumulative), 6 points, (*1)	
			1ms		T127, 1 points, (*1) T246~T249(accumulative), 4 points, (*1)	
	C	Counter	16-bit count up		C0~C111, 112 points, (*1) C128~C199, 72 points, (*1)	Total 233 points
					C112~C127, 16 points, (*2)	
			32-bit count up/down		C200~C223, 24 points, (*1)	
					C224~C232, 9 points, (*2)	
			32bit high-speed count up/down	Soft-ware	C235~C242, 1 phase 1 input, 8 points, (*2)	Total 22 points
					C233~C234, 2 phase 2 input, 2 points, (*2)	
				Hard-ware	C243~C244, 1 phase 1 input, 2 points, (*2)	
					C245~C250, 1 phase 2 input, 6 points, (*2)	
					C251~C254 2 phase 2 input, 4 points, (*2)	
	S	Step point	Initial step point		S0~S9, 10 points, (*2)	Total 1024 points
			Zero point return		S10~S19, 10 points (use with IST instruction), (*2)	
			Latched		S20~S127, 108 points, (*2)	
			General		S128~S911, 784 points, (*1)	
			Alarm		S912~S1023, 112 points, (*2)	

Specifications					
Word Register	T	Current value		T0~T255, 256 words	
	C	Current value		C0~C199, 16-bit counter, 200 words	
				C200~C254, 32-bit counter, 55 words	
	D	Data register	General	D0~D407, 408 words, (*1) D600~D999, 400 words, (*1) D3920~D4999, 1080 words, (*1)	Total 5000 points
			Latched	D408~D599, 192 words, (*2) D2000~D3919, 1920 words, (*2)	
			Special	D1000~D1999, 1000 words, some are latched	
			Index	E0~E7, F0~F7, 16 words, (*1)	
Pointer	N	Master control loop		N0~N7, 8 points	
	P	Pointer		P0~P255, 256 points	
	I	Interrupt Service	External interrupt	I000/I001(X0), I100/I101(X1), I200/I201(X2), I300/I301(X3), I400/I401(X4), I500/I501(X5), I600/I601(X6), I700/I701(X7), 8 points (01: rising-edge trigger ┘┌, 00: falling-edge trigger ┐└)	
			Timer interrupt	I602~I699, I702~I799, 2 points (Timer resolution = 1ms)	
			High-speed counter interrupt	I010, I020, I030, I040, I050, I060, I070, I080, 8 points	
			Communication interrupt	I140(COM1), I150(COM2), 2 points, (*3)	
Constant	K	Decimal		K-32,768 ~ K32,767 (16-bit operation), K-2,147,483,648 ~ K2,147,483,647 (32-bit operation)	
	H	Hexadecimal		H0000 ~ HFFFF (16-bit operation), H00000000 ~HFFFFFFFF (32-bit operation)	
Serial ports				COM1: built-in RS-232 ((Master/Slave) COM2: built-in RS-485 (Master/Slave)	
Special I/O Modules				Up to 8 special I/O modules can be connected	

Notes:

1. Non-latched area cannot be modified.
2. Latched area cannot be modified.
3. COM1: built-in RS232 port. COM2: built-in RS485 port.
4. ELC2-PB MPU occupies 16 input points (X0~X17) and 16 output points (Y0~Y17).

2.6 ELC Memory Map for ELC2-PC/ELC2-PA controllers

Specifications						
Control Method					Stored program, cyclic scan system	
I/O Processing Method					Batch processing method (when END instruction is executed)	
Execution Speed					LD instructions – 0.54μs, MOV instructions – 3.4μs	
Program language					Instruction List + Ladder + SFC	
Program Capacity					15872 steps	
Bit Contacts	X	External inputs			X0~X377, octal number system, 256 points max.	Total 480+32 I/O(*4)
	Y	External outputs			Y0~Y377, octal number system, 256 points max.	
	M	Auxiliary relay	General		M0~M511, 512 points, (*1) M768~M999, 232 points, (*1) M2000~M2047, 48 points, (*1)	Total 4096 points
			Latched		M512~M767, 256 points, (*2) M2048~M4095, 2048 points, (*2)	
			Special		M1000~M1999, 1000 points, some are latched	
	T	Timer	100ms (M1028=ON, T64~T126: 10ms)		T0~T126, 127 points, (*1) T128~T183, 56 points, (*1)	Total 256 points
					T184~T199 for Subroutines, 16 points (*1)	
					T250~T255(accumulative), 6 points (*1)	
			10ms (M1038=ON, T200~T245: 1ms)		T200~T239, 40 points, (*1)	
					T240~T245(accumulative), 6 points, (*1)	
			1ms		T127, 1 points, (*1) T246~T249(accumulative), 4 points, (*1)	
	C	Counter	16-bit count up		C0~C111, 112 points, (*1) C128~C199, 72 points, (*1)	Total 233 points
					C112~C127, 16 points, (*2)	
			32-bit count up/down		C200~C223, 24 points, (*1)	
					C224~C232, 9 points, (*2)	
			32bit high-speed count up/down	Soft-ware	C235~C242, 1 phase 1 input, 8 points, (*2)	Total 22 points
					C233~C234, 2 phase 2 input, 2 points, (*2)	
				Hard-ware	C243~C244, 1 phase 1 input, 2 points, (*2)	
					C245~C250, 1 phase 2 input, 6 points, (*2)	
					C251~C254 2 phase 2 input, 4 points, (*2)	
	S	Step point	Initial step point		S0~S9, 10 points, (*2)	Total 1024 points
			Zero point return		S10~S19, 10 points (use with IST instruction), (*2)	
			Latched		S20~S127, 108 points, (*2)	
			General		S128~S911, 784 points, (*1)	
			Alarm		S912~S1023, 112 points, (*2)	

Specifications					
Word Register	T	Current value		T0~T255, 256 words	
	C	Current value		C0~C199, 16-bit counter, 200 words	
				C200~C254, 32-bit counter, 55 words	
	D	Data register	General	D0~D407, 408 words, (*1) D600~D999, 400 words, (*1) D3920~D9799, 5880 words, (*1)	Total 10000 points
			Latched	D408~D599, 192 words, (*2) D2000~D3919, 1920 words, (*2)	
			Special	D1000~D1999, 1000 words, some are latched	
			Righ-side special module	D9900~D9999, 100 words (*1) (*6)	
			Left-side special module	D9800~D9899, 100 words (*1) (*7)	
			Index	E0~E7, F0~F7, 16 words, (*1)	
Pointer	N	Master control loop		N0~N7, 8 points	
	P	Pointer		P0~P255, 256 points	
	I	Interrupt Service	External interrupt	I000/I001(X0), I100/I101(X1), I200/I201(X2), I300/I301(X3), I400/I401(X4), I500/I501(X5), I600/I601(X6), I700/I701(X7), 8 points (01: rising-edge trigger ⎍, 00: falling-edge trigger ⎏)	
			Timer interrupt	I602~I699, I702~I799, 2 points (Timer resolution = 1ms)	
			High-speed counter interrupt	I010, I020, I030, I040, I050, I060, I070, I080, 8 points	
			Communication interrupt	I140(COM1), I150(COM2), I160(COM3), 3 points, (*3)	
Constant	K	Decimal		K-32,768 ~ K32,767 (16-bit operation), K-2,147,483,648 ~ K2,147,483,647 (32-bit operation)	
	H	Hexadecimal		H0000 ~ HFFFF (16-bit operation), H00000000 ~HFFFFFFFF (32-bit operation)	
Serial Ports			ELC2-PC	COM1: built-in RS-232 ((Master/Slave) COM2: built-in RS-485 (Master/Slave) COM3: built-in RS-485 (Master/Slave)	
			ELC2-PA	COM1: built-in RS-232 ((Master/Slave) COM2: built-in RS-485 (Master/Slave) COM3: built-in USB (Slave)	
Real Time Clock				Year, Month, Day, Week, Hour, Minute, Second Keep 1~2 week when the 24VDC power off.	
Special I/O Modules				Right side: Up to 8 I/O modules can be connected Left side: Up to 8 high-speed I/O module can be connected	
File Register (*5)				K0~K4999, 5000 points (*2)	

Notes:

1. Non-latched area cannot be modified
2. Latched area cannot be modified
3. Please refer to the table above for more information about serial ports. SX2 does not support I160.
4. There are 8 input points (X0~X17) and 4 output points (Y0~Y3) in an ELC2-PC series MPU. An ELC2-PC series MPU occupies 16 input points (X0~X17), and 16 output points (Y0~Y17). There are 8 input points (X0~X17), and 6 output points (Y0~Y5) in an ELC2-PA series MPU. An ELC2-PA series MPU occupies 16 input points (X0~X17), and 16 output points (Y0~Y17). Extension input points start from X20, and extension output points start from Y17.

5. Please refer to the instruction MEMR/MEMW for more information about the reading/writing of data.
6. If an ELC2-PC/PA series MPU is connected to a right-side special module, and M1183 is Off, the range of data registers can be used. Every special module connected to an ELC2-PC/PA series MPU occupies ten data registers.
7. If an ELC2-PC/PA series MPU is connected to a left-side special module, and M1182 is Off, the range of data registers can be used. Every special module connected to an ELC2-PC/PA series MPU occupies ten data registers.

2.7 ELC Memory Map for ELC2-PE controllers

Specifications						
Control Method					Stored program, cyclic scan system	
I/O Processing Method					Batch processing method (when END instruction is executed)	
Execution Speed					LD instructions – 0.64μs, MOV instructions – 2μs, 1000 steps – approximately 1ms	
Program language					Instruction List + Ladder diagram+ SFC	
Program Capacity					15872 steps	
Bit Contacts	X	External inputs			X0~X377, octal number system, 256 points max.	Total 480+12 I/O(*4)
	Y	External outputs			Y0~Y377, octal number system, 256 points max.	
	M	Auxiliary relay	General		M0~M511, 512 points, (*1) M768~M999, 232 points, (*1) M2000~M2047, 48 points, (*1)	Total 4096 points
			Latched		M512~M767, 256 points, (*2) M2048~M4095, 2048 points, (*2)	
			Special		M1000~M1999, 1000 points, some are latched	
	T	Timer	100ms (M1028=ON, T64~T126: 10ms)		T0~T126, 127 points, (*1) T128~T183, 56 points, (*1)	Total 256 points
					T184~T199 for Subroutines, 16 points, (*1)	
					T250~T255(accumulative), 6 points (*1)	
			10ms (M1038=ON, T200~T245: 1ms)		T200~T239, 40 points, (*1)	
					T240~T245(accumulative), 6 points, (*1)	
			1ms		T127, 1 points, (*1) T246~T249(accumulative), 4 points, (*1)	
	C	Counter	16-bit count up		C0~C111, 112 points, (*1) C128~C199, 72 points, (*1)	Total 232 points
					C112~C127, 16 points, (*2)	
			32-bit count up/down		C200~C223, 24 points, (*1)	
					C224~C231, 8 points, (*2)	
			32bit high-speed count up/down	Soft-ware	C235~C242, 1 phase 1 input, 8 points, (*2)	Total 20 points
					C233~C234, 2 phase 2 input, 2 points, (*2)	
				Hard-ware	C243~C244, 1 phase 1 input, 2 points, (*2)	
					C245~C248, 1 phase 2 input, 4 points, (*2)	
					C251~C254 2 phase 2 input, 4 points, (*2)	

Specifications					
Bit Contacts	S	Step point	Initial step point	S0~S9, 10 points, (*2)	Total 1024 points
			Zero point return	S10~S19, 10 points (use with IST instruction), (*2)	
			Latched	S20~S127, 108 points, (*2)	
			General	S128~S911, 784 points, (*1)	
			Alarm	S912~S1023, 112 points, (*2)	
Word Register	T	Current value		T0~T255, 256 words	
	C	Current value		C0~C199, 16-bit counter, 200 words	
				C200~C254, 32-bit counter, 55 words	
	D	Data register	General	D0~D407, 408 words, (*1) D600~D999, 400 words, (*1) D3920~D9799, 5880 words, (*1) D10000~D11999, 2000 words, (*1)	Total 12000 points
			Latched	D408~D599, 192 words, (*2) D2000~D3919, 1920 words, (*2)	
			Special	D1000~D1999, 1000 words, some are latched	
			Right-side special module	D9900~D9999, 100 words, (*1) (*5)	
			Left-side special module	D9800~D9899, 100 words, (*1) (*6)	
			Index	E0~E7, F0~F7, 16 words, (*1)	
Pointer	N	Master control loop		N0~N7, 8 points	
	P	Pointer		P0~P255, 256 points	
	I	Interrupt Service	External interrupt	I000/I001(X0), I100/I101(X1), I200/I201(X2), I300/I301(X3), I400/I401(X4), I500/I501(X5), I600/I601(X6), I700/I701(X7), 8 points (01: rising-edge trigger ┘‾, 00: falling-edge trigger ‾└)	
			Timer interrupt	I602~I699, I702~I799, 2 points (Timer resolution = 1ms)	
			High-speed counter interrupt	I010, I020, I030, I040, I050, I060, I070, I080, 8 points	
			Communication interrupt	I150 (COM2), I160 (COM3), 2 points, (*3)	
Constant	K	Decimal		K-32,768 ~ K32,767 (16-bit operation), K-2,147,483,648 ~ K2,147,483,647 (32-bit operation)	
	H	Hexadecimal		H0000 ~ HFFFF (16-bit operation), H00000000 ~HFFFFFFFF (32-bit operation)	
Serial Ports				COM1: built-in USB (Slave) COM2: built-in RS-485 (Master/Slave) COM3: built-in RS-485 (Master/Slave) Ethernet: built-in Ethernet (Please refer to Appendix B for more information.) COM1 is typically the programming port.	
Real Time Clock				Year, Month, Day, Week, Hours, Minutes, Seconds Keep 1~2 week when 24VDC power off.	
Special I/O Modules				Right side: Up to 8 I/O modules can be connected Left side: Up to 8 high-speed I/O modules can be connected	

Notes:

1. Non-latched area cannot be modified.
2. Latched area cannot be modified.

3. COM2: built-in RS485 port. COM3: built-in RS485 port.
4. There are 8 input points (X0~X7) and 4 output points (Y0~Y3) in an ELC2-PE MPU. An ELC2-PE MPU occupies 16 input points (X0~X17) and 16 output points (Y0~Y17). Extension input points start from X20, and output points start from Y20.
5. If an ELC2-PE series MPU is connected to a right-side special module, and M1183 is Off, the range of data registers can be used. Every special module connected to an ELC2-PE series MPU occupies ten data registers.
6. If an ELC2-PE series MPU is connected to a left-side special module, and M1182 is Off, the range of data registers can be used. Every special module connected to an ELC2-PE series MPU occupies ten data registers.

2.8 ELC Latched Memory Settings

ELC-PA Controllers

M Auxiliary relay	General	Latched	Special auxiliary relay	Latched
	M0~M511	M512~M999	M1000~M1999	M2000~M4095
	Not latched	Latched (default) Start: D1200 (K512) End: D1201 (K999)	Some are latched and they cannot be changed.	Latched (default) Start: D1202 (K2000) End: D1203 (K4095)

T Timer	100 ms	10 ms	10ms	1 ms	100 ms
	T0 ~T199	T200~T239	T240~T245	T246~T249	T250~T255
	non-latched	non-latched	Accumulative latched		

C Counter	16-bit count up		32-bit count up/down		32-bit high-speed count up/down
	C0~C95	C96~C199	C200~C215	C216~C234	C235~C255
	Non-latched	Latched (default) Start: D1208 (K96) End: D1209 (K199)	Non-latched	Latched (default) Start: D1210 (K216) End: D1211 (K234)	Latched (default) Start: D1212 (K235) End: D1213 (K255)

S Step relay	Initial	Zero return	General	Latched	Step alarm
	S0~S9	S10~S19	S20~S511	S512~S895	S896~S1023
	Non-latched			Latched (default) Start: D1214 (K512) End: D1215 (K895)	Latched

D Register	General	Latched	Special register	Latched
	D0~D199	D200~D999	D1000~D1999	D2000~D4999
	Non-latched	Latched (default)	Some are latched,	Latched (default)

		Start: D1216 (K200) End: D1217 (K999)	and cannot be changed	Start: D1218 (K2000) End: D1219 (K4999)

File Register	K0-1599
	Latched

ELC-PV Controllers

M Auxiliary relay	General	Latched	Special auxiliary relay	Latched
	M0~M499	M500~M999	M1000~M1999	M2000~M4095
	Non-latched (default)	Latched (default)	Some are latched and they cannot be changed.	Latched (default)
	Start: D1200 (K500) End: D1201 (K999)			Start: D1202 (K2000) End: D1203 (K4095)

T Timer	100 ms	10 ms	10ms	1 ms	100 ms
	T0 ~T199	T200~T239	T240~T245	T246~T249	T250~T255
	Non-latched (default)	Latched (default)	Accumulative latched		
	Start: D1204 (K-1)*1 End: D1205 (K-1)*1	Start: D1206 (K-1)*1 End: D1207 (K-1)*1			

C Counter	16-bit count up		32-bit count up/down		32-bit high-speed count up/down
	C0~C99	C100~C199	C200~C219	C220~C234	C235~C255
	Non-latched (default)	Latched (default)	Non-latched (default)	Latched (default)	Latched (default)
	Start: D1208 (K96) End: D1209 (K199)		Start: D1210 (K216) End: D1211 (K234)		Start: D1212 (K235) End: D1213 (K255)

S Step relay	General	Latched	Special	Latched	General
	S0~S9	S10~S19	S20~S499	S500~S899	S900~S1023
	Non-latched (default)			Latched (default)	Latched
	Start: D1214 (K500) End: D1215 (K999)				

D Register	General	Latched	Special register	Latched
	D0~D199	D200~D999	D1000~D1999	D2000~D9999
	Non-latched (default)	Latched (default)	Some is latched, and cannot be changed	Latched (default)
	Start: D1216 (K200) End: D1217 (K999)			Start: D1218 (K2000) End: D1219 (K9999)

File Register	K0-9999
	Latched

*1: K-1 refers to the default setting is non-latched.

ELC2-PV Controllers

M (Auxiliary relay)	General purpose	Latched	Special auxiliary relay	Latched
	M0 ~ M499	M500 ~ M999	M1000 ~ M1999	M2000 ~ M4095
	Start: D1200 (K500) End: D1201 (K999)	Some are latched and cannot be modified.	Start: D1202 (K2,000) End: D1203 (K4,095)	

T (Timer)	100 ms	10 ms	10 ms	1 ms	100 ms
	T0 ~ T199	T200 ~ T239	T240 ~ T245	T246 ~ T249	T250 ~ T255
	Default: non-latched	Default: non-latched	Accumulative type It is fixed to be latched.		
	Start: D1204 (K-1)*1 End: D1205 (K-1)*1	Start: D1206 (K-1)*1 End: D1207 (K-1)*1			

C (Counter)	16-bit counting up		32-bit counting up/down		32-bit high-speed counting up/down	
	C0 ~ C99	C100 ~ C199	C200 ~ C219	C220 ~ C234	C235 ~ C245	C246 ~ C255
	Default: non-latched	Default: latched	Default: non-latched	Default: latched	Default: latched	
	Start: D1208 (K100) End: D1209 (K199)		Start: D1210 (K220) End: D1211 (K234)		Start: D1212 (K235) End: D1213 (K255)	

S (Step relay)	Initial	Zero return	General purpose	Latched	Step alarm
	S0 ~ S9	S10 ~ S19	S20 ~ S499	S500 ~ S899	S900 ~ S1023
	Non-latched (default)			Latched (default)	It is fixed to be latched.
	Start: D1214 (K500) End: D1215 (K899)				

D (Register)	General purpose	Latched	Special register	Latched
	D0 ~ D199	D200 ~ D999	D1000 ~ D1999	D2000 ~ D11999
	Default: non-latched	Default: latched	Some is latched and cannot be modified.	Default: latched
	Start: D1216 (K200) End: D1217 (K999)			Start: D1218 (K2,000) End: D1219 (K9,999)

File register	K0 ~ K9,999
	It is fixed to be latched.

ELCM-PH/ELCM-PA/ELC2-PB/ELC2-PC/ELC2-PA/ELC2-PE Controllers

M Auxiliary relay	General	Latched	Special auxiliary relay
	M0~M511 M768~M999 M2000~M2047	M512~M999 M2048~M4095	M1000~M1999
	Not latched	Latched	Some are latched and cannot be changed.

T Timer	100 ms	100 ms	1 ms	10 ms	10ms	1 ms	100 ms
	T0 ~T126 T128~T183	T184~T199	T127	T200~T239	T240~T245	T246~T249	T250~T255
	M1028=1,T64 ~T126:10ms	For subroutine	-	M1038=1,T200~T245: 1ms		-	
	non-latched	non-latched			Accumulative non-latched		

C Counter	16-bit count up		32-bit count up/down		32-bit high-speed count up/down
	C0~C111 C128~C199	C112~C127	C200~C223	C224~C232	C233~C254
	Non-latched	Latched	Non-latched	Latched	Latched

S Step relay	Initial	Zero return	General	Latched	Step alarm
	S0~S9	S10~S19	S20~S127	S128~S911	S912~S1023
	Latched			Non-latched	Latched

D Register	General	Latched	Special register	For AIO
	D0~D407 D600~D999 D3920~D11999	D408~D599 D2000~D3919	D1000~D1999	D9800~D9999
	Non-latched	Latched	Some are latched, and cannot be changed	Non-latched

2.9 ELC Latched Memory Modes

ELCB-PB Controllers

Memory type	Power OFF=>ON	STOP=>RUN	RUN=>STOP	Clear all M1031 Non-latched area	Clear all M1032 latched area	Factory setting
Non-latched	Clear	When M1033=OFF, clear When M1033=ON, No change		Clear	Unchanged	0
Latched	Unchanged			Unchanged	Clear	Unchanged
Special M, Special D, Index Register	Initial	Unchanged		Unchanged		Initial setting

ELC-PA/ PV, ELC2-PV, ELCM-PH/ PA, ELC2- PB/PH/PA/PE Controllers

Memory type	Power OFF=>ON	STOP=>RUN	RUN=>STOP	Clear all M1031 Non-latched area	Clear all M1032 latched area	Factory setting
Non-latched	Clear	Unchanged	When M1033=OFF, clear When M1033=ON, No change	Clear	Unchanged	0
Latched	Unchanged			Unchanged	Clear	0
Special M, Special D, Index Register	Initial	Unchanged		Unchanged		Initial setting
File Register	Unchanged					0/HFFFF*

*: For ELC-PA/PV and ELC2-PV, it is K0. For ELCM-PH/PA and ELC2-PB/PH/PA/PE, it is HFFFF.
*: In ELCM-PH/PA, only ELCM-PH/PAV2.0 (and above) has file registers.

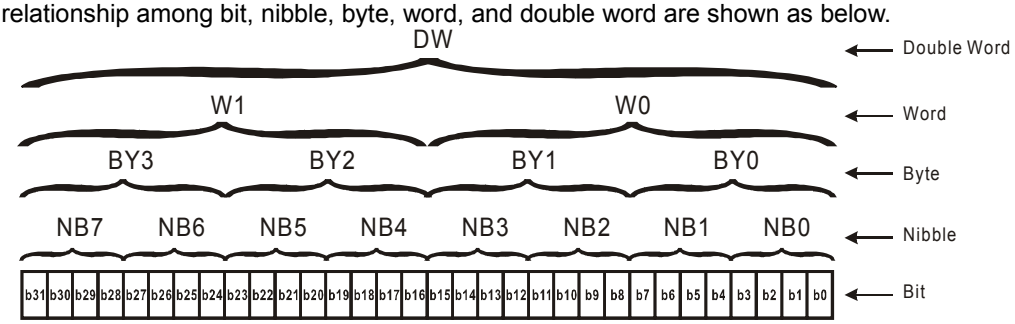
2.10 ELC Bits, Nibbles, Bytes, Words, etc

ELC controllers utilize five numeric types to perform different instructions. The following is the explanation of numeric types.

Numeric	Description
Bit	Bit is the basic unit of a binary number system. Range is 0 or 1
Nibble	4 consecutive bits, such as b3~b0. Range 0 – 9 (BCD) or 0~F hexadecimal.
Byte	8 consecutive bits b7~b0 Range 0 – 255 or 00 - FF hexadecimal
Word	16 consecutive bits (2 consecutive bytes) b15~b0 Range -32,768 ~ 32,767 or 0000 ~ FFFF hexadecimal

Numeric	Description
Double Word	32 consecutive bits (2 consecutive words) b31~b0 Range -2,147,483,648 ~ 2,147,483,647 or 00000000 - FFFFFFFF hexadecimal

The relationship among bit, nibble, byte, word, and double word are shown as below.



2.11 Binary, Octal, Decimal, BCD, Hex

ELC is capable of using many different numbering systems.

1. Binary Number, (BIN)
   ELC internally calculates, operates, and stores the value in Binary format.
2. Octal Number, (OCT)
   The external I/O points of the ELC are numbered in octal format.

   e.g.
   External inputs: X0～X7, X10～X17, …, X377. (No. of input)
   External outputs: Y0～Y7, Y10～Y17, …, Y377. (No. of output)
3. Decimal Number, (DEC)
   ELC applies decimal operation in situations below:
   - Set value for timers and counters, e.g. TMR C0 K50. (K value)
   - No. of S, M, T, C, D, E, F, P, I, e.g. M10, T30. (No. of device)
   - For use of operand in API instructions, e.g. MOV K123 D0. (K value)
   - Constant K:

   Decimal value in ELC operation is attached with a “K”, e.g. K100 indicates the value 100 in Decimal format.

   > Exception:
   > When a constant K is used with bit devices X, Y, M, S, the value specified after K indicates the groups of 4-bit units, which forms a digit(4-bit), byte(8 bit), word(16bit), or double word(32-bit) data, e.g. K2Y10, K4M100, representing Y10 ~ Y17 and M100~M115.
4. BCD (Binary Coded Decimal)
   BCD format uses 1 Decimal digit to represent a 4 bit value, so that 16 consecutive data bits can be represented by a 4-digit decimal value. Used mainly for reading values from DIP switches or sending data to 7-segement displays
5. Hexadecimal Number, HEX
   ELC applies Hexadecimal operation in situations below:
   - For use of operand in API instructions, e.g. MOV H1A2B D0 ◦ (H value)
   - Constant H:

   Hexadecimal value in ELC operation is attached with an “H”, e.g. H100 indicates the value 100 in Hex format.

Reference Table:

Binary (BIN)	Octal (OCT)	Decimal (K) (DEC)	BCD (Binary Code Decimal)	Hexadecimal (H) (HEX)

For ELC internal operation	No. of X, Y relay	Constant K, No. of registers M, S, T, C, D, E, F, P, I	For DIP Switch and 7-segment display	Constant H
0000	0	0	0000	0
0001	1	1	0001	1
0010	2	2	0010	2
0011	3	3	0011	3
0100	4	4	0100	4
0101	5	5	0101	5
0110	6	6	0110	6
0111	7	7	0111	7
1000	10	8	1000	8
1001	11	9	1001	9
1010	12	10	0000	A
1011	13	11	0001	B
1100	14	12	0010	C
1101	15	13	0011	D
1110	16	14	0100	E
1111	17	15	0101	F
10000	20	16	0110	10
10001	21	17	0111	11

2.12 Special M Relay

The special auxiliary relay (special M) are as shown in the following. Please notice that some SM will be different to the different Controller. In the following chart, the values in the “Type” column are: “R”: can only read. “R/W”: can read/write. A “-“means it can do nothing.

For ELCB-PB, ELC-PA, ELC-PV, and ELC2-PV:

SM	Function	ELCB -PB	ELC -PA	ELC -PV	ELC2 -PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1000	Normally open contact (a contact). This contact is ON when running and it is ON when the status is set to RUN.	Y	Y	Y	Y	OFF	ON	OFF	R	NO	OFF
M1001	Normally OFF contact (b contact). This contact is OFF in running and it is OFF when the status is set to RUN.	Y	Y	Y	Y	ON	OFF	ON	R	NO	ON
M1002	ON only for 1 scan after RUN. Initial pulse is contact a. It will get positive pulse in the RUN moment. Pulse width=scan period.	Y	Y	Y	Y	OFF	ON	OFF	R	NO	OFF
M1003	OFF only for 1 scan after RUN. Initial pulse is contact a. It will get negative pulse in the RUN moment. Pulse width=scan period.	Y	Y	Y	Y	ON	OFF	ON	R	NO	ON
M1004	ON when error occurs	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1008	Monitor timer flag (ON: ELC WDT time out)	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1009	History of LV signal due to 24VDC insufficiency	Y	Y	-		OFF	-	-	R	NO	OFF
M1010	PLSY Y0 mode selection. ON = continuous output	Y	Y	-	-	OFF	-	-	R/W	NO	OFF
	Pulse output when reaching END instruction	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1011	10ms clock pulse, 5ms ON/5ms OFF	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1012	100ms clock pulse, 50ms ON / 50ms OFF	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1013	1s clock pulse, 0.5s ON / 0.5s OFF	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1014	1min clock pulse, 30s ON / 30s OFF	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1015	High-speed connection counter	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1016	Display year bit. When OFF = display two right-most bits. When ON = display (two right-most bits + 2000).	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1017	±30 seconds adjustment	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1018	Flag for Radian/Degree, ON for degree	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1019	Enabling frequency measurement card	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1020	Zero flag	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1021	Barrow flag	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1022	Carry flag	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1023	PLSY Y1 mode selection, ON = continuous output.	Y	Y	-	-	OFF	-	-	R/W	NO	OFF
M1024	COM1 monitor request	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1025	If the ELC receives an illegal communication request when PC or HMI connects to an ELC, M1025 =ON and save the error code in D1025.	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1026	Startup flag of RAMP module	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1027	PR output flag	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1028	10ms time switch flag. The base setting flag of T64~T126 is 100ms, when timer is OFF and the base setting flag is 10ms when it is ON.	Y	-	-	-	OFF	-	-	R/W	NO	OFF
M1029	Pulse output Y0 of PLSY and PLSR instruction execution completed or other relative instruction execution completed	Y	Y	-	-	OFF	-	-	R	NO	OFF
	The 1st group pulse output CH0 (Y0, Y1) is completed, or other relevant instructions complete their executions.	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1030	Pulse output Y1 of PLSY and PLSR instruction execution completed	Y	Y	-	-	OFF	-	-	R	NO	OFF
	The 2nd group pulse output CH1 (Y2, Y3) is completed, or other relevant instructions complete their executions.	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1031	Clear all non-latched memory	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF

SM	Function	ELCB -PB	ELC -PA	ELC -PV	ELC2 -PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1032	Clear all latched memory	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1033	Memory latched at STOP	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1034	All Y outputs disable	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1035	Start X input point to be RUN/STOP switch and correspond to D1035 (ELC-PA controllers indicate X3 only..)	-	Y	Y	Y	-	-	-	R/W	YES	OFF
M1036	The 3rd group pulse output CH2 (Y4, Y5) is completed.	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1037	The 4th group pulse output CH3 (Y6, Y7) is completed.	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1038	OFF: The time base of T200~T255 is 10ms. ON: The time base of T200~T255 is 1ms.	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1039	Constant scan mode	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1040	Step transition inhibit	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1041	Step transition start	Y	Y	Y	Y	OFF	-	OFF	R/W	NO	OFF
M1042	Start pulse	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1043	Zero point return completed	Y	Y	Y	Y	OFF	-	OFF	R/W	NO	OFF
M1044	Zero point condition	Y	Y	Y	Y	OFF	-	OFF	R/W	NO	OFF
M1045	All outputs clear inhibit	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1046	STL state setting (ON)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1047	STL monitor enable	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1048	Flag for alarm point state	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1049	Monitor flag for alarm point	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1050	I000/I001 masked	Y	Y	-	-	OFF	-	-	R/W	NO	OFF
M1051	I100/I101 masked	Y	Y	-	-	OFF	-	-	R/W	NO	OFF
M1052	I200/I201 masked	Y	Y	-	-	OFF	-	-	R/W	NO	OFF
M1053	I300/I301 masked	Y	Y	-	-	OFF	-	-	R/W	NO	OFF
	Enabling X4 speed detection	-	-	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1054	I400/I401 masked	-	Y	-	-	OFF	-	-	R/W	NO	OFF
	Enabling X10 speed detection	-	-	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1055	I500/I501 masked	-	Y	-	-	OFF	-	-	R/W	NO	OFF
	Enabling X14 speed detection	-	-	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1056	I602~ I699 masked	Y	Y	-	-	OFF	-	-	R/W	NO	OFF
	Enabling X1 interrupt to get the counting value of C241	-	-	-	Y	OFF	OFF	OFF	R/W	NO	OFF
M1057	I702~I799 masked	-	Y	-	-	OFF	-	-	R/W	NO	OFF
	Enabling X2 interrupt to get the counting value of C241	-	-	-	Y	OFF	OFF	OFF	R/W	NO	OFF
M1058	COM3 monitor request	-	-	-	Y	OFF	-	-	R/W	NO	OFF
M1059	I010~I060 masked	-	Y	-	-	OFF	-	-	R/W	NO	OFF
	Enabling X3 interrupt to get the counting value of C241	-	-	-	Y	OFF	OFF	OFF	R/W	NO	OFF
M1060	System error message 1: The peripheral circuit of the CPU breaks down.	Y	Y	Y	-	OFF	-	-	R	NO	OFF
M1061	System error message 2: The CPU flag register breaks down.	-	Y	Y	-	OFF	-	-	R	NO	OFF
	System error message 2: An error occurs when the data in the latched area is read.	Y	-	-	-	OFF	-	-	R	NO	OFF
M1062	System error message 3: The CPU BIOS ROM breaks down.	Y	Y	Y	-	OFF	-	-	R	NO	OFF
M1063	System error message 4: The RAM in the CPU breaks down.	Y	Y	Y	-	OFF	-	-	R	NO	OFF
M1064	Operator error	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1065	Syntax error	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1066	Program error	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1067	Program execution error	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1068	Execution error latched (D1068)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1070	Y1 time base switching for PWM instruction (ON: 100us; OFF: 1ms)	Y	Y	-		OFF	-	-	R/W	NO	OFF
	Y0 time base switching for PWM instruction (ON: 100us; OFF: 1ms) when On, D1371 will decide the time base)	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1071	Y2 time base switching for PWM instruction (On: 100us; Off: 1ms) when On, D1372 will decide the time base)	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1072	Executing ELC RUN instruction	Y	Y	Y	Y	OFF	On	Off	R/W	NO	OFF

SM	Function	ELCB -PB	ELC -PA	ELC -PV	ELC2 -PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1074	SRAM access error	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1075	Error occurring when writing FLASH card or Flash ROM	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1076	Real time clock malfunction	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1077	Battery in low voltage, malfunction or no battery	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1078	Immediately stopping Y0 pulse output for PLSY instruction	Y	Y	-	-	OFF	-	-	R/W	NO	OFF
M1079	Immediately stopping Y1 pulse output for PLSY instruction	Y	Y	-	-	OFF	-	-	R/W	NO	OFF
M1080	Requesting COM2 monitoring	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1081	Changing direction for FLT instruction	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1082	Real time clock has been changed	-	Y	Y	Y	OFF	-	-	R	NO	Off
M1083	Allowing interruption subroutine in FROM/TO instructions	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1084	Detecting bandwidth	Y	Y	-	-	OFF	OFF	OFF	R/W	NO	OFF
M1085	Selecting ELC-ACPGMXFR duplicating function	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1086	Setting up the switch for enabling password function of ELC-ACPGMXFR	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1087	Enabling LV signal	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1088	Matrix comparison. Comparing between equivalent values (M1088 = 1) or different values (M1088 = 0).	-	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1089	Matrix search end flag. When the comparison reaches the last bit, M1089 = 1.	-	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1090	Matrix search start flag. Compare from the first bit and M1090=1.	-	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1091	Matrix finding bit flag. When find it, it will stop comparing and M1091=1.	-	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1092	Matrix pointer error flag. When pointer Pr exceeds this range, M1092=1.	-	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1093	Matrix pointer increase flag. It will add 1 to present pointer.	-	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1094	Matrix pointer clear flag. It will clear present pointer to 0.	-	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1095	Carry flag for matrix rotate/shift output	-	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1096	Complement flag for matrix shift input	-	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1097	Direction flag for matrix rotate/shift	-	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1098	Matrix count bit 0 or 1 flag	-	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1099	It is ON when matrix count result 0	-	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1100	SPD instruction sampling once	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1101	Start file register or not	-	Y	Y	Y	-	-	-	R/W	Yes	OFF
M1112	AY0 output point on 2DO card (transistor)	-	-	Y	-	OFF	OFF	-	R/W	NO	OFF
M1113	AY1 output point on 2DO card (transistor)	-	-	Y	-	OFF	OFF	-	R/W	NO	OFF
M1115	Start switch for accel/decel pulse output	Y	Y	-	-	OFF	OFF	OFF	R/W	NO	OFF
M1116	Acceleration flag	Y	Y	-	-	OFF	OFF	OFF	R/W	NO	OFF
M1117	Target attained frequency flag	Y	Y	-	-	OFF	OFF	OFF	R/W	NO	OFF
M1118	Deceleration flag	Y	Y	-	-	OFF	OFF	OFF	R/W	NO	OFF
M1119	Completed function flag Using the instruction DDRVI/DDRVA to enable two target frequencies.	Y -	Y -	- Y	- Y	OFF OFF	OFF OFF	OFF OFF	R/W R/W	NO NO	OFF OFF
M1120	Set COM2 (RS-485) protocol kept on. D1120 cannot be changed after the setting.	Y	Y	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1121	Waiting for the sending of COM2 (RS-485) communication data	Y	Y	Y	Y	OFF	OFF	ON	R	NO	OFF
M1122	COM2 (RS-485) sending request	Y	Y	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1123	Receiving through COM2 (RS-485) is completed	Y	Y	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1124	Waiting for receiving through COM2 (RS-485)	Y	Y	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1125	COM2 (RS-485) communication reset	Y	Y	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1126	Selecting COM2 (RS-485) STX/ETX user defined or system defined	Y	Y	Y	Y	OFF	OFF	OFF	R/W	NO	OFF

SM	Function	ELCB -PB	ELC -PA	ELC -PV	ELC2 -PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1127	Sending/receiving data of COM2 (RS-485) communication instruction is completed (RS instruction not included)	Y	Y	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1128	Sending COM2 (RS-485)/receiving COM2 (RS-485) indication	Y	Y	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1129	COM2 (RS-485) receiving time-out	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1130	Selecting COM2 (RS-485) STX/ETX user defined or system defined	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1131	On during COM2 (RS-485) MODRD/RDST/MODRW data are converted to hex data	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1132	ON= no relative communications instruction in ELC program.	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1133	Special high speed pulse (50KHz) output switch (ON = start)	-	Y	-	-	OFF	OFF	OFF	R/W	NO	OFF
M1134	ON is continuous output switch	-	Y	-	-	OFF	OFF	-	R/W	NO	OFF
M1135	Output pulse numbers attained flag	-	Y	-	-	OFF	OFF	OFF	R/W	NO	OFF
M1137	Retain the data in DNET mapping area in STOP state	-	-	Y	Y	-	-	-	R/W	NO	OFF
M1138	Set COM1 (RS-232) protocol kept on. D1036 cannot be changed after setting	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1139	COM1 (RS-232) ASCII/RTU mode (OFF:ASCII, ON:RTU)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1140	MODRD/MODWR/MODRW data received error	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1141	MODRD/MODWR/MODRW instruction error	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1143	COM2 (RS-485) ASCII/RTU mode. (OFF:ASCII, ON:RTU)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1144	Output start switch of accel/decel pulse output function of adjustable slope	-	Y	-	-	OFF	OFF	OFF	R/W	NO	OFF
M1145	Acceleration flag of accel/decel pulse output function of adjustable slope	-	Y	-	-	OFF	OFF	-	R	NO	OFF
M1146	Target attained frequency flag of accel/decel pulse output function of adjustable slope	-	Y	-	-	OFF	OFF	-	R	NO	OFF
M1147	Deceleration flag of accel/decel pulse output function of adjustable slope	-	Y	-	-	OFF	OFF	-	R	NO	OFF
M1148	Complete function flag of accel/decel pulse output function of adjustable slope	-	Y	-	-	OFF	OFF	OFF	R/W	NO	OFF
M1149	Stop counting temporality flag of accel/decel pulse output function of adjustable slope	-	Y	-	-	OFF	OFF	-	R/W	NO	OFF
M1150	DHSZ instruction in multiple set values comparison mode	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1151	The execution of DHSZ multiple set values comparison mode is completed.	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1152	Setting up DHSZ instruction as frequency control mode	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1153	DHSZ frequency control mode has been executed.	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1154	Start designated deceleration function flag of accel/decel pulse output function of adjustable slope	-	Y	-	-	OFF	-	-	R/W	NO	OFF
	PWD bandwidth detection duty-off/duty-on	-	-	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1155	Enable auto ramp up/down function for DCIMA, DCIMR instructions	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1156	Enable CH0 pulse output pause (ramp down) function when interrupt signal is triggered from X0. (When both M1156 and M1538 is ON, the remaining pulses will be executed by resetting M1108)	-	-	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1157	Enabling X1 interruption, immediate decelerating and stopping CH1 high-speed output	-	-	Y	Y	OFF	OFF	-	R/W	NO	OFF

SM	Function	ELCB -PB	ELC -PA	ELC -PV	ELC2 -PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1158	Enabling X2 interruption, immediate decelerating and stopping CH2 high-speed output	-	-	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1159	Enabling X3 interruption, immediate decelerating and stopping CH3 high-speed output	-	-	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1160	X4, X5 bandwidth detection flag	-	Y	-	-	OFF	OFF	-	R/W	NO	OFF
M1161	8-bit mode On: in 8-bit mode	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1162	Switching between decimal integer and binary floating point for SCLP instruction ON: binary floating point; OFF: decimal integer	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1163	Read/write memory card according to value in D1063 (automatically Off once the execution is completed)	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1164	Enable flash ROM access function according to the value set in D1064 (M1164 resets automatically when data access is completed)	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1165	When ON, the program and password on flash will be copied to the ELC when the ELC is powered.	-	-	Y	-	-	-	-	R/W	YES	OFF
M1166	When ON, the recipe on flash will be copied to the ELC when the ELC is powered.	-	-	Y	-	-	-	-	R/W	YES	OFF
M1167	HKY input is 16 bits mode	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1168	SMOV working mode indication	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1169	Selecting PWD modes	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1170	Enabling single step execution	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1171	Single step execution	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1172	2-phase pulse output switch (on is start)	-	Y	-	-	OFF	OFF	OFF	R/W	NO	OFF
M1173	ON is continuous output switch	-	Y	-	-	OFF	-	-	R/W	NO	OFF
M1174	Output pulse number attained flag	-	Y	-	-	OFF	OFF	OFF	R/W	NO	OFF
M1175	Losing ELC parameter data	-	-	Y	Y	-	-	-	R	YES	OFF
M1176	Losing the data in the ELC program	-	-	Y	Y	-	-	-	R	YES	OFF
M1178	VR0 Variable resistor enable	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1179	VR1 Variable resistor enable	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1181	Enabling X2 interruption (I201) followed by immediately clearing X0 high-speed counting input value. PS1: Only supports ELC-PA_V1.8 and above. PS2: After the high-speed counting value is obtained, the high-speed counting present value will be cleared immediately.	-	Y	-	-	OFF	OFF	-	R/W	NO	OFF
M1182	Enabling X3 interruption (I301) followed by immediately clearing X1 high-speed counting input value. PS1: Only supports ELC-PA _V1.8 and above. PS2: After the high-speed counting value is obtained, the high-speed counting present value will be cleared immediately.	-	Y	-	-	OFF	OFF	-	R/W	NO	OFF
M1183	On: The automatic matching read/write function of the special module is enabled. PS1: The right side module should support this function.	-	-	-	Y	ON	-	-	R/W	NO	ON
M1189	Read/write of Memory card/Flash ROM completed flag (Automatically reset to Off every time when enabled)	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1190	Set PLSY Y0 output as 0.01~100Hz	-	-	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1191	Set PLSY Y2 output as 0.01~100Hz	-	-	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1192	Set PLSY Y4 output as 0.01~100Hz	-	-	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1193	Set PLSY Y6 output as 0.01~100Hz	-	-	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1194	I40X, I50X interruptions is able to immediately update the present pulse output value at CH0.	-	-	Y	Y	OFF	OFF	-	R/W	NO	OFF

SM	Function	ELCB -PB	ELC -PA	ELC -PV	ELC2 -PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1195	I40X, I50X interruptions is able to immediately update the present pulse output value at CH1.	-	-	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1196	7-Seg Display mode. ON=Hex, OFF=Decimal. PA controllers only	-	Y	-	-	OFF	-	-	R/W	NO	OFF
M1197	7-Seg display. Display decimal point to the right of the LSD. PA controllers only	-	Y	-	-	OFF	-	-	R/W	NO	OFF
M1198	7-Seg display. Display decimal point to the right of the MSD. PA controllers only	-	Y	-	-	OFF	-	-	R/W	NO	OFF
M1200	C200 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1201	C201 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1202	C202 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1203	C203 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1204	C204 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1205	C205 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1206	C206 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1207	C207 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1208	C208 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1209	C209 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1210	C210 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1211	C211 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1212	C212 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1213	C213 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1214	C214 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1215	C215 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1216	C216 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1217	C217 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1218	C218 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1219	C219 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1220	C220 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1221	C221 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1222	C222 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1223	C223 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1224	C224 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1225	C225 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1226	C226 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF

SM	Function	ELCB -PB	ELC -PA	ELC -PV	ELC2 -PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1227	C227 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1228	C228 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1229	C229 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1230	C230 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1231	C231 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1232	C232 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1233	C233 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1234	C234 counting mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1235	C235 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1236	C236 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1237	C237 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1238	C238 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1239	C239 counter mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1240	C240 counter mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1241	C241 counter mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1242	C242 counter mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1243	C243 counter mode setting (ON: count down)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1244	C244 counter mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1245	C245 counter mode setting (ON: count down)	-	Y	-	-	OFF	-	-	R/W	NO	OFF
M1246	C246 counter monitor (ON: count down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1247	C247 counter monitor (ON: count down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1248	C248 counter monitor (ON: count down)	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1249	C249 counter monitor (ON: count down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1250	C250 counter monitor (ON: count down)	-	Y	-	-	OFF	-	-	R	NO	OFF
M1251	C251 counter monitor (ON: count down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1252	C252 counter monitor (ON: count down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1253	C253 counter monitor (ON: count down)	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1254	C254 counter monitor (ON: count down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1255	C255 counter monitor (ON: count down)	-	Y	-	-	OFF	-	-	R	NO	OFF
M1257	Set the ramp up/down of Y0, Y2 to be “S curve.” ON = S curve.	-	-	Y	Y	OFF	OFF	-	R	NO	OFF
M1258	Y0 pulse output signal reversing for PWM instruction	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1259	Y2 pulse output signal reversing for PWM instruction	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1260	Let X7 be the reset input signal of high-speed counters C235~C241	-	Y	-	-	OFF	-	-	R/W	NO	OFF
M1261	High-speed comparator comparison flag for DHSCR instruction	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1262	Enable cyclic output for table output function of DPTPO instruction	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1264	Enabling reset function of HHSC0	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1265	Enabling start function of HHSC0	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1266	Enabling reset function of HHSC1	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1267	Enabling start function of HHSC1	-	-	Y	Y	OFF	-	-	R/W	NO	OFF

SM	Function	ELCB -PB	ELC -PA	ELC -PV	ELC2 -PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1268	Enabling reset function of HHSC2	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1269	Enabling start function of HHSC2	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1270	Enabling reset function of HHSC3	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1271	Enabling start function of HHSC3	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1272	Reset control of HHSC0	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1273	Start control of HHSC0	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1274	Reset control of HHSC1	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1275	Start control of HHSC1	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1276	Reset control of HHSC2	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1277	Start control of HHSC2	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1278	Reset control of HHSC3	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1279	Start control of HHSC3	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1280	Inhibiting I000/I001	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1281	Inhibiting I100/I101	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1282	Inhibiting I200/I201	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1283	Inhibiting I300/I301	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1284	Inhibiting I400/I401	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
	For I400/I401, reverse interrupt trigger pulse direction (Rising/Falling)	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1285	Inhibiting I500/I501	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1286	Inhibiting I601~I699	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1287	Inhibiting I701~I799	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1288	Inhibiting I8801~I899	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1289	Inhibiting I010	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1290	Inhibiting I020	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1291	Inhibiting I030	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1292	Inhibiting I040	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1293	Inhibiting I050	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1294	Inhibiting I060	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1295	Inhibiting I110	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1296	Inhibiting I120	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1297	Inhibiting I130	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1298	Inhibiting I140	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1299	I150 flag disable	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1300	Inhibiting I160	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1301	Inhibiting I170	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1302	Inhibiting I180	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1303	High / low exchanged flag for XCH instruction	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1304	X input point can decide to be ON-OFF	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1305	Operation direction of the 1st group pulse CH0 (Y0, Y1) for PLSV/DPLSV/DRVI /DDRVI/DRVA/DDRVA instruction	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1306	Operation direction of the 2nd group pulse CH1 (Y2, Y3) for PLSV/DPLSV/DRVI /DDRVI/DRVA/DDRVA instruction	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1310	Immediately shut down Y10 pulse output starting flag	-	Y	-	-	OFF	OFF	-	R/W	NO	OFF
	Off->On: The 1st pulse group CH2 (Y4, Y5) high-speed output immediately stops. On->Off: Completing remaining number of output pulses	-	-	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1311	Immediately shut down Y11 pulse output starting flag	-	Y	-	-	OFF	OFF	-	R/W	NO	OFF
	Off->On: The 1st pulse group CH3 (Y6, Y7) high-speed output immediately stops. On->Off: Completing remaining number of output pulses	-	-	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1312	Controlling start input point of C235	-	-	Y	-	OFF	-	-	R/W	NO	OFF
	For COM1(RS-232), sending request (Only applicable for MODRW and RS instruction)	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1313	Controlling start input point of C236	-	-	Y	-	OFF	-	-	R/W	NO	OFF
	For COM1(RS-232), ready for data receiving (only applicable for MODRW and RS instruction)	-	-	-	Y	OFF	OFF	-	R	NO	OFF
M1314	Controlling start input point of C237	-	-	Y	-	OFF	-	-	R/W	NO	OFF
	For COM1(RS-232), data receiving	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF

SM	Function	ELCB -PB	ELC -PA	ELC -PV	ELC2 -PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
	completed (only applicable for MODRW and RS instruction)										
M1315	Controlling start input point of C238	-	-	Y	-	OFF	-	-	R/W	NO	OFF
	For COM1(RS-232), data receiving error (only applicable for MODRW and RS instruction)	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1316	Controlling start input point of C239	-	-	Y	-	OFF	-	-	R/W	NO	OFF
M1317	Controlling start input point of C240	-	-	Y	-	OFF	-	-	R/W	NO	OFF
M1320	Controlling reset input point of C235	-	-	Y	-	OFF	-	-	R/W	NO	OFF
M1321	Controlling reset input point of C236	-	-	Y	-	OFF	-	-	R/W	NO	OFF
M1322	Controlling reset input point of C237	-	-	Y	-	OFF	-	-	R/W	NO	OFF
M1323	Controlling reset input point of C238	-	-	Y	-	OFF	-	-	R/W	NO	OFF
M1324	Controlling reset input point of C239	-	-	Y	-	OFF	-	-	R/W	NO	OFF
M1325	Controlling reset input point of C240	-	-	Y	-	OFF	-	-	R/W	NO	OFF
M1328	Enabling start/reset of C235	-	-	Y	-	OFF	-	-	R/W	NO	OFF
M1329	Enabling start/reset of C236	-	-	Y	-	OFF	-	-	R/W	NO	OFF
M1330	Enabling start/reset of C237	-	-	Y	-	OFF	-	-	R/W	NO	OFF
M1331	Enabling start/reset of C238	-	-	Y	-	OFF	-	-	R/W	NO	OFF
M1332	Enabling start/reset of C239	-	-	Y	-	OFF	-	-	R/W	NO	OFF
M1333	Enabling start/reset of C240	-	-	Y	-	OFF	-	-	R/W	NO	OFF
M1334	Stopping the 1st group pulse output CH0 (Y0, Y1)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1335	Stopping the 2nd group pulse output CH1 (Y2, Y3)	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1336	Sending out the 1st group pulse output CH0 (Y0, Y1)	-	-	Y	Y	OFF	OFF	OFF	R	NO	OFF
M1337	Sending out the 2nd group pulse output CH1 (Y2, Y3)	-	-	Y	Y	OFF	OFF	OFF	R	NO	OFF
M1338	Enabling offset pulses of the 1st group pulse output CH0 (Y0, Y1)	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1339	Enabling offset pulses of the 2nd group pulse output CH1 (Y2, Y3)	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1340	Generating interruption I110 after the 1st group pulse output CH0 (Y0, Y1) is sent out	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1341	Generating interruption I120 after the 2nd group pulse output CH1 (Y2, Y3) is sent out	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1342	Generating interruption I130 when the 1st group pulse output CH0 (Y0, Y1) is sent out	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1343	Generating interruption I140 when the 2nd group pulse output CH1 (Y2, Y3) is sent out	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1344	Enabling the offset of the 1st group pulse output CH0 (Y0, Y1)	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1345	Enabling the offset of the 2nd group pulse output CH1 (Y2, Y3)	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1346	Enabling ZRN CLEAR output signal	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1347	Reset after the 1st group pulse output CH0 (Y0, Y1) is completed for PLSY instruction	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
	Auto-reset Y0 when high speed pulse output completed	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1348	Reset after the 2nd group pulse output CH1 (Y2, Y3) is completed for PLSY instruction	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
	Auto-reset Y1 when high speed pulse output completed	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1350	Enable the function of ELC Link	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1351	Enable auto mode on ELC Link	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1352	Enable manual mode on ELC Link	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1353	Enable 32 slave unit linkage and up to 100 data length of data exchange on ELC LINK	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1354	Enable simultaneous data read/write in a polling of ELC Link	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1355	Select Slave linking mode in ELC LINK (ON: manual; OFF: auto-detection)	-	Y	Y	Y	-	-	-	R/W	YES	OFF
M1356	When the ELC link is enabled and M1356 is ON, the values in D1900~D1931 are taken as the station address. The default station address in D1399 is not used.	-	-	-	Y	-	-	-	R/W	NO	OFF

SM	Function	ELCB -PB	ELC -PA	ELC -PV	ELC2 -PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1360	Slave ID#1 status on ELC Link network	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1361	Slave ID#2 status on ELC Link network	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1362	Slave ID#3 status on ELC Link network	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1363	Slave ID#4 status on ELC Link network	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1364	Slave ID#5 status on ELC Link network	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1365	Slave ID#6 status on ELC Link network	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1366	Slave ID#7 status on ELC Link network	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1367	Slave ID#8 status on ELC Link network	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1368	Slave ID#9 status on ELC Link network	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1369	Slave ID#10 status on ELC Link network	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1370	Slave ID#11 status on ELC Link network	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1371	Slave ID#12 status on ELC Link network	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1372	Slave ID#13 status on ELC Link network	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1373	Slave ID#14 status on ELC Link network	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1374	Slave ID#15 status on ELC Link network	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1375	Slave ID#16 status on ELC Link network	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1376	Indicating Slave ID#1 data transaction status on ELC Link	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1377	Indicating Slave ID#2 data transaction status on ELC Link	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1378	Indicating Slave ID#3 data transaction status on ELC Link	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1379	Indicating Slave ID#4 data transaction status on ELC Link	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1380	Indicating Slave ID#5 data transaction status on ELC Link	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1381	Indicating Slave ID#6 data transaction status on ELC Link	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1382	Indicating Slave ID#7 data transaction status on ELC Link	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1383	Indicating Slave ID#8 data transaction status on ELC Link	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1384	Indicating Slave ID#9 data transaction status on ELC Link	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1385	Indicating Slave ID#10 data transaction status on ELC Link	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1386	Indicating Slave ID#11 data transaction status on ELC Link	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1387	Indicating Slave ID#12 data transaction status on ELC Link	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1388	Indicating Slave ID#13 data transaction status on ELC Link	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1389	Indicating Slave ID#14 data transaction status on ELC Link	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1390	Indicating Slave ID#15 data transaction status on ELC Link	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1391	Indicating Slave ID#16 data transaction status on ELC Link	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1392	Slave ID#1 linking error	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1393	Slave ID#2 linking error	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1394	Slave ID#3 linking error	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1395	Slave ID#4 linking error	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1396	Slave ID#5 linking error	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1397	Slave ID#6 linking error	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1398	Slave ID#7 linking error	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1399	Slave ID#8 linking error	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1400	Slave ID#9 linking error	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1401	Slave ID#10 linking error	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1402	Slave ID#11 linking error	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1403	Slave ID#12 linking error	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1404	Slave ID#13 linking error	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1405	Slave ID#14 linking error	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1406	Slave ID#15 linking error	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1407	Slave ID#16 linking error	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1408	Indicating reading from Slave ID#1 is	-	Y	Y	Y	OFF	-	-	R	NO	OFF

SM	Function	ELCB -PB	ELC -PA	ELC -PV	ELC2 -PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
	completed										
M1409	Indicating reading from Slave ID#2 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1410	Indicating reading from Slave ID#3 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1411	Indicating reading from Slave ID#4 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1412	Indicating reading from Slave ID#5 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1413	Indicating reading from Slave ID#6 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1414	Indicating reading from Slave ID#7 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1415	Indicating reading from Slave ID#8 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1416	Indicating reading from Slave ID#9 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1417	Indicating reading from Slave ID#10 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1418	Indicating reading from Slave ID#11 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1419	Indicating reading from Slave ID#12 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1420	Indicating reading from Slave ID#13 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1421	Indicating reading from Slave ID#14 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1422	Indicating reading from Slave ID#15 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1423	Indicating reading from Slave ID#16 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1424	Indicating writing to Slave ID#1 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1425	Indicating writing to Slave ID#2 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1426	Indicating writing to Slave ID#3 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1427	Indicating writing to Slave ID#4 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1428	Indicating writing to Slave ID#5 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1429	Indicating writing to Slave ID#6 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1430	Indicating writing to Slave ID#7 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1431	Indicating writing to Slave ID#8 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1432	Indicating writing to Slave ID#9 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1433	Indicating writing to Slave ID#10 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1434	Indicating writing to Slave ID#11 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1435	Indicating writing to Slave ID#12 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1436	Indicating writing to Slave ID#13 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1437	Indicating writing to Slave ID#14 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1438	Indicating writing to Slave ID#15 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1439	Indicating writing to Slave ID#16 is completed	-	Y	Y	Y	OFF	-	-	R	NO	OFF
M1440	Slave ID#17 status on ELC LINK network	-	-	Y	-	OFF	-	-	R	NO	OFF

SM	Function	ELCB -PB	ELC -PA	ELC -PV	ELC2 -PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
		-	-	-	Y	-	-	-	R	YES	OFF
M1441	Slave ID#18 status on ELC LINK network	-	-	Y	-	OFF	-	-	R	NO	OFF
		-	-	-	Y	-	-	-	R	YES	OFF
M1442	Slave ID#19 status on ELC LINK network	-	-	Y	-	OFF	-	-	R	NO	OFF
		-	-	-	Y	-	-	-	R	YES	OFF
M1443	Slave ID#20 status on ELC LINK network	-	-	Y	-	OFF	-	-	R	NO	OFF
		-	-	-	Y	-	-	-	R	YES	OFF
M1444	Slave ID#21 status on ELC LINK network	-	-	Y	-	OFF	-	-	R	NO	OFF
		-	-	-	Y	-	-	-	R	YES	OFF
M1445	Slave ID#22 status on ELC LINK network	-	-	Y	-	OFF	-	-	R	NO	OFF
		-	-	-	Y	-	-	-	R	YES	OFF
M1446	Slave ID#23 status on ELC LINK network	-	-	Y	-	OFF	-	-	R	NO	OFF
		-	-	-	Y	-	-	-	R	YES	OFF
M1447	Slave ID#24 status on ELC LINK network	-	-	Y	-	OFF	-	-	R	NO	OFF
		-	-	-	Y	-	-	-	R	YES	OFF
M1448	Slave ID#25 status on ELC LINK network	-	-	Y	-	OFF	-	-	R	NO	OFF
		-	-	-	Y	-	-	-	R	YES	OFF
M1449	Slave ID#26 status on ELC LINK network	-	-	Y	-	OFF	-	-	R	NO	OFF
		-	-	-	Y	-	-	-	R	YES	OFF
M1450	Slave ID#27 status on ELC LINK network	-	-	Y	-	OFF	-	-	R	NO	OFF
		-	-	-	Y	-	-	-	R	YES	OFF
M1451	Slave ID#28 status on ELC LINK network	-	-	Y	-	OFF	-	-	R	NO	OFF
		-	-	-	Y	-	-	-	R	YES	OFF
M1452	Slave ID#29 status on ELC LINK network	-	-	Y	-	OFF	-	-	R	NO	OFF
		-	-	-	Y	-	-	-	R	YES	OFF
M1453	Slave ID#30 status on ELC LINK network	-	-	Y	-	OFF	-	-	R	NO	OFF
		-	-	-	Y	-	-	-	R	YES	OFF
M1454	Slave ID#31 status on ELC LINK network	-	-	Y	-	OFF	-	-	R	NO	OFF
		-	-	-	Y	-	-	-	R	YES	OFF
M1455	Slave ID#32 status on ELC LINK network	-	-	Y	-	OFF	-	-	R	NO	OFF
		-	-	-	Y	-	-	-	R	YES	OFF
M1456	Indicating Slave ID#17 data transaction status on ELC LINK	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1457	Indicating Slave ID#18 data transaction status on ELC LINK	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1458	Indicating Slave ID#19 data transaction status on ELC LINK	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1459	Indicating Slave ID#20 data transaction status on ELC LINK	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1460	Indicating Slave ID#21 data transaction status on ELC LINK	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1461	Indicating Slave ID#22 data transaction status on ELC LINK	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1462	Indicating Slave ID#23 data transaction status on ELC LINK	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1463	Indicating Slave ID#24 data transaction status on ELC LINK	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1464	Indicating Slave ID#25 data transaction status on ELC LINK	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1465	Indicating Slave ID#26 data transaction status on ELC LINK	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1466	Indicating Slave ID#27 data transaction status on ELC LINK	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1467	Indicating Slave ID#28 data transaction status on ELC LINK	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1468	Indicating Slave ID#29 data transaction status on ELC LINK	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1469	Indicating Slave ID#30 data transaction status on ELC LINK	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1470	Indicating Slave ID#31 data transaction status on ELC LINK	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1471	Indicating Slave ID#32 data transaction status on ELC LINK	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1472	Slave ID#17 linking error	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1473	Slave ID#18 linking error	-	-	Y	Y	OFF	-	-	R	NO	OFF

SM	Function	ELCB -PB	ELC -PA	ELC -PV	ELC2 -PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1474	Slave ID#19 linking error	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1475	Slave ID#20 linking error	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1476	Slave ID#21 linking error	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1477	Slave ID#22 linking error	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1478	Slave ID#23 linking error	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1479	Slave ID#24 linking error	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1480	Slave ID#25 linking error	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1481	Slave ID#26 linking error	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1482	Slave ID#27 linking error	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1483	Slave ID#28 linking error	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1484	Slave ID#29 linking error	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1485	Slave ID#30 linking error	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1486	Slave ID#31 linking error	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1487	Slave ID#32 linking error	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1488	Indicating reading from Slave ID#17 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1489	Indicating reading from Slave ID#18 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1490	Indicating reading from Slave ID#19 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1491	Indicating reading from Slave ID#20 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1492	Indicating reading from Slave ID#21 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1493	Indicating reading from Slave ID#22 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1494	Indicating reading from Slave ID#23 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1495	Indicating reading from Slave ID#24 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1496	Indicating reading from Slave ID#25 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1497	Indicating reading from Slave ID#26 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1498	Indicating reading from Slave ID#27 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1499	Indicating reading from Slave ID#28 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1500	Indicating reading from Slave ID#29 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1501	Indicating reading from Slave ID#30 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1502	Indicating reading from Slave ID#31 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1503	Indicating reading from Slave ID#32 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1504	Indicating writing to Slave ID#17 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1505	Indicating writing to Slave ID#18 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1506	Indicating writing to Slave ID#19 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1507	Indicating writing to Slave ID#20 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1508	Indicating writing to Slave ID#21 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1509	Indicating writing to Slave ID#22 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1510	Indicating writing to Slave ID#23 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1511	Indicating writing to Slave ID#24 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1512	Indicating writing to Slave ID#25 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF

SM	Function	ELCB -PB	ELC -PA	ELC -PV	ELC2 -PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1513	Indicating writing to Slave ID#26 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1514	Indicating writing to Slave ID#27 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1515	Indicating writing to Slave ID#28 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1516	Indicating writing to Slave ID#29 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1517	Indicating writing to Slave ID#30 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1518	Indicating writing to Slave ID#31 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1519	Indicating writing to Slave ID#32 is completed	-	-	Y	Y	OFF	-	-	R	NO	OFF
M1520	Stopping the 3rd group pulse output CH2 (Y4, Y5)	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1521	Stopping the 4th group pulse output CH3 (Y6, Y7)	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1522	Sending out the 3rd group pulse output CH2 (Y4, Y5)	-	-	Y	Y	OFF	-	OFF	R	NO	OFF
M1523	Sending out the 4th group pulse output CH3 (Y6, Y7)	-	-	Y	Y	OFF	-	OFF	R	NO	OFF
M1524	Auto-reset Y2 when high speed pulse output completes	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1525	Auto-reset Y3 when high speed pulse output completes	-	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1526	Reversing Y4 pulse output signal for PWM instruction	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1527	Reversing Y6 pulse output signal for PWM instruction	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1528	Enabling the instruction DICF to execute the constant speed output section	-	-	-	Y	OFF	OFF	OFF	R/W	NO	OFF
M1529	Enabling the instruction DICF to execute the final output section	-	-	-	Y	OFF	OFF	OFF	R/W	NO	OFF
M1530	Switching time resolution of Y4 output for PWM instruction (ON: 100us; OFF: 1ms)	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1531	Switching time resolution of Y6 output for PWM instruction (ON: 100us; OFF: 1ms)	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1532	Reverse operation of the 3rd group pulse CH2 (Y4, Y5) for PLSV/DPLSV/DRVI /DDRVI/DRVA/DDRVA instruction	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1533	Reverse operation of the 4th group pulse CH3 (Y6, Y7) for PLSV/DPLSV/DRVI /DDRVI/DRVA/DDRVA instruction	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1534	CH0 being able to designate deceleration time. Has to be used with D1348.	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1535	CH1 being able to designate deceleration time. Has to be used with D1349.	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1536	CH2 being able to designate deceleration time. Has to be used with D1350.	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1537	CH3 being able to designate deceleration time. Has to be used with D1351.	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1538	Indicating pause status of high speed output in CH0	-	-	Y	Y	OFF	-	OFF	R/W	NO	OFF
M1539	Indicating pause status of high speed output in CH1	-	-	Y	Y	OFF	-	OFF	R/W	NO	OFF
M1540	Indicating pause status of high speed output in CH2	-	-	Y	Y	OFF	-	OFF	R/W	NO	OFF
M1541	Indicating pause status of high speed output in CH3	-	-	Y	Y	OFF	-	OFF	R/W	NO	OFF
M1542	CH0 executes the function that the constant speed output section reaches the target frequency.	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF

SM	Function	ELCB -PB	ELC -PA	ELC -PV	ELC2 -PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1543	CH0 executed the function that the constant speed output section reaches the target number.	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1544	CH1 executes the function that the constant speed output section reaches the target frequency.	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1545	CH1 executed the function that the constant speed output section reaches the target number.	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1546	CH2 executes the function that the constant speed output section reaches the target frequency.	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1547	CH2 executed the function that the constant speed output section reaches the target number.	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1548	CH3 executes the function that the constant speed output section reaches the target frequency.	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1549	CH3 executed the function that the constant speed output section reaches the target number.	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1550	Used with the instruction DCIF to clear the high-speed output couting number	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1560	Inhibiting I900 and I901	-	-	-	Y	OFF	-	-	R/W	NO	OFF
M1561	Inhibiting I910 and I911	-	-	-	Y	OFF	-	-	R/W	NO	OFF
M1562	Inhibiting I920 and I921	-	-	-	Y	OFF	-	-	R/W	NO	OFF
M1563	Inhibiting I930 and I931	-	-	-	Y	OFF	-	-	R/W	NO	OFF
M1564	Inhibiting I940 and I941	-	-	-	Y	OFF	-	-	R/W	NO	OFF
M1565	Inhibiting I950 and I951	-	-	-	Y	OFF	-	-	R/W	NO	OFF
M1566	Inhibiting I960 and I961	-	-	-	Y	OFF	-	-	R/W	NO	OFF
M1567	Inhibiting I970 and I971	-	-	-	Y	OFF	-	-	R/W	NO	OFF
M1570	Enabling the negative limit function of the high-speed output CH0	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1571	Enabling the negative limit function of the high-speed output CH1	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1572	Enabling the negative limit function of the high-speed output CH2	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1573	Enabling the negative limit function of the high-speed output CH3	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1574	The DOG of CH0 in the instruction ZRN is positive stop function.	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1575	The DOG of CH1 in the instruction ZRN is positive stop function.	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1576	The DOG of CH2 in the instruction ZRN is positive stop function.	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1577	The DOG of CH3 in the instruction ZRN is positive stop function.	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1578	Off: Number of times the instruction ZRN search for the Z phase On: The output designates the displacement. The flag is used with D1312.	-	-	-	Y	OFF	OFF	-	R/W	NO	OFF
M1579	Disable the checksum mechanism of the left/right side module (On: Disable; Off: Enable)	-	-	-	Y	-	-	-	R/W	YES	OFF

For ELCM-PH/ELCM-PA, ELC2-PB, ELC2-PC/ELC2-PE, and ELC2-PA:

SM	Function	ELCM -PH -PA	ELC2- PB	ELC2- PC -PE	ELC2 -PA	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1000	Normally open contact (a contact). This contact is ON when running and it is ON when the status is set to RUN.	Y	Y	Y	Y	OFF	ON	OFF	R	NO	OFF

SM	Function	ELCM -PH -PA	ELC2-PB	ELC2-PC -PE	ELC2 -PA	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1001	Normally OFF contact (b contact). This contact is OFF in running and it is OFF when the status is set to RUN.	Y	Y	Y	Y	ON	OFF	ON	R	NO	ON
M1002	ON only for 1 scan after RUN. Initial pulse is contact a. It will get positive pulse in the RUN moment. Pulse width=scan period.	Y	Y	Y	Y	OFF	ON	OFF	R	NO	OFF
M1003	OFF only for 1 scan after RUN. Initial pulse is contact a. It will get negative pulse in the RUN moment. Pulse width=scan period.	Y	Y	Y	Y	ON	OFF	ON	R	NO	ON
M1004	ON when error occurs	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1008	Monitor timer flag (ON: ELC WDT time out)	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1009	History of LV signal due to 24VDC insufficiency	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1011	10ms clock pulse, 5ms ON/5ms OFF	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1012	100ms clock pulse, 50ms ON / 50ms OFF	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1013	1s clock pulse, 0.5s ON / 0.5s OFF	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1014	1min clock pulse, 30s ON / 30s OFF	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1015	High-speed connection counter	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1016	Display year bit. When OFF = display two right-most bits. When ON = display (two right-most bits + 2000).	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1017	±30 seconds adjustment	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1018	Flag for Radian/Degree, ON for degree	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1020	Zero flag	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1021	Barrow flag	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1022	Carry flag	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1024	COM1 monitor request	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1025	If the ELC receives an illegal communication request when PC or HMI connects to an ELC, M1025 =ON and save the error code in D1025.	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1026	Startup flag of RAMP module	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1027	PR output flag	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1028	10ms time switch flag. The base setting flag of T64~T126 is 100ms, when timer is OFF and the base setting flag is 10ms when it is ON.	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1029	Y0 or CH0 (Y0, Y1) pulse output execution completed.	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1030	Pulse output Y1 execution completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1031	Clear all non-latched memory	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1032	Clear all latched memory	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1033	Memory latched at STOP	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1034	All Y outputs disable	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1035	X7 input point to be RUN/STOP switch	Y	Y	Y	Y	-	-	-	R/W	YES	OFF
M1037	Enable 8-sets SPD function (Has to be used with D1037)	1.4	1.2	Y	1.2	OFF	OFF	OFF	R/W	NO	OFF
M1038	Switching T200~T255 timer resolution (10ms/1ms). ON = 1ms.	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1039	Constant scan mode	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1040	Step transition inhibit	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1041	Step transition start	Y	Y	Y	Y	OFF	-	OFF	R/W	NO	OFF
M1042	Start pulse	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1043	Zero point return completed	Y	Y	Y	Y	OFF	-	OFF	R/W	NO	OFF
M1044	Zero point condition	Y	Y	Y	Y	OFF	-	OFF	R/W	NO	OFF
M1045	All outputs clear inhibit	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF

SM	Function	ELCM -PH -PA	ELC2-PB	ELC2-PC -PE	ELC2 -PA	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1046	STL state setting (ON)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1047	STL monitor enable	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1048	Flag for alarm point state	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1049	Monitor flag for alarm point	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1050	I000/I001 masked	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1051	I100/I101 masked	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1052	I200/I201 masked	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1053	I300/I301 masked	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1054	I400/I401 masked	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1055	I500/I501 masked	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1056	I602~ I699 masked	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1057	I702~I799 masked	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1058	COM3 monitor request	Y	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1059	Disable high-speed counter interruptions I010~I080	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1060	System error message 1	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1061	System error message 2	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1062	System error message 3	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1063	System error message 4	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1064	Operator error	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1065	Syntax error	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1066	Program error	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1067	Program execution error	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1068	Execution error latched (D1068)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1070	Y1 time base switching for PWM instruction (ON: 100us; OFF: 1ms)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1071	Switching clock pulse of Y3 for PWM instruction (ON: 100us; OFF: 1ms)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1072	ELC status (RUN/STOP), ON = RUN	Y	Y	Y	Y	OFF	ON	OFF	R/W	NO	OFF
M1075	Error occurring when write in Flash ROM	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1078	Y0/CH0(Y0, Y1) pulse output pause (immediate)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1079	Y1 pulse output pause (immediate)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1080	COM2 monitor request	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1081	Changing conversion mode for FLT instruction	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1083	Selecting X6 pulse-width detecting mode. M1083 = ON, detecting pulse-width when X6 = ON; M1083 = OFF, detecting pulse-width when X6 = OFF.	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1084	Enabling X6 Pulse width detecting function. (has to be used with M1183 and D1023)	Y	Y	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1085	Selecting ELC-ACPGMXFR duplicating function	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1086	Enabling password function for ELC-ACPGMXFR	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1088	Matrix comparison. Comparing between equivalent values (M1088 = ON) or different values (M1088 = OFF).	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1089	Indicating the end of matrix comparison. When the comparison reaches the last bit, M1089 = ON.	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1090	Matrix search start flag. Compare from the first bit and M1090=1.	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1091	Matrix finding bit flag. When find it, it will stop comparing and M1091=1.	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1092	Matrix pointer error flag. When pointer Pr exceeds this range, M1092=1.	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1093	Matrix pointer increase flag. It will add 1 to present pointer.	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF

SM	Function	ELCM -PH -PA	ELC2-PB	ELC2-PC -PE	ELC2 -PA	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1094	Matrix pointer clear flag. It will clear present pointer to 0.	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1095	Carry flag for matrix rotate/shift output	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1096	Complement flag for matrix shift input	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1097	Direction flag for matrix rotate/shift	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1098	Matrix count bit 0 or 1 flag	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1099	It is ON when matrix count result 0	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1102	Y2 pulse or CH1 (Y2, Y3) pulse output execution completed	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1103	Y3 pulse output completed	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1104	Y2 pulse or CH1 (Y2, Y3) pulse output pause (immediate)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1105	Y3 pulse output pause (immediate)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1106	Zero point selection. M1106=ON, change the zero point to the right of DOG switch for zero return on CH0.	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1107	Zero point selection. M1107=ON, change the zero point to the right of DOG switch for zero return on CH1.	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1108	Y0 pulse or CH0 (Y0, Y1) pulse output pause (ramp down)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1109	Y1 pulse output pause (ramp down)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1110	Y2 pulse or CH1 (Y2, Y3) pulse output pause (ramp down)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1111	Y3 pulse output pause (ramp down)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1112	Switching time resolution of Y0 for PWM instruction (ON: 100us; OFF: 1ms)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1113	Switching time resolution of Y2 for PWM instruction (ON: 100us; OFF: 1ms)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1119	Enable 2-speed output function of DDRVI instruction	Y	-	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1120	Set COM2 (RS-485) protocol kept on. D1120 cannot be changed after the setting.	Y	Y	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1121	Transmission ready	Y	Y	Y	Y	OFF	OFF	ON	R	NO	OFF
M1122	Sending request	Y	Y	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1123	Receiving completed	Y	Y	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1124	Receiving wait	Y	Y	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1125	Communication reset	Y	Y	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1126	STX/ETX user/system selection	Y	Y	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1127	MODRD/RDST/MODRW instructions. Data receiving completed.	Y	Y	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1128	Transmitting/Receiving Indication	Y	Y	Y	Y	OFF	OFF	OFF	R/W	NO	OFF
M1129	Receiving time out	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1130	STX/ETX selection	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1131	MODRD/RDST/MODRW, M1131=ON when data convert to HEX	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1132	ON= no relative communications instruction in ELC program.	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1136	For COM3(RS-485), retaining communication setting	Y	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1137	Retain the data in DNET mapping area in STOP state	-	-	Y	Y	-	-	-	R/W	NO	OFF
M1138	Set COM1 (RS-232) protocol kept on. D1036 cannot be changed after setting	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1139	COM1 (RS-232) ASCII/RTU mode (OFF:ASCII, ON:RTU)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1140	MODRD/MODWR/MODRW data received error	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF

SM	Function	ELCM -PH -PA	ELC2-PB	ELC2-PC -PE	ELC2 -PA	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1141	MODRD/MODWR/MODRW instruction error	Y	Y	Y	Y	OFF	OFF	-	R	NO	OFF
M1143	1. COM2 (RS-485) ASCII/RTU mode (OFF:ASCII, ON:RTU)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1148	If DELAY is used, the delay time unit is 5us. If M1148 is ON, the delay time unit us 5us. If M1148 is OFF, the delay time unit us 100us. If M1148 is ON, it will become OFF after DELAY is executed.	V3.2	V3.0	V2.6 V1.4	V2.4	OFF	OFF	OFF	R/W	NO	OFF
M1156	Enabling the mask and alignment mark function on I400/I401(X4) corresponding to Y0	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1158	Enabling the mask and alignment mark function on I600/I601(X6) corresponding to Y2	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1161	8/16 bits mode (ON = 8 bit mode)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1162	Switching between decimal integer and binary floating point for SCLP instruction ON: binary floating point; OFF: decimal integer	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1167	HKY input is 16 bits mode	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1168	SMOV working mode indication	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1178	VR0 Variable resistor enable	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1179	VR1 Variable resistor enable	-	-	Y	Y	OFF	-	-	R/W	NO	OFF
M1180	Read analog-to-digital values immediately. (M1180 is used with REF.)	ELCM-PA	-	-	Y	OFF	-	-	R/W	NO	OFF
M1181	Output digital-to-analog values immediately. (M1181 is used with REF.)	ELCM-PA	-	-	Y	OFF	-	-	R/W	NO	OFF
M1182	M1182 = ON, disable auto-mapping function when connected with left-side modules(values will be auto-mapped to D9800 and above.).	-	-	Y	Y	ON	-	-	R/W	NO	ON
M1183	M1183 = ON, disable auto mapping function when connected with AIO modules #: ELCM-PH/PA: OFF; ELC2-PB/PH/PA/PE: ON	Y	Y	Y	Y	#	-	-	R/W	NO	#
M1190	Set PLSY Y0 output as 0.01~10Hz	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1191	Set PLSY Y1 output as 0.01~10Hz	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1192	Set PLSY Y2 output as 0.01~10Hz	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1193	Set PLSY Y3 output as 0.01~10Hz	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1200	C200 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1201	C201 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1202	C202 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1203	C203 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1204	C204 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1205	C205 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1206	C206 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1207	C207 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1208	C208 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF

SM	Function	ELCM -PH -PA	ELC2-PB	ELC2-PC -PE	ELC2 -PA	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1209	C209 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1210	C210 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1211	C211 counting mode setting (ON: cour down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1212	C212 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1213	C213 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1214	C214 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1215	C215 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1216	C216 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1217	C217 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1218	C218 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1219	C219 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1220	C220 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1221	C221 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1222	C222 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1223	C223 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1224	C224 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1225	C225 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1226	C226 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1227	C227 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1228	C228 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1229	C229 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1230	C230 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1231	C231 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1232	C232 counting mode setting (ON: count down)	-	Y	-	-	OFF	-	-	R/W	NO	OFF
	C232 counter mode monitor (ON: count down)	Y	-	Y	Y	OFF	-	-	R	NO	OFF
M1233	C233 counter mode monitor (ON: count down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1234	C234 counter mode monitor (ON: count down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1235	C235 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1236	C236 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1237	C237 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1238	C238 counting mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF

SM	Function	ELCM -PH -PA	ELC2-PB	ELC2-PC -PE	ELC2 -PA	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1239	C239 counter mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1240	C240 counter mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1241	C241 counter mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1242	C242 counter mode setting (ON: count down)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1243	C243 Reset enable	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1244	C244 Reset enable	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1245	C245 counter mode monitor (ON: cour down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1246	C246 counter monitor (ON: count down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1247	C247 counter monitor (ON: count down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1248	C248 counter monitor (ON: count down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1249	C249 counter monitor (ON: count down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1250	C250 counter monitor (ON: count down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1251	C251 counter monitor (ON: count down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1252	C252 counter monitor (ON: count down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1253	C253 counter monitor (ON: count down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1254	C254 counter monitor (ON: count down)	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1257	Set the ramp up/down of Y0, Y2 to be “S curve.” ON = S curve.	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1260	Let X7 be the reset input signal of high-speed counters C235~C241	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1262	Enable cyclic output for table output function of DPTPO instruction	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1270	C235 counting mode setting (ON: falling-edge count)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1271	C236 counting mode setting (ON: falling-edge count)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1272	C237 counting mode setting (ON: falling-edge count)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1273	C238 counting mode setting (ON: falling-edge count)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1274	C239 counting mode setting (ON: falling-edge count)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1275	C240 counting mode setting (ON: falling-edge count)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1276	C241 counting mode setting (ON: falling-edge count)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1277	C242 counting mode setting (ON: falling-edge count)	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1280	For I000/I001, reverse interrupt trigger pulse direction (Rising/Falling)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1284	For I400/I401, reverse interrupt trigger pulse direction (Rising/Falling)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1286	For I600 / I601, reverse interrupt trigger pulse direction (Rising/Falling)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1303	High / low exchanged flag for XCH instruction	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1304	X input point can decide to be ON-OFF	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1305	Reverse Y1 pulse output direction in high speed pulse output instructions	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF

SM	Function	ELCM -PH -PA	ELC2-PB	ELC2-PC -PE	ELC2 -PA	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1306	Reverse Y3 pulse output direction in high speed pulse output instructions	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1307	For ZRN instruction, enable left limit switch	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1312	For COM1(RS-232), sending request (Only applicable for MODRW and RS instruction)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1313	For COM1(RS-232), ready for data receiving (only applicable for MODRW and RS instruction)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1314	For COM1(RS-232), data receiving completed (only applicable for MODRW and RS instruction)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1315	For COM1(RS-232), data receiving error (only applicable for MODRW and RS instruction)	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1316	For COM3(RS-485), sending request (only applicable for MODRW and RS instruction)	Y	-	Y	-	OFF	OFF	-	R/W	NO	OFF
M1317	For COM3(RS-485), ready for data receiving (only applicable for MODRW and RS instruction)	Y	-	Y	-	OFF	OFF	-	R/W	NO	OFF
M1318	For COM3(RS-485), data receiving completed (only applicable for MODRW and RS instruction)	Y	-	Y	-	OFF	OFF	-	R/W	NO	OFF
M1319	For COM3(RS-485), data receiving error (only applicable for MODRW and RS instruction)	Y	-	Y	-	OFF	OFF	-	R/W	NO	OFF
M1320	For COM3 (RS-485), ASCII/RTU mode selection. (OFF: ASCII; ON: RTU)	Y	-	Y	-	OFF	-	-	R/W	NO	OFF
M1346	Enabling ZRN CLEAR output signal	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1347	Auto-reset Y0 when high speed pulse output completed	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1348	Auto-reset Y1 when high speed pulse output completed	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1350	Enable the function of ELC Link	Y	Y	Y	Y	OFF	-	OFF	R/W	NO	OFF
M1351	Enable auto mode on ELC Link	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1352	Enable manual mode on ELC Link	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1353	Enable 32 slave unit linkage and up to 100 data length of data exchange on ELC LINK	ELCM-PH	-	-	-	OFF	-	-	R/W	NO	OFF
	Enable up to 100 data length of data exchange	V1.3	Y	Y	Y	-	-	-	R/W	YES	OFF
M1354	Enable simultaneous data read/write in a polling of ELC Link	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1355	Select Slave linking mode in ELC LINK (ON: manual; OFF: auto-detection)	Y	Y	Y	Y	-	-	-	R/W	YES	OFF
M1356	Enable station number selection function. When both M1353 and M1356 are ON, the user can specify the station number in D1900~D1931	Y	×	Y	Y	-	-	-	R/W	YES	OFF
M1360	Slave ID#1 status on ELC Link network	Y	Y	Y	Y	-	-	-	R/W	YES	OFF
M1361	Slave ID#2 status on ELC Link network	Y	Y	Y	Y	-	-	-	R/W	YES	OFF
M1362	Slave ID#3 status on ELC Link network	Y	Y	Y	Y	-	-	-	R/W	YES	OFF
M1363	Slave ID#4 status on ELC Link network	Y	Y	Y	Y	-	-	-	R/W	YES	OFF
M1364	Slave ID#5 status on ELC Link network	Y	Y	Y	Y	-	-	-	R/W	YES	OFF
M1365	Slave ID#6 status on ELC Link network	Y	Y	Y	Y	-	-	-	R/W	YES	OFF

SM	Function	ELCM -PH -PA	ELC2-PB	ELC2-PC -PE	ELC2 -PA	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1366	Slave ID#7 status on ELC Link network	Y	Y	Y	Y	-	-	-	R/W	YES	OFF
M1367	Slave ID#8 status on ELC Link network	Y	Y	Y	Y	-	-	-	R/W	YES	OFF
M1368	Slave ID#9 status on ELC Link network	Y	Y	Y	Y	-	-	-	R/W	YES	OFF
M1369	Slave ID#10 status on ELC Link network	Y	Y	Y	Y	-	-	-	R/W	YES	OFF
M1370	Slave ID#11 status on ELC Link network	Y	Y	Y	Y	-	-	-	R/W	YES	OFF
M1371	Slave ID#12 status on ELC Link network	Y	Y	Y	Y	-	-	-	R/W	YES	OFF
M1372	Slave ID#13 status on ELC Link network	Y	Y	Y	Y	-	-	-	R/W	YES	OFF
M1373	Slave ID#14 status on ELC Link network	Y	Y	Y	Y	-	-	-	R/W	YES	OFF
M1374	Slave ID#15 status on ELC Link network	Y	Y	Y	Y	-	-	-	R/W	YES	OFF
M1375	Slave ID#16 status on ELC Link network	Y	Y	Y	Y	-	-	-	R/W	YES	OFF
M1376	Indicating Slave ID#1 data transaction status on ELC Link	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1377	Indicating Slave ID#2 data transaction status on ELC Link	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1378	Indicating Slave ID#3 data transaction status on ELC Link	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1379	Indicating Slave ID#4 data transaction status on ELC Link	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1380	Indicating Slave ID#5 data transaction status on ELC Link	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1381	Indicating Slave ID#6 data transaction status on ELC Link	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1382	Indicating Slave ID#7 data transaction status on ELC Link	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1383	Indicating Slave ID#8 data transaction status on ELC Link	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1384	Indicating Slave ID#9 data transaction status on ELC Link	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1385	Indicating Slave ID#10 data transaction status on ELC Link	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1386	Indicating Slave ID#11 data transaction status on ELC Link	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1387	Indicating Slave ID#12 data transaction status on ELC Link	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1388	Indicating Slave ID#13 data transaction status on ELC Link	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1389	Indicating Slave ID#14 data transaction status on ELC Link	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1390	Indicating Slave ID#15 data transaction status on ELC Link	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1391	Indicating Slave ID#16 data transaction status on ELC Link	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1392	Slave ID#1 linking error	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1393	Slave ID#2 linking error	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1394	Slave ID#3 linking error	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1395	Slave ID#4 linking error	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1396	Slave ID#5 linking error	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1397	Slave ID#6 linking error	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1398	Slave ID#7 linking error	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1399	Slave ID#8 linking error	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1400	Slave ID#9 linking error	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1401	Slave ID#10 linking error	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1402	Slave ID#11 linking error	Y	Y	Y	Y	OFF	-	-	R	NO	OFF

SM	Function	ELCM -PH -PA	ELC2-PB	ELC2-PC -PE	ELC2 -PA	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1403	Slave ID#12 linking error	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1404	Slave ID#13 linking error	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1405	Slave ID#14 linking error	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1406	Slave ID#15 linking error	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1407	Slave ID#16 linking error	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1408	Indicating reading from Slave ID#1 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1409	Indicating reading from Slave ID#2 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1410	Indicating reading from Slave ID#3 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1411	Indicating reading from Slave ID#4 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1412	Indicating reading from Slave ID#5 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1413	Indicating reading from Slave ID#6 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1414	Indicating reading from Slave ID#7 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1415	Indicating reading from Slave ID#8 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1416	Indicating reading from Slave ID#9 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1417	Indicating reading from Slave ID#10 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1418	Indicating reading from Slave ID#11 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1419	Indicating reading from Slave ID#12 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1420	Indicating reading from Slave ID#13 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1421	Indicating reading from Slave ID#14 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1422	Indicating reading from Slave ID#15 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1423	Indicating reading from Slave ID#16 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1424	Indicating writing to Slave ID#1 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1425	Indicating writing to Slave ID#2 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1426	Indicating writing to Slave ID#3 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1427	Indicating writing to Slave ID#4 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1428	Indicating writing to Slave ID#5 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1429	Indicating writing to Slave ID#6 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1430	Indicating writing to Slave ID#7 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1431	Indicating writing to Slave ID#8 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1432	Indicating writing to Slave ID#9 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1433	Indicating writing to Slave ID#10 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1434	Indicating writing to Slave ID#11 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1435	Indicating writing to Slave ID#12 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1436	Indicating writing to Slave ID#13 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1437	Indicating writing to Slave ID#14 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF

SM	Function	ELCM -PH -PA	ELC2-PB	ELC2-PC -PE	ELC2 -PA	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
M1438	Indicating writing to Slave ID#15 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1439	Indicating writing to Slave ID#16 is completed	Y	Y	Y	Y	OFF	-	-	R	NO	OFF
M1524	Auto-reset Y2 when high speed pulse output completes	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1525	Auto-reset Y3 when high speed pulse output completes	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1534	Y0 being able to designate deceleration time. Has to be used with D1348	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1535	Y2 being able to designate deceleration time. Has to be used with D1349	Y	Y	Y	Y	OFF	-	-	R/W	NO	OFF
M1538	Indicating pause status of Y0	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1539	Indicating pause status of Y1	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1540	Indicating pause status of Y2	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1541	Indicating pause status of Y3	Y	Y	Y	Y	OFF	OFF	-	R/W	NO	OFF
M1580	The absolute position of Delta ASDA-A2 servo is read successfully by means of the instruction DABSR.	V3.2	-	V2.6 V1.4	V2.4	OFF	OFF	OFF	R/W	NO	OFF
M1581	The absolute position of Delta ASDA-A2 servo is not read successfully by means of the instruction DABSR.	V3.2	-	V2.6 V1.4	V2.4	OFF	OFF	OFF	R/W	NO	OFF
M1584	If the left limit switch of CH0 is enabled, it can be triggered either by a rising-edge signal or by a falling-edge signal. (OFF: Rising-edge signal; ON: Falling-edge signal)	V3.2	V3.0	V2.8 V1.4	V2.6	OFF	OFF	-	R/W	NO	OFF
M1585	If the left limit switch of CH1 is enabled, it can be triggered either by a rising-edge signal or by a falling-edge signal. (OFF: Rising-edge signal; ON: Falling-edge signal)	V3.2	V3.0	V2.8 V1.4	V2.6	OFF	OFF	-	R/W	NO	OFF

2.13 S Relay

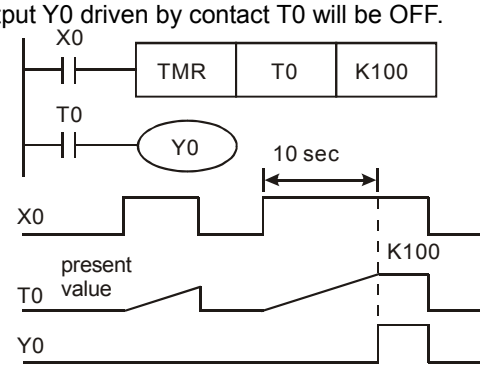
Initial step relay	Starting instruction in Sequential Function Chart (SFC).
Zero return step relay	Returns to zero point when using IST instruction in program. Zero return step relays not used for IST instruction can be used as general step relays.
General purpose step relay	S20 ~ S511, total 492 points (for ELC-PA controllers); S20 ~ S499, total 480 points (for ELC-PV, ELC2-PV controllers); S128 ~ S911, total 784 points (for ELCM-PH/PA, ELC2-PB/PH/PA/PE controllers). General relays in sequential function chart (SFC). They will be cleared if power is lost after running.
Latched step relay	S512 ~ S895, total 384 points (for ELC-PA controllers); S20 ~ S127, total 108 points (for ELCB-PB, ELCM-PH/PA, ELC2-PB/PH/PA/PE controllers); S500 ~ S899, total 400 points (for ELC-PV, ELC2-PV controllers). In sequential function chart (SFC), latched step relay will be saved when power is lost after running. The state of power on after power is lost will be the same as the state before power loss.
Alarm step relay	S896 ~ S1023, total 128 points (for ELC-PA controllers); S900 ~ S1023, total 124 points (for ELC-PV, ELC2-PV controllers); S912 ~ S1023, total 112 points (for ELCM-PH/PA, ELC2-PB/PH/PA/PE controllers). The step relay for alarm uses with alarm drive instruction ANS to the contact for alarm. It is used to record warnings and eliminate external malfunctions.

2.14 T (Timer)

The timer increment in units of 1ms, 10ms and 100ms and the counting method is counting up. When the present value in the timer equals the set value, the associated output coil will be ON. The set value should be a K value in decimal and can be specified by the content of data register D.

The actual set time of the timer = timer resolution × set value
Ex: If the set value is K200 and timer resolution is 10ms, the actual set time in the timer will be 10ms * 200 = 2000ms = 2 sec.

General Timer
For ELC-PA, ELCB-PB, ELCM-PH/PA, and ELC2- PB/PH/PA/PE controllers: The timer executes once when the program reaches END instruction. When TMR instruction is executed, the timer coil will be ON when the timing reaches its preset value.
For ELC-PV and ELC2-PV controllers: The timer executes once when the program reaches TMR instruction. When TMR instruction is executed, the timer coil will be ON when the timing reaches its preset value.
When X0 = ON, TMR instruction is driven. When current value reaches K100, the associated timer contact, T0, turns ON to drive output Y0. If X0 = OFF or the power is off, the current value in T0 will be cleared as 0 and output Y0 driven by contact T0 will be OFF.

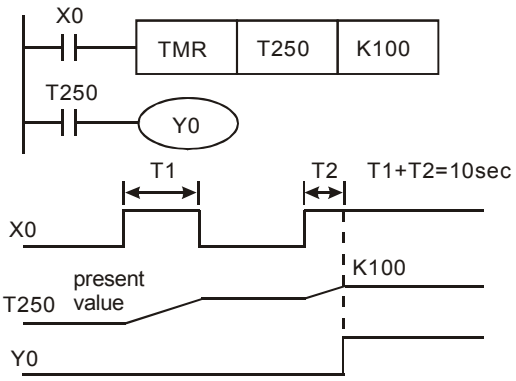


Accumulative Timer

For ELC-PA, ELCB-PB, ELCM-PH/PA, and ELC2-PB/PH/PA/PE controllers: The timer executes once when the program reaches END instruction. When TMR instruction is executed, the timer coil will be ON when the current value reaches its preset value. For accumulative timers, current value will not be cleared when timing is interrupted.

For ELC-PV, ELC2-PV controllers: The timer executes once when the program reaches TMR instruction. When TMR instruction is executed, the timer coil will be ON when the timing reaches its preset value.

Timer T250 will begin timing when X0=ON. If T250 has not reached its preset value by the time X0=OFF, then T0 will pause. When X0=ON, T250 will resume timing from where it was paused.



Timers for Subroutines and Interrupts

Timers for subroutines and interrupts count once when END instruction is met. The associated output coils will be ON if the set value is achieved when End instruction executes. Timers T192~T199 (ELC-PA/PV, ELC2-PV controllers), T184~T199 (ELCM-PH/PA, ELC2-PB/PH/PA/PE controllers) are the only timers that can be used in a subroutine or interrupt. Generals timers used in subroutines and interrupts will not work if the subroutines or interrupts are not executing.

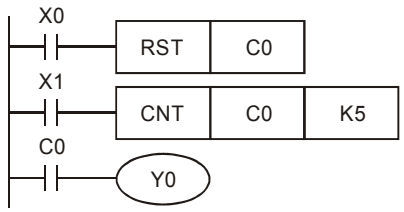
2.15 C (Counter)

Counters will increment their present count value when the input signal transitions from OFF→ON.

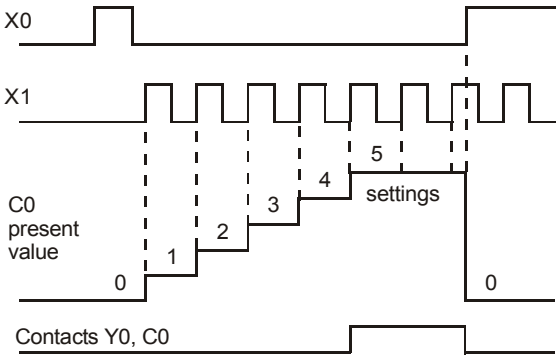
Item	16 bits counters	32 bits counters		
Type	General	General	High speed	
ELCB-PB Counters	C0~C127	-	C235~C238, C241, C242, C244, C246, C247, C249, C251, C252, C254	-
ELC-PA Counters	C0~C199	C200~C234	C235~C242, C244, C246, C247, C249, C251~C254	-
ELC-PV, ELC2-PV Counters	C0~C199	C200~C234	C235~C244, C246~C249, C251~C254	-
ELCM-PH/PA Counters	C0~C199	C200~C231	C232~C242, C245~C254	C243, C244
ELC2-PB/ PH/PA Counters	C0~C199	C200~C232	C233~C242 C245~C254	C243, C244
ELC2-PE Counters	C0~C199	C200~C231	C233~C242 C245~C248 C251~C254	C243, C244
Count direction	Count up	Count up/down		Count up
Range	0~32,767	-2,147,483,648~+2,147,483,647		0~2,147,483,647
Preset value register	Constant K or data register D (Word)	Constant K or data register D (Dword)		
Output operation	Counter will stop when preset value reached	Counter will keep on counting when preset value reached. The count value will become -2,147,483,648 if one more count is added to +2,147,483,647		Counter will keep on counting when preset value is reached. The count value will become 0 if one more count is added to +2,147,483,647
Output contact function	Output Coil will be ON when counter reaches preset value.	Output coil is ON when counter reaches or is above preset value. Output coil is OFF when counter is below preset value.		Output coil is ON when counter reaches or is above preset value
High speed comparison	-	-		Associated devices are activated immediately when preset value is reached
Reset action	The present value will reset to 0 when RST instruction is executed, output coil will be OFF.			
Update method	During every scan	During every scan	Immediate – update is independent of scan time.	

Example:

```
LD      X0
RST     C0
LD      X1
CNT     C0   K5
LD      C0
OUT     Y0
```



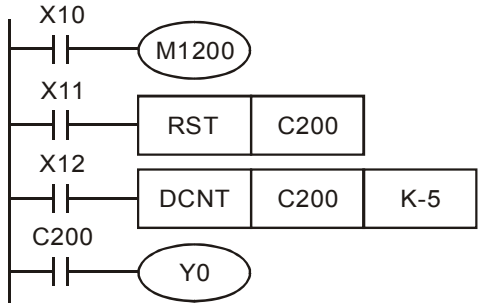
When X0=ON, RST instruction will reset C0. When X1 transitions OFF→ON, C0 will count up (add 1). When C0 reaches the preset value K5, C0 output coil Y0 will = ON and C0 will stop counting and ignore the X1 trigger signal.



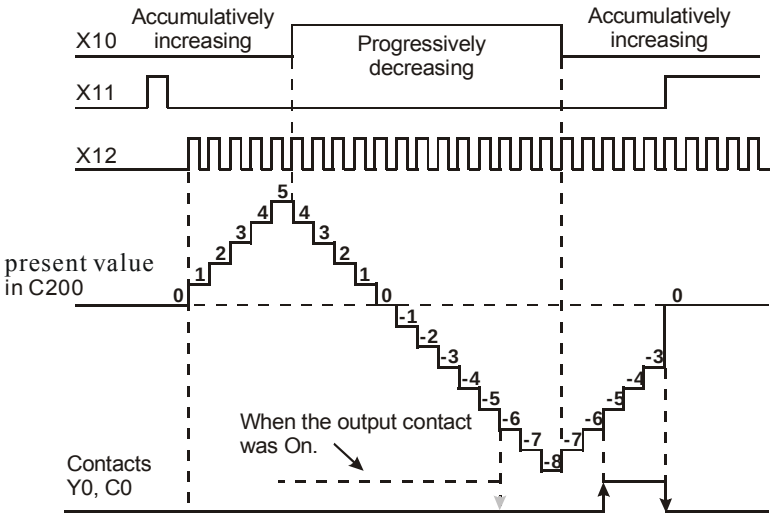
M relays M1200 ~ M1255 are used to set the up/down count direction for C200 ~ C255 respectively. Setting the corresponding M relay ON will set the counter to count down.

Example:

```
LD      X10
OUT     M1200
LD      X11
RST     C200
LD      X12
CNT     C200   K-5
LD      C200
OUT     Y0
```



1. X10 drives M1200 to determine counting direction (up / down) of C200
2. When X11 goes from OFF to ON, RST instruction will be executed and the present value in C200 will be cleared and contact C200 is OFF.
3. When X12 goes from OFF to ON, present value of C200 will count up (plus 1) or count down (minus 1).
4. When present value in C200 changes from K-6 to K-5, the contact C200 will be energized.
5. When present value in C200 changes from K-5 to K-6, the contact of C200 will be reset.
6. If the MOV instruction is applied through ELCSoft to designate a value bigger than the set value to the present value register of C0, next time when X1 goes from OFF to ON, the contact C0 will be ON and present value of C0 will equal setting value.



2.16 High-speed Counters

ELC- PA, ELCB-PB:
ELC High-speed counters can be 1-phase or 2 phases and can count up to a frequency of 20KHz. The table below displays the relation to inputs X0-X5, X10-X11, and counters C235-C255. (ELCB-PB: X0-X3, ELC-PA: X0-X5, ELC-PH: X0-X5, X10, X11). ELC-PA: C253 (2 phase input) High-speed counter can count up to a frequency of 20KHz.

	1-phase input											1-phase 2 inputs				2-phase inputs				
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C246	C247	C249	C250	C251	C252	C253	C254	C255
X0	U/D						U/D			U/D		U	U	U		A	A	B	A	
X1		U/D					R			R		D	D	D		B	B	A	B	
X2			U/D					U/D					R	R			R		R	
X3				U/D				R		S				S					S	
X4					U/D															
X5						U/D														
X10									U/D						U					A
X11											U/D				D					B
Maximum Count Frequency for each counter (Unit: kHz)																				
PB	20	20	10	10	-	-	20	10	-	20	-	20	20	20	-	5	5	-	5	-
PA	20	20	10	10	-	-	20	10	-	20	-	20	20	20	-	5	5	20	5	-

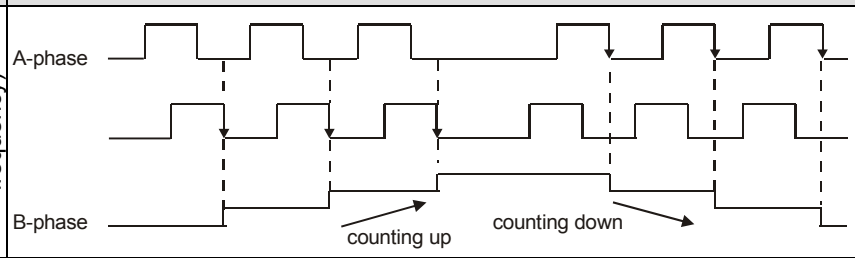
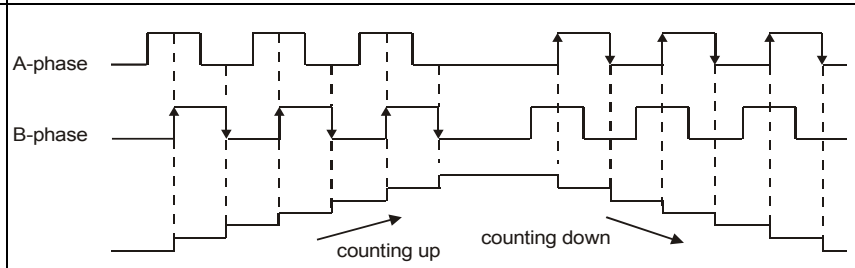
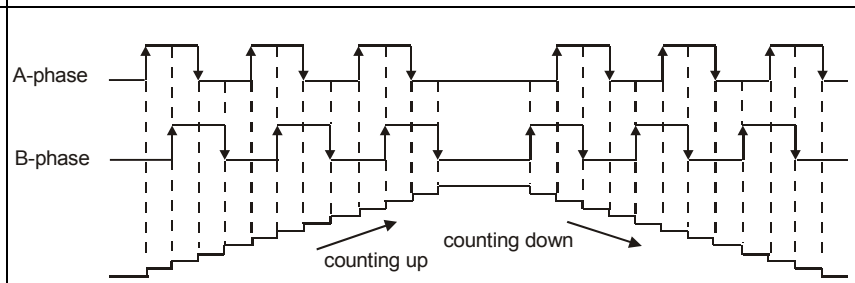
U: Increasing A: A phase input S: Start input
D: Decreasing B: B phase input R: Clear input

Input X5 has two functions:
M1260=OFF C240 is general U/D high-speed counter.
M1260=ON It is Global reset for C235~C239.

Counting mode selection
The high-speed counter uses special D1022 in 2-phase inputs counting mode to select double frequency mode. D1022 content will be loaded in at the first scan time when the ELC switches from STOP to RUN.

D1022	Functions
D1022	Double frequency setting of counter counting method
D1022=K1	Normal frequency mode
D1022=K2	Double frequency mode (factory setting)
D1022=K4	Four times frequency mode

Double frequency mode (↑ , ↓ means the action of counting)

Counting mode		Wave for counting mode
2-phase inputs	1 (normal frequency)	
	2 (double frequency)	
	4 (four times frequency)	

ELC-PV and ELC2-PV:
ELC-PV, ELC2-PV supports high speed counters. C235 ~ C240 are program-interruption 1-phase high speed counter with a total bandwidth of 20kHz, can be used alone with a counting frequency of up to 10kHz. C241 ~ C254 are hardware high speed counter (HHSC). There are four HHSC in ELC-PV, HHSC0 ~ 3. The pulse input frequency of HHSC0 ~ 3 of the ELC-PV, ELC2-PV can reach 200kHz, among which:

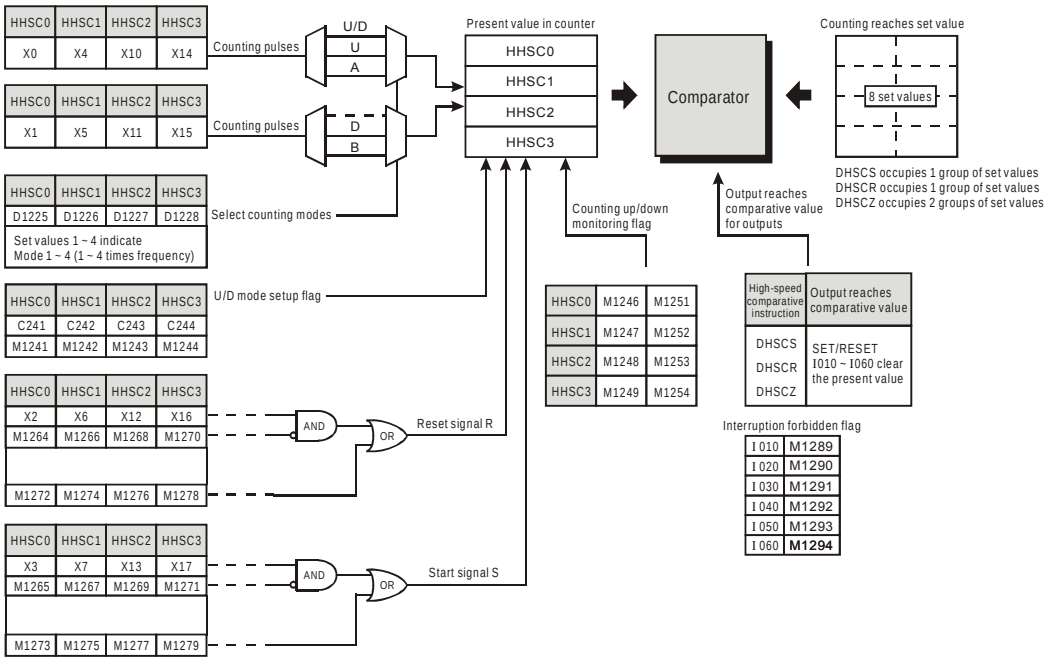
- C241, C246 and C251 share HHSC0
- C242, C247 and C252 share HHSC1
- C243, C248 and C253 share HHSC2
- C244, C249 and C254 share HHSC3

1. Every HHSC can only be designated to one counter by DCNT instruction.
2. There are three counting modes in every HHSC (see the table below):
   a) 1-phase 1 input refers to “pulse/direction” mode.
   b) 1-phase 2 inputs refers to “clockwise/counterclockwise (CW/CCW)” mode.
   c) 2-phase 2 inputs refers to “A-B phase” mode.

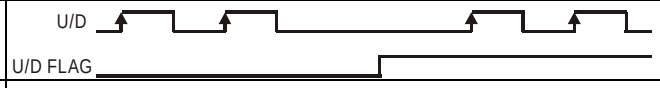
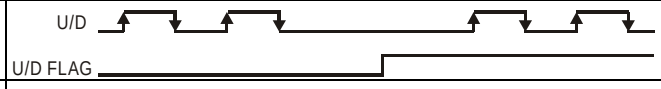
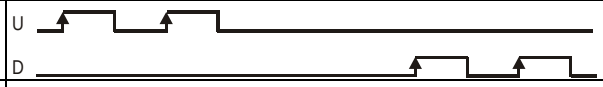
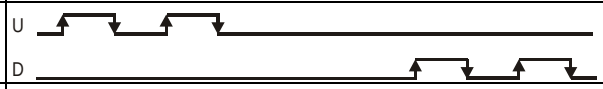
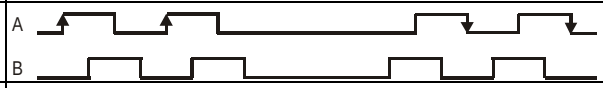
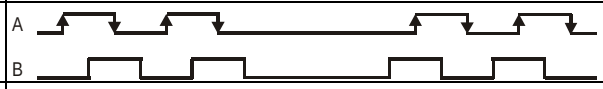
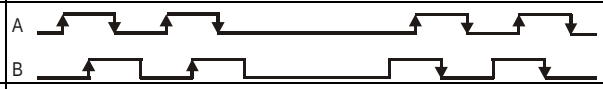
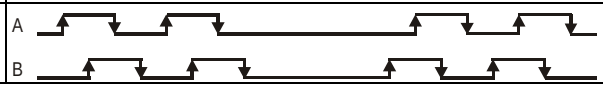
Counter type	Program-interruption high speed counter						Hardware high speed counter											
Type / X	1-phase 1 input						1-phase 1 input				1-phase 2 inputs				2-phase 2 inputs			
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C246	C247	C248	C249	C251	C252	C253	C254
X0	U/D						U/D				U				A			
X1		U/D									D				B			
X2			U/D				R				R				R			
X3				U/D			S				S				S			
X4					U/D			U/D				U				A		
X5						U/D						D				B		
X6								R				R				R		
X7								S				S				S		
X10									U/D				U				A	
X11													D				B	
X12									R				R				R	
X13									S				S				S	
X14										U/D				U				A
X15														D				B
X16										R				R				R
X17										S				S				S
Maximum Count Frequency for each counter (Unit: kHz)																		
PV	10	10	10	10	10	10	200	200	20	20	200	200	20	20	200	200	20	20

U: Progressively increasing input
A: A phase input
S: Input started
B: Progressively decreasing input
B: B phase input
R: Input cleared

3. System structure of the hardware high speed counters:
   a) HHSC0 ~ 3 have reset signals and start signals from external inputs. Settings in M1272, M1274, M1276 and M1278 are reset signals of HHSC0, HHSC1, HHSC2 and HHSC3. Settings in M1273, M1275, M1277 and M1279 are start signals of HHSC0, HHSC1, HHSC2 and HHSC3.
   b) If the external control signal inputs of R and S are not in use, you can set M1264/M1266/M1268/M1270 and M1265/M1267/M1269/M1271 as True and disable the input signals. The corresponding external inputs can be used again as general input points (see the figure below).
   c) When special M is used as a high speed counter, the inputs controlled by START and RESET will be affected by the scan time.



4. Counting modes:
The counting modes of the hardware high-speed counters in the ELC-PV, ELC2-PV can be set in D1225 ~ D1228.

Counting modes		Wave pattern
Type	Set value in special D	Counting up(+1) Counting down(-1)
1-phase 1 input	1 (Normal frequency)	
	2 (Double frequency)	
1-phase 2 inputs	1 (Normal frequency)	
	2 (Double frequency)	
2-phase 2 inputs	1 (Normal frequency)	
	2 (Double frequency)	
2-phase 2 inputs	3 (Triple frequency)	
	4 (4 times frequency)	

ELCM-PH/PA, ELC2-PB/PA/PH/PE:
There are two types of high speed counters provided by ELCM-PH/PA including Software High Speed Counter (SHSC) and Hardware High Speed Counter (HHSC). The same Input point (X) can be designated with only one high speed counter. Double designation on the same input or the same counter will result in syntax error when executing DCNT instruction.

Applicable Software High Speed Counters:

C / X	1-phase input								2 phase 2 input		
	C235	C236	C237	C238	C239	C240	C241	C242	C232*2	C233	C234
X0	U/D								A		
X1		U/D									
X2			U/D						B		
X3				U/D							
X4					U/D					A	
X5						U/D				B	
X6							U/D				A
X7								U/D			B
R/F*3	M1270	M1271	M1272	M1273	M1274	M1275	M1276	M1277	-	-	-
U/D*1	M1235	M1236	M1237	M1238	M1239	M1240	M1241	M1242	-	-	-
Maximum Count Frequency for each counter (Unit: kHz)											
ELCM	10	10	10	10	10	10	10	10	15	5	5
ELC2	10	10	10	10	10	10	10	10	15	5	5

U: Count up D: Count down A: Phase A input B: Phase B input

Note:
1. U/D (Count up/Count down) can be specified by special M. OFF = count up; ON = count down.
2. ELC2-PB/ PH/PE does not support a two-phase two-input counter (C232 with the input points X0 and X2).
3. R/F (Rising edge trigger/ Falling edge trigger) can also be specified by special M. OFF = Rising; ON = Falling.
4. SHSC supports max 10kHz input pulse on single point. Max. 8 counters are applicable in the same time.
5. For 2-phase 2-input counting, (X4, X5) (C233) and (X6, X7) (C234), max 5kHz. (X0,X2) (C232), max 15kHz.
6. 2-phase 2-input counting supports double and 4 times frequency, which is selected in D1022 as the table in next page.

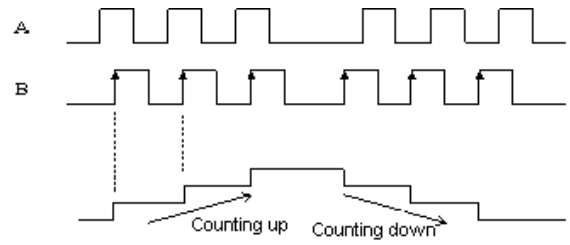
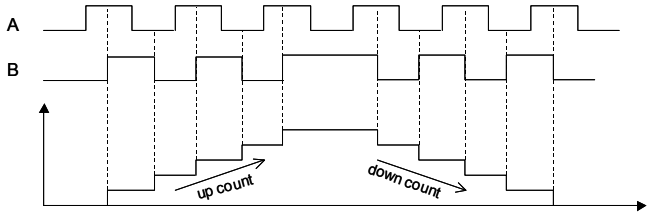
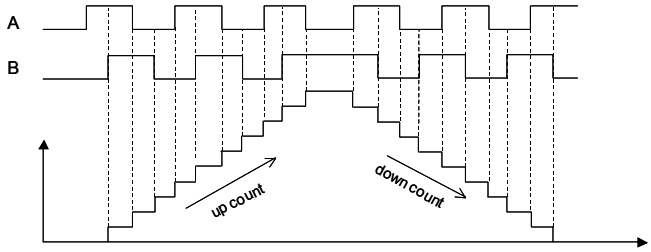
Applicable Hardware High Speed Counters:

C / X	1-phase input		1-phase 2-input						2-phase 2-input			
	C243	C244	C245	C246	C247	C248	C249*2	C250*2	C251	C252	C253	C254
X0	U		U/D	U/D	U	U			A	A		
X1	R		Dir	Dir	D	D			B	B		
X2		U					U/D	U/D			A	A
X3		R					Dir	Dir			B	B
X4				R		R				R		
X5								R				R
Maximum Count Frequency for each counter (Unit: kHz)												
ELCM	100	100	100	100	10	10	100	100	5	5	5	5
ELC2-PB	20	20	20	20	10	10	20	20	5	5	5	5
ELC2-PA	100	100	100	100	10	10	100	100	5	5	5	5
ELC2-PC	100	100	100	100	100	100	100	100	50	50	5	5
ELC2-PE	100	100	100	100	100	100	--	--	50	50	5	5

U: Count up A: Phase A input Dir: Direction signal input
D: Count down B: Phase B input R: Reset signal input

Note:
1. The max frequency of the 1-phase input counters X0 (C243) and X2 (C244) is 100kHz on ELCM-PH/ PA, ELC2-PC/PA and 20kHz on ELC2-PB.
2. ELC2-PE does not support the counters C249 and C250.
3. 2-phase 2-input counting supports double and 4 times frequency, which is selected in D1022 as the table in next page. Please refer to the below table for detailed counting wave form.

D1022	Counting mode

D1022	Counting mode
K1	
K2 (Double Frequency)	
K4 or other value (4 times frequency) (Default)	

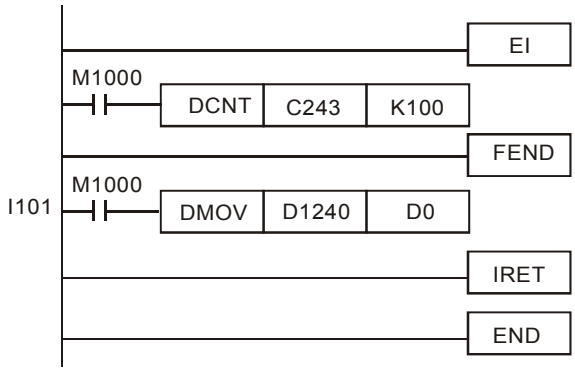
4. ELCM-PH/PA, version 1.0 did not support the single frequency mode. The ELCM version 2.0 and other ELC2 series support the three modes.
5. C243 and C244 support count-up mode only and occupy the associate input points X1 and X3 as reset (“R”) function. If users do not need to apply reset function, set ON the associated special M relays (M1243 and M1244) to disable the reset function.
6. “Dir” refers to direction control function. OFF indicates counting up; ON indicates counting down.
7. When X1, X3, X4 and X5 is applied for reset function and associated external interrupts are disabled, users can define the reset function as Rising/Falling-edge triggered by special M relays

Reset Function	X1	X3	X4	X5
R/F	M1271	M1273	M1274	M1275

8. When X1, X3, X4 and X5 is applied for reset function and external interrupts are applied, the interrupt instructions have the priority in using the input points. In addition, the ELC will move the current data in the counters to the associated data registers below then reset the counters.

Special D	D1241, D1240				D1243, D1242		
Counter	C243	C246	C248	C252	C244	C250	C254
External Interrupt	X1	X4			X3	X5	

Example:

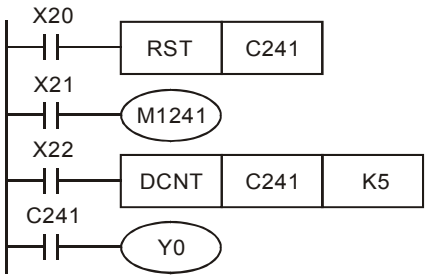


When C243 is counting and external interrupt is triggered from X1(I101), counted value in C243 will be move to (D1241, D1240) immediately then C243 is reset. After this interrupt I101 executes.

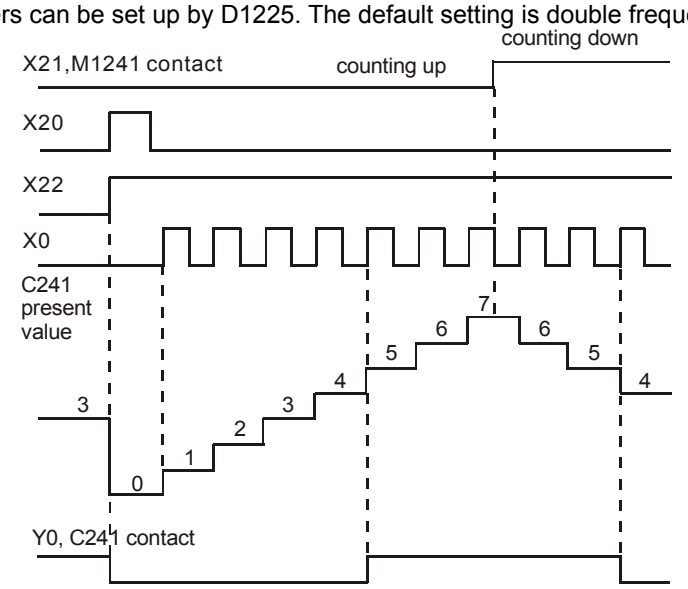
1-phase inputs high-speed counter:

Example:

```
LD      X20
RST     C241
LD      X21
OUT     M1241
LD      X22
DCNT    C241   K5
LD      C241
OUT     Y0
```



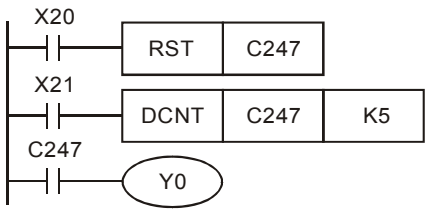
X21 drives M1241 to decide C241 is addition or subtraction.
When X20=ON and RST instruction is executed, clear C241 to 0 and reset output contact to off.
When X22=ON, C241 receives count signal from X0 and counter will count up (+1) or count down (-1).
When counter C241 attains settings K5, C241 will be ON. If there is still signal input for X0, it will keep on counting.
C241 in ELCB-PB and ELC-PA controllers has external input signals to reset X1.
C241 in ELC-PV, ELC2-PV controllers has external input signals to reset X2 and start X3.
The external input contact of reset signal of C241 (HHSC0) in ELC-PV, ELC2-PV controllers is disabled by M1264. The external input contact of start signal is disabled by M1265.
The internal input contact of reset signal of C241 (HHSC0) in ELC-PV, ELC2-PV controllers is disabled by M1272. The internal input contact of start signal is disabled by M1273.
The counting modes (normal frequency or double frequency) of C246 (HHSC0) in ELC-PV, ELC2-PV controllers can be set up by D1225. The default setting is double frequency mode.



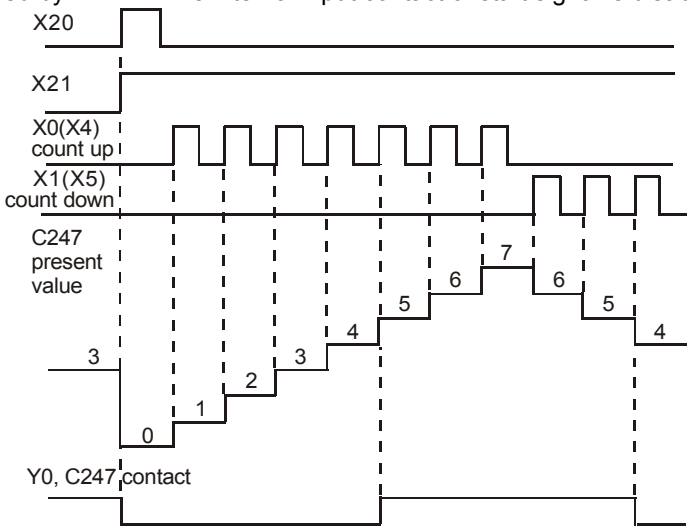
1-phase 2 inputs high-speed counters:

Example:

```
LD      X20
RST     C247
LD      X21
DCNT    C247 K5
LD      C247
OUT     Y0
```



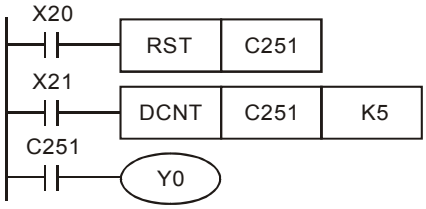
When X20=ON and RST instruction is executed, clear C247 to 0 and reset output contact to off.
When X21=ON, ELCB-PB and ELC-PA controllers C247 receives count signal from X0 input terminal and counter will count up (+1) or receive count signal from X1 input terminal and counter will count down (-1). ELC-PV, ELC2-PV controllers C247 receives count signal from X4 input terminal and counter will count up (+1) or receive count signal from X5 input terminal and counter will count down (-1).
When C247 attains settings K5, C247 will be on. After C247 is ON, if there is counter pulse input, C247 will keep on counting.
C247 in ELCB-PB, ELC-PA controllers has external input signals to reset X2.
C247 in ELC-PV, ELC2-PV controllers has external input signals to reset X6 and start X7.
The counting modes (normal frequency or double frequency) of C247 (HHSC1) in ELC-PV, ELC2-PV controllers can be set up by D1226. The default setting is double frequency mode.
The external input contact of reset signal of C247 (HHSC1) in PV controllers is disabled by M1266.
The external input contact of start signal is disabled by M1267.
The internal input contact of reset signal of C247 (HHSC1) in ELC-PV, ELC2-PV controllers is disabled by M1274. The internal input contact of start signal is disabled by M1275.



2-phase AB input high-speed counter:

Example:

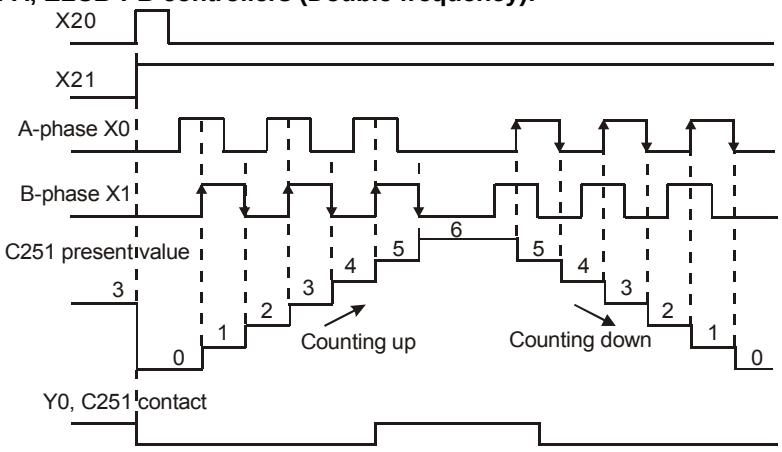
```
LD      X20
RST     C251
LD      X21
DCNT    C251 K5
LD      C251
OUT     Y0
```



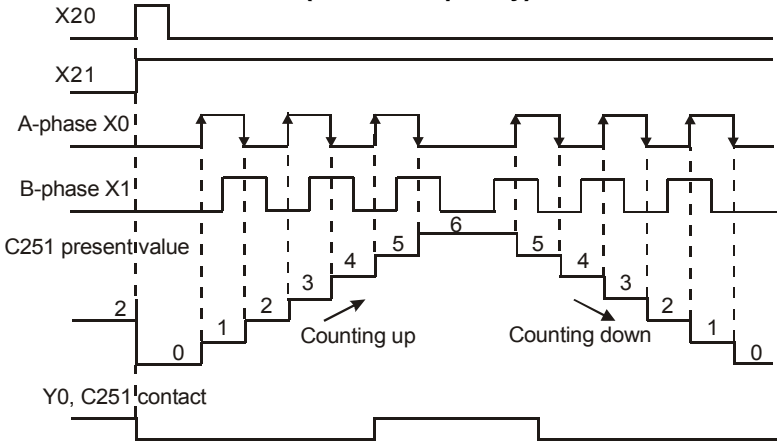
When X20=ON, RST instruction is executed and resets C251 to 0, output contact is reset to off.

C251 receives A phase counting signal of X0 input terminal and B phase counting signal of X1 input terminal to execute add 1 (count up) or subtract 1 (count down) when X21=on.
When counter C251 attains settings K5, C251 contact will be ON. After C251 is ON, if there is counter pulse input, C251 will keep on counting.
In ELCB-PB and ELC-PA controllers frequency can be set to normal, double frequency or four times frequency by D1022 (counting mode setting). Factory setting is double frequency.
C251 in ELC-PV, ELC2-PV controllers has external input signals to reset X2 and start X3.
The counting modes (normal frequency, double frequency, triple frequency or 4 times frequency) of C251 (HHSC0) in ELC-PV, ELC2-PV controllers can be set up by D1225. The default setting is double frequency mode.
The external input contact of reset signal of C246 (HHSC0) in ELC-PV, ELC2-PV controllers is disabled by M1264. The external input contact of start signal is disabled by M1265.
The internal input contact of reset signal of C246 (HHSC0) in ELC-PV, ELC2-PV controllers is disabled by M1272. The internal input contact of start signal is disabled by M1273.

ELC-PA, ELCB-PB controllers (Double frequency):



ELC-PV, ELC2-PV controllers (Double frequency):



2.17 Special Data Register

The special registers (special D) are as shown in the following. Please notice that some equipments with the same number will be different to the different model. In the following chart, the values in the "Type" column are: "R": can only read. "R/W": can read/write. A "-" means it can do nothing. "#" means this is a system setting. You can read the detailed explanation of the setting in the manual.

ELCB-PB, ELC-PA, ELC-PV, and ELC2-PV:

Special D	Function	ELCB -PB	ELC -PA	ELC-PV	ELC 2-PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
D1000	Watchdog timer (WDT) value (Unit: 1ms)	Y	Y	Y	Y	200	-	-	R/W	NO	200
D1001	ELC program version (user can read the ELC program version from this register. For example, D1001 = H XX10 means version 1.0)	Y	Y	Y	Y	-	-	-	R	NO	#
D1002	Program capacity # -> ELC-PV:15,872; ELC-PA: 7,920; ELCB-PB: 3,792; ELC2-PV: 30000	Y	Y	Y	Y	#	-	-	R	NO	#
D1003	Sum of the ELC internal program memory # -> ELC-PV: -15,873; ELC-PA: -7,920; ELCB-PB: -3,792; ELC2-PV: -30000	Y	Y	Y	Y	-	-	-	R	YES	#
D1004	Grammar detective number	Y	Y	Y	Y	0	0	-	R	NO	0
D1008	STSC address when WDT timer is ON	Y	Y	Y	Y	0	-	-	R	NO	0
D1009	Number of LV signal occurrence	Y	Y	-	-	0	-	-	R	YES	0
	Register for SRAM lost data error code	-	-	Y	Y						
D1010	Present scan time (Unit: 0.1ms)	Y	Y	Y	Y	0	0	0	R	NO	0
D1011	Minimum scan time (Unit: 0.1ms)	Y	Y	Y	Y	0	0	0	R	NO	0
D1012	Maximum scan time (Unit: 0.1ms)	Y	Y	Y	Y	0	0	0	R	NO	0
D1015	0~32,767(unit: 0.1ms) addition type of high-speed connection timer	-	Y	Y	Y	0	-	-	R/W	NO	0
D1018	π PI (Low byte)	Y	Y	Y	Y	0FDB	0FDB	0FDB	R/W	NO	0FDB
D1019	π PI (High byte)	Y	Y	Y	Y	4049	4049	4049	R/W	NO	4049
D1020	X0~X7 input filter (unit: ms); modulation range: 2~20ms	Y	Y	Y	Y	10	-	-	R/W	NO	10
D1021	X10~X17 input filter (unit: ms)	Y	-	-	Y	0	-	-	R/W	NO	0
D1022	Double frequency selection for AB phase counter	Y	Y	-	-	#	-	-	R/W	NO	#
D1023	Storing detected pulse width (unit: 0.1ms)	Y	Y	-	-	0	-	-	R/W	NO	0
D1025	Communication error code	Y	Y	Y	Y	0	-	-	R	NO	0
D1026	Pulse number for masking Y0 when M1156 = ON (Low word)	-	-	Y	Y	0	0	-	R/W	NO	0
D1027	Pulse number for masking Y0 when M1156 = ON (High word)	-	-	Y	Y	0	0	-	R/W	NO	0
D1028	Index register E0	Y	Y	Y	Y	0	-	-	R/W	NO	0
D1029	Index register F0	Y	Y	Y	Y	0	-	-	R/W	NO	0
D1030	Output numbers of Y0 pulse (Low word)	Y	Y	-	-	0	-	-	R	NO	0
D1031	Output numbers of Y0 pulse (High word)	Y	Y	-	-	0	-	-	R	NO	0
D1032	Output numbers of Y1 pulse (Low word)	Y	Y	-	-	0	-	-	R	NO	0
D1033	Output numbers of Y1 pulse (High word)	Y	Y	-	-	0	-	-	R	NO	0
D1035	No. of input point X as RUN/STOP	-	-	Y	Y	-	-	-	R/W	YES	0
D1036	COM1 (RS-232) Communications protocol	Y	Y	Y	Y	0086	-	-	R/W	NO	0086
D1037	Repetition time of HKY key	-	-	Y	Y	-	-	-	R/W	NO	0
D1038	Delay time of data response when ELC MPU as slave in RS-485 communication, range: 0 ~ 10,000 (unit: 0.1ms) ELC-PA: delay time for sending the next communication data in ELC LINK (unit for ELC-PA: 1 scan cycle; ELC-PV/ELC2-PV: 0.1ms)	Y	Y	Y	Y	-	-	-	R/W	YES	0
D1039	Constant scan time (ms)	Y	Y	Y	Y	0	-	-	R/W	NO	0
D1040	ON state number 1 of STEP point S	-	Y	Y	Y	0	-	-	R	NO	0
D1041	ON state number 2 of STEP point S	-	Y	Y	Y	0	-	-	R	NO	0
D1042	ON state number 3 of STEP point S	-	Y	Y	Y	0	-	-	R	NO	0
D1043	ON state number 4 of STEP point S	-	Y	Y	Y	0	-	-	R	NO	0

Special D	Function	ELCB -PB	ELC -PA	ELC-PV	ELC 2-PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
D1044	ON state number 5 of STEP point S	-	Y	Y	Y	0	-	-	R	NO	0
D1045	ON state number 6 of STEP point S	-	Y	Y	Y	0	-	-	R	NO	0
D1046	ON state number 7 of STEP point S	-	Y	Y	Y	0	-	-	R	NO	0
D1047	ON state number 8 of STEP point S	-	Y	Y	Y	0	-	-	R	NO	0
D1049	ON number of alarm point	-	Y	Y	Y	0	-	-	R	NO	0
D1050 ↓ D1055	ELC will automatically convert the ASCII data saved in D1070~D1085 to HEX.	Y	Y	Y	Y	0	-	-	R	NO	0
D1056	Present value of PA controller analog input channel 0 (CH0)	-	Y	-	-	0	-	-	R	NO	0
D1057	Present value of PA controller analog input channel 1 (CH1)	-	Y	-	-	0	-	-	R	NO	0
D1058	Enabling X1 interrupt tp get the counting value of C241 (M1056 is On)-Low word	-	-	-	Y	0	0		R	NO	0
D1059	Enabling X1 interrupt tp get the counting value of C241 (M1056 is On)-High word	-	-	-	Y	0	0		R	NO	0
D1061	Error record of non-latched area	Y	-	-	-	-	-	-	R	YES	0
D1062	Average times of AD (CH0, CH1): 2~4	-	Y	-	-	2	-	-	R/W	NO	2
D1064	ELC flash ROM access function for program, password and data. ELC reads flash: H55AA ELC writes flash: HAA55 H55A9/ H99AB/ HA955/ HAB55/ H8888 are added to ELC2-PV.	-	-	Y	Y	0	-	-	R/W	NO	0
D1067	Algorithm error code	Y	Y	Y	Y	0	0	-	R	NO	0
D1068	Lock the algorithm error address	Y	Y	Y	Y	0	-	-	R	NO	0
D1070 ↓ D1085	When the ELC's built-in RS-485 communication instruction receives feedback signals from receiver. The data will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.	Y	Y	Y	Y	0	-	-	R	NO	0
D1086	ELC-ACPGMXFR High word of password setting (Display by HEX value corresponding to ASCII word)	Y	Y	Y	Y	0	-	-	R/W	NO	0
D1087	ELC-ACPGMXFR Low word of password setting (Display by HEX value corresponding to ASCII word)	Y	Y	Y	Y	0	-	-	R/W	NO	0
D1089 ↓ D1099	When the ELC built-in RS-485 communication instruction is executed, the transmitting signals will be stored in the registers D1089~D1099. User can use the contents saved in registers to check the feedback data.	Y	Y	Y	Y	0	-	-	R	NO	0
D1100	Corresponding status after LV signal is enabled	-	-	Y	Y	0	-	-	R/W	NO	0
D1101	Start address of file register	-	Y	Y	Y	-	-	-	R/W	Yes	0
D1102	Copy numbers of file register	-	Y	Y	Y	-	-	-	R/W	Yes	1600
D1103	Set start D number for file register to store (the number should be large than 2000)	-	Y	Y	Y	-	-	-	R/W	Yes	2000
D1104	Parameter index for Accel/Decel pulse output Y0 (corresponds to device D)	Y	Y	-	-	0	0	-	R/W	NO	0
D1109	COM3 (RS-485) Communication protocol	-	-	Y	Y	0086	-	-	R/W	NO	0086
D1110	Average value of the analog input CH0 When the average time is set to 1, D1110 indicates the present value.	-	Y	-	-	0	-	-	R	NO	0
D1111	Average value of the analog input CH When the average time is set to 1, D1111 indicates the present value.	-	Y	-	-	0	-	-	R	NO	0
D1116	Analog output channel 0 (CH 0)	-	Y	-	-	0	0	0	R/W	NO	0
D1117	Analog output channel 1 (CH 1)	-	Y	-	-	0	0	0	R/W	NO	0

Special D	Function	ELCB -PB	ELC -PA	ELC-PV	ELC 2-PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
D1118	Sampling time of analog/digital conversion. Default: 5. Unit: 1ms. Sampling time will be regarded as 5ms if D1118≦5	-	Y	-	-	5	-	-	R/W	NO	5
D1120	COM2 (RS-485) communication protocol	Y	Y	Y	Y	0086	-	-	R/W	NO	0086
D1121	ELC communication address (the address that save the ELC communication address)	Y	Y	Y	Y	-	-	-	R/W	Yes	1
D1122	Residual words of transmitting data	Y	Y	Y	Y	0	0	-	R	NO	0
D1123	Residual words of receiving data	Y	Y	Y	Y	0	0	-	R	NO	0
D1124	Start character definition (STX)	Y	Y	Y	Y	003A	-	-	R/W	NO	003A
D1125	First ending character definition (ETX1)	Y	Y	Y	Y	000D	-	-	R/W	NO	000D
D1126	Second ending character definition (ETX2)	Y	Y	Y	Y	000A	-	-	R/W	NO	000A
D1127	RS instruction, interrupt request when receiving specified data character (I150)	Y	-	-	-	0	-	-	R/W	NO	0
	Number of pulses for ramp-up operation of positioning instruction (Low word)	-	-	Y	Y	0	-	-	R	NO	0
D1128	Number of pulses for ramp-up operation of positioning instruction (High word)	-	-	Y	Y	0	-	-	R	NO	0
D1129	RS-485 time-out setting (ms)	Y	Y	Y	Y	0	-	-	R/W	NO	0
D1130	MODBUS return error code record	Y	Y	Y	Y	0	-	-	R	NO	0
D1131	Low 16 bytes of high-speed counter value extracted by interruption I501	-	Y	-	-	0	-	-	R	NO	0
	Input/output percentage value of CH0(Y0,Y1) close loop control	-	-	Y	Y	100	-	-	R/W	NO	100
D1132	High 16 bytes of high-speed counter value extracted by interruption I501	-	Y	-	-	0	-	-	R	NO	0
	Input/output percentage value of CH1(Y2,Y3) close loop control	-	-	Y		100	-	-	R/W	NO	100
D1133	Special high-speed pulse output Y0 (50KHz) register (D) index	-	Y	-	-	0	-	-	R/W	NO	0
	Number of pulses for ramp-down operation of positioning instruction (Low word)	-	-	Y	Y	0	-	-	R	NO	0
D1134	Number of pulses for ramp-down operation of positioning instruction (High word)	-	-	Y	Y	0	-	-	R/W	NO	0
D1135	Pulse number for masking Y2 when M1158 = ON (Low word)	-	-	-	Y	0	0	-	R/W	NO	0
D1136	Pulse number for masking Y2 when M1158 = ON (High word)	-	-	-	Y	0	0	-	R/W	NO	0
D1137	Address of operator error occurs	Y	Y	Y	Y	0	0	-	R	NO	0
D1140	Special expansion module number, maximum is 8 modules	Y	Y	Y	Y	0	-	-	R	NO	0
D1142	Input points (X) of expansion module	Y	Y	Y	Y	0	-	-	R	NO	0
D1143	Output points (Y) of expansion module	Y	Y	Y	Y	0	-	-	R	NO	0
D1144	The instruction DRVI calculates in advance the value in the data register for Y0.	-	-	V2.0	-	0	-	-	R/W	NO	0
	Parameter index for Accel/Decel pulse output Y0 of adjustable slope (corresponds to component D)	-	Y	-	-	0	-	-	R/W	NO	0
D1145	Number of left-side special expansion modules (max. 8)	-	-	Y	-	0	-	-	R	NO	0
D1150	Table count register in multi-group setting comparison mode of DHSZ command	-	-	Y	Y	0	0	0	R	NO	0
D1151	Table counting register for DHSZ multiple set values comparison mode	-	-	Y	Y	0	0	0	R	NO	0
D1152	High word of changed D value for DHSZ instruction	-	-	Y	Y	0	0	0	R	NO	0
D1153	Low word of changed D value for DHSZ instruction	-	-	Y	Y	0	0	0	R	NO	0
D1154	Recommended Interval of accelerated time (10~32767 ms) of Accel/Decel pulse output Y0 of adjustable slope	-	Y	-	-	200	-	-	R/W	NO	200

Special D	Function	ELCB -PB	ELC -PA	ELC-PV	ELC 2-PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
	Pulse number for masking Y4 (Low word)	-	-	V2.0	Y	0	0	-	R/W	NO	0
D1155	Recommended Interval of decelerated time (-1~ -32700 ms) of Accel/Decel pulse output Y0 of adjustable slope	-	Y	-	-	-1000	-	-	R/W	NO	-1000
	Pulse number for masking Y4 (High word)	-	-	V2.0	Y	0	0	-	R/W	NO	0
D1156	Designated special D for RTMU, RTMD instructions (K0)	-	-	Y	Y	0	-	-	R/W	NO	0
D1157	Designated special D for RTMU, RTMD instructions (K1)	-	-	Y	Y	0	-	-	R/W	NO	0
D1158	Designated special D for RTMU, RTMD instructions (K2)	-	-	Y	Y	0	-	-	R/W	NO	0
D1159	Designated special D for RTMU, RTMD instructions (K3)	-	-	Y	Y	0	-	-	R/W	NO	0
D1160	Designated special D for RTMU, RTMD instructions (K4)	-	-	Y	Y	0	-	-	R/W	NO	0
	Present pulse output frequency of CH0 (Y0/Y1) (Low word)	-	-	-	V1.4	0	-	-	R/W	NO	0
D1161	Designated special D for RTMU, RTMD instructions (K5)	-	-	Y	Y	0	-	-	R/W	NO	0
	Present pulse output frequency of CH0 (Y0/Y1) (High word)	-	-	-	V1.4	0	-	-	R/W	NO	0
D1162	Designated special D for RTMU, RTMD instructions (K6)	-	-	Y	Y	0	-	-	R/W	NO	0
	Present pulse output frequency of CH1 (Y2/Y3) (Low word)	-	-	-	V1.4	0	-	-	R/W	NO	0
D1163	Designated special D for RTMU, RTMD instructions (K7)	-	-	Y	Y	0	-	-	R/W	NO	0
D1163	Present pulse output frequency of CH1 (Y2/Y3) (High word)	-	-	-	V1.4	0	-	-	R/W	NO	0
D1164	Designated special D for RTMU, RTMD instructions (K8)	-	-	Y	Y	0	-	-	R/W	NO	0
	Present pulse output frequency of CH2 (Y4/Y5) (Low word)	-	-	-	V1.4	0	-	-	R/W	NO	0
D1165	Designated special D for RTMU, RTMD instructions (K9)	-	-	Y	Y	0	-	-	R/W	NO	0
	Present pulse output frequency of CH2 (Y4/Y5) (High word)	-	-	-	V1.4	0	-	-	R/W	NO	0
D1166	Exchange of X10 rising/falling-edge count modes (for ELC-PH only)	-	Y	-	-	0	-	-	R/W	NO	0
	Present pulse output frequency of CH3 (Y6/Y7) (Low word)	-	-	-	V1.4	0	-	-	R/W	NO	0
D1167	Present pulse output frequency of CH3 (Y6/Y7) (High word)	-	-	-	V1.4	0	-	-	R/W	NO	0
D1168	Interruption request (I150) for receiving specific word in RS instruction (COM2)	-	Y	Y	Y	0	-	-	R/W	NO	0
D1169	Interruption request (I160) for receiving specific length in RS instruction (COM2)	-	-	Y	Y	0	-	-	R/W	NO	0
D1170	PC value when executing single step	-	-	Y	Y	0	0	0	R	NO	0
D1172	2-phase pulse output frequency (12Hz~20KHz)	-	Y	-	-	0	-	-	R/W	NO	0
	X4 speed detecting time setting	-	-	Y	Y	0	0	-	R/W	NO	0
D1173	2-phase pulse output mode selection (K1and K2)	-	Y	-	-	0	-	-	R/W	NO	0
	X4 speed detecting value	-	-	Y	Y	0	0	-	R/W	NO	0
D1174	Target number for 2-phase pulse outputs (low 16-bit)	-	Y	-	Y	0	-	-	R/W	NO	0
	X10 speed detecting time	-	-	Y	Y	0	0	-	R/W	NO	0
D1175	Target number for 2-phase pulse outputs (high 16-bit)	-	Y	-	-	0	-	-	R/W	NO	0
	X10 speed detecting value	-	-	Y	Y	0	0	-	R/W	NO	0
D1176	Present output number of 2-phase pulse (low 16-bit)	-	Y	-	-	0	-	-	R/W	NO	0
D1176	X14 speed detecting time	-	-	Y	Y	0	0	-	R/W	NO	0
D1177	Present output number of 2-phase pulse (high 16-bit)	-	Y	-	-	0	-	-	R/W	NO	0
	X14 speed detecting value	-	-	Y	Y	0	0	-	R/W	NO	0

Special D	Function	ELCB -PB	ELC -PA	ELC-PV	ELC 2-PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
D1178	VR0 Variable resistor value	-	Y	Y	Y	0	-	-	R	NO	0
D1179	VR1 Variable resistor value	-	Y	Y	Y	0	-	-	R	NO	0
D1180	When X2 interruption (I201) occurs, immediately extracting the low 16 bytes from X0 high-speed counting value.	-	V1.8	-	-	0	0	-	R/W	NO	0
	Enabling X2 to get the counting value of the high-speed counter C241 (M1057 is On)(Low word)	-	-	-	Y	0	0	-	R	NO	0
D1181	When X2 interruption (I201) occurs, immediately extracting the high 16 bytes from X0 high-speed counting value.	-	V1.8	-	-	0	0	-	R/W	NO	0
	Enabling X2 to get the counting value of the high-speed counter C241 (M1057 is On)(High word)	-	-	-	Y	0	0	-	R	NO	0
D1182	Index register E1	-	Y	Y	Y	0	-	-	R/W	NO	0
D1183	Index register F1	-	Y	Y	Y	0	-	-	R/W	NO	0
D1184	Index register E2	-	Y	Y	Y	0	-	-	R/W	NO	0
D1185	Index register F2	-	Y	Y	Y	0	-	-	R/W	NO	0
D1186	Index register E3	-	Y	Y	Y	0	-	-	R/W	NO	0
D1187	Index register F3	-	Y	Y	Y	0	-	-	R/W	NO	0
D1188	Index register E4	-	-	Y	Y	0	-	-	R/W	NO	0
D1189	Index register F4	-	-	Y	Y	0	-	-	R/W	NO	0
D1190	Index register E5	-	-	Y	Y	0	-	-	R/W	NO	0
D1191	Index register F5	-	-	Y	Y	0	-	-	R/W	NO	0
D1192	Index register E6	-	-	Y	Y	0	-	-	R/W	NO	0
D1193	Index register F6	-	-	Y	Y	0	-	-	R/W	NO	0
D1194	Index register E7	-	-	Y	Y	0	-	-	R/W	NO	0
D1195	Index register F7	-	-	Y	Y	0	-	-	R/W	NO	0
D1196	Content in the display	-	Y	-	-	0	-	-	R/W	NO	0
D1197	The refresh time for 7-segment displayer (Unit: 100ms)	-	Y	-	-	5	-	-	R/W	NO	5
D1198	When X3 interruption (I301) occurs, immediately extracting the low 16 byte from X1 high-speed counting value.	-	V1.8	-	-	0	-	-	R	NO	0
	Enabling X3 to get the counting value of the high-speed counter C241 (M1058 is On) (Low word)	-	-	-	Y	0	0	-	R	NO	0
D1199	When X3 interruption (I301) occurs, immediately extracting the high 16 byte from X1 high-speed counting value.	-	V1.8	-	-	0	-	-	R	NO	0
	Enabling X3 to get the counting value of the high-speed counter C241 (M1058 is On) (High word)	-	-	-	Y	0	0	-	R	NO	0
D1200	Start address of M0~M999 auxiliary relay latched	-	Y	Y	Y	-	-	-	R/W	Yes	512
D1201	End address of M0~M999 auxiliary relay latched	-	Y	Y	Y	-	-	-	R/W	Yes	999
D1202	Start address of M2000~M4095 auxiliary relay latched	-	Y	Y	Y	-	-	-	R/W	Yes	2000
D1203	End address of M2000~M4095 auxiliary relay latched	-	Y	Y	Y	-	-	-	R/W	Yes	4095
D1204	Start latched address for 100ms timers T0 ~ T199	-	-	Y	Y	-	-	-	R/W	YES	H'FFFF
D1205	End latched address for 100ms timers T0 ~ T199	-	-	Y	Y	-	-	-	R/W	YES	H'FFFF
D1206	Start latched address for 10ms timers T200 ~ T239	-	-	Y	Y	-	-	-	R/W	YES	H'FFFF
D1207	End latched address for 10ms timers T200 ~ T239	-	-	Y	Y	-	-	-	R/W	YES	H'FFFF
D1208	Start latched address of 16-bit counter C0~C199	-	Y	Y	Y	-	-	-	R/W	Yes	96
D1209	End latched address of 16-bit counter C0~C199	-	Y	Y	Y	-	-	-	R/W	Yes	199
D1210	Start latched address of 32-bit counter C200~C234	-	Y	Y	Y	-	-	-	R/W	Yes	216
D1211	End latched address of 32-bit counter C200~C234	-	Y	Y	Y	-	-	-	R/W	Yes	234
D1212	Start latched address of 32-bit	-	Y	Y	Y	-	-	-	R/W	Yes	235

Special D	Function	ELCB -PB	ELC -PA	ELC-PV	ELC 2-PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
	high-speed counter C235~C255										
D1213	End latched address of 32-bit high-speed counter C235~C255	-	Y	Y	Y	-	-	-	R/W	Yes	255
D1214	Start latched address of step point S0~S1023	-	Y	Y	Y	-	-	-	R/W	Yes	512
D1215	End latched address of step point S0~S1023	-	Y	Y	Y	-	-	-	R/W	Yes	895
D1216	Start latched address of register D0~D999	-	Y	Y	Y	-	-	-	R/W	Yes	200
D1217	End latched address of register D0~D999	-	Y	Y	Y	-	-	-	R/W	Yes	999
D1218	Start latched address of register D2000~D4999	-	Y	Y	Y	-	-	-	R/W	Yes	2000
D1219	End latched address of register D2000~D4999	-	Y	Y	Y	-	-	-	R/W	Yes	4999
D1220	Phase of the 1st group pulse output CH0 (Y0, Y1)	-	-	Y	Y	0	-	-	R/W	NO	0
D1221	Phase of the 2nd group pulse output CH1 (Y2, Y3)	-	-	Y	Y	0	-	-	R/W	NO	0
D1222	Time difference between direction signal and pulse output for the 1st group pulse CH0 (Y0, Y1) in DRVI, DDRVI, DRVA, DDRVA, PLSV, DPLSV	-	-	Y	Y	0	-	-	R/W	NO	0
D1223	Time difference between direction signal and pulse output for the 2nd group pulse CH1 (Y2, Y3) in DRVI, DDRVI, DRVA, DDRVA, PLSV, DPLSV	-	-	Y	Y	0	-	-	R/W	NO	0
D1225	Counting mode of the counter HHSC0	-	-	Y	Y	2	-	-	R/W	NO	2
D1226	Counting mode of the counter HHSC1	-	-	Y	Y	2	-	-	R/W	NO	2
D1227	Counting mode of the counter HHSC2	-	-	Y	Y	2	-	-	R/W	NO	2
D1228	Counting mode of the counter HHSC3	-	-	Y	Y	2	-	-	R/W	NO	2
D1129	Phase of the 3rd group pulse output CH2 (Y4, Y5)	-	-	Y	Y	0	-	-	R/W	NO	0
D1130	Phase of the 4th group pulse output CH3 (Y6, Y7)	-	-	Y	Y	0	-	-	R/W	NO	0
D1232	Output pulse number for ramp-down stop when CH0 mark sensor receives signals. (Low Word).	-	-	Y	Y	0	-	-	R/W	NO	0
D1233	Output pulse number for ramp-down stop when CH0 mark sensor receives signals. (High Word).	-	-	Y	Y	0	-	-	R/W	NO	0
D1234	Output pulse number for ramp-down stop when CH1 mark sensor receives signals (Low Word).	-	-	Y	Y	0	-	-	R/W	NO	0
D1235	Output pulse number for ramp-down stop when CH1 mark sensor receives signals (High Word).	-	-	Y	Y	0	-	-	R/W	NO	0
D1236	Output pulse number for ramp-down stop when CH2 mark sensor receives signals (Low Word)	-	-	Y	Y	0	-	-	R/W	NO	0
D1237	Output pulse number for ramp-down stop when CH2 mark sensor receives signals (High Word)	-	-	Y	Y	0	-	-	R/W	NO	0
D1238	Output pulse number for ramp-down stop when CH3 mark sensor receives signals (Low Word)	-	-	Y	Y	0	-	-	R/W	NO	0
D1239	Output pulse number for ramp-down stop when CH3 mark sensor receives signals (High Word)	-	-	Y	Y	0	-	-	R/W	NO	0
D1240	The low 16 bits of the end frequency of CH0 (available when the acceleration and the deceleration are separate)	-	-	Y	Y	0	0	-	R/W	NO	0
D1241	The high 16 bits of the end frequency of CH0 (available when the acceleration and the deceleration are separate)	-	-	Y	Y	0	0	-	R/W	NO	0
D1244	Number of idle speed output from CH0	-	-	Y	Y	0	0	-	R/W	NO	0

Special D	Function	ELCB -PB	ELC -PA	ELC-PV	ELC 2-PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
	in the instruction DCLLM										
D1245	Number of idle speed output from CH1 in the instruction DCLLM	-	-	Y	Y	0	0	-	R/W	NO	0
D1246	Number of idle speed output from CH2 in the instruction DCLLM	-	-	Y	Y	0	0	-	R/W	NO	0
D1247	Number of idle speed output from CH3 in the instruction DCLLM	-	-	Y	Y	0	0	-	R/W	NO	0
D1249	Communication timeout of COM1 instruction (unit: 1ms; the maximum value is 50ms; the value less than 50ms is count as 50ms.) (Only the instruction MODRW and RS are supported.) RS: 0 indicates that the timeout is not set.	-	-	-	Y	0	-	-	R/W	NO	0
D1250	COM1 (RS-232) communication error code (only applicable for MODRW/RS instruction)	-	-	-	Y	0	-	-	R/W	NO	0
D1256 ↓ D1295	MODRW instruction of COM2 (RS-485) is built-in. The characters sent during execution are saved in D1256~D1295. User can check according to the content of these registers.	Y	Y	Y	Y	0	-	-	R	NO	0
D1296 ↓ D1311	MODRW instruction of COM2 (RS-485) is built-in. The ELC system will convert ASCII content of the register to HEX and save it in D1296 – D1311.	Y	Y	Y	Y	0	-	-	R	NO	0
D1312	Number of times the instruction ZRN searches for Z phase and the number of displacement	-	-	-	Y	0	0	-	R/W	NO	0
D1313	Perpetual calendar (RTC) second 00~59	-	Y	Y	Y	0	-	-	R/W	NO	0
D1314	Perpetual calendar (RTC) minute 00~59	-	Y	Y	Y	0	-	-	R/W	NO	0
D1315	Perpetual calendar (RTC) hour 00~23	-	Y	Y	Y	0	-	-	R/W	NO	0
D1316	Perpetual calendar (RTC) day 01~31	-	Y	Y	Y	0	-	-	R/W	NO	1
D1317	Perpetual calendar (RTC) month 01~12	-	Y	Y	Y	0	-	-	R/W	NO	1
D1318	Perpetual calendar (RTC) week 1~7	-	Y	Y	Y	0	-	-	R/W	NO	6
D1319	Perpetual calendar (RTC) year 00~99	-	Y	Y	Y	0	-	-	R/W	NO	0
D1320	ID of the 1st right-side expansion module	-	-	-	Y	0	-	-	R	NO	0
D1321	ID of the 2nd right-side expansion module	-	-	-	Y	0	-	-	R	NO	0
D1322	ID of the 3rd right-side expansion module	-	-	-	Y	0	-	-	R	NO	0
D1323	ID of the 4th right-side expansion module	-	-	-	Y	0	-	-	R	NO	0
D1324	ID of the 5th right-side expansion module	-	-	-	Y	0	-	-	R	NO	0
D1325	ID of the 6th right-side expansion module	-	-	-	Y	0	-	-	R	NO	0
D1326	ID of the 7th right-side expansion module	-	-	-	Y	0	-	-	R	NO	0
D1327	ID of the 8th right-side expansion module	-	-	-	Y	0	-	-	R	NO	0
D1328	Low word of offset pulse the 1st group pulses CH0 (Y0, Y1)	-	-	Y	Y	-	-	-	R/W	NO	0
D1329	High word of offset pulse the 1st group pulses CH0 (Y0, Y1)	-	-	Y	Y	0	-	-	R/W	NO	0
D1330	Low word of offset pulse the 2nd group pulses CH1 (Y2, Y3)	-	-	Y	Y	0	-	-	R/W	NO	0
D1331	High word of offset pulse the 2nd group pulses CH1 (Y2, Y3)	-	-	Y	Y	0	-	-	R/W	NO	0
D1332	Low word of the remaining number of pulses of the 1st group pulses CH0 (Y0, Y1)	-	-	Y	Y	0	-	-	R	NO	0
D1333	High word of the remaining number of pulses of the 1st group pulses CH0 (Y0, Y1)	-	-	Y	Y	0	-	-	R	NO	0
D1334	Low word of the remaining number of pulses of the 2nd group pulses CH1 (Y2, Y3)	-	-	Y	Y	0	-	-	R	NO	0

Special D	Function	ELCB -PB	ELC -PA	ELC-PV	ELC 2-PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
D1335	High word of the remaining number of pulses of the 2nd group pulses CH1 (Y2, Y3)	-	-	Y	Y	0	-	-	R	NO	0
D1336	Low word of the present value of the 1st group pulses CH0 (Y0, Y1)	-	-	Y	Y	-	-	-	R	YES	0
D1337	High word of the present value of the 1st group pulses CH0 (Y0, Y1)	-	-	Y		-	-	-	R	YES	0
D1338	Low word of the present value of the 2nd group pulses CH1 (Y2, Y3)	-	-	Y	Y	-	-	-	R	YES	0
D1339	High word of the present value of the 2nd group pulses CH1 (Y2, Y3)	-	-	Y	Y	-	-	-	R	YES	0
D1340	start/end frequency of the 1st group pulse output CH0 (Y0, Y1)	-	-	Y	Y	-	-	-	R/W	YES	200
D1341	Low word of max. output frequency	-	-	Y	Y	-	-	-	R	YES	H'04D0
D1342	High word of max. output frequency	-	-	Y	Y	-	-	-	R	YES	3
D1343	Ramp up/down time of the 1st group pulse output CH0 (Y0, Y1):	-	-	Y	Y	-	-	-	R/W	YES	100
D1344	Low word of the number of compensation pulses of the 1st group pulses CH0 (Y0, Y1)	-	-	Y	Y	-	-	-	R/W	YES	0
D1345	High word of the number of compensation pulses of the 1st group pulses CH0 (Y0, Y1)	-	-	Y	Y	-	-	-	R/W	YES	0
D1346	Low word of the number of compensation pulses of the 2nd group pulses CH1 (Y2, Y3)	-	-	Y	Y	-	-	-	R/W	YES	0
D1347	High word of the number of compensation pulses of the 2nd group pulses CH1 (Y2, Y3)	-	-	Y	Y	-	-	-	R/W	YES	0
D1348	CH0 pulse output. When M1534 = ON, it refers to the deceleration time	-	-	Y	Y	-	-	-	R/W	YES	100
D1349	CH1 pulse output. When M1535 = ON, it refers to the deceleration time	-	-	Y	Y	-	-	-	R/W	YES	100
D1350	CH2 pulse output. When M1536 = ON, it refers to the deceleration time	-	-	Y	Y	-	-	-	R/W	YES	100
D1351	CH3 pulse output. When M1537 = ON, it refers to the deceleration time	-	-	Y	Y	-	-	-	R/W	YES	100
D1352	Start/end frequency of the 2nd group pulse output CH1 (Y2, Y3)	-	-	Y	Y	-	-	-	R/W	YES	200
D1353	Acceleration/deceleration time of the 2nd group pulse output CH1 (Y2, Y3)	-	-	Y	Y	-	-	-	R/W	YES	100
D1354	Scan cycle for the ELC link (unit: 1ms) PS1: The maximum value is K32000 PS2: K0: The ELC link stops or the first detection is complete.	-	-	-	Y	0	0	0	R	NO	0
D1355	Starting reference for Master to read from Slave ID#1	-	Y	Y	Y	-	-	-	R/W	NO	1064
D1356	Starting reference for Master to read from Slave ID#2	-	Y	Y	Y	-	-	-	R/W	NO	1064
D1357	Starting reference for Master to read from Slave ID#3	-	Y	Y	Y	-	-	-	R/W	NO	1064
D1358	Starting reference for Master to read from Slave ID#4	-	Y	Y	Y	-	-	-	R/W	NO	1064
D1359	Starting reference for Master to read from Slave ID#5	-	Y	Y	Y	-	-	-	R/W	NO	1064
D1360	Starting reference for Master to read from Slave ID#6	-	Y	Y	Y	-	-	-	R/W	NO	1064
D1361	Starting reference for Master to read from Slave ID#7	-	Y	Y	Y	-	-	-	R/W	NO	1064
D1362	Starting reference for Master to read from Slave ID#8	-	Y	Y	Y	-	-	-	R/W	NO	1064
D1363	Starting reference for Master to read from Slave ID#9	-	Y	Y	Y	-	-	-	R/W	NO	1064
D1364	Starting reference for Master to read from Slave ID#10	-	Y	Y	Y	-	-	-	R/W	NO	1064
D1365	Starting reference for Master to read from Slave ID#11	-	Y	Y	Y	-	-	-	R/W	NO	1064

Special D	Function	ELCB -PB	ELC -PA	ELC-PV	ELC 2-PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
D1366	Starting reference for Master to read from Slave ID#12	-	Y	Y	Y	-	-	-	R/W	NO	1064
D1367	Starting reference for Master to read from Slave ID#13	-	Y	Y	Y	-	-	-	R/W	NO	1064
D1368	Starting reference for Master to read from Slave ID#14	-	Y	Y	Y	-	-	-	R/W	NO	1064
D1369	Starting reference for Master to read from Slave ID#15	-	Y	Y	Y	-	-	-	R/W	NO	1064
D1370	Starting reference for Master to read from Slave ID#16	-	Y	Y	Y	-	-	-	R/W	NO	1064
D1371	Time unit of PWM Y0 pulse output when M1070=ON	-	-	Y	Y	1	-	-	R/W	NO	1
D1372	Time unit of PWM Y2 pulse output when M1071=ON	-	-	Y	Y	1	-	-	R/W	NO	1
D1373	Time unit of PWM Y4 pulse output when M1530=ON	-	-	Y	Y	1	-	-	R/W	NO	1
D1374	Time unit of PWM Y6 pulse output when M1531=ON	-	-	Y	Y	1	-	-	R/W	NO	1
D1375	Low word of the present value of the 3rd group pulses CH2 (Y4, Y5)	-	-	Y	Y	-	-	-	R/W	YES	0
D1376	High word of the present value of the 3rd group pulses CH2 (Y4, Y5)	-	-	Y	Y	-	-	-	R/W	YES	0
D1377	Low word of the present value of the 4th group pulses CH3 (Y6, Y7)	-	-	Y	Y	-	-	-	R/W	YES	0
D1378	High word of the present value of the 4th group pulses CH3 (Y6, Y7)	-	-	Y	Y	-	-	-	R/W	YES	0
D1379	Start frequency of the 1st section and end frequency of the last section for the 3rd group pulse output CH2 (Y4, Y5)	-	-	Y	Y	-	-	-	R/W	YES	200
D1380	Start frequency of the 1st section and end frequency of the last section for the 4th group pulse output CH3 (Y6, Y7)	-	-	Y	Y	-	-	-	R/W	YES	200
D1381	Acceleration/deceleration time for the 3rd pulse output CH2 (Y4, Y5)	-	-	Y	Y	-	-	-	R/W	YES	100
D1382	Acceleration/deceleration time for the 4th pulse output CH3 (Y6, Y7)	-	-	Y	Y	-	-	-	R/W	YES	100
D1383	Time difference between direction signal and pulse output for the 3rd group pulse CH2 (Y4, Y5) in DRVI, DDRVI, DRVA, DDRVA, PLSV, DPLSV (Unit: ms)	-	-	Y	Y	0	-	-	R/W	NO	0
D1384	Time difference between direction signal and pulse output for the 4th group pulse CH3 (Y6, Y7) in DRVI, DDRVI, DRVA, DDRVA, PLSV, DPLSV (Unit: ms)	-	-	Y	Y	0	-	-	R/W	NO	0
D1386	ID of the 1st left-side expansion module	-	-	Y	-	0	-	-	R	NO	0
D1387	ID of the 2nd left-side expansion module	-	-	Y	-	0	-	-	R	NO	0
D1388	ID of the 3rd left-side expansion module	-	-	Y	-	0	-	-	R	NO	0
D1389	ID of the 4th left-side expansion module	-	-	Y	-	0	-	-	R	NO	0
D1390	ID of the 5th left-side expansion module	-	-	Y	-	0	-	-	R	NO	0
D1391	ID of the 6th left-side expansion module	-	-	Y	-	0	-	-	R	NO	0
D1392	ID of the 7th left-side expansion module	-	-	Y	-	0	-	-	R	NO	0
D1393	ID of the 8th left-side expansion module	-	-	Y	-	0	-	-	R	NO	0
D1399	Assigning ID number of the starting slave on ELC Link network	-	Y	Y	Y	1	-	-	R/W	YES	1
D1415	Starting reference for Master to write in Slave ID#1	-	Y	Y	Y	-	-	-	R/W	NO	10C8
D1416	Starting reference for Master to write in Slave ID#2	-	Y	Y	Y	-	-	-	R/W	NO	10C8
D1417	Starting reference for Master to write in Slave ID#3	-	Y	Y	Y	-	-	-	R/W	NO	10C8
D1418	Starting reference for Master to write in Slave ID#4	-	Y	Y	Y	-	-	-	R/W	NO	10C8
D1419	Starting reference for Master to write in Slave ID#5	-	Y	Y	Y	-	-	-	R/W	NO	10C8
D1420	Starting reference for Master to write in Slave ID#6	-	Y	Y	Y	-	-	-	R/W	NO	10C8
D1421	Starting reference for Master to write in Slave ID#7	-	Y	Y	Y	-	-	-	R/W	NO	10C8

Special D	Function	ELCB -PB	ELC -PA	ELC-PV	ELC 2-PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
D1422	Starting reference for Master to write in Slave ID#8	-	Y	Y	Y	-	-	-	R/W	NO	10C8
D1423	Starting reference for Master to write in Slave ID#9	-	Y	Y	Y	-	-	-	R/W	NO	10C8
D1424	Starting reference for Master to write in Slave ID#10	-	Y	Y	Y	-	-	-	R/W	NO	10C8
D1425	Starting reference for Master to write in Slave ID#11	-	Y	Y	Y	-	-	-	R/W	NO	10C8
D1426	Starting reference for Master to write in Slave ID#12	-	Y	Y	Y	-	-	-	R/W	NO	10C8
D1427	Starting reference for Master to write in Slave ID#13	-	Y	Y	Y	-	-	-	R/W	NO	10C8
D1428	Starting reference for Master to write in Slave ID#14	-	Y	Y	Y	-	-	-	R/W	NO	10C8
D1429	Starting reference for Master to write in Slave ID#15	-	Y	Y	Y	-	-	-	R/W	NO	10C8
D1430	Starting reference for Master to write in Slave ID#16	-	Y	Y	Y	-	-	-	R/W	NO	10C8
D1431	Setting of times of ELC Link polling cycle	-	Y	Y	Y	0	-	-	R/W	NO	0
D1432	The current times of ELC Link polling cycle	-	Y	Y	Y	0	-	-	R/W	NO	0
D1433	Number of slave modules linked on ELC Link network	-	Y	Y	Y	0	-	-	R/W	NO	0
D1434	Data length to be read on Slave ID#1	-	Y	Y	Y	-	-	-	R/W	NO	16
D1435	Data length to be read on Slave ID#2	-	Y	Y	Y	-	-	-	R/W	NO	16
D1436	Data length to be read on Slave ID#3	-	Y	Y	Y	-	-	-	R/W	NO	16
D1437	Data length to be read on Slave ID#4	-	Y	Y	Y	-	-	-	R/W	NO	16
D1438	Data length to be read on Slave ID#5	-	Y	Y	Y	-	-	-	R/W	NO	16
D1439	Data length to be read on Slave ID#6	-	Y	Y	Y	-	-	-	R/W	NO	16
D1440	Data length to be read on Slave ID#7	-	Y	Y	Y	-	-	-	R/W	NO	16
D1441	Data length to be read on Slave ID#8	-	Y	Y	Y	-	-	-	R/W	NO	16
D1442	Data length to be read on Slave ID#9	-	Y	Y	Y	-	-	-	R/W	NO	16
D1443	Data length to be read on Slave ID#10	-	Y	Y	Y	-	-	-	R/W	NO	16
D1444	Data length to be read on Slave ID#11	-	Y	Y	Y	-	-	-	R/W	NO	16
D1445	Data length to be read on Slave ID#12	-	Y	Y	Y	-	-	-	R/W	NO	16
D1446	Data length to be read on Slave ID#13	-	Y	Y	Y	-	-	-	R/W	NO	16
D1447	Data length to be read on Slave ID#14	-	Y	Y	Y	-	-	-	R/W	NO	16
D1448	Data length to be read on Slave ID#15	-	Y	Y	Y	-	-	-	R/W	NO	16
D1449	Data length to be read on Slave ID#16	-	Y	Y	Y	-	-	-	R/W	NO	16
D1450	Data length to be written on Slave ID#1	-	Y	Y	Y	-	-	-	R/W	NO	16
D1451	Data length to be written on Slave ID#2	-	Y	Y	Y	-	-	-	R/W	NO	16
D1452	Data length to be written on Slave ID#3	-	Y	Y	Y	-	-	-	R/W	NO	16
D1453	Data length to be written on Slave ID#4	-	Y	Y	Y	-	-	-	R/W	NO	16
D1454	Data length to be written on Slave ID#5	-	Y	Y	Y	-	-	-	R/W	NO	16
D1455	Data length to be written on Slave ID#6	-	Y	Y	Y	-	-	-	R/W	NO	16
D1456	Data length to be written on Slave ID#7	-	Y	Y	Y	-	-	-	R/W	NO	16
D1457	Data length to be written on Slave ID#8	-	Y	Y	Y	-	-	-	R/W	NO	16
D1458	Data length to be written on Slave ID#9	-	Y	Y	Y	-	-	-	R/W	NO	16
D1459	Data length to be written on Slave ID#10	-	Y	Y	Y	-	-	-	R/W	NO	16
D1460	Data length to be written on Slave ID#11	-	Y	Y	Y	-	-	-	R/W	NO	16
D1461	Data length to be written on Slave ID#12	-	Y	Y	Y	-	-	-	R/W	NO	16
D1462	Data length to be written on Slave ID#13	-	Y	Y	Y	-	-	-	R/W	NO	16
D1463	Data length to be written on Slave ID#14	-	Y	Y	Y	-	-	-	R/W	NO	16
D1464	Data length to be written on Slave ID#15	-	Y	Y	Y	-	-	-	R/W	NO	16
D1465	Data length to be written on Slave ID#16	-	Y	Y	Y	-	-	-	R/W	NO	16
D1466	Number of pulses required per revolution of motor at CH0 (low word)	-	-	Y	Y	-	-	-	R	YES	2,000
D1467	Number of pulses required per revolution of motor at CH0 (high word)	-	-	Y	Y	-	-	-	R	YES	0
D1468	Number of pulses required per revolution of motor at CH1 (low word)	-	-	Y	Y	-	-	-	R	YES	2,000
D1469	Number of pulses required per revolution of motor at CH1 (high word)	-	-	Y	Y	-	-	-	R	YES	0
D1470	Distance created for 1 revolution of motor at CH0 (low word)	-	-	Y	Y	-	-	-	R	YES	1,000
D1471	Distance created for 1 revolution of motor at CH0 (high word)	-	-	Y	Y	-	-	-	R	YES	0
D1472	Distance created for 1 revolution of	-	-	Y	Y	-	-	-	R	YES	1,000

Special D	Function	ELCB -PB	ELC -PA	ELC-PV	ELC 2-PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
	motor at CH1 (low word)										
D1473	Distance created for 1 revolution of motor at CH1 (high word)	-	-	Y	Y	-	-	-	R	YES	0
D1474	Machine unit of CH0 movement (low word)	-	-	Y	Y	-	-	-	R	YES	0
D1475	Machine unit of CH0 movement (high word)	-	-	Y	Y	-	-	-	R	YES	0
D1476	Machine unit of CH1 movement (low word)	-	-	Y	Y	-	-	-	R	YES	0
D1477	Machine unit of CH1 movement (high word)	-	-	Y	Y	-	-	-	R	YES	0
D1478	Input / output percentage value of CH2(Y4, Y5) close loop control	-	-	Y	Y	100	-	-	R/W	NO	100
D1479	Input / output percentage value of CH3 CH3(Y6, Y7) close loop control	-	-	Y	Y	100	-	-	R/W	NO	100
D1480 ↓ D1495	The data which is read from slave ID#1 in the ELC LINK at the time when M1353 is OFF.	-	Y	Y	Y	0	-	-	R	YES	0
	The initial data register where the communication address read from slave ID#1~ID#16 in the ELC LINK is stored at the time when M1353 is ON	-	-	Y	Y	-	-	-	R/W	YES	0
D1496 ↓ D1511	The data which is written into slave ID#1 in the ELC LINK at the time when M1353 is OFF	-	Y	Y	Y	0	-	-	R	YES	0
	The initial data register where the communication address written into slave ID#1~ID#16 in the ELC LINK is stored at the time when M1353 is ON	-	-	Y	Y	-	-	-	R/W	YES	0
D1512 ↓ D1527	The data which is read from slave ID#2 in the ELC LINK	-	Y	Y	Y	0	-	-	R	YES	0
	The initial data register where the communication address read from slave ID#17~ID#32 in the ELC LINK is stored at the time when M1353 is ON	-	-	Y	Y	0	-	-	R	YES	0
D1528 ↓ D1543	The data which is written into slave ID#2 in the ELC LINK	-	Y	Y	Y	0	-	-	R	YES	0
	The initial data register where the communication address written into slave ID#17~ID#32 in the ELC LINK is stored at the time when M1353 is ON	-	-	Y	Y	0	-	-	R/W	YES	0
D1544 ↓ D1559	The data which is read from slave ID#3 in the ELC LINK	-	Y	Y	Y	0	-	-	R	YES	0
	The initial data register where the number of data that ID#17~ID#32 read in the ELC LINK is stored at the time when M1353 is ON	-	-	Y	Y	-	-	-	R	YES	0
D1560* ↓ D1575*	The data which is written into slave ID#3 in the ELC LINK	-	Y	Y	Y	0	-	-	R	YES	0
	The initial data register where the number of data that ID#17~ID#32 write in the ELC LINK is stored at the time when M1353 is ON	-	-	Y	Y	-	-	-	R/W	YES	0
D1576* ↓ D1591*	The data which is read from slave ID#4 in the ELC LINK	-	Y	Y	Y	0	-	-	R	YES	0
	The initial data register where the data that ID#17~ID#32 read in the ELC LINK is stored at the time when M1353 is ON	-	-	Y	Y	-	-	-	R	YES	0
D1592* ↓ D1607*	The data which is written into slave ID#4 in the ELC LINK	-	Y	Y	Y	0	-	-	R	YES	0
	The initial data register where the data that ID#17~ID#32 write in the ELC LINK is stored at the time when M1353 is ON	-	-	Y	Y	-	-	-	R/W	YES	0
D1608 ↓ D1623	Data buffer to store the data read from Slave ID#5.	-	Y	Y	Y	-	-	-	R	YES	0
D1624	Data buffer to store the data to be written on Slave ID#5.	-	Y	Y	Y	-	-	-	R/W	YES	0

Special D	Function	ELCB -PB	ELC -PA	ELC- PV	ELC 2-PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
↓ D1639											
D1640 ↓ D1655	Data buffer to store the data read from Slave ID#6.	-	Y	Y	Y	-	-	-	R	YES	0
D1656 ↓ D1671	Data buffer to store the data to be written on Slave ID#6.	-	Y	Y	Y	-	-	-	R/W	YES	0
D1672 ↓ D1687	Data buffer to store the data read from Slave ID#7.	-	Y	Y	Y	-	-	-	R	YES	0
D1688 ↓ D1703	Data buffer to store the data to be written on Slave ID#7.	-	Y	Y	Y	-	-	-	R/W	YES	0
D1704 ↓ D1719	Data buffer to store the data read from Slave ID#8.	-	Y	Y	Y	-	-	-	R	YES	0
D1720 ↓ D1735	Data buffer to store the data to be written on Slave ID#8.	-	Y	Y	Y	-	-	-	R/W	YES	0
D1736 ↓ D1751	Data buffer to store the data read from Slave ID#9.	-	Y	Y	Y	-	-	-	R	YES	0
D1752 ↓ D1767	Data buffer to store the data to be written on Slave ID#9.	-	Y	Y	Y	-	-	-	R/W	YES	0
D1768 ↓ D1783	Data buffer to store the data read from Slave ID#10.	-	Y	Y	Y	-	-	-	R	YES	0
D1784 ↓ D1799	Data buffer to store the data to be written on Slave ID#10.	-	Y	Y	Y	-	-	-	R/W	YES	0
D1800 ↓ D1815	Data buffer to store the data read from Slave ID#11.	-	Y	Y	Y	-	-	-	R	YES	0
D1816 ↓ D1831	Data buffer to store the data to be written on Slave ID#11.	-	Y	Y	Y	-	-	-	R/W	YES	0
D1832 ↓ D1847	Data buffer to store the data read from Slave ID#12.	-	Y	Y	Y	-	-	-	R	YES	0
D1848 ↓ D1863	Data buffer to store the data to be written on Slave ID#12.	-	Y	Y	Y	-	-	-	R/W	YES	0
D1864 ↓ D1879	Data buffer to store the data read from Slave ID#13.	-	Y	Y	Y	-	-	-	R	YES	0
D1880 ↓ D1895	Data buffer to store the data to be written on Slave ID#13.	-	Y	Y	Y	-	-	-	R/W	YES	0
D1896 ↓ D1911	Data buffer to store the data read from Slave ID#14.	-	Y	Y	Y	-	-	-	R	YES	0
D1900 ↓ D1915	When M1356 is ON, the values in these registers are defined as the station address (ID1~ID16). The default station address in D1399 is not used. Only when M1356 is ON is rhe latched function available.	-	-	-	Y	1~16	-	-	R/W	NO	1~16
D1916 ↓ D1931	When M1356 is ON, the values in these registers are defined as the station address (ID17~ID32). The default station address in D1399 is not used. Only when M1356 is ON is rhe latched function available.	-	-	-	Y	17~32	-	-	R/W	NO	17~32
D1912	Data buffer to store the data to be written	-	Y	Y	Y	-	-	-	R/W	YES	0

Special D	Function	ELCB -PB	ELC -PA	ELC-PV	ELC 2-PV	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
↓ D1927	on Slave ID#14.										
D1928 ↓ D1943	Data buffer to store the data read from Slave ID#15.	-	Y	Y	Y	-	-	-	R	YES	0
D1944 ↓ D1959	Data buffer to store the data to be written on Slave ID#15.	-	Y	Y	Y	-	-	-	R/W	YES	0
D1960 ↓ D1975	Data buffer to store the data read from Slave ID#16.	-	Y	Y	Y	-	-	-	R	YES	0
D1976 ↓ D1991	Data buffer to store the data to be written on Slave ID#16.	-	Y	Y	Y	-	-	-	R/W	YES	0
D1994	Limit of ID setting times for ELC-ACPGMXFR	-	Y	Y	Y	0	-	-	R/W	NO	0
D1995	Data length of ELC ID Setting for ELC-ACPGMXFR	-	-	Y	Y	0	-	-	R/W	NO	0
D1996	1st Word of ELC ID Setting for ELC-ACPGMXFR (Indicated by Hex format corresponding to ASCII codes)	-	Y	Y	Y	0	-	-	R/W	NO	0
D1997	2nd Word of ELC ID Setting for ELC-ACPGMXFR (Indicated by Hex format corresponding to ASCII codes)	-	Y	Y	Y	0	-	-	R/W	NO	0
D1998	3rd Word of ELC ID Setting for ELC-ACPGMXFR (Indicated by Hex format corresponding to ASCII codes)	-	Y	Y	Y	0	-	-	R/W	NO	0
D1999	4th word of ELC ID Setting for ELC-ACPGMXFR (Indicated by Hex format corresponding to ASCII codes)	-	Y	Y	Y	0	-	-	R/W	NO	0
D9900 ↓ D9979	For Analog I/O modules only.	-	-	-	Y	0	-	-	R/W	NO	0

ELCM-PH/ELCM-PA, ELC2-PB, ELC2-PC/ELC2-PE, and ELC2-PA:

Special D	Function	ELCM -PH -PA	ELC 2-PB	ELC2 -PC -PE	ELC2 -PA	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
D1000	Watchdog timer (WDT) value (Unit: 1ms)	Y	Y	Y	Y	200	-	-	R/W	NO	200
D1001	ELC program version (user can read the ELC program version from this register. For example, D1001 = H XX10 means version 1.0	Y	Y	Y	Y	-	-	-	R	NO	#
D1002	Program capacity (ELCM-PH/PA, ELC2-PC/PA/PE: 15872; ELC2-PB: 7920)	Y	Y	Y	Y	#	-	-	R	NO	#
D1003	Sum of the ELC internal program memory (#:ELCM-PH/PA, ELC2-PC/PA/PE: -15872; ELC2-PB: -7920)	Y	Y	Y	Y	-	-	-	R	YES	#
D1004	Grammar detective number	Y	Y	Y	Y	0	0	-	R	NO	0
D1008	STEP address when WDT timer is ON	Y	Y	Y	Y	0	-	-	R	NO	0
D1009	Number of LV signal occurrence	Y	Y	Y	Y	-	-	-	R	YES	0
D1010	Present scan time (Unit: 0.1ms)	Y	Y	Y	Y	#	#	#	R	NO	0
D1011	Minimum scan time (Unit: 0.1ms)	Y	Y	Y	Y	#	#	#	R	NO	0
D1012	Maximum scan time (Unit: 0.1ms)	Y	Y	Y	Y	#	#	#	R	NO	0
D1015	0~32,767(unit: 0.1ms) addition type of high-speed connection timer	Y	Y	Y	Y	0	-	-	R/W	NO	0
D1018	π PI (Low byte)	Y	Y	Y	Y	0FDB	0FDB	0FDB	R/W	NO	0FDB
D1019	π PI (High byte)	Y	Y	Y	Y	4049	4049	4049	R/W	NO	4049
D1020	X0~X7 input filter (unit: ms) 0~1,000ms adjustable	Y	Y	Y	Y	10	-	-	R/W	NO	10
D1022	Double frequency selection for AB	Y	Y	Y	Y	#	-	-	R/W	NO	#

Special D	Function	ELCM -PH -PA	ELC 2-PB	ELC2 -PC -PE	ELC2 -PA	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
	phase counter										
D1023	Storing detected pulse width (unit: 0.1ms)	Y	Y	Y	Y	0	-	-	R/W	NO	0
D1025	Communication error code	Y	Y	Y	Y	0	-	-	R	NO	0
D1026*	Pulse number for masking Y0 when M1156 = ON (Low word)	Y	Y	Y	Y	0	0	-	R/W	NO	0
D1027	Pulse number for masking Y0 when M1156 = ON (High word)	Y	Y	Y	Y	0	0	-	R/W	NO	0
D1028	Index register E0	Y	Y	Y	Y	0	-	-	R/W	NO	0
D1029	Index register F0	Y	Y	Y	Y	0	-	-	R/W	NO	0
D1030	Output numbers of Y0 pulse (Low word)	Y	Y	Y	Y	-	-	-	R	NO	0
D1031	Output numbers of Y0 pulse (High word)	Y	Y	Y	Y	-	-	-	R	NO	0
D1032	Output numbers of Y1 pulse (Low word)	Y	Y	Y	Y	0	-	-	R	NO	0
D1033	Output numbers of Y1 pulse (High word)	Y	Y	Y	Y	0	-	-	R	NO	0
D1036	COM1 (RS-232) Communications protocol	Y	Y	Y	Y	0086	-	-	R/W	NO	0086
D1037	Register for setting 8-sets SPD function (has to be used with M1037)	Y	Y	Y	Y	0	-	-	R/W	NO	0
D1038	1. Delay time setting for data response when ELC is SLAVE in COM2 / COM3 RS-485 communication. Range: 0 ~ 10,000 (unit: 0.1ms). 2. By using ELC LINK in COM2 (RS-485), D1038 can be set to send next communication data with delay. Range: 0 ~ 10,000 (Unit: one scan cycle)	Y	Y	Y	Y	-	-	-	R/W	NO	0
D1039	Constant scan time (ms)	Y	Y	Y	Y	0	-	-	R/W	NO	0
D1040	ON state number 1 of STEP point S	Y	Y	Y	Y	0	-	-	R	NO	0
D1041	ON state number 2 of STEP point S	Y	Y	Y	Y	0	-	-	R	NO	0
D1042	ON state number 3 of STEP point S	Y	Y	Y	Y	0	-	-	R	NO	0
D1043	ON state number 4 of STEP point S	Y	Y	Y	Y	0	-	-	R	NO	0
D1044	ON state number 5 of STEP point S	Y	Y	Y	Y	0	-	-	R	NO	0
D1045	ON state number 6 of STEP point S	Y	Y	Y	Y	0	-	-	R	NO	0
D1046	ON state number 7 of STEP point S	Y	Y	Y	Y	0	-	-	R	NO	0
D1047	ON state number 8 of STEP point S	Y	Y	Y	Y	0	-	-	R	NO	0
D1049	ON number of alarm point	Y	Y	Y	Y	0	-	-	R	NO	0
D1050 ↓ D1055	ELC will automatically convert the ASCII data saved in D1070~D1085 to HEX.	Y	Y	Y	Y	0	-	-	R	NO	0
D1062	Average number of times an analog signal is input to the ELCM-PA/ELC2-PA The default value is K10 for ELCM-PA version 2.6 and version 2.8.	Y	-	-	Y	2	-	-	R/W	YES	2
D1067	Algorithm error code	Y	Y	Y	Y	0	0	-	R	NO	0
D1068	Lock the algorithm error address	Y	Y	Y	Y	0	-	-	R	NO	0
D1070 ↓ D1085	When the ELC's built-in RS-485 communication instruction receives feedback signals from receiver. The data will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.	Y	Y	Y	Y	0	-	-	R	NO	0
D1086	ELC-ACPGMXFR High word of password setting (Display by HEX value corresponding to ASCII word)	Y	Y	Y	Y	0	-	-	R/W	NO	0
D1087	ELC-ACPGMXFR Low word of password setting (Display by HEX value corresponding to ASCII word)	Y	Y	Y	Y	0	-	-	R/W	NO	0

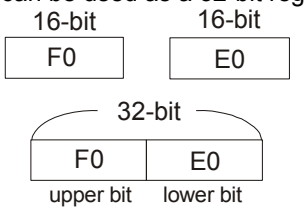
Special D	Function	ELCM -PH -PA	ELC 2-PB	ELC2 -PC -PE	ELC2 -PA	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Type	Latched	Factory setting
D1089 ↓ D1099	When the ELC built-in RS-485 communication instruction is executed, the transmitting signals will be stored in the registers D1089~D1099. User can use the contents saved in registers to check the feedback data.	Y	Y	Y	Y	0	-	-	R	NO	0
D1109	COM3 (RS-485) Communication protocol	Y	-	Y	Y	0086	-	-	R/W	NO	0086

2.18 E, F Index Registers

An Index register functions the same way as a general operand. It can be used to move, compare, or be used as an index for byte device (KnX, KnY, KnM, KnS, T, C, D) and bit device (X, Y, M, S). It cannot be used for constant (K, H). Only for ELC-PV and ELC2-PV controllers, it can be used for constant (K, H).

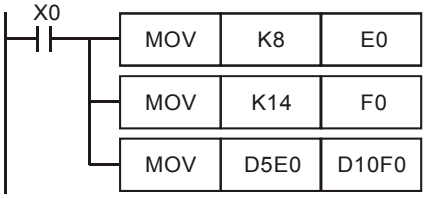
Index register [E], [F]
Index registers are 16-bit registers. There are 2 points, E0 and F0, for ELCB-PB controllers. There are 8 points, E0~E3 and F0~F3, for ELC-PA controllers. There are 16 points, E0~E7 and F0~F7, for ELC-PV, ELCM-PH/PA, ELC2-PB/PH/PA/PE/PV controllers. If you want to use index register to be a 32-bit register, you should indicate E and at this moment F cannot be used.

Example: "MOV K10 D0F0"
Index registers E, and F are 16-bit data registers, similar to the general data register. They are read/write.
They can be used as a 32-bit register.



It is recommended to use the instruction DMOVP K0 E to clear E and F to 0 at power on.
The combination of E and F when used as a 32-bit register are:
(E0, F0) , (E1, F1) (E2, F2) (E3, F3) (E4, F4) , (E5, F5) (E6, F6) (E7, F7)

When X0=ON, if E0=8, and F0=14,
D5E0=D(5+8)=D13,
D10F0 =D(10+14) = D24,
The contents of D13 will be moved to D24.



2.19 File Register

There is no device number (name) for file register; to execute read/writes of file register use instruction MEMR or MEMW, peripheral device ELCSoft. The other controllers are not support.
The models have file registers:

model	ELC-PA	ELC-PV	ELC2-PV	ELC2-PC/PA ELCM-PH/PA
file registers	1,600 (K0~K1599)	10,000 (K0~K9,999)	50,000 (K0~K49,999)	5,000 (K0~K4,999)

The ELC will check the following when the ELC is powered on(Only for ELC-PA/PV, ELC2-PV).

1. M1101 - starts file register function
2. D1101 - starting address of file register, for ELC-PA controllers: K0~K1599; for ELC-PV, ELC2-PV controllers: K0~K9999.
3. D1102 - number of item to read/write, for ELC-PA controllers: K1~ K1600; for ELC-PV, ELC2-PV controllers: K1~K8000.
4. D1103 - starting address of file register D register, ELC-PA: D2000 ~ D4999; ELC-PV: D2000 ~ D9999; ELC2-PV: D2000~D11999;

Note:

1. Reading from file register to data register D will not be executed when D1101 is greater than 1600 in ELC-PA controllers, D1101 greater than 8000 and D1103 smaller than 2000 or greater than 9999/11999 in ELC-PV/ELC2-PV controllers.
2. When starting the action to read data from the file register to the data register, the ELC will stop executing once the address of file register or data register D exceeds the viable address range.
3. There are 1600 file registers in ELC-PA controllers, 10000 in ELC-PV controllers, 50000 in ELC2-PV controllers. The file register does not have an exact device number; therefore the read/write function of file registers has to be executed by instruction API 148 MEMR, API 149 MEMW or through ELCSoft.

2.20 Nest Level Pointer[N], Pointer[P], Interrupt Pointer [I]

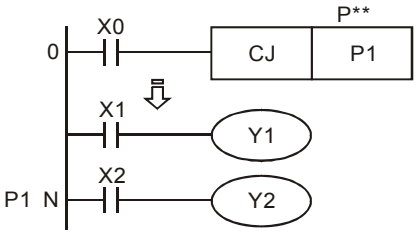
Pointer	N	Master control nested		N0~N7, 8 points	The control point of master control nested
	P	For CJ, CALL instructions		ELCB-PB Controllers = P0~P63 64 points ELC- PA/PV, ELC2-PV, ELCM-PH/PA, ELC2-PB/PH/PA/PE/SV Controllers = P0~P255, 256 points	The location point of CJ, CALL
	I	For interrupt	Insert external interrupt	ELCB-PB Controllers = I001, I101, I201, I301 4 points ELC-PA controllers = I001, I101, I201, I301, I401, I501, total is 6 points ELC-PV controllers = I000/I001(X0), I100/I101 (X1), I200/I201 (X2), I300/I301 (X3), I400/I401 (X4), I500/I501 (X5), 6 points (01, rising-edge trigger ⌐, 00, falling-edge trigger ⌙) ELC2-PV controllers = I000/I001(X0), I100/I101 (X1), I200/I201 (X2), I300/I301 (X3), I400/I401 (X4), I500/I501 (X5), I600/I601 (X6), I700/I701 (X7), I900/I901 (X10), I910/I911 (X11), I920/I921 (X12), I930/I931 (X13), I940/I941 (X14), I950/I951 (X15), I960/I961 (X16), I970/I971 (X17), 16 points (01, rising-edge trigger ⌐, 00, falling-edge trigger ⌙) ELCM-PH/PA, ELC2-PB/PH/PA/PE controllers = I000/I001(X0), I100/I101(X1), I200/I201(X2), I300/I301(X3), I400/I401(X4), I500/I501(X5), I600/I601(X6), I700/I701(X7), 8 points (01, rising-edge trigger ⌐, 00, falling-edge trigger ⌙)	The location point of interrupt subroutine.
			Insert time interrupt	ELCB-PB Controllers = I610~I699, (Timer resolution: 1ms), 1 point ELC-PA Controllers = I601~I699, I701~I799, (Timer resolution: 1ms) , 2 points ELC-PV controllers = I601~I699, I701~I799, (Timer resolution: 1ms); I801~I899, (Timer resolution: 0.1ms) , 3 points ELC2-PV controllers = I602~I699, I702~I799,(Timer resolution: 1ms); I801~I899, (Timer resolution: 0.1ms), 3 points ELCM-PH/PA, ELC2-PB/PH/PA/PE controllers = I602~I699, I702~I799, (Timer resolution: 1ms) , 2 points	The location point of interrupt subroutine

Pointer	I	For interrupt	High-speed counter interruption	ELC-PA Controllers = I010, I020, I030, I040, I050, I060, 6 points ELC-PV Controllers = I010, I020, I030, I040, I050, I060, 6 points ELC2-PV Controllers = I010, I020, I030, I040, I050, I060, 6 points ELCM-PH/PA, ELC2-PB/PH/PA/PE Controllers = I010, I020, I030, I040, I050, I060, I070, I080, 8 points	
			Pulse interruption	ELC-PV, ELC2-PV Controllers = I110, I120, I130, I140, 4 points	
			Insert communication interrupt	ELCB-PB, ELC-PA Controllers= I150, 1 point ELC-PV, ELC2-PV Controllers = I150, I160, I170, 3 points ELCM-PH/PA, ELC2-PC/PA Controllers = I140, I150, I160, 3 points ELC2-PB Controllers = I140, I150, 2 points ELC2-PE Controllers = I150, I160, 2 points	

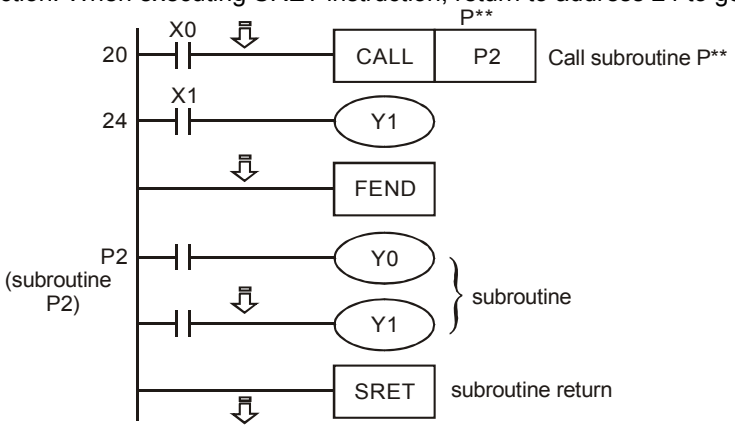
Nest Level Pointer N: used with instruction MC and MCR. MC is master start instruction. When the MC instruction is executed, the instructions between MC and MCR will be executed normally. MC-MCR master instruction supports nested program structure and the maximum is 8 levels, which is numbered from N0 to N7.

Pointer P: use with application instructions CJ, CALL, and SRET.
CJ condition jump:

When X0=ON, program will jump from 0 to N (designated label P1) and keep on executing without executing the instructions between 0 and N.
When X0=OFF, program will execute from 0 and keep on executing the followings. CJ instruction will not be executed at this time.



CALL subroutine, SRET subroutine END:
When X0 is ON, it will jump to P2 to execute the designated subroutine as executing CALL instruction. When executing SRET instruction, return to address 24 to go on executing.



Interrupt pointer I:
It is used with application instruction EI, DI, IRET. There are five functions below. Interrupt insert should be used with EI, interrupt insert enable, interrupt insert disable and IRET interrupt insert return, etc.

1. External interrupt

In ELC- PA, ELCB-PB, When input signal of input terminal X0~X5 is triggered on rising-edge, it will interrupt the present program and jump to the designated interrupt subroutine pointer I001(X0), I101(X1), I201(X2), I301(X3), I401(X4), I501(X5) to execute and return to the previous address to execute when executing IRET instruction.
In ELC-PA V1.2 and above, when I401 (X4) works with X0 (C235, C251 or C253), the value of (C243 or C255) will be stored in (D1180, D1181) and I501 (X5) works with X1 (C236), the value of high-speed counter (C236) will be stored in (D1198, D1199).
In ELC-PV, ELC2-PV, when input signal of input terminal X0~X5 is triggered on rising-edge or falling-edge, it will interrupt current program execution and jump to the designated interrupt subroutine pointer I000/I001(X0), I100/I101(X1), I200/I201(X2), I300/I301(X3), I400/I401(X4), I500/I501(X5),. When IRET instruction is executed, program execution returns to the address before interrupt occurs.
In ELCM-PH/PA , ELC2-PB/PH/PA/PE when input signal of input terminal X0~X7 is triggered on rising-edge or falling-edge, it will interrupt current program execution and jump to the designated interrupt subroutine pointer I000/I001(X0), I100/I101(X1), I200/I201(X2), I300/I301(X3), I400/I401(X4), I500/I501(X5), I600/I601(X6), I700/I701(X7). When IRET instruction is executed, program execution returns to the address before interrupt occurs.
In ELCM-PH/PA, ELC2-PB/PH/PA/PE when X0 (C243) works with I100/I101 (X1), X0/X1 (C246, C248, C252) works with I400/I401, the value of C243, C246, C248, C252 will be stored in (D1240, D1241)
In ELCM-PH/PA, ELC2-PB/PH/PA/PE when X2 (C244) works with I300/I301 (X3), X2/X3 (C250, C254) works with I500/I501, the value of C244, C250, C254 will be stored in (D1242, D1243).

2. Timer interrupt

ELC will stop the present program and jump to the designated interrupt subroutine. Then ELC will execute automatically for every time period set by 1ms~99ms (0.1ms~9.9ms)

3. Pulse interruption

In ELC-PV, ELC2-PV series, the pulse output instruction API 57 PLSY can be set up so the interrupt signal is sent out synchronously when the first pulse is sent out by enabling flags M1342 and M1343. The corresponding interrupts are I130 and I140. You can also set up so the interrupt signal is sent out after the last pulse is sent out, by enabling flags M1340 and M1341. The corresponding interrupts are I110 and I120.

4. Counter attained interrupt

The comparison instruction DHSCS of the high-speed counter can be used to interrupt the present program and jump to the designated interrupt insert subroutine to execute the interrupt pointer I010, I020, I030, I040, I050, I060, I070, I080 when the comparison is attained.

5. Communication interrupt

ELC-PB/PA/PV, ELCB-PB:
I150:
When using the communication instruction RS, it can be set to have an interrupt request when receiving specific characters. Interrupt I150 and specific characters are set to the low byte of D1168 (ELC-PA), D1127 (ELCB-PB).
When the ELC connects to a communication device and the received data length is not the expected length, the end character in D1168 (D1127) and the interrupt subroutine I150 is set. When the ELC receives this end character, it will execute interrupt subroutine I150.
I160:
The RS instruction sends out an interrupt request when receiving a specific length of data. When the data received equals the low byte of D1169, I160 will be triggered. When D1169 = 0, I160 will not be triggered.
I170:
In Slave mode, interrupt I170 will be generated when the received data message is complete. Normally when the communication terminal of the ELC is in Slave mode, the ELC will not immediately process the communication data entered, but process it after the END is executed. Therefore, when the scan time is very long and you need the communication data to be processed immediately, you can use interruption I170 for this functionality.

ELCM-PH/PA, ELC2-PB/PH/PA/PE:
I140:
The communication instruction RS (COM1 RS-232) can be used to send an interrupt request when specific characters are received. Interrupt I140 and the specific characters are set to low byte of D1167.
This function can be used when the ELC receives data of a different length during communications. Set up the specific end word in D1167 and write the interruption subroutine I140. When the ELC receives the end word, the program will execute I140.
I150:
The communication instruction RS (COM2 RS-485) can be used to send an interrupt request when specific characters are received. Interrupt I150 and the specific characters are set to low byte of D1168.
This function can be used when the ELC receives data of a different length during the communications. Set up the specific end word in D1168 and write the interruption subroutine I150. When the ELC receives the end word, the program will execute I150.
I160:
The communication instruction RS (COM3 RS-485) can be used to send an interrupt request when specific characters are received. Interrupt I160 and the specific characters are set to low byte of D1169
This function can be used when the ELC receives data of a different length during the communications. Set up the specific end word in D1169 and write the interruption subroutine I160. When the ELC receives the end word, the program will execute I160

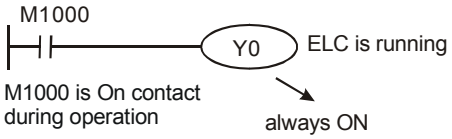
2.21 Applications of Special M Relay and Special D Register

Function Group ELC Operation Flag

Number M1000~M1003

Contents:
These relays provide information about the ELC when switched to run mode.

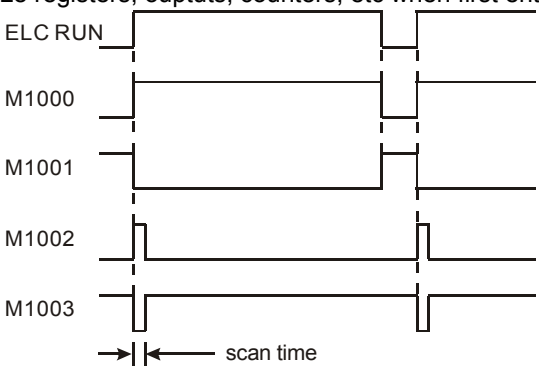
M1000:
Always ON when in run mode.



M1001:
Always OFF when in run mode.

M1002:
ON for the first scan when the ELC starts then OFF the rest of the time during run mode. Use to initialize registers, ouptuts, counters, etc. when first entering run mode.

M1003:
OFF for the first scan when the ELC starts, then ON the rest of the time during run mode. Use to initialize registers, ouptuts, counters, etc when first entering run mode.

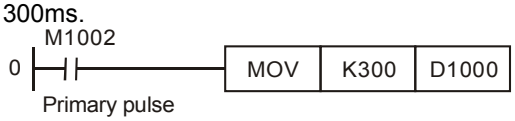


Function Group Watchdog Timer (WDT)

Number D1000

Contents:

1. Monitor timer is used for monitoring the ELC scan time. When the scan time exceeds the set value (SV) in the monitor timer, the ELC ERROR LED will be ON and all outputs will be “OFF”.
2. The default in the monitor timer is 200ms. If the program is long or the operation is too complicated, MOV instruction can be used to modify SV. See the example below for SV = 300ms.



3. The maximum SV in the monitor timer is 32,767ms. However, care should be taken when adjusting SV. If SV in D1000 is too large, it will take much longer for operation errors to be detected. Therefore, SV is recommended to be shorter than 200ms.
4. Scan time could be prolonged due to complicated instruction operations or by the use of many I/O modules. Check D1010 ~ D1012 to see if the scan time exceeds the SV value in D1000. Besides modifying the SV value in D1000, you can also apply the WDT instruction (API 07). When program execution progresses to WDT instruction, the internal monitor timer will be reset and therefore the scan time will not exceed the set value in the monitor timer.

Function Group Program Capacity

Number D1002

Contents:
This register holds the program capacity of the ELC.
1. ELCB-PB controllers: 3,792 Steps (Word)
2. ELC-PA, ELC2-PB controllers: 7,920 Steps (Word)
3. ELC-PV, ELCM-PH/M-PA, ELC2-PC/PA/PE controllers: 15,872 Steps (Word)
4. ELC2-PV controllers: 30,000 Steps (Word)

Function Group Grammar

Number M1004, D1004, D1137

Contents:
1. When errors occur in the syntax check, the ERROR LED indicator will flash and special relay M1004 = ON.
2. Timings for the ELC syntax check:
   a) When the power goes from “OFF” to “ON”.
   b) When ELCSoft writes the program into the ELC.
   c) When on-line editing is being conducted between the ELCM-PH/PA and ELCSoft.
3. Errors might result from a parameter error or a program error. The error code of the error will be placed in D1004. The address where the fault is located is saved in D1137. If the error belongs to a loop error, it may not have an address associated with it. In this case the value in D1137 is invalid.

Function Group Scan Time-out Timer

Number M1008, D1008

Contents:
1. When scan time-outs occur during program execution, the ELC ERROR LED will light and M1008=ON.
2. D1008 saves which address the STEP timeout occurred on.

Function Group Checking Lost ELC SRAM Data

Number D1009, M1175, M1176

Contents:
1. bit0 ~ bit7 record the types of data lost. bit = 1 refers to losing data; bit = 0 refers to correct data.
2. What are lost

bit8 ~ 15	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reserved	ELC program	D register	T register	C register	File register	M relay	S step	password

3. After the ELC is powered, the data in SRAM will be verified. If the SRAM data are lost, the ELC will record the error in D1009 and set on M1175 or M1176 according to the content of the data.

Function Group Scan Time Monitor

Number D1010~D1012

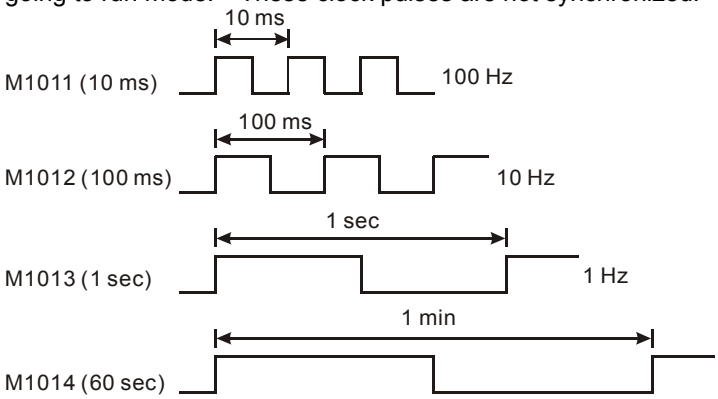
Contents:
The present value, minimum value and maximum value of scan time are saved in these registers.
D1010: present scan time value.
D1011: minimum scan time value.
D1012: maximum scan time value.

Function Group Internal Clock Pulse

Number M1011~M1014

Contents:

The ELC provides four different clock pulses to aid the application. When the ELC is powered on, these four clock pulse will start automatically. All are 50% duty cycle. The ELC starts the timing when going to run mode. These clock pulses are not synchronized.



Function Group High-speed Timer

Number M1015, D1015

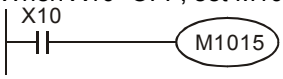
Contents:

The steps for using special M and special D directly:

3. Only valid when the ELC is in RUN for ELC-PV, ELC2-PV, ELCM-PH/PA, but is valid when the ELC is in RUN or STOP status for ELC-PA.
4. When M1015=ON, it will start high-speed timer D1015 once the ELC finish executing the END instruction of that scan period. The minimum unit of D1015 is 100us.
5. The range of D1015 is 0~32,767. When it counts to 32,767, it will start from 0.
6. When M1015=OFF, D1015 will stop timing immediately.
7. ELC-PV, and ELC2-PV controllers offers high-speed timer instruction HST. See the API 196 HST for more details.

Example:

1. When X10 is ON, set M1015=ON to start high-speed timer and record in D1015.
2. When X10=OFF, set M1015=OFF to close high-speed timer.



Function Group Perpetual Calendar Clock

Number M1016~M1017, M1076, D1313~D1319

Contents:

1. Special M and special D relevant to RTC

	Name	Function
M1016	Year Display	OFF: show the 2 right most bits ON: show the (2 right most bits + 2000)
M1017	Make ±30 seconds adjustment to clock	When OFF→ON, adjust is triggered RTC = 0~29 seconds, second will be reset to 0 and minute will not change. RTC = 30~59 seconds, reset second to 0 and add 1 to minute.
M1076	RTC malfunction	ON when setting exceeds range or battery has run down. RTC will reset to Jan. 1, 2000. 00:00

	Name	Function
M1082	Flag change on RTC	On: Modification on RTC
D1313	Second	0~59
D1314	Minute	0~59
D1315	Hour	0~23
D1316	Day	1~31
D1317	Month	1~12
D1318	Week	1~7
D1319	Year	0~99(2 right-most bit)

2. If the set value for the RTC is invalid: RTC will display the time as Second→0, Minute→0, Hour →0, Day→1, Month→1, Week→1, Year→0.
3. Only when power is on can RTCs of ELC2-PB perform the fuction of timing. Memory of the RTC is latched. The RTC will continue to time when power is down. For higher accuracy of RTC, please verify RTC time when power resumes.
4. D1313 ~ D1319 will immediately update the RTC only when in TRD instruction or ECISoft monitoring mode.
5. RTCs of ELC2-PC/PE/PA and ELCM-PH/PA version 2.0 (and above) can still operate for one or two weeks after the power is off (they vary with the ambient temperature). Therefore, if the machine has not operated since one or two weeks ago, please reset RTC.
6. RTCs of ELC-PV and ELC2-PV can still operate for 6 months after the power is off (they vary with the ambient temperature). Therefore, if the machine has not operated since 6 months ago, please reset RTC.
7. Adjust method of the perpetual clock:
   a) You can use a specific command, TWR, to adjust for ELC-PA/PV,ELC2-PV/PA/PH/PE, and ELCM-PH/PA version 2.0 controller's built-in real time clock. Refer to the TWR command for detail.
   b) Using peripherial ELCSoft to set.

Function Group π (PI)

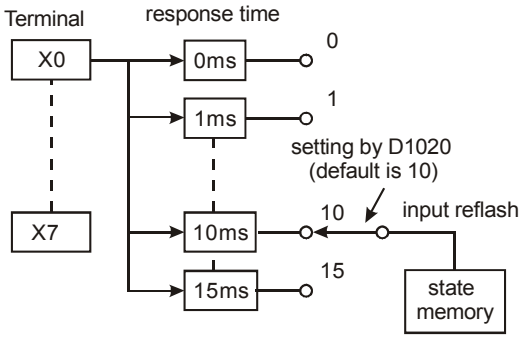
Number D1018~D1019

Contents:
1. D1018 and D1019 are combined as 32-bit data register containing floating point value of PI.
2. Floating point value = H 40490FDB

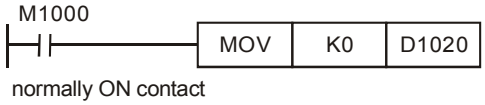
Function Group Input points time delay

Number D1020~D1021

Contents:
1. ELC-PA, ELCM-PH/PA controllers X0~X7: 10ms (factory default), 0~20ms adjustable. Refer to the usage of special registers D1020.
2. D1020 can be used for setting up the response time of receiving pulses at X0 ~X7 for ELC-PB/PA controllers. (Setup range: 0 ~ 20; Unit: ms)
3. D1021 can be used for setting up the response time of receiving pulses at X10 ~X17 for ELC-PB controllers. (Setup range: 0 ~ 20; Unit: ms)
4. D1020 can be used for setting up the response time of receiving pulses at X0 ~X7 for ELC-PV/ELC2-PV series MPU. (Setup range: 0 ~ 60; Unit: ms)
5. D1021 can be used for setting up the response time of receiving pulses at X10 ~X17 for ELC-PV/ELC2-PV series MPU. (Setup range: 0 ~ 60; Unit: ms)
6. When power to the ELC goes from OFF→ON, the content of D1020 and D1021 will become to 10 (default) automatically.



7. When setting the X0~X7 response time to 0ms to execute following program, the faster response time input terminal will be 50μs due to input terminal connection to an RC filter circuit in series.



8. It is not necessary to adjust response time when using high-speed counter or an interrupt in the program.
9. It is the same to use instruction REFF or change the content of D1020 and D1021.

Function Group Communication Error Code

Number M1025, D1025

Contents:
Slave mode error code when receiving incorrect communication request:

01: illegal instruction code
02: illegal device address.
03: requested data exceeds the range.
07: checksum error

Function Group Execution Completed Flag

Number M1029, M1030, M1036, M1037, M1102, M1103

Contents:
Execution Completed Flag:

MTR, HKY, DSW, SEGL, PR:
M1029=ON for a scan period once the instruction finishes executing.

PLSY, PLSR:
1. M1029 will be “ON” after Y0 pulse output of ELC-PA/PB, ELCB-PB is completed. M1030 will be “ON” after Y1 pulse output is completed. M1102 will be ON after Y2 pulse output is completed. When PLSY, PLSR instruction are OFF, M1029, M1030, M1102, M1103 will be OFF as well. When pulse output instructions executes again, M1029, M1030, M1102, M1103 will be OFF and turn ON when execution completes. You have to reset M1029 and M1030 after the action is completed.
2. M1029 will be “ON” after Y0 and Y1 pulse output of ELC- PV, ELC2-PV is completed. M1030 will be “ON” after Y2 and Y3 pulse output is completed. M1036 will be “ON” after Y4 and Y5 pulse output of ELC-PV is completed. M1037 will be “ON” after Y6 and Y7 pulse output is completed. When PLSY and PLSR instruction is “OFF”, M1029, M1030, M1036 and M1037 turn “OFF”. When the instruction is re-executed for the next time, M1029, M1030, M1036 and M1037 will turn “OFF” and “ON” again when the execution is completed.
3. ELCM-PH/PA M1029 = ON when Y0 pulse output completes. M1030 = ON when Y1 pulse output completes. M1102 = ON when Y2 pulse output completes. M1103 = ON when Y3 pulse output completes. When PLSY, PLSR instruction are OFF, M1029, M1030, M1102,

M1103 will be OFF as well. When pulse output instructions executes again, M1029, M1030, M1102, M1103 will be OFF and turn ON when execution completes.

4. User should clear M1029, M1030, M1036, M1037, M1102 and M1103 manually.

INCD:
M1029 will be ON for a scan period when the designated group finishes comparison.

RAMP, SORT:
1. M1029= ON after completing execution, M1029 must be cleared by user.
2. If this instruction is OFF, M1029 will be OFF.

DABSR:
1. M1029 = On when the 1st output group Y0 and Y1 of ELC-PV, ELC2-PV is completed. M1030 = On when the 2nd output group Y2 and Y3 is completed.
2. M1036 = On when the 3rd output group Y4 and Y5 of ELC-PV is completed. M1037 = On when the 4th output group Y6 and Y7 is completed.
3. When the instruction is re-executed for the next time, M1029 or M1030 will turn “Off” and “On” again when the execution is completed.
4. M1029=ON after completing execution.
5. M1029 will be OFF when execute this instruction the next time and it will be ON after completing execution.

ZRN, DRVI, DRVA:
1. M1029 will be “ON” after Y0 and Y1 pulse output is completed. M1102 will be “ON” after Y2 and Y3 pulse output is compeleted.
2. When the instruction is re-executed for the next time, M1029 / M1102 will turn off first then ON again when the instruction is completed.
3. For ELC-PV, ELC2-PV controllers, M1029 = ON when the first output group Y0 and Y1 is completed. M1030 = ON when the second output group Y2 and Y3 is completed. M1036 = ON when the third output group Y4 and Y5 is completed. M1037 = ON when the fourth output group Y6 and Y7 is completed. When the instruction is re-executed for the next time, M1029 or M1030 will turn “OFF” and “ON” again when the execution is completed.
4. For ELCM-PH/PA, ELC2-PB/PA/PH/PE controllers, M1029 will be “ON” after Y0 and Y1 pulse output is completed. M1102 will be “ON” after Y2 and Y3 pulse output is completed. When the instruction is re-executed for the next time, M1029 / M1102 will turn off first then ON again when the instruction is completed.

Function Group Clear Command

Number M1031, M1032

Contents:
M1031 (clear unlatched area), M1032 (clear latched area)

	The component that will be cleared
M1031 Clear unlatched area	The contact state of Y, general M, general S T contact for general and time coil C contact for general, time coil reset coil D present register for general T present register for general C present register for general
M1032 Clear latched area	The contact state of M and S for latched Accumulative timer T contact and time coil Latched C and high-speed counter C contact, count coil Present register D for latched Present register of accumulative timer T Latched C and present register of high-speed counter C

Function Group M1033

Number Output Latched in STOP mode

Contents:
When M1003 = ON, the ELC outputs will be latched in their current state when the ELC is switched from RUN to STOP.

Function Group Turn all outputs off

Number M1034

Contents:
When M1034 = ON, all outputs will turn off.

Function Group RUN/STOP Switch

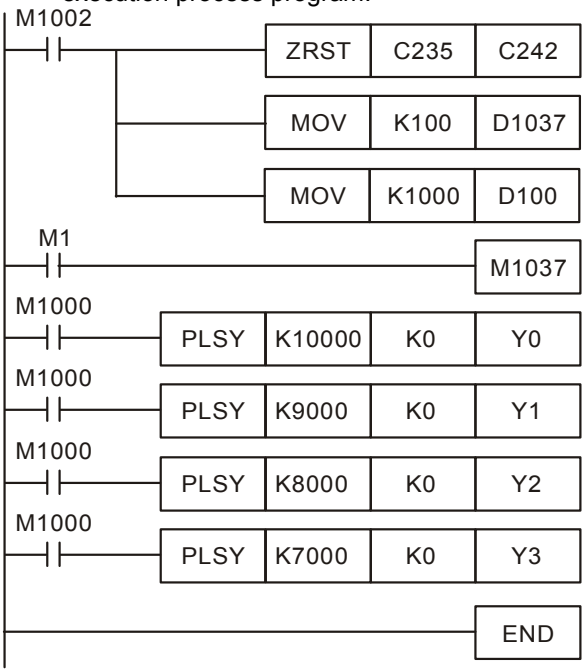
Number M1035, D1035

Contents:
1. When M1035 = ON, ELCM-PH/PA, ELC2-PB/PH/PA/PE use input point X7, ELC-PA use input point X3 . ELCB-PB does not support.
2. When M1035 = ON, ELC-PV, ELC2-PV controllers will determine the content (K0 ~ K15) in D1035 to enable input points X0 ~ X17 as the RUN/STOP switch.

Function Group Enable SPD function

Number M1037, D1037

Contents:
1. M1037 and D1037 can be used to enable 8 sets of SPD instructions. When M1037 is ON, 8 sets of SPD instructions will be enabled. When M1037 is OFF, the function will be disabled.
2. The detected speed will be stored in the registers designated by D1037, e.g. if D1037 = K100, the user has to set up the value in D100, indicating the interval for capturing the speed value (unit: ms). In addition, the captured speed value will be stored in D101 ~ D108 in order.
   ※ When the function is enabled, C235~C242 will be occupied and unavailable in ELC execution process program.



Function Group Communication Response Delay

Number D1038

Contents:

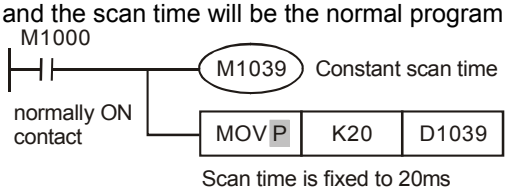
1. When ELC is used as slave station, in RS-485 communication interface, users can set up communication response delay time ranging from 0 to 10,000 (0 ~ 1 second). If the time is without the range, D1038 = O (time unit: 0.1ms). The set value of time must be less than that in D1000(scan time-out timer WDT).
2. By using ELC-Link (Master), D1038 can be set to send next communication data with delay. (Unit: 1 scan period for ELC-PA, ELCM-PH/PA, ELC2-PB/PH/PA/PE, 0.1ms for ELC-PV, ELC2-PV)

Function Group Constant scan time

Number M1039, D1039

Contents:

1. When M1039 is ON, program scan time is determined by D1039 (if D1039 is greater than the maximum scan time). When program finishes executing, it will execute the next scan after the time set by D1039. If D1039 is less than program scan time, the program scan will complete, and the scan time will be the normal program scan time.



2. The relative instructions of scan time are RAMP, HKY, SEGL, ARWS and PR. They should be used with “constant scan time” or “constant time insert interrupt”.
3. Especial for instruction HKY, scan time should be set to 20ms and above when it is used 4×4 matrix to be 16 keys to operate.
4. Scan time D1010~D1012 display also include constant scan time.

Function Group Analog Function

Number D1056~D1057, D1062, D1110~D1113, D1116~D1118

Contents:

1. ELC-PA, ELC2-PA, ELCM-PA Controllers only.
2. Resolution of analog input channel: 12 bits.
   Voltage: -10V~10V ⇔ Value: -2000~2000.
   Current: -20mA~20mA⇔ Value: -2000~2000.
   Current: 4mA~20mA⇔ Value: 0~2000.
3. Resolution of analog output channel: 12 bits
   Voltage: -10V~10V ⇔ Value: -2000~2000.
   Current: 0~20mA ⇔ Value: 0~4000.
   Current: 4mA ~20mA ⇔ Value: 0~4000.
4. ELCM-PA, default of analog input average times: (K2). If set value = K1, the ELC takes the present value.

	Function
D1056	Present value of analog input channel 0 (CH0), ELC-PA support.
D1057	Present value of analog input channel 1 (CH 1), ELC-PA support.
D1062	For ELC-PA controllers, average times of AD (CH0, CH1): 2~4, Default = K2 For ELCM-PA/ELC2-PA controllers, average times of AD (CH0~CH3): 1~20, Default = K2
D1110	Average value of analog input channel 0 (AD 0)
D1111	Average value of analog input channel 1 (AD 1)
D1112	Average value of analog input channel 2 (AD 2), ELC-PA not support.
D1113	Average value of analog input channel 3 (AD 3), ELC-PA not support.

D1114	Enable/disable ELCM-PA/ELC2-PA AD channels (0: enable (default) / 1: disable) bit0~bit3 sets AD0~AD3.
D1115	Analog mode selection (Voltage / Current) 0: voltage, 1: current (Default: voltage) bit0~bit3 refer to AD0~AD3 bit4~bit5 refer to DA0, DA1 bit8~bit13: current mode selection bit8~ bit11 refer to AD0~AD3, 0: -20mA~20mA, 1: 4~20mA bit12, bit13 refer to DA0~DA1, 0: 0~20mA, 1: 4~20mA
D1116	Analog output channel 0 (DA 0).
D1117	Analog output channel 1 (DA 1).
D1118	Analog input filter setting (ms). For ELC-PA controllers, sampling time of analog/digital conversion Sampling time will be regarded as 5ms If D1118≦5. For ELCM-PA/ELC2-PA controllers, sampling time of analog/digital conversion Sampling time will be regarded as 2ms If D1118≦2.

Function Group Algorithm Error Flag

Number M1067~M1068, D1067~D1068

Contents:

Algorithm error flag:

Component	Explanation	Latched	STOP→RUN	RUN→STOP
M1067	Algorithm error flag	None	Clear	Latched
M1068	Algorithm error lock flag	None	Unchanged	Latched
D1067	Algorithm error code	None	Clear	Latched
D1068	STEP value of algorithm error	None	Unchanged	Latched

Error code explanation:

D1067 error code	Function
0E18	BCD conversion error
0E19	Divisor is 0
0E1A	Usage exceeds limit (include E and F)
0E1B	It is negative number after doing radical
0E1C	FROM/TO communication error

Function Group ELC- PA, ELCB-PB series, X0 input point can detect pulse width

Number M1084、D1023

Contents:

X0 input point of ELCB-PB/ELC-PA controllers can detect pulse width. Whenever X0 turns from ON to OFF, the value will be updated once and stored in D1023 (unit: 0.1ms). The minimum detectable width is 0.1ms and maximum 10,000ms.

Function Group ELCM-PH/PA, ELC2-PB/PA/PH/PE series, X6 pulse width detecting function

Number M1083,M1084, D1023

Contents:

When M1084 = ON, X6 pulse width detecting function is enabled and the detected pulse width is stored in D1023 (unit: 0.1ms)

M1083 ON：detecting width of negative half cycle (OFF→ON)

M1083 OFF：detecting width of positive half cycle (ON→OFF)

Function Group Pulse output Mark and Mask function

Number M1108, M1110, M1156, M1158, M1538, M1540, D1026, D1027, D1135, D1136, D1232, D1233, D1234, D1235, D1348, D1349

Contents:
Please refer to explanations of API 59 PLSR / API 158 DDRVI / API 197 DCLLM instructions.

Function Group Low Voltage, for ELC-PV, ELC2-PV series.

Number M1087, D1100

Contents:
1. When the ELC detects LV (Low Voltage) signal, it will check if M1087 is “ON” or not. If M1087 is “ON”, the content in D1100 will be stored in Y0 ~ Y17.
2. Bit0 (LSB) of D1100 corresponds to Y0, bit1 corresponds to Y1, bit8 corresponds to Y10 and so on.

Function Group File Register

Number M1101, D1101~D1103

Contents:
1. Only support ELC-PA/PV, ELC2-PV.
2. When the ELC is powered on or from STOP to RUN, it will check start file register function from M1101, the start number of file register from D1101, read item number of file register from D1102, to determine if it should send file register data to the designated data register automatically or not.
   M1101: Whether to automatically downland data from file register
   D1101: Start No. of file register K0 ~ K1,599 (for ELC-PA)
   Start No. of file register K0 ~ K9,999 (for ELC-PV, ELC2-PV)
   D1102: Number of data read from file register K0 ~ K1,600 (for ELC-PA)
   Number of data read from file register K0 ~ K8,000 (for ELC-PV, ELC2-PV)
   D1103: Location for storing data read from file register
   Start No. of assigned data register D K2,000 ~ K4,999 (for ELC-PA)
   Start No. of assigned data register D K2,000 ~ K9,999 (for ELC-PV)
   Start No. of assigned data register D K2,000 ~ K9,999 (for ELC2-PV)
3. Please refer to instructions MEMR and MEMW explanation.

Function Group Pulse Output With Speed Acceleration/Deceleration

Number M1115 ~ M1119, D1104

Contents:
1. Special D and special M for acceleration/ deceleration of speed pulse output for ELCB-PB/ELC-PA:

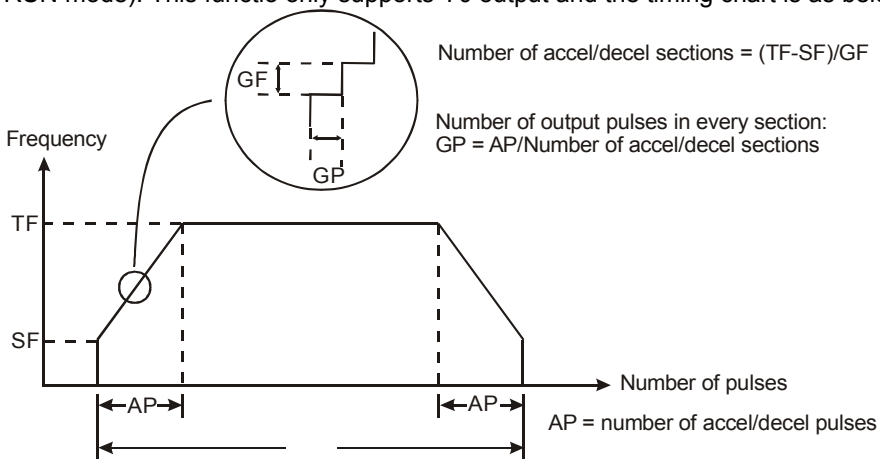
Device No.	Function
M1115	Activation switch
M1116	“Accelerating” flag
M1117	“Target frequency reached” flag
M1118	“Decelerating” flag
M1119	“Function completed” flag
D1104	Start No. of control register (D)

2. Parameters for D1104 (frequency range: 25Hz ~ 10kHz)

Index	Function	
+ 0	Start frequency (SF)	
+ 1	Gap frequency (GF)	
+ 2	Target frequency (TF)	
+ 3	The lower 16 bits of the 32 bits for the total number of output pulses	(TP)

Index	Function	
+ 4	The higher 16 bits of the 32 bits for the total number of output pulses	(TP)
+ 5	The lower 16 bits of the 32 bits for the total number of output pulses in accelerating/decelerating section	(AP)
+ 6	The higher 16 bits of the 32 bits for the total number of output pulses in accelerating/decelerating section	

3. No instruction is needed, users need only to fill out the parameter table and enable M1115 (in RUN mode). This functio only supports Y0 output and the timing chart is as below.



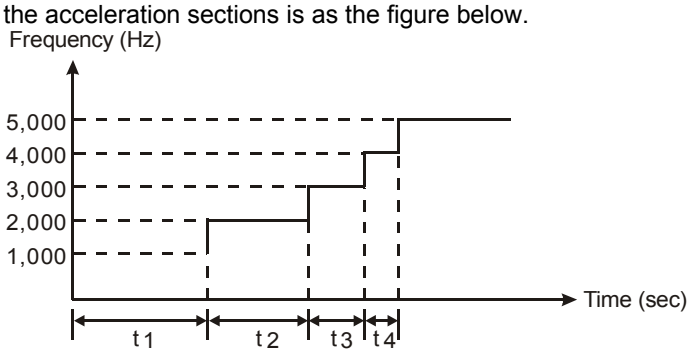
4. Note: this function is applicable only when “all” the conditions below are met.
   a) Start frequency < target frequency.
   b) Gap frequency ≤ (target frequency – start frequency)
   c) Total number of pulses > (accel/decel number of pulses × 2)
   d) For start frequency and target frequency: Min. 25Hz; Max. 10kHz
   e) Number of accel/decel pulses > number of accel/decel sections

When M1115 turns from “On” to “Off”, M1119 will be reset and M1116, M1117 and M1118 remain unchanged. When ELC goes from “STOP” to “RUN”, M1115 ~ M1119 will be reset as “Off”. D1104 will only be cleared as “0” when it turns from “Off” to “On”.

Either accel/decel pulse output function or PLSY Y0 output can be executed at a time when ELC is operating.

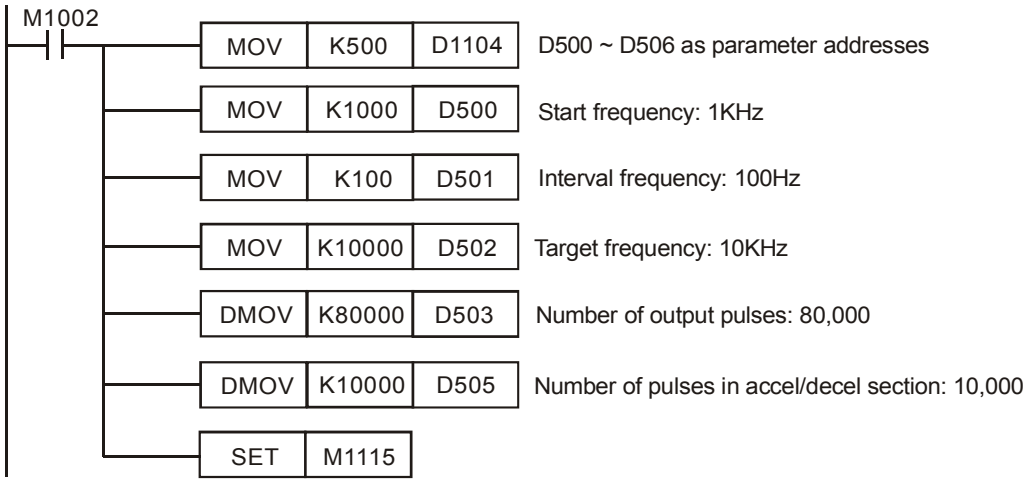
5. How to calculate the action time of each section

   Assume the start frequency is set as 1kHz, gap frequency as 1kHz, target frequency as 5kHz, total number of pulses as 100 and number of acceleration pulses as 40, the timing diagram of the acceleration sections is as the figure below.

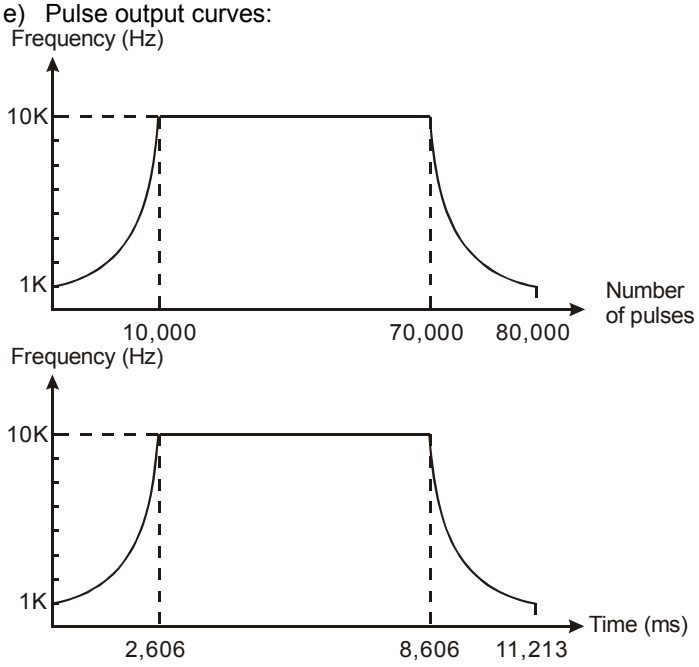


   From the conditions above, we can obtain the number of acceleration/deceleration sections is (5K – 1K)/1K = 4 and the number of output pulses in each section is 40/4 = 10. Therefore, in the diagram, t1 = (1/1K) × 10 = 10ms, t2 = (1/2K) × 10 = 5ms, t3 = (1/3K) × 10 = 3.33ms, t4 = (1/4K) × 10 = 2.5ms.

6. Program example: Forward/reverse acceleration/deceleration step motor control



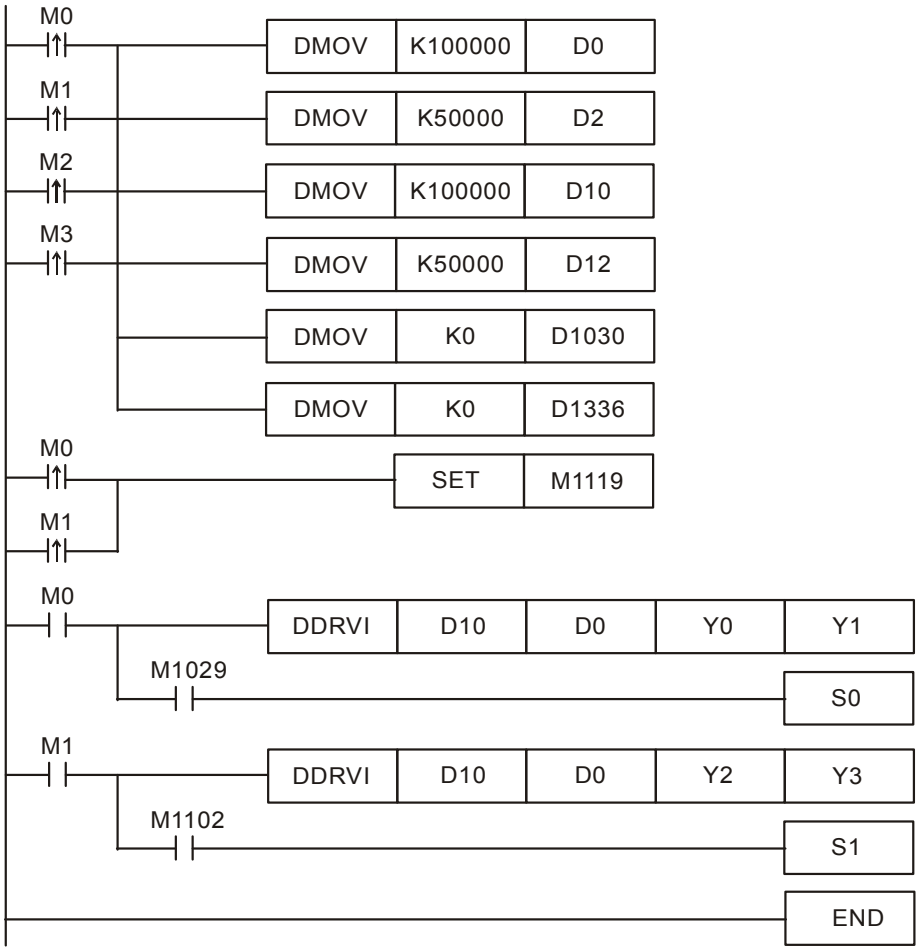
a) When ELC is in RUN status, store all parameter settings into the registers designated in D1104.
b) When M1115 = On, the acceleration/deceleration pulse output will start.
c) M1116 = On in the acceleration process. When the speed reaches its target, M1117 will be On. M1118 = On in the deceleration process. When the speed reaches its target, M1119 will be On.
d) M1115 will not be reset automatically. You have to check the conditions during the process and reset it.
e) Pulse output curves:

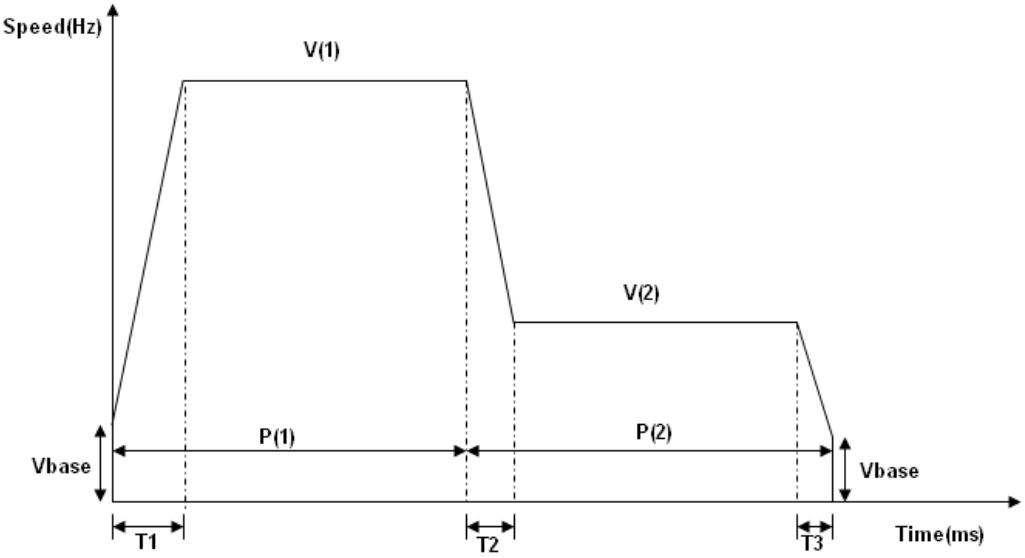


Function Group Enable 2-speed output function of DDRVI instruction

Number M1119

Contents:
ELCB-PB, ELC-PA did not support.
When M1119 is ON, 2-speed output function of DDRVI will be enabled.
Example: Assume that D0 (D1) is the first speed and D2(D3) is the second speed. D10(D11) is the output pulse number of the first speed and D12(D13) is the output pulse number of the second speed.





Vbase	T1	T2+T3	P(1)	V(1)	P(2)	V(2)
Initial frequency	Ramp-up time	Ramp-down time	Position of the first speed	The first speed	Position of the second speed	The second speed

Function Group COM Port Function

Number

Item \ Port	COM1	COM2	COM3
Communication format	D1036	D1120	D1109
Communication setting holding	M1138	M1120	M1136
ASCII/RTU mode	M1139	M1143	M1320
Slave communication address	D1121		D1255

Contents:

The communication interfaces which are supported are listed below. (All the ports can carry out communication simultaneously.):

COM port	COM1	COM2	COM3
ELC-PA/PV, ELCB-PB, ELC2-PB/PV	RS-232	RS-485	-
ELCM-PH/PA, ELC2-PC	RS-232	RS-485	RS-485
ELC2-PE	USB	RS-485	RS-485
ELC2-PA	RS-232	RS-485	USB

Communication protocol of master/slave ELC:

COM port	COM1	COM2	COM3
Configuration Register	D1036	D1120	D1109
ELC-PA/PV, ELCB-PB	Slave	Slave/Master	-
ELCM-PH/PA, ELC2-PC	Slave/Master	Slave/Master	Slave/Master
ELC2-PB/PV	Slave/Master	Slave/Master	-
ELC2-PE	Slave	Slave/Master	Slave/Master
ELC2-PA	Slave/Master	Slave/Master	Slave

You can use D1036/D1120/D1109 to set communication protocol of master/slave ELC. Both support communication format of MODBUS ASCII/RTU, and you can use M1139/M1143/M1320 to set ASCII/RTU.

Communication port summary:

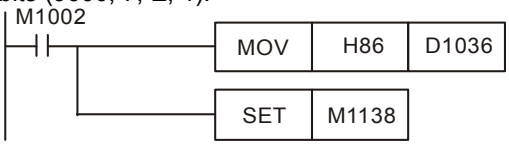
Parameter \ COM	COM1	COM2	COM3)
Baud rate	ELCB-PB, ELC-PA/PV, ELC2-PV:110~115200 bps In ELCM-PH/PA, ELC2-PB/PH/PA/PE, COM1: 110~115200 bps, COM2/COM3: 110~921,000 bps.		
Data length	7~8bits		
Parity	Even/Odd/None parity check		
Stop bits length	1~2 bits		
Configuration Register	D1036	D1120	D1109
communication mode	ASCII/RTU Mode		
Data length for access	100 registers (Both of ASCII/RTU)		

For more information about setting configuration registers (D1036, D1120, D1109), please refer to the table below.

	Content		
b0	Data Length	0: 7 data bits, 1: 8 data bits (RTU supports 8 data bits only)	
b1 b2	Parity bit	00: None 01: Odd 11: Even	
b3	Stop bits	0: 1 bit, 1: 2bits	
b4 b5 b6 b7	Baud rate	0001(H1): 110 0010(H2): 150 0011(H3): 300 0100(H4): 600 0101(H5): 1200 0110(H6): 2400 0111(H7): 4800 1000(H8): 9600 1001(H9): 19200 1010(HA): 38400 1011(HB): 57600 1100(HC): 115200 1101(HD): 500000 (COM2 / COM3) 1110(HE): 31250 (COM2 / COM3) 1111(HF): 921000 (COM2 / COM3)	
b8	Select start bit	0: None	1: D1124 (COM2)
b9	Select the 1st end bit	0: None	1: D1125 (COM2)
b10	Select the 2nd end bit	0: None	1: D1126 (COM2)
b11~b15	Undefined		

Example 1: Modifying COM1 communication format

1. Add the below instructions on top of the program to modify the communication format of COM1. When the ELC switches from STOP to RUN, the program will detect whether M1138 is ON in the first scan. If M1138 is ON, the program will modify the communication settings of COM1 according to the value set in D1036
2. Modify COM1 communication format to ASCII mode, 9600bps, 7 data bits, even parity, 1 stop bits (9600, 7, E, 1).

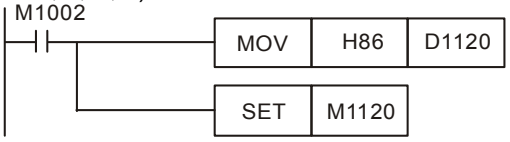


Note:

1. After the communication format is modified, the format will stay intact when the ELC switches from RUN to STOP.
2. Communication format will be reset to default setting after power is shut down.

Example 2: Modifying COM2 communication format

1. Add the below instructions on top of the program to modify the communication format of COM2. When the ELC switches from STOP to RUN, the program will detect whether M1120 is ON in the first scan. If M1120 is ON, the program will modify the communication settings of COM2 according to the value set in D1120
2. Modify COM2 communication format to ASCII mode, 9600bps, 7 data bits, even parity, 1 stop bits (9600, 7, E, 1)

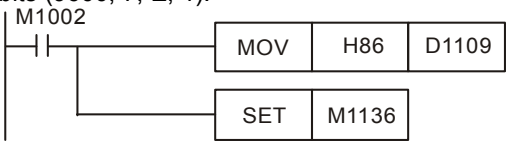


Note:

1. Do NOT write any communication instruction in the program when COM2 is used as slave.
2. After the communication format is modified, the format will stay intact when the ELC switches from RUN to STOP.
3. Communication format will be reset to default setting after power is shut down.

Example 3: Modifying COM3 communication format

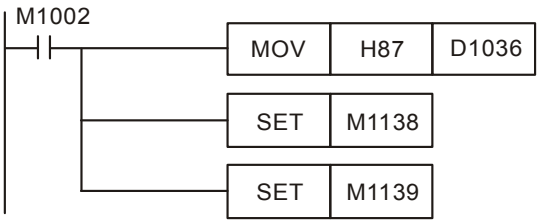
1. Add the below instructions on top of the program to modify the communication format of COM3. When the ELC switches from STOP to RUN, the program will detect whether M1136 is ON in the first scan. If M1136 is ON, the program will modify the communication settings of COM3 according to the value set in D1109
2. Modify COM3 communication format to ASCII mode, 9600bps, 7 data bits, even parity, 1 stop bits (9600, 7, E, 1).



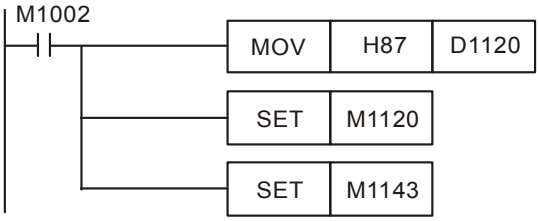
Example 4: RTU mode setting of COM1、COM2、COM3

1. COM1, COM2 and COM3 support ASCII/RTU mode. COM1 is set by M1139, COM2 is set by M1143 and COM3 is set by M1320. Set the flags ON enables RTU mode while OFF enables ASCII mode.
2. Modify COM1/COM2/COM3 communication format to RTU mode, 9600bps, 8 data bits, even parity, 1 stop bits (9600, 8, E, 1).

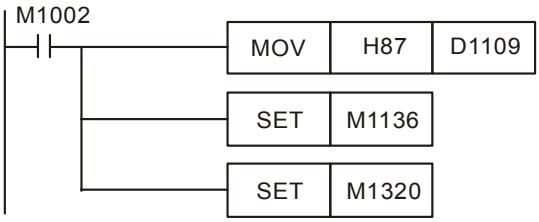
COM1:



COM2:



COM3:

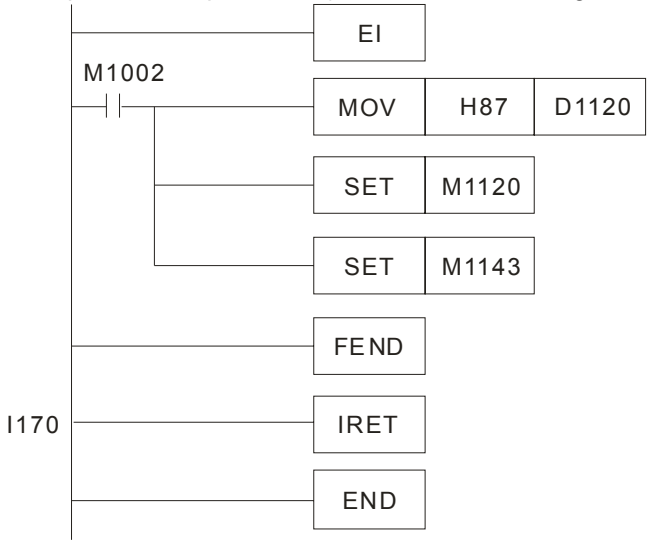


Note:

1. The modified communication format will not be changed when the ELC state turns from RUN to STOP.
2. If the ELC is powered OFF then ON again after communication format is modified, COM1~COM3 will be reset to default communication format (9600, 7, E, 1).

Example 5: RTU mode of setting COM2 with the generation of interruption I170.

1. Only ELC-PV/ELC2-PV support the generation of interruption I170 when the data receiving is completed in Slave mode.
2. Normally when the communication terminal of the ELC is in Slave mode, ELC will not immediately process the communication data entered but process it after the END is executed. Therefore, when the scan time is very long and you need the communication data to be processed immediately, you can use interruption I170 for this matter.
3. Example of interruption I170 (after the data receiving is completed in Slave mode)



With I170 in the program, when COM2 is in Slave mode and there are communication data coming in, ELC will process the data and respond immediately.

Notes:

1. DO NOT updat program on-line when using I170.
2. The scan time of ELC will be slightly longer.

Function Group Special High-speed pulse output (ELC-PA)

Number M1133~M1135, D1133~D1136

Contents:

ELC- PA controllers: Special High-speed pulse output

1. The definition of special D and special M for special high-speed pulse (50KHz) output function:

	Function
M1133	Output switch (ON is start executing) for special high-speed pulse (50KHz)
M1134	ON is continuous output switch for special high-speed pulse Y0 (50KHz)
M1135	Output pulse number attained flag for special high-speed pulse Y0 (50KHz)
D1133	Start number of control register (D) for special high-speed pulse Y0 (50KHz)

2. Corresponding table for D1133 parameter

Index	Function
+0	Special high-speed pulse output frequency (lower 16-bit of 32 bits)
+1	Special high-speed pulse output frequency (upper 16-bit of 32 bits)
+2	Special high-speed pulse output number (lower 16-bit of 32 bits)
+3	Special high-speed pulse output number (upper 16-bit of 32 bits)
+4	Display present special high-speed pulse output number (lower 16-bit of 32 bits)
+5	Display present special high-speed pulse output number (upper 16-bit of 32 bits)

Function explanation:

1. Output frequency and output numbers above can be modified when M1133=on and M1135=OFF. It will not affect present output pulse once output

2. Frequency or output target number is changed. Present output pulse number will be displayed once a scan time update. It will be cleared to 0 when M1133 is from OFF→ON and it will keep that last output number when M1133 is from ON→OFF.

Note:

1. This special high-speed pulse output function can use special Y0 output point in RUN. It can exist with PLSY Y0 at the same time and PLSY (Y1) will not be affected. If instruction PLSY (Y0) is executed prior to this function, this function cannot be used and vice versa. When executing this function, general Y0 output will be invalid and outputs point of Y1~Y7 can be used.
2. The difference between this function and instruction PLSY is higher than output frequency. The maximum output can up to 50KHz.

Function Group Expansion Connected Detection

Number D1140, D1142, D1143, D1145

Contents:

D1140: Number of right-side modules (AIO, PT, TC, etc.), max. 8 modules can be connected.

D1142: Digital expansion input X point number.

D1143: Digital expansion input Y point number.

D1145: Number of special left-side expansion modules (Analog in, Analog out, PT, TC, etc.); Max. 8 (available in ELC-PV, ELC2-PC/PA/PE/PV only)

Function Group Adjustable Acceleration/Deceleration Pulse Output Function Explanation

Number M1144~M1149, M1154, D1030, D1031, D1144, D1154, D1155

Contents:

1. For the ELC-PA Controllers, the definition of special D and special M of adjustable accel/decel pulse output function:

	Function
M1144	Start switch of accel/decel pulse output
M1145	Flag that is used in acceleration
M1146	Target frequency attained flag
M1147	Flag that is used in deceleration
M1148	Completed function flag
M1149	stop counting temporarily flag
M1154	Start designated deceleration gap time flag and frequency flag
D1030	Lower 16-bit of 32-bit of Y0 pulse accumulative output numbers
D1031	Upper 16-bit of 32-bit of Y0 pulse accumulative output numbers
D1144	Using parameter index (correspond to D component)
D1154	Recommended value of designated deceleration gap time (10~32767 ms)
D1155	Recommended value of designated acceleration gap frequency (-1~ - 32700 Hz)

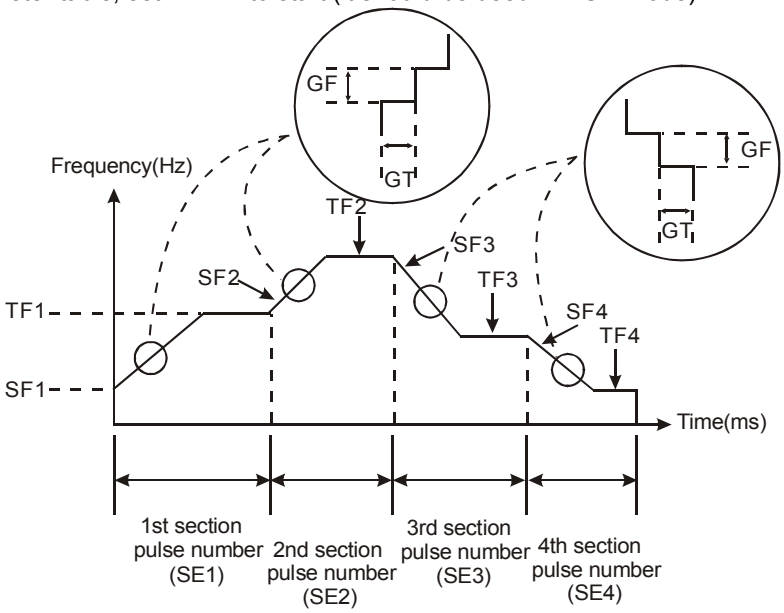
2. Corresponding table of parameter D1144

Index	Function
+0	Total segment number (n) (the maximum number is 10)
+1	Present execution segment (read only)
+2	Start frequency of first segment (SF1)
+3	Interval time of first segment (GT1)
+4	Interval frequency of first segment (GF1)
+5	Target frequency of first segment (TF1)
+6	Lower 16-bit of 32-bit of target number of first segment output pulse
+7	Upper 16-bit of 32-bit of target number of first segment output pulse
+8	Start frequency of second segment (SF2)
+9	Interval time of second segment (GT2)
+10	Interval frequency of second segment (GF2)
+11	Target frequency of second segment (TF2)

Index	Function
+12	Lower 16-bit of 32-bit of target number of second segment output pulse
+13	Upper 16-bit of 32-bit of target number of second segment output pulse
:	:
+n*6+2	Start frequency of nth segment (SFn)
+n*6+3	Interval time of nth segment (GTn)
+n*6+4	Interval frequency of nth segment (GFn)
+n*6+5	Target frequency of nth segment (TFn)
+n*6+6	Lower 16-bit of 32-bit of target number of nth segment output pulse
+n*6+7	Upper 16-bit of 32-bit of target number of nth segment output pulse

Function Explanation:

This function can only be used for Y0 output point and the timing will be as follows. After filling parameter table, set M1144 to start (it should be used in RUN mode)



Usage rule and restriction:

1. The minimum frequency of start frequency and target frequency should be equal to or greater than 200Hz. If it is less than 200Hz, it means finish executing or not to execute.
2. The maximum frequency of start frequency of target frequency is 32,700Hz. It will execute in 32,700Hz as it is greater than 32,700Hz.
3. The interval time range is 1~32,767ms and its unit is ms.
4. The interval frequency range in acceleration segment is 1Hz~32,700Hz and in deceleration segment is -1~-32,700Hz. If it is set to 0Hz, the executed segment cannot be up to target frequency, but it will transfer to execute next segment after reaching target number.
5. Target number of segment pulse output should be greater than ((GF*GT/1000)* ((TF-SF)/GF). Refer to example 1 for detail. Once Target number of segment pulse output isn't greater than ((GF*GT/1000)* ((TF-SF)/GF), this function cannot be used. The improve method is to add interval time or add target number of pulse output.
6. If there is Y0 output designated by high-speed instruction in RUN mode, Y0 output
7. Instruction will be started as high priority.
8. After starting to execute M1144, if M1148 outputs without attaining completed function flag and M1144 is closed, this function will start deceleration function. If designated acceleration function flag M1154 is OFF, it will reduce 200Hz per 200ms and stop output pulse till output frequency is less than 200Hz and set M1147 to deceleration flag. But if designated deceleration flag M1154 is ON, it will be executed by interval time and frequency that defined by user. And interval time cannot be less than or equal to 0 (if it is less than or equal to 0, factory setting will be set to 200ms). Interval frequency cannot be greater than or equal to 0

(factory setting will be set to -1KHz when it is equal to 0 and factory setting will be added negative sign automatically when it is greater than 0.)

9. When M1148 attains completed function flag and M1144 is closed, this function will not start deceleration function and it will clear M1148 flag. Once M1144 is closed, it will clear M1149 flag.
10. The execution segment of this function will execute by total segment number. The maximum segment is 10 segments.
11. The acceleration/deceleration of this function will execute by start frequency of the next segment, i.e. when target frequency of execution segment is less than start frequency of the next segment, the next segment is acceleration and the target frequency of the next segment must be greater than start frequency of the next segment. When target frequency of execution segment is greater than the next segment frequency, the next segment is deceleration; therefore, target frequency of the next segment must be less than start frequency of the next segment. If user cannot set it by this way, we cannot ensure that you can get correct output pulse.
12. When STOP→RUN, M1144~M1149 will be cleared to OFF. When RUN→STOP, M1144 will be cleared and M1145~M1149 will not be cleared. D1144 will be cleared to 0 when it is from OFF→ON and unchanged in other case.
13. The valid parameter range is D0~D999 and D2000~D4999. the ELC will not execute this instruction, and close M1144 if the parameter is out of range (includes all segment parameters).

Example 1:

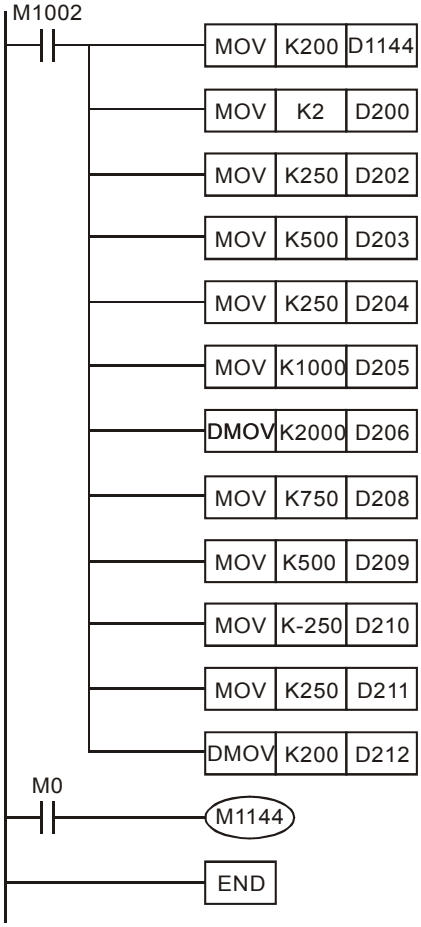
To calculate output number of acceleration/deceleration of each segment and target frequency If setting start frequency of segment to 200Hz, segment interval time to 100ms, segment gap frequency to 100Hz, segment target frequency to 500Hz and target number of segment pulse is 1000 pulses. The calculation will be in the following:

1. Output pulse number at start acceleration/deceleration is 200*100/1000 = 20 pulses
2. Output pulse number of the first acceleration interval is 300*100/1000 = 30 pulses
3. Output pulse number of the second acceleration interval is 400*100/1000 = 40 pulses
4. Output pulse number of target frequency is 1000 – (40+30+20) = 910 pulses

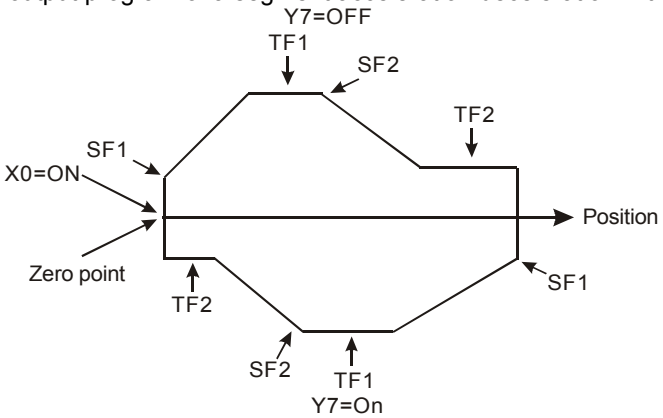
(NOTE: it is recommended to set this number to be greater than 10)

5. Output time of target frequency is 1 / 500 * 910 = 1820 ms
6. Total time of this segment is 1820 + 3*100 = 2120 ms

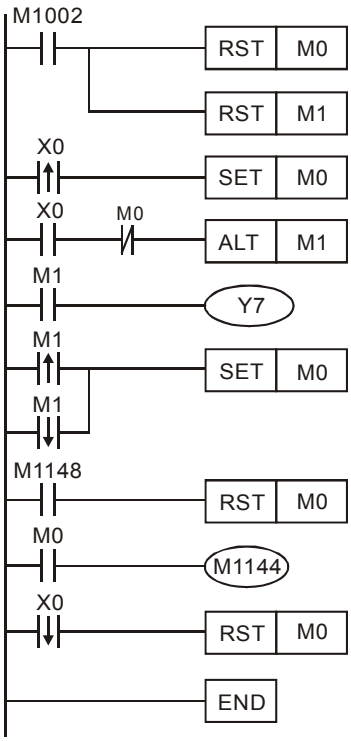
Example 2:
Simple acceleration/deceleration pulse output program of a segment acceleration and a segment deceleration



Example 3:
Pulse output program of a segment acceleration/deceleration with direction

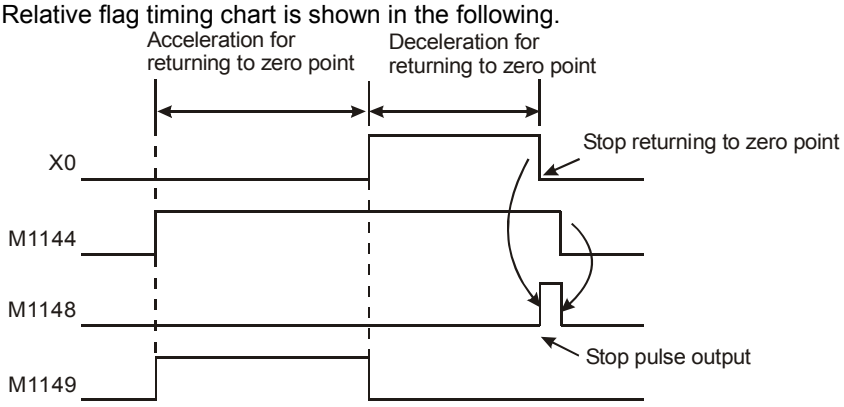


Explanation:
1. Acceleration/deceleration setting is as example 2.
2. Figure above is the example of position movement. When X0 contact is ON, it will start to move and it will stop when X0 contact is OFF. (Y7 is for direction setting)
3. Program is shown in the following.

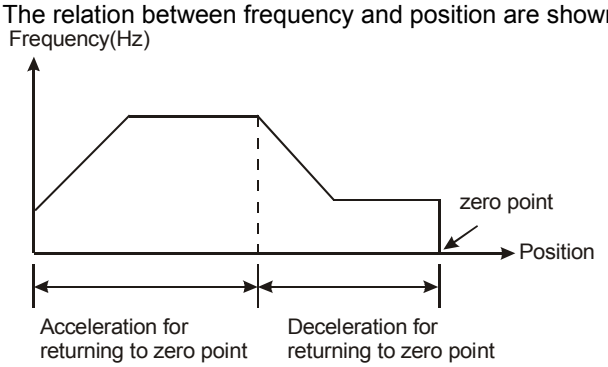


Example 4: apply acceleration and deceleration of a segment to zero point return program.

1. Relative flag timing chart is shown in the following.



2. The relation between frequency and position are shown in the following.

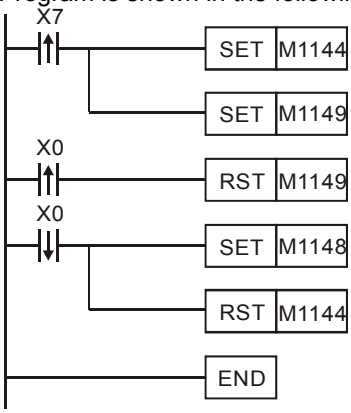


3. Number setting of acceleration/deceleration, frequency and pulse are shown in the following. (correspond to component D)

Index	Settings
+0	2
+2	250(Hz)

Index	Settings
+3	100(ms)
+4	500(Hz)
+5	10000(Hz)
+6, +7	10(pulse)
+8	9750(Hz)
+9	50(ms)
+10	-500(Hz)
+11	250(Hz)
+12, +13	30000(pulse)

4. Program is shown in the following: (it assumes contact X7 to be start reset trigger switch)



5. Explanation:
   a) After contact X7 is triggered, M1144 will set to start acceleration and set M1149 not to count pulse number. And it will send 10 pulses once deceleration switch X0 is triggered and then enter deceleration segment.
   b) To set M1148 to end pulse output by manual and close this function once X0 is closed.
   c) Note: This example is just an application method that user should adjust parameters settings used in acceleration/deceleration segment according to actual machine characteristics and limitation.

Function Group Single Step Execution

Number M1170, M1171, D1170

Contents:

1. Special D and special M for single step execution for the ELC-PV, ELC2-PV:

	Function
M1170	Start flag
M1171	Action flag
D1170	STEP No. of the currently executed instruction

2. The function:
   a. Execution timing: The flag is valid only when the ELC is in RUN status.
   b. Action steps:
      i) When M1170 is enabled, the ELC enters the single step execution mode. the ELC stays at a specific instruction, stores the location of STEP in D1170 and executes the instruction once.
      ii) When M1171 is forced “ON”, the ELC executes the next instruction and stops. At the same time, the ELC auto-force “OFF” M1171 and stops at the next instruction. D1170 stores the present STEP value.
      iii) When Y output is in single step execution mode, Y outputs immediately without having to wait until END instruction is being executed.
3. Note:
   a) Instruction that will be affected by scan time will be executed incorrectly due to the single step execution. For example, when HKY instruction is executed, it takes 8 scan times to

obtain a valid input value from a key. Therefore, the single step execution will result in incorrect actions.

b) High-speed pulse input/output and high-speed counter comparison instructions are executed by hardware; therefore, they will not be affected by the single step execution.

Function Group 2-phase Pulse Output Function

Number M1172~M1174, D1172~D1177

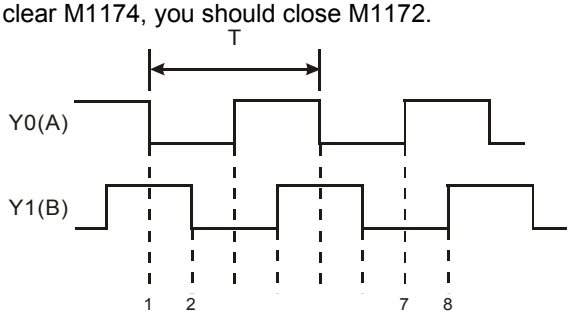
Contents:

For the ELC-PA Controllers, the definition of special D and special M of 2-phase output function:

	Function Explanation
M1172	2-phase pulse output switch
M1173	ON is continuous output switch
M1174	Output pulse number attained flag
D1172	2-phase output frequency (12Hz~20KHz)
D1173	2-phase output mode selection (k1and k2)
D1174	Lower bit of 32-bit of 2-phase output pulse target number
D1175	Upper bit of 32-bit of 2-phase output pulse target number
D1176	Lower bit of 32-bit of 2-phase present output pulse number
D1177	Upper bit of 32-bit of 2-phase present output pulse number

Function Explanation:

1. Output frequency = 1/T as shown in the figure below. There are two output modes, k1 and k2, k1 means A phase gets ahead of B phase and k2 means B phase gets ahead of A phase. Output number calculation adds 1 once there is a phase difference, such as figure below, there are 8 output pulses. When output numbers attains, M1174 will be ON and if you want to clear M1174, you should close M1172.



2. Output frequency, output target number and mode selection can be modified when M1172=ON and M1174=OFF. The modification of output frequency and output target number will not affect present output pulse number but mode selection modification will clear present output pulse number to 0. Present output pulse number will be updated once scan time updates and it will clear to 0 when M1172 is from Stop→Run, and keep that last output number when M1172 is from Run→Stop.

Note:

This function just can be used at RUN mode and can exist in program with PLSY instruction. But if instruction PLSY is executed first, this function cannot be used, and vice versa.

Function Group VR Volume

Number M1178~M1179, D1178~D1179

Contents:

For the ELC-PV/ELC2-PV/PH/PE/PA controllers, the definition of special D and special M of built-in 2 points VR Variable resistor function:

	Function
M1178	Start VR0
M1179	Start VR1
D1178	VR0 value

	Function
D1179	VR1 value

Function explanation:

1. This function only can be used at RUN mode. When M1178=ON, the variable value of VR 0 will be converted to digit 0~255 and saved in D1178. When M1179=ON, the variable value of VR 1 will be converted to digit 0~255 and saved in D1179.
2. Refer to instruction VRRD for detail.

Function Group Interrupt instruction for reading pulse number

Number D1180~D1181, D1198~D1199

Contents:

3. In ELC-PA controllers, it is possible to use Interrupt instruction to read the value of high-speed counter and store in D1180~D1181, D1198~D1199.
4. Function:
   a) In ELC-PA V1.2 and above, X0 (counter input) and X4 (external Interrupt) will correspondingly work together with C235, C251, C253, and I401. Use D1180 and D1181 those are total 32 bit to set X0 and X4. X1 (counter input) and X5 (external Interrupt) will correspondingly work together with C236 and I501. Use D1198 and D1199 those are total 32 bit to set X1 and X5.

Function Group Power Loss Latched Range Setting

Number D1200~D1219

Contents:

For ELC-PA/PV, ELC2-PV Controllers, to set latched range. The latched range will be from start address number to end address number.

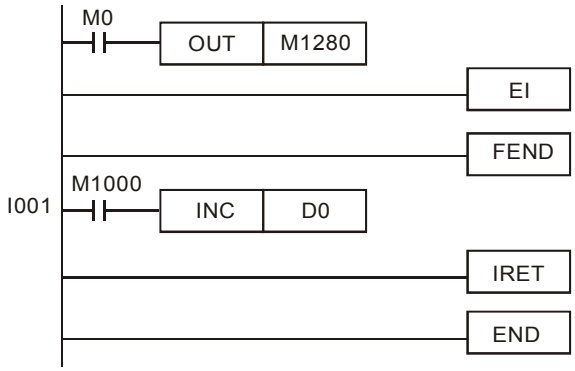
Function Group Reverse Interrupt Trigger Pulse Direction

Number M1280, M1284, M1286

Contents:

1. The flags should be used with EI instruction and should be inserted before EI instruction
2. The default setting of interrupt I101 (X0) is rising-edge triggered. If M1280 is ON and EI instruction is executed, the ELC will reverse the trigger direction as falling-edge triggered. The trigger pulse direction of X1 will be set as rising-edge again by resetting M1280.
3. When M0 = OFF, M1280 = OFF. X0 external interrupt will be triggered by rising-edge pulse.

4. When M0 = ON, M1280 = ON. X0 external interrupt will be triggered by falling-edge pulse. Users do not have to change I101 to I000.



Function Group Stores Value of High-speed Counter when Interrupt Occurs

Number D1240~D1241, D1242~D1243

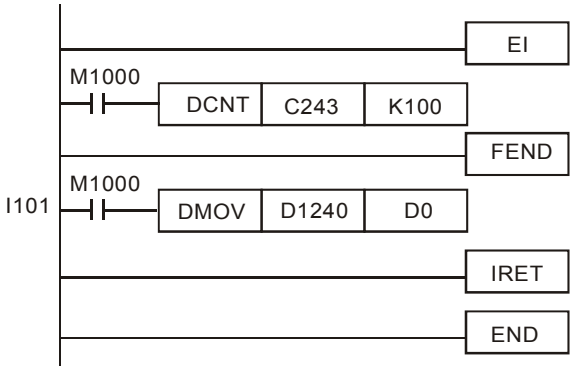
Contents:

1. If external interrupts are applied on input points for Reset, the interrupt instructions have the priority in using the input points. In addition, the ELC will move the current data in the counters to the associated data registers below then reset the counters.

Special D	D1241, D1240				D1243, D1242		
Counter	C243	C246	C248	C252	C244	C250	C254
Interrupt signal	X1	X4			X3	X5	

2. Function:
   a) When X0 (counter input) and X1 (external Interrupt) correspondingly work together with C243, and I100/I101, the ELC will move the count value to D1241 and D1240.
   b) When X0 (counter input) and X4 (external Interrupt) correspondingly work together with C246, C248, C252 and I400/I401, the ELC will move the count value to D1241 and D1240
   c) When X2 (counter input) and X3 (external Interrupt) correspondingly work together with C244, and I300/I301, the ELC will move the count value to D1243 and D1242.
   d) When X2 (counter input) and X5 (external Interrupt) correspondingly work together with C250, C254 and I500/I501, the ELC will move the count value to D1243 and D1242.

Example:



When external interrupt (X1, I101) occurs during counting process of C243, the count value in C243 will be stored in (D1241, D1240) and C243 is reset. After this, the interrupt subroutine I101 will be executed

Function Group Input Point X can force to be ON/OFF

Number M1304

Contents:

1. For ELCB-PB, when M1304 = On, the X input points (X0 ~ X17) on MPU can be set On/Off by peripheral devices. However, the LED indicators will not respond to the setup.
2. For ELC-PA, when M1304 = On, peripheral devices can set On/Off of X0 ~ X17 on the MPU, but the LED indicators will not respond to it.
3. For ELC-PV/ELC2-PV, when M1304 = On, peripheral devices can set On/Off of X input points on the MPU, but the LED indicators will not respond to it.
4. For ELCM-PH/PA, ELC2-PB/PH/PA/PE when M1304 = ON, input point X of the ELC can be forced to be ON-OFF by using peripheral ELCSoft, but the associated hardware LED will not respond to it.

Function Group Output specified pulses or seek Z phase signal when zero point is achieved.

Number M1308, D1312

Contents:

When zero point is achieved, ELC can output specified pulses or seek Z phase signal by this function. Input terminals X2, X3 are the Z-phase signal input point of CH1, CH2. When M1308= ON, D1312 is the setting register to specify the additional pulses within the range -30,000~30,000. Specified value exceeds the range will be changed as the max/min value automatically. When D1312 is set to 0, the additional pulses output function will be disabled.

Functions of other input terminals:

X4 → CH1 DOG signal input
X5 → CH1 LSN signal input
X6 → CH2 DOG signal input
X7 → CH2 LSN signal input

Function Group Right-Side Special Expansion Module ID

Number D1320 ~ D1327

Contents:

1. The ID of right-side special extension module, if any, connected to ELC-PV, ELC2-PV are stored in D1320 ~ D1327 in sequence.
2. ID of each AIO module for ELC and ELC2 series:

Module Name	Module ID (hex)	Module Name	Module ID (hex)
ELC-AN04ANNN	H'0088	ELC-AN06AANN	H'00CC
ELC-AN02NANN	H'0049	ELC-PT04ANNN	H'008A
ELC-AN04NANN	H'0089	ELC-TC04ANNN	H'008B

3. The ID of special expansion module, if any, connected to ELCM-PH/PA are stored in D1320 ~ D1327 in sequence.
4. ID of each AIO module for ELCM-PH/PA:

Module Name	Module ID (hex)	Module Name	Module ID (hex)
ELCM-AN04ANNN	H'0080	ELCM-AN06AANN	H'00C4
ELCM-AN02NANN	H'0041	ELCM-PT04ANNN	H'0082
ELCM-AN04NANN	H'0081	ELCM-TC04ANNN	H'0083

Function Group Left-Side High-Speed Special Expansion Module ID

Number D1386 ~ D1393

Contents:

1. The ID of left-side special extension module, if any, connected to ELC-PV/ELC2-PV/PA/PH/PE are stored in D1386 ~ D1393 in sequence.
2. Left-side special expansion module ID for ELC and ELC2 series.

Name	ID (HEX)
ELC-CODNETM	H'4131
ELC-COENETM	H'4050

Name	ID (HEX)
ELC-COCANOM	H'4133

Function Group Mapping function for right-side high-speed special modules

Number M1183, D9900~ D9979, D4900~D4979

Contents:
The default value of M1183 in ELCM-PH/ PA is Off. When M1183 is Off, the mapping function is enabled.
The default value of M1183 in ELC2-PB/PH/PA/PE/PV is On. When M1183 is On, the mapping function is disabled.

Model name	ELC2-PV/ELCM-PH/ELCM-PA/ ELC2-PC/ELC2-PA/ELC2-PE	ELC2-PB
Mapping range	D9900~D9979	D4900~D4979

Example:
If the modules connected to ELCM-PH from left to right are ELCM-AN04NANN and ELCM-AN04ANNN, and M1183 is Off, D9900~D9903 will be assigned to the first ELCM-AN04NANN, and D9910~D9913 will be assigned to the second ELCM-AN04NANN.

Model name	ELCM-PH	ELCM-AN04NANN	ELCM-AN04ANNN
Channel 1 (Ch1)		D9900	D9910
Channel 2 (Ch2)		D9901	D9911
Channel 3 (Ch3)		D9902	D9912
Channel 4 (Ch4)		D9903	D9913

If the modules connected to ELC2-PB from left to right are ELCM-AN04NANN and ELCM-AN04ANNN, and M1183 is Off, D4900~D4903 will be assigned to the first ELCM-AN04NANN, and D4910~D4913 will be assigned to the second ELCM-AN04NANN.

Model name	ELC2-PB	ELCM-AN04NANN	ELCM-AN04ANNN
Channel 1 (Ch1)		D4900	D4910
Channel 2 (Ch2)		D4901	D4911
Channel 3 (Ch3)		D4902	D4912
Channel 4 (Ch4)		D4903	D4913

Function Group Output clear signals when ZRN is completed

Number M1346

Contents:
When M1346 = ON, ELC will output clear signals when ZRN is completed. The clear signals to Y0, Y1 will be sent by Y4 for 20ms, and the clear signals to Y2, Y3 will be sent by Y5 for 20ms.

Function Group ELC Link

Number M1350-M1356, M1360-M1519, D1355-D1370, D1399, D1415-D1465, D1480-D1991

Contents:

Explanation of Special D and special M explanation of ELC LINK ID1–ID8 for ELC-PA/PV, ELC2- PB/PA/PH/PE/PV, ELCM-PH/PA:

MASTER ELC															
SLAVE ID 1		SLAVE ID 2		SLAVE ID 3		SLAVE ID 4		SLAVE ID 5		SLAVE ID 6		SLAVE ID 7		SLAVE ID 8	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D1480 | D1495	D1496 | D1511	D1512 | D1527	D1528 | D1543	D1544 | D1559	D1560 | D1575	D1576 | D1591	D1592 | D1607	D1608 | D1623	D1624 | D1639	D1640 | D1655	D1656 | D1671	D1672 | D1687	D1688 | D1703	D1704 | D1719	D1720 | D1735
Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num
D1434	D1450	D1435	D1451	D1436	D1452	D1437	D1453	D1438	D1454	D1439	D1455	D1440	D1456	D1441	D1457
Slave Device Internal Address to Read/Write															
D1355	D1415	D1356	D1416	D1357	D1417	D1358	D1418	D1359	D1419	D1360	D1420	D1361	D1421	D1362	D1422
M1355 = ON, Slave status is user-defined. Set the linking status of Slave manually by M1360~M1375. M1355 = OFF, Slave status is auto-detected. Linking status of Slave can be monitored by M1360~M1375															
M1360		M1361		M1362		M1363		M1364		M1365		M1366		M1367	
Action indication flag for master ELC do to slave ELC															
M1376		M1377		M1378		M1379		M1380		M1381		M1382		M1383	
Read/write error flag															
M1392		M1393		M1394		M1395		M1396		M1397		M1398		M1399	
Read completed flag (Whenever finishing a ELC read/write, this flag will be set to OFF automatically)															
M1408		M1409		M1410		M1411		M1412		M1413		M1414		M1415	
Write completed flag (whenever finishing a ELC read/write, this flag will be OFF automatically)															
M1424		M1425		M1426		M1427		M1428		M1429		M1430		M1431	
↓		↓		↓		↓		↓		↓		↓		↓	
SLAVE ID 1		SLAVE ID 2		SLAVE ID 3		SLAVE ID 4		SLAVE ID 5		SLAVE ID 6		SLAVE ID 7		SLAVE ID 8	
Read out	Write in	Read out	Write in	Read out	Write in	Readout	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215

Factory setting of starting communication address to read is H1064 (D100).
Factory setting of starting communication address to write is H10C8 (D200).

Explanation of Special D and special M explanation of ELC LINK ID9–ID16 for ELC-PA/PV, ELC2-PB/PA/PH/PE/PV, ELCM-PH/PA:

MASTER ELC															
SLAVE ID 9		SLAVE ID 10		SLAVE ID 11		SLAVE ID 12		SLAVE ID 13		SLAVE ID 14		SLAVE ID 15		SLAVE ID 16	
Read out	Write in	Read out	Write in	Read out	Write in	Readout	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D1736 | D1751	D1752 | D1767	D1768 | D1783	D1784 | D1799	D1800 | D1815	D1816 | D1831	D1832 | D1847	D1848 | D1863	D1864 | D1879	D1880 | D1895	D1896 | D1911	D1912 | D1927	D1928 | D1943	D1944 | D1959	D1960 | D1975	D1976 | D1991
Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num
D1442	D1458	D1443	D1459	D1444	D1460	D1445	D1461	D1446	D1462	D1447	D1463	D1448	D1464	D1449	D1465
Slave Device Internal Address to Read/Write															
D1363	D1423	D1364	D1424	D1365	D1425	D1366	D1426	D1367	D1427	D1368	D1428	D1369	D1429	D1370	D1430
M1355 = ON, Slave status is user-defined. Set the linking status of Slave manually by M1360~M1375. M1355 = OFF, Slave status is auto-detected. Linking status of Slave can be monitored by M1360~M1375															
M1368		M1369		M1370		M1371		M1372		M1373		M1374		M1375	
Action indication flag for master ELC do to slave ELC															
M1384		M1385		M1386		M1387		M1388		M1389		M1390		M1391	
Read/write error flag															
M1400		M1401		M1402		M1403		M1404		M1405		M1406		M1407	
Read completed flag (Whenever finishing a ELC read/write, this flag will be set to OFF automatically)															
M1416		M1417		M1418		M1419		M1420		M1421		M1422		M1423	
Write completed flag (whenever finishing a ELC read/write, this flag will be set to OFF automatically)															
M1432		M1433		M1434		M1435		M1436		M1437		M1438		M1439	

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

SLAVE ID 9		SLAVE ID 10		SLAVE ID 11		SLAVE ID 12		SLAVE ID 13		SLAVE ID 14		SLAVE ID 15		SLAVE ID 16	
Read out	Write in	Read out	Write in	Read out	Write in	Readout	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215

Factory setting of Communication address for reading is H1064 (D100).
Factory setting of Communication address for writing is H10C8 (D200).

Special D and special M for ID1 ~ ID8 of the 32 stations in ELC LINK (M1353 = ON) for ELC-PV, ELC2-PV:

MASTER ELC															
SLAVE ID 1		SLAVE ID 2		SLAVE ID 3		SLAVE ID 4		SLAVE ID 5		SLAVE ID 6		SLAVE ID 7		SLAVE ID 8	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
M1353 = ON: Enable 32 stations in the Link and the function of reading/writing more than 16 data (SET M1353); the No. of D registers for storing the read/written data.															
D1480	D1496	D1481	D1497	D1482	D1498	D1483	D1499	D1484	D1500	D1485	D1501	D1486	D1502	D1487	D1503
Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num
D1434	D1450	D1435	D1451	D1436	D1452	D1437	D1453	D1438	D1454	D1439	D1455	D1440	D1456	D1441	D1457
Slave Device Internal Address to Read/Write															
D1355	D1415	D1356	D1416	D1357	D1417	D1358	D1418	D1359	D1419	D1360	D1420	D1361	D1421	D1362	D1422
M1355 = ON, Slave status is user-defined. Set the linking status of Slave manually by M1360~M1375. M1355 = OFF, Slave status is auto-detected. Linking status of Slave can be monitored by M1360~M1375															
M1360		M1361		M1362		M1363		M1364		M1365		M1366		M1367	
Action indication flag for master ELC do to slave ELC															
M1376		M1377		M1378		M1379		M1380		M1381		M1382		M1383	
Read/write error flag															
M1392		M1393		M1394		M1395		M1396		M1397		M1398		M1399	
Read completed flag (Whenever finishing a ELC read/write, this flag will be set to OFF automatically)															
M1408		M1409		M1410		M1411		M1412		M1413		M1414		M1415	
Write completed flag (whenever finishing a ELC read/write, this flag will be set to OFF automatically)															
M1424		M1425		M1426		M1427		M1428		M1429		M1430		M1431	

SLAVE ID 1		SLAVE ID 2		SLAVE ID 3		SLAVE ID 4		SLAVE ID 5		SLAVE ID 6		SLAVE ID 7		SLAVE ID 8	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215

Default start communication address D1355 ~ D1362 to be read = H1064 (D100)
Default start communication address D1415 ~ D1422 to be written = H10C8 (D200)

Special D and special M for ID9 ~ ID16 of the 32 stations in ELC LINK (M1353 = ON) for ELC-PV, ELC2-PV:

MASTER ELC															
SLAVE ID 9		SLAVE ID 10		SLAVE ID 11		SLAVE ID 12		SLAVE ID 13		SLAVE ID 14		SLAVE ID 15		SLAVE ID 16	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read Out	Write in	Read out	Write in	Read out	Write in
M1353 = ON: Enable 32 stations in the Link and the function of reading/writing more than 16 data (SET M1353); the No. of D registers for storing the read/written data.															
D1488	D1504	D1489	D1505	D1490	D1506	D1491	D1507	D1492	D1508	D1493	D1509	D1494	D1510	D1495	D1511
Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num
D1442	D1458	D1443	D1459	D1444	D1460	D1445	D1461	D1446	D1462	D1447	D1463	D1448	D1464	D1449	D1465
Slave Device Internal Address to Read/Write															
D1363	D1423	D1364	D1424	D1365	D1425	D1366	D1426	D1367	D1427	D1368	D1428	D1369	D1429	D1370	D1430
M1355 = ON, Slave status is user-defined. Set the linking status of Slave manually by M1360~M1375. M1355 = OFF, Slave status is auto-detected. Linking status of Slave can be monitored by M1360~M1375															
M1368		M1369		M1370		M1371		M1372		M1373		M1374		M1375	
Action indication flag for master ELC do to slave ELC															
M1384		M1385		M1386		M1387		M1388		M1389		M1390		M1391	
Read/write error flag															
M1400		M1401		M1402		M1403		M1404		M1405		M1406		M1407	
Read completed flag (Whenever finishing a ELC read/write, this flag will be set to OFF automatically)															
M1416		M1417		M1418		M1419		M1420		M1421		M1422		M1423	
Write completed flag (whenever finishing a ELC read/write, this flag will be set to OFF automatically)															
M1432		M1433		M1434		M1435		M1436		M1437		M1438		M1439	

SLAVE ID 9		SLAVE ID 10		SLAVE ID 11		SLAVE ID 12		SLAVE ID 13		SLAVE ID 14		SLAVE ID 15		SLAVE ID 16	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215

Default start communication address D1363 ~ D1370 to be read = H1064 (D100)
Default start communication address D1423 ~ D1430 to be written = H10C8 (D200)

Special D and special M for ID17 ~ ID24 of the 32 stations in ELC LINK (M1353 = ON) for ELC-PV, ELC2-PV:

MASTER ELC															
SLAVE ID 17		SLAVE ID 18		SLAVE ID 19		SLAVE ID 20		SLAVE ID 21		SLAVE ID 22		SLAVE ID 23		SLAVE ID 24	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
M1353 = ON: Enable 32 stations in the Link and the function of reading/writing more than 16 data (SET M1353); the No. of D registers for storing the read/written data.															
D1576	D1592	D1577	D1593	D1578	D1594	D1579	D1595	D1580	D1596	D1581	D1597	D1582	D1598	D1583	D1599
Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num
D1544	D1560	D1545	D1561	D1546	D1562	D1547	D1563	D1548	D1564	D1549	D1565	D1550	D1566	D1551	D1567
Slave Device Internal Address to Read/Write															
D1512	D1528	D1513	D1529	D1514	D1530	D1515	D1531	D1516	D1532	D1517	D1533	D1518	D1534	D1519	D1535
M1355 = ON, Slave status is user-defined. Set the linking status of Slave manually by M1360~M1375. M1355 = OFF, Slave status is auto-detected. Linking status of Slave can be monitored by M1360~M1375															
M1440		M1441		M1442		M1443		M1444		M1445		M1446		M1447	
Action indication flag for master ELC do to slave ELC															
M1456		M1457		M1458		M1459		M1460		M1461		M1462		M1463	
Read/write error flag															
M1472		M1473		M1474		M1475		M1476		M1477		M1478		M1479	
Read completed flag (Whenever finishing a ELC read/write, this flag will be set to OFF automatically)															
M1488		M1489		M1490		M1491		M1492		M1493		M1494		M1495	
Write completed flag (whenever finishing a ELC read/write, this flag will be set to OFF automatically)															
M1504		M1505		M1506		M1507		M1508		M1509		M1510		M1511	

SLAVE ID 17		SLAVE ID 18		SLAVE ID 29		SLAVE ID 20		SLAVE ID 21		SLAVE ID 22		SLAVE ID 23		SLAVE ID 24	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215

Default start communication address D1512 ~ D1519 to be read = H1064 (D100)
Default start communication address D1528 ~ D1535 to be written = H10C8 (D200)

Special D and special M for ID25 ~ ID32 of the 32 stations in ELC LINK (M1353 = ON) for ELC-PV, ELC2-PV:

MASTER ELC															
SLAVE ID 25		SLAVE ID 26		SLAVE ID 27		SLAVE ID 28		SLAVE ID 29		SLAVE ID 30		SLAVE ID 31		SLAVE ID 32	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
M1353 = ON: Enable 32 stations in the Link and the function of reading/writing more than 16 data (SET M1353); the No. of D registers for storing the read/written data.															
D1584	D1600	D1585	D1601	D1586	D1602	D1587	D1603	D1588	D1604	D1589	D1605	D1590	D1606	D1591	D1607
Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num	Item num
D1552	D1568	D1553	D1569	D1554	D1570	D1555	D1571	D1556	D1572	D1557	D1573	D1558	D1574	D1559	D1575
Slave Device Internal Address to Read/Write															
D1520	D1536	D1521	D1537	D1522	D1538	D1523	D1539	D1524	D1540	D1525	D1541	D1526	D1542	D1527	D1543
M1355 = ON, Slave status is user-defined. Set the linking status of Slave manually by M1360~M1375. M1355 = OFF, Slave status is auto-detected. Linking status of Slave can be monitored by M1360~M1375															
M1448		M1449		M1450		M1451		M1452		M1453		M1454		M1455	
Action indication flag for master ELC do to slave ELC															
M1464		M1465		M1466		M1467		M1468		M1469		M1470		M1471	
Read/write error flag															
M1480		M1481		M1482		M1483		M1484		M1485		M1486		M1487	
Read completed flag (Whenever finishing a ELC read/write, this flag will be set to OFF automatically)															
M1496		M1497		M1498		M1499		M1500		M1501		M1502		M1503	
Write completed flag (whenever finishing a ELC read/write, this flag will be set to OFF automatically)															
M1512		M1513		M1514		M1515		M1516		M1517		M1518		M1519	

SLAVE ID 25		SLAVE ID 26		SLAVE ID 27		SLAVE ID 28		SLAVE ID 29		SLAVE ID 30		SLAVE ID 31		SLAVE ID 32	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215

Default start communication address D1520 ~ D1527 to be read = H1064 (D100)
Default start communication address D1536 ~ D1543 to be written = H10C8 (D200)

Explanation:

1. The basic communication protocol for ELC LINK is MODBUS
2. ELC-PV/ELC2-PV supports 32 stations in the LINK and reading/writing of more than 16 data (SET1353) (M1353 = On). ELC-PA, ELCM-PH/PA, ELC2-PB/PA/PH/PE supports 16 devices in the LINK and reading/writing of more than 16 data.
3. ELC-PV/ELC2-PV: When a MASTER ELC and a Slave ELC is connected, they are able to read/write maximum 100 WORD data (M1353 = On). ELC2-PB/PA/PH/PE: When a MASTER ELC and a Slave ELC is connected, they are able to read/write maximum 50 WORD data (M1353 = On). ELC-PA: Does not support M1353. When a Master ELC and a Slave ELC is connected, they are able to read/write maximum 16 WORD data.
4. When the Master ELC is connected through COM2 (RS-485), baud rates and communication formats of all Slave ELCs must be the same (set in D1120). When ELC-PA/PV, ELC2-PV as Master, it supports ASCII and RTU format.
5. When Slave ELC is connected through COM2 (RS-485), baud rate and communication format of all Slaves must be the same (set in D1120). ELC supports both ASCII and RTU mode. When ELC-PA/PV, ELC2 series as Slave, it supports ASCII and RTU format.
6. When Slave ELC is connected through COM3 (RS-485), baud rate and communication format of all Slaves must be the same (set in D1109). When ELC-PA/ PV, ELC2-PV as Slave, it only supports ASCII format (Max. baud rate = 38,400bps).
7. For one-to-many LINK: Connected through RS-485. ELC COM2, COM3 support many communication formats.
8. When M1356 = OFF(Default), the station number of the starting Slave (ID1) can be designated by D1399 of Master ELC through ELC LINK, and ELC will automatically assign ID2~ID16 with consecutive station numbers according to the station number of ID1. For example, if D1399 = K3, Master ELC will send out communication commands to ID1~ID16 which carry station number K3~K18. In addition, care should be taken when setting the station number of Slaves. All station numbers of slaves should not be the same as the station number of the Master ELC, which is set up in D1121/D1255.
9. When both M1353 and M1356 are ON, the station number of ID1~ID16 can be specified by the user in D1900~D1915 of Master ELC. For example, when D1900~D1903 = K3, K3, K5, K5, Master ELC will access the Slave with station number K3 for 2 times, then the slave with station number K5 for 2 times as well. Note that all station numbers of slaves should not be the same as the station number of the Master ELC, and M1353 must be set ON for this function.
10. The ID number of the starting slave can be designated by D1399 and should be limited to the range K1~K214. Slave ID cannot be repeated or the same as Master ID (set in D1121/D1255)
11. Station number selection function (M1356 = ON) is supported by ELCM-PH/PA and ELC2-PB/PA/PH/PE/PV.

Operation:

1. Set up the baud rates and communication formats of Master ELC and all connected Slave ELCs and make them the same. COM2_RS-485: D1120; COM3_RS-485: D1109.
2. Set up Master ELC ID by D1121 and the starting slave ID by D1399. Then, set slave ID of each slave ELC. The ID of master ELC and slave ELC cannot be the same.
3. Set up the number of connected Slave stations and the number of data to be read in/written to Slave stations. For ELC-PV/ELC2-PV/PB/PA/PH/PE ELCM-PH/PA (M1353 = On): Enable the function of the 32(or 16) conncected Slaves and reading/writing of more than 16 data (Max. 100 data). Next, set up the No. of D registers for storing the read data (D1480 ~ D1495, D1576 ~ D1591) and written data (D1496 ~ D1511, D1592 ~ D1607) (See the explanations above on special D). ELC-PA, only supports reading/writing of 16 data. (If data length is not specified, ELC will take default setting or the previous value as the set value. For details of data length registers, please refer to the tables above)
4. Set device communication address to read/write to slave. (Refer to Special D explanation above for special D setting. Factory setting of communication address for reading is H1064 (D100) and writing is H10C8 (D200).
5. Steps to start ELC LINK:
   - Enable the function of more than 32 stations connected to ELC LINK and reading/writing of 16 data (Max. 100 data) (M1353).
   - Set ON M1354 to enable simultaneous data read/write in a polling of ELC LINK.

- When M1355 = On, M1360 ~ M1375 (M1440 ~ M1455) will be the flags for the ELC designated to be connected to. When M1355 = Off, there will be detection on the slaves connected, and M1360 ~ M1375 (M1440 ~ M1455) will become the flags for the existence of connected ELC.
- Select auto mode on ELC LINK by M1351 or manual mode by M1352. (Note that the 2 flags should not be set ON at the same time.)Then, set up the times of polling cycle by D1431.
- Start MASTER ELC LINK (M1350)

Master ELC action explanation:

1. M1353=ON, Enable the function of more than 32 stations connected to ELC LINK.
2. M1355 = ON, Slave status is user-defined. Set the linking status of Slave manually by M1360~M1375 (M1440 ~ M1455).
   - Select auto mode on ELC LINK by M1351 or manual mode by M1352. Auto mode and manual mode should not be enabled at the same time.
   - Enable ELC LINK (M1350). The linking status is specified by M1360~M1375 (M1440 ~ M1455), therefore Master ELC will access the designated Slaves according to M1352~M1375 (M1440 ~ M1455) continuously no matter how many Slave ELC are physically connected.
3. M1355 = OFF, Slave status is auto-detected. Linking status of Slave can be monitored by M1360~M1375, M1440 ~ M1455.
   - Select auto mode on ELC LINK by M1351 or manual mode by M1352. Auto mode and manual mode should not be enabled at the same time.
   - Enable ELC LINK (M1350). Master ELC will detect the connected Slaves and store the number of connected ELCs in D1433. The time for detection differs by number of connected Slaves and time-out setting in D1129.
   - M1360~M1375 (M1440 ~ M1455) indicate the linking status of Slave ID 1~16 (ID 17~31)
   - If no slave is detected, M1350 will be OFF and ELC Link will be stopped.
   - ELC will only detect the number of slaves at the first time when M1350 turns ON.
   - After auto-detection is completed, master ELC starts to access each connected slave. Once slave ELC is added after auto-detection, master ELC cannot access it unless auto-detection is conducted again.
4. Synchronous read/write function (M1354) has to be set up before enabling ELC LINK. Setting up this flag during ELC LINK execution will not take effect.
5. When M1354=ON, it will use Modbus Function H17 (synchronous read/write function) for ELC LINK communication function. If item number for writing is set to 0, the ELC will use Modbus Function H03 (read multiple WORDs) for ELC LINK communication function. In the same way, if item number for reading is set to 0, the ELC will use Modbus Function H06 (write one WORD) or Modbus Function H10 (write multiple WORDs) for ELC LINK communication function.
6. When M1353 = OFF, ELC LINK accesses the Slave with max 16 words, and the data is automatically stored in the corresponding registers. When M1353 = ON, up to 50 words are accessible and the user can specify the starting register for storing the read/written data. For example, if the register for storing the read/written data on Slave ID1 is specified as D1480 = K500, D1496 = K800, access data length D1434 = K50, D1450 = K50, registers of Master ELC D500~D549 will store the data read from Slave ID1, and the data stored in D800~D849 will be written into Slave ID1.
7. Master ELC conducts reading before writing. Both reading and writing is executed according to the range specified by user.
8. Master ELC will read/write to slave ELC in order, i.e. it will read/write to the next slave after finishing a slave.

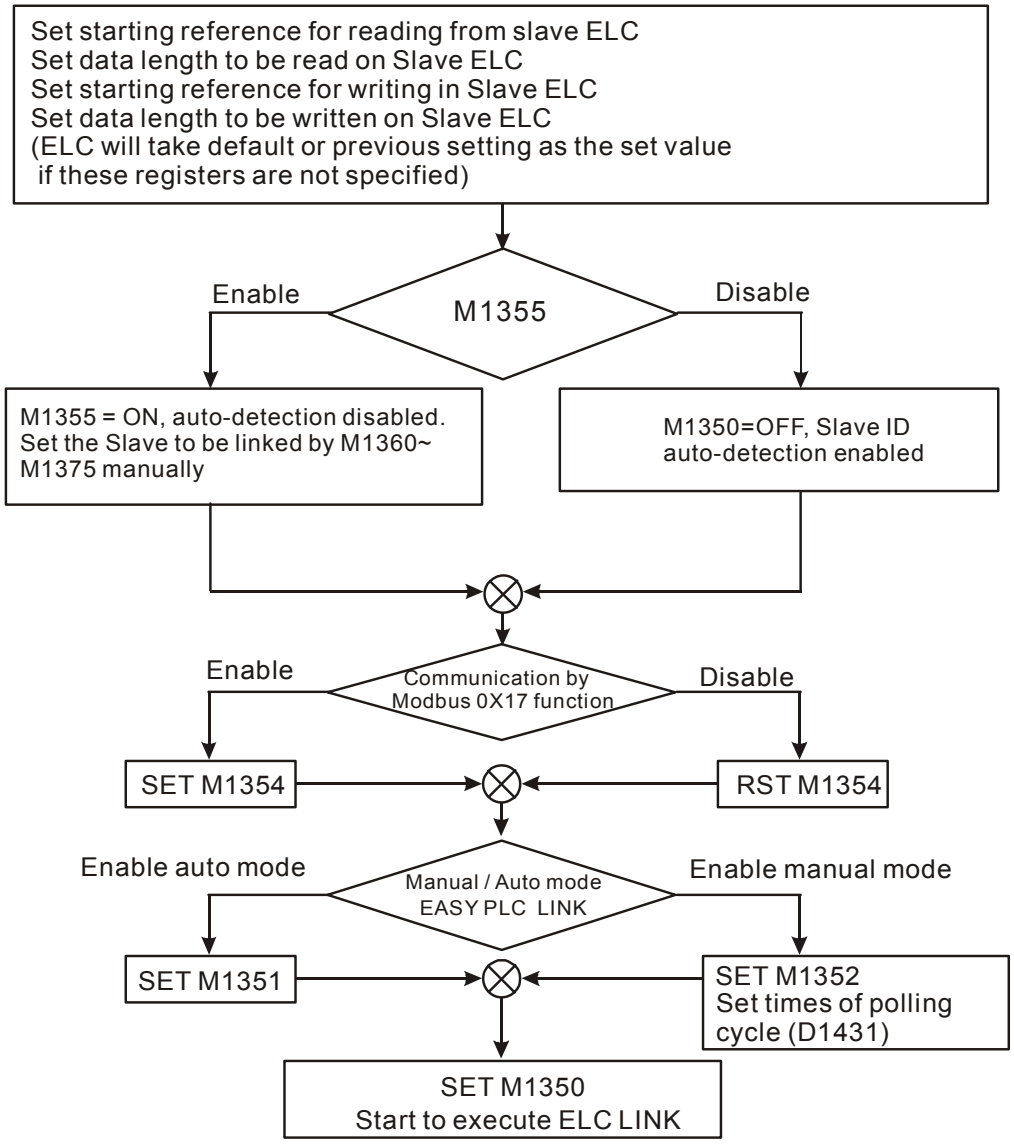
Automatic / Manual mode explanation:

1. Auto mode (M1351): when M1351 = ON, Master ELC will access slave ELCs as the operation described above, and stop the polling till M1350 or M1351 is OFF.
2. Manual mode (M1352): When manual mode is selected, times of polling cycle in D1431 has to be set up. A full polling cycle refers to the completion of accessing all Slaves. When ELC LINK is enabled, D1432 starts to store the times of polling. When D1431 = D1432, ELC LINK stops and M1352 is reset. When M1352 is set ON again, the ELC will start the polling according to times set in D1431 automatically.

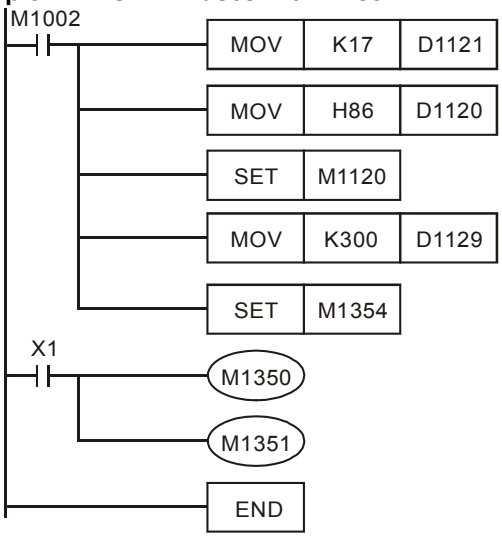
Note:

1. Auto mode M1351 and manual mode M1352 cannot be enabled at the same time. If M1351 is enabled after M1352 is ON, ELC LINK will stop and M1350 will be reset.
2. Communication timeout setting can be modified by D1129 with available range 200 ≦D1129 ≦ 3000. The ELC will take the upper / lower bound value as the set value if the specified value is out of the available range. D1129 has to be set up before M1350 = ON.
3. ELC LINK function is only valid when baud rate is higher than 1200 bps. When baud rate is less than 9600 bps, please set communication time-out to more than 1 second.
4. The communication is invalid when data length to be accessed is set to 0.
5. Access on 32-bit high speed counters (C200~C255) is not supported.
6. Available range for D1399: 1 ~ 230. The ELC will take the upper / lower bound value as the set value if the specified value exceeds the available range.
7. D1399 has to be set up before enabling ELC LINK. Setting up this register during ELC LINK execution will not take effect.
8. Advantage of using D1399 (Designating the ID of starting Slave): In old version of ELC LINK, the ELC detects slaves from ID1 to ID16. Therefore, when ELC LINK is applied in multi-layer networks, e.g. 3 layers of networks, the Slave ID of 2nd and 3rd layer will be repeated. When Slave ID is repeated, i.e. the same as Master ID, the Slave will be passed. In this case, only 15 Slaves can be connected in 3rd layer. To solve this problem, D1399 can be applied for increasing the connectable Slaves in multi-layer network structure.

Operation flow chart:

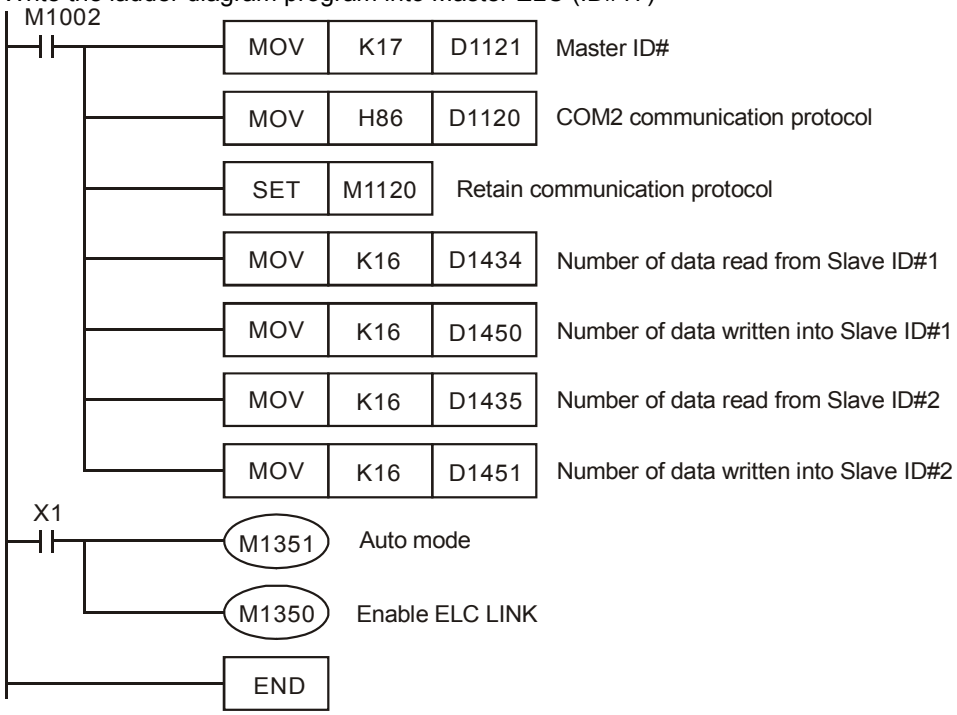


Example 1: ELC LINK uses with M1354



Example 2: Connection of 1 Master and 2 Slaves by RS-485 and exchange of 16 data between Master and Slaves through ELC LINK (M1353 = OFF, linkage of 16 stations, 16 data read/write mode)

1. Write the ladder diagram program into Master ELC (ID#17)



2. When X1 = ON, the data exchange between Master and the two Slaves will be automatically done in ELC LINK, i.e. the data in D100 ~ D115 in the two Slaves will be read into D1480 ~ D1495 and D1512 ~ D1527 of the Master, and the data in D1496 ~ D1511 and D1528 ~ D1543 will be written into D200 ~ D215 of the two Slaves.

Master ELC *1		Slave ELC*2
D1480 ~ D1495	Read ←	D100 ~ D115 of Slave ID#1
D1496 ~ D1511	Write →	D200 ~ D215 of Slave ID#1
D1512 ~ D1527	Read ←	D100 ~ D115 of Slave ID#2
D1528 ~ D1543	Write →	D200 ~ D215 of Slave ID#2

3. Assume the data in D for data exchange between Master and Slave before ELC LINK is enabled (M1350 = OFF) are as the follow:

Master ELC	Preset value	Slave ELC	Preset value
D1480 ~ D1495	K0	D100 ~ D115 of Slave ID#1	K5,000
D1496 ~ D1511	K1,000	D200 ~ D215 of Slave ID#1	K0
D1512 ~ D1527	K0	D100 ~ D115 of Slave ID#2	K6,000
D1528 ~ D1543	K2,000	D200 ~ D215 of Slave ID#2	K0

After ELC LINK is enabled (M1350 = ON), the data in D for data exchange will become:

Master ELC	Preset value	Slave ELC	Preset value
D1480 ~ D1495	K5,000	D100 ~ D115 of Slave ID#1	K5,000
D1496 ~ D1511	K1,000	D200 ~ D215 of Slave ID#1	K1,000
D1512 ~ D1527	K6,000	D100 ~ D115 of Slave ID#2	K6,000
D1528 ~ D1543	K2,000	D200 ~ D215 of Slave ID#2	K2,000

4. The Master ELC has to be ELC-PA/PV, ELC2-PB/PH/PA/PE/PV, ELCM-PH/PA controllers, and the Slave ELC can be any ELC controller.
5. There can be maximum 16 Slave ELCs in ELC LINK. See the special Ds in the Master ELC corresponding to D100 ~ D115 and D200 ~ D215 in every Slave ELC in the tables of special M and special D.
6. D1354 is ELC link scan cycle with unit is 1ms and max. display value is K32000. D1354 = K0 when ELC Link stops or when the first scan is completed.

- Internal Device Communication Address

Device	Range		Type	ELC communication address (Hex)	Modbus communication address (Dec)
S	000~255		bit	0000~00FF	000001~000256
S	246~511		bit	0100~01FF	000247~000512
S	512~767		bit	0200~02FF	000513~000768
S	768~1023		bit	0300~03FF	000769~001024
X	000~377 (Octal)		bit	0400~04FF	101025~101280
Y	000~377 (Octal)		bit	0500~05FF	001281~001536
T	000~255		bit	0600~06FF	001537~001792
			word	0600~06FF	401537~401792
M	000~255		bit	0800~08FF	002049~002304
M	256~511		bit	0900~09FF	002305~002560
M	512~767		bit	0A00~0AFF	002561~002816
M	768~1023		bit	0B00~0BFF	002817~003072
M	1024~1279		bit	0C00~0CFF	003073~003328
M	1280~1535		bit	0D00~0DFF	003329~003584
M	1536~1791		bit	B000~B0FF	045057~045312
M	1792~2047		bit	B100~B1FF	045313~045568
M	2048~2303		bit	B200~B2FF	045569~045824
M	2304~2559		bit	B300~B3FF	045825~046080
M	2560~2815		bit	B400~B4FF	046081~046336
M	2816~3071		bit	B500~B5FF	046337~046592
M	3072~3327		bit	B600~B6FF	046593~046848
M	3328~3583		bit	B700~B7FF	046849~047104
M	3584~3839		bit	B800~B8FF	047105~047360
M	3840~4095		bit	B900~B9FF	047361~047616
C	0~199	16-bit	bit	0E00~0EC7	003585~003784
			word	0E00~0EC7	403585~403784
	200~255	32-bit	bit	0EC8~0EFF	003785~003840
			Dword	0EC8~0EFF	403785~403840
D	000~256		word	1000~10FF	404097~404352
D	256~511		word	1100~11FF	404353~404608
D	512~767		word	1200~12FF	404609~404864
D	768~1023		word	1300~13FF	404865~405120
D	1024~1279		word	1400~14FF	405121~405376
D	1280~1535		word	1500~15FF	405377~405632

Device	Range	Type	ELC communication address (Hex)	Modbus communication address (Dec)
D	1536~1791	word	1600~16FF	405633~405888
D	1792~2047	word	1700~17FF	405889~406144
D	2048~2303	word	1800~18FF	406145~406400
D	2304~2559	word	1900~19FF	406401~406656
D	2560~2815	word	1A00~1AFF	406657~406912
D	2816~3071	word	1B00~1BFF	406913~407168
D	3072~3327	word	1C00~1CFF	407169~407424
D	3328~3583	word	1D00~1DFF	407425~407680
D	3584~3839	word	1E00~1EFF	407681~407936
D	3840~4095	word	1F00~1FFF	407937~408192
D	4096~4351	word	9000~90FF	436865~437120
D	4352~4607	word	9100~91FF	437121~437376
D	4608~4863	word	9200~92FF	437377~437632
D	4864~5119	word	9300~93FF	437633~437888
D	5120~5375	word	9400~94FF	437889~438144
D	5376~5631	word	9500~95FF	438145~438400
D	5632~5887	word	9600~96FF	438401~438656
D	5888~6143	word	9700~97FF	438657~438912
D	6144~6399	word	9800~98FF	438913~439168
D	6400~6655	word	9900~99FF	439169~439424
D	6656~6911	word	9A00~9AFF	439425~439680
D	6912~7167	word	9B00~9BFF	439681~439936
D	7168~7423	word	9C00~9CFF	439937~440192
D	7424~7679	word	9D00~9DFF	440193~440448
D	7680~7935	word	9E00~9EFF	440449~440704
D	7936~8191	word	9F00~9FFF	440705~440960
D	8192~8447	word	A000~A0FF	440961~441216
D	8448~8703	word	A100~A1FF	441217~441472
D	8704~8959	word	A200~A2FF	441473~441728
D	8960~9215	word	A300~A3FF	441729~441984
D	9216~9471	word	A400~A4FF	441985~442240
D	9472~9727	word	A500~A5FF	442241~442496
D	9728~9983	word	A600~A6FF	442497~442752
D	9984~10239	word	A700~A7FF	442753~443008
D	10234~10495	word	A800~A8FF	443009~443246

Device	Range	Type	ELC communication address (Hex)	Modbus communication address (Dec)
D	10496~10751	word	A900~A9FF	443247~443502
D	10752~11007	word	AA00~AAFF	443503~443758
D	11008~11263	word	AB00~ABFF	443759~444014
D	11264~11519	word	AC00~ACFF	444015~444270
D	11520~11775	word	AD00~ADFF	444271~444526
D	11776~11999	word	AE00~AEDF	444527~444750

- **Devices which are supported are listed below.**

Device / Model	X	Y	M	S	T	C	D
ELCB-PB	X0~X177	Y0~Y177	M0~M1535	S0~S127	T0~T127	C0~C127, C232~C255	D0~D599, D1000~D1311
ELC-PA	X0~X177	Y0~Y177	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D4999
ELC-PV	X0~X377	Y0~X377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999
ELCM-PH **ELCM-PA** **ELC2-PC** **ELC2-PA**	X0~X377	Y0~X377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999
ELC2-PB	X0~X377	Y0~X377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D4999
ELC2-PV **ELC2-PE**	X0~X377	Y0~X377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999

MEMO

Instruction Set

This chapter contains all of the instructions that are used with the ELC controllers as well as detailed information concerning the usage of the instructions.

This Chapter Contains

3 Instruction Set

3.1 Basic Instructions (without API numbers)

Code	Function	Execution speed (us)				STEPS
		ELCB-PB ELC-PA	ELCM-PH ELCM-PA ELC2-PB ELC2-PC ELC2-PA	ELC2-PE	ELC-PV ELC2-PV	
LD	Load contact A	3.8	0.76	0.64	0.24(0.56)	1~3
LDI	Load contact B	3.88	0.78	0.68	0.24(0.56)	1~3
AND	Series connection with A contact	2.32	0.54	0.58	0.24(0.56)	1~3
ANI	Series connection with B contact	2.4	0.56	0.62	0.24(0.56)	1~3
OR	Parallel connection with A contact	2.32	0.54	0.62	0.24(0.56)	1~3
ORI	Parallel connection with B contact	2.4	0.56	0.64	0.24(0.56)	1~3
ANB	Series connects the circuit block	1.76	0.68	0.68	0.24	1~3
ORB	Parallel connects the circuit block	1.76	0.76	0.76	0.24	1~3
MPS	Save the operation result	1.68	0.74	0.68	0.24	1~3
MRD	Read the operation result (the pointer not moving)	1.6	0.64	0.54	0.24	1
MPP	Read the result	1.6	0.64	0.54	0.24	1
OUT	Drive coil	5.04	0.88	0.68	0.24(0.56)	1~3
SET	Action latched (ON)	3.8	0.76	0.68	0.24(0.56)	1~3
RST	Clear the contacts or the registers	7.8	2.2	1.04	0.24(0.56)	3
MC	Connect the common series connection contacts	5.6	1	0.8	5.6	3
MCR	Disconnect the common series connection contacts	5.7	1	0.8	5.7	3
END	Program end	5	1	0.8	0.24	1
NOP	No function	0.88	0.4	0.5	0.16	1
STL	Step transition ladder start command	11.6	2.2	2	0.56	1
RET	Step transition ladder return command	7.04	1.6	1.4	0.24	1

Note: For ELC-PV/ELC2-PV, the execution speed in the brackets ( ) refers to the execution speed of designated operand M1536 ~ M4095.

3.2 Basic Instruction Explanations

Mnemonic	Operands	Function	Program steps
LD	X, Y, M, S, T, C	Load A contact	1

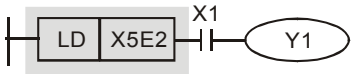
ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

Description:

The LD instruction is used on the A contact that has its start from the left BUS or the A contact that is the start of a new block of program when using the ORB and ANB instructions (see later sections). ELC2-PB/PH/PA/PE/PV V1.0 and ELCM-PH/PA V2.0 support the operands X, Y, M, S. These operands can be qualified by E or F. Users have to use ELCSoft version 2.02 (or above).

Program Example:

Instruction:		Operation:
		; Load contact A of X0
LD	**X0**	
AND	X1	; Connect to contact A of X1 in series
OUT	Y1	; Drive Y1 coil



Instruction:		Operation:
		; Load contact A of X3 (E2=K-2)
LD	**X5E2**	
AND	X1	; Connect to contact A of X1 in series
OUT	Y1	; Drive Y1 coil

Mnemonic	Operands	Function	Program steps
LDI	X, Y, M, S, T, C	Load B contact	1

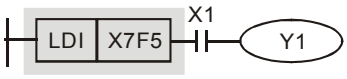
ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

Description:

The LDI instruction is used on the B contact that has its start from the left BUS or the B contact that is the start of a new block of program when using the ORB and ANB instructions (see later sections). ELC2-PB/PH/PA/PE/PV V1.0 and ELCM-PH/PA V2.0 support the operands X, Y, M, S. These operands can be qualified by E or F. Users have to use ELCSoft version 2.02 (or above).

Program Example:

Instruction:		Operation:
LDI	**X0**	; Load contact B of X0
AND	X1	; Connect to contact A of X1 in series
OUT	Y1	; Drive Y1 coil



Instruction:		Operation:
LDI	**X7F5**	;Load contact B of X10 **(F5=K3)**
AND	X1	; Connect to contact A of X1 in series
OUT	Y1	; Drive Y1 coil

Mnemonic	Operands	Function	Program steps
AND	X, Y, M, S, T, C	Series connection- A contact	1

ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

3

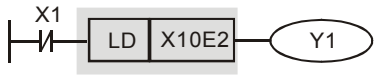
Description:

The AND command is used in the series connection of an A contact.

ELC2-PB/PH/PA/PE/PV V1.0 and ELCM-PH/PA V2.0 support the operands X, Y, M, S. These operands can be qualified by E or F. Users have to use ELCSoft version 2.02 (or above).

Program Example:

Instruction:		Operation:
LDI	X1	; Load contact B of X1
AND	**X0**	; Connect to contact A of X0 in series
OUT	Y1	; Drive Y1 coil



Instruction:		Operation:
LDI	X1	;Load contact B of X1
AND	**X10E2**	;Connect to contact A of X20(E2 = K8) in series
OUT	Y1	;Drive Y1 coil

Mnemonic	Operands	Function	Program steps
ANI	X, Y, M, S, T, C	Series connection- B contact	1

ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

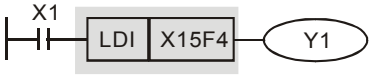
Descriptions:

The ANI command is used in the series connection of a B contact.

ELC2-PB/PH/PA/PE/PV V1.0 and ELCM-PH/PA V2.0 support the operands X, Y, M, S. These operands can be qualified by E or F. Users have to use ELCSoft version 2.02 (or above).

Program Example:

Instruction:		Operation:
LD	X1	; Load contact A of X1
ANI	**X0**	; Connect to contact B of X0 in series
OUT	Y1	; Drive Y1 coil



Instruction:		Operation:
LD	X1	; Load contact A of X1
ANI	**X15F4**	; Connect to contact B of X11(F4=K-4) in series
OUT	Y1	; Drive Y1 coil

Mnemonic	Operands	Function	Program steps
OR	X, Y, M, S, T, C	Parallel connection- A contact	1

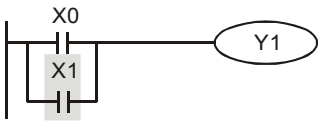
ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

Description:

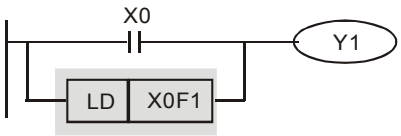
The OR command is used in the parallel connection of an A contact.

ELC2-PB/PH/PA/PE/PV V1.0 and ELCM-PH/PA V2.0 support the operands X, Y, M, S. These operands can be qualified by E or F. Users have to use ELCSoft version 2.02 (or above).

Program Example:



Instruction:		Operation:
LD	X0	; Load contact A of X0
OR	**X1**	; Connect to contact A of X1 in parallel
OUT	Y1	; Drive Y1 coil



Instruction:		Operation:
LD	X0	; Load contact A of X0
OR	**X0F1**	; Connect to contact A of X5(F1=K5) in parallel
OUT	Y1	; Drive Y1 coil

Mnemonic	Operands	Function	Program steps
ORI	X, Y, M, S, T, C	Parallel connection- B contact	1

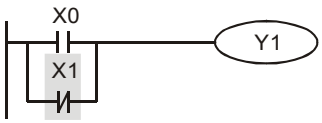
ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

Description:

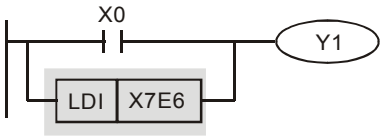
The ORI command is used in the parallel connection of a B contact.

ELC2-PB/PH/PA/PE/PV V1.0 and ELCM-PH/PA V2.0 support the operands X, Y, M, S. These operands can be qualified by E or F. Users have to use ELCSoft version 2.02 (or above).

Program Example:



Instruction:		Operation:
LD	X0	; Load contact A of X0
ORI	**X1**	; Connect to contact B of X1 in parallel
OUT	Y1	; Drive Y1 coil



Instruction:		Operation:
LD	X0	; Load contact A of X0
ORI	**X7E6**	; Connect to contact B of X4(E6=K-3) in parallel
OUT	Y1	Drive Y1 coil

Mnemonic	Operands	Function	Program steps
ANB	X, Y, M, S, T, C	Series connection (Multiple Circuits)	1

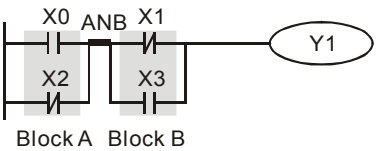
ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

Description:

The “AND” operation between the previous logic and the contents of the accumulative register.

ELC2-PB/PH/PA/PE/PV V1.0 and ELCM-PH/PA V2.0 support the operands X, Y, M, S. These operands can be qualified by E or F. Users have to use ELCSoft version 2.02 (or above).

Program Example:



Instruction:		Operation:
LD	X0	; Load contact A of X0
ORI	X2	; Connect to contact B of X2 in parallel
LDI	X1	; Load contact B of X1
OR	X3	; Connect to contact A of X3 in parallel
ANB		; Connect circuit block in series
OUT	Y1	; Drive Y1 coil

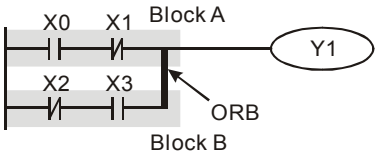
Mnemonic	Operands	Function	Program steps
ORB	X, Y, M, S, T, C	Parallel connection (Multiple circuits)	1

ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

Description:

The “OR” operation between the previous logic and the contents of the accumulative register.

Program Example:



Instruction:		Operation:
LD	X0	; Load contact A of X0
ANI	X1	; Connect to contact B of X1 in series
LDI	X2	; Load contact B of X2
AND	X3	; Connect to contact A of X3 in series
ORB		; Connect circuit block in parallel
OUT	Y1	; Drive Y1 coil

Mnemonic	Operands	Function	Program steps
MPS	X, Y, M, S, T, C	Store the current result of the internal ELC operations	1

ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

Description:

MPS stores the connection point of the ladder circuit so that further coil branches can recall the value later.

Mnemonic	Operands	Function	Program steps
MRD	X, Y, M, S, T, C	Reads the current result of the internal ELC operations	1

ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

Description:

MRD recalls or reads the previously stored connection point data and forces the next contact to connect to it.

Mnemonic	Operands	Function	Program steps
MPP	X, Y, M, S, T, C	Pops (recalls and removes) the currently stored result	1

ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

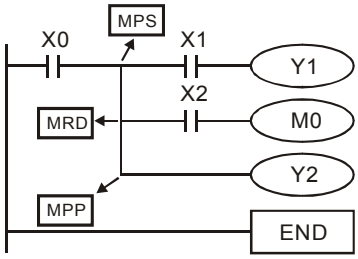
Description:

MPP pops (recalls and removes) the stored connection point. First, it connects the next contact, then it removes the point from the temporary storage area.

Basic points to remember:

7. Use these instructions to connect output coils to the left hand side of a contact. Without these instructions connections can only be made to the right hand side of the last contact.
8. For every MPS instruction there MUST be a corresponding MPP instruction.
9. The last contact or coil circuit must connect to an MPP instruction.
10. At any programming step, the number of active MPS-MPP pairs must be no greater than 8.

Program Example:




```
Instruction:      Operation:
LD      X0        ; Load contact A of X0
MPS               ; Save to stack
AND     X1        ; Connect to contact A of X1 in series
OUT     Y1        ; Drive Y1 coil
MRD               ; Read from stack
AND     X2        ; Connect to contact A of X2 in series
OUT     M0        ; Drive M0 coil
MPP               ; Read from stack and pop pointer
OUT     Y2        ; Drive Y2 coil
END               ; Program end
```

MPS, MRD and MPP usage:

1. When writing a program in ladder format, programming tools automatically add all MPS, MRD and MPP instructions at the program conversion stage. If the generated instruction program is viewed, the MPS, MRD and MPP instructions are present.
2. When writing a program in instruction format, it is entirely up to the user to enter all relevant MPS, MRD and MPP instructions as required.

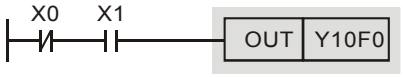
Mnemonic	Operands	Function	Program steps
OUT	X, Y, M, S, T, C	Output coil	1

ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

Use the OUT Instruction to turn a bit on when the conditions preceding it are evaluated as true. ELC2-PB/PH/PA/PE/PV V1.0 and ELCM-PH/PA V2.0 support the operands X, Y, M, S. These operands can be qualified by E or F. Users have to use ELCSoft version 2.02 (or above).

Program Example:

Instruction:		Operation:
LDI	X0	; Load contact B of X0
AND	X1	; Connect to contact A of X1 in series
OUT	**Y1**	; Drive Y1 coil



Instruction:		Operation:
LDI	X0	; Load contact B of X0
AND	X1	; Connect to contact A of X1 in series
OUT	**Y10F0**	; Drive Y5 (F0=K-3) coil

Mnemonic	Operands	Function	Program steps
SET	X, Y, M, S, T, C	Latch（ON）	1

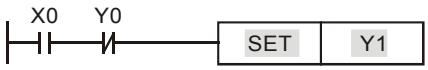
ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

Description:

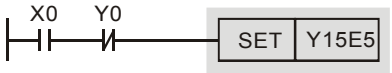
When the SET command is energized, the addressed bit is turned on. This instruction can only turn on a bit. To turn a bit off, use the RST command.

ELC2-PB/PH/PA/PE/PV V1.0 and ELCM-PH/PA V2.0 support the operands X, Y, M, S. These operands can be qualified by E or F. Users have to use ELCSoft version 2.02 (or above).

Program Example:



Instruction:		Operation:
LD	X0	; Load contact A of X0
ANI	Y0	; Connect to contact B of Y0 in series
SET	**Y1**	;Deive Y1 latch (ON)



Instruction:		Operation:
LD	X0	; Load contact A of X0
ANI	Y0	; Connect to contact B of Y0 in series
SET	**Y15E5**	;Deive **Y20(E5=K3)** latch (ON)

Mnemonic	Operands	Function	Program steps
RST	Y, M, S, T, C, D, E, F	Clear the contact or the register	1

ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

Description:

When the RST command is energized, the action taken depends on the data type as follows:

Device	Status
S, Y, M	Coil and contact will be set to “OFF”.
T, C	Present values and contacts of the timer or counter will be cleared.
D, E, F	The content value will be set to 0.

ELC2-PB/PH/PA/PE/PV V1.0 and ELCM-PH/PA V2.0 support the operands Y, M, S. These operands can be qualified by E or F. Users have to use ELCSoft version 2.02 (or above).

Program Example:

Instruction: Operation:

LD X0 ; Load contact A of X0

RST Y5 ; Clear contact Y5

Instruction: Operation:

LD X0 ; Load contact A of X0

RST Y5E0 ; Clear contact **Y5(E0=K0)**

Mnemonic	Operands	Function	Program steps
MC/MCR	N0~N7	Master control Start/Reset	1

ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

Description:

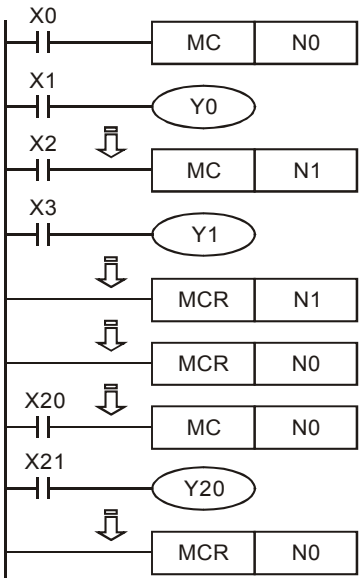
MC is the main-control start command. When the MC command is evaluated as true, the execution of the instructions between MC and MCR will not be interrupted. When MC is false, the instructions between MC and MCR are described as follows:

General Timer	The timer value is set back to zero, the coil and the contact are both turned OFF
Timers for Subroutines and Interrupts	The timer value is set back to zero, the coil and the contact are both turned OFF
Accumulative timer	The timer value and the contact stay at their present condition
Counter	The counting value and the contact stay at their present condition
OUT	All turned OFF
SET/RST	Stay at present condition
Application instructions	Remain unchanged

11. MCR is the main-control command that is placed at the end of the main-control program.

12. The MC and MCR instructions support nesting up to 8 levels. When using MC/MCR instructions you must use the numbers associated with them in numerical order beginning with N0 as shown in the example below. Valid numbers for MC/MCR pairs are N0-N7.

Program Example:




```
Instruction:      Operation:
LD    X0     ; The control loop N0 is active when X0
MC    N0       is ON
LD    X1     ; Load A contact of X1
OUT   Y0     ; Energize Y0
:
LD    X2     ; The control loop N1is active when X2
MC    N1       is ON
LD    X3     ; Load A contact of X3
OUT   Y1     ; Energize Y1
:
MCR   N1     ; The control loop N1 terminates
:
MCR   N0     ; The control loop N0 terminates
:
LD    X20    ; The control loop N0 is active when X20
MC    N0       is ON
LD    X21    ; Load A contact of X21
OUT   Y20    ; Energize Y20
:
MCR   N0     ; The control loop N0 terminates
```

Mnemonic	Function	Program steps
END	Program End	1

ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

Description:

An END statement must be placed at the end of an ELC program. An ELC controller will scan from program line 0 to the END statement, then return to line 0 again. .

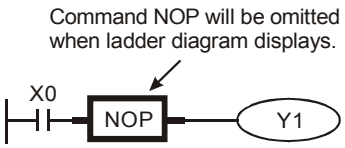
Mnemonic	Function	Program steps
NOP	No operation	1

ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

Description:

This is a no-operation command and has no effect on the operation of the line of code it's in. NOP is used to delete a command without changing the number of steps. (Overwrite with NOP)

Program Example:



Instruction:		Operation:
LD	X0	; Load A contact of X0
NOP		; No operation
OUT	Y1	; Drive Y1 coil

Mnemonic	Function	Program steps
NP	Negative contact to Positive contact	1

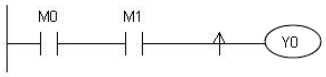
ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

Description:

When the conditions preceding NP command change from false to true, NP command (works as contact A) will be ON for a scan cycle. In the next scan cycle it turns OFF.

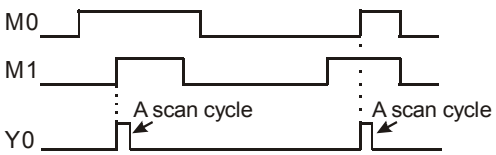
ELC-PV only support V1.6(above) version.

Program Example:



Instruction:		Operation:
LD	M0	; Load A contact of M0
AND	M1	; And A contact of M1
NP		; Negative contact to Positive contact
OUT	Y0	; Drive Y0 coil

Timing Diagram:



Mnemonic	Function	Program steps
PN	Positive contact to Negative contact	1

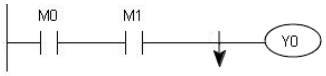
ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

Description:

When the conditions preceding PN command change from true to false, PN command (works as contact A) will be ON for a scan cycle. In the next scan cycle it turns OFF.

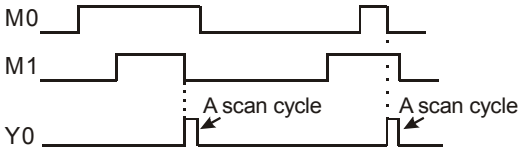
ELC-PV only support ECL-PVV1.6(above) .

Program Example:



Instruction:		Operation:
LD	M0	; Load A contact of M0
AND	M1	; And A contact of M1
PN		; Positive contact to Negative contact
OUT	Y0	; Drive Y0 coil

Timing Diagram:



3.3 Pointers

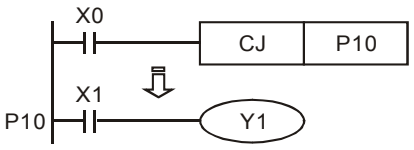
Mnemonic	Function	Program steps
P	Pointer (P0~P255)	1

ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

Description:

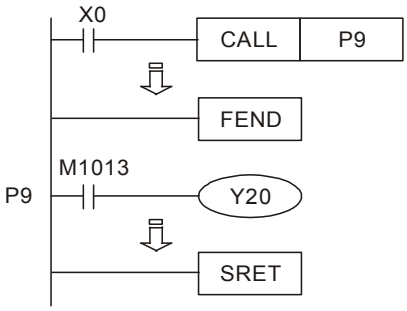
Pointers are used with the jump instructions (CJ, CALL) to jump to another area of logic or to call a subroutine. Pointer numbers can only be used once.

Program Example 1:




```
Instruction:        Operation:
LD      X0          ; Load A contact of X0
CJ      P10         ; Jump from CJ to P10
:
P10                 ; Pointer P10
LD      X1          ; Load A contact of X1
OUT     Y1          ; Drive Y1 coil
```

Program Example 2:




```
Instruction:        Operation:
LD      X0          ; Call the Subroutine P9 when X0 is ON
CALL    P9          ; Jump from CALL to P9
:
FEND                ; Program end
P9                  ; Pointer P9
LD      M1013       ; Load A contact of M1013
OUT     Y20         ; Drive Y20 coil
:
SRET                ; Subroutine return
```

Available devices:

13. ELCB-PB have 64 pointers; available from the range of P0 to P63.
14. ELC-PA/PV, ELC2-PB/PH/PA/PE/PV and ELCM-PH/PA have 256 pointers; available from the range of P0~P255.

3.4 Interrupt Pointers

Mnemonic	Function	Program steps
I	Interrupt program marker	1

ELCB	ELC		ELC2			ELCM
PB	PA	PV	PB	PA/PH/PE	PV	PH/PA

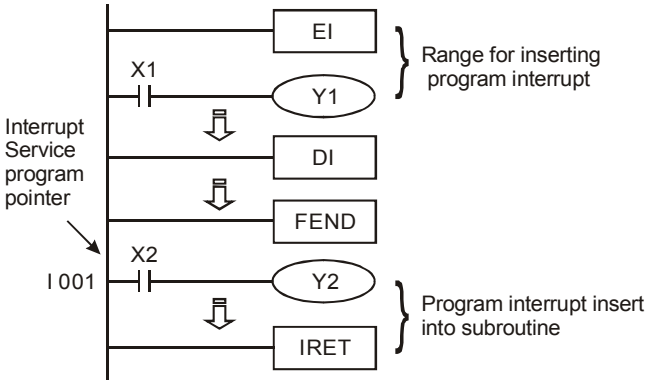
Description:

Interrupt programs begin with an interrupt pointer（I□□□）and end with the application command to terminate the interrupt and return (IRET). A special numbering system is used based on the interrupt type and input trigger method used.

Additional application instructions:

Interrupts must be used with application commands API 03 IRET, API 04 EI, API 05 DI.

Program Example:




```
Instruction:        Operation:
EI                  ; Interrupt Enable
LD      X1          ; Load A contact of X1
OUT     Y1          ; Energize Y1
:
DI                  ; Interrupt Disable
:
FEND                ; Program end
I001                ; Insert interrupt point
LD      X2          ; Load A contact of X2
OUT     Y2          ; Energize Y2
:
IRET                ; Interrupt return
```

Input Interrupts:

15. ELCB-PB have 4 input interrupts: (I001, X0), (I101, X1), (I201, X2) and (I301, X3).
16. ELC-PA have 6 input interrupts: (I001, X0), (I101, X1), (I201, X2), (I301, X3), (I401, X4) and (I501, X5).
17. In ELC-PA V1.2 and above, when I401 (X4) works with X0 (C235, C251 or C253), the value of (C243 or C255) will be stored in (D1180, D1181) and I501 (X5) works with X1 (C236), the value of high-speed counter (C236) will be stored in (D1198, D1199)
18. ELC-PV has 6 input interrupts: I000/I001(X0), I100/I101 (X1), I200/I201 (X2), I300/I301 (X3), I400/I401 (X4), I500/I501 (X5), 6 points (01, rising-edge trigger ⌐↑, 00, falling-edge trigger ↓¬)

19. ELC-PV2 has 16 input interrupts: I000/I001(X0), I100/I101 (X1), I200/I201 (X2), I300/I301 (X3), I400/I401 (X4), I500/I501 (X5), I700/I701(X6), I900/I901 (X10), I910/I911 (X11), I920/I921 (X12), I930/I931 (X13), I940/I941 (X14), I950/I951 (X15), I960/I961 (X16), I970/I971 (X17), 16 points (01, rising-edge trigger ⌐↑, 00, falling-edge trigger ↓¬)
20. ELCM-PH/PA, ELC2-PB/PH/PA/PE have 8 input interrupts: I000/I001 (X0), I100/I101 (X1), I200/I201 (X2), I300/I301 (X3), I400/I401 (X4), I500/I501 (X5), I600/I601 (X6), I700/I701 (X7), 8 points (01, rising-edge trigger ⌐↑, 00, falling-edge trigger ↓¬)

Timer Interrupts:

21. ELCB-PB have 1 timer interrupt point: I610~I699, (Timer resolution: 1ms)
22. ELC-PA have 2 timer interrupts: I601~I699, I701~I799, (Timer resolution: 1ms)
23. ELC-PV, ELC-PV2 has 3 timer interrupts: I601~I699, I701~I799, (Timer resolution: 1ms); I801~I899, (Timer resolution: 0.1ms)
24. ELCM-PH/PA, ELC2-PB/PH/PA/PE have 2 timer interrupts:I602~I699, I702~I799, (Timer resolution: 1ms)

Communication Interrupts:

25. ELCB-PB have 1 communication interrupt: I150
26. ELC-PV, ELC-PV2 has 3 communication interrupt: I150, I160, I170
27. ELCM-PH/PA, ELC2-PB/PH/PA/PE have 3 communication interrupt: I140, I150, I160

Counter Interrupts:

28. ELC-PA have 6 high-speed counter attained interrupt points:
    I010 (use with C235, C241, C244, C246, C247, C249, C251, C252, C254)
    I020 (use with C236, C243, C246, C247, C249, C251, C252, C254)
    I030 (use with C237, C242)
    I040 (use with C238, C245)
    I050 (use with C239)
    I060 (use with C240, C250)
29. ELC-PV, ELC-PV2 has 6 high-speed counter attained interrupt points: I010, I020, I030, I040, I050, I060.
30. ELCM-PH/PA, ELC2-PB/PH/PA/PE have 8 high-speed counter attained interrupt points: I010, I020, I030, I040, I050, I060, I070 and I080

Pulse interruption

ELC-PV, ELC-PV2 has four pulse interrupt points: I110, I120, I130, and I140.

Please see the following pages for more details on the interrupt functions.

Input interrupts – see page 3-46

Timer interrupts – see page 3-46

Communication interrupt – see page 3-46

Counter interrupts – see page 3-46

Pulse interruption – see page 3-46

Enabling interrupts, API 04 EI, Disabling interrupts, API 05 DI, Interrupt return, API 03 IRET – see page 3-45

3.5 Application Programming Instructions

1. ELC instructions each have a unique mnemonic. Most instructions are also given a number so the ELC knows how to execute the instruction. In the example below the numerical value given to the instruction is 00, also called the API (Application Programming Instruction) number. The mnemonic name is CJ and the function is Conditional Jump.

API	Mnemonic		Operands	Function
00	CJ	P	**S**	Conditional Jump

OP	Range	Program Steps
S	P0~P255	CJ, CJP: 3 steps

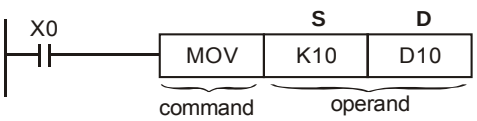
2. The table will be found at the beginning of each new instruction description. The area identified as 'Operands' will list the various operands that can be used with the instruction. Various identification letters will be used to associate each operand with its function, i.e. D-destination, S-source, n, m-number of elements. Additional numeric suffixes will be attached if there are more than one operand with the same function.
3. When using ELCSoft to create the application it is not necessary to remember the API number of an instruction since ELCSoft uses a drop down list to select an instruction or there is a button on the toolbar for the instruction.
4. Not all instructions and conditions apply to all ELC's models. Applicable controllers are identified by the boxes at the bottom right hand corner of the table. For more detailed information on each instruction, a second indicator box is used to identify the availability of pulse (P), single word (16) and double word (32) format.

ELCB			ELC						ELC2									ELCM		
			PA			PV			PB			PA/PH/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

5. No modification of the instruction mnemonic is required for 16 bit operation. However, pulse operation requires a 'P' to be added directly after the mnemonic while 32 bit operations require a 'D' to be added before the mnemonic. This means that if an instruction is being used with both pulse and 32 bit operation it would look line…D***P where *** was the basic mnemonic.

Instruction Composition

Instructions consist of either just the instruction or the instruction followed by operand parameters



Command : Indicates the instruction type

Operand : Indicates additional parameters needed

S	Source operand: if there is more than 1 source operand, then S_1, S_2… is used to describe them.
D	Destination operand
If an operand is only represented as a constant (K, H) then m, m_1, m_2, n, n_1, n_2 is used to describe them.	

The Length of Operand (16-bit or 32-bit instruction)

The length of an operand can be divided into two groups: 16-bit and 32-bit. A"D" before an instruction means it's a 32-bit instruction.

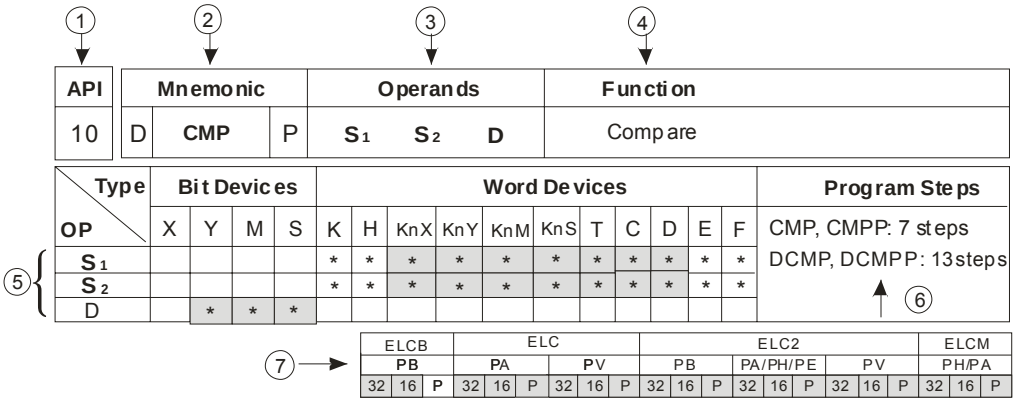
16-bit MOV instruction

When X0=ON, K10 is moved into D10.

32-bit DMOV instruction

When X1=ON, the 32-bit contents of D11, D10 is moved to D21, D20.

Description of the format for the application programming instructions



① API number for the instruction

② The core mnemonic code of the instruction
A "D" in this box means the instruction is a 32 bit instruction if a "D" is added as a prefix
A "P" in this box indicates the instruction can be used as a pulse instruction

③ The operand format of the instruction

④ The description of the instruction

⑤ The symbol **"*"** means this instruction supports the associated data type

⑥ The number of program steps for the instruction

⑦ The ELC models that support the following for each instruction:
A "16" represents 16-bit instruction format
A "32" represents 32-bit instruction format

A "P" represents pulse instruction format

Note: If the cells adjacent to the operands are grayed, this means they support index registers E and F for those cells in the table above. For example, device D of operand S1 supports index registers E and F.

Continuous execution vs. Pulse execution

1. The execution type for instructions can be divided into two types: continuous execution instructions and pulse execution instructions. The execution time is shorter when instructions are not executed. Using the pulse option for an instruction will reduce the scan time of the program.
2. The 'pulse' function allows the associated instruction to be activated on the rising edge of the control input. The output instructions are true for one program scan.
3. Thereafter, while the control input remains ON, the associated instruction is false. To re-execute the instruction the control input must be turned OFF then ON again.

Pulse execution instruction

When X0 goes from OFF→ON, the MOVP instruction will be executed one time and instruction cannot be re-executed again in the same program scan. This is a pulse execution instruction.

Continuous execution instruction

When X1=ON, the MOV instruction will execute every program scan This is called continuous execution instruction.

The above figures show that when X0, X1=OFF, the instruction will not be executed and the contents of the destination operand "D12" will remain unchanged.

Operands

31. Bit addresses such as X, Y, M, S can be used in numerical order to define a WORD or a portion of a word. Successive bits can be written to a word address and a word address can be broken into bits. This is accomplished using the MOV instruction and a special designator in front of the bit address, such as: KnX, KnY, KnM, KnS. The n represents 4 bits, 8 bits, 12 bits or 16 bits to be moved into a word address or moved from a word address. n=1 represents 4 bits, n=2 represents 8 bits, n=3 represents 12 bits and n=4 represents 16 bits.
32. Data register D, Timer T, Counter C and Index Register E, F can all be assigned as operands.
33. A D register is a 16-bit register. It can also be assigning as a 32-bit register which consumes two consecutive D registers.
34. If the operand of a 32-bit instruction is assigned to D0, D1 is also used to comprise a 32-bit value. The upper 16-bits is D1 and D0 is the lower 16-bits.

35. If the 32-bit counters (C200~C255) are used as Data registers, each counter is a 32-bit counter. Only instructions using 32-bit operands can be assigned.

Operand Data format

36. X, Y, M, S are bit addresses and are either ON/OFF.
37. 16-bit (or 32-bit) addresses T, C, D, E, F are data registers and are defined as word or double word addresses.
38. If a Kn is placed in front of X, Y, M and S it will be defined as a word device, where n=1 means 4-bits. So 16-bits are defined from K1 to K4, and 32-bits are defined from K1 to K8. For example, K2M0 represents 8-bits from M0 to M7.

When X0=ON, move the contents of M0 to M7 to D10 bits 0 to 7, and bits 8 to 15 are set to 0.

Kn values

16-bit instruction	
Specified Number of Digits (16-bit instruction): K-32,768~K+32,767	
16-bit instruction: (K1~K4)	
K1 (4 points)	0~15
K2 (8 points)	0~255
K3 (12 points)	0~4,095
K4 (16 points)	-32,768~+32,767

32-bit instruction	
Specified Number of Digits (32-bit instruction): K-2,147,483,648~K+2,147,483,647	
32-bit instruction: (K1~K8)	
K1 (4 points)	0~15
K2 (8 points)	0~255
K3 (12 points)	0~4,095
K4 (16 points)	0~65,535
K5 (20 points)	0~1,048,575
K6 (24 points)	0~167,772,165
K7 (28 points)	0~268,435,455
K8 (32 points)	-2,147,483,648~+2,147,483,647

Flags

1. General Flags

   The following flags are available for the ELC:

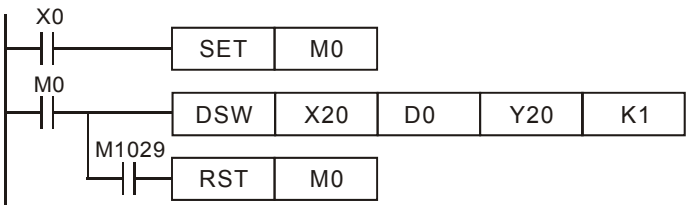
   M1020: Zero flag

   M1021: Borrow flag

   M1022: Carry flag

   M1029: Instruction execution completed flag

   When executing an instruction, all flags will be turned to ON or OFF according to the operation result of the instruction. However, if the instruction is not executed, the ON/OFF state of the flags will remain unchanged.



When X0=ON, the DSW instruction is energized. When X0=OFF, M0 is not reset until the instruction execution is complete

2. Error Operation Flags

If the instruction or the assigned operands result in an error, the error flags and associated error codes in the following table will be displayed during the execution of the application instructions.

M1067 **D1067** **D1069**	When error operations occur, M1067=ON, D1067 will show the error code and D1069 will show the error address. If other errors occur, the contents of D1067 and D1069 will be refreshed. (when the error is reset, M1067=OFF)
M1068 **D1068**	When error operations occur, M1068=ON, D1068 will show the error address. If other errors occur, the contents of D1068 will not be refreshed. M1068 must be reset to OFF, otherwise the error will remain.

3. Flags to Extend Functions

Some instructions can extend their function by using some special flags.

Example: The RS instruction can be either an 8-bit or a 16-bit instruction depending on the state of M1161.

Limited Use Instructions

Some instructions can be used unlimited times in the program, others can be used only a particular number of times in a program.

1. These instructions can only be used once in a program:

PWM	ELCB-PB Model
IST	ELC-PA/PV, ELCB-PB, ELCM-PH/PA, ELC2-PB/PH/PA/PE/PV Models
SEGL	ELCB-PB Model
DABSR	ELC- PV, ELCM-PH/PA, ELC2-PB/PH/PA/PE/PV Models

2. These instructions can only be used twice in a program:

PLSY	ELCB-PB Models
PLSR	ELCB-PB Models
PR	ELC-PA/PV, ELCM-PH/PA, ELC2-PB/PH/PA/PE/PV Models
SEGL	ELC-PV, ELC2-PV Models

3. These instructions can only be used four times in a program:

HOUR	ELC-PA Models

4. These instructions can only be used eight times in a program:

TTMR	ELC-PA, ELCM-PH/PA, ELC2-PB/PH/PA/PE/PV Models

5. The DHSCS and DHSCR instructions can only be executed simultaneously a maximum of 4 times in the program for ELCB-PB models.
6. The DHSCS, DHSCR and DHSZ instructions can only be executed simultaneously a maximum of five times in the program for ELC-PA models.
7. For counters C232~C242, the maximum number of times DHSCS, DHSCR and DHSZ instructions can be used in a program is 6. DHSZ can only be used a maximum of 6 times in a program for ELCM-PH/PA, ELC2-PB/PH/PA/PE models.
8. For counters C243, C245~C248, C251, C252, the maximum number of DHSCS, DHSCR and DHSZ instructions can be used in a program is 4. DHSZ takes up 2 times of the total available number of times this instruction can be used in a program for ELCM-PH/PA, ELC2-PB/PH/PA/PE models.
9. For counters C244, C249, C250, C253, C254, the maximum number of times DHSCS, DHSCR and DHSZ instructions can be used in a program is 4. DHSZ takes up 2 times of the total available number of times this instruction can be used in a program for ELCM-PH/PA , ELC2-PB/PH/PA/PE models.

Limitations on executing the same instruction multiple times

There is no limitation on the number of times an instructions listed below may be used in a program. But there are limitations on how many occurances of the same instruction that can be simultaneously executed. in the program.

1. Instructions which can be executed only once:

ELCB-PB, ELC-PA/PV, ELC2-PV2: API 56 SPD, API 75 ARWS, API 80 RS, API 100 MODRD, API 101 MODWR, API 150 MODRW.

ELC-PA/PV, ELC2-PV2: API 52 MTR, API 69 SORT, API 70 TKY, API 71 HKY, API 151 PWD.

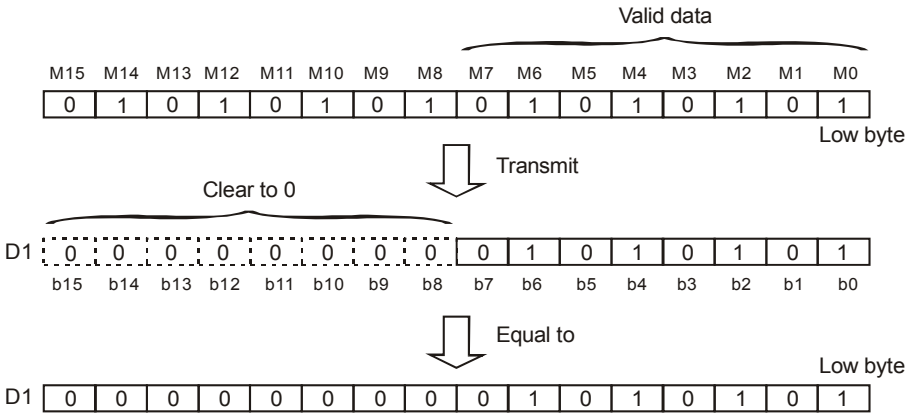
ELC-PA: API 72 DSW, API 74 SEGL.

2. Instructions which can be executed only twice: API 58 PWM (ELC-PV, ELC2-PV), API 59 PLSR (ELC-PV, ELC2-PV).
3. Instructions which can be executed only 4 times: API 57 PLSY (ELC-PV, ELC2-PV), API 58 PWM (ELC-PV, ELC2-PV).
4. Instructions which can be executed only 8 times: API 64 TTMR(ELC-PV, ELC2-PV)
5. In ELC-PA, there is on limitation on the times of using the high-speed output instructions PLSY, PWM and PLSR, bit only one high-speed output instruction will be enabled in every scan.

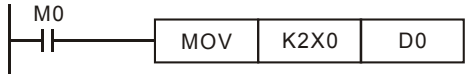
6. In ELC-PV, ELC2-PV, there is no limitation on the times of using hardware high-speed counter instructions DHSCS, DHSCR and DHSZ, but when the three instructions are enabled at the same time, DHSCS will occupy 1 memory unit, DHSCR 1 memory unit, and DHSZ 2 memory units. The total memeory units occupied by the three instructions cannot be more than 8 units. If there are more than 8 memory units occupied, the ELC system will execute the instruction that is scanned first and ignore the rest.
7. In ELCM-PH/PA, ELC2-PB/PH/PA/PE only 1 instruction can be executed at the same scan cycle: API 52 MTR, API 69 SORT, API 70 TKY, API 71 HKY, API 72 DSW, API 74 SEGL, API 75 ARWS.
8. In ELCM-PH/PA, ELC2-PB/PH/PA/PE, only 4 instruction can be executed at the same scan cycle: API 56 SPD, API 169 HOUR.
9. In ELCM-PH/PA, ELC2-PB/PH/PA/PE, there is no limitation on the times of using the high-speed output instructions API 57 PLSY, API 58 PWM, API 59 PLSR, API 156DZRN, API 158 DDRVI, API 159 DDRVA and API 195 DPTPO, but only one high-speed output instruction will be executed in the same scan time.
10. In ELCM-PH/PA, ELC2-PB/PH/PA/PE, there is no limitation on the times of using the communication instructions API 80 RS, API 100 MODRD, API 101 MODWR, API 150 MODRW, but only one communication instruction will be executed on single COM port during the same scan cycle.

Numeric Values

1. Devices such as X, Y, M, S are bit addresses and there are only two states, ON and OFF. However, T, C, D, E, F are data registers or word addresses. Although a bit device can only be a single point ON/OFF, they can also be used as numeric values in the operands of instructions if the Kn designation is used in front of the bit address.. The designation Kn placed in front of a bit address indicates that the bit address is a starting address and the n determines how many bits follow, where “n” is in the range of 1 to 8. Each n number represents 4 bits. For example, K1M0 represents M0-M3.
2. 16-bits can be represented with K1 to K4, and 32-bits can be represented with K1 to K8. For example, K2M0 means there are 8-bits from M0 to M7. For the example below, MOV K2M0 D1:



3. Moving K1M0, K2M0, K3M0 to a 16-bit register clears the upper unused bits in the destination word. It's the same as for K1M0, K2M0, K3M0, K4M0, K5M0, K6M0, K7M0 to 32-bit registers. The upper bits not included in the move are cleared in the 32-bit destination value.
4. The unused upper bits will be 0 if K1 to K3 is used in a 16-bit operation or K1 to K7 is used in a 32-bit operation. Therefore, the operation result is a positive decimal value



The state of input bits X0-X7 will be assigned to the lower 8 bits of D0. Each of the upper 8 bits of D0 will contain 0

Assign Continuous Bit Numbers

As already explained, bit devices can be grouped into 4 bit units. The "n" in KnM0 defines the number of groups of 4 bits that are used for this data operation. For data register D, consecutive D registers refers to D0, D1, D2, D3, D4…; For bit devices with Kn, consecutive numbers refers to:

K1X0	K1X4	K1X10	K1X14…
K2Y0	K2Y10	K2Y20	Y2X30…
K3M0	K3M12	K3M24	K3M36…
K4S0	K4S16	K4S32	K4S48…

Note: When moving bits to a word or a double word, realize that the upper bits of the destination that are not included in the source, i.e. n=1, 2 or 3, will be filled with 0s. For example, if K4M0 is moved to a double word using a DMOV instruction, the upper word will contain all 0s after the DMOV is executed.

Floating Point Operation

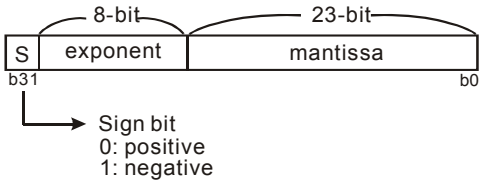
When performing integer math, the decimal point from a division operation for example will be discarded. In other words, 40 ÷ 3 = 13, remainder is 1 and the decimal point will be discarded. But using floating point math, the decimal point is used.

The application instructions related to floating point operation are shown in the following table.

FLT	DECMP	DEZCP	DMOVR	DRAD
DDEG	DEBCD	DEBIN	DEADD	DESUB
DEMUL	DEDIV	DEXP	DLN	DLOG
DESQR	DPOW	INT	DSIN	DCOS
DTAN	DASIN	DACOS	DATAN	DSINH
DCOSH	DTANH	DADDR	DSUBR	DMULR
DDIVR	FLD※	FAND※	FOR※	

Binary Floating Point

The ELC controllers use the IEEE754 method to represent floating point numbers with 32-bits:



The range of a 32-bit floating point value is from $\pm 2^{-126}$ to $\pm 2^{+128}$, i.e. from $\pm 1.1755 \times 10^{-38}$ to $\pm 3.4028 \times 10^{+38}$.

Example 1: using 32-bit floating point to represent decimal number 23

Step 1: convert 23 to binary number: 23.0=10111

Step 2: Normalizing the binary: $10111=1.0111 \times 2^4$, 0111 is mantissa and 4 is an exponent.

Step 3: get exponent: ∵E-B=4 →E-127=4 ∴E=131=10000011_2

Step 4: We can now combine the sign, exponent, and normalized mantissa into the binary IEEE short real representation.

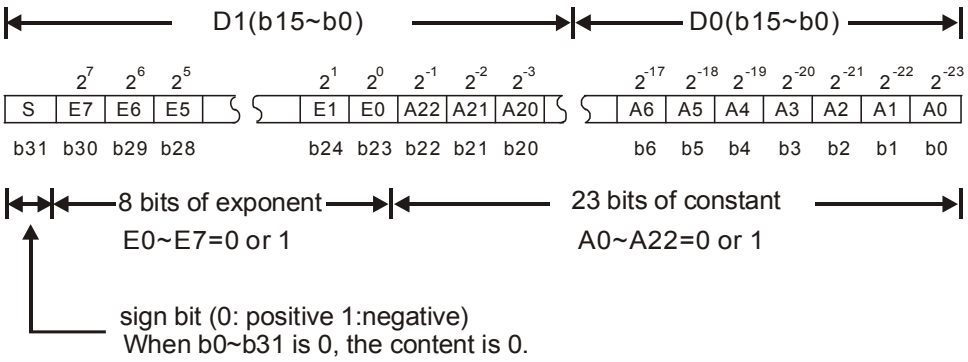
0 100**0001**1 011**1000**0000**0000**0000**0000**$_2$=41B80000$_{16}$

Example 2: using 32-bit floating point to represent decimal number –23

The conversion steps are the same as decimal number 23. Only need to modify sign bit from 0 to 1 to get value.

1 100**0001**1 011**1000**0000**0000**0000**0000**$_2$=C1B80000$_{16}$

ELC also uses two registers with continuous number to store binary floating point. The following is the example that uses register (D1, D0) to store binary floating point.



Decimal Floating Point

1. Binary floating point is not practical for most applications, therefore, binary floating point format can be converted to decimal floating point format for performing floating point math..
2. Decimal floating point values are stored in two consecutive registers. The least significant register is where the constant value (mantissa) is stored and the upper register is where the exponent is stored.

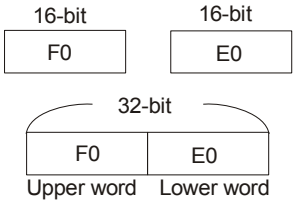
   For example, using registers (D1, D0) to store a decimal floating point value.
   Decimal floating point = [constant D0] X $10^{[\text{exponent D1}]}$
   constant D0 = ±1,000~±9,999, exponent D1 = - 41~+35
   the most significant bit of (D1, D0) is the sign bit.
   The range of decimal numbers is from $\pm 1175\times10^{-41}$ to $\pm 3402\times10^{+35}$.
3. Decimal floating point can be used in the following instructions.
   The conversion instruction for Binary floating point → Decimal floating point (DEBCD)
   The conversion instruction for Decimal floating point → Binary floating point (DEBIN)
4. Zero flag (M1020), Borrow flag (M1021) and carry flag (M1022). The flags that correspond to the floating point instructions are:
   1. Zero flag: when the result is 0, M1020=ON.
   2. Borrow flag: when a borrow is generated, M1021=ON
   3. Carry flag: when the absolute value of result exceeds usage range, M1022=ON

Index register E, F

The index registers are 16-bit registers. There are 2 index registers for ELC-PB, ELCB-PB models (E and F), 8 for ELC-PC/PA/PH models (E0~E3, F0~F3), and 16 for ELCM-PH/PA, ELC-PV models (E0 ~ E7 and F0 ~ F7).

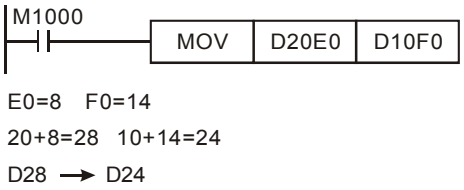
E and F are also 16-bit registers just the same as D-registers. They are read /write registers.

When using a 32-bit index register, the combination of E, F are as follows. (E0, F0), (E1, F1), (E2, F2), (E3, F3), (E4, F4), (E5, F5), (E6, F6), (E7, F7).



If using a 32-bit register, you must specify an E index register. In this case, the specified E register is used and its corresponding F register is also used (When the ELC starts-up, it is recommended to use the MOVP instruction to clear the contents of F and reset it to 0)

As the right figure shows, the addresses will change based the contents of E, F. This is called “Index” addressing.



For example, if E0=8 then D20E0 represents address D (20+8). When F0=14, the destination address becomes D24.

Data types supported in ELC-PB, ELCB-PB series: P, X, Y, M, S, KnX, KnY, KnM, KnS, T, C, D.

Data types supported in ELC-PC/PA/PH, ELCM-PH/PA series: P, X, Y, M, S, KnX, KnY, KnM, KnS, T, C, D.

Data types supported in ELC-PV series: P, I, X, Y, M, S, K, H, KnX, KnY, KnM, KnS, T, C, D.

If E and F are used with a constant value, use @ between the K or H value and the E or F index register. For example: "MOV K10@E0 D0F0”

3.6 Numerical List of Instructions

Loop Control

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB-PB	ELC-PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC-PV ELC2-PV	16	32
00	CJ	-	✓	Conditional Jump	✓	✓	✓	✓	3	-
01	CALL	-	✓	Call Subroutine	✓	✓	✓	✓	3	-
02	SRET	-	-	Subroutine Return	✓	✓	✓	✓	1	-
03	IRET	-	-	Interrupt Return	✓	✓	✓	✓	1	-
04	EI	-	-	Enable Interrupt	✓	✓	✓	✓	1	-
05	DI	-	-	Disable Interrupt	✓	✓	✓	✓	1	-
06	FEND	-	-	Terminate the main routine program	✓	✓	✓	✓	1	-
07	WDT	-	✓	Reset the Watchdog Timer	✓	✓	✓	✓	1	-
08	FOR	-	-	Loop Begin	✓	✓	✓	✓	3	-
09	NEXT	-	-	Loop End	✓	✓	✓	✓	1	-

Transmission Comparison

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB-PB	ELC-PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC-PV ELC2-PV	16	32
10	CMP	DCMP	✓	Compare	✓	✓	✓	✓	7	13
11	ZCP	DZCP	✓	Zone Compare	✓	✓	✓	✓	9	17
12	MOV	DMOV	✓	Move	✓	✓	✓	✓	5	9
13	SMOV	-	✓	Shift Move	-	✓	✓	✓	11	-
14	CML	DCML	✓	Compliment and Move	✓	✓	✓	✓	5	9
15	BMOV	-	✓	Block Move	✓	✓	✓	✓	7	-
16	FMOV	DFMOV	✓	Fill and Move	✓	✓	✓	✓	7	13
17	XCH	DXCH	✓	Data Exchange	✓	✓	✓	✓	5	9
18	BCD	DBCD	✓	Convert BIN to BCD	✓	✓	✓	✓	5	9
19	BIN	DBIN	✓	Convert BCD to BIN	✓	✓	✓	✓	5	9

Four Fundamental Operations Arithmetic

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB -PB	ELC -PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC -PV ELC2 -PV	16	32
20	ADD	DADD	✓	Addition	✓	✓	✓	✓	7	13
21	SUB	DSUB	✓	Subtraction	✓	✓	✓	✓	7	13
22	MUL	DMUL	✓	Multiplication	✓	✓	✓	✓	7	13
23	DIV	DDIV	✓	Division	✓	✓	✓	✓	7	13
24	INC	DINC	✓	Increment	✓	✓	✓	✓	3	5
25	DEC	DDEC	✓	Decrement	✓	✓	✓	✓	3	5
26	WAND	DAND	✓	Logical AND	✓	✓	✓	✓	7	13
27	WOR	DOR	✓	Logical OR	✓	✓	✓	✓	7	13
28	WXOR	DXOR	✓	Exclusive XOR	✓	✓	✓	✓	7	13
29	NEG	DNEG	✓	Negative (2's Compliment)	✓	✓	✓	✓	3	5

Rotation and Displacement

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB -PB	ELC -PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC -PV ELC2 -PV	16	32
30	ROR	DROR	✓	Rotate Right	✓	✓	✓	✓	5	9
31	ROL	DROL	✓	Rotate Left	✓	✓	✓	✓	5	9
32	RCR	DRCR	✓	Rotate Right with Carry	✓	✓	✓	✓	5	9
33	RCL	DRCL	✓	Rotate Left with Carry	✓	✓	✓	✓	5	9
34	SFTR	-	✓	Bit Shift Right	✓	✓	✓	✓	9	-
35	SFTL	-	✓	Bit Shift Left	✓	✓	✓	✓	9	-
36	WSFR	-	✓	Word Shift Right	-	✓	✓	✓	9	-
37	WSFL	-	✓	Word Shift Left	-	✓	✓	✓	9	-
38	SFWR	-	✓	Shift Register Write	-	✓	✓	✓	7	-
39	SFRD	-	✓	Shift Register Read	-	✓	✓	✓	7	-

Data Operation

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB -PB	ELC -PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC -PV ELC2 -PV	16	32
40	ZRST	-	✓	Zone Reset	✓	✓	✓	✓	5	-

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB -PB	ELC -PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC -PV ELC2 -PV	16	32
41	DECO	-	✓	Decode	✓	✓	✓	✓	7	-
42	ENCO	-	✓	Encode	✓	✓	✓	✓	7	-
43	SUM	DSUM	✓	Sum of ON bits	✓	✓	✓	✓	5	9
44	BON	DBON	✓	Bit ON Test	✓	✓	✓	✓	7	13
45	MEAN	DMEAN	✓	Mean Value	✓	✓	✓	✓	7	13
46	ANS	-	-	Alarm Set	-	✓	✓	✓	7	-
47	ANR	-	✓	Alarm Reset	-	✓	✓	✓	1	-
48	SQR	DSQR	✓	Square Root	✓	✓	✓	✓	5	9
49	FLT	DFLT	✓	Floating Point	✓	✓	✓	✓	5	9

High Speed Processing

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB -PB	ELC -PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC -PV ELC2 -PV	16	32
50	REF	-	✓	Refresh I/O Immediately	✓	✓	✓	✓	5	-
51	REFF	-	✓	Refresh and Filter Adjust	-	✓	✓	✓	3	-
52	MTR	-	-	Input Matrix	-	✓	✓	✓	9	-
53	-	DHSCS	-	High Speed Counter Set	✓	✓	✓	✓	-	13
54	-	DHSCR	-	High Speed Counter Reset	✓	✓	✓	✓	-	13
55	-	DHSZ	-	HSC Zone Compare	-	✓	✓	✓	-	17
56	SPD	-	-	Speed Detection	✓	✓	✓	✓	7	-
57	PLSY	DPLSY	-	Pulse Output	✓	✓	✓	✓	7	13
58	PWM	-	-	Pulse Width Modulation	✓	✓	✓	✓	7	-
59	PLSR	DPLSR	-	Pulse Ramp	✓	✓	✓	✓	9	17

Convenience Instruction

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB -PB	ELC -PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC -PV ELC2 -PV	16	32
60	IST	-	-	Manual/Auto Control	✓	✓	✓	✓	7	-
61	SER	DSER	✓	Search a Data Stack	-	✓	✓	✓	9	17
62	ABSD	DABSD	-	Absolute Drum Sequencer	-	✓	✓	✓	9	17

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB -PB	ELC -PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC -PV ELC2 -PV	16	32
63	INCD	-	-	Incremental drum sequencer	-	✓	✓	✓	9	-
64	TTMR	-	-	Alternate timer	-	✓	✓	✓	5	-
65	STMR	-	-	Special timer	-	✓	✓	✓	7	-
66	ALT	-	✓	ON/OFF alternate instruction	✓	✓	✓	✓	3	-
67	RAMP	DRAMP	-	Ramp signal	-	✓*	✓	✓	9	17
68	DTM	-	✓	Data transform and move	-	-	✓	✓*	9	-
69	SORT	DSORT	-	Data sort	-	✓*	✓	✓	11	21

*:API 67 DRAMP, API69 DSORT don't support ELC-PA; API 68 DTM don't support ELC-PV.

External I/O Display

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB -PB	ELC -PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC -PV ELC2 -PV	16	32
70	TKY	DTKY	-	10-key keypad input	-	✓	✓	✓	7	13
71	HKY	DHKY	-	16-key keypad input	-	✓	✓	✓	9	17
72	DSW	-	-	Digital Switch input	-	✓	✓	✓	9	-
73	SEGD	-	✓	Decode the 7-step display panel	✓	✓	✓	✓	5	-
74	SEGL	-	-	7-step display scan output	✓	✓	✓	✓	7	-
75	ARWS	-	-	Arrow keypad input	-	✓	✓	✓	9	-
76	ASC	-	-	ASCII code conversion	-	✓	✓	✓	11	-
77	PR	-	-	Output ASCII code	-	✓	✓	✓	5	-

Serial I/O

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB -PB	ELC -PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC -PV ELC2 -PV	16	32
78	FROM	DFROM	✓	Read special module CR data	✓	✓	✓	✓	9	17
79	TO	DTO	✓	Special module CR data write in	✓	✓	✓	✓	9	17
80	RS	-	-	Serial data communication	✓	✓	✓	✓	9	-
81	PRUN	DPRUN	✓	Octal number system transmission	-	✓	✓	✓	5	9
82	ASCII	-	✓	Convert HEX to ASCII	✓	✓	✓	✓	7	-

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB-PB	ELC-PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC-PV ELC2-PV	16	32
83	HEX	-	✓	Convert ASCII to HEX	✓	✓	✓	✓	7	-
84	CCD	-	✓	Check sum	-	✓	✓	✓	7	-
85	VRRD	-	✓	Volume read	-	-	-	✓	5	-
86	VRSC	-	✓	Volume scale	-	-	-	✓	5	-
87	ABS	DABS	✓	Absolute value	✓	✓	✓	✓	3	5
88	PID	DPID	-	PID calculation	✓	✓	✓	✓	9	17

3

Basic Instruction

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB-PB	ELC-PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC-PV ELC2-PV	16	32
89	PLS	-	-	Rising-edge output	✓	✓	✓	✓	3	-
90	LDP	-	-	Rising-edge pulse	✓	✓	✓	✓	3	-
91	LDF	-	-	Falling-edge pulse	✓	✓	✓	✓	3	-
92	ANDP	-	-	Serial connection of rising-edge pulse	✓	✓	✓	✓	3	-
93	ANDF	-	-	Serial connection falling-edge pulse	✓	✓	✓	✓	3	-
94	ORP	-	-	Parallel connection of rising-edge pulse	✓	✓	✓	✓	3	-
95	ORF	-	-	Parallel connection of falling-edge pulse	✓	✓	✓	✓	3	-
96	TMR	-	-	Timer	✓	✓	✓	✓	4	-
97	CNT	DCNT	-	Counter	✓	✓	✓	✓	4	6
98	INV	-	-	Inverting operation	✓	✓	✓	✓	1	-
99	PLF	-	-	Falling-edge output	✓	✓	✓	✓	3	-
255	ATMR	-	-	Contact type timer	-	-	✓	✓*	5	-

*:API 255 ATMR does not support ELC-PV

Communication Instruction

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB -PB	ELC -PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC -PV ELC2 -PV	16	32
100	MODRD	-	-	MODBUS data Read	✓	✓	✓	✓	7	-
101	MODWR	-	-	MODBUS data write in	✓	✓	✓	✓	7	-
107	LRC	-	✓	LRC check sum	✓	✓	✓	✓	7	-
108	CRC	-	✓	CRC check sum	✓	✓	✓	✓	7	-
150	MODRW	-	-	MODBUS data read/write in	✓	✓	✓	✓	11	-
113	ETHRW	-	-	Ethernet communication	-	-	✓*	✓*	9	-

*:API 113 ETHRW does not support ELC-PV,ELC2-PB,ELCM-PH/PA

Floating Operation

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB -PB	ELC -PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC -PV ELC2 -PV	16	32
110	-	DECMP	✓	Floating point compare	✓	✓	✓	✓	-	13
111	-	DEZCP	✓	Floating point zone compare	✓	✓	✓	✓	-	17
112		DMOVR	✓	Floating point data Move	✓	✓	✓	✓		9
116	-	DRAD	✓	Degree → Radian	-	✓	✓	✓	-	9
117	-	DDEG	✓	Radian → Degree	-	✓	✓	✓	-	9
118	-	DEBCD	✓	Float to scientific conversion	✓	✓	✓	✓	-	9
119	-	DEBIN	✓	Scientific to float conversion	✓	✓	✓	✓	-	9
120	-	DEADD	✓	Floating point addition	✓	✓	✓	✓	-	13
121	-	DESUB	✓	Floating point subtraction	✓	✓	✓	✓	-	13
122	-	DEMUL	✓	Floating point multiplication	✓	✓	✓	✓	-	13
123	-	DEDIV	✓	Floating point division	✓	✓	✓	✓	-	13
124	-	DEXP	✓	Floating point exponent operation	✓	✓	✓	✓	-	9
125	-	DLN	✓	Floating natural logarithm operation	✓	✓	✓	✓	-	9
126	-	DLOG	✓	Floating point logarithm operation	✓	✓	✓	✓	-	13
127	-	DESQR	✓	Square root of binary floating point	✓	✓	✓	✓	-	9

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB-PB	ELC-PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC-PV ELC2-PV	16	32
128	-	DPOW	✓	Floating point power operation	✓	✓	✓	✓	-	13
129	INT	DINT	✓	Floating point to integer	✓	✓	✓	✓	5	9
130	-	DSIN	✓	Floating point Sine operation	✓	✓	✓	✓	-	9
131	-	DCOS	✓	Floating point Cosine operation	✓	✓	✓	✓	-	9
132	-	DTAN	✓	Floating point Tangent operation	✓	✓	✓	✓	-	9
133	-	DASIN	✓	Floating point Arcsine operation	-	✓	✓	✓	-	9
134	-	DACOS	✓	Floating point Arccosine operation	-	✓	✓	✓	-	9
135	-	DATAN	✓	Floating point Arctangent operation	-	✓	✓	✓	-	9
136	-	DSINH	✓	Hyperbolic Sine	-	-	-	✓	-	9
137	-	DCOSH	✓	Hyperbolic Cosine	-	-	-	✓	-	9
138	-	DTANH	✓	Hyperbolic Tangent	-	-	-	✓	-	9
172	-	DADDR	✓	Floating Point Number Addition	✓	✓	✓	✓	-	13
173	-	DSUBR	✓	Floating Point Number Subtraction	✓	✓	✓	✓	-	13
174	-	DMULR	✓	Floating Point Number Multiplication	✓	✓	✓	✓	-	13
175	-	DDIVR	✓	Floating Point Number Division	✓	✓	✓	✓	-	13

Additional Instruction

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB-PB	ELC-PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC-PV ELC2-PV	16	32
143	DELAY	-	✓	Delay instruction	-	✓	✓	✓	3	-
144	GPWM	-	-	General pulse width modulation	-	✓	✓	✓	7	-
145	FTC	-	-	Fuzzy temperature control	-	✓	✓	✓	9	-
146	CVM	-	-	Valve Control	-	-	-	✓	7	-
147	SWAP	DSWAP	✓	Swap high/low byte	✓	✓	✓	✓	3	5
148	MEMR	DMEMR	✓	MEMORY read	-	✓	✓*	✓	7	13
149	MEMW	DMEMW	✓	MEMORY write in	-	✓	✓*	✓	7	13
151	PWD	-	-	Detection of Input Pulse Width	-	-	-	✓	5	-

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB-PB	ELC-PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC-PV ELC2-PV	16	32
152	RTMU	-	-	Start of the Measurement of Execution Time of I Interruption	-	-	-	✓	5	-
153	RTMD	-	-	End of the Measurement of the Execution Time of I Interruption	-	-	-	✓	3	-
154	RAND	DRAND	✓	Random value	-	✓	✓	✓	7	13
168	MVM	DMVM	✓	Masked Move	-	-	✓	✓	7	13
176	MMOV	-	✓	16-bit→32-bit Conversion	-	✓	✓	✓	5	-
177	GPS	-	-	GPS data receiving	-	-	✓*	✓*	5	-
178	-	DSPA	-	Solar cell positioning	-	-	✓*	✓*	-	9
179	WSUM	DWSUM	✓	Sum of multiple devices	-	-	✓	✓	7	13
196	HST	-	✓	High Speed Timer	-	-	-	✓	3	-
202	SCAL	-	✓	Calculation of Proportional Value	✓	✓	✓	✓	9	-
203	SCLP	DSCLP	✓	Calculation of Parameter Proportional Value	✓	✓	✓	✓	7	13
205	CMPT	DCMPT	✓	Compare table	-	-	✓	✓*	9	17
207	CSFO	-	-	Catch speed and proportional output	-	-	✓*	✓*	7	-

*: ELC2-PB/PE does not support API148 DMEMR/API149 DMEMW.

*: ELC-PV does not support API205 DCMPT.

*: ELC-PV and ELC2-PE do not support API177 GPS and API178 DSPA and API207 CSFO.

Positioning Control

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB-PB	ELC-PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC-PV ELC2-PV	16	32
155	-	DABSR	-	ABS current value read	-	✓	✓	✓	-	13
156	ZRN	DZRN	-	Zero point return	-	-	✓*	✓	9	17
157	PLSV	DPLSV	-	Adjustable Speed Pulse Output	-	-	✓	✓	7	13
158	DRVI	DDRVI	-	Relative positioning	-	-	✓*	✓	9	17
159	DRVA	DDRVA	-	Absolute positioning	-	-	✓*	✓	9	17
191	-	DPPMR	-	2-Axis Relative Point to Point Motion	-	-	✓*	✓	-	17

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB-PB	ELC-PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC-PV ELC2-PV	16	32
192	-	DPPMA	-	2-Axis Absolute Point to Point Motion	-	-	✓*	✓	-	17
193	-	DCIMR	-	2-Axis Relative Position Arc Interpolation	-	-	✓*	✓	-	17
194	-	DCIMA	-	2-Axis Absolute Position Arc Interpolation	-	-	✓*	✓	-	17
195	-	DPTPO	-	Single-Axis pulse output by table	-	-	✓	✓	-	13
197	-	DCLLM	-	Close loop position control	-	-	✓	✓	-	17
198	-	DVSPO	-	Variable speed pulse output	-	-	✓	✓*	-	17
199	-	DICF	✓	Immediately change frequency	-	-	✓	✓*	-	13

*:16-bits API156 ZRN,API 158 DRVI, API159 DRVA don't support ELCM-PH/PA, ELC2-PB/PH/PA/PE.

.*: API191 DPPMR, API192 DPPMA, API193 DCIMR, API194 DCIMA don't support ELC2-PB.

*: API198 DVSPO and API199 DICF don't support ELC-PV.

Perpetual Calendar

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB-PB	ELC-PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC-PV ELC2-PV	16	32
160	TCMP	-	✓	Calendar data comparison	-	✓	✓	✓	11	-
161	TZCP	-	✓	Calendar data zone comparison	-	✓	✓	✓	9	-
162	TADD	-	✓	Calendar data addition	-	✓	✓	✓	7	-
163	TSUB	-	✓	Calendar data subtraction	-	✓	✓	✓	7	-
166	TRD	-	✓	Calendar data read	-	✓	✓	✓	3	-
167	TWR	-	✓	Calendar data write in	-	✓	✓	✓	3	-
169	HOUR	DHOUR	-	Hour meter	-	✓	✓	✓	7	13

Gray Code

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB-PB	ELC-PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC-PV ELC2-PV	16	32
170	GRY	DGRY	✓	Convert BIN to Gray code	-	✓	✓	✓	5	9
171	GBIN	DGBIN	✓	Convert Gray code to BIN	-	✓	✓	✓	5	9

Matrix Handling

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB -PB	ELC -PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC -PV ELC2 -PV	16	32
180	MAND	-	✓	Matrix AND	-	✓	✓	✓	9	-
181	MOR	-	✓	Matrix OR	-	✓	✓	✓	9	-
182	MXOR	-	✓	Matrix XOR	-	✓	✓	✓	9	-
183	MXNR	-	✓	Matrix XNR	-	✓	✓	✓	9	-
184	MINV	-	✓	Matrix inverse	-	✓	✓	✓	7	-
185	MCMP	-	✓	Matrix compare	-	✓	✓	✓	9	-
186	MBRD	-	✓	Matrix bit read	-	✓	✓	✓	7	-
187	MBWR	-	✓	Matrix bit write	-	✓	✓	✓	7	-
188	MBS	-	✓	Matrix bit shift	-	✓	✓	✓	7	-
189	MBR	-	✓	Matrix bit rotate	-	✓	✓	✓	7	-
190	MBC	-	✓	Matrix bit state count	-	✓	✓	✓	7	-

Contact Type Logic Operation

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB -PB	ELC -PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC -PV ELC2 -PV	16	32
215	LD&	DLD&	-	S_1 & S_2	-	✓	✓	✓	5	9
216	LD|	DLD|	-	S_1 | S_2	-	✓	✓	✓	5	9
217	LD^	DLD^	-	S_1 ^ S_2	-	✓	✓	✓	5	9
218	AND&	DAND&	-	S_1 & S_2	-	✓	✓	✓	5	9
219	AND|	DAND|	-	S_1 | S_2	-	✓	✓	✓	5	9
220	AND^	DAND^	-	S_1 ^ S_2	-	✓	✓	✓	5	9
221	OR&	DOR&	-	S_1 & S_2	-	✓	✓	✓	5	9
222	OR|	DOR|	-	S_1 | S_2	-	✓	✓	✓	5	9
223	OR^	DOR^	-	S_1 ^ S_2	-	✓	✓	✓	5	9

Contact Type Compare Instruction

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB -PB	ELC -PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC -PV ELC2 -PV	16	32
224	LD=	DLD=	-	S_1 = S_2	✓	✓	✓	✓	5	9

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB -PB	ELC -PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC -PV ELC2 -PV	16	32
225	LD>	DLD>	-	$S_1 > S_2$	✓	✓	✓	✓	5	9
226	LD<	DLD<	-	$S_1 < S_2$	✓	✓	✓	✓	5	9
228	LD<>	DLD<>	-	$S_1 \neq S_2$	✓	✓	✓	✓	5	9
229	LD<=	DLD<=	-	$S_1 \leqq S_2$	✓	✓	✓	✓	5	9
230	LD>=	DLD>=	-	$S_1 \geqq S_2$	✓	✓	✓	✓	5	9
232	AND=	DAND=	-	$S_1 = S_2$	✓	✓	✓	✓	5	9
233	AND>	DAND>	-	$S_1 > S_2$	✓	✓	✓	✓	5	9
234	AND<	DAND<	-	$S_1 < S_2$	✓	✓	✓	✓	5	9
236	AND<>	DAND<>	-	$S_1 \neq S_2$	✓	✓	✓	✓	5	9
237	AND<=	DAND<=	-	$S_1 \leqq S_2$	✓	✓	✓	✓	5	9
238	AND>=	DAND>=	-	$S_1 \geqq S_2$	✓	✓	✓	✓	5	9
240	OR=	DOR=	-	$S_1 = S_2$	✓	✓	✓	✓	5	9
241	OR>	DOR>	-	$S_1 > S_2$	✓	✓	✓	✓	5	9
242	OR<	DOR<	-	$S_1 < S_2$	✓	✓	✓	✓	5	9
244	OR<>	DOR<>	-	$S_1 \neq S_2$	✓	✓	✓	✓	7	13
245	OR<=	DOR<=	-	$S_1 \leqq S_2$	✓	✓	✓	✓	7	-
246	OR>=	DOR>=	-	$S_1 \geqq S_2$	✓	✓	✓	✓	1	-
296	LDZ>	DLDZ>	-	$\lvert S_1 - S_2 \rvert > \lvert S_3 \rvert$	-	-	✓	✓*	7	13
297	LDZ>=	DLDZ>=	-	$\lvert S_1 - S_2 \rvert \geqq \lvert S_3 \rvert$	-	-	✓	✓*	7	13
298	LDZ<	DLDZ<	-	$\lvert S_1 - S_2 \rvert < \lvert S_3 \rvert$	-	-	✓	✓*	7	13
299	LDZ<=	DLDZ<=	-	$\lvert S_1 - S_2 \rvert \leqq \lvert S_3 \rvert$	-	-	✓	✓*	7	13
300	LDZ=	DLDZ=	-	$\lvert S_1 - S_2 \rvert = \lvert S_3 \rvert$	-	-	✓	✓*	7	13
301	LDZ<>	DLDZ<>	-	$\lvert S_1 - S_2 \rvert \neq \lvert S_3 \rvert$	-	-	✓	✓*	7	13
302	ANDZ>	DANDZ>	-	$\lvert S_1 - S_2 \rvert > \lvert S_3 \rvert$	-	-	✓	✓*	7	13
303	ANDZ>=	DANDZ>=	-	$\lvert S_1 - S_2 \rvert \geqq \lvert S_3 \rvert$	-	-	✓	✓*	7	13
304	ANDZ<	DANDZ<	-	$\lvert S_1 - S_2 \rvert < \lvert S_3 \rvert$	-	-	✓	✓*	7	13
305	ANDZ<=	DANDZ<=	-	$\lvert S_1 - S_2 \rvert \leqq \lvert S_3 \rvert$	-	-	✓	✓*	7	13
306	ANDZ=	DANDZ=	-	$\lvert S_1 - S_2 \rvert = \lvert S_3 \rvert$	-	-	✓	✓*	7	13
307	ANDZ<>	DANDZ<>	-	$\lvert S_1 - S_2 \rvert \neq \lvert S_3 \rvert$	-	-	✓	✓*	7	13
308	ORZ>	DORZ>	-	$\lvert S_1 - S_2 \rvert > \lvert S_3 \rvert$	-	-	✓	✓*	7	13
309	ORZ>=	DORZ>=	-	$\lvert S_1 - S_2 \rvert \geqq \lvert S_3 \rvert$	-	-	✓	✓*	7	13
310	ORZ<	DORZ<	-	$\lvert S_1 - S_2 \rvert < \lvert S_3 \rvert$	-	-	✓	✓*	7	13
311	ORZ<=	DORZ<=	-	$\lvert S_1 - S_2 \rvert \leqq \lvert S_3 \rvert$	-	-	✓	✓*	7	13

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB-PB	ELC-PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC-PV ELC2-PV	16	32
312	ORZ=	DORZ=	-	$\|S_1 - S_2\| = \|S_3\|$	-	-	✓	✓*	7	13
313	ORZ<>	DORZ<>	-	$\|S_1 - S_2\| \neq \|S_3\|$	-	-	✓	✓*	7	13

*:ELC-PV does not support API296~API313

Specified Bit Instruction

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB-PB	ELC-PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC-PV ELC2-PV	16	32
266	BOUT	DBOUT	-	Output Specified Bit of a Word	-	-	✓	✓	5	9
267	BSET	DBSET	-	Set ON Specified Bit of a Word	-	-	✓	✓	5	9
268	BRST	DBRST	-	Reset Specified Bit of a Word	-	-	✓	✓	5	9
269	BLD	DBLD	-	Load NO Contact by Specified Bit	-	-	✓	✓	5	9
270	BLDI	DBLDI	-	Load NC Contact by Specified Bit	-	-	✓	✓	5	9
271	BAND	DBAND	-	Connect NO Contact in Series by Specified Bit	-	-	✓	✓	5	9
272	BANI	DBANI	-	Connect NC Contact in Series by Specified Bit	-	-	✓	✓	5	9
273	BOR	DBOR	-	Connect NO Contact in Parallel by Specified Bit	-	-	✓	✓	5	9
274	BORI	DBORI	-	Connect NC Contact in Parallel by Specified Bit	-	-	✓	✓	5	9

Floating-Point Contact Type Comparison

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB-PB	ELC-PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC-PV ELC2-PV	16	32
275	-	FLD=	-	$S_1 = S_2$	-	-	✓	✓	-	9
276	-	FLD>	-	$S_1 > S_2$			✓	✓	-	9
277	-	FLD<	-	$S_1 < S_2$			✓	✓	-	9
278	-	FLD<>	-	$S_1 \neq S_2$			✓	✓	-	9
279	-	FLD<=	-	$S_1 \leqq S_2$			✓	✓	-	9

API	Mnemonic		P	Function	Availability				STEPS	
	16 bits	32 bits			ELCB -PB	ELC -PA	ELCM-PH/PA ELC2-PB/PH ELC2-PA/PE	ELC -PV ELC2 -PV	16	32
280	-	FLD>=	-	$S_1 \geqq S_2$			✓	✓	-	9
280	-	FAND=	-	$S_1 = S_2$			✓	✓	-	9
282	-	FAND>	-	$S_1 > S_2$			✓	✓	-	9
283	-	FAND<	-	$S_1 < S_2$			✓	✓	-	9
284	-	FAND<>	-	$S_1 \neq S_2$			✓	✓	-	9
285	-	FAND<=	-	$S_1 \leqq S_2$			✓	✓	-	9
286	-	FAND>=	-	$S_1 \geqq S_2$			✓	✓	-	9
287	-	FOR=	-	$S_1 = S_2$			✓	✓	-	9
288	-	FOR>	-	$S_1 > S_2$			✓	✓	-	9
289	-	FOR<	-	$S_1 < S_2$			✓	✓	-	9
290	-	FOR<>	-	$S_1 \neq S_2$			✓	✓	-	9
291	-	FOR<=	-	$S_1 \leqq S_2$			✓	✓	-	9
292	-	FOR>=	-	$S_1 \geqq S_2$			✓	✓	-	9

Note: ELCB-PB does not support pulse execution type instructions (P instruction).

3.7 Detailed Instruction Explanation

API	Mnemonic		Operands	Function
00	CJ	P	**S**	Conditional Jump

OP	Range	Program Steps
S	P0~P255	CJ, CJP: 3 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: The destination pointer of the conditional jump, P can be modified by Index register E, F

39. ELCB-PB have 64 pointers; available from the range of P0 to P63.
40. ELC-PA/PV, ELC2-PB/PH/PA/PE/PV and ELCM-PH/PA have 256 pointers; available from the range of P0~P255.

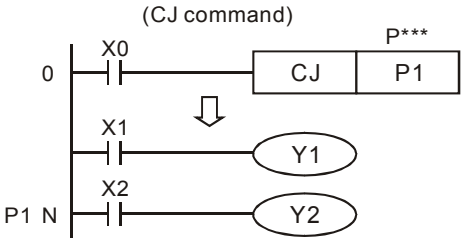
Description:

41. When the CJ instruction is true it forces the program to jump to the specified program marker. While the jump takes place the intervening program steps are skipped. This means they are not processed in any way. The resulting effect is to speed up the program scan time.
42. When the destination of the pointer P is before the CJ instruction, note that this is a program loop and there must be a way out of the loop. If it continually jumps backwards, the Watchdog timer will eventually time out faulting the processor. Loops are generally not recommended in PLC programming. It's always better to allow the program scan to run. The I/O is updated once per scan and if the program is caught in a loop, the I/O is not being scanned and updated.
43. What happens to each data type when the CJ instruction jumps over them :
    1. Y, M, S remains its previous state before the condition jump occurs.
    2. General timers, accumulative timers and general counters will freeze their current values if they are skipped by a CJ instruction.
    3. Timers for Subroutines and Interrupts are an exception to this as they are processed independently of the main program.
    4. High speed counters are also an exception to this as they are processed independently of the main program.
    5. The ordinary counters stop executing.
    6. If the "reset instruction" of a timer is executed before the conditional jump, the device will still be in the reset status while jump is being executed.
    7. Application instructions are also skipped if they are programmed between the CJ instruction and the destination pointer. However, the DHSCS, DHSCR, DHSZ, SPD, PLSY, PWM, PLSR, PLSV, DDRVI and DDRVA instructions will operate continuously if they were active before the CJ instruction was executed, otherwise they will not be processed. .

Program Example 1:

When X0=ON the program will skip from address 0 to N (label P1) automatically and continue executing. Logic between 0 and N will be skipped and will not be executed.

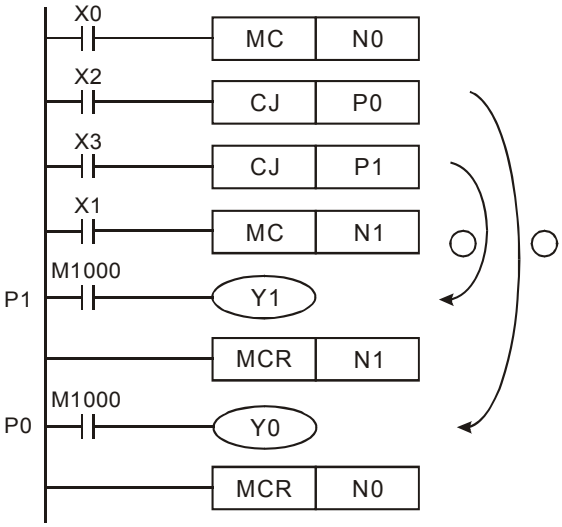
When X0=OFF, all lines of code will be executed.



Program Example 2:

There are five conditions where the CJ instruction can be used between the MC and MCR instructions.

1. Outside of an MC~MCR.
2. Valid in the Loop P1 shown below
3. In the same level N, inside of MC~MCR.
4. Inside of MC, out of MCR.
5. Jump from one MC~MCR to another MC~MCR.



API	Mnemonic		Operands	Function
01	CALL	P	**S**	Call Subroutine

OP	Range	Program Steps
S	P0~P255	CALL, CALLP: 3 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: The destination pointer of the call subroutine. P can be modified by Index registers E, F.

44. ELCB-PB have 64 pointers; available from the range of P0 to P63.
45. ELC-PA/PV, ELC2-PB/PH/PA/PE/PV and ELCM-PH/PA have 256 pointers; available from the range of P0~P255.

Description:

46. When the CALL instruction is active it forces the program to run the subroutine associated with the called pointer.
47. A CALL instruction must be used in conjunction with FEND (API 06) and SRET (API 02) instructions.
48. The program jumps to the subroutine pointer (located after the FEND instruction) and processes the contents until an SRET instruction is encountered. This forces the program flow back to the line of ladder immediately following the original CALL instruction.

Points to note:

49. Subroutines must be placed after the FEND instruction.
50. Subroutines must end with the SRET instruction.
51. CALL and CJ instruction pointers are not allowed to use the same pointer number.
52. CALL instructions can call any subroutine any number of times.
53. Subroutines can be nested 5 levels deep including the initial CALL instruction.

API	Mnemonic	Function
02	SRET	Subroutine Return

OP	Range	Program Steps
N/A	Automatically returns to the step immediately following the CALL instruction which activated the subroutine	SRET: 1 steps

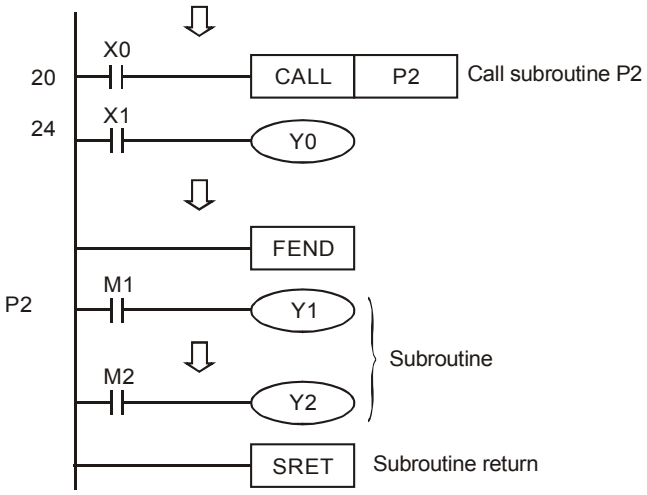
ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Description:

Indicates the end of a subroutine program. The subroutine will return to the main program and begin execution with the instruction after the CALL instruction.

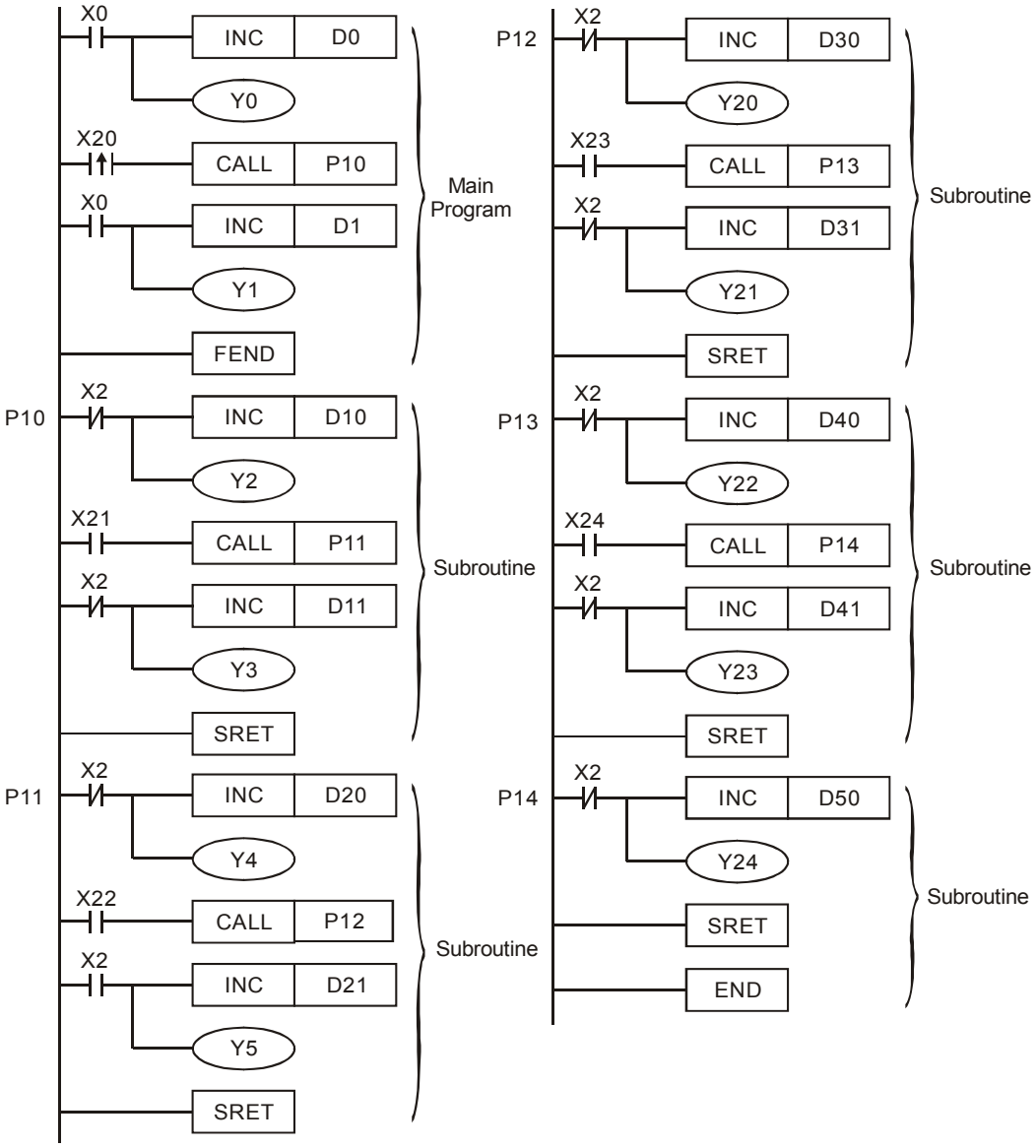
Program Example 1:

When X0 = ON, the CALL instruction will jump to P2 and run the subroutine. With the execution of the SRET instruction, it will jump back to step 24 and continue execution.



Program Example 2:

54. When the rising-edge of X20 is triggered, the CALL P10 instruction will transfer execution to subroutine P10.
55. When X21 is ON, execute CALL P11, jump to and run subroutine P11.
56. When X22 is ON, execute CALL P12, jump to and run subroutine P12.
57. When X23 is ON, execute CALL P13, jump to and run subroutine P13.
58. When X24 is ON, execute CALL P14, jump to and run subroutine P14. When the SRET instruction is reached, jump back to the last P*** subroutine and keep executing until the last SRET instruction is reached which will return execution back to the main program.



API	Mnemonic	Function
03	IRET	Interrupt Return

OP	Range	Program Steps
N/A	IRET ends the processing of an interrupt subroutine and returns execution back to the main program	IRET: 1 step

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

API	Mnemonic	Function
04	EI	Enable All Interrupts

OP	Range	Program Steps
N/A	Enables Interrupts. This instruction is used with the DI (disable interrupts) instruction. See the DI instruction for more information. If users want to enable a certain interrupt, they can set the corresponding special M.	EI: 1 step

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

API	Mnemonic	Function
05	DI	Disable All Interrupts

OP	Range	Program Steps
N/A	DI instruction disables the ELC to accept all interrupts; like Time interrupts or High-speed counter interrupts or External interrupts. If users want to disable a certain interrupt, they can set the corresponding special M.	DI: 1 step

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Description:

59. Interrupt subroutines must be placed after the FEND instruction.
60. Other interrupts are not allowed during execution of a current interrupt routine.
61. Priority is given to the interrupt occurring first. If interrupts occur simultaneously, the interrupt with the lower pointer number will be given the higher priority.
62. Any interrupt request occurring between DI and EI instructions will not be executed immediately. The interrupt will be saved and executed when the next EI instruction occurs.

63. Care should be used when using external interrupts and using the same inputs for high speed counter inputs.
64. During the execution of an interrupt routine, an immediate I/O instruction can be performed by using the REF instruction.

Points to note:

65. ELCB-PB models interrupt pointers (I):
    1. External interrupts: (I001, X0), (I101, X1), (I201, X2), (I301, X3) 4 points.
    2. Time interrupts: I6□□, 1 point (□□＝10~99, time base=1ms)
    3. Communication interrupt for specific characters received (I150)
    4. Flags:

Flag	Function
M1050	External interrupt, I 001 masked
M1051	External interrupt, I 101 masked
M1052	External interrupt, I 201 masked
M1053	External interrupt, I 301 masked
M1056	Disable time interrupt I6□□

66. FOR ELC-PA models Interrupt pointers (I):
    1. External interrupts: (I001, X0), (I101, X1), (I201, X2), (I301, X3), (I401, X4), (I501, X5) 6 points.
    2. Time interrupts: I6□□, I7□□ 2 points. (□□＝1~99ms, time base=1ms)
    3. High-speed counter interrupts: I010, I020, I030, I040, I050, I060 6 points. (used with DHSCS instruction)
    4. Communication interrupt for specific characters received (I150)
    5. The priority of interrupt pointer I: high-speed counter interrupt, external interrupt, time interrupt and communication interrupt for specific characters received
    6. Among the following 6 interrupts, (I001, I010), (I101, I020), (I201, I030), (I301, I040), (I401, I050), (I501, I060), the program allows the user to use only one of the two numbers in a pair. If the user uses the two numbers in the pair, syntax check errors may occur when the program is written to the ELC controller.
    7. Flags:

Flag	Function
M1050	External interrupt, I 001 masked
M1051	External interrupt, I 101 masked
M1052	External interrupt, I 201 masked
M1053	External interrupt, I 301 masked
M1054	External interrupt, I 401 masked
M1055	External interrupt, I 501 masked

Flag	Function
M1056	Timer interrupt, I6□□ masked
M1057	Timer interrupt, I7□□ masked
M1059	High-speed counter interrupt, I010~I060 masked
M1299	Communication interrupt, I150 masked

67. FOR ELC-PV, ELC-PV2 models Interrupt pointers (I):

a) External interrupts: (I00□, X0), (I10□, X1), (I20□, X2), (I30□, X3), (I40□, X4), (I50□, X5) 6 points. (□ = 0 designates interrupt on falling-edge, □ = 1 designates interrupt on rising-edge)

b) Time interrupts: I601~I699, I701~I799, 2 points. (Timer resolution: 1ms), I801~I899 1 point. (Timer resolution: 0.1ms)

c) High-speed counter interrupts: I010, I020, I030, I040, 1050, 1060 6 points. (used with API 53 DHSCS instruction to generate interrupt signals)

d) When pulse output interrupts I110, I120 (triggered when pulse output is finished), I130, I140 (triggered when the first pulse output starts) are executed, the currently executed program is interrupted and jumps to the designated interrupt subroutine.

e) Communication interrupt: I150, I160, I170

f) The order for execution of interrupt pointer I: external interrupt, time interrupt, high-speed counter interrupt, pulse interrupt, communication interrupt.

g) External interruptions only for ELC2-PV: (I90□, X10), (I91□, X11), (I92□, X12), (I93□, X13), (I94□, X14), (I95□, X15), (I96□, X16), (I97□, X17) 8 points. (□ = 0 designates interruption in falling-edge, □ = 1 designates interruption in rising-edge

h) Flags:

Flag	Function
M1280	Disable external interrupt I00□
M1281	Disable external interrupt I10□
M1282	Disable external interrupt I20□
M1283	Disable external interrupt I30□
M1284	Disable external interrupt I40□
M1285	Disable external interrupt I50□
M1286	Disable time interrupt I6□□
M1287	Disable time interrupt I7□□
M1288	Disable time interrupt I8□□
M1289	Disable high-speed counter interrupt I010
M1290	Disable high-speed counter interrupt I020
M1291	Disable high-speed counter interrupt I030
M1292	Disable high-speed counter interrupt I040

Flag	Function
M1293	Disable high-speed counter interrupt I050
M1294	Disable high-speed counter interrupt I060
M1295	Disable pulse output interrupt I110
M1296	Disable pulse output interrupt I120
M1297	Disable pulse output interrupt I130
M1298	Disable pulse output interrupt I140
M1299	Disable communication interrupt I150
M1300	Disable communication interrupt I160
M1301	Disable communication interrupt I170
M1340	Generate interrupt I110 after CH0 pulse is sent
M1341	Generate interrupt I120 after CH1 pulse is sent
M1342	Generate interrupt I130 when CH0 pulse is being sent
M1343	Generate interrupt I140 when CH1 pulse is being sent
M1560 ~ M1567	Disable external interrupt I90□ ~ I97□, only for ELC2-PV

68. FOR ELCM-PH/PA, ELC2-PB/PH/PA/PE models Interrupt pointers (I):

a) External interrupts: (I00□, X0), (I10□, X1), (I20□, X2), (I30□, X3), (I40□, X4), (I50□, X5), (I60□, X6), (I70□, X7) 8 points. (□ = 0 designates interrupt on falling-edge, □ = 1 designates interrupt on rising-edge)

b) Time interrupts: I602~I699, I702~I799, 2 points. (Timer resolution: 1ms)

c) High-speed counter interrupts: I010, I020, I030, I040, 1050, 1060, 1070, 1080 8 points. (used with API 53 DHSCS instruction to generate interrupt signals)

d) Communication interrupt: I140, I150, I160

e) The order for execution of interrupt pointer I: external interrupt, time interrupt, high-speed counter interrupt, communication interrupt.

f) Flags:

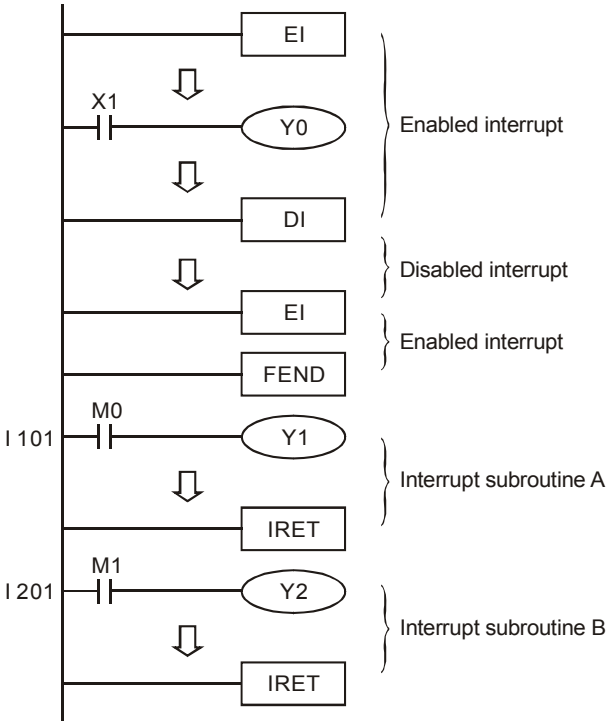
Flag	Function
M1050	Disable external interrupt I000 / I001
M1051	Disable external interrupt I100 / I101
M1052	Disable external interrupt I200 / I201
M1053	Disable external interrupt I300 / I301
M1054	Disable external interrupt I400 / I401
M1055	Disable external interrupt I500 / I501, I600 / I601, I700 / I701
M1056	Disable timer interrupts I602~I699

Flag	Function
M1057	Disable timer interrupts I702~I799
M1059	Disable high-speed counter interrupts I010~I080
M1280	I000/I001 Reverse interrupt trigger pulse direction (Rising/Falling)
M1284	I400/I401 Reverse interrupt trigger pulse direction (Rising/Falling)
M1286	I600/I601 Reverse interrupt trigger pulse direction (Rising/Falling)

Note: The default setting of I000 is falling-edge triggered. When M1280 = ON and EI instruction is enabled, ELC will reverse the interrupt at X0 as rising-edge triggered. If the interrupt needs to trigger on falling-edge, M1280 must be reset (OFF) first and then the DI instruction must be enabled. Then the interrupt will be reset on the falling-edge when EI is executed again.

Program Example:

During the ELC operation, the program scans the instructions between EI and DI, if X1 or X2 are ON, subroutine A or B will be executed. When IRET is reached, the main program will resume.



API	Mnemonic	Function
06	FEND	Terminate the Main Routine Program

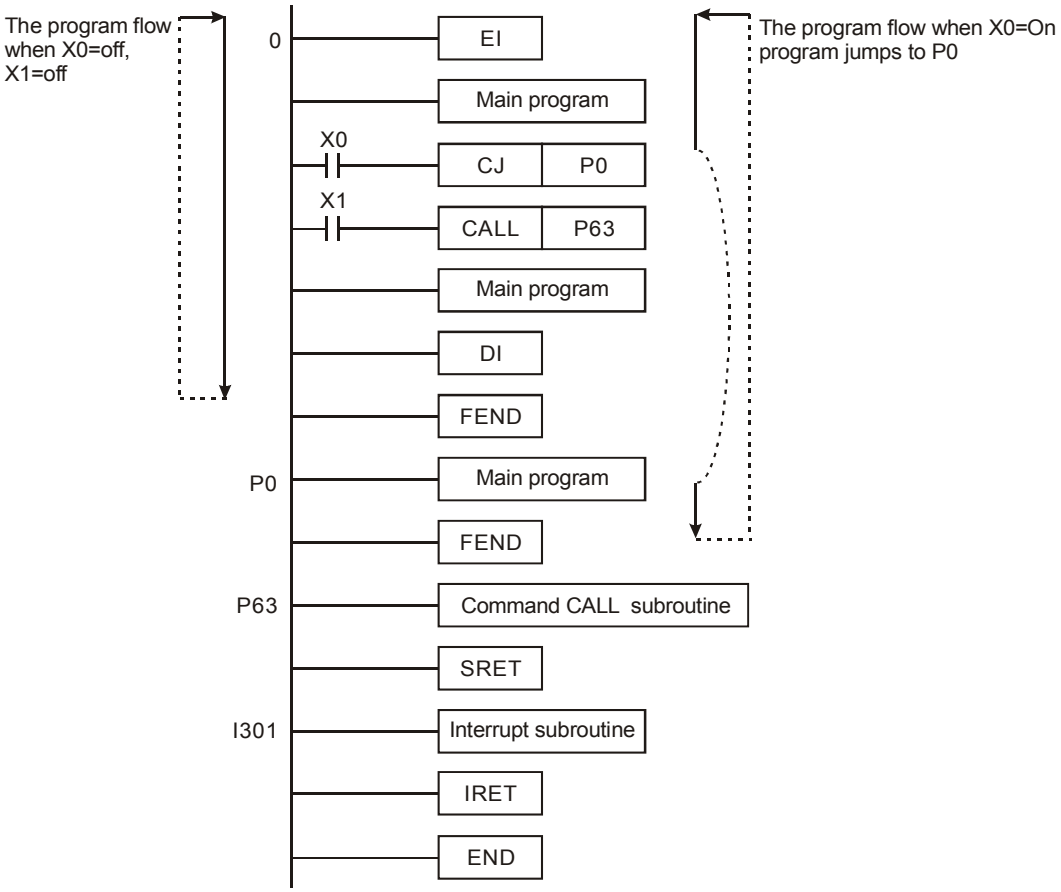
OP	Range	Program Steps
N/A	Instruction driven by contact is not necessary.	FEND: 1 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

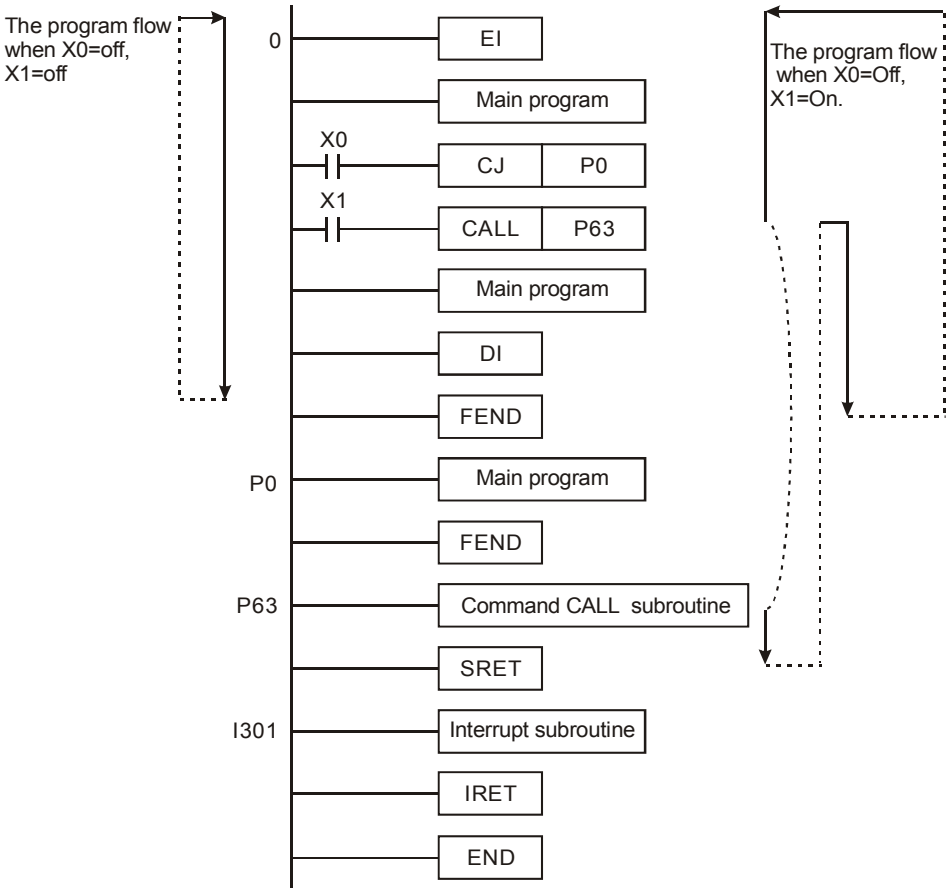
Description:

69. Use the FEND instruction when the application uses either standard subroutines or interrupt routines. If subroutines are not used then use the END instruction to end the main program.
70. This instruction denotes the end of the main program when subroutines are used. It has the same function as the END instruction during ELC operation.
71. Subroutines must be placed after the FEND instruction. Each subroutine must end with the SRET instruction.
72. Interrupt subroutines must be placed after the FEND instruction. Each interrupt subroutine must end with the IRET instruction.
73. When using the FEND instruction, an END instruction is still required, but should be placed as the last instruction after the main program and all subroutines.
74. If using several FEND instructions, place the subroutines between the FEND and END instructions.
75. During execution of a subroutine, if a FEND instruction is scanned before the SRET instruction, an error will occur.
76. During execution of a FOR instruction, if a FEND instruction is scanned before the NEXT instruction, an error will occur.

CJ Command Program Flow



CALL Command Program Flow



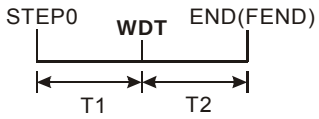
API	Mnemonic		Function
07	WDT	P	Reset the Watchdog Timer

OP	Range	Program Steps
N/A		WDT, WDTP: 1 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Description:

1. The WDT instruction can be used to reset the Watch Dog Timer. If the ELC scan time (from step 0 to END or FEND instruction) is more than 200ms, the ERROR LED will flash. The user will have to turn the ELC OFF and then back ON to clear the fault. The ELC will determine the status of RUN/STOP according to RUN/STOP switch when power is restored.
2. When to use WDT:
   1. When an error occurs in the ELC.
   2. When the scan time of the program exceeds the WDT value in D1000. It can be modified by using the following two methods.
      i. Use WDT instruction (see the program example below for more information)



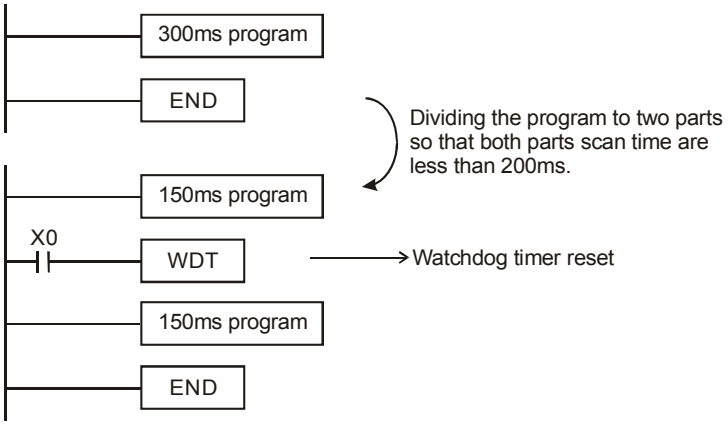
      ii. Use D1000 (default is 200ms) to change the watchdog time.

Points to note:

77. When the WDT instruction is used it will operate on every program scan so long as its input conditions are true. To force the WDT instruction to operate for only ONE scan, use the suffix P with the WDT instruction (WDTP).
78. The watchdog timer has a default setting of 200ms for ELC controllers. This time limit may be modified by moving another value into the contents of data register D1000, the wathdog timer register.

Program Example:

If the program scan time is over 300ms, users can divide the program into 2 parts. Insert the WDT instruction in the middle of the program, so both halves of the program's scan time will be less than 200ms.



API	Mnemonic	Operands	Function
08	FOR	**S**	Loop Begin

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	FOR: 3 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: The number of times the loop will be executed

API	Mnemonic	Function
09	NEXT	Loop End

OP	Range	Program Steps
N/A		NEXT: 1 steps

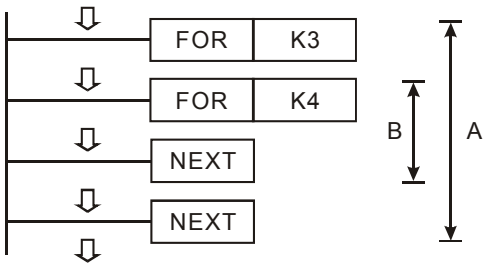
ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Description:

1. The FOR and NEXT instructions are used when FOR/NEXT loops are needed.
2. “N” (the number of times the loop is executed – this is the operand S for the FOR instruction) must be within the range of K1 to K32767. If the range N≦K1, N will always be K1.
3. An error will occur in the following conditions:
   - The NEXT instruction is before the FOR instruction.
   - A FOR instruction doesn’t have a NEXT instruction.
   - There is a NEXT instruction after the FEND or END instruction.
   - A different number of FOR and NEXT instructions.
4. The FOR to NEXT loop can be nested to five levels. If the execution time of the loops is too long, the ELC scan time will increase and it may cause the watchdog timer to be activated and result in an error. The WDT instruction can be used to prevent watchdog faults..

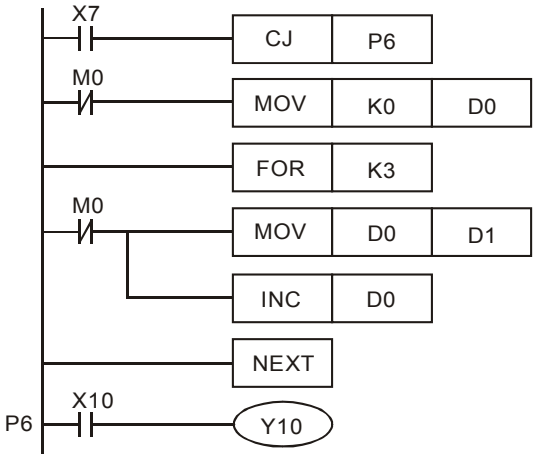
Program Example 1:

After loop A operates 3 times, the program after the last NEXT instruction will be scanned. For every complete cycle of loop A, loop B will execute 4 times. Therefore, the total number of times that loop B operates will be 3 × 4＝12 times.



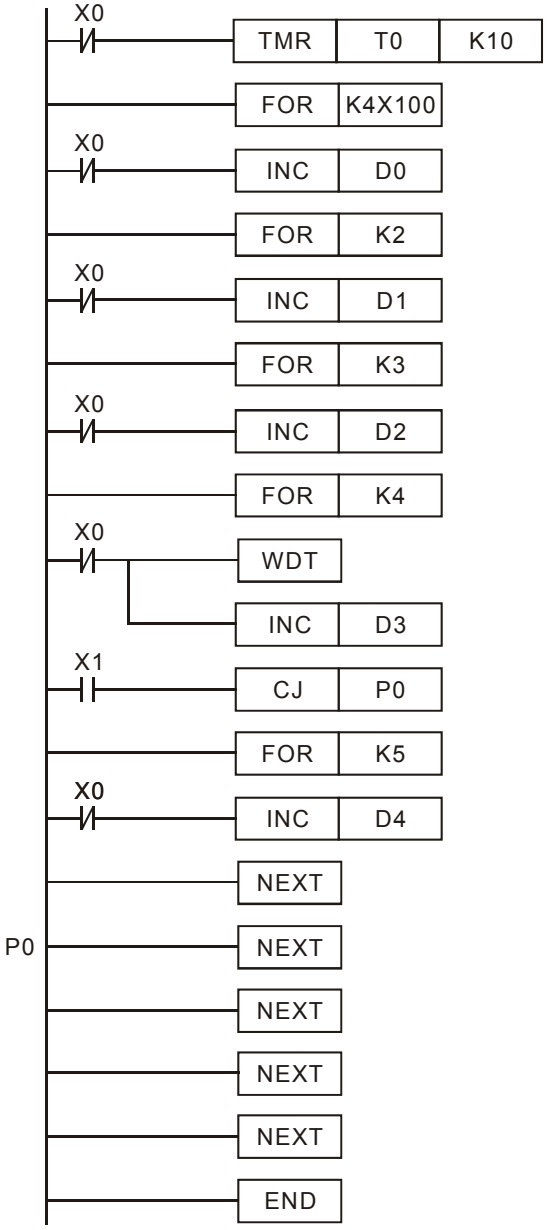
Program Example 2:

When X7 = Off, the ELC will execute the program between the FOR ~ NEXT loop. When X7 = On, the CJ instruction jumps to P6 and does not execute the logic between the CJ and P6, which includes the FOR/NEXT loop.



Program Example 3:

When the FOR / NEXT instructions are not to be executed, a CJ instruction can be used to jump around the loop. When X1=ON, the CJ instruction will jump to P0 and not execute the inner most FOR / NEXT loop.



API	Mnemonic			Operands	Function
10	D	CMP	P	S_1, S_2, D	Compare

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	CMP, CMPP: 7 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	DCMP, DCMPP: 13 steps
D		*	*	*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

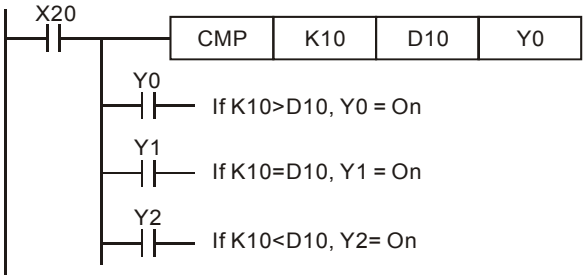
S_1: First comparison value S_2: Second comparison value **D**: Comparison result (starting bit address – uses 3 consecutive bit addresses)

Description:

79. The contents of S_1 and S_2 are compared and **D** denotes the compare result.
80. Operand **D** occupies 3 continuous bits (Y, M or S).
81. The values are binary values. If bit 15=1 in 16-bit instruction or bit 31=1 in 32-bit instruction, the comparison will regard the value as a negative binary value.
82. **D**, **D** +1, **D** +2 hold the comparison results,
    D = ON if $S_1 > S_2$,
    D +1 = ON if $S_1 = S_2$
    D +2 = ON if $S_1 < S_2$
83. If operand S_1, S_2 use index register F, only a 16 bit compare is available.

Program Example:

84. If **D** is set to Y0, then Y0, Y1, Y2 will display the results of the compare as shown below.
85. When X20=ON, the CMP instruction is executed and one of Y0, Y1, Y2 will be ON. When X20=OFF, the CMP instruction is not executed and Y0, Y1, Y2 remain in their previous states.



86. Use RST or ZRST instructions to reset the comparison result.

API	Mnemonic			Operands	Function
11	D	ZCP	P	**S1, S2, S, D**	Zone Compare

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	ZCP, ZCPP: 9 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	DZCP, DZCPP: 17 steps
S					*	*	*	*	*	*	*	*	*	*	*	
D		*	*	*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: First comparison value (Minimum) **S_2**: Second comparison value (Maximum)

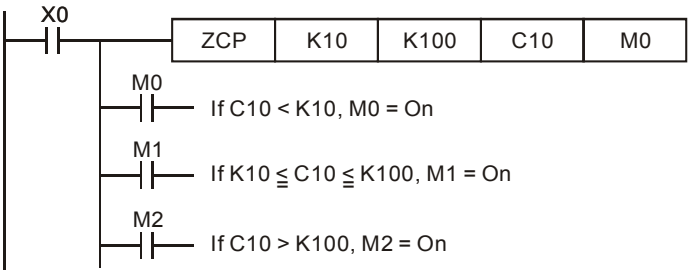
S: Comparison value **D**: Comparison result result (starting bit address – uses 3 consecutive bit addresses)

Description:

87. **S** is compared with the lower limit, **S_1** and the upper limit **S_2** and **D** contains the compare result.
88. The values are binary values. If bit 15=1 in 16-bit instruction or bit 31=1 in 32-bit instruction, the comparison will regard the value as a negative binary value.
89. If operand **S_1**, **S_2** , **S** use index register F, only a 16 bit compare is available.
90. Operand **S_1** should be less than Operand **S_2**,
    Operand **D** occupies 3 continuous addresses.

Program Example:

91. If **D** is set to M0, then M0, M1, M2 will display the result of the ZCP as shown below.
92. When X0=ON, ZCP instruction is evaluated and one of M0, M1 or M2 will be ON. When X0=OFF, ZCP instruction is not evaluated and M0, M1, M2 remain in the previous status.



93. Use RST or ZRST instructions to reset the comparison result.

API	Mnemonic			Operands	Function
12	D	MOV	P	S, D	Move

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	MOV, MOVP: 5 steps
D								*	*	*	*	*	*	*	*	DMOV, DMOVP: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

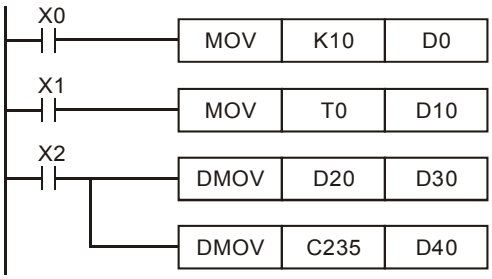
S: Data source **D**: Data destination

Description:

When the MOV instruction is executed, the data in **S** is moved to **D** without any change to **S**. If the MOV instruction is not executed, the content of **D** will remain unchanged.

Program Example:

94. MOV will move a 16-bit value from the source location to the destination.
95. When X0=OFF, the content of D0 remains unchanged. If X0=ON, the data K10 is moved to D0.
96. When X1=OFF, the content of D10 remain unchanged. If X1=ON, the data of T0 is moved to the D10 data register. In a word instruction, T0 is the accumulated value of timer T0.
97. DMOV will move a 32-bit value from the source location to the destination.
98. When X2=OFF, the content of (D31, D30) and (D41, D40) remain unchanged. If X2=ON, the data of (D21, D20) is moved to (D31, D30) data register. Meanwhile, the data of C235 is moved to (D41, D40) data register.



API	Mnemonic		Operands	Function
13	SMOV	P	**S, m1, m2, S2, D, n**	Shift Move

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*	*	*	SMOV, SMOVP: 11 steps
m_1					*	*										
m_2					*	*										
D								*	*	*	*	*	*	*	*	
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Data source **m_1**: Source position (nibble) of the first digit to be moved **m_2**: Number of source digits (nibbles) to be moved **D**: Destination **n**: Destination position for the first digit (nibble)

Description:

99. BCD mode (M1168=OFF):

    This mode of the SMOV operation allows BCD numbers to be manipulated in exactly the same way as the 'normal' SMOV manipulates decimal numbers, i.e. this instruction copies a specified number of digits from a 4 digit BCD source(**S**) and places them at a specified location within a destination (**D**) address (also a 4 digit BCD number).

100. BIN mode (M1168=ON):

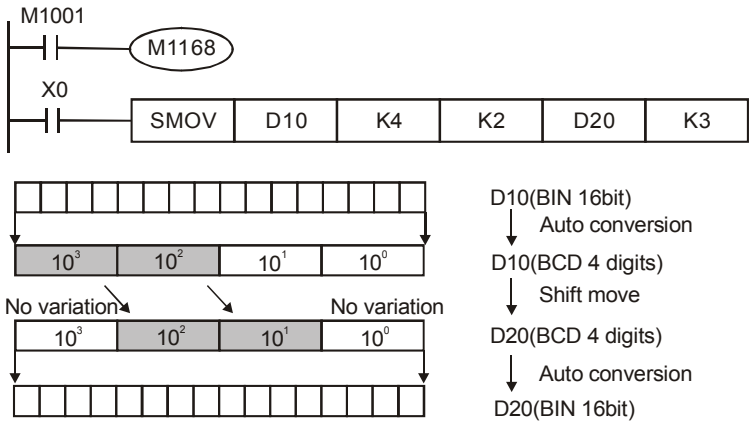
    This instruction copies a specified number of digits from a 4 digit decimal source (**S**) and places them at a specified location within a destination (**D**) address (also a 4 digit decimal). The existing data in the destination is overwritten.

Points to note:

101. The range of **m_1**: 1 ~ 4
102. The range of **m_2**: 1 ~ **m_1** (cannot be great than **m_1**)
103. The range of **n**: **m_2** ~ 4 (cannot be less than **m_2**)

Program Example 1:

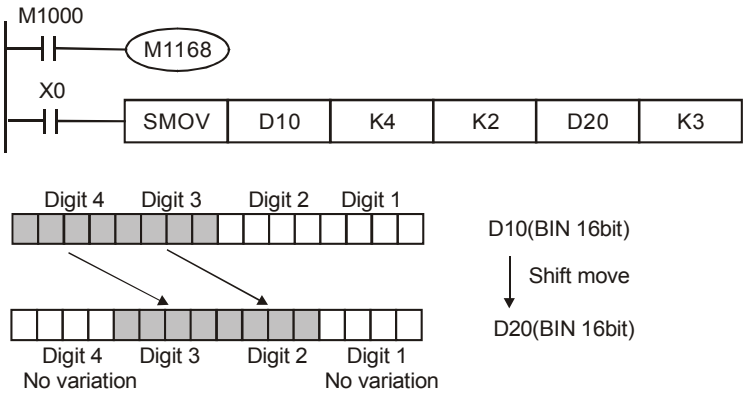
104. When M1168=OFF and X0=ON, the two upper digits of D10 are moved to the two middle digits of D20. The contents of the high and low digits of D20 remain unchanged after SMOV is executed.
105. If the source is not a valid BCD number an operation error will occur in ELC. The instruction will not be executed and M1067 and M1068 = ON, D1067 = error code H0E18.



106. If D10=H1234, D20=H5678 before execution, D10 remains unchanged and D20=H5128 after execution.

Program Example 2:

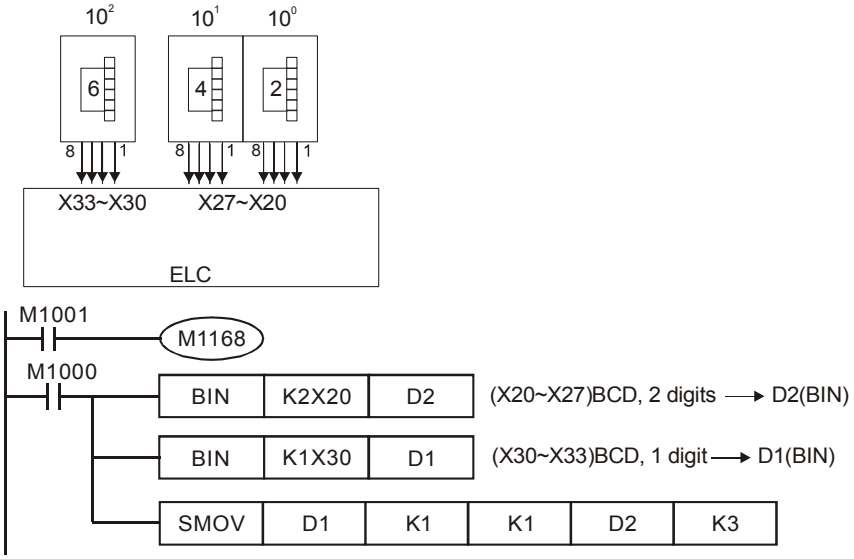
When M1168=ON and X0=ON, SMOV is executed, the two high digits of D10 will be moved to the middle two digits of D20 in hex format.



Program Example 3:

Move the first digit of D1 to the third digit of D2 after the low byte of D2 is populated with X20-X27 and the low byte of D1 is populated with X30-X37.

Use SMOV to move the first digit of D1 to the third digit of D2 and combining these two digit switches into one word (D2).



API	Mnemonic			Operands	Function
14	D	CML	P	**S, D**	Compliment and Move

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	CML, CMLP: 5 steps
D								*	*	*	*	*	*	*	*	DCML, DCMLP: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

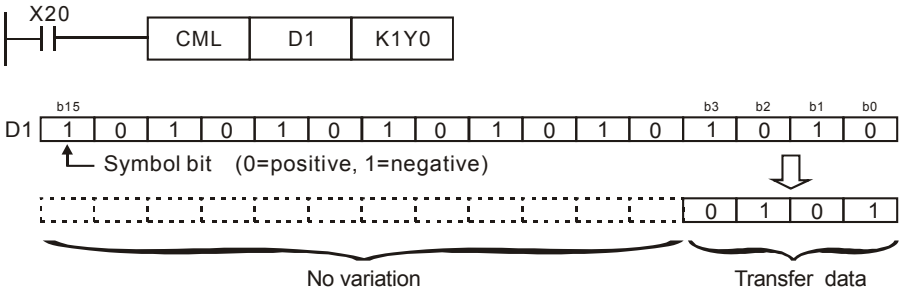
S: Data source **D**: Destination

Description:

107. Take the data in the source **S**, compliment (0→1, 1→0) it and move to the assigned destination **D**.
108. If operand **S** and **D** use index register F, only a 16 bit value is available

Program Example:

When X20=ON, the contents of D1, b0~b3, will be complimented and moved to Y0 – Y3.



API	Mnemonic		Operands	Function
15	BMOV	P	**S, D, n**	Block Move

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			BMOV, BMOVP: 7 steps
D								*	*	*	*	*	*			
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

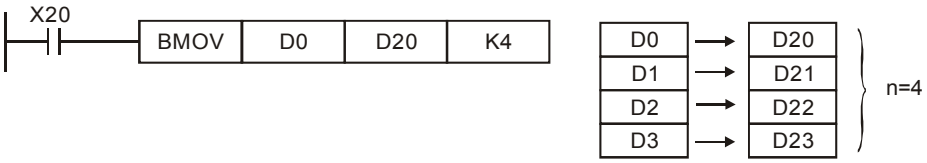
S: Source **D**: Destination **n**: Number of data registers to move

Description:

109. This instruction is used to move an assigned block of data to a new destination. Move the contents of **S** through **S** + **n** to **D** through **D** + **n** registers. If the number of registers **n** exceeds the valid range shown below, only the values that are within the valid range will be moved.
110. The range of **n**=1 – 512.
111. ELCB-PB models do not support KnX, KnY, KnM, KnS addresses.

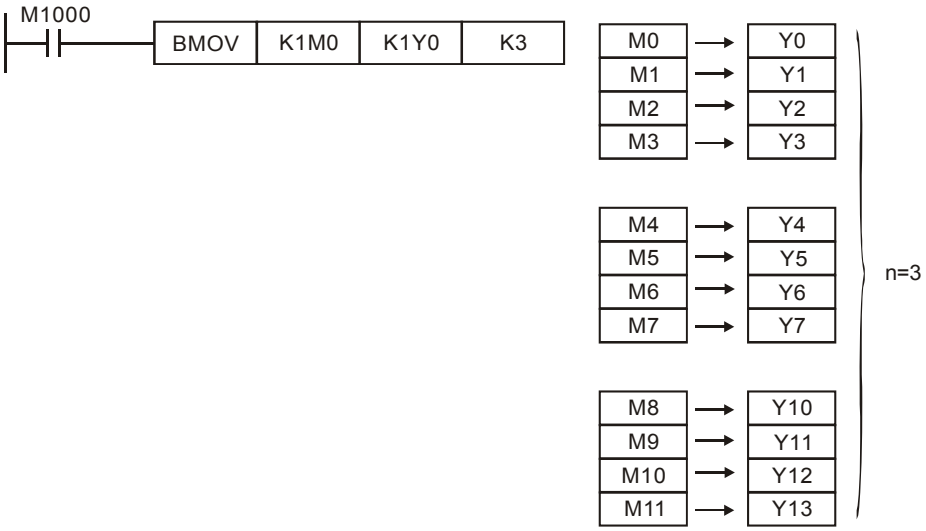
Program Example 1:

When X20=ON, move the contents of the four registers D0~D3 to their corresponding registers D20~D23.



Program Example 2:

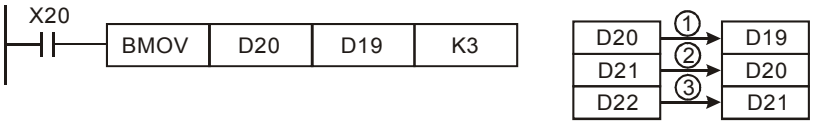
If BMOV is used to move bits, KnX, KnY, KnM, KnS, the digit numbers of **S** and **D** should be of the same data type.



Program Example 3:

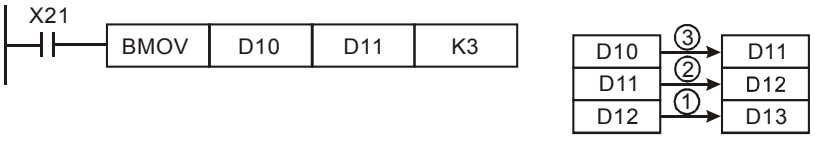
The BMOV instruction will operate differently, depending on the addresses assigned to **S** and **D** as follows.

112. When **S** > **D**, the BMOV instruction is processed in the order ①→②→③.



113. In ELC-PV, when **S** < **D**, the instruction is processed following the order ①→②→③

114. In ELC-PB/PC/PA/PH and ELCM-PH/PA, when **S** < **D**, the BMOV instruction is processed in the order ③→②→①, then D11~D13 all equal to D10.



API	Mnemonic			Operands	Function
16	D	FMOV	P	**S, D, n**	Fill and Move

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	FMOV, FMOVP: 7 steps DFMOV, DFMOVP: 13 steps
D								*	*	*	*	*	*			
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

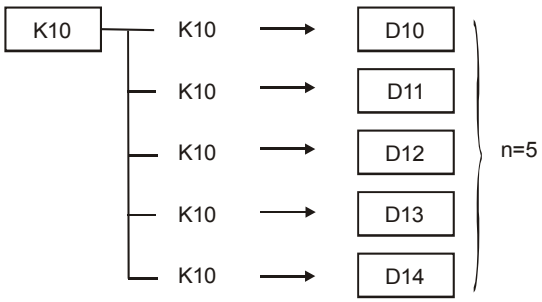
S: Source **D**: Destination **n**: Number of data registers to move

Description:

115. This instruction is used to move a value to a block of values. . Move the contents of **S** to each **D** through **D** + **n** registers. If the number of registers **n** exceeds the valid range shown below, only the values that are within the valid range will be moved. For example, this instruction can be used to clear a file or assign a single value to a file of registers.
116. ELCB-PB models do not support KnX, KnY, KnM, and KnS devices.
117. If operand **S** uses index register F, only 16 bit values are available
118. The range of n: 1~ 512(16-bit instruction), 1~ 256 (32-bit instruction)

Program Example:

When X20=ON, move constant K10 to the consecutive five registers (D10~D14) starting from D10.



API	Mnemonic			Operands	Function
17	D	XCH	P	D1, D2	Data Exchange

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D_1								*	*	*	*	*	*	*	*	XCH, XCHP: 5 steps
D_2								*	*	*	*	*	*	*	*	DXCH, DXCHP: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

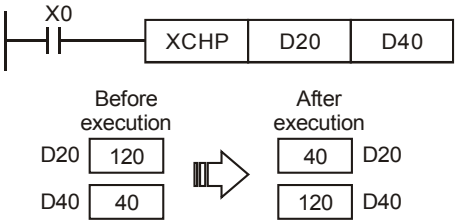
D_1: First exchange device **D_2**: Second exchange device

Description:

119. Exchange the contents of **D_1** and **D_2**.
120. This instruction is best used as a pulse execution (XCHP) to avoid assigning the same values back and forth.
121. If operand **D1** and **D2** use index register F, only a 16-bit value is available.

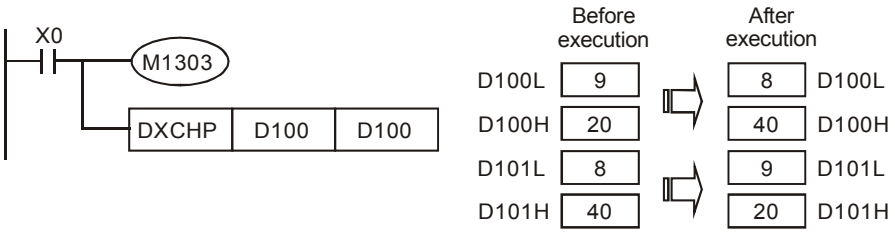
Program Example:

When X0=OFF→ON, the contents of D20 and D40 are exchanged.



Points to note:

122. When **D_1** and **D_2** are the same, and M1303=ON, the upper and lower 16-bits will be exchanged. ELCB-PB does not support this.
123. When X0=ON and M1303=ON, the upper and lower 16-bit contents of D100, D101 will exchange.



API	Mnemonic			Operands	Function
18	D	BCD	P	**S, D**	Convert BIN to BCD

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*	*	*	BCD, BCDP: 5 steps
D								*	*	*	*	*	*	*	*	DBCD, DBCDP: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Source **D**: Converted result

Description:

124. Convert BIN data (0 to 9999) of **S** into BCD and transfer the result to **D**.
125. If the BCD conversion result is outside the valid range of 0 to 9999 (16-bit) or 0 to 99,999,999 (32-bit), an operation error occurs, the error flag M1067 and M1068 =ON, and D1067 will hold error code H0E18.
126. If operand **S** and **D** use index register F, only 16-bit values are available.
127. Flags: M1067 (operation error), M1068 (operation error), D1067 (error code)

Program Example:

128. When X0=ON, the binary data D10 is converted into BCD number, and stored at K1Y0 (Y0~Y3).

129. When D10=001E (Hex) =0030(decimal), the result will be Y0~Y3=0000(BIN).

API	Mnemonic			Operands	Function
19	D	BIN	P	S, D	Convert BCD to BIN

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*	*	*	BIN, BINP: 5 steps
D								*	*	*	*	*	*	*	*	DBIN, DBINP: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Source **D**: Converted result

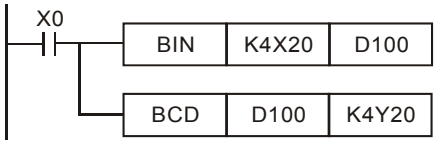
Description:

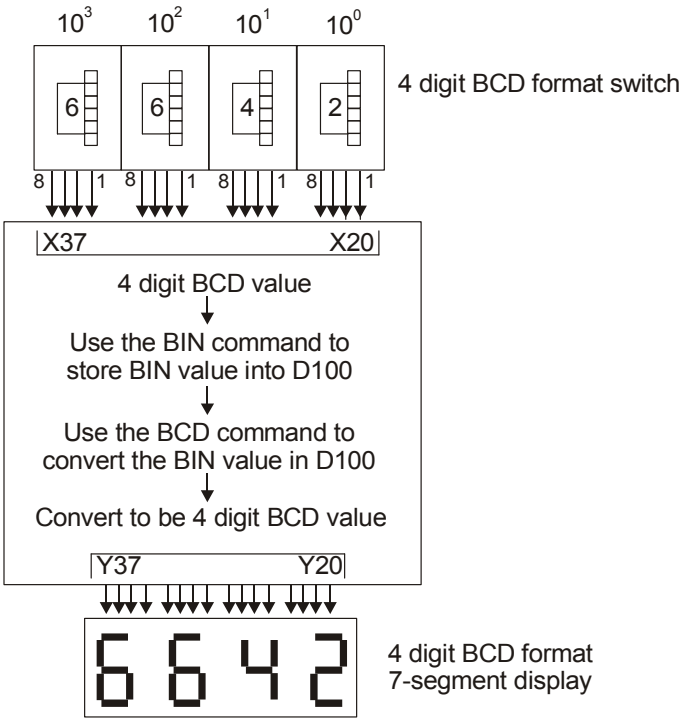
130. Converts BCD data (0 to 9,999) of **S** into BIN and transfer the result to **D**.
131. The valid range of source **S**: BCD (0 to 9,999), DBCD (0 to 99,999,999)
132. If the content of **S** is not a valid BCD value, an operation error will occur, error flags M1067 and M1068 =ON, and D1067 holds error code H0E18.
133. If operand S and D use index register F, only a 16-bit compare is available.
134. Flags: M1067 (operation error), M1068 (operation error), D1067 (error code)

Program Example:

When X0=ON, the BCD data K1X20 is converted to BIN data, and result stored at D10.

135. The BIN instruction is used to covert the source data into BIN data. An example could be when the ELC reads a BCD value from a thumbwheel switch and this BCD value needs to be converted to BIN.
136. When X0=ON, convert K4X20 (BCD data) into BIN data and move it to D100. Then, convert BIN data of D100 into BCD data and move it to K4Y20.





API	Mnemonic			Operands	Function
20	D	ADD	P	**S_1, S_2, D**	Addition

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	ADD, ADDP: 7 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	DADD, DADDP: 13 steps
D								*	*	*	*	*	*	*	*	

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Augend **S_2**: Addend **D**: Addition result

Description:

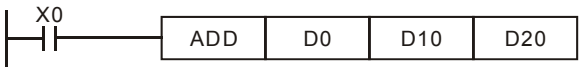
1. The data contained within the source devices (**S_1**, **S_2**) are added together and the result is stored at the specified destination device (**D**).
2. The most significant bit is the sign bit. 0 indicates positive and 1 indicates negative. All calculation are algebraically processed, i.e. 3 + (-9) = -6.
3. If operands **S_1**, **S_2**, **D** use index register F, then only 16-bit instruction is available.
4. Flags: M1020 (Zero flag), M1021 (Borrow flag), M1022 (Carry flag)

Program Example 1:

16-bit instruction:

When X0 = ON, the data in D0 and data in D10 are added together and the result stored in D20. D0

and D10 are unchanged.



(D0) + (D10) = (D20)

Program Example 2:

32-bit instruction:

When X0 = ON, the data in (D31, D30) and data in (D41, D40) are added together and the result stored in (D51, D50). (D31, D30) and (D41, D40) are unchanged. (D30, D40, D50 is the lower 16-bit data, while D31, D41, D51 is the higher 16-bit data)

(D31, D30) + (D41, D40) = (D51, D50)

Flag operations:

16-bit instruction:

137. If the operation result is “0”, then the Zero flag, M1020 is set to ON.

138. If the operation result is less than -32,768, the borrow flag, M1021 is set to ON.

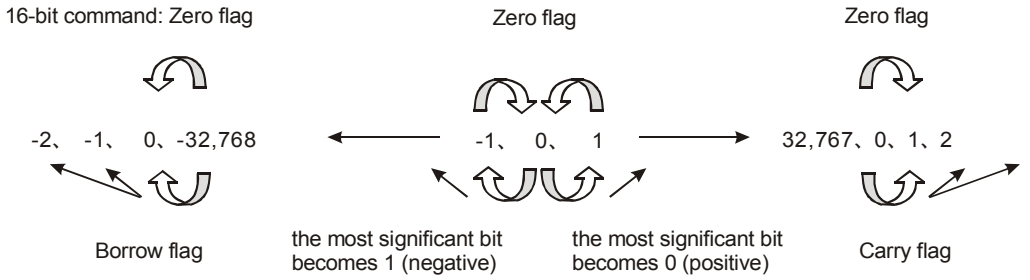
139. If the operation result exceeds 32,767, the carry flag, M1022 is set to ON.

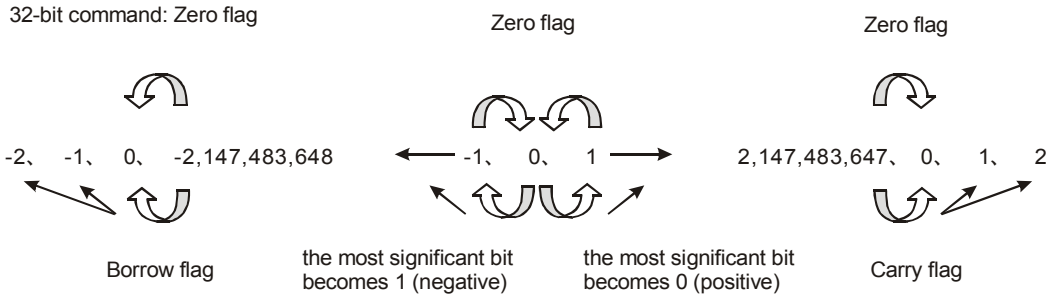
32-bit instruction:

140. If the operation result is “0”, then the Zero flag, M1020 is set to ON.

141. If the operation result is less than -2,147,483,648, the borrow flag, M1021 is set to ON.

142. If the operation result exceeds 2,147,483,647, the carry flag, M1022 is set to ON





API	Mnemonic			Operands	Function
21	D	SUB	P	**S_1, S_2, D**	Subtraction

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	SUB, SUBP: 7 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	DSUB, DSUBP: 13 steps
D								*	*	*	*	*	*	*	*	

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Minuend **S_2**: Subtrahend **D**: Subtraction result

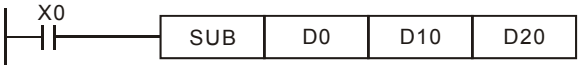
Description:

143. The data contained within the source device, **S_2** is subtracted from the contents of source device **S_1**. The result of this calculation is stored in the destination device **D**.
144. The most significant bit is the sign. 0 indicates positive and 1 indicates negative. All calculation is algebraically processed.
145. If operand **S_1, S_2, D** use index register F, then only 16-bit instruction is available.
146. Flags: M1020 (Zero flag), M1021 (Borrow flag), M1022 (Carry flag).

Program Example 1:

16-bit instruction:

When X0 = ON, the data in D10 is subtracted from the data in D0 and the result is placed in D20.

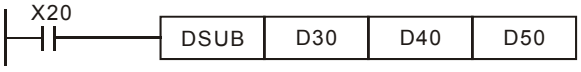


(D0) – (D10) = (D20)

Program Example 2:

32-bit instruction:

When X20 = ON, the data in (D41, D40) is subtracted from the data in (D31, D30) and the result is placed in (D51, D50). (D30, D40, D50 is the lower 16-bit data, and D31, D41, D51 is the higher 16-bit data)



(D31, D30) – (D41, D40) = (D51, D50)

API	Mnemonic			Operands	Function
22	D	MUL	P	S_1, S_2, D	Multiplication

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		MUL, DMULP: 7 steps
S_2					*	*	*	*	*	*	*	*	*	*		DMUL, DMULP: 13 steps
D								*	*	*	*	*	*	*		

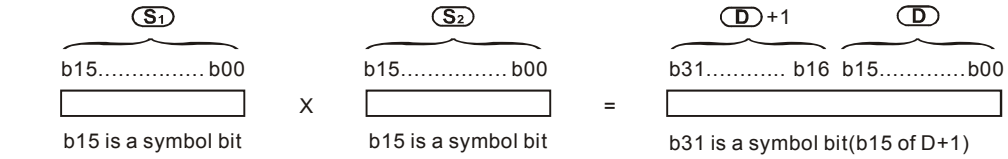
ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Multiplicand S_2: Multiplier **D**: Multiplication result

Description:

147. The contents of the two source devices (S_1, S_2) are multiplied together and the result is stored at the destination device (**D**).
148. MSB = 0, positive; MSB = 1, negative.
149. If operands S_1, S_2 use index register F, then only 16-bit instruction is available.
150. If operand **D** uses index register E, then the 32-bit instruction must be used.
151. 16-bit instruction



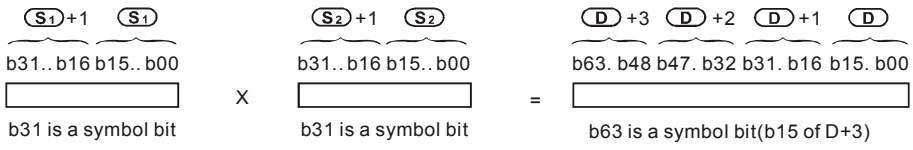
b15=0,S1 is a positive value
b15=1,S1 is a negative value

b15=0,S2 is a positive value
b15=1,S2 is a negative value

b31=0,D(D+1) is a positive value
b31=1,D(D+1) is a negative value

If **D** is specified with a bit address, it must use K1 ~ K4 to store a 16-bit result. The result of the MUL instruction is always a 32-bit value. So, the D-register address used for the destination (D) for this instruction always uses 2 consecutive D registers. .

152. 32-bit instruction



b31=0,S1(S1+1) are positive value
b31=1,S1(S1+1) are negative value

b31=0,S2(S2+1) are positive value
b31=1,S2(S2+1) are negative value

b63=0, D~(D+3) are positive value
b63=1, D~(D+3) are negative value

If **D** is specified with a bit address, it must utilize K1~K8 to store a 32-bit result. If **D** is specified with a word address and the controller is a ELCB-PB, it will only store the low 32-bit data. The ELC-PA/PV, ELCM-PH/PA, ELC2-PB/PH/PA/PE/PV will store 64-bit data. 4 consecutive 16-bit registers will be used to store 64-bit data.

Program Example:

The value in D10 is multiplied by the value in D0 and the total is a 32-bit result stored in (D21, D20). The upper 16-bit data is stored in D21 and the lower one is stored in D20. The polarity of the result is indicated by the OFF/ON of the most significant bit. OFF indicates the value of positive (0) and ON indicates the value of negative (1).

(D0) × (D10) = (D21, D20)

16-bit × 16-bit = 32-bit

API	Mnemonic			Operands	Function
23	D	DIV	P	S_1, S_2, D	Division

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		DIV, DIVP: 7 steps DDIV, DDIVP: 13 steps
S_2					*	*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*		

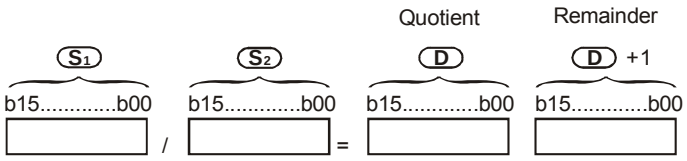
ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: dividend S_2: divisor **D**: Quotient and Remainder

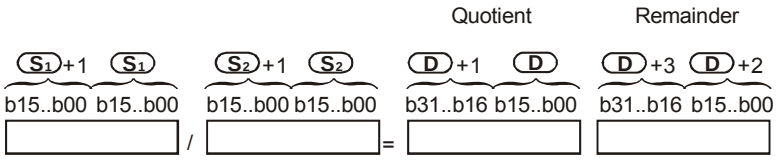
Description:

1. The primary source (S_1) is divided by the secondary source (S_2). The result is stored in the destination (**D**).
2. This instruction is not executed when the divisor is "0". Then, the flag M1067, M1068 = ON and D1067 holds error code H0E19.
3. If operands S_1, S_2, **D** use index F, then only the 16-bit instruction is available.
4. 16-bit instruction:



If D is specified with a bit address, it must utilize K1 ~ K4 to store a 16-bit result. 2 consecutive 16-bit registers will be used to store the 32-bit data consisting of the quotient and remainder.

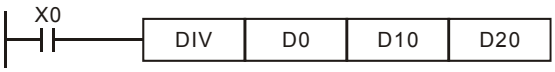
5. 32-bit instruction:



If D is specified with a bit address, it must utilize K1 ~ K8 to store a 32-bit result for ELCB-PB controllers. 4 consecutive 16-bit registers are used to store the quotient and remainder for ELC-PA/PV, ELCM-PH/PA, ELC2-PB/PH/PA/PE/PV controllers.

Program Example:

When X0 = ON, the value in D0 (dividend) is divided by the value in D10 (divisor). The quotient is stored in D20 and the remainder is stored in D21. The polarity of the result is indicated by the OFF/ON of the most significant bit. OFF indicates the value of positive and ON indicates the value of negative.



API	Mnemonic			Operands	Function
24	D	INC	P	**D**	Increment

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	INC, INCP: 3 steps DINC, DINCP: 5 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

D: Destination

Description:

153. If the instruction is not in pulse mode, “1” is added to the value of the destination **D** every execution of the instruction, which could be every scan.
154. This instruction works best using pulse mode (INCP, DINCP).
155. In the 16-bit instruction, when +32,767 is reached, “1” is added and it will write a value of –32,768 to the destination. In 32-bit instruction, when +2,147,483,647 is reached, “1” is added and it will write a value of -2,147,483,648 to the destination.
156. Flags M1020~M1022 won’t be affected by the operation result of this instruction.
157. If operand **D** uses index register F, then only 16-bit instruction is available.

Program Example:

When X0 = OFF → ON, the content of D0 will be incremented by 1.

API	Mnemonic			Operands	Function
25	D	DEC	P	**D**	Decrement

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	DEC, DECP: 3 steps DDEC, DDECP: 5 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

D: Destination

Description:

158. If the instruction is not in pulse mode, “1” is subtracted from the value of destination **D** on every execution of the instruction, which could be every scan.
159. This instruction typically works best using pulse mode (DECP, DDECP).
160. In 16-bit instruction, when –32,768 is reached, “1” is subtracted and it will write a value of +32,767 to the destination. In 32-bit instruction, when -2,147,483,648 is reached, “1” is subtracted and it will write a value of +2,147,483,647 to the destination.
161. Flags M1020~M1022 won’t be affected by the operation result of this instruction.
162. If operand **D** uses index register F, then only 16-bit instruction is available.

Program Example:

When X0 = OFF → ON, the value in D0 will be decremented by 1.

API	Mnemonic		Operands	Function
26	WAND	P	S_1, S_2, D	Logical AND 16-bit

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	WAND, WANDP: 7 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

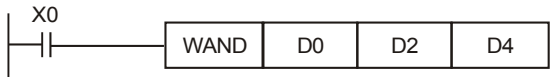
S_1: First data source **S_2**: Second data source **D**: Operation result

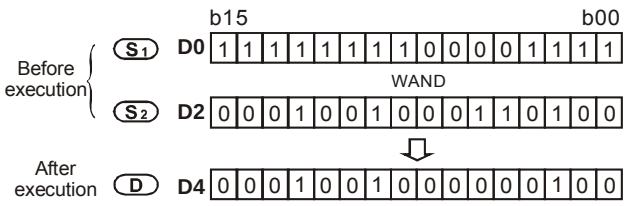
Description:

1. A logical AND operation is performed on the bit patterns of the contents of the two source addresses (**S_2** and **S_1**). The result of the logical AND is stored in the destination address (**D**).
2. For 32-bit operation please refer to the DAND instruction.

Program Example:

When X0 = ON, the 16-bit sources D0 and D2 are analyzed and the result of the logical WAND (Word AND) is stored in D4.





API	Mnemonic		Operands	Function
26	DAND	P	S_1, S_2, D	Logical AND 32-bit

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		DAND, DANDP: 13 steps
S_2					*	*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*		

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

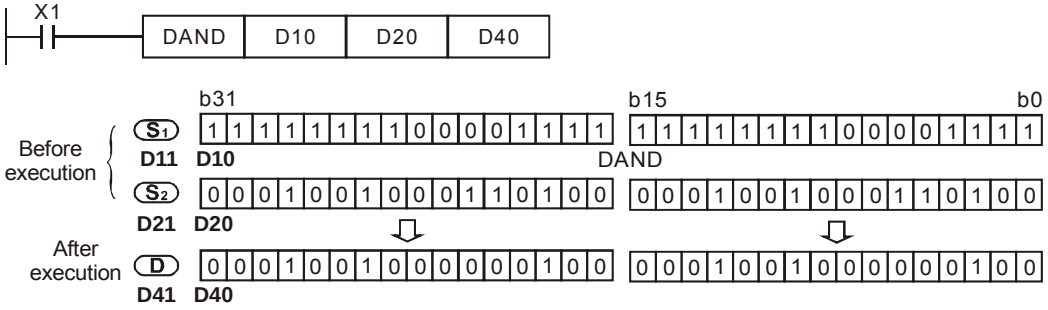
S_1: First data source **S_2**: Second data source **D**: Operation result

Description:

163. Logical Double word AND operation.
164. A logical AND operation is performed on the bit patterns of the contents of the two source addresses (**S_2** and **S_1**). The result of the logical AND is stored in the destination device (**D**).
165. For 16-bit operation please refer to the WAND instruction.

Program Example:

When X1 = ON, the 32-bit source (D11, D10) and (D21, D20) are analyzed and the result of the logical DAND (Double word AND) is stored in (D41, D40).



API	Mnemonic		Operands	Function
27	WOR	P	S_1, S_2, D	Logical OR 16-bit

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	WOR, WORP: 7 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

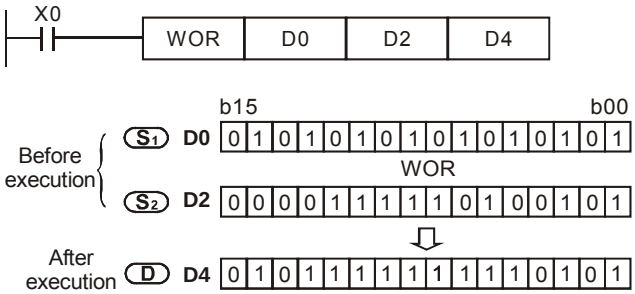
S_1: First data source **S_2**: Second data source **D**: Operation result

Description:

1. A logical OR operation is performed on the bit patterns of the contents of the two source addresses (**S_2** and **S_1**). The result of the logical OR is stored in the destination device (**D**).
2. For 32-bit operation please refer to the DOR instruction.

Program Example:

When X0 = ON, the 16-bit data source D0 and D2 are analyzed and the result of the logical WOR is stored in D4.



API	Mnemonic		Operands	Function
27	DOR	P	S_1, S_2, D	Logical OR 32-bit

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		DOR, DORP: 13 steps
S_2					*	*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*		

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

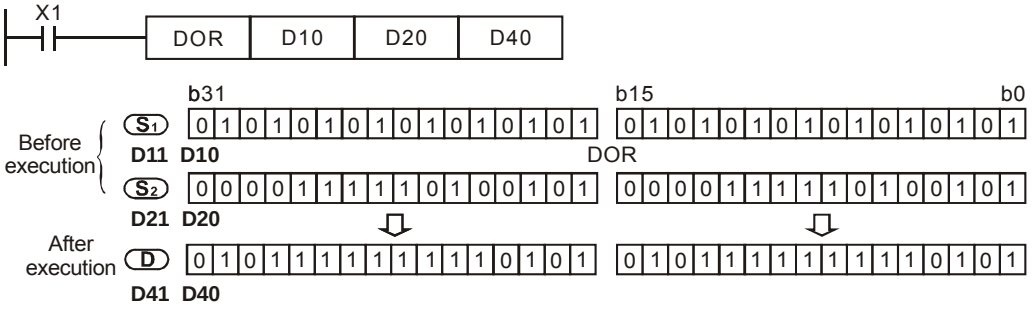
S_1: First data source **S_2**: Second data source **D**: Operation result

Description:

166. Logical Double word OR operation.
167. A logical OR operation is performed on the bit patterns of the contents of the two source addresses (**S_2** and **S_1**). The result of the logical OR analysis is stored in the destination device (**D**)
168. For 16-bit operation please refer to the WOR instruction.

Program Example:

When X1 is ON, the 32-bit data source (D11, D10) and (D21, D20) are analyzed and the operation result of the logical DOR is stored in (D41, D40).



API	Mnemonic		Operands	Function
28	WXOR	P	S_1, S_2, D	Exclusive XOR 16-bit

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	WXOR, WXORP: 7 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

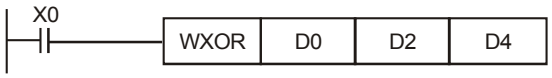
S_1: First data source **S_2**: Second data source **D**: Operation result

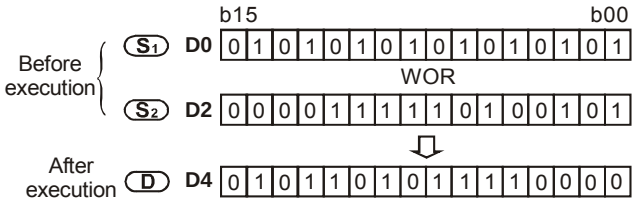
Description:

169. A logical XOR operation is performed on the bit patterns of the contents of the two source addresses (**S_2** and **S_1**). The result of the logical XOR is stored in the destination device (**D**)
170. For 32-bit operation please refer to the DXOR instruction.

Program Example:

When X0 = ON, the 16-bit data source D0 and D2 are analyzed and the operation result of the logical WXOR is stored in D4.





API	Mnemonic		Operands	Function
28	DXOR	P	S_1, S_2, D	Exclusive XOR 32-bit

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		DXOR, DXORP: 13 steps
S_2					*	*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*		

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

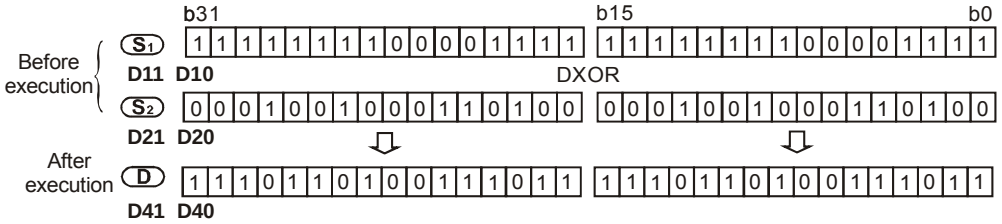
S_1: First data source **S_2**: Second data source **D**: Operation result

Description:

171. Logical Double word XOR operation.
172. A logical XOR operation is performed on the bit patterns of the contents of the two source addresses (**S_2** and **S_1**). The result of the logical DXOR is stored in the destination device (**D**)
173. If operands **S_1**, **S_2**, **D** use index F, only a 16-bit instruction is available.
174. For 16-bit operation please refer to the WXOR instruction.

Program Example:

When X1 = ON, the 32-bit data source = (D11, D10) and (D21, D20) are analyzed and the operation result of the logical DXOR = is stored in (D41, D40).



API	Mnemonic	Operands	Function
29	D NEG P	**D**	Negative (2's Compliment)

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	NEG, NEGP: 3 steps DNEG, DNEGP: 5 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

D: Store the result of the 2's Compliment operation

Description:

1. The bit pattern of the selected device is inverted. This means any occurrence of a '1' becomes a '0' and any occurrence of a '0' will be as a '1'. When this is complete, a binary 1 is added to the bit pattern. The result is the logical sign change of the value, e.g. a positive number will become a negative number or a negative number will become a positive.
2. This instruction works best using pulse instruction (NEGP, DNEGP).
3. If operand **D** uses index F, then only 16-bit instruction is available.

Program Example 1:

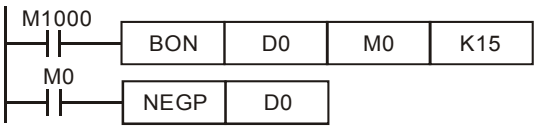
When X0 goes from OFF → ON, all bits in D10 will be inverted (0→1, 1→0) and then 1 will be added and saved in the original register, D10.

Program Example 2:

Obtaining the absolute value of a negative value:

175. When the 15th bit of D0 is "1", M0 = ON. (D0 is a negative value).

176. When M0 = ON, the absolute value of D0 can be obtained using the NEG instruction.



Program Example 3:

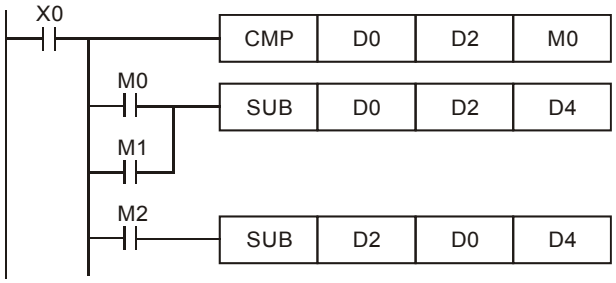
Obtain the absolute value of the remainder of the subtraction. When X0 = ON,

a) If D0 > D2, M0 = ON.

b) If D0 = D2, M1 = ON.

c) If D0 < D2, M2 = ON.

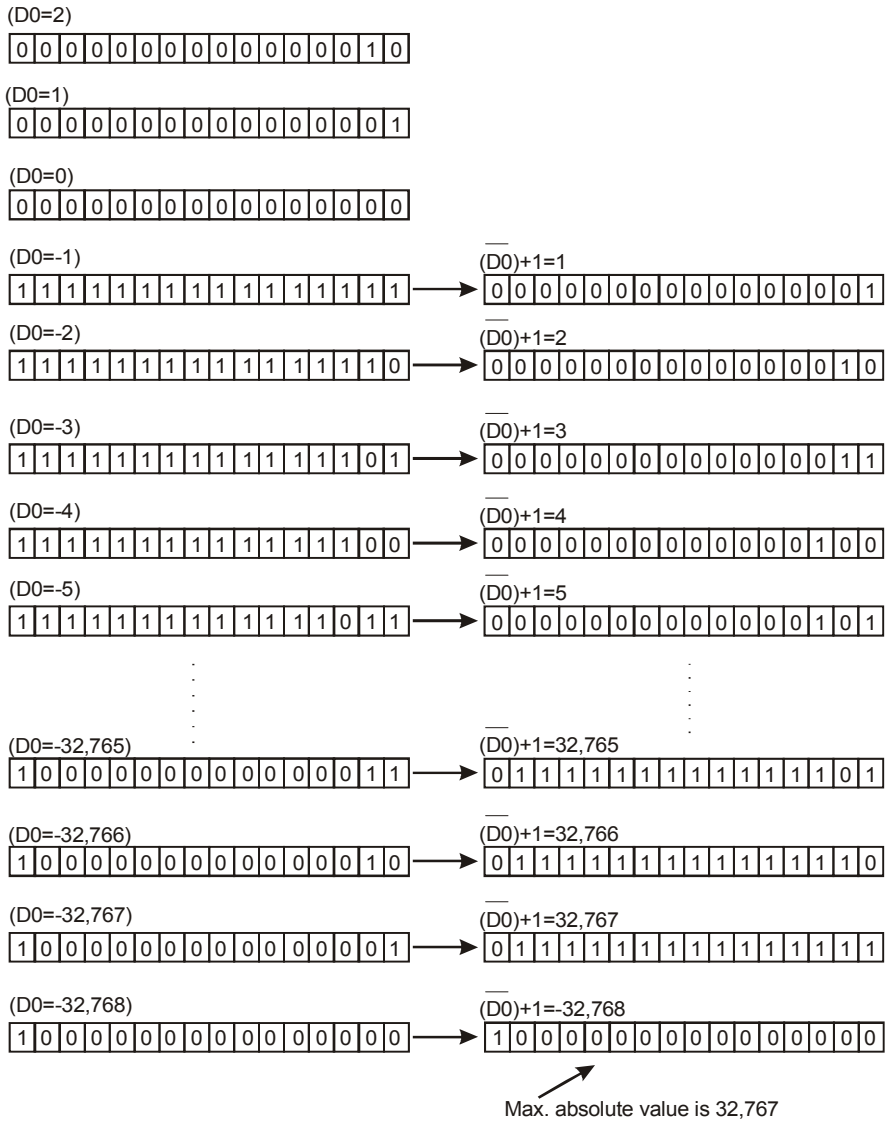
d) D4 will always be a positive value.



Indication of the negative value and absolute value

177. The content of the most significant bit of the register indicates whether the value is positive or negative. It is a positive value when the most significant bit (MSB) = “0” and it is a negative value when the MSB = “1”.

178. If it is a negative value, the absolute value can be obtained by using the NEG instruction.



API	Mnemonic			Operands	Function
30	D	ROR	P	D, n	Rotate Right

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	ROR, RORP: 5 steps DROR, DRORP: 9 steps
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

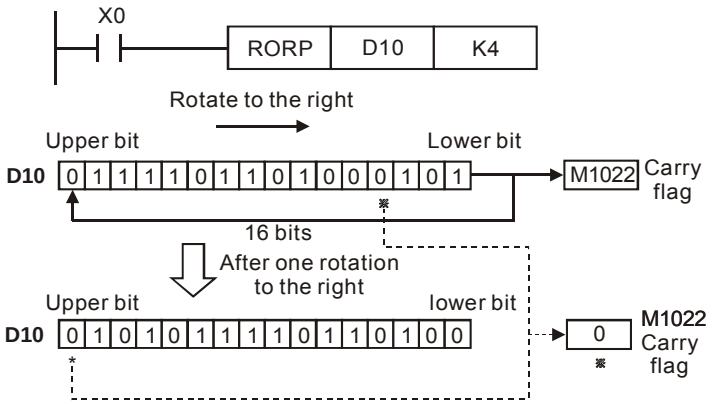
D: Address to be rotated **n**: Number of bits to be rotated in 1 rotation

Description:

179. All the bits of **D** are rotated **n** bit places to the right on every operation of the instruction, which could be every scan.
180. The state of the last bit rotated is copied to the carry flag M1022 (Carry flag)
181. This instruction works best using pulse instruction (RORP, DRORP).
182. If operand **D** uses index F, then only 16-bit instruction is available.
183. If operand **D** is specified as KnY, KnM, KnS, only K4 (16-bit) and K8 (32-bit) is valid.
184. Valid range of operand **n**: 1≤ **n** ≤16 (16-bit), 1≤ **n** ≤32 (32-bit)

Program Example:

When X0 goes from OFF → ON, the 16 bit data of D10 will rotate 4 bits to the right, as shown in the diagram, bit b3 (prior to rotation) will be moved to the carry flag (CY) M1022.



API	Mnemonic			Operands	Function
31	D	ROL	P	**D, n**	Rotate Left

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	ROL, ROLP: 5 steps DROL, DROLP: 9 steps
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

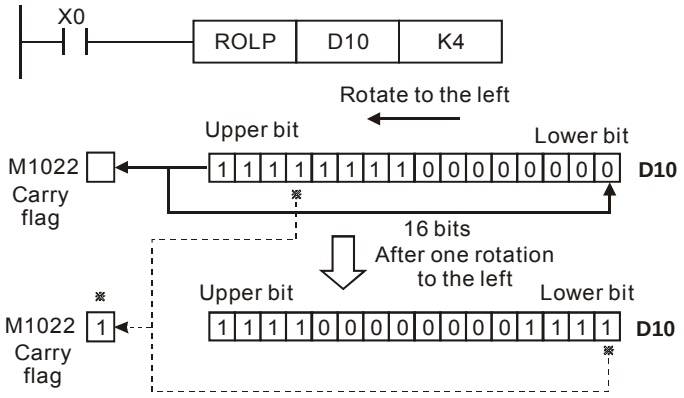
D: Address to be rotated **n**: Number of bits to be rotated in 1 rotation

Description:

185. All the bits of **D** are rotated **n** bit places to the left on every operation of the instruction, which could be every scan.
186. The status of the last bit rotated is copied to the carry flag M1022.
187. This instruction works best using pulse instruction (ROLP, DROLP).
188. If operand **D** uses index F, then only 16-bit instruction is available.
189. If operand **D** is specified as KnY, KnM, KnS, only K4 (16-bit) and K8 (32-bit) are valid.
190. Valid range of operand **n**: 1≤ **n** ≤16 (16-bit), 1≤ **n** ≤32 (32-bit)

Program Example:

When X0 goes from OFF → ON, all 16 bits of D10 will rotate 4 bits to the left, as shown in the diagram, and b12 (prior to rotation) will be moved to the carry flag (CY) M1022.



API	Mnemonic			Operands	Function
32	D	RCR	P	D, n	Rotate Right with Carry

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	RCR, RCRP: 5 steps DRCR, DRCRP: 9 steps
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

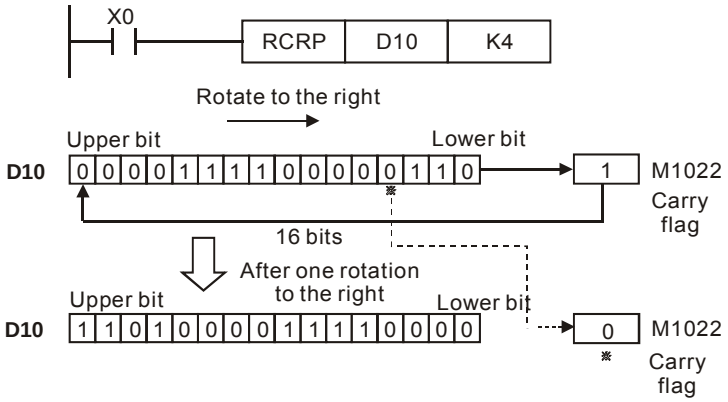
D: Address to be rotated **n**: Number of bits to be rotated in 1 rotation

Description:

191. All the bits of **D** are rotated **n** bit places to the right including the carry flag on every operation of the instruction, which could be every scan.
192. The status of the last bit rotated is moved into the carry flag M1022. On the following operation of the instruction M1022 is the first bit to be moved back into the destination device.
193. This instruction works best with the pulse instruction (RCRP, DRCRP).
194. If operand **D** uses index F, then only 16-bit instruction is available.
195. If operand **D** is specified as KnY, KnM, KnS, only K4 (16-bit) and K8 (32-bit) are valid.
196. Valid range of operand **n**: 1≤ **n** ≤16 (16-bit), 1≤ **n** ≤32 (32-bit)

Program Example:

When X0 goes from OFF →ON, the 16 bit value in D10, including the carry flag (M1022) will rotate 4 bits to the right, as shown in the diagram. b3 (prior to rotation) will be moved to the carry flag M1022, and the original contents of the carry flag M1022 will be moved to bit b12.



API	Mnemonic			Operands	Function
33	D	RCL	P	D, n	Rotate Left with Carry

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	RCL, RCLP: 5 steps DRCL, DRCLP: 9 steps
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

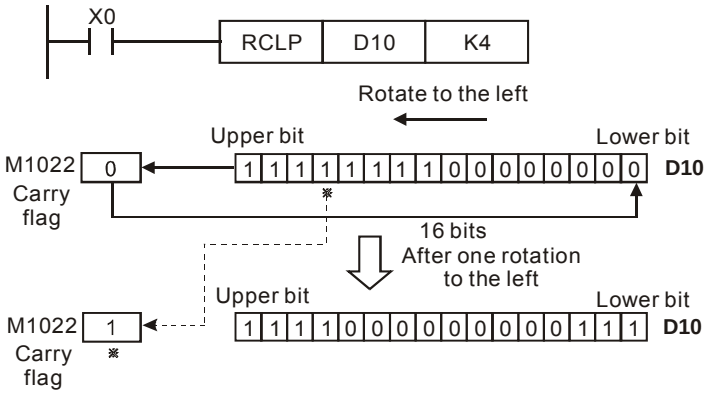
D: Address to be rotated **n**: Number of bits to be rotated in 1 rotation

Description:

197. All the bits of **D** are rotated **n** bit places to the left including the carry flag on every operation of the instruction, which could be every scan.
198. The status of the last bit rotated is moved into the carry flag M1022. On the following operation of the instruction M1022 is the first bit to be moved back into the destination device.
199. This instruction works best with the pulse instruction (RCLP, DRCLP).
200. If operand **D** uses index F, then only 16-bit instruction is available.
201. If operand **D** is specified as KnY, KnM, KnS, only K4 (16-bit) and K8 (32-bit) is valid.
202. Valid range of operand **n**: 1≤ **n** ≤16 (16-bit), 1≤ **n** ≤32 (32-bit)

Program Example:

When X0 goes from OFF →ON, the 16 bit value in D10, including the carry flag (M1022) will rotate 4 bits to the left, as shown in the diagram. b12 (prior to rotation) will be moved to the carry flag M1022, and the original contents of the carry flag M1022 will be moved to bit b3.



API	Mnemonic		Operands	Function
34	SFTR	P	**S, D, n_1, n_2**	Bit Shift Right

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												SFTR, SFTRP: 9 steps
D		*	*	*												
n_1					*	*										
n_2					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Starting address of the source device **D**: Starting address of the destination device

n_1: Length of data to be shifted n_2: Number of bits to be shifted as a group

Description:

203. Shift n_1 bits of **S** to the right by n_2 bits. Shift n_2 bits of **D** to the most significant bits of **S**.
204. This instruction works best with the pulse instruction (SFTRP).
205. Valid range of operand **n1**, **n2** : 1≤ **n2** ≤ **n1** ≤1024, ELCB-PB models: 1≦ **n2** ≦ **n1** ≦512

Program Example:

206. When X0 OFF →ON, the 16 bits M0~M15 will shift 4 bits to the right, and 4 bits from X0-X3 into M12-M15..
207. Please refer to the following ❶~❺ steps to perform SFTR instruction during a single scan.

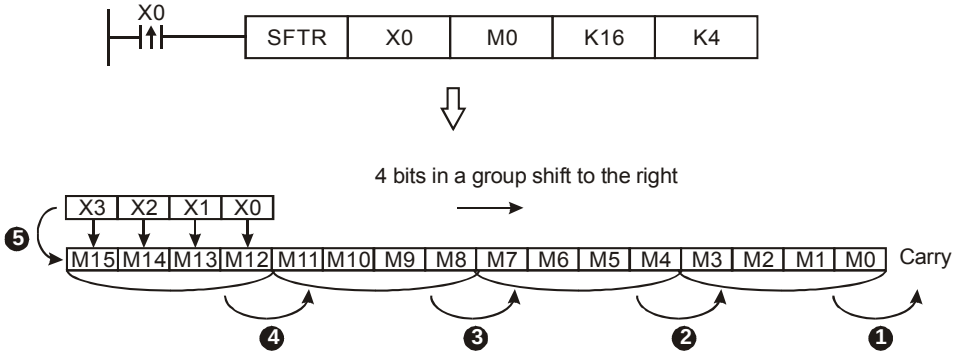
❶ M3~M0 → Carry

❷ M7~M4 → M3~M0

❸ M11~M8 → M7~M4

❹ M15~M12 → M11~M8

❺ X3~X0 → M15~M12 complete



API	Mnemonic		Operands	Function
35	SFTL	P	**S, D, n_1, n_2**	Bit Shift Left

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												SFTL, SFTLP: 9 steps
D		*	*	*												
n_1					*	*										
n_2					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Starting source address **D**: Starting destination address **n_1**: Number of bits to be shifted

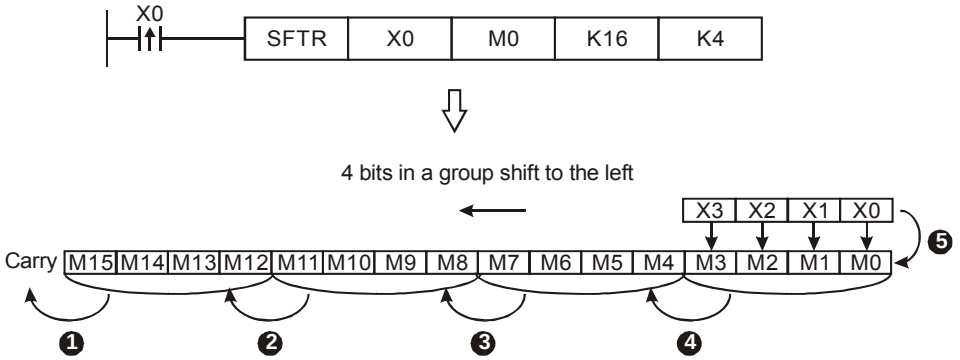
n_2: Number of bit positions to shift the data bits as a group

Description:

208. Shift **n_1** bits of **S** to the left by **n_2** bit positions. Shift **n_2** bits of **D** to the least significant bits of **S.**
209. This instruction works best with the pulse instruction (SFTLP).
210. Valid range of operand **n1**, **n2** : 1≤ **n2** ≤ **n1** ≤1024, In ELCB-PB models: 1≦ **n2** ≦ **n1** ≦ 512

Program Example:

211. When X0 OFF →ON, the 16 bit data of M0~M15 will shift 4 bits to the left. And 4 bits from X0 into the low order bits of M0.
212. Please refer to the following ❶~❺ steps to perform SFTR instruction during a single scan.
    - ❶ M15~M12 → Carry
    - ❷ M11~M8 → M15~M12
    - ❸ M7~M4 → M11~M8
    - ❹ M3~M0 → M7~M4
    - ❺ X3~X0 → M3~M0 complete



API	Mnemonic		Operands	Function
36	WSFR	P	**S, D, n_1, n_2**	Word Shift Right

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	WSFR, WSFRP: 9 steps
S							*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n_1					*	*										
n_2					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Starting source address **D**: Starting destination address **n_1**: Length of data to be shifted

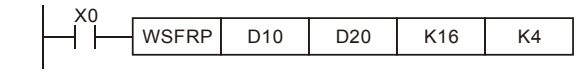
n_2: Number of registers to be shifted as a group

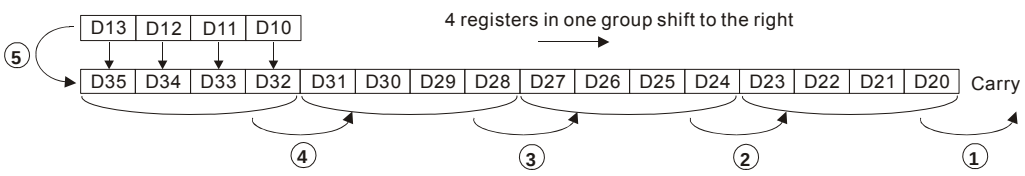
Description:

213. Shift **n_1** registers of **S** to the right by **n_2** registers. Shift **n_2** registers of **D** to the most significant registers of **S**.
214. This instruction works best with the pulse instruction (WSFRP).
215. When using operand **S** and **D** for bit data types, the data types must be equal. For example, if one of these bit or word types is used for **S**, it must also be used for **D**: KnX, KnY, KnM, KnS and the other kind is T, C, D.
216. When using operand **S** and **D** bit data types, the Kn value must be equal.
217. Valid range of operand **n1**, **n2** : 1≤ **n2** ≤ **n1** ≤512

Program Example 1:

218. When X0 OFF →ON, the registers starting at D20~D35 will shift 4 registers to the right. And 4 registers from D10 will shift into the upper registers of the destination.
219. Refer to the following ❶~❺ steps to perform WSFR instruction during a single scan.
    - ❶ D23~D20 → Carry
    - ❷ D27~D24 → D23~D20
    - ❸ D31~D28 → D27~D24
    - ❹ D35~D32 → D31~D28
    - ❺ D13 ~D10 → D35~D32 complete





Program Example 2:

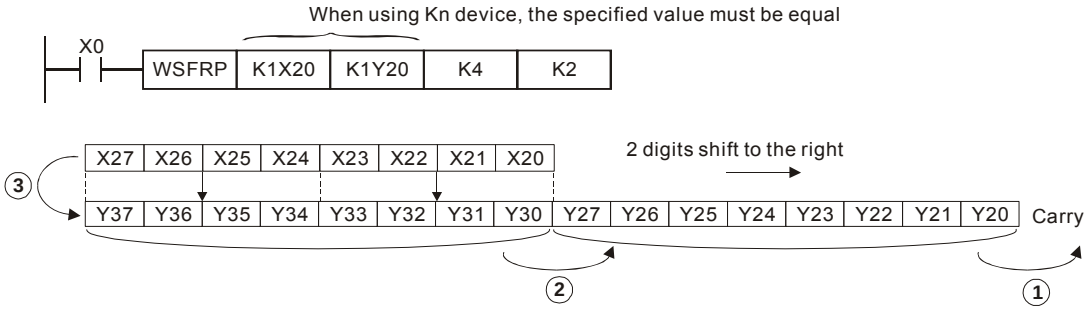
220. When X0 OFF →ON, the bit registers of Y20~Y37 are shifted 2 digits to the right.

221. Please refer to the following ❶~❸ steps to perform WSFR instruction of one time shift.

❶ Y27~Y20 → carry

❷ Y37~Y30 → Y27~Y20

❸ X27~X20 → Y37~Y30 complete



API	Mnemonic		Operands	Function
37	WSFL	P	**S, D, n_1, n_2**	Word Shift Left

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			WSFL, WSFLP: 9 steps
D								*	*	*	*	*	*			
n_1					*	*										
n_2					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Starting source address **D**: Starting destination address **n_1**: Length of data to be shifted

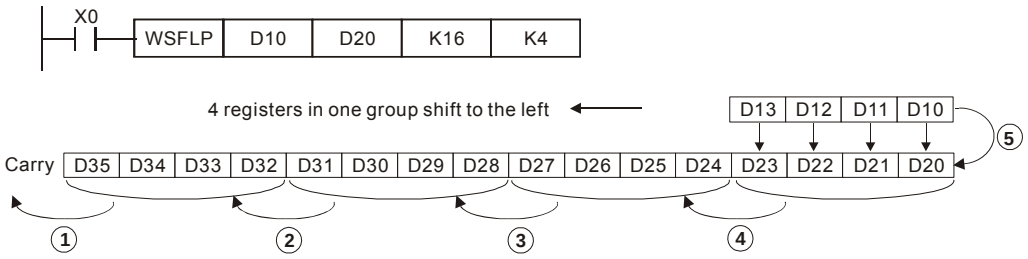
n_2: Number of registers to be shifted as a group

Description:

222. Shift **n_1** registers of **D** to the left by **n_2** registers. Shift **n_2** registers of **S** to the least significant registers of **D.** The high **n_2** registers of **D** are moved to the Carry.
223. This instruction works best with the pulse instruction (WSFLP).
224. When using operand **S** and **D** for bit data types, the data types must be equal. For example, if one of these bit or word types is used for **S**, it must also be used for **D**: KnX, KnY, KnM, KnS and the other kind is T, C, D.
225. When using operand **S** and **D** bit data types, the Kn value must be equal.
226. Valid range of operand **n1**, **n2** : 1≤ **n2** ≤ **n1** ≤512

Program Example:

227. When X0 OFF →ON, the registers starting at D20~D35 will shift 4 registers to the left. And 4 registers from D10 will shift into the lower registers of the destination.
228. Please refer to the following ❶~❺ steps to perform WSFL instruction during a single scan.
    - ❶ D35~D32 → Carry
    - ❷ D31~D28 → D35~D32
    - ❸ D27~D24 → D31~D28
    - ❹ D23~D20 → D27~D24
    - ❺ D13~D10 → D23~D20 complete



API	Mnemonic		Operands	Function
38	SFWR	P	**S, D, n**	Shift Register Write

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	SFWR, SFWRP: 7 steps
D								*	*	*	*	*	*			
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Source address **D**: Starting address of data stack **n**: Length of data stack

Description:

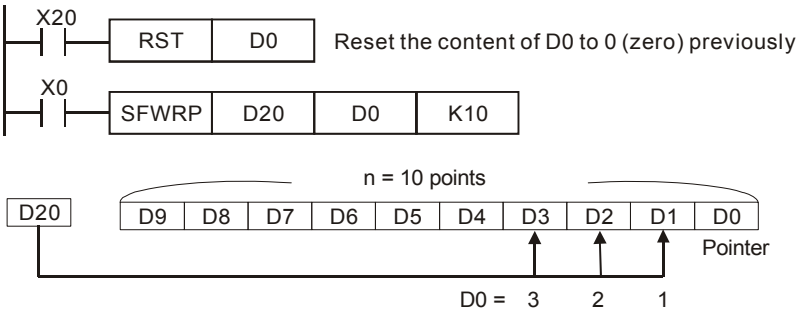
229. This instruction defines the data stack of **n** words starting with **D.** This first address of the data stack is the pointer into the remainder of the data stack. When D=1, the value of S is moved into position 1 of the data stack (when the instruction is executed the first time). D is incremented with each execution of the instruction and should be reset when it reaches the last element of the data stack. When the contents of the pointer (D) exceeds **n-1**, the instruction stops and carry flag M1022= ON.
230. This instruction works best with the pulse instruction (SFWRP).
231. Valid range of operand **n**: 2≤ **n** ≤512

Program Example:

232. First, X20=ON resets the contents of D0 to 0. When X0 goes from OFF to ON, the contents of D0 becomes 1 and the contents of D20 is moved into D1. After changing the content of D20, X0 goes from OFF to ON again, then the contents of D0 becomes 2 and the contents of D20 is moved into D2.
233. Please refer to the following ❶~❷steps to perform SFWR instruction.

❶ The content of D20 is created and built in D1.

❷ The content of D0 becomes 1.



Point to note:

The API 38 SFWR instruction can be used with the API 39 SFRD instruction create a first-in, first-out (FIFO) application.

API	Mnemonic		Operands	Function
39	SFRD	P	**S, D, n**	Shift Register Read

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			SFRD, SFRDP: 7 steps
D								*	*	*	*	*	*	*	*	
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

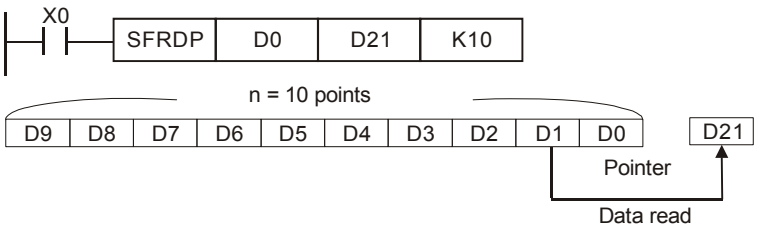
S: Starting address of data stack **D**: Destination address **n**: Length of data stack

Description:

234. This instruction defines the data stack of **n** words starting from **S** as a FIFO data stack and specifies the first device as the pointer (**S**). The contents of the pointer indicates the current position in the stack. When SFRDP is executed, the first data register (**S+1**) will be moved to **D** and all data in the stack moves up one register. The contents of the pointer is decremented by 1. When the content in pointer = 0, the instruction stops and carry flag M1022= ON
235. This instruction works best with the pulse instruction (SFRDP).
236. Valid range of operand **n**: 2≤ **n** ≤512

Program Example:

237. When X0 goes from OFF to ON, D9~D2 are all shifted one register to the right and the pointer content of D0 is decremented by 1 and the content of D1 is moved to D21.
238. Please refer to the following ❶~❸ steps to perform SFRD instruction.
    ❶ The content of D1 is moved to D21.
    ❷ D9~D2 are all shifted one register to the right.
    ❸ The content of D0 is decremented by 1.



API	Mnemonic		Operands	Function
40	ZRST	P	D_1, D_2	Zone Reset

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D_1		*	*	*							*	*	*			ZRST, ZRSTP: 5 steps
D_2		*	*	*							*	*	*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

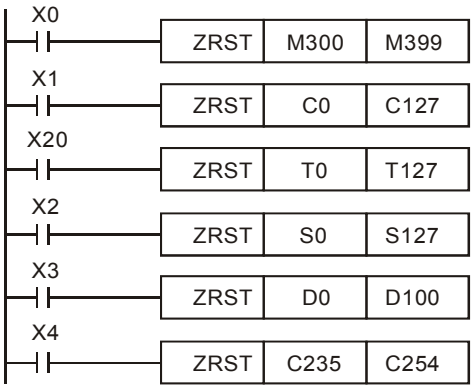
D_1: Starting destination address D_2: Ending destination address

Description:

239. When the instruction is executed, range D_1 to D_2 will be reset.
240. When $D_1 > D_2$, then only device D_2 is reset.
241. Operand D_1 and D_2 must be the same data type, Valid range: $D_1 \leqq D_2$.
242. ELCB-PB models, standard and High speed counters cannot be mixed.
243. This instruction works best with the pulse instruction (ZRSTP).

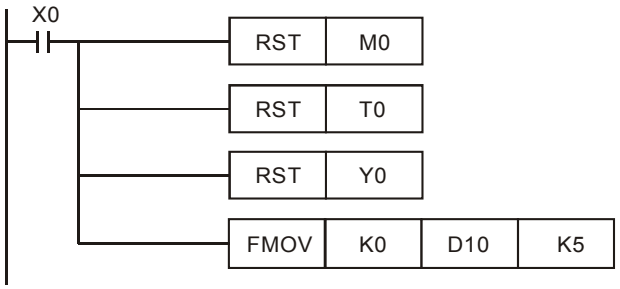
Program Example:

244. When X0 = ON, M300 to M399 will be reset to OFF.
245. When X1 = ON, C0 to C127 will all be reset. Their present value =0, coil output will be reset to OFF.
246. When X20 = ON, T0 to T127 will all be reset. Their present value =0, coil output will be reset to OFF.
247. When X2 = ON, the status of S0 to S127 will be reset to OFF.
248. When X3 = ON, the data of D0 to D100 will be reset to 0.
249. When X4 = ON, C235 to C254 will all be reset. Their present value =0, coil output will be reset to OFF.



Points to note:

1. Bit addresses Y, M, S and word addresses T, C, D, can be reset individually with the RST instruction.
2. For clearing multiple devices, API 16 FMOV instruction can be used to send K0 to word addresses T, C, D or bit addresses KnY, KnM, KnS.



API	Mnemonic		Operands	Function
41	DECO	P	**S, D, n**	Decode

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*	*	*					*	*	*	*	*	DECO, DECOP: 7 steps
D		*	*	*							*	*	*	*	*	
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

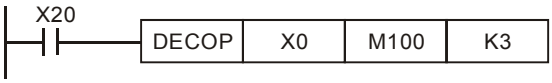
S: Source to decode **D**: Destination **n**: Number of bits to decode

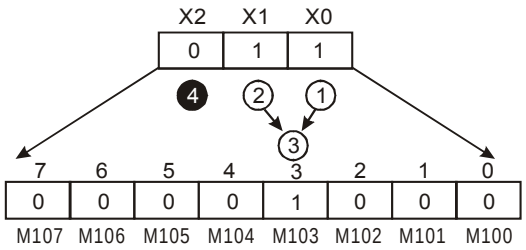
Description:

1. Decodes the lower "**n**" bits of source **S** and stores the result of "2^{n}" bit in **D**.
2. This instruction works best with the pulse instruction (DECOP).
3. When operand **D** is a bit device, **n**=1~8, when operand **D** is a word device, **n**=1~4

Program Example 1:

1. **n** valid range: 0< **n** ≦8. But if **n**=0 or **n**>8, an error will occur.
2. If **n** = 8, the decoded data is 2^{8}= 256 bits of data.
3. When X20 goes from OFF → ON, the data of X0~X2 will be decoded to M100~M107.
4. If the source data is 3, M103 (third bit from M100) = ON.
5. After the execution is completed, X20 is changed to OFF. The result in M103 remains.





Program Example 2:

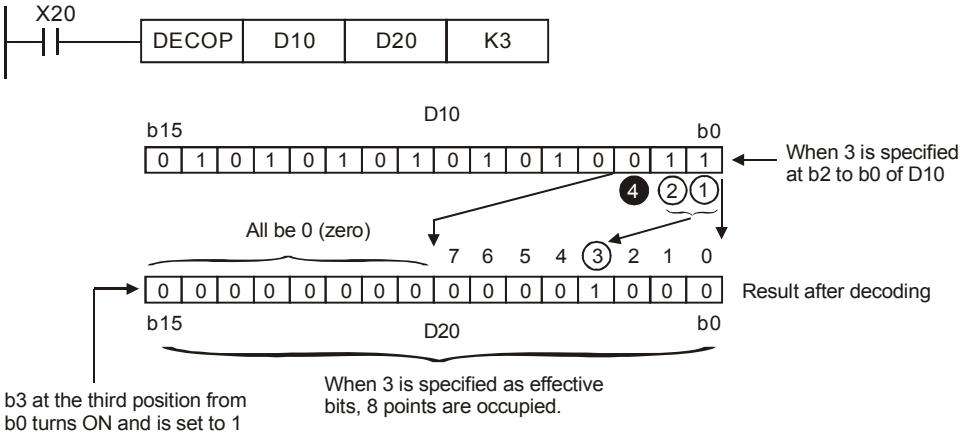
250. **D** valid range: 0< **n** ≦4, if **n**=0 or **n**>4, an error will occur.

251. When **n**=4, the maximum decoded data is 2^4 = 16 points.

252. When X20 goes from OFF → ON, the data in D10 (b2 to b0) will be decoded and stored in D20 (b7 to b0). The unused bits in D20 (b15 to b8) will be all set to 0.

253. This example decodes the three lower bits in D10 and sets the appropriate bit in D20. The bit number in D20 is determined by the value of the 3 low bits in D10. The content of the eight upper bits of D20 are all set to 0.

254. After the execution is completed, X20 is changed to OFF. The result in D20 remains.



API	Mnemonic		Operands	Function
42	ENCO	P	**S, D, n**	Encode

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*							*	*	*	*	*	DECO, DECOP: 7 steps
D											*	*	*	*	*	
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

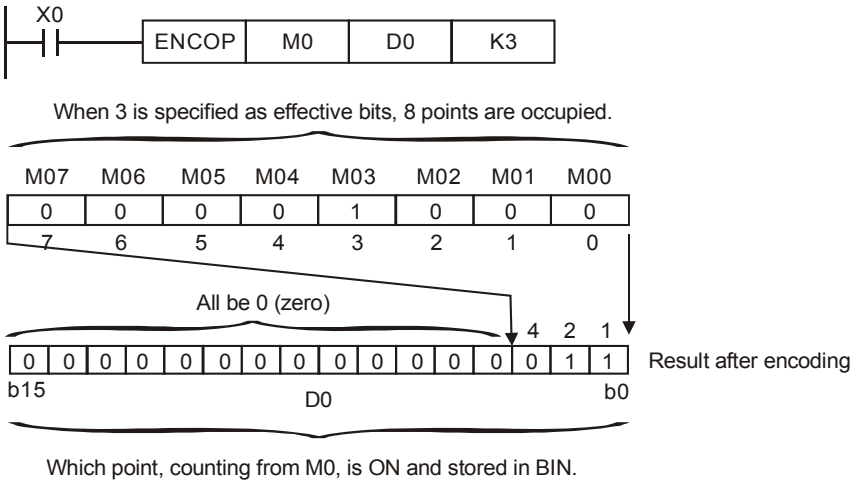
S: Source to encode **D**: Destination for storing encoded data **n**: Number of bits to encode

Description:

255. Encodes the lower "2^{n}" bits of source **S** and stores the result in **D**.
256. If the source device **S** has multiple bits set to a 1, processing is performed on the highest bit position. The bit number set in S is encoded to the low bits of D.
257. This instruction works best with the pulse instruction (ENCOP).
258. When operand **S** is a bit device, **n**=1~8, when operand **S** is a word device, **n**=1~4
259. If no bits in **S** is active (1), M1067, M1068 = ON and D1067 records the error code H0E1A.

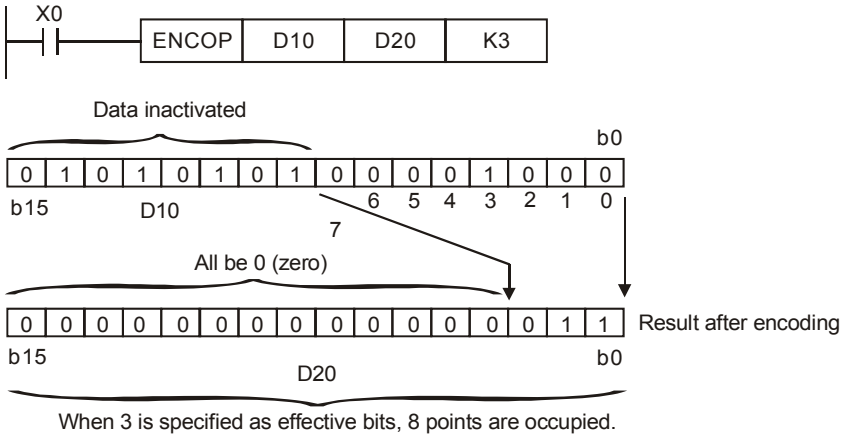
Program Example 1:

260. **S** valid range: 0< **n** ≦8. If **n**=0 or **n**>8, an error will occur.
261. When **n**=8, the maximum decoded data is 2^{8} = 256 points.
262. When X0 goes from OFF → ON, the data in (M0 to M7) will be encoded and stored in the low 3 bits of D0 (b2 to b0). The unused bits in D0 (b15 to b3) will be all set to 0.
263. After the execution is completed, X0 is changed to OFF and the data in **D** remain unchanged.



Program Example 2:

264. **S** Valid rang: 0< **n** ≦4. If **n**=0 or **n**>4, an error will occur.
265. When **n**=4, the maximum decoded data is 2^4 = 16 points.
266. When X0 goes from OFF → ON, the data in D10 will be encoded and stored in the three low bits of D20 (b2 to b0). The unused bits in D20 (b15 to b3) will be all set to 0.
267. After the execution is completed, X0 is changed to OFF and the data in **D** remains unchanged.



API	Mnemonic			Operands	Function
43	D	SUM	P	**S, D**	Sum of ON bits

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	SUM, DSUMP: 5 steps DSUM, DSUMP: 9 steps
D											*	*	*	*	*	

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

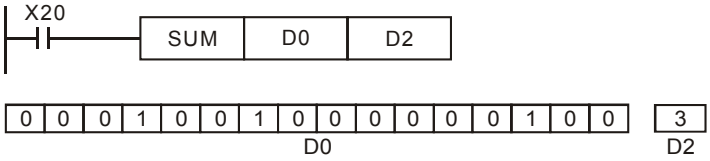
S: Source address **D**: Destination address stores number of ON bits

Description:

268. If the contents of the16 bit source are all “0”, the “Zero” flag, M1020=ON.
269. **D** will occupy two registers when using as a 32-bit instruction.
270. If operands **S**, **D** use index register F, it is only available as a 16-bit instruction.
271. If no bits are ON then the zero flag, M1020 is set.

Program Example:

When X20 =ON, all the bits that = “1” in D0 will be counted and the number stored in D2.



API	Mnemonic			Operands	Function
44	D	BON	P	**S, D, n**	Bit ON Test

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	BON, BONP: 7 steps DBON, DBONP: 13 steps
D		*	*	*												
n					*	*					*	*	*	*	*	

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

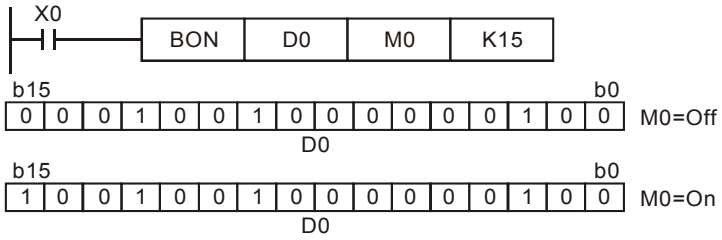
S: Source address **D**: Destination address for storing the result **n**: Bit number to test

Description:

272. The instruction checks the status of a designated bit (specified by **n**) in **S** and stores the result in **D**
273. If operands **S**, **n** use index F, then only 16-bit instruction is available.
274. Valid range of operand **n** : **n**=0~15 (16-bit), **n**=0~31 (32-bit)

Program Example:

275. When X0 = ON, and if the 15th bit of D0 = “1”, M0 is ON. But if the 15th bit of D0 is “0”, M0 is OFF.
276. Once X0 is switched to OFF, M0 will stay at its previous state.



API	Mnemonic			Operands	Function
45	D	MEAN	P	**S, D, n**	Mean Value

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MEAN, MEANP: 7 steps
D								*	*	*	*	*	*	*	*	DMEAN, DMEANP: 13 steps
n					*	*	*	*	*	*	*	*	*	*	*	

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

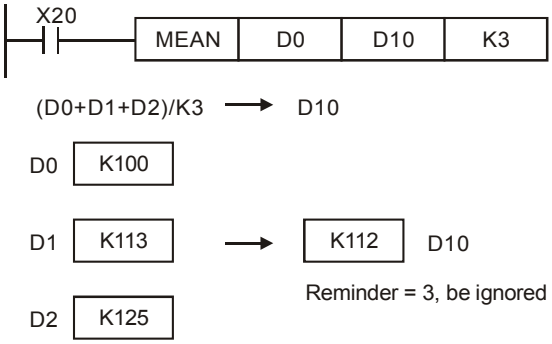
S: Starting source address **D**: Destination address for the result **n**: Length of the **S** file

Description:

277. The instruction obtains the mean value from **n** consecutive registers from **S** and stores the value in **D**.
278. Remainders in the operation will be ignored.
279. If **n** is out of the valid range (1~64), an error will be generated.
280. If operands **D**, **n** use index F, then only 16-bit instruction is available.
281. Valid range of operand **n** : **n**=1~64

Program Example:

When X20 = ON, add up the contents of the three registers starting from D0, and divide the sum by three to get the mean value. Store this mean value in D10 and ignore the remainder.



API	Mnemonic	Operands	Function
46	ANS	**S, m, D**	Alarm Set

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*					ANS: 7 steps
m					*											
D				*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Alarm timer **m**: Time setting prior to alarm **D**: Alarm

Description:

282. ANS instruction is used to drive the output alarm device.
283. ELC-PA: Operand **S** valid range: T0~T191

    ELC-PV, ELC2-PV: Operand **S** valid range: T0~T199

    ELCM-PH/PA, ELC2-PB/PH/PA/PE: Operand **S** valid range: T0~T183

    Operand **m** valid range: K1~K32,767 in units of 100 ms

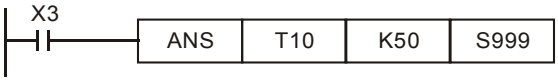
    ELC-PA: Operand **D** valid range: S896~S1023

    ELC-PV, ELC2-PV: Operand **D** valid range: S900~S1023

    ELCM-PH/PA, ELC2-PB/PH/PA/PE: Operand **D** valid range: S912~S1023
284. See ANR for more information
285. Flag: M1048 (ON = Alarm Active), M1049 (ON = Enable Alarms)

Program Example:

If alarm device S999=ON and X3 = ON for more than 5 seconds, S999 will stay ON after X3=OFF. T10 will be reset to OFF, present value=0)



API	Mnemonic		Function
47	ANR	P	Alarm Reset

OP	Range	Program Steps
N/A	Instruction driven by contact is necessary.	ANR, ANRP: 1 steps

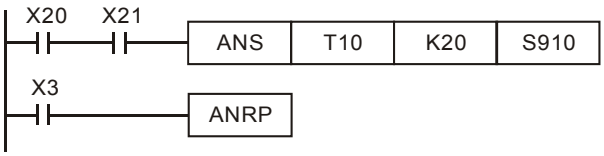
ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Description:

286. ANR instruction is used to reset an alarm.
287. When several alarm devices are ON, the lowest alarm number will be reset.
288. This instruction works best with the pulse instruction (ANRP).

Program Example:

289. When X20 and X21 are simultaneously ON more than 2 seconds, the alarm S910 = ON. If X20 or X21 change to OFF, alarm S910 will remain ON. T10 will reset to OFF, present value is 0.
290. When X20 and X21 are simultaneously ON less than 2 seconds, the present value of T10 is reset to 0.
291. When X3 goes from OFF → ON, activated alarms S896~S1023 (ELC-PA) or S900~S1023 (ELC-PV, ELC2-PV) or S912~S1023 (ELCM-PH/PA, ELC2-PB/PH/PA/PE) will be reset.
292. When X3 goes from OFF → ON again, the second lowest alarm number will be reset.



Points to note:

Flags:

293. M1048: When M1049 = ON, any alarm S896~S1023 (ELC-PA) =ON or S900~S1023 (ELC-PV, ELC2-PV) =ON or S912~S1023 (ELCM-PH/PA, ELC2-PB/PH/PA/PE) =ON will turn M1048 ON. If M1049=OFF, M1048 will not be affected if alarms occur.
294. M1049: When M1049 = ON, D1049 will automatically hold the lowest alarm number during the execution of this instruction.

Application example for Alarms:

X0=forward switch X1=backward switch

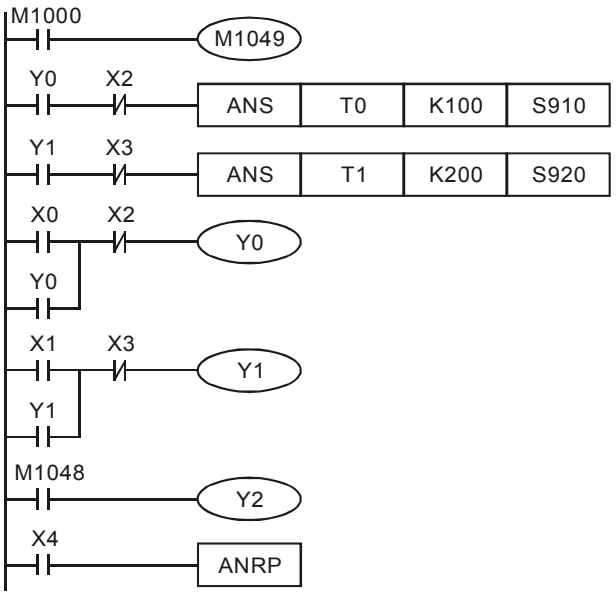
X2=front location switch X3=back location switch

X4=alarm device reset button

Y0=forward Y1=forward

Y2=alarm indicator

S910=forward alarm device S920=backward alarm



295. When M1049=ON, Alarms are enabled. If M1048=ON, an alarm has occurred, D1049=lowest alarm number.
296. If Y0=ON > 10 seconds and has not reached the front location X2, S910=ON.
297. If Y1=ON > 20 seconds and not reached the back location X3, S920=ON.
298. When X1=ON, and Y1=ON, and X3=ON, Y1 = OFF.
299. If an alarm occurs, alarm indicator Y2=ON.
300. Each alarm that is activated will be reset one by one, each time the reset button X4 = ON during the execution of this instruction. The lowest numbered alarm is reset every execution of this instruction.

API	Mnemonic			Operands	Function
48	D	SQR	P	**S, D**	Square Root

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			SQR, SQRP: 5 steps
D													*			DSQR, DSQRP: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

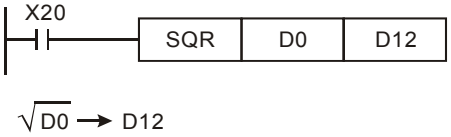
S: Source **D**: Destination to store result

Description:

301. Perform a square root on source **S** and store the result in **D**.
302. **S** can only be a positive value. Performing a square root operation on a negative value will result in an error and the instruction will not be executed. The error flag M1067 and M1068 = ON and D1067 records error code H0E1B.
303. SQR result **D** is calculated as an integer only, fractional values are ignored. If the result of the SQR is not a whole number, the Borrow flag M1021=ON.
304. When SQR result **D** = 0, the Zero flag M1020=ON.
305. Performing any square root operation (even on a calculator) on a negative number will result in an error. This will result in M1067, the Program Execution Error bit being set.

Program Example:

When X20=ON, SQR of D0 will be stored in D12.



$\sqrt{D0} \rightarrow D12$

API	Mnemonic			Operands	Function
49	D	FLT	P	**S, D**	Floating Point

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*	*	*			FLT, FLTP: 5 steps
D											*	*	*			DFLT, DFLTP: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Source **D**: Destination

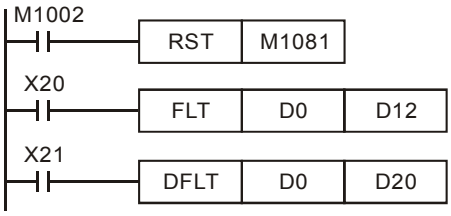
Description:

1. When M1081 = OFF, the source is converted from integer to floating point. **S** source = 16-bit and **D** = 32 bits.
   a) If the absolute value of the conversion result is larger than the maximum floating point value, the carry flag M1022=ON.
   b) If the absolute value of the conversion result is less than the minimum floating point value, the borrow flag M1021=ON.
   c) If conversion result is 0, zero flag M1020=ON.
2. When M1081 is ON, the source is converted from floating point to integer (ignore decimal points). **S** source = 32-bit and **D** Destination occupies 16-bit. If conversion result exceeds the integer range of **D** (16-bit, -32,768~32,767 and 32-bit, -2,147,483,648~2,147,483,647), **D** will hold either max or min value, and the carry flag will be set M1022=ON.
   a) If the decimal point was ignored, the borrow flag M1021=ON.
   b) If the conversion result = 0, zero flag M1020=ON.
3. ELCB-PB/ELC-PA/ELC-PVV1.2 don't support T,C registers.

Program Example 1:

306. When M1081 = OFF, the source data is converted from integer to floating point.
307. When X20 = ON, D0 (16-bit integer) is converted to D13, D12 (floating point).
308. When X21 = ON, D1, D0 (32-bit integer) is converted to D21, D20 (floating point).
309. If D0=K10 and X20=ON, the 32-bit floating point result will be H41200000 and it will be saved in the 32-bit register D13,D12

If the 32-bit register D1, D0=K100,000 and X21 = ON, The 32-bit of floating point result will be H47C35000 and it will be saved in the 32-bit register D21,D20.

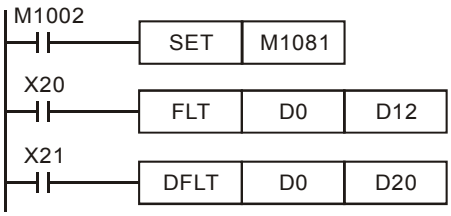


Program Example 2:

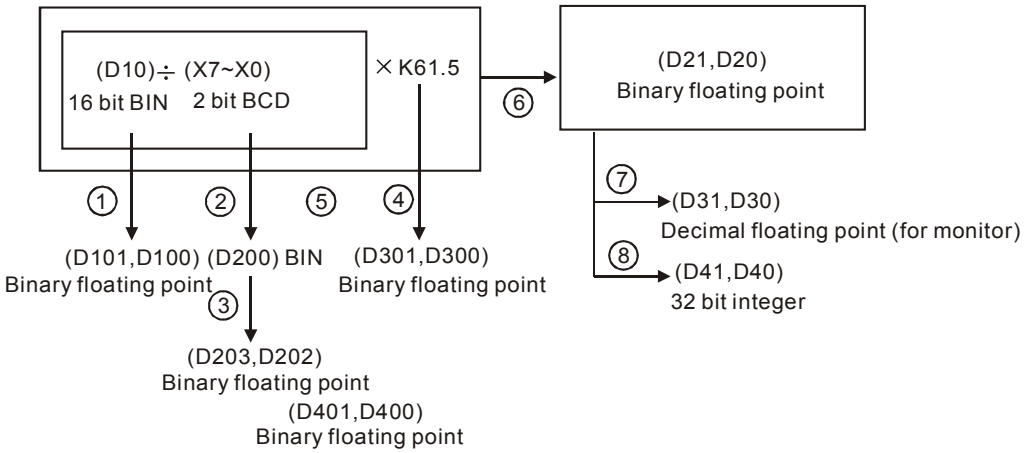
310. When M1081 = ON, the source data is converted from floating point to integer. (ignore the decimal point)

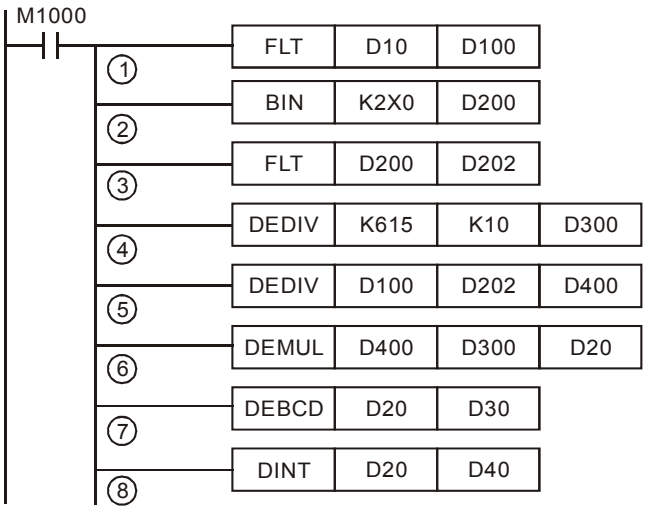
311. When X20 = ON, D1, D0 (floating point value) is converted to D12 (16-bit integer). If D1 (D0) =H47C35000, the floating point result is 100,000. The result will be D12=K32,767, M1022=ON, since the value exceeds the max value of the 16-bit register D12.

312. When X21 = ON, D1, D0 (floating point value) is converted to D21, D20 (32-bit integer). If D1, D0=H47C35000, the floating point result is 100,000. The result will be saved in 32-bit register D21, D20.



Program Example 3:





313. Covert D10 (16-bit integer) to D101, D100 (floating point).
314. Covert the value of X7~X0 (BCD value) to D200 (16-bit integer value).
315. Covert D200 (16-bit integer) to D203, D202 (floating point).
316. Save the result of K615 ÷ K10 to D301, D300 (floating point).
317. Divide the floating point:
     Save the result of (D101, D100) ÷ (D203, D202) to D401, D400 (floating point).
318. Multiply floating point:
     Save the result of (D401, D400) × (D301, D300) to D21, D20 (floating point).
319. Covert floating point (D21, D20) to decimal floating point (D31, D30).
320. Covert floating point (D21, D20) to 32-bit integer (D41, D40).

API	Mnemonic		Operands	Function
50	REF	P	**D, n**	Refresh I/O Immediately

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D	*	*														REF, REFP: 5 steps
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

D: Starting source of I/O **n**: Number of I/O to refresh

Description:

1. The Input image is updated at the end of each program scan. The output image is also sent to the outputs at the end of each program scan. If particular I/O points need to be updated during the program scan, the REF instruction can be used for this purpose.
2. **D** must always be a multiples of 10 (Y0, X10, Y20). **n** must always be a multiples of 8.
3. Range of **n**: 8 ~ 256 in multiples of 8. An error will be generated if **n** is out of range.
4. For ELC-PA, the input and output points processed by this instruction are the I/O points: X0~X7, Y0~Y7 and **n**=K8.
5. For ELCM-PH/PA, ELC2-PA/PB/PH/PE, Only the I/O points on the controller can be specified for operand D for I/O refresh.
   - When **D** specifies X0 and **n** ≦ 8, only X0~X7 will be refreshed. If **n** > 8, all I/O points on MPU will be refreshed.
   - When **D** specifies Y0 and **n** = 4, only Y0~X3 will be refreshed. If **n** > 4, all I/O points on the controller will be refreshed.
   - When **D** specifies X10 or Y10 I/O points on the controller starting from X10 or Y4 will all be refreshed regardless of **n** value.
6. For ELCM-PH/PA, Range for **n:** 4 ~ total I/O points on the controller. **n** should always be a multiple of 4.
7. For ELCM-PA and ELC2-PA MPU only: If M1180 = ON and REF instruction executes, the ELC will read the A/D value and update the read value to D1110~D1113. If M1181 = ON and REF instruction executes, the ELC will output the D/A value to D1116 and D1117 immediately. When A/D or D/A values are refreshed, the ELC will reset M1180 or M1181 automatically.

Program Example 1:

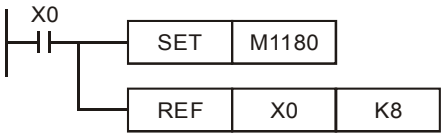
When X0 = ON, ELC will read the state of X0~X7 input points immediately. No input delay occurs.

Program Example 2:

When X0 = ON, the output signal Y0~Y7 (8 points) are sent to the output terminals immediately.

Program Example 3:

For ELCM-PA, ELC2-PA only: When X0 = ON and M1180 = ON, analog input values will be immediately read and placed into D1110~D1113. This is independent of the settings of operands **D** and **n**



API	Mnemonic		Operands	Function
51	REFF	P	n	Refresh and Filter Adjust

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
n					*	*										REFF, REFFP: 3 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

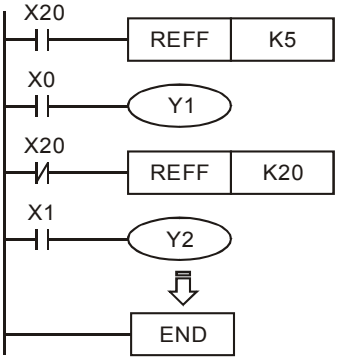
n: Response time setting, in ms

Description:

321. The ELC provides an input filter to prevent noise or interference from creating a false reading. Inputs X0~X17 are digitally filtered. REFF can adjust the response time of the input filters. D1020 (adjusts the filter time of X0~X7) and D1021 (adjusts the filter time of X10~X17).
322. When the ELC turns OFF → ON or the END instruction is reached, the response time is dictated by the values in D1020 and D1021.
323. During program execution, the values in D1020 and D1021 may be modified using the MOV instruction.
324. The response time can also be changed by using the REFF instruction in the program. When executed, the response time specified in the REFF instruction will be moved to both D1020 and D1021 and will take effect the next program scan.
325. Range of **n**: for ELC-PA, **n** = K0 ~ K20; for ELC-PV and ELC-PV2, **n** = K0 ~ K60, for ELCM-PH/PA and ELC2-PB/PH/PA/PE, **n** = K2 ~ K20.

Program Example:

326. When power to the ELC turns from OFF →ON, the filter time of X0~X7 inputs is dictated by the value in D1020 and D1021.
327. When X20=ON, REFF K5 instruction is executed, the response time will change to 5 ms and will take affect the next program scan.
328. When X20=OFF, the filter time will change to 20ms and will take affect the next program scan.



Points to note:

The response time is ignored (no delay set) If any external interrupts, high-speed counters or SPD instructions are used in the program, the filter times for the I/O associated with these instructions will be ignored.

API	Mnemonic	Operands	Function
52	MTR	**S, D_1, D_2, n**	Input Matrix

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*															MTR: 9 steps
D_1		*														
D_2		*	*	*												
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

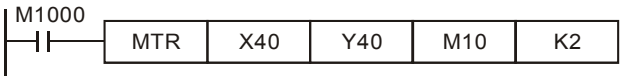
S: Starting address of input matrix **D_1**: Starting address of output matrix **D_2**: Corresponding starting address of matrix scan **n**: Number of banks for the matrix

Description:

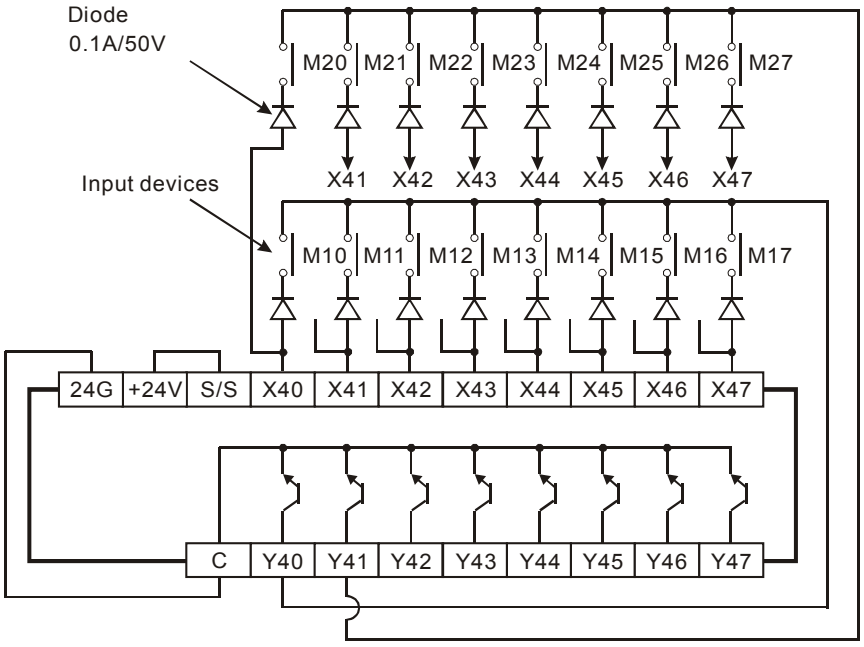
329. **S** is the starting address that specifies all inputs of the matrix. Once the input is specified, a selection of 8 continuous input addresses is called the “input matrix”. **D_1** is the starting address of the transistor outputs.
330. This instruction allows a selection of 8 continuous inputs to be used multiple (**n**) times. Each input has more than one input device wired to it. . Each set of 8 input signals are grouped into “rows” and there are **n** number of rows. Each row is selected by turning a different output on. The quantity of outputs from **D_1** is equal to the number of rows **n**. The results are stored in a matrix-table with a starting address specified by **D_2**.
331. The maximum number of inputs supported by this instruction is 64 (8 inputs x 8 rows).
332. When this instruction is used in an interrupt routine, each row of inputs will be processed every 25msec. This results in a matrix of 8 rows, i.e. 64 inputs (8 inputs x 8 rows) being read in 200msec. Hence, this instruction is not available for the input signal with an ON/OFF rate is more than 200ms.
333. It is recommended to condition a MTR rung with M1000 (normally open contact).
334. When the MTR instruction execution is complete, the instruction execution completed flag M1029 is turned ON. This flag is automatically reset when the MTR instruction is turned OFF.
335. This instruction can only be used ONCE in the program.
336. Flag: M1029, execution completed flag.

Program Example:

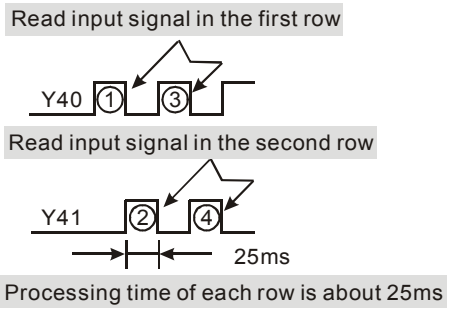
When the ELC is in the run mode, the MTR instruction starts to execute. The external 2 rows, total 16 devices are read in order and the results are stored in the internal addresses M10~M17, M20~M27.



The figure below is an example wiring diagram for the operation of the MTR instruction. The external 2 rows consist of 2 sets of inputs wired to X40~47 and Y40~41. A total of 16 addresses are used to store the states of the two rows of inputs: M10~M17, M20~M27. For a general precaution to aid in successful operation, diodes should be placed after each input devices. These diodes should have a rating of 0.1A, 50V.



When output Y40 is ON, only those inputs in the first row are read. These results are stored in addresses M10~M17. The second step involves Y40 going OFF and Y41 coming ON. Then only inputs in the second row are read. These results are stored in M20~M27.



Points to note:

1. Operand **S** must be a multiple of 10, i.e. X0, X10, X20… etc. and occupies 8 continuous input addresses.
2. Operand $\mathbf{D_1}$ must be a multiple of 10, i.e. Y0, Y10, Y20… etc. and occupies **n** continuous output addresses.

3. Operand **D_2** should be a multiple of 10, i.e. Y0, M10, S20… etc.

4. Valid range: **n**=2~8

API	Mnemonic		Operands	Function
53	D	HSCS	**S_1, S_2, D**	High Speed Counter Set

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		DHSCS: 13 steps
S_2												*				
D		*	*	*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Compare value **S_2**: High-speed counter number **D**: Compare result

Description:

337. All high-speed counters use an interrupt process; therefore all compare results **D** are updated immediately.

338. DHSCS instruction compares the current value of the selected high-speed counter **S_2** against a selected compare value **S_1**. When the counters current value equals **S_1**, then **D** = ON. Once set to ON if values **S_1** and **S_2** are not longer equal, **D** will still be ON.

339. If for example **D** is specified as Y0~Y7 and, when **S_1** and **S_2** are equal, the compare result will immediately energize output Y0~Y7... If M and S addresses are used, they are also immediately updated independent of the program scan. Interrupt pointers can also be used for **D**, to execute an interrupt subroutine when **S_1** and **S_2** are equal.

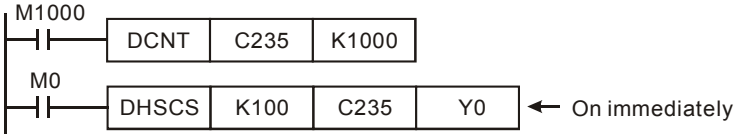
340. The **D** does not support E, F index registers.

341. Operand limitation:
Operand **S_2** must be high speed counter. For example: C235~C255.
Interrupt pointers I010 to I080 can be used for **D**, but ELCB-PB models do not support these interrupts.

342. Flags: M1289 ~ M1294 are used to inhibit interrupts for the high speed counters. See Program Example 3 for more details.

Program Example 1:

If M0=ON, the DHSCS instruction starts to operate. Y0 is turned ON immediately when C235's present value goes from 99→100 or from 101→100, and will remain on.

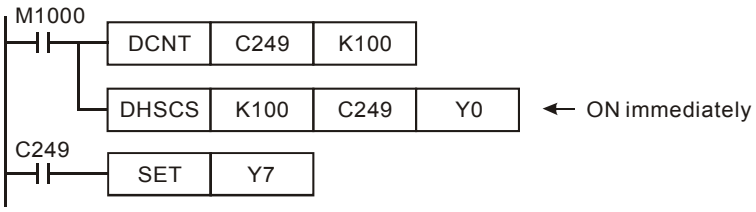


Program Example 2:

Difference between a Y output for DHSCS instructions and general Y output:

When C249's value changes from 99→100 or 101→100, Y0 output of DHSCS immediately energizes the output Y0 using the interrupt process which is independent of the scan time. However, there will still be a delay due to the output module: relay output delay: 10ms. Transistor output delay: 10us.

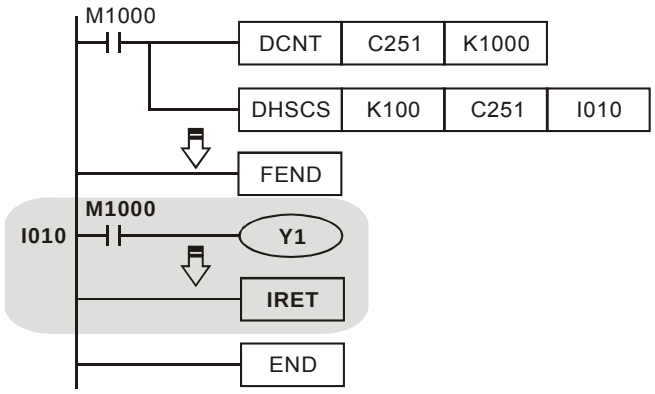
When the present value of high-speed counter C249 changes from 99 to 100, C249 will be activated, and Y7 will be ON, but the actual output Y7 will not be energized until after the program scan is complete. The same output delays for relay and transistor outputs apply as well.



Program Example 3:

High-speed counter interrupt:

343. ELC-PA models support the high-speed counter interrupt
344. When using the DHSCS instruction to execute an interrupt routine when **S_1** and **S_2** are equal, the specified high-speed counter can not be use in other DHSCS, DHSCR or DHSZ instructions. If it is used in any of these instructions, it will result in an error.
345. The interrupt pointers I010 to I060 can be used as the **D** operand for DHSCS instructions and this enables the interrupt routine to be executed when the value of the specified high-speed counter reaches the value in the DHSCS instruction.
346. For ELC-PA model, there are six high-speed counter interrupts: I010, I020, I030, I040, I050, I060 (6 points) that can be used. I010 is used with C235, C241, C244, C246, C247, C249, C251, C252, and C254. I020 is used with C236, C243; I030 is used with C237, C242; I040 is used with C245, C238; I050 is used with C239 and I060 is used with C240, C250 and C255.
347. When the present value of C251 changes from 99→100 or 101→100, the program will jump to the interrupt pointer I010 to execute the interrupt routine.



348. ELC-PA, ELCM-PH/PA, ELC2-PA/PB/PH/PE models, M1059 is the high-speed counter interrupt disable flag

349. ELC-PV, ELC2-PV models, M1289 ~ M1294 are the disable flags for high speed counter interrupts for I010 ~ I060, i.e. when M1294 = On, I060 interrupt will be disabled per the table below.

Interruption pointer I No.	Interrupt disable flag	Interruption pointer I No.	Interrupt disable flag
I010	M1289	I040	M1292
I020	M1290	I050	M1293
I030	M1291	I060	M1294

Notes for using the DHSCS Instruction with ELCM-PH/PA, ELC2-PA/PB/PH/PE controllers:

1. If **D** is specified as Y0~Y3, when the instruction is executed and the count value equals $\mathbf{S}_1$, the compare result will immediately energize Y0~Y3. All other outputs will be updated normally. Also, M and S devices, not affected by the program scan time, will be immediately updated.
2. Operand **D** can use the following interrupts: I0□0, □=1~8
3. High speed counters include software high speed counters and hardware high speed counters. In addtiion, there are also two types of comparators including software comparators and hardware comparators.
4. Software comparators:

a) There are 6 software comparators available that are associated with high speed counter interrupts.

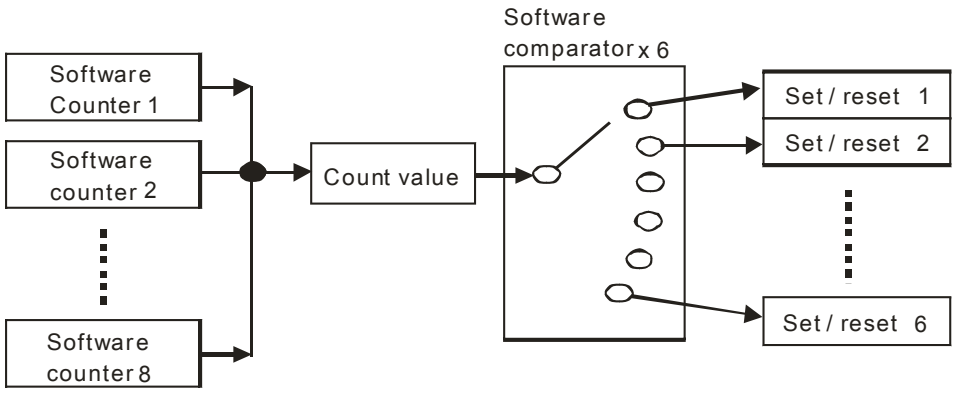
b) When programming the DHSCS and DHSCR instructions, the total of Set/Reset comparisons for both instructions can not be more than 6, otherwise a syntax check error will occur.

c) Table of software counters and software comparators:

Counter	C232	C233	C234	C235	C236	C237
DHSCS Hi-speed interrupt	I010	I050	I070	I010	I020	I030
Hi-speed compare Set/Reset	C232~C242 share 6 software comparators					

Counter	C238	C239	C240	C241	C242
DHSCS Hi-speed interrupt	I040	I050	I060	I070	I080
Hi-speed compare Set/Reset	C232~C242 share 6 software comparators				

d) Block diagram of software counters and comparators:



5. Hardware comparators:

a) There are 2 groups of hardware comparators provided respectively for 2 groups of hardware counters (A group and B group). Each group shares 4 comparators with individual Compare Set/Reset functions.

b) When programming the DHSCS and DHSCR instructions, the total of Set/Reset comparisons for both instructions can not be more than 4, otherwise a syntax check error will occur.

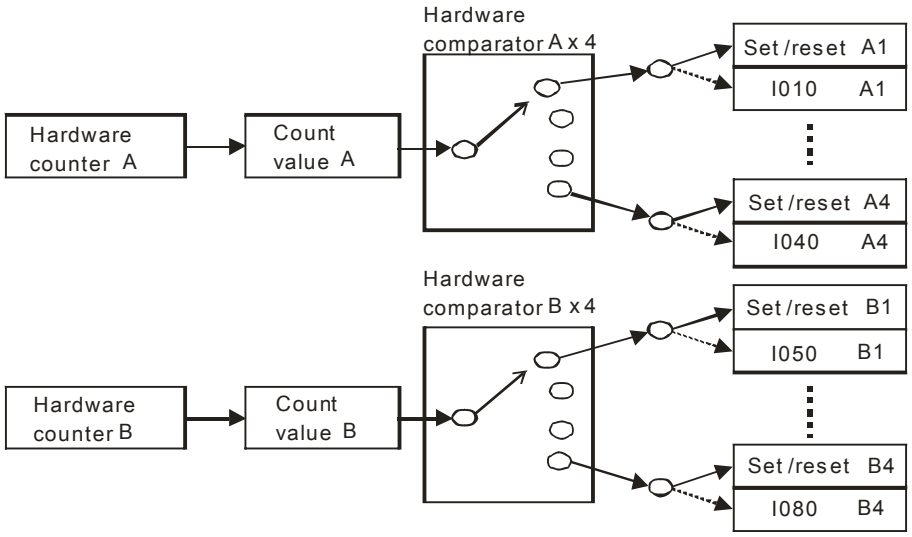
c) Each high-speed counter interrupt occupies an associated hardware comparator; consequently the interrupt number can not be repeated. Also, I010~I040 can only be applied for group A comparators and I050~I080 for group B.

d) If DCNT instruction enables C243 as high speed counter (group A) and DHSC/DHSC instruction uses C245 as high speed counter (group A) at the same time, the ELC takes C243 as the source counter automatically and no syntax check error will occur.

e) Table of settings for hardware counters and comparators:

Hardware counter	A group				B group			
	A1	A2	A3	A4	B1	B2	B3	B4
Counter No.	C243, C245~C248, C251,C252				C244, C249, C250, C253, C254			
High-speed counter interrupt	I010	I020	I030	I040	I050	I060	I070	I080
Hi-speed compare Set/Reset	Share 4 hardware comparators for group A				Share 4 hardware comparators for group B			

f) Block diagram of hardware counters and comparators:



6. Differences between software and hardware comparators:

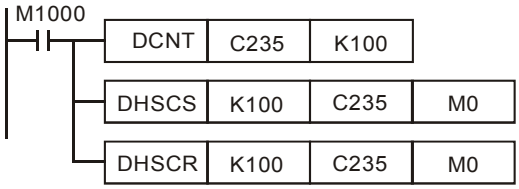
a) 6 comparators are available for software counters while 8 comparators are available for 2 groups of hardware counters (4 comparators for each group)

b) Output timing of software comparator → count value equals the compare value in both count up and down modes.

c) Output timing of hardware comparator → count value equals the compare value+1 in count-up mode; count value equals the compare value -1 in count-down mode.

Program Example 4:

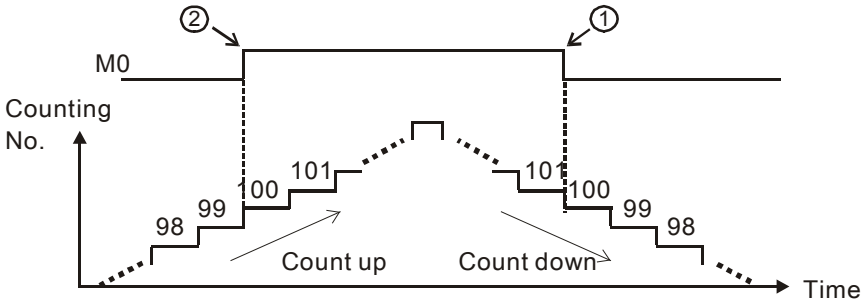
1. Set/reset M0 by utilizing the software comparator



2. When the accumulated value in C235 increments from 99 to100, the DHSCS instruction sets M0 ON. (M1235 = OFF, C235 counts up)

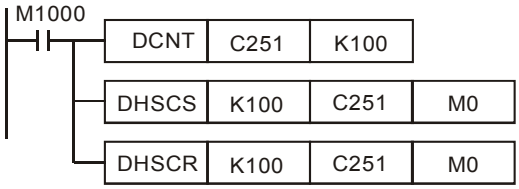
3. When the accumulated value in C235 decrements from 101 to100, the DHSCR instruction resets M0. (M1235 = ON, C235 counts down)

4. Timing diagram for the comparison:

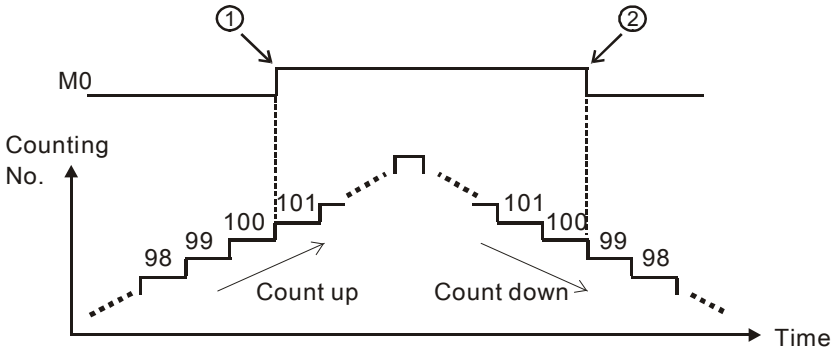


Program Example 5: for ELCM-PH/PA version 1.00

1. Set/reset M0 by utilizing the hardware comparator

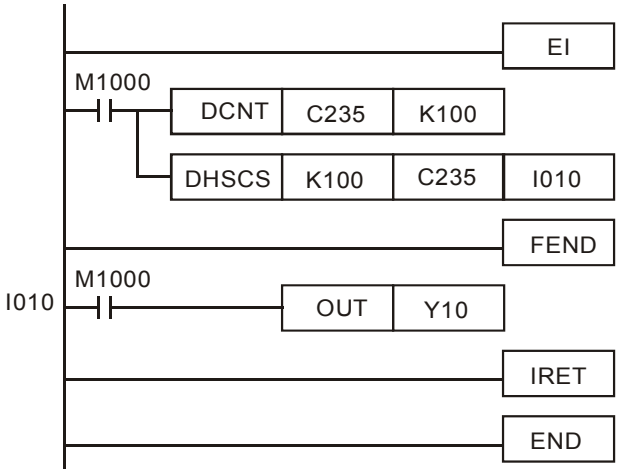


2. When the accumulated value in C251 increments from 100 to101, the DHSCS instruction sets M0 ON.
3. When the accumulated value in C251 decrements from 100 to 99, DHSCR instruction resets M0.
4. Timing diagram for the comparison:



Program Example 6:

1. Executing an interrupt subroutine with a software comparator.
2. When the accumulative value in C235 increments from 99 to100, the interrupt subroutine I010 executes immediately to set Y0 ON.



Points to note:

1. The output contact of a high-speed counter and the compare outputs of the DHSCS instruction, the DHSCR instruction and the DHSZ instruction will all operate normally when there are counted inputs. If instructions such as, DADD, DMOV…etc. are used to change the present value of a high-speed counter to equal the preset or compare value, the output will not energize because there are no counted inputs.
2. High-speed counters for ELCB-PB series models:

   ELCB-PB models: total counting frequency (X0~X3) is 20 KHz.

Type / Input	1-phase 1 input							1-phase 2 inputs			2-phase 2 inputs		
	C235	C236	C237	C238	C241	C242	C244	C246	C247	C249	C251	C252	C254
X0	U/D				U/D		U/D	U	U	U	A	A	A
X1		U/D			R		R	D	D	D	B	B	B
X2			U/D			U/D			R	R		R	R
X3				U/D		R	S			S			S

U: Increasing input
D: Decreasing input
A: A phase input
B: B phase input
S: Start input
R: Reset input

Input points X0 and X1 can be used as high speed counter inputs for 1-phase 1 input up to 20KHz. Input point X2 and X3 can be used as high speed counter inputs for 1-phase 1 input up to 10KHz. But total counting frequency of these input points should be less than or equal to total frequency 20KHz. If the input is 2-phase 2 inputs signal, the frequency will be four times the counting frequency. Therefore, the counting frequency of 2-phase 2 inputs is 4KHz. In ELCB-PB models, DHSCS and DHSCR instruction cannot be used more than 4 times.

3. High-speed counter provided in ELC-PA series models:

ELC-PA models: total counting frequency is 30 KHz

Type / Input	1-phase 1 input									1-phase 2 inputs			2-phase 2 inputs			
	C235	C236	C237	C238	C239	C240	C241	C242	C244	C246	C247	C249	C251	C252	C253	C254
X0	U/D						U/D		U/D	U	U	U	A	A	B	A
X1		U/D					R		R	D	D	D	B	B	A	B
X2			U/D					U/D			R	R		R		R
X3				U/D				R	S			S				S
X4					U/D											
X5						U/D										

U: Increasing input
D: Decreasing input
A: A phase input
B: B phase input
S: Start input
R: Reset input

Input points X0 and X1 can be high speed counter inputs for 1-phase 1 input up to 30KHz. Input points X2 ~ X5 can be high speed counter inputs for 1-phase 1 input up to 10KHz. But the total counting frequency of these input points should be less than or equal to the total frequency 30KHz. If the input is 2-phase 2 inputs signal, the frequency will be four times the counting frequency. Therefore, the counting frequency of 2-phase 2 inputs (C251, C252, C254) is 4KHz and the counting frequency of 2-phase 2 inputs (C253) is 25KHz.

In ELC-PA series models, DHSCS, DHSCR and DHSZ instruction cannot be used more than 6 times.

4. High-speed counter provided in ELC-PV, ELC2-PV series Models:

ELC-PV, ELC2-PV series supports high speed counters. C235 ~ C240 are program-interrupt 1-phase high speed counters with a total bandwidth of 20KHz. They can be used alone with a counting frequency of up to 10KHz. C241 ~ C254 are hardware high speed counters (HHSC). There are four HHSC in ELC-PV series, HHSC0 ~ 3. The pulse input frequency of HHSC0 and HHSC1 can be a maximum of 200KHz and that of HHSC2 and HHSC3 a maximum of 20KHz (1 phase or A-B phase). The pulse input frequency of HHSC0 ~ 3 can reach 200KHz, among which:

C241, C246 and C251 share HHSC0

C242, C247 and C252 share HHSC1

C243, C248 and C253 share HHSC2

C244, C249 and C254 share HHSC3

a) Every HHSC can only be assigned to one counter with the DCNT instruction.

b) There are three counting modes in every HHSC (see the table below):

i) 1-phase 1 input refers to “pulse/direction” mode.

ii) 1-phase 2 inputs refers to “clockwise/counterclockwise (CW/CCW)” mode.

iii) 2-phase 2 inputs refers to “A-B phase” mode.

Counter type	Program-interruption high speed counter						Hardware high speed counter											
Type / Input	1-phase 1 input						1-phase 1 input				1-phase 2 inputs				2-phase 2 inputs			
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C246	C247	C248	C249	C251	C252	C253	C254
X0	U/D						U/D				U				A			
X1		U/D									D				B			
X2			U/D				R				R				R			
X3				U/D			S				S				S			
X4					U/D			U/D				U				A		
X5						U/D						D				B		
X6								R				R				R		
X7								S				S				S		
X10									U/D				U				A	
X11													D				B	
X12									R				R				R	
X13									S				S				S	
X14										U/D				U				A
X15														D				B
X16										R				R				R
X17										S				S				S

U: Progressively increasing input

A: A phase input

S: Input started

B: Progressively decreasing input

B: B phase input

R: Input cleared

c) In the ELC-PV, ELC2-PV series, there is no limit on the number of times the hardware high speed counter related instructions, DHSCS, DHSCR and DHSZ can be used. However, when these instructions are enabled at the same time, there will be some limitations. DHSCS instruction will occupy 1 group of settings, DHSCR 1 group of settings and DHSZ 2 groups of settings. These three instructions cannot occupy 8 groups of settings in total; otherwise the system will ignore the instructions which are not the first scanned and enabled.

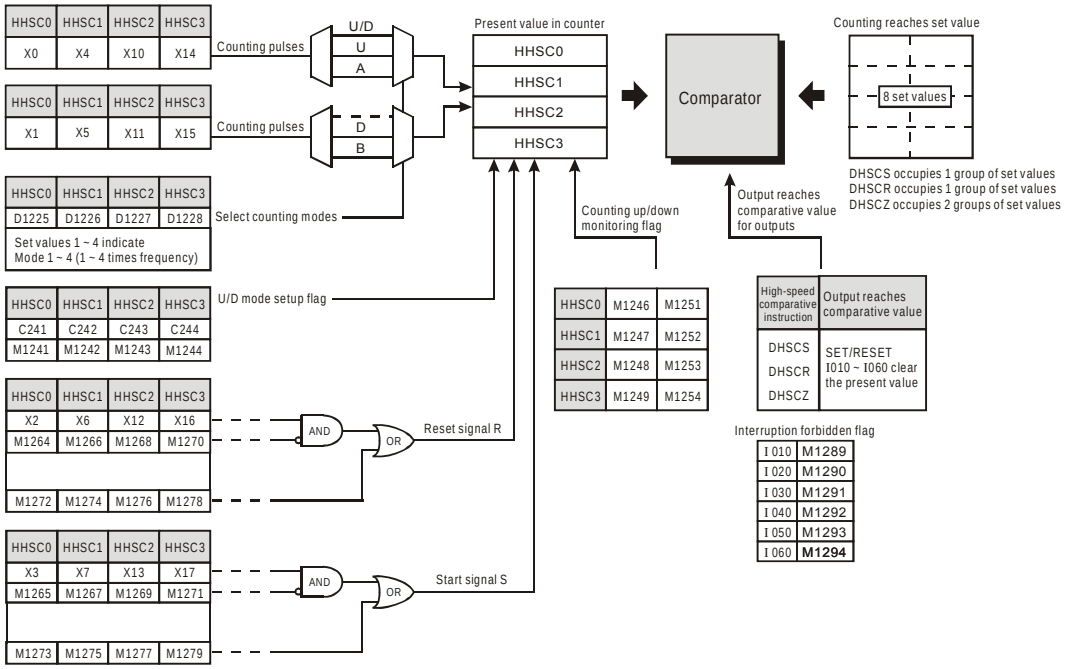
d) System structure of the hardware high speed counters:

i) HHSC0 ~ 3 have reset signals and start signals from external inputs. M1272, M1274, M1276 and M1278 are reset signals for HHSC0, HHSC1, HHSC2 and HHSC3. M1273, M1275, M1277 and M1279 are start signals for HHSC0, HHSC1, HHSC2 and HHSC3.

ii) If the external control signal inputs of R and S are not in use, you can set M1264/M1266/M1268/M1270 and M1265/M1267/M1269/M1271 as True and disable the input signals. The corresponding external inputs can be used again as general input points (see the figure below).

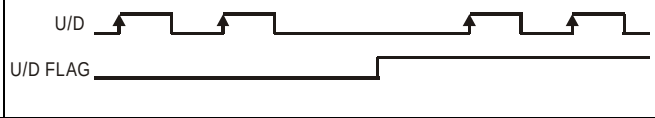
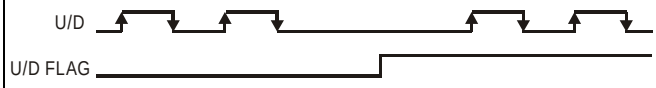
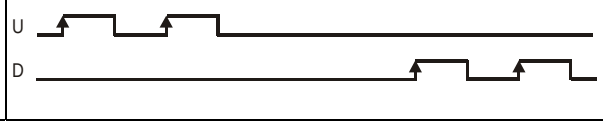
iii) When special M is used as a high speed counter, the inputs controlled by START

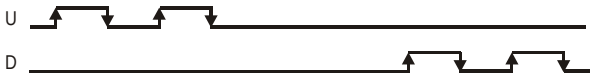
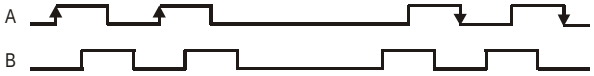
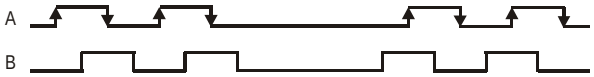
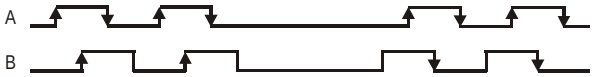
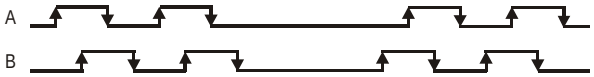
and RESET will be affected by the scan time.



e) Counting modes:

Special registers D1225 ~ D1228 are for setting up different counting modes for the hardware high speed counters (HHSC0 ~ 3) in ELC-PV, ELC2-PV series. These are normally 4 times the frequency for counting, with a default setting that is double the frequency.

Counting modes		Wave pattern
Type	Set value in special D	Counting up(+1) Counting down(-1)
1-phase 1 input	1 (Normal frequency)	
	2 (Double frequency)	
1-phase 2 inputs	1 (Normal frequency)	

Counting modes		Wave pattern
Type	Set value in special D	Counting up(+1) Counting down(-1)
	2 (Double frequency)	
2-phase 2 inputs	1 (Normal frequency)	
	2 (Double frequency)	
	3 (Triple frequency)	
2-phase 2 inputs	4 (4 times frequency)	

5. High-speed counter provided in ELCM-PH/PA, ELC2-PA/PB/PH/PE series Models:

There are two types of high speed counters provided by ELCM and ELC2 including Software High Speed Counter (SHSC) and Hardware High Speed Counter (HHSC). The same Input point (X) can be used with only one high speed counter. Using the same input with 2 high speed counters will result in a syntax error when executing DCNT instruction.

Applicable Software High Speed Counters:

C / X	1-phase input								2 phase 2 input		
	C235	C236	C237	C238	C239	C240	C241	C242	C232	C233	C234
X0	U/D								A		
X1		U/D									
X2			U/D						B		
X3				U/D							
X4					U/D					A	
X5						U/D				B	
X6							U/D				A
X7								U/D			B

R/F	M1270	M1271	M1272	M1273	M1274	M1275	M1276	M1277	-	-	-
U/D	M1235	M1236	M1237	M1238	M1239	M1240	M1241	M1242	-	-	-

U: Count up D: Count down A: Phase A input B: Phase B input

Note:

a) U/D (Count up/Count down) can be specified by special M bit. OFF = count up; ON = count down.

b) R/F (Rising edge trigger/ Falling edge trigger) can also be specified by special M bit. OFF = Rising; ON = Falling.

c) SHSC supports max 10kHz input pulse on single input point. Max. 8 counters are allowed at the same time.

d) For 2-phase 2-input counting, (X4, X5) (C233) and (X6, X7) (C234), max 5kHz. (X0,X2) (C232), max 15kHz.

e) 2-phase 2-input counting supports double and 4 times frequency, which is selected in D1022 as follows:

Applicable Hardware High Speed Counters:

C / X	1-phase input		1-phase 2-input						2-phase 2-input			
	C243	C244	C245	C246	C247	C248	C249	C250	C251	C252	C253	C254
X0	U		U/D	U/D	U	U			A	A		
X1	R		Dir	Dir	D	D			B	B		
X2		U					U/D	U/D			A	A
X3		R					Dir	Dir			B	B
X4				R		R				R		
X5								R				R

U: Count up A: Phase A input Dir: Direction signal input

D: Count down B: Phase B input R: Reset signal input

Note:

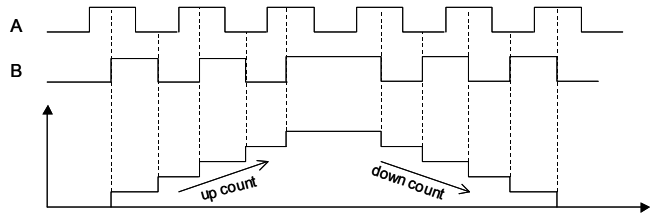
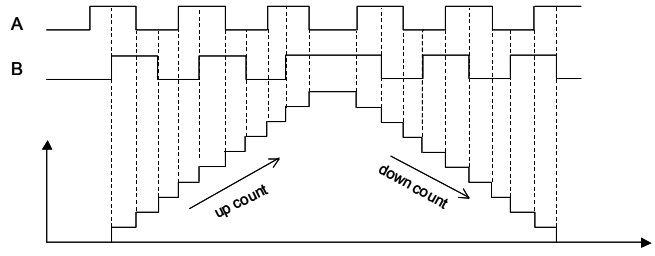
a) The max. frequency of the 1-phase input X0 (C243, C245, C246) and X2(C242) is 100kHz.

b) The max. frequency of the 1-phase 2-input (X0, X1) (C245, C246) and (X2, X3) (C249, C250) is 100kHz.

c) The max. frequency of the 1-phase 2-input (X0, X1) (C247, C248) is 10kHz.

d) The max. frequency of the 2-phase 2-input (X0, X1) (C251, C252) and (X2, X3) (C253, C254) is 5kHz.

e) 2-phase 2-input counting supports double and 4 times frequency, which is selected in D1022 as the table in next page. Please refer to the below table for detailed counting wave form.

D1022	Counting mode
K2 (Double Frequency)	
K4 or other value (4 times frequency) (Default)	

f) C243 and C244 support count-up mode only and occupy input points X1 and X3 as reset

("R") functions. If the reset function is not needed, set the special M relays ON (M1243 and M1244) to disable the reset function.

g) "Dir" refers to direction control function. OFF indicates counting up; ON indicates counting down.

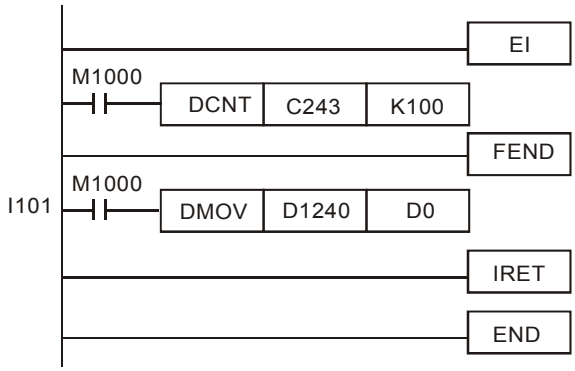
h) When X1, X3, X4 and X5 are applied for the reset function and the external interrupts are disabled, users can define the reset function as Rising/Falling-edge triggered by special M bits.

Reset Function	X1	X3	X4	X5
R/F	M1271	M1273	M1274	M1275

i) When X1, X3, X4 and X5 are applied for the reset function and external interrupts are disabled, users can define the reset function as Rising/Falling-edge triggered by special M bits specified in the table: Applicable Software High Speed Counters. However, if external interrupts are applied, the interrupt instructions have the priority in using the input points. In addition, the ELC will move the current data in the counters to the data registers below then reset the counters.

Special D	D1241, D1240				D1243, D1242		
Counter	C243	C246	C248	C252	C244	C250	C254
External Interrupt	X1	X4			X3	X5	

- When X0 (counter input) and X1 (external Interrupt I100/I101) work with C243, the count value will be moved to D1240 and D1241 when the interrupt occurs. Then the counter will be reset.
- When X2 (counter input) and X3 (external Interrupt I300/I301) work with C244, the count value will be moved to D1242 and D1243 when the interrupt occurs. Then the counter will be reset.
- When X0 (counter input) and X4 (external Interrupt I400/I401) work with C246, C248, C252, the count value will be moved to D1240 and D1241 when the interrupt occurs, Then the counter will be reset.
- When X2 (counter input) and X5 (external Interrupt I500/I501) work with C244, C250, C254, the count value will be moved to D1242 and D1243 when the interrupt occurs. Then the counter will be reset.
- Example: When C243 is counting and an external interrupt is triggered from X1(I101), the count value in C243 will be moved to D0 immediately then C243 is reset. Then, interrupt I101 executes.



6. Special registers for relevant flags and settings of high speed counters:

Flag	Function
M1059	Disable high-speed counter interruptions I010~I080
M1150	DHSZ instruction in multiple set values comparison mode
M1151	The execution of DHSZ multiple set values comparison mode is completed.
M1152	Set DHSZ instruction as frequency control mode
M1153	DHSZ frequency control mode has been executed.
M1232 ~ M1245	Designating the counting direction of high speed counters C232 ~ C245 When M12□□ = Off, C2□□ will count up. When M12□□ = On, C2□□ will count down.
M1246 ~ M1255	Monitor the count direction of high speed counters C246 ~ C255 When M12□□ = Off, C2□□ will count up. When M12□□ = On, C2□□ will count down.
M1260	ELC-PA: X5 is the reset input signal of the global reset of C235~C239 ELCM-PH/PA, ELC2-PA/PB/PH/PE: Let X7 be the reset input signal of high-speed counters C235~C241
M1261	High-speed comparison flag for DHSCR instruction
M1264	Disable the external control signal input point of HHSC0 reset signal point (R)
M1265	Disable the external control signal input point of HHSC0 start signal point (S)
M1266	Disable the external control signal input point of HHSC1 reset signal point (R)
M1267	Disable the external control signal input point of HHSC1 start signal point (S)
M1268	Disable the external control signal input point of HHSC2 reset signal point (R)
M1269	Disable the external control signal input point of HHSC2 start signal point (S)
M1270	ELC-PV, ELC2-PV: Disable the external control signal input point of HHSC3 reset signal point (R)

Flag	Function
M1271	ELCM-PH/PA, ELC2-PA/PB/PH/PE: C235 counting mode setting (ON: falling-edge count) ELC-PV, ELC2-PV: Disable the external control signal input point of HHSC3 start signal point (S) ELCM-PH/PA, ELC2-PA/PB/PH/PE: C236 counting mode setting (ON: falling-edge count)
M1272	ELC-PV, ELC2-PV: Internal control signal input point of HHSC0 reset signal point (R) ELCM-PH/PA, ELC2-PA/PB/PH/PE: C237 counting mode setting (ON: falling-edge count)
M1273	ELC-PV, ELC2-PV: Internal control signal input point of HHSC0 start signal point (S) ELCM-PH/PA, ELC2-PA/PB/PH/PE: C238 counting mode setting (ON: falling-edge count)
M1274	ELC-PV, ELC2-PV: Internal control signal input point of HHSC1 reset signal point (R) ELCM-PH/PA, ELC2-PA/PB/PH/PE: C239 counting mode setting (ON: falling-edge count)
M1275	ELC-PV, ELC2-PV: Internal control signal input point of HHSC1 start signal point (S) ELCM-PH/PA, ELC2-PA/PB/PH/PE: C240 counting mode setting (ON: falling-edge count)
M1276	ELC-PV, ELC2-PV: Internal control signal input point of HHSC2 reset signal point (R) ELCM-PH/PA, ELC2-PA/PB/PH/PE: C241 counting mode setting (ON: falling-edge count)
M1277	ELC-PV, ELC2-PV: Internal control signal input point of HHSC2 start signal point (S) ELCM-PH/PA, ELC2-PA/PB/PH/PE: C242 counting mode setting (ON: falling-edge count)
M1278	Internal control signal input point of HHSC3 reset signal point (R)
M1279	Internal control signal input point of HHSC3 start signal point (S)
M1289	High speed counter I010 interruption forbidden

Flag	Function
M1290	High speed counter I020 interruption forbidden
M1291	High speed counter I030 interruption forbidden
M1292	High speed counter I040 interruption forbidden
M1293	High speed counter I050 interruption forbidden
M1294	High speed counter I060 interruption forbidden
M1312	C235 Start input point control
M1313	C236 Start input point control
M1314	C237 Start input point control
M1315	C238 Start input point control
M1316	C239 Start input point control
M1317	C240 Start input point control
M1320	C235 Reset input point control
M1321	C236 Reset input point control
M1322	C237 Reset input point control
M1323	C238 Reset input point control
M1324	C239 Reset input point control
M1325	C240 Reset input point control
M1328	Enable Start/Reset of C235
M1329	Enable Start/Reset of C236
M1330	Enable Start/Reset of C237
M1331	Enable Start/Reset of C238
M1332	Enable Start/Reset of C239
M1333	Enable Start/Reset of C240

Special D	Function
D1022	Multiplied frequency of A-B phase counters for ELC-PA, ELCB-PB, ELCM-PH/PA, ELC2-PA/PB/PH/PE series
D1180 (LW) D1181 (HW)	ELC-PA：When interrupt X2 / I201 occurs, D1180 will store the low word of high speed counting value at X0, D1181 will store the high word of high speed counting value at X0.
D1198 (LW)	ELC-PA：When interrupt X3 / I301 occurs, D1198 will store the low word of

D1199 (HW)	high speed counting value at X0, D1199 will store the high word of high speed counting value at X0.
D1225	The counting mode of the 1st group counters (C241, C246, C251)
D1226	The counting mode of the 2nd group counters (C242, C247, C252)
D1227	The counting mode of the 3rd group counters (C243, C248, C253)
D1228	The counting mode of the 4th group counters (C244, C249, C254)

Special D	Function
D1225 ~ D1228	Counting modes of HHSC0 ~ HHSC3 in ELC-PV, ELC2-PV series (default = 2) 1: Normal frequency counting mode 2: Double frequency counting mode 3: Triple frequency counting mode 4: 4 times frequency counting mode
D1240	When interrupt I400/I401/I100/I101 occurs, D1240 stores the low Word of the high-speed counter.
D1241	When interrupt I400/I401/I100/I101 occurs, D1241 stores the high Word of the high-speed counter.
D1242	When interrupt I500/I501/I300/I301 occurs, D1242 stores the low Word of the high-speed counter.
D1243	When interrupt I500/I501/I300/I301 occurs, D1243 stores the high Word of the high-speed counter.

API	Mnemonic	Operands	Function
54	D HSCR	S_1, S_2, D	High Speed Counter Reset

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		DHSCR: 13 steps
S_2												*				
D		*	*	*								*				

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Compare value **S_2**: High-speed counter number **D**: Compare result

Description:

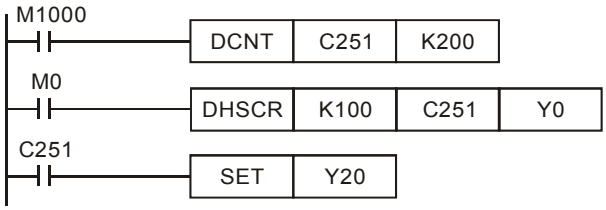
350. DHSCR compares the current value of the counter **S_2** against a compare value **S_1**. When the counters current value changes to a value equal to **S_1**, address **D** is reset to OFF. Once reset, even if the compare result is no longer equal, **D** will remain OFF.
351. If **D** is specified as Y0~Y7 (ELCM-PH/PA: Y0~Y3), when the compare value and the present value of the high-speed counter are equal, the compare result will immediately de-energize the external output Y0~Y7 (ELCM-PH/PA: Y0~Y3). . M and S addresses are also allowed for **D.**
352. Operand S_2 must be high speed counter. For example: C235~C255.
353. Operand D of ELC-PA, ELCB-PB cannot use the counter addresses**.**

354. Operand D of other series can use same with Operand S_2.

Program Example 1:

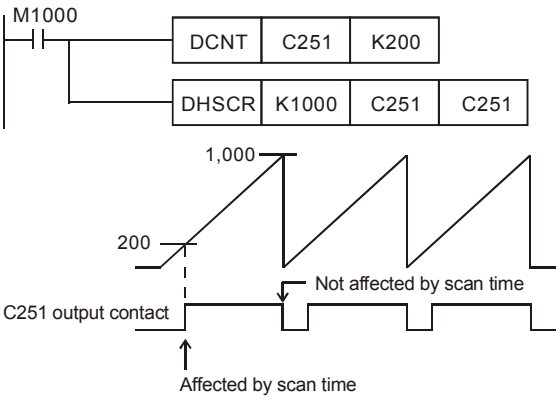
When M0=ON and C251's present value goes from 99→100 or 101→100, Y0 will be reset to OFF.

When C251's present value changes from 199 to 200, the contact C251 will be ON and Y20=ON.



- Program Example 2:

When DHSCR instruction uses the same high speed counter number as a DCNT instruction, and when the present value in the high speed counter C251 changes from 999 to 1,000 or 1,001 to 1,000, C251 will be reset.

- 

Remarks:

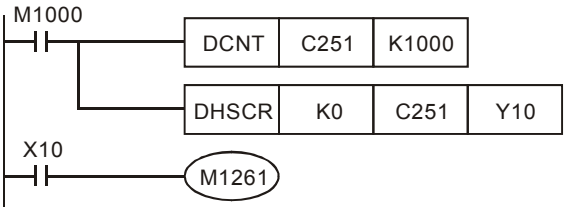
1. All ELC series controllers support high speed counters. For the limitatios on the use of the high speed counter instructions, see the remarks for API 53 DHSCS for more details.
2. M1261 of ELC-PV, ELC2-PV series designates the external reset modes of the high speed counter. Some high speed counters have input points for external reset; therefore, when the input point is On, the present value in the corresponding high speed counter will be cleared to 0 and the output contact will be Off. If the reset needs to be executed immediately by the external input, you must set M1261 ON.
3. M1261 can only be used in the hardware high speed counters C241 ~ C255.
4. Example:

a) X2 is the input for the external reset of C251.

b) Assume Y10 = On.

c) When M1261 = Off and X2 = On, the present value in C251 will be cleared to 0 and the contact of C251 will be Off. When the DHSCR instruction is executed, there will be no counting input pulses and the comparison result will not reset Y10. The external output will not execute the reset; so Y10 = On will remain unchanged.

d) When M1261 = On and X2 = On, the present value in C251 will be cleared to 0 and the contact of C251 will be Off. When DHSCR instruction is executed, there will be no counting the input pulses but the comparison result will occur. Therefore, Y10 will be reset.



API	Mnemonic		Operands	Function
55	D	HSZ	S_1, S_2, S, D	HSC Zone Compare

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		DHSZ: 17 steps
S_2					*	*	*	*	*	*	*	*	*	*		
S												*				
D		*	*	*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Low-limit value of zone comparison **S_2**: High-limit value of zone comparison **S**: high-speed counter number **D**: compared result (occupies 3 continuous bit addresses)

Description: (ELC-PA/PV, ELC2-PV)

355. **S_1** should be equal to or less than **S_2** (**$S_1 \leqq S_2$**).
356. If **D** is specified as Y0~Y7, when **S_1** and **S_2** are equal, the compare result will immediately energize the external output Y0~Y7. M and S devices are immediately set, not being affected by the scan cycle.
357. All outputs with the zone comparison use interrupts.
358. Operand **S** should be high speed counter. For example: C235~C255.
359. Flags: M1059~M1260 (Refer to the DHSCS instruction for more details)

Description: (ELCM-PH/PA, ELC2-PA/PB/PH/PE)

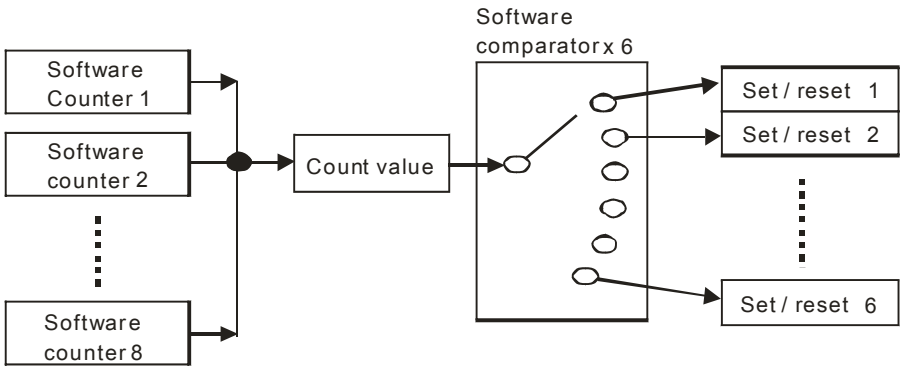
360. If **D** is specified as Y0~Y3 in this instruction, the compare result will immediately energize the external outputs Y0~Y3. M and S devices, not affected by the program scan cycle, will be updated immediately.
361. High speed counters include software high speed counters and hardware high speed counters. In addtiion, there are also two types of comparators including software comparators and hardware comparators.
362. Explanations on software comparators for DHSZ instruction

    a) Corresponding table for software counters and comparators:

Counter	C232	C233	C234	C235	C236	C237	C238	C239	C240	C241	C242
Hi-speed compare Set/Reset	Share 6 software comparators										

•

b) Block diagram of software counters and comparators:



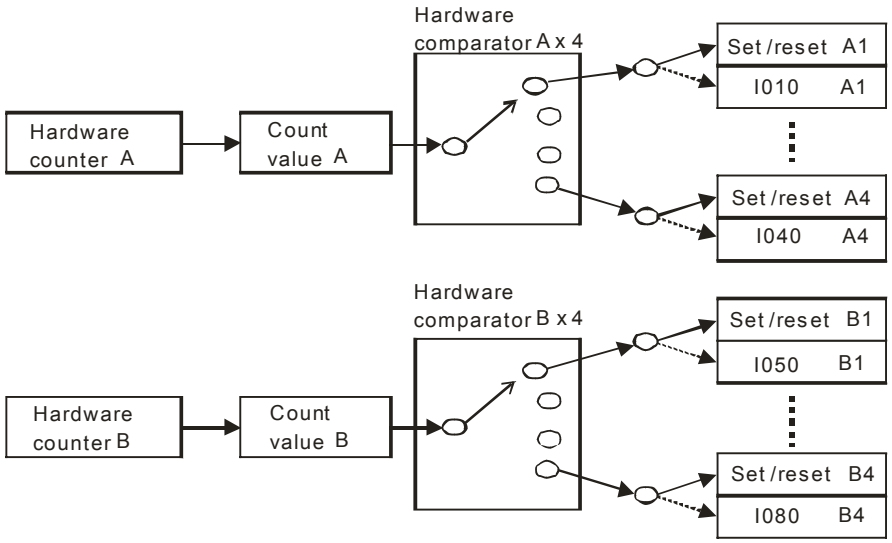
c) There are 6 software zone comparators available exclusively for zone compare operations. The limit of 6 comparisons for zone compares does not include the comparisons of DHSCS and DHSCR.

363. Descriptions of hardware comparators for the HSZ instruction:

a) Corresponding table for hardware counters and comparators

Hardware counter	A group				B group			
	A1	A2	A3	A4	B1	B2	B3	B4
Counter No.	C243, C245~C248, C251,C252				C244, C249, C250, C253, C254			
Hi-speed compare Set/Reset	Shares 4 hardware comparators for group A				Shares 4 hardware comparators for group B			

b) Block diagram of hardware counters and comparators:

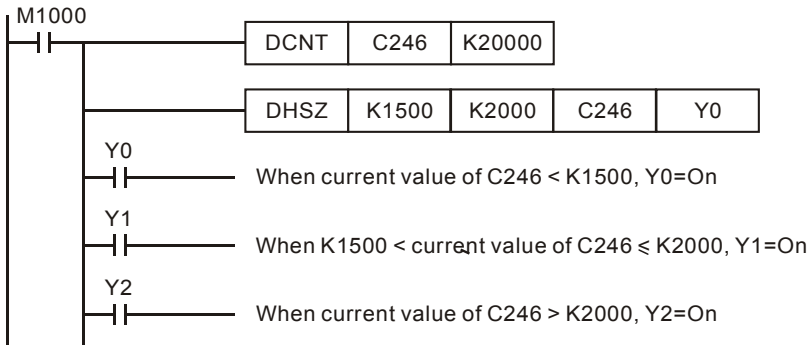


c) The two groups can only be used once for each group, occupying 2 comparators. For example, when DHSZ instruction uses A3 and A4 of group A comparators, only the other 2 comparators (A1, A2) are available for DHSCS and DHSCR instructions.

d) When DHSCS uses I030 or I040, comparators A3 and A4 are no longer available for DHSZ instruction. Also, when DHSCS uses I070 or I080, comparators B3 and B4 are no longer available for DHSZ instruction. If comparators are used repeatedly, the syntax error will be detected on the previous instruction..

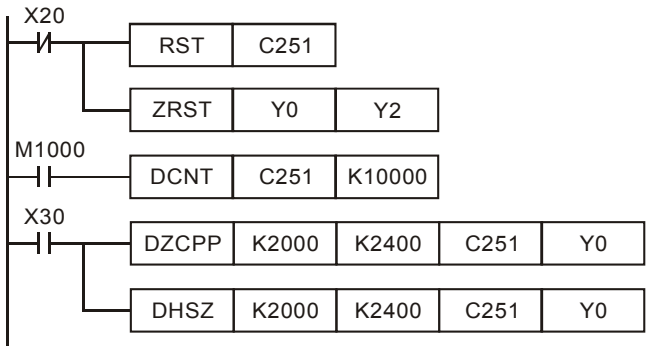
Program Example 1:

364. When **D** is specified as Y0, then Y0~Y2 will also be used..
365. When the DHSZ instruction is executed and the high-speed counter C246 is counting, if the high or low limit value is reached, one of Y0~Y2 will be ON.

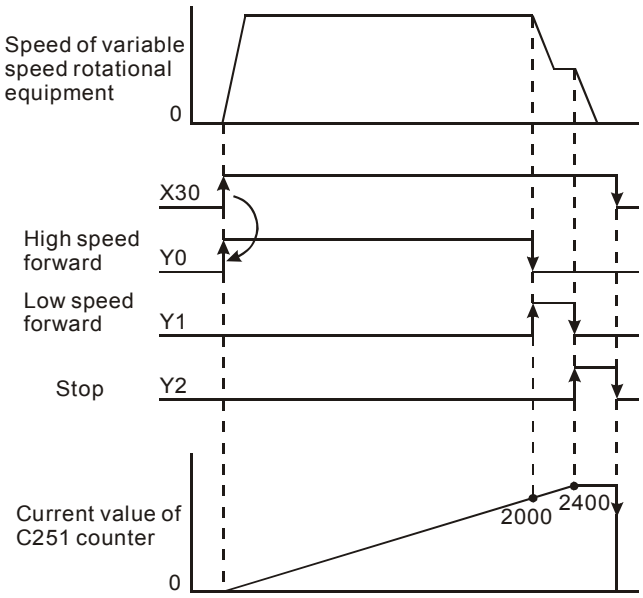


Program Example 2:

366. When using the DHSZ instruction to control stop, high, or low speed, C251 is set as an AB phase high-speed counter.
367. When X30=ON, DHSZ will turn Y0=ON when the current value ≦ K2,000. In order to improve this, use the DZCPP instruction to compare C251 against K2,000 during the first program scan after going to RUN. When counting the current value ≦ K2,000, Y0=ON and the DZCPP instruction is Pulsed. Instruction DZCPP can only be executed ONCE in a program. Y0 will continue to be ON.
368. When X20=OFF, Y0~Y2 will be reset to OFF.



TIMING DIAGRAM

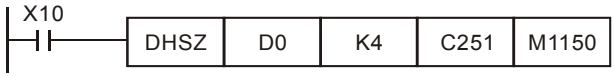


- Program Example 3:

369. Program Example 3 is only applicable to ELC-PV, ELC2-PV series.
370. The multiple set values comparison mode: If D of DHSZ instruction designates a special auxiliary relay M1150, the instruction will be able to compare the present value in the high speed counter with many values in a table.
371. In this mode,
    - **S_1**: starting value in the comparison table. **S_1** can only use the D data registers and can also utilize the index registers E and F. Once this mode is enabled, **S_1** cannot be changed even if the index registers E and F are changed.
    - **S_2**: the amount of data in the group to be compared. **S_2** can only be a constant value K1 ~ K255 or H1 ~ HFF and does not support the index registers E and F. Once this mode is enabled, **S_2** cannot be changed. If **S_2** is not within its range, error code 01EA (hex) will be displayed and the instruction will not be executed.
    - **S**: high speed counter number (C241 ~ C254).
    - **D**: Designated mode (determined by M1150)
372. The number of start registers designated in S1 and the number of rows (groups) designated in S2 construct a comparison table. Enter the values in every register in the table before executing the instruction.
373. When the present value of the counter C251 (S) equals the values in D1 and D0, the Y output designated by D2 will be reset to Off (D3 = K0) or On (D3 = K1) and be maintained. Output Y will be processed as an interrupt. The number of Y output points are in decimal (range: 0 ~

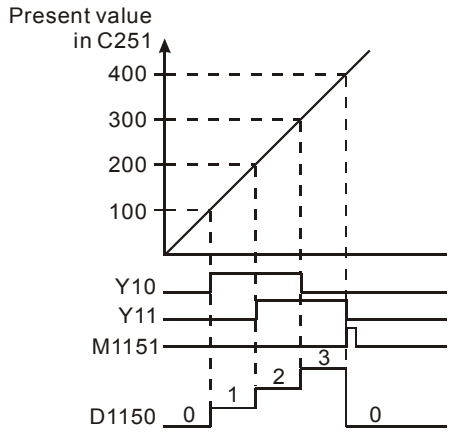
255). If the number falls outside of the range, SET/RESET will not be enabled when the comparison reaches its target.

374. When this mode is enabled, the ELC will first acquire the values in D0 and D1 as the target values for the first comparison section. At the same time, the index value displayed in D1150 will be 0, indicating that ELC performed the comparison based on the group 0 data.
375. When the group 0 data in the table has been compared, the ELC will first execute the Y output set in the group 0 data and determine if the comparison reaches the target number of groups. If the comparison reaches the target, M1151 will be On; if the comparison has not reached the final group, one will be added to the contents of D1150 and continue the comparison for the next group.
376. M1151 is the flag for the completion of one execution of the table. It can be turned Off by the user, or when the next comparison cycle takes place and the group 0 data has been compared, the ELC will automatically reset the flag.
377. When X10 is turned Off, the execution of the instruction will be interrupted and the content in D1150 will be reset to 0. However, the On/Off status of all outputs will be maintained.
378. When the instruction is being executed, all values in the comparison table will be regarded as valid values only when the program scan reaches the END statement for the first time.
379. This mode can only be used once in the program.
380. This mode can only be used on the hardware high speed counters C241 ~ C254.
381. When in this mode, the frequency of the input pulses cannot exceed 50KHz or the neighboring two groups of compare values cannot differ by 1; otherwise there will not be enough time for the ELC to react and the result will be an error.



-
-
- The comparison table:

32-bit data for comparison		No. of Y output	On/Off indication	Table counting register D1150
High word	Low word			
D1 (K0)	D0 (K100)	D2 (K10)	D3 (K1)	0
D5 (K0)	D4 (K200)	D6 (K11)	D7 (K1)	1
D9 (K0)	D8 (K300)	D10 (K10)	D11 (K0)	2
D13 (K0)	D12 (K400)	D14 (K11)	D15 (K0)	3
		K10: Y10 K11: Y11	K0: Off K1: On	0→1→2→3→0 Cyclic scan



382. Special registers for flags and relevant settings:

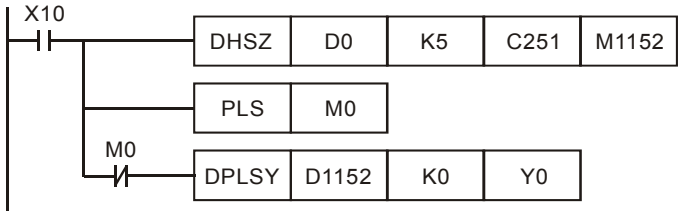
Flag	Function
M1150	DHSZ instruction in multiple set values comparison mode
M1151	The execution of DHSZ multiple set values comparison mode is completed.

Special D	Function
D1150	Table counting register for DHSZ multiple set values comparison mode

- Program Example 4:

1. Program Example 4 is only applicable to ELC-PV, ELC2-PV series.
2. DHSZ and DPLSY instructions are combined for frequency control. If **D** of the DHSZ instruction is a special auxiliary relay M1152, the present value in the counter will be able to control the pulse output frequency of DPLSY instruction.
3. In this mode,
   - $\mathbf{S_1}$: starting address in the comparison table. $\mathbf{S_1}$ can only use D registers which can utilize the index registers E and F. Once this mode is enabled, $\mathbf{S_1}$ will not be allowed to change even if E and F change.
   - $\mathbf{S_2}$: the amount of data in the group to be compared. $\mathbf{S_2}$ can only be a constant value K1 ~ K255 or H1 ~ HFF and does not support the index registers E and F. Once this mode is enabled, $\mathbf{S_2}$ cannot be changed. If $\mathbf{S_2}$ is not within its range, error code 01EA (hex) will display and the instruction will not be executed.
   - **S**: high speed counter number (designated as C241 ~ C254).
   - **D**: Designated mode (determined by M1152)

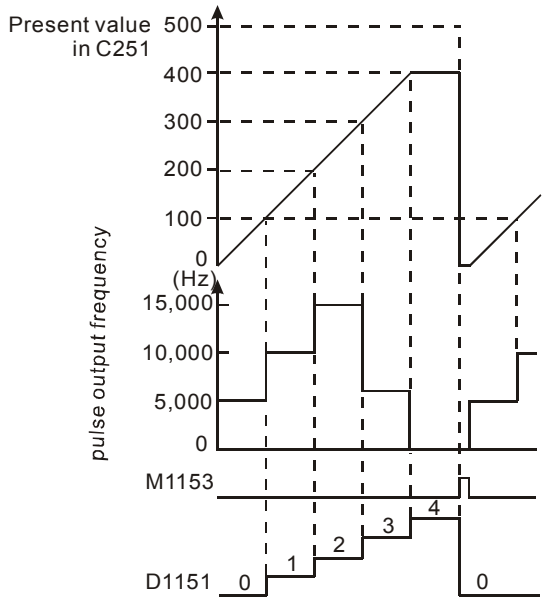
4. This mode can only be used once. For PV series, this mode can only be used with the hardware high speed counters C241 ~ C254. Enter the values in every register in the table before executing the instruction.
5. When this mode is enabled, the ELC will first acquire the values in D0 and D1 as the target values for the first comparison section. At the same time, the index value displayed in D1152 will be 0, indicating that ELC performs the comparison based on the group 0 data.
6. When the group 0 data in the table has been compared, the ELC will first execute at the frequency set in group 0 data (D2, D3) and copy the data to D1152 and D1153, determining if the comparison reaches the target number of groups. If the comparison reaches the target, M1153 will be On; if the comparison has not reached the final group, the content in D1151 will add 1 and continue the comprison for the next group.
7. M1153 is the flag for the completion of one execution of the table. It can be turned Off by the user. Or when the next comparion cycle takes place and the group 0 data has been compared, the ELC will automatically reset the flag.
8. If you wish to use this mode with the PLSY instruction, preset the value in D1152.
9. If you wish to stop the execution at the last row, set the value in the last row to K0.
10. When X10 turns Off, the execution of the instruction will be interrupted and the content in D1151 will be reset to 0.
11. When in this mode, the frequency of the input pulses cannot exceed 50KHz or the neighboring two groups of comparative values cannot differ by 1; otherwise there will not be enough time for the ELC to react and result in errors.



- The comparison table:

32-bit data for comparison		Pulse output frequency	Table counting register D1151
High word	Low word	0 ~ 200KHz	
D1 (K0)	D0 (K0)	D3, D2 (K5,000)	0
D5 (K0)	D4 (K100)	D7, D6 (K10,000)	1
D9 (K0)	D8 (K200)	D11, D10 (K15,000)	2
D13 (K0)	D12 (K300)	D15, D14 (K6,000)	3

D17 (K0)	D16 (K400)	D19, D18 (K0)	4
			0→1→2→3→4 Cyclic scan

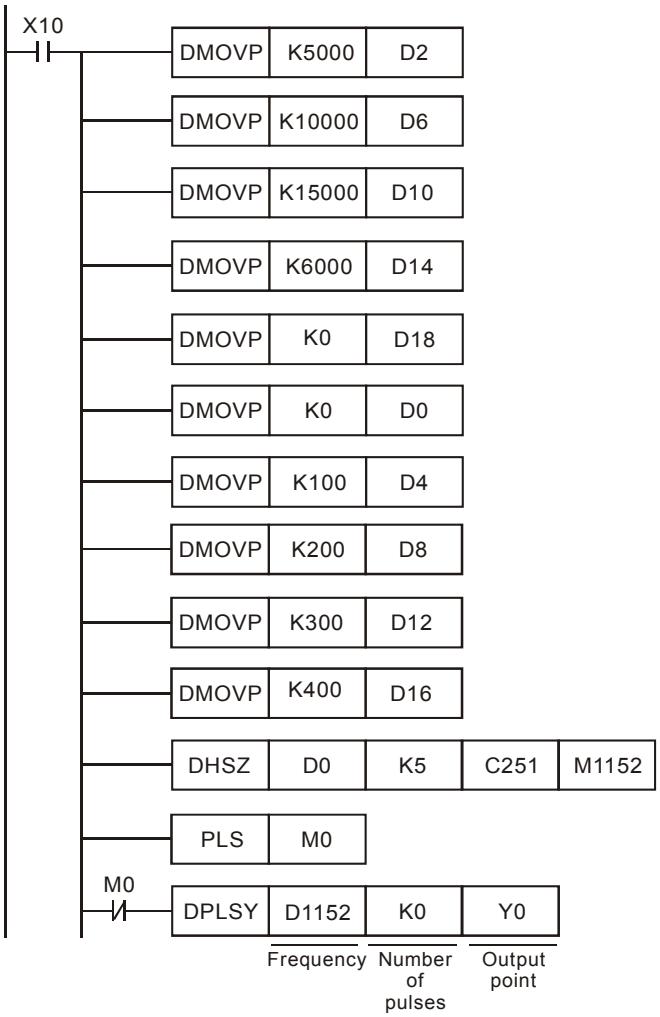


12. Special registers for flags and relevant settings:

Flag	Function
M1152	DHSZ instruction in frequency control mode
M1153	The execution of DHSZ frequency control mode is complete.

Special D	Function
D1151	Table counting register for DHSZ multiple set values comparison mode
D1152 (low word) D1153 (high word)	In frequency control mode, DHSZ reads the upper and lower limits in the table counting register D1153 and D1152.
D1336 (low word)	Current number of pulses sent out by DPLSY instruction

13. The complete program:



14. During the execution of DHSZ instruction, do not modify the values set in the comparison table.

15. The designated data will be arranged into the the above program when the program reaches the END statement. Therefore, the PLSY instruction has to be executed after DHSZ instruction has been executed once.

API	Mnemonic	Operands	Function
56	SPD	**S_1, S_2, D**	Speed Detection

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1	*															SPD: 7 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D											*	*	*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: External pulse input **S_2**: Pulse time (ms) **D**: Result (occupies 5 continuous addresses)

Description: ELC-PA/PV, ELCB-PB

383. This instruction counts the number of pulses received at input terminal **S_1** during the time **S_2** (ms) and stores the result in the register **D**.
384. External pulse input terminals designated in **S_1** for all ELC controllers:

Input \ ELC	ELCB-PB	ELC-PA (new input X0 is added for V1.2 and above)	ELC-PV
Available input points	X1, X2	X0, X1, X2	X0 ~ X3

385. ELC-PA models, if X0, X1 or X2 are used with a SPD instruction, then the related high-speed counters or external interrupts I001, I101, I201 can not be used.
386. Count the number of pulses received at the inputs specified by **S_1** during the time specified by **S_2** (ms) and store the result in the register specified by **D**.
387. **D** occupies 5 registers, **D**+1, **D** stores the result of the previous SPD operation, **D**+3, **D**+2 indicate the present accumulated count value of pulses and **D**+4 indicates the remaining time, the max. is 32767ms. Both are 32-bit integer values.
388. ELCB-PB models: Maximum frequency: X1=20KHz, X2=10KHz, Total frequency must be less than 20KHz.
389. ELC-PA models: Maximum frequency: X1=30KHz, X2=10KHz, Total frequency must be less than 30KHz. If X0 is used in a SPD instruction, X0 and X1 will be the inputs of A, B phase signals to detect the speed of A, B phase. The maximum input frequency is 4KHz. Use D1022 to set time frequency.
390. ELC-PV models: Maximum frequency: X0 and X1 = 100KHz, X2 and X3 = 10KHz, Total frequency must be less than 100KHz. If X0 is used in a SPD instruction, X0 and X1 will be the inputs of A, B phase signals to detect the speed of A, B phase.
391. This instruction is mainly used to obtain a proportional value of rotation speed. The result **D** and rotation speed are in proportion. The following equation can be used to obtain the rotation speed of motor.

$$N=\frac{60(D0)}{nt}\times 10^3(rpm)$$

N: Rotation speed
n: The number of pulses per rotation of the motor
t: Detection time specified by S_2 (ms)

392. ELC-PV models: When the SPD instruction is enabled and M1100 = On, the SPD instruction will perform a sampling at the moment when M1100 goes from Off to On, then stop the sampling. If you wish to resume the sampling, you must turn Off M1100 and re-enable the SPD instruction.

Description: (ELCM-PH/PA, ELC2-PA/PB/PH/PE)

393. External pulse input terminals designated in S_1 for ELCM and ELC2 series ELC controllers:

Available input points	X0, X2	X1(X0/X1), X3(X2/X3), X5(X4/X5), X7(X6/X7)	X6, X7
Input mode	1-phase input (Supports single frequency)	AB-phase input (Supports 4 tmes frequency)	1-phase input (Supports single frequency)
Maximum frequency	100KHz	5KHz	10KHz

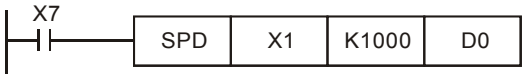
394. **D** occupies 5 consecutive registers, **D** + 1 and **D** store the results of previous pulse detection; **D** +3 and **D** + 2 store the current accumulated number of pulses; **D** + 4 stores the current time remaining (max. 32,767ms).
395. If X0 ~ X7 are used in a SPD instruction, their associated high-speed counters and external interrupts I000/I001, I100/I101, I200/I201, I300/I301, I400/I401, I500/I501, I600/I601 or I700/I701 cannot be used.
396. When X0, X2, X6 and X7 are used, they will be detected as 1-phase inputs. When X1, X3, X5, X7, is used, X0, X2, X4, X6 (A) and X1, X3, X5, X7 (B) will be detected as AB-phase input.
397. This instruction is mainly used to obtain the value of rotation speed and the results in **D** are in proportion to the rotation speed. Rotation speed **N** can be calculated by the following equation

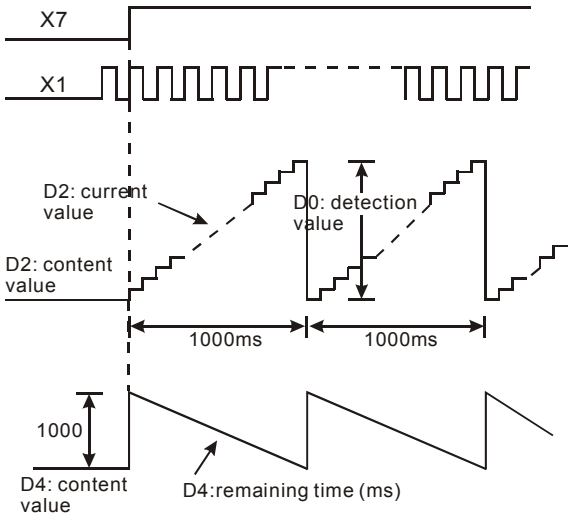
$$N=\frac{60(D0)}{nt}\times 10^3(rpm)$$

N: Rotation speed
n: The number of pulses produced per rotation
t: Detecting time specified by S_2 (ms)

Program Example:

398. When X7=ON, D2 will count the high-speed pulses from X1. After 1,000ms, it will stop counting automatically and store the result in D0.
399. After 1000ms of counting has completed, the content of D2 will be reset to 0. When X7 turns ON again, D2 will begin counting again.





API	Mnemonic		Operands	Function
57	D	PLSY	S_1, S_2, D	Pulse Output

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	PLSY: 7 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	DPLSY: 13 steps
D		*														

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Pulse output frequency **S_2**: Number of output pulses **D**: External output (only Y0 and Y1 can be specified)

Description: ELC-PA/PV, ELCB-PB, ELC2-PV Controllers

400. When the PLSY instruction is executed, the specified quantity of pulses **S_2** will be sent out the output terminal specified by **D** at the specified pulse output frequency **S_1.**

401. **S_1** - the pulse output frequency

Output frequency range of each series models			
Models	ELCB-PB	ELC-/PA	ELC-PV, ELC2-PV
Output frequency range	Y0：1~10,000Hz Y1：1~10,000Hz	Y0：1~30,000Hz Y1：1~30,000Hz	Y0: 0~200,000Hz Y2: 0~200,000Hz Y4: 0~200,000Hz Y6: 0~200,000Hz
M1190=ON	-	-	Y0: 0.01~100Hz
M1191=ON	-	-	Y2: 0.01~100Hz
M1192=ON	-	-	Y4: 0.01~100Hz
M1193=ON	-	-	Y6: 0.01~100Hz

402. **S_2** - Number of output pulses.

16-bit instruction: 1~32,767. 32-bit instruction: 1~2,147,483,647.

Continuous pulses	M1010（Y0） ON, pulse output is continuous M1023（Y1） ON pulse output is continuous In ELC-PV, set S2 (the number of output pulses for Y0, Y2, Y4 and Y6) to K0

403. For ELC-PV series, when the number of output pulses is set to 0, there will be continuous pulse output with no limit on the number of pulses. For ELC-PB/PC/PA/PH, ELCB-PB series, you must to turn M1010 (Y0) or M1023 (Y1) On to allow a continuous pulse output with no limit on the number of pulses.

404. For the pulse output terminal specified in D, the ELC-PV, ELC2-PV series can use Y0, Y2, Y4 and Y6, The ELC-PA, ELCB-PB series can use Y0 and Y1.

405. The ELC-PV, ELC2-PV series has four groups of A-B phase pulse outputs from CH0 (Y0, Y1), CH1 (Y2, Y3), CH2 (Y4, Y5) and CH3 (Y6, Y7). See the remarks in section 2.3 concerning how to set this up. .
406. When the PLSY instruction has been executed, the specified quantity of pulses $\mathbf{S_2}$ will be sent out the output terminal specified by **D** at the specified pulse output frequency $\mathbf{S_1}$.
407. When the PLSY instruction is used in a program, the outputs used in the PLSY instruction cannot be used in the PWM instruction or PLSR instruction.
408. Pulse output complete flags for each controller type:

ELC	ELC-PA ELCB-PB		ELC-PV, ELC2-PV			
Output device	Y0	Y1	Y0	Y2	Y4	Y6
Flag	M1029	M1030	M1029	M1030	M1036	M1037

a) For ELC-PA, ELCB-PB series, after the Y0 pulse output is complete, M1029 = ON; after the Y1 pulse output is complete, M1030 = ON. When the PLSY instruction is OFF, M1029 and M1030 will be OFF.

b) The execution complete flags M1029, M1030 should be cleared by the user after the execution of the instruction has been completed.

c) For ELC-PV, ELC2-PV series, M1029 is set to ON after Y0 finishes sending the specified number of pulses , M1030 is set to ON after Y2 finishes sending the specified number of pulses, M1036 is set to ON after Y4 finishes sending the specified number of pulses, and M1037 is set to ON after Y6 finishes sending the specified number of pulses. When the PLSY instruction is OFF, M1029, M1030, M1036, and M1037 will not be turned OFF automatically. The user program will need to reset them.

409. For the ELC-PA, ELCB-PB series controllers, when the PLSY and DPLSY instructions are disabled, the pulse output completed flags will all be turned Off automatically.
410. For the ELC-PV, ELC2-PV series, when the PLSY and DPLSY instructions are disabled, the user will have to reset the pulse output complete flags. They are not reset automatically like with the other ELC controllers.
411. While the PLSY instruction is being executed, the output will not be affected if $\mathbf{S_2}$ is changed. To change the number of output pulses, the PLSY instruction must be disabled, then change the number the number of pulses.
412. $\mathbf{S_1}$ the pulse output frequency can be changed while the PLSY instruction is being executed.
413. The pulses sent out by the PLSY instruction are sent at a 50% duty cycle. For example, if each pulse is sent at 1000Hz, it will be On for .5ms and Off for .5ms.
414. If operands $\mathbf{S_1}$, $\mathbf{S_2}$ use index register F, then only the 16-bit instruction is available (PLSY).
415. For ELCB-PB series, the PLSY instruction can only be used twice in the program.

416. For ELC-PA/PV, ELC2-PV series, there is no limit on the number of times this instruction may be used in the program. However, for the ELC-PA series, the program only allows two PLSY instructions to be executed at the same time. For ELC-PV series, the program allows four PLSY instructions to be executed at the same time.

ELCM-PH/PA, ELC2-PA/PB/PH/PE Controllers

417. S_1 specifies the pulse output frequency

Output frequency range			
Series	ELC2-PB	ELCM-PH/PA ELC2-PA/PH/PE	
Output	Y0~Y3	Y0, Y2	Y1, Y3
16-bit instruction	0~10,000Hz	0~32,767 Hz	0~10,000Hz
32-bit instruction	0~10,000Hz	0~100,000Hz	0~10,000Hz
If 0Hz is specified, pulse output will be disabled			

418. S_2 specifies the number of output pulses.

16-bit instruction: -32,768~32,767. 32-bit instruction: -2,147,483,648~2,147,483,647.

When S_2 is specified as K0, the pulse will be continuous output.

419. ELCM-PH/PA, ELC2-PA/PB/PH/PE series have four pulse output modes as shown below.

Mode / Output	D1220						D1221					
	K0		K1	K2	K3		K0		K1	K2	K3#	
Y0	Pulse		Pulse	A	CW							
Y1		Pulse	Dir	B		Pulse						
Y2							Pulse		Pulse	A	CCW	
Y3								Pulse	Dir	B		Pulse

Pulse: Pulse A: A phase pulse CW: clockwise

Dir: Direction B: B phase pulse CCW: Counter-clockwise

Note #: When D1220 contains the value K3, D1221 is invalid.

420. Pulse output flags for ELCM-PH/PA, ELC2-PA/PB/PH/PE series:

Output device	Y0	Y1	Y2	Y3
Completed Flag	M1029	M1030	M1102	M1103
pause	M1078	M1079	M1104	M1105
0.01~10Hz output	M1190	M1191	M1192	M1193

a) M1029 = ON after Y0/Y1 (D1220=K1, pulse/Dir) is complete.

M1102 = ON after Y2/Y3 (D1221=K1, pulse/Dir) is complete.

M1029 = ON after the Y0/Y2 (D1220 = K3, CW/CCW) is complete.

b) The execution complete flag M1029, M1030, M1102, and M1103 should be manually reset by the user program after the pulse output has completed.

c) When M1190~M1192 is ON, frequency 0.01~100Hz is available for outputs Y0~Y3.

d) When the PLSY / DPLSY instruction is OFF, the pulse output complete flags will all be reset.

e) When M1190~M1192 = ON, the available output range for PLSY Y0~Y3 is 0.01~100Hz.

421. While the PLSY instruction is being executed, the output will not be affected if $\mathbf{S_2}$ is changed. To change the number of pulses, stop the PLSY instruction, then change the number of pulses.

422. $\mathbf{S_1}$ the pulse output frequency can be changed while the PLSY instruction is being executed.

423. The pulses sent out by the PLSY instruction are sent at a 50% duty cycle. For example, if each pulse is sent at 1000Hz, it will be On for .5ms and Off for .5ms.

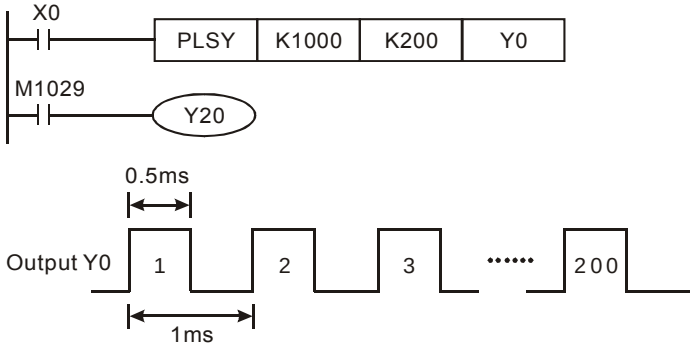
424. If operand $\mathbf{S_1}$, $\mathbf{S_2}$ use the index register F, only the 16-bit instruction (PLSY) is available.

425. There is no limit on the number of times this instruction can be used in the program. However the program only allows 4 pulse output instructions (PLSY, PWM, PLSR) to be executed at the same time. If Y1 is used for several high speed pulse output instructions, the controller will execute each one in the order they are enabled and scanned in the program.

Program Example:

426. When X0=ON, 200 pulses will be sent out Y0 at 1KHz. When the instruction is finished, M1029=ON (the complete flag) which will turn on Y20.

427. When X0=OFF, the pulse output to Y0 will immediately stop. When X0 turns ON again, the pulse output will restart.



Points to note:

428. Flags and special registers for ELC-PB, ELCB-PB series:

M1010: When M1010=ON, continuous pulses will be sent out Y0. When M1010=OFF, the PLSY instruction will send the number of pulses specified in $\mathbf{S_2}$.

M1023: When M1023=ON, Y1 will send continuous pulses. When M1023=OFF, the PLSY instruction will send the number of pulses specified in $\mathbf{S_2}$.

M1029: M1029= ON after Y0 pulse output is complete.

M1030: M1030= ON after Y1 pulse output is complete.

M1078: Pause sending pulses out Y0.

M1079: Pause sending pulses out Y1.

D1030: Present number of pulses sent out Y0 (LOW WORD).

D1031: Present number of pulses sent out Y0 (HIGH WORD).

D1032: Present number of pulses sent out Y1 (LOW WORD).

D1033: Present number of pulses sent out Y1 (HIGH WORD).

429. Flags and special registers for ELC-PA series:

M1010: When On, Y0 output will be continuous with no limit on the number of pulses. When Off, the number of output pulses from Y0 will be determined by $\mathbf{S_2}$.

M1023: When On, Y1 output will be continuous with no limit on the number of pulses. When Off, the number of output pulses from Y1 will be determined by $\mathbf{S_2}$.

M1029: On when Y0 pulse output is complete.

M1030: On when Y1 pulse output is complete.

M1078: Pause sending pulses out Y0

M1079: Pause sending pulses out Y1

M1347: For ELC-PA, Auto reset Y0 when high speed pulse output is complete

M1348: For ELC-PA, Auto reset Y1 when high speed pulse output is complete

D1030: Low word of the current number of output pulses from Y0

D1031: High word of the current number of output pulses from Y0

D1032: Low word of the current number of output pulses from Y1

D1033: High word of the current number of output pulses from Y1

430. Flags and special registers for ELC-PV, ELC2-PV series:

M1010: When On, CH0, CH1, CH2 and CH3 will send pulses at END of program scan. Off when the output starts.

M1029: On when CH0 pulse output is complete.

M1030: On when CH1 pulse output is complete.

M1036: On when CH2 pulse output is complete.

M1037: On when CH3 pulse output is complete.

M1190: Set Y0 high speed output as 0.01~100Hz

M1191: Set Y2 high speed output as 0.01~100Hz

M1192: Set Y4 high speed output as 0.01~100Hz

M1193: Set Y6 high speed output as 0.01~100Hz

M1334: Pause sending pulses out CH0

M1335: Pause sending pulses out CH1

M1520: Pause sending pulses out CH2

M1521: Pause sending pulses out CH3

M1336: CH0 pulse output has been sent.

M1337: CH1 pulse output has been sent.

M1522: CH2 pulse output has been sent.

M1523: CH3 pulse output has been sent.

M1338: CH0 offset pulses enabled.

M1339: CH1 offset pulses enabled.

M1340: I110 interrupt occurs after CH0 pulse output is complete.

M1341: I120 interrupt occurs after CH1 pulse output is complete.

M1342: I130 interrupt occurs when CH0 pulse output is in process.

M1343: I140 interruption occurs when CH0 pulse output is in process.

M1344: CH0 pulse compensation enabled.

M1345: CH1 pulse compensation enabled.

M1347: CH0 pulse output reset flag

M1348: CH1 pulse output reset flag

M1524: CH2 pulse output reset flag

M1525: CH3 pulse output reset flag

D1220: Phase setting for CH0 (Y0, Y1): D1220 determines the phase type for CH0 with the two low bits; other bits in this word are invalid.

1. K0: Y0 output
2. K1: Y0, Y1 AB-phase output; A ahead of B.
3. K2: Y0, Y1 AB-phase output; B ahead of A.
4. K3: Y1 output

D1221: Phase setting for CH1 (Y2, Y3): D1221 determines the phase type for CH1 with the low two bits; other bits in this word are invalid.

1. K0: Y2 output
2. K1: Y2, Y3 AB-phase output; A ahead of B.
3. K2: Y2, Y3 AB-phase output; B ahead of A.
4. K3: Y3 output

D1229: Phase setting for CH2 (Y4, Y5): D1229 determines the phase type for CH2 with the low two bits; other bits in this word are invalid.

1. K0: Y4 output
2. K1: Y4, Y5 AB-phase output; A ahead of B.
3. K2: Y4, Y5 AB-phase output; B ahead of A.
4. K3: Y5 output

D1230: Phase setting for CH3 (Y6, Y7): D1230 determines the phase type for CH3 with the low two bits; other bits in this word are invalid.

1. K0: Y6 output
2. K1: Y6, Y7 AB-phase output; A ahead of B.
3. K2: Y6, Y7 AB-phase output; B ahead of A.
4. K3: Y7 output

D1328: Low word of the number of CH0 offset pulses

D1329: High word of the number of CH0 offset pulses

D1330: Low word of the number of CH1 offset pulses

D1331: High word of the number of CH1 offset pulses

D1332: Low word of the number of remaining pulses at CH0

D1333: High word of the number of remaining pulses at CH0

D1334: Low word of the number of remaining pulses at CH1

D1335: High word of the number of remaining pulses at CH1

D1336: Low word of the current number of output pulses at CH0

D1337: High word of the current number of output pulses at CH0

D1338: Low word of the current number of output pulses at CH1

D1339: High word of the current number of output pulses at CH1

D1375: Low word of the current number of output pulses at CH2

D1376: High word of the current number of output pulses at CH2

D1377: Low word of the current number of output pulses at CH3

D1378: High word of the current number of output pulses at CH3

D1344: Low word of the number of compensation pulses at CH0

D1345: High word of the number of compensation pulses at CH0

D1346: Low word of the number of compensation pulses at CH1

D1347: High word of the number of compensation pulses at CH1

431. Flags and special registers for ELCM-PH/PA, ELC2-PA/PB/PH/PE series:

M1029: M1029 = ON when Y0 pulse output is complete.

M1030: M1030 = ON when Y1 pulse output is complete.

M1102: M1102 = ON when Y2 pulse output is complete.

M1103: M1103 = ON when Y3 pulse output is complete.

M1078: Pause sending pulses out Y0

M1079: Pause sending pulses out Y1

M1104: Pause sending pulses out Y2

M1105: Pause sending pulses out Y3

M1190 Set Y0 high speed output as 0.01~10Hz

M1191 Set Y1 high speed output as 0.01~10Hz

M1192 Set Y2 high speed output as 0.01~10Hz

M1193 Set Y3 high speed output as 0.01~10Hz

M1347: Auto reset Y0 when high speed pulse output is complete

M1348: Auto reset Y1 when high speed pulse output is complete

M1524: Auto reset Y2 when high speed pulse output is complete

M1525: Auto reset Y3 when high speed pulse output is complete

M1538: Indicates if Y0 is paused

M1539: Indicates if Y1 is paused

M1540: Indicating if Y2 is paused

M1541: Indicating if Y3 is paused

D1030: Present number of Y0 output pulses (LOW WORD).

D1031: Present number of Y0 output pulses (HIGH WORD).

D1032: Present number of Y1 output pulses (LOW WORD).

D1033: Present number of Y1 output pulses (HIGH WORD).

D1336: Present number of Y2 output pulses (LOW WORD).

D1337: Present number of Y2 output pulses (HIGH WORD).

D1338: Present number of Y3 output pulses (LOW WORD).

D1339: Present number of Y3 output pulses (HIGH WORD).

D1220: Phase setting of the 1st pulse output group (Y0, Y1),

D1221: Phase setting of the 2nd pulse output group (Y2, Y3),

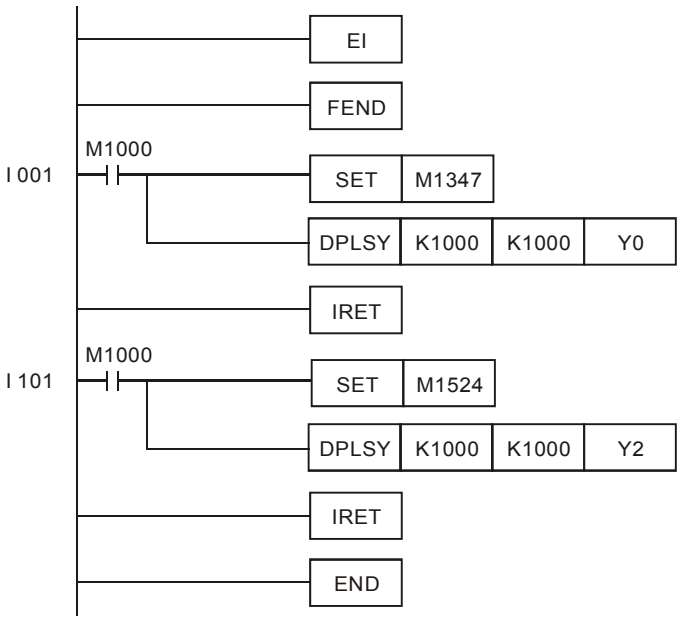
432. When several pulse output instructions (PLSY, PWM, PLSR) use Y0 as the output in the same program, and are executed simultaneously in the same scan cycle, the ELC will perform the instruction which has fewest step numbers.

433. More information for M1347,M1348, M1524, M1525:

Generally when the pulse output is complete, the PLSY instruction must be reset so that the instruction can be executed again. When M1347, M1348, M1524 or M1525 is enabled, the associated output terminals (Y0~Y3) will be reset automatically when the pulse output is complete, i.e. the PLSY instruction is reset. When the ELC scans the PLSY instruction again, the pulse output will automatically start. Also, the ELC updates the 4 flags at the END of the program scan. This means that the PLSY instruction in continuous pulse output mode requires a delay time of one scan cycle for the next pulse output operation.

The function is mainly used in subroutines or interrupts which require high speed pulse outputs. Here are a couple of examples:

Program Example 1:

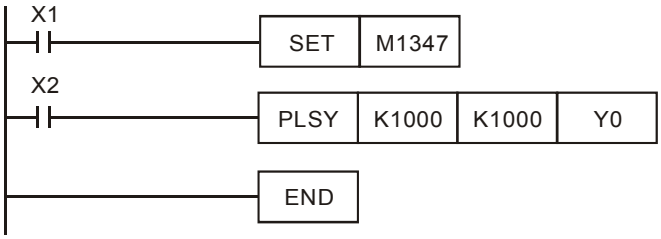


Description:

a) Whenever I001 is triggered, Y0 will send 1,000 pulses; whenever I101 is triggered, Y2 will send 1,000 pulses.

b) When the pulse output is complete, there should be an interval of at least one scan cycle before the next pulse output operation can be triggered. .

Program Example 2:



Description:

When both X1 and X2 are ON, the pulse output to Y0 will operate continuously. However, there will be a delay of approx. 1 scan cycle every 1000 pulses.

API	Mnemonic	Operands	Function
58	PWM	S_1, S_2, D	Pulse Width Modulation

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	PWM: 7 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D		*														

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Pulse output width S_2: Pulse output period **D**: Pulse output address

Description:

434. S_1 is the pulse output width with a range of 0~32,767ms. S_2 is specified as pulse output period with a range of 1~32,767ms, where $S_1 \leqq S_2$.

435. **D** for all ELC controllers:

ELC	ELC-PA, ELCB-PB	ELC-PV, ELC2-PV	ELCM-PH/PA, ELC2-PA/PB/PH/PE
Output point	Y1	Y0, Y2, Y4, Y6	Y0 ~ Y3

436. For the ELC-PB, ELCB-PB models, The output specified in **D** cannot be the same as the output used in a PLSY or PLSR instruction. The PWM instruction can only be used once in the program.

437. When several pulse output instructions (PLSY, PWM, PLSR) enable same output point in the same scan cycle, the ELC will perform the instruction which is executed first.

438. When $S_1 \leqq 0$, $S_2 \leqq 0$ or $S_1 > S_2$, the ELC controller will fault (M1067 and M1068 will not be ON) and no output will be generated from the pulse output instruction.. When $S_1 = S_2$, the pulse output device will be ON continuously.

439. S_1, S_2 can be changed during the execution of PWM instruction.

440. For ELC-PA, ELCM-PH/PA, ELC2-PA/PB/PH/PE series, when M1070 = ON, the units for the output pulses sent to Y1 are 100μs. When M1070 = OFF, the units are 1ms. When M1071 = ON, the units for the output pulses sent to Y3 are 100μs. When M1070 = OFF, the units are 1ms.

441. For ELC-PV, ELC2-PV series, setting the time units for CH0 is determined by the contents of D1371. Setting the time units for CH1 is determined by the contents of D1372.

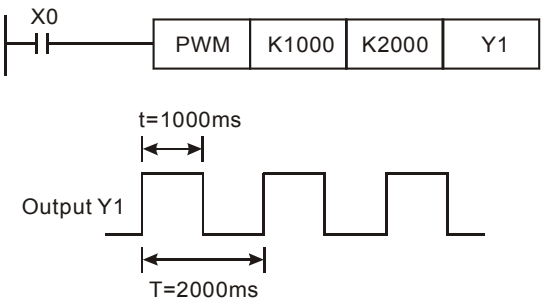
442. For ELC-PA/PV, ELCM-PH/PA, ELC2-PA/PB/PH/PE/PV series, there is no limit on the number of times this instruction can be used in the program. For ELC-PV, ELC2-PV series,

four instructions are allowed to be executed at the same time. For ELCM-PH/PA, ELC2-PA/PB/PH/PE series, two instructions are allowed to be executed at the same time.

Program Example:

When X0=ON, the following pulses will be sent out Y1.

When X0=OFF, output Y1 will turn OFF.



Note:

1. Flags and special registers for the ELC-PB/PA, ELCB-PB series controllers:

   M1070: Selecting the clock pulse units of Y1 for the PWM instruction (ON:100 us, OFF: 1ms)

2. Flags and special registers for ELC-PV, ELC2-PV series controllers:

   M1010: When On, CH0, CH1, CH2 and CH3 will output pulses when the END instruction is executed. Off when the output starts.

   M1070: The setting of time units for CH0 is determined by the contents of D1371.

   M1071: The setting of time units for CH1 is determined by the contents of D1372.

   M1258: Reverse CH0 pulse output signals.

   M1259: Reverse CH1 pulse output signals.

   M1334: Pause CH0 pulse output.

   M1335: Pause CH1 pulse output.

   M1336: CH0 pulse output has been sent.

   M1337: CH1 pulse output has been sent.

   M1520: CH2 pulse output is paused.

   M1521: CH3 pulse output is paused.

   M1522: CH2 pulse output has been sent.

   M1523: CH3 pulse output has been sent.

   M1526: Reverse CH2 pulse output signals.

   M1527: Reverse CH3 pulse output signals.

   M1530: M1530 = ON, the setting of time units for CH2 is determined by the contents of D1373. When = Off, the time units are 1ms.

   M1531: M1531 = ON, the setting of time units for CH3 is determined by the contents of D1374. When = Off, the time units are 1ms.

   D1336: Low word of the current number of output pulses from CH0.

D1337: High word of the current number of output pulses from CH0.

D1338: Low word of the current number of output pulses from CH1.

D1339: High word of the current number of output pulses from CH1.

D1371: Time unit of CH0 output pulses when M1070 = On.

D1372: Time unit of CH1 output pulses when M1071 = On.

D1373: Time unit of CH2 output pulses when M1530 = On.

D1374: Time unit of CH3 output pulses when M1531 = On.

D1375: Low word of the current number of output pulses from CH2.

D1376: High word of the current number of output pulses from CH2.

D1377: Low word of the current number of output pulses from CH3.

D1378: High word of the current number of output pulses from CH3.

3. Flags and special registers for ELCM-PH/PA, ELC2-PA/PB/PH/PE series ELC:

M1070: Switching clock pulse of Y1 for PWM instruction (ON:100 us, OFF: 1ms)

M1071: Switching clock pulse of Y3 for PWM instruction (ON:100 us, OFF: 1ms)

D1032: Low word of the present value of Y1 pulse output

D1033 High word of the present value of Y1 pulse output

D1338: Low word of the present value of Y3 pulse output.

D1339: High word of the present value of Y3 pulse output.

4. Time unit settings for ELC-PV, ELC2-PV series:

You cannot modify M1070 in the program.

D1371, D1372, D1373 and D1374 determine the time units of the output pulses for CH0, CH1, CH2 and CH3. The default setting is K1. If the value is not within range, the default value will be used.

D1371, D1372, D1373, D1374	K0	K1	K2	K3
Time unit	10us	100us	1ms	10ms

API	Mnemonic		Operands	Function
59	D	PLSR	**S_1, S_2, S_3, D**	Pulse Ramp

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	PLSR: 9 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	DPLSR: 17 steps
S_3					*	*	*	*	*	*	*	*	*	*	*	
D		*														

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Maximum speed (Hz) **S_2**: Number of pulses **S_3**: Acceleration/Deceleration time (ms)

D: Pulse output address

Description: (ELC-PB/PA/PV, ELCB-PB, ELC2-PV)

443. **S_1**: the maximum frequency (Hz) for the 16-bit instruction: 10 to 32,767 Hz. For the 32-bit instruction: 10 to 200,000 Hz. The maximum speed must be multiples of 10, if not, the ones digit will be discarded. 1/10 of the maximum speed is the maximum variation per accel/decel step. Note: This condition meets acceleration requirements of a stepper motor and would not result in stepper motor damage.

444. Range of **S_1** pulse output frequency:

Range of **S_1** pulse output frequency:			
	ELCB-PB	ELC-PA	ELC-PV, ELC2-PV
Output frequency:	Y0：1~10,000Hz Y1：1~10,000Hz	Y0：1~30,000Hz Y1：1~30,000Hz	Y0：10~200,000Hz Y1：10~200,000Hz Y4：10~200,000Hz Y6：10~200,000Hz

445. **S_2**: Number of pulses for the 16-bit instruction: 110~32,767, For the 32-bit instruction: 110~2,147,483,647. The minimum value is 110.

446. **S_3**: Acceleration/Deceleration time (ms). Set-points: below 5,000ms. The acceleration time and the deceleration time are the same setting. For ELCB-PB series.

a) The accel/decel time must be larger than 10 times the maximum scan time (D1012). If the set-point is below this, the slope of the accel/decel may be inaccurate.

b) Minimum set-point of the accel/decel time can be obtained from the following equation.

$$(S_3) \geqslant \frac{90000}{(S_1)}$$

If the set-point is smaller than the result of the above equation, the accel/decel time will be greater, and if the set-point is smaller than the 90000/ **S_1**, the result value of 90000/ **S_1**

will be treated as its regular set-point.

c) Maximum set-point of the accel/decel time can be obtained from the following equation.

$$(S_3) \leqslant \frac{(S_2)}{(S_1)} \times 818$$

d) The number of the accel/decel speed variation steps is fixed at 10 steps. If the input acceleration/deceleration time is greater than the maximum set-point, the maximum set-point will be treated as its regular set-point. If the set-point is smaller than the minimum set-point, the minimum set-point will be treated as its regular set-point.

447. **D** for all series:

ELC	ELC-PA, ELCB-PB	ELC-PV, ELC2-PV
Output point	Y0, Y1	Y0, Y2, Y4, Y6

448. The acceleration is performed when the pulses go from the static state to reaching its targeted speed. Then it slows down the closer it gets to its targeted distance. The pulse will stop its output once the targeted distance is reached.

449. When the PLSR instruction has been executed, its output frequency is first increased in increments of 1/10 of the maximum frequency $\mathbf{S_1}$/10 and the time of each output frequency is fixed at 1/9 of $\mathbf{S_3}$.

450. The output will not be affected if $\mathbf{S_1}$, $\mathbf{S_2}$ or $\mathbf{S_3}$ are changed when PLSR instruction is being executed.

451. For ELC-PA, ELCB-PB series, when all the Y0 pulses have been sent, M1029 will be On; when all the Y1 pulses have been sent, M1030 will be On. Next time when PLSR instruction is enabled, M1029 or M1030 will be reset.

452. For ELC-PV, ELC2-PV series, when all the CH0 (Y0, Y1) pulses have been sent, M1029 will be On; when all the CH1 (Y2, Y3) pulses have been sent, M1030 will be On; when CH2 (Y4, Y5) pulses have been sent, M1036 will be On; when CH3 (Y6, Y7) pulses have been sent, M1037 will be On. Next time when the PLSR instruction is enabled, M1029, M1030, M1036 or M1037 will be reset.

453. For ELC-PV, ELC2-PV series, if the set value falls out of the allowable range of operands it will be automatically corrected with the min. or max available value.

454. ELC-PV, ELC2-PV series has four groups of A-B phase pulse outputs CH0 (Y0, Y1), CH1 (Y2, Y3), CH2 (Y4, Y5) and CH3 (Y6, Y7). See the remarks in API 57 PLSY for more information.

455. During the acceleration of each step, the pulse numbers (each frequency x time) may not all be integer values, but the ELC controller only uses integer values for this operation. Therefore, the time of each interval may have some deviation. The offset is determined by the frequency value and by discarding the decimal point value. In order to ensure the output pulse values are correct, the ELC will fill in pulses as needed to keep any deviation to a minimum.

456. For ELCB-PB series, PLSR instruction can be used twice in the program but the outputs cannot be repeated.

457. For ELC-PV, ELC2-PV series, there is no limit on the number of times this instruction can be used in the program. However, for ELC-PA series, two instructions can be executed at the same time; for ELC-PV, ELC2-PV series, four instructions can be executed at the same time.

ELCM-PH/PA, ELC2-PA/PB/PH/PE controllers

458. The PLSR instruction performs a frequency ramp up/down process for positioning. The speed ramp up process is activated between the static state and the target speed. The pulse output continues at the target speed until it gets close to the target position. When it nears the target position, the speed ramp down process begins, and the pulse output stops when the target position is reached.

459. Set range of S_1 pulse output frequency:

Output frequency range			
Series	ELC2-PB	ELCM-PH/PA, ELC2-PA/PH/PE	
Output	Y0 ~ Y3	Y0, Y2	Y1, Y3
16-bit instruction	6~10,000Hz	6~32,767 Hz	6~10,000Hz
32-bit instruction	6~10,000Hz	6~100,000Hz	6~10,000Hz

460. When the output is specified as Y0, Y2, the start/end frequency of Y0 is set with D1340 and the start/end frequency of Y2 is set with D1352.

461. When the output is specified as Y1, Y3, the start/end frequency is 0Hz.

462. The PLSR instruction supports two modes of pulse output per the table below.

Mode / Output	D1220			D1221		
	K0		K1	K0		K1
Y0	Pulse		Pulse			
Y1		Pulse	Dir			
Y2				Pulse		Pulse
Y3					Pulse	Dir

463. When using Y0 and Y2 in output mode Pulse, i.e. D1220 = K0, D1221 = K0, the available range for S_2 is 1~32,767 (16-bit instruction) and 1~2,147,483,647 (32-bit instruction).

464. When using Y0 and Y2 in output mode Pulse/Dir, i.e. D1220 = K1, D1221 = K1, the available range for S_2 is 1~32,767 or -1~-32,768 (16-bit instruction) and 1~2,147,483,647 or -1~-2,147,483,648 (32-bit instruction)

465. When using outputs Y1 and Y3, the available range for S_2 is 1~32,767 (16-bit instruction) and 1~2,147,483,647 (32-bit instruction).

466. S_3: Ramp up/down time units: ms, min. 20ms. When using outputs Y1 and Y3, the value for the ramp up and ramp down time are the same. When assigning outputs Y0 and Y2, and if:

M1534 = OFF (Y0) and M1535 = OFF (Y2), then the ramp up and ramp down times are the same.

When M1534 = ON and M1535 = ON, S_3 specifies the ramp up time only. The ramp down time is specified by the value set in D1348 (Y0) and D1349 (Y2).

467. **D** for Y0, Y1, Y2, Y3.
468. When M1257 = OFF, the ramp up/down curve of Y0 and Y2 is a straight line. When M1257 = ON, the ramp up/down curve will be an S curve.
469. The ramp up/down curve of Y1 and Y3 is a straight line.
470. The output will not be affected if S_1, S_2 or S_3 are changed when the PLSR instruction is being executed. The PLSR instruction must be stopped to change the S_1, S_2 or S_3 values.
471. Pulse output status flags

Output	Y0	Y1	Y2	Y3
Completion	M1029	M1030	M1102	M1103
Pause	M1078	M1079	M1104	M1105

a) When the pulse output on Y0/Y1 specified as Pulse/Dir (D1220 = K1) is complete, the complete flag M1029 = ON.

b) When the pulse output on Y2/Y3 specified as Pulse/Dir (D1221 = K1) is complete, the complete flag M1102 = On。

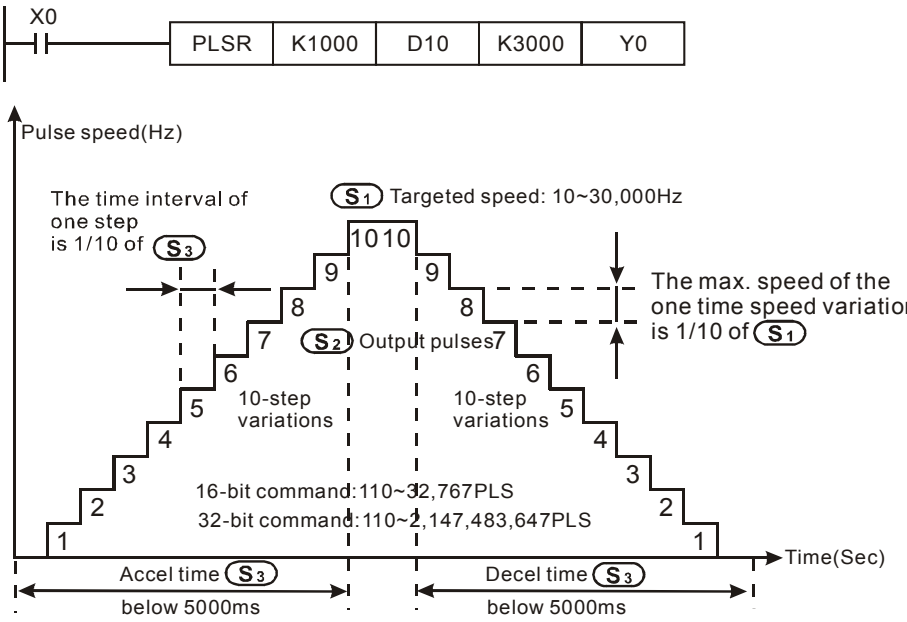
c) When the PLSR/DPLSR instruction is activated again, the complete flags will automatically be reset.

472. During the ramp up process, the pulse numbers (frequency x time) of each speed shift may not all be integer values, and the ELC will operates with integer values only. In this case, the omitted decimal points will result in errors between each speed shift. To insure the correct number of output pulses, the, ELC will fill in pulses as need to the last shift in order to correct the deviation.
473. For ELCM-PH/PA series, there is no limit on the number of times this instruction can be used in the program. However, only 4 instructions can be executed in the same scan. When several pulse output instructions (PLSY, PWM, PLSR) use Y1 as the output device in the same scan cycle, the ELC will execute the instructions in the order scanned.
474. If the set value falls out of the available range of operands, it will be automatically corrected with the available min. or max value.

Program Example: (ELC-PA/PV, ELCB-PB, ELC2-PV)

475. When X0=ON, the maximum frequency of the PLSR instruction is 1,000Hz. D10 is the total quantity of output pulses, the accel/decel time is 3,000ms and pulses are sent out Y0. The output frequency changes 1,000/10 Hz each step. The frequency of each pulse is fixed at 3,000/9.

476. When X0 = OFF, the output will be interrupted, and when turned ON again, the pulses will restart at zero.

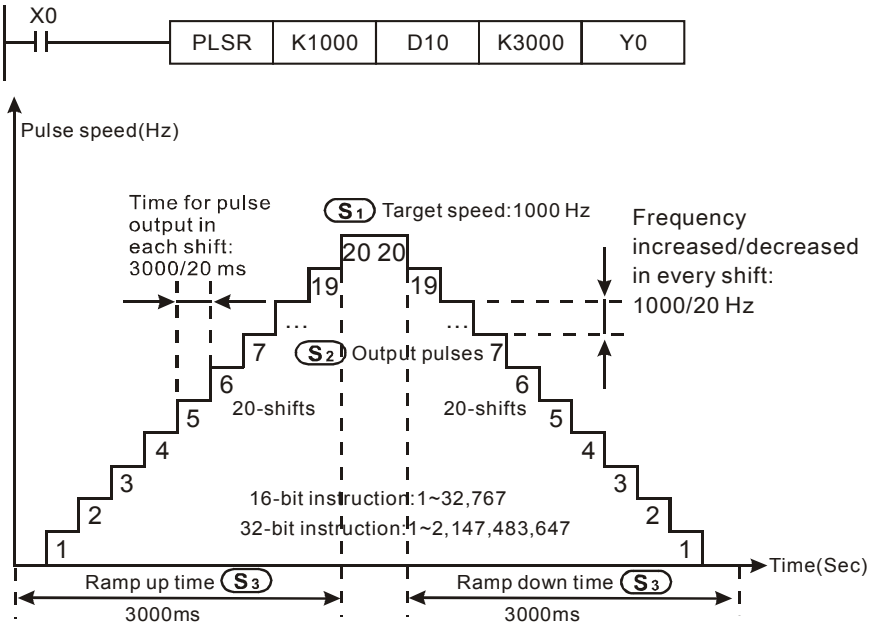


Program Example: (ELCM-PH/PA, ELC2-PA/PB/PH/PE)

477. When X0 = ON, the PLSR instruction sends pulses out Y0 with a target speed of 1000Hz, an output pulse value of D10 and a ramp up/down time of 3000ms. The ramp up process begins to increase 1000/20 Hz in every shift and every shift sends D10/40 pulses for 3000/20 ms.

478. When X0 = OFF, the output stops immediately and starts from the count value in D1030, D1031 when PLSR is executed again.

479. Ramp up/down shifts for Y0, Y2: 20. Ramp up/down shifts for Y1, Y3: 10



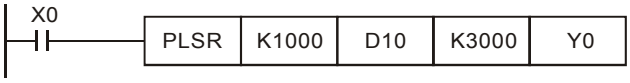
Note:

480. When the PLSR instruction is used in a program, the outputs used in the PLSR instruction cannot be used in the PLSY instruction or PWM instruction.

481. When several pulse output instructions (PLSY, PWM, PLSR) use Y0 as the pulse output in the same program, and if they are executed simultaneously in the same scan cycle, the ELC will perform the instruction with the fewest step numbers.

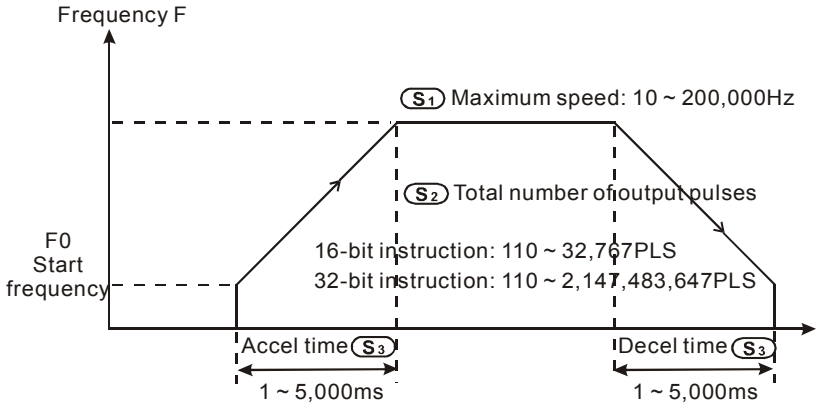
Functions in ELC-PV, ELC2-PV series:

1. Relevant devices for the ELC-PV, ELC2-PV series:



2. The range of the pulse rate for this instruction is 10 ~ 200,000Hz. If the values of maximum speed and acceleration/deceleration time exceeds the range, the ELC will operate with the default value that is within the range.

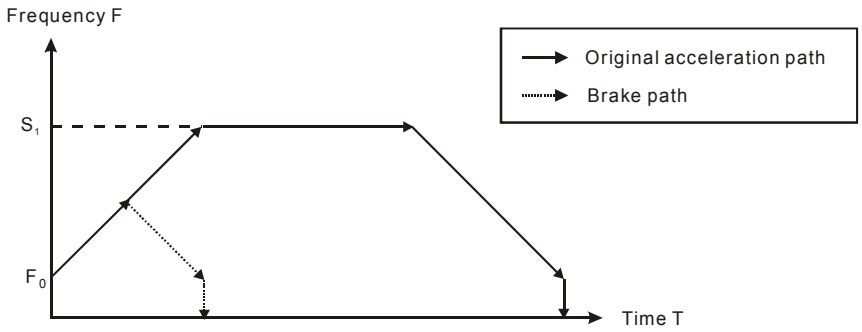
Operand		S_1	S_2	S_3	D
Explanation		Max. frequency	Total number of pulses	Accel/Decel time	Output point
Range	16-bit	10 ~ 32,767Hz	110 ~ 32,767	1 ~ 5,000ms	Y0 ~ Y7
	32-bit	10 ~ 200KHz	110 ~ 2,147,483,647		
Definition		K0: No output Kn: Designated frequency	Kn: Designated number	Flag: M1067, M1068	See settings of D1220, D1221



3. The acceleration/deceleration of the PV series is based on the number of pulses. If the output cannot reach the maximum acceleration frequency within the acceleration/deceleration time, the instruction will automatically adjust the acceleration/deceleration time and the maximum frequency.

4. The operands must be set before the execution of the instruction.

5. All acceleration/deceleration instructions are included with the brake function. The brake function will be enabled when the ELC is performing acceleration and the switch contact is turned Off. The deceleration will operate at the slope of the acceleration.



Functions in ELCM-PH/PA, ELC2-PA/PB/PH/PE series:

482. Description on associated flags:

For M1029, M1030, M1102, M1103, M1078, M1079, M1104, M1105, M1538, M1539, M1540, M1541, M1347, M1348, M1524, M1525, please refer to PLSY instruction for more information.

M1108: Pause output Y0 pulses (ramp down). ON = pause, OFF = resume

M1109: Pause output Y1 pulses e (ramp down). ON = pause, OFF = resume

M1110: Pause output Y2 pulses (ramp down). ON = pause, OFF = resume

M1111: Pause output Y3 pulses (ramp down). ON = pause, OFF = resume

M1156: Enabling the mask and alignment mark function on I400/I401(X4) corresponding to Y0.

M1257: Set the ramp up/down for Y0, Y2 to be “S curve.” ON = S curve, OFF = straight line

M1158: Enabling the mask and alignment mark function on I600/I601(X6) corresponding to Y2.

M1534: Enable ramp-down time setting for Y0. Must be used with D1348

M1535: Enable ramp-down time setting for Y2. Must be used with D1349

483. Description of associated special registers:

For D1030~D1033, D1336~D1339, D1220, D1221, please refer to the PLSY instruction

D1026: M1156 = ON, D1026 stores the number of pulses for masking Y0 (LOW WORD).

D1027: M1156 = ON, D1026 stores the number of pulses for masking Y0 (HIGH WORD).

D1135: M1158 = ON, D1135 stores the number of pulses for masking Y2 (LOW WORD).

D1136: M1158 = ON, D1135 stores the number of pulses for masking Y2 (HIGH WORD).

D1232: The number of output pulses for ramp-down stop when Y0 mark sensor receives signals. (LOW WORD).

D1233: The number of output pulses for ramp-down stop when Y0 mark sensor receives signals. (HIGH WORD).

D1234: The number of output pulses for ramp-down stop when Y2 mark sensor receives

signals (LOW WORD).

D1235: The number of output pulses for ramp-down stop when Y2 mark sensor receives signals (HIGH WORD).

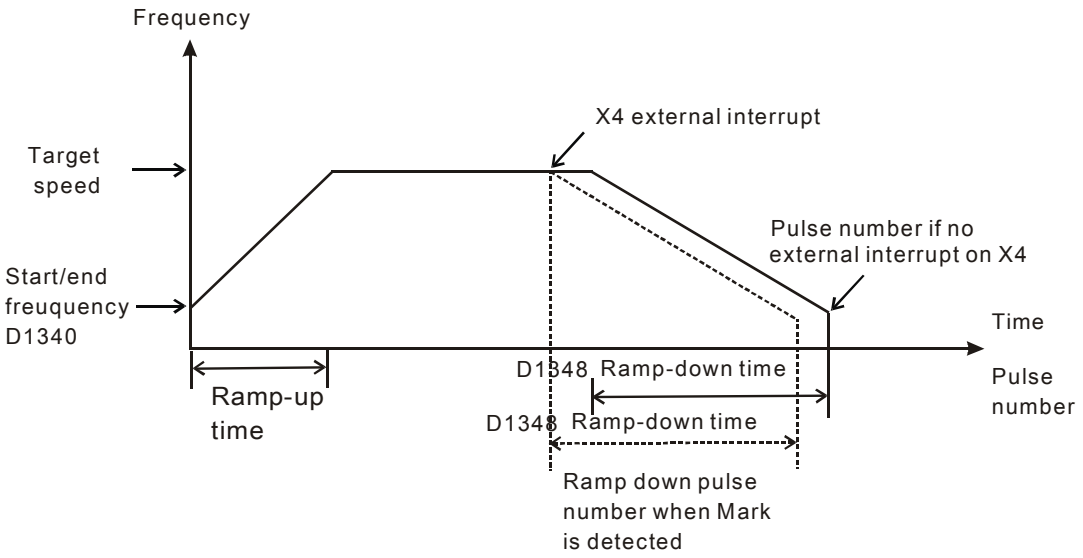
D1348: M1534 = ON, D1348 stores the ramp-down time of the CH0(Y0, Y1) pulse output.

D1349: M1535 = ON, D1349 stores the ramp-down time of the CH1(Y2, Y3) pulse output.

D1340 Start/end frequency of the pulse output CH0 (Y0, Y1)

D1352 Start/end frequency of the pulse output CH1 (Y2, Y3)

484. Operation of Mark function on Y0:



a) When M1156 = ON, enable ramp-down pause on Y0 when X4 receives interrupt signal.

b) When Mark function is enabled, ramp down time is independent of the ramp up time. Users can set the ramp up time in **S_3** and ramp down time in D1348. (Range: 20ms~32767ms)

c) When the PLSR instruction is executed, the ELC will automatically calculate the minimum pulse number required for ramp down when Mark is detected, and store the number of pulses in D1232~D1233

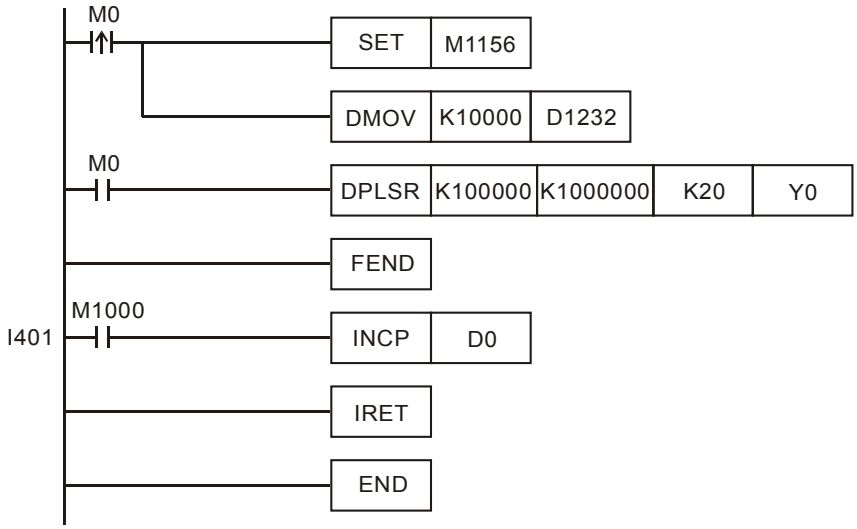
d) The number of ramp down pulses for the Mark function can be specified by users. However, the specified number should be more than the number the ELC calculated for the ramp-down time, otherwise the ELC will fill D1232~D1233 with the minimum Number of pulses.

e) When the Mark signal is detected, M1108 = ON. The Y0 pulse output will pause during the ramp down process.

f) Y0,Y2 relative parameters for Mask and Alignment Mark function:

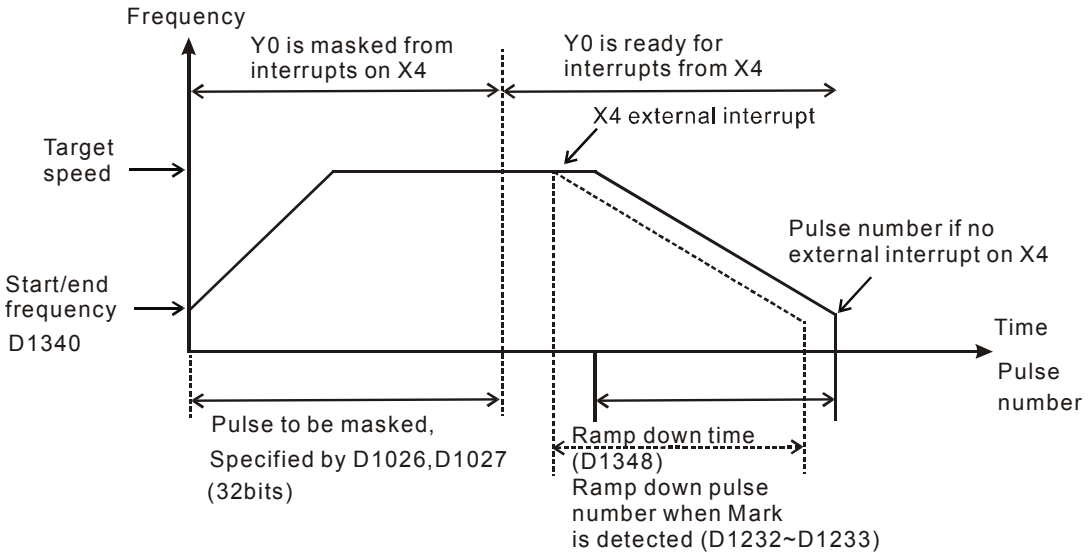
Parameter / Output	Mark flag	Input points	Ramp down time	Pulse number for masking output	Pulse number for ramp-down of Mark function	Output pause (ramp down)	Pause status
Y0	M1156	X4	D1348	D1026, D1027	D1232, D1233	M1108	M1538
Y2	M1158	X6	D1349	D1135, D1136	D1234, D1235	M1110	M1540

g) Program example 1:



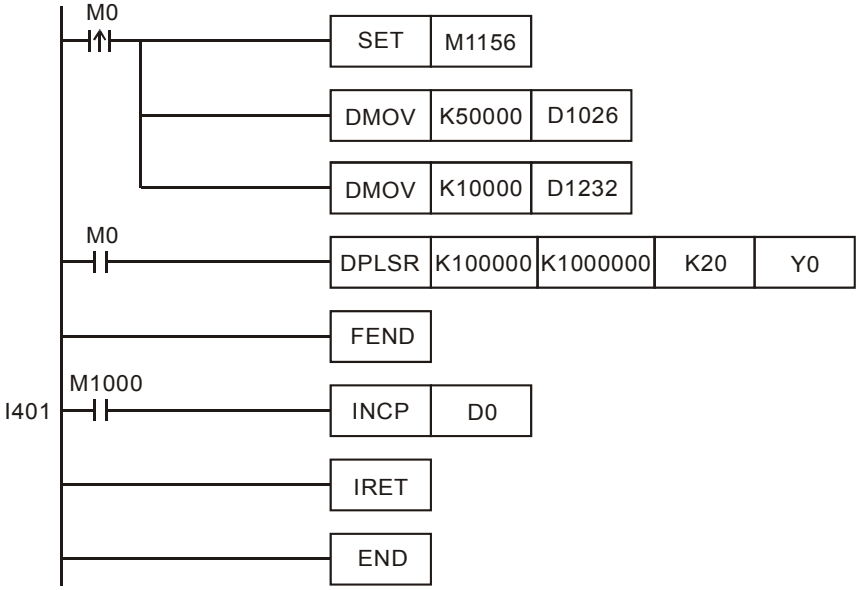
- When M0 is triggered, Y0 executes the pulse output. If an external interrupt is detected on X4, the pulse output will execute the ramp down process for 10,000 pulses and then stop. M1108 will be ON to indicate the pause status (ramp down). If no interrupt is detected, Y0 pulse output will stop after 1,000,000 pulses are completed.
- When the pulse output ramps down and stops after the Mark is detected, M1538 will be ON to indicate the pause status. If the remaining pulses need to be completed, set the M1108 flag OFF and the pulse output will resume.

485. Operation of the Mask function on Y0:



a) The Mask function is enabled when D1026 and D1027 contain values other than 0. The Mask function is disabled when D1026 and D1027 contain a value of 0.

b) Program example 2:



- When M0 is triggered, the DPLSR instruction begins sending pulses out Y0.. When an external interrupt is detected at X4 after 50,000 pulses have been sent, the pulse output will perform the ramp down process for 10,000 pulses and then stop. M1108 will be ON. If no interrupt is detected at X4, pulses will continue to be sent out Y0 until 1,000,000 pulses have been sent, then it will stop..
- An interrupt triggered between 0 ~ 50,000 pulses will be invalid, i.e. no ramp-down process will be performed before 50,000 pulses have been sent..

API	Mnemonic	Operands	Function
60	IST	**S, D_1, D_2**	Manual/Auto Control

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*													IST: 7 steps
D_1				*												
D_2				*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

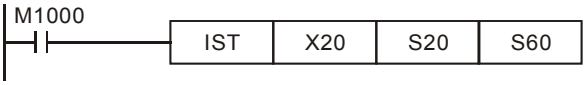
Operands:

S: The starting input address (Operand **S** will occupy 8 continuous addresses). **D_1** The smallest number for the designated-status step point in auto mode. **D_2**: The greatest number for the designated-status step point in auto mode.

Description:

486. The IST is a convenient instruction made specifically for the initial state of the step function control procedure.
487. ELCB-PB model, the range **D_1** and **D_2** = S20~S127 and **D_1** < **D_2**. ELC-PA/PV and ELC-PV2 models, the range **D_1** and **D_2** = S20~S899 and **D_1** < **D_2**. ELCM-PH/PA and ELC2-PB/PH/PA/PE models, the range **D_1** and **D_2** = S20~S911 and **D_1** < **D_2**.
488. IST instruction can only be used one time in a program.

Program Example 1:



S: X20: Individual operation (Manual operation) X24: Continuous operation
X21: Zero point return X25: Zero point return start switch
X22: Step operation X26: Start switch
X23: One cycle operation X27: Stop switch

489. When the IST instruction is executed, the following special auxiliary relays will switch automatically.
M1040: Movement inhibited S0: Manual operation/initial state step point
M1041: Movement start S1: Zero point return/initial state step point
M1042: Status pulse S2: Auto operation/initial state step point
M1047: STL monitor enable
490. When the IST instruction is used, S10~S19 are for the zero point return operation and the step point of this state can't be used as a general step point. However, when using S0~S9 step points, S0 initiates "manual operation", S1 initiates "zero point return operation" and S2

initiates “auto operation”. Thus, there should be three circuits of these three initial state step points written first in the program.

491. When switching to S1 (zero point return mode), zero point return won’t take any action once any of S10~S19 = ON.

492. When switching to S2 (auto operation mode), auto operation won’t take any action when S is between $\mathbf{D_1}$ to $\mathbf{D_2}$ or if M1043=ON

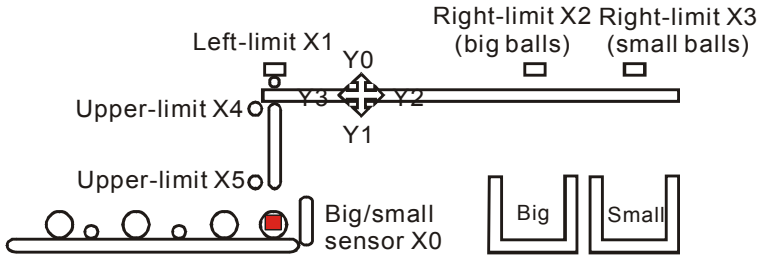
Program Example 2:

The Robot arm control (using the IST instruction):

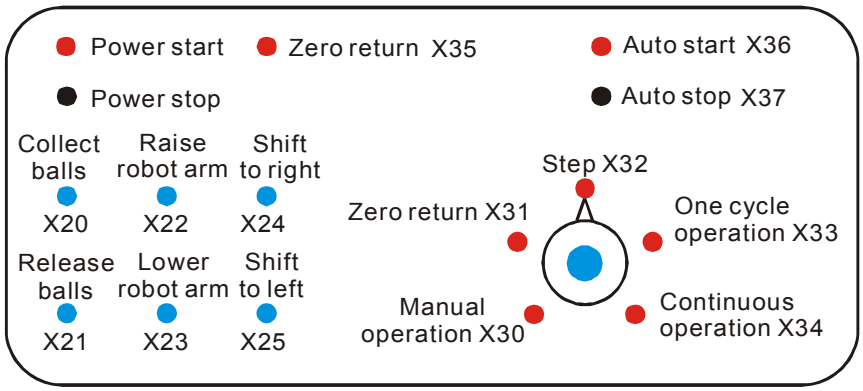
493. Control purpose: Select the big balls and small balls and move them to corresponding boxes. Configure the control panel for each operation.

494. Motion of the Robot arm: lower robot arm, collect balls, raise robot arm, shift to right, lower robot arm, release balls, raise robot arm, shift to left to finish motion in order.

495. I/O Device



496. Control panel

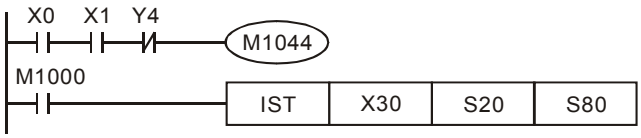


a) Big/small sensor X0.

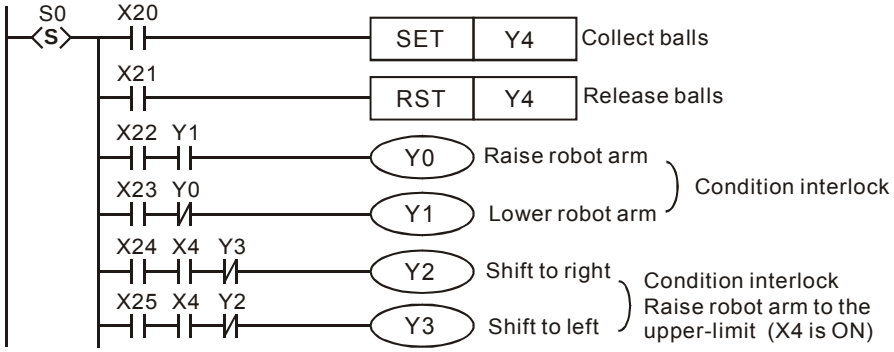
b) The left-limit of the robot arm X1, the right-limit X2 (big balls), the right-limit X3 (small balls), the upper-limit X4, and the lower-limit X5.

c) Raise robot arm Y0, lower robot arm Y1, shift to right Y2, shift to left Y3, and collect balls Y4.

497. START circuit:

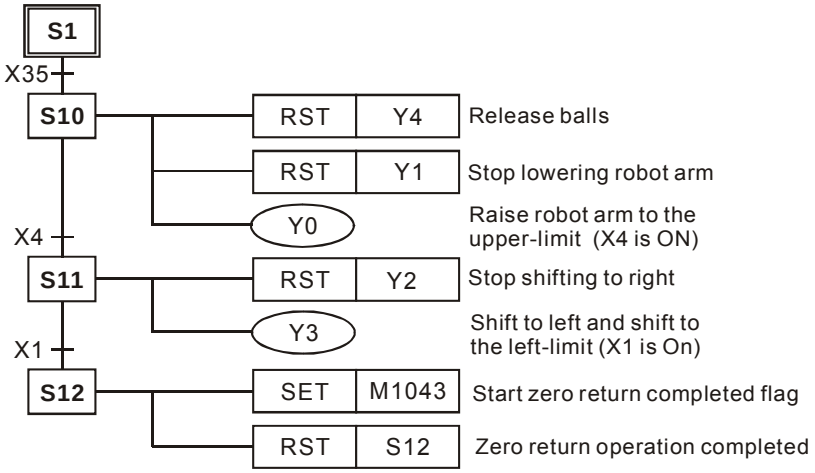


498. Manual operation mode:

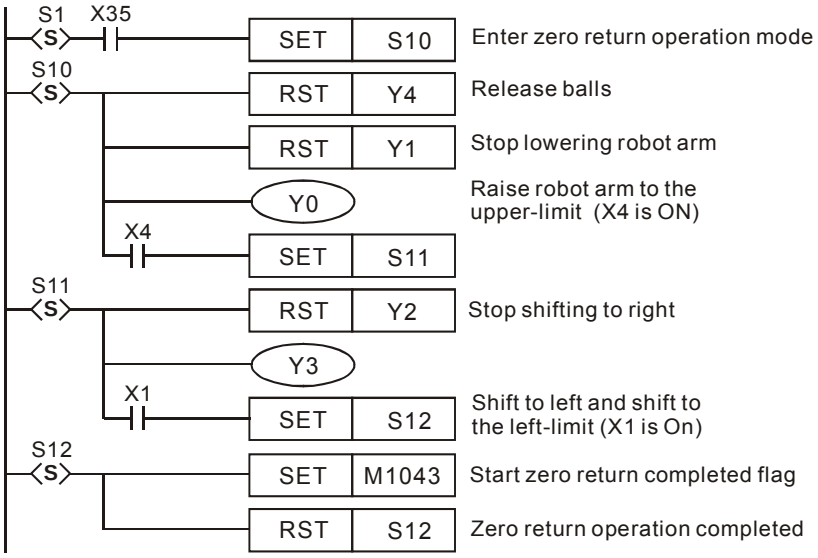


499. Zero point return mode:

a) SFC figure:

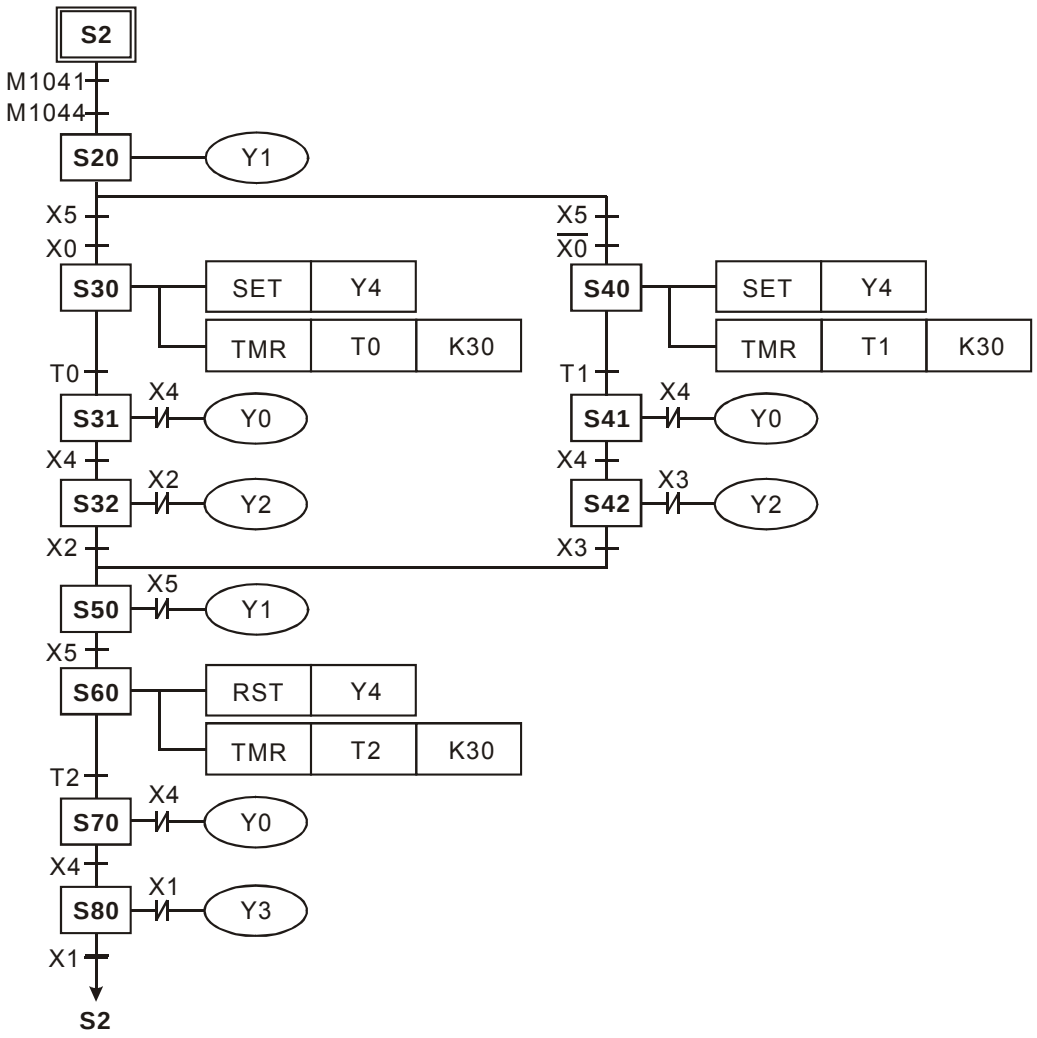


b) Ladder Diagram:

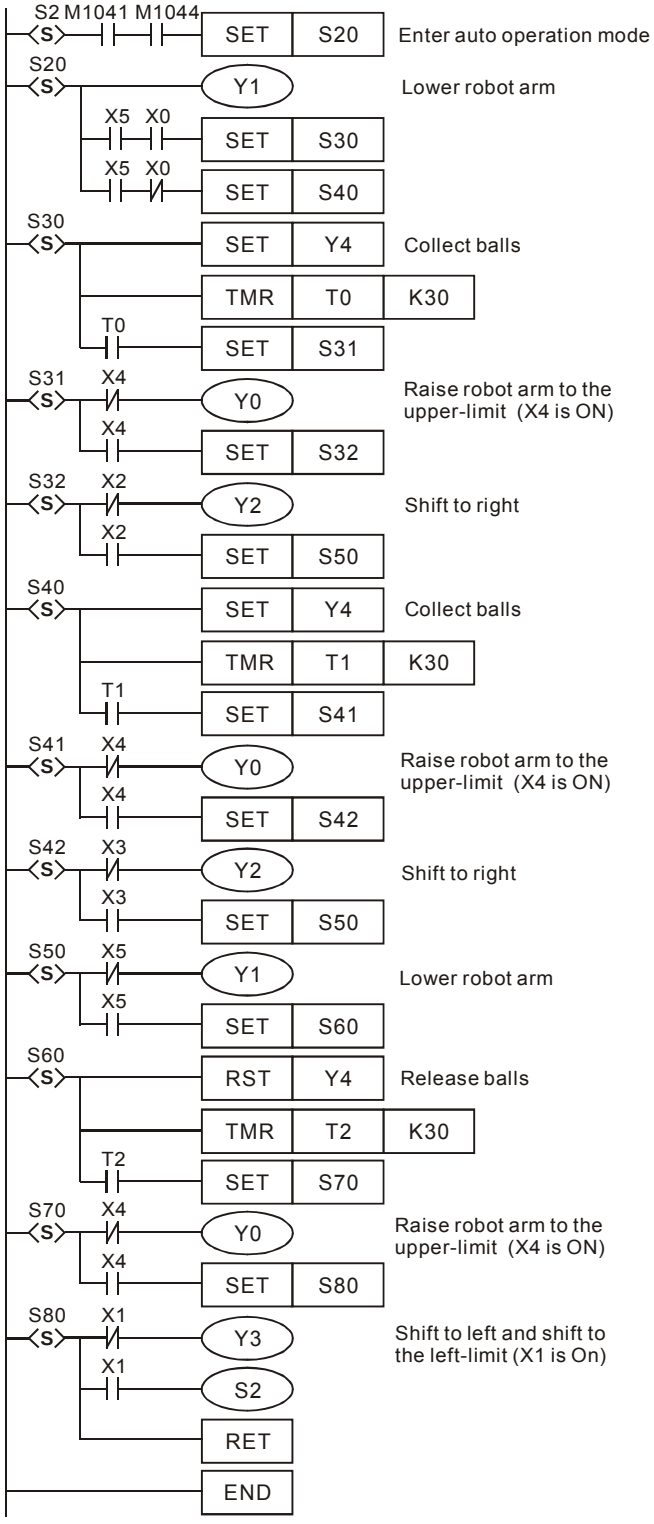


500. Auto operation (step/one-cycle/continuous operation modes):

a) SFC figure:



b) Ladder Diagram:



Flag explanation:

M1040:

Step point movement disabled. When M1040=ON, all movements of the step point are disabled.

501. **Manual operation mode:** M1040 = ON.
502. **Zero point return mode/one cycle operation mode:** Pressing the STOP button and pressing START button again, M1040 = ON.
503. **Step operation mode:** M1040 = ON, and will only be OFF when the START button is pressed.
504. **Continuous operation mode:** When ELC goes from STOP→RUN, M1040 = ON, and will be OFF when the START button is pressed.

M1041:

Step point movement start. the special auxiliary relay that reflects the movement of the primary step point (S2) to the next step point.

505. **Manual operation mode/Zero point return mode:** M1041 = OFF.
506. **Step operation mode/One cycle operation mode:** M1041 = ON when the START button is pressed.
507. **Continuous operation mode:** Stays ON when the START button is pressed, and turns OFF when the STOP button is pressed.

M1042:

START pulse: Only once pulses will be sent out when the button is pressed.

M1043:

Zero point return complete: Once M1043 =ON, indicates that the RESET motion has been executed.

M1044:

Conditions of the origin: In continuous operation mode, conditions of the origin, M1044= ON to execute the initial step point (S2) moving to the next step point.

M1045:

All output reset inhibit. If executing conditions:

508. From manual control S0 to zero point return S1
509. From auto operation S2 to manual operation S0
510. From auto operation S2 to zero point return S1
    a) When M1045=OFF and one of S of **D_1~D_2** is ON, step point of SET Y output and actions will be cleared to OFF.
    b) When M1045 =ON, SET Y output will be reserved, and step point during action will be

cleared to OFF.

c) If executing from zero point return S1 to manual operation S0, no matter if M1045=ON or M1045=OFF, SET Y output will be reserved, and step point action will be cleared to OFF.

M1046:

When STL action = ON: If one of step point S is ON, M1046=ON. After M1047 = ON, M1046 = ON once one of S is ON. Besides, 8 prior points numbers is ON of step point S will be recorded in D1040~D1047.

M1047:

STL monitor enabled. When IST instruction starts executing, M1047 will be forced to be ON and it will be forced to ON for each scan time once IST instruction is still ON. This flag is used to monitor all S.

D1040~D1047:

ON state number 1-8 of step point S.

API	Mnemonic			Operands	Function
61	D	SER	P	**S_1, S_2, D, n**	Search a Data Stack

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			SER, SERP: 9 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	DSER, DSERP: 17 steps
D								*	*	*	*	*	*			
N					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Starting source address **S_2**: Compare value **D**: Starting destination for storing compared result (occupies 5 continuous addresses) **n**: Number of addresses to compare

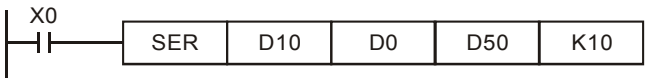
Description:

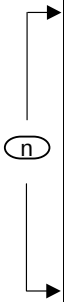
511. **S_1** specifies the starting address of the registers to compare, **n** specifies how many registers to compare to the value specified by **S_2**, and the compare result is stored in destination registers specified by **D**.
512. **D** stores the total of the matched results; **D+1** stores the address number of the first matched result; **D+2** stores the address number of the last matched result; **D+3** stores the address number of the smallest value; **D+4** stores the address number of the largest value.
513. If the 32-bit instruction is used, operands **S_1**, **S_2**, **D**, **n** will reference 32-bit registers.
514. If operand **S_2** uses index F, only 16-bit instruction is available
515. The range of operand **n**: **n**=1~256 (16-bit instruction), **n**=1~128 (32-bit instruction)

Program Example:

516. When X0=ON, the data stack D10~D19 is compared against D0 and the result is stored in D50~D54. If there are no equal values, the contents of D50~D52 will be 0.

517. The offset into the data file of the largest value of all compared data will be stored in D54 and the offset into the data file of the smallest value of all compared data will be stored in D53. When there are more then one largest value and/or smallest value, only the last location in the data file will be recorded for each. For example, if there were a K5 in D17 and D18 in the table below, the value stored in D53 will be 8 instead of 7. This is because the last occurance of the smallest value in this case is in position 8 in the data file.





S_1	Content value	Compare data	Data number	Result
D10	88	S_2	0	
D11	100		1	Equal
D12	110		2	
D13	150	D0=K100	3	
D14	100		4	Equal
D15	300		5	
D16	100		6	Equal
D17	5		7	Smallest
D18	100		8	Equal
D19	500		9	Largest

D	Content value	Explanation
D50	4	The number of matches found
D51	1	The offset into the file where the first match is located
D52	8	The offset into the file where the last match is located
D53	7	The offset into the file where the smallest value is located
D54	9	The offset into the file where the largest value is located

API	Mnemonic	Operands	Function
62	D ABSD	**S_1, S_2, D, n**	Absolute Drum Sequencer

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			ABSD: 9 steps
S_2											*	*	*			DABSD: 17 steps
D		*	*	*												
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

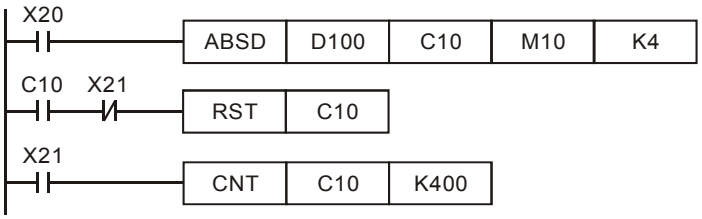
S_1: Starting address of the data compare file **S_2**: Counter number **D**: Starting address of compare result **n**: Groups of multi-step comparison (**n**=1~64)

Description:

518. The ABSD instruction creates various output wave forms according to the current value of the counter designated by **S_2**. The instruction is typically used for absolute cam control applications.
519. **S_2** of DABSD specifies a high-speed, 32-bit counter. However, when the current value of the high-speed counter is compared against the set-point value, the result cannot happen immediately, because it is influenced by the scan time. If an immediate output is required, use the DHSZ instruction.
520. When operand **S_1** uses KnX, KnY, KnM or KnS, where n=4 only a 16-bit instruction can be used. Use K8 for the 32-bit instruction.

Program Example:

521. Before executing the ABSD instruction, preload the set-point values into D100~D107. The contents of the even number D register is the lower-limit value and the contents of the odd number D register is the upper-limit value.
522. When X20=ON, the current value of counter C10 is compared against the upper and lower-limit values of D100~D107 (four groups). The compared result is displayed in M10~M13.
523. When X20=OFF, the origin ON/OFF state of M10~M13 will be unchanged.

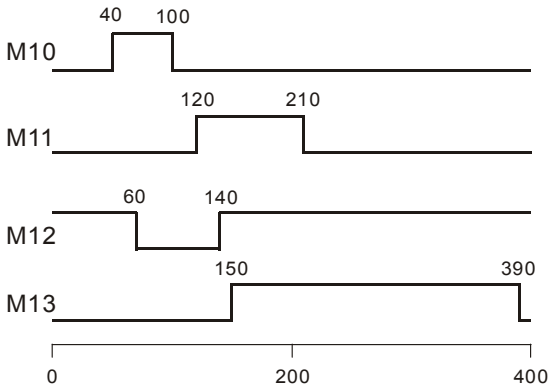


524. M10~ M13 = ON when the current value of C10 is equal to or greater than the lower-limit value and equal to or less than the upper-limit value.

Lower-limit value	Upper-limit value	Current value of C10	Output
D100= 40	D101=100	40≦C10≦100	M10=ON
D102=120	D103=210	120≦C10≦210	M11=ON
D104=140	D105=170	140≦C10≦170	M12=ON
D106=150	D107=390	150≦C10≦390	M13=ON

525. When the lower-limit value is greater than the upper-limit value, if the current value of C10 is greater than the lower-limit value (C10>140) and less than the upper-limit value (C10<60), M12=ON.

Lower-limit value	Upper-limit value	Current value of C10	Output
D100= 40	D101=100	40≦C10≦100	M10=ON
D102=120	D103=210	120≦C10≦210	M11=ON
D104=140	D105= 60	60≦C10≦140	M12=OFF
D106=150	D107=390	150≦C10≦390	M13=ON



API	Mnemonic	Operands	Function
63	INCD	S_1, S_2, D, n	Incremental drum sequencer

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			INCD: 9 steps
S_2												*				
D		*	*	*												
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

3

Operands:

S_1: Starting address of the compare data table S_2: Counter number **D**: Starting address of compare result **n**: Groups of multi-step comparison (**n**=1~64)

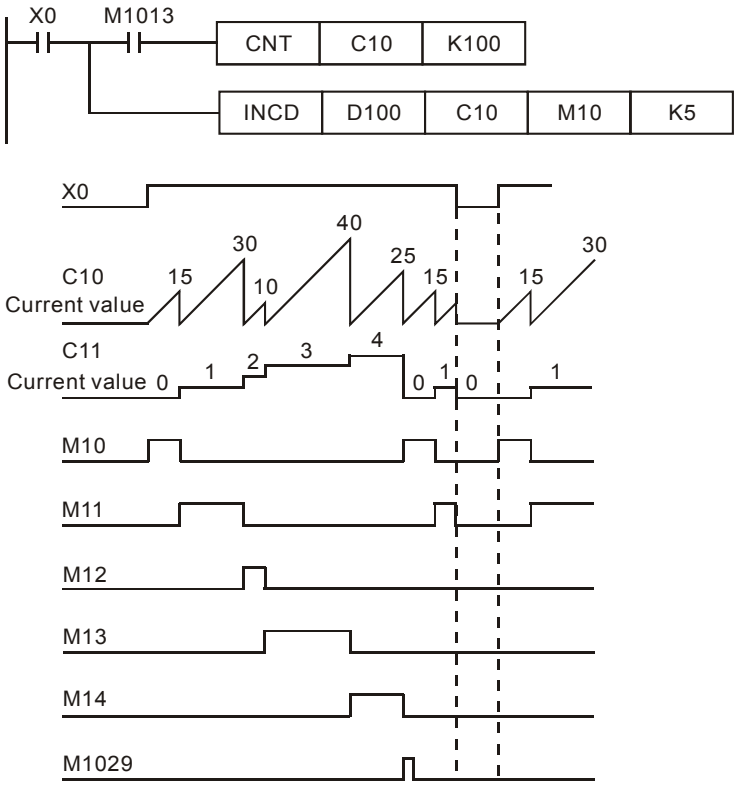
Description:

526. The INCD instruction is a multi-step comparison instruction and is typically used for relative cam control.
527. The current value of S_2 is compared against the set-point values of S_1, one at a time.. Once the current value is equal to the present set-point value, the current value of S_2 will be reset to 0 and be compared again. The total number of equal comparisons is stored in S_2+1.
528. When the comparison of n groups of data has been completed, the execution complete flag M1029 = ON for one scan cycle.
529. When operand S_1 is specified as KnX, KnY, KnM or KnS, n=4 must be used.
530. In 16-bit instructions, operand S_2 must be C0~C198 and will occupy 2 consecutive counters.
531. **Flag:** M1029 is the execution complete flag.

Program Example:

532. Before executing the INCD instruction, preload the set-point values into D100~D104 in advance. D100=15, D101=30, D102=10, D103=40, D104=25.
533. The current value of counter C10 is compared against the set-point values of D100~D104, one at a time. Each time the current value is equal to the current set-point value, the current value of C10 will be reset to 0 and will be compared again. Meanwhile C11 keeps a count of the number of equal comparisons..
534. When the content of C11 increments by 1, M10~M14 will also change. The state of each of these bits indicates where in the compare cycle the instruction is. Refer to the timing diagram below.
535. When the comparison of 5 groups of data has been completed, the execution completed flag M1029 = ON for one scan cycle.

536. When X0 turns from ON →OFF, C10 and C11 will both be reset to 0 and M10~M14 =OFF. When X0 turns ON again, this instruction will be executed again.



API	Mnemonic	Operands	Function
64	TTMR	D, n	Alternate Timer

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D													*			TTMR: 5 steps
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

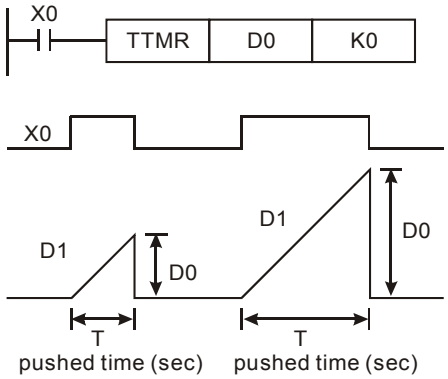
D: Device number for storing the ON time of the input **n**: Multiple set-points (**n**=0~2)

Description:

537. The ON time of the external switch is measured and stored in **D** +1. The units are 100ms increments.
538. **n** determines the units for **D.** When **n**=0, **D** is in seconds. When **n**=1, **D** is in 100ms increments.. When **n**=2, **D** is in 10ms increments.
539. Operand **D** occupies 2 consecutive registers.
540. ELC-PA, ELCM-PH/PA, ELC2-PB/PH/PA/PE: The TTMR instruction can only be used eight times in a program.

Program Example 1:

541. The On duration of X0 is stored in D1 in 100ms increments. **n** is used to specify the units for the On duration of X0 stored in D0. Then the switch can be used to adjust the set-point value of a timer, for example.
542. When X0 = OFF, the contents of D1 will be reset to 0 but the contents of D0 is unchanged.

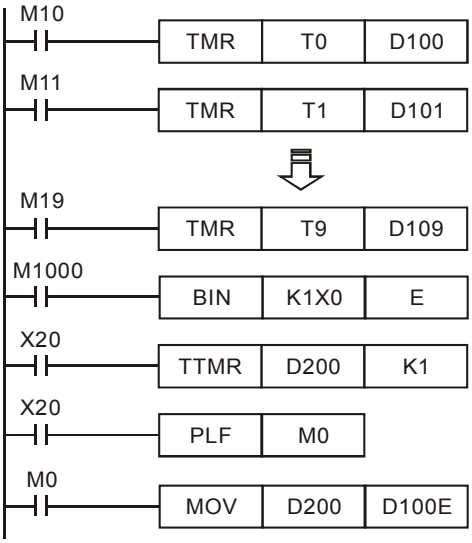


543. If the ON duration of X0 is T seconds, the relationship between D0, D1 and **n** are shown as the table below.

n	D0	D1(unit: 100 ms)
K0 (unit: s)	1×T	D1=D0×10
K1 (unit: 100 ms)	10×T	D1=D0
K2 (unit: 10 ms)	100×T	D1=D0/10

Program Example 2:

544. Using the TTMR instruction write preset values to 10 timers.
545. Write the preset values for timers T0-T9 to D100~D109. Use the thumbwheel switch wired to X0-X4 to determine which timer's preset to set..
546. T0~T9 are 100ms timebase timers. So, we will want the time values measured with the TTMR instruction to also be in 100ms increments, because these values will be moved to the presets for timers T0-T9.
547. Connect a single digit thumbwheel switch to X0~X3 and use the BIN instruction to convert the set-point value of the switch to a BIN value and move it to the index register E. This allows the thumbwheel switch to determine the timer number to move a new preset values to.
548. The ON duration (in 100ms increments) of X20 is stored in D200.
549. M0 is a pulse for one scan cycle generated when the alternate timer button X20 is released.
550. Use the set-point number of the thumbwheel switch as the pointer for index register E. Then move the new preset value to the appropriate timer. If E=0, the contents of D200 is moved to the preset for timer T0 (D100). If E=1, the contents of D200 is moved to the preset for timer T1 (D101) and so on.



Note:

ELC-PA, ELCM-PH/PA, ELC2-PB/PB/PA/PE models, can only use the TTMR instruction eight times

in a program. If used in a CALL subroutine or interrupt subroutine, it only can be use once.

API	Mnemonic	Operands	Function
65	STMR	**S, m, D**	Special Timer

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*					STMR: 7 steps
m					*	*										
D		*	*	*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Timer number **m**: Set-point value of the timer (**m**=1~32,767), units in 100ms

D: Starting address of the results (occupies 4 continuous bits)

Description:

551. The STMR instruction can be used for OFF-delay, ON/OFF triggering, or as a flashing circuit.
552. The timer number (**S**) specified by STMR instruction can be used only once.
553. Range of **S**: ELC-PA T0 ~ T191; ELC-PV T0 ~ T199; ELCM-PH/PA, ELC2-PB/PA/PE/PV T0 ~ T183

Program Example:

554. The preset value of timer T0 is 5 seconds.
555. Y0 is the OFF-delay output :

a) When X20 turns from OFF →ON, Y0= ON.

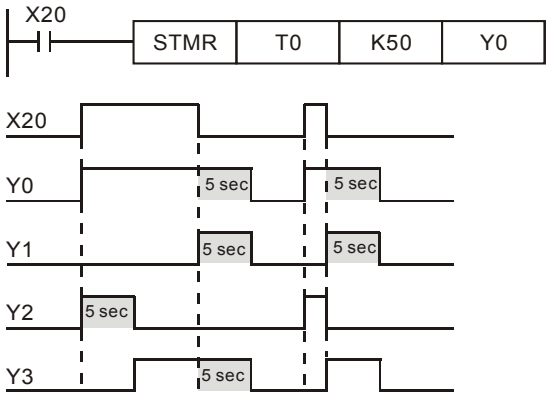
b) When X20 turns ON→OFF after a delay 5 seconds, Y0=OFF.

c) When X20 turns from ON→OFF, Y1= ON for 5 seconds, then turns Off.

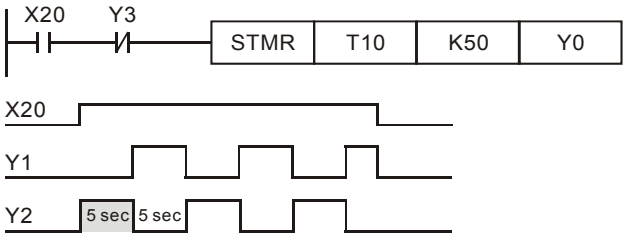
d) When X20 turns from OFF→ON, Y2=ON for 5 seconds, then turns Off.

e) When X20 turns from OFF→ON, Y3=ON after a 5 second delay.

f) When X20 turns from ON→OFF, Y3=OFF after a 5 second delay.



g) Add a b contact addressed with Y3 (LDI Y3) in series with contact X20. Then Y1 and Y2 can be used for a flash circuit. Y1 and Y2 will alternate, each on for 5 seconds, then off for 5 seconds. When X20 turns OFF, Y0, Y1 and Y3 = OFF and the content of T10 will be reset to 0.



API	Mnemonic		Operands	Function
66	ALT		D	Alternate ON/OFF

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D		*	*	*												ALT, ALTP: 3 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

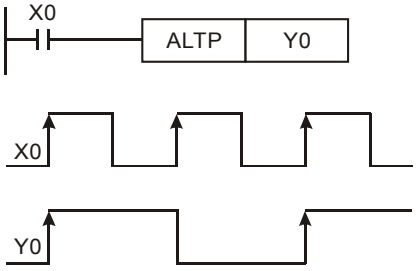
D: Destination address

Description:

556. The status of the destination address (**D**) alternates between on and off with each transition of the ALT instruction..
557. This toggling of **D** will occur every program scan unless the pulse option P is used with this instruction..
558. The ALT instruction is ideal for switching between two modes of operation e.g. start and stop, ON and OFF etc.
559. This instruction should be used as a pulse instruction (ALTP)

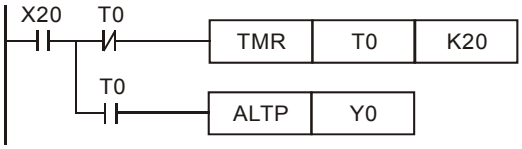
Program Example 1:

When X0 turns from OFF →ON for the first time, Y0=ON. When X0 turns from OFF →ON for the second time, Y0=OFF.



Program Example 2:

When X20= ON, T0 will generate a pulse every two seconds and output Y0 will toggle between on and off every 2 seconds..



API	Mnemonic	Operands	Function
67	RAMP	S_1, S_2, D, n	Ramp variable Value

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			RAMP: 9 steps
S_2													*			DRAMP: 17 steps
D													*			
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Starting set-point of ramp signal **S_2**: Ending set-point of ramp signal **D**: Current value of ramp signal **n**: Scan times (**n**=1~32,767)

Description:

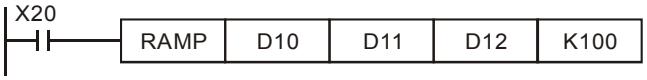
560. This instruction creates a ramp output. The linearity of the ramp depends on a consistent program scan time. Therefore, set the scan time to a fixed time prior to using this RAMP instruction. This can be done by setting the constant scan mode bit M1039 and the constant scan time register D1039 (ms).
561. When the RAMP instruction is executed, the ramp signal will ramp from **S_1** to **S_2**. The Current value of the ramp signal is stored in **D. D+1** stores the current number of accumulated scans. When ramp signal reaches **S_2**, or when the conditions preceding the RAMP instruction turn OFF, the contents in **D** varies according to the setting of M1026 which is explained later in **Points to note**.
562. When **n** specifies a D register, the value in D cannot be modified during the execution of the instruction. The contents of D can only be modified when the instruction is stopped.
563. In ELC-PV, DRAMP only supports version 1.4 and above.

Program example:

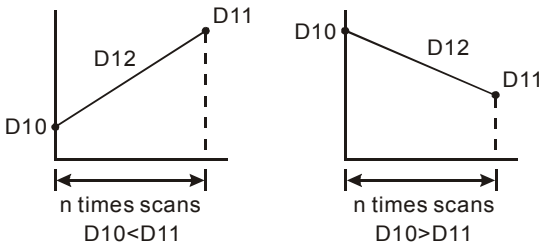
564. Before executing the instruction, first set bit M1039 = ON to fix the scan time and move a value in ms into the constant scan time register D1039. Assume 30 was moved into D1039, which fixes the scan time at 30ms. If n = K100, the time for D10 to increase to D11 will be 3 seconds (30ms × 100).
565. When X20 turns OFF, the instruction will stop execution. When X20 goes ON again, the content in D12 and D13 will be reset to 0.
566. When M1026 = OFF, the RAMP instruction will operate continuously. The ramp complete bit (M1029) will turn on for one scan at the end of each ramp cycle and the contents of D10 will be moved into D12 for the start of the next ramp cycle. When M1026 = ON, the RAMP

instruction will operate for one ramp cycle and stop. When X20 turns off, then on again, the contents of D10 will be moved into D12 and the next ramp cycle will execute. Each time the instruction completes a ramp cycle, the ramp complete bit M1029 will turn on. M1029 will reset when X20 turns off.

567. Set the Start and End of ramp signal in D10 and D11. When X20 = ON, D10 increases towards D11, the current value of the ramp is stored in D12 and the number of current scans is stored in D13.



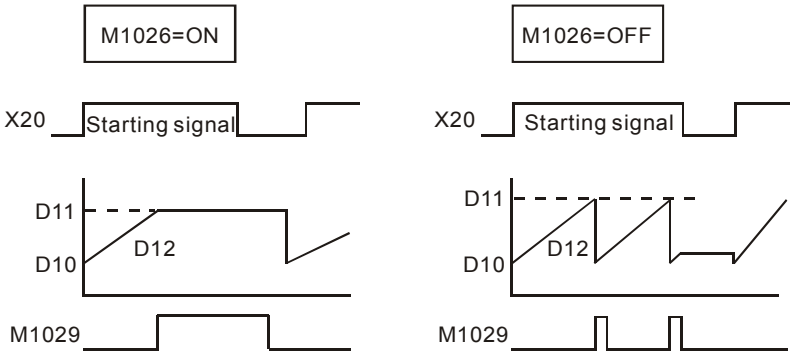
568. If X20 = ON,



Scan time is stored in D13

Points to note:

The ramp cycles of D12 based on the state of M1026:



API	Mnemonic	Operands	Function
68	DTM	S_1, D, m, n	Data Transform and Move

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			DTM: 9 steps
D													*			
m					*	*							*			
n					*	*							*			

ELCB	ELC	ELC2	ELCM

PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

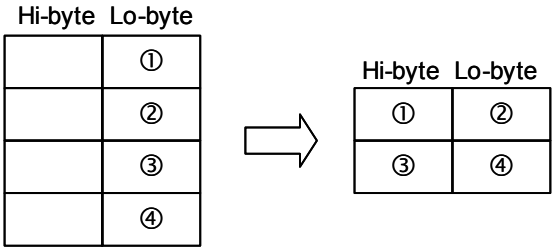
S_1: Starting address of the source data stack **D**: Starting address of the destination data stack

m: Transformation mode **n**: Length of the source data stack

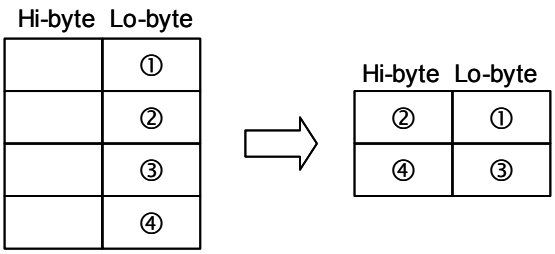
Description:

1. Operand **m**, the Transformation Mode can use any of the following data types: K, H, D**.** If the value is not in the available range, no transformation or move operation will be executed and no error will occur. The available range for m is K1-K19.
2. K, H, D data types can also be used for operand **n,** which determines the length of the source data stack. The range for **n** is K1~K256. If the value is out of range, the ELC will take the max value (256) or the min value (1) and use it for operand **n.**
3. The DTM instruction modes set in operand **m**:

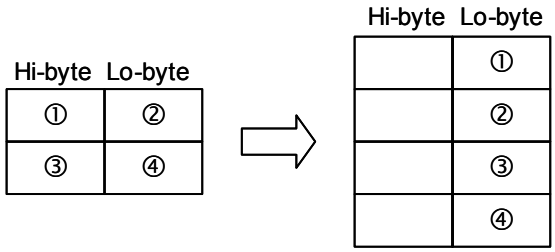
k0: Transform 8-bit data into 16-bit data (Hi-byte, Lo-byte) in the following format:



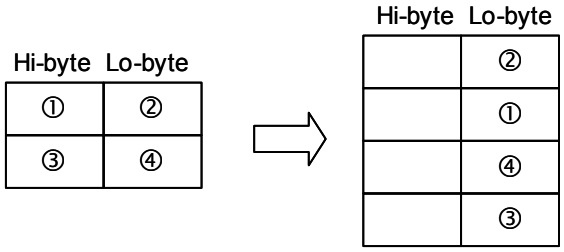
k1: Transform 8-bit data into 16-bit data (Lo-byte, Hi-byte) in the following format:



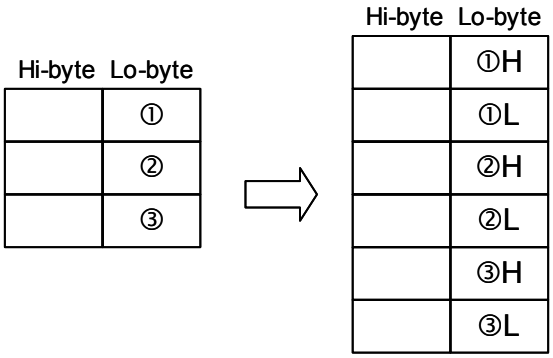
k2: Transform 16-bit data (Hi-byte, Lo-byte) into 8-bit data in the following format:



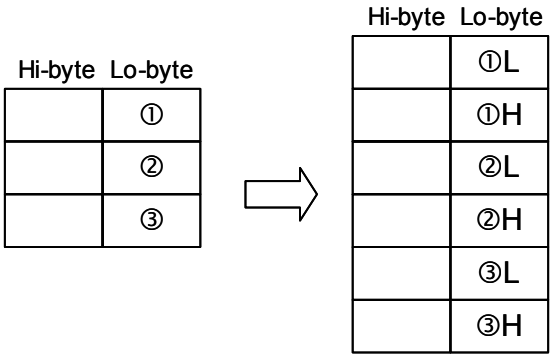
k3: Transform 16-bit data (Lo-byte, Hi-byte) into 8-bit data in the following format:



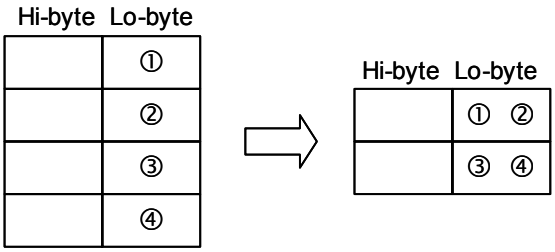
k4: Transform 8-bit HEX data into ASCII data (higher 4 bits, lower 4 bits) in the following format:



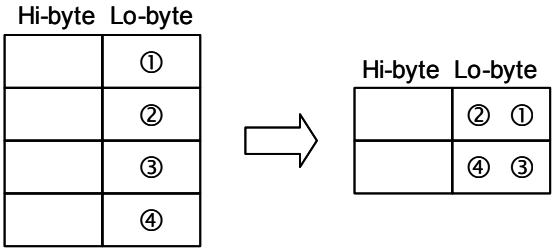
k5: Transform 8-bit HEX data into ASCII data (lower 4 bits, higher 4 bits) in the following format:



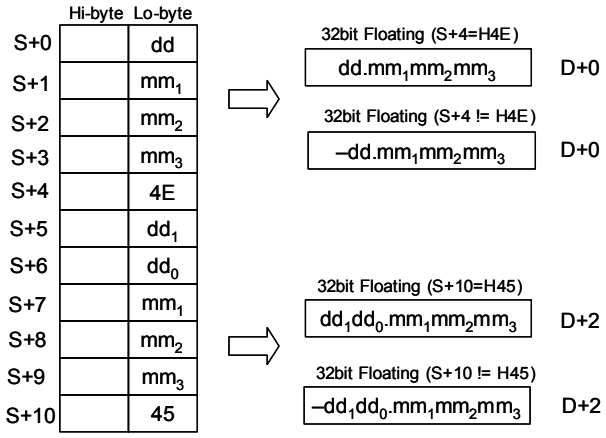
k6: Transform 8-bit ASCII data (higher 4 bits, lower 4 bits) into HEX data in the following format: (ASCII value to be transformed includes 0 ~ 9 (0x30~0x39), A ~ F (0x41~0x46), and a ~ f (0x61~0x66).)



k7: Transform 8-bit ASCII data (lower 4 bits, higher 4 bits) into HEX data in the following format:



K8: Transform 8-bit GPS data into 32-bit floating point data in the following format:



K9: Calculate the optimal frequency for the positioning instructions with the ramp up/ down function.

Only the total number of pulses need to be set up for positioning with the total time for positioning first. The DTM instruction will automatically calculate the optimal max output frequency as well as the optimal start frequency for positioning instructions with ramp-up/down function such as PLSR, DDRVI and DCLLM.

Points to note:

- When the calculation results exceed the max frequency of the ELC, the output frequency will be set as 0.
- When the total of ramp-up and ramp-down time exceeds the total time for operation, the ELC will change the total time for operation (**S**+2) into "ramp-up time (**S**+3) + ramp-down time (**S**+4) + 1" automatically.
- Description of the operands:

  S+0, **S**+1: Total number of pulses (32-bit value)

  S+2: Total time for operation (unit: ms)

  S+3: Ramp-up time (unit: ms)

  S+4: Ramp-down (unit: ms)

  D+0, **D**+1: Optimal max output frequency (unit: Hz) (32-bit)

  D+2: Optimal start frequency (Unit: Hz)

  n: Reserved

K11: Conversion from Local Time to Local Sidereal Time

Unlike the common local time defined by time zones, local sidereal time is calculated based on actual longitude. The conversion helps the user obtain the more accurate time difference of each location within the same time zone.

Explanation on operands:

S+0, **S**+1: Longitude (32-bit floating point value; East: positive, West: negative)

S+2: Time zone (16-bit integer; unit: hour)

S+3~ **S**+8: Year, Month, Day, Hour, Minute, Second of local time (16-bit integer)

D+0~**D**+5: Year, Month, Day, Hour, Minute, Second of the converted local sidereal time (16-bit integer)

n: Reserved

Example:

Input: Longitude F121.55, Time zone: +8, Local time: AM 8:00:00, Jan/6/2011

Conversion results: AM 8:06:12, Jan/6/2011

K12: Proportional Value Calculation Function of Multi-point Areas (16-bit values)

Explanation on operands (16-bit values):

S: input value

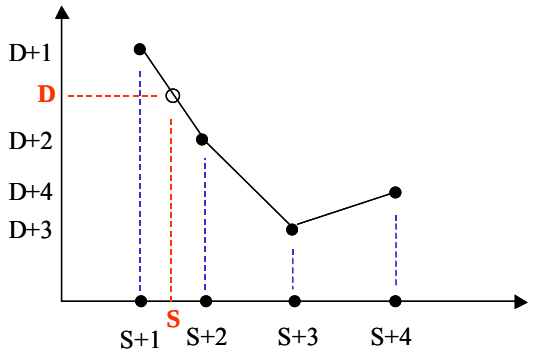
S+1, **S**+2….. **S**+n: set values of multi-point areas. **S**+1 must be the minimum value, **S**+2 must be larger than **S**+1 and so on. Therefore, **S**+n must be the maximum value.

D: output value gotten from the proportional value calculation

D+1, **D** +2 … **D**+n: the range of values gotten from the proportional value calculation

n: set values of multi-point areas. The range of set values is K2~K50. When the set value exceeds the range, it will not be executed.

The sample curve: (n is set to be K4)



The explanation of the sample:

1. When input value S is larger than S+1 (S_1 for short) and smaller than S+2 (S_2 for short), D+1 (D_1 for short) and D+2 (D_2 for short), D= ((S – S_1) x (D_2 – D_1) / (S_2 – S_1)) + D_1.
2. When input value S is smaller than S+1, D= D+1; when input value S is larger than S+n, D= D+n.
3. The operation of instructions uses floating-point values. After the decimal value of the output values is omitted, the value will be output in the 16-bit form.

K13: Proportional Value Calculation Function of Multi-point Areas (32-bit values)

The explanations of source and destination devices are illustrated as the explanation of K12, but devices S and D are indicated by 32-bit values.

K14: Proportional Value Calculation Function of Multi-point Areas (floating-point values)

The explanations of source and destination devices are illustrated as the explanation of K12, but devices S and D are indicated by 32-bit floating-point values.

K16: String combination

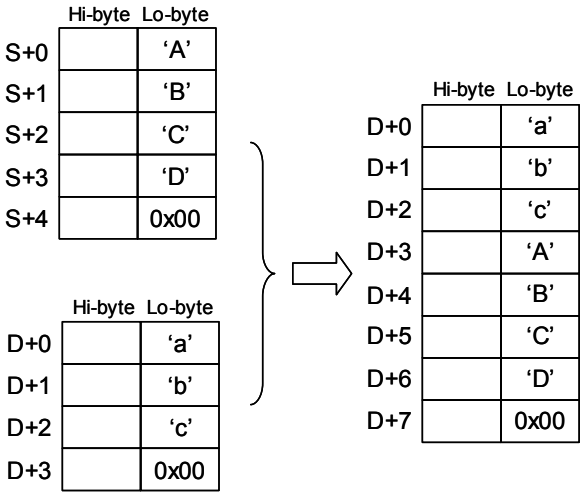
Explanation:

The system searches for the location of ETX (value 0x00) of the destination data string (lower 8 bits), then copies the data string starting of the source register (lower 8 bits) to the end of the destination data string. The source data string will be copied in byte order until the ETX (value 0x00) is reached.

Points to note:

The operand **n** sets the max data length after the string combination (max 256). If the ETX is not reached after the combination, the location indicated by **n** will be the ETX and filled with 0x00.

The combination will be performed in the following rule:

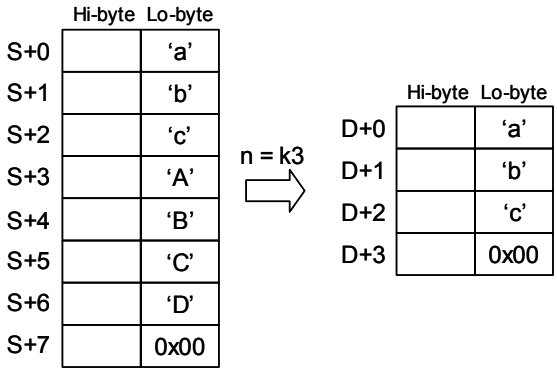


K17: String capture

Explanations:

The system copies the source data string (lower 8 bits) with the data length specified by operand n to the destination registers, where the n+1 register will be filled with 0x00. If value 0x00 is reached before the specified capture length n is completed, the capture will also be ended.

The capture will be performed in the following rule:



K18: Convert data string to floating point value

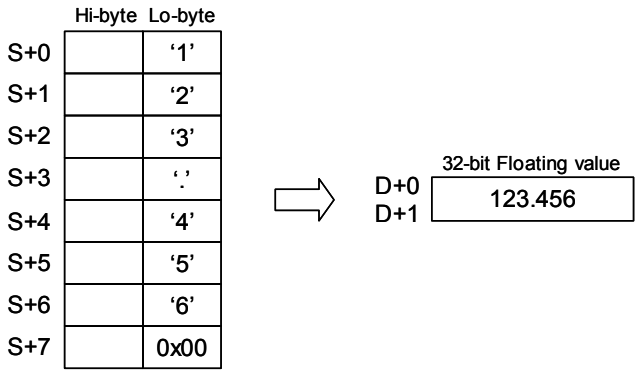
Explanations:

The system converts **n** words (lower 8 bits) of the source data string (decimal point is not included) to floating point value and stores the converted value in the destination device.

Points to note:

1. Operand **n** sets the number of total digits for the converted floating value. Max 8 digits are applicable and the value over **n** digit will be omitted. For example, **n** = K6, data string “123.45678” will be converted to “123.456”.
2. When there are characters other than numbers 0~9 or the decimal point in the source data string, the character before the decimal point will be regarded as 0, and the value after the decimal point will be regarded as the ETX.
3. If the source data string contains no decimal point, the converted value will be displayed by a **n**-digit floating point value automatically.

The conversion will be performed in the following rule:



K19: Convert floating point value to data string

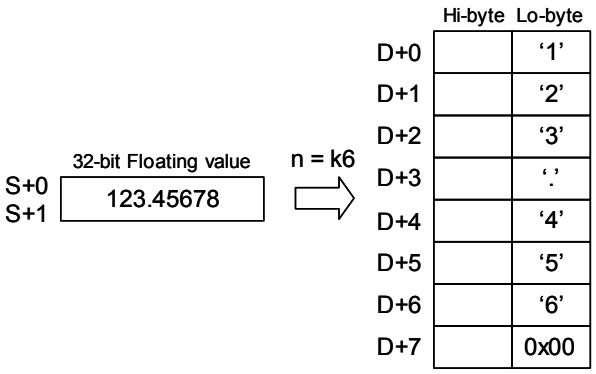
Explanations:

The system converts the floating point value in the source device S to data string with specified length **n** (decimal point is not included).

Points to note:

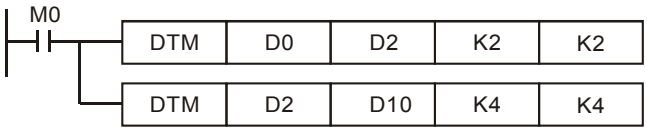
1. Operand **n** sets the number of total digits for the floating point value to be converted. Max 8 digits are applicable and the value over **n** digit will be omitted. For example, **n** = K6, floating value F123.45678 will be converted to data string “123.456”.
2. When the digits of source value are more than the specified **n** digits, only the **n** digits from the left will be converted. For example, source value F123456.78 with **n**=K4 will be converted as data string "1234”.
3. If the source value is a decimal value without integers, e.g. 0.1234, the converted data string will be “.1234” where the first digit is the decimal point.

The conversion will be performed in the following rule:



Program Example1: m = K2, m = K4

1. When M0 = ON, transform 16-bit data in D0, D1 into ASCII data in the following order: H byte – L byte – H byte – Low byte, and store the results in D10 ~ D17.



2. Value of source data D0, D1:

Register	D0	D1
Value	H1234	H5678

3. When the 1st DTM instruction executes, the ELC transforms the 16-bit data (Hi-byte, Lo-byte) into 8-bit data and moves it to registers D2~D5.

Register	D2	D3	D4	D5
Value	H12	H34	H56	H78

4. When the 2nd DTM instruction executes, the ELC transforms the 8-bit HEX data into ASCII data and moves it to registers D10~D17.

Register	D10	D11	D12	D13	D14	D15	D16	D17
Value	H0031	H0032	H0033	H0034	H0035	H0036	H0037	H0038

Program Example 2: m = K9

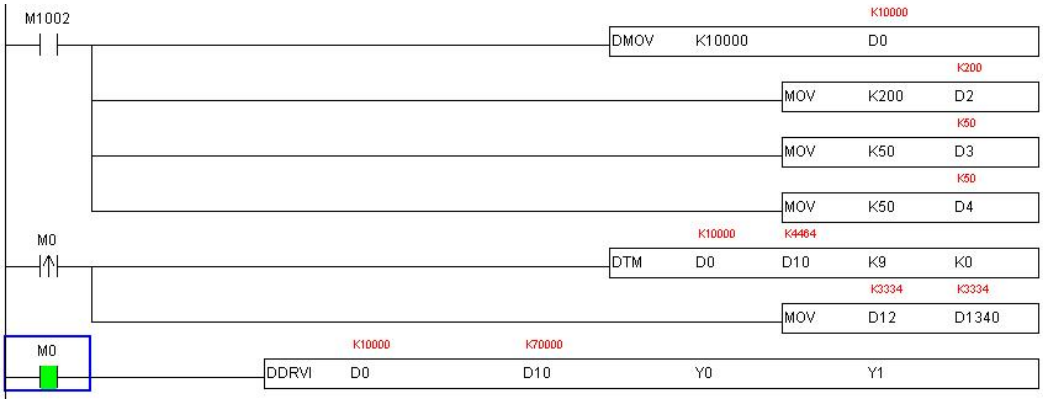
569. Set up the total number of pulses, total time, ramp-up time and ramp-down time in the source starting with D0. Execute the DTM instruction and the optimal max frequency as well as optimal start frequency can be obtained and executed by the positioning instructions.

570. Assume the source data is set up per the following:

Total Pulses	Total Time	Ramp-up Time	Ramp-down Time
D0, D1	D2	D3	D4
K10000	K200	K50	K50

571. The optimal positioning results can be obtained below:

Max frequency	Start frequency
D10, D11	D12
K70000	K3334



API	Mnemonic	Operands	Function
69	D SORT	**S, m_1, m_2, D, n**	Data sort

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			SORT: 11 steps
m_1					*	*										DSORT: 21 steps
m_2					*	*										
D													*			
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

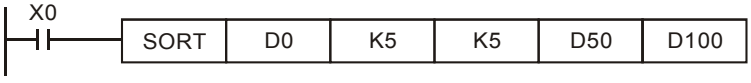
S: Starting address of source data **m_1**: The number of sort data groups (**m_1** =1~32) **m_2**: The number of columns for the data (**m_2** =1~6) **D**: Starting address for storing the sort data **n**: Present column of the sort data matrix (**n**=1~ **m2**)

Description:

572. The sorted data is stored in the **m_1** × **m_2** table of registers beginning with **D**. If devices **S** and **D** specify the same register, the resulting sorted data will be the same as the contents of source device **S**.
573. Once the SORT instruction has completed, the Flag M1029 (Execution complete flag) = ON.
574. There is no limit to the number of times this instruction may be used in the program. However, only one instruction can be executed at a time
575. In ELC-PV, DSORT only supports version 1.4 and above.
576. The function of sorting one-dimensional data is added. If **m_1** is 1, and **m_2** is 1, the function will be enabled, and the operand **n** represents the number of data (**n**=1~32). The data in **n** devices starting from the operand **S** are sorted. The sort result is stored in the devices starting from the operand **D**. It takes one scan cycle for the data to be sorted. After the data is sorted, M1029 will be On. This function supports ELC2-PB/PH/PA/PE/PV, ELCM-PH/PAV2.0.

Program Example:

577. When X0 = ON, it starts to sort the specified data. After the data sort is complete, M1029= ON. During the execution of the SORT instruction, data being sorted should not be changed. If the sort data needs to be changed the SORT instruction should be turned OFF, modify the data, then turn SORT instruction back on.



578. The tables below show how the matrix is established and how the data is sorted in numerical order in the selected column, based on the column number specified in D100. The first sort

table below using D0-D24 must be entered by the user. When a number from 1-5 for this example is written to D100, that column will be sorted numerically in the matrix formed by D50-D74. All data will be moved properly to the output matrix (D50-D74), but only the data in the selected column will be sorted into numerical order. All other data in the destination matrix will be placed in the proper position based on the data order in the selected column.

Example table of data sort

Data numbers: m_2

Data numbers: m_1	Data Column				
Column / Row	1	2	3	4	5
	Students No.	English	Math.	Physics	Chemistry
1	（D0）1	（D5）90	（D10）75	（D15）66	（D20）79
2	（D1）2	（D6）55	（D11）65	（D16）54	（D21）63
3	（D2）3	（D7）80	（D12）98	（D17）89	（D22）90
4	（D3）4	（D8）70	（D13）60	（D18）99	（D23）50
5	（D4）5	（D9）95	（D14）79	（D19）75	（D24）69

579. Sort data table when D100=K3

Data numbers: m_2

Data numbers: m_1	Data Column				
Column / Row	1	2	3	4	5
	Students No.	English	Math.	Physics	Chemistry
1	（D50）4	（D55）70	（D60）60	（D65）99	（D70）50
2	（D51）2	（D56）55	（D61）65	（D66）54	（D71）63
3	（D52）1	（D57）90	（D62）75	（D67）66	（D72）79
4	（D53）5	（D58）95	（D63）79	（D68）75	（D73）69
5	（D54）3	（D59）80	（D64）98	（D69）89	（D74）90

580. Sort data table when D100=K5

Data numbers: m_2

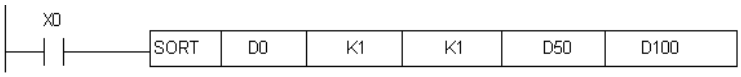
	Data Column

Data numbers: m_1

Column / Row	1	2	3	4	5
	Students No.	English	Math.	Physics	Chemistry
1	（D50）4	（D55）70	（D60）60	（D65）99	（D70）50
2	（D51）2	（D56）55	（D61）65	（D66）54	（D71）63
3	（D52）5	（D57）95	（D62）79	（D67）75	（D72）69
4	（D53）1	（D58）90	（D63）75	（D68）66	（D73）79
5	（D54）3	（D59）80	（D64）98	（D69）89	（D74）90

Program Example 1: (Sorting one-dimensional data)

If X0 is On, the data specified will be sorted. After the data is sorted, M1029 will be On.



If m_1 is K1, and m_2 is K1, one-dimensional data will be sorted. The value in D100 is K5. The values in D0~D4 are shown below.

1. The values in D0~D4 are listed below.

Data source (**S**)	D0	D1	D2	D3	D4
Data	75	65	98	60	79

2. The sort result is stored in D50~D54.

Sort result (**D**)	D50	D51	D52	D53	D54
Data	60	65	75	79	98

API	Mnemonic		Operands	Function
70	D	TKY	**S ,D_1, D_2**	Ten Key Input

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												TKY: 7 steps
D_1								*	*	*	*	*	*	*	*	DTKY: 13 steps
D_2		*	*	*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

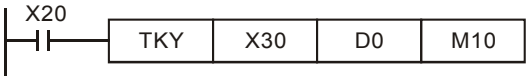
S: Starting input address (occupies 10 consecutive bits) **D_1**: Destination for storing key input value **D_2**: Key input signal (occupies 11 consecutives bits)

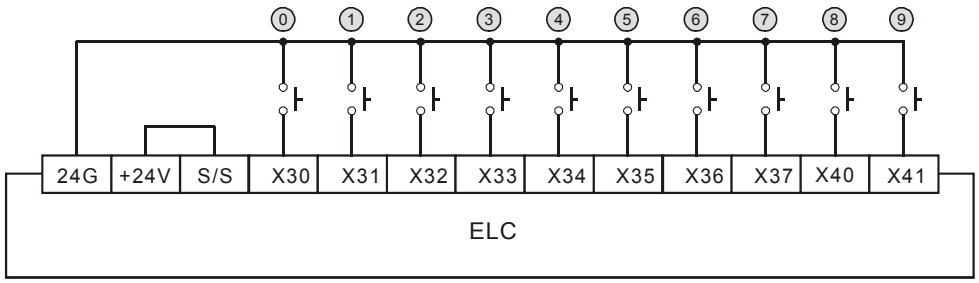
Description:

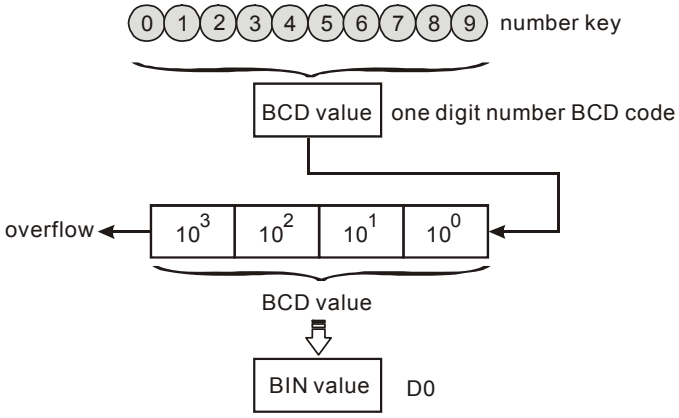
581. This instruction specifies ten external inputs with **S** and these ten external inputs are identified as decimal values of 0 to 9. These ten external input devices are connected to ten keys. When each key is pressed, the decimal value for that key from 0 to 9,999 (max. 4 digits in 16-bit instruction) or from 0 to 99,999,999 (max. 8 digits in 32-bit instruction) is stored in destination **D_1**. **D_2** is used to store the state of the current key number pressed.
582. For ELC-PA, **S** and **D_2** do not support E, F index registers modification.
583. There is no limit on the number of times this instruction is used in the program, however only one instruction is allowed to be executed at a time.

Program Example:

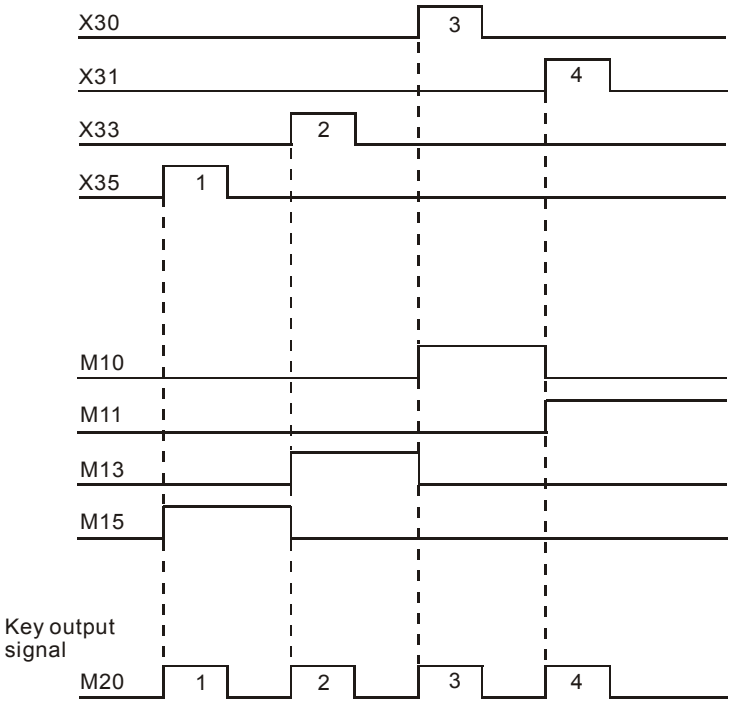
584. Specify ten input terminals beginning with X30 to connect the ten keys to. These keys are numbered from 0 to 9. When X20=ON, the instruction is executed and it will store the BIN value of the key pressed into D0 and set the associated bit (M10~M19). M0-M9 in this example are used to store the condition of the key that has been pressed last.







585. The chart below has four keys connected to X35, X33, X30 and X31 of a number keyboard. After pressing the four keys in the following order, ①②③④, the number 5,301 will be entered into D0, one digit at a time. The maximum number which can be entered in D0 is 9,999 i.e. 4 digits. If the entered number exceeds the allowable range, the highest digits will overflow.
586. After X35 is pressed, M15=ON until another key is pressed. The process is the same as other keys are pressed.
587. As each key is pressed, the associated M-bit will be turned ON.
588. M20 = ON when any of the keys is pressed.
589. When X20 is OFF, the value in D0 remains unchanged but all the M-bits will be turned OFF.



API	Mnemonic		Operands	Function
71	D	HKY	S ,D_1, D_2, D_3	Hexadecimal Key Input

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*															HKY: 9 steps
D_1		*														DHKY: 17 steps
D_2											*	*	*	*	*	
D_3		*	*	*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

3

Operands:

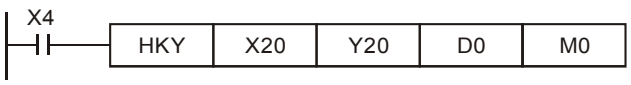
S: Starting scan input address (occupies 4 consecutive bits) **D_1**: Starting scan output address (occupies 4 consecutive bits) **D_2**: Destination for storing the key input value **D_3**: Key input signal (occupies 8 consecutive bits)

Description:

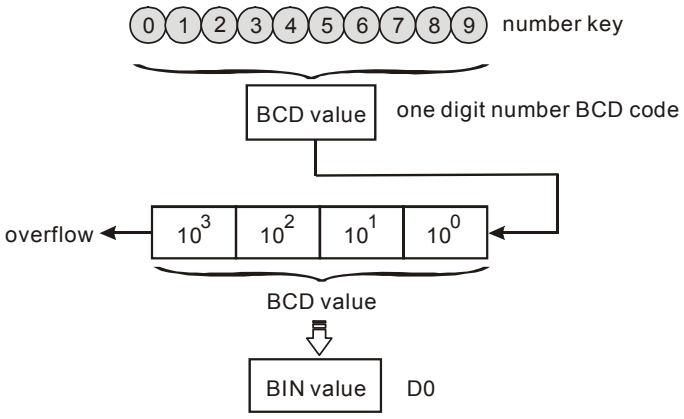
590. This instruction creates a 16-key keyboard by multiplexing 4 consecutive external inputs, with the starting input address entered into parameter **S** and 4 consecutive external outputs with the starting output address entered into parameter **D_1**. The key input value will be stored in **D_2**. **D_3** stores the condition of keys A~F and indicates the key input status of 0~9 and A~F.
591. Every time this instruction is executed, the execution complete flag M1029 = ON for the duration the key is pressed (one scan cycle).
592. If two or more keys are pressed at the same time, only the key activated first will be used.
593. When the HKY instruction is used in a 16-bit instruction, **D_2** can store numbers from 0 to 9,999 (max. 4 digits). When DHKY instruction is used in a 32-bit instruction (DHKY), **D_2** can store numbers from 0 to 99,999,999 (max. 8 digits). If the entered number exceeds the allowable ranges, the highest digits will overflow.
594. For ELC-PA, **S, D_1** and **D_2** do not support the E, F index registers modification.
595. There is no limit on the number of times this instruction can be used in the program, but only one instruction is allowed to be executed at a time.

Program Example:

Use this instruction to create a 16-key keyboard which multiplexes 4 continuous external input devices X20~X23 and 4 continuous external output devices Y20~Y23. When X4=ON, the instruction is executed and it will store the BIN value of the 4 inputs into D0. M0~M7 in this example are used to store the condition of the key that has been pressed last.



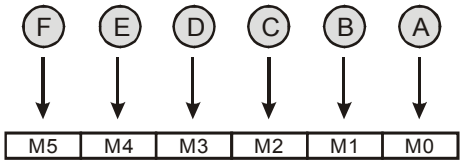
Number input:



Function key input:

a) When the A key is pressed, M0=ON and latched. Next, press the D key and, M0=OFF, M3=ON and latched.

b) If two or more keys are pressed at the same time, only the key activated first is used.



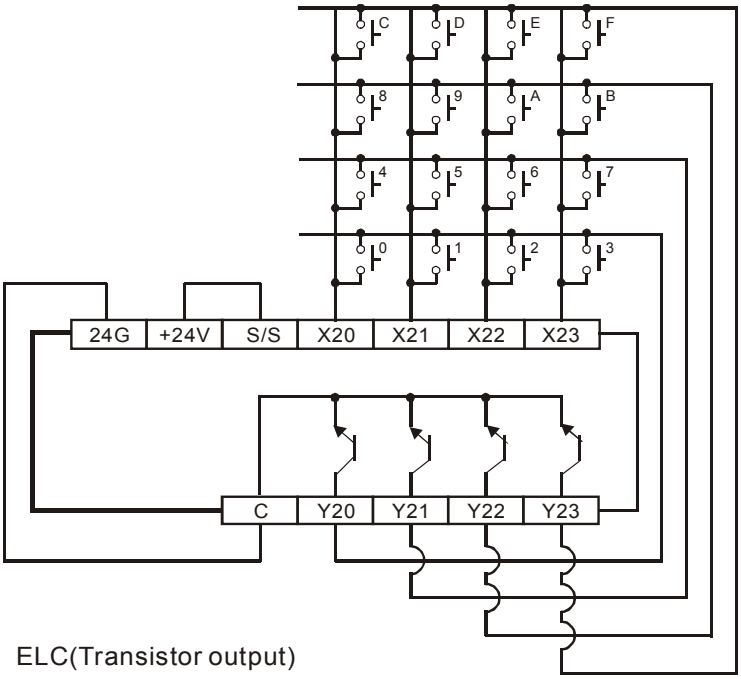
Key output signal:

a) When any key of A – F is pressed, M6=ON for one scan time.

b) When any key of 0 to 9 is pressed, M7=ON for one scan time.

When X4 = OFF, D0 remains unchanged but M0~M7 = OFF.

External wiring:



ELC(Transistor output)

Points to note:

596. When this instruction is executed, 8 scan time cycles are required to read the input values of the keys. If the scan cycle is too long or too short, it may cause the key values to be read incorrectly. Therefore, a fixed scan time is suggested. Use Bit M1039 and register D1039 to fix the scan time. If the scan time is too long, use this instruction in a time interrupt subroutine.
597. The function of flag M1167:
    a) When M1167=ON, the HKY instruction can input hexadecimal values from 0~F.
    b) When M1167=OFF, A~F of the HKY instruction are used as function keys.
598. D1037 functionality (only supported by the ELC-PV, ELC2-PV):
    - Write the overlapping time for the keys into D1037 (unit: ms). The overlapping time will vary upon different program scan times and the settings in D1037.

API	Mnemonic	Operands	Function
72	DSW	**S, D_1, D_2, n**	Digital Switch

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*															DSW: 9 steps
D_1		*														
D_2											*	*	*			
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

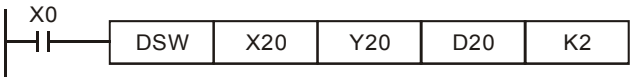
S: Starting address of switch input **D_1**: Starting address of switch output **D_2**: Destination address for storing the set-point value **n**: Number of digits (**n**=1~2)

Description:

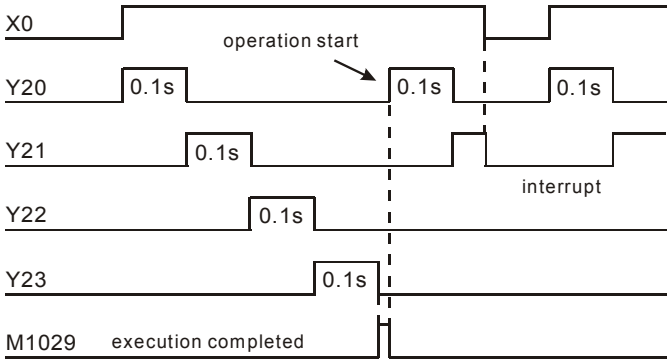
599. This instruction creates 1 or 2 groups of 4-digit DIP switches by combining 4 or 8 consecutive input points starting from **S** and 4 consecutive output points starting from **D_1**. The value of the inputs will be read in **D_2** and the value in **n** specifies the number of groups (1 or 2) of the DIP switches.
600. **If n** = K1, **D_2** occupies 1 register. If **n** = K2, **D_2** occupies 2 consecutive registers..
601. There is no limit on the number of times this instruction may be used in the program. However only one instruction is allowed to be executed at the same time.

Program Example:

602. The first group of switches consists of X20~X23 and Y20~Y23. The second group of switches consists of X24~X27 and Y20~Y23. When X0=ON, the first group of switches are read and converted to binary and stored in D20. Then the second group of switches are read and converted to binary and stored in D21.

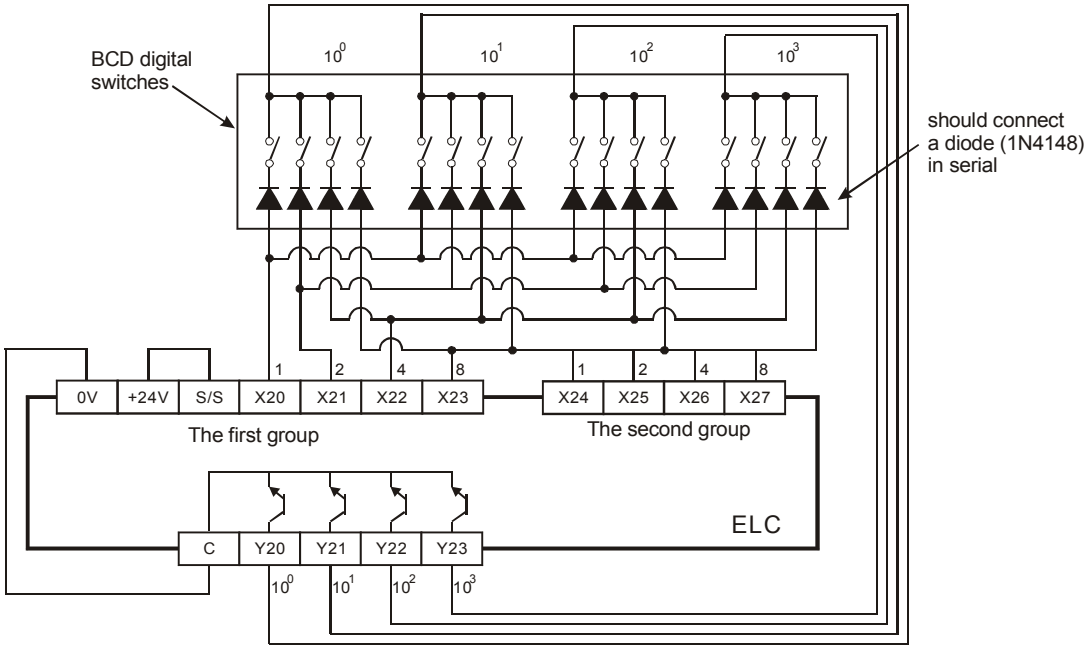


603. When X0=ON, Y20~Y23 will each turn ON one at a time in order and continue to cycle over and over. After the completion of each cycle, the execution completed flag M1029 will turn ON for one scan after the completion of a cycle.



604. Transistor outputs must be used for Y20~Y23. Also, be sure that a diode is connected to every input terminal (0.1A/50V diode) as shown below.

Wiring diagram of digital switch:



API	Mnemonic		Operands	Function
73	SEGD	P	S, D	7-segment Decoder

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	SEGD, SEGDP: 5 steps
D								*	*	*	*	*	*	*	*	

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Source address for decoding **D**: Output address after decoding

Description:

A single hexadecimal digit (0 to 9, A to F) occupying the lower 4 bits of source device **S** is decoded into a data format used to drive a seven segment display. A representation of the hex digit is then displayed. The decoded data is stored in the lower 8 bits of destination address **D**. The upper 8 bits of the address are not written to.

Program Example:

When X20=ON, the contents of the lower 4 bits (b0~b3) of D10 will be decoded per the table below, to be displayed on a 7-segment display. The decoded results will be moved to Y20~Y27.

Decoding Chart of the 7-segment Display Panel

16 bits	Bit Combination	Composition of the 7-SEG display	Status of each segment						
			B0(a)	B1(b)	B2(c)	B3(d)	B4(e)	B5(f)	B6(g)
0	0000		On	On	On	On	On	On	Off
1	0001		Off	On	On	Off	Off	Off	Off
2	0010		On	On	Off	On	On	Off	On
3	0011		On	On	On	On	Off	Off	On
4	0100		Off	On	On	Off	Off	On	On
5	0101		On	Off	On	On	Off	On	On
6	0110		On	Off	On	On	On	On	On
7	0111		On	On	On	Off	Off	Off	Off
8	1000		On	On	On	On	On	On	On
9	1001		On	On	On	On	Off	On	On
A	1010		On	On	On	Off	On	On	On
B	1011		Off	Off	On	On	On	On	On
C	1100		On	Off	Off	On	On	On	Off
D	1101		Off	On	On	On	On	Off	On
E	1110		On	Off	Off	On	On	On	On
F	1111		On	Off	Off	Off	On	On	On

API	Mnemonic	Operands	Function
74	SEGL	**S, D, n**	7-segment with Latch

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	SEGL: 7 steps
D		*														
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

3

Operands:

S: Source address for the 7-segment display **D**: Starting address of 7-segment display outputs

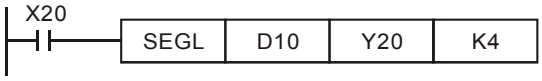
n: Polarity set-point of output signal and scan signal (**n**=0~7)

Description:

605. 8 or 12 consecutive external output points with the starting address defined by **D** can be regarded as display and scan signal outputs of 1 or 2 groups of 4 digits for a 7-segment display. A 7-segment display module has the ability to convert input BCD code to a 7-segment display and uses a control signal to latch it.
606. **n** will determine the number of groups of 4 digits for a 7-segment display..
607. Each time this instruction executes, it cycles through writing each group of outputs to the outputs for the 7-segment display, one at a time.
608. For ELCB-PB series controllers, the instruction can only be used once in the program.
609. For ELC-PV, ELC2-PV series controllers, the instruction can be used twice in the program.
610. For ELC-PA, ELCM-PH/PA, ELC2-PB/PH/PA/PE series controllers, there is no limitation on the times of using the instruction, but only one instruction can be executed at a time.
611. Flag: When SEGL is completed, M1029 = ON for one scan cycle.

Program Example:

612. When X20=ON, instruction will start to execute. 7-segment display is connected to outputs Y20~Y27. The value of D10 will be converted to BCD code and sent to the first group for the 7-segment display. The value of D11 will be converted to BCD code and sent to the second group for the 7-segment display. If any value of D10 or D11 is greater than 9,999, an operation error will occur.



613. When X20=ON, Y24~Y27 will begin cycling ON, one at a time. Each cycle needs 12 scan cycles. M1029=ON when each cycle completes.
614. 4 digits per group, **n**=0~3

a) After terminals 1, 2, 4, 8 for the first 7-segment display are decoded, the result will be written to Y20~Y23 for this display. As terminals Y24-Y27 cycle ON and OFF, these values will be driven and latched on the display..

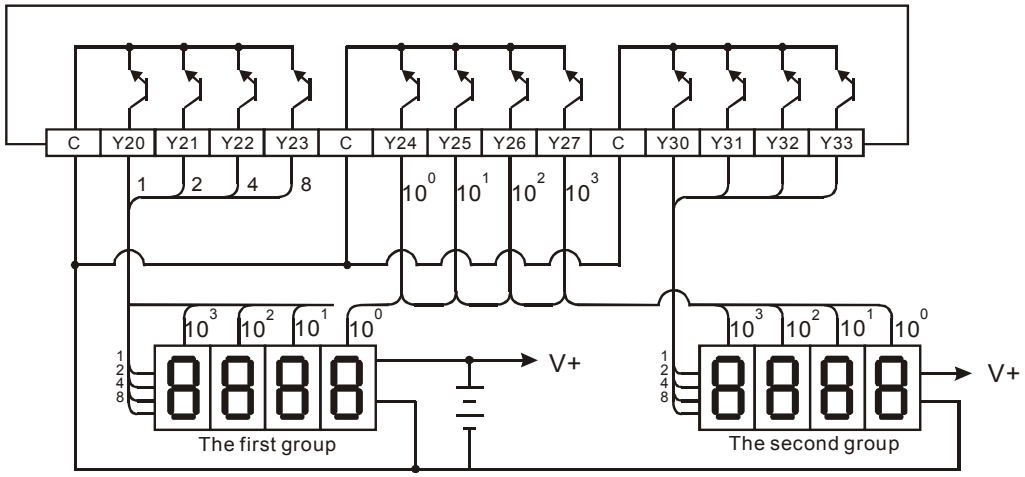
b) When X20=ON, the contents of D10 will be transmitted to the 7-segment display to display in sequential order the data to outputs Y24~Y27 cycling ON/OFF in sequence.

615. 4 digits of two groups, **n**=4~7

a) After the terminal of 1, 2, 4, 8 for the first 7-segment display are decoded and displayed, , the terminals for the second 7-segment displayY30~Y33 will be decoded. Terminals Y24~Y27 are used to latch the display values for both sets of displays by continually cycling On and OFF.

b) The contents of D10 will be decoded and written to the first group of outputs for the first 7-segment display and the contents of D11 will be decoded and written to the second group of outputs for the second 7-segment display to display. If D10=K1234 and D11=K4321, the first group will display 1 2 3 4 and the second group will display 4 3 2 1.

7-segment display output wiring



Points to note:

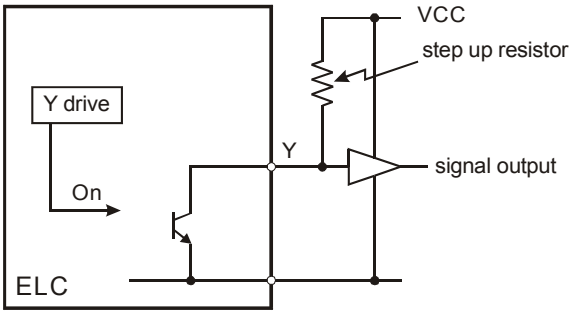
616. ELC-PB, ELCB-PB series only provides a group of 4 digits for a 7-segment display and use 8 output points. The SEGL instruction only can be used once in the program and the usage range of operand **n** is 0 to 3.

617. The scan time must be longer than 10ms for this instruction to execute properly. If the scan time is shorter than 10ms, use the fixed scan time function to fix the scan time to 10ms.

618. **n** is used to set the polarity of transistor output loop. It can be set to positive polarity or negative polarity. The type of 7-segment display it connects to (a group of 4 digits or two groups of 4 digits) determines the value of **n**.

619. ELC transistor output is NPN type and it is open collect output. When wiring, connect the output to a pull-up resistor to VCC (less than 30VDC). When the output point Y is ON, the

output will be low voltage.



620. Positive logic (Negative polarity) output of BCD code

BCD value				Y output (BCDcode)				Signal output			
b_3	b_2	b_1	b_0	8	4	2	1	A	B	C	D
0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	1	0	0	0	1	1	1	1	0
0	0	1	0	0	0	1	0	1	1	0	1
0	0	1	1	0	0	1	1	1	1	0	0
0	1	0	0	0	1	0	0	1	0	1	1
0	1	0	1	0	1	0	1	1	0	1	0
0	1	1	0	0	1	1	0	1	0	0	1
0	1	1	1	0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	0	0	1	1	1
1	0	0	1	1	0	0	1	0	1	1	0

621. Negative logic (Positive polarity) output of BCD code

BCD value				Y output (BCDcode)				Signal output			
b_3	b_2	b_1	b_0	8	4	2	1	A	B	C	D
0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	0	0	1	0
0	0	1	1	1	1	0	0	0	0	1	1
0	1	0	0	1	0	1	1	0	1	0	0
0	1	0	1	1	0	1	0	0	1	0	1
0	1	1	0	1	0	0	1	0	1	1	0
0	1	1	1	1	0	0	0	0	1	1	1
1	0	0	0	0	1	1	1	1	0	0	0

1	0	0	1	0	1	1	0	1	0	0	1

622. Display scan (latch) signal

Positive logic (Negative polarity) output		Negative logic (Positive polarity) output	
Y output (Latch)	Output control signal	Y output (Latch)	Output control signal
1	0	0	1

623. Parameter **n** set-points:

Groups number of 7-segment display	A group				Two groups			
Y of BCD code outputs	+		−		+		−	
Display scan latch signal	+	−	+	−	+	−	+	−
n	0	1	2	3	4	5	6	7

'+': Positive logic (Negative polarity) output

'−': Negative logic (Positive polarity) output

624. The output polarity of the ELC transistor and input polarity of 7-segment display is determined by **n**.

API	Mnemonic	Operands	Function
75	ARWS	**S, D_1, D_2, n**	Arrow Key Input

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												ARWS: 9 steps
D_1											*	*	*	*	*	
D_2		*														
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Starting address of the key inputs (occupies 4 continuous addresses) **D_1**: Display address for the 7-segment display **D_2**: Starting output address for the 7-segment display **n**: Polarity set-point of output signal and scan signal (**n**=0~3)

Description:

625. This instruction displays the contents of a single integer **D_1** on a set of 4 digit, seven segment displays. The data within **D_1** is actually in a standard decimal format but is automatically converted to BCD to display on the seven segment units. Each digit of the displayed number can be selected and edited. The editing procedure directly changes the value of **D_1**.

626. **S** and **D_2** of ELC-PA, ELC2-PB/PA/PE, ELCM-PH/PA do not support the E, F index registers modification.

627. **D_2** of ELCM-PH/PA, ELC2-PB/PH/PA/PE can only be specified as a multiple of 10 as well, e.g. Y0, Y10, Y20…etc.

628. Only transistor outputs should be used for this instruction.

629. When using this instruction, the scan time should be fixed or the instruction should be placed into a timed interrupt.

630. There is no limit on the number of times this instruction can be used in the program, but only one instruction is allowed to be executed at a time.

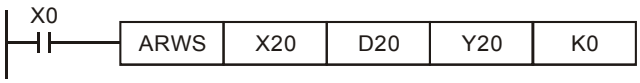
Program Example:

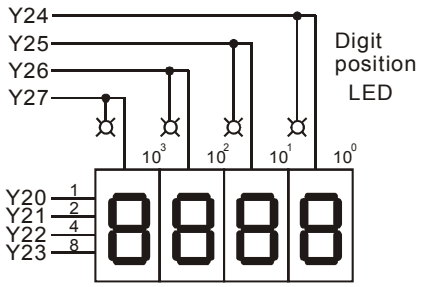
631. When the instruction is executed, X20 is defined as the down key, X21 is defined as the up key, X22 is defined as the right key and X23 is defined as the left key. These keys are used to edit and display the external set-point value. The set-point value is stored in D20 and its set-point range is from 0 to 9,999.

632. When X0=ON, 10^3 is the effective set-point digit number. Pressing the left key, the effective digit number will be displayed and it will jump from $10^3 \rightarrow 10^0 \rightarrow 10^1 \rightarrow 10^2 \rightarrow 10^3 \rightarrow 10^0$.

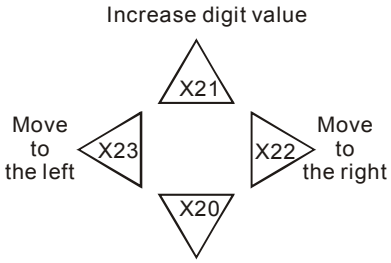
633. Pressing the right key, the effective digit number will be displayed and it will jump from $10^3 \rightarrow 10^2 \rightarrow 10^1 \rightarrow 10^0 \rightarrow 10^3 \rightarrow 10^2$. At the same time, the digit position LED connected to Y24 to Y27 will indicate the effective set-point digit number.

634. Pressing the up key to increase the value , the number will change from 0→1→2→…8→9→0→1. Pressing the down key, the number will change from 0→9→8→…1→0→9. The changed value will be displayed on the 7-segment display.





7-step display which displays setting value (4 digits data)



The 4 switches is used to move digit position to the left or to the right and increase or decrease the setting value of the digits

API	Mnemonic	Operands	Function
76	ASC	S, D	ASCII Code Conversion

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S																ASC: 11 steps
D											*	*	*			

ELCB			ELC						ELC2										ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA			
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	

Operands:

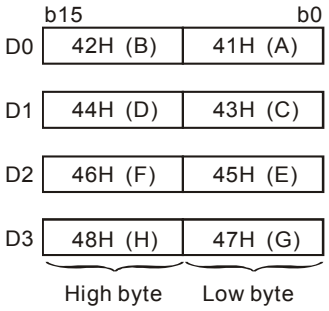
S: The alphanumeric character to be converted to ASCII code **D**: The destination address for storing the ASCII code

Explanation:

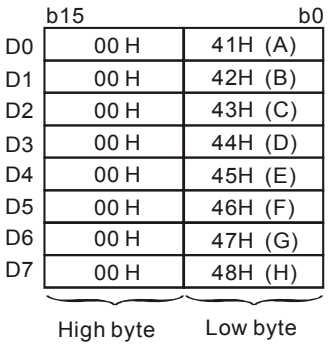
635. The ASC instruction converts 8 characters stored in **S** to ASCII code and stores the result in **D**. The values in **S** can be entered with ELCSoft programming software.
636. **S** in ELC-PA only accepts A, B, C, D, E, F, G, H, the 8 English character.
637. The alphanumeric characters can be used to display error messages.
638. Flag: M1161 selects 8 or 16 bit mode.

Program Example:

When X0=ON, A~H is converted to ASCII code and stored in D0~D3.



When M1161=ON, the ASCII code converted from each character will be placed in the lower 8-bits (b7~b0) of one register. The high byte will be invalid and its contents filled with 0. This also means that one register is used to store one character.



API	Mnemonic	Operands	Function
77	PR	**S, D**	Outputs ASCII Code

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*	*	*			PR: 5 steps
D		*														

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

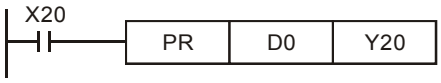
S: The address for storing ASCII data (occupies 4 continuous addresses) **D**: The external output address for the ASCII code (occupies 10 continuous bit addresses)

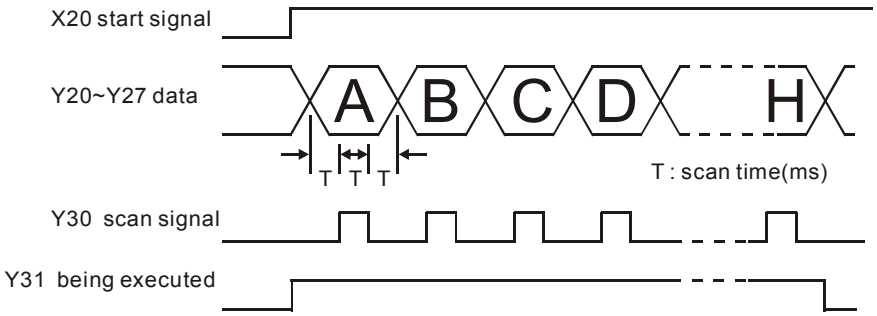
Description:

639. This instruction will display ASCII codes stored in 4 continuous registers (**S)** on the outputs specified by **D**.
640. **D ~ D+7** map the source data (ASCII code) directly in order, **D+10** is the scan signal and **D+11** is the execution flag.
641. The PR instruction can only be used twice in the program.
642. Flag: M1029 execution complete flag; M1027 outputs number flag.

Program Example 1:

643. First, use the ASC instruction to convert A~H to ASCII code and store the data in D0~D3. Then, using the PR instruction move the data to the 8 outputs.
644. When M1027=OFF and X20=ON, this instruction will execute to write the ASCII data to Y20 (lower bit) through Y27 (upper bit). Y30 is the scan signal, and Y31 is the monitor signal. This mode can sequentially display 8 ASCII characters on the 8 outputs.
645. If X20 turns from ON →OFF while the instruction being executed, all the outputs will be turned OFF. When X20 turns ON again, the instruction will begin executing again.





Program Example 2:

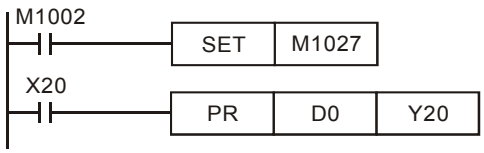
646. The PR instruction provides an 8 character serial string output operation. When M1027=OFF, a maximum of 8 character strings can be serially displayed on the outputs. When M1027=ON, a 1 to 16 character string will be serially displayed on the outputs.

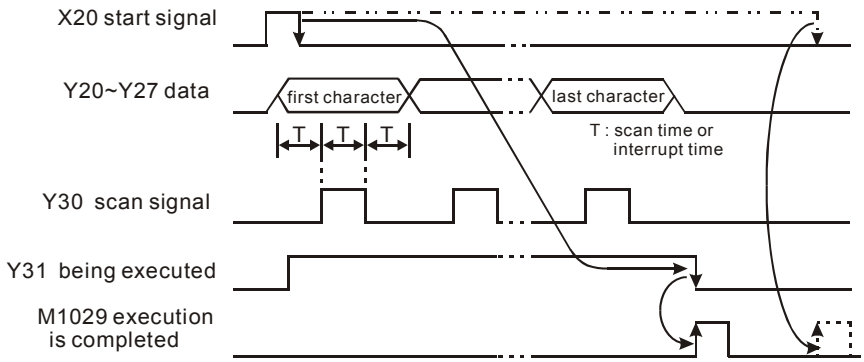
647. When M1027=ON and X20 transitions from OFF to ON, the PR instruction will execute to write the ASCII data to Y20 (lower bit) through Y27 (upper bit), Y30 is the scan signal, and Y31 is the monitor signal. This mode can sequentially display 16 ASCII characters on the 8 outputs.

648. . If X20 turns OFF during execution, the outputs will turn OFF.

649. If the instruction encounters a 00H (NULL) in the character string, it means the end of the character string and the operation of PR instruction will stop.

650. If X20 is always ON, the instruction will stop automatically after one operation is complete. If X20 remains ON at the end of execution, the execution complete bit (M1029) will not be active.





Points to note:

651. Only transistor outputs should be used with this instruction.

652. When using this instruction, the scan time should be fixed or execute this instruction in a timed interrupt subroutine.

API	Mnemonic			Operands	Function
78	D	FROM	P	**m_1, m_2, D, n**	Read CR from Module

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
m_1					*	*							*			FROM, FROMP: 9 steps
m_2					*	*							*			DFROM, DFROMP: 17 steps
D								*	*	*	*	*	*	*	*	
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

m_1: Specialty module number **m_2**: CR (Control Register) number in the specialty module

D: Location to save the read data **n**: number of data words to read

Description:

653. The ELC uses this instruction to read CR data from specialty modules.

654. The operands m1, m2, and n should be in the ranges listed in the table below:

ELCB-PB, ELC-PA:

Operand	**m_1**	**m_2**	16-bit **n**	32-bit **n**
Right-side Range	0~7	0~48	1 ~ (49 – **m_2**)	1 ~ (49 – **m_2**)/2
Left-side Range	Left-side modules are not supported.			

ELC-PV, ELC2-PV:

Operand	**m_1**	**m_2**	16-bit **n**	32-bit **n**
Right-side Range	0~7	0 ~ 499	1 ~ (500 – **m_2**)	1 ~ (500 – **m_2**)/2
Left-side Range	100~107	0 ~ 499	1 ~ (500 – **m_2**)	1 ~ (500 – **m_2**)/2

ELCM-PH/PA:

Operand	**m_1**	**m_2**	16-bit **n**	32-bit **n**
Right-side Range	0~7	0~255	1~4 V2.0(above)1~6	1~2 V2.0(above)1~3
Left-side Range	Left-side modules are not supported.			

ELC2-PC/PA:

Operand	**m_1**	**m_2**	16-bit **n**	32-bit **n**

Right-side Range	0~7	0~48	1~6	1~3
Left-side Range	100~107	0~255	1~(256-m2)	1~(256-m2)/2

ELC2-PE:

Operand	m_1	m_2	16-bit **n**	32-bit **n**
Right-side Range	0~7	0~48	1~6	1~3
Left-side Range	100~108	0~255	1~(256-m2)	1~(256-m2)/2

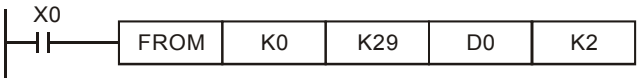
ELC2-PB:

Operand	m_1	m_2	16-bit **n**	32-bit **n**
Right-side Range	0~7	0~48	1~6	1~3
Left-side Range	Left-side modules are not supported.			

655. ELCM-PH/PA and ELC2-PB/PH/PA/PE: The operand **D** only supports the D devices.
656. When **D** indicates a bit device operand, use K1~K4 for the16-bit instruction and K5~K8 for the 32-bit instruction.
657. Flag: When M1083=ON, it enables interrupts during the execution of the FROM/TO instructions. M1083 only supports ELC-PA/PV, and ELC2-PV.
658. ELCB-PB doesn't support the index registers E, F modification.

Program Example:

659. Read the contents of CR#29 and CR#30 from specialty module#0 and place the data in D0 and D1.in the ELC controller. 2 words are read because **n**=2.
660. The instruction will be executed when X0=ON.



API	Mnemonic			Operands	Function
79	D	TO	P	m_1, m_2, S, n	Write CR to Module

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
m_1					*	*							*			TO, TOP: 9 steps
m_2					*	*							*			DTO, DTOP: 17 steps
S					*	*	*	*	*	*	*	*	*	*	*	
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

m_1: Specialty module number **m_2**: CR (Control Register) number of the special module

S: Data address of the data to write to the specialty module **n**: Number of words to write

Description:

661. The operands m1, m2, and n should be in the ranges listed in the table below:

ELCB-PB, ELC-PA:

Operand	m_1	m_2	16-bit **n**	32-bit **n**
Right-side Range	0~7	0~48	1 ~ (49 – m_2)	1 ~ (49 – m_2)/2
Left-side Range	Left-side modules are not supported.			

ELC-PV, ELC2-PV:

Operand	m_1	m_2	16-bit **n**	32-bit **n**
Right-side Range	0~7	0 ~ 499	1 ~ (500 – m_2)	1 ~ (500 – m_2)/2
Left-side Range	100~107	0 ~ 499	1 ~ (500 – m_2)	1 ~ (500 – m_2)/2

ELCM-PH/PA:

Operand	m_1	m_2	16-bit **n**	32-bit **n**
Right-side Range	0~7	0~255	1~4 V2.0(above)1~6	1~2 V2.0(above)1~3
Left-side Range	Left-side modules are not supported.			

ELC2-PC/PA:

Operand	m_1	m_2	16-bit **n**	32-bit **n**
Right-side Range	0~7	0~48	1~6	1~3

Left-side Range	100~107	0~255	1~(256-m2)	1~(256-m2)/2

ELC2-PE:

Operand	m_1	m_2	16-bit **n**	32-bit **n**
Right-side Range	0~7	0~48	1~6	1~3
Left-side Range	100~108	0~255	1~(256-m2)	1~(256-m2)/2

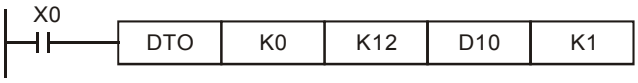
ELC2-PB:

Operand	m_1	m_2	16-bit **n**	32-bit **n**
Right-side Range	0~7	0~48	1~6	1~3
Left-side Range	Left-side modules are not supported.			

662. ELCM-PH/PA and ELC2-PB/PH/PA/PE: The operand **S** only supports the D devices.
663. When **S** is a bit operand, K1~K4 can be used for 16-bit instruction and K1~K8 can be used for 32-bit instruction.
664. Flag: When M1083=ON, it enables interrupts during the execution of the FROM/TO instructions. M1083 only supports ELC-PA/PV, and ELC2-PV
665. ELCB-PB doesn't support the index registers E, F modification.

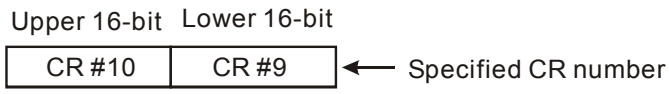
Program Example:

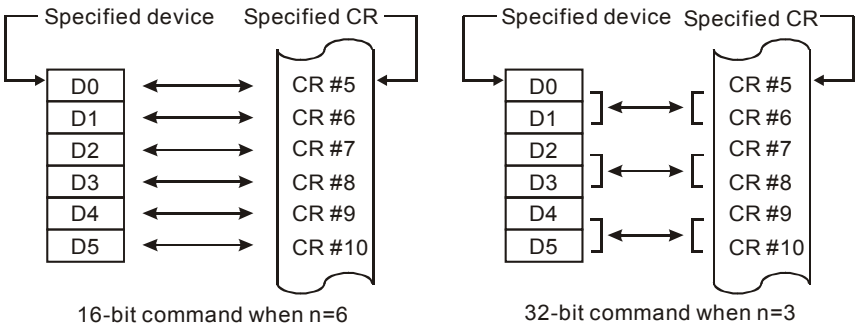
666. Using the 32-bit instruction DTO, the program will write D11 and D10 into CR#13 and CR#12 of the specialty module#0.
667. The instruction will be executed when X0=ON.



The rule of instruction operand:

668. m_1: The number of the specialty module. The right specialty module closest to the ELC controller will be assigned 0, the next closest will be assigned 1, etc. for a maximum of 8 modules (0~7). The left special module closest to the ELC controller will be assigned 100, the next closest will be assigned 101, etc. for a total of 8 modules (100~107).
669. m_2: CR number. 16-bit memory in specialty modules is called CR (Control Registers). Valid CR numbers are 0~499.





670. In ELCB-PB and ELC2-PB/PH/PA/PE, M1083 is not supported. When the FROM/TO instruction is executed, all interrupts (including external or internal interrupt subroutines) will be disabled. All interrupts will be executed after the FROM/TO instruction is completed. The, FROM/TO instruction can also be executed in an interrupt subroutine.

671. In ELC-PA/PV and ELC2-PV2, M1083 is supported. The functions of M1083 are described below.:

a) When M1083=OFF, the FROM/TO instruction is executed, all interrupts (including external or internal interrupt subroutines) will be disabled. All interrupts will be executed after the FROM/TO instruction is completed.

b) When M1083=ON, if an interrupt occurs while the FROM/TO instruction is executing, the FROM/TO instruction will be interrupted to execute the interrupt. In this case, the FROM/TO instruction cannot be executed in the interrupt subroutine

Application program example of FROM/TO instruction:

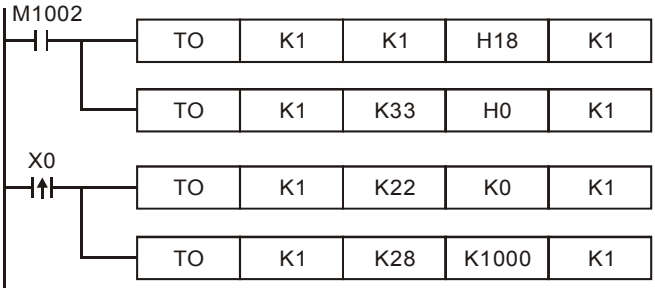
672. Example 1: Configure the scaling parameters for the analog input CH1 of the ELC-AN04ANNN with a set-point OFFSET value of 0V(=$K0_{LSB}$) and a GAIN value of 2.5V(=$K2000_{LSB}$).

1. Write H0 to CR#1 of analog input module No. 0, which sets CH1 to mode 0 (voltage input: -10V to +10V).
2. Write H0 to CR#33 and allow it to adjust the characteristics of CH1 to CH4.
3. When X0 turns from OFF →ON, K0 OFFSET value will be written to CR#18 and K2000 GAIN value will be written to CR#24.

b) Example 2: Adjust A/D conversion characteristic curve of ELC-AN04ANNN by the set-point OFFSET value of CH2 to 2mA(=K400) and GAIN value of CH2 to 18 mA(=K3600).

1. Write H18 to CR#1 of analog input module No. 0, which sets CH2 to mode 3 (current input: -20mA to +20mA).
2. Write H0 to CR#33 to adjust characteristics of CH1 to CH4.
3. When X0 turns from OFF →ON, K400, the OFFSET value will be written to CR#19 and K3600, the GAIN value will be written to CR#25.

c) Example 3: Adjust D/A conversion characteristic curve of ELC-AN02NANN by set-point OFFSET value for CH2 to 0mA(=K0) and GAIN value for CH2 to 10mA(=K1000).



1. Write H18 to CR#1 of analog input mode No. 1 which sets CH2 to mode 3 (current input: 0mA to +20mA).
2. Write H0 to CR#33 to adjust characteristics of CH1 and CH2.
3. When X0 turns from OFF →ON, write K0, the OFFSET value to CR#22 and K1000, the GAIN value will be written to CR#28.

API	Mnemonic	Operands	Function
80	RS	**S, m, D, n**	Serial Data Communication

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			RS: 9 steps
m					*	*							*			
D													*			
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Starting address of transmit data **m**: Transmit data length (**m**=0~255) **D**: Starting address of receive data **n**: receive data length (**n**=0~255)

Description:

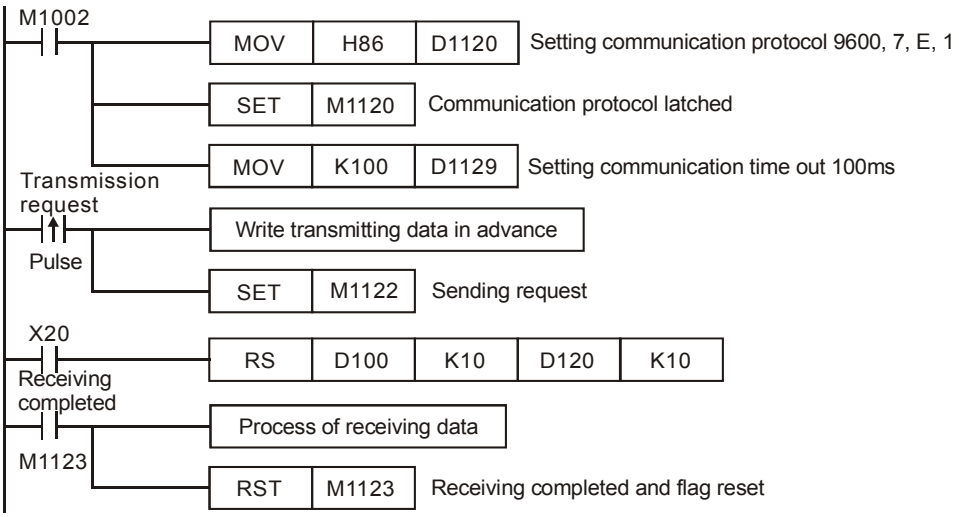
673. The RS instruction is used for transmitting data and/or receiving data between an ELC controller and an external device. Data needs to be entered into registers starting from **S** with length **m**. Then, specify the data receive register **D** and the receiving data length **n**.
674. Please refer to the following table for more information about the communication ports supported by RS:

COM	COM1(RS-232)	COM2(RS-485)	COM3(RS-485)
ELC-PA/PV	-	V	-
ELCB-PB	-	V	-
ELCM-PH/PA	V	V	V
ELC2-PB/PA/PV	V	V	-
ELC2-PC	V	V	V
ELC2-PE	-	V	V

675. If data does not need to be transmitted, **m** must be K0. If it doesn't need to receive data, **n** must be K0.
676. There is no limit on the number of times this instruction can be used, however, only 1 instruction can be executed on one communication port at the same time.
677. During execution of the RS instruction, the data being sent cannot be changed.
678. For Flags: M1120~M1131, M1140~M1143, M1161, see the examples below.
679. The ELCB-PB do not support the index registers E, F modification.

Program Example 1: COM2 RS-485

680. Enter data into the registers that start with D100 and set M1122=ON (send request-flag).
681. If the RS instruction is executed when X20=ON, the ELC will transmit the 10 data values that start from D100, then wait for the receive data. M1122 will be set to OFF at the end of transmitting (do not use the program to reset M1122). After a 1ms delay, it will start to receive the 10 external data values and store them into consecutive registers that start from D120.
682. When the receive operation is complete, M1123 will be set to ON. The ELC program should reset M1123 = OFF when the receive is complete. Do not use ELC program to continuously reset M1123. Reset it once each time the receive is complete.

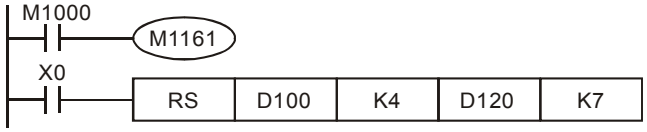


Program Example 2: COM2 RS-485

8-bit mode (M1161=ON) / 16-bit mode (M1161=OFF) switch:

8-bit mode:

683. Start and End Characters for the ELC transmission data will be set with M1126 and M1130 with the data in D1124~D1126. Once the start and end characters are set, the ELC will send start and end data automatically when executing the RS instruction.
684. When M1161=ON, the mode will be 8-bit. 16-bit data will be divided into high byte and low byte. High byte will be ignored and low byte will be received and transmitted.



Transmit data: (ELC→External equipment)

STX	D100L	D101L	D102L	D103L	ETX1	ETX2

Head code

(S) source data register will start from low byte of D100

(m) length = 4

Tail code 1

Tail code 2

Receive data: (External equipment→ELC)

D120L	D121L	D122L	D123L	D124L	D125L	D126L

Head code

(S) receive data register will start from low byte of D120

(n) length = 7

Tail code 1

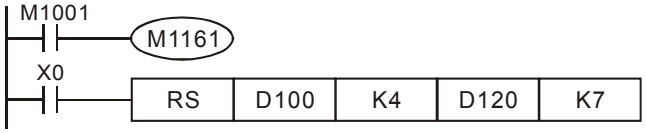
Tail code 2

685. The ELC will receive all data transmitted from the external equipment, including header and footer.

16-bit mode:

686. Start and End Characters for the ELC to transmit are set by using M1126 and M1130 and the data must be entered into D1124~D1126. The ELC will send start and end data automatically when executing the RS instruction.

687. When M1161=OFF, the mode will be 16-bit. 16-bit data will be divided into high byte and low byte for data transmitting and receiving.



Transmit data: (ELC→External equipment)

STX	D100L	D100L	D101L	D101L	ETX1	ETX2

Head code

(S) source data register will start from low byte of D100

(m) length = 4

Tail code 1

Tail code 2

Receive data: (External equipment→ELC)

D120L	D120H	D121L	D121H	D122L	D122H	D123L

Head code

(D) receive data register will start from low byte of D120

(n) length = 7

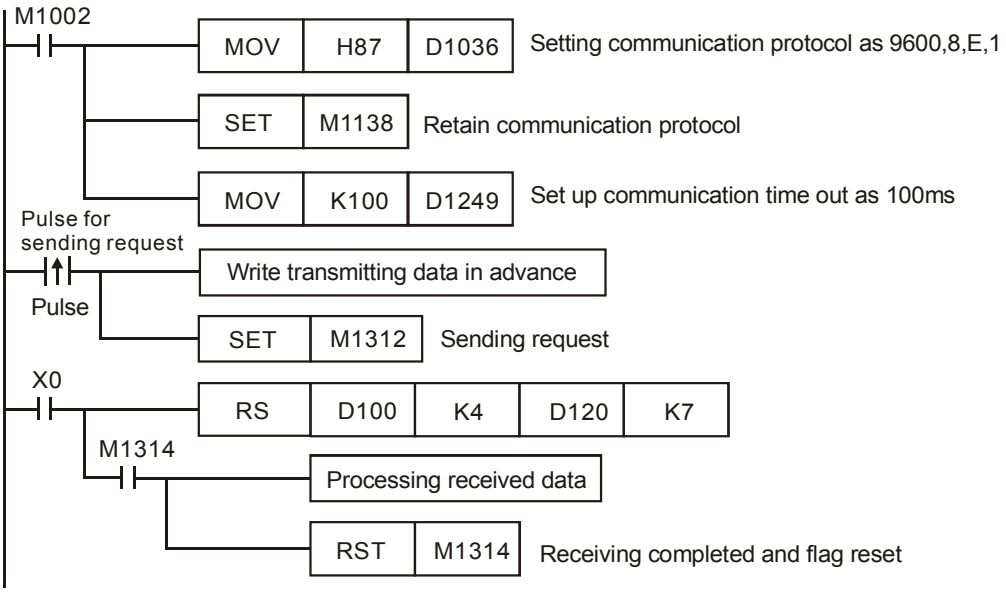
Tail code 1

Tail code 2

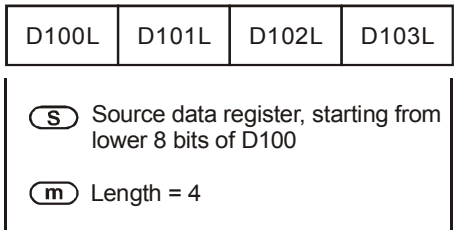
688. ELC will receive all data transmitted from external equipment, including start and end characters..

Program Example 3: COM1 RS-232

1. Only 8-bit mode is supported (ELC sets M1161 = ON automatically)
2. STX/ETX function (M1126/M1130/D1124~D1126) is not supported.
3. High byte of 16-bit data is not available. Only low byte is valid for data communication.
4. Enter the data to be transmitted into registers starting from D100 and set M1312 (COM1 sending request) ON
5. When X10 = ON, the RS instruction executes and the ELC is ready for communications. The instruction will then start to send out 4 data bytes beginning with D100. When transmitting data is complete, M1312 will be automatically reset. (DO NOT reset M1312 in the program). After approximate 1ms, the ELC will start to receive 7 data bytes and store them in 7 consecutive registers starting from D20.
6. When the data read is complete, M1314 will turn ON. When data processing on the received data is complete, M1314 must be reset (OFF) and the ELC will be ready for communication again. However, DO NOT continuously execute RST M1314.



Sending data: (ELC→External equipment)



Receving data: (External equipment→ELC)

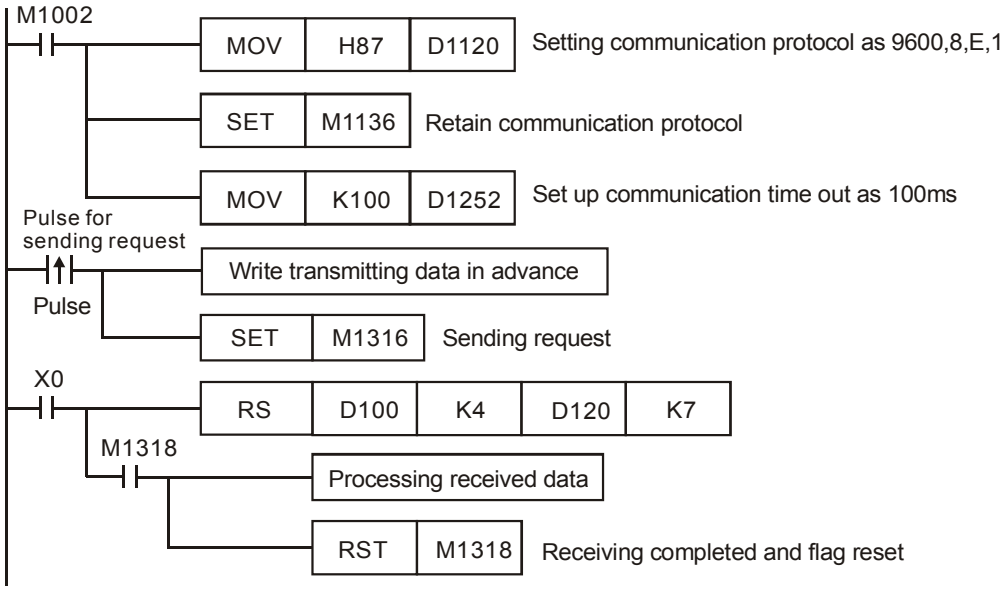
D120L	D121L	D122L	D123L	D124L	D125L	D126L

(S) Registers for received data, starting from lower 8 bits of D120

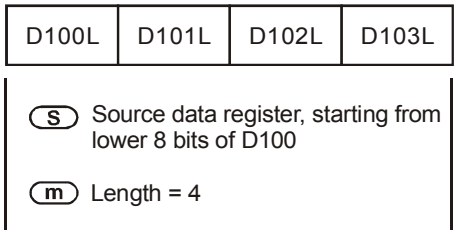
(n) Length = 7

Program Example 4: COM3 RS-485

1. Only the 8-bit mode is supported (ELC sets M1161 = ON automatically)
2. STX/ETX function (M1126/M1130/D1124~D1126) is not supported.
3. The high byte of the 16-bit data is not available. Only the low byte is valid for data communications.
4. Enter the data to be transmitted into registers starting with D100 then set M1316 (COM3 sending request) to ON
5. When X10 = ON, the RS instruction executes and the ELC is ready for communications. 4 bytes of data beginning with D100 will be sent. Each time the data has been sent, M1318 will be automatically reset. (DO NOT reset M1318 in the program). After approximate 1ms, the ELC will start to receive 7 data bytes and store them in 7 consecutive registers starting from D120.
6. When the data received is complete, M1318 will automatically be ON. When the received data is completed, M1318 must be reset (OFF) and the ELC will be ready for communications again.



Sending data: (ELC→External equipment)



Receving data: (External equipment→ELC)

D120L	D121L	D122L	D123L	D124L	D125L	D126L

(S) Registers for received data, starting from lower 8 bits of D120

(n) Length = 7

Points to note:

689. **ELC COM1 RS-232:** Special bits (M-bits) and special registers (D-registers) used for communication instructions RS / MODRD

Flag	Function	Action
M1138	COM1 lock communication settings. Communication settings will be set according to the contents in D1036 after every scan cycle, unless after setting the interface parameters with D1036, M1038 is Set. This will retain the communication protocol settings. When M1138 = ON, the communication settings for COM1 will not be changed when the content in D1036 is changed. Supported communication instructions: RS / MODRW	User sets and resets
M1139	COM1 ASCII / RTU mode selection, ON: RTU, OFF: ASCII. Supported communication instructions: RS / MODRW	User sets and resets
M1312	COM1 sending request. Before executing communication instructions, users need to set M1312 to ON with a trigger pulse, to initiate data transmission. When the communication is completed, the ELC will reset M1312 automatically. Supported communication instructions: RS / MODRW	User sets and system resets
M1313	COM1 ready to receive data. When M1313 is ON, the ELC is ready to receive data Supported communication instructions: RS / MODRW	System
M1314	M1314 will be ON when the data received is complete. Process the data received when M1314 is on. When the data has been processed, M1314 must be reset by the program. Supported communication instructions: RS / MODRW	System sets and user resets
M1315	COM1 receiving error. M1315 will be set ON when errors occur and the error code will be stored in D1250. Supported communication instructions: RS / MODRW	System sets and user resets

Special register	Function
D1036	COM1 (RS-232) communication protocol. Refer to the table in point number 4 below for the necessary settings.
D1167	The specific end word to be detected for the RS instruction to execute an interruption request (I140) on COM1 (RS-232). Supported communication instructions: RS
D1121	COM1 (RS-232) and COM2 (RS-485) communication address.
D1249	COM1 (RS-232) Communication time-out setting (unit: ms). If a time-out value is set in D1249 and if the data being received times-out, M1315 will be set ON and the error code K1 will be stored in D1250. M1315 must be reset by the program when the time-out status is cleared.
D1250	COM1 (RS-232) communication error code. Supported communication instructions: MODRW

690. **ELC COM2 RS-485:** Special M-bits and special D-registers for communication instructions RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW.

Flag	Function	Action
M1120	Retain communication settings. Communication settings will be reset (changed) according to the content in D1120 after every scan cycle. Users can set M1120 ON if the communication protocol requires the port configuration data to be retained. When M1120 = ON, communication settings cannot be modified when communication instructions are being processed, even if the content in D1120 is changed.	User sets/resets
M1121	Data transmission ready. M1121 = OFF indicates that RS-485 in COM2 is transmitting	System sets
M1122	Sending request. Before executing communication instructions, users need to set M1122 to ON by trigger pulse, to initiate sending and receiving data. When the communications is complete, the ELC will reset M1122 automatically.	User sets, system resets

Flag	Function	Action
M1123	Receiving data complete. When data is received via the RS instruction, M1123 will be ON. The received data can be processed when M1123 is ON. When data processing is completed, M1123 must be reset by the program. Supported communication instructions: RS	System sets ON and user resets
M1124	Data receiving ready. When M1124 is ON, the ELC is ready to receive data.	System sets
M1125	Communication ready status reset. When M1125 is set ON, the ELC resets the communication (transmitting/receiving) ready status. M1125 has to be reset by user after resetting the communication ready status.	User sets/resets
M1126	Set STX/ETX as user-defined or system-defined for RS communications. For details refer to the table in point 5. M1126 only supports the RS instruction.	
M1130	Set STX/ETX as user-defined or system-defined in RS communications. For details please refer to the table in point 5. M1130 only supports the RS instruction	
M1127	COM2 (RS-485) data sending/receiving/converting completed. RS instruction is NOT supported. Supported communication instructions: MODRD / MODWR / MODRW	System sets and user resets
M1128	Transmitting/receiving status indication.	System sets
M1129	Receiving time out. If time-out value is set up in D1129 and the data receiving time exceeds the time-out value, M1129 will be set ON.	System sets and user resets
M1131	In ASCII mode, M1131 = ON only when MODRD / MODRW data is being converted to HEX. Supported communication instructions: MODRD / MODRW	System sets
M1140	MODRD/MODWR/MODRW data receiving error Supported communication instructions: MODRD / MODWR / MODRW	

Flag	Function	Action
M1141	MODRD/MODWR/MODRW parameter error Supported communication instructions: MODRD / MODWR/ MODRW	System sets
M1143	ASCII / RTU mode selection. ON : RTU mode, OFF: ASCII mode. Supported communication instructions: RS / MODRD / MODWR / MODRW	User sets and resets
M1161	8/16-bit mode. ON: 8-bit mode. OFF: 16-bit mode Supported communication instructions: RS	User sets

Special register	Function
D1038	Delay time of data response when ELC is a SLAVE on COM2, COM3 RS-485 communication, Range: 0~10,000. (Unit: 0.1ms). By using ELC LINK on COM2, D1038 can be set to send next communication data with delay. (unit: one scan cycle)
D1050~D1055	Converted data for Modbus communication data processing. ELC automatically converts the ASCII data in D1070~D1085 into Hex data and stores the 16-bit Hex data in D1050~D1055 Supported communication instructions: MODRD
D1070~D1085	Received data (ASCII) from Modbus communications. When the ELC's RS-485 communication instruction receives data, the data will be saved in registers D1070~D1085. RS instruction is not supported.
D1089~D1099	Sent data of Modbus communication. When the ELC's RS-485 communication instruction (MODRD), the data will be stored in D1089~D1099. RS instruction is not supported
D1120	COM2 (RS-485) communication protocol. Refer to the following table in point 4 below for protocol settings.
D1121	COM1 (RS-232) and COM2 (RS-485) ELC communication address.
D1122	COM2 (RS-485) number of words of transmitted data.
D1123	COM2 (RS-485) number of words of the received data.
D1124	COM2 (RS-485) Definition of start character (STX)

Special register	Function
	Supported communication instruction: RS
D1125	COM2 (RS-485) Definition of first ending character (ETX1) Supported communication instruction: RS
D1126	COM2 (RS-485) Definition of second ending character (ETX2) Supported communication instruction: RS
D1129	COM2 (RS-485) Communication time-out setting (unit: ms). If a time-out value is moved into D1129 and the data received time exceeds the time-out value, M1129 will be set ON and the error code K1 will be stored in D1130. M1129 must be reset manually when the time-out status is cleared.
D1130	COM2 (RS-485) Error code returned from Modbus. RS instruction is not included. Supported communication instructions: MODRD / MODWR / MODRW
D1168	The specific end word to be detected for RS instruction to execute an interrupt request (I150) on COM2 (RS-485). Supported communication instruction: RS
• D1169	For RS instruction, when the received data length = the low byte of D1169, the interrupt I160 will be triggered. When D1169 = 0, I160 will not be triggered. Supported communication instruction: RS
D1256~D1295	For COM2 RS-485 MODRW instruction. D1256~D1295 store the sent data of MODRW instruction. When MODRW instruction sends out data, the data will be stored in D1256~D1295. Users can check the sent data in these registers. Supported communication instruction: MODRW
D1296~D1311	For COM2 RS-485 MODRW instruction. D1296~D1311 store the converted hex data from D1070 ~ D1085 (ASCII). ELC automatically converts the received ASCII data in D1070 ~ D1085 into hex data and places it into D1296-D1311. Supported communication instruction: MODRW

3

691. **ELC COM3 RS-485:** Special M-bits and D-registers for communication instructions RS / MODRW.

Flag	Function	Action
M1136	COM3 retain communication settings. Communication settings will be reset (changed) according to the content in D1109 after every scan cycle. Users can set ON M1136 if the communication protocol requires to be retained. When M1136 = ON, communication settings will not be reset (changed) when communication instructions are being processed, even if the content in D1109 is changed. Supported communication instructions: RS / MODRW	User sets and resets
M1320	COM3 ASCII / RTU mode selection. ON: RTU, OFF: ASCII. Supported communication instructions: RS / MODRW	
M1316	COM3 sending request. Before executing communication instructions, set M1316 to ON by with a pulse, to initiate sending and receiving data. When communications is completed, ELC will reset M1316 automatically. Supported communication instructions: RS / MODRW	User sets, system resets
M1317	Data receive ready. When M1317 is ON, ELC is ready to receive data. Supported communication instructions: RS / MODRW	System sets
M1318	COM3 data receive complete. Supported communication instructions: RS / MODRW	System sets, user resets
M1319	COM3 data receive error. M1319 will be set ON when errors occur and the error code will be stored in D1252 Supported communication instructions: RS / MODRW	System sets, user resets

Special register	Function
D1038	Delay time of data response when the ELC is a SLAVE on COM2, COM3 RS-485 ports, Range: 0~10,000. (unit: 0.1ms). By using ELC LINK on COM2, D1038 can be set to send next communication data with delay. (unit: one scan cycle)
D1109	COM3 (RS-485) communication protocol. Refer to the following table in point 4 below for protocol settings.
D1169	The specific end word to be detected for RS instruction to execute an interrupt request (I160) on COM3 (RS-485). Supported communication instructions: RS
D1252	COM3 (RS-485) Communication time-out setting (ms). If a time-out value is set up in D1252 and the data receiving time exceeds the time-out value, M1319 will be set ON and the error code K1 will be stored in D1253. M1319 has to be reset manually when time-out status is cleared.
D1253	COM3 (RS-485) communication error code
D1255	COM3 (RS-485) ELC communication address when ELC is Slave.

692. COM port and communication settings/status table.

	COM1	COM2	COM3	Function Description
Protocol setting	M1138	M1120	M1136	Retain communication setting
	M1139	M1143	M1320	ASCII/RTU mode selection
	D1036	D1120	D1109	Communication protocol
	D1121	D1121	D1255	ELC communication address
Sending request	-	M1161	-	8/16 bit mode selection
	-	M1121	-	Indicates transmission status
	M1312	M1122	M1316	Sending request
	-	M1126	-	Set STX/ETX as user/system defined. (RS)
	-	M1130	-	Set STX/ETX as user/system defined. (RS)
	-	D1124	-	Definition of STX (RS)
	-	D1125	-	Definition of ETX1 (RS)
	-	D1126	-	Definition of ETX2 (RS)
	D1249	D1129	D1252	Communication timeout setting (ms)
	-	D1122	-	Residual number of words of transmitting data

	COM1	COM2	COM3	Function Description
Sending request	-	D1256 ~ D1295	-	Store the sent data of MODRW instruction.
	-	D1089 ~ D1099	-	Store the sent data of MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF instruction
Data receiving	M1313	M1124	M1317	Data receiving ready
	-	M1125	-	Communication ready status reset
	-	M1128	-	Transmitting/Receiving status Indication
	-	D1123	-	Residual number of words of the receiving data
	-	D1070 ~ D1085	-	Store the feedback data of Modbus communication. RS instruction is not supported.
	D1167	D1168	D1169	Store the specific end word to be detected for executing interrupts I140/I150/I160 (RS)
Receiving completed	M1314	M1123	M1318	Data receiving completed
	-	M1127	-	COM2 (RS-485) data sending / receiving / converting completed. (RS instruction is not supported)
	-	M1131	-	ON when MODRD/RDST/MODRW data is being converted from ASCII to Hex
	-	D1296 ~ D1311	-	Store the converted HEX data of MODRW instruction.
	-	D1050 ~ D1055	-	Store the converted HEX data of MODRD instruction

	COM1	COM2	COM3	Function Description
Errors	M1315	-	M1319	Data receiving error
	D1250	-	D1253	Communication error code
	-	M1129	-	COM2 (RS-485) receiving time out
	-	M1140	-	COM2 (RS-485) MODRD/MODWR/MODRW data receiving error
	-	M1141	-	MODRD/MODWR/MODRW parameter error (Exception Code exists in received data) Exception Code is stored in D1130
	-	M1142	-	Data receiving error of VFD-A handy instructions (FWD/REV/STOP/RDST/RSTEF)
Errors	-	D1130	-	COM2 (RS-485) Error code returning from Modbus communication

693. Communication protocol settings: D1036(COM1 RS-232) / D1120(COM2 RS-485) / D1109(COM3 RS-485)

	Content		
b0	Data Length	**0:** 7 data bits	**1:** 8 data bits
b1 b2	Parity bit	00: None 01: Odd 11: Even	
b3	Stop bits	**0:** 1 bit	**1:** 2bits
b4 b5 b6 b7	Baud rate	0001(H1):	110 bps
		0010(H2):	150 bps
		0011(H3):	300 bps
		0100(H4):	600 bps
		0101(H5):	1200 bps
		0110(H6):	2400 bps
		0111(H7):	4800 bps
		1000(H8):	9600 bps
		1001(H9):	19200 bps
		1010(HA):	38400 bps
		1011(HB):	57600 bps
		1100(HC):	115200 bps
		1101(HD):	500000 bps (ELCM-PH/PA COM2 / COM3 support)
		1110(HE):	31250 bps

	Content			
		1111(HF):	921000 bps	
b8 (D1120)	STX	0: None	1: D1124	
b9 (D1120)	ETX1	0: None	1: D1125	
b10 (D1120)	ETX2	0: None	1: D1126	
b11~b15	N/A			

694. When RS instruction is applied for communication between ELC and peripheral devices on COM2 RS-485, usually STX (Start of the text) and ETX (End of the text) have to be set into communication format. In this case, b8~10 of D1120 should be set to 1, so that users can set up STX/ETX as user-defined or system-defined by using M1126, M1130, and D1124~D1126. For settings of M1126 and M1130, please refer to the following table.

		M1130	
		0	1
M1126	0	D1124: user defined D1125: user defined D1126: user defined	D1124: H 0002 D1125: H 0003 D1126: H 0000 (no setting)
	1	D1124: user defined D1125: user defined D1126: user defined	D1124: H 003A (':') D1125: H 000D (CR) D1126: H 000A (LF)

695. Example of setting communication format in D1120:

Communication format:

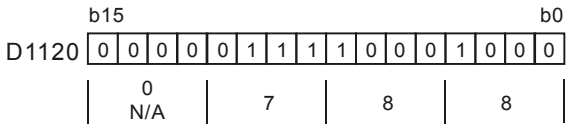
Baud rate: 9600, 7, N, 2

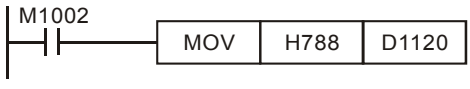
STX: ": "

ETX1: "CR"

ETX2: "LF"

Check to the table in point 4 and the set value H788 can be referenced corresponding to the baud rate. Set the value into D1120.





When STX, ETX1 and ETX2 are applied, care should be taken on setting the ON/OFF status of M1126 and M1130.

696. D1250(COM1)、D1253(COM3) communication error code:

Value	Error Description
H0001	Communication time-out
H0002	Checksum error
H0003	Exception Code exists
H0004	Command code error / data error
H0005	Communication data length error

697. Corresponding table between D1167~D1169 and the associated interrupt pointers. (Only lower 8 bits are valid)

COM Port	I1□0 interrupt	Special D
COM1	I140	D1167
COM2	I150	D1168
COM3	I160	D1169

698. Take standard MODBUS format for example:

ASCII mode

Field Name	Descriptions
STX	Start word = ': ' (3AH)
Address Hi	Communication address:
Address Lo	The 8-bit address consists of 2 ASCII codes
Function Hi	Function code:
Function Lo	The 8-bit function code consists of 2 ASCII codes
DATA (n-1)	Data content:
…….	n × 8-bit data content consists of 2n ASCII codes
DATA 0	
LRC CHK Hi	LRC check sum:
LRC CHK Lo	8-bit check sum consists of 2 ASCII code
END Hi	End word:
END Lo	END Hi = CR (0DH), END Lo = LF(0AH)

The communication protocol is in Modbus ASCII mode, i.e. every byte is composed of 2 ASCII characters. For example, 64Hex is '64' in ASCII, composed by '6' (36Hex) and '4' (34Hex). Every character '0'…'9', 'A'…'F' corresponds to an ASCII code.

Character	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'
ASCII code	30H	31H	32H	33H	34H	35H	36H	37H

Character	'8'	'9'	'A'	'B'	'C'	'D'	'E'	'F'

ASCII code	38H	39H	41H	42H	43H	44H	45H	46H

Start word (STX): ':' (3AH)

Address:

'0' '0': Broadcasting to all drives (Broadcast)

'0' '1': toward the drive at address 01

'0' 'F': toward the drive at address 15

'1' '0': toward the drive at address 16

… and so on, max. address: 254 ('FE')

Function code:

'0' '3': read contents from multiple registers

'0' '6': write one word into a single register

'1' '0': write contents to multiple registers

Data characters:

The data sent by the user

LRC checksum:

LCR checksum is 2's complement of the value added from Address to Data Characters.

For example: 01H + 03H + 21H + 02H + 00H + 02H = 29H. 2's complement of 29H = D7H.

End word (END):

Fix the END as END Hi = CR (0DH), END Lo = LF (0AH)

Example:

Read 2 continuous data stored in the registers of the drive at address 01H (see the table below). The start register is at address 2102H.

Inquiry message:

STX	': '
Address	'0'
	'1'
Function code	'0'
	'3'
Start address	'2'
	'1'
	'0'
	'2'
Number of data (count by word)	'0'
	'0'
	'0'
	'2'
LRC Checksum	'D'
	'7'
END	CR
	LF

Response message:

STX	': '
Address	'0'
	'1'
Function code	'0'
	'3'
Number of data (count by byte)	'0'
	'4'
Content of start address 2102H	'1'
	'7'
	'7'
	'0'
Content of address 2103H	'0'
	'0'
	'0'
	'0'
LRC Checksum	'7'
	'1'
END	CR

	LF

RTU mode

Field Name	Descriptions
START	Refer to the following explanation
Address	Communication address: n 8-bit binary
Function	Function code: n 8-bit binary
DATA (n-1)	Data: n × 8-bit data
…….	
DATA 0	
CRC CHK Low	CRC checksum: 16-bit CRC consists of 2 8-bit binary data
CRC CHK High	
END	Refer to the following explanation

START/END:

RTU Timeout Timer:

Baud rate(bps)	RTU timeout timer (ms)	Baud rate (bps)	RTU timeout timer (ms)
300	40	9,600	2
600	21	19,200	1
1,200	10	38,400	1
2,400	5	57,600	1
4,800	3	115,200	1

Address:

00 H: Broadcasting to all drives (Broadcast)

01 H: toward the drive at address 01

0F H: toward the drive at address 15

10 H: toward the drive at address 16

… and so on, max. address: 254 (‘FE’)

Function code:

03 H: read contents from multiple registers

06 H: write one word into single register

10 H: write contents to multiple registers

Data characters:

The data sent by the user

CRC checksum: Starting from Address and ending at Data Content. The calculation is as follows:

- Step 1: Set the 16-bit register (CRC register) = FFFFH
- Step 2: Operate XOR on the first 8-bit message (Address) and the lower 8 bits of CRC register. Store the result in the CRC register.

- Step 3: Right shift CRC register for a bit and fill “0” into the highest bit.
- Step 4: Check the lowest bit (bit 0) of the shifted value. If bit 0 is 0, fill in the new value obtained at step 3 to CRC register; if bit 0 is NOT 0, operate XOR on A001H and the shifted value and store the result in the CRC register.
- Step 5: Repeat step 3 – 4 to finish all operation on all the 8 bits.
- Step 6: Repeat step 2 – 5 until the operation of all the messages are completed. The final value obtained in the CRC register is the CRC checksum. Care should be taken when placing the LOW byte and HIGH byte of the obtained CRC checksum.

Example:

Read 2 continuous data words stored in the registers of the drive at address 01H (see the table below). The start register is at address 2102H

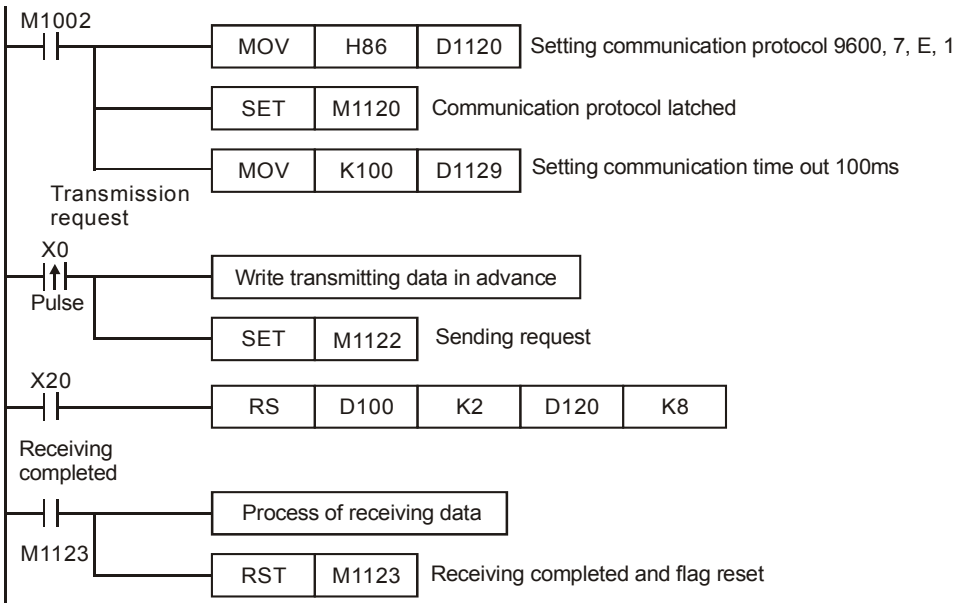
Inquiry message:

Field Name	Data (Hex)
Address	01 H
Function	03 H
Start data address	21 H
	02 H
Number of data (count by word)	00 H
	02 H
CRC CHK Low	6F H
CRC CHK High	F7 H

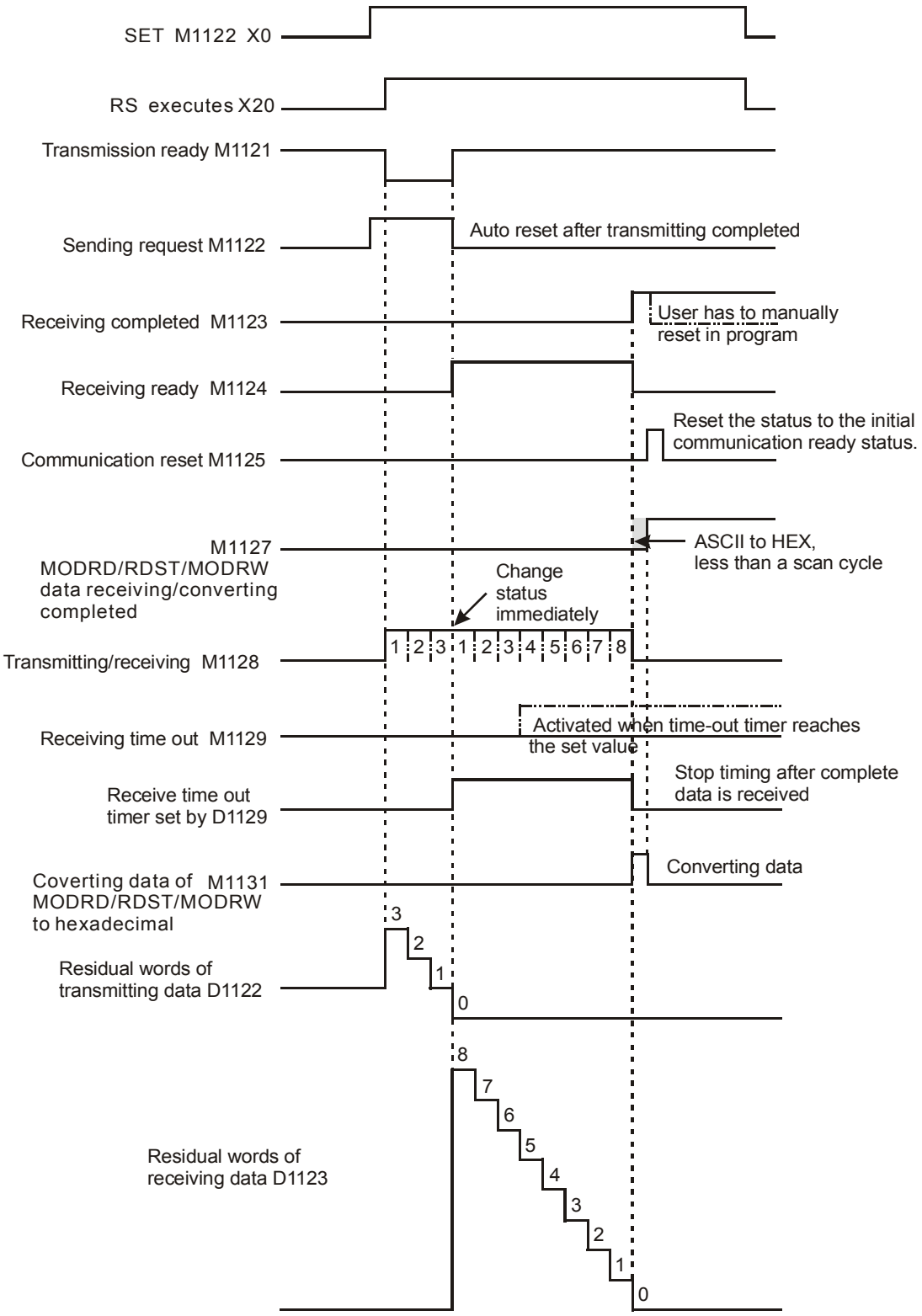
Response message:

Field Name	Data (Hex)
Address	01 H
Function	03 H
Number of data (count by byte)	04 H
Content of data address 2102H	17 H
	70 H
Content of data address 2103H	00 H
	00 H
CRC CHK Low	FE H
CRC CHK High	5C H

Example program of RS-485 communication:



- Timing diagram:



API	Mnemonic		Operands	Function
81	D	PRUN	S, D	Parallel Run

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*		*							PRUN, PRUNP: 5 steps
D								*	*							DPRUN, DPRUNP: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

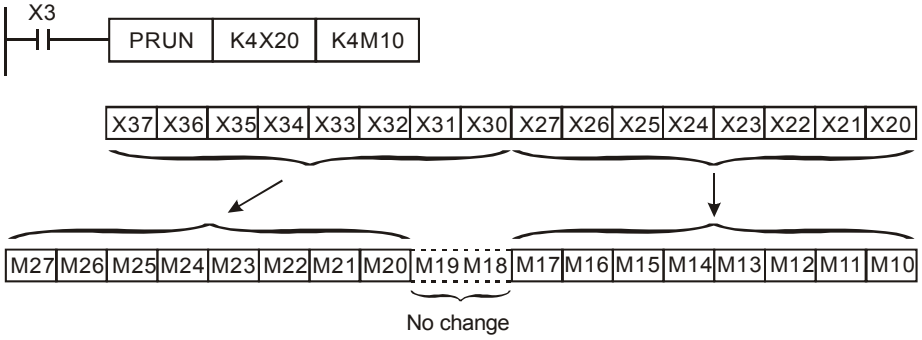
S: Transmission source address **D**: Destination address

Description:

699. Transmit the content of **S** to **D** in octal number system format.
700. X, Y, M of KnX, KnY, KnM should be a multiple of 10, e.g. X20, M20, Y20.
701. When operand **S** is specified as KnX, operand **D** should be specified as KnM.
702. When operand **S** is specified as KnM, operand **D** should be specified as KnY.

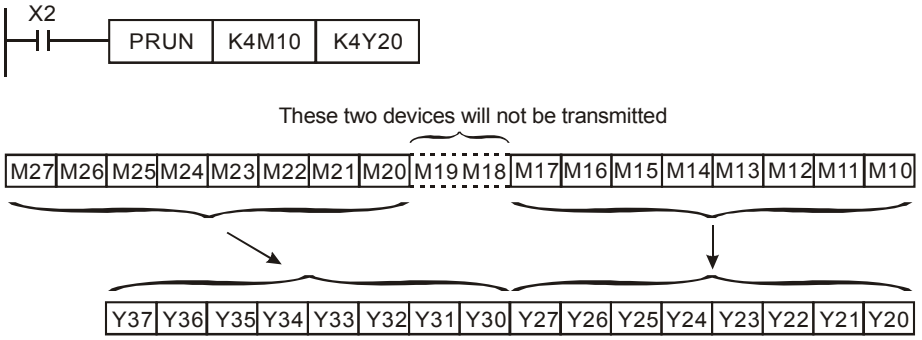
Program Example 1:

When X3=ON, transmit the contents of K4X20 to K4M10 in octal number system format.



Program Example 2:

When X2=ON, transmit the content of K4M10 to K4Y20 in octal number system format.



API	Mnemonic		Operands	Function
82	ASCI	P	**S, D, n**	Converts HEX to ASCII

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*			ASCI, ASCIP: 7 steps
D								*	*	*	*	*	*			
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

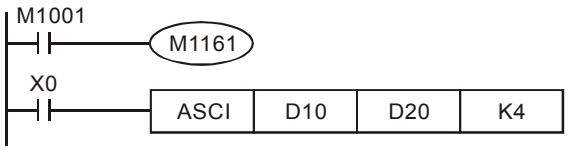
S: Source data **D**: Destination of the result **n**: Number of digits to convert (**n**=1~256)

Description:

703. 16-bit conversion mode: When M1161=OFF (16-bit conversion mode), read **n** hexadecimal characters from source **S** and convert them to ASCII. Then, store the result into high and low bytes of **D**.
704. 8-bit conversion mode: When M1161=ON (8-bit conversion mode), read **n** hexadecimal characters from source **S** and convert them to ASCII. Then, store the result into the low byte of **D** (high byte of **D** will be set to 0).
705. Available range for Hex data: 0~9, A~F.

Program Example 1:

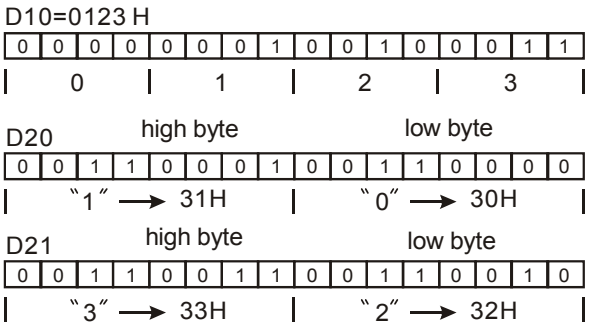
706. When M1161=OFF, conversion mode is 16-bit.
707. When X0=ON, read four hexadecimal characters from D10 and convert them into ASCII. Then, store the converted characters in the register started from D20.



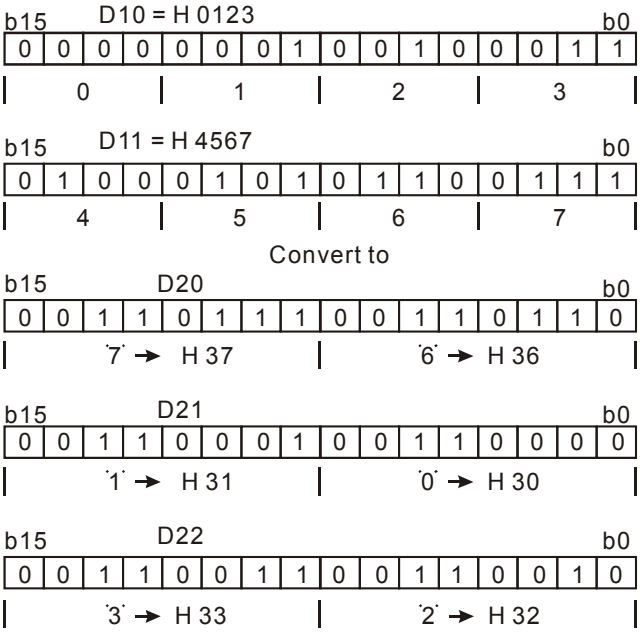
708. Assume:

(D10) = 0123 H	'0' = 30H	'4' = 34H	'8' = 38H
(D11) = 4567 H	'1' = 31H	'5' = 35H	'9' = 39H
(D12) = 89AB H	'2' = 32H	'6' = 36H	'A' = 41H
(D13) = CDEF H	'3' = 33H	'7' = 37H	'B' = 42H

709. When **n** is 4, the bit structure is:



710. When **n** is 6, the bit structure is:



711. When **n** = 1 to 16:

D \ n	K1	K2	K3	K4	K5	K6	K7	K8
D20 low byte	“3”	“2”	“1”	“0”	“7”	“6”	“5”	“4”
D20 high byte		“3”	“2”	“1”	“0”	“7”	“6”	“5”
D21 low byte			“3”	“2”	“1”	“0”	“7”	“6”
D21 high byte				“3”	“2”	“1”	“0”	“7”
D22 low byte					“3”	“2”	“1”	“0”
D22 high byte						“3”	“2”	“1”
D23 low byte							“3”	“2”
D23 high byte								“3”
D24 low byte								
D24 high byte								
D25 low byte								
D25 high byte								
D26 low byte								
D26 high byte								
D27 low byte								
D27 high byte								

No change

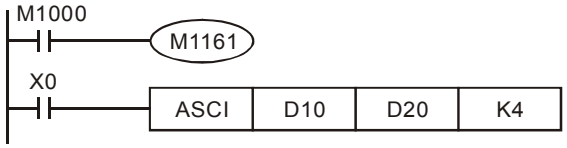
D \ n	K9	K10	K11	K12	K13	K14	K15	K16
D20 low byte	“B”	“A”	“9”	“8”	“F”	“E”	“D”	“C”
D20 high byte	“4”	“B”	“A”	“9”	“8”	“F”	“E”	“D”
D21 low byte	“5”	“4”	“B”	“A”	“9”	“8”	“F”	“E”
D21 high byte	“6”	“5”	“4”	“B”	“A”	“9”	“8”	“F”
D22 low byte	“7”	“6”	“5”	“4”	“B”	“A”	“9”	“8”
D22 high byte	“0”	“7”	“6”	“5”	“4”	“B”	“A”	“9”
D23 low byte	“1”	“0”	“7”	“6”	“5”	“4”	“B”	“A”
D23 high byte	“2”	“1”	“0”	“7”	“6”	“5”	“4”	“B”
D24 low byte	“3”	“2”	“1”	“0”	“7”	“6”	“5”	“4”
D24 high byte		“3”	“2”	“1”	“0”	“7”	“6”	“5”
D25 low byte			“3”	“2”	“1”	“0”	“7”	“6”
D25 high byte				“3”	“2”	“1”	“0”	“7”
D26 low byte					“3”	“2”	“1”	“0”
D26 high byte						“3”	“2”	“1”
D27 low byte							“3”	“2”
D27 high byte								“3”

No change

Program Example 2:

712. When M1161=ON, conversion mode is 8-bit.

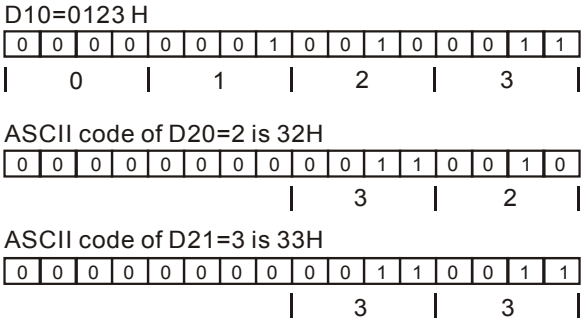
713. When X0=ON, read four hexadecimal data characters starting from D10 and convert them to ASCII. Then, store the converted data in the register starting from D20.



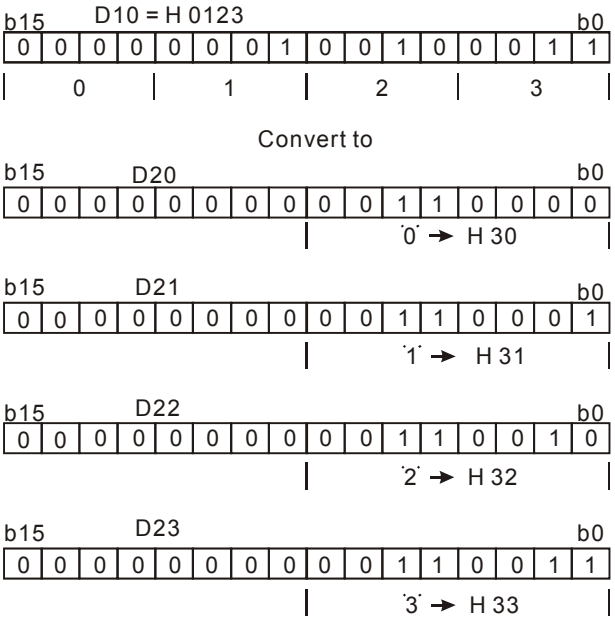
714. Assume:

(D10) = 0123 H	'0' = 30H	'4' = 34H	'8' = 38H
(D11) = 4567 H	'1' = 31H	'5' = 35H	'9' = 39H
(D12) = 89AB H	'2' = 32H	'6' = 36H	'A' = 41H
(D13) = CDEFH	'3' = 33H	'7' = 37H	'B' = 42H

715. When **n** is 2, the bit structure is:



716. When **n** is 4, the bit structure is:



717. When **n** = 1 to 16:

D \ n	K1	K2	K3	K4	K5	K6	K7	K8
D20	“3”	“2”	“1”	“0”	“7”	“6”	“5”	“4”
D21		“3”	“2”	“1”	“0”	“7”	“6”	“5”
D22			“3”	“2”	“1”	“0”	“7”	“6”
D23				“3”	“2”	“1”	“0”	“7”
D24					“3”	“2”	“1”	“0”
D25						“3”	“2”	“1”
D26							“3”	“2”
D27								“3”
D28								
D29								
D30								
D31								
D32								
D33								
D34								
D35								

No change

D \ n	K9	K10	K11	K12	K13	K14	K15	K16
D20	“B”	“A”	“9”	“8”	“F”	“E”	“D”	“C”
D21	“4”	“B”	“A”	“9”	“8”	“F”	“E”	“D”
D22	“5”	“4”	“B”	“A”	“9”	“8”	“F”	“E”
D23	“6”	“5”	“4”	“B”	“A”	“9”	“8”	“F”
D24	“7”	“6”	“5”	“4”	“B”	“A”	“9”	“8”
D25	“0”	“7”	“6”	“5”	“4”	“B”	“A”	“9”
D26	“1”	“0”	“7”	“6”	“5”	“4”	“B”	“A”
D27	“2”	“1”	“0”	“7”	“6”	“5”	“4”	“B”
D28	“3”	“2”	“1”	“0”	“7”	“6”	“5”	“4”
D29		“3”	“2”	“1”	“0”	“7”	“6”	“5”
D30			“3”	“2”	“1”	“0”	“7”	“6”
D31				“3”	“2”	“1”	“0”	“7”
D32					“3”	“2”	“1”	“0”
D33						“3”	“2”	“1”
D34							“3”	“2”
D35								“3”

No change

API	Mnemonic		Operands	Function
83	HEX	P	S, D, n	Converts ASCII to HEX

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*			HEX, HEXP: 7 steps
D								*	*	*	*	*	*			
n					*	*										

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

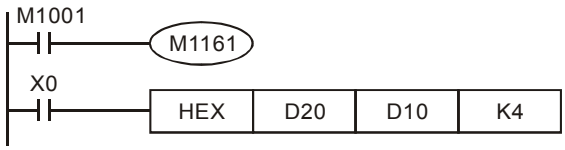
S: source data **D**: Destination for the result **n**: number of digits to convert (**n**=1~256)

Description:

718. 16-bit conversion mode: When M1161=OFF, the conversion mode is 16-bit. Convert 16-bit ASCII data of **S** (high and low byte) to hexadecimal and then store the result in **D**. The number of converted ASCII characters is set by **n**.
719. 8-bit conversion mode: When M1161=ON, the conversion mode is 8-bit. Convert 16-bit ASCII code of **S** (high and low byte) to hexadecimal data characters and then transmit to low byte of **D**. The number of converted ASCII codes is set by **n**. (high byte of **D** set to 0)
720. Available range for Hex data: 0~9, A~F

Program Example 1:

721. When M1161=OFF, it is 16-bit conversion mode.
722. When X0=ON, read ASCII bytes from starting register D20 and convert them to hexadecimal characters. Then, store the converted data to four registers starting with D10.

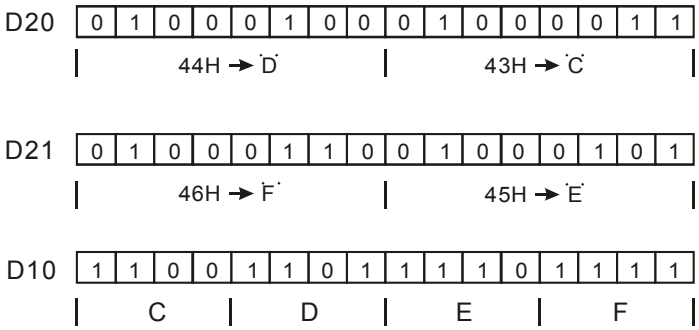


723. Assume:

S	ASCII code	HEX conversion	S	ASCII code	HEX conversion
D20 low byte	H 43	“C”	D24 low byte	H 34	“4”
D20 high byte	H 44	“D”	D24 high byte	H 35	“5”
D21 low byte	H 45	“E”	D25 low byte	H 36	“6”
D21 high byte	H 46	“F”	D25 high byte	H 37	“7”
D22 low byte	H 38	“8”	D26 low byte	H 30	“0”
D22 high byte	H 39	“9”	D26 high byte	H 31	“1”
D23 low byte	H 41	“A”	D27 low byte	H 32	“2”

S	ASCII code	HEX conversion	S	ASCII code	HEX conversion
D23 high byte	H 42	“B”	D27 high byte	H 33	“3”

724. When **n** is 4, the bit structure is:



725. When **n** = 1 to 16:

n \ D	D13	D12	D11	D10
1	The used registers which are not specified are all 0			***C H
2				**CD H
3				*CDE H
4				CDEF H
5			***C H	DEF8 H
6			**CD H	EF89 H
7			*CDE H	F89A H
8			CDEF H	89AB H
9		***C H	DEF8 H	9AB4 H
10		**CD H	EF89 H	AB45 H
11		*CDE H	F89A H	B456 H
12		CDEF H	89AB H	4567 H
13	***C H	DEF8 H	9AB4 H	5670 H
14	**CD H	EF89 H	AB45 H	6701 H
15	*CDE H	F89A H	B456 H	7012 H
16	CDEF H	89AB H	4567 H	0123 H

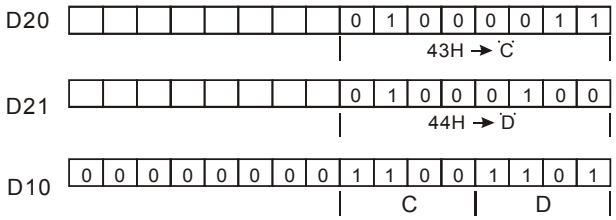
Program Example 2:

726. When M1161=ON, it is 16-bit conversion mode.

727. Assume:

S	ASCII code	HEX conversion	S	ASCII code	HEX conversion
D20	H 43	“C”	D28	H 34	“4”
D21	H 44	“D”	D29	H 35	“5”
D22	H 45	“E”	D30	H 36	“6”
D23	H 46	“F”	D31	H 37	“7”
D24	H 38	“8”	D32	H 30	“0”
D25	H 39	“9”	D33	H 31	“1”
D26	H 41	“A”	D34	H 32	“2”
D27	H 42	“B”	D35	H 33	“3”

728. When **n** is 2, the bit structure is



729. When **n** = 1 to 16:

n \ D	D13	D12	D11	D10
1	The used registers which are not specified are all 0			***C H
2				**CD H
3				*CDE H
4				CDEF H
5			***C H	DEF8 H
6			**CD H	EF89 H
7			*CDE H	F89A H
8			CDEF H	89AB H
9		***C H	DEF8 H	9AB4 H
10		**CD H	EF89 H	AB45 H
11		*CDE H	F89A H	B456 H
12		CDEF H	89AB H	4567 H
13	***C H	DEF8 H	9AB4 H	5670 H
14	**CD H	EF89 H	AB45 H	6701 H
15	*CDE H	F89A H	B456 H	7012 H
16	CDEF H	89AB H	4567 H	0123 H

API	Mnemonic		Operands	Function
84	CCD	P	S, D, n	Check Code

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			CCD, CCDP: 7 steps
D									*	*	*	*	*			
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

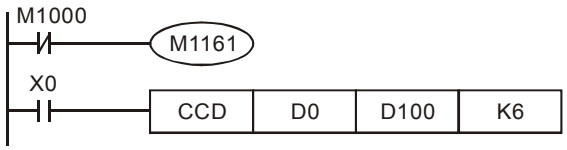
S: source address **D**: Destination for storing the check sum **n**: Number of values to use in the instruction (**n**=1~256)

Description:

730. This instruction is used to create the check sum for data packets to ensure data integrity.
731. 16-bit conversion mode: When M1161=OFF, the conversion mode is 16-bit. Create a check sum of **n** words from the register specified by **S** and store the check sum in the register specified by **D**. The parity bits are in **D** +1.
732. 8-bit conversion mode: When M1161=ON, conversion mode is 8-bit. Create a checksum of **n** words (8-bits per byte, only low bytes are available) from the register specified by **S** and store the check sum in the register specified **D**. The parity bits are in **D** +1

Program Example 1:

733. When M1161=OFF, the conversion mode is 16-bit.
734. When X0=ON, the checksum for the 6 words from registers D0-D5 is calculated and stored in D100. The parity bits are stored in D101.



(S)	Content of data(words)									
D0 low byte	K100	=	0	1	1	0	0	1	0	0	
D0 high byte	K111	=	0	1	1	0	1	1	1	①	←
D1 low byte	K120	=	0	1	1	1	1	0	0	0	
D1 high byte	K202	=	1	1	0	0	1	0	1	0	
D2 low byte	K123	=	0	1	1	1	1	0	1	①	←
D2 high byte	K211	=	1	1	0	1	0	0	1	①	←
D100	K867										Total
D101			0	0	0	1	0	0	0	①	

• An even result is indicated by the use of 0(zero)
• An odd result is indicated by the use of 1(one)

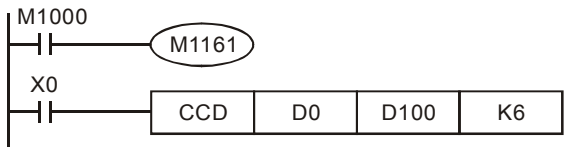
D100	0	0	0	0	0	0	1	1	0	1	1	0	0	0	1	1

D101	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	⟸ Parity

Program Example 2:

735. When M1161=ON, it is 8-bit conversion mode.

736. When X0=ON, the checksum of the 6 words from registers D0-D5 is calculated and stored in D100. The parity bits are stored in D101.



(S)	Content of data(words)									
D0 low byte	K100	=	0	1	1	0	0	1	0	0	
D1 low byte	K111	=	0	1	1	0	1	1	1	①	←
D2 low byte	K120	=	0	1	1	1	1	0	0	0	
D3 low byte	K202	=	1	1	0	0	1	0	1	0	
D4 low byte	K123	=	0	1	1	1	1	0	1	①	←
D5 low byte	K211	=	1	1	0	1	0	0	1	①	←
D100	K867										Total
D101			0	0	0	1	0	0	0	①	

• An even result is indicated by the use of 0(zero)
• An odd result is indicated by the use of 1(one)

D100	0	0	0	0	0	0	1	1	0	1	1	0	0	0	1	1

D101	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	⟸ Parity

API	Mnemonic		Operands	Function
85	VRRD	P	**S, D**	Volume Read

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*										VRRD, VRRDP: 5 steps
D								*	*	*	*	*	*	*	*	

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

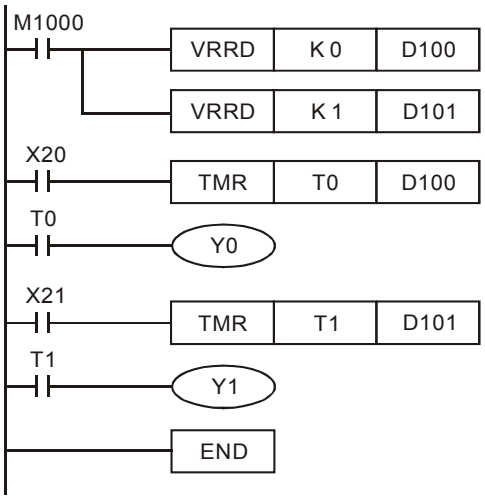
S: Variable resistor number (0~1) **D**: Destination for storing the values

Description:

737. VRRD instruction is used to read the two variable resistors on ELC. The read value will be scaled to a decimal range of 0 to 255 and stored in destination **D**.
738. For information on flags: M1178 and M1179 see the Note below.

Program Example:

Variable resistor values are used in this example as set points for two timers. The VRRD instructions read the VR values as decimal values from 0-255. Since the timers are 100ms time-base timers, the range for the VR values as presets for these timers are 0-25.5 seconds.



Note:

739. VR means VARIABLE RESISTOR.
740. The Variable Resistor. values are from the special D registers below and the M bits are used to enable each VR.

Device	Function
M1178	Start volume VR0
M1179	Start volume VR1
D1178	VR0 value
D1179	VR1value

API	Mnemonic		Operands	Function
86	VRSC	P	**S, D**	Volume Scale Read

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*										VRSC, VRSCP: 5 steps
D								*	*	*	*	*	*	*	*	

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

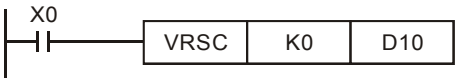
S: Variable resistor number (0~1) **D**: Destination address for the scaled value

Explanation:

The VRSC instruction reads the variable resistor values and scales them to 0-10 decimal. The scaled values from the 2 Variable Resistors on ELC are referenced by VR 0 and VR 1. The read data will be stored in destination address **D**.

Program Example 1:

When X0=ON, the scaled value (0 to10) of VR 0 is stored in address D10.



API	Mnemonic			Operands	Function
87	D	ABS	P	**D**	Absolute Value

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	ABS, ABSP: 3 steps DABS, DABSP: 5 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

D: Source and destination for absolute value

Explanation:

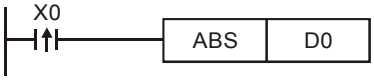
741. When the instruction is executed, take the absolute value of the specified value in **D**.

742. This instruction works best using the pulse option (ABSP, DABSP).

743. If **D** uses index F, then only the 16-bit instruction is available.

Program Example:

When X0 goes from OFF→ON, take the absolute value of the contents of D0.



API	Mnemonic		Operands	Function
88	D	PID	S_1, S_2, S_3, D	PID Calculation

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			PID : 9 steps
S_2													*			DPID: 17 steps
S_3													*			
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

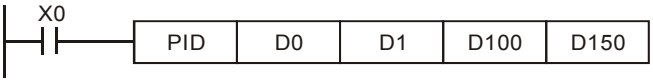
S_1: Target value (SV) S_2: Present value (PV) S_3: Parameters (for 16-bit instruction, uses 20 continuous addresses, for 32-bit instruction, uses 21 continuous addresses) **D**: Output value (MV)

Description:

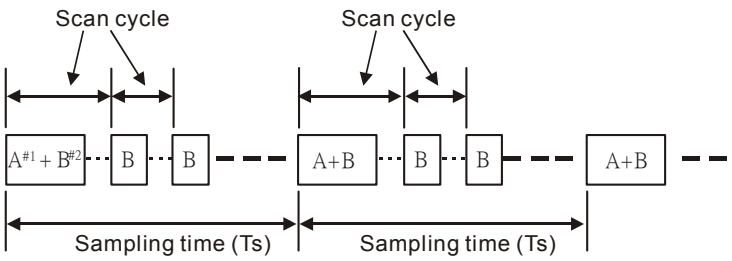
1. This instruction is specifically for PID control. PID operation will be executed only when the sampling time is reached. PID refers to “proportion, integration and derivative”. PID control is widely applied to mechanical, pneumatic and electronic equipment.
2. After all the parameters are set up, the PID instruction can be executed and the results will be stored in **D**. **D** must be an unlatched data register. (If users want to designate a latched data register area, clear the latched registers to 0 at the beginning of the user program.

Program Example:

744. Pre-write the PID parameters into all the registers before executing the PID instruction.
745. This instruction will be executed when X0=ON and the result will be stored in D150. The instruction will not be executed when X0=OFF and the data will be unchanged.



746. The timing chart for the PID instruction (ELC-PA/PV, ELCB-PB max. operation time is 474us, ELCM-PH/PA max. operation time is 80us)



Note1: The time for equation calculation during PID operation (ELC- PA/PV, ELCB-PB, ELC2-PV about 430us) (ELCM-PH/PA, ELC2-PB/PH/PA/PE approx. 72us)

Note2: The PID operation time without equation calculation (ELC- PA/PV, ELCB-PB, ELC2-PV about 44us) (ELCM-PH/PA, ELC2-PB/PH/PA/PE approx. 8us)

Points to note:

747. There is no limit on the number of times the PID instruction can be used in a program, but the addresses S_3~ S_3+19 cannot be repeated.

748. For 16-bit instructions: S_3 uses 20 registers. In the example below the parameter area of the PID instruction for S_3 is D100~D119.

749. Parameter table of 16-bit S_3:

Device No.	Function	Set-point range	Explanation
S_3:	Sampling time (T_S)	1~2,000 (unit: 10ms)	Time interval between PID calculations and updates of MV. If T_S = 0, the PID instruction will not be enabled. If T_S is less than 1 program scan time, the PID instruction sets S_3 to 1 program scan time, i.e. the value of T_S cannot be less than one program scan time.
S_3+1:	Propotional gain (K_P)	0~30,000(%)	The proportion for minimizing the error between SV and PV.
S_3+2:	Integral gain (K_I)	0~30,000(%)	The proportion for minimizing the integral value (The accumulated error). For control mode K0~K8.
	Integral time constant (T_I)	0~30,000 (ms)	For control mode K10
S_3+3:	Differential gain (K_D)	-30,000~30,000 (%)	The proportion for minimizing the derivative value (The rate of change of the process error). For control mode K0~K8
	Derivative time constant (T_D)	-30,000~30,000 (ms)	For control mode K10

Device No.	Function	Set-point range	Explanation
S_3+4:	Control method		0: Automatic control 1: Forward control (E = SV - PV). 2: Reverse control (E = PV - SV). 3: Auto-tuning of parameters exclusively for the temperature control. The device will automatically become K4 when the auto-tuning is completed and K_P, K_I and K_D are set with appropriate values (not avaliable in the 32-bit instruction). 4: Exclusively for the adjusted temperature control (not avaliable in the 32-bit instruction). 5: Automatic mode with MV upper/lower bound control. When MV reaches upper/lower bound, the accumulation of integral value stops. 7: Manual control 1: User set an MV. The accumulated integral value increases according to the error. It is suggested that the control mode should be used in a control environment which change more slowly. ELCM-PH/PA, ELC2-PB/PH/PA/PE/PV are supported. 8: Manual control 2: User set an MV. The accumulated integral value will stop increasing. When the control mode becomes the automatic mode (the control mode K5 is used), the instruction PID outputs an appropriate accumulated integral value according to the last MV. ELCM-PH/PA, ELC2-PB/PH/PA/PE/PV are supported. 10: T_I / T_D : The control changes the integra gain and the differential gain into integral time constant and differential time constant.

Device No.	Function	Set-point range	Explanation
S$_3$+5:	The range in which Error value (E) will not work	0~32,767	E = the error between SV and PV. If **S**$_3$ +5 is set as 5, when E is between -5 and 5, MV will be 0. When **S**$_3$ +5 = K0, the function will not be enabled.
S$_3$+6:	Upper limit of saturated output (MV)	-32,768 ~32,767	Ex: if **S**$_3$+6 is set as 1,000, MV will be 1,000 when it exceeds 1,000. **S**$_3$+6 has to be bigger or equal to **S**$_3$+7, otherwise the upper bound and lower bound values will switch.
S$_3$+7:	Lower limitof saturated output (MV)	-32,768 ~32,767	Ex: if **S**$_3$+7 is set as -1,000, MV will be -1,000 when it is smaller than -1,000..
S$_3$+8:	Upper limit of saturated integration	-32,768 ~32,767	Ex: if **S**$_3$+8 is set as 1,000, the integral value will be 1,000 when it is bigger than 1,000 and the integration will stop. **S**$_3$+8 has to be bigger or equal **S**$_3$ +9; otherwise the upper bound and lower bound values will switch
S$_3$+9:	Lower limit of saturated integration	-32,768 ~32,767	Ex: if **S**$_3$+9 is set as -1,000, the integral value will be -1,000 when it is smaller than -1,000 and the integration will stop.
S$_3$+10, 11:	Save accumulated integral value temporality	32-bit floating point range	The accumulated integral value is usually for reference. Users can clear or modify it (in 32-bit floating point) according to specific needs.
S$_3$ +12:	Save previous PV value temporality	-32,768~32, 767	The previous PV is usually for reference. Users can clear or modify it according to specific needs.

Device No.	Function	Set-point range	Explanation
S$_3$ +13: ~ **S**$_3$ +19:	For system use only.		

750. When a parameter setting exceeds its range, the upper / lower bound will be selected as the set value. However, if the control direction (DIR) exceeds the available range, it will be set to 0.

751. The PID instruction can be used in interrupt subroutines, step ladders and with the CJ instruction.
752. The maximum error of sampling time T_S = - (1 scan time + 1ms) ~ + (1 scan time). When the error affects the output, fix the scan time or execute the PID instruction in timed interrupt.
753. The PV parameter of the PID instruction must be stable before the PID operation executes.
754. For the 32-bit instruction, **S_3** occupies 21 registers. In the program example above, the area designated in **S_3** will be D100 ~ D120. Before execution of the PID instruction, users must move the parameters to the designated register area with a MOV instruction. If the designated registers are latched, use the MOVP instruction to transmit all parameters once. If **S_3**+1~3: are out of range, either the upper or lower limit value will be used.

755. Parameter table of 32-bit **S_3**:

Device No.	Function	Set-point range	Explanation
S_3 :	Sampling time (T_S)	1~2,000 (unit: 10ms)	Time interval between PID calculations and updates of MV. If T_S= 0, the PID instruction will not be enabled. If T_S is less than 1 program scan time, the PID instruction sets **S_3** to 1 program scan time, i.e. the value of T_S cannot be less than one program scan time.
S_3+1:	Proportional gain (K_P)	0~30,000(%)	The proportion for minimizing the error between SV and PV.
S_3+2:	Integration gain (K_I)	0~30,000(%)	The proportion for minimizing the integral value (The accumulated error). For control mode K0~K2, K5.
	Integral time constant (T_I)	0~30,000 (ms)	For control mode K10
S_3+3:	Derivative gain (K_D)	-30,000~30,000(%)	The proportion for minimizing the derivative value (The rate of change of the process error). For control mode K0~K2, K5.
	Derivative time constant (T_D)	-30,000~30,000 (ms)	For control mode K10
S_3+4:	Control mode	0: Automatic control 1: Forward control (E = SV - PV). 2: Reverse control (E = PV - SV). 5: Automatic mode with MV upper/lower bound control. When MV reaches upper/lower bound, the accumulation of integral value stops. 10: T_I / T_D mode with MV upper/lower bound control. When MV reaches upper/lower bound, the accumulation of integral value stops.	

Device No.	Function	Set-point range	Explanation
S_3+5, 6:	Tolerable range for error (E), 32-bit	0~ 2,147,483,647	E = the error between SV and PV. If S_3 +5 is set as 5, when E is between -5 and 5, MV will be 0. When S_3 +5 = K0, the function will not be enabled.
S_3+7, 8:	Upper bound of output value (MV) , 32-bit	-2,147,483,648~ 2,147,483,647	Ex: if S_3+6 is set as 1,000, MV will be 1,000 when it exceeds 1,000. S_3+6 has to be bigger or equal to S_3+7, otherwise the upper bound and lower bound value will switch
S_3+9, 10:	Lower bound of output value (MV) , 32-bit	-2,147,483,648~ 2,147,483,647	Ex: if S_3+7 is set as -1,000, MV will be -1,000 when it is smaller than -1,000.
S3+11, 12:	Upper bound of integral value, 32-bit	-2,147,483,648~ 2,147,483,647	Ex: if S3+8 is set as 1,000, the integral value will be 1,000 when it is bigger than 1,000 and the integration will stop. S3+8 has to be bigger or equal S3 +9; otherwise the upper bound and lower bound value will switch.
S_3+13, 14:	Lower bound of integral value, 32-bit	-2,147,483,648~ 2,147,483,647	Ex: if S_3+9 is set as -1,000, the integral value will be -1,000 when it is smaller than -1,000 and the integration will stop.
S_3+15, 16:	Accumulated integral value, 32-bit	Available range of 32-bit floating point	The accumulated integral value is usually for reference. Users can clear or modify it (in 32-bit floating point) according to specific needs.
S_3+17, 18:	The previous PV, 32-bit	-2,147,483,648~ 2,147,483,647	The previous PV is usually for reference. Users can clear or modify it according to specific needs.
S_3+19, 20	For system use only.		

756. The description of 32-bit **S**$_3$ and 16-bit **S**$_3$ are nearly the same. The difference is the capacity of **S**$_3$+5 ~ **S**$_3$+20.

PID Equations:

1. When control mode (**S**$_3$+4) is selected as K0, K1, K2 or K5:

   - In this control mode, the PID operation can be selected as Automatic, Forward, Reverse or Automatic with MV upper/lower bound control modes. Forward / Reverse direction is designated in **S**$_3$+4. Other relevant settings of the PID operation are set by the registers designated in **S**$_3$ ~ **S**$_3$+5.

   - PID equation for control mode k0~k2:

$$MV = K_P * E(t) + K_I * E(t)\frac{1}{S} + K_D * PV(t)S$$

where

MV : Output value

K_P : Proprotional gain

$E(t)$: Error value

$PV\ (t)$: Present measured value

$SV\ (t)$: Target value

K_D : Derivative gain

$PV(t)S$: Derivative value of PV(t)

K_I : Integral gain

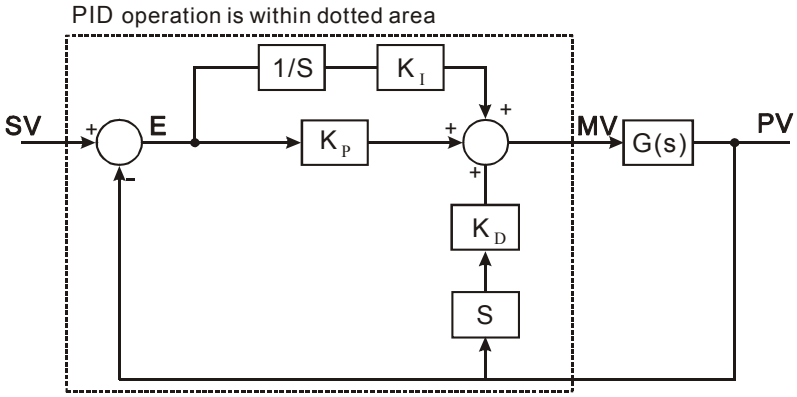
$E(t)\frac{1}{S}$: Integral value of E(t)

   - When E(t) is smaller than 0 as the control mode is selected as forward or inverse, E(t) will be regarded as “0"

Control mode	PID equation
Forward, automatic	E(t) = SV – PV
Inverse	E(t) = PV – SV

- Control diagram:

In diagram below, S is the derivative value, referenced to "(PV– previous PV) ÷ sampling time". 1 / S is integral value, referenced to "previous integral value + (error value × sampling time)". G(S) refers to the device being controlled.



- The equation above illustrates that this operation is different from a general PID operation in the way the derivative value is applied. To avoid the fault that the transient derivative value could be too big when a general PID instruction is first executed, our PID instruction monitors the derivative value of the PV. When the variation of PV is excessive, the instruction will reduce the output MV

2. When control mode (**S_3**+4) is selected as K3 and K4:

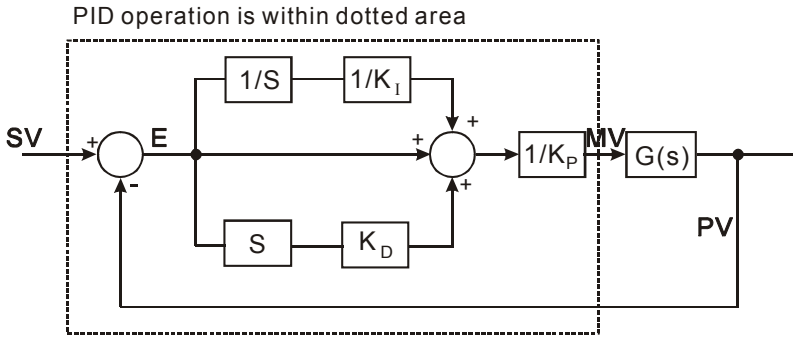
- The equation is exclusively for temperature control:

$$MV = \frac{1}{K_P}\left[E(t) + \frac{1}{K_I}\left(E(t)\frac{1}{S}\right) + K_D * E(t)S\right],$$

where $E(t) = SV(t) - PV(t)$

- Control diagram:

In diagram below, 1/KI and 1/KP refer to "divided by KI" and "divided by KP". Because this mode is exclusively for temperature control, users must use the PID instruction together with the GPWM instruction. See Example 3 below for more details.



- This equation is exclusively designed for temperature control. Therefore, when the sampling time (T_S) is set to 4 seconds (K400), the range of output value (MV) will be K0 ~ K4,000 and the cycle time of the GPWM instruction used together has to be set as 4 seconds (K4000) as well.
- Auto tuning is available by, selecting K3 (auto-tuning). After all the parameters are adjusted (the control direction will be automatically set as K4), parameters can be modified based on the adjusted results.

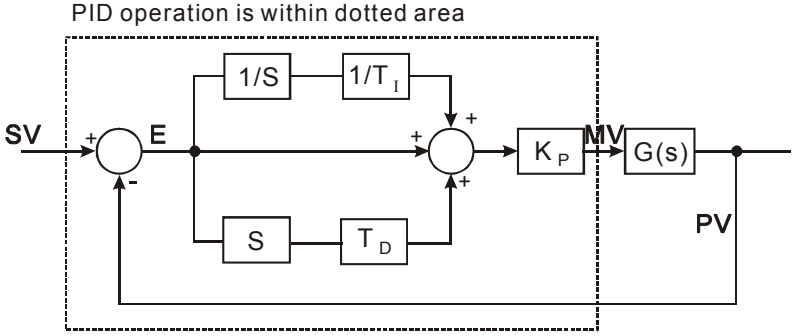
3. When control mode (**S_3**+4) is selected as K10:
   - **S_3**+2 (K_I) and **S_3**+3 (K_D) in this mode will be switched to parameter settings of Integral time constant (T_I) and Derivative time constant (T_D).
   - When the output value (MV) reaches the upper bound, the accumulated integral value will not increase. Also, when MV reaches the lower bound, the accumulated integral value will not decrease.
   - The equation for this mode will be:

$$MV = K_P \times \left[E(t) + \frac{1}{T_I} \int E(t)dt + T_D \frac{d}{dt} E(t) \right]$$

Where

$$E(t) = SV(t) - PV(t)$$

- Control diagram:



Notes and suggestion:

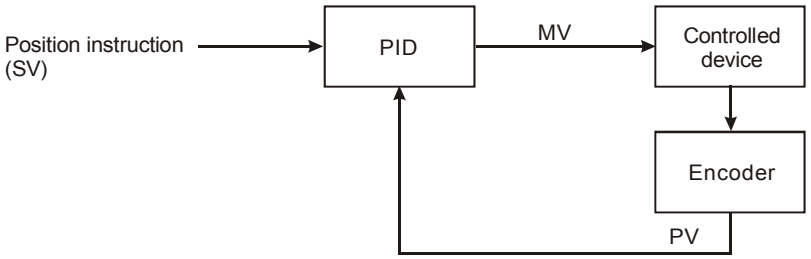
757. S_3 + 3 can only be in the range 0 ~ 30,000.

758. When the three main parameters are adjusted, K_P, K_I and K_D (S_3 + 4 = K0 ~ K2), adjust K_P first and reset K_I and K_D to 0. When the output can roughly be controlled, proceed to increase K_I and K_D (see example 4 below for adjustment methods). K_P = 100 refers to 100%, i.e. the proportional gain to the error is 1. K_P < 100% will decrease the error and K_P > 100% will increase the error

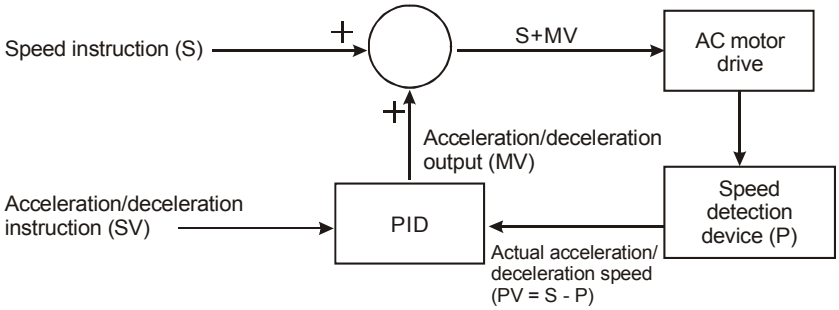
759. When the temperature auto-tuning function is selected(S_3 + 4 = K3, K4), it is suggested to store the parameters in D registers in the latched area to prevent the adjusted parameters from being cleared when the controller is powered down or when it is placed in program mode. There is no guarantee that the adjusted parameters are suitable for every control requirement. Users can modify the adjusted parameters according to specific needs, but it is suggested to modify only K_I or K_D.

760. The PID instruction has to be controlled with many parameters; therefore care should be taken when setting each parameter to prevent the PID operation from going out of control.

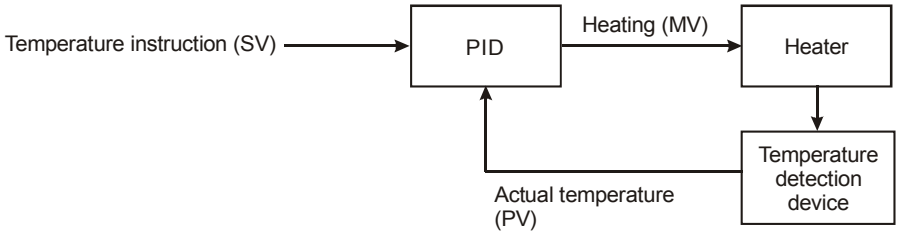
Example 1: Block diagram for positioning application (S_3+4 = 0)



Example 2: Block diagram for an AC motor drive application (S_3+4 = 0)



Example 3: Block diagram for a temperature control application (S_3+4 = 1)

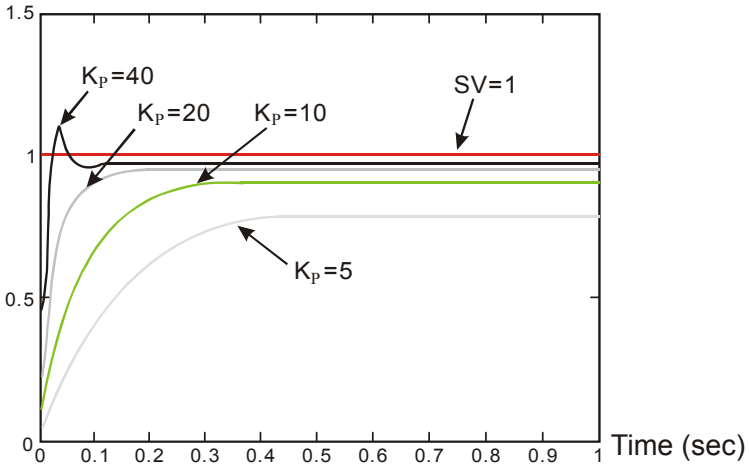


Example 4: Adjusting PID parameters

Assume that the transfer function of the controlled device G(S) in a control system is a first-order function $G(s)=\frac{b}{s+a}$ (model of general motors), SV = 1, and sampling time (T_S) = 10ms. Suggested steps for adjusting the parameters are as follows:

Step1:

Set K_I and K_D as 0, and K_P as 5, 10, 20, 40. Record the SV and PV respectively and the results are per the figure below.



Step 2:

When K_P is 40, response overshoot occurs, so we will not select it.

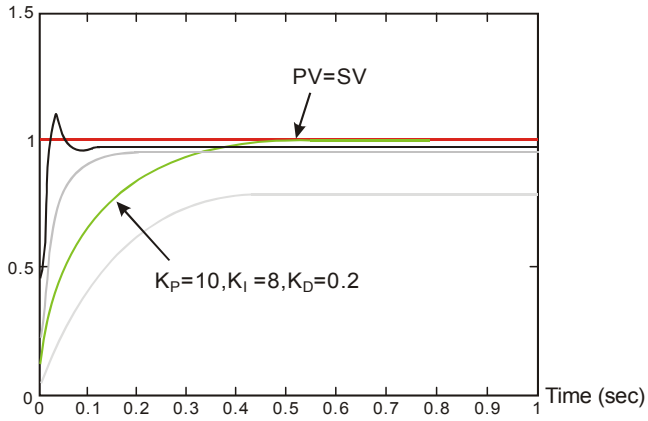
When K_P is 20, PV response is close to SV and won't overshoot, but transient MV will be too large due to a fast start-up. A better curve is needed.

When K_P is 10, PV response is close to SV and is smooth. We can consider using it.

When K_P is 5, the response is too slow. So we won't use it.

Step 3:

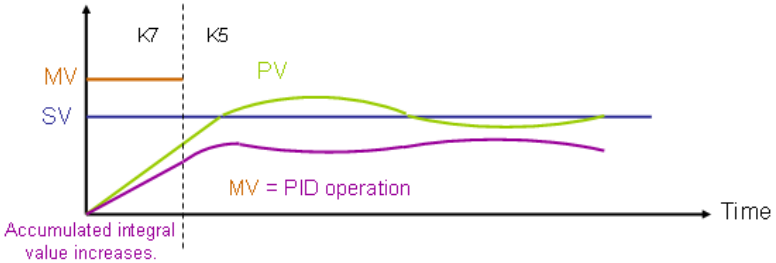
Select K_P = 10 and increase K_I gradually, e.g. 1, 2, 4, 8. K_I should not be bigger than K_P. Then, increase K_D as well, e.g. 0.01, 0.05, 0.1, 0.2. K_D should not exceed 10% of K_P. Finally we obtain the curves for PV and SV below.



Note: The example is only for reference. Users have to adjust parameters according to the condition of the actual control system.

Example 5: Transition between the manual mode (K7) and the automatic mode (K5)

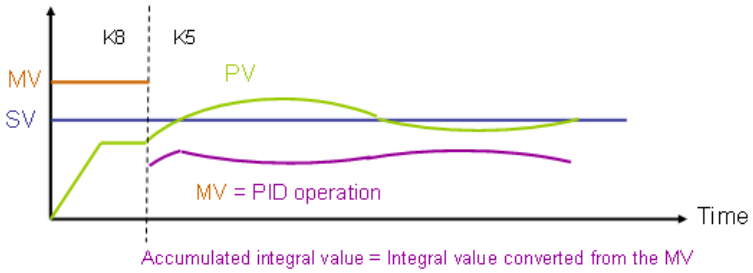
If the setting of the PID parameters is complete, and the control mode is the manual mode (K7), the control curve will be as shown below.



If the control mode becomes the automatic mode (K5), the output value MV changes from the output value set by users to the output value of the PID operation.

Example 6: Transition between the manual mode (K8) and the automatic mode (K5)

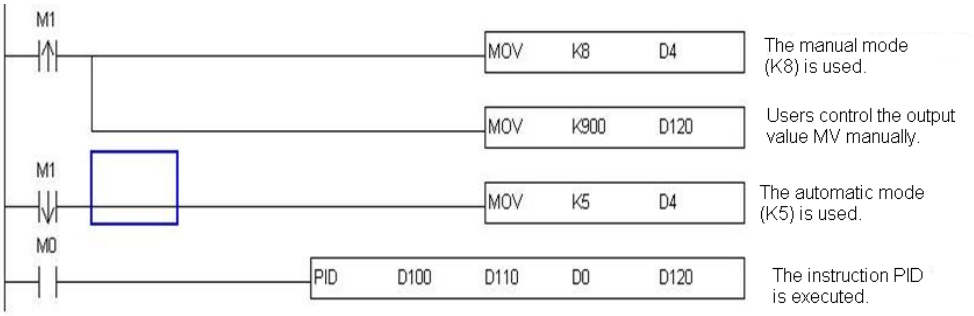
If the setting of the PID parameters is complete, and the control mode is the manual mode (K8), the control curve will be as shown below.



If the control mode becomes the automatic mode (K5), the accumulated integral value will be the integral value converted from the last MV, and the accumulated integral value will be converted into the output value of the PID operation.

The program for example 5 and program 6 are shown below. In the figure below, ,M0 is a flag for

enabling the instruction PID. When M1 is On, the manual mode is used. When M1 is Off, the automatic mode is used.



Application 1:

PID instruction in pressure control system. (Use the block diagram in example 1)

Control purpose:

Enabling the control system to reach the target pressure.

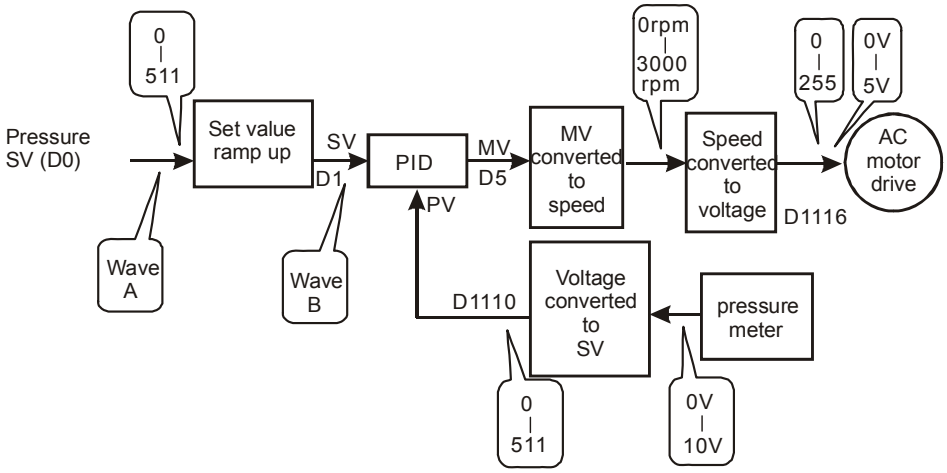
Control properties:

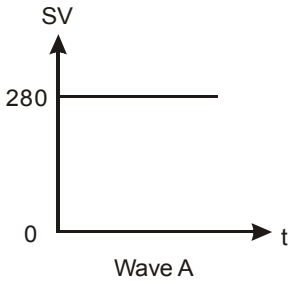
The system requires a gradual control. Therefore, the system will be overloaded or out of control if the process progresses too fast.

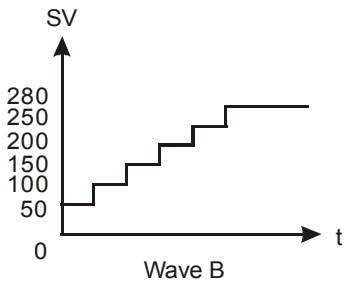
Suggested solution:

Solution 1: Longer sampling time

Solution 2: Using delay instruction. See the figure below





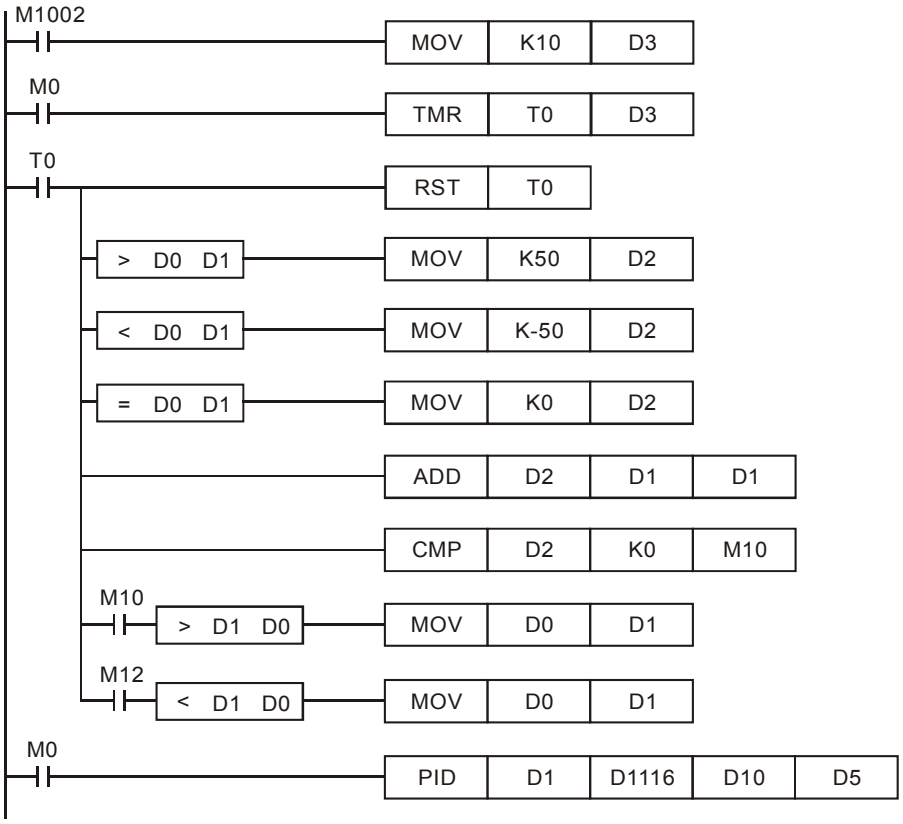


D2 stores increased value of each shift
D3 stores the time interval of each shift

Values in can modify D2 and D3 according to actual requirement

3

Example program of SV ramp up function:



Application 2:

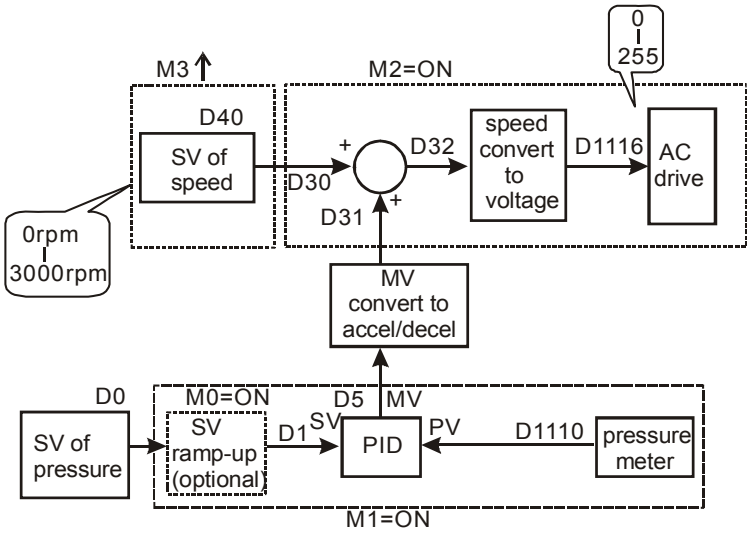
Speed control systems and pressure control systems operate separately (use the diagram of Example 2)

Control purpose:

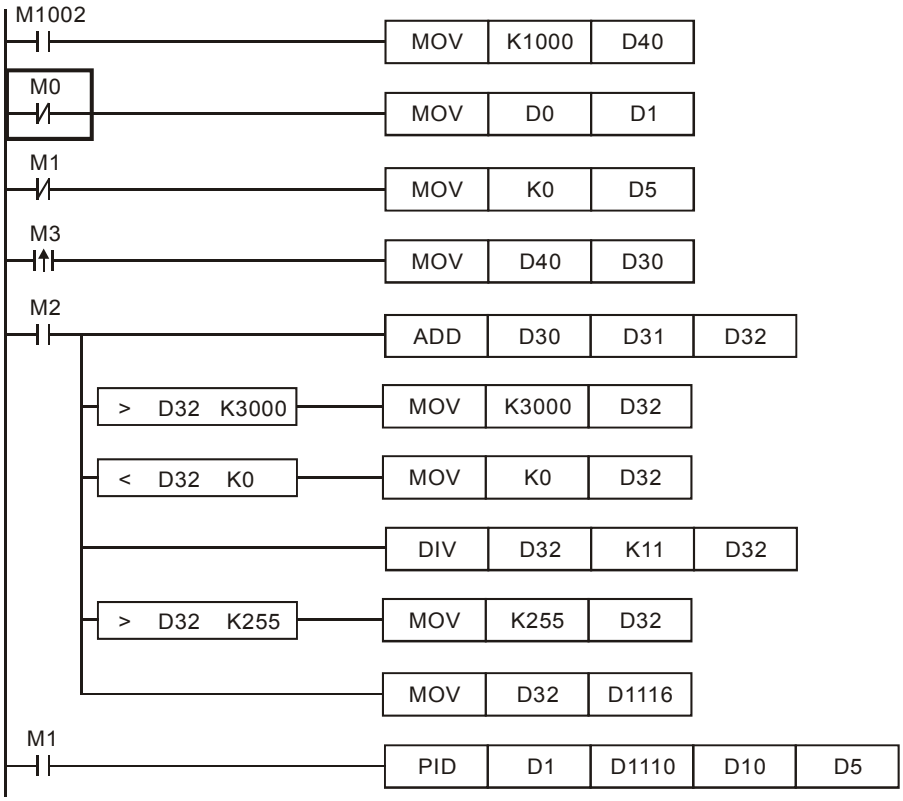
After the speed control operates in open loop for a period of time, add a pressure control system (PID instruction) to perform a close loop control.

Control properties:

Since the speed and pressure control systems are not interrelated, we have to structure an open loop for speed control first followed by a close loop pressure control. If there is concern that the pressure control system changes excessively, consider adding the SC ramp-up function illustrated in **Application 1** into this control. See the control diagram below.



Part of the example program:



Application 3:

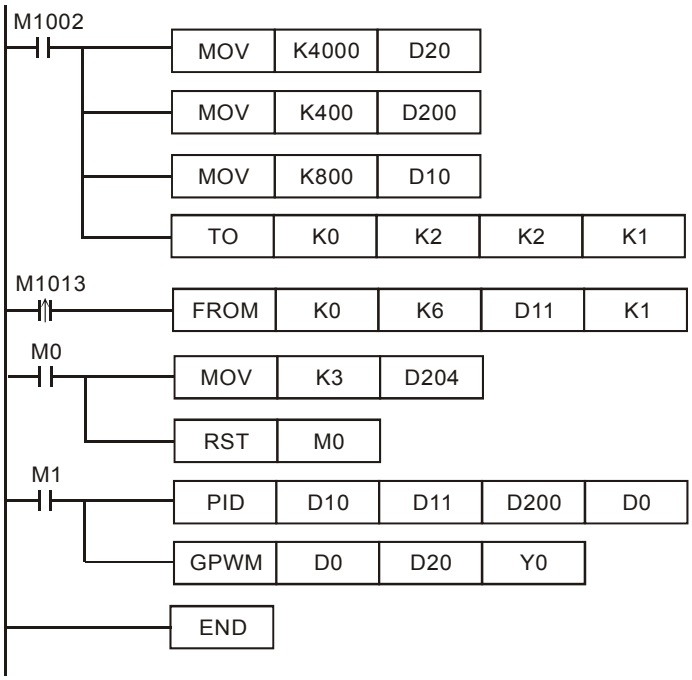
Using auto-tuning for temperature control

Control purpose:

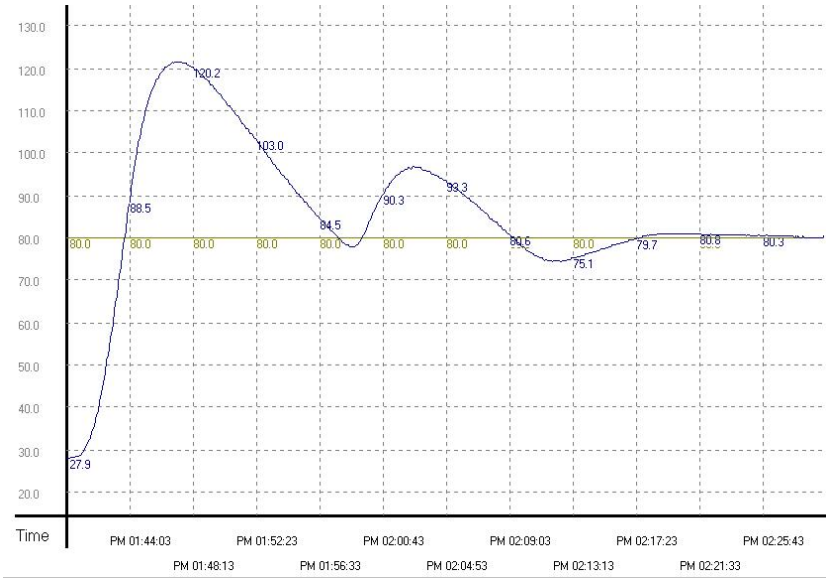
Calculating optimal parameters for the PID instruction for temperature control

Control properties:

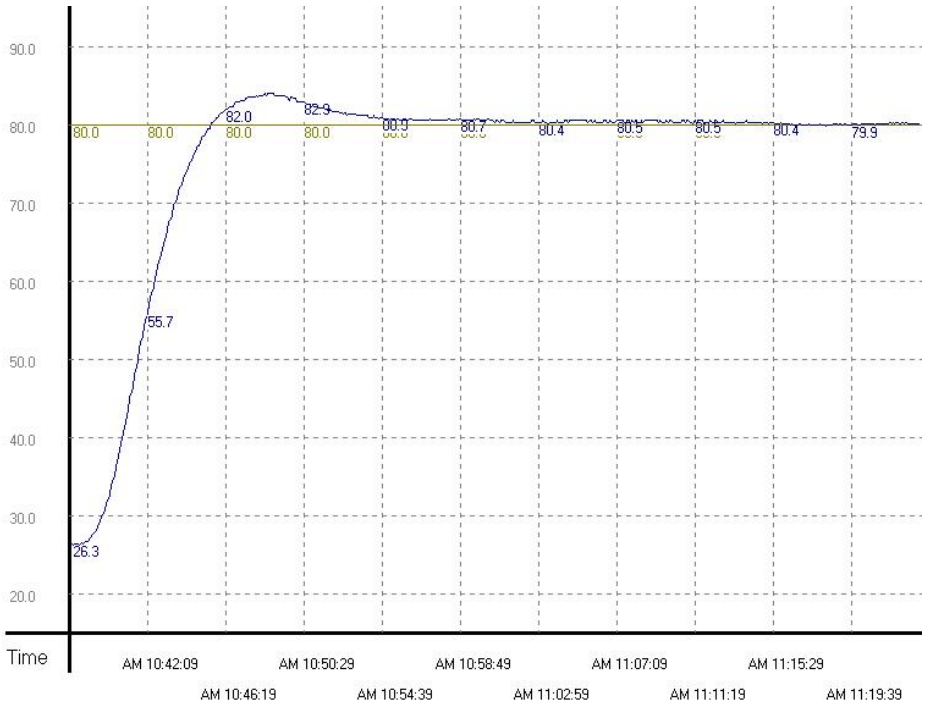
Users may not be familiar with a new temperature environment. In this case, selecting auto-tuning (S_3+4 = K3) for an initial adjustment is suggested. After initial tuning is complete, the instruction will auto modify the control mode to the mode exclusively for adjusted temperature (S_3+4 = K4). In this example, the control environment is an oven. See the example program below.



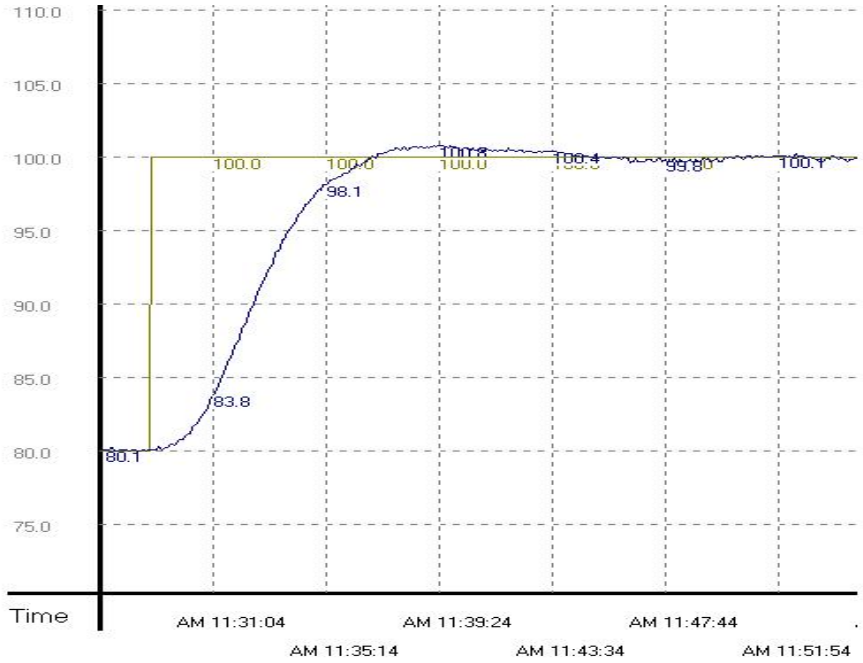
Results of initial auto-tuning



Results of using adjusted parameters generated by initial auto-tuning function.



From the figure above, we can see that the temperature control after auto-tuning is working fine and it used only approximately 20 minutes for the control. Next, we modify the target temperature from 80°C to 100°C and obtain the result below.



From the result above, we can see that when the parameter is 100°C, temperature control works fine and requires only 20 minutes which is the same as that in 80°C above.

API	Mnemonic	Operands	Function
89	PLS	**S**	Rising-edge Output

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S		*	*													PLS: 3 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

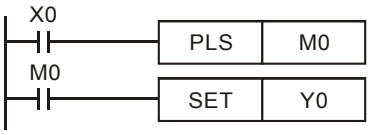
S: Rising edge pulse output

Description:

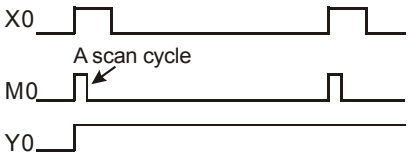
When X0=OFF→ON (rising-edge trigger), the PLS instruction will be executed and M0 will be On for one program scan..

Program Example:

Ladder Diagram:



Timing Diagram:



Instruction:		Operation:
LD	X0	; Load A contact of X0
PLS	**M0**	; M0 rising-edge output
LD	M0	; Load the contact A of M0
SET	Y0	; Y0 latched (ON)

API	Mnemonic	Operands	Function
90	LDP	**S**	Rising-edge Detection Operation

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*							*	*				LDP: 3 steps

ELCB			ELC						ELC2										ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Bit address

Description:

The LDP instruction is similar to the LD instruction, but LDP requires a false-to-true transition of **S** to be energized for one program scan. Then, the state of **S** must go false, then true again to be energized.

Program Example:

Ladder Diagram:

Instruction:		Operation:
LDP	**X0**	; Start X0 rising-edge detection
AND	X1	; Series connection A contact of X1
OUT	Y1	; Drive Y1 coil

Points to Note:

761. If a specific rising-edge contact state is ON before the ELC is powered, the rising-edge contact will be True after power is applied to the ELC
762. If X1 is ON and X0 transitions from OFF to ON, Y1 will be energized for one program scan, then it will turn OFF. Y1 will not turn on again until X0 goes OFF, then ON again. .

API	Mnemonic	Operands	Function
91	LDF	**S**	Falling-edge Detection Operation

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*							*	*				LDF: 3 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Bit address

Description:

The LDF instruction is similar to the LD and LDP instructions, but LDF requires a true-to-false transition of **S** to be energized for one program scan. Then, the state of **S** must go true, then false again to be energized.

Program Example:

Ladder Diagram:

Instruction:		Operation:
LDF	**X0**	; Start X0 falling-edge detection
AND	X1	; Series connection A contact of X1
OUT	Y1	; Drive Y1 coil

Points to Note:

1 If specific rising-edge contact state is ON before ELC is powered, the rising-edge contact will be True after power is applied to the ELC

2 If X1 is ON and X0 transitions from ON to OFF, Y1 will be energized for one program scan, then it will turn OFF. Y1 will not turn on again until X0 goes ON, then OFF again. .

API	Mnemonic	Operands	Function
92	ANDP	**S**	Rising-edge Series Connection

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*							*	*				ANDP: 3 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Bit address

Description:

The ANDP instruction is used to detect the rising edge of series contacts.

Program Example:

Ladder Diagram:

Instruction:		Operation:
LD	X0	; Load A contact of X0
ANDP	**X1**	; X1 rising-edge detection in series connection
OUT	Y1	; Drive Y1 coil

API	Mnemonic	Operands	Function
93	ANDF	**S**	Falling-edge Series Connection

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*							*	*				ANDF: 3 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

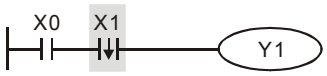
S: Bit address

Description:

The ANDF instruction is used to detect the falling edge of series contacts.

Program Example:

Ladder Diagram:



Instruction:		Operation:
LD	X0	; Load A contact of X0
ANDF	**X1**	; X1 falling-edge detection in series connection
OUT	Y1	; Drive Y1 coil

API	Mnemonic	Operands	Function
94	ORP	**S**	Rising-edge Parallel Connection

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*							*	*				ORP: 3 steps

ELCB			ELC						ELC2										ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

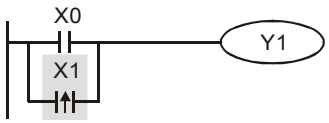
S: The parallel connection device that is detected switching from OFF to ON

Description:

The ORP instruction is used to detect the rising edge of a parallel contact.

Program Example:

Ladder Diagram:



Instruction:		Operation:
LD	X0	; Load A contact of X0
ORP	**X1**	; X1 rising-edge detection in parallel connection
OUT	Y1	; Drive Y1 coil

API	Mnemonic	Operands	Function
95	ORF	**S**	Falling-edge Parallel Connection

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*							*	*				ORF: 3 steps

ELCB			ELC						ELC2										ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

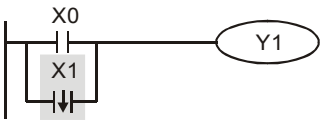
S: The parallel connection device that is detected switching from ON to OFF

Description:

The ORF instruction is used to detect the falling edge of a parallel contact.

Program Example:

Ladder Diagram:



Instruction:		Operation:
LD	X0	; Load A contact of X0
ORF	**X1**	; X1 falling-edge detection in parallel connection
OUT	Y1	; Drive Y1 coil

API	Mnemonic	Operands	Function
96	TMR	S_1, S_2	Timer

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*					TMR: 5 steps
S_2					*								*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

3

Operands:

S_1: Timer number (T0~T255) S_2: Set value (K0~K32,767)

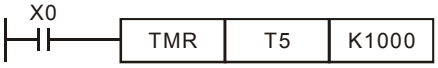
Description:

When the TMR instruction is executed, the timer is energized and will start timing. When the Set value of the timer is reached (accumulated value >= Set value), the timer done bit will be set. The timer done bit for any timer must use a bit instruction addressed with the timer number (T5 for example). The accumulative value of a timer must be used in a word type instruction and is also addressed with the timer number (T5 for example).

A timer's time base is determined by the timer number. Each type of ELC controller contains a different number of timers, so to determine the time base of a timer, refer to the memory maps for the various controllers located early in Chapter 2. For example, for all controllers timer T0 is a 100ms time base timer. This means that if the Set is K200, timer T0 will be a 20 second timer (200 x 100ms = 20 seconds). There are also 10ms and 1ms time base timers for most ELC controllers.

Program example:

Ladder Diagram:



Instruction:			Operation:
LD	X0		; Load A contact of X0
TMR	**T5**	**K1000**	; T5 timer Setting is K1000

API	Mnemonic	Operands	Function
97	CNT	S_1, S_2	Counter 16-bit

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1												*				CNT: 5 steps
S_2					*								*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

3

Operands:

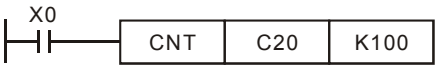
S_1: 16 bit counter number (C0~C199, ELCB-PB only C0~C127) **S_2**: Set value (K0~K32,767)

Description:

Each time the conditions preceding a CNT instruction transition from false-to-true, the counter will increment by one. When the accumulative value of a counter equals its Set value, the counter done bit will turn on. The done bit of a counter is the counter number (C2 for example) and must be used in a bit instruction. When the counter number is used in a word instruction, it is the accumulative value (the count value). If the conditions preceding a CNT instruction transition from false-to-true after the Set value has been reached, the done bit contact and the accumulative value will remain unchanged. To reset a counter, use the RST instruction.

Program example:

Ladder Diagram:



Instruction:			Operation:
LD	X0		; Load A contact of X0
CNT	**C20**	**K100**	; C20 counter Setting is K100

API	Mnemonic	Operands	Function
97	DCNT	S_1, S_2	Counter 32-bit

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1												*				DCNT: 9 steps
S_2					*								*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: 32 bit counter number S_2: Set value

Description:

763. DCNT is the startup instruction for the 32-bit counters C200 ~ C254(ELCB-PB: only C235~C254).
764. S_2 Set value: K-2,147,483,648~K2,147,483,647. (C243/C244 in ELCM-PH/PA: 0~K2,147,483,647).
765. The 32-bit counters are listed in the following table.

Model	General Counter	High speed Counter
ELCB-PB	-	C235~C254
ELC-PV/PA, ELC2-PV	C200~C234	C235~C254
ELCM-PH/PA	C200~C231	C232~C254
ELC2-PB/PH/PA/PE	C200~C232	C233~C254

766. General counters are 32-bit up/down counters. The present value will count up (add 1) or count down (subtract 1) according to the flags M1200~M1234 (setting count mode) when command DCNT is OFF->ON.

Model	General Counter	Flag (Count direction)
ELCB-PB	-	-
ELC-PV/PA, ELC2-PV	C200~C234	M1200~M1234
ELCM-PH/PA	C200~C231	M1200~M1231
ELC2-PB/PH/PA/PE	C200~C232	M1200~M1232

767. The count direction is determined by M1200-M1234. Each M-bit is for a specific counter number (C200-C234). When a particular counters direction bit is off, the counter will count up. When the bit is on, the counter will count down.
768. For information on high speed counters, refer to section 2.12.
769. When the DCNT instruction is OFF, the counter will stop counting, but the count values will not be cleared. Use the RST instruction to reset a counter. High-speed counters can use a specific external input to reset these counters.

Program Example:

Ladder Diagram:

Instruction:			Operation:
LD	M0		; Load A contact of M0
DCNT	**C254**	**K1000**	; C254 counter Setting is K1000

API	Mnemonic	Operands	Function
98	**INV**	-	Inverse Operation

OP	Description	Program Steps
N/A	Invert the current result of the internal ELC operations	INV: 1 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Description:

Invert the state of the conditions preceding the INV Instruction. If the conditions preceding the INV instruction are true, the output will be off. If the conditions preceding the INV instruction are false, the output will be on.

Program Example:

Ladder Diagram:

Instruction:		Operation:
LD	X0	; Load A contact of X0
INV		; Inverting the operation result
OUT	Y1	; Drive Y1 coil

In the example above, when X0 is on, output Y1 will be off. When X0 is off, output Y1 will be on.

API	Mnemonic	Operands	Function
99	PLF	**S**	Falling-edge Output

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S		*	*													PLF: 3 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

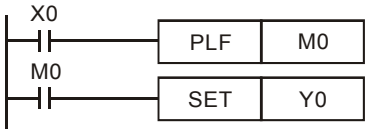
S: Output address

Description:

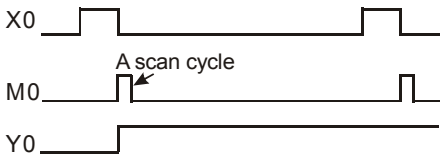
When X0= ON→OFF (falling-edge trigger), the PLF instruction will be executed and M0 will be on for one program scan..

Program Example:

Ladder Diagram:



Timing Diagram:



Command Code:		Operation:
LD	X0	; Load A contact of X0
PLF	**M0**	; M0 falling-edge output
LD	M0	; Load the contact A of M0
SET	Y0	; Y0 latched (ON)

API	Mnemonic	Operands	Function
100	MODRD	**S1, S2, n**	Modbus Data Read

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			MODRD: 7 steps
S_2					*	*							*			
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: External device's node address (K0~K254) S_2: Modbus Data address of the external device

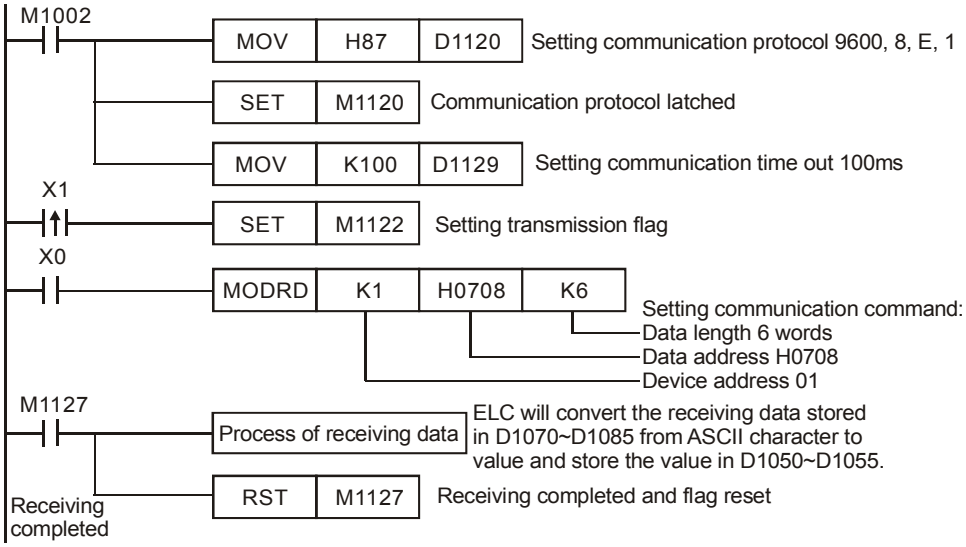
n: Number of registers to read (K1＜**n**≦K6)

Description:

770. MODRD instruction supports COM2 (RS-485).
771. MODRD is a Modbus read instruction for the MODBUS ASCII / RTU communication modes. The MODRD instruction can be used to read MODBUS data from external devices that support MODBUS.
772. If the Modbus data address S_2 is illegal for the external device, the device will respond with an error. The ELC will record the error code in D1130 and set bit M1141.
773. The response data from the external device will be stored in D1070 to D1085. After receiving the reply, the ELC verifies that the reply is correct. If there is an error, then M1140 = ON.
774. If ASCII mode is selected, the ELC will automatically convert the response data to hex and store it in D1050 to D1055. D1050 to D1055 will be invalid if using RTU mode for ELC-PV, ELC-PV2 controllers.
775. After M1140 or M1141 = ON, the command will be sent to the external device again. If the response is correct, then the flags M1140, M1141 will be reset.
776. There is no limit on the number of times this instruction can be used in the program, but only one instruction can be executed at a time on the same COM port.
777. Rising-edge contact (LDP, ANDP, ORP) and falling-edge contact (LDF, ANDF, ORF) can not be used with MODRD instruction.
778. Refer to the RS instruction for more information. The RS instruction also uses COM2.
779. ELCB-PB doesn't support the index registers E, F modification.

Program Example 1:

Communication between the ELC and MVX AC drives (ASCII Mode, M1143= OFF)



ELC → MVX, ELC transmits: "01 03 0708 0006 E7"

MVX → ELC , ELC receives: "01 03 0C 0100 1766 0000 0000 0136 0000 3B"

ELC transmit message

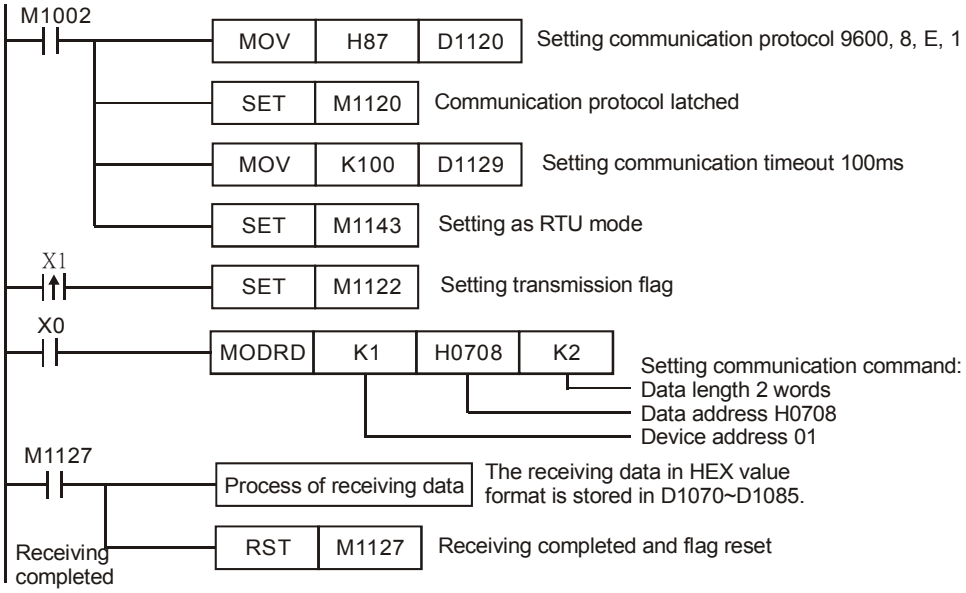
Register	Data		Descriptions	
D1089 low byte	'0'	30 H	ADR 1	ADR (1,0) is AC drive address
D1089 high byte	'1'	31 H	ADR 0	
D1090 low byte	'0'	30 H	CMD 1	CMD (1,0) is instruction code
D1090 high byte	'3'	33 H	CMD 0	
D1091 low byte	0'	30 H	Starting data address	
D1091 high byte	'7'	37 H		
D1092 low byte	'0'	30 H		
D1092 high byte	'8'	38 H		
D1093 low byte	'0'	30 H	Number of data (count by word)	
D1093 high byte	'0'	30 H		
D1094 low byte	'0'	30 H		
D1094 high byte	'6'	36 H		
D1095 low byte	'E'	45 H	LRC CHK 1	LRC CHK (0,1) is error check code
D1095 high byte	'7'	37 H	LRC CHK 0	

ELC receive message

Register	Data		Descriptions	
D1070 low	'0'	30 H	ADR 1	
D1070 high	'1'	31 H	ADR 0	
D1071 low byte	'0'	30 H	CMD 1	
D1071 high byte	'3'	33 H	CMD 0	
D1072 low byte	'0'	30 H	Number of data (count by byte)	
D1072 high byte	'C'	43 H		
D1073 low byte	'0'	30 H	Content of address 0708 H	ELC automatically converts ASCII codes to hex and stores the converted values in D1050 = 0100 H
D1073 high byte	'1'	31 H		
D1074 low byte	'0'	30 H		
D1074 high byte	'0'	30 H		
D1075 low byte	'1'	31 H	Content of address 0709 H	ELC automatically converts ASCII codes to hex and stores the converted values in D1051 = 1766 H
D1075 high byte	'7'	37 H		
D1076 low byte	'6'	36 H		
D1076 high byte	'6'	36 H		
D1077 low byte	'0'	30 H	Content of address 070A H	ELC automatically converts ASCII codes to hex and stores the converted values in D1052 = 0000 H
D1077 high byte	'0'	30 H		
D1078 low byte	'0'	30 H		
D1078 high byte	'0'	30 H		
D1079 low byte	'0'	30 H	Content of address 070B H	ELC automatically converts ASCII codes to hex and stores the converted values in D1053 = 0000 H
D1079 high byte	'0'	30 H		
D1080 low byte	'0'	30 H		
D1080 high byte	'0'	30 H		
D1081 low byte	'0'	30 H	Content of address 070CH	ELC automatically converts ASCII codes to hex and stores the converted values in D1054 = 0136 H
D1081 high byte	'1'	31 H		
D1082 low byte	'3'	33 H		
D1082 high byte	'6'	36 H		
D1083 low byte	'0'	30 H	Content of address 070D H	ELC automatically converts ASCII codes to hex and stores the converted values in D1055 = 0000 H
D1083 high byte	'0'	30 H		
D1084 low byte	'0'	30 H		
D1084 high byte	'0'	30 H		
D1085 low byte	'3'	33 H	LRC CHK 1	
D1085 high byte	'B'	42 H	LRC CHK 0	

Program Example 2:

Communication between ELC and MVX AC drive (RTU Mode, M1143= ON)



ELC → MVX, ELC transmits: 01 03 0708 0002 44 BD

MVX → ELC, ELC receives: 01 03 04 1770 0000 FE 5C

ELC transmit message

Register	Data	Descriptions
D1089 low byte	01 H	Address
D1090 low byte	03 H	Function
D1091 low byte	07 H	Starting data address
D1092 low byte	08 H	
D1093 low byte	00 H	Number of data (count by word)
D1094 low byte	02 H	
D1095 low byte	44 H	CRC CHK Low
D1096 low byte	BD H	CRC CHK High

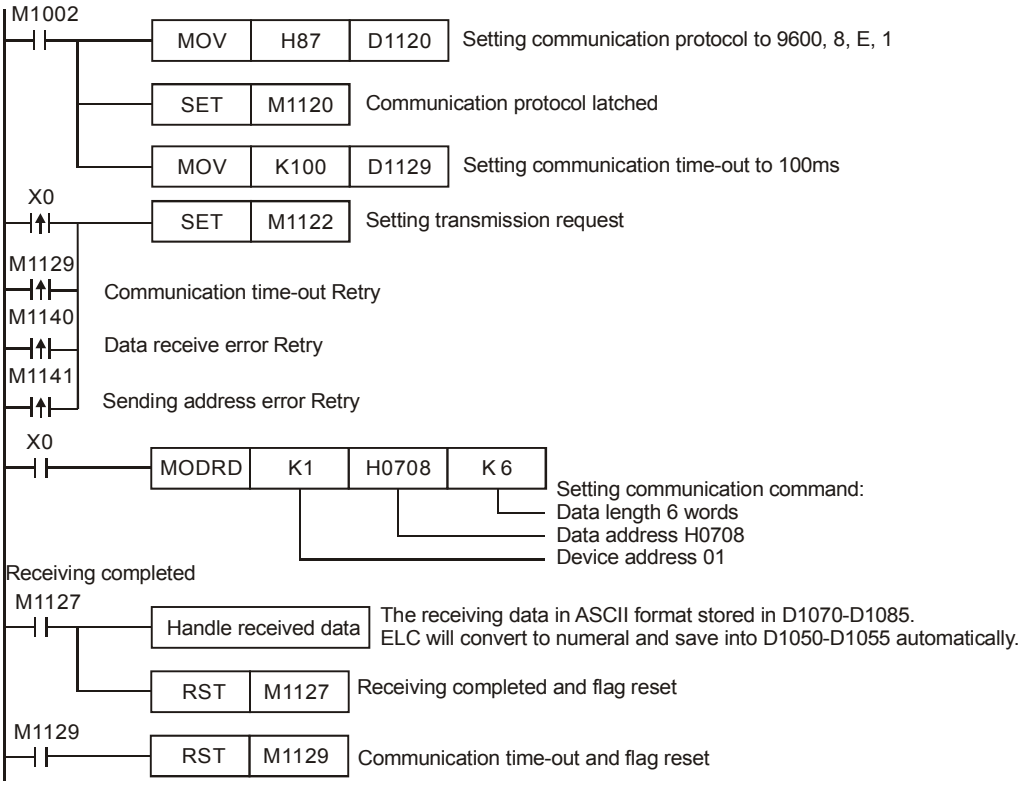
ELC receive message

Register	Data	Descriptions
D1070 low byte	01 H	Address
D1071 low byte	03 H	Function
D1072 low byte	04 H	Number of data (count by byte)
D1073 low byte	17 H	Content of address 2102 H
D1074 low byte	70 H	
D1075 low byte	00 H	Content of address 2103 H

D1076 low byte	00 H	
Register	**Data**	**Descriptions**
D1077 low byte	FE H	CRC CHK Low
D1078 low byte	5C H	CRC CHK High

Program Example 3:

780. The ELC is connected to an MVX AC drive (ASCII Mode, M1143= OFF). If a message times-out occurs, retry the message.
781. When X0= ON, read data from address H0708 of device 01 (MVX) and save the reply data in D1070~D1085 in ASCII format. The ELC will auto convert its contents to hex and save it in D1050~D1055.
782. If flag M1129 = ON, the previous message timed out.
783. If flag M1140 = ON, an error was received for the previous message
784. If flag M1141 = ON, an invalid Modbus data address occurred.
785. When any of the above flags are set, the message was not delivered successfully. The program can clear the flag and retransmit the message. This is demonstrated in the example program below.



API	Mnemonic	Operands	Function
101	MODWR	**S1, S2, n**	Modbus Data Write

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			MODWR: 7 steps
S_2					*	*							*			
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

3

Operands:

S_1: External device's node address (K0~K254) **S_2**: Modbus data address in external device

n: Data to be written (ELC address or constant value)

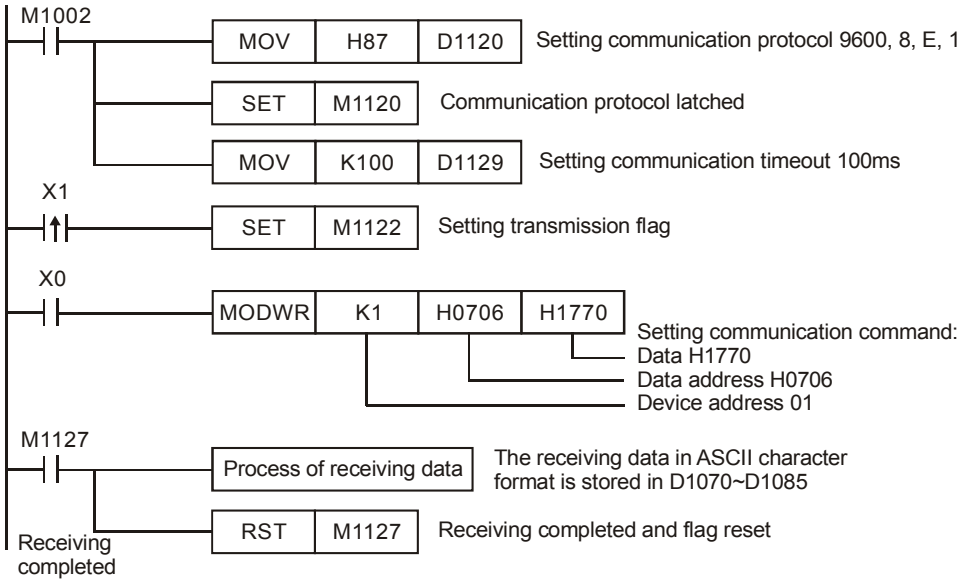
Description:

786. MODWR instruction supports COM2 (RS-485).
787. MODWR is a specific instruction for the MODBUS ASCII / RTU mode communications. The MODWR instruction can be used to write MODBUS data to external devices that support MODBUS communications.
788. If the address of **S_2** is illegal for the designated communication device, the device will respond with an error and the ELC will record the error code in D1130 and M1141 will be ON. For example, 8000H is an illegal register address in the MVX drive, resulting in M1141 = ON, D1130=2. The error code is generated by the external device. To determine the cause of the error, the user manual for the external device will need to be referenced. In this case the MVX series user manual would need to be referenced.
789. The response from the external device will be stored in D1070 to D1076. After receiving the response, the ELC will check if there are any data errors. If there is an error, then M1140 = ON.
790. After M1140 or M1141 = ON, resend the correct data to the external device again. If the response is correct, then the flags M1140, M1141 will be cleared.
791. There is no limit on the number of times this instruction can be used in the program, but only one instruction can be executed at a time for the same COM port.
792. If rising-edge contacts (LDP, ANDP, ORP) or falling-edge contacts (LDF, ANDF, ORF) are used as conditions for the MODWR instruction, energizing flag M1122 is required to initiate the message request.
793. Refer to the RS instruction for more information. The RS instruction also uses COM2.
794. ELCB-PB doesn't support index registers E, F modification.

Program Example 1:

Communication between the ELC and MVX AC drives (ASCII Mode, M1143= OFF)

Program diagram



ELC → MVX, ELC transmits: "01 06 0706 1770 65 "

MVX → ELC, ELC receives: "01 06 0706 1770 65 "

ELC transmit message

Register	Data		Descriptions	
D1089 low	'0'	30 H	ADR 1	ADR (1,0) is AC drive address
D1089 high	'1'	31 H	ADR 0	
D1090 low	'0'	30 H	CMD 1	CMD (1,0) is instruction code
D1090 high	'6'	36 H	CMD 0	
D1091 low	'0'	30 H	Data address	
D1091 high	'7'	37 H		
D1092 low	'0'	30 H		
D1092 high	'6'	36 H		
D1093 low	'1'	31 H	Data contents	
D1093 high	'7'	37 H		
D1094 low	'7'	37 H		
D1094 high	'0'	30 H		
D1095 low	'6'	36 H	LRC CHK 1	LRC CHK (0,1) is error check code
D1095 high	'5'	35 H	LRC CHK 0	

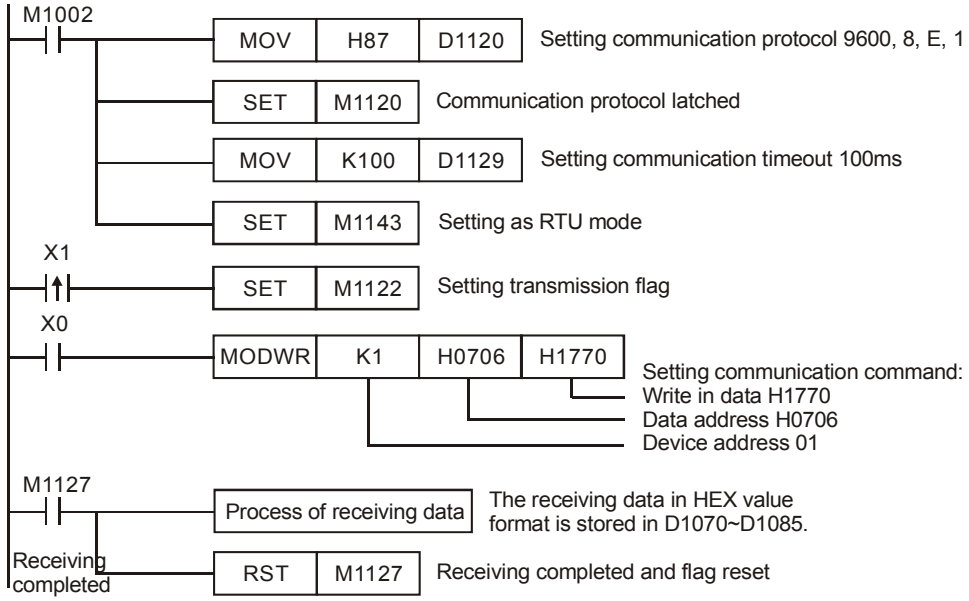
ELC receive message

Register	Data		Descriptions
D1070 low	'0'	30 H	ADR 1
D1070 high	'1'	31 H	ADR 0
D1071 low	'0'	30 H	CMD 1
D1071 high	'6'	36 H	CMD 0
D1072 low	'0'	30 H	Data address
D1072 high	'7'	37 H	
D1073 low	'0'	30 H	
D1073 high	'6'	36 H	
D1074 low	'1'	31 H	Data content
D1074 high	'7'	37 H	
D1075 low	'7'	37 H	
D1075 high	'0'	30 H	
D1076 low	'6'	36 H	LRC CHK 1
D1076 high	'5'	35 H	LRC CHK 0

Program Example 2:

Communication between the ELC and MVX AC drives (RTU Mode, M1143= ON)

Program diagram



ELC → MVX, ELC transmits: 01 06 0706 1770 66 AB

MVX → ELC, ELC receives: 01 06 0706 1770 66 AB

ELC transmit message

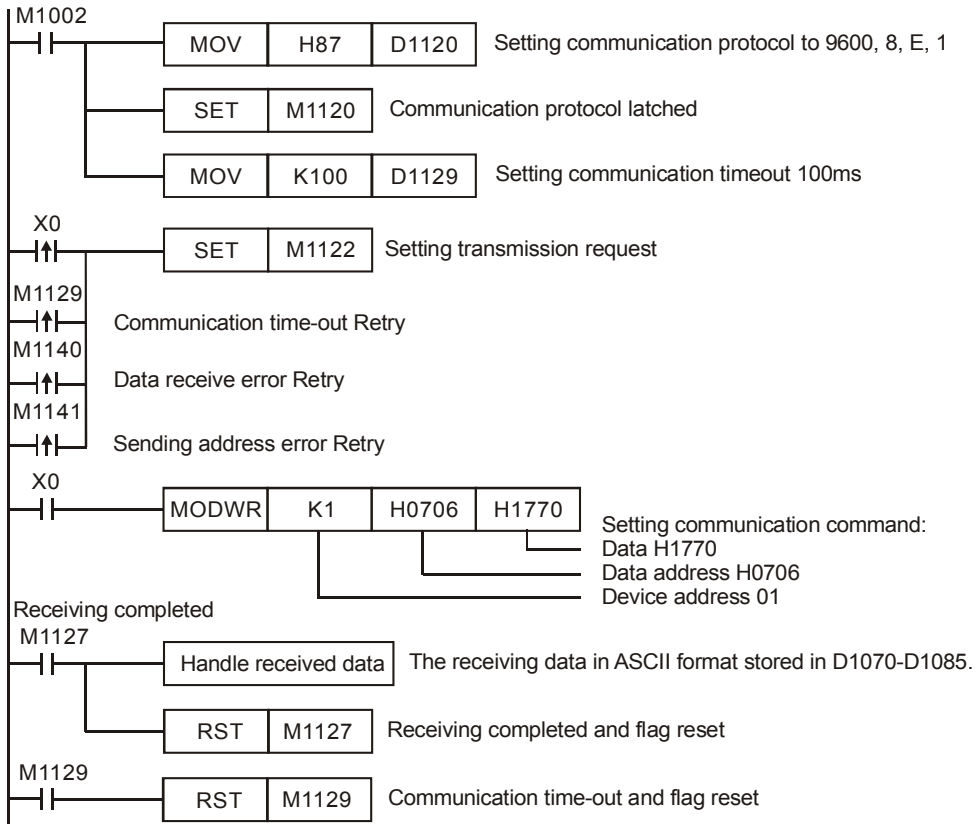
Register	Data	Descriptions
D1089 low	01 H	Address
D1090 low	06 H	Function
D1091 low	07 H	Data address
D1092 low	06 H	
D1093 low	17 H	Data content
D1094 low	70 H	
D1095 low	66 H	CRC CHK Low
D1096 low	AB H	CRC CHK High

ELC receive message

Register	Data	Descriptions
D1070 low	01 H	Address
D1071 low	06 H	Function
D1072 low	07 H	Data address
D1073 low	06 H	
D1074 low	17 H	Data content
D1075 low	70 H	
D1076 low	66 H	CRC CHK Low
D1077 low	AB H	CRC CHK High

Program Example 3:

795. The ELC connects to an MVX AC drive (ASCII Mode, M1143= OFF). When communication times-out, retry when the error occurs.
796. When X0= ON, the ELC will write data H1770(K6000) into address H0706 of device 01 (MVX).
797. Flag M1129 = ON when communication times-out and the program will send a request from M1129 to re-energize M1122 to resend the message.
798. Flag M1140 = ON when response is a receive error, the program will send a request from M1140 and re-energize M1122 to resend the message.
799. Flag M1141 = ON when the response is a received address error, the program will send a request from M1141 and re-energize M1122 to resend the message.



API	Mnemonic		Operands	Function
107	LRC	P	S, n, D	LRC Generator

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			LRC, LRCP: 7 steps
n					*	*							*			
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

3

Operands:

S: Starting address for the checksum operation (ASCII mode) **n**: Number of values to perform the operation on (**n**=K1~K256) **D**: Destination for storing the operation result

Description:

800. **n**: **n** must be even. If **n** is out of range, an error will occur and the instruction will not be executed. If an error occurs, M1067 and M1068 = ON and error code H'0E1A will be recorded in D1067.
801. 16-bit conversion mode: When M1161= OFF, hexadecimal data that starts from the source **S** will be divided into upper 8-bit and lower 8-bit values and the checksum operation will be performed on **n** values. Then, the result will be stored in the upper and lower 8-bits of **D**.
802. 8-bit conversion mode: When M1161= ON, divide the hexadecimal data that starts with source address **S** into upper 8-bit (invalid data) and lower 8-bit and perform the checksum operation on **n** values. Then, store the result in the lower 8-bits of **D** (upper 8-bits of **D** will be zero (0)).
803. Flag: M1161 8/16-bit mode

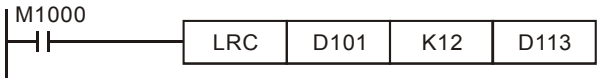
Program Example:

ASCII communication mode: Data stored as following:

Register	Data		Descriptions	
D100 low byte	': '	3A H	STX	
D101 low byte	'0'	30 H	ADR 1	ADR (1,0) is AC drive address
D102 low byte	'1'	31 H	ADR 0	
D103 low byte	'0'	30 H	CMD 1	CMD (1,0) is instruction code
D104 low byte	'3'	33 H	CMD 0	
D105 low byte	'0'	30 H	Starting data address	
D106 low byte	'7'	37 H		
D107 low byte	'0'	30 H		
D108 low byte	'8'	38 H		
D109 low byte	'0'	30 H	Number of data (count by word)	

Register	Data		Descriptions	
D110 low byte	'0'	30 H		
D111 low byte	'0'	30 H		
D112 low byte	'6'	36 H		
D113 low byte	'E'	45 H	LRC CHK 0	LRC CHK (0,1) error check code
D114 low byte	'7'	37 H	LRC CHK 1	
D115 low byte	CR	D H	END	
D116 low byte	LF	A H		

The LRC CHK (0,1) above is error check code and it can be calculated with the LRC instruction (8-bit Mode, M1161= ON).



LRC check: 01 H + 03 H + 07 H + 08 H + 00 H + 06 H = 19 H, then take the 2's complement, E7 H. At that time, 'E'(45 H) is stored in the lower 8-bits of D113 and '7' (37 H) is stored in the lower 8-bits of D114.

API	Mnemonic		Operands	Function
108	CRC	P	S, n, D	CRC Generator

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			CRC, CRCP: 7 steps
n					*	*							*			
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Starting address for the checksum operation (RTU mode) **n**: Number of values to perform the operation on (**n**=K1~K256) **D**: Destination for storing the operation result

Description:

804. If **n** is out of range, an error will occurred and the instruction will not be executed. If an error occurs, M1067 and M1068 = ON and error code H'0E1A will be recorded in D1067.
805. 8-bit conversion mode: When M1161 = ON, divide the hexadecimal data that starts with source **S** into the high byte (invalid data) and low byte and have the CRC operation performed on **n** values and store the result in the low byte of **D** (upper 8-bit of **D** will be zero)
806. 16-bit conversion mode: W hen M1161 = OFF, divide hexadecimal data that starts with source **S** into the high byte and low byte and have the CRC operation performed on **n** values and store the result in the low byte and high byte of **D**

Program Example:

RTU communication mode: Data stored as following:

Register	Data	Descriptions
D100 low byte	01 H	Address
D101 low byte	06 H	Function
D102 low byte	07 H	Data address
D103 low byte	06 H	
D104 low byte	17 H	Data content
D105 low byte	70 H	
D106 low byte	66 H	CRC CHK 0
D107 low byte	AB H	CRC CHK 1

The CRC CHK (0,1) above is the error check code and it can be calculated using the CRC instruction (8-bit Mode, M1161= ON).

CRC check: 66 H is stored in the lower 8-bits of D106 and AB H is stored in the lower 8-bits of D107.

API	Mnemonic			Operands	Function
110	D	ECMP	P	S_1, S_2, D	Floating Point Compare

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DECMP, DECMPP: 13 steps
S_2					*	*							*			
D		*	*	*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

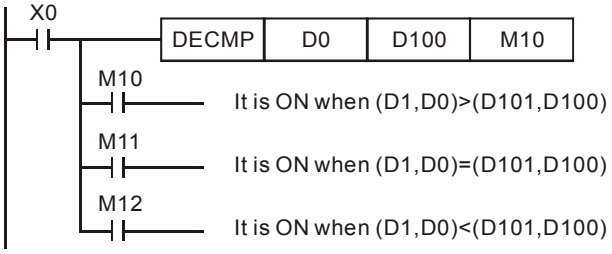
S_1: 1st comparison value **S_2**: 2nd comparison value **D**: Destination result, 3 consecutive bit addresses

Description:

807. The data of **S_1** is compared to the data of **S_2** and the result (>, =, <) is displayed via three bit addresses beginning with **D**.
808. If the source operand **S_1** or **S_2** is a constant K or H, the integer value will automatically be converted to binary floating point to compare.

Program Example:

809. If M10 is entered for **D**, M10~M12 will automatically be used.
810. When X0= ON, one of M10~M12 = ON. When X0= OFF, DECMP is not executed, M10~M12 will retain their previous state before X0= OFF.
811. M10~M12 are used to display the result of the DECMP.
812. Use RST or ZRST instruction to reset the result.



API	Mnemonic	Operands	Function
111	D EZCP P	S_1, S_2, S, D	Floating Point Zone Compare

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DEZCP, DEZCPP: 17 steps
S_2					*	*							*			
S					*	*							*			
D		*	*	*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Lower limit of zone comparison S_2: Upper limit of zone comparison **S**: Comparison value

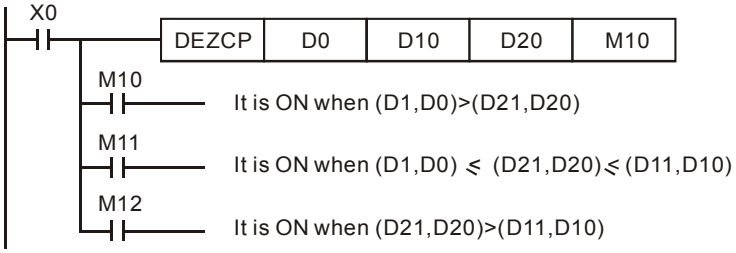
D: Comparison result, 3 consecutive bit addresses

Description:

813. The contents of **S** is compared to the data range of S_1~ S_2 and the result (>, =, <) is displayed by three bit addresses beginning with **D**.
814. If the source operand S_1 or S_2 is a constant K or H, the integer value will automatically be converted to binary floating point for the compare.
815. Operand S_1 should be smaller than operand S_2, when $S_1 > S_2$, S_1 will be used as upper and lower limit for the comparison.

Program Example:

816. If M10 is entered for **D**, M10~M12 will automatically be used.
817. When X0= ON, one of M10~M12 = ON. When X0= OFF, DEZCP instruction is not executed, M10~M12 will retain their previous state before X0= OFF.
818. Use RST or ZRST instruction to reset the result.



API	Mnemonic			Operands	Function
112	D	MOVR	P	**S, D**	Floating Point Data Move

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S																DMOVR, DMOVRP: 9 steps
D								*	*	*	*	*	*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Data source of the floating point value **D**: Destination of the move

Description:

819. Directly enter a floating point value in **S**.

820. When instruction is enabled, the contents of **S is moved** to **D**.

Program Example:

When X0 = OFF, D10 and D11 will not change. When X0 = ON, move F1.200E+0 (Input F1.2, and scientific notation F1.200E+0 to D11/D10. In ELCSoft, under “Edit Monitored Devices” enter D10 and D11 and view the floating point data value for those addresses in the floating point column. ..

Remarks:

DMOVR/DMOVRP supports ELCB-PB V1.2 and ELC-PA V1.2 above.

API	Mnemonic	Operands	Function
113	ETHRW	**S_1, S_2, D, n**	Ethernet communication

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			ETHRW: 9 steps
S_2					*	*							*			
D													*			
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: IP address, communication port number, and read/write mode **S_2**: Device address **D**: Source/Destination data register **n**: Data length (Unit: Word; Range: K1~K96)

Description:

1. **S_1**: IP address, communication port number, and read/write mode

   The operand **S_1** occupies five consecutive data registers. The functions are as follows.

   - IP address: Two data registers are occupied, that is, **S_1**+0 and **S_1**+1.

     IP address→IP3.IP2.IP1.IP0→192.168.0.2

     If **S_1** is D100, the values in D100 and D101 are H'0002 and H'C0A8 respectively.

D100 (**S_1**+0)		D101 (**S_1**+1)	
High	Low	High	Low
IP1	IP0	IP3	IP2
0	2	192	168
H'0002		H'C0A8	

   - **S_1**+2: Communication port number

     The communication port number(k108) of the Ethernet port on ELC2-PE. The communication ports on the left-side Ethernet modules connected to a CPU module are numbered according to their distances from the CPU module. The numbers start from K100 to K107.

   - **S_1**+3: Station address of a slave
   - **S_1**+4: Read/Write mode

     The definition is the same as Modbus. The function codes supported are H'03, H'04, H'06, and H'10.

2. **S_2**: Device address

   The definition is the same as Modbus.

3. The operand **D** specifies a source data register or a destination data register.

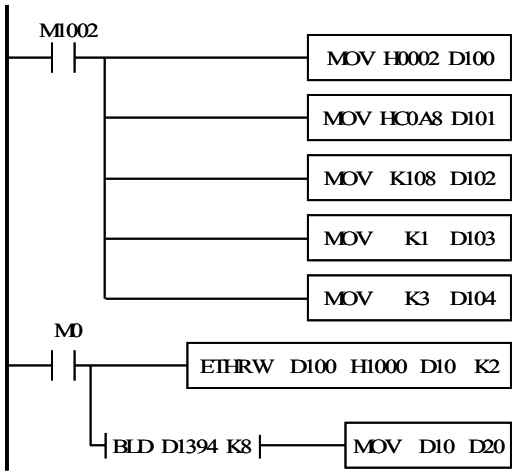
4. **n**: Length of data (Unit: word)

   The setting range is K1~K96. If **n** exceeds the range, it will be taken as the maximum value or the minimum value.
5. Whenever the instruction is executed, the communication command is sent. Users do not need to enable a special flag to send the communication command.
6. The instruction can be used several times. However, if an ETHRW instruction specifies a module, other ETHRW instructions can not send communication commands to the module. The next communication command can not be sent until the reception is complete or the module replies that an error occurs.
7. If a communication command is being received, the reception stops when the execution of the instruction stops. Besides, the flag related to the command's having being received and the error flag are not ON.
8. The communication timeout is stored in D1349. The default timeout is 3000 milliseconds. The range of digital values is 1~32767. If the communication timeout exceeds the range, it will be taken as 3000 milliseconds.
9. The values of bit0~bit8 in D1395 indicate which communication port has received a command. For example, if the communication port built in ELC2-PE has received a command, "BLD D1395 K8" is satisfied.
10. The values of bit0~bit8 in D1396 indicate which module For example, if a reception error occur in the first left-side EN01-SL, "BLD D1396 K0" is satisfied.
11. When the instruction is executed, user can not use the online editing function. Otherwise, the data received will not be stored correctly.

Program Example:

(The command is sent and received through the Ethernet port built in ELC2-PE.)

The IP address stored in D100 and D101 is 192.168.0.2, the communication port number stored in D102 is K108, the station address stored in D103 is K1, and the function code stored in D104 is H'03. The device address is H'1000, and two pieces of data are read. When M0 is ON, ETHRW is executed. After the reception of the communication command is complete, bit8 in D1394 is ON. The data received is stored in D10 and D11.



API	Mnemonic			Operands	Function
116	D	RAD	P	S, D	Degree → Radian

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DRAD, DRADP: 9 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: source (degrees) **D**: Destination result (radians)

Description:

821. This instruction uses the following formula to convert degrees to radians:

Radians = degrees × (π/180)

822. Flags: M1020 Zero flag, M1021 Borrow flag, M1022 Carry flag

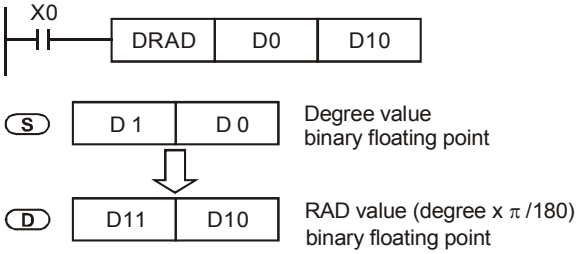
If the absolute value of the result is larger than the maximum floating point value, the carry flag M1022= ON.

If absolute value of the results is less than the minimum floating point value, the borrow flag M1021= ON.

If the conversion result is 0, zero flag M1020= ON.

Program Example:

When X0= ON, convert the degrees value of the binary floating point value in D1/D0 to radians and save the result in D11/D10. The result is binary floating point.



API	Mnemonic			Operands	Function
117	D	DEG	P	S, D	Radian → Degree

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DDEG, DDEGP: 9 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: data source (radians) **D**: Destination result (degrees)

DESCRIPTION:

823. This instruction uses the following formula to convert radians to degrees:

Degrees = Radians × (180/π)

824. Flags: M1020 Zero flag, M1021 Borrow flag and M1022 Carry flag.

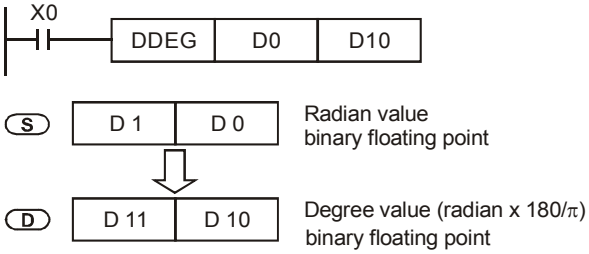
If the absolute value of the result is larger than the maximum floating point value, the carry flag M1022= ON.

If absolute value of the results is less than the minimum floating point value, the borrow flag M1021= ON.

If the conversion result is 0, zero flag M1020= ON.

Program Example:

When X0= ON, convert the degrees value of the binary floating point in D1/D0 to radians and save the result in D11/D10 The result is binary floating point.



API	Mnemonic	Operands	Function
118	D EBCD P	S, D	Floating to Scientific Conversion

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			DEBCD, DEBCDP: 9 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Data source **D**: Destination result

Description:

825. Convert the binary floating point value in the register specified by **S** to a decimal floating point value and store the result in the register specified by **D**.

826. ELC controllers utilize the binary floating point format. The DEBCD instruction is the instruction used to convert binary floating point to decimal floating point.

827. Flag: M1020 Zero flag, M1021 Borrow flag, M1022 Carry flag

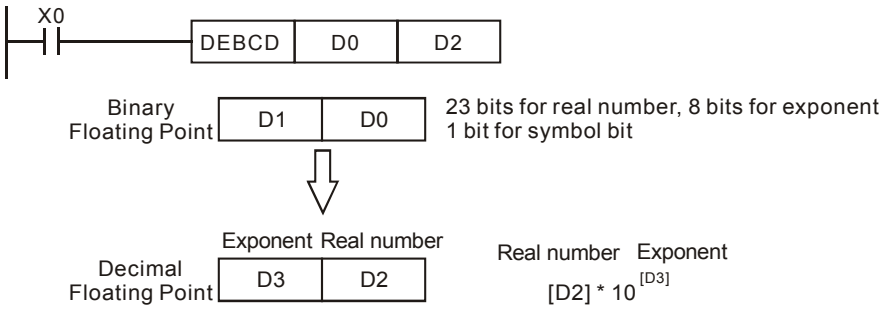
If the absolute value of the result is larger than the maximum floating point value, the carry flag M1022= ON.

If absolute value of the results is less than the minimum floating point value, the borrow flag M1021= ON.

If the conversion result is 0, zero flag M1020= ON.

Program Example:

When X0= ON, the binary floating point value in D1/D0 will be converted to decimal floating point and the result stored in D3/D2.



API	Mnemonic			Operands	Function
119	D	EBIN	P	S, D	Scientific to Floating Conversion

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			DEBIN, DEBINP: 9 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

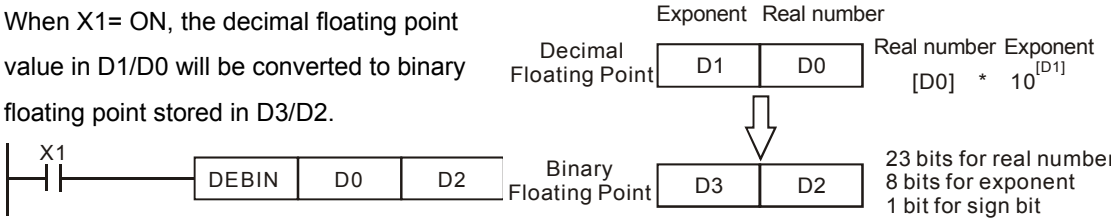
S: Data source **D**: Destination result

Description:

828. Convert a decimal floating point value in the register specified by **S** to a binary floating point value and store the result in the register specified by **D**.
829. For example, **S** =1234, **S** +1= 3 will become **S** =1.234 x 10^6
830. **D** must be in binary floating point format. **S** and **S** +1 represent the real number and the exponent of the floating point number respectively.

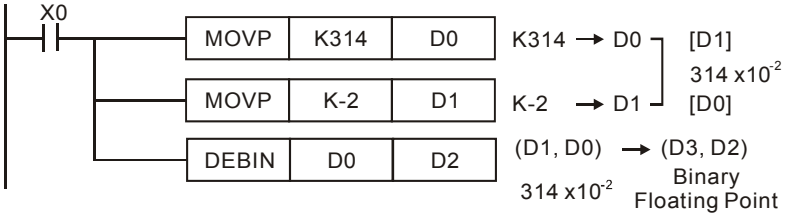
Program Example 1:

When X1= ON, the decimal floating point value in D1/D0 will be converted to binary floating point stored in D3/D2.



Program Example 2:

831. Before performing this floating point operation, you must use the FLT instruction to convert BIN integer to binary floating point. The source data should be a BIN integer. The DEBIN instruction can be used to convert a floating point value to a binary floating point value.
832. When X0= ON, move K314 to D0 and move K-2 to D1 to generate decimal floating point format (3.14 = 314 × 10^{-2}).



API	Mnemonic			Operands	Function
120	D	EADD	P	S_1, S_2, D	Floating Point Addition

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DEADD, DEADDP: 13 steps
S_2					*	*							*			
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

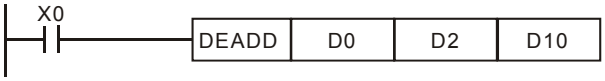
S_1: Augend S_2: Addend **D**: Addition result

Description:

833. $S_1 + S_2 = D$. The floating point value in the registers specified by S_1 and S_2 are added together and the result is stored in the registers specified by **D**.
834. If the source operand S_1 or S_2 is a constant K or H, the integer value will automatically be converted to binary floating point to perform the addition operation.
835. S_1 and S_2 can specify the same register number (the same address can be used for S_1 and S_2).
836. This instruction works best when used as a pulse instruction (EADDP).
837. Flags: M1020 (Zero flag), M1021 (Borrow flag) and M1022 (Carry flag)
    If the absolute value of the result is larger than the maximum floating point value, the carry flag M1022= ON.
    If absolute value of the results is less than the minimum floating point value, the borrow flag M1021= ON.
    If the conversion result is 0, zero flag M1020= ON.

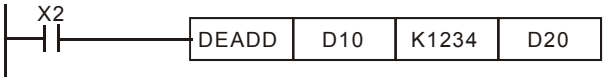
Program Example 1:

When X0= ON, add the binary floating point value in D1/D0 and the binary floating point value in D3 /D2 and store the result in D11/D10.



Program Example 2:

When X2= ON, add the binary floating point value of D11/D10 and K1234 (automatically converted to binary floating point) and store the result in D21/D20.



API	Mnemonic			Operands	Function
121	D	ESUB	P	S_1, S_2, D	Floating Point Subtraction

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DESUB, DESUBP: 13 steps
S_2					*	*							*			
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Minuend S_2: Subtrahend **D**: Subtraction result

Description:

838. $S_1 - S_2 = D$. The floating point value in the registers specified by S_2 is subtracted from the floating point value in the registers specified by S_1 and the result is stored in the registers specified by **D**. All data will be operated in floating point format and the result will be also stored in floating point format.
839. If the source operand S_1 or S_2 is a constant K or H, the integer value will automatically be converted to binary floating point to perform the subtraction operation.
840. S_1 and S_2 can specify the same register number (the same address can be used as S_1 and S_2). If this is the case and if the continuous execution of the DESUB instruction is used, the data in the register will be subtracted one time in every program scan when the condition contact is ON. Therefore, the pulse instruction (DESUBP) is recommended.
841. Flags: M1020 (Zero flag), M1021 (Borrow flag) and M1022 (Carry flag)
    If the absolute value of the result is larger than the maximum floating point value, the carry flag M1022= ON.
    If absolute value of the results is less than the minimum floating point value, the borrow flag M1021= ON.
    If the conversion result is 0, zero flag M1020= ON.

Program Example1:

When X0= ON, the binary floating point value in D3/D2 is subtracted from binary floating point value in D1/D0 and the result is stored in D11/D10.

Program Example 2:

When X2 = ON, the binary floating point value in D1/D0 will be subtracted from K1234 (automatically converted into binary floating point) and the result will be stored in D11/D10.

API	Mnemonic			Operands	Function
122	D	EMUL	P	S_1, S_2, D	Floating Point Multiplication

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DEMUL, DEMULP: 13 steps
S_2					*	*							*			
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Multiplicand S_2: Multiplier **D**: Multiplication result

Description:

842. $S_1 \times S_2 = D$. The floating point value in the registers specified by S_1 is multiplied by the floating point value in the registers specified by S_2 and the result is stored in the registers specified by **D**. All data will be operated in floating point format and the result will also be stored in floating point format.

843. If the source operands S_1 or S_2 are a constant K or H, the integer value will automatically be converted to binary floating point to perform the multiplication operation.

844. S_1 and S_2 can specify the same register number (the same address can be used as S_1 and S_2). If in this case and continuous execution of the DEMUL instruction is used, the data in the register will be multiplied one time in every program scan as long as the condition contact = ON. Therefore, the pulse instruction (DEMULP) is recommended.

845. Flags: M1020 Zero flag, M1021 Borrow flag and M1022 Carry flag

If the absolute value of the result is larger than the maximum floating point value, the carry flag M1022= ON.

If absolute value of the results is less than the minimum floating point value, the borrow flag M1021= ON.

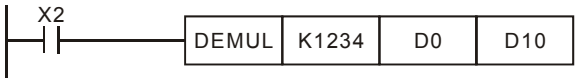
If the conversion result is 0, zero flag M1020= ON.

Program Example1:

When X1= ON, the binary floating point value in D1/D0 is multiplied by the binary floating point value in D11/D10 and the result is stored in D21/D20.

Program Example 2:

When X2 = ON, K1234 (automatically converted into binary floating point) is multiplied by the binary floating point value in D1/D0 and the result is stored in D11/D10..



API	Mnemonic			Operands	Function
123	D	EDIV	P	S_1, S_2, D	Floating Point Division

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DEADD, DEADDP: 13 steps
S_2					*	*							*			
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Dividend S_2: Divisor **D**: Quotient and Remainder

Description:

846. $S_1 \div S_2 = D$. The floating point value in the registers specified by S_1 is divided by the floating point value in the registers specified by S_2 and the result is stored in the registers specified by **D**. All data will be operated in floating point format and the result will be also stored in floating point format.
847. If the source operands S_1 or S_2 are a constant K or H, the integer value will automatically be converted to binary floating point to perform the division operation.
848. If S_2 is 0 (zero), the operation will fail and will result in an “operand error” and the instruction will not be executed.
849. Flags: M1020 Zero flag, M1021 Borrow flag and M1022 Carry flag
     If the absolute value of the result is larger than the maximum floating point value, the carry flag M1022= ON.
     If absolute value of the results is less than the minimum floating point value, the borrow flag M1021= ON.
     If the conversion result is 0, zero flag M1020= ON.

Program Example 1:

When X1= ON, the binary floating point value in D1/D0 is divided by the binary floating point value in D11/D10 and the result is stored in D21/D20.

Program Example 2:

When X2= ON, the binary floating point value in D1/D) is divided by K1234 (automatically converted to binary floating point) and the result is stored in D1/D10.

API	Mnemonic			Operands	Function
124	D	EXP	P	**S, D**	Floating Point Exponent Operation

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DEXP, DEXPP: 9 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

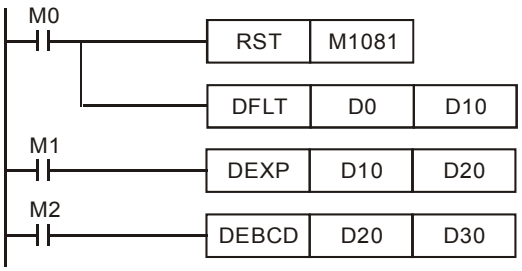
S: operand source address **D**: operand result address

Description:

850. The DEXP instruction calculates e^{S}, where the exponent is **S**
851. $e^{[S+1, S]}$=[**D** +1, **D**]
852. Both S and D must be 32 bit floating point values and therefore each value uses two consecutive D-registers.
853. **S** can be a positive or negative value.
854. When operand **D**= e^{S}, e=2.71828 and **S** is a floating point value.
855. Error flags: M1067 and M1068. Error codes: D1067 and D1068.
856. Flags: M1020 Zero flag, M1021 Borrow flag and M1022 Carry flag.
    If the absolute value of the result is larger than the maximum allowable floating point value, the carry flag M1022= ON.
    If the absolute value of the result is less than the minimum allowable floating point value, the borrow flag M1021= ON.
    If the conversion result is 0, zero flag M1020= ON.

Program Example:

857. When M0= ON, convert (D0, D1) to binary floating point and save the result in register (D10, D11).
858. When M1= ON, use (D10, D11) as the exponent to perform the operation: e^{S}. The result is a binary floating point value that is saved in register (D20, D21).
859. When M2= ON, convert (D20, D21) from binary floating point to decimal floating point and save the result in register (D30, D31).



API	Mnemonic			Operands	Function
125	D	LN	P	S, D	Floating Natural Logarithm Operation

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DEXP, DEXPP: 9 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: operand source address **D**: operand result address

Description:

860. Perform natural logarithm operation to operand **S**:

LN[**S** +1, **S**]=[**D** +1, **D**]

861. Only a positive number is valid for **S**. Operand D must use a 32-bit floating point value for this instruction. Therefore, **S** needs to be converted to floating point.

862. e^{D}=**S** and **D**=ln **S**.

863. Flags: M1020 Zero flag, M1021 Borrow flag and M1022 Carry flag.

If the absolute value of the result is larger than the maximum allowable floating point value, the carry flag M1022= ON.

If the absolute value of the result is less than the minimum allowable floating point value, the borrow flag M1021= ON.

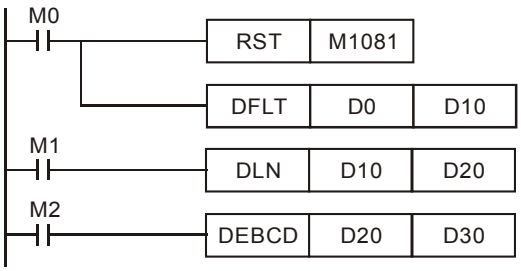
If the conversion result is 0, zero flag M1020= ON.

Program Example:

864. When M0= ON, convert (D0, D1) to binary floating point and save the result in register (D10, D11).

865. When M1= ON, use (D10, D11) as a real number and perform natural logarithm operation. The result is a binary floating point value saved in register (D20, D21).

866. When M2= ON, convert (D20, D21) binary floating point to decimal floating point and save the result in register (D30, D31).



API	Mnemonic			Operands	Function
126	D	LOG	P	S_1, S_2, D	Floating Point Logarithm Operation

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DLOG, DLOGP: 13 steps
S_2					*	*							*			
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

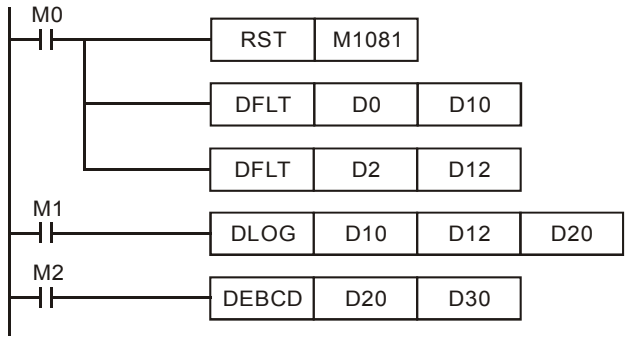
S_1: operand base address S_2: operand source address **D**: operand result address

Description:

867. Perform logarithm operation to S_1 and S_2 and save the result to **D**. LOG base S_1 of S_2 = **D.**
868. Only positive numbers are valid for S_2 (positive and negative numbers are valid for S_1). Operand D is a 32-bit floating point value. Therefore, S_1 and S_2 need to be converted to floating point.
869. Flag: M1020 Zero flag, M1021 Borrow flag and M1022 Carry flag.
    If the absolute value of the result is larger than the maximum allowable floating point value, the carry flag M1022= ON.
    If the absolute value of the result is less than the minimum allowable floating point value, the borrow flag M1021= ON.
    If the conversion result is 0, zero flag M1020= ON.

Program Example:

870. When M0= ON, convert (D0, D1) and (D2, D3) to binary floating point and save the result in register (D10, D11) and (D12, D13). (D10, D11) is the base of the log and (D12,D13) is the value we are taking the log of.
871. When M1= ON, use (D10, D11) and (D12, D13) binary floating point of 32-bit registers to perform logarithm operation and save the result in 32-bit register (D20, D21).
872. When M2= ON, convert (D20, D21) binary floating point of 32-bit registers to decimal floating point and save the result in register (D30, D31).



API	Mnemonic			Operands	Function
127	D	ESQR	P	S, D	Floating point Square Root

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DESQR, DESQRP: 9 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

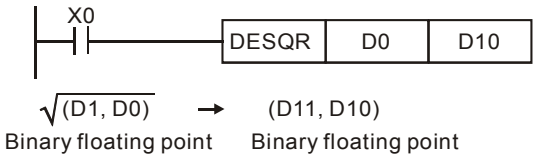
S: Source address **D**: Destination address to store the result

Description:

873. This instruction performs a square root operation on the floating point value of source **S** and stores the result at the destination **D**. All operations are performed in the floating point format and the result will also be stored in floating point format.
874. If the source **S** is a constant K or H, the integer value will automatically be converted to binary floating point to perform the ESQR operation.
875. If the ESQR operation result in **D** is 0 (zero), the Zero flag M1020= ON.
876. **S** can only be a positive value. Performing any square root operation on a negative value will result in an “operation error” and this instruction will not be executed. M1067 and M1068 = ON and error code “0E1B” will be recorded in D1067.
877. Flags: M1020 (Zero flag), M1067 (Program execution error)

Program Example 1:

When X0= ON, the square root of binary floating point value in (D1, D0) is stored in register (D11, D10) in binary floating point.



Program Example 2:

When X2= ON, the square root of K1234 (automatically converted to binary floating point) is stored in (D11, D10).

API	Mnemonic			Operands	Function
128	D	POW	P	S_1, S_2, D	Floating Point Power Operation

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DPOW, DPOWP: 13 steps
S_2					*	*							*			
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: base address. S_2: exponent. **D**: result

Description:

878. Perform the power operation to binary floating point S_1 to the power of S_2 and save the result to **D**.

POW [S_1+1, S_1]^[S_2+1, S_2]= **D.** That is, S_1 to the power of S_2 .Only a positive number is valid for S_1. Result **D** needs to use two consecutive D-registers to accommodate the 32-bit floating point value. S_1 and S_2 also need to be converted to floating point before executing the DPOW instruction.

879. Error flags: M1067 and M1068, read D1067 and D1068.

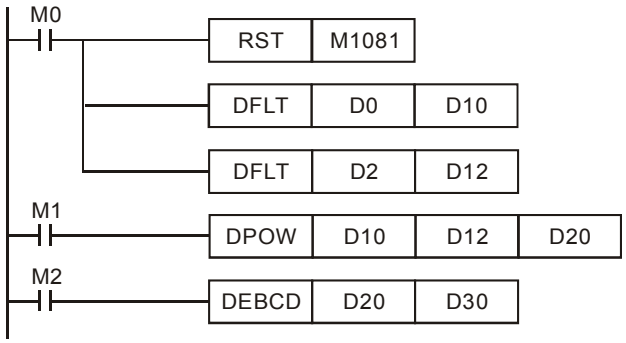
If the absolute value of the result is larger than the maximum allowable floating point value, the carry flag M1022= ON.

If the absolute value of the result is less than the minimum allowable floating point value, the borrow flag M1021= ON.

If the conversion result is 0, zero flag M1020= ON.

Program Example:

880. When M0= ON, convert (D0, D1) and (D2, D3) to binary floating point and save the results in register (D10, D11) and (D12, D13).
881. When M1= ON, use (D10, D11) and (D12, D13) to perform the power operation and save the result in 32-bit register (D20, D21).
882. When M2= ON, convert (D20, D21) binary floating point 32-bit registers to decimal floating point and save the result in register (D30, D31).



API	Mnemonic			Operands	Function
129	D	INT	P	**S, D**	Floating Point to Integer

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*	*	*			INT, INTP: 5 steps
D											*	*	*			DINT, DINTP: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Source address **D**: Destination address to store the result

Description:

883. The binary floating point value of register **S** is converted to BIN integer and stored in register **D**.
884. This instruction is the opposite of the API 49 (FLT) instruction.
885. Flags: M1020 (Zero flag), M1021 (Borrow flag), M1022 (Carry flag)

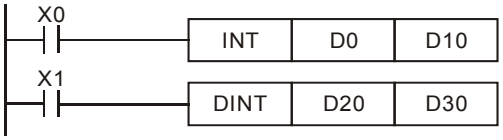
     If the absolute value of the result is larger than the maximum allowable floating point value, the carry flag M1022= ON.

     If the absolute value of the result is less than the minimum allowable floating point value, the borrow flag M1021= ON.

     If the result exceeds the integer or double integer range (an overflow occurs), the Carry flag M1022= ON.
886. ELCB-PB/ELC-PA/ELC-PVV1.2 don't support T,C registers.

Program Example:

887. When X0= ON, the binary floating point value of (D1, D0) will be converted to BIN integer and the result is stored in (D10).
888. When X1= ON, the binary floating point value of (D21, D20) will be converted to BIN integer and the result is stored in (D31, D30).



API	Mnemonic			Operands	Function
130	D	SIN	P	S, D	Floating Point Sine Operation

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DSIN, DSINP: 9 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

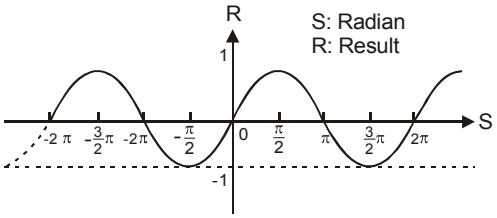
Operands:

S: Specified RAD value or an angle ($0^\circ \leqq$ **S** $< 360^\circ$) **D**: Result of the SIN operation

Description:

889. Source **S** can be in radians or an angle based on flag M1018.
890. When M1018= OFF, it is set to radian mode. RAD=angle $\times \pi$ /180.
891. When M1018= ON, it is set to angle mode. Angle range: $0^\circ \leqq$ angle $< 360^\circ$.
892. The SIN value of an angle specified by **S** is calculated and the result is stored in the register specified by **D**.
893. Flag: M1018 flag for radian/angle.

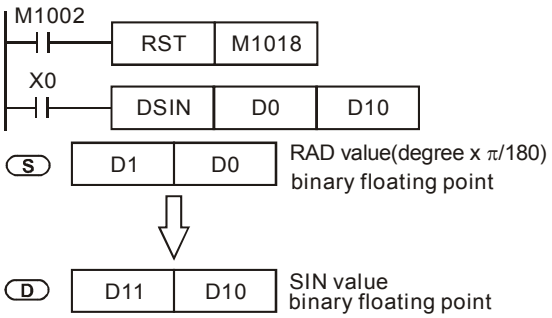
Following shows the relation between radian and result:



894. If result of D is 0, the Zero flag M1020=ON.

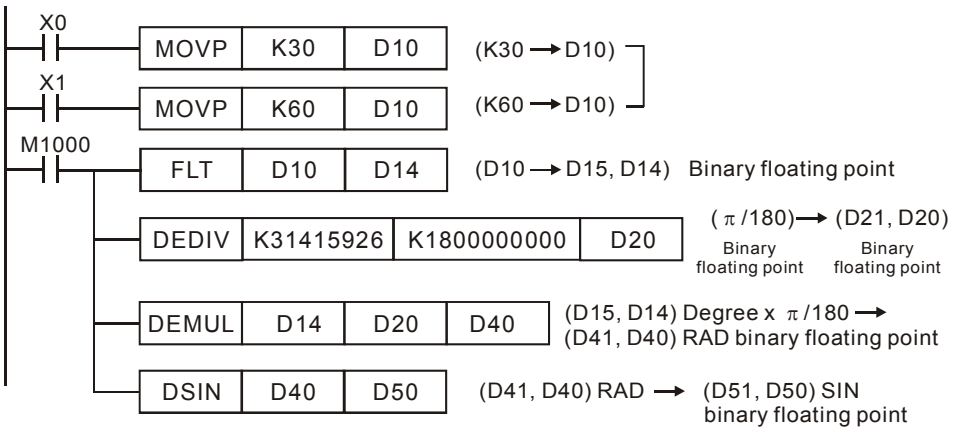
Program Example 1:

When M1018= OFF, the radian mode is used. When X0= ON, the SIN of the RAD value (D1, D0) is calculated and the result is stored in (D11, D10). The result in (D11, D10) is in binary floating point format.



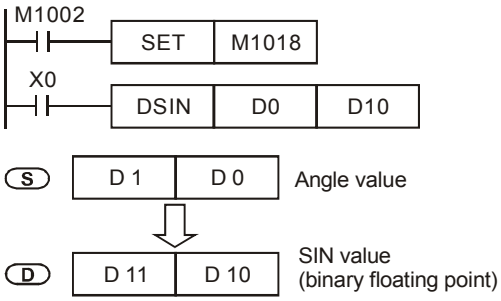
Program Example 2:

When M1018= OFF, the radian mode is used. Select an angle of 30 degrees with X0 or 60 degrees with X1 and convert the angle to a RAD value to calculate the SIN.



Program Example 3:

When M1018= ON, the angle mode is used. When X0= ON, the SIN of the angle value in (D1, D0) is calculated and the result is stored in (D11, D10). The angle range is: 0°≦angle value＜360°. The result is saved in (D11, D10) in the binary floating point format.



API	Mnemonic	Operands	Function
131	D COS P	S, D	Floating Point Cosine Operation

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DCOS, DCOSP: 9 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

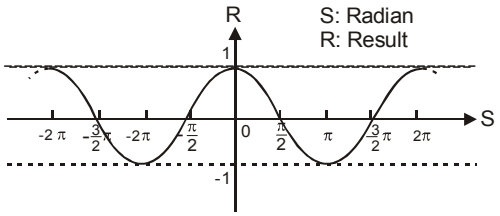
Operands:

S: Specified RAD value ($0^\circ \leqq S < 360^\circ$) **D**: Result of the COS operation

Description:

895. Source designated by **S** can be radians or an angle based on flag M1018.
896. When M1018= OFF, is set to radian mode. RAD=angle $\times \pi$ /180.
897. When M1018= ON, is set to angle mode. Angle range: $0^\circ \leqq$ angle $< 360^\circ$.
898. The COS value of an angle specified by **S** is calculated and the result is stored in the register specified by **D**.

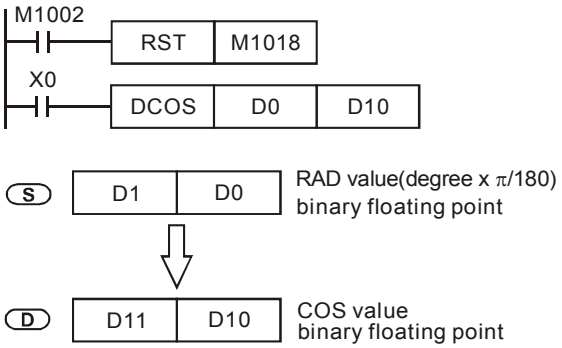
Following shows the relation between radian and result:



899. If result of D is 0, the Zero flag M1020=ON.

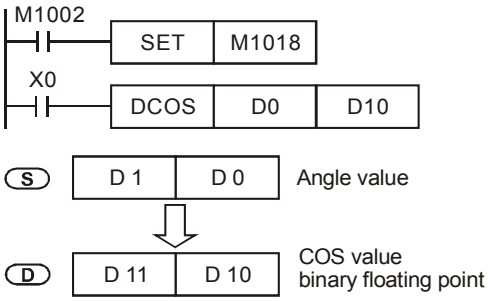
Program Example 1:

When M1018= OFF, it is in radian mode. When X0= ON, calculate the COS value of the angle specified in (D1, D0) and store the result in (D11, D10). The value in (D1, D0) and the result stored in (D11, D10) are all in binary floating point format.



Program Example 2:

When M1018= ON, it is angle mode. When X0= ON, take the COS of the angle specified in (D1, D0) and store the result in (D11, D10) in the binary floating point format.



API	Mnemonic			Operands	Function
132	D	TAN	P	S, D	Floating Point Tangent Operation

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DTAN, DTANP: 9 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Specified RAD value ($0^\circ \leqq$ **S** $< 360^\circ$) **D**: Area where calculated result is stored

Description:

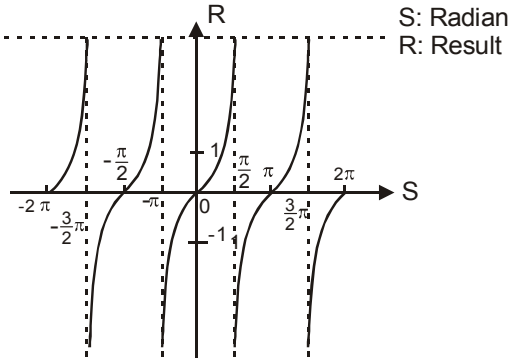
900. Source designated by **S** can be in radians or an angle based on flag M1018.

901. When M1018= OFF, is set to radian mode. RAD=angle $\times \pi$ /180.

902. When M1018= ON, is set to angle mode. Angle range: $0^\circ \leqq$ angle $< 360^\circ$.

903. The TAN value of an angle specified by **S** is calculated and the result is stored in the register specified by **D**.

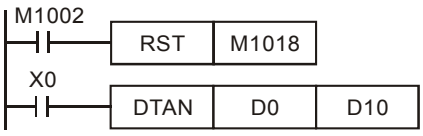
Following shows the relation between radian and result:

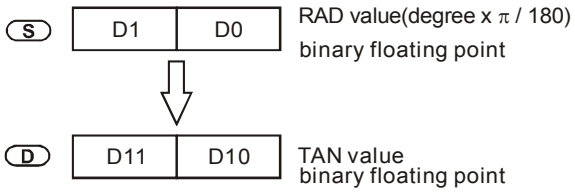


904. If result of D is 0, the zero flag M1020=ON.

Program Example 1:

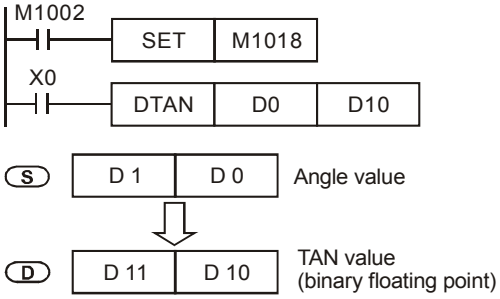
When M1018= OFF, it is in radian mode. When X0= ON, take the TAN of RAD value (D1, D0) and store the result in (D11, D10). The value in (D1, D0) and the result stored in (D11, D10) are all in binary floating point format.





Program Example 2:

When M1018= ON, it is angle mode. When X0= ON, take the TAN of the angle specified by (D1, D0) and store the result in (D11, D10) in the binary floating point format.



API	Mnemonic			Operands	Function
133	D	ASIN	P	S, D	Floating Point Arcsine Operation

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DASIN, DASINP: 9 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

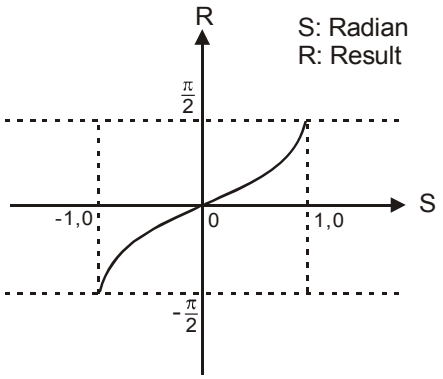
Operands:

S: Specified source (binary floating point) **D**: Area where calculated result is stored

Description:

905. ASIN value=SIN^{-1}

906. See the figure below for the relation between radians and the result of the arcsine operation:

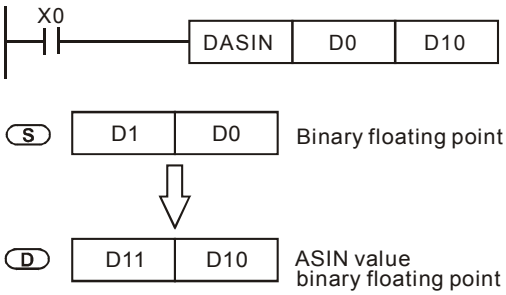


907. If the result in D is 0, the zero flag M1020=ON.

908. The decimal floating point of the SIN value designated by **S** should be within -1.0 ~ +1.0. If the value falls outside of this range, error bits M1067 and M1068 will be ON.

Program Example:

When X0= ON, calculate the ASIN of the specified binary floating point value (D1, D0) and save the result in (D11, D10). The result stored in (D11, D10) is in the binary floating point format.



API	Mnemonic			Operands	Function
134	D	ACOS	P	S, D	Floating Point Arccosine Operation

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DACOS, DACOSP: 9 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

3

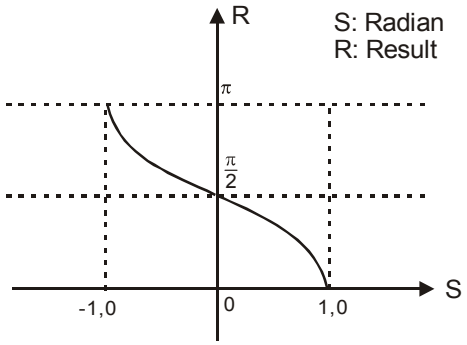
Operands:

S: Specified source (binary floating point) **D**: Area where calculated result is stored

Description:

909. ACOS value=COS^{-1}

910. See the figure below for the relation between radian and result:

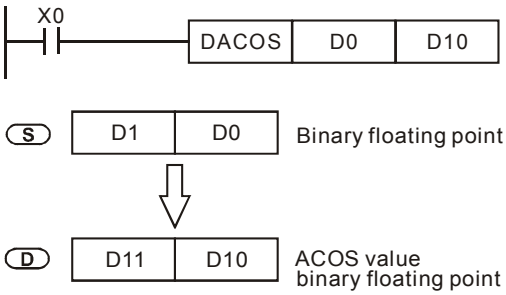


911. If the result of D is 0, the zero flag M1020=ON.

912. The decimal floating point of the COS value designated by **S** should be within -1.0 ~ +1.0. If the value falls outside of this range, error bits M1067 and M1068 will be ON.

Program Example:

When X0= ON, the ACOS of binary floating point value (D1, D0) is calculated and the result is saved in (D11, D10). The result (D11, D10) is in binary floating point format.



API	Mnemonic			Operands	Function
135	D	ATAN	P	S, D	Floating Point Arctangent Operation

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DATAN, DATANP: 9 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

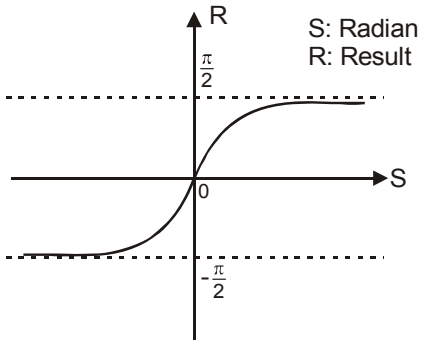
Operands:

S: Specified source (binary floating point) **D**: Area where calculated result is stored

Description:

913. ATAN value=TAN^{-1}

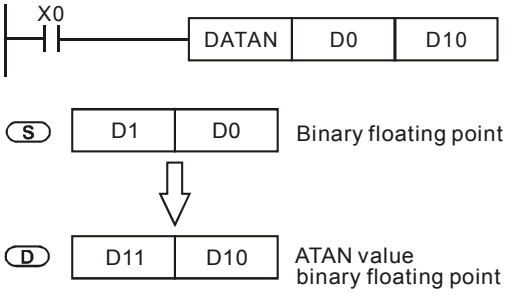
914. See the figure below for the relation between radian and result:



915. If the result of D is 0, the zero flag M1020=ON.

Program Example:

When X0= ON, calculate the ATAN of the binary floating point value (D1, D0) and save the result in (D11, D10). The result stored in (D11, D10) is in binary floating point format.



API	Mnemonic			Operands	Function
136	D	SINH	P	S, D	Hyperbolic Sine

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DSINH, DSINHP: 9 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

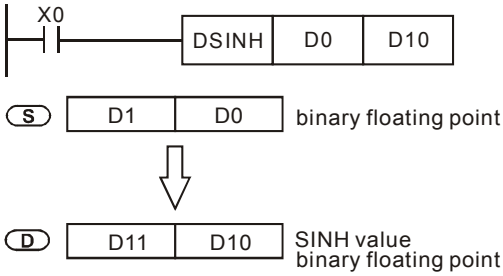
- **S**: Source value (binary floating point) **D**: SINH result

Description:

1. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
2. SINH **S** =(e^{S}-e^{-S})/2. The result is stored in **D**.

- Program Example:

1. When X0 = On, obtain the SINH of binary floating point value (D1, D0) and store the binary floating point result in (D11, D10).



2. If the absolute value of the result ＞ maximum available floating point value, the carry flag M1022 = On.
3. If the absolute value of the result ＜ minimum available floating point value, the borrow flag M1021 = On.
4. If the result = 0, the zero flag M1020 = On.

API	Mnemonic			Operands	Function
137	D	COSH	P	S, D	Hyperbolic Cosine

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DCOSH, DCOSHP: 9 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

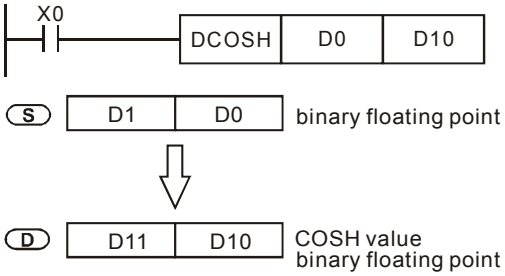
- **S**: Source value (binary floating point) **D**: COSH result

Description:

1. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
2. COSH **S** =(e^{S}+e^{-S})/2. The result is stored in **D**.

- Program Example:

1. When X0 = On, obtain the COSH of binary floating point value (D1, D0) and store the binary floating point result in (D11, D10).



2. If the absolute value of the result ＞ maximum available floating point value, the carry flag M1022 = On.
3. If the absolute value of the result ＜ minimum available floating point value, the borrow flag M1021 = On.
4. If the result = 0, the zero flag M1020 = On.

API	Mnemonic			Operands	Function
138	D	TANH	P	S, D	Hyperbolic Tangent

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DTANH, DTANHP: 9 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

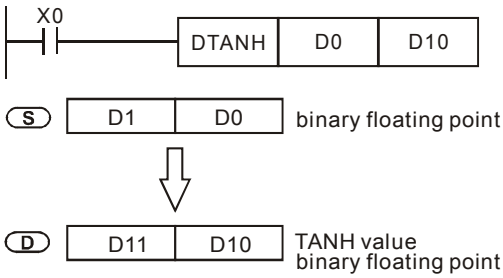
- **S**: Source value (binary floating point) **D**: TANH result

Description:

1. See the specifications of each model for their range of use.
2. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
3. TANH **S** $=(e^{S}-e^{-S})/(e^{S}+e^{-S})$. The result is stored in **D**.

- Program Example:

1. When X0 = On, obtain the TANH of binary floating point value (D1, D0) and store the binary floating point result in (D11, D10).



2. If the absolute value of the result ＞ maximum available floating point value, the carry flag M1022 = On.
3. If the absolute value of the result ＜ minimum available floating point value, the borrow flag M1021 = On.
4. If the result = 0, the zero flag M1020 = On.

API	Mnemonic		Operands	Function
143	DELAY	P	**S**	Delay

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DELAY, DELAYP: 3 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

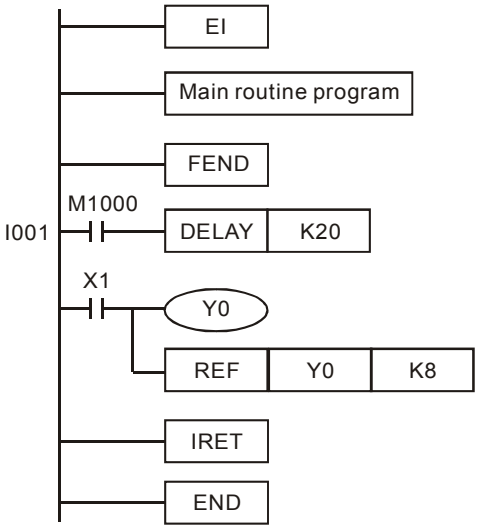
S: Delay time, units are 0.1ms (K1~K1000)

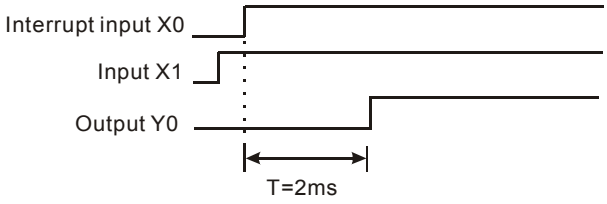
Description:

After executing the DELAY instruction, in every scan the program will be delayed by the time specified in the Delay instruction. The delay will occur in each scan immediately following the execution of the Delay instruction.

Program Example:

If an external interrupt occurs when X0 goes from OFF to ON, the interrupt subroutine will execute the DELAY instruction and Y0=ON will be delayed by 2 seconds when X1=ON.





Points to note:

916. When executing the DELAY instruction, the delay time may increase upon the execution of communications, high-speed counters and high-speed pulse output commands.
917. If the external output (transistor output or relay output) is specified, the delay time may increase due to the transistor or relay turn-on delay.

API	Mnemonic	Operands	Function
144	GPWM	S_1, S_2, D	General PWM Output

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			GPWM: 7 steps
S_2													*			
D		*	*	*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

3

Operands:

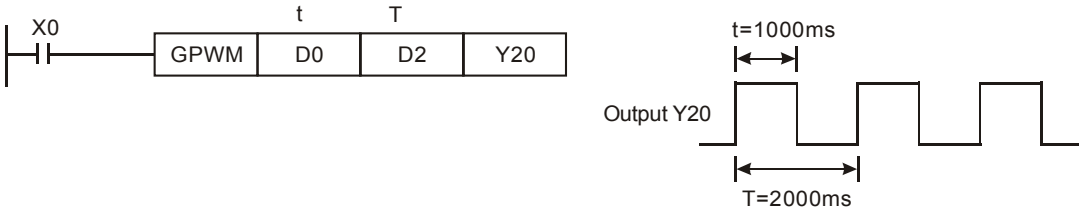
S_1: Pulse output width S_2: Pulse output cycle (occupies 3 addresses) **D**: Pulse output address

Description:

918. S_1 is specified as the pulse output width as t: 0~32,767ms.
919. S_2 is specified as the pulse output cycle as T:1~32,767ms, $S_1 \leqq S_2$.
920. S_2 +1 and S_2 +2 are for internal system use. They should not be used in the program. If the data in these addresses is modified in any way, unexpected operation of the GPWM instruction could occur. The address for the pulse output (**D)** must be: Y, M or S.
921. When the GPWM instruction is executed, pulses will be sent to the Pulse output address (**D)** based on the pulse output width S_1 and pulse output cycle S_2.
922. When $S_1 \leqq 0$, no pulses will be generated. When $S_1 \geqq S_2$, the pulse output address will be always ON.
923. S_1 and S_2 can be modified when executing GPWM instruction.

Program Example:

When D0=K1000, D2=K2000 and X0= ON, the following pulses will be sent out Y20. When X0= OFF, the Y20 output will also be OFF.



Points to note:

924. This instruction counts by scan cycle so the maximum offset will be an ELC program scan cycle. The value of S_1, S_2 and (S_2 - S_1) should be larger than one ELC program scan cycle. Otherwise, an error will be generated.
925. If using this instruction in a subroutine or interrupt, the GPWM output may not be accurate.

API	Mnemonic	Operands	Function
145	FTC	S_1, S_2, S_3, D	Fuzzy Temperature Control

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			FTC: 9 steps
S_2					*	*							*			
S_3													*			
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

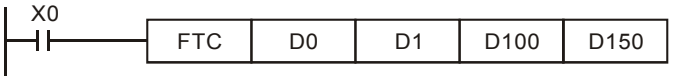
S_1: Target value (SV) S_2: Present measured value (PV) S_3: Parameter list (occupies 6 consecutive addresses) **D**: Output value (MV)

Description:

926. Operand S_1 range is 1~5000 representing 0.1°C ~500°C. The units are 0.1°C. If S_3+1 (refer to the Notes below) is set to K0 it represents 0.1°C~500°C.
927. Operand S_2 range is 1~5000 representing 0.1°C ~500°C. The units are 0.1°C. If S_3+1 (refer to the Notes below) is set to K0 it represents 0.1°C~500°C.
928. The analog temperature input will be scaled to the range 1~5000.
929. S_3 is the sampling time setting. If this setting is less than K1, the instruction will not operate. If this setting exceeds K200, it will use K200.
930. Bit 0 of S_3+1 configures the temperature for °C or °F. If bit 0 is K0, it's °C and if bit 0 is K1 it's °F. If Bit 1 of S_3+1 is K0 the input filter is disabled. If bit 1 of S_3+1 is k1, the input filter is active. Refer to the Notes below for information on bits 2-5.
931. The range of **D** is 0~(sampling time*100). When using the FTC instruction, it is necessary to use it in conjunction with other instructions, based on the heater type. For example, the FTC instruction can be used with the GPWM instruction to control output pulses. (Sampling time*100) would be the output cycle time of GPWM and MV the output width of the GPWM pulse. See Example 1 below.
932. There is no limit on the number of times this instruction can be used in a program.

Program Example:

933. Enter the parameters before executing the FTC instruction.
934. When X0= ON, the instruction is executed and the result is saved in D150. When X0= OFF, instruction is not executed and the data is unchanged.

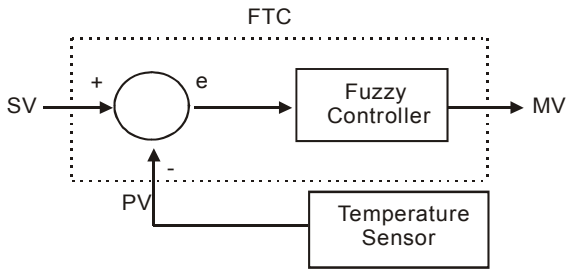


Notes:

935. The settings for **S_3** are as follows:

Device	Function	Usage range	Explanation
S_3:	Sampling time (T_S)	1~200 (unit: 100ms)	When T_S is less than the scan time of the program, the PID instruction will execute for one scan time. When T_S=0, the instruction will not execute. The minimum setting of T_S should be larger than a program scan time. When this setting exceeds 200, the value 200 will be used.
S_3 +1:	b0:Temperature unit b1:filter function b2~b5:heat environment b6~b15 preserved	b0=0 : °C b0=1 : °F	Temperature units
		b1=0:no filter b1=1:filter	When the filter is disabled, the present value(PV)=present measured value. If the filter is enabled, the present value(PV)=(present measure value + previous measure value)/2
		b2=1	Slow heat environment
		b3=1	General heat environment
		b4=1	Rapid heat environment
		b5=1	High-speed heat environment
S_3 +2: ≀ **S_3** +6:	For system use only, do not use.		

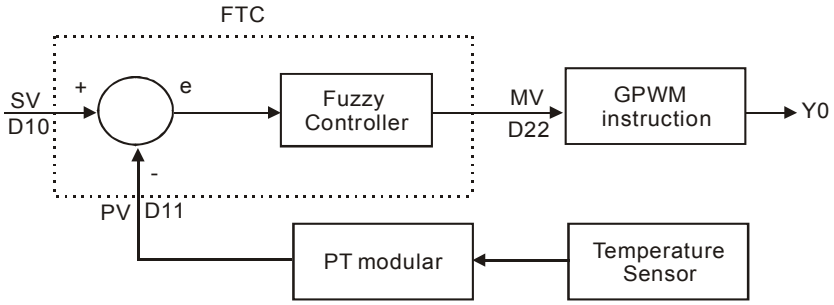
936. Control Diagram:



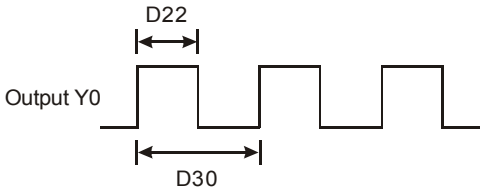
Points to notes:

It is recommended to set the sampling time to twice that of the sampling time of the temperature sensor for better temperature control.

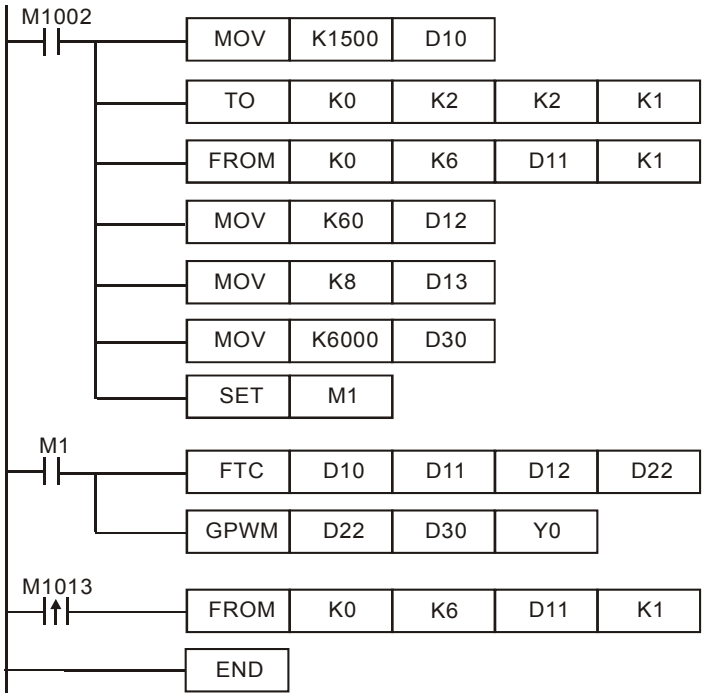
Example 1: control diagram



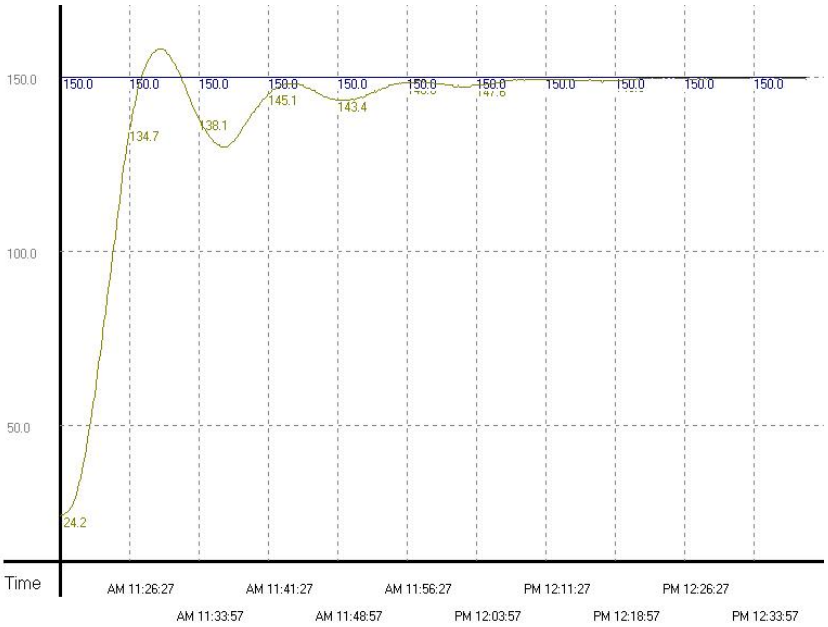
The output D22 (MV) of the FTC instruction is used as the input to the GPWM instruction. D22 represents the pulse output width and D30 is the pulse output cycle. The timing chart for output Y0 is as follows.



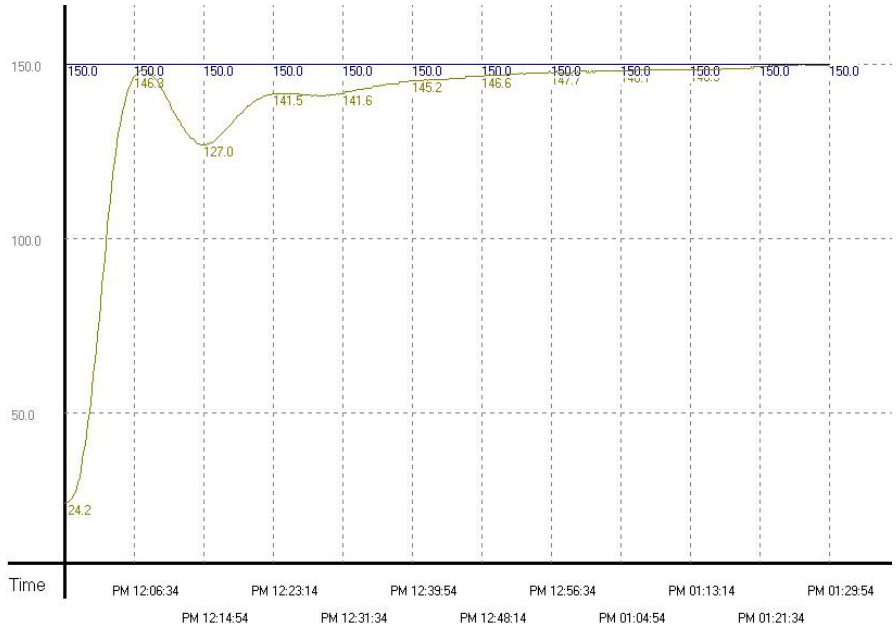
The settings for the FTC instruction for this example are D10=k1500(target temperature), D12=k60(sampling time is 6 seconds), D13=k8 (Bit 3=1) and D30=k6000 (=D12*100). The ladder diagram is as follows:



This example application is controlling an oven (The temperature it can heat is 250℃). A plot of the target and real temperatures are shown below.

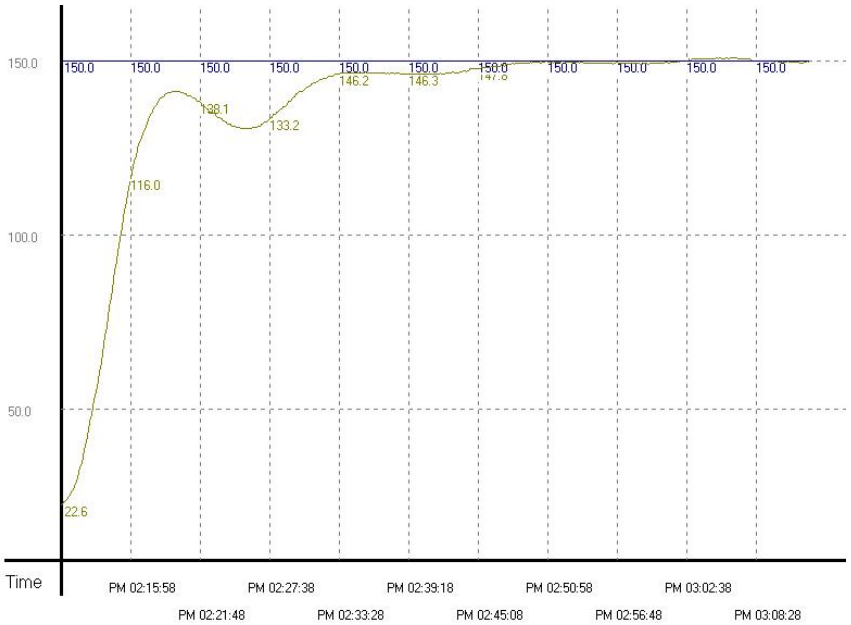


Example 2: This example application requires rapid heating. Use D13=k16 due to overshoot. The result after testing is shown as follows.



From the plot above, there is no overshoot but it takes more than one hour and fifteen minutes to make the error of the target temperature ± 1℃. This means that the current test environment is correct but there will be long sampling time delays.

Example 3: To make example 2 reach the target temperature faster, set the sampling time to 4 seconds (D12=k40 and D30=k4000). The result is shown as below.



API	Mnemonic	Operands	Function
146	CVM	S_1, S_2, D	Valve Control

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			CVM: 7 steps
S_2					*	*							*			
D		*	*	*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

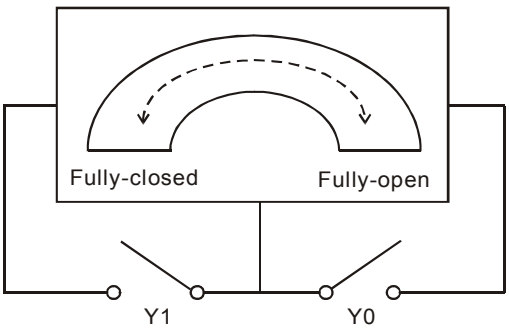
- S_1: Target time of valve (absolute position) S_2: Time from fully-closed to fully-open of valve (destination) **D**: Startingoutput address

Description:

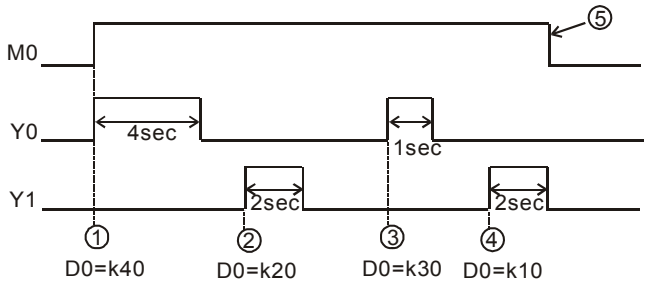
1. S_1 occupies 3 consecutive registers. S_1 + 0 is for storing the designated value; S_1 + 1 is the current position of the valve and S_1 + 2 is for storing the parameters recorded in the instruction execution. DO NOT alter the last two registers.
2. **D** occupies 2 consecutive output addresses. **D** + 0 is the “open” contact and **D** + 1 is the “closed” contact.
3. The unit of time: 0.1 seconds. When the scan time of the program exceeds 0.1 seconds, DO NOT use this instruction to adjust the position of the valve.
4. Frequency of the output device: 10Hz.
5. When the time of S_1 + 0 ＞ the fully-opened time set in S_2, **D** + 0 will remain On and **D** + 1 will remain Off. When the time of S_1 + 0 ＜ 0, **D** + 0 will remain Off and **D** + 1 will remain On.
6. When the instruction is enabled, it will start to control the valve from “0” time position. If it cannot be determined that the valve is at “0” before executing the instruction, be sure that S_1 + 0 is less than 0 and execute the instruction for S_2 (time) before sending in the correct target control time.

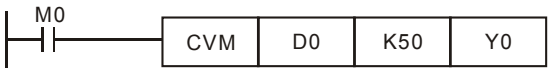
- Program Example 1:

1. The control valve



2. Definitions of the control valve:
   a) When Y0 and Y1 = Off: No valve action
   b) When Y0 = On and Y1 = Off: Valve “open”
   c) When Y0 = Off and Y1 = On: Valve “closed”
   d) When Y0 and Y1 = On: The action is prohibited.
3. Timing diagram and program of the control:

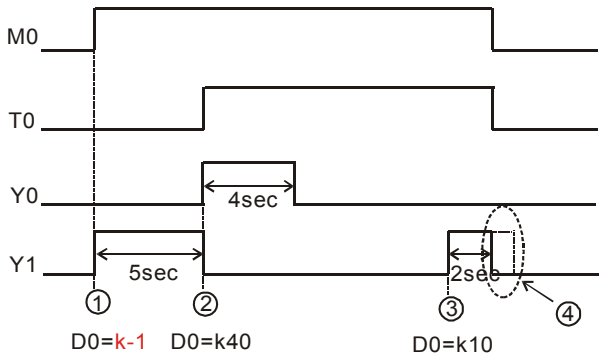


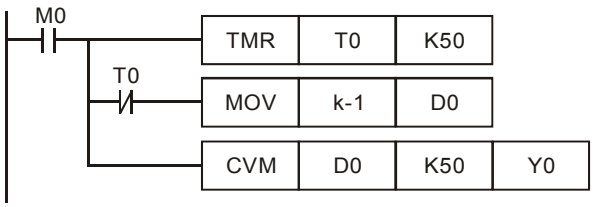


4. Control phases:
   1) Phase ①: When M0 = On, D0 = K40 implies the valve shall be open (Y0 = On, Y1 = Off) until 4 seconds is reached.
   2) Phase ②: Change the position of the valve. Make D0 = K20. Because the previous position was at 4 seconds, the valve shall be closed (Y0 = Off, Y1 = On) for 2 seconds.
   3) Phase ③: Change the position of the valve. make D0 = K30. Because the previous position was at 2 seconds, the valve shall be open (Y0 = On, Y1 = Off) for 1 second, moving the valve to the position of 3 seconds.
   4) Phase ④: Change the position of the valve. Make D0 = K10. Because the previous position was at 2 seconds, the valve shall be closed (Y0 = Off, Y1 = On) for 2 seconds, moving the valve to the position of 1 second.
   5) Phase ⑤: Switch off M0 and the valve will remain at it’s last state (Y0 = Off, Y1 = Off).

◆ Program Example 2:

1. Timing diagram and program control:





2. Control phases:

1) Phase ①: When M0 = On, because the valve position is not known, set D0 = K-1 to deliberately close the valve (Y0 = Off, Y1 = On) for 5 seconds and make sure the valve is at the position of 0 seconds before moving on to the next step.
2) Phase ②: When T0 = On, allow D0 = K40 to begin execution. Open the valve (Y0 = On, Y1 = Off) for 4 seconds, moving the valve to the position of 4 seconds.
3) Phase ③: Change the position of the valve and make D0 = K10. Because the previous position was at 4 seconds, the valve shall be closed (Y0 = Off, Y1 = On) for 3 seconds, moving the valve to the position of 1 second.
4) Phase ④: Switch off M0 and the valve will no longer move (Y0 = Off, Y1 = Off).

API	Mnemonic			Operands	Function
147	D	SWAP	P	**S**	Swap High/Low Byte

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*	*	*	SWAP, SWAPP: 3 steps DSWAP, DSWAPP: 5 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

3

Operands:

S: Address of the value to be byte swapped.

Description:

937. When used as a 16-bit instruction, swap high/low bytes.

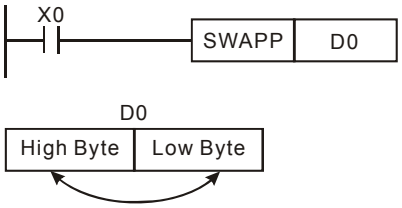
938. When used as a 32-bit instruction, swap high/low bytes of each word separately.

939. This instruction works best using the pulse option (SWAPP, DSWAPP).

940. If operand **D** is used with index register F, it is only available as a 16-bit instruction.

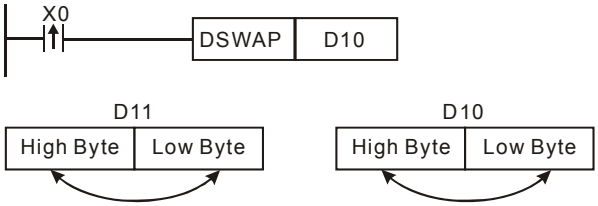
Program Example 1:

When X0=ON, swap the high/low bytes of D0.



Program Example 2:

When X0=ON, swap the high/low bytes of word D10 and of word D11.



API	Mnemonic			Operands	Function
148	D	MEMR	P	m, D, n	File Memory Read

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
m					*	*							*			MEMR, MEMRP: 7 steps DMEMR, DMEMRP: 13 steps
D													*			
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB/PE			PH/PA			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

m: Constant offset for reading from the data file register **D**: Address for storing read data

n: Quantity of data read from the file register

Description:

941. Range of **m**, **D** and **n** in each model:

model	ELC-PA	ELC-PV	ELC2-PV	ELC2-PC/PA ELCM-PH/PA
m	K0~K1599	K0~K9,999	K0~K49,999	K0~K4,999
D	D2000~D4999	D2000~D9999	D2000~D11999	D2000~D9999
16-bit n	K1~K1,600	K1~K8,000	K1~K8,000	K1~K5,000
32-bit n	K1~K800	K1~K4,000	K1~K4,000	K1~K5,000

942. The number of 16-bit file registers in each model(all can be read):

model	ELC-PA	ELC-PV	ELC2-PV	ELC2-PC/PA ELCM-PH/PA
16-bit file registers	1,600	10,000	50,000	5,000

943. **m** and **n** of ELC-PA do not support E and F index registers modification.

944. If operands **m**, **D** and **n** are out of range, an operand error will occurr. M1067, M1068= ON and error code 0E1A will be recorded in D1067.

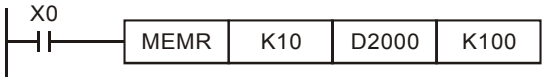
945. Flag: M1101, please refer the Notes below in API 149.

946. MEMR/MEMRP supports ELCM-PH/PA version 2.0 (and above).

Program Example 1:

947. 16-bit instruction MEMR reads 100 16-bit words of data from the 10th address of the file register and stores the read data in the data register starting with D2000.

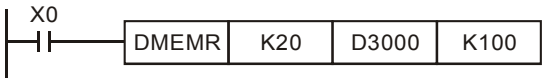
948. When X0= ON, the instruction is executed. When X0 goes to OFF, the instruction is not executed.



Program Example 2:

949. 32-bit instruction DMEMR reads 100 32-bit data words from the 20th address of the file register and store the read data in the data register starting with D3000.

950. When X0= ON, the instruction is executed. When X0 goes to OFF, the instruction is not executed.



API	Mnemonic			Operands	Function
149	D	MEMW	P	**S, m, n**	File Memory Write

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			MEMW, MEMWP: 7 steps
m					*	*							*			DMEMW, DMEMWP: 13 steps
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB/PE			PH/PA			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

3

Operands:

S: Address for storing the data to be written to the file register **m**: Constant offset into the file register **n**: Quantity of data to be written to the file register

Description:

951. Range of **m**, **D** and **n** in each model:

model	ELC-PA	ELC-PV	ELC2-PV	ELC2-PC/PA ELCM-PH/PA
m	K0~K1599	K0~K9,999	K0~K49,999	K0~K4,999
D	D2000~D4999	D2000~D9999	D2000~D11999	D2000~D9999
16-bit **n**	K1~K1,600	K1~K8,000	K1~K8,000	K1~K100
32-bit **n**	K1~K800	K1~K4,000	K1~K4,000	K1~K100

952. ELC uses this instruction to write data register data into file registers.

953. The number of 16-bit file registers in each model:

model	ELC-PA	ELC-PV	ELC2-PV	ELC2-PC/PA ELCM-PH/PA
16-bit file registers	1,600	10,000	50,000	5,000
all can be write by MEMW	Yes	Yes	No	Yes
Write times limited	None	None	10,000	10,000

954. There are 50,000 files registers in ELC2-PV. Owing to the fact that 40,000 of the file registers (K10,000~K49,999) are stored permanently in a flash ROM, values can only be written into

these file registers by means of ELCSoft, and API149 MEMW does not support these file registers.

955. **m** and **n** of ELC- PA do not support E and F index registers modification.
956. If operands **m**, **D** and **n** are out of range, an operand error will occur. M1067, M1068= ON and error code 0E1A will be recorded in D1067.
957. Flag: M1101, please refer the Note below .
958. MEMW/MEMWP supports ELCM-PH/PA version 2.0 (and above).

Program Example:

959. When X0= ON, the double word instruction DMEMW is executed. Write 100 32-bit data words starting from D2001, D2000 into the file register address 0 to 199.
960. When X0= ON, the instruction is executed. When X0 goes to OFF, the instruction is not executed.

File Register Notes:

961. ELCM-PH/PA and ELC2-PC/PA do not support M1101..
962. ELC-PA, when the ELC starts up, it will determine the state of M1101 and the contents of D1101 (file register provides numbers, K0~K1,599), D1102 (the number of file registers to be read, K1~K1,600), D1103 (destination address which stores the read data from the file register, specified by a data register D number, K2,000~K4,999) and automatically transfers the contents of the file register to the specified data register.
963. ELC-PA, when the value of D1101 is less than 0 or more than 1,599, or the value of D1103 is less than 2,000 or more than 4,999, reading data from file register to data register is disabled.
964. ELC-PV, ELC2-PV: When the ELC is powered, it will decide whether to automatically send the data in the file register to the designated data register based on the state of M1101, the contents of D1101 (starting address in file register K0 ~ K9,999), D1102 (amount of data to be read in file register K1 ~ k8,000), and D1103 (address for storing read data, starting from D, ELC-PV, ELC2-PV: K2,000 ~ K9,999; ELC2-PV: K2,000 ~ K11,999).
965. ELC-PV, ELC2-PV: The reading of data from file register to data register D will not be executed if D1101 < 0, D1101 > K9,999, D1103 < K2,000 or (ELC-PV: D1103 > K9,999; ELC2-PV: D1103>K11,999).
966. When the file register read executes,, if the address of the file register or data register exceeds the allowable range, the ELC will stop reading.
967. File registers do not have actual addresses. Reading and writing to the file registers can only be done through API 148 MEMR, API 149 MEMW, using an offset into the file register.
968. If the address in the file register to be read exceeds its range, the read value will be 0.

969. Related special relays and registers of file register:

Flag	Function Explanation
M1101	Startup function for the file register, Latched, Default is OFF

Special Register	Function Explanation
D1101	Starting address of the file register. ELC-PA: K0 ~ K1,599; ELC-PV, ELC2-PV: K0 ~ K9,999; latched; default = 0
D1102	Maximum amount of data to read/write to the file register. ELC-PA: K1 ~ K1,600; ELC-PV, ELC2-PV: K1 ~ K8,000; latched; default = 1600
D1103	D-register address for storing read/write data for the file register. ELC-PA: K2,000 ~ K4,999; ELC-PV: K2,000 ~ K9,999; ELC2-PV: K2,000~K11,999; latched; default = 2,000

API	Mnemonic	Operands	Function
150	MODRW	S_1, S_2, S_3, S, n	MODBUS Data Read/Write

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			MODRW: 11 steps
S_2					*	*							*			
S_3					*	*							*			
S													*			
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: External device address (K0~K254) **S_2**: Function code K2(H02), K3(H03), K4(H04), K5(H05), K6(H06), K15(H0F), K16(H10) , K23(H17) **S_3**: Data address being read from or written to in the external device **S**: Internal address to store a read message reply or to store write data to be sent **n**: Length of read/write data.

Description:

970. Please refer to the following table for more information about the communication ports supported by MODRW:

COM port	COM1(RS-232)	COM2(RS-485)	COM3(RS-485)
ELC-PA/PV	-	V	-
ELCB-PB	-	V	-

ELCM-PH/PA	V	V	V
ELC2-PB/PA/PV	V	V	-
ELC2-PC	V	V	V
ELC2-PE	-	V	V

971. S_1: Address of the external device the ELC will communicate with. The valid range is K0~K254. The address specified by the function codes K2, K3, K4, and K23 can not be K0.

972. S_2: Function code. Only the function codes listed below are available currently; other function codes are not executable. Please refer to the program examples below for more information.

Function code	Description	Models supported
H02	Read multiple bits devices	All series (except ELCB-PB)
H03	Read multiple word devices	All series
H04	Read multiple word devices	ELCM-PH/PA V2.0 (above) ELC2-PB/PH/PA/PE/PV
H05	Force On/Off a single bit devices	All series (except ELCB-PB)
H06	Write single word device	All series
H0F	Write multiple bit devices	All series (except ELCB-PB)
H10	Write multiple word devices	All series
H17	Read/Wrie multiple word devices	ELCM-PH/PA V2.0 (above) ELC2-PB/PH/PA/PV

973. S_3: Modbus data address in the external device. If the address is illegal for the external device, it will respond with an error message and ELC will store the error code and associated error flag will be ON. If the function code is K23, S_3 only can specify a data register. Besides, S_3 is a data register from which data is read, S_3+1 is a data register into which data is written.

- Associated registers and flags indicating errors on ELC com ports: (For additional details refer to **Points to note** for API 80 RS instruction.)

ELC COM	COM1	COM2	COM3
Error flag	M1315	M1141	M1319
Error code	D1250	D1130	D1253

- If an ELC controller receives a Modbus message with an illegal Modbus data address, the error will be indicated by a different set of flags and registers. For COM2, M1141 will be ON and D1130 = 2; for COM1, M1315 = ON and D1250 = 3, for COM3, M1319 = ON and D1253 = 3.

974. **S**: Registers for storing read/written data. Registers starting from **S** stores the data to be written into the communication device or the data read from the communication device. When COM2 sends the function code of reading (K2/K3/K4), the registers from **S** directly receive the data

string and stored the converted data in D1296~D1311. Please refer to program example 1 and 3 for further explanation. When COM1 or COM3 sends the function code of reading (K2/K3/K4), the registers store the converted data directly. Refer to program example 2 and program example 4 for further explanations. If the function code is K23, the index value in **S** indicates a data register into which data is read, and the index value **S**+1 indicates a data register from which data is written. If the function code for COM2 is K23, the data which is received and converted will not be stored in D1296~D1311. Please refer to program example 13 and program example 14 for more information about the function code K23.

975. **n**: Data length for accessing.

- When $\mathbf{S_2}$ (MODBUS function code) is specified as H05 which designates the PLC force ON/OFF status, **n** = 0 indicates ON and **n** = 1 indicates OFF.
- When $\mathbf{S_2}$ is specified as H02, H03, H04, H0F, H10, H17 which designate the data length for accessing, the available set range will be K1~Km, where m value should be specified according to communication modes and COM ports as the table below. (H02/H0F, unit: Bit. H03/H04/H10/H17, unit: Word.) If the function code is H17, n is the number of data registers from which data is read, n+1 is the number of data registers into which data is written.

Communication mode	**Communication port**	**Function code**				
		H02	H03/H04	H0F	H10	H17
RTU	COM1(RS-232)	K 64	K 16	K 64	K 16	K 16
	COM2(RS-485)	K 64	K 16	K 64	K 16	K 16
	COM3(RS-485)	K 64	K 16	K 64	K 16	K 16
ASCII	COM1(RS-232)	K 64	K 16	K 64	K 16	K 16
	COM2(RS-485)	K 64	K 8	K 64	K 8	K 16
	COM3(RS-485)	K 64	K 16	K 64	K 16	K 16

976. The functions of $\mathbf{S_3}$, **S**, and **n** vary with the function code used.

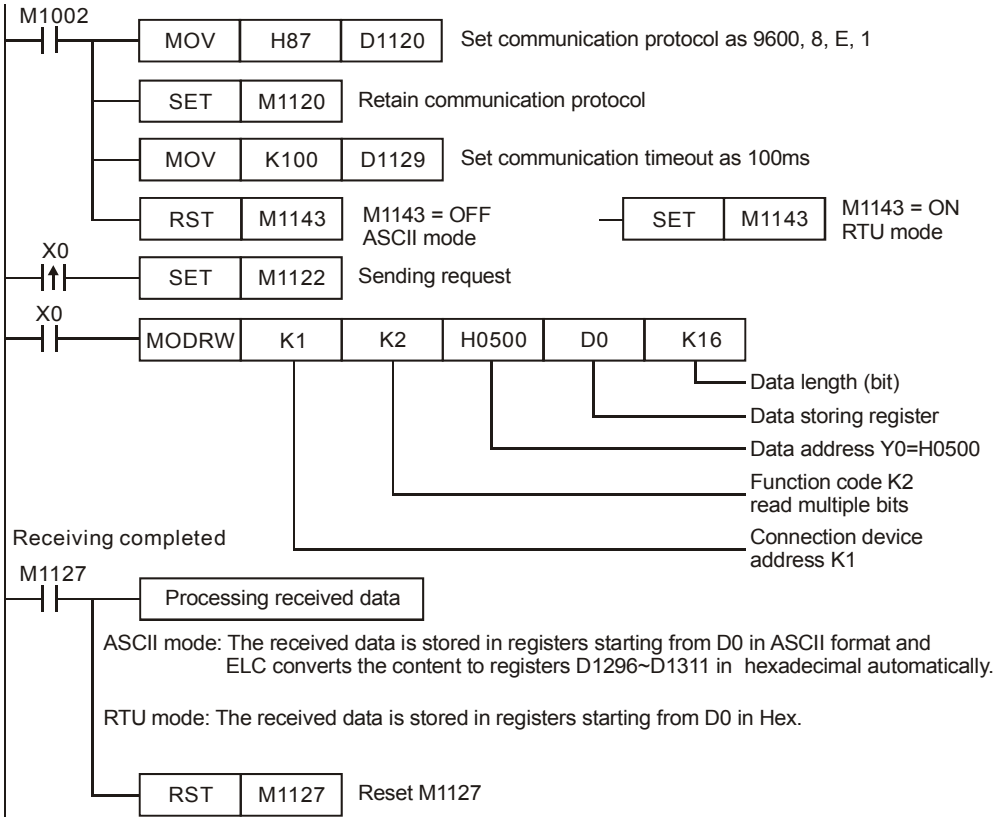
Function code	S_3	S	n
H02	Address from which the data is read	Register in which the data read is stored	Length of data read
H03	Address from which the data is read	Register in which the data read is stored	Length of data read
H04	Address from which the data is read	Register in which the data read is stored	Length of data read
H05	Address into which the data is written	No meaning	Status value written

Function code	S_3	S	n
H06	Address into which the data is written	Register in which the data written is stored	No meaning
H0F	Address into which the data is written	Register in which the data written is stored	Length of data written
H10	Address into which the data is written	Register in which the data written is stored	Length of data written
H17	**S_3**: Address from which the data is read **S_3**+1: Address into which the data is written	**S**: Register in which the data read is stored **S**+1: Register in which the data written is stored	**n**: Length of data read **n**+1: Length of data written

977. There is no limit on the number of times this instruction can be used, however only one instruction can be executed on the same COM port at a time.
978. Rising-edge contacts (LDP, ANDP, ORP) and falling-edge contacts (LDF, ANDF, ORF) can not be used as conditions for the MODRW instruction, using Function Codes H02, H03, H04, H17 or the data stored in the receive registers will be incorrect.
979. The following bits need to be energized after the MODRW instruction is executed to send the message: M1122(COM2) / M1312(COM1) / M1316(COM3).
980. MODRW instruction determines the COM port according to the communication request. The COM port determination is made following the order: COM1→COM3→COM2. Therefore, please insert every MODRW instruction right after the sending request instruction for avoiding errors on the target location for data access.
981. Flags: M1120~M1131, M1140~M1143, refer to the RS (API 80) instruction or section 2.9 for information.

Program Example 1: COM2(RS-485), Function Code H02

982. Function code K2 (H02): read multiple bit addresses, up to 64 bits can be read.
983. M1143 = OFF, ASCII mode, M1143 = ON, RTU Mode
984. In ASCII or RTU mode, when the ELC's COM2 sends a read message, the data in the reply will be stored in D1296~D1311. The data sent with a write message will be stored in registers D1256~D1295 (the data should be placed into the D-register address in the instruction).
985. The tables below show the status of the outputs when ELC1 reads Y0~Y17 from ELC2.



- ASCII Mode (M1143 = OFF):

When X0 = ON, MODRW instruction executes the function specified by Function Code 02.

ELC1⇨ ELC2，ELC1 sends: “01 02 0500 0010 E8”

ELC2 ⇨ELC1，ELC1 receives: “01 02 02 3412 B5”

Registers for data to be sent

Register	Data		Descriptions	
D1256 Low	‘0’	30 H	ADR 1	Device address: ADR (1,0)
D1256 High	‘1’	31 H	ADR 0	
D1257 Low	‘0’	30 H	CMD 1	Control parameter: CMD (1,0)
D1257 High	‘2’	32 H	CMD 0	
D1258 Low	‘0’	30 H	Y0 = H0500 Starting Data Address	
D1258 High	‘5’	35 H		
D1259 Low	‘0’	30 H		
D1259 High	‘0’	30 H		

Register	Data		Descriptions
D1260 Low	‘0’	30 H	Number of Data bits
D1260 High	‘0’	30 H	

D1261 Low	'1'	31 H		
D1261 High	'0'	30 H		
D1262 Low	'E'	45 H	LRC CHK 1	Checksum: LRC CHK (0,1)
D1262 High	'8'	38 H	LRC CHK 0	

Registers for received data

Register	Data		Descriptions	
D0 Low	'0'	30 H	ADR 1	
D0 High	'1'	31 H	ADR 0	
D1 Low	'0'	30 H	CMD 1	
D1 High	'2'	33 H	CMD 0	
D2 Low	'0'	30 H	Number of Data (count by Byte)	
D2 High	'2'	32 H		
D3 Low	'3'	33 H	Content of address 0500H~ 0515H	1234 H ELC automatically converts ASCII codes and store the converted value in D1296
D3 High	'4'	34 H		
D4 Low	'1'	31H		
D4 High	'2'	32H		
D5 Low	'B'	52H	LRC CHK 1	
D5 High	'5'	35 H	LRC CHK 0	

State of the bits read from ELC2 Y0~Y17: 1234H

Device	Status	Device	Status	Device	Status	Device	Status
Y0	OFF	Y1	OFF	Y2	ON	Y3	OFF
Y4	ON	Y5	ON	Y6	OFF	Y7	OFF
Y10	OFF	Y11	ON	Y12	OFF	Y13	OFF
Y14	ON	Y15	OFF	Y16	OFF	Y17	OFF

- RTU Mode (M1143 = ON):

When X0 = ON, MODRW instruction sends the message using Function Code 02

ELC1⇨ ELC2，ELC1sends: “01 02 0500 0010 79 0A”

ELC2 ⇨ ELC1，ELC1receives: “01 02 02 34 12 2F 75”

Registers for data to be sent

Register	Data	Descriptions
D1256 Low	01 H	Address
D1257 Low	02 H	Function
D1258 Low	05 H	Y0 = H0500
D1259 Low	00 H	Starting Data Address
Register	**Data**	**Descriptions**
D1260 Low	00 H	Number of Data (count by word)
D1261 Low	10 H	
D1262 Low	79 H	CRC CHK Low
D1263 Low	0A H	CRC CHK High

Received data registers

Register	Data	Descriptions	
D0	1234 H	ELC stores the value 1234H into D1296	
D1 Low	02 H	Function	
D2 Low	02 H	Number of Data (Byte)	
D3 Low	34 H	Content of address	
D4 Low	12 H	H0500~H0515	
D5 Low	2F H	CRC CHK Low	
D6 Low	75 H	CRC CHK High	

State of the bits read from ELC2 Y0~Y17: 1234H

Device	Status	Device	Status	Device	Status	Device	Status
Y0	OFF	Y1	OFF	Y2	ON	Y3	OFF

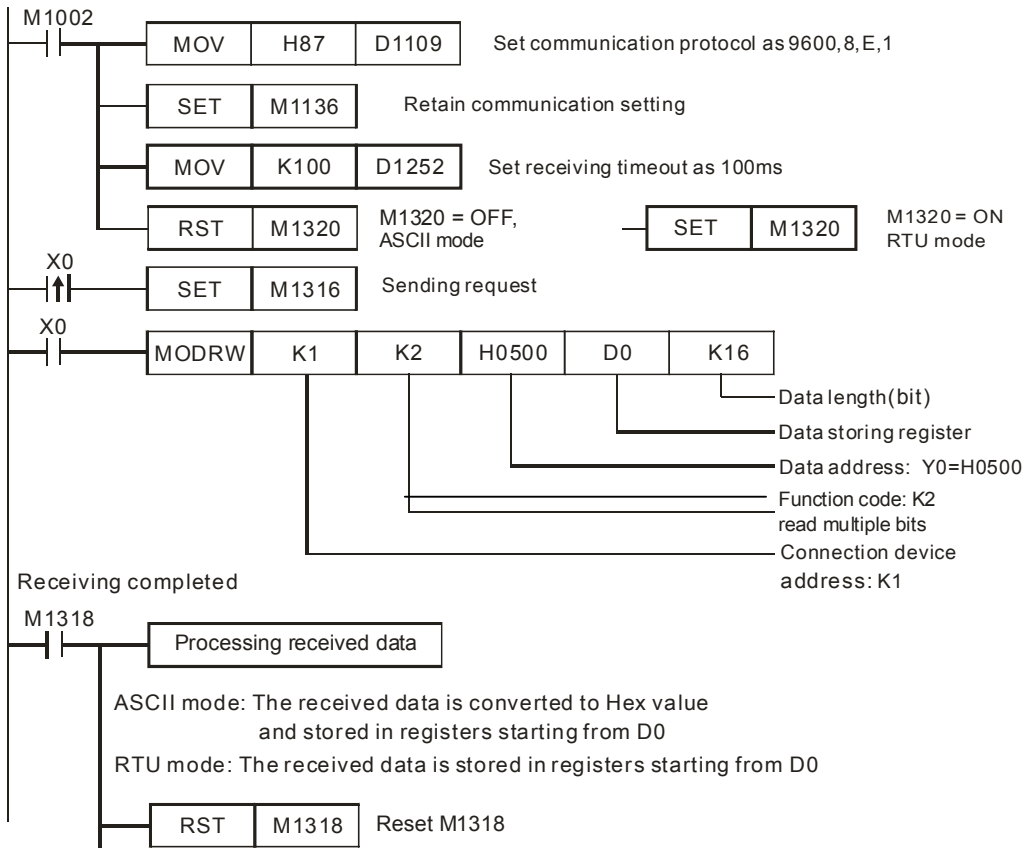
Device	Status	Device	Status	Device	Status	Device	Status
Y4	ON	Y5	ON	Y6	OFF	Y7	OFF
Y10	OFF	Y11	ON	Y12	OFF	Y13	OFF
Y14	ON	Y15	OFF	Y16	OFF	Y17	OFF

Program Example 2: COM1(RS-232) / COM3(RS-485), Function Code H02

1. Function code K2 (H02): read multiple bit devices. Up to 64 bits can be read.
2. ELC1 connects to ELC2. COM3: (M1320 = OFF, ASCII mode), (M1320 = ON, RTU mode)
3. For both ASCII and RTU modes, ELC COM1/COM3 only stores the received data in registers starting from **S**, and does not store the data to be sent.
4. The connection between ELC1 (ELC COM3) and ELC2(ELC COM1) is shown in the tables below when ELC1 reads Y0~Y17 from ELC2

   If ELC1 uses COM1 for communication rather than COM3, change the following parameters in the program below:

   1. D1109→D1036: communication protocol
   2. M1136→M1138: retain communication setting
   3. D1252→D1249: Set value for data receiving timeout
   4. M1320→M1139: ASCII/RTU mode selection
   5. M1316→M1312: sending request
   6. M1318→M1314: receiving completed flag



ASCII mode (COM3: M1320 = OFF, COM1: M1139 = OFF):

When X0 = ON, the MODRW instruction sends the message using Function Code 02.

ELC1⇨ ELC2, ELC1 sends: “01 02 0500 0010 E8”

ELC2 ⇨ELC1, ELC1 receives: “01 02 02 3412 B5”

ELC1 receives data in register D0

Register	Data	Descriptions
D0	1234H	ELC converts the ASCII data in address 0500H~0515H and stores the converted data automatically.

State of the bits read from ELC2 Y0~Y17: 1234H

Device	Status	Device	Status	Device	Status	Device	Status
Y0	OFF	Y1	OFF	Y2	ON	Y3	OFF
Y4	ON	Y5	ON	Y6	OFF	Y7	OFF
Y10	OFF	Y11	ON	Y12	OFF	Y13	OFF
Y14	ON	Y15	OFF	Y16	OFF	Y17	OFF

RTU mode (COM3: M1320 = ON, COM1: M1139 = ON):

When X0 = ON, MODRW instruction sends the message using Function Code 02

ELC1⇨ ELC2, ELC1 sends: “01 02 0500 0010 79 0A”

ELC2 ⇨ ELC1, ELC1 receives: “01 02 02 34 12 2F 75”

ELC data receiving register:

Register	Data	Descriptions
D0	1234 H	ELC converts the data in address 0500H ~ 0515H and stores the converted data automatically.

State of the bits read from ELC2 Y0~Y17: 1234H

Device	Status	Device	Status	Device	Status	Device	Status
Y0	OFF	Y1	OFF	Y2	ON	Y3	OFF
Y4	ON	Y5	ON	Y6	OFF	Y7	OFF
Y10	OFF	Y11	On	Y12	OFF	Y13	OFF
Y14	ON	Y15	OFF	Y16	OFF	Y17	OFF

5. Flags and data registers for COM1 / COM2 / COM3 when they act as a Modbus Master:

	COM2	COM1	COM3	Function
COM. setting	M1120	M1138	M1136	Retain communication setting
	M1143	M1139	M1320	ASCII/RTU mode selection
	D1120	D1036	D1109	Communication protocol
	D1121	D1121	D1255	ELC communication address
Sending request	M1122	M1312	M1316	Sending request
	D1129	D1249	D1252	Set value for data receiving timeout (ms)
Receiving completed	M1127	M1314	M1318	Data receiving completed
Errors	-	M1315	M1319	Data receiving error
	-	D1250	D1253	Communication error code
	M1129	-	-	Receiving timeout
	M1140	-	-	Data receiving error
	M1141	-	-	Parameter error. Exception Code is stored in D1130
	D1130	-	-	Error code (Exception code) returning from Modbus communication

Program Example 3: COM2 (RS-485), Function Code H03 (The function code H04 is the same as the function code H03.)

986. Function code K3 (H03): read multiple registers. For COM2 ASCII mode, only 8 words can be read.

ELC connects to a drive on Modbus. (ASCII Mode, when M1143=OFF)

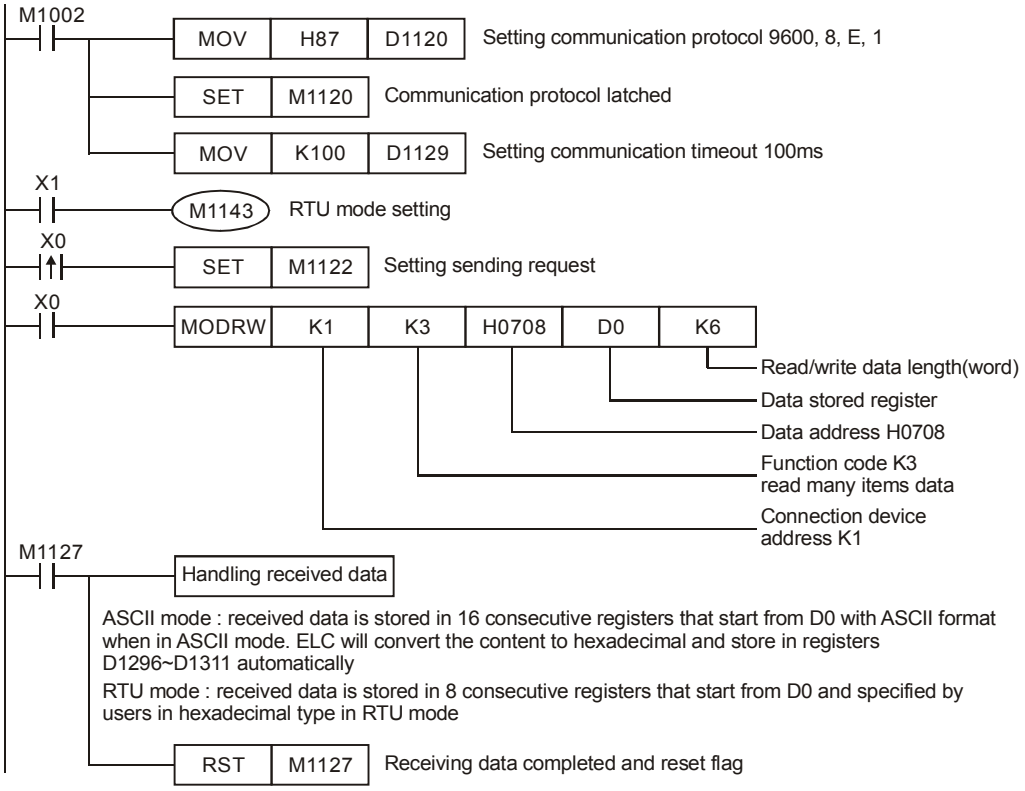
ELC connects to a drive on Modbus. (RTU Mode, when M1143=ON)

987. For ASCII or RTU mode, ELC COM2 stores the data to be sent in D1256~D1295, and stores the command reply in registers starting from **S**, and stores the converted 16-bit data in D1296 ~ D1311. M1131=ON when ELC starts converting to hexadecimal and M1131 = OFF after the conversion is complete.

988. Use MOV, DMOV or BMOV instructions to move the hexadecimal data from D1296~D1311 and store in D-registers.

989. After receiving data, the ELC will automatically check if the received data is correct. If there is any fault, M1140 = ON and the fault code will be stored in D1130.

990. After M1140=ON or M1141=ON, the ELC will re-transmit the data to the drive. If the received data is correct, M1140 and M1141 will be reset.



- ASCII Mode (M1143=OFF): ELC connects to the drive.

  ELC → the drive, ELC transmits: “ 01 03 0708 0006 E7 ”

  MVX → the drive, ELC receives: “ 01 03 OC 0100 1766 0000 0000 0136 0000 3B ”

ELC transmits data register (transmit message)

Register	Data		Descriptions	
D1256 Low byte	'0'	30 H	ADR 1	ADR (1,0) is MVX drive address
D1256 High byte	'1'	31 H	ADR 0	
D1257 Low byte	'0'	30 H	CMD 1	CMD (1,0) is instruction code
D1257 High byte	'3'	33 H	CMD 0	
D1258 Low byte	'0'	30 H	Data Address	
D1258 High byte	'7'	37 H		
D1259 Low byte	'0'	30 H		
D1259 High byte	'8'	38 H		
D1260 Low byte	'0'	30 H	Number of data (count by word)	
D1260 High byte	'0'	30 H		
D1261 Low byte	'0'	30 H		
D1261 High byte	'6'	36 H		
D1262 Low byte	'E'	45 H	LRC CHK 1	LRC CHK (0,1) error check
D1262 High byte	'7'	37 H	LRC CHK 0	

ELC receive data register (response message)

Register	Data		Descriptions	
D0 low byte	'0'	30 H	ADR 1	
D0 high byte	'1'	31 H	ADR 0	
D1 low byte	'0'	30 H	CMD 1	
D1 high byte	'3'	33 H	CMD 0	
D2 low byte	'0'	30 H	Number of data (count by byte)	
D2 high byte	'C'	43 H		
D3 low byte	'0'	30 H	Content of address 0708 H	ELC automatically converts ASCII codes to hex and store the converted value in D1296 = 0100 H
D3 high byte	'1'	31 H		
D4 low byte	'0'	30 H		
D4 high byte	'0'	30 H		
D5 low byte	'1'	31 H	Content of address 0709 H	ELC automatically converts ASCII codes to hex and store the converted value in D1297 = 1766 H
D5 high byte	'7'	37 H		
D6 low byte	'6'	36 H		
D6 high byte	'6'	36 H		
D7 low byte	'0'	30 H	Content of address 070A H	ELC automatically converts ASCII codes to hex and store the converted value in D1298
D7 high byte	'0'	30 H		
D8 low byte	'0'	30 H		

D8 high byte	'0'	30 H		= 0000 H

Register	Data		Descriptions	
D9 low byte	'0'	30 H	Content of address 070B H	ELC automatically converts ASCII codes to hex and store the converted value in D1299 = 0000 H
D9 high byte	'0'	30 H		
D10 low byte	'0'	30 H		
D10 high byte	'0'	30 H		
D11 low byte	'0'	30 H	Content of address 070C H	ELC automatically converts ASCII codes to hex and store the converted value in D1300 = 0136 H
D11 high byte	'1'	31 H		
D12 low byte	'3'	33 H		
D12 high byte	'6'	36 H		
D13 low byte	'0'	30 H	Content of address 070D H	ELC automatically converts ASCII codes to hex and store the converted value in D1301 = 0000 H
D13 high byte	'0'	30 H		
D14 low byte	'0'	30 H		
D14 high byte	'0'	30 H		
D15 low byte	'3'	33 H	LRC CHK 1	
D15 high byte	'B'	42 H	LRC CHK 0	

- RTU Mode (M1143=ON): ELC connects to the drive

  ELC → the drive, ELC transmits: 01 03 0708 0006 45 7E

  The drive → ELC, ELC receives: 01 03 0C 0000 0503 0BB8 0BB8 0000 012D 8E C5

ELC transmit data

Register	Data	Descriptions
D1256 Low byte	01 H	Address
D1257 Low byte	03 H	Function
D1258 Low byte	07 H	Data Address
D1259 Low byte	08 H	
D1260 Low byte	00 H	Number of data (count by word)
D1261 Low byte	06 H	
D1262 Low byte	45 H	CRC CHK Low
D1263 Low byte	7E H	CRC CHK High

ELC receive data

Register	Data	Descriptions	
D0 low byte	01 H	Address	
D1 low byte	03 H	Function	
D2 low byte	0C H	Number of data (count by byte)	
D3 low byte	00 H	Content of address 0708 H	ELC automatically store the value in D1296 = 0000 H
D4 low byte	00 H		
D5 low byte	05 H	Content of address 0709 H	ELC automatically store the value in D1297 = 0503 H
D6 low byte	03 H		
D7 low byte	0B H	Content of address 070A H	ELC automatically store the value in D1298 = 0BB8 H
D8 low byte	B8 H		
D9 low byte	0B H	Content of address 070B H	ELC automatically store the value in D1299 = 0BB8 H
D10 low byte	B8 H		
D11 low byte	00 H	Content of address 070C H	ELC automatically store the value in D1300 = 0000 H
D12 low byte	00 H		
D13 low byte	01 H	Content of address 070D H	ELC automatically store the value in D1301 = 012D H
D14 low byte	2D H		
D15 low byte	8E H	CRC CHK Low	
D16 low byte	C5 H	CRC CHK High	

Program example 4: COM1(RS-232) / COM3(RS-485), Function Code H03 (The function code H04 is the same as the function code H03.)

991. Function code K3 (H03): read multiple Words, up to 16 words can be read.
992. ELC COM1 / COM3 stores the received data in registers starting from **S**.
993. The data sent between the ELC and the drive, is shown in the tables below.
994. If the ELC uses COM1 rather than COM2, the program below would be changed as follows:

D1109→D1036: Communication protocol

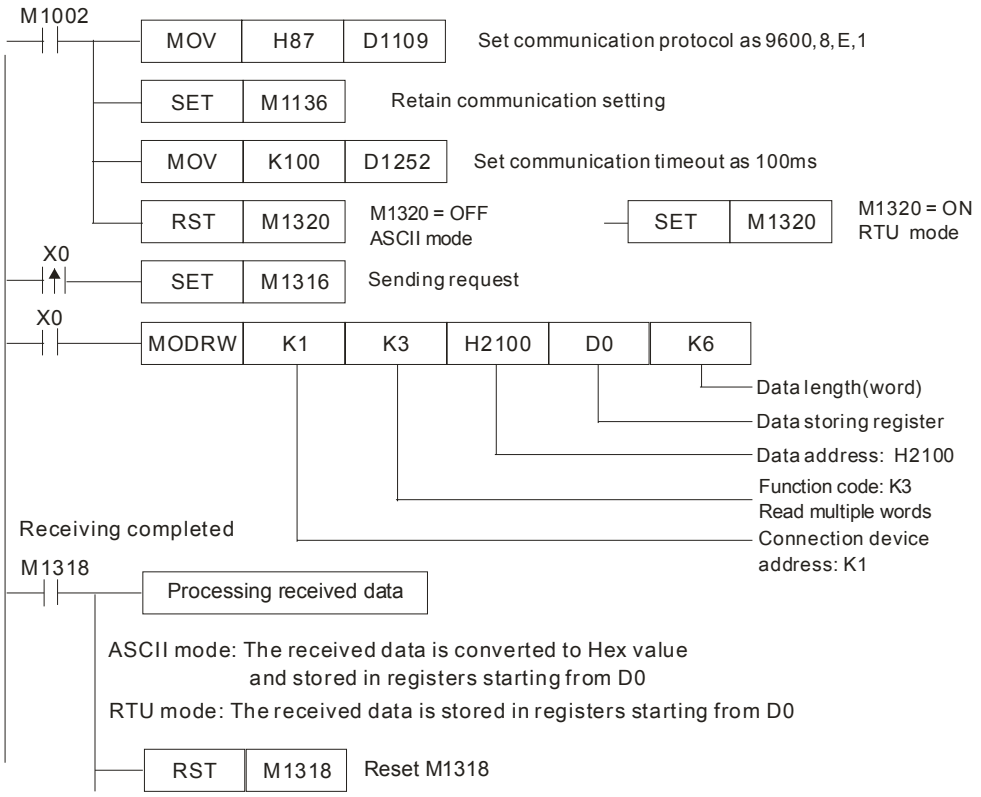
M1136→M1138: Retain communication setting

D1252→D1249: Set value for data receiving timeout

M1320→M1139: ASCII/RTU mode selection

M1316→M1312: Sending request

M1318→M1314: Receiving completed flag



- ASCII mode (COM3: M1320 = OFF, COM1: M1139 = OFF):

  When X0 = ON, MODRW instruction executes the function specified by Function Code 03

  ELC ⇨ the drive , ELC sends: " 01 03 0708 0006 E7 "

  The drive⇨ ELC, ELC receives: "01 03 0C 0100 1766 0000 0000 0136 0000 3B"

Registers for receive data

Register	Data	Descriptions
D0	0100 H	ELC converts ASCII codes in 0708 H and stores the converted data automatically.
D1	1766 H	ELC converts ASCII codes in 0709 H and stores the converted data automatically.
D2	0000 H	ELC converts ASCII codes in 070A H and stores the converted data automatically.
D3	0000 H	ELC converts ASCII codes in 070B H and stores the converted data automatically.
D4	0136 H	ELC converts ASCII codes in 070C H and stores the converted data automatically.

Register	Data	Descriptions
D5	0000 H	ELC converts ASCII codes in 070D H and stores the converted data automatically.

- RTU mode (COM3: M1320 = ON COM1: M1139 = ON):

  When X0 = ON, MODRW instruction executes the function specified by Function Code 03

  ELC ⇨ the drive, ELC sends: ” 01 03 0708 0006 45 7E”

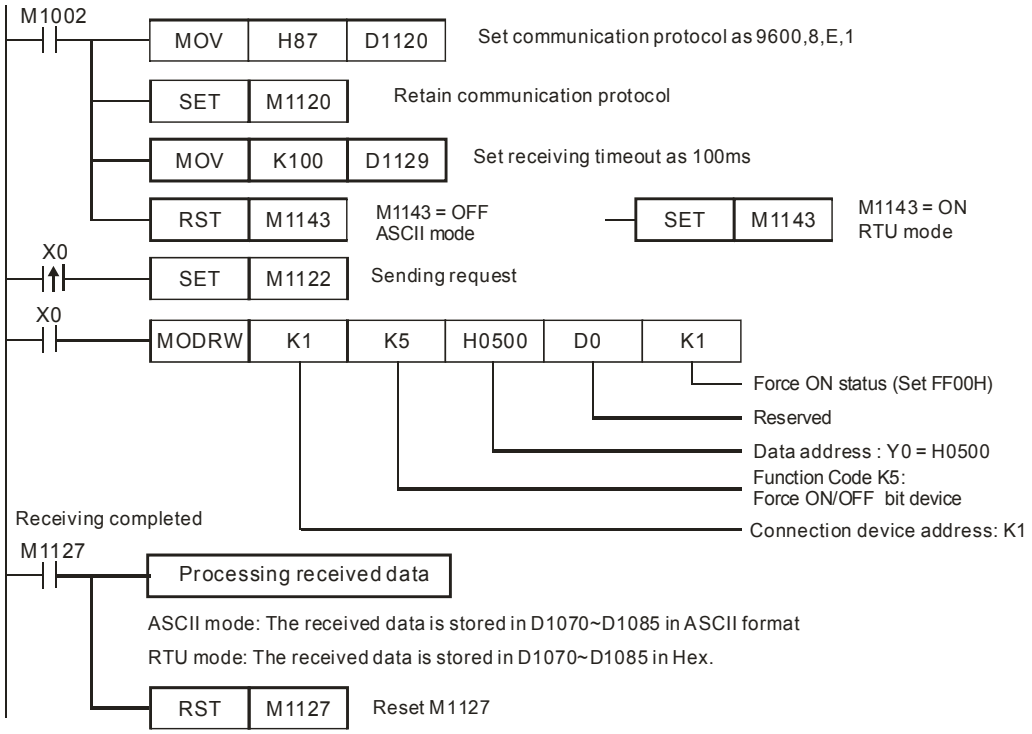
  The drive⇨ ELC, ELC receives: “01 03 0C 0000 0503 0BB8 0BB8 0000 012D 8E C5”

  Registers for received data

Register	Data	Descriptions
D0	0000 H	ELC converts data in 0708 H and stores the converted data automatically.
D1	0503 H	ELC converts data in 0709 H and stores the converted data automatically.
D2	0BB8 H	ELC converts data in 070A H and stores the converted data automatically.
D3	0BB8 H	ELC converts data in 070B H and stores the converted data automatically.
D4	0136 H	ELC converts data in 070C H and stores the converted data automatically.
D5	012D H	ELC converts data in 070D H and stores the converted data automatically.

Program example 5: COM2(RS-485), Function Code H05

995. Function code K5(H05): Force ON/OFF bit device
996. ELC1 connects to ELC2: (M1143 = OFF, ASCII mode), (M1143 = ON, RTU Mode)
997. **n** = 1 indicates Force ON (set FF00H) and **n** = 0 indicates Force OFF (set 0000H)
998. For ASCII or RTU mode, ELC COM2 stores the data to be sent in D1256~D1295 and stores the received data in D1070~D1085
999. Take the connection between ELC1 (ELC COM2) and ELC2 (ELC COM1) for example, the tables below explain the status when ELC1 Force ON ELC2 Y0.



- ASCII mode (M1143 = OFF):

  When X0 = ON, MODRW instruction executes the function specified by Function Code 05

  ELC1 ⇨ ELC2, ELC sends: “01 05 0500 FF00 6F”

  ELC2 ⇨ ELC1, ELC receives: “01 05 0500 FF00 6F”

Data to be sent

Register	Data		Descriptions	
D1256 low byte	‘0’	30 H	ADR 1	Device address: ADR (1,0)
D1256 high byte	‘1’	31 H	ADR 0	
D1257 low byte	‘0’	30 H	CMD 1	CMD (1,0) Control parameter
D1257 high byte	‘5’	35H	CMD 0	
D1258 low byte	‘0’	30 H	Data Address	
D1258 high byte	‘5’	35 H		
D1259 low byte	‘0’	30 H		
D1259 high byte	‘0’	30 H		
D1260 low byte	‘F’	46 H	High byte to be force ON/OFF	
D1260 high byte	‘F’	46 H		
D1261 low byte	‘0’	30H	Low byte to be force ON/OFF	

Register	Data		Descriptions	
D1261 high byte	‘0’	30 H		
D1262 low byte	‘6’	36 H	LRC CHK 1	Checksum: LRC CHK (0,1)
D1262 high byte	‘F’	46 H	LRC CHK 0	

Received data

Register	Data		Descriptions
D1070 low byte	‘0’	30 H	ADR 1
D1070 high byte	‘1’	31 H	ADR 0
D1071 low byte	‘0’	30 H	CMD 1
D1071 high byte	‘5’	35H	CMD 0
D1072 low byte	‘0’	30 H	Data Address
D1072 high byte	‘5’	35 H	
D1073 low byte	‘0’	30 H	
D1073 high byte	‘0’	30 H	
D1074 low byte	‘F’	46 H	High byte to be force ON/OFF
D1074 high byte	‘F’	46 H	
D1075 low byte	‘0’	30H	Low byte to be force ON/OFF
D1075 high byte	‘0’	30 H	
D1076 low byte	‘6’	36 H	LRC CHK 1
D1076 high byte	‘F’	46 H	LRC CHK 0

- RTU mode (M1143 = ON)

  When X0 = ON, MODRW instruction executes the function specified by Function Code 05

  ELC1⇨ ELC2, ELC1 sends: “01 05 0500 FF00 8C F6”

  ELC2 ⇨ELC1, ELC1 receives: “01 05 0500 FF00 8C F6”

Data to be sent

Register	Data	Descriptions
D1256 Low byte	01 H	Address
D1257 Low byte	05 H	Function
D1258 Low byte	05 H	Data Address
D1259 Low byte	00 H	
D1260 Low byte	FF H	Data content (ON = FF00H)
D1261 Low byte	00 H	

Register	Data	Descriptions
D1262 Low byte	8C H	CRC CHK Low
D1263 Low byte	F6 H	CRC CHK High

Received data

Register	Data	Descriptions
D1070 Low byte	01 H	Address
D1071 Low byte	05 H	Function
D1072 Low byte	05 H	Data Address
D1073 Low byte	00 H	
D1074 Low byte	FF H	Data content (ON = FF00H)
D1075 Low byte	00 H	
D1076 Low byte	8C H	CRC CHK Low
D1077 Low byte	F6 H	CRC CHK High

Program example 6: COM1(RS-232) / COM3(RS-485), Function Code H05

1000. Function Code K5 (H05): Force ON/OFF bit.

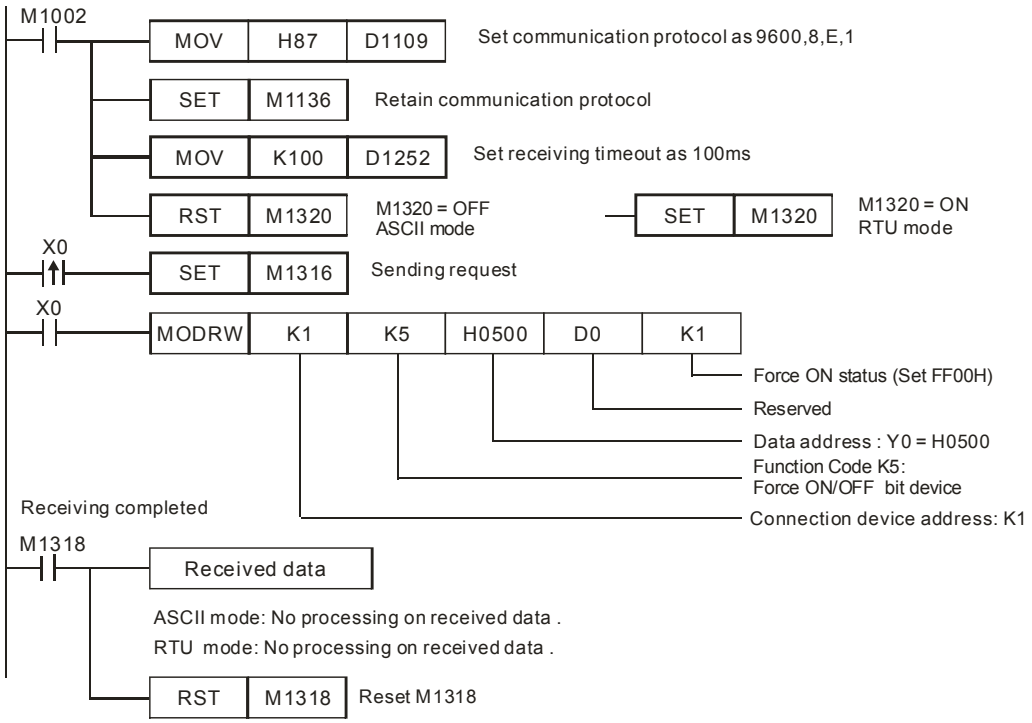
1001. M1320 = OFF, ASCII Mode , M1320 = ON, RTU Mode

1002. **n** = 1 indicates Force ON (set FF00H) and **n** = 0 indicates Force OFF (set 0000H)

1003. The messages sent between ELC1 (ELC COM3) and ELC2(ELC COM1) are shown in the tables below.

1004. If ELC1 uses COM1 for communications, the program below can be usable by changing:

1. D1109→D1036: communication protocol
2. M1136→M1138: retain communication setting
3. D1252→D1249: Set value for data receiving timeout
4. M1320→M1139: ASCII/RTU mode selection
5. M1316→M1312: sending request
6. M1318→M1314: receiving completed flag



- ASCII mode (COM3: M1320 = OFF, COM1: M1139 = OFF):
  When X0 = ON, MODRW instruction executes the function specified by Function Code 05
  ELC1 ⇨ ELC2, ELC sends: “01 05 0500 FF00 6F”
  ELC2 ⇨ ELC1, ELC receives: “01 05 0500 FF00 6F”
  (No data processing on received data)
- RTU mode (COM3: M1320 = ON, COM1: M1139 = ON):
  When X0 = ON, MODRW instruction executes the function specified by Function Code 05
  ELC1⇨ ELC2, ELC1 sends: “01 05 0500 FF00 8C F6”
  ELC2 ⇨ELC1, ELC1 receives: “01 05 0500 FF00 8C F6”
  (No data processing on received data)

Program Example 7: COM2(RS-485), Function Code H06

1005. Function code K6(H6) : write one data word
ASCII Mode when M1143=OFF
RTU Mode when M1143=ON

1006. When in ASCII mode, store the data that will be written to the drive in ASCII format in register D50. Data received from the drive will be stored in registers D1070~D1076.

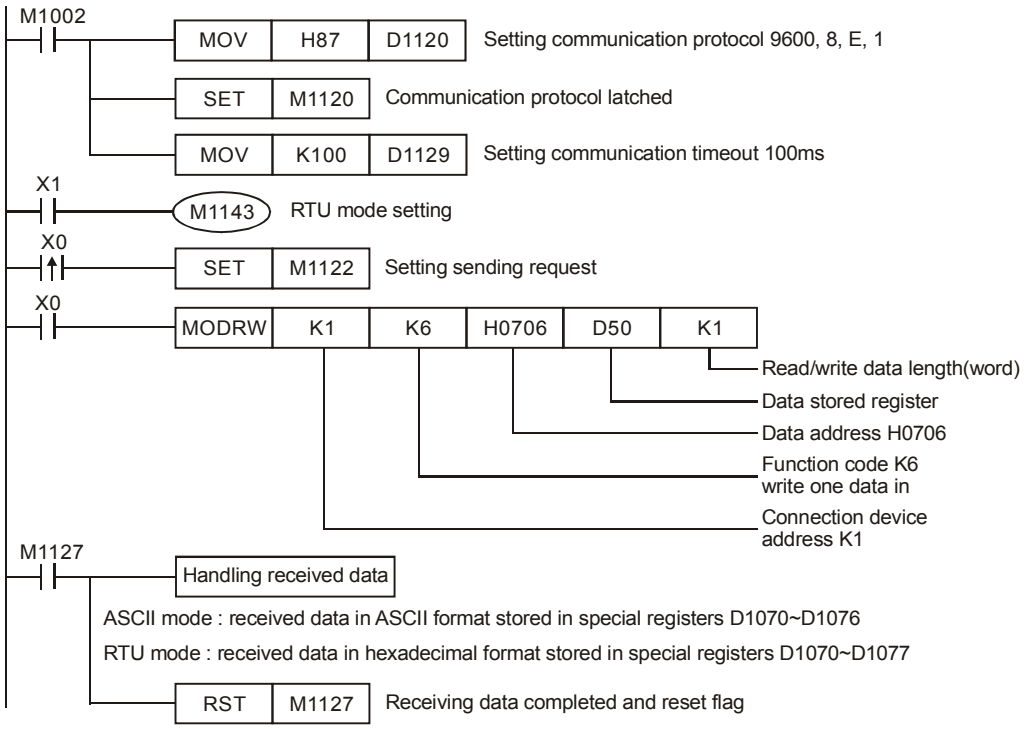
1007. When in RTU mode, store data that will be written to the drive in hexadecimal format in register D50. Data received from the drive will be stored in registers D1070~D1077.

1008. When in ASCII or RTU mode, the ELC will transmit the data stored in registers D1256~D1295.

1009. After receiving a response, the ELC automatically checks if the received data is correct. If there is any fault, M1140 will be set to ON.

1010. If the register address of the drive is illegal, a fault code will be stored in D1130 and M1141 = ON. For example, 8000H is illegal for the drive and M1141=ON and D1130=2. Refer to MVX user manual to fault code.

1011. After M1140=ON or M1141=ON, the ELC will re-transmit the same data to the drive. If the received data is correct, M1140 and M1141 will be reset.



- ASCII Mode (M1143=OFF): ELC connects to the drive.
  ELC → the drive, ELC transmits: " 01 06 0706 1770 65 "
  The drive → ELC, ELC receives: " 01 06 0706 1770 65 "

ELC transmit data

Register	Data		Descriptions	
D1256 Low byte	'0'	30 H	ADR 1	ADR (1,0) is MVX drive address
D1256 High byte	'1'	31 H	ADR 0	
D1257 Low byte	'0'	30 H	CMD 1	CMD (1,0) is function code
D1257 High byte	'6'	36 H	CMD 0	
D1258 Low byte	'0'	30 H	Data Address	
D1258 High byte	'7'	37 H		

D1259 Low byte	'0'	30 H		
D1259 High byte	'6'	36 H		
D1260 Low byte	'1'	31 H		
D1260 High byte	'7'	37 H	Data contents	The content of register D50 (H1770=K6000)
D1261 Low byte	'7'	37 H		
D1261 High byte	'0'	30 H		
D1262 Low byte	'6'	36 H	LRC CHK 1	LRC CHK (0,1) is error check
D1262 High byte	'5'	35 H	LRC CHK 0	

Data received

Register	Data		Descriptions
D1070 Low byte	'0'	30 H	ADR 1
D1070 High byte	'1'	31 H	ADR 0
D1071 Low byte	'0'	30 H	CMD 1
D1071 High byte	'6'	36 H	CMD 0
D1072 Low byte	'0'	30 H	Data Address
D1072 High byte	'7'	37 H	
D1073 Low byte	'0'	30 H	
D1073 High byte	'6'	36 H	
D1074 Low byte	'1'	31 H	Data content
D1074 High byte	'7'	37 H	
D1075 Low byte	'7'	37 H	
D1075 High byte	'0'	30 H	
D1076 Low byte	'6'	36 H	LRC CHK 1
D1076 High byte	'5'	35 H	LRC CHK 0

- RTU Mode (M1143=ON): ELC connects to the drive

  ELC → the drive, ELC transmits: 01 06 0706 1770 66 AB

  The drive → ELC, ELC receives: 01 06 0706 1770 66 AB

ELC transmit data

Register	Data	Descriptions	
D1256 Low byte	01 H	Address	
D1257 Low byte	06 H	Function	
D1258 Low byte	07 H	Data Address	
D1259 Low byte	06 H		
D1260 Low byte	17 H	Data content	The content of register D50 (H1770=K6000)
D1261 Low byte	70 H		
D1262 Low byte	66 H	CRC CHK Low	
D1263 Low byte	AB H	CRC CHK High	

Data received

Register	Data	Descriptions
D1070 Low byte	01 H	Address
D1071 Low byte	06 H	Function
D1072 Low byte	07 H	Data Address
D1073 Low byte	06 H	
D1074 Low byte	17 H	Data content
D1075 Low byte	70 H	
D1076 Low byte	66 H	CRC CHK Low
D1077 Low byte	AB H	CRC CHK High

Program example 8: COM1 (RS-232) / COM3 (RS-485), Function Code H06

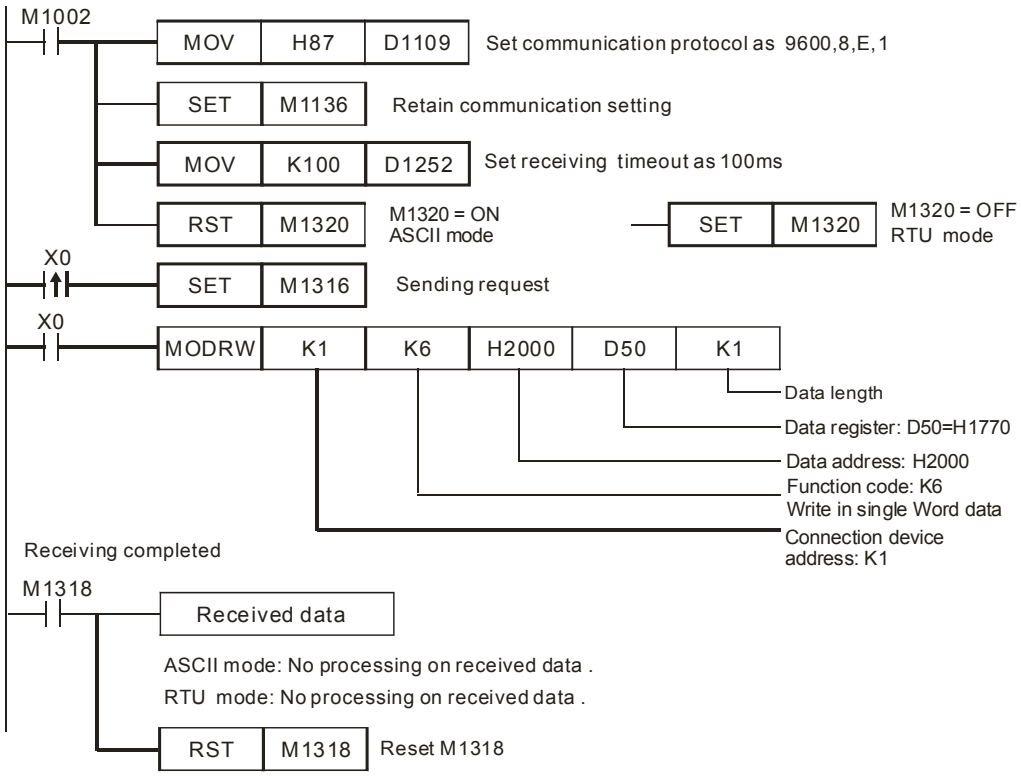
1012. Function code K6 (H06): Write single Word.

1013. Set the value to be written to the drive in the register specified by operand **S**.

1014. The messages sent between ELC (ELC COM3) and the drive is shown in the tables below.

1015. If ELC uses COM1 for communications, the program below can be usable by changing:

1. D1109→D1036: communication protocol
2. M1136→M1138: retain communication setting
3. D1252→D1249: Set value for data receiving timeout
4. M1320→M1139: ASCII/RTU mode selection
5. M1316→M1312: sending request
6. M1318→M1314: receiving completed flag



- ASCII mode (COM3: M1320 = OFF, COM1: M1139 = OFF):

  When X0 = ON, MODRW instruction executes the function specified by Function Code 06

  ELC ⇨ the drive , ELC sends: “01 06 0706 1770 65”

  The drive ⇨ ELC, ELC receives: “01 06 0706 1770 65”

  (No data processing on received data)

- RTU mode (COM3: M1320 = ON, COM1: M1139 = ON)

  When X0 = ON, MODRW instruction executes the function specified by Function Code 06

  ELC ⇨ the drive, ELC sends: “01 06 0706 1770 66 AB”

  The drive → ELC, ELC receives: “01 06 0706 1770 66 AB”

  (No data processing on received data)

Program Example 9: COM2 (RS-485), Function Code H0F

1016. Function code K15 (H0F): write multiple bits. Up to 64bits can be written.

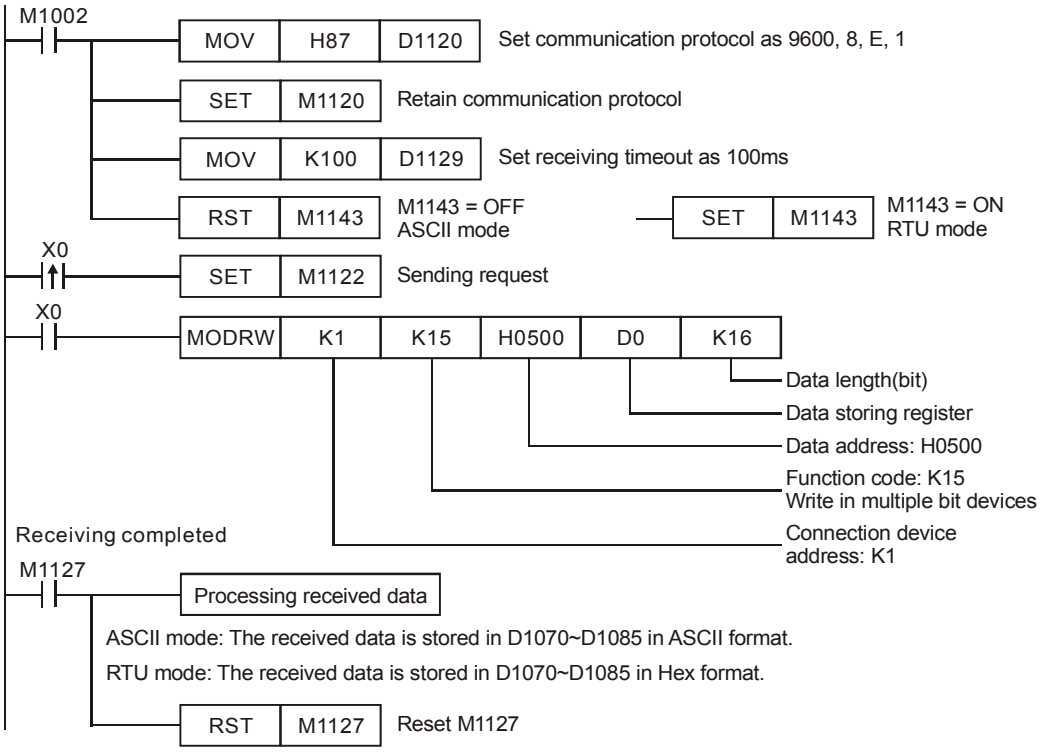
1017. M1143 = OFF, ASCII Mode, M1143 = ON, RTU Mode

1018. For ASCII or RTU mode, ELC COM2 stores the data to be sent in D1256~D1295 and the received data in D1070~D1085.

1019. The messages for ELC1 (ELC COM2) and ELC2 (ELC COM1) are shown in the tables below.

Set value: K4Y0=1234H

Device	Status	Device	Status	Device	Status	Device	Status
Y0	OFF	Y1	OFF	Y2	ON	Y3	OFF
Y4	ON	Y5	ON	Y6	OFF	Y7	OFF
Y10	OFF	Y11	ON	Y12	OFF	Y13	OFF
Y14	ON	Y15	OFF	Y16	OFF	Y17	OFF



- ASCII mode (M1143 = OFF)

When X0 = ON, MODRW instruction executes the function specified by Function Code H0F.

ELC1 ⇨ ELC2, ELC sends: " 01 0F 0500 0010 02 3412 93 "

ELC2 ⇨ ELC1, ELC receives: " 01 0F 0500 0010 DB "

Data to be sent

Register	Data		Descriptions	
D1256 Low byte	'0'	30 H	ADR 1	Device address: ADR (1,0)
D1256 High byte	'1'	31 H	ADR 0	

D1257 Low byte	'0'	30 H	CMD 1	Control parameter: CMD (1,0)
D1257 High byte	'F'	46 H	CMD 0	
D1258 Low byte	'0'	30 H	Data Address	
D1258 High byte	'5'	35 H		
D1259 Low byte	'0'	30 H		
D1259 High byte	'0'	30 H		
D1260 Low byte	'0'	30 H	Number of Data (count by bit)	
D1260 High byte	'0'	30 H		
D1261 Low byte	'1'	31H		
D1261 High byte	'0'	30 H		
D1262 Low byte	'0'	30 H	Byte Count	
D1262 High byte	'2'	32 H		
D1263 Low byte	'3'	33 H	Data contents	1234H Content of register D0
D1263 High byte	'4'	46 H		
D1264 Low byte	'1'	33 H		
D1264 High byte	'2'	46 H		
D1265 Low byte	'9'	39 H	LRC CHK 1	Checksum: LRC CHK (0,1)
D1265 High byte	'3'	33 H	LRC CHK 0	

Received data

Register	Data		Descriptions
D1070 Low byte	'0'	30 H	ADR 1
D1070 High byte	'1'	31 H	ADR 0
D1071 Low byte	'0'	31 H	CMD 1
D1071 High byte	'F'	46 H	CMD 0
D1072 Low byte	'0'	30 H	Data Address
D1072 High byte	'5'	35 H	
D1073 Low byte	'0'	30 H	
D1073 High byte	'0'	30 H	
D1074 Low byte	'0'	30 H	Number of Data(count by bit)
D1074 High byte	'0'	30 H	
D1075 Low byte	'1'	31 H	
D1075 High byte	'0'	30 H	
D1076 Low byte	'D'	44 H	LRC CHK 1
D1076 High byte	'B'	42 H	LRC CHK 0

- RTU mode (M1143 = ON)

When X0 = ON, MODRW instruction executes the function specified by Function Code H0F

ELC1 ⇨ ELC2，ELC1 sends: “01 0F 0500 0010 02 34 12 21 ED”

◆ ELC2 ⇨ ELC1，ELC1 receives: “01 0F 0500 0010 54 CB”

Data to be sent

Register	Data	Descriptions	
D1256 Low byte	01 H	Address	
D1257 Low byte	0F H	Function	
D1258 Low byte	05 H	Data Address	
D1259 Low byte	00 H		
D1260 Low byte	00 H	Number of Data(count by bit)	
D1261 Low byte	10 H		
D1262 Low byte	02 H	Byte Count	
D1263 Low byte	34 H	Data content 1	Content of D0: H34
D1264 Low byte	12 H	Data content 2	Content of D1: H12
D1265 Low byte	21 H	CRC CHK Low	
D1266 Low byte	ED H	CRC CHK High	

Received data

Register	Data	Descriptions
D1070 Low byte	01 H	Address
D1071 Low byte	0F H	Function
D1072 Low byte	05 H	Data Address
D1073 Low byte	00 H	
D1074 Low byte	00 H	Number of Data(count by bit)
D1075 Low byte	10H	
D1076 Low byte	54H	CRC CHK Low
D1077 Low byte	CB H	CRC CHK High

Program example 10: COM1 (RS-232) / COM3 (RS-485), Function Code H0F

1020. Function code K15 (H0F): write multiple bit devices. Up to 64 bits can be written

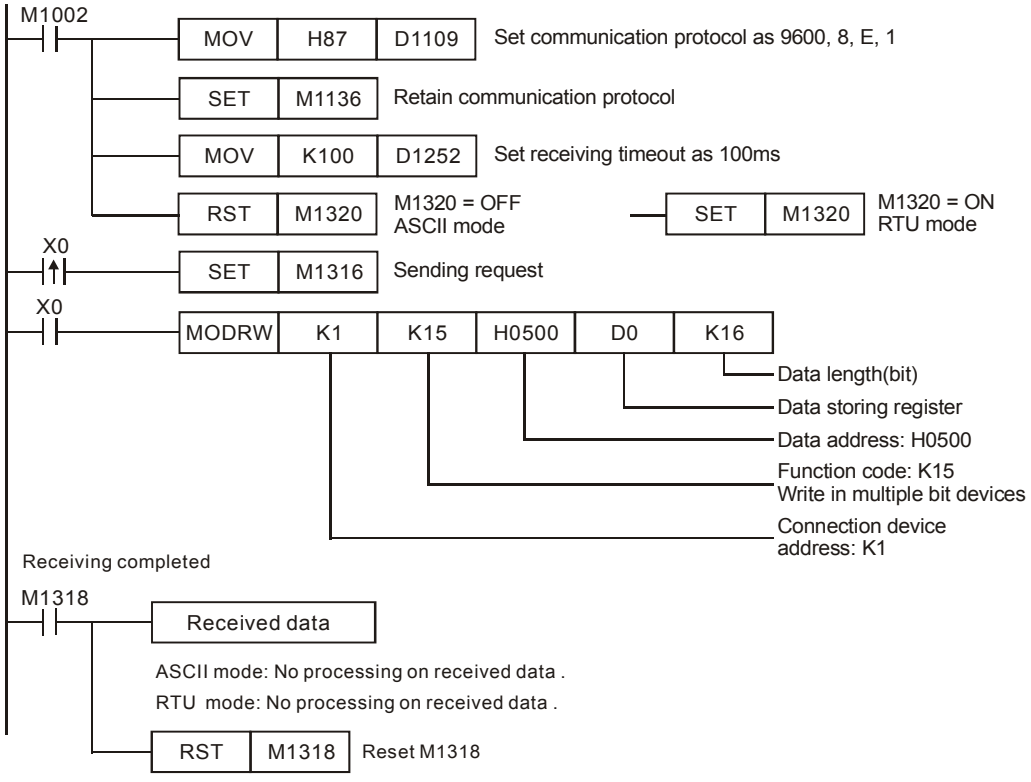
1021. M1143 = OFF, ASCII mode, M1143 = ON, RTU mode

1022. The messages sent between ELC1 (ELC COM3) and ELC2 (ELC COM1) is shown in the tables below.

Set value: K4Y0=1234H

Device	Status	Device	Status	Device	Status	Device	Status
Y0	OFF	Y1	OFF	Y2	ON	Y3	OFF
Y4	ON	Y5	ON	Y6	OFF	Y7	OFF
Y10	OFF	Y11	ON	Y12	OFF	Y13	OFF
Y14	ON	Y15	OFF	Y16	OFF	Y17	OFF

- If the ELC uses COM1 for communication, the program below can be usable by changing:
  a) D1109→D1036: communication protocol
  b) M1136→M1138: retain communication setting
  c) D1252→D1249: Set value for data receiving timeout
  d) M1320→M1139: ASCII/RTU mode selection
  e) M1316→M1312: sending request
  f) M1318→M1314: receiving completed flag



- ASCII mode (COM3: M1320 = OFF, COM1: M1139 = OFF):
  When X0 = ON, MODRW executes the function specified by Function Code H0F
  ELC1 ⇨ ELC2, ELC sends: " 01 0F 0500 0010 02 3412 93 "

ELC2 ⇨ ELC1, ELC receives: “ 01 0F 0500 0010 DB ”

(No data processing on received data)

- RTU mode (COM3: M1320 = ON, COM1: M1139 = ON):

  When X0 = ON, MODRW executes the function specified by Function Code H0F

  ELC1 ⇨ ELC2, ELC1 sends: “01 0F 0500 0010 02 34 12 21 ED”

  ELC2 ⇨ ELC1, ELC1 receives: “01 0F 0500 0010 54 CB” ,

  (No data processing on received data)

Program Example 11: COM2 (RS-485), Function Code H10

1023. Function code K16(H10): write multiple words of data. For ELC COM2 ASCII mode, only 8 words can be written.

ASCII Mode when M1143=OFF

RTU Mode when M1143=ON

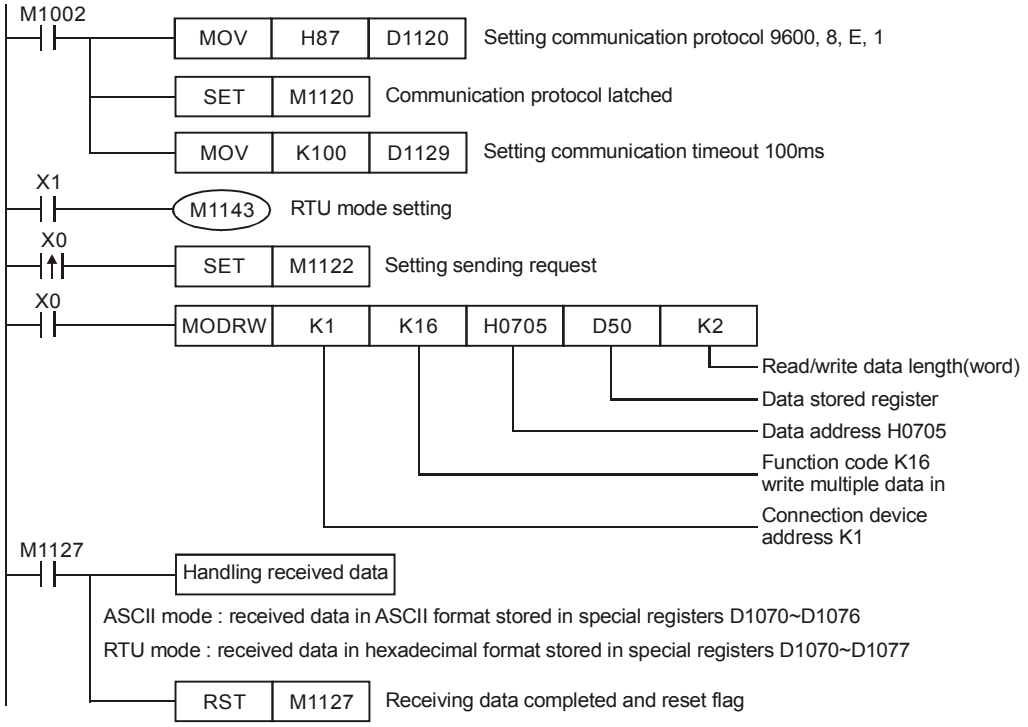
1024. In ASCII mode, store the data that will be written to the drive in ASCII format in 8 continuous registers started from D50. Received data from the drive will be stored in registers D1070~D1076.

1025. Received data from the drive will be stored in registers D1070~D1077.

1026. After receiving a response, the ELC automatically checks if the received data is correct. If there is any fault, M1140 = ON.

1027. If the register address of the drive is illegal, a fault code will be stored in D1130 and M1141 = ON.

1028. After M1140=ON or M1141=ON, the ELC will re-transmit the same data to the drive. If the received data is correct, M1140 and M1141 will be reset.



- ASCII Mode: ELC connects to MVX drive.

  ELC → the drive, ELC transmits: " 01 10 0705 0002 04 1770 0012 44 "

  The drive → ELC, ELC receives: " 01 10 0705 0002 E1 "

ELC transmit data

Register	Data		Descriptions	
D1256 Low byte	'0'	30 H	ADR 1 ADR 0	ADR (1,0) is MVX drive address
D1256 High byte	'1'	31 H		
D1257 Low byte	'1'	31 H	CMD 1 CMD 0	CMD (1,0) is instruction code
D1257 High byte	'0'	30 H		
D1258 Low byte	'0'	30 H	Data Address	
D1258 High byte	'7'	37 H		
D1259 Low byte	'0'	30 H		
D1259 High byte	'5'	35 H		
D1260 Low byte	'0'	30 H	Number of Register	
D1260 High byte	'0'	30 H		
D1261 Low byte	'0'	30 H		
D1261 High byte	'2'	32 H		
D1262 Low byte	'0'	30 H	Byte Count	
D1262 High byte	'4'	34 H		
D1263 Low byte	'1'	31 H	Data contents 1	The content of register D50 (H1770=K6000)
D1263 High byte	'7'	37 H		
D1264 Low byte	'7'	37 H		
D1264 High byte	'0'	30 H		
D1265 Low byte	'0'	30 H	Data contents 2	The content of register D51 (H12)
D1265 High byte	'0'	30 H		
D1266 Low byte	'1'	31 H		
D1266 High byte	'2'	32 H		
D1267 Low byte	'4'	34 H	LRC CHK 1 LRC CHK 0	LRC CHK (0,1) is error check
D1267 High byte	'4'	34 H		

ELC receive data

Register	Data		Descriptions
D1070 Low byte	'0'	30 H	ADR 1
D1070 High byte	'1'	31 H	ADR 0
D1071 Low byte	'1'	31 H	CMD 1
D1071 High byte	'0'	30 H	CMD 0
D1072 Low byte	'0'	30 H	Data Address
D1072 High byte	'7'	37 H	
D1073 Low byte	'0'	30 H	
D1073 High byte	'5'	35 H	
D1074 Low byte	'0'	30 H	Number of Register
D1074 High byte	'0'	30 H	
D1075 Low byte	'0'	30 H	
D1075 High byte	'2'	32 H	
D1076 Low byte	'E'	45 H	LRC CHK 1
D1076 High byte	'1'	31 H	LRC CHK 0

- RTU Mode: ELC connects to MVX drives

  ELC → the drive, ELC transmits: 01 10 0705 0002 04 1770 0012 91 C2

  The drive → ELC, ELC receives: 01 10 0705 0002 50 BD

ELC transmit data

Register	Data	Descriptions	
D1256 Low byte	01 H	Address	
D1257 Low byte	10 H	Function	
D1258 Low byte	07 H	Data Address	
D1259 Low byte	05 H		
D1260 Low byte	00 H	Number of Register	
D1261 Low byte	02 H		
D1262 Low byte	04 H	Byte Count	
D1263 Low byte	17 H	Data content 1	The content of register D50 (H1770=K6000)
D1264 Low byte	70 H		
D1265 Low byte	00 H	Data content 2	The content of register D51 (H12)
D1266 Low byte	12 H		
D1262 Low byte	91 H	CRC CHK Low	
D1263 Low byte	C2 H	CRC CHK High	

ELC receive data

Register	Data	Descriptions
D1070 Low byte	01 H	Address
D1071 Low byte	10 H	Function
D1072 Low byte	07 H	Data Address
D1073 Low byte	05 H	
D1074 Low byte	00 H	Number of Register
D1075 Low byte	02 H	
D1076 Low byte	50 H	CRC CHK Low
D1077 Low byte	BD H	CRC CHK High

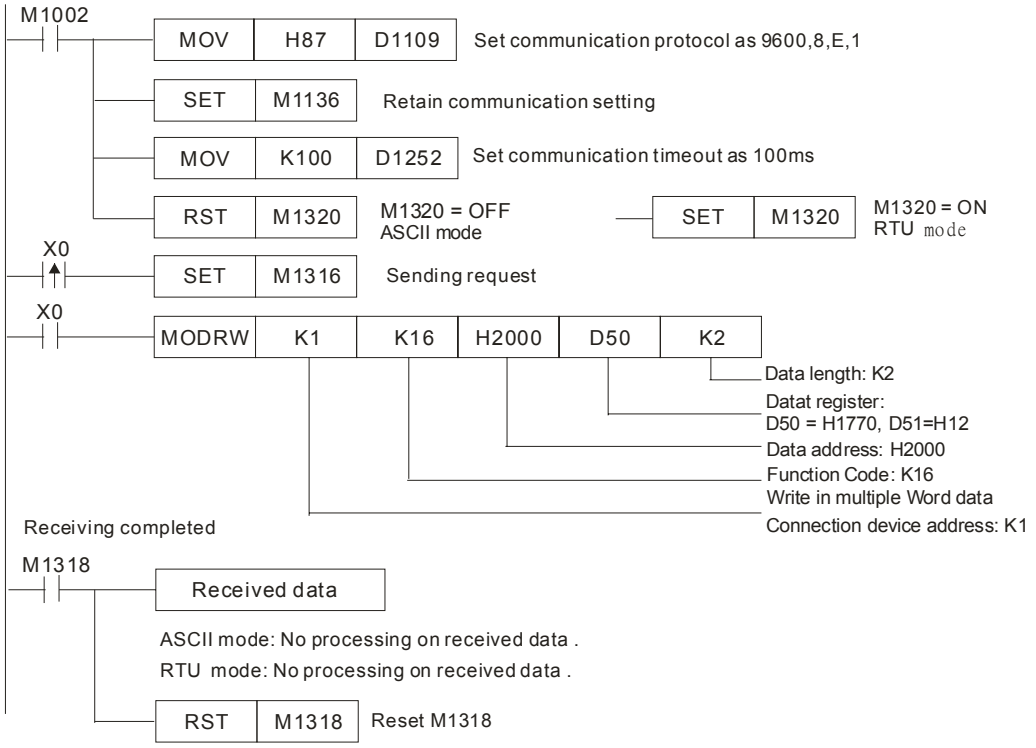
Program example 12: COM1 (RS-232) / COM3 (RS-485), Function Code H10

1029. Function code K16 (H10): Write multiple words. Up to 16 Words can be written.

1030. The message data between ELC COM3 and the drive is shown in the tables below.

1031. If the ELC uses COM1 for communications, the program below can be usable by changing:

1. D1109→D1036: communication protocol
2. M1136→M1138: retain communication setting
3. D1252→D1249: Set value for data receiving timeout
4. M1320→M1139: ASCII/RTU mode selection
5. M1316→M1312: sending request
6. M1318→M1314: receiving completed flag

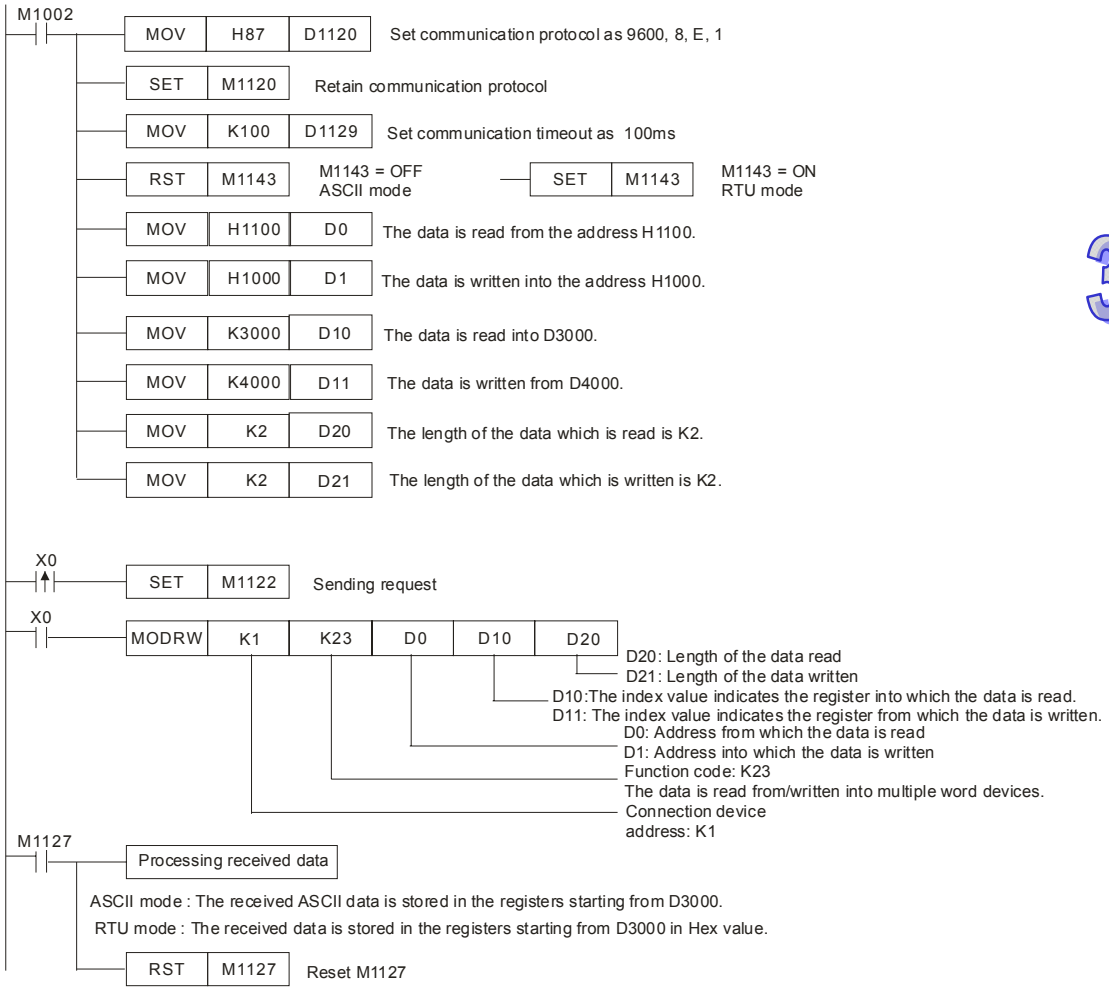


- ASCII mode (COM3: M1320 = OFF, COM1: M1139 = OFF):
  When X0 = ON, MODRW executes the function specified by Function Code H10
  ELC ⇨ the drive, ELC sends: “01 10 0705 0002 04 1770 0012 44”
  The drive⇨ELC, ELC receives: “01 10 0705 0002 E1”
  (No processing on received data)
- RTU Mode (COM3: M1320=On, COM1: M1139=On):
  When X0 = ON, MODRW executes the function specified by Function Code H10
  ELC ⇨ the drive,,ELC sends: “01 10 0705 0002 04 1770 0012 91 C2”
  The drive,⇨ELC, ELC receives :” 01 10 0705 0002 50 BD”
  (No processing on received data)

Program example 13: COM2 (RS-485)), Function Code H17

1. Function code K23 (H17): Data is read from multiple word devices and data is written into multiple word devices. Data can be read from 16 word devices at most, and data can be written into 16 word devices at most.
2. In the ASCII or RTU mode, the data received is stored in the registers starting from the register indicated by the index value in **S**.
3. The connection between ELC-A (PLC COM2) and ELC-B:

- Data is read from multiple word devices in ELC-B into ELC-A, and data is written into multiple word devices in ELC-B from ELC-A. (M1143=OFF, ASCII Mode) (M1143=ON, RTU Mode)



- ASCII Mode (M1143=OFF)

  When X0=ON, MODRW executes the function specified by the function ode H17.

  ELC-A ⇨ELC-B, ELC-A sends: “01 17 1100 0002 1000 0002 04 1770 0012 06”

  ELC-B⇨ELC-A, ELC-A receives: “01 17 04 0100 1766 66”

Registers in ELC-A for received data (responding messages)

Register	Data		Description
D3000 Low byte	'0'	30 H	ADR 1
D3000 High byte	'1'	31 H	ADR 0
D3001 Low byte	'1'	31 H	CMD 1
D3001 High byte	'7'	37 H	CMD 0
D3002 Low byte	'0'	30 H	Number of data (bytes)
D3002 High byte	'4'	34 H	
D3003 Low byte	'0'	30 H	Contents of the address 1100H
D3003 High byte	'1'	31 H	
D3004 Low byte	'0'	30 H	
D3004 High byte	'0'	30 H	
D3005 Low byte	'1'	31 H	Contents of the address 1101H
D3005 High byte	'7'	37 H	
D3006 Low byte	'6'	36H	
D3006 High byte	'6'	36H	
D3007 Low byte	'6'	36H	LRC CHK 1
D3007 High byte	'6'	36H	LRC CHK 0

- RTU Mode (M1143=ON)

When X0=ON, MODRW executes the function specified by the function ode H17.

ELC-A ⇨ELC-B,ELC-A sends: "01 17 1100 0002 1000 0002 04 1770 0012 A702"

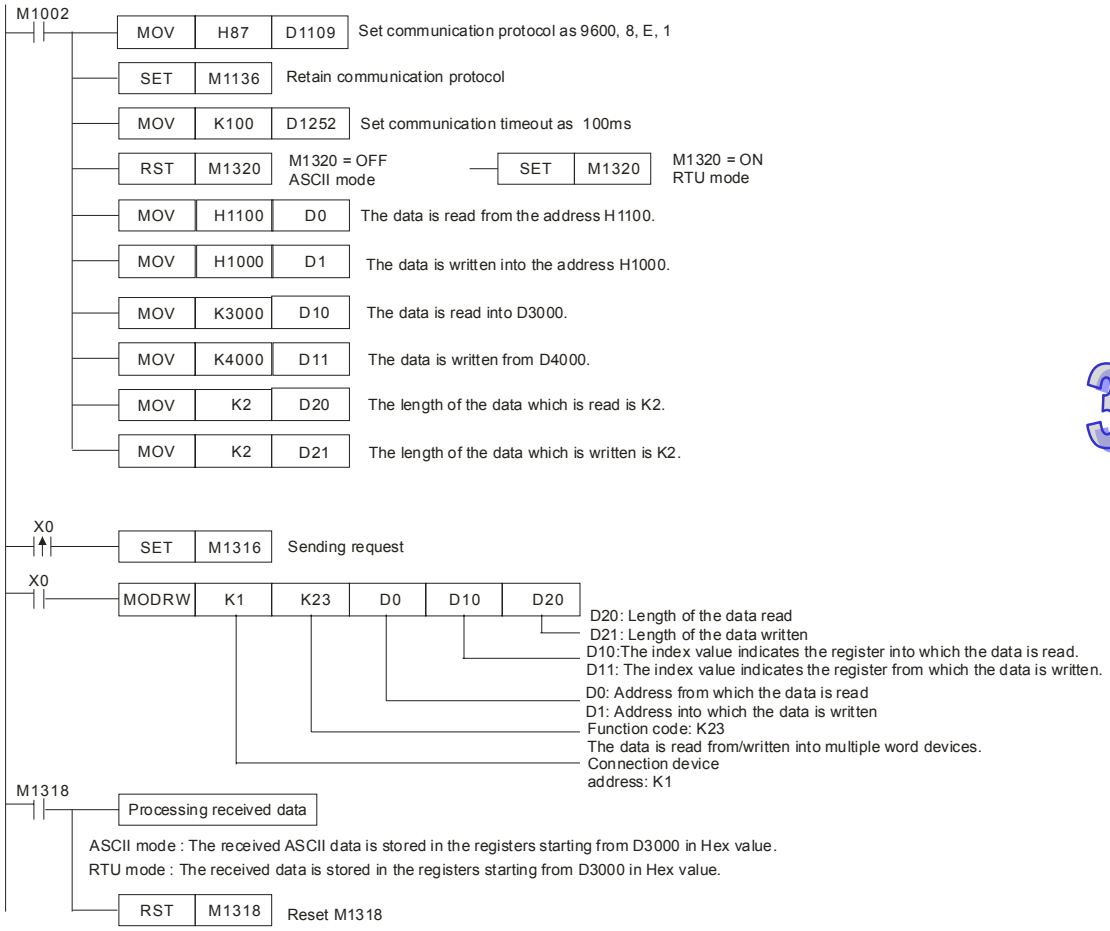
ELC-B⇨ELC-A, ELC-A receives: "01 17 04 0100 1766 7701"

Registers in ELC-A for received data (responding messages)

Register	Data	Description
D3000 Low byte	01 H	Address
D3001 Low byte	17 H	Function
D3002 Low byte	04 H	Number of data (bytes)
D3003 Low byte	01 H	Contents of the address 1100H
D3004 Low byte	00 H	
D3005 Low byte	17 H	Contents of the address 1101H
D3006 Low byte	66 H	
D3007 Low byte	77 H	CRC CHK Low
D3008 Low byte	01 H	CRC CHK High

Program example 14: COM1 (RS-232)/ COM3 (RS-485), Function Code H17

1. Function code K23 (H17): Data is read from multiple word devices and data is written into multiple word devices. Data can be read from 16 word devices at most, and data can be written into 16 word devices at most.
2. In the ASCII or RTU mode, the data received through COM1/COM3 on the PLC is stored in the registers starting from the register indicated by the index value in **S**+1. Users can use the instruction DTM to transform and move the data.
3. The connection between ELC-A (PLC COM3) and ELC-B:
   - Data is written into multiple word devices in ELC-B from ELC-A. (M1320=OFF, ASCII Mode) (M1320=ON, RTU Mode)
   - If COM1 on ELC-A is connected, the program can be modified as shown below.
     1. D1109→D1036: Communication protocol
     2. M1136→M1138: The communication setting is retained.
     3. D1252→D1249: Communication timeout
     4. M1320→M1139: Choice between the ASCII mode and the RTU mode
     5. M1316→M1312: The sending of the data though the communication instruction is requested.
     6. M1318→M1314: The receiving of the data through the communication instruction is complete.



- ASCII Mode (COM3: M1320=OFF; COM1: M1139=OFF):

  When X0=ON, MODRW executes the function specified by the function ode H17.

  ELC-A ⇨ELC-B, ELC-A sends: “01 17 1100 0002 1000 0002 04 1770 0012 06”

  ELC-B⇨ELC-A, ELC-A receives: “01 17 04 0100 1766 66”

  Registers in ELC-A for received data (responding messages)

Register	Data	Description
D3000	0100H	ELC-A converts ASCII codes in 1100H and stores the converted data automatically.
D3001	1766H	ELC-A converts ASCII codes in 1101H and stores the converted data automatically.

- RTU Mode (COM3: M1320=ON; COM1: M1139=ON):

  When X0=ON, MODRW executes the function specified by the function ode H17.

  ELC-A ⇨ELC-B,ELC-A sends: “01 17 2100 0002 2000 0002 04 1770 0012 A702”

ELC-B⇨ELC-A, ELC-A receives: “01 17 04 0100 1766 7701”

Registers in ELC-A for received data (responding messages)

Register	Data	Description
D3000	0100 H	ELC-A converts data in 1100H and stores the converted data automatically.
D3001	1766 H	ELC-A converts data in 1101H and stores the converted data automatically.

API	Mnemonic	Operands	Function
151	PWD	S, D	Detection of Input Pulse Width

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*															PWD: 5 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

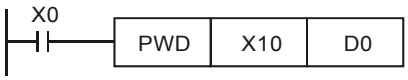
S: Source address **D**: Destination address for storing the result

Description:

1. Range of **S**: X10 ~ X17
2. Range of **D**: D0 ~ D999, occupying 2 consecutive devices. Can only be used once in the program.
3. PWD instruction is used for detecting the time span of input signals from X10 ~ X17; the valid frequency range is 1 ~1KHz. When M1169 = Off, the instruction will detect the time span from the rising edge to the falling edge of the input signals (units: 100us). When M1169 = On, the instruction will detect the time span from one rising edge of the input signal to the next rising edge of the input signal (units: 1us). It cannot use the same input as is used in the DCNT and ZRN instructions.
4. **D** occupies two consecutive words. The longest detectable time is 21,474.83647 seconds, about 357.9139 minutes or 5.9652 hours.
5. There is no limit on the number of times this instruction can be used in a program. However, only one instruction can be executed at a time.

◆ Program Example:

When X0 = On, record the time span of X10 = On and store it in D1 and D0.



API	Mnemonic	Operands	Function
152	**RTMU**	**S, D**	Start of the Measurement of Execution Time of I Interruption

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D					*	*							*			RTMU: 5 steps
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

D: The address offset for the execution time of an I interrupt (unit: 1us) **n**: time base. Parameter range: K10 ~ K500 (units: 1us)

Description:

1. Range of **D**: K0 ~ K9
2. Range of **n**: K10 ~ K500
3. The special D registers (D1156 ~ D1165) are used to measure the execution time of up to 10 interrupt subroutines. For example, when **D** = K5, the D register will be D1161.
4. When RTMU is executed, if the **D** and **n** entered by the user are legal, the timer will be enabled and the special D register designated by **D** is cleared to 0. When RTMD is executed, the instruction will measure the execution time of the interrupt and store the result in the special D register specified by **D.**
5. Refer to API 153 below for more information.

◆

API	Mnemonic	Operands	Function
153	RTMD	D	End of the Measurement of the Execution Time of I Interrupt

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D					*	*							*			RTMD: 3 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

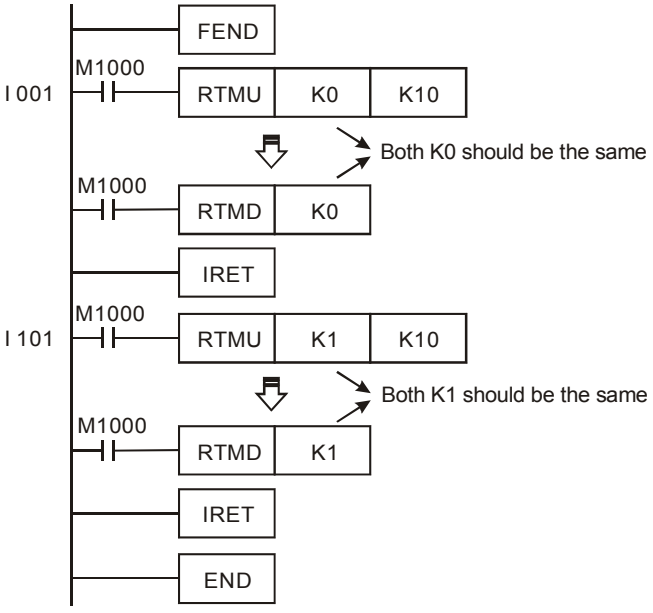
Operands:

- **D**: Address offset to store the measured time (unit: 1us).

Description:

1. Range of **D**: K0 ~ K9. The No. of **D** must be the same as that designated by **D** in API 152.
2. This instruction stops the RTMU instruction. It is necessary when using the RTMU instruction to measure the execution time of an I interrupt, to terminate the measurment with the RTMD instruction prior to the end of the I subroutine, per the example below.

- Program Example:
- When X0 goes from Off to On, the program will enter I001 interrupt subroutine. RTMU will activate an 8-bit timer (unit: 10us) and RTMD (when D = K0) will store the execution time in the special D registers D1156 ~ D1165, referenced by the offset value in **D** (K0 ~ K9).



Remarks:

1. It is recommended that this be used only for test purposes and that this logic be removed when testing is complete..
2. If RTMU is activated but RTMD is not activated before the end of the interrupt routine, the interrupt timer will not stop.

API	Mnemonic			Operands	Function
154	D	RAND	P	S_1, S_2, D	Random Number

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	RAND, RANDP: 7 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	DRAND, DRANDP: 13 steps
D								*	*	*	*	*	*	*	*	

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: lower limit for the random numbers S_2: upper limit for the random numbers **D**: Random number result

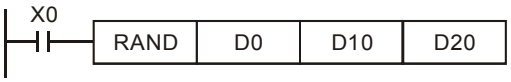
Description:

1032. The range of 16-bit operands S_1, S_2 is: K0 ≦ S_1 , S_2 ≦K32,767, the range of 32-bit operands S_1, S_2 is: K0 ≦ S_1 , S_2 ≦K2,147,483,647.

1033. If S_1 > S_2, the ELC will produce an operand error and M1067 and M1068=ON, and error code 0E1A(HEX) will be recorded in D1067.

Program Example:

When X0=ON, the random number produced between lower limit D0 and upper limit D10 will be saved in D20.



API	Mnemonic	Operands	Function
155	D ABSR	S, D_1, D_2	Absolute Position Read

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												DABSR: 13 steps
D_1		*	*	*												
D_2								*	*	*	*	*	*	*		

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Input signal from Servo (occupies 3 continuous addresses) **D_1**: Control signal for controlling the Servo (occupies 3 continuous addresses) **D_2**: Absolute position data (32-bit) read from Servo

Description:

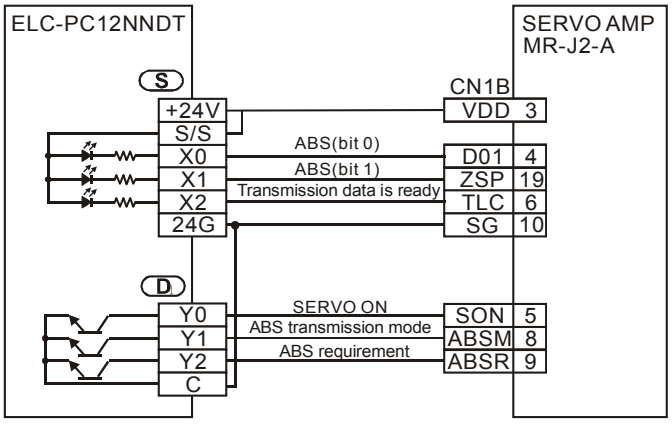
1034. This instruction provides continuous absolute position data read function of the Mitsubishi servo drive MR-J2 (with absolute position check function).

1035. There is no 16-bit instruction for API 155, it can only be used as a 32-bit instruction, DABSR is available and it can only be used once in the program.

1036. Flags: For the description of M1010, M1029, M1102, M1103, M1334, M1335, M1336, M1337, M1346, please refer to the Notes below.

1037. **S** is the input signal from the Servo and it will use 3 continuous addresses **S**, **S** +1, **S** +2. Device **S** and **S** +1 are connected to the ABS (bit0, bit1) of the Servo for data transmitting. Device **S** +2 is connected to Servo for transmitting the data ready flag.

1038. **D_1** is the control signal for controlling the Servo and it will use 3 continuous addresses **D_1**, **D_1**+1, **D_1**+2. Device **D_1** is connected to the Servo ON (SON), device **D_1**+1 is connected to ABS data transmitting mode and **D_1**+2 is connected to ABS data request signal.



1039. **D_2** is the absolute position data (32 bit) read from Servo and it will use 2 continuous addresses **D_2**, **D_2**+1. **D_2** is low word and **D_2**+1 is high word. The absolute position data should be stored in the current value registers (D1348, D1349) corresponding to CH0 pulse or the

current value registers (D1350, D1351) corresponding to CH1 pulse, so it is recommended to use these two registers. If other registers are used, the data will need to be transmitted into the current value registers (D1348, D1349) corresponding to CH0 pulse or the current value registers (D1350, D1351) corresponding to CH1 pulse.

1040. When the DABSR instruction is enabled and reading starts, the command execution completed flag M1029 will be ON. The flags must be reset by the user.

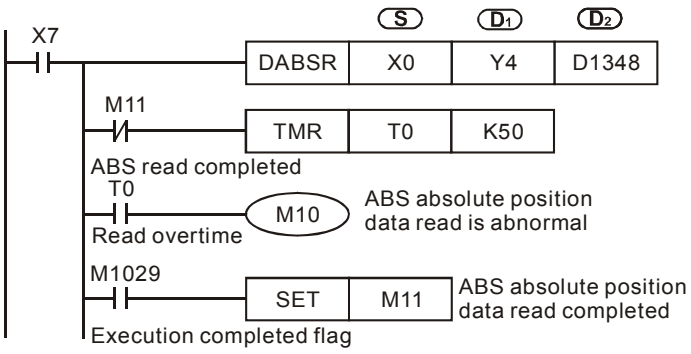
1041. When executing the DABSR instruction, use a normally open contact. If the contact of DABSR command turns OFF when DABSR command read starts, the execution of the absolute current value read will be interrupted and result will be incorrect.

1042. If the DABSR instruction is disabled after the read is complete, the Servo ON (SON) signal connected to **D_1** will also turn OFF and the operation will be disabled.

Program Example:

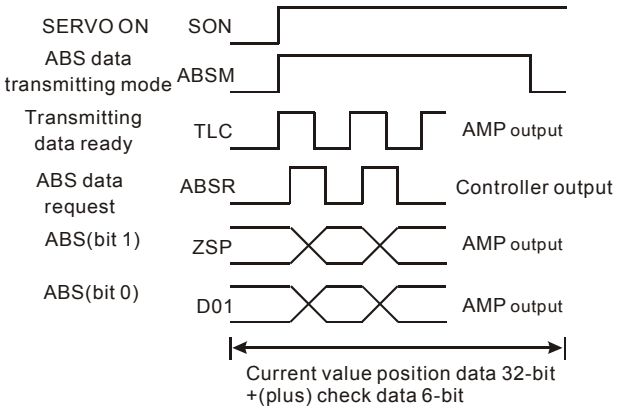
1043. When X7= ON, the absolute position data (32 bit) read from Servo should be stored in the current value register (D1348, D1349) corresponding to the CH0 pulse. At the same time, use a timer T0 to time for 5 seconds. If the time exceeds 5 seconds and the absolute position data (32 bit) read is not complete, M10=ON and this means the absolute position data (32 bit) read is abnormal.

1044. When connecting to the system, set the power of ELC-PV series and SERVO AMP to be ON at the same time or set the SERVO AMP to be ON before power is applied to ELC.



Points to notes:

1045. The Timing chart for the DABSR instruction absolute position data read:



1046. When the DABSR instruction starts to execute, it will turn the Servo ON (SON) and prompt it to send ABS data.

1047. With the transmit data ready flag and the ABS request signal, confirmation of the transmission and receipt from both sides is verified.

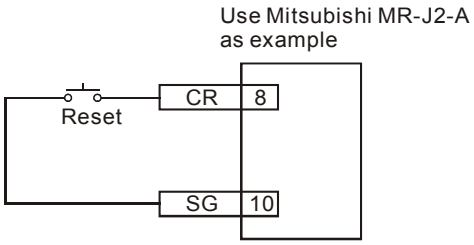
1048. Data is transmitted by ABS (bit0, bit1) two bits.

1049. This command is applicable to the Servo motor equipped with absolute position detect function, such as Mitsubishi MR-J2-A Servo drive.

1050. The Servo motor with absolute position detect function should be started at zero reference more than one degree revolution and given the reset signal before manufacturing equipments.

1051. Complete zero point return by using the reset signal function by executing API 156 ZRN command.

1052. After using JOG or manual operation to adjust the zero point position of the equipment, reset the SERVO AMP. For details on the wiring between ELC and Mitsubishi MR-J2-A, please refer to the API 159 DRVA.



Flags and Special Data register descriptions:

1053. M1029: (For ELC-PV series) M1029=ON after the command execution is complete.

Range of settings:

API	Mnemonic		Operands	Function
156	D	ZRN	S_1, S_2, S_3, D	Zero Return

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		DZRN: 17 steps
S_2					*	*	*	*	*	*	*	*	*	*		
S_3	*															
D		*														

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Zero return speed **S_2**: Creep speed **S_3**: Near point signal (DOG) **D**: Pulse output address

Description: (ELC-PV)

1054. **S_1** is specified as the zero point return speed. For the ELC-PV series, the range of the 16-bit value is from 10 ~ 32,767Hz and the range of the 32-bit value is from 10 ~ 200,000Hz. For the ELC-PH series, the range of the 32-bit value is from 100 to 100,000Hz. When the speed is greater (less) than the maximum (minimum) range, the maximum (minimum) range value will be used.

1055. **S_2** is specified as the creep speed (start and end frequency), the lower speed after the near point signal (DOG) turns ON. For the ELC-PV series, the range is from 10 ~ 32,767Hz.

1056. **S_3** is specified as the near point signal (DOG) input. In the ELC-PV series, if devices other than the input device (X10 ~ X17), e.g. X, Y, M, S are used, they will be affected by the scan time, resulting in dispersion of the zero point. In addition, please note that the same input points X10 ~ X17 used for this instruction should not be used by the DCNT and PWD instructions.

1057. The ELC-PV series has four groups of A/B phase pulse outputs, CH0 (Y0, Y1), CH1 (Y2, Y3), CH2 (Y4, Y5) and CH3 (Y6, Y7).

1058. When executing the API 158 DRVI and API 159 DRVA instructions, the ELC stores the current pulse count in D-registers so that the machine position is always known. For the ELC-PV series, Y0: D1337, D1336; Y2: D1339, D1338, Y4: D1376, D1375; Y6: D1378, D1377. This allows the user to keep track of the position of the machine at any time. However, the data may be lost when power to the ELC is turned off. Therefore, the machine should execute a zero point return at power up.

1059. Flag: For the description of M1010, M1029, M1102, M1103, M1334, M1335, M1336, M1337, M1346, refer to the PLSY (API 57) instruction.

1060. Zero return output addresses in the two ELC models

◆ Model	◆ ELC-PV

• Zero return output	• Y0, Y2, Y4, Y6

The Timing Chart of the reset signal

1. Timing chart of the reset signal output for ELC-PV series.

   a) When the reset signal flag M1346 = On, after zero return is completed, the ELC can send the reset signal to the servo drive for approximately 20ms. After 20ms, the reset signal will return to Off again.

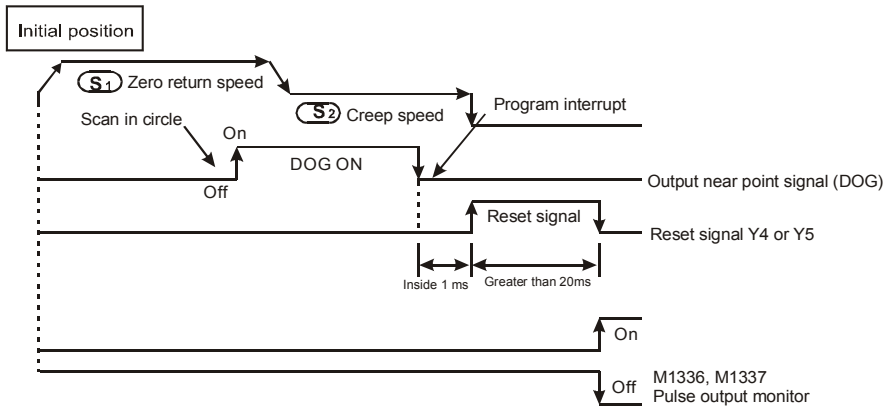
   b) Output addresses for reset signals of ELC-PV series:

CH0 (Y0, Y1) reset output device (Y10)

CH1 (Y2, Y3) reset output device (Y11)

CH2 (Y4, Y5) reset output device (Y12)

CH3 (Y6, Y7) reset output device (Y13)



Description of the zero point return operation:

1061. When the ZRN instruction is executed, accelerate to Zero point return speed **S_1** and start to move; Y10 and Y11 will use S2 as the initial frequency. For the ELC-PV series, the accel/decel time is set by Y10 (D1343) and Y11 (D1353).

1062. When the Near point signal (DOG) goes from OFF to ON, it will decelerate to the Creep speed **S_2** according to the accel/decel time setting (D1343 and D1353).

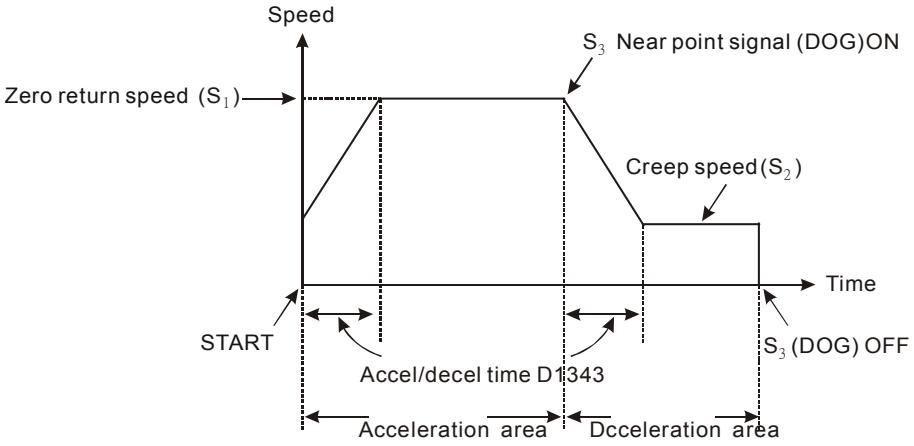
1063. When the Near point signal (DOG) goes from ON to OFF and the pulse output stops, 0 will be written to Y10, the current pulse value (D1337, D1336), Y11 current pulse value (D1339, D1338), Y12 current pulse value (D1375,D1376), and Y13 current pulses value (D1377, D1378) in ELC-PV series. 0 will also be written to Y10 current pulse value (D1348,D1349) or Y11 current pulse value (D1350,D1351) in ELC-PH series.

1064. When the Near point signal (DOG) goes from On to Off and the reset signal flag M1346=On, Y10 (CH0), Y11 (CH1) , Y12 (CH2) and Y13 (CH3) in the ELC-PV series will send a reset signal.

1065. For the ELC-PV series, when the pulse output is completed and M1029, M1030, M1036 and M1037 are enabled, flag M1336 will be reset for the CH0 pulses, M1337 for CH1, M1522

for CH2 and M1523 for CH3. For the ELC-PH series, when the pulse output is completed, M1102 and M1103 will be enabled.

1066. The instruction can not search the position of the Near point signal (DOG) and the operation of the ZRN instruction can only be processed in one direction. For the ZRN instruction, the pulse value register (D1337,D1336) for Y10 or the pulse value register (D1339,D1338) for Y11 will decrease in the ELC-PV series; the pulse value register (D1348, D1349) for Y10 or the pulse value register (D1350, D1351) for Y11 will also decrease in ELC-PH series.



1067. This instruction is applicable to the Servo motor equipped with the absolute positioning function. It can record the current position even when the power is OFF. Because the current position of the servo drive can be read by API 155 the DABSR instruction in the ELC-PV series , the ZRN instruction should only be executed for one time.

1068. When instruction is in deceleration and the output frequency reaches creep speed, the output pulses will stop when the near point signal transitions from ON to OFF.

1069. During instruction execution, parameters cannot be modified until instruction execution is completed.

1070. When the instruction is OFF, all outputs will stop.

Description: (ELCM-PH/PA, ELC2-PA/PB/PH/PE)

1071. **S_1** (zero return speed): max. 100kHz. **S_2** (JOG speed for DOG) must be less than **S_1**. The JOG speed for the DOG also refers to the start frequency.

1072. **S_3** and **D** operands must be used as an input/output set according to the table below, i.e. when **S_3** is specified as X4, **D** must be specified as Y0; also when **S_3** is specified as X6, **D** must be specified as Y2.

1073. M1307 enables (ON) / disables (OFF) the left limit switch of CH0 (Y0, Y1) and CH1 (Y2, Y3). M1307 must be set up before the instruction executes. M1305 and M1306 can reverse the pulse output direction on Y1 and Y3 and must be set up before instruction executes.

The left limit switch for CH0 (Y0, Y1) is X5; the left limit switch for CH1 (Y2, Y3) is X7.

◆ Channel ◆ Input		
◆ DOG point		
◆ Left limit switch (		

◆ Channel ◆ Input		
M1307 = ON)		
◆ Reverse pulse		

◆ Channel ◆ Input		
output direction		

1074. When **D** is specified as Y0, its direction signal output is Y1; when **D** is specified as Y2, its direction signal output is Y3.

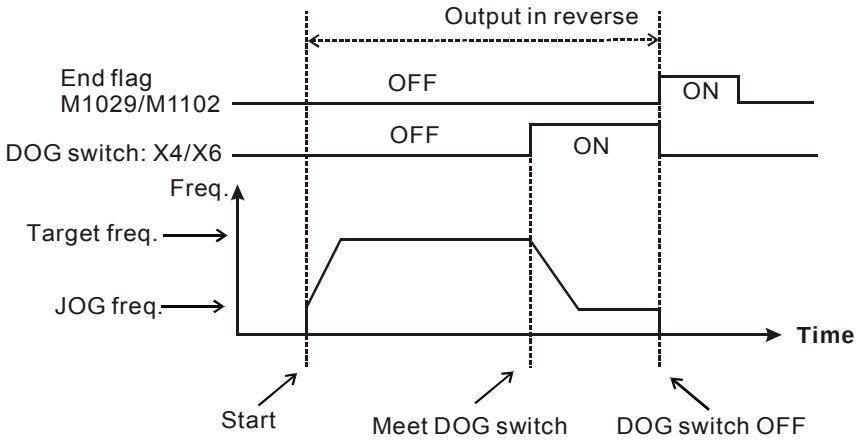
1075. When the instruction executes, the pulse output starts in the reverse direction. The ELC then controls the direction based on the limit switch and DOG switch. The current position of the Y0 output: (D1030,D1031); current position of the Y1 output (D1032, D1033)

1076. When the pulse output reaches the zero point, the pulse output execution complete flag M1029 (CH0), M1102 (CH1) is ON and the register indicating current position is reset to 0.

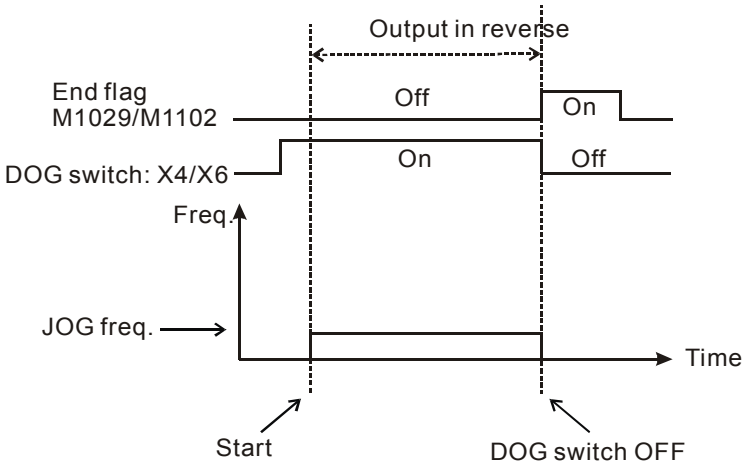
1077. When DZRN instruction executes, external interrupt I40x (Y0) or I60x (Y2) in program will be disabled until DZRN instruction is complete. Also, if the left limit switch (X5 / X7) is enabled during instruction execution, external interrupts will be disabled as well.

1078. Timing Diagram:

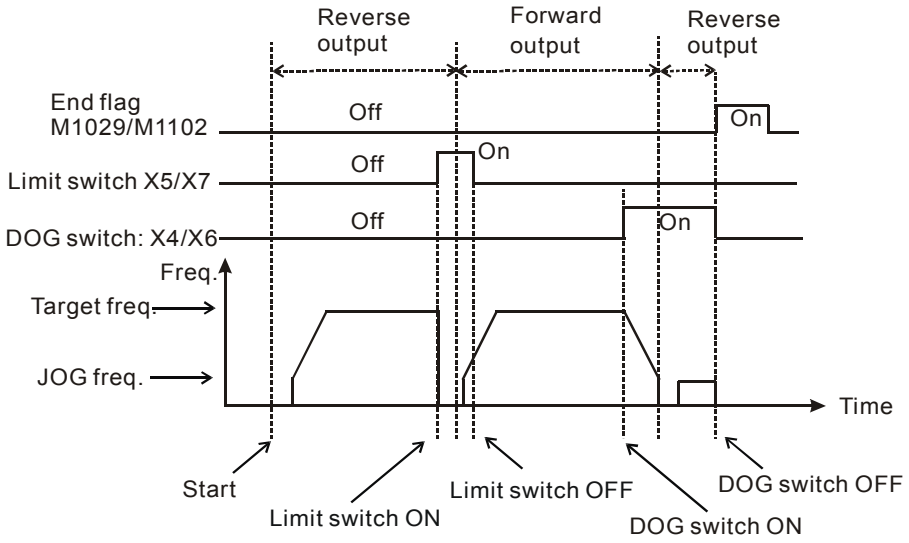
Mode1: Current position at right side of DOG switch, pulse output in reverse, limit switch disabled.



Mode 2: DOG switch is ON, pulse output in reverse, limit switch disabled.

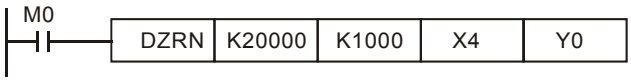


Mode 3: Current position at left side of zero point, pulse output in reverse, limit switch enabled.



Program Example:

When M0 = ON, Y0 pulse output executes zero return with a frequency of 20kHz. When it reaches the DOG switch, X4 = ON, the frequency changes to the JOG frequency of 1kHz. Y0 will then stop when X4 = OFF.



Description: (ELC2-PV)

1. When S1 and S2 are used in device F, only 16-bit instruction is applicable.
2. Flag: see remarks API 158 DDRVI for more details.
3. S1 is the starting speed of zero return operation. The 16-bit instruction can designate the range of the speed, which is 10 ~ 32,767Hz and the range designated by the 32-bit instruction

is 10 ~ 200,000Hz. If the designated speed is slower than 10Hz, the zero return will operate at 10Hz and when the designated speed is faster than 200kHz, the zero return will operate at 200kHz.

4. S2 is the designated low speed after the near point signal (DOG) is On. ELC2-PV can designate the range of S2, which is 10 ~ 32,767Hz.

5. S3 is the designated near point signal (DOG) input (input from A contact). In ELC2-PV series MPU, if devices other than the external output device (X10 ~ X17), e.g. X, Y, M, S are designated, they will be affected by the scan period, resulting in dispersion of the zero point. In addition, please note that the MPU cannot designate the same input points X10 ~ X17 as those designated by DCNT and PWD instructions.

6. Zero return output device in different models:

Model	ELC2-PV
Zero return output	Y0, Y2, Y4, Y6

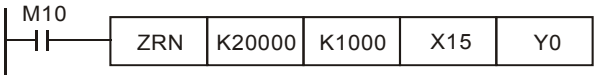
7. The instruction DZRN can be used to detect the limit switch, nake the pulsed output stop at the positive position, search for the Z phase, and output a certain number of displacement. Therefore, when the instrcution is writtem, the input number of the DOG point should be consisten with the description in the table below.

Output point number (**D**)		Y0	Y2	Y4	Y6
Corresponding output point number		Y1	Y3	Y5	Y7
DOG point number (S_3)		X2	X6	X12	X16
Disabling the left limit		M1570=On	M1571=On	M1572=On	M1573=On
Left limit input point		X3	X7	X13	X17
Stopping at the right side of DOG		M1574=On	M1575=On	M1576=On	M1577=On
Searching for the Z phase (M1578=Off)	Z phase number	X1	X5	X11	X15
	The number of times the Z phase is searched for is stored in D1312.	Positive value: Searching for the Z phase in the positive direction Negative value: Searching for the Z phase in the negative direction			
Number of displacement (M1578=On)	The number of displacement is stored in D1312.	Positive value: The pulse output is in the positive direction Negative value: The pulse output is in the negative direction			
Clearing the output (M1346=On)		Y10	Y11	Y12	Y13

8. When executing API 158 DRVI (releative positioning) or API 159 DRVA (absolute positioning), PLC will automatically store the increasing or decreasing forward/reverse pulses in the present value registers. For ELC2-PV series MPU, Y0: D1337, D1336; Y2: D1339, D1338, Y4: D1376, D1375; Y6: D1378, D1377.

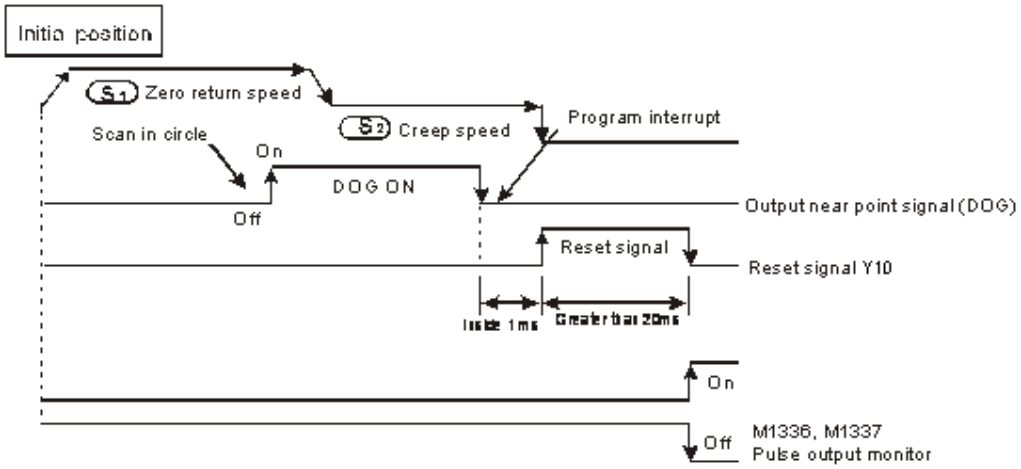
Program Example:

When M10= On, Y0 output pulses start to operate zero return at the frequency of 20kHz. When the zero return meets DOG X15 = On, Y0 output pulses will start to operate by creep speed 1kHz until X15 is Off.



Remarks:

1. Timing chart of the reset signal output.
   a) When the reset signal flag M1346 = On, after zero return is completed, the PLC can send the reset signal to the servo drive and the signal will last for approximately 20ms. After 20ms, the reset signal will return to Off again.
   b) Output devices for reset signals:
   CH0 (Y0, Y1) reset output device (Y10)
   CH1 (Y2, Y3) reset output device (Y11)
   CH2 (Y4, Y5) reset output device (Y12)
   CH3 (Y6, Y7) reset output device (Y13)



Note: The designated devices, X, Y, M, and S, other than the external input devices X10 ~ X17 will be affected by the scan period, 2 times of the scan period at worst.

API	Mnemonic	Operands	Function
157	D PLSV	**S, D_1, D_2**	Adjustable Speed Pulse Output

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	PLSV: 7 steps DPLSV: 13 steps
D_1		*														
D_2		*	*	*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Pulse output frequency **D_1**: Pulse output address (must use transistor outputs)

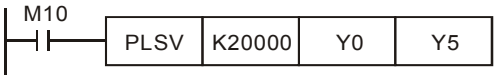
D_2: Output address for the rotation direction signal

Description: (ELC-PV/ELC2-PV)

1. Flag: see remarks of API 158 DDRVI for more details.
2. **S** is the designated pulse output frequency. The 16-bit instruction ranges are 0 ~ +32,767Hz, 0 ~ -32,768Hz. The ranges for the 32-bit instruction are 0 ~ +200,000Hz and 0 ~ -200,000Hz. "+/-" signs indicate forward/backward directions. During the pulse output, the frequency can be changed, but not the direction.
3. **D_1** is the pulse output address. ELC-PV, ELC2-PV can use Y0, Y2, Y4 and Y6.
4. The operation of **D_2** corresponds to the "+" or "-" of **S**. When **S** is "+", **D_2** will be On; when **S** is "-", **D_2** will be Off.
5. The PLSV instruction does not have settings for acceleration and deceleration. Use API 67 RAMP for the acceleration and deceleration of the pulse output frequency.
6. When the absolute value of the input frequency during the execution of DPLSV is greater than 200KHz, the output will operate at 200KHz.
7. D1222, D1223, D1383 and D1384 contain the Time difference between the direction signal and the pulse output points for CH0, CH1, CH2 and CH3.
8. M1305, M1306, M1532 and M1533 are the flags for the direction signals for CH0, CH1, CH2 and CH3. When S is "+", the output will operate in the forward direction and the flag will go Off. When S is "-", the output will operate in the backward direction and the flag will go On.

◆ Program Example:

When M10 = ON, Y0 will send pulses at 20KHz. Y5 = ON indicates forward direction.



Description: (ELCM-PH/PA, ELC2-PB/PH/PA/PE)

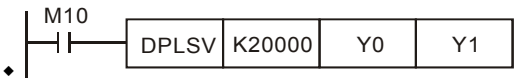
1. **S** is the designated pulse output frequency. Available range: -100,000Hz ~ +100,000 Hz. "+/-"

signs indicate forward/reverse output direction. The frequency can be changed during pulse output execution. However, if the specified output direction is diferent from the current output direction, the instruction will stop for 1 scan cycle then restart with the changed frequency.

2. **D_1** is the pulse output address. It can use CH0(Y0) and CH1(Y2).
3. **D_2** is the direction signal output device. It can use CH0(Y1) and CH1(Y3).
4. The operation of **D_2** corresponds to the "+" or "-" of **S**. When **S** is "+", **D_2** will be OFF; when **S** is "-", **D_2** will be ON.
5. M1305 and M1306 can change the output direction of CH0/CH1 set in **D_2**. When **S** is "-", **D_2** will be ON, however, if M1305/M1306 is set ON before instruction executes, **D_2** will be OFF during execution of instruction.
6. PLSV instruction does not support the settings for ramp up or ramp down. If ramp up/down is needed, use API 67 the RAMP instruction.
7. If the drive contact turns off during the pulse output process, the pulse output will stop immediately.
8. **D_2** doesn't support the index registers E, F modification.

◆ Program Example:

When M10 = ON, Y0 will send pulses at 20kHz. Y1 = OFF indicates forward direction.



API	Mnemonic		Operands	Function
158	D	DRVI	S_1, S_2, D_1, D_2	Relative Position Control

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	DDRVI: 17 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D_1		*														
D_2		*	*	*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

3

Operands:

S_1: Numbers of pulses (Target device) S_2: Pulse output frequency D_1: Pulse output address

D_2: Rotation direction signal

Description: (ELC-PV, ELC2-PV)

1079. S_1 is specified as the numbers of pulses (relative position). The 16-bit instruction range is -32,768 ~ +32,767, and the range for the 32-bit instruction is -2,147,483,648 ~ +2,147,483,647. The positive (+) and negative (-) symbol indicates the forward and reverse direction.

1080. S_2 is specified as the pulse output frequency. The 16-bit instruction range is 10 ~ 32,767Hz, and the range of the 32-bit instruction is 10 ~ 200,000Hz.

1081. Has four groups of A/B phase pulse outputs, CH0 (Y0, Y1), CH1 (Y2, Y3), CH2 (Y4, Y5) and CH3 (Y6, Y7).

1082. D_1 is specified as the pulse output address.

◆ Model	◆ ELC-PV, ELC2-PV
◆ Pulse output end	◆ Y0, Y2, Y4, Y6

1083. D_2 is specified as rotation direction signal and it operates based on the polarity of S_1. When S_1 is negative (-), **D2** is OFF. When S_1 is positive (+), **D2** is ON and D_2 will not be OFF immediately following the completion of the pulse output. D_2 will be turned OFF when contact switch is OFF.

1084. S_1 is

- The 32-bit data stored in the present value registers (D1337, D1336) of CH0 (Y0, Y1).
- The 32-bit data stored in the present value registers (D1339, D1338) of CH1 (Y2, Y3).
- The 32-bit data stored in the present value registers (D1376, D1375) of CH2 (Y4, Y5).

- The 32-bit data stored in the present value registers (D1378, D1377) of CH3 (Y5, Y6).
- When in the reverse direction, the contents in the present value register will decrease.

1085. The contents of each operand cannot be changed while the DRVI instruction is executed. The contents will be changed with the next execution of the instruction.

1086. If the conditions preceding the DRVI instruction turn OFF when the DRVI command is executing, the machine will decelerate to a stop and the execution completed flag M1102 and M1103 will turn ON.

1087. When the conditions preceding the DRVI instruction are Off, even if the indication flag M1336 generated by CH0 pulses, M1337 generated by CH1 pulses, M1522 generated by CH2 pulses and M1523 generated by CH3 pulses are “On”, the DRVI instruction will not execute again.

1088. When the absolute value of the input frequency of the DDRVI instruction is greater than 200KHz, the output will be operated at 200KHz. When the absolute value of the input frequency is less than 10Hz, the output will be operated at 10Hz.

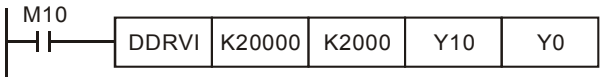
1089. D1343 (D1353) is the accel/decel time setting. The acceleration and deceleration time shall not be shorter than 10ms. The output will be operatedp for 10ms if the time is shorter than 10ms or longer than 10,000ms.

1090. Flags: For the description of M1010, M1029, M1102, M1103, M1334, M1335, M1336, M1337, M1346, refer to the PLSY (API 57) instruction.

1091. M1305 (M1306) is the direction signal for CH0 (CH1). When $\mathbf{S_1}$ is a positive number, the output will be operated in the forward direction and M1305 (M1306) will be Off. When $\mathbf{S_1}$ is a negative number, the output will be operated in the reverse direction and M1305 (M1306) will be On.

Program Example:

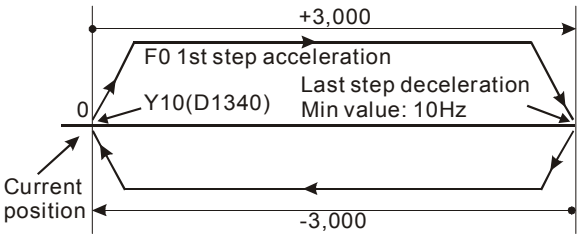
When M10= ON, 20000 pulses at a frequency of 2KHz will be sent out Y10 (relative position). Y0= ON represents the positive direction.



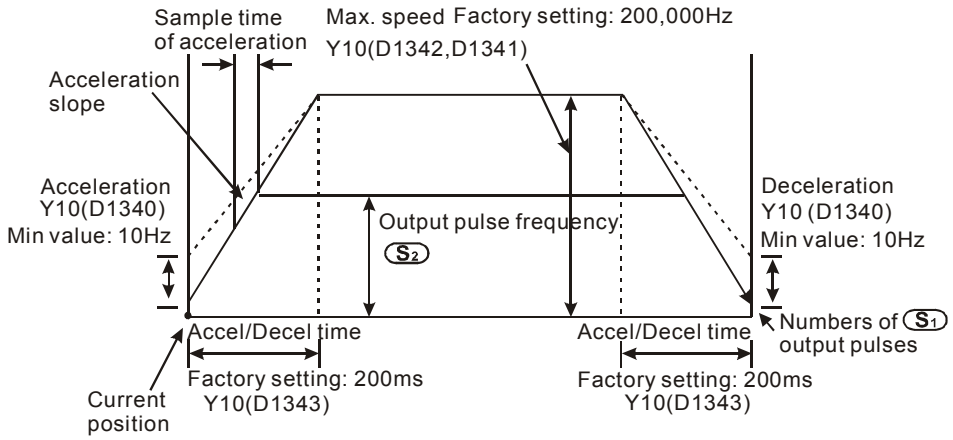
Points to note:

1092. Relative position control: Using a positive or a negative value to specify travel distance from the current position is also a way of doing relative position control.

ELC-PV, ELC2-PV series



1093. The setting of relative position and acceleration/deceleration:



a) This instruction can be used many times in the program.

b) After Y10 is activated by the DDRVI instruction, output Y10 will be disabled as will Y11.

c) Once the instruction is active, parameters cannot be modified until the instruction is de-activated.

d) When the instruction is OFF but the output is not complete, if M1334=ON, Y10 will stop immediately. If M1334=OFF, Y10 will deceleration to the end frequency based on the deceleration time and then stop.

e) When the instruction is OFF but the output is not complete, if M1335=ON, Y10 will stop immediately. If M1335=OFF, Y10 will deceleration to the end frequency based on the deceleration time and then stop.

1094. Flags description:

M1010: When M1010 = On, CH0, CH1, CH2 and CH3 will send pulses when the END instruction is being executed. M1010 will be Off automatically when the output starts.

M1029: M1029 = On after CH0 pulse output is complete.

M1030: M1030 = On after CH1 pulse output is complete.

M1036: M1036 = On after CH2 pulse output is complete.

M1037: M1037 = On after CH3 pulse output is complete.

M1305: Direction signal of CH0.

M1306: Direction signal of CH1.

M1334: CH0 pulse output stops.

M1335: CH1 pulse output stops.

M1336: CH0 sending pulses" indication.

M1337: CH1 sending pulses" indication.

M1520: CH2 pulse output stops.

M1521: CH3 pulse output stops.

M1522: CH2 sending pulses" indication.

M1523: CH3 sending pulses" indication.

M1534: Deceleration time of CH0 (used with D1348).

M1535: Deceleration time of CH1 (used with D1349).

M1536: Deceleration time of CH2 (used with D1350).

M1537: Deceleration time of CH3 (used with D1351).

M1532: Direction signal of CH2.

M1533: Direction signal of CH3.

1095. Special registers:

D1220: Phase setting of CH0 (Y0, Y1): D1220 determines the phase by the last two bits; other bits are invalid.

1. K0: Y0 output

1096. K1: Y0, Y AB-phase output; A ahead of B.

1097. K2: Y0, Y1 AB-phase output; B ahead of A.

1098. K3: Y1 output

D1221: Phase setting of CH1 (Y2, Y3): D1221 determines the phase by the last two bits; other bits are invalid.

1. K0: Y2 output
2. K1: Y2, Y3 AB-phase output; A ahead of B.
3. K2: Y2, Y3 AB-phase output; B ahead of A.

1099. K3: Y3 output

D1222: The time difference between the direction signal and pulse output sent by CH0.

D1223: The time difference between the direction signal and pulse output sent by CH1.

D1229: Phase setting of CH2 (Y4, Y5): D1229 determines the phase by the last two bits; other bits are invalid.

1. K0: Y4 output

1100. K1: Y4, Y5 AB-phase output; A ahead of B.

1101. K2: Y4, Y5 AB-phase output; B ahead of A.

1102. K3: Y5 output

D1230: Phase setting of CH3 (Y6, Y7): D1230 determines the phase by the last two bits; other bits are invalid.

1. K0: Y6 output

1103. K1: Y6, Y7 AB-phase output; A ahead of B.

1104. K2: Y6, Y7 AB-phase output; B ahead of A.

1105. K3: Y7 output

D1336: Low word of the current number of output pulses from CH0.

D1337: High word of the current number of output pulses from CH0.

D1338: Low word of the current number of output pulses from CH1.

D1339: High word of the current number of output pulses from CH1.

D1340: The first start frequency and the last end frequency of CH0.

D1343: Acceleration/deceleration time for CH0 pulse output.

D1352: The first start frequency and the last end frequency of CH1.

D1353: Acceleration/deceleration time for CH1 pulse output.

D1375: Low word of the current number of output pulses from CH2.

D1376: High word of the current number of output pulses from CH2.

D1377: Low word of the current number of output pulses from CH3.

D1378: High word of the current number of output pulses from CH3.

D1379: The first start frequency and the last end frequency of CH2.

D1380: The first start frequency and the last end frequency of CH3.

D1348: Deceleration time for CH0 pulse output when M1534 = On.

D1349: Deceleration time for CH1 pulse output when M1535 = On.

D1350: Deceleration time for CH2 pulse output when M1536 = On.

D1351: Deceleration time for CH3 pulse output when M1537 = On.

D1381: Acceleration/deceleration time for CH2 pulse output.

D1382: Acceleration/deceleration time for CH3 pulse output.

D1383: The time difference between the direction signal and pulse output sent by CH2.

D1384: The time difference between the direction signal and pulse output sent by CH3.

Description: (ELCM-PH/PA, ELC2-PB/PH/PA/PE)

1106. **S_1** is the number of pulses (relative position). Available range: -2,147,483,648 ~ +2,147,483,647. "+/-" signs indicate forward and reverse direction.

1107. **S_2** is the pulse output frequency. Available range: 6 ~ 100,000Hz.

1108. **D_1** is the pulse output address. It can use CH0 (Y0) or CH1 (Y2).

1109. **D_2** is the direction signal for the output. It can use CH0 (Y1) or CH1 (Y3).

1110. The operation of D_2 corresponds to the "+" or "-"of **S**. When **S** is "+", D_2 will be OFF; when **S** is "-", D_2 will be ON. D_2 will not be OFF immediately after pulse output complete and will be OFF when the drive contact is OFF.

1111. The value in S_1 is the relative position of

- The current position (32-bit data) of CH0 (Y0, Y1) which is stored in D1031(high), D1030 (low)
- The current position (32-bit data) of CH1 (Y2, Y3) which is stored in D1337(high), D1336 (low).

◆ In reverse direction the pulse output values are in (D1031, D1330) and (D1336, D1337) decreases.

1112. D1343 (D1353) is the ramp up/down time setting of CH0 (CH1). Available range: 20 ~ 32,767ms. Default: 100ms. The ELC will take the upper/lower bound value as the set value when the specified value exceeds the available range.

1113. D1340 (D1352) is the start/end frequency setting of CH0 (CH1). Available range: 6 to 100,000Hz. The ELC will take the upper/lower bound value as the set value when specified value exceeds the available range.

1114. M1305 and M1306 can change the output direction of CH0/CH1 set in D_2. When **S** is "-", D_2 will be ON, however, if M1305/M1306 is set ON before the instruction executes, D_2 will be OFF during execution of the instruction..

1115. Ramp-down time of CH0 and CH1 can be modified by using (M1534, D1348) and (M1535, D1349). When M1534 / M1535 = ON, CH0 / CH1 ramp-down time is specified by D1348 / D1349.

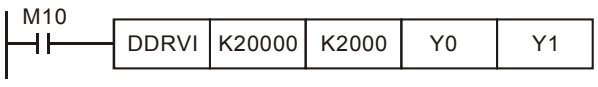
1116. If M1078 / M1104 = ON during instruction execution, Y0 / Y2 will pause immediately and M1538 / M1540 = ON indicates the pause status. When M1078 / M1104 = OFF, M1538 / M1540 = OFF, Y0 / Y2 will proceed to finish the remaining pulses.

1117. DRVI instruction supports Alignment Mark and Mask function. Please refer to the explanation in API 59, the PLSR instruction.

1118. D_2 doesn't support the index registers E, F modification.

Program Example:

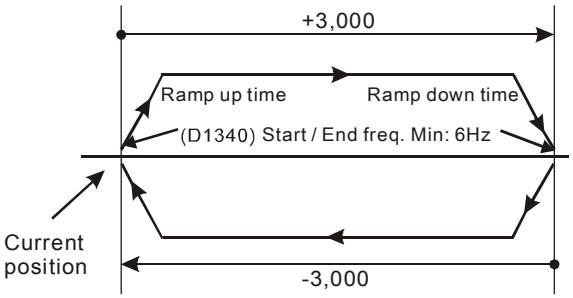
When M10= ON, 20,000 pulses (relative position) at 2kHz frequency will be generated from Y0. Y1= OFF indicates positive direction.



Points to note:

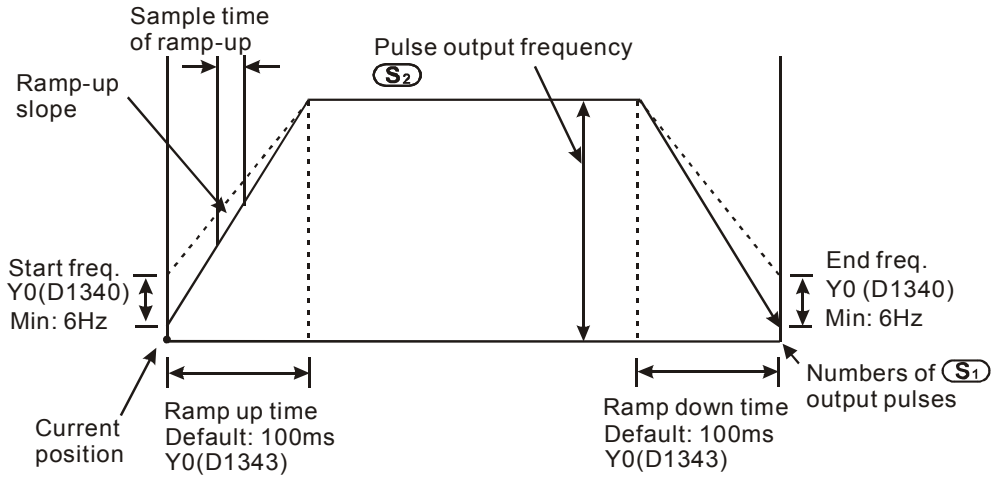
1119. Operation of relative positioning:

The Pulse output executes according to the relative distance and direction from the current position.



1120. Registers for setting ramp up/down time and start/end frequency:

a) Output Y0:



b) This instruction can be used many times in the program, but only one instruction will be activated at a time.

c) After activating the instruction, parameters cannot be modified unless the instruction is OFF.

1121. Related Flags:

M1029: CH0 (Y0, Y1) pulse output execution complete.

M1102: CH1 (Y2, Y3) pulse output execution complete

M1078: CH0 (Y0, Y1) pulse output pause (immediate)

M1104: CH1 (Y2, Y3) pulse output pause (immediate)

M1108: CH0 (Y0, Y1) pulse output pause (ramp down).

M1110: CH1 (Y2, Y3) pulse output pause (ramp down)

M1156: Enabling the mask and alignment mark function on I400/I401(X4)

corresponding to Y0.

M1158: Enabling the mask and alignment mark function on I600/I601(X6) corresponding to Y2.

M1305: Reverse Y1 pulse output direction in high speed pulse output instructions

M1306: Reverse Y3 pulse output direction in high speed pulse output instructions

M1347: Auto-reset Y0 when high speed pulse output complete

M1524: Auto-reset Y2 when high speed pulse output complete

M1534: Enable ramp-down time setting on Y0. Has to be used with D1348

M1535: Enable ramp-down time setting on Y2. Has to be used with D1349.

M1538: Indicating pause status of CH0 (Y0, Y1)

M1540: Indicating pause status of CH1 (Y2, Y3)

1122. Special D registers:

D1030: Low word of the present value of Y0 pulse output

D1031: High word of the present value of Y0 pulse output

D1336: Low word of the present value of Y2 pulse output

D1337: High word of the present value of Y2 pulse output

D1340: Start/end frequency of the 1st group pulse output CH0 (Y0, Y1)

D1352: Start/end frequency of the 2nd group pulse output CH1 (Y2, Y3)

D1343: Ramp up/down time of the 1st group pulse output CH0 (Y0, Y1)

D1353: Ramp up/down time of the 2nd group pulse output CH1 (Y2, Y3)

D1348: CH0(Y0, Y1) pulse output. When M1534 = ON, D1348 stores the ramp-down time

D1349: CH1(Y2, Y3) pulse output. When M1535 = ON, D1349 stores the ramp-down time

D1232: Number of output pulses for ramp-down stop when Y0 masking sensor receives signals. (LOW WORD)

D1233: Number of output pulses for ramp-down stop when Y0 masking sensor receives signals. (HIGH WORD).

D1234: Number of output pulses for ramp-down stop when Y2 masking sensor receives signals (LOW WORD).

D1235: Number of output pulses for ramp-down stop when Y2 masking sensor receives signals (HIGH WORD).

D1026: Number of pulses for masking Y0 when M1156 = ON (Low word)

D1027: Number of pulses for masking Y0 when M1156 = ON (High word)

D1135: Number of pulses for masking Y2 when M1158 = ON (Low word)

D1136: Number of pulses for masking Y2 when M1158 = ON (High word)

API	Mnemonic	Operands	Function
159	D DRVA	S_1, S_2, D_1, D_2	Absolute Position Control

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	DDRVA: 17 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D_1		*														
D_2		*	*	*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Numbers of pulses (Target device) S_2: pulse output frequency D_1: pulse output address

D_2: Rotation direction signal

Description: (ELC- PV, ELC2-PV)

1123. S_1 is specified as the number of output pulses (absolute position). The 16-bit instruction range is -32,768 ~ +32,767. The range for the 32-bit instruction is -2,147,483,648 ~ +2,147,483,647. The positive (+) and negative (-) symbol indicates the forward and reverse direction.

1124. S_2 is specified as the pulse output frequency. The 16-bit instruction range is 10 ~ 32,767Hz. The range for the 32-bit instruction is 10 ~ 200,000Hz.

1125. Has four groups of A/B phase pulse output, CH0 (Y0, Y1), CH1 (Y2, Y3), CH2 (Y4, Y5) and CH3 (Y6, Y7).

1126. D_1 is pulse output address:

◆ Model	◆ ELC-PV, ELC2-PV
◆ Pulse output end	◆ Y0, Y2, Y4, Y6

1127. D_2 is specified as rotation direction signal.

1128. When S_1 is larger than the current absolute position, D_2 is OFF.

1129. When S_1 is less than the current absolute position, D_2 is ON.

1130. After the pulse output completes, D_2 will not immediately turn OFF. D_2 will be ON until the conditions preceding the instruction are OFF.

1131. S_1 is

- The 32-bit data stored in the present value registers (D1337, D1336) for CH0 (Y0, Y1).

- The 32-bit data stored in the present value registers (D1339, D1338) for CH1 (Y2, Y3).
- The 32-bit data stored in the present value registers (D1376, D1375) for CH2 (Y4, Y5).
- The 32-bit data stored in the present value registers (D1378, D1377) for CH3 (Y5, Y6).

1132. When in the reverse direction, the contents of the present value register will decrease.

1133. When the DRVA instruction is sending pulses, you cannot change the operands. The changes will not be valid until the next time the instruction is enabled.

1134. When the conditions preceding the DRVA instruction are OFF, the pulse output will decelerate to a stop and M1029 and M1030 will turn ON.

1135. When the conditions preceding the DRVA instruction are OFF, and the flag M1336 for the CH0 pulses or M1337 for the CH1 pulses are “ON”, the DRVA instruction will not execute.

1136. When the absolute value of the input frequency of DRVA and DDRVA instructions is greater than 200KHz, the output will be at 200KHz. When the absolute value of the input frequency is less than 10Hz, the output will be at 10Hz.

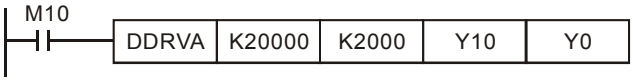
1137. D1343 (D1353) is the acceleration/deceleartion time of the first step acceleration and the last step deceleration. The acceleration and deceleration time shall not be shorter than 10ms. The output will be set to 10ms if the time is shorter than 10ms and set to 100ms if the time is longer than 10,000ms.

1138. M1305 (M1306) is the direction signal for CH0 (CH1). When $\mathbf{S_1}$ is a positive number, the output will be operated in a forward direction and M1305 (M1306) will be OFF. When $\mathbf{S_1}$ is a negative number, the output will be operated in the reverse direction and M1305 (M1306) will be ON.

1139. Flags: Refer to API 158 DRVI for information on M1010, M1102, M1103, M1334, M1335, M1336, M1337, M1346

Program Example:

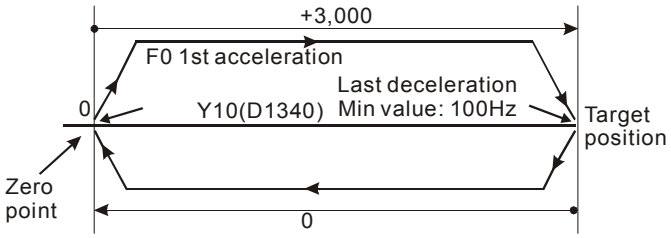
When M10=ON, it will output 20,000 pulses from Y10 with 2KHz frequency. Y0=ON indicates positive direction.



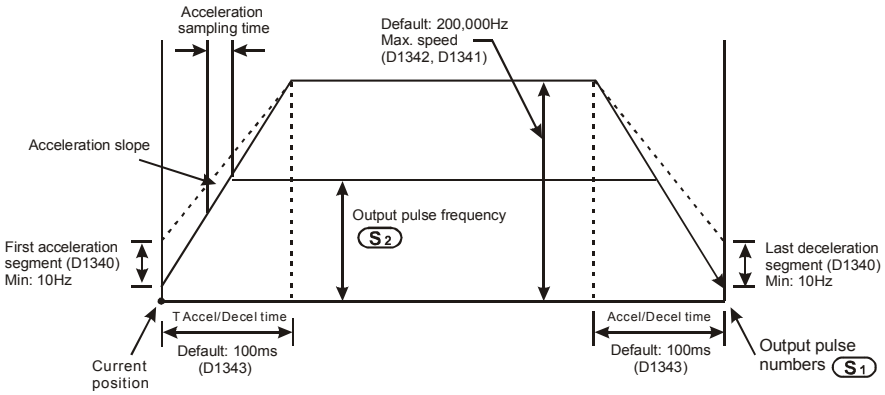
Points to note:

1. :

a) Absolute position control: The travel distance starting from the zero point (0); also known as an absolute driving method.



b) Settings for absolute positioning and the acceleration/deceleration speed:



2. When this instruction is OFF and the output is not complete:

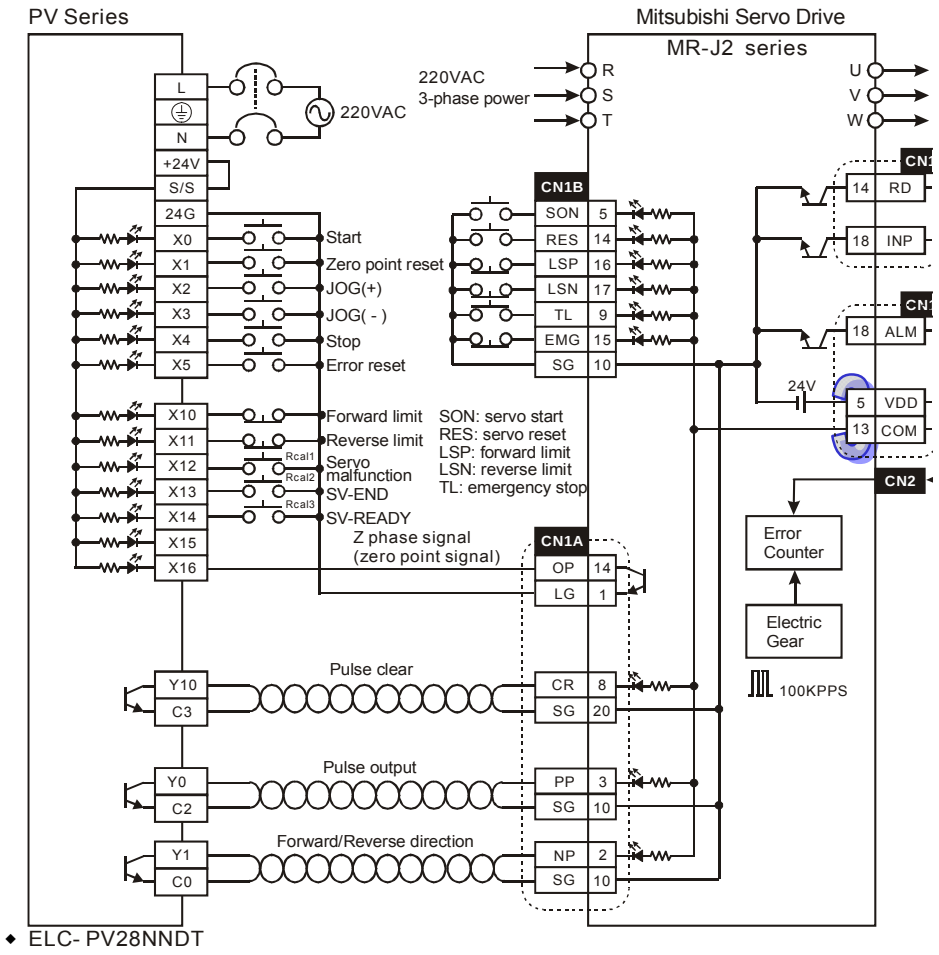
   If M1334=ON, Y10 will stop the pulse output immediately.

   If M1334=OFF, Y10 will decelerate from the deceleration time to the end frequency and then stop the pulse output.

   It is the same way for M1335 with respect to Y11 output.

3. Please refer to the DDRVI instruction: “Remarks for flag information.

4. Wiring of the ELC-PV and a Servo drive:



Note:

(a) When detecting an absolute position by using DABSR instruction, the parameter settings of a Mitsubishi MR-J2-□A servo drive that connects to ELC-PV:

P0: position mode.

P1: using absolute value.

P21: pulse input type as Pulse+DIR.

(b) The forward/reverse limit switch should be connected to SERVO AMP.

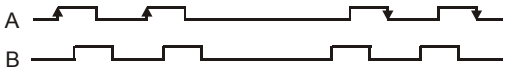
(c) When using OP (Z-phase signal) in a servo and given that the Z-phase signal is a high-frequency signal, when the motor is running at high speed, the valid detection can only be possible when the signal is within the range detectable by the ELC. When using OP (Z phase signal) of the servo, if the Z phase signal is a high frequency signal during high-speed motor operation, the high frequency signal shall be within the available range that can be detected by the ELC.

5. Settings of the pulse output signals in the operation of position control :

a) Pulse + DIR (recommended)

b) CW/CCW (limited frequency at 10KHz)

c) A/B-phase output (limited frequency at 10KHz)

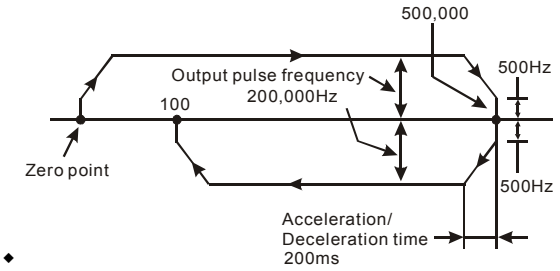


6. When the Y0 output is used with many high-speed pulse output instructions (PLSY, PWM, PLSR) and position control instructions (ZRN, PLSV, DRVI, DRVA) in a program and these instructions are executed synchronously in the same program scan, the ELC will execute the instruction with the fewest step numbers.

Programming example for forward/reverse operation:

For the wiring, see the wiring drawing of the ELC-PV series and the servo drive above.

One operation mode performs positioning by absolute position:



Explanations: (ELCM-PH/PA,ELC2-PB/PH/PA/PE)

1140. S_1 is the number of pulses (Absolute positioning). Available range: -2,147,483,648 ~ +2,147,483,647. "+/-" signs indicate forward and reverse direction.

1141. S_2 is the pulse output frequency. Available range: 6 ~ 100,000Hz.

1142. D_1 is the pulse output address. It can use CH0 (Y0) or CH1 (Y2).

1143. D_2 is the direction signal output address. If a Y output is used, only CH0 (Y1) and CH1 (Y3) are available.

1144. S_1 is the target position for absolute positioning. The actual number of output pulses (S_1 – current position) will be calculated by the ELC. When the result is positive, the pulse output executes in the forward direction, i.e. D_2 = OFF; when the result is negative, the pulse output executes in the reverse direction, i.e. D_2 = ON.

1145. The value in S_1 is the absolute position from the zero point. The calculated actual number of output pulses will be the relative position of:

- The current position (32-bit data) of CH0 (Y0, Y1) which is stored in D1031(high), D1030 (low)
- The current position (32-bit data) of CH1 (Y2, Y3) which is stored in D1337(high), D1336 (low).

◆ In the reverse direction of the pulse output, the value in (D1031, D1330) and (D1336, D1337) decreases.

1146. D1343 (D1353) is the ramp up/down time setting of CH0 (CH1). Available range: 20 ~ 32,767ms. Default: 100ms. The ELC will use the upper/lower value as the set value when the value exceeds the allowable range.

1147. D1340 (D1352) is the start/end frequency setting of CH0 (CH1). Available range: 6 to 100,000Hz. The ELC will use the upper/lower value as the set value when the value exceeds the allowable range.

1148. M1305 and M1306 can change the output direction of CH0/CH1 set in D_2. When **S** is "-", D_2 will be ON. If M1305/M1306 is set ON before the instruction executes, D_2 will be OFF during execution of the instruction.

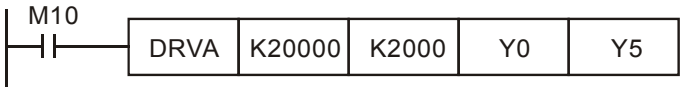
1149. The ramp-down time of CH0 and CH1 can be modified by using (M1534, D1348) and (M1535, D1349). When M1534 / M1535 = ON, CH0 / CH1 ramp-down time is specified by D1348 / D1349.

1150. If M1078 / M1104 = ON during instruction execution, Y0 / Y2 will pause immediately and M1538 / M1540 = ON indicates the pause status. When M1078 / M1104 = OFF, M1538 / M1540 = OFF, Y0 / Y2 will proceed to finish the remaining pulses.

1151. DRVA/DDRVA instructions do NOT support Alignment Mark and Mask function.

Program Example:

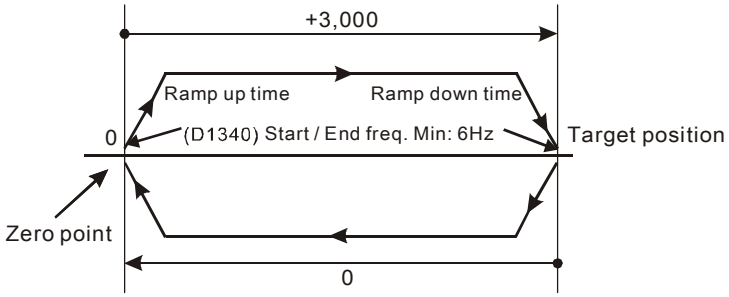
When M10 = ON, the DRVA instruction executes absolute positioning on Y0 at the target position 20000 and the target frequency 2kHz. Y5 = OFF indicates positive direction.



Points to note:

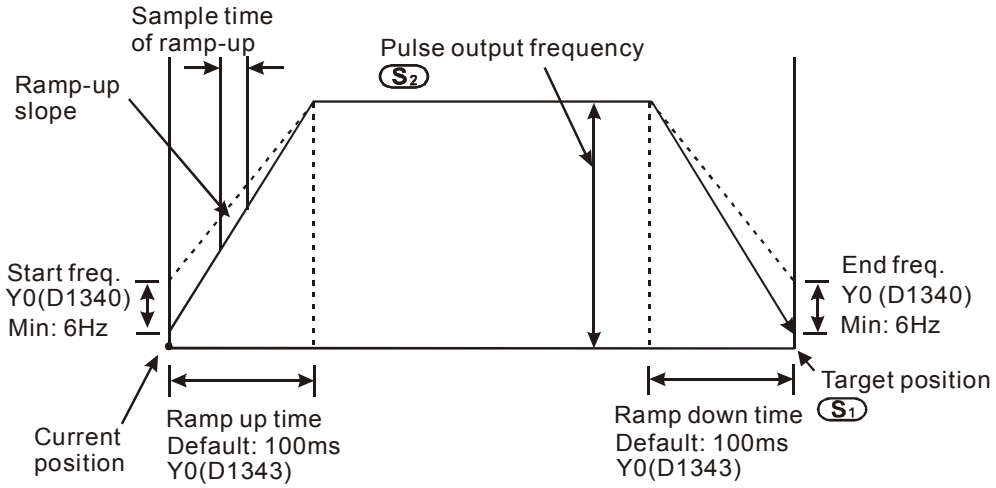
1152. Operation of absolute positioning:

Pulse output executes according to the specified absolute position from zero point



1153. Registers for setting ramp up/down time and start/end frequency:

a) Output Y0:



b) This instruction can be used many times in the program, but only one instruction will be allowed to be active at a time. The instructions first activated will be first executed.

c) After activating the instruction, parameters cannot be modified unless the instruction is OFF.

d) For related special flags and special registers, please refer to **Points to note** of DDRVI instruction.

◆

API	Mnemonic		Operands	Function
160	TCMP	P	S_1, S_2, S_3, S, D	Calendar Compare

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	TCMP, TCMPP: 11 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
S_3					*	*	*	*	*	*	*	*	*	*	*	
S											*	*	*			
D		*	*	*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: hours of comparison time, range is K0~K23 S_2: minutes of comparison time, range is K0~K59 S_3: seconds of comparison time, range is K0~K59 **S**: Current clock information (occupies 3 continuous addresses) **D**: Comparison result (occupies 3 continuous addresses)

Description:

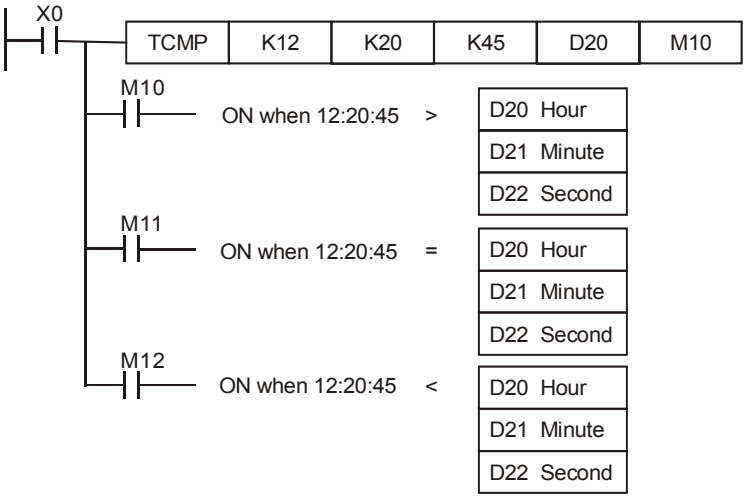
1154. S_1, S_2, S_3 are compared to the current values of the real-time clock **S** and save the comparison result in **D**.

1155. **S** is the hour of the current time and the range is K0~K23. **S** +1 is the minutes of the current time and the range is K0~K59. **S** +2 is the seconds of the current time and the range is K0~K59.

1156. The current time of the RTC specified by **S** is read by using the TRD instruction previously in the program and then compared by using TCMP instruction. If the contents of **S** exceeds the range, it will result in “operation error”, the instruction will not execute and M1067=ON, M1068=ON, and the error code 0E1A (HEX) will be in D1067.

Program Example:

1157. When X0= ON, the instruction is executed and the current time of the calendar is placed in (D20~D22) and is compared to the value 12:20:45 and the result is shown at M10~M12. When X0 goes from ON→OFF, the instruction stops executing, but the ON/OFF state of M10~M12 is unchanged..



API	Mnemonic		Operands	Function
161	TZCP	P	S_1, S_2, S, D	Calendar Zone Compare

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			TZCP, TZCPP: 9 steps
S_2											*	*	*			
S											*	*	*			
D		*	*	*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Lower limit time data S_2: Upper limit time data **S**: Current time of the RTC

D: Comparison result (occupies 3 continuous addresses)

Operands S_1, S_2, **S** occupy 3 continuous addresses S_1 should be less than S_2

Description:

1158. **S** is compared to the time of S_1~ S_2 and the comparison result is stored in **D**.

1159. S_1, S_1 +1, S_1 +2: represents “Hour”, “Minute”, “Second” of the lower limit time data.

1160. S_2, S_2 +1, S_2 +2: represents “Hour”, “Minute”, “Second” of the Upper limit time data。

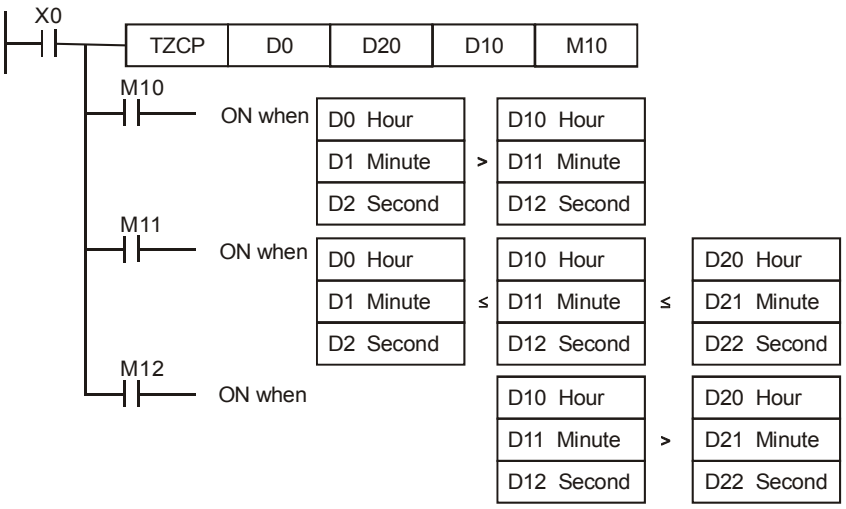
1161. **S**, **S** +1, **S** +2: represents “Hour”, “Minute”, “Second” of the current time of calendar.

1162. The current time of the calendar specified by **S** is read by using the TRD instruction previously in the program and then compared by using TZCP instruction. If the content of **S**, S_1, S_2 exceeds the range, it will result in “operation error”. At this time, the instruction will not be executed and M1067=ON, M1068=ON, and error code 0E1A (HEX) will be written to D1067.

1163. If **S** < S_1 and **S** < S_2, **D** is ON. If **S** > S_1 and **S** > S_2, **D**+2 is ON. Except for these two situations, **D** +1 is ON. (Lower limit S_1 should be less than upper limit S_2.)

Program Example:

When X0= ON, the instruction is executed and one of M10~M12 = ON. When X0=OFF, the instruction is not executed but the state of M10~M12 remain unchanged.



API	Mnemonic		Operands	Function
162	TADD	P	S_1, S_2, D	Calendar Data Addition

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			TADD, TADDP: 7 steps
S_2											*	*	*			
D											*	*	*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Time augend **S_2**: Time addend **D**: Addition result

Operand **S_1**, **S_2**, **D** occupies 3 continuous addresses

Description:

1164. **$S_1 + S_2 = D$**. The time data in the registers specified by **S_1** is added to the time data in the registers specified by **S_2** and the result is stored in the registers specified by **D**.

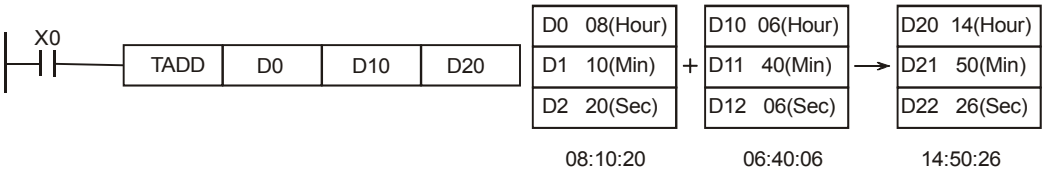
1165. If the time data in **S_1**, **S_2** exceeds the allowable range, it will result in "operation error". At this time, the instruction will not be executed and M1067=ON, M1068=ON, and error code 0E1A (HEX) will be written to D1067.

1166. If the addition result is a value greater than 24 hours, the Carry flag M1022=ON. The value of the result shown in **D** is the time remaining above 24 hours.

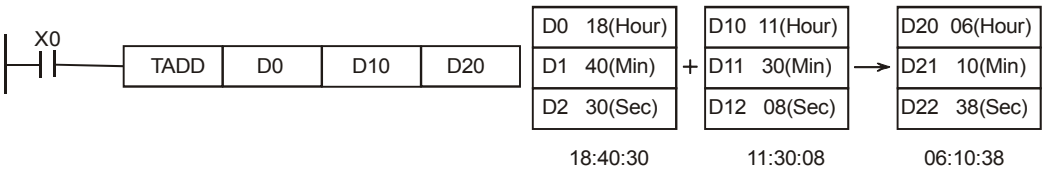
1167. If the addition result is equal to 0 (zero, 0 hour, 0 minute, 0 second), the Zero flag M1020= ON.

Program Example:

When X0= ON, the instruction is executed. Add the time data specified by D0~D2 and D10~D12 and store the result in the register specified by D20~D22.



If the addition result is in a value greater than 24 hours, the Carry flag M1022=ON.



API	Mnemonic		Operands	Function
163	TSUB	P	S_1, S_2, D	Calendar Data Subtraction

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			TSUB, TSUBP: 7 steps
S_2											*	*	*			
D											*	*	*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Time Minuend S_2: Time Subtrahend **D**: Subtraction result

Operand S_1, S_2, **D** occupies 3 continuous addresses.

Description:

1168. $S_1 - S_2 = D$. The time data in the registers specified by S_2 is subtracted from the time data in the registers specified by S_1 and the result is stored in the registers specified by **D**.

1169. If the time data in S_1, S_2 exceeds the allowable range, it will result in "operation error". At this time, the instruction won't be executed and M1067=ON, M1068=ON, and error code 0E1A (HEX) will be written to D1067.

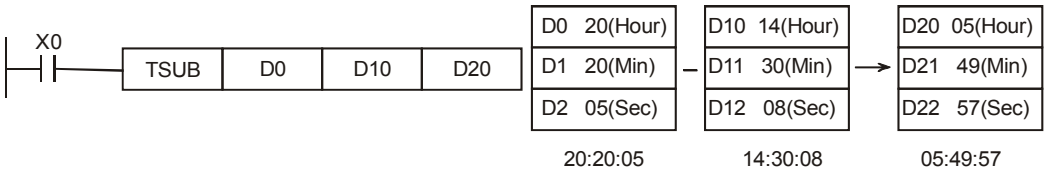
1170. If the subtraction result is a negative value, the Zero flag M1020= ON. The value of the result shown in **D** is the time remaining.

1171. If the subtraction result is equal to 0 (zero, 0 hour, 0 minute, 0 second), the Zero flag M1020= ON.

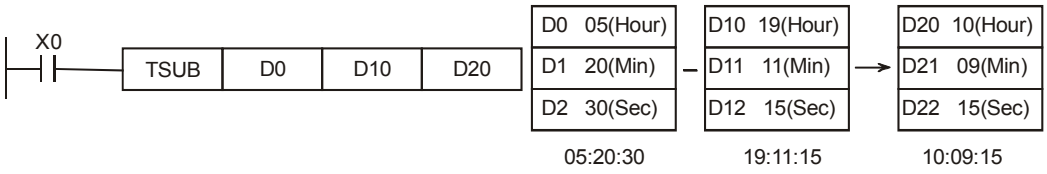
1172. Besides using the TRD instruction, the MOV instruction can also be used to move the special registers D1315 (Hour), D1314 (Minute), D1313 (Second) to the three registers specified to read the current time of the clock/calendar.

Program Example:

When X0= ON, the instruction is executed. The time data specified by D10~D12 is subtracted from the time data specified by D0~D2 and the result is stored in the register specified by D20~D22.



If the subtraction result is a negative value (less than 0), the borrow flag M1021= ON.



API	Mnemonic		Operands	Function
166	TRD	P	D	Calendar Data Read

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D											*	*	*			TRD, TRDP: 3 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

D: The starting address of the current RTC(clock/calendar) data (occupies 7 continuous words)

Description:

1173. The RTC,provides the data about the year, week, month, date, hours, minutes and seconds, total 7 data words stored in D1319~D1313. The TRD instruction reads the current clock/calendar information and stores the values in the 7 data registers specified by **D**.

1174. D1319 is read as a two digit number for the year, but can be changed to a four digit number per the notes below.

1175. The Real Time Clock in the ELC2-PB, ELCM-PH/PA version 1.0 maintains normal operation only while powered. The RTC data registers D1319~D1313 are latched. When power is re-applied, the RTC will resume from where it left off. Therefore, it is recommended that each time the controller is powered that the current RTC information be moved into D1313-D1319 using the TWR (API 167) instruction. For the ELC-PV/PA, ELC2-PA/PH/PE/PV, ELCM-PH/PA version 2.0 (and above), the RTC information in D1313-D1319 will continue when power is turned off to the controller. If the RTC information in D1313-D1319 was correct when the controller was powered down, when power is re-applied to the unit the RTC information will be correct.

1176. Flags: M1016, M1017, M1076, please see the Notes below.

Program Example:

When X0=ON, read the current time of calendar to the specified register D0~D6.

The content of D1318: 1 is Monday, 2 is Tuesday, 7 is Sunday.

Special D device	Meaning	Content		General D device	Meaning
D1319	Year (A.D.)	00~99	→	D0	Year (A.D.)
D1318	Day (Mon.~Sun.)	1~7	→	D1	Day (Mon.~Sun.)
D1317	Month	1~12	→	D2	Month

Special D device	Meaning	Content		General D device	Meaning
D1316	Date	1~31	→	D3	Date
D1315	Hour	0~23	→	D4	Hour
D1314	Minute	0~59	→	D5	Minute
D1313	Second	0~59	→	D6	Second

Notes:

The best method for setting the RTC is to use the TWR (API 167) instruction.

1. To display a four digit number for the year:

   By default, ELC controllers display a 2 digit number for the year (for example: displays 03 for year 2003). If you want to display a 4 digit number for the year, set special M bit M1016 at power up, per the following:

2. The RTC will display 4 digits in place of the original 2 digit number.

   When using the 4-digit format for the year and a new value needs to be written to D1319, writing a 2-digit value using the TWR instruction will result in a 4-digit value for the year if M1016 is set. Writing a 4-digit value to D1319 when M1016 is set is not necessary. For example, 00=year 2000, 50=year 2050 and 99=year 2099.

3. Error Flag of the calendar:

Device	Name	Function
M1016	year format for the calendar	Displays the 2 right-most digits for the year in D1319 when M1016 is OFF. Displays the 4-digit value for the year when M1016 is ON.
M1017	±30 seconds correction	A false-to-true transition of this bit will result in the clock changing the following: if it is 0-29 seconds, it will reset to 0. If it is 30-59 seconds, add 1 to minutes and reset seconds to 0.
M1076	calendar fault	ON when RTC settings are out of range or the battery is no longer working.

Device	Name	Range
D1313	Second	0-59

Device	Name	Range
D1314	Minute	0-59
D1315	Hour	0-23
D1316	Day	1-31
D1317	Month	1-12
D1318	Week	1-7
D1319	Year	0-99 (two right-most digit number of year)

API	Mnemonic		Operands	Function
167	TWR	P	D	Calendar Data Write In

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D											*	*	*			TWR, TWRP: 5 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

D: Starting address for new RTC data (occupies 7 continuous words)

Description:

1177. This instruction writes the values in **D** into the RTC registers.

1178. If any of the RTC data in **D** is out of the allowable range, it will result in an “operation error”. The instruction will not execute and M1067=ON, M1068=ON, and error code 0E1A (HEX) will be in D1067.

1179. To make adjustments to the RTC, use this instruction to write correct data into the RTC registers (D1313-D1319).

1180. Flags: M1016, M1017, M1076. Refer to the TRD instruction for information on these flags.

Program Example 1:

When X0= ON, write the new time/date into the RTC.

	General D device	Meaning	Content		Special D device	Meaning	
New setting time	D20	Year (A.D.)	00~99	→	D1319	Year (A.D.)	Calendar Clock
	D21	Day (Mon.~Sun.)	1~7	→	D1318	Day (Mon.~Sun.)	
	D22	Month	1~12	→	D1317	Month	
	D23	Date	1~31	→	D1316	Date	
	D24	Hour	0~23	→	D1315	Hour	
	D25	Minute	0~59	→	D1314	Minute	
	D26	Second	0~59	→	D1313	Second	

Program Example 2:

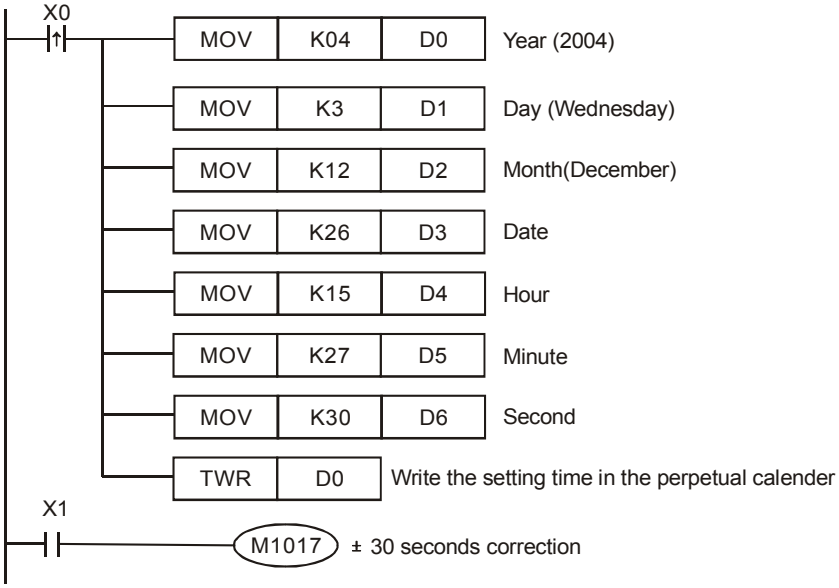
Adjust the clock/calendar information to 2004/12/15, Tuesday, 15:27:30.

The contents of D0~D6 is the RTC data.

When X0= ON, move the new RTC data to D0-D6 and execute the TWR instruction to write the new

RTC data to D1313-D1319.

When X1=ON, the calendar clock will perform the ±30 seconds correction. “Correction” means that if the seconds of the clock is 1~29 seconds, the seconds value will be automatically set to “0” (zero) seconds and the minutes will not change. If the seconds of the clock are between 30~59 seconds, the seconds value will be automatically set to “0” (zero) seconds but the minutes will increase by 1 minute.



API	Mnemonic			Operands	Function
168	D	MVM	P	S_1, S_2, D	Masked Move

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*	*	*	MVM, MVMP: 7 steps DMVM, DMVMP: 13 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D							*	*	*	*	*	*	*	*	*	

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Source address 1 S_2: Bits to be masked (OFF) **D**: Source device 2 / Operation results

[**D** = (S_1 & S_2) | (**D** & ~S_2)]

Description:

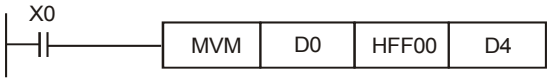
1181. This instruction performs a move with mask operation. This means that data is moved bit by bit from S_1 through the mask S_2 to **D.** Bit data will be moved from S_1 to **D** only where there is a logical 1 in that position in the mask.

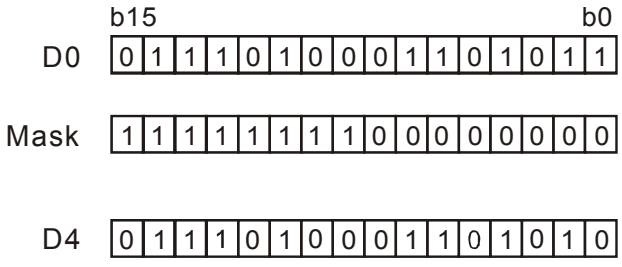
1182. Where there is a logical 1 in the mask, data is passed. Where there is a logical 0 in the mask, the data is blocked, i.e. the destination bit is unaffected.

1183. MVM is the new instruction for ELC-PV V1.6.

Program Example:

When X0 = ON, the MVM instruction moves the data from D0 to D4 bit by bit, only allowing bit data to be passed where there is a logical 1 in the mask. The bit positions where there is a logical 0 in the mask leaves the destination bit position unaffected. The value in D4 prior to executing the MVM instruction is: 1000110001101010.





Note: The low 8 bits of D4 are unchanged. The high 8 bits of D0 are moved to the high 8 bits of D4.

API	Mnemonic		Operands	Function
169	D	HOUR	S, D_1, D_2	Hour Meter

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	HOUR: 7 steps
D_1													*			DHOUR: 13 steps
D_2		*	*	*												

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Hours set-point value **D_1**: Current time (occupies 2 continuous addresses) **D_2**: Output address

Description:

1184. **D_2** turns ON when the current accumulated time equals **S** (in hours), and the range is K1~K32,767. **D_1** range is K0~K32,767. **D_1**+1 is the current time in seconds. Its range is K0~K3,599.

1185. This instruction is used to measure the number of hours a process is running. This can be used for example, to measure the time a machine is running to plan for scheduled maintenance.

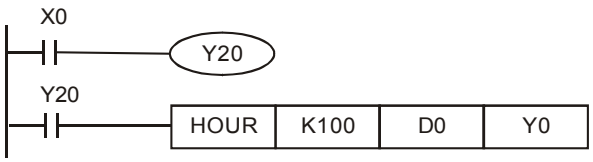
1186. After the output **D_2** is ON, the timer will keep on timing.

1187. When the 16-bit timer counts to its max. value (32,767 hours and 3,599 seconds), it will stop. When the 32-bit timer counts to its max. value (2,147,483,647 hours and 3,599 seconds), it will stop.

1188. This Instruction can be used four times in the program.

Program Example:

For 16-bit instruction: When X0=ON, Y20 = ON, timing starts. When the time reaches 100 hours, Y0 = ON and D0 will record the current time (units are hours for D0 and, D1 will contain the number of seconds that have elapsed (range is 0~3599 seconds).



API	Mnemonic			Operands	Function
170	D	GRY	P	**S, D**	BIN → GRAY CODE

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	GRY, GRYP: 5 steps
D								*	*	*	*	*	*	*	*	DGRY, DGRYP: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Source value **D**: Destination address to store the Gray code result

Description:

1189. The BIN value, **S** is converted to the equivalent GRAY CODE value and the result is stored in the device specified by **D**.

1190. The range of **S** that can be converted to GRAY CODE is shown as follows:

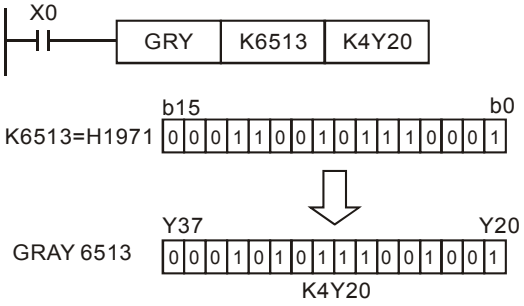
16-bit instruction: 0~32,767

32-bit instruction: 0~2,147,483,647

1191. If the BIN value is outside the range shown above, it is an “Operation Error”. The instruction will not be executed, M1067=ON, M1068=ON, and the error code 0E1A (HEX) in D1067.

Program Example:

When X0=ON, constant K6513 is converted to GRAY CODE and stored in the K4Y20.



API	Mnemonic			Operands	Function
171	D	GBIN	P	S, D	GRAY CODE → BIN

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	GBIN, GBINP: 5 steps
D								*	*	*	*	*	*	*	*	DGBIN, DGBINP: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Source GRAY CODE **D**: Destination address for the converted BIN result

Description:

1192. The GRAY CODE value in **S** is converted to an equivalent BIN value and the result is stored in the device specified by **D**.

1193. This instruction can be used to read the value from an absolute position type encoder (a gray code encoder) which is connected to the ELC inputs. Convert the value to a BIN value and store it in the specified register.

1194. If the source is inputs X20~X37, it can speed up the input response time by using the REFF instruction or D1020 (adjust input response time).

1195. The range of **S** that can be converted to the GRAY CODE is as follows:

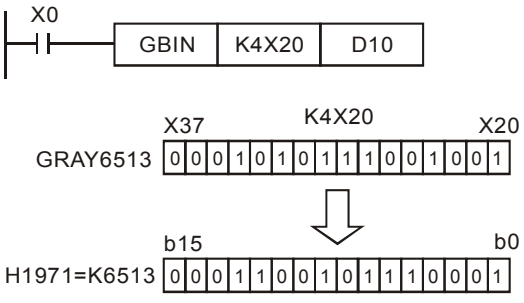
16-bit instruction: 0~32,767

32-bit instruction: 0~2,147,483,647

1196. If the GRAY CODE value is outside the range shown above, it is an “Operation Error”.

Program Example:

When X0=ON, the GRAY CODE value in the absolute position type encoder connected to X20~X37 inputs is converted to BIN value and stored in D10.



API	Mnemonic			Operands	Function
172	D	ADDR	P	S_1, S_2, D	Floating Point Number Addition

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			DADDR, DADDRP: 13 steps
S_2													*			
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

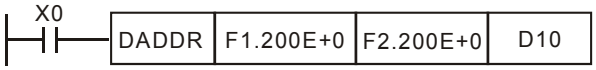
S_1: Floating point number summand **S_2:** Floating point number addend **D:** Sum

Description:

1. S_1 and S_2 are Floating point numbers
2. ELCB-PB model does not support index register E and F modification.
3. The DADDR operands, **S_1** and **S_2** can each be 2 D-register addresses or actual floating point values (e.g. F1.2).
4. When **S_1** and **S_2** use D registers, the function will be the same as **API** 120 EADD.
5. If the absolute value of the operation result > the maximum floating point number, the carry flag M1022=On.
6. If the absolute value of operation result < the minimum floating point number, the borrow flag M1021=On.
7. If the operation result equals 0, the zero flag M1020=On.

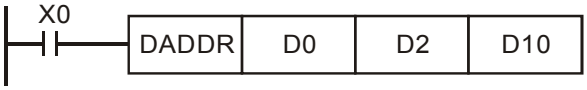
Program Example 1:

When X0 = ON, add floating point number F1.200E+0 (enter F1.2, scientific notation F1.200E+0 will be displayed in the instruction.) to F2.200E+0 and store the result F3.400E+0 in register D10 and D11.



Program example 2:

When X0=On, store the result of (D1, D0)＋(D3, D2) into (D11, D10).



API	Mnemonic			Operands	Function
173	D	SUBR	P	S_1, S_2, D	Floating Point Number Subtraction

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			DSUBR, DSUBRP: 13 steps
S_2													*			
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

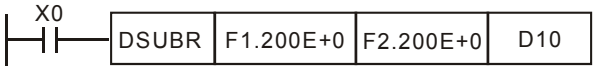
S_1: Floating point number minuend **S_2:** Floating point number subtrahend **D:** Result

Description:

1. S_1 and S_2 are Floating point numbers
2. ELCB-PB model does not support index register E and F modification.
3. The DSUBR operands, **S_1** and **S_2** can each be 2 consecutive D-register addresses or actual floating point values (e.g. F1.2).
4. When **S_1** and **S_2** use D registers, the function will be the same as **API** 120 ESUB.
5. If the absolute value of the operation result > the maximum floating point numbers, the carry flag M1022=On.
6. If the absolute value of operation result < the minimum floating point numbers, the borrow flag M1021=On.
7. If the operation result equals 0, the zero flag M1020=On.

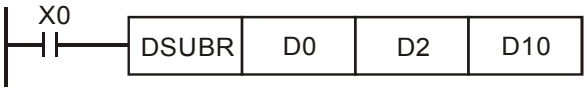
Program example 1:

When X0 = ON, subtract floating point value F1.200E+0 (enter F1.2, scientific notation F1.200E+0 will be displayed in the instruction.) with F2.200E+0 and store the result F-1.000E+0 in register D10 and D11.



Program example 2:

When X0=On, store the result of floating point number value (D1, D0) － (D3, D2) into (D11, D10).



API	Mnemonic			Operands	Function
174	D	MULR	P	**S_1, S_2, D**	Floating Point Number Multiplication

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			DMULR, DMULRP: 13 steps
S_2													*			
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

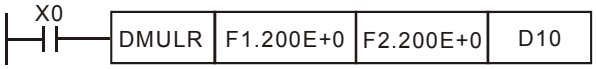
S_1: Floating point number multiplicand **S_2:** Floating point number multiplicator **D:** Product

Description:

1. S_1 and S_2 are Floating point numbers
2. ELCB-PB model does not support index register E and F modification.
3. The DMULR operands, **S_1** and **S_2** can each be 2 consecutive D-register addresses or actual floating point values (e.g. F1.2).
4. When **S_1** and **S_2** use D registers, the function will be the same as **API** 122 EMUL.
5. If the absolute value of the operation result > the maximum floating point numbers, the carry flag M1022=On.
6. If the absolute value of operation result < the minimum floating point numbers, the borrow flag M1021=On.
7. If the operation result equals 0, the zero flag M1020=On.

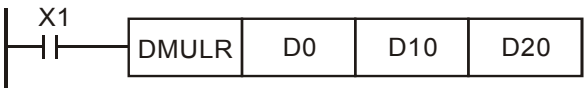
Program Example 1:

When X0= ON, multiply floating point number F1.200E+0 (enter F1.2, scientific notation F1.200E+0 will be displayed in the instruction.) by F2.200E+0 and store the result F2.640E+0 in register D10 and D11.



Program example 2:

When X1=On, store the result of floating point number value (D1, D0) × (D11, D10) into (D21, D20).



API	Mnemonic	Operands	Function

175	D	DIVR	P	S_1, S_2, D	Floating Point Number Division

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			DDIVR, DDIVRP: 13 steps
S_2													*			
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

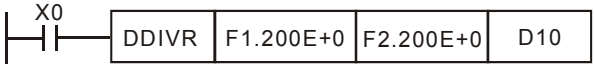
S_1: Floating point number dividend **S_2:** Floating point number divisor **D:** Quotient

Description:

1. S_1 and S_2 are Floating point numbers
2. ELCB-PB model does not support index register E and F modification.
3. The DDIVR operands, **S_1** and **S_2** can each be 2 consecutive D-register addresses or actual floating point values (e.g. F1.2).
4. When **S_1** and **S_2** use D registers, the function will be the same as **API** 123 EDIV.
5. If **S_2** = 0, this is a bad operation", DDIVR will not execute, M1067, M1068=On, and D1067 records the error code H'0E19.
6. If the absolute value of the operation result > the maximum floating point numbers, the carry flag M1022=On.
7. If the absolute value of operation result < the minimum floating point numbers, the borrow flag M1021=On.
8. If the operation result equals 0, the zero flag M1020=On.

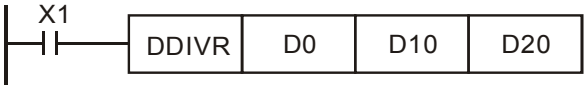
Program example 1:

When X0 = ON, divide floating point number F1.200E+0 (enter F1.2, scientific notation F1.200E+0 will be displayed in the instruction) by F2.200E+0 and store the result F0.545E+0 in D10 and D11.



Program example 2:

When X1=On, divide the floating point number value (D1, D0) by (D11, D10) and store the quotient into registers assigned by (D21, D20).



API	Mnemonic		Operands	Function
176	MMOV	P	S, D	16-bit→32-bit Conversion

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*			MMOV, MMOVP: 5 steps
D											*	*	*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

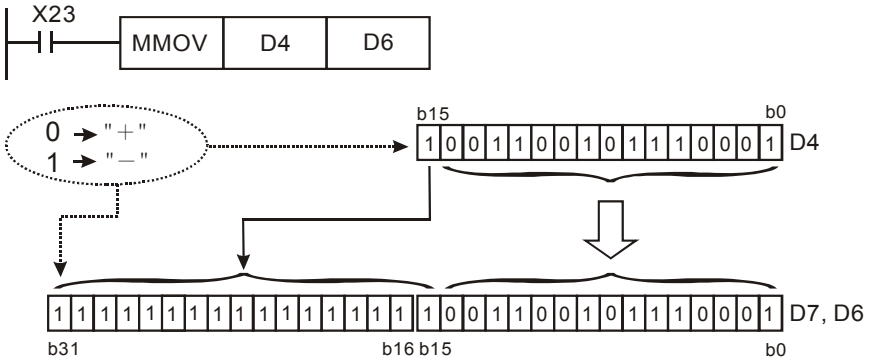
S: Source address (16-bit) **D:** Destination address (32-bit)

Description:

MMOV instruction sends the data in the 16-bit source **S** to **the** 32-bit destination **D**. The sign bit (MSB) of the source will be copied to the high word of **D** and the high bit of the low word of **D**.

Program example:

When X23 = 0N, 16-bit data in D4 will be moved to D6 and D7.



In the example above, b15 in D4 will be sent to b15~b31 of D7/D6.

API	Mnemonic	Operands	Function
177	GPS	**S, D**	GPS Data Receiving

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			GPS: 5 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Sentence identifier for GPS data received **D**: Destination address for feedback data

Description:

1. ELC2-PE doesn't support API177 GPS instruction.
2. The GPS instruction receives GPS data from COM1 (RS-232) only, with communication settings: 9600,8,N,1 protocol: NMEA-0183, and communication frequency: 1Hz.
3. Operand **S** is the sentence identifier for receiving GPS data. K0: $GPGGA, K1: $GPRMC.
4. Operand **D** stores the received data. Up to 17 consecutive words will be used. Please refer to the table below for a description of each **D** value.

When **S** is set to K0, the sentence identifier $GPGGA is specified. The **D** values are:

No.	*Content*	*Range*	*Format*	*Note*
***D** + 0*	*Hour*	*0 ~ 23*	*Word*	
***D** + 1*	*Minute*	*0 ~ 59*	*Word*	
***D** + 2*	*Second*	*0 ~ 59*	*Word*	
***D** + 3~4*	*Latitude*	*0 ~ 90*	*Float*	*Unit: dd.mmmmmm*
***D** + 5*	*North / South*	*0 or 1*	*Word*	*0(+)→North, 1(-)→South*
***D** + 6~7*	*Longitude*	*0 ~ 180*	*Float*	*Unit: ddd.mmmmmm*
***D** + 8*	*East / West*	*0 or 1*	*Word*	*0(+)→East, 1(-)→West*
***D** + 9*	*GPS data valid / invalid*	*0, 1, 2*	*Word*	*0 = invalid*
***D** + 10~11*	*Altitude*	*0 ~9999.9*	*Float*	*Unit: meter*
D + 12~13	*Latitude*	*-90 ~ 90*	*Float*	*Unit: ±dd.ddddd*
D + 14~15	*Longitude*	*-180 ~ 180*	*Float*	*Unit: ±ddd.ddddd*

- When **S** is set to K1, the sentence identifier $GPRMC is specified. The **D** values are:

No.	*Content*	*Range*	*Format*	*Note*

No.	Content	Range	Format	Note
D *+ 0*	*Hour*	*0 ~ 23*	*Word*	
D *+ 1*	*Minute*	*0 ~ 59*	*Word*	
D *+ 2*	*Second*	*0 ~ 59*	*Word*	
D *+ 3~4*	*Latitude*	*0 ~ 90*	*Float*	*Unit: dd.mmmmmm*
D *+ 5*	*North / South*	*0 or 1*	*Word*	*0(+)→North, 1(-)→South*
D *+ 6~7*	*Longitude*	*0 ~ 180*	*Float*	*Unit: ddd.mmmmmm*
D *+ 8*	*East / West*	*0 or 1*	*Word*	*0(+)→East, 1(-)→West*
D *+ 9*	*GPS data valid / invalid*	*0, 1, 2*	*Word*	*0 = invalid*
D *+ 10*	*Day*	*1 ~ 31*	*Word*	
D *+ 11*	*Month*	*1 ~ 12*	*Word*	
D *+ 12*	*Year*	*2000 ~*	*Word*	
D + 13~14	*Latitude*	*-90 ~ 90*	*Float*	*Unit: ±dd.ddddd*
D + 15~16	*Longitude*	*-180 ~ 180*	*Float*	*Unit: ±ddd.ddddd*

5. When applying the GPS instruction, COM1 has to be in Master mode, i.e. M1312 must be enabled to send the request. In addition, M1314 = ON indicates receiving completed. M1315 = ON indicates an error. (D1250 = K1, receive time-out; D1250 = K2, checksum error)
6. Associated M flags and special D registers:

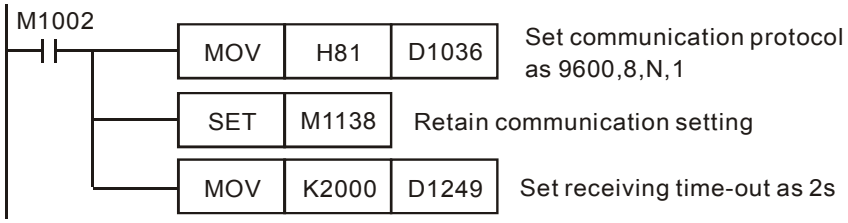
No.	Function
M1312	COM1 (RS-232) sending request
M1313	COM1 (RS-232) ready for data receiving
M1314	COM1 (RS-232) data receiving completed
M1315	COM1 (RS-232) data receiving error
M1138	Retaining communication setting of COM1
D1036	COM1 (RS-232) Communication protocol
D1249	COM1 (RS-232) data receiving time-out setting. (Suggested value: >1s)
D1250	COM1 (RS-232) communication error code

7. Before using the GPS data, check the value in **D**+9. If **D**+9 = 0, the GPS data is invalid.
8. If an error occurs, the previous data in **D** registers will not be cleared, i.e. the previous received data remains.

Program example: Sentence identifier: $GPGGA

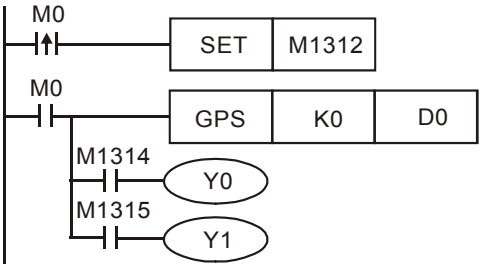
1. Set COM1communication protocol first

a)



2. Then enable M0 to execute the GPS instruction with sentence identifier $GPGGA

b)

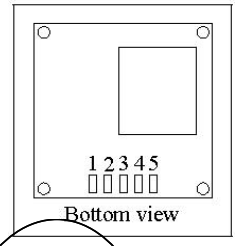


3. When the instruction finishes execution, M1314 = ON. If it fails, M1315 = ON. The data will be stored in addresses beginning with D0.

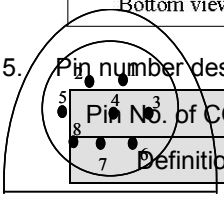
No.	*Content*	*No.*	*Content*
D0	*Hour*	*D8*	*East / West*
D1	*Minute*	*D9*	*GPS data valid / invalid*
D2	*Second*	*D10~D11*	*Altitude*
D3~D4	*Latitude*	*D12~D13*	*Latitude. Unit: ±dd.ddddd*
D5	*North / South*	*D14~D15*	*Longitude. Unit: ±ddd.ddddd*
D6~D7	*Longitude*		

4. Pin number description on the GPS module (LS20022)

Pin No. of GPS	1	2	3	4	5
Definition	VCC(+5V)	Rx	Tx	GND	GND



5. Pin number description on ELC COM1:



Pin No. of COM1	1	2	3	4	5	6	7	8
Definition	VCC(+5V)		--	Rx	Tx	--	--	GND

API	Mnemonic		Operands	Function
178	D	SPA	S, D	Solar Positioning Algorithm

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DSPA: 9 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Starting address for input parameters **D**: Starting address for output parameters

Description:

1. ELC2-PE doesn't support API178 SPA instruction.
2. Operand **S** occupies 208 consecutive word registers. The function of each word is shown below:

No.	*Content*	*Range*	*Format*	*Note*
S *+ 0*	*Year*	*2000 ~*	*Word*	
S *+ 1*	*Month*	*1 ~ 12*	*Word*	
S *+ 2*	*Day*	*1 ~ 31*	*Word*	
S *+ 3*	*Hour*	*0 ~ 23*	*Word*	
S *+ 4*	*Minute*	*0 ~ 59*	*Word*	
S *+ 5*	*Second*	*0 ~ 59*	*Word*	
S *+ 6~7*	*Time difference (Δt) (sec)*	*± 8000*	*Float*	
S *+ 8~9*	*Local time zone*	*± 12*	*Float*	*West=negative*
S *+ 10~11*	*Longitude*	*± 180*	*Float*	*West=negative*
S *+ 12~13*	*Latitude*	*± 90*	*Float*	*South=negative*
S *+ 14~15*	*Elevation*	*0~ 6500000*	*Float*	*Unit: meter*
S *+ 16~17*	*Pressure*	*0 ~ 5000*	*Float*	*Unit: millibar*
S *+ 18~19*	*Mean annual temperature (MAT)*	*-273~6000*	*Float*	*Unit: °C*
S *+ 20~21*	*Slope*	*± 360*	*Float*	
S *+ 22~23*	*Azimuth*	*± 360*	*Float*	
S *+ 24~25*	*Refraction of sunrise/sunset*	*± 5*	*Float*	
S *+26~207*	*Reserved for system operation*			

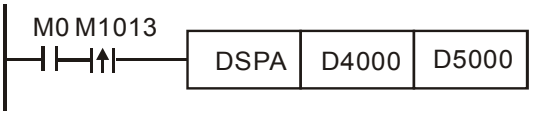
3. Operand **D** occupies 8 consecutive word registers. The function of each word is shown below:

No.	Content	Range	Format	Note
D *+ 0~1*	*Zenith*	*0 ~ 90*	*Float*	*Horizontal=0*
D *+ 2~3*	*Azimuth*	*0 ~ 360*	*Float*	*North point=0*
D *+ 4~5*	*Incidence*	*0 ~ 90*	*Float*	
D *+ 6*	*Converted DA value of Zenith*	*0 ~ 2000*	*Word*	*1LSB = 0.045 degree*
D *+ 7*	*Converted DA value of Azimuth*	*0 ~ 2000*	*Word*	*1LSB = 0.18 degree*

The execution time of the SPA instruction is up to 50ms.

Program example:

Input parameters starting from D4000: 2009/3/23/(y/m/d),10:10:30, Δt = 0, Local time zone = +8, Longitude/Latitude = +119.192345 East, +24.593456 North, Elevation = 132.2M, Pressure = 820m, MAT = 15.0℃, Slope = 0 degree, Azimuth = -10 degree.



Output results:

D5000: Zenith = F37.2394 degree

D5002: Azimuth = F124.7042 degree

API	Mnemonic			Operands	Function
179	D	WSUM	P	**S, D, n**	Sum of Multiple Devices

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*	*	*			WSUM, WSUMP: 7 steps
n					*	*							*			DWSUM, DWSUMP: 13 steps
D											*	*	*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

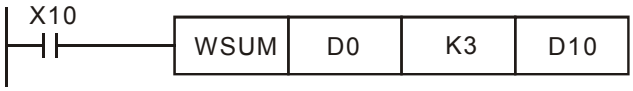
S: Source address **n**: Data length to be summed **D**: Address for storing the result

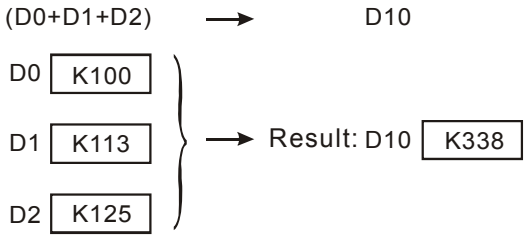
Description:

9. In ELC-PV, the WSUM is a new instruction in ELC-PV V1.4.
10. The WSUM instruction adds **n** values starting from **S** and stores the result in **D.**
11. Valid range for **n**: 1~64. If the **n** value is out of the available range (1~64), ELC will use the upper (64) or lower (1) bound.

Program example:

When X10 = ON, 3 consecutive addresses (**n** = 3) starting from D0 will be added and the result will be stored in D10





API	Mnemonic		Operands	Function
180	MAND	P	S_1, S_2, D, n	Matrix AND

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			MAND, MANDP: 9 steps
S_2							*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

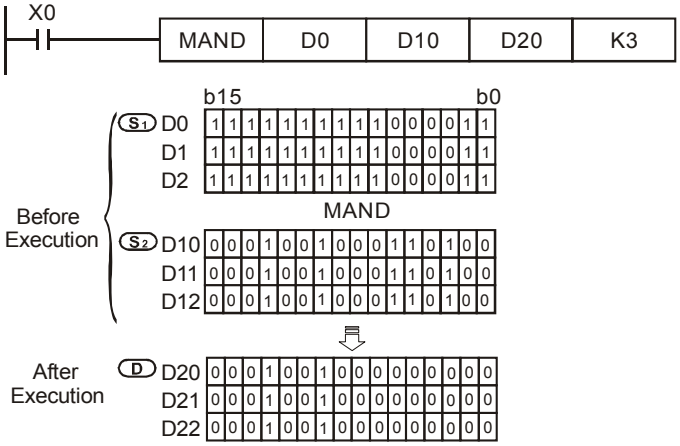
S_1: matrix source address 1 **S_2**: matrix source address 2 **D**: Area where calculated result is stored **n**: matrix length (**n**=K1~K256)

Description:

12. Execute the matrix AND operation to the matrix source addresses 1 and 2, with a length of **n** and save the result in **D**.
13. If **S_1**, and **S_2** designate KnX, KnY, KnM and KnS; **D** designates KnY, KnM and KnS. ELC-PA, ELCM-PH/PA, ELC2-PB/PH/PA/PE can designate n = 4. ELC-PV, ELC2-PV can designate n=1- 4.

Program Example:

When X0=ON, execute the MAND instruction and the AND operation to 3 rows (D0-D2) of 16-bit registers and 3 rows (D10-D12) of 16-bit registers. Then save the result in 3 rows (D20-D22) of 16-bit registers.



API	Mnemonic		Operands	Function
181	MOR	P	S_1, S_2, D, n	Matrix OR

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			MOR, MORP: 9 steps
S_2							*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: matrix source address 1 **S_2**: matrix source address 2. **D**: Area where calculated result is stored **n**: matrix length (**n**=K1~K256)

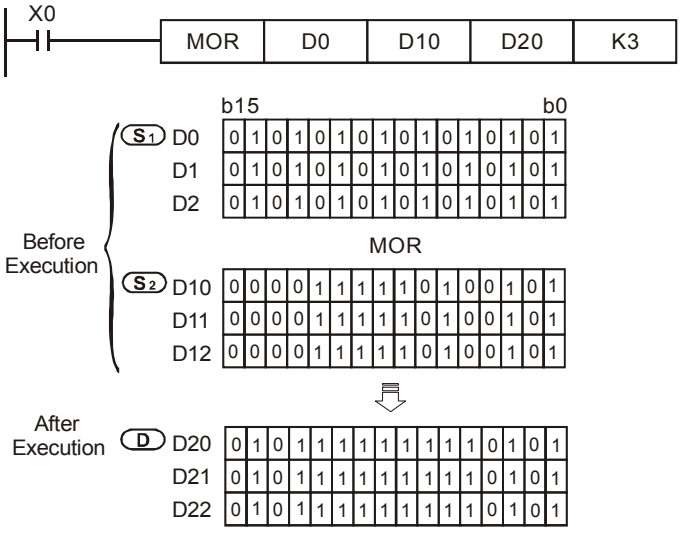
Description:

1 Execute the OR operation to the matrix source addresses 1 and 2, with a length of **n** and save the result in **D**.

2. If **S_1**, and **S_2** designate KnX, KnY, KnM and KnS; **D** designates KnY, KnM and KnS. ELC-PA, ELCM-PH/PA, ELC2-PB/PH/PA/PE can designate n = 4. ELC-PV, ELC2-PV can designate n=1- 4.

Program Example:

When X0=ON, execute the MOR instruction and the OR operation to 3 rows (D0-D2) of 16-bit register and 3 rows (D10-D12) of 16-bit register. Then save the result in 3 rows (D20-D22) of 16-bit register.



API	Mnemonic		Operands	Function
182	MXOR	P	S_1, S_2, D, n	Matrix XOR

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			MXOR, MXORP: 9 steps
S_2							*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: matrix source address 1 **S_2**: matrix source address 2 **D**: Area where calculated result is stored **n**: matrix length (**n**=K1~K256)

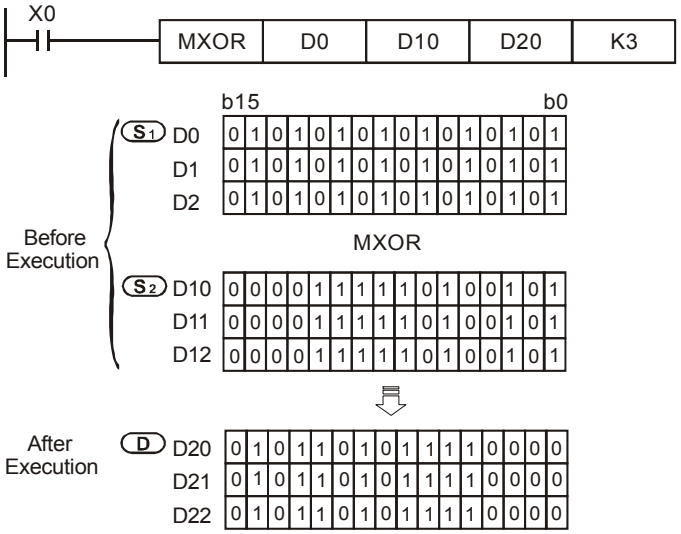
Description:

1. Execute the XOR operation to the matrix source addresses 1 and 2, with a length of **n** and save the result in **D**.

2. If **S_1**, and **S_2** designate KnX, KnY, KnM and KnS; **D** designates KnY, KnM and KnS. ELC-PA, ELCM-PH/PA, ELC2-PB/PH/PA/PE can designate n = 4. ELC-PV, ELC2-PV can designate n=1- 4.

Program Example:

When X0=ON, execute the MXOR instruction and the XOR operation to 3 rows (D0-D2) of 16-bit register and 3 rows (D10-D12) of 16-bit register. Then save the result in 3 rows (D20-D22) of 16-bit register.



API	Mnemonic		Operands	Function
183	MXNR	P	S_1, S_2, D, n	Matrix XNR

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			MXNR, MXNRP: 9 steps
S_2							*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: matrix source address 1 **S_2**: matrix source address 2 **D**: Area where calculated result is stored **n**: matrix length (K1~K256)

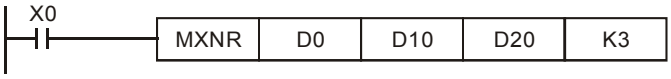
Description:

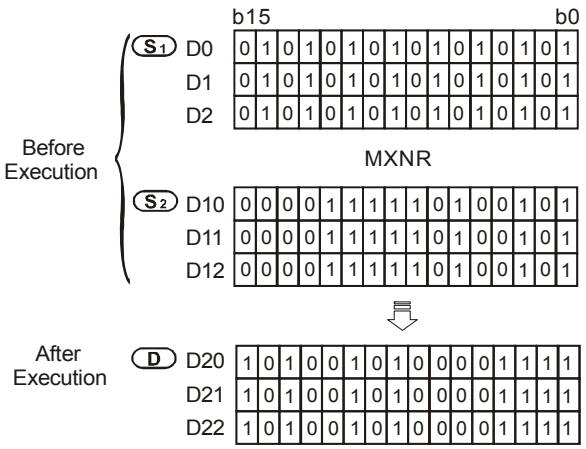
1. Execute the matrix XNR operation to the matrix source addresses 1 and 2, with a length of **n** and save the result in **D**.
2. The matrix XNR operation is: the result bit is 1 when 2 bits are the same otherwise it is 0.
3. If **S_1**, and **S_2** designate KnX, KnY, KnM and KnS; **D** designates KnY, KnM and KnS. ELC-PA, ELCM-PH/PA, ELC2-PB/PH/PA/PE can designate n = 4. ELC-PV, ELC2-PV can designate n=1- 4.

Program Example:

When X0=ON, Execute the matrix XNR operation to the matrix source addresses 1 and 2, with a length of **n** and save the result in **D**.

XNR operation to 3 rows (D0-D2) of 16-bit register and 3 rows (D10-D12) of 16-bit register. Then save the result in 3 rows (D20-D22) of 16-bit register.





API	Mnemonic		Operands	Function
184	MINV	P	S, D, n	Matrix Inverse

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MINV, MINVP: 7 steps
D								*	*	*	*	*	*			
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

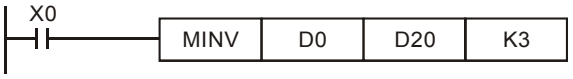
S: Matrix source address **D**: result **n** : matrix length (K1~K256)

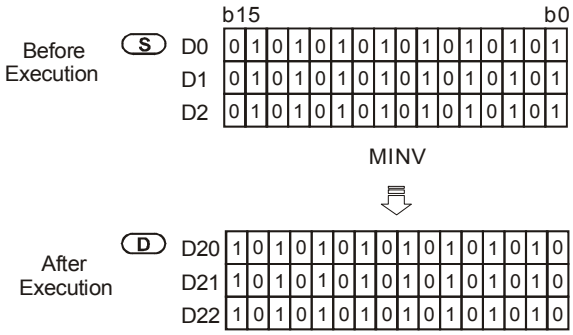
Description:

14. Do matrix inverse operation to the matrix source with a length of **n** and save the result in **D**.
15. If **S** designate KnX, KnY, KnM and KnS; **D** designates KnY, KnM and KnS. ELC-PA, ELCM-PH/PA, ELC2-PB/PH/PA/PE can designate n = 4. ELC-PV, ELC2-PV n=1- 4.
16. The inverse operation simply changes logical 1s to 0s and 0s to 1s.

Program Example:

When X0=ON, execute the MINV operation to 3 rows (D0-D2) of 16-bit registers by 3 rows (D10-D12) of 16-bit registers. Then save the result in 3 rows (D20-D22) of 16-bit registers.





API	Mnemonic		Operands	Function
185	MCMP	P	S_1, S_2, n, D	Matrix Compare

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			MCMP, MCMPP: 9 steps
S_2							*	*	*	*	*	*	*			
n					*	*							*			
D								*	*	*	*	*	*	*	*	

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: matrix source address 1 **S_2**: matrix source address 2 **n**: matrix length (K1~K256)

D: Address containing the pointer to the target address

Description:

17. For each comparison, it will compare each bit of **S_1** with each bit of **S_2**, one at a time and provide the pointer number for the bit position when two bits are equal in **D**. To find the address of the next equal comparison and save the address in **D**, a false-to-true transition of the instruction is needed.
18. Flag M1088. If M1088=1 an equal comparison was found and the pointer value is in **D**. Once a comparison is found, it will stop comparing immediately and set bit M1091=1. When the comparison reaches the last bit in the matrix, the matrix search end flag M1089 = ON and the comparison pointer value is saved in **D**. For next scan period, it will start comparing from the first bit and set the matrix search start flag M1090=1. If the **D** value exceeds its range, the pointer error flag M1092 =1.
19. The matrix pointer should be a 16-bit D-register.
20. If **S_1**, and **S_2** designate KnX, KnY, KnM and KnS; **D** designates KnY, KnM and KnS. ELC-PA, ELCM-PH/PA, ELC2-PB/PH/PA/PE can designate n = 4. ELC-PV, ELC2-PV can designate n=1- 4.
21. Flag: M1088-M1092

Program Example:

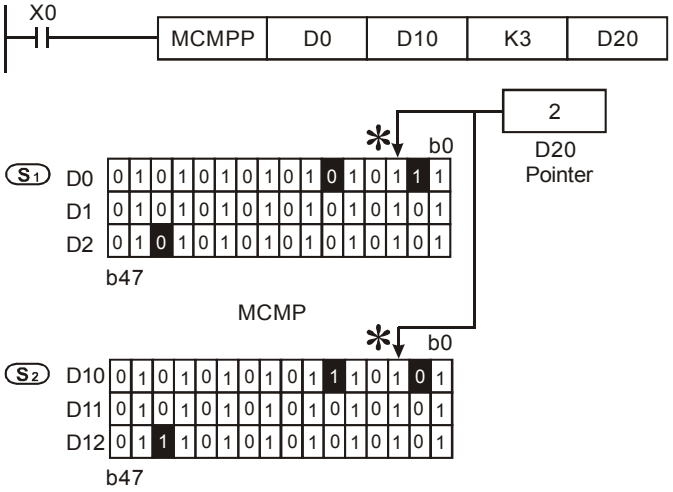
When X0 transitions from OFF→ON, the matrix search start flag M1090=0 and it will start comparing to find the first occurrence of two equal bits As long as M1088=0, there is no match.

When the present value of the pointer D20=2, and the instruction is executed 4 times, the following pointer values will be found. Each execution requires a false-to-true transition of the MCMPP instruction.

22. D20=5, matrix bit search flag M1091=1, matrix search end flag M1089=0.
23. D20=45, matrix bit search flag M1091=1, matrix search end flag M1089=0.

24. D20=47, matrix bit search flag M1091=0, matrix search end flag M1089=1.

25. D20=1, matrix bit search flag M1091=1, matrix search end flag M1089=0.



Description of the flags:

M1088: Matrix comparison flag, if M1088=1, the result of the comparison is equal, otherwise M1088=0.

M1089: Matrix search end flag, when the last bit in the matrix is encountered, M1089=1.

M1090: Matrix search start flag, start comparing from the first bit, M1090=1.

M1091: Matrix bit search flag, it will stop comparing once an equal comparison is found and M1091=1.

M1092: Matrix pointer error flag, pointer exceeds its range, M1092=1.

API	Mnemonic		Operands	Function
186	MBRD	P	**S, n, D**	Matrix Bit Read

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MBRD, MBRDP: 7 steps
n					*	*							*			
D								*	*	*	*	*	*	*	*	

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: matrix source address **n**: matrix length (K1~K256). **D**: pointer to where the target address is saved

Description:

26. When executing the instruction, it will look at M1094 (matrix pointer clear flag) to see if it is ON. If it is ON, pointer **D** will be cleared to 0 and it will read **S** starting from bit 0 and read the ON/OFF state of each bit and write it to M1095 (matrix rotate/shift/output/carry). It will see if M1093 (matrix pointer increase flag) is ON after reading each bit and increase **D** by 1 if it is ON. When reading to the last bit, M1089 (matrix search end flag) =ON, pointer **D** records the bit number of the read bit and ends execution of the instruction.
27. The Pointer should be a 16-bit D-register.
28. If **S** designate KnX, KnY, KnM and KnS; **D** designates KnY, KnM and KnS. ELC-PA, ELCM-PH/PA, ELC2-PB/PH/PA/PE can designate n = 4. ELC-PV, ELC2-PV n=1- 4.

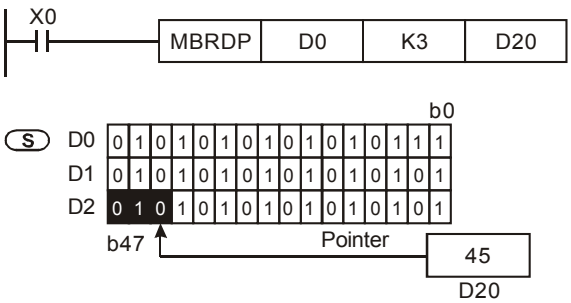
Program Example:

29. When X0 transitions from OFF→ON, the pointer clear flag M1094=ON, the matrix pointer increases by 1 if flag M1093=1, after the first bit is read.
30. When present value of pointer D20=45, it will read each of the last 3 bits with each false-to-true transition of the instruction, while M1093 = 1. The following three results will be obtained.

D20=46, matrix rotate/shift/output carry flag M1095=0, matrix search end flag M1089=0.

a) D20=47, matrix rotate/shift/output carry flag M1095=1, matrix search end flag M1089=0.

b) D20=47, matrix rotate/shift/output carry flag M1095=1, matrix search end flag M1089=1.



Description of the flags:

M1089: Matrix search end flag, when reaching the last bit, M1089=1.

M1092: Matrix pointer error flag, the pointer exceeds its range, M1092=1.

M1093: Matrix pointer increase flag, when set, add 1 to the present pointer.

M1094: Matrix pointer clear flag, when set, clear the present pointer to 0.

M1095: Matrix rotate/shift/output carry flag.

API	Mnemonic		Operands	Function
187	MBWR	P	**S, n, D**	Matrix Bit Write

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MBWR, MBWRP: 7 steps
n					*	*							*			
D								*	*	*	*	*	*	*	*	

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

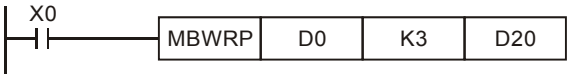
S: matrix source address **n**: matrix length (K1~K256) **D**: pointer to where the target address is saved

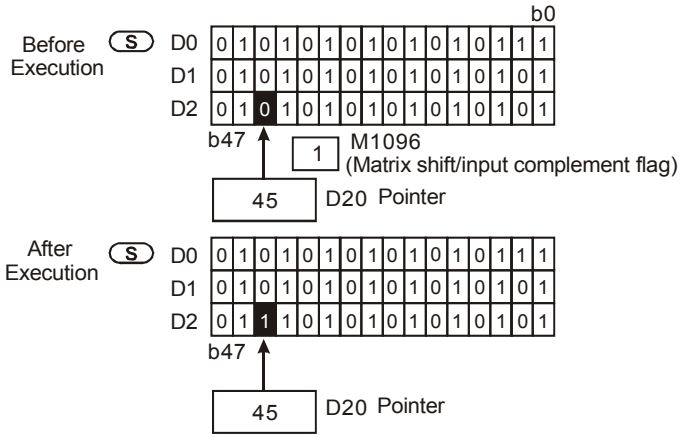
Description:

31. When executing the instruction, it will look at M1094 (matrix pointer clear flag) to see if it is ON. If it is ON, pointer **D** will be cleared to 0 and write the state of M1096 (matrix shift/input complement flag) into bit 0 of **S**. It will see if M1093 (matrix pointer increase flag) is ON after writing each bit and increase **D** by 1 if it is ON. When writing to the last bit, M1089 (matrix search end flag) =ON, and pointer **D** will record the number of the bit written to and end executing the instruction. If **D** exceeds its range, M1092=1.
32. The Pointer should be a 16-bit D-register.
33. If **S** designate KnX, KnY, KnM and KnS; **D** designates KnY, KnM and KnS. ELC-PA, ELCM-PH/PA, ELC2-PB/PH/PA/PE can designate n = 4. ELC-PV, ELC2-PV n=1- 4.
34. Flags: Refer to explanation for M1089-M1096 below.

Program Example:

35. When X0 transitions from OFF→ON, the pointer clear flag M1094=OFF and if the matrix pointer flag M1093=1, the pointer (D20) is increased by 1.
36. When the present pointer is D20=45, M1096 (matrix shift/input complement flag) =1. When X0 is executed once from OFF→ON, it will obtain following result:





Description of the flags:

M1089: Matrix search end flag, when reaching the last bit, M1089=1.

M1092: Matrix pointer error flag, the pointer exceeds its range, M1092=1.

M1093: Matrix pointer increase flag, when set, add 1 to the present pointer.

M1094: Matrix pointer clear flag, when set, clear the present pointer to 0.

M1096: Matrix shift/input complement flag

API	Mnemonic		Operands	Function
188	MBS	P	**S, D, n**	Matrix Bit Shift

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MBS, MBSP: 7 steps
D								*	*	*	*	*	*			
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

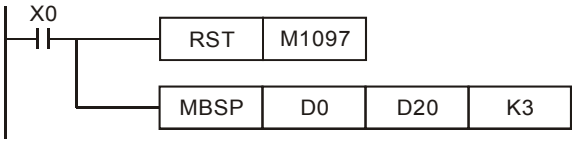
S: matrix source address **D**: result **n**: matrix length (K1~K256)

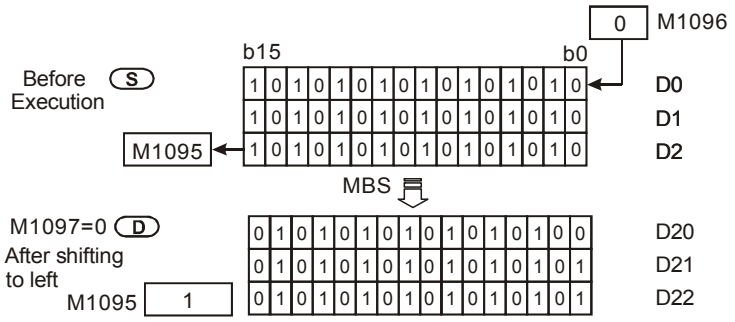
Description:

37. This instruction is used to shift bits left or right within a matrix **S**. When M1097=0 bits are shifted to the left in the matrix. When M1097=1 bits are shifted to the right in the matrix.
38. M1096 is used to fill the empty bit when a shift occurs. M1095 is the carry flag and is where the bit shifted out of the matrix goes**.**
39. This instruction works best when used as a pulse instruction (MBSP).
40. If **S** designate KnX, KnY, KnM and KnS; **D** designates KnY, KnM and KnS. ELC-PA, ELCM-PH/PA, ELC2-PB/PH/PA/PE can designate n = 4. ELC-PV, ELC2-PV n=1- 4.
41. Flags:
    M1095: matrix rotate/shift/output carry flag
    M1096: matrix shift/input complement flag
    M1097: matrix rotate/shift direction flag

Program Example 1:

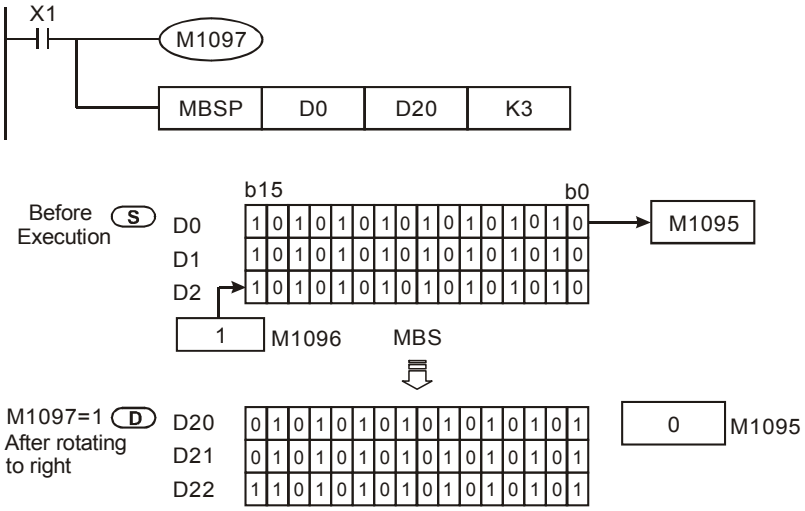
When X0=ON, M1097=OFF means shift the matrix to the left by one bit for each false-to-true transition of the instruction. For this example, flag M1096 was 0. This bit is moved into the first position of the matrix as shown below when a shift occurs. The bit shifted out of the matrix goes to the Carry Bit M1095. The resulting matrix after the shift is the matrix represented by D20-D22. The source matrix represented by D0-D2 is unaffected.





Program Example 2:

When X1=ON, M1097=ON means shift the matrix to the right by one bit for each false-to-true transition of the instruction. For this example, flag M1096 was 1. This bit is moved into the first position of the matrix as shown below when a shift occurs. The bit shifted out of the matrix goes to the Carry Bit M1095. The resulting matrix after the shift is the matrix represented by D20-D22. The source matrix represented by D0-D2 is unaffected.



API	Mnemonic		Operands	Function
189	MBR	P	**S, D, n**	Matrix Bit Rotator

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MBR, MBRP: 7 steps
D								*	*	*	*	*	*			
n					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: matrix source address **D**: result **n**: matrix length (K1~K256)

Description:

1 This instruction is used to shift bits left or right within a matrix **S**. When M1097=0 bits are shifted to the left in the matrix. When M1097=1 bits are shifted to the right in the matrix.

2 For this instruction, the empty bit is filled by the bit shifted out of the other end of the matrix. when a shift occurs. M1095 is the carry flag and is also where the bit shifted out of the matrix goes**.**

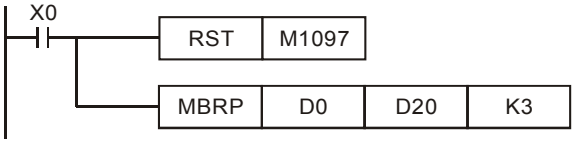
3 This instruction works best when used as a pulse instruction (MBSP).

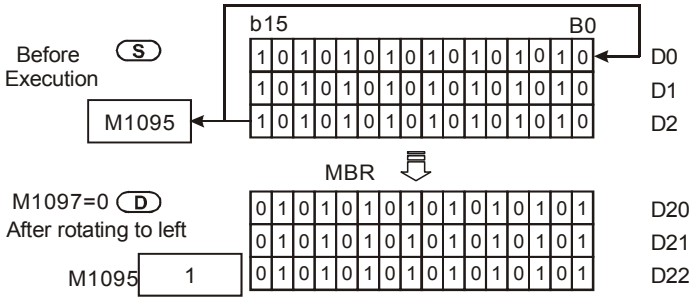
4 If **S** uses KnX, KnY, KnM and KnS or if **D** uses KnY, KnM and KnS ELC-PA, ELCM-PH/PA, ELC2-PB/PH/PA/PE can designate n = 4. ELC-PV, ELC2-PV n ≦4..

5. Flags:
   M1095: matrix rotate/shift/output carry flag
   M1097: matrix rotate/shift direction flag

Program Example 1:

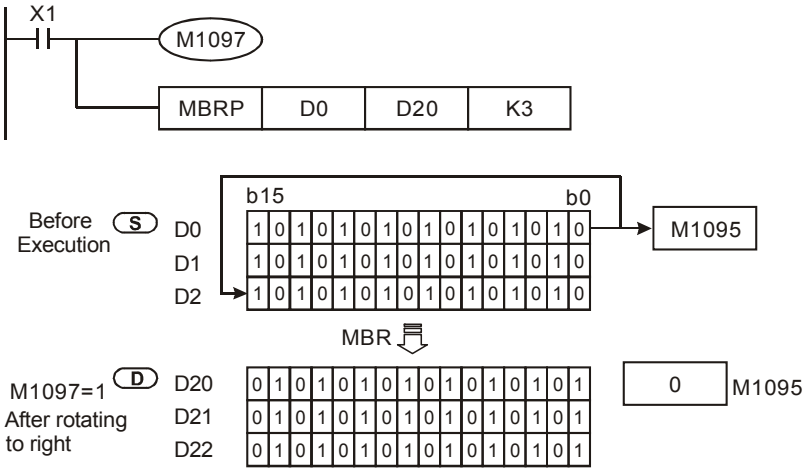
When X0=ON, M1097=OFF means shift the matrix to the left by one bit for each false-to-true transition of the instruction. The bit moved into the bit vacated by the shift is the bit shifted out of the other end of the matrix. This bit shifted out of the matrix also goes to the Carry Bit M1095. The resulting matrix after the shift is the matrix represented by D20-D22. The source matrix represented by D0-D2 is unaffected.





Program Example 2:

When X1=ON, M1097=ON means shift the matrix to the right by one bit for each false-to-true transition of the instruction. The bit moved into the bit vacated by the shift is the bit shifted out of the other end of the matrix. The bit shifted out of the matrix also goes to the Carry Bit M1095. The resulting matrix after the shift is the matrix represented by D20-D22. The source matrix represented by D0-D2 is unaffected.



API	Mnemonic		Operands	Function
190	MBC	P	**S, n, D**	Matrix Bit State Count

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MBC, MBCP: 7 steps
n					*	*							*			
D								*	*	*	*	*	*	*	*	

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Matrix source address **n**: matrix length (K1~K256) **D**: result

Description:

42. This instruction counts the number of 1s or 0s in a matrix of length **n**, and places the result in **D**. M1098 determines if the instruction counts 1s or 0s. If M1098=0, it counts the number of 0s in the matrix. If M1098=1, it counts the number of 1s in the matrix.
43. If **S** uses KnX, KnY, KnM and KnS or if **D** uses KnY, KnM and KnS ELC-PA, ELCM-PH/PA, ELC2-PB/PH/PA/PE can designate n = 4. ELC-PV, ELC2-PV n ≦4.

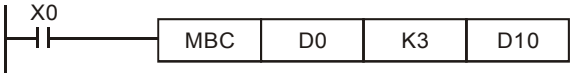
3. Flags:

   When M1099 is ON, this means the result is 0.

   When M1098=1, count the number of 1s in the matrix. When M1098=0, count the number of 0s in the matrix.

Program Example:

When X0=ON and M1098=1, the instruction counts the number of 1s in the matrix D0-D2 and saves the count in D10. When M1098=0, it counts the number of 0s in the matrix D0~D2 and saves the count in D10.



D0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
D1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
D2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1

D10 | 12 | M1098=0

D10 | 36 | M1098=1

API	Mnemonic		Operands	Function
191	D	PPMR	S_1, S_2, S, D	2-Axis Relative Point to Point Motion

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DPPMR: 17 steps
S_2					*	*							*			
S					*	*							*			
D		*														

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Number of output pulses for the X axis **S_2**: Number of output pulses for the Y axis **S**: Max. point to point output frequency **D**: Pulse output address

Description: (ELC-PV, ELC2-PV)

44. This instructin only supports “pulse + direction” mode.
45. **S_1** and **S_2** are the designated number of output pulses in the X axis (Y0 or Y4) and Y axis (Y2 or Y6). The range of these values is -2,147,483,648 ~ +2,147,483,647 (+/- represents the forward/reverse directions). When in the forward direction, the present value registers are: CH0 (D1337 high word, D1336 low word), CH1 (D1339 high word, D1338 low word), CH2 (D1376 high word, D1375 low word) and CH3 (D1378 high word, D1377 low word) and they will increase. When in the reverse direction, the present value will decrease.
46. **D** can use Y0 and Y4.

When Y0 is used:

Y0 refers to the 1st group of X-axis pulse outputs.

Y1 refers to 1st group of the X-axis direction signal.

Y2 refers to 1st group of Y-axis pulse outputs.

Y3 refers to 1st group of the Y-axis direction signal.

Y4 refers to 2nd group of X-axis pulse outputs.

Y5 refers to 2nd group of the X-axis direction signal.

Y6 refers to 2nd group of Y-axis pulse outputs.

Y7 refers to 2nd group of the Y axis direction signal.

When the instruction execution is complete, the direction signal output will not turn off until the conditions driving the instruction go false.

D1340/D1379 are the start/end frequencies of the 2-axis of motion. D1343/D1381 are the times for the first acceleration segment and last deceleration segment of the 2-axis of motion. The time shall be longer than 10ms. If the time is shorter than 10ms or longer than 10,000ms, the output will be

operated at 10ms. Default setting = 100ms.

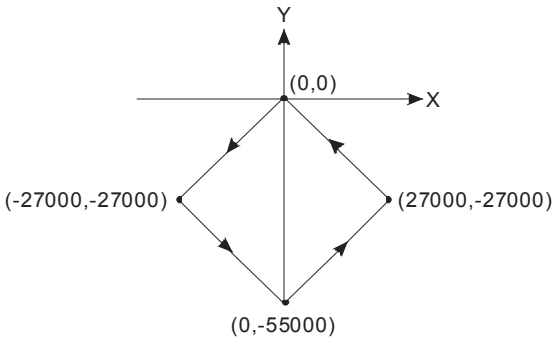
47. If the maximum output frequency setting is less than 10Hz, the output will be operated at 10Hz. If the setting is more than 200KHz, the output will be operated at 200KHz.
48. When the 2-axis synchronous motion instruction is enabled, the start frequency and acceleration/deceleration time for the Y axis will be the same as the settings for the X axis.
49. The number of output pulses for the 2-axis motion shall not be less than 59; otherwise the line drawn will not be straight.
50. There is no limit on the number of times this instruction can be used in the program. However, if CH1 or CH2 is in use, the 1st group will not be able to send pulses. If CH3 or CH4 is in use, the 2nd group will not be able to send pulses.
51. Flags: M1029, M1030, M1334, M1335. See Points to note below for more details.

Description: (ELCM-PH/PA, ELC2-PC/PA/PE)

52. If the maximum output frequency setting is less than 100Hz, the output will be operated at 100Hz. If the setting is more than 100KHz, the output will be operated at 100KHz.
53. When the 2-axis synchronous motion instruction is enabled, the start frequency and acceleration/deceleration time in the Y axis will be same as the settings in the X axis.
54. The number of output pulses for the 2-axis motion shall not be less than 20; otherwise the line drawn will not be straight.
55. There is no limitation on the number of times using the instruction.
56. Flags: M1029. See Points to note below for more details.

Program Example:

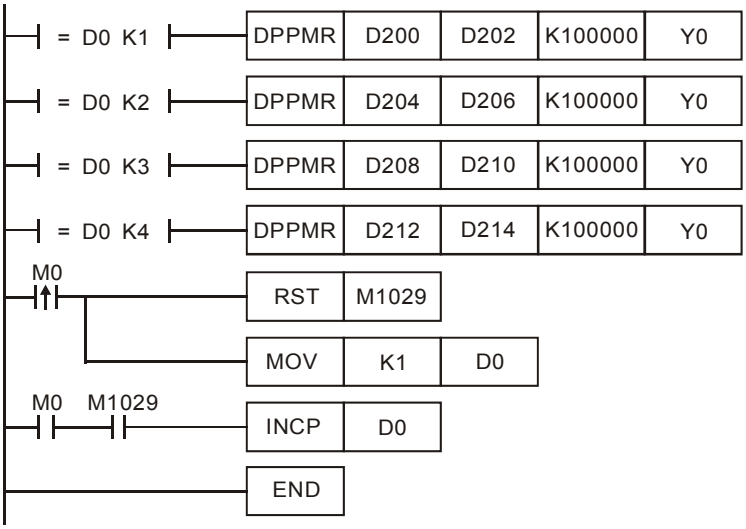
57. Draw a rhombus like the figure below.



58. Steps:

a) Set the four coordinates (0,0), (-27000, -27000), (0, -55000), (27000, -27000) per the figure above. Calculate the relative coordinates of the four points and obtain (-27000, -27000), (27000, -28000), (27000, 27000), and (-27000, 27000). Place them in the 32-bit registers (D200, D202), (D204, D206), (D208, D210), (D212, D214).

b) Set M0 On to start the 2-axis motion.



59. Motion description:

When M0 = On, the ELC will start the first point-to-point motion at 100KHz. D0 will increment by 1 whenever a motion step is complete, which will start the next motion step. This will continue until the fourth motion step is complete.

Points to note:

1. ELC-PV, ELC2-PV Flag descriptions:

M1029 : On when the 1st group of 2-axis motion is complete.

M1036 : On when the 2nd group of 2-axis motion is complete.

M1334 : On when the 1st group of 2-axis motion is in error.

M1336 : 1st group of 2-axis pulse output indication flag

M1520 : On when the 2nd group of 2-axis motion is in error.

M1522 : 2nd group of 2-axis pulse output indication flag

2. ELC-PV, ELC2-PV Special register descriptions:

D1336, D1337 : Pulse present value register for the Y0 output of the 1st group of X-axis motion. The present value increases or decreases based on the rotation direction. (D1337 high word; D1336 low word)

D1338, D1339 : Pulse present value register for the Y2 output of the 1st group of Y-axis motion. The present value increases or decreases based on the rotation direction. (D1339 high word; D1338 low word)

D1340 : Frequency settings for the first acceleration and the last deceleration segment for the Y0 output of the 1st group of X-axis motion and Y2 of the Y-axis motion for API 191 DPPMR and API 192 DPPMA.

D1343 : Time settings for the first acceleration and last deceleration segment for the Y0 output of the 1st group of X-axis motion and Y2 of the Y-axis motion for API 191 DPPMR and API 192 DPPMA.

D1375, D1376 : Pulse present value register for the Y4 output of the 2nd group of X-axis motion. The present value increases or decreases based on the rotation direction. (D1337 high word; D1336 low word)

D1377, D1378 : Pulse present value register for the Y6 output of the 2nd group of Y-axis motion. The present value increases or decreases based on the rotation direction. (D1339 high word; D1338 low word)

D1379 : Frequency settings for the first acceleration and last deceleration segment for the Y4 output of the 2nd group of X-axis motion and Y6 of the Y-axis motion for API 191 DPPMR and API 192 DPPMA.

D1381 : Time settings for the first acceleration and last deceleration segment for the Y4 output of the 2nd group of X-axis motion and Y6 of the Y-axis motion for API 191 DPPMR and API 192 DPPMA.

3. ELCM-PH/PA, ELC2-PC/PA/PE Flag explanations:

M1029 : On when the 2-axis pulse output is complete.

4. ELCM-PH/PA, ELC2-PC/PA/PE Special register explanations:

D1030, D1031 : Pulse present value register for the Y0 output for the X-axis motion. The present value increases or decreases based on the rotation direction. (D1031 high word; D1030 low word)

D1336, D1337 : Pulse present value register for the Y2 output for the Y-axis motion. The present value increases or decreases based on the rotation direction. (D1337 high word; D1336 low word)

D1340 : Frequency settings for the first acceleration and last deceleration segment for the Y0 output of the X-axis motion and Y2 of the Y-axis motion for API 191 DPPMR and API 192 DPPMA.

D1343 : Time settings for the first acceleration and last deceleration segment for the Y0 output of the X-axis motion and Y2 of the Y-axis motion for API 191 DPPMR and API 192 DPPMA.

API	Mnemonic		Operands	Function
192	D	PPMA	**S_1, S_2, S, D**	2-Axis Absolute Point to Point Motion

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DPPMA: 17 steps
S_2					*	*							*			
S					*	*							*			
D		*														

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Number of X axis output pulses **S_2**: Number of Y axis output pulses **S**: Maximum output frequency **D**: Pulse output address

Description: (ELC-PV, ELC2-PV)

1. Flags: M1029, M1030, M1334, M1335. Refer to API 191 DPPMR for details.
2. This instructin only supports “pulse + direction” mode.
3. **S_1** and **S_2** are the designated number of output pulses in the X axis (Y0 or Y4) and Y axis (Y2 or Y6). The range of the values are -2,147,483,648 ~ +2,147,483,647 (+/- represents the forward/reverse direction). When in the forward direction, the pulse present value registers CH0 (D1337 high word, D1336 low word), CH1 (D1339 high word, D1338 low word), CH2 (D1376 high word, D1375 low word) and CH3 (D1378 high word, D1377 low word) will increase. When in the reverse direction, the present value will decrease.
4. **D** can use Y0 and Y4.

When Y0 is used:

Y0 refers to the 1st group of X-axis pulse outputs.

Y1 refers to the 1st group of the X-axis direction signal.

Y2 refers to the 1st group of Y-axis pulse outputs.

Y3 refers to the 1st group of the Y axis direction signal.

Y4 refers to the 2nd group of X-axis pulse outputs.

Y5 refers to the 2nd group of the X-axis direction signal.

Y6 refers to the 2nd group of Y-axis pulse outputs.

Y7 refers to the 2nd group of the Y axis direction signal.

When the instruction execution is complete, the direction signal output will not turn off until the conditions driving the instruction go false.

5. D1340/D1379 refer to the settings of the start/end frequencies of the 2-axis motion.
   D1343/D1381 refer to the time of the first acceleration segment and last deceleration segment

of the 2-axis motion. The time must be longer than 10ms. If the time is shorter than 10ms or longer than 10,000ms, the output will be operated at 10ms. Default setting = 100ms.

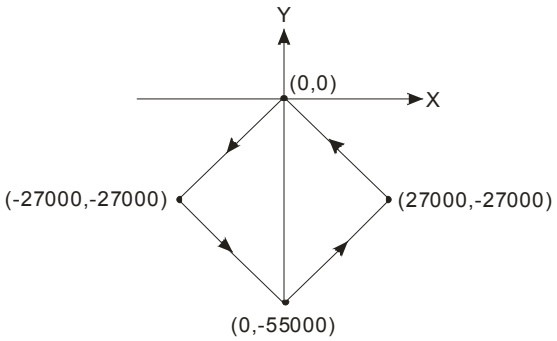
6. If the maximum output frequency setting is less than 10Hz, the output will be operated at 10Hz. If the setting is more than 200KHz, the output will be operated at 200KHz.
7. When the 2-axis synchronous motion instruction is enabled, the start frequency and the acceleration/deceleration time in the Y axis will be the same as the settings for the X axis.
8. The number of output pulses for the 2-axis motion must be greater than 59, otherwise the line drawn will not be straight.
9. There is no limit on the number of times this instruction can be used in the program.

Description: (ELCM-PH/PA, ELC2-PC/PA/PE)

1. Flags: M1029. See API 191 DPPMR for more details.
2. S_1 and S_2 are the designated number of output pulses in the X axis (Y0) and Y axis (Y2). The range of the values is -2,147,483,648 ~ +2,147,483,647 (+/- represents the forward/backward direction). When in forward direction, the pulse present value registers CH0 (D1031 high word, D1030 low word), CH1 (D1337 high word, D1336 low word) will increase. When in the reverse direction, the present value will decrease.

Program Example:

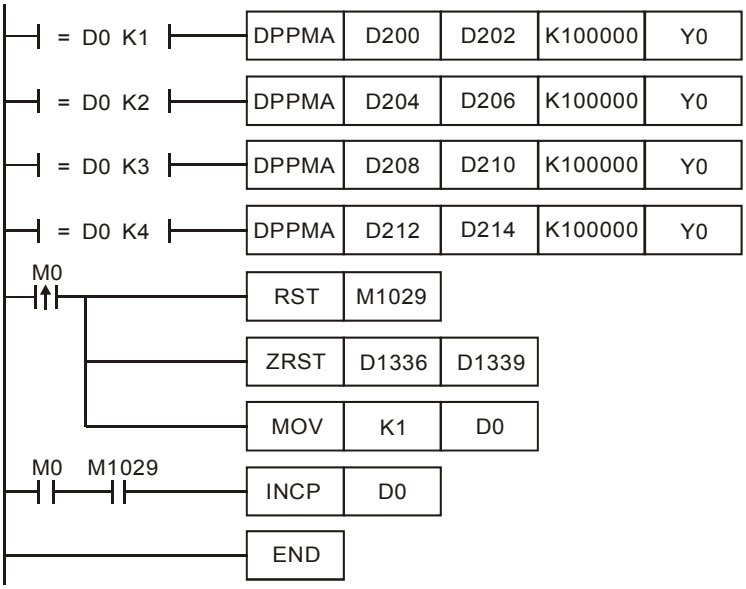
1. Draw a rhombus per the figure below.



2. Steps:

a) Set the four coordinates (0,0), (-27000, -27000), (0, -55000), (27000, -27000) per the figure above. Place them in the 32-bit registers (D200, D202), (D204, D206), (D208, D210), (D212, D214).

b) Set M0 On and start the 2-axis motion.



3. Motion description:

When M0 = On, the ELC will start the first point-to-point motion at 100KHz. D0 will increment by 1 whenever a motion step is complete, which will start the next motion step. This will continue until the fourth motion step is completed.

API	Mnemonic		Operands	Function
193	D	CIMR	S_1, S_2, S, D	2-Axis Relative Position Arc Interpolation

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DCIMR: 17 steps
S_2					*	*							*			
S													*			
D		*														

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S1: Number of X axis output pulses **S2:** Number of Y axis output pulses **S:** Parameter settings **D:** Pulse output address

Description: (ELC-PV, ELC2-PV)

1. Flags: M1029, M1030, M1334, M1335. See API 191 DPPMR for details.
2. This instructin only supports “pulse + direction” mode.
3. S_1 and S_2 are the designated number of output pulses in the X axis (Y0 or Y4) and Y axis (Y2 or Y6). The range of the values are -2,147,483,648 ~ +2,147,483,647 (+/- represents the forward/backward direction). When in the forward direction, the pulse present value registers CH0 (D1337 high word, D1336 low word), CH1 (D1339 high word, D1338 low word), CH2 (D1376 high word, D1375 low word) and CH3 (D1378 high word, D1377 low word) will increase. When in the reverse direction, the present value will decrease.
4. When the lower 16 bits of **S** are K0, this refers to a clockwise 10-segment (average resolution) output; K2 refers to a clockwise 20-segment (higher resolution) output for a 90° arc (see figure 1 and 2). K1 refers to a counterclockwise 10-segment (average resolution) output; K3 refers to a counterclockwise 20-segment (higher resolution) output for a 90° arc (see figure 3 and 4).
5. The higher 16 bits of **S** (time settings for the motion): K1 refers to 0.1 seconds. The setting range for average resolution is K1 ~ K100 (0.1 sec. ~ 10 secs.), for higher resolution use K2 ~ K200 (0.2 sec. ~ 20 secs.).

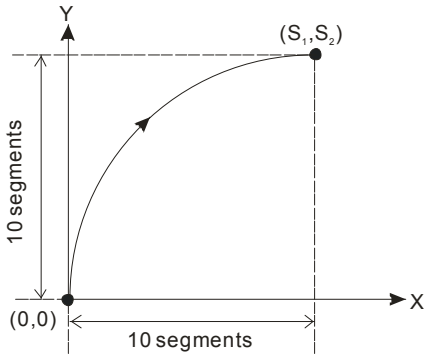


Figure 1

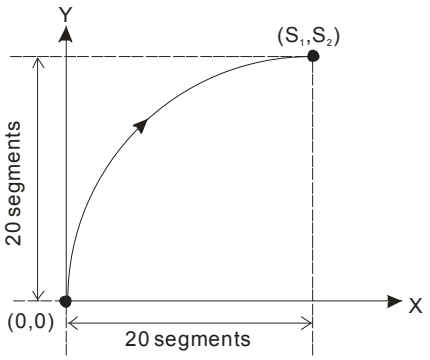


Figure 2

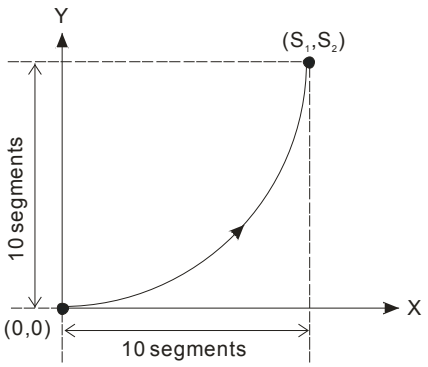


Figure 3

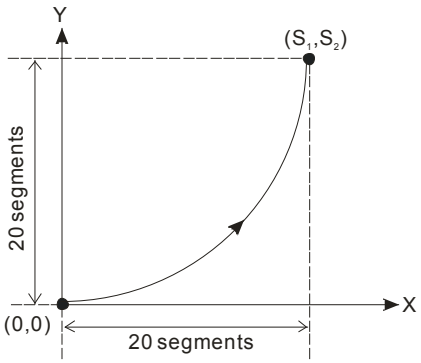


Figure 4

6. **D** can use Y0 and Y4.

When Y0 is used:

Y0 refers to the 1st group of X-axis pulse outputs.

Y1 refers to the 1st group of the X-axis direction signal.

Y2 refers to the 1st group of Y-axis pulse outputs.

Y3 refers to the 1st group of the Y-axis direction signal.

When Y4 is used:

Y4 refers to the 2nd group of X-axis pulse outputs.

Y5 refers to the 2nd group of the X-axis direction signal.

Y6 refers to the 2nd group of Y-axis pulse outputs.

Y7 refers to the 2nd group of the Y-axis direction signal.

When the instruction execution is complete, the direction signal output will not turn off until the conditions driving the instruction go false.

7. Draw four 90° arcs.
8. When the direction signal is On, the direction is positive. When the direction signal is Off, the direction is negative. When S is set to K0, K2, the arcs will be clockwise (see figure 5). When S is set as K1, K3, the arcs will be counterclockwise (see figure 6).

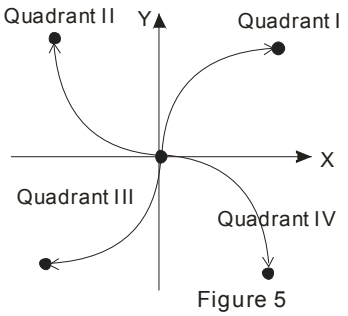


Figure 5

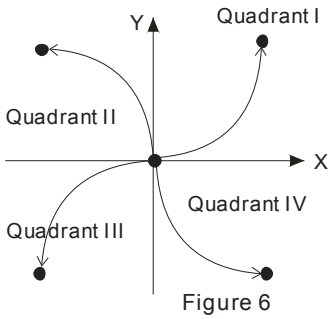


Figure 6

9. When the 2-axis motion is being executed in 10 segments (of average resolution), the operation time of the instruction when the instruction is first enabled is approximately 5ms. The number of output pulses cannot be less than 100 and more than 1,000,000; otherwise, the instruction

cannot be enabled.

10. When the 2-axis motion is being executed in 20 segments (high resolution), the operation time of the instruction when the instruction is first enabled is approximately 10ms. The number of output pulses cannot be less than 1,000 and more than 10,000,000; otherwise, the instruction cannot be enabled.
11. If the number of pulses in 10-segment or 20-segment motion needs to be lower or higher than the allowable range, adjust the gear ratio of the servo to obtain the desired value.
12. Each time the instruction is executed, only one 90° arc can be drawn.
13. There are no settings for the start frequency and acceleration/deceleration time.
14. There is no limit on the number of times this instruction can be used in the program..
15. The settings for the direction and resolution are in the lower 16 bits of S and can only be K0 ~ K3.
16. The settings for the motion time are in the high 16 bits of S and can be slower than the the fastest suggested time but must not be faster than the fastest suggested time.
17. The fastest suggested time for the arc interpolation:

Segments	Max. target position (pulse)	Fastest suggested set time (unit:100ms)
Average resolution	100 ~ 10,000	1
	10,001 ~ 19,999	2
	:	:
	Less than 1,000,000	Less than 100
Higher resolution	1,000 ~ 20,000	2
	20,000 ~ 29,999	3
	:	:
	Less than 10,000,000	Less than 200

Description: (ELCM-PH/PA, ELC2-PC/PA/PE)

1. Flags: M1029. See API 191 DPPMR for details.
2. $\mathbf{S_1}$ and $\mathbf{S_2}$ are the designated number of output pulses in the X axis (Y0) and Y axis (Y2). The range of the values is -2,147,483,648 ~ +2,147,483,647 (+/- represents the forward/backward direction). When in the forward direction, the pulse present value registers CH0 (D1031 high word, D1030 low word), CH1 (D1337 high word, D1336 low word) will increase. When in reverse direction, the present value will decrease.
3. The lower 16 bits of **S** (the settings for the direction and resolution): K0 refers to clockwise 20-segment output for a 90° arc (see figure 7). K1 refers to counterclockwise 20-segment output for a 90° arc (see figure 8).
4. The higher 16 bits of **S** (the settings for the motion time): K1 refers to 0.1 seconds. The range for the resolution is K2 ~ K200 (0.2 sec. ~ 20 secs.).

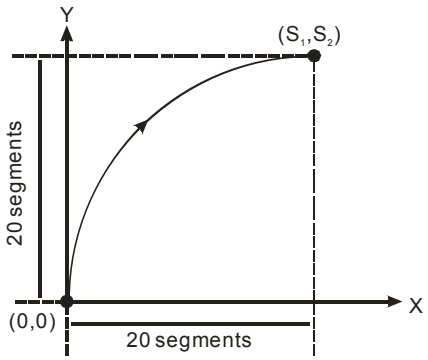


Figure 7

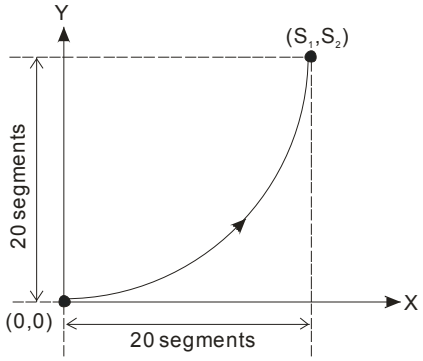


Figure 8

5. **D** can use Y0.

When Y0 is used:

Y0 refers to the X-axis pulse outputs.

Y1 refers to the X-axis direction signal.

Y2 refers to the Y-axis pulse outputs.

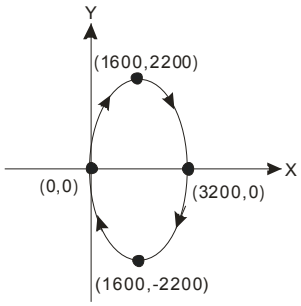
Y3 refers to the Y-axis direction signal.

6. Draw four 90° arcs. When the direction signal is Off, the direction is positive. When the direction signal is On, the direction is negative. When S is set as K0, the arcs will be clockwise (see figure 5). When S is set as K1, the arcs will be counterclockwise (see figure 6).
7. When the 2-axis motion is being executed, the operation time of the instruction when the instruction is first enabled is approximately 2ms.
8. Each time the instruction is executed, only one 90° arc can be drawn.
9. There are no settings for the start frequency and acceleration/deceleration time.
10. There is no limit on the number of times this instruction can be used in the program.
11. The settings for the direction and resolution are in the lower 16 bits of **S** and can only be K0 ~ K1.
12. The settings for the motion time are in the high 16 bits of S and can be slower than the fastest suggested time but shall not be faster than the fastest suggested time.
13. The fastest suggested time for the arc interpolation:

Segments	Max. target position (pulse)	Fastest suggested set time (unit:100ms)
20-segment resolution	1,000 ~ 20,000	2
	20,000 ~ 29,999	3
	:	:
	Less than 10,000,000	Less than 200

Program Example 1:

1. Draw an ellipse per the figure below.

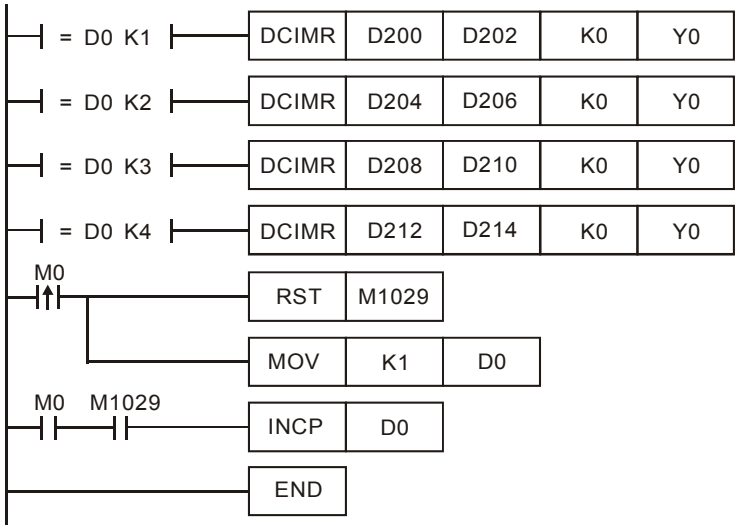


2. Steps:

   a) Set the four coordinates (0,0), (1600, 2200), (3200, 0), (1600, -2200) per the figure above. Calculate the relative coordinates of the four points and obtain (1600, 2200), (1600, -2200), (-1600, -2200), and (-1600, 2200). Place them in the 32-bit registers (D200, D202), (D204, D206), (D208, D210), (D212, D214).

   b) Select “draw clockwise arc” and “average resolution” (**S** = K0)

   c) Set M0 to On to start the motion.

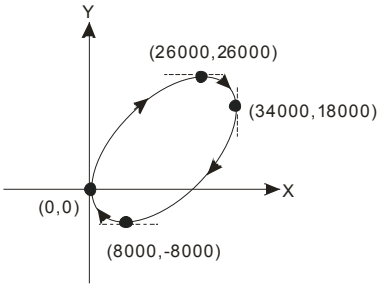


3. Motion description:

   When M0 = On, the ELC will start drawing the first segment of the arc. D0 will increment each time a segment of arc is complete, which also initiates the second segment of the arc. This execution will continue until the fourth segment of the arc is complete.

Program Example 2:

1. Draw a tilted ellipse per the figure below.

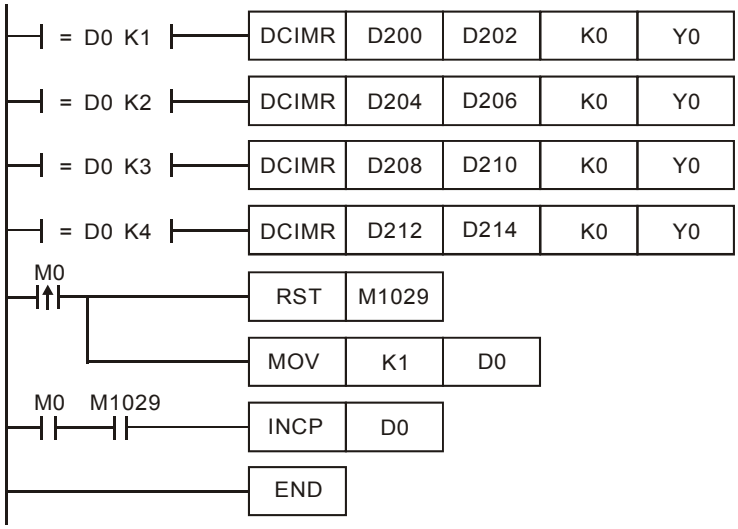


2. Steps:

a) Find the max. and min. coordinates for the X and Y axes (0,0), (26000,26000), (34000,18000), (8000,-8000) per the figure above. Calculate the relative coordinates of the four points and obtain (26000,26000)、(8000,-8000)、(-26000,-26000), (-8000,8000). Place them in the 32-bit registers (D200,D202), (D204,D206), (D208,D210) and (D212,D214).

b) Select “draw clockwise arc” and “average resolution” (S = K0).

c) Set M0 to On to start the motion.



3. Motion explanation:

When M0 = On, the ELC will start drawing the first segment of the arc. D0 will increment each time a segment of the arc is complete, which initializes the second segment of the arc. This execution will continue until the fourth segment of the arc is complete.

API	Mnemonic		Operands	Function
194	D	CIMA	S_1, S_2, S, D	2-Axis Absolute Position Arc Interpolation

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DCIMA: 17 steps
S_2					*	*							*			
S													*			
D		*														

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S1: Number of X axis output pulses **S2:** Number of Y axis output pulses **S:** Parameter settings **D:** Pulse output address

Description: (ELC-PV, ELC2-PV)

1. Flags: M1029, M1030, M1334, M1335. See API 191 DPPMR for details.
2. This instructin only supports “pulse + direction” mode.
3. S_1 and S_2 are the number of output pulses for the X axis (Y0 or Y4) and Y axis (Y2 or Y6). The range of the values is -2,147,483,648 ~ +2,147,483,647. When S_1 and S_2 are greater than the pulse present value registers CH0 (D1337 high word, D1336 low word), CH1 (D1339 high word, D1338 low word), CH2 (D1376 high word, D1375 low word), and CH3 (D1378 high word, D1377 low word), the output direction will be positive and direction signals Y1, Y3, Y5, Y7 will be On. When S_1 and S_2 are less than the pulse present value registers, the output direction will be negative and direction signals Y1, Y3, Y5, Y7 will be Off.
4. The lower 16 bits of **S** (settings for the direction and resolution): K0 refers to clockwise 10-segment (average resolution) output; K2 refers to clockwise 20-segment (higher resolution) output for a 90° arc (see figure 1 and 2). K1 refers to counterclockwise 10-segment (average resolution) output; K3 refers to counterclockwise 20-segment (higher resolution) output for a 90° arc (see figure 3 and 4).
5. The higher 16 bits of **S** (the settings for the motion time): K0 refers to 0.1 second. The range for the average resolution is K1 ~ K100 (0.1 sec. ~ 10 secs.), for the higher resolution the range is K2 ~ K200 (0.2 sec. ~ 20 secs.).

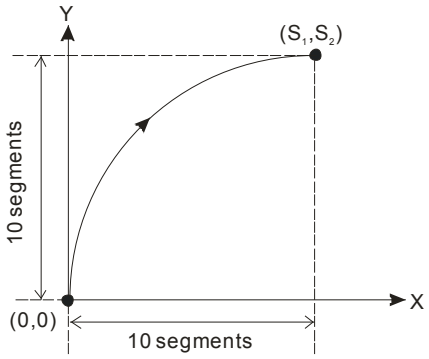


Figure 1

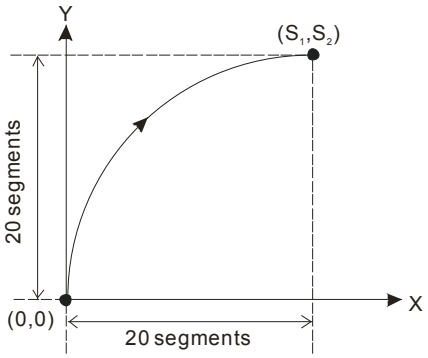


Figure 2

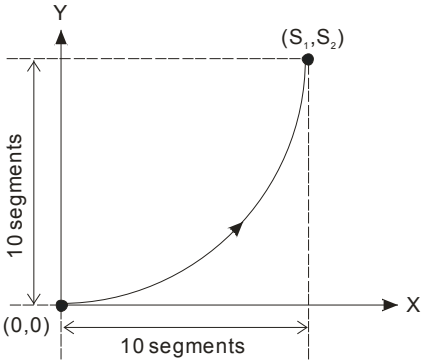


Figure 3

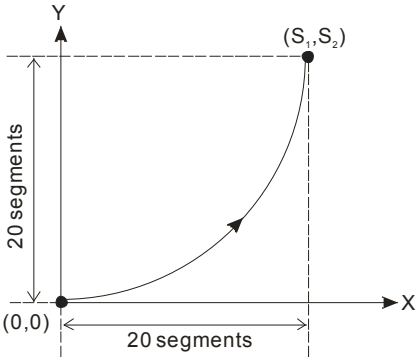


Figure 4

6. **D** can use Y0 and Y4.

When Y0 is used:

Y0 refers to the 1st group of X-axis pulse outputs.

Y1 refers to the 1st group of the X-axis direction signal.

Y2 refers to the 1st group of Y-axis pulse outputs.

Y3 refers to the 1st group of the Y-axis direction signal.

When Y4 is used:

Y4 refers to the 2nd group of X-axis pulse outputs.

Y5 refers to the 2nd group of the X-axis direction signal.

Y6 refers to the 2nd group of Y-axis pulse outputs.

Y7 refers to the 2nd group of the Y-axis direction signal.

When the instruction execution is complete, the direction signal output will not turn off until the conditions driving the instruction go false.

7. Draw four 90° arcs.
8. When the direction signal is On, the direction is positive. When the direction signal is Off, the direction is negative. When **S** is set to K0, K2, the arcs will be clockwise (see figure 5). When **S** is set to K1, K3, the arcs will be counterclockwise (see figure 6).

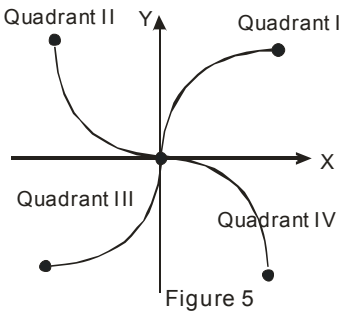


Figure 5

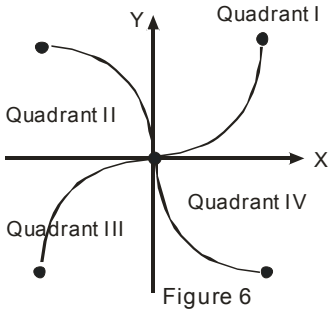


Figure 6

9. When the 2-axis motion is being executed in 10 segments (of average resolution), the operation time of the instruction when the instruction is first enabled is approximately 5ms. The number of output pulses cannot be less than 100 or more than 1,000,000; otherwise, the instruction cannot be enabled.
10. When the 2-axis motion is being executed in 20 segments (of high resolution), the operation time of the instruction when the instruction is first enabled is approximately 10ms. The number of output pulses cannot be less than 1,000 or more than 10,000,000; otherwise, the instruction cannot be enabled.
11. If the number of pulses in 10-segment or 20-segment motion needs to be lower or higher than the allowable range, adjust the gear ratio of the servo to obtain the desired value
12. Each time the instruction is executed, only one 90° arc can be drawn.
13. There are no settings for the start frequency and acceleration/deceleration time.
14. There is no limit on the number of times this instruction can be used in the program
15. The settings for the direction and resolution are in the lower 16 bits of **S** and can only be K0 ~ K3.
16. The settings for the motion time are in the high 16 bits of **S** and can be slower than the fastest suggested time but shall not be faster than the fastest suggested time.
17. The fastest suggested time for the arc interpolation:

Segments	Max. target position (pulse)	Fastest suggested set time (unit:100ms)
Average resolution	100 ~ 10,000	1
	10,001 ~ 19,999	2
	:	:
	Less than 1,000,000	Less than 100
Higher resolution	1,000 ~ 20,000	2
	20,000 ~ 29,999	3
	:	:
	Less than 10,000,000	Less than 200

Description: (ELCM-PH/PA, ELC2-PC/PA/PE)

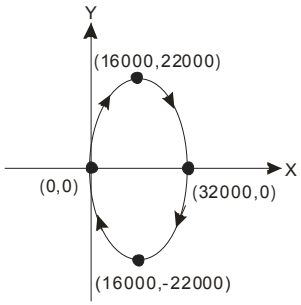
1. Flags: M1029. See API 191 DPPMR for details.
2. $\mathbf{S_1}$ and $\mathbf{S_2}$ are the designated number of output pulses in the X axis (Y0) and the Y axis (Y2).

The range of the values is -2,147,483,648 ~ +2,147,483,647 (+/- represents the forward/backward direction). When in forward direction, the pulse present value registers CH0 (D1031 high word, D1030 low word), CH1 (D1337 high word, D1336 low word) will increase. When in backward direction, the present value will decrease.

3. For additional information refer to API 193 DCIMR.

Program Example 1:

1. Draw an ellipse per the figure below.

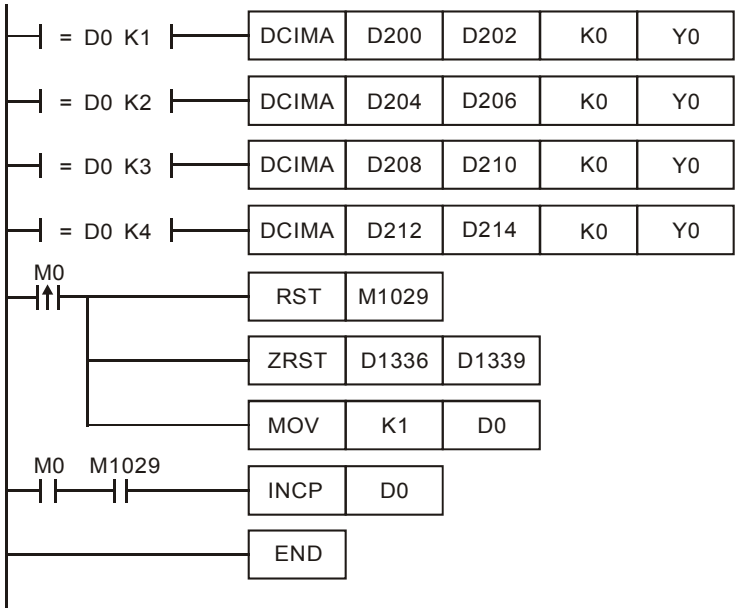


2. Steps:

   a) Set the four coordinates (0,0), (16000, 22000), (32000, 0), (16000, -22000) per the figure above. Place them in the 32-bit registers (D200, D202), (D204, D206), (D208, D210), (D212, D214).

   b) Select “draw clockwise arc” and “average resolution” (S = K0).

   c) Set M0 to On to start the motion.



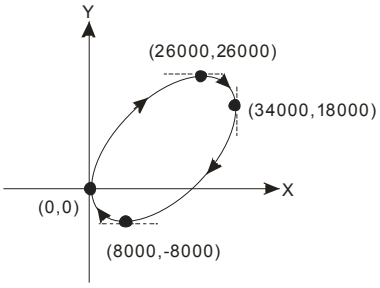
3. Motion description:

When M0 = On, the ELC will start drawing the first segment of the arc. D0 will increment by 1 each

time a segment of the arc is complete, which also initiates the second segment of the arc. This execution will continue until the fourth segment of the arc is complete.

Program Example 2:

1. Draw a tilted ellipse as the figure below.

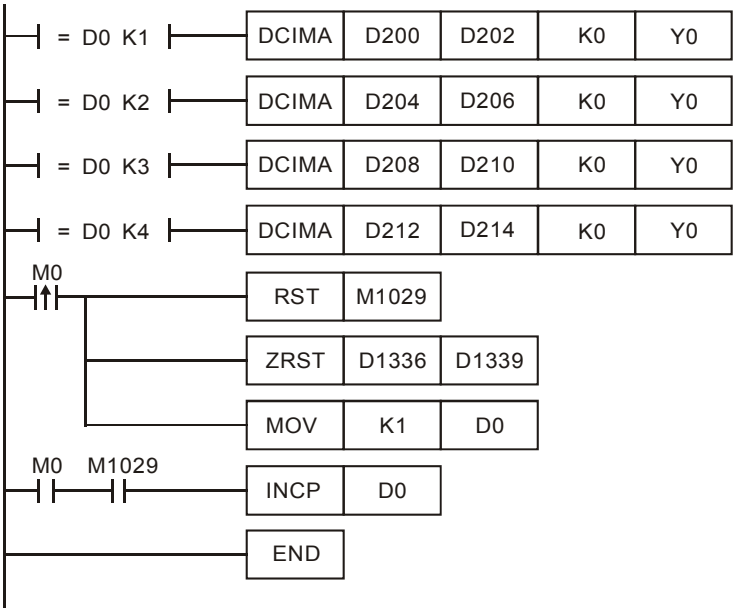


2. Steps:

   a) Find the max. and min. coordinates on the X and Y axes (0,0), (26000,26000), (34000,18000), (8000,-8000) per the figure above. Place them in the 32-bit registers (D200,D202), (D204,D206), (D208,D210) and (D212,D214).

   b) Select “draw clockwise arc” and “average resolution” (S = K0).

   c) Set M0 to On to start the motion.



3. Motion description:

When M0 = On, the ELC will start drawing the first segment of the arc. D0 will increment each time a segment of the arc is complete, which also initiates the second segment of the arc. This execution will continue until the fourth segment of the arc is complete.

API	Mnemonic		Operands	Function
195	D	PTPO	S_1, S_2, D	Single-Axis Pulse Output by Table

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			DPTPO: 13 steps
S_2													*			
D		*														

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Starting source address S_2: Number of segments **D**: Pulse output address

Description: (ELC-PV, ELC2-PV)

1. Flags: M1029, M1030, M1334, M1335. See the remarks below for details.
2. The value of S_2, determines the number of segments and each segment uses four D registers. (S_1 + 0) refers to the output frequency. (S_1 + 2) refers to the number of output pulses.
3. When the output frequency of S_1 is less than 1, the ELC will use a value of 1. When the value is greater than 200,000KHz, the ELC will use 200,000KHz.
4. S_2 + 0: the number of segments (range: 1 ~ 60). S_2 + 1: the number of segments being executed. Each time the program scans the instruction, it will update the segment number that is currently being executed.
5. **D** can only designate output addresses Y0, Y2, Y4 and Y6 and can only perform pulse output control.
6. This instruction does not offer acceleration and deceleration functions. When the instruction is disabled, the output pulses will stop immediately.
7. There is no limit to the number of times this instruction can be used in the program, but only one can be executed at a time for each channel.
8. When the instruction is being executed, the frequency and the number of the segments cannot be modified.

Description: (ELCM-PH/PA, ELC2-PB/PH/PA/PE)

1. S_1 specifies the output frequency and the number of pulses based on the number of segments set by S_2. Each segment occupies 4 consecutive registers in S_1. (S_1+0) and (S_1+1) store the output frequency; (S_1+2) and (S_1+3) store the number of output pulses.
2. Available output frequencies for S_1: 6Hz~100,000Hz.
3. S_2 is the total number of segments (range: 1 ~ 40). S_2 + 1: The number of the current executing segment. The number in S_2 + 1 will be updated when the ELC scans this instruction.
4. **D** can only be designated with output addresses Y0 and Y2.

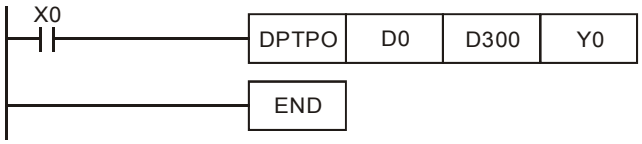
5. This instruction does not offer the ramp up/down function. Therefore, when the instruction is disabled, the output pulses will stop immediately.
6. There is no limit to the number of times this instruction can be used in the program, but only one can be executed at a time for each channel.
7. When the instruction is being executed, changes to the instruction parameters will be invalid.

Program Example:

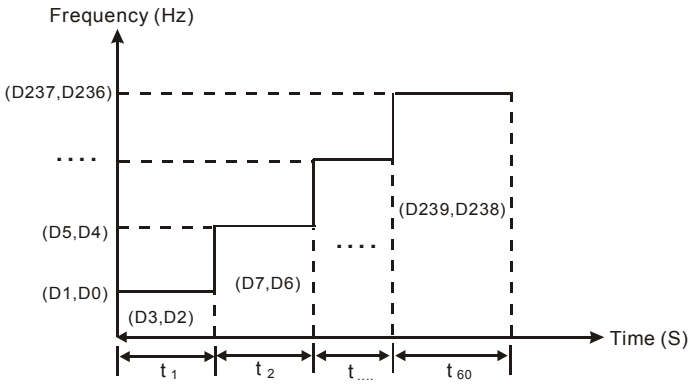
1. When X0 = On, the output will be operated according to the frequency and number of pulses per segment.
2. Format of the table:

S_2 = D300, number of segments (D300 = K60)	S_1 = D0, frequency (S_1 + 0)	S_1 = D0, number of output pulses (S_1 + 2)
K1 (1st segment)	D1, D0	D3, D2
K2 (2nd segment)	D5, D4	D7, D6
: :	: :	: :
K60 (60th segment)	D237, D236	D239, D238

3. Monitor the segment number that is currently being executed in register D301.



4. The pulse output curve:



Points to note:

1. ELC-PV, ELC2-PV Flag descriptions:

   M1029: On when CH0 (Y0) pulse output is complete.

M1030: On when CH1 (Y2) pulse output is complete.

M1036: On when CH2 (Y4) pulse output is complete.

M1037: On when CH3 (Y6) pulse output is complete.

M1334: When On, CH0 (Y0) pulse output is in error.

M1335: When On, CH1 (Y2) pulse output is in error.

M1520: When On, CH2 (Y4) pulse output is in error.

M1521: When On, CH3 (Y6) pulse output is in error.

M1336: CH0 (Y0) pulse output indication flag

M1337: CH1 (Y2) pulse output indication flag

M1522: CH2 (Y4) pulse output indication flag

M1523: CH3 (Y6) pulse output indication flag

2. ELC-PV, ELC2-PV Special register descriptions:

D1336, D1337: Pulse present value register for CH0 (Y0) (D1337 high word, D1336 low word)

D1338, D1339: Pulse present value register for CH1 (Y2) (D1339 high word, D1338 low word)

D1375, D1376: Pulse present value register forCH2 (Y4) (D1376 high word, D1375 low word)

D1377, D1378: Pulse present value register forCH3 (Y6) (D1378 high word, D1377 low word)

3. ELCM-PH/PA, ELC2-PB/PH/PA/PE Flag descriptions:

M1029 CH0 (Y0) pulse output execution complete.

M1102 CH1 (Y2) pulse output execution complete

M1078 CH0 (Y0) pulse output pause (immediate)

M1104 CH1 (Y2) pulse output pause (immediate)

M1538 Indicating pause status of Y0

M1540 Indicating pause status of Y2

4. ELCM-PH/PA, ELC2-PB/PH/PA/PE Special register explanations:

D1030 Low word of the present value of the Y0 pulse output

D1031 High word of the present value of the Y0 pulse output

D1336 Low word of the present value of the Y2 pulse output

D1337 High word of the present value of the Y2 pulse output

API	Mnemonic		Operands	Function
196	HST	P	**S**	High Speed Timer

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*										HST, HSTP: 3 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

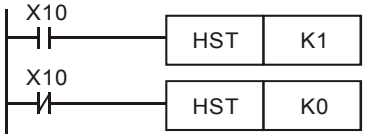
S: Start/stop condition for the high speed timer

Description:

1. Range of **S**: S = K0 or K1.
2. Flag: M1015: See below for details.
3. When **S** = 1, and the high speed timer is executed, M1015 will be On and, the high speed timer will start to time and record the present value in D1015 (units: 100us).
4. The timing range of D1015: K0 ~ K32,767. When the accumulated value equals K32,767, the timer will roll over to 0..
5. When **S** = 0, the high speed timer will be disabled and M1015 = Off. D1015 will stop the timing immediately.
6. When **S** is other than 0 or 1, the HST instruction will not execute.

◆ Program Example:

1. When X10 = On, M1015 will be On. The high speed timer will start to time and record the present value in D1015.
2. When X10 = Off, M1015 will be Off. The high speed timer will be shut down.



Points to note:

1. Flag descriptions:
   M1015: high speed timer start-up indication
   D1015: high speed timer accumulated value

API	Mnemonic		Operands	Function
197	D	CLLM	**S_1, S_2, S_3, D**	Close Loop Position Control

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1	*											*				DCLLM: 17 steps
S_2					*	*							*			
S_3					*	*							*			
D		*														

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

- **S_1**: Feedback source address **S_2**: Target number of feedbacks
  S_3: Target frequency of the output **D**: Pulse output address
- Description: (ELC-PV, ELC2-PV)

1. Flags: M1029, M1030, M1334, M1335. See the remarks below for details.
2. The corresponding interrupt of S1:

					• C241 ~ C254			

- ☐ = 1: rising-edige trigger; ☐ = 0: falling-edge trigger

a) When **S_1** designates X as the input points and the pulse output reaches the set target number of feedbacks in **S_2**, the output will continue to operate with the frequency of the last segment until the input interrupt occurs.

b) When **S_1** designates a high speed counter and the pulse output reaches the set target number of feedbacks in **S_2**, the output will continue to operate with the frequency of the last segment until the feedback pulses reach the target number.

c) S1 can be a high speed counter C or an external input interrupt. If S1 is C, the DCNT instruction should be the first executed to enable the high-speed counting function.

d) The range of S_2 is -2,147,483,648 ~ +2,147,483,647 (+/- represents the forward/backward direction). When in forward direction, the pulse present value registers CH0 (D1337 high word, D1336 low word), CH1 (D1339 high word, D1338 low word), CH2 (D1376 high word, D1375 low word) and CH3 (D1378 high word, D1377 low word) will increase. When in reverse direction, the present value will decrease.

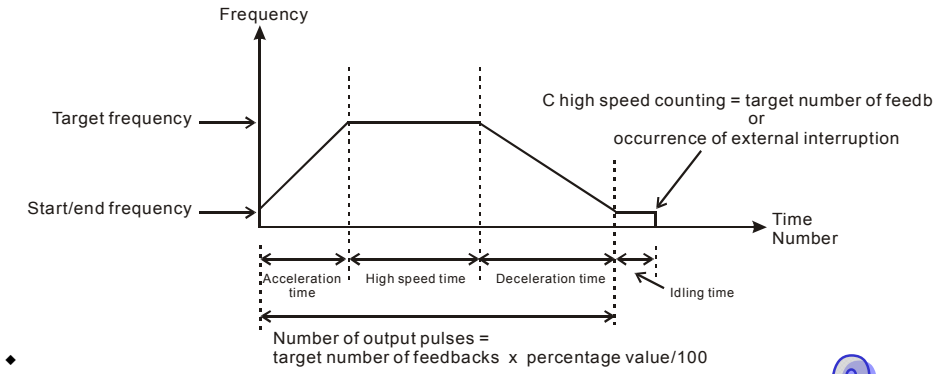
3. If S_3 is less than 10Hz, the output will operate at 10Hz; if S_3 is greater than 200KHz, the output will operate at 200KHz.
4. D can only designate Y0, Y2, Y4 and Y6 and the direction signals are Y1, Y3, Y5 and Y7.

- When the instruction execution is complete, the direction signal output will not turn off until the the conditions driving the instruction go false.

5. D1340, D1352, D1379 and D1380 are the settings for the start/end frequencies of CH0 ~ CH3. The minimun frequency is 10Hz and the default is 200Hz.
6. D1343, D1353, D1381 and D1382 are the settings for the time of the first segment and the last deceleration segment of CH0 ~ CH3. The acceleration/deceleration time cannot be shorter than 10ms. The outptut will be operated at 10ms if the time set is less than 10ms or greater than 10,000ms. The dafault setting is 100ms.
7. D1198, D1199, D1478 and D1479 are the output/input ratio of the close loop control in CH0 ~ CH3. K1 refers to 1 output pulse out for the 100 target feedback input pulses; K200 refers to 200 output pulses out of the 100 target feedback input pulses. D1198, D1199, D1478 and D1479 are the numerators of the ratio (range: K1 ~ K10,000) and the denominator is fixed at K100 (the denominator does not need to be entered).
8. M1305, M1306, M1532 and M1533 are the direction signal flags for CH0 ~ CH3. When S_2 is a positive value, the output will be in the forward direction and the flag will be Off. When S_2 is a negative value, the output will be in the reverse direction and the flag will be On.

- Close Loop description:

1. Function: Immediately stop the high-speed pulse output based on the number of feedback pulses or external interrupt signals.
2. The execution:



◆

3

3. How to adjust the time for the positioning:

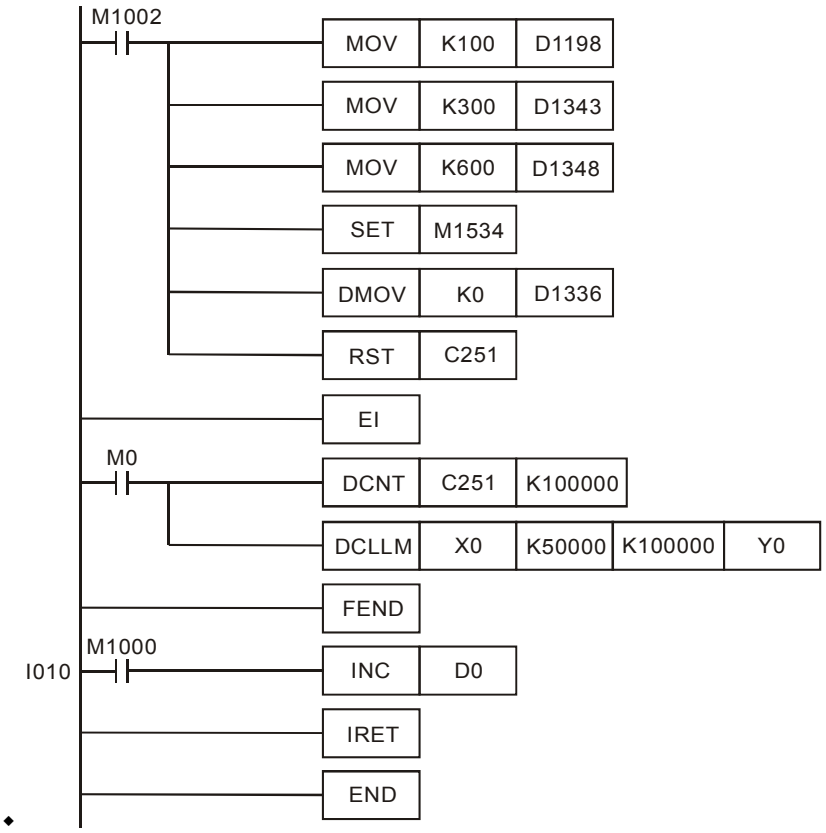
   a) The time for the completion of the positioning refers to the time for “acceleration + high speed + deceleration + idling” (see the figure above). For example, the entire number of output pulses can be increased or decreased by making the adjustment to the percentage value and further increase or decrease the time required for the positioning.

   b) Among the four segments of time, only the idling time cannot be adjusted directly. However, you can determine if the execution result is good or bad by the length of the idle time.

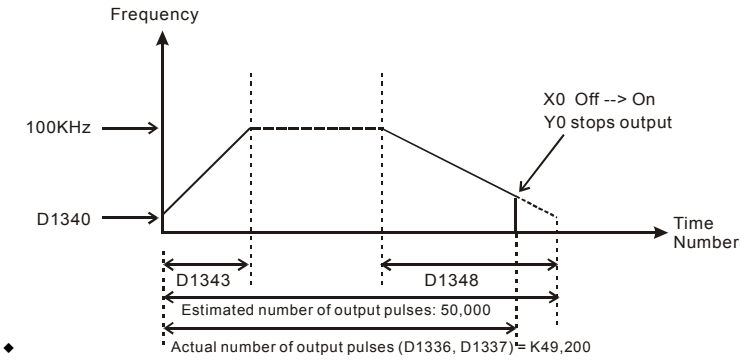
   c) Because this is a close loop operation, the length of the idle time will not be the same for each execution. Therefore, when the content of the D register for displaying the actual number of output pulses is smaller or larger than the calculated number of output pulses (taget number of feedbacks x percentage value/100), this can be improved by adjusting the percentage value, acceleration/decelartion time or the target frequency.

◆ Program Example:

1. Assume we choose X0 as the external interrupt, with I001 (rising-edge trigger) interrupt; target number of feedbacks = 50,000; target frequency = 10KHz; Y0, Y1 (CH0) for the output pulses; start/end frequency (D1340) = 200Hz; acceleration time (D1343) = 300ms; deceleration time (D1348) = 600ms; percentage value (D1198) = 100; current number of output pulses (D1336, D1337) = 0.



2. Assume the first execution result as:

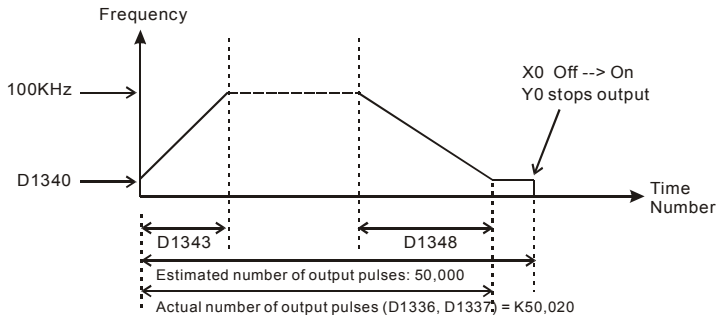


3. Note the result of the first execution:

   a) The actual output number 49,200 – estimated output number 50,000 = -800 (a negative value). A negative value indicates that the entire execution finished early and has not completed.

   b) Try shortening the acceleration time (D1343) to 250ms and the deceleration time (D1348) to 550ms.

4. The result of the second execution:



5. Note the result of the second execution:

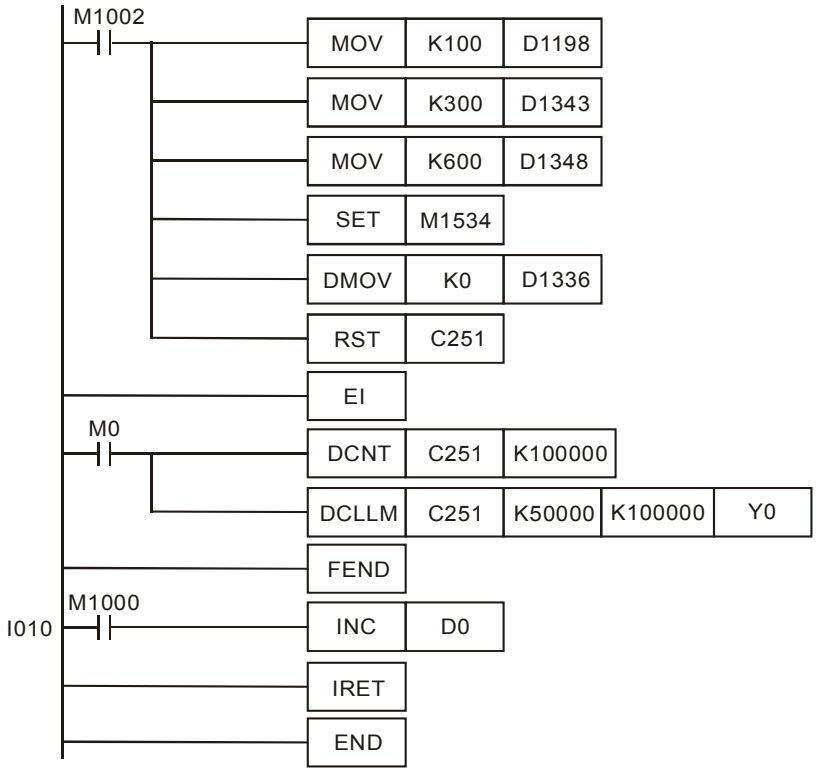
   a) The actual output number 50,020 – estimated output number 50,000 = 20

   b) 20 x (1/200Hz) = 100ms (idle time)

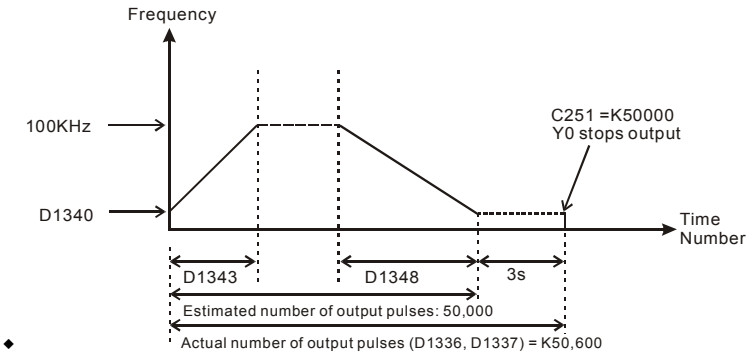
   c) 100ms is an appropriate value. Therefore, set the acceleration time to 250ms and the deceleration time to 550ms.

◆ Program Example 2:

1. Assume the feedback of the encoder is an A/B phase input and we use the C251 high speed counter (reset the counter before execution); target number of feedbacks = 50,000; target output frequency = 100KHz; Y0, Y1 (CH0) as output pulses; start/end frequency (D1340) = 200Hz; acceleration time (D1343) = 300ms; deceleration time (D1348) = 600ms; precentage value (D1198) = 100; current number of output pulses (D1336, D1337) = 0.



2. Assume the first execution result as:

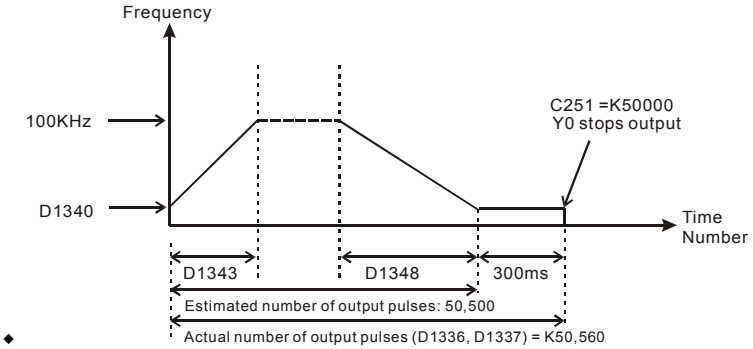


3. Note the result of the first execution:

   a) The actual output number 50,600 – estimated output number 50,000 = 600

   b) 600 x (1/200Hz) = 3s (idle time)

   c) 3 seconds are too long. Therefore, increase the percentage value (D1198) to K101.



4. The result of the second execution:

   a) The actual output number 50,560 – estimated output number 50,500 = 60

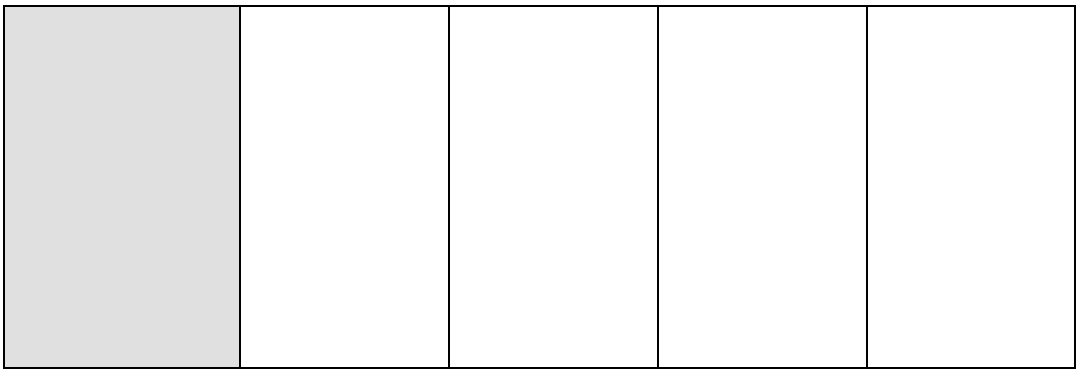
   b) 60 x (1/200Hz) = 300ms (idling time)

   c) 300ms is an appropriate value. Therefore, set the percentage value (D1198) to K101.

◆ Description: (ELCM-PH/PA, ELC2-PB/PH/PA/PE)

1. The corresponding interrupt pointers of **S_1**:

			◆ C2 4 3 ~ C 2 5 4



◆ □ = 1: rising-edige triggered; □ = 0: falling-edge triggered

a) When **S_1** designates input points X and the pulse output reaches the target number of feedbacks in **S_2**, the output will continue to operate with the last frequency until interrupts occur on input points X.

b) When **S_1** designates high speed counters and the pulse output reaches the target number of feedbacks in **S_2**, the output will continue to operate with the last frequency until the feedback pulses reache the target number.

c) **S_1** can be a high speed counter C or an input point X using an external interrupt. If **S_1** is C, the DCNT instruction should be executed before enabling the high-speed counting function. If input interrupts are used, the EI instruction with I0x0 should be enabled for external interrupts.

2. The range of **S_2** is -2,147,483,648 ~ +2,147,483,647 (+ / - indicates the positive / negative rotation direction). the present value of the pulse output in CH0 (Y0, Y1) and CH1 (Y2, Y3) increases in the positive direction and decreases in the negative direction. Registers storing the present value of the pulse output are CH0(D1031 High, 1030 Low), CH1(D1337 High, D1336 Low)

3. If **S_3** is less than 6Hz, the output will operate at 6Hz; if **S_3** is greater than 100kHz, the output will operate at 100kHz.

4. **D** can only designate Y0 (Direction signal output: Y1) or Y2 (Direction signal output: Y3). The direction signal output will be OFF only when the instruction is not being executed, i.e. the completion of the pulse output will not reset Y1 or Y3.

5. D1340 and D1352 store the start/end frequencies of CH0 and CH1. Min. 6Hz, default: 100Hz.

6. D1343 and D1353 store the ramp up/down time of CH0 and CH1. If the ramp up/down time is shorter than 20ms, ELC will operate in 20ms. Dafault: 100ms.

7. Ramp-down time of CH0 and CH1 can be specified by the setting (M1534, D1348) and (M1535, D1349). When M1534 / M1535 is ON, ramp-down time of CH0 and CH1 is set with D1348 and D1349.

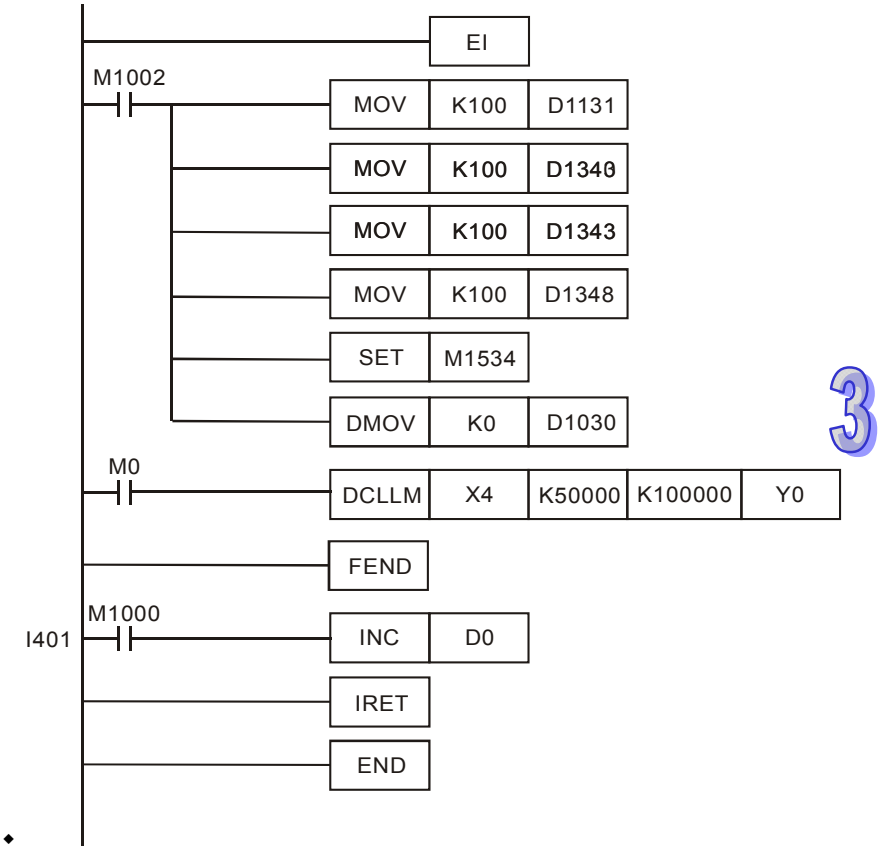
8. D1131 and D1132 are the output/input ratio(%) of the close loop control in CH0 and CH1. K1 refers to 1 output pulse out of 100 feedback pulses; K200 refers to 200 output pulses out of

the 100 feedback pulses. In the general percentage equation, the value set in D1131 and D1132 represents numerators (output pulses, available range: K1 ~ K10,000) and the denominator (the input feedbacks) is fixed at K100.

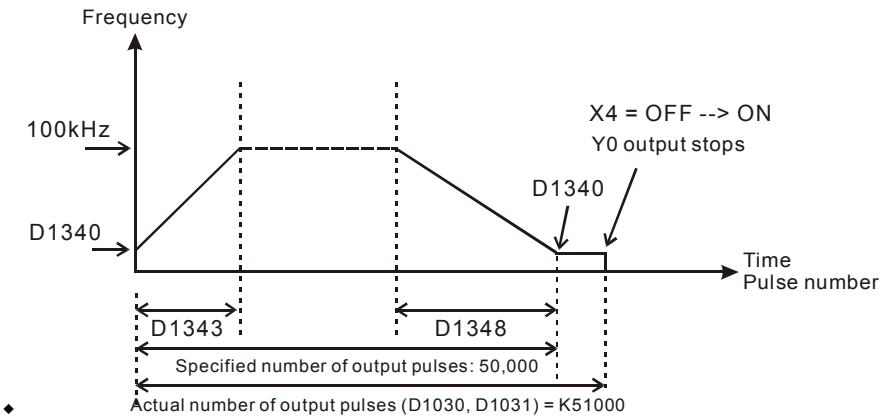
9. M1305 and M1306 can reverse the direction of CH0, CH1 pulse output. For example, when direction signal output (Y1/Y3) is OFF, the pulse output will operate in positive direction. If M1305/M1306 is set ON before the execution of this instruction, the pulse output will operate in the reverse direction.
10. When S_1 designates input points X with interrupt pointers, D1244 / D1255 can be used for setting the idle time to limited the number of pulses, in case the interrupt is not properly triggered.
11. DCLLM instruction supports the Alignment Mark and Mask function. The **Points to note** below provide more details.

- **Program Example3:** Immediately stop the high-speed pulse output with an external interrupt

1. X4 is used as the input for the external interrupt and I401 (rising-edge trigger) as the interrupt pointer. Set the target number of feedbacks = 50,000; target frequency = 100kHz; pulse output address: Y0, Y1 (CH0); start/end frequency (D1340) = 100Hz; ramp-up time (D1343) = 100ms; ramp-down time (D1348) = 100ms; percentage value (D1131) = 100; present value of output pulses (D1030, D1031) = 0.



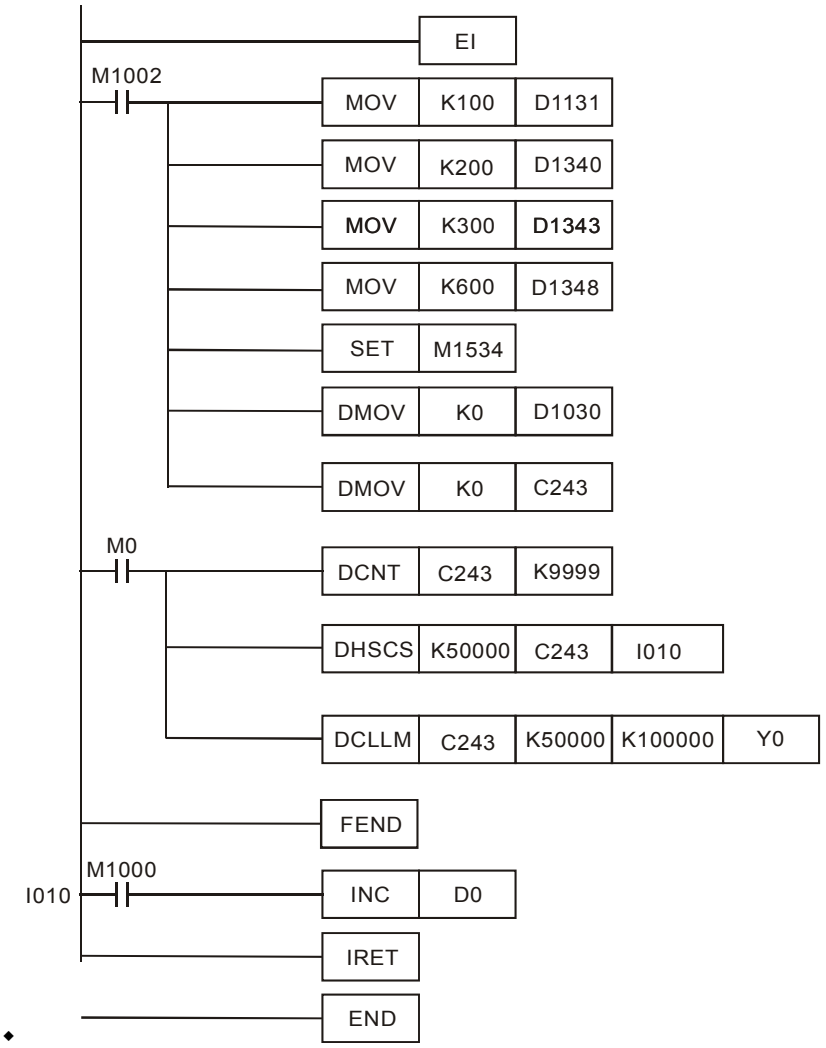
2. Execution result:



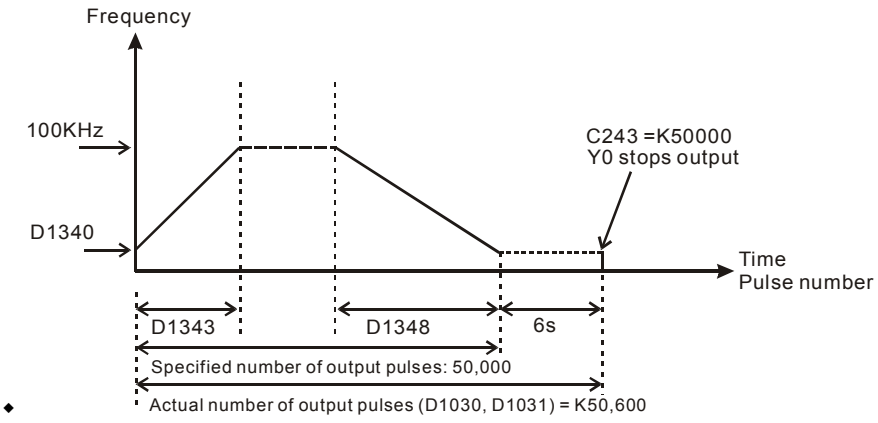
- **Program Example 4:** Immediate stop the high-speed pulse output with a high speed counter

1. C243 is used with an AB-phase input from the encoder. Set the target number of feedbacks = 50,000; target frequency = 100kHz; pulse output addresses: Y0, Y1 (CH0); start/end frequency (D1340) = 100Hz; ramp-up time (D1343) = 100ms; ramp-down time (D1348) =

100ms; percentage value (D1131) = 100; present value of output pulses (D1030, D1031) = 0..



2. Assume the first execution results are below:



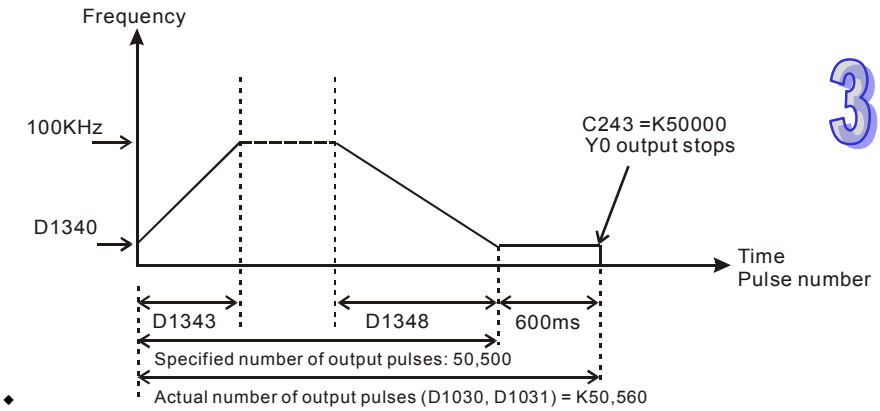
3. Note the results of the first execution:

   a) The actual output number 50,600 – specified output number 50,000 = 600

   b) 600 x (1/100Hz) = 6s (idle time)

   c) 6 seconds are too long. Therefore, increase the percentage value (D1131) to K101.

4. Obtain the results of the second execution:



5. Note the results of the second execution:

   a) The actual output number 50,560 – specified output number 50,500 = 60

   b) 60 x (1/100Hz) = 600ms (idle time)

   c) 600ms is an appropriate value. Therefore, set the percentage value (D1131) to K101.

◆ Points to note:

1. ELC-PV, ELC2-PV Flag explanations:

◆ When On, CH0, CH1, CH2 and CH3 will send pulses when encountering END instruction. Off when the output starts.

◆ On when CH0 pulse output is complete.

- On when CH1 pulse output is complete.

- On when CH2 pulse output is complete.

- On when CH3 pulse output is complete.

- When On, CH0 pulse output will is in error

- When On, CH1 pulse output will is in error.

- When On, CH2 pulse output will is in error.

- When On, CH3 pulse output will is in error.

3

- CH0 pulse output indication flag

- CH1 pulse output indication flag

- CH2 pulse output indication flag

- CH3 pulse output indication flag

- CH0 direction signal flag

- CH1 direction signal flag

- CH2 direction signal flag

- CH3 direction signal flag

- Deceleration time of CH0 setup flag (must be used with D1348)

- Deceleration time of CH1 setup flag (must be used with D1349)

- Deceleration time of CH2 setup flag (must be used with D1350)

- Deceleration time of CH3 setup flag (must be used with D1351)

2. ELC-PV, ELC2-PV Special register explanations:

◆ D1198:	◆ Close loop output/input ratio of CH0 (default: K100)
◆ D1199:	◆ Close loop output/input ratio of CH1 (default: K100)
◆ D1478:	◆ Close loop output/input ratio of CH2 (default: K100)

- D1479: Close loop output/input ratio of CH3 (default: K100)
- D1220: Phase setting of CH0 (Y0, Y1): determined by the last 2 digits of D1220; other digits are invalid.
  1. K0: Y0 output
  2. K1: Y0, Y1 AB-phase output; A ahead of B
  3. K2: Y0, Y1 AB-phase output; B ahead of A
- D1221: Phase setting of CH1 (Y2, Y3): determined by the last 2 digits of D1221; other digits are invalid.
  1. K0: Y2 output
  2. K1: Y2, Y3 AB-phase output; A ahead of B
  3. K2: Y2, Y3 AB-phase output; B ahead of A
- D1229: Phase setting of CH2 (Y4, Y5): determined by the last 2 digits of D1229; other digits are invalid.
  1. K0: Y4 output
  2. K1: Y4, Y5 AB-phase output; A ahead of B
  3. K2: Y4, Y5 AB-phase output; B ahead of A
- D1230: Phase setting of CH3 (Y6, Y7): determined by the last 2 digits of D1230; other digits are invalid.
  1. K0: Y6 output
  2. K1: Y6, Y7 AB-phase output; A ahead of B

3. K2: Y6, Y7 AB-phase output; B ahead of A

◆ D12 22:	◆ Time difference between the direction signal and the pulse output of CH0
◆ D12 23:	◆ Time difference between the direction signal and the pulse output of CH1
◆ D13 83:	◆ Time difference between the direction signal and the pulse output of CH2
◆ D13 84:	◆ Time difference between the direction signal and the pulse output of CH3
◆ D13 36:	◆ Low word of the current number of output pulses of CH0
◆ D13 37:	◆ High word of the current number of output pulses of CH0
◆ D13 38:	◆ Low word of the current number of output pulses of CH1
◆ D13 39:	◆ High word of the current number of output pulses of CH1
◆ D13 75:	◆ Low word of the current number of output pulses of CH2

◆ D13 76:	◆ High word of the current number of output pulses of CH2
◆ D13 77:	◆ Low word of the current number of output pulses of CH3
◆ D13 78:	◆ High word of the current number of output pulses of CH3
◆ D13 40:	◆ Start/end frequency settings for CH0 (default: K200)
◆ D13 52:	◆ Start/end frequency settings for CH1 (default: K200)
◆ D13 79:	◆ Start/end frequency settings for CH2 (default: K200)
◆ D13 80:	◆ Start/end frequency settings for CH3 (default: K200)
◆ D13 48:	◆ Deceleration time of CH0 pulse output when M1534 = On (default: K100)
◆ D13 49:	◆ Deceleration time of CH1 pulse output when M1535 = On (default: K100)
◆ D13 50:	◆ Deceleration time of CH2 pulse output when M1536 = On (default: K100)
◆ D13 51:	◆ Deceleration time of CH3 pulse output

	when M1537 = On (default: K100)
◆ D1343:	◆ Acceleration/deceleration time for CH0 pulse output (default: K100)
◆ D1353:	◆ Acceleration/deceleration time for CH1 pulse output (default: K100)
◆ D1381:	◆ Acceleration/deceleration time for CH2 pulse output (default: K100)
◆ D1382:	◆ Acceleration/deceleration time for CH3 pulse output (default: K100)

3. ELCM-PH/PA, ELC2-PB/PH/PA/PE Flag explanations:

M1029	CH0 (Y0, Y1) pulse output execution complete.
M1102	CH1 (Y2, Y3) pulse output execution complete.
M1078	M1078 = ON, CH0 (Y0, Y1) pulse output pause (immediate)
M1104	M1104 = ON CH1 (Y2, Y3) pulse output pause (immediate)
M1108	CH0 (Y0, Y1) pulse output pause (ramp down). M1108 = ON during ramp down.
M1110	CH1 (Y2, Y3) pulse output pause (ramp down). M1110 = ON during ramp down.
M1156	Enabling the mask and alignment mark function on I400/I401(X4) corresponding to Y0.
M1158	Enabling the mask and alignment mark function on I600/I601(X6) corresponding to Y2.
M1538	Indicating pause status for CH0 (Y0, Y1).M1538 = ON when output paused.
M1540	Indicating pause status for CH1 (Y2, Y3). M1540 = ON when output paused
M1305	Reverse CH0 (Y0, Y1) pulse output direction. M1305 = ON, pulse output direction is reversed.
M1306	Reverse CH1 (Y2, Y3) pulse output direction. M1306 = ON, pulse output direction is reversed
M1347	Auto-reset CH0 (Y0, Y1) when high speed pulse output completed. M1347 will

be reset after CH0 (Y0, Y1) pulse output is completed.

M1524 Auto-reset CH1 (Y2, Y3) when high speed pulse output completed. M524 will be reset after CH1 (Y2, Y3) pulse output is completed.

M1534 Enable ramp-down time setting on Y0. Must be used with D1348

M1535 Enable ramp-down time setting on Y2. Must be used with D1349

◆

4. ELCM-PH/PA, ELC2-PB/PH/PA/PE Special register explanations:

D1026:	Pulse number for masking Y0 when M1156 = ON (Low word). The function is disabled when set value≦0. (Default = 0)
D1027:	Pulse number for masking Y0 when M1156 = ON (High word). The function is disabled when set value≦0. (Default = 0)
D1135:	Pulse number for masking Y2 when M1156 = ON (Low word). The function is disabled when set value≦0. (Default = 0)
D1136:	Pulse number for masking Y2 when M1156 = ON (High word). The function is disabled when set value≦0. (Default = 0)
D1030:	Low word of the present value for CH0 (Y0, Y1) pulse output
D1031:	High word of the present value for CH0 (Y0, Y1) pulse output
D1131:	Input/output percentage value for CH0 (Y0, Y1) close loop control. Default: K100
D1132:	Input/output percentage value for CH1 (Y2, Y3) close loop control. Default: K100
D1244:	Idle time (pulse number) setting for CH0 (Y0, Y1) The function is disabled if set value≦0.
D1245:	Idle time (pulse number) setting for CH2 (Y2, Y3) The function is disabled if set value≦0.
D1336:	Low word of the present value for CH1 (Y2, Y3) pulse output
D1337:	High word of the present value for CH1 (Y2, Y3) pulse output
D1340:	Start/end frequency of the 1st group of pulse outputs CH0 (Y0, Y1). Default: K100
D1352:	Start/end frequency of the 2st group of pulse outputs CH1 (Y2, Y3). Default: K100
D1343:	Ramp up/down time of the 1st group of pulse outputs CH0 (Y0, Y1). Default: K100
D1353:	Ramp up/down time of the 2nd group of pulse outputs CH1 (Y2, Y3). Default: K100
D1348:	CH0(Y0, Y1) pulse output. When M1534 = ON, D1348 stores the ramp-down time. Default: K100
D1349:	d) CH1(Y2, Y3) pulse output. When M1535 = ON, D1349 stores the ramp-down time. Default: K100

API	Mnemonic		Operands	Function
198	D	VSPO	S_1, S_2, S_3, D	Variable Speed Pulse Output

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			DVSPO: 17 steps
S_2					*	*							*			
S_3					*	*							*			
D		*														

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

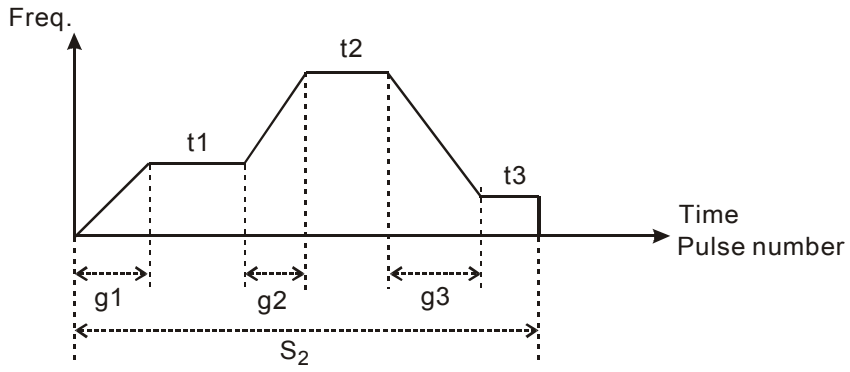
- S_1: Target frequency of the output S_2: Target number of pulses
  S_3: Gap time and gap frequency **D**: Pulse output address (Y0, Y2)

- Description:

1. The Maximum frequency for S_1 is 100kHz. The target frequency can be modified during the execution of the instruction. If S_1 is modified, VSPO will ramp up/down to the target frequency according to the gap time and gap frequency set in S_3.
2. S_2, target number of pulses is valid only when the instruction is executed the first time. S_2 can NOT be modified during the execution of the instruction. S_2 can be a negative value, however, if the output direction is not specified in D1220/D1221, the ELC will assume this value is a positive value.
3. S_3 occupies 2 consecutive 16-bit registers. S_3+0 stores the gap frequency S_3+1 stores the gap time. These parameters can be modified during the execution of the instruction. The range for S_3+0: 6Hz ~ 32767Hz; the range for S_3+0: 5ms ~ 80ms. If the value exceeds the available range, the ELC will use the upper or lower limits.
4. **D,** the pulse output address supports only Y0 and Y2. If Y1 and Y3 are required for output direction control, D1220 or D1221 must be set to K1(Pulse/Dir).
5. Parameters set in S_3 can only be modified when the value in S_1 is modified accordingly. When the target frequency is set to 0, the ELC will ramp down to a stop based on the parameters set in S_3. When the output is stopped, the ELC will enable the flags indicating pause status (Y0: M1538, Y2: M1540). If the target frequency is other than 0, the pulse output will ramp up to the target frequency and operates untill the target number of pulses are complete..

Function description:

Pulse output diagram:



1. Definitions:

   t1 → target frequency of 1st shift

   t2 → target frequency of 2nd shift

   t3 → target frequency of 3rd shift

   g1 → ramp-up time of 1st shift

   g2 → ramp-up time of 2nd shift

   g3 → ramp-down time of 3rd shift

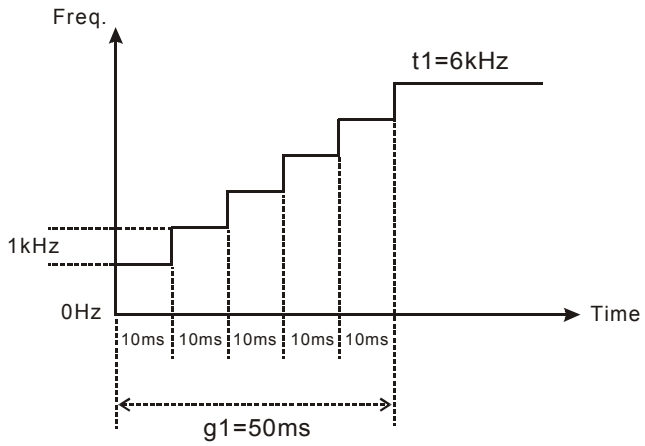
   S_2 → total output pulses

2. Explanations on each shift:

   ◆ 1st shift:

   Assume t1 = 6kHz, gap freqency = 1kHz, gap time = 10ms

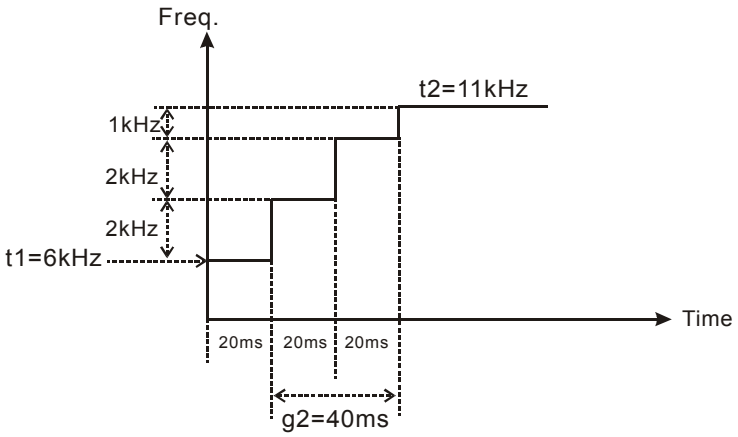
   Ramp-up steps of 1st shift:



   ◆ 2nd shift:

   Assume t2 = 11kHz, internal frequency = 2kHz, gap time = 20ms

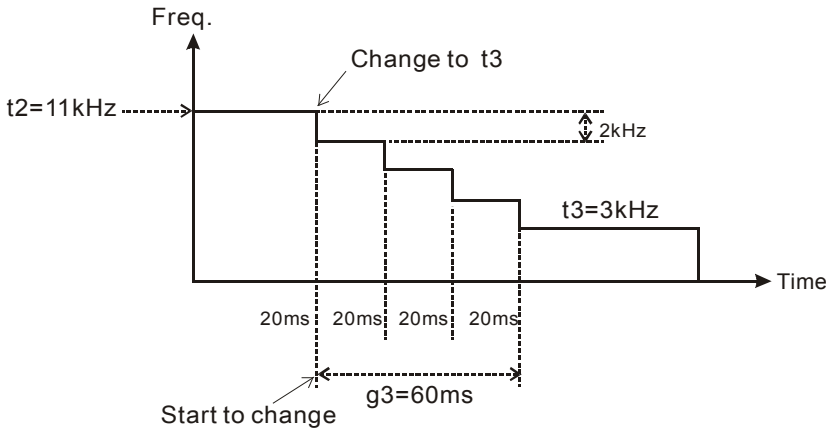
   Ramp-up steps of 2nd shift:



◆ 3rd shift:

Assume t3 = 3kHz, gap frequency = 2kHz, gap time = 20ms

Ramp-down steps of 3rd shift:



◆ For program examples please refer to API 199

◆ Points to note:

◆ Flag explanations:

M1029: CH0 (Y0, Y1) pulse output execution complete

M1102: CH1 (Y2, Y3) pulse output execution complete

M1078: Y0 pulse output pause (immediate)

M1104: Y2 pulse output pause (immediate)

M1305: Reverse Y1 pulse output direction for the high speed pulse output instructions

M1306: Reverse Y3 pulse output direction for the high speed pulse output instructions

M1538: Indicating pause status of Y0

M1540: Indicating pause status of Y2

◆ Special register descriptions:

D1030: Low word of the present value of Y0 pulse output

D1031: High word of the present value of Y0 pulse output

D1336: Low word of the present value of Y2 pulse output

D1337: High word of the present value of Y2 pulse output

D1220: Pulse output mode setting of CH0 (Y0, Y1). Please refer to PLSY instruction.

D1221: Pulse output mode setting of CH1 (Y2, Y3). Please refer to PLSY instruction

API	Mnemonic	Operands	Function
199	D ICF	S_1, S_2, D	Immediately Change Frequency

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			DVSPO: 13 steps
S_2					*	*							*			
D		*														

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

3

Operands:

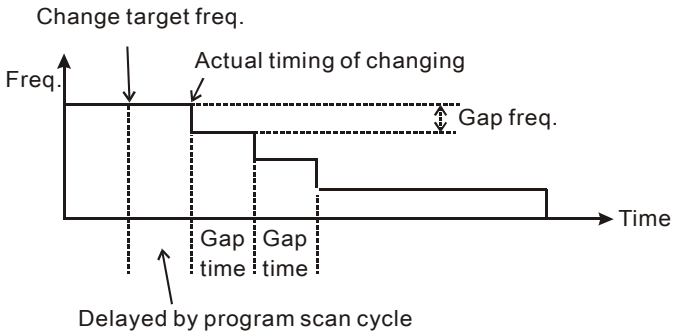
- S_1: Target frequency to be changed S_2: Gap time and gap frequency D: Pulse output address (Y0, Y2)

- Description:

1. Max frequency for S_1: 100kHz. When the ICF instruction executes, it will immediately start to change the frequency with the ramp-up/down process.
2. The ICF instruction must be executed after the execution of DVSPO or DPLSY instructions. When this instruction is used together with the DVSPO instruction, operands S_1, S_2, **D** of the DICF instruction must be assigned the same addresses S_1, S_3, **D** as DVSPO. When the instruction is used with DPLSY, operands S_1 and **D** must be assigned the same addresses as S_1 and **D** of DPLSY**.**
3. If the ICF instruction is used with the DPLSY instruction, operand S_2 is invalid.
4. When the ICF instruction is used with the DVSPO instruction, parameter S_2 functions the same as S_3 in the DVSPO instruction, specifying the gap time and gap frequency of the ramp-up/down process.
5. **D,** the pulse output address supports only Y0 and Y2.
6. It is recommended that the instruction be applied in an interrupt subroutine to obtain a better response time and more accurate execution...
7. For associated flags and registers, please refer to **Points to note** for the API 198 DVSPO instruction.

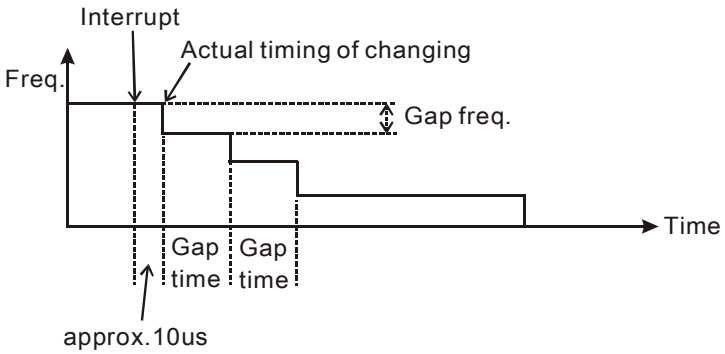
Function Description:

1. If the target frequency is changed by using the DVSPO instruction, the timing change will be delayed due to the program scan time and the gap time as shown below.



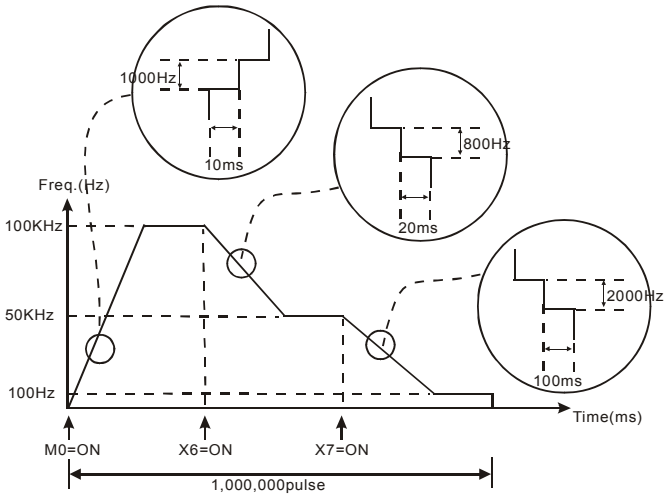
2. If the target frequency is changed with the DICF instruction in an interrupt subroutine, the change will be executed immediately with an approximate delay of 10us (the execution time of the DICF instruction).

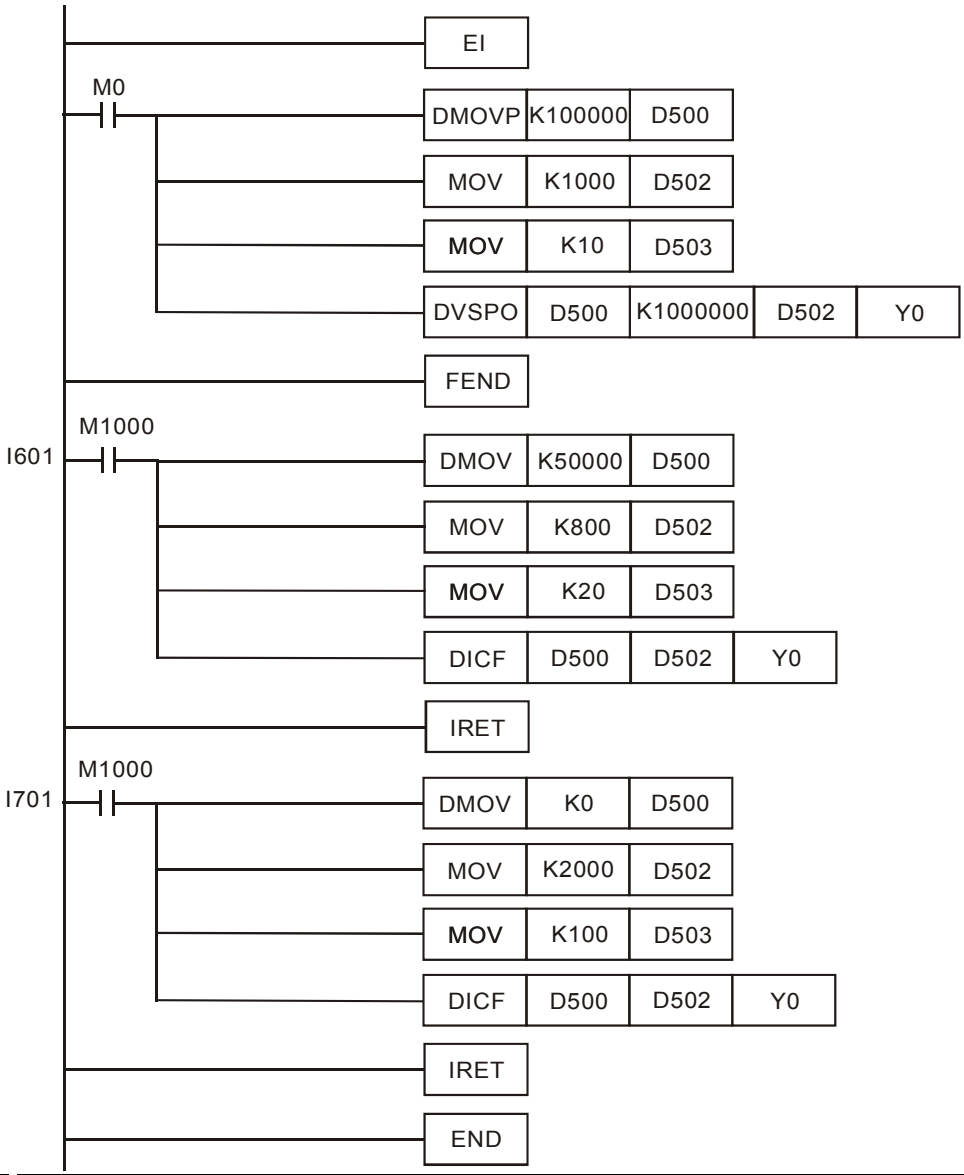
   The timing diagram is as below:



Program Example:

1. When M0 = ON, pulse output ramps up to 100kHz. Total shifts: 100, Gap frequency: 1000Hz, Gap time: 10ms. Calculation of total shifts: (100,000 – 0) ÷ 1000 = 100.
2. When the X6 external interrupt executes, the target frequency is changed and is ramped down to 50kHz immediately. Total shifts: 150, Gap frequency: 800Hz, Gap time: 20ms. Calculation of total shifts: (100,000 – 50,000) ÷ 800 = 125
3. When the X7 external interrupt executes, the target frequency is changed and is ramped down to 100Hz immediately. Total shifts: 25, Gap frequency: 2000Hz, Gap time: 100ms. Calculation of total shifts: (50,000 – 100) ÷ 2000 = 25.
4. When the pulse output reaches 100Hz, the frequency is kept constant and the pulse output stops when 1,000,000 pulses are completed.





API	Mnemonic	Operands	Function

202	SCA L	P	S_1, S_2, S_3, D	Calculation of Proportional Value

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			SCAL,SCLAP: 9 steps
S_2					*	*							*			
S_3					*	*							*			
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

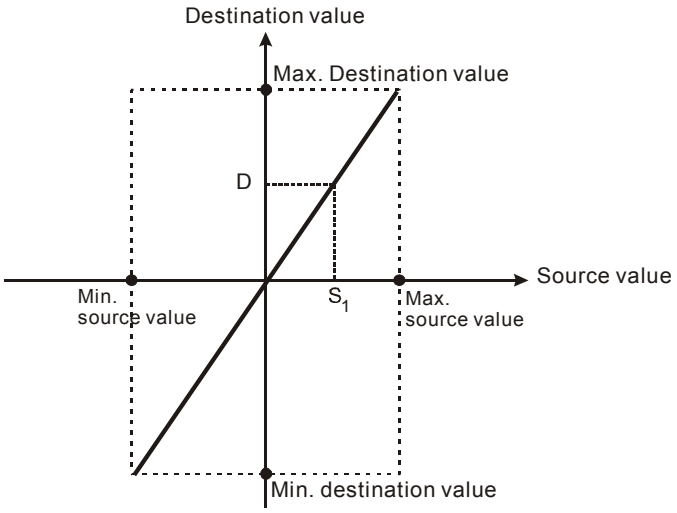
Operands:

S_1: Source value **S_2**: Slope. The units of S_2 are 0.001 **S_3**: Offset **D**: Destination address

The range for operands **S1, S2, S3** is -32767~32767.

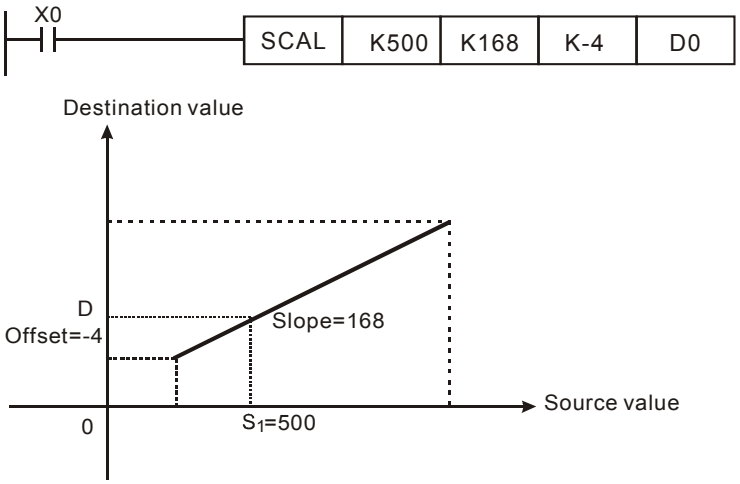
Description:

60. Equation: **D** = (**S_1** × **S_2**)÷1000+ **S_3**
61. The values for **S_2** and **S_3** must be calculated using the slope and offset equations below with decimal rounding to get 16-bit integer values.
62. Slope equation: **S_2** = [(Max. destination value – Min. destination value)÷(Max. source value – Min. source value)] ×1000
63. Offset equation: **S_3** =Min. destination value – (Min. source value × (**S_2**÷1000))
64. The output curve is as follow:



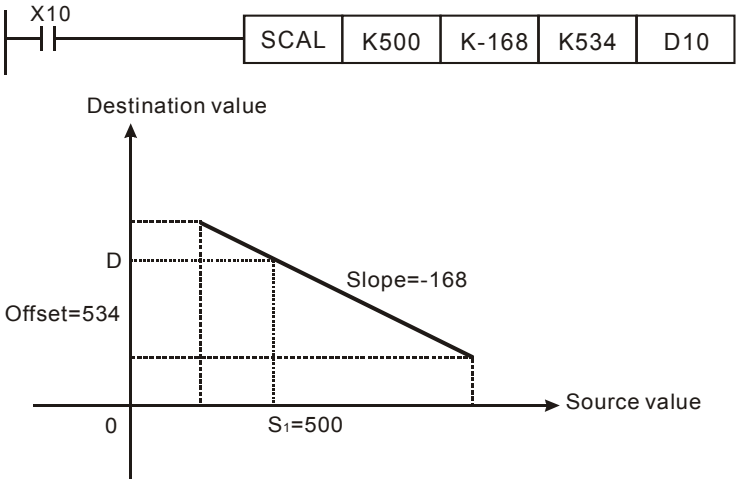
Program Example 1:

65. **S_1** is 500, **S_2** is 168 and **S_3** is –4. When X0=On, SCAL instruction executes and the scaled value is stored in D0.
66. Equation：D0 = (500 × 168)÷1000+ (-4)=80



Program Example 2:

67. S_1 is 500, S_2 is –168 and S_3 is 534. When X10=On, the SCAL instruction executes and the scaled value is stored in D10.
68. Equation：D10 = (500 × -168)÷1000+ 534=450



Points to notes:

69. If you do not wish to calculate the slope and offset, use the SCLP instruction.
70. When using the slope equation, the user must be aware that the Max. source value must be larger than the Min. source value. The Max. destination value does not need to be larger than the Min. destination value.
71. If **D** > 32,767, **D** =32,767. If **D** < -32,768, **D** =-32,768.
72. In ELCB-PB, ELC-PA, SCAL instruction support ELCB-PB V1.4(above) and ELC-PA V1.4(above).

API	Mnemonic	Operands	Function
203	D SCLP P	**S_1, S_2, D**	Calculation of Parameter Proportional Value

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			SCLP, SCLPP: 9 steps
S_2													*			DSCLP, DSCLPP: 13 steps
D													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Source value **S_2**: Parameters **D**: Destination address

Description:

73. **S_2** Parameter settings for the 16-bit SCLP instruction are as follows

Device No.	Parameter	Range
S_2	Max. source value	-32768~32767
S_2+1	Min. source value	-32768~32767
S_2+2	Max. destination value	-32768~32767
S_2+3	Min. destination value	-32768~32767

74. For the 16-bit instruction, operand **S_2** will use 4 continuous registers.

75. **S_2** Parameter settings for the 32-bit SCLP instruction are as follows

Device No.	Parameter	Range	
		Integer	Floating point number
S_2、**S_2**+1	Max. source value	-2,147,483,648~2,147,483,647	Range of 32-bit floating point number
S_2+2、3	Min. source value		
S_2+4、5	Max. destination value		
S_2+6、7	Min. destination value		

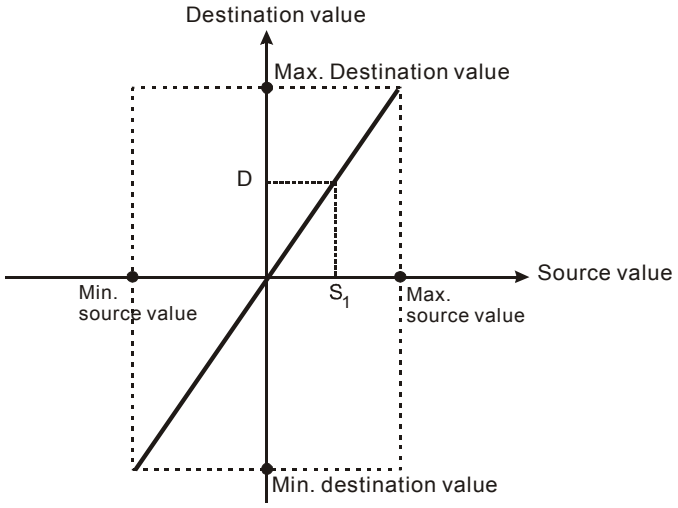
4. For the 32-bit instruction, operand **S_2** will use 8 continuous registers.

5. Equation used for scaling by this instruction is **D** = [(**S_1** – Min. source value) × (Max. destination value – Min. destination value)] ÷ (Max. source value – Min. source value) + Min. destination value

6. Equation of source value and destination value:

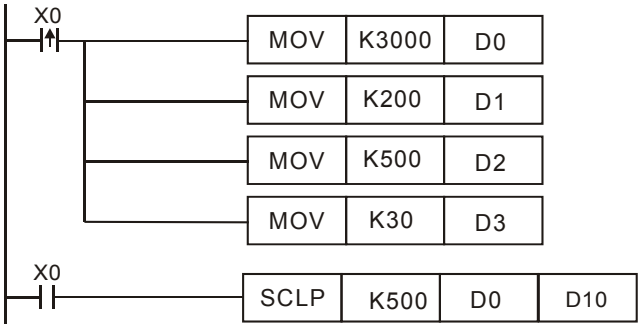
y=kx+b, y=Destination value (**D**) , k=slope=(Max. destination value – Min. destination value)÷ (Max. source value – Min. source value) , x=Source value (**S_1**) , b=offset=Min. destination value – Min. source value × slope

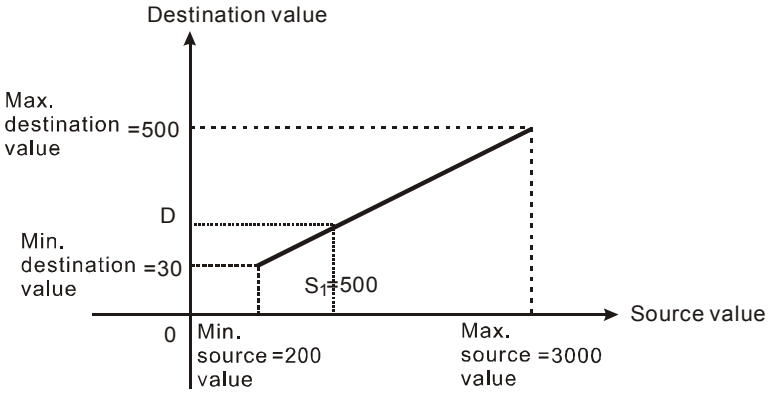
7. If **S_1** > Max. source value, **S_1** = Max. source value. If **S_1** < Min. source value, **S_1** = Min. source value. When the input value and parameters are set, the output curve is shown below:



Program Example 1:

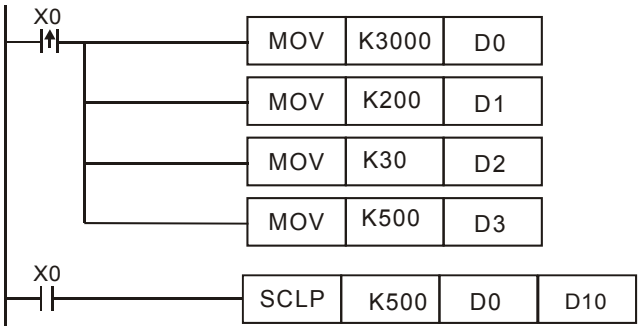
8. **S_1** is 500, the Max. source value is D0=3000, the Min. source value is D1=200, the Max. destination value is D2=500, and the Min. destination value is D3=30. When X0=On, the SCLP instruction executes and the scaled value is stored in D10.
9. Equation: D10 = [(500 –200) × (500–30)] ÷(3000–200) +30=80.35. Rounding off the result into an integer, D10 =80.

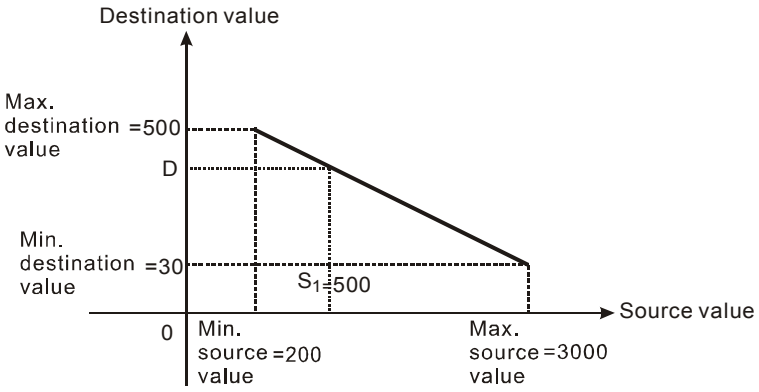




Program Example 2:

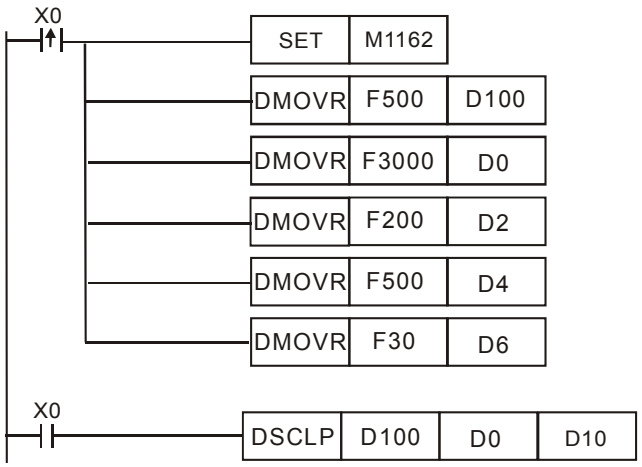
10. **S_1** is 500, the Max. source value is D0=3000, the Min. source value is D1=200, the Max. destination value is D2=30 and Min. destination value is D3=500. When X0=On, the SCLP instruction executes and the scaled value is stored in D10.
11. Equation: D10 = [(500 –200) × (30–500)] ÷(3000–200) +500=449.64. Rounding off the result into an integer, D10 =450.





Program Example 3:

12. The source value S_1 is D100=F500, the Max. source value is D0=F3000, the Min. source value is D2=F200, the Max. destination value is D4=F500 and the Min. destination value is D6=F30. When X0=On, SET M1162 which means the DSCLP instruction will assume floating point values. The scaled value will be stored in D10.
13. Equation: D10 = [(F500 –F200) × (F500–F30)] ÷(F3000–F200) +F30=F80.35. Rounding off the result into an integer, D10 =F80.



Points to note:

14. The range for the 16-bit operand S_1: Max. source value ≧ S_1 ≧ Min. source value. -32768~32767. If the value exceeds a boundary value, use the boundary value for the calculation.
15. The range for the 32-bit integer operand S_1: Max. source value≧S_1≧Min. source value. -2,147,483,648~2,147,483,647. If the value exceeds a boundary value, use the boundary value for the calculation.
16. The range for 32-bit floating point operand S_1: Max. source value≧S_1≧Min. source value, according to the range of 32-bit floating point number. If the value exceeds a boundary value, use the boundary value for the calculation.
17. The Max. source value should be larger than Min. source value. The Max. destination value does not need to be larger than Min. destination value.
18. In ELCB-PB, ELC-PA, SCLP instruction supports version 1.4 (and above).

API	Mnemonic		Operands	Function
205	CMPT	P	**S_1, S_2, n, D**	Compare Table

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			CMPT: 9 steps
S_2											*	*	*			CMPTP: 9 steps
n					*	*							*			
D								*	*	*	*	*	*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Source address 1 **S_2**: Source address 2 **n**: Data length (**n** = 1~16) **D**: Destination address

Description:

19. **S_1** and **S_2** can be T/C/D addresses. For Counters, only 16-bit addresses are applicable (C0~C199).
20. In ELCM-PH/PA, DCMPT instruction supports ELCM-PH/PAV2.0.
21. The value in the high 16 bits of **n** used in the 32-bit instruction is an invalid value.
22. The value in the low 8 bits of **n** indicates the data length. For the 16-bit instruction, **n** is 1 ~ 16. For the 32-bit instruction, **n** is 1 ~ 32. If **n**< 1, then **n** = 1. If **n** >maximum value, then **n** = maximum value.
23. The 16-bit data is written into **D**. If the data length is less than 16 bits, the bit which does not have a corresponding value is 0. For example, if **n** is K8, bit0~7 have corresponding values, and bit8~15 are 0.
24. The value in the high 8 bits of **n** indicates the comparison condition. The relation between the comparison conditions and the values are shown in the following table.

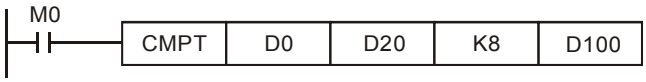
Value	K0	K1	K2	K3	K4
Comparison condition	$S_1 = S_2$	$S_1 < S_2$	**S_1 <= S_2**	$S_1 > S_2$	**S_1 >= S_2**

25. The example of setting **n**: If **n** used in the 16-bit instruction is H0108, eight pieces of data are compared with eight pieces of data in terms of “larger than”. If **n** used in the 32-bit instruction is H00000320, 32 pieces of data are compared with 32 pieces of data in terms of “less than”.
26. If the setting value of the comparison condition exceeds the range, or the firmware version does not support the comparison condition, the default comparion condition “equal to” is executed. ELCM-PH/PA V1.0(below), ELC-PV don’t support the setting of the comparison condition.
27. The 16-bit comparison values used in the 16-bit instruction are signed values. The comparison values used in the 32-bit instruction are 32-bit signed values (M1162=OFF), or floating-point numbers (M1162=ON).

28. The 16-bit data or 32-bit data is written into **D**. If the data length is less than 16 bits or 32 bits, the bit which does not have a corresponding value is 0. For example, if **n** is K8, bit0~7 have corresponding values, and bit8~bit15 or bit8~bit31 are 0.
29. If the comparison result meets the comparison condition, the corresponding bit is 1. If the comparison result does not meet the comparison condition, the corresponding bit is 0..

Program example:

When M0 = ON, compare the 16-bit values in D0~D7 with those in D20~D27 and store the result in D100.



- Contents of D0~D7:

No.	D0	D1	D2	D3	D4	D5	D6	D7
Value	K10	K20	K30	K40	K50	K60	K70	K80

- Contents of D20~D27:

No.	D20	D21	D22	D23	D24	D25	D26	D27
Value	K12	K20	K33	K44	K50	K66	K70	K88

- After the execution of the CMPT instruction, the associated bit will be 1 if two devices have the same value, and 0 for those that are not equal. Therefore the results in D100 will be as below:

	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8~15
D100	0	1	0	0	1	0	1	0	0…0
	H0052 (K82)								

API	Mnemonic	Operands	Function
207	CSFO	**S, S_1, D**	Catch Speed and Proportional Output

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1	*															CSFO: 7 steps
S_2													*			
S													*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S: Source address of the signal input (Only X0~X3 are available) **S_1**: Sample time setting and the input speed information **D**: Output proportion setting and output speed information

Description:

1. When **S** specifies X0, the ELC only uses the X0 input point and its associated high speed pulse output: Y0. When **S** specifies X1, the ELC uses X0 (A phase) and X1 (B phase) input points and their associated outputs: Y0 (Pulse) / Y1 (Dir). When **S** specifies X2, the ELC only uses the X2 input point and its associated high speed pulse output: Y2. When **S** specifies X3, the ELC uses X2 (A phase) and X3 (B phase) input points and their associated outputs: Y2 (Pulse) / Y3 (Dir). ELC2-PV only supported X0 and X1 inputs and their associated Y0/Y1 output.
2. The execution of CSFO requires a hardware high speed counter function as well as the high speed output function. Therefore, when program scans the CSFO instruction with high speed counter input points (X0, X1) or (X2, X3) enabled by with the DCNT instruction, or high speed pulse outputs (Y0, Y1) or (Y2, Y3) enabled by some other high speed output instructions, the CSFO instruction will not be activated.
3. If **S** specifies X1 / X3 with 2-phase 2 inputs, the counting mode is fixed at 4-times frequency.
4. During the pulse output process of Y0 or Y2, special registers (D1031, D1330 / D1337, D1336) for storing the current number of output pulses will be updated when program scans the instruction.
5. **S_1** occupies 4 consecutive 16-bit registers. **S_1** +0 specifies the sample time, i.e. when **S_1** +0 uses K1, the ELC catches the speed every time when 1 pulse is sent. Valid range for **S_1** +0 in 1-phase 1-input mode: K1~K100, and in 2-phase 2-input mode: K2~K100. If the specified value exceeds the valid range, the ELC will take the lower/upper limit value as the set value. Sample time can be changed during ELC operation, however the modified value will not take effect until the instruction is scanned. **S_1**+1 indicates the latest speed sampled by the ELC (Read-only). Units: 1Hz. Valid range: ±10kHz. **S_1**+2 and **S_1**+3 indicate the accumulated value of pulses as a 32-bit value (Read-only).

6. S_1 +0 specifies the sample time. The value of the sample time is recommended to be larger when the input speed increases, so as to achieve a higher accuracy for trapping the speed. For example, set S_1 +0 to K1 for the speed range 1Hz~1KHz, K10 for the speed range 10Hz~10KHz, K100 for the speed range 100Hz~10KHz.
7. **D** occupies 3 consecutive 16-bit registers. **D** +0 specifies the output proportional value. Valid range: K1 (1%) ~ K10000 (10000%). If the specified value exceeds the valid range, the ELC will take the lower/upper bound value as the set value. Output proportion can be changed during ELC operation, however the modified value will not take effect until the next time the program scans the instruction. **D**+2 and **D**+1 indicates the output speed as a 32-bit value. Units: 1Hz. Valid range: ±100kHz.
8. The speed sampled by the ELC will be multiplied with the output proportional value **D**+0, then the ELC will generate the actual output speed. The ELC will take the integer of the calculated value, i.e. if the calculated result is smaller than 1Hz, the ELC will output 0Hz. For example, input speed: 10Hz, output proportion: K5 (5%), then the calculation result will be 10 x 0.05 = 0.5Hz. The Pulse output will be 0Hz; if output proportion is changed to K15 (15%), then the calculation result will be 10 x 0.15 = 1.5Hz. Pulse output will be 1Hz.

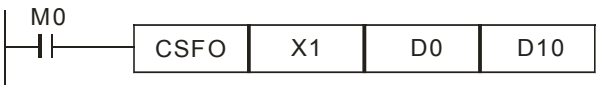
Program Example:

1. If D0 is set to K2 and D10 is set to K100:

   When the sampled speed is on (X0, X1) is +10Hz (D1 = K10), (Y0, Y1) pulses will be sent at +10Hz (D12, D11 = K10); When the sampled speed is -10Hz (D1 = K-10), (Y0, Y1) pulses will be sent at -10Hz (D12, D11 = K-10)
2. If D0 is set to K2 and D10 is set to K1000:

   When the sampled speed is on (X0, X1) is +10Hz (D1 = K10), (Y0, Y1) will output pulses with +100Hz (D12, D11 = K100); When the sampled speed is -100Hz (D1 = K-100), (Y0, Y1) pulses will be sent at -100Hz (D12, D11 = K-100)
3. If D0 is set to K10 and D10 is set to K10:

   When the sampled speed is on (X0, X1) is +10Hz (D1 = K10), (Y0, Y1) will output pulses with +1Hz (D12, D11 = K1); When the sampled speed is -10Hz (D1 = K-10), (Y0, Y1) pulses will be sent at -1Hz (D12, D11 = K-1)



API	Mnemonic		Operands	Function
215~217	D	LD#	S_1, S_2	Logic Contact Operation

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	LD#: 5 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	DLD#: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Data source value 1 S_2: Data source value 2

Description:

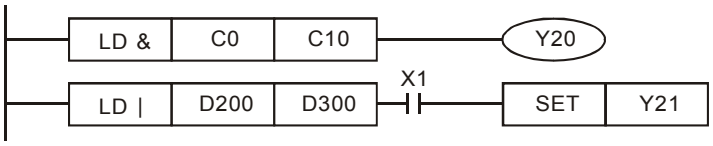
30. Compare the contents of S_1 and S_2. For example, "LD&" a logical AND of S_1 and S_2, if the comparison result is NOT 0, the instruction is true, and if it is 0, the instruction is false.
31. This instruction must be the first instruction of a rung.

API No.	16 -bit instruction	32 -bit instruction	Instruction is true	Instruction is false
215	LD&	DLD&	S_1 & $S_2 \neq 0$	S_1 & $S_2=0$
216	LD|	DLD|	S_1 | $S_2 \neq 0$	S_1 | $S_2=0$
217	LD^	DLD^	S_1 ^ $S_2 \neq 0$	S_1 ^ $S_2=0$

32. Operators: & : Logical "AND" operation, | : Logical "OR" operation, ^ : Logical "XOR" operation
33. If a 32-bit length counter is used with this instruction, be sure to use the 32-bit instruction (DLD#). If the 16-bit instruction (LD#) is used with a 32-bit counter, a "Program Error", will occur and the red "ERROR" indicator on the ELC will blink.

Program Example:

34. The LD& (Logical "AND" operation) instruction is used to compare the contents of C0 and C10. If the result is not equal to 0, Y20=ON.
35. Tthe LD| (Logical "OR" operation) instruction is used to compare the contents of D200 and D300. If the result is not equal to 0 and X1=ON, set Y21=ON.



API	Mnemonic		Operands	Function
218~220	D	AND#	S_1, S_2	Series Logic Contact Operation

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	AND#: 5 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	DAND#: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Data source value 1 S_2: Data source value 2

Description:

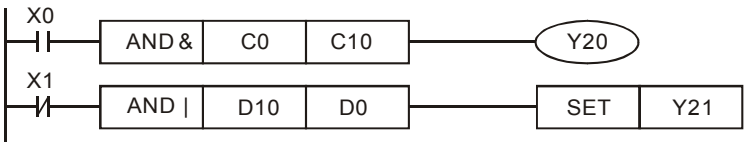
36. Compare the contents of S_1 and of S_2. For example, “AND&” compares S_1 and of S_2 , and if the comparison result is NOT 0, the instruction is true, and if it is 0, the instruction is false.
37. The AND# instruction is an input instruction.

API No.	16 -bit instruction	32 -bit instruction	Instruction is true	Instruction is false
218	AND&	DAND&	S_1 & $S_2 \neq 0$	S_1 & $S_2=0$
219	AND|	DAND|	S_1 | $S_2 \neq 0$	S_1 | $S_2=0$
220	AND^	DAND^	S_1 ^ $S_2 \neq 0$	S_1 ^ $S_2=0$

38. Operators:
    & : Logical “AND” operation, | : Logical “OR” operation, ^ : Logical “XOR” operation
39. If a 32-bit counter is used with this instruction, be sure to use the 32-bit instruction (DAND#). If the 16-bit instruction (AND#) is used for a 32-bit counter, a “Program Error”, will occur and the red “ERROR” indicator on the ELC panel will blink.

Program Example:

40. When X0=ON, using the AND& (Logical “AND” operation) instruction to compare the contents of C0 and C10. If the result is not equal to 0, Y20=ON.
41. When X1=OFF, using the AND| (Logical “OR” operation) instruction to compare the contents of D10 and D0. If the result is not equal to 0, set Y21=ON.



API	Mnemonic	Operands	Function
221~223	D **OR#**	S_1, S_2	Parallel Logic Contact Operation

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	OR#: 5 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	DOR#: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Data source value 1 S_2: Data source value 2

Description:

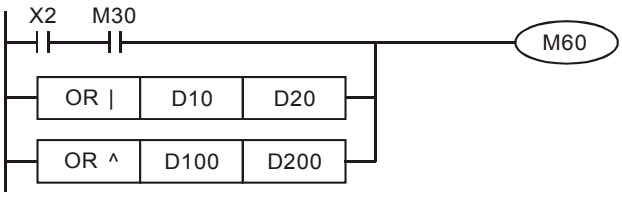
42. Compare the contents of S_1 and S_2. For example"OR&" compares S_1 and S_2, if the comparison result is NOT 0, the instruction is true, and if it is 0, the instruction is false.
43. The OR# instruction is an input instruction.

API No.	16 -bit instruction	32 -bit instruction	Instruction is true	Instruction is false
221	OR&	DOR&	S_1 & $S_2 \neq 0$	S_1 & $S_2=0$
222	OR|	DOR|	S_1 | $S_2 \neq 0$	S_1 | $S_2=0$
223	OR^	DOR^	S_1 ^ $S_2 \neq 0$	S_1 ^ $S_2=0$

44. Operators:
    & : Logical "AND" operation, | : Logical "OR" operation, ^ : Logical "XOR" operation
45. If a 32-bit counter is used with this instruction for comparison, be sure to use the 32-bit instruction (DOR#). If the 16-bit instruction (OR#) for a 32-bit counter is used, a "Program Error", will occur and the red "ERROR" indicator on the ELC panel will blink.

Program Example:

If both X2 and M30 are "ON", or when using the OR| (Logic "OR" operation) instruction to compare the contents of D10 and D20 and the result is not equal to 0, or when using the OR^ (Logic "XOR" operation) instruction to compare the contents of D100 and D200 and the result is not equal to 0, M60=ON.



API	Mnemonic	Operands	Function
224~230	D **LD***	S_1, S_2	Contact Comparison

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	LD*: 5 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	DLD*: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

3

Operands:

S_1: Data source value 1 S_2: Data source value 2

Description:

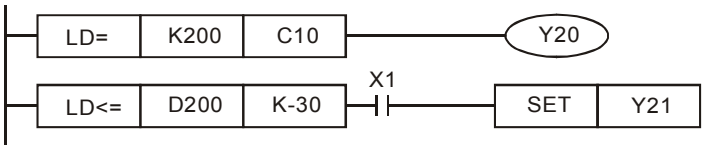
46. Compare the S1 and S_2. For example, “LD=” compares S_1 and $S_{2,\ and}$ if the comparison result is “=”, the instruction is true. If they are not “=” the instruction is false.
47. This instruction must be the first instruction of a rung.

API No.	16 -bit instruction	32 -bit instruction	Instruction is true	Instruction is false
224	LD=	DLD=	$S_1=S_2$	$S_1\neq S_2$
225	LD>	DLD>	$S_1>S_2$	$S_1\leqq S_2$
226	LD<	DLD<	$S_1<S_2$	$S_1\geqq S_2$
228	LD<>	DLD<>	$S_1\neq S_2$	$S_1=S_2$
229	LD<=	DLD<=	$S_1\leqq S_2$	$S_1>S_2$
230	LD>=	DLD>=	$S_1\geqq S_2$	$S_1<S_2$

48. If a 32-bit counter is used with this instruction for comparison, be sure to use the 32-bit instruction (DLD*). If the 16-bit instruction (LD*) is used with a 32-bit counter, a “Program Error”, will occur and the red “ERROR” indicator on the ELC panel will blink.

Program Example:

49. If the accumulated value of counter C10 is equal to K200, Y20=ON.
50. When the content of D200 is less than or equal to K－30, and X1=ON, set Y21=ON.



API	Mnemonic		Operands	Function
232~238	D	AND*	S_1, S_2	Series Contact Comparison

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	AND*: 5 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	DAND*: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Data source value 1 S_2: Data source value 2

Description:

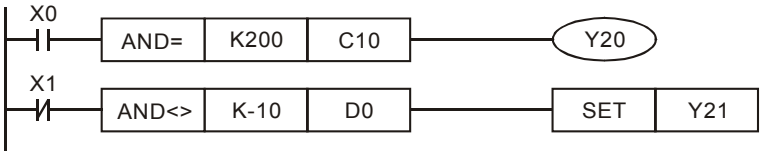
51. Compare the contents of S_1 and of S_2. For example, “AND=”,compares S_1 and of S_2 and if the comparison result is “=” , the instruction is true, and if it is “≠” , the instruction is false.
52. The AND* instruction is an input instruction

API No.	16 -bit instruction	32 -bit instruction	Instruction is true	Instruction is false
232	AND=	DAND=	$S_1=S_2$	$S_1\neq S_2$
233	AND>	DAND>	$S_1>S_2$	$S_1\leq S_2$
234	AND<	DAND<	$S_1<S_2$	$S_1\geq S_2$
236	AND<>	DAND<>	$S_1\neq S_2$	$S_1=S_2$
237	AND<=	DAND<=	$S_1\leq S_2$	$S_1>S_2$
238	AND>=	DAND>=	$S_1\geq S_2$	$S_1<S_2$

53. If a 32-bit counter is used with this instruction for comparison, be sure to use the 32-bit instruction (DAND*). If the 16-bit instruction (AND*) is used with a 32-bit counter, a “Program Error”, will occur and the red “ERROR” indicator on the ELC panel will blink.

Program Example:

54. If X0=ON and the accumulated value of counter C10 equals K200, Y20=ON.
55. If X1=OFF and the contents of register D0 is not equal to K－10, set Y21=ON.



API	Mnemonic		Operands	Function
240~246	D	OR*	S_1, S_2	Parallel Contact Comparison

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	OR*: 5 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	DOR*: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Data source value 1 S_2: Data source value 2

Description:

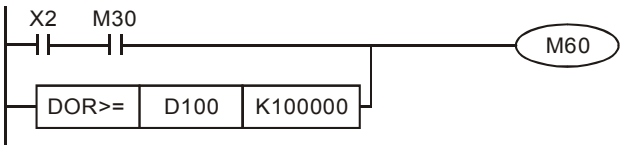
56. Compare the contents of S_1 and of S_2. For example "OR=" compares S_1 and of S_2 ,and if the comparison result is "=", the instruction is true, and if it is "≠", the instruction is false.
57. The OR*instruction is an input instruction

API No.	16-bit instruction	32-bit instruction	Instruction is true	Instruction is false
240	OR=	DOR=	$S_1=S_2$	$S_1\neq S_2$
241	OR>	DOR>	$S_1>S_2$	$S_1\leqq S_2$
242	OR<	DOR<	$S_1<S_2$	$S_1\geqq S_2$
244	OR<>	DOR<>	$S_1\neq S_2$	$S_1=S_2$
245	OR<=	DOR<=	$S_1\leqq S_2$	$S_1>S_2$
246	OR>=	DOR>=	$S_1\geqq S_2$	$S_1<S_2$

58. When the left most bit, MSB (the 16-bit instruction: b15, the 32-bit instruction: b31), from S_1 and S_2 is 1, this comparison value will be viewed as a negative value for comparison.
59. If the 32-bit length counter (C200~) is used with this instruction for comparison, be sure to use the 32-bit instruction (DOR*). If the 16-bit instruction (OR*) is used, a "Program Error", will occur and the red "ERROR" indicator on the ELC panel will blink.

Program Example:

If both X2 and M30 are "ON", or if the contents of the 32-bit registers D101 and D100 are greater or equal to K100,000, M60=ON.



API	Mnemonic	Operands	Function
258	ATMR	S_1, S_2	Contact type timer

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*					ATMR: 5 steps
S_2					*	*							*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Timer number (T0~T255) **S_2**: Setting value。

Description:

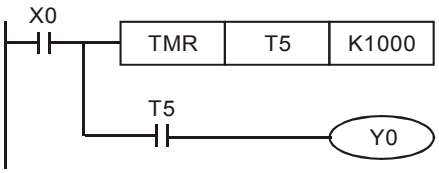
60. **S_2**: Setting value is K0~K32767
61. In ELCM-PH/PA, ATMR instruction supports ELCM-PH/PA V2.0(above).
62. When the instruction ATMR is executed, the coil of the timer specified is driven. When the timer value is equal to the setting value, the state of the normally-open contact is On, and the normally-closed contact is Off.

Normally-open contact	On
Normally-closed contact	Off

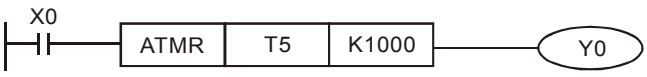
Program Example:

When the normally-open contact X0 is On, the timer T5 begins to measure time intervals. If the timer value is larger than or equal to K1000, the normally-open contact Y0 will be On.

Ladder diagram (The instruction TMR is used.)



Ladder diagram (The instruction ATMR is used.)



API	Mnemonic		Operands	Function
266	D	BOUT	**D, n**	Output Specified Bit of a Word

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*			BOUT: 5 steps
n					*	*	*	*	*	*	*	*	*	*	*	DBOUT: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

D: Destination output address **n**: Address specifying the output bit

Description:

63. Available range for the value in operand **n**: K0~K15 for the 16-bit instruction; K0~K31 for the 32-bit instruction.
64. The BOUT instruction turns an output on/off based on the value in **n.**
65. In ELC-PV, BOUT instruction support ELC-PV V1.6(above).

Status of Coils and Associated Contacts:

Evaluation result	BOUT instruction		
	Coil	Associated Contacts	
		NO contact（normally open）	NC contact（normally closed）
FALSE	OFF	Current blocked	Current flows
TRUE	ON	Current flows	Current blocked

Program Example:


```
Instruction:        Operation:
LDI   X0            ; Load NC contact X0
AND   X1            ; Connect NO contact
                      X1 in series.
BOUT  K4Y0  D0      ; When D0 = k1,
                      executes output on Y1
                    ; When D0 = k2,
                      executes output on Y2
```

In the program above, the outputs Y0-Y7 and Y10-Y17 are turned on and off based on the value in D0. If D0=0, Y0 turns on. If D0=1, output Y1 turns on and so on.

API	Mnemonic		Operands	Function
267	D	BSET	**D, n**	Set ON Specified Bit of a Word

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*			BSET: 5 steps
n					*	*	*	*	*	*	*	*	*	*	*	DBSET: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

D: Destination address to be Set ON **n**: Value specifying the bit to be Set ON

Description:

66. Available range for the value in operand **n**: K0~K15 for the 16-bit instruction; K0~K31 for the 32-bit instruction.
67. In ELC-PV, BOUT instruction support ELC-PV V1.6(above).
68. When the BSET instruction executes, the output device specified by operand **n** will be turned ON and latched. To reset the ON state of the device, BRST instruction is required.

Program Example:


```
Instruction:        Operation:
LDI   X0            ; Load NC contact X0
AND   X1            ; Connect NO contact
                      X1 in series.
BSET  K4Y0  D0      ; When D0 = k1,
                      Y1 is ON and latched
                    ; When D0 = k2,
                      Y2 = ON and latched
```

API	Mnemonic		Operands	Function
268	D	BRST	**D, n**	Reset Specified Bit of a Word

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*			BRST: 5 steps
n					*	*	*	*	*	*	*	*	*	*	*	DBRST: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

D: Destination address to be reset **n:** Value specifying the bit to be reset

Description:

69. Available range for the value in operand **n**: K0~K15 for the 16-bit instruction; K0~K31 for the 32-bit instruction.
70. When the BRST instruction executes, the output device specified by operand **n** will be reset (OFF).
71. In ELC-PV, BOUT instruction support ELC-PV V1.6(above).

Program Example:

Instruction: Operation:

LD X0 ; Load NO contact X0

BRST K4Y0 D0 ; When D0 = k1, Y1 is reset

; When D0 = k2, Y2 is reset

API	Mnemonic	Operands	Function
269	D BLD	**S, n**	Load NO Contact by Specified Bit

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			BLD: 5 steps
n					*	*	*	*	*	*	*	*	*	*	*	DBLD: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

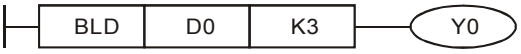
Operands:

S: Reference source address **n**: Reference bit address

Description:

72. Available range for the value in operand **n**: K0~K15 for the 16-bit instruction; K0~K31 for the 32-bit instruction.
73. The BLD instruction is used to evaluate the state of a bit in D0 and if that bit is ON in the program below, the instruction will be true. If it's OFF, the instruction will be false. **n** specifies the bit in S. If bit 3 in D0 is a 1 in the program below, the instruction is true and output Y0 will be energized..
74. In ELC-PV, BOUT instruction support ELC-PV V1.6(above).

Program Example:



Instruction:			Operation:
BLD	**D0**	**K3**	; Load NO contact with bit status of bit3 in D0
OUT	Y0		; Drive coil Y0

API	Mnemonic		Operands	Function
270	D	BLDI	**S, n**	Load NC Contact by Specified Bit

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			BLDI: 5 steps
n					*	*	*	*	*	*	*	*	*	*	*	DBLDI: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

3

Operands:

S: Reference source address **n**: Reference bit address

Description:

75. Available range for the value in operand **n**: K0~K15 for the 16-bit instruction; K0~K31 for the 32-bit instruction.
76. The BLDI instruction is used to evaluate the state of a bit in D0 in the program below and if that bit is OFF, the instruction will be true. If it's ON, the instruction will be false. **n** specifies the bit in **S**. If bit 1 in D0 is a 0 in the program below, the instruction is true and output Y0 will be energized.
77. In ELC-PV, BOUT instruction support ELC-PV V1.6(above).

Program Example:


```
Instruction:        ; Operation:
BLDI  D0   K1       ; Load NC contact with bit
                      status of bit1 in D0
OUT   Y0            ; Drive coil Y0
```

API	Mnemonic		Operands	Function
271	D	BAND	**S, n**	Connect NO Contact in Series by Specified Bit

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			BLDI: 5 steps
n					*	*	*	*	*	*	*	*	*	*	*	DBLDI: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

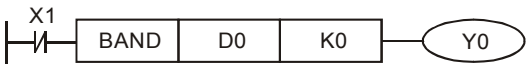
Operands:

S: Reference source address **n**: Reference bit address

Description:

78. Available range for the value in operand **n**: K0~K15 for the 16-bit instruction; K0~K31 for the 32-bit instruction.
79. The BAND instruction is used to connect NO contacts in series, whose state is defined by the reference bit **n** in reference device **S**, i.e. if the bit specified by **n** is ON, the instruction will be true and if X1=0, Y0 will be energized in the program below.
80. In ELC-PV, BOUT instruction support ELC-PV V1.6(above).

Program Example:




```
Instruction:          Operation:
LDI    X1             ; Load NC contact X1
BAND   D0   K0        ; Connect NO contact in
                        series, whose state is
                        defined by bit0 of D0
OUT    Y0             ; Drive coil Y0
```

API	Mnemonic		Operands	Function
272	D	BANI	**S, n**	Connect NC Contact in Series by Specified Bit

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			BANI: 5 steps
n					*	*	*	*	*	*	*	*	*	*	*	DBANI: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

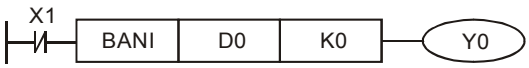
Operands:

S: Reference source address **n**: Reference bit address

Description:

81. Available range for the value in operand **n**: K0~K15 for 16-bit instruction; K0~K31 for 32-bit instruction.
82. The BANI instruction is used to connect NC contacts in series, whose state is defined by the reference bit **n** in reference device **S**, i.e. if the bit specified by **n in D** is OFF, the instruction will be true and if X1=0, Y0 will be energized.
83. In ELC-PV, BOUT instruction support ELC-PV V1.6(above).

Program Example:



Instruction:			Operation:
LDI	X1		; Load NC contact X1
BANI	**D0**	**K0**	; Connect NC contact in series, whose state is defined by bit0 of D0
OUT	Y0		; Drive coil Y0

API	Mnemonic	Operands	Function
273	D BOR	**S, n**	Connect NO Contact in Parallel by Specified Bit

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			BOR: 5 steps
n					*	*	*	*	*	*	*	*	*	*	*	DBOR: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

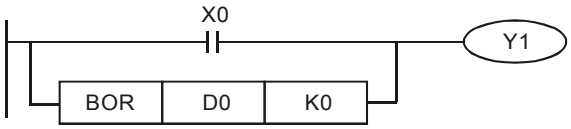
Operands:

S: Reference source address **n:** Reference bit address

Description:

84. Available range for the value in operand **n**: K0~K15 for the 16-bit instruction; K0~K31 for the 32-bit instruction.
85. The BOR instruction is used to connect NO contacts in parallel, whose state is defined by the reference bit **n** in reference device **S**, i.e. if the bit specified by **n in S** is ON, the instruction will be true and in the program below, Y1 will be energized.
86. In ELC-PV, BOUT instruction support ELC-PV V1.6(above).

Program Example:




```
Instruction:        Operation:
LD    X0            ; Load NO contact X0
BOR   D0   K0       ; Connect NO contact in
                      parallel, whose state
                      is defined by bit0 of
                      D0
OUT   Y1            ; Drive coil Y1
```

API	Mnemonic		Operands	Function
274	D	BORI	**S, n**	Connect NC Contact in Parallel by Specified Bit

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			BORI: 5 steps
n					*	*	*	*	*	*	*	*	*	*	*	DBORI: 9 steps

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

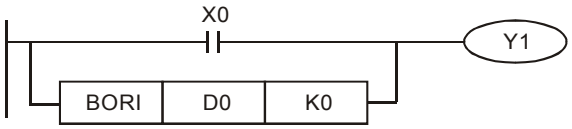
Operands:

S: Reference source address **n:** Reference bit address

Description:

87. Available range for the value in operand **n**: K0~K15 for the 16-bit instruction; K0~K31 for the 32-bit instruction.
88. The BORI instruction is used to connect NC contacts in parallel, whose state is defined by the reference bit **n** in reference device **S**, i.e. if the bit specified by **n** is OFF, the instruction will be true and in the program below output Y1 will be energized.
89. In ELC-PV, BOUT instruction support ELC-PV V1.6(above).

Program Example:



Instruction:			Operation:
LD	X0		; Load NO contact X0
BORI	**D0**	**K0**	; Connect NC contact in parallel, whose state is defined by bit 0 of D0
OUT	Y1		; Drive coil Y1

API	Mnemonic	Operands	Function
275~280	**FLD***	S_1, S_2	Floating Point Number Contact Comparison

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			FLD*: 9 steps
S_2											*	*	*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Data source value 1 S_2: Data source value 2

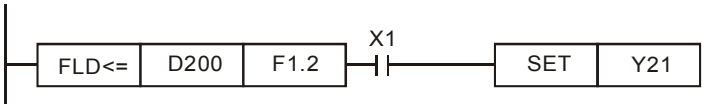
Description:

90. Compare the S1 and S_2. For example, "FLD=" compares S_1 and $S_{2,\ and}$ if the comparison result is "=", the instruction is true. If they are not "=" the instruction is false.
91. The FLD* operands, can directly insert floating point numbers into S_1 and S_2 can each be 2 T, C, D or actual floating point values (e.g. F1.2).
92. In ELC-PV, BOUT instruction support ELC-PV V1.6(above).
93. This instruction must be the first instruction of a rung.

API No.	32 -bit instruction	Instruction is true	Instruction is false
275	FLD=	$S_1=S_2$	$S_1\neq S_2$
276	FLD>	$S_1>S_2$	$S_1\leqq S_2$
277	FLD<	$S_1<S_2$	$S_1\geqq S_2$
278	FLD<>	$S_1\neq S_2$	$S_1=S_2$
279	FLD<=	$S_1\leqq S_2$	$S_1>S_2$
280	FLD>=	$S_1\geqq S_2$	$S_1<S_2$

Program Example:

When the content of floating point numbers D201, D200 is less than or equal to F1.2, and X1=ON, set Y21=ON.



API	Mnemonic	Operands	Function
281~286	FAND*	S_1, S_2	Floating Point Number Series Contact Comparison

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			FAND*: 9 steps
S_2											*	*	*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

3

Operands:

S_1: Data source value 1 S_2: Data source value 2

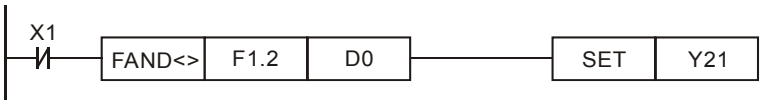
Description:

94. Compare the contents of S_1 and of S_2. For example, “FAND=”,compares S_1 and of S_2 and if the comparison result is “=” , the instruction is true, and if it is “≠” , the instruction is false.
95. The FAND* operands, can directly insert floating point numbers into S_1 and S_2 can each be 2 T, C, D or actual floating point values (e.g. F1.2).
96. In ELC-PV, BOUT instruction support ELC-PV V1.6(above).
97. The FAND* instruction is an input instruction

API No.	32 -bit instruction	Instruction is true	Instruction is false
232	FAND＝	$S_1=S_2$	$S_1\neq S_2$
233	FAND＞	$S_1>S_2$	$S_1\leqq S_2$
234	FAND＜	$S_1<S_2$	$S_1\geqq S_2$
236	FAND＜＞	$S_1\neq S_2$	$S_1=S_2$
237	FAND＜＝	$S_1\leqq S_2$	$S_1>S_2$
238	FAND＞＝	$S_1\geqq S_2$	$S_1<S_2$

Program Example:

If X1=OFF and the contents of floating point numbers D1, D0 is not equal to F1.2, set Y21=ON.



API	Mnemonic	Operands	Function
287~292	FOR*	S_1, S_2	Floating Point Number Parallel Contact Comparison

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			FOR*: 9 steps
S_2											*	*	*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Data source value 1 S_2: Data source value 2

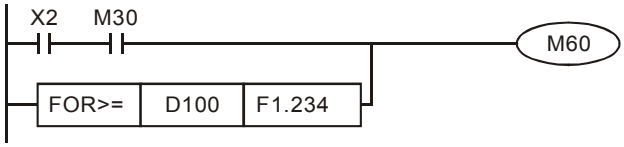
Description:

98. Compare the contents of S_1 and of S_2. For example "FOR=" compares S_1 and of S_2 ,and if the comparison result is "=", the instruction is true, and if it is "≠", the instruction is false.
99. The FOR* operands, can directly insert floating point numbers into S_1 and S_2 can each be 2 T, C, D or actual floating point values (e.g. F1.2)
100. In ELC-PV, BOUT instruction support ELC-PV V1.6(above)..
101. The FOR*instruction is an input instruction

API No.	32-bit instruction	Instruction is true	Instruction is false
287	FOR=	$S_1=S_2$	$S_1\neq S_2$
288	FOR>	$S_1>S_2$	$S_1\leqq S_2$
289	FOR<	$S_1<S_2$	$S_1\geqq S_2$
290	FOR<>	$S_1\neq S_2$	$S_1=S_2$
291	FOR<=	$S_1\leqq S_2$	$S_1>S_2$
292	FOR>=	$S_1\geqq S_2$	$S_1<S_2$

Program Example:

If both X2 and M30 are "ON", or if the contents of the floating point numbers D101, D100 are greater or equal to F1.234, M60=ON.



API	Mnemonic		Operands	Function
296~301	D	LDZ※	S_1, S_2, S_3	Comparing contact type absolute values LDZ※

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*			LDZ※: 7 steps
S_2					*	*	*	*	*	*	*	*	*			DLDZ※: 13 steps
					*	*	*	*	*	*	*	*	*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Data source value 1 S_2: Data source value 2 S_3: Data source value 3

Description:

102. In ELCM-PH/PA, LDZ※ supported ELCM-PH/PAV2.0(above).
103. The absolute value of the difference between S_1 and S_2 is compared with the absolute value of S_3. Take LDZ= for example. If the comparison result is that the absolute value of the difference between S_1 and S_2 is equal to the absolute value of S_3, the condition of the instruction is met. If the comparison result is that the absolute value of the difference between S_1 and S_2 is not equal to the absolute value of S_3, the condition of the instruction is not met.
104. The instruction can be connected to a busbar.

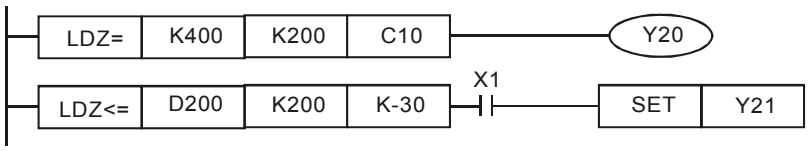
API No.	16-bit instruction	32-bit instruction	Comparison result	
			On	Off
296	LDZ＞	DLDZ＞	$\|S_1 - S_2\| > \|S_3\|$	$\|S_1 - S_2\| \leqq \|S_3\|$
297	LDZ＞＝	DLDZ＞＝	$\|S_1 - S_2\| \geqq \|S_3\|$	$\|S_1 - S_2\| < \|S_3\|$
298	LDZ＝	DLDZ＜	$\|S_1 - S_2\| < \|S_3\|$	$\|S_1 - S_2\| \geqq \|S_3\|$
299	LDZ＜＝	DLDZ＜＝	$\|S_1 - S_2\| \leqq \|S_3\|$	$\|S_1 - S_2\| > \|S_3\|$
300	LDZ＝	DLDZ＝	$\|S_1 - S_2\| = \|S_3\|$	$\|S_1 - S_2\| \neq \|S_3\|$
301	LDZ＜＞	DLDZ＜＞	$\|S_1 - S_2\| \neq \|S_3\|$	$\|S_1 - S_2\| = \|S_3\|$

105. If the values of the most significant bits in S_1, S_2, and S_3 are 1, the values in S_1, S_2, and S_3 are negative values.
106. A 32-bit counter (C200~) must be used with the 32-bit instruction DLDZ※. If it is used with the 16-bit instruction LDZ※, a program error will occur, and the ERROR LED indicator on the PLC will blink.

Program Example:

107. If the value in C10 is equal to K200 or K-200, Y20 will be On.

108. If the value in D200 is less than or equal to K230, and is larger than or equal to K170, and X1 is On, Y21 will be On and latched.



API	Mnemonic		Operands	Function
302~307	D	ANDZ※	S_1, S_2, S_3	Comparing contact type absolute values ANDZ※

Type / OP	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*			ANDZ※: 7 steps
S_2					*	*	*	*	*	*	*	*	*			DANDZ※: 13 steps
					*	*	*	*	*	*	*	*	*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Data source value 1 S_2: Data source value 2 S_3: Data source value 3

Description:

109. In ELCM-PH/PA, ANDZ※supported ELCM-PH/PAV2.0(above).
110. The absolute value of the difference between S_1 and S_2 is compared with the absolute value of S_3. Take AND= for example. If the comparison result is that the absolute value of the difference between S_1 and S_2 is equal to the absolute value of S_3, the condition of the instruction is met. If the comparison result is that the absolute value of the difference between S_1 and S_2 is not equal to the absolute value of S_3, the condition of the instruction is not met.
111. The instruction ANDZ※ is connected to a contact in series.

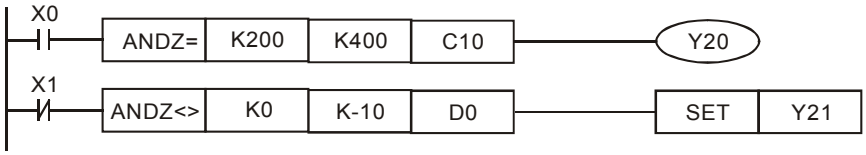
API No.	16-bit instruction	32-bit instruction	Comparison result	
			On	Off
302	ANDZ＞	DANDZ＞	$\lvert S_1 - S_2 \rvert > \lvert S_3 \rvert$	$\lvert S_1 - S_2 \rvert \leqq \lvert S_3 \rvert$
303	ANDZ＞＝	DANDZ＞＝	$\lvert S_1 - S_2 \rvert \geqq \lvert S_3 \rvert$	$\lvert S_1 - S_2 \rvert < \lvert S_3 \rvert$
304	ANDZ＜	DANDZ＜	$\lvert S_1 - S_2 \rvert < \lvert S_3 \rvert$	$\lvert S_1 - S_2 \rvert \geqq \lvert S_3 \rvert$
305	ANDZ＜＝	DANDZ＜＝	$\lvert S_1 - S_2 \rvert \leqq \lvert S_3 \rvert$	$\lvert S_1 - S_2 \rvert > \lvert S_3 \rvert$
306	ANDZ＝	DANDZ＝	$\lvert S_1 - S_2 \rvert = \lvert S_3 \rvert$	$\lvert S_1 - S_2 \rvert \neq \lvert S_3 \rvert$
307	ANDZ＜＞	DANDZ＜＞	$\lvert S_1 - S_2 \rvert \neq \lvert S_3 \rvert$	$\lvert S_1 - S_2 \rvert = \lvert S_3 \rvert$

112. If the values of the most significant bits in S_1, S_2, and S_3 are 1, the values in S_1, S_2, and S_3 are negative values.
113. A 32-bit counter (C200~) must be used with the 32-bit instruction DANDZ※. If it is used with the 16-bit instruction ANDZ※, a program error will occur, and the ERROR LED indicator on the PLC will blink.

Program Example:

114. If X0 is On, and the present value in C10 is equal to K200 or K-200, Y20 will be On.

115. If X1 is Off, and the value in D0 is not equal to K10 or K-10, Y21 will be On and latched.



API	Mnemonic		Operands	Function
308~313	D	ORZ※	S_1, S_2, S_3	Comparing contact type absolute values ORZ※

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*			ORZ※: 7 steps
S_2					*	*	*	*	*	*	*	*	*			DORZ※: 13 steps
					*	*	*	*	*	*	*	*	*			

ELCB			ELC						ELC2									ELCM		
PB			PA			PV			PB			PH/PA/PE			PV			PH/PA		
32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P	32	16	P

Operands:

S_1: Data source value 1 S_2: Data source value 2 S_3: Data source value 3

Description:

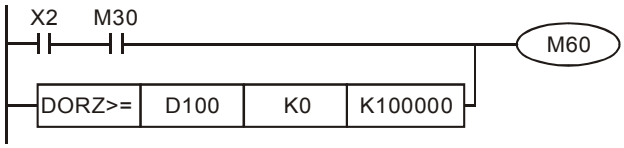
116. In ELCM-PH/PA, ANDZ※supported ELCM-PH/PAV2.0(above).
117. The absolute value of the difference between S_1 and S_2 is compared with the absolute value of S_3. Take ORZ= for example. If the comparison result is that the absolute value of the difference between S_1 and S_2 is equal to the absolute value of S_3, the condition of the instruction is met. If the comparison result is that the absolute value of the difference between S_1 and S_2 is not equal to the absolute value of S_3, the condition of the instruction is not met.
118. The instruction ANDZ※ is connected to a contact in parallel.

API No.	16-bit instruction	32-bit instruction	Comparison result	
			On	Off
308	ORZ＞	DORZ＞	$\|S_1 - S_2\| > \|S_3\|$	$\|S_1 - S_2\| \leqq \|S_3\|$
309	ORZ＞＝	DORZ＞＝	$\|S_1 - S_2\| \geqq \|S_3\|$	$\|S_1 - S_2\| < \|S_3\|$
310	ORZ＜	DORZ＜	$\|S_1 - S_2\| < \|S_3\|$	$\|S_1 - S_2\| \geqq \|S_3\|$
311	ORZ＜＝	DORZ＜＝	$\|S_1 - S_2\| \leqq \|S_3\|$	$\|S_1 - S_2\| > \|S_3\|$
312	ORZ＝	DORZ＝	$\|S_1 - S_2\| = \|S_3\|$	$\|S_1 - S_2\| \neq \|S_3\|$
313	ORZ＜＞	DORZ＜＞	$\|S_1 - S_2\| \neq \|S_3\|$	$\|S_1 - S_2\| = \|S_3\|$

119. If the values of the most significant bits in S_1, S_2, and S_3 are 1, the values in S_1, S_2, and S_3 are negative values.
120. A 32-bit counter (C200~) must be used with the 32-bit instruction DORZ※. If it is used with the 16-bit instruction ORZ※, a program error will occur, and the ERROR LED indicator on the PLC will blink.

Program Example:

If X2 and M30 are On, or the value in the 32-bit register (D101, D100) is larger than or equal to K100000, or is less than or equal to K-100000, M60 will be On.



MEMO

Sequential Function Chart

This chapter contains information in programming in SFC mode.

This Chapter Contains

4 Sequential Function Chart

4.1 Sequential Function Chart (SFC)

Sequential Function Charts are a graphical method of organizing a PLC program in which the entire structure of the program resembles a flow chart. The two primary components of an SFC are steps and transitions. A step is a group of logic used to accomplish a particular control task. Steps in an SFC are very similar to the steps in a flow chart. Transitions are the mechanisms used to decide which step should occur next and precisely when that step should become active. Transitions are like the arrows in a flow chart.
Each step and transition contains a so-called “inner ladder” which allows the programmer to create ladder logic that will run whenever that step or transition is active. While a given step is active its inner ladder will execute continuously until the condition(s) for the transition are met, at which point the next step(s) in the sequence will become active.
SFC programming is beneficial for use in applications which have a repeatable multi-step process or series of repeatable processes. Programmers who use the State-Machine method of programming will find SFCs to be a useful programming method. The visual nature of the SFC makes it easy to maintain. The active step(s) are highlighted, pointing out what logic is being executed. Due to the fact that the program is divided into steps, one “section” of the program can be viewed rather than viewing the entire program.

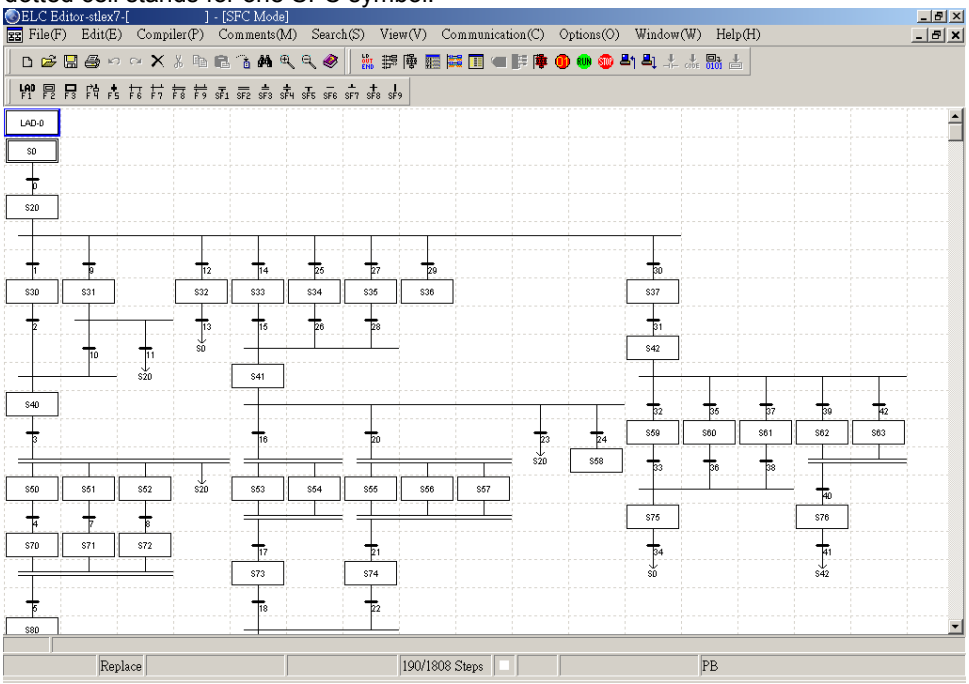
4.2 Basic Operation

■ SFC Toolbar and Icons Explanation:

Icons shown on SFC toolbar	Description of Icons	Explanation:
	Common ladder	Located in front of an initial SFC Step Point. Contains code that is run on every scan (outside of the Step procedure). No transition point is used.
	Initial diagram of SFC	Diagram for the Initial Step Point: double-framed diagram designates the start of a Step Procedure. Available devices are only S0~S9. Only one can be used per procedure.
	Common SFC	Diagram for the Common Step Point: Used for all other step points outside of the Initial Step Point. Available devices vary by processor.
	Jump diagram	Jump diagram: Used to move from one step point to another that is not sequenced in order or not in the same procedure.
	Condition diagram	Transition condition diagram: The condition used to transition to the next step.
	Divergence of condition diagram	Alternative divergence: Used to move from a single step point to one of multiple subsequent step points, each move having a different transition condition.
	Convergence of condition diagram	Alternative convergence: Used to move from multiple step points to the same single step point, each move having a different transition condition.
	Divergence diagram	Simultaneous divergence: Used to move from a single step point to multiple subsequent step points using a single transition condition. The subsequent step points will execute cyclically.
	Convergence diagram	Simultaneous convergence: Used to move from multiple step points to a single step point using a single transition condition.

■ SFC Editing Environment:
In SFC mode, up to 16 columns can be used for programming, whereas there is no limit to rows.

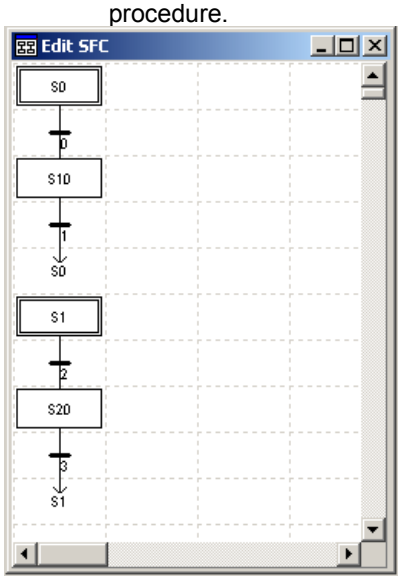
Every dotted cell stands for one SFC symbol.



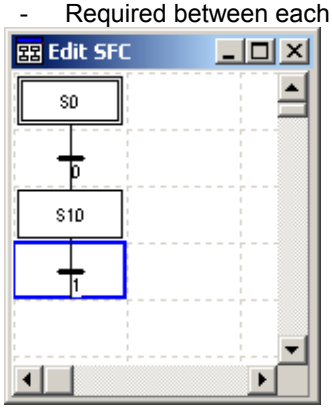
When programming in SFC, use the far left cell as the starting position and then work from left to right, top to bottom. Each object (LAD, Step Point, and Transition) contains its own ladder logic segment called **Inner Ladder**. The entire SFC diagram can be constructed first and then the inner ladder of each SFC symbol can be created, or each SFC symbol and its internal ladder logic can be developed simultaneously as the SFC structure diagram is constructed.

SFC Structure

1. LAD
   - Logic that is needed outside of SFC logic is put here.
   - Logic is executed every scan.
   - At a minimum logic to enable Initial Step Point must be located here.
2. Initial Step
   - The start of the procedure
   - Only one per procedure
   - Available addresses are S0 – S9
   - No transition condition is used between this and the LAD therefore logic must be included in LAD to turn on Initial Step addresses.
   - If more than one procedure is needed, additional Initial Steps can be added. Place the next Initial Step in first column, but after completion of previous procedure.



3. Transition
   - Condition to move between steps
   - Required between each step point



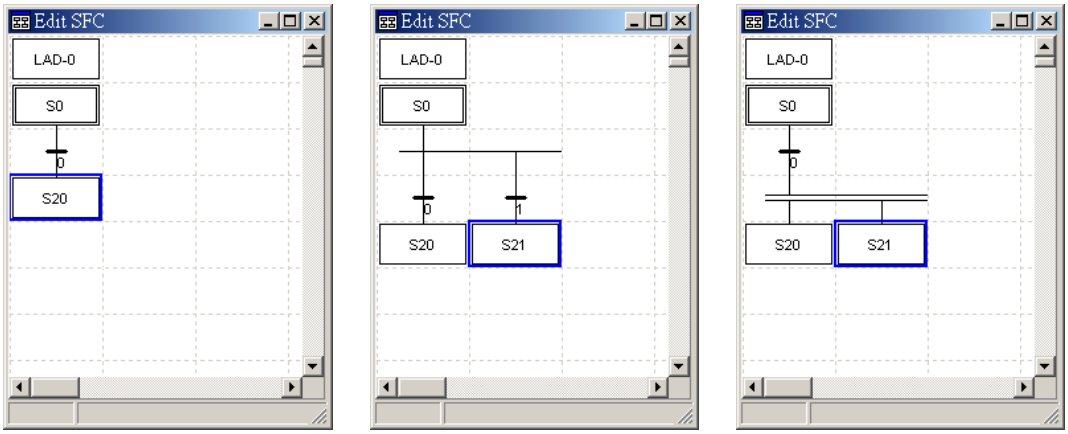
4. Common Step Point
   - These compose the bulk of the SFC
   - Step Points in a procedure other than the Initial Step Point
   - Ladder logic is contained within the Step Points.
   - Available addresses depend on the processor.
5. Convergence/Divergence
   - The ways to transition to and from multi-steps
   - Two types of each: Alternative and Simultaneous

- Alternative: Each step has a unique transition condition
- Simultaneous: All steps share one transition condition

1. **No divergence:**
   Move from S0 to S20 according to the transition condition 0.

2. **Alternative Divergence:**
   This is an OR condition. Move from S0 to *either* S20 *or* S21 by different transition conditions 0 and 1.

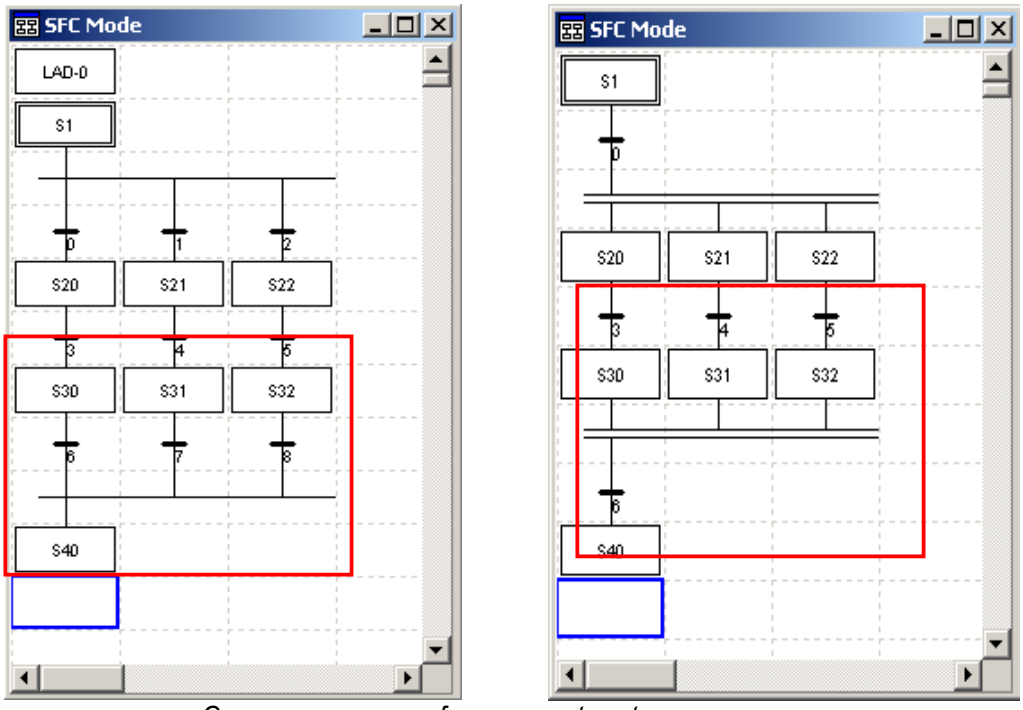
3. **Simultaneous Divergence:**
   This is an AND condition. Move from S0 to *both* S20 *and* S21 by the single transition condition 0.



Divergence moves from one step to many.

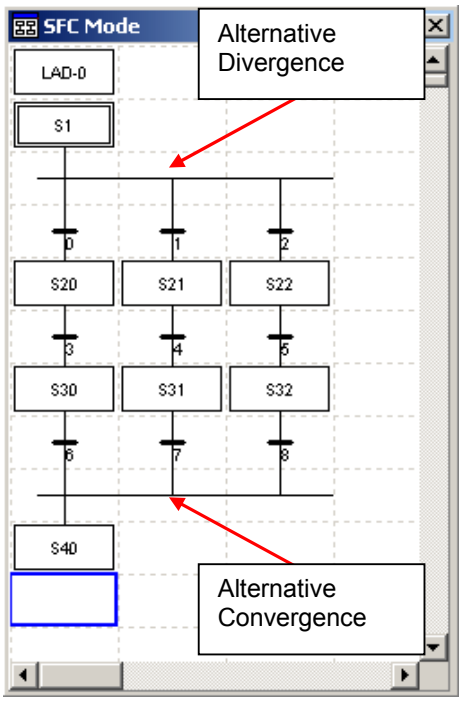
1. Alternative Convergence:
This is an OR condition. There is only one path. Move from S20 to S30 or S21 to S31 or S22 to S32 to the step point S40 when the appropriate transition (6 or 7 or 8) is true.

2. Simultaneous Convergence:
This is an AND condition. Move from S30, S31, and S32 to the step point S40 according to the same transition condition 6. S30, S31 and S32 must all be active before transitioning to S40.

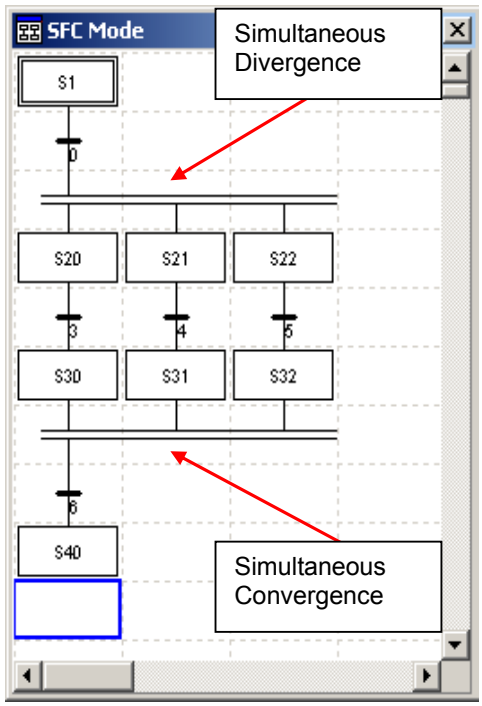


Convergence moves from many steps to one.

1. Alternative:
Move to or from multiple step points according to different transition conditions.

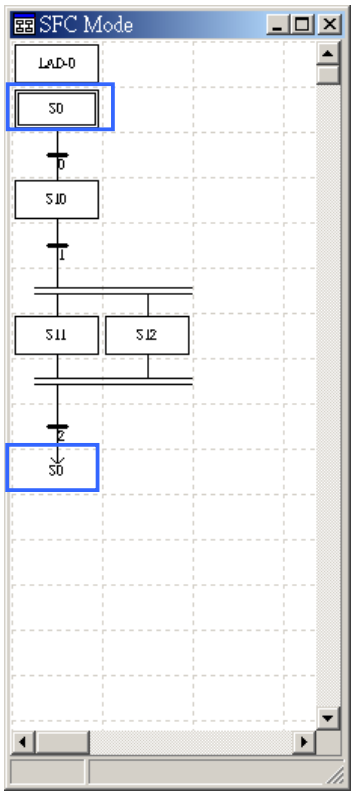


2. Simultaneous:
Move to or from multiple step points according to one transition condition.



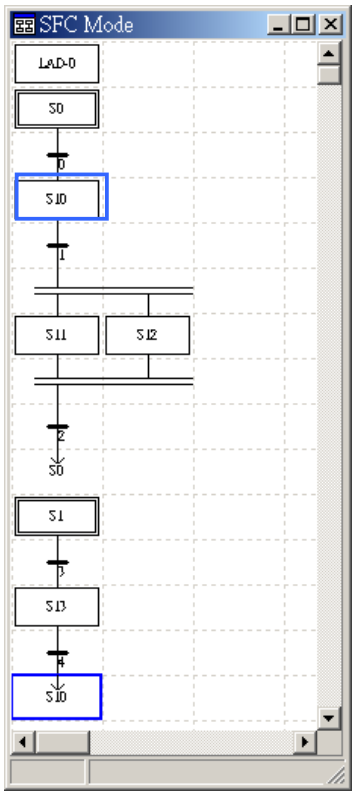
6. Jumps
   - Allows movement to step points outside the sequenced order or even to different program processes

1. The step points S10, S11 and S12 are sequenced in order, and therefore can be connected by using the common SFC diagram. However, to return back to S0 from S11 or S12, a Jump has to be used to complete the transition.



Return to the initial step point S0 in the same procedure.

2. To move from S20 within the procedure of S1 to S10 within the procedure of S0, a Jump has to be used to complete the transition.



Transition between SFC diagrams in two different procedures.

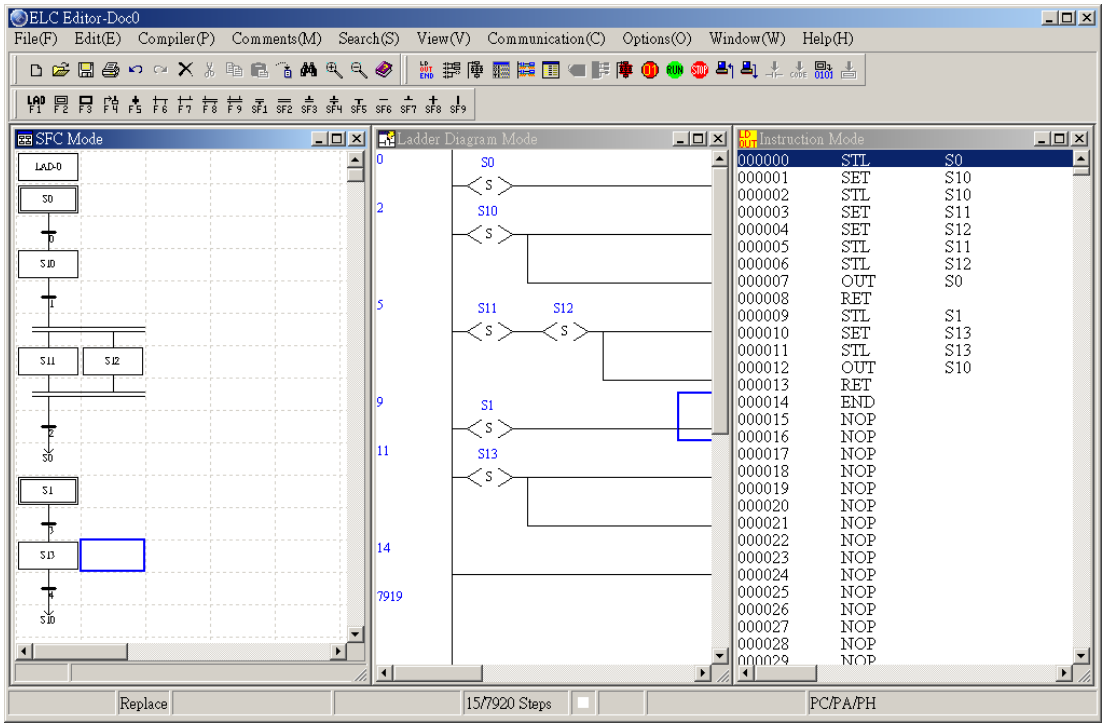
4.3 SFC Viewed as Ladder and Instruction List

■ Rules of SFC

1. Segment and Row comments cannot be used in SFC mode – only Device comments can be used.
2. Inner Ladder mode is used to develop ladder logic in SFC mode.
3. The device number of each step point (including the initial step point) must be unique.
4. Transition numbers are automatically assigned in sequential order.
5. A RET statement is generated in the Ladder logic of the last step of a procedure when the SFC is compiled.
6. SFC code compiles into Instruction List which can then be viewed in Ladder. If a change made in Ladder is to also be viewed in SFC, the Ladder must be compiled into Instruction List and then from Instruction List into SFC. It is recommended to make changes in Inner Ladder Mode.

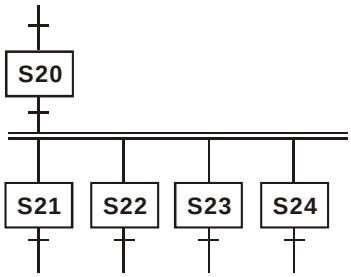
■ Graphical Examples

Example of the simultaneous divergence and simultaneous convergence procedures



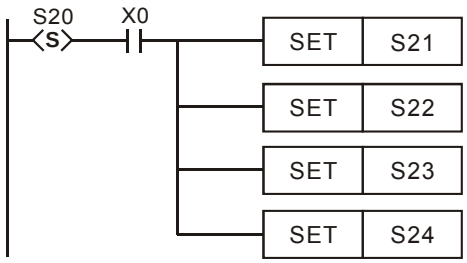
SFC LADDER INSTRUCTION

SFC

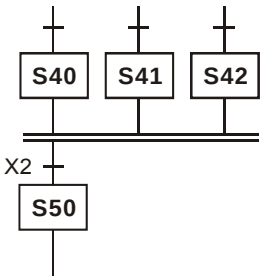


Alternative Divergence

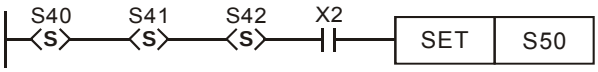
LADDER



Alternative Divergence



Simultaneous Convergence



Simultaneous Convergence

* This does not show the inner ladder logic of each step.

■ For additional information consult corresponding software manual.

MEMO

Setting and Using an Ethernet ELC/Module

This Chapter Contains

4.4 Specifications for an Ethernet ELC/Module

The specifications for an ELC series Ethernet port and the functions of an ELC series Ethernet port are listed below.

Specifications for an Ethernet interface:

Item	Specifications
Interface	RJ-45 with Auto MDI/MDIX
Number of ports	1 Port
Transmission method	IEEE802.3, IEEE802.3u
Transmission cable	Category 5e
Transmission rate	10/100 Mbps Auto-Defect
Protocol	ICMP, IP, TCP, UDP, DHCP, SMTP, NTP, MODBUS TCP

Ethernet functions:

Function Item	ELC2-PE Series	ELC-COENETM
MODBUS/TCP	Master & Slave	Master & Slave
Connection number of servers	16	16
Connection number of clients	8	16
Connection number of data exchanged	8	24
RTU mapping	-	4
E-mail	-	4
SNMP	-	2
IP filter	4	8

4.5 Ethernet Control Registers

In order to control and monitor Ethernet communication, users can read the data in the control registers listed below by means of the instruction FROM, and write data into the control registers listed below by means of the instruction TO. (Please refer to the explanation of API 78 and that of API 79 in chapter 3 for more information about FROM/TO.)

[Note 1] Please refer to ELC-COENETM Manual for more information about control registers.

[Note 2] The ELC2-PE Built-in Ethernet port, it always is like a 9th left side module. For example: read IP address(CR88, CR89) to D100/D101, the command will be “FROM K108 K88 D100 K2”.

[Note 3] The ELC-COENETM can be connected to left side of ELC. For example: it is 1st module at left side of ELC MPU module, and it read model name(CR0) to D0, the command will be “FROM K100 K0 D0 K1”

4.5.1 ELC2-PE SERIES ELC (ETHERNET ELC)

CR number		Attribute	Register name	Description
HW	LW			
#12	#0	-	Reserved	
	#13	R/W	Enabling the data exchange	Users can set CR#13 to “sending the data” or “not sending the data”.
	#14	R/W	Writing function of the RTU mapping	0: The ELC writes data continually. 1: The ELC writes data when the input changes.
	#15	R/W	Enabling flag for RTU mapping	1: Enable; 0: Disable. Default = 1
	#16	R/W	Connection status of RTU mapping slave	b0: Status of RTU slave 1 b1: Status of RTU slave 2 b2: Status of RTU slave 3 b3: Status of RTU slave 4
	#17	R/W	Execution cycle of the data exchange	Time unit: ms
	#18	-	Reserved	
	#19	R	States of the slaves involved in the data exchange	If the value of a bit is 1, an error occurs in the slave corresponding to the bit. b[0:7] indicate the states of the slaves 1~8 involved in the data exchange.
#86 ~ #20		-	Reserved	
	#87	R/W	IP address setting mode	0: Static IP 1: DHCP
#89	#88	R/W	IP address	When the IP address is 192.168.1.5, the data in CR#89 is 192.168, and the data in CR#88 is 1.5.
#91	#90	R/W	Mask address	When the mask address is 255.255.255.0 the data in CR#91 is 255.255, and the data in CR#90 is 255.0.
#93	#92	R/W	Gateway IP address	When the GIP address is 192.168.1.1, the data in CR#89 is 192.168, and the data in CR#88 is 1.1.
	#94	R/W	Enabling the IP address setting	0: The setting of the IP address is not executed. 1: The setting of the IP address is executed.
	#95	R	IP address setting status	0: The setting is unfinished. 1: The setting is being executed. 2: The setting is complete.
#113 ~ #96		-	Reserved	

CR number		Attribute	Register name	Description
HW	LW			
	#114	R/W	MPDBUS TCP time-out	Setting up MODBUS TCP time-out (in ms) Default: 3000
	#115	R/W	MODBUS TCP trigger	Setting up whether to send out data in MODBUS TCP mode
	#116	R/W	MODBUS TCP status	Displaying current status of MODBUS TCP mode
#118	#117	R/W	MODBUS TCP destination IP	Setting up destination IP address for MODBUS TCP transaction
	#119	R/W	MODBUS TCP data length	Setting up the data length for MODBUS TCP transaction
#219~#120		R/W	MODBUS TCP data buffer	Data buffer of MODBUS TCP for storing sending/receiving data
#248~#220		-	Reserved	
	#249	R	Sub-version	
	#250	R	Update date	0xC820 (April 8, 2012)
	#251	R	Error code	Displaying the errors. See the error code table for more information.
#255~#252		-	Reserved	
Symbols “R” refers to “able to read data by FROM instrcution”; “W” refers to “able to write data by TO instrcution”.				

4.5.2 ELC-COENETM (ETHERNET COMMUNICATION MODULE)

ELC-COENETM Ethernet communication module				
CR number		Attribute	Register name	Description
HW	LW			
	#0	R	Model name	Set up by the system; read only. Model code of ELC-COENETM = H’4050
	#1	R	Firmware version	Displaying the current firmware version in hex.
	#2	R	Communication mode	b0: MODBUS TCP mode b1: data exchange mode
	#3	W	E-Mail Event 1 trigger	Set up whether to send E-Mail 1
	#4	W	E-Mail Event 2 trigger	Set up whether to send E-Mail 2
	#5	W	E-Mail Event 3 trigger	Set up whether to send E-Mail 3
	#6	W	E-Mail Event 4 trigger	Set up whether to send E-Mail 4
	#7	R	Status of E-Mail 1, 2	b0 ~ b7: Current status of E-Mail 2 b8 ~ b15: Current status of E-Mail 1

ELC-COENETM Ethernet communication module				
CR number		Attribute	Register name	Description
HW	LW			
	#8	R	Status of E-Mail 3, 4	b0 ~ b7: Current status of E-Mail 4 b8 ~ b15: Current status of E-Mail 3
	#9	R/W	E-Mail 1 additional message	Filled in by the user, and it will be sent by E-mail.
	#10	R/W	E-Mail 2 additional message	Filled in by the user, and it will be sent by E-mail.
	#11	R/W	E-Mail 3 additional message	Filled in by the user, and it will be sent by E-mail.
	#12	R/W	E-Mail 4 additional message	Filled in by the user, and it will be sent by E-mail.
	#13	R/W	Data exchange trigger	Set up whether to send out data in data exchange mode
	#14	R	Status of data exchange	Displaying current status of data exchange.
	#15	R/W	Enabling flag for RTU mapping	1: Enable; 0: Disable. Default = 0
	#16	R/W	Connection status of RTU mapping slave	b0: Status of RTU slave 1 b1: Status of RTU slave 2 b2: Status of RTU slave 3 b3: Status of RTU slave 4
#24 ~ #17		-	Reserved	
#26	#25	R/W	Destination IP	Destination IP address for data exchange
	#27	-	Reserved	
	#28	R/W	Destination Slave ID	Destination Slave ID for data exchange
#48 ~ #29		R/W	Data transmission buffer	Buffer for transmitted data in data exchange
#68 ~ #49		R	Data receiving buffer	Buffer for received data in data exchange
#69~#80		-	Reserved	
	#81	R/W	Read address for data exchange	Slave transmission buffer address for data exchange
	#82	R/W	Read length for data exchange	Number of registers for read data
	#83	R/W	Received address for data exchange	Buffer address for the receiving Master in data exchange
	#84	R/W	Written-in address for data exchange	Buffer address for the receiving Slave in data exchange
	#85	R/W	Written-in length for data exchange	Number of registers for data transmission
	#86	R/W	Transmission address for data exchange	Master transmission buffer address for data exchange
#110 ~ #87		-	Reserved	
	#111	R/W	8-bit processing mode	Setting up MODBUS TCP Master control as 8-bit mode
	#112	R/W	MODBUS TCP Keep-Alive Time-out	MODBUS TCP Keep-Alive Time-out (s)
	#113	-	Reserved	
	#114	R/W	MODBUS TCP time-out	Setting up MODBUS TCP time-out (in ms)

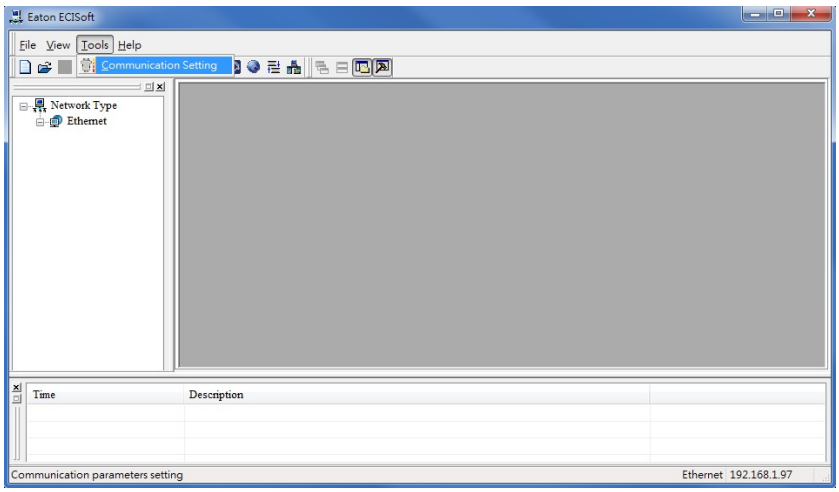
ELC-COENETM Ethernet communication module				
CR number		Attribute	Register name	Description
HW	LW			
	#115	R/W	MODBUS TCP trigger	Setting up whether to send out data in MODBUS TCP mode
	#116	R/W	MODBUS TCP status	Displaying current status of MODBUS TCP mode
#118	#117	R/W	MODBUS TCP destination IP	Setting up destination IP address for MODBUS TCP transaction
	#119	R/W	MODBUS TCP data length	Setting up the data length for MODBUS TCP transaction
#219 ~ #120		R/W	MODBUS TCP data buffer	Data buffer of MODBUS TCP for storing sending/receiving data
#248 ~ #220		-	Reserved	
	#251	R	Error code	Displaying the errors. See the error code table for more information.
#255 ~ #252		-	Reserved	

4.6 Searching for an Ethernet ELC

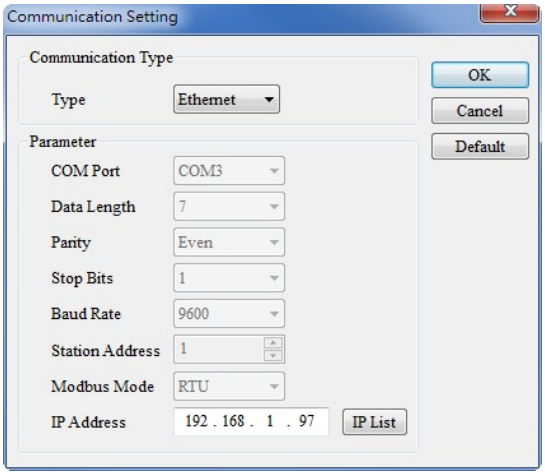
This section introduces how to search for and set an Ethernet ELC by ECISoft. Before you start a setup page, you have to select **Ethernet** in the **Communication Setting** window. Next, you can search by a broadcast, or an IP address. An Ethernet ELC is set up by UDP port 20006; therefore, you have to be aware of the relevant settings of the firewall.

4.6.1 COMMUNICATION SETTING

(1) Start ECISoft in your PC, and click **Communication Setting** on the **Tools** menu.

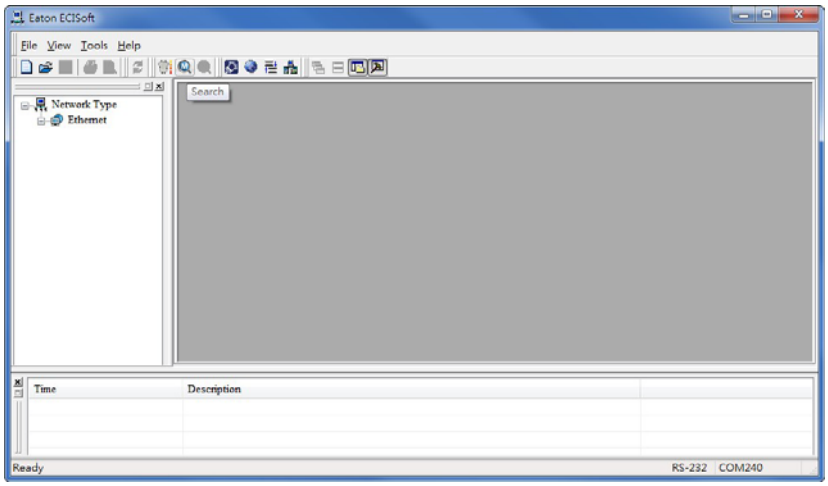


(2) Select **Ethernet** in the **Type** drop-down list box.

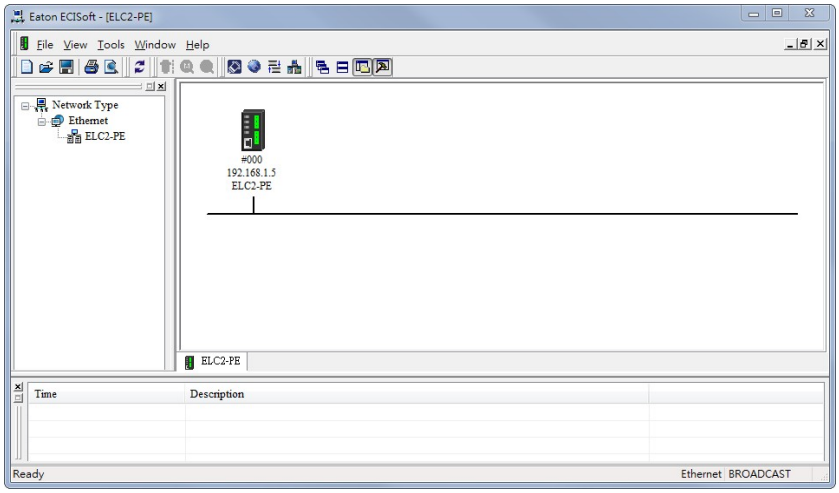


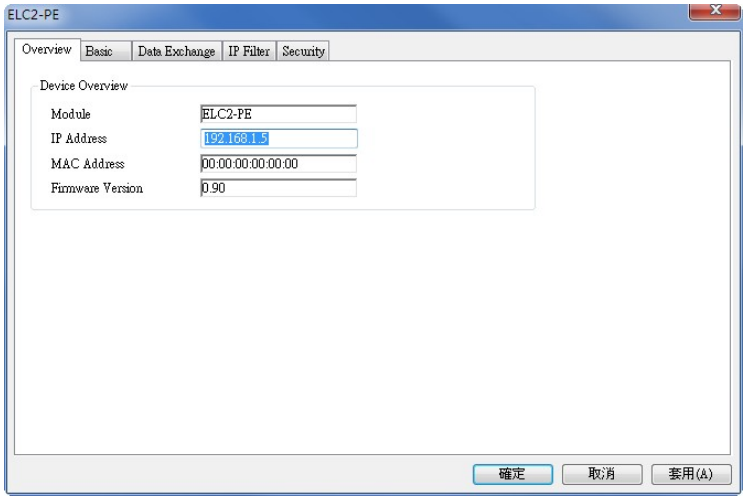
4.6.2 BROADCAST SEARCH

(1) Click **Search** on the toolbar in ECISoft to search for all Eaton Ethernet products on the network. The window on the left hand side shows the models found, and the window on the right hand side displays the device list of all models.



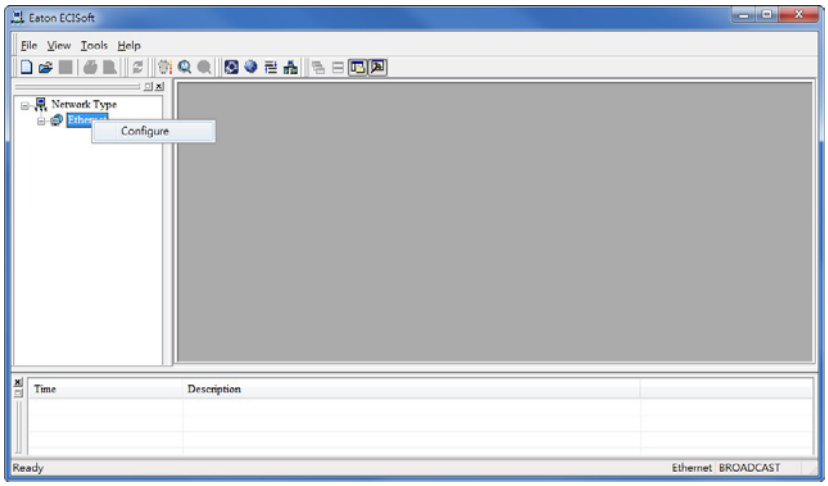
(2) Click a model on the left hand side, and you will see the device list of the model selected on the right hand side. Click the device to be set up to enter the setup page.



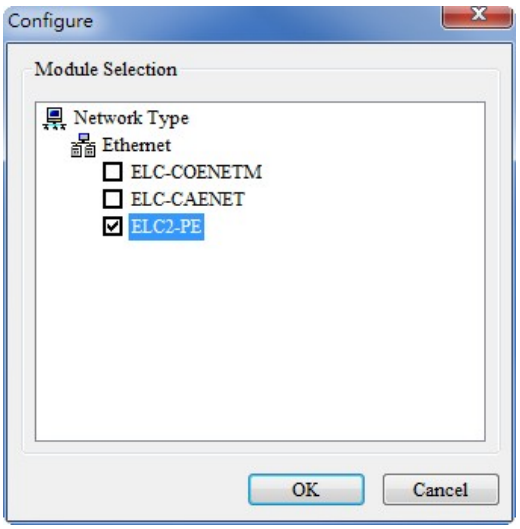


4.6.3 SEARCHING FOR A MODEL SPECIFIED

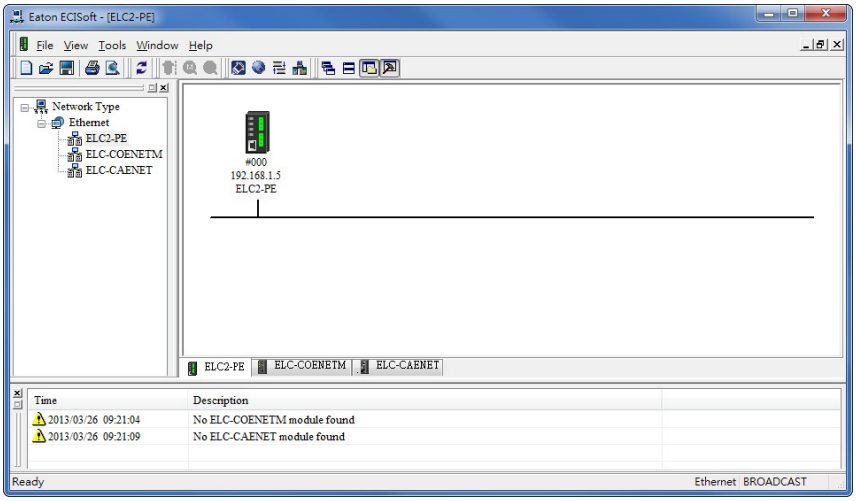
(1) Right-click **Ethernet** in the left hand side window, and click **Configure** to designate a model to be searched for.



(2) After users select a model which will be searched for, they can click **OK** to auto-search for the model on the network. In the window shown below, the **ELC-COENETM** checkbox is selected.

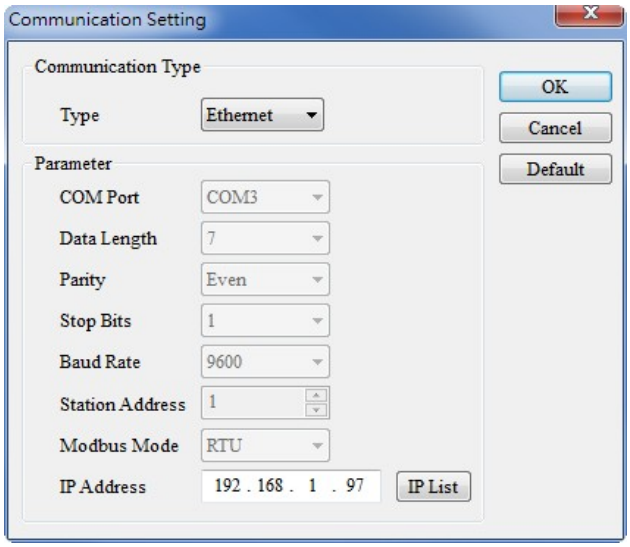


(3) A list of specified devices is in the window. If the users have selected several models, they can view these models.

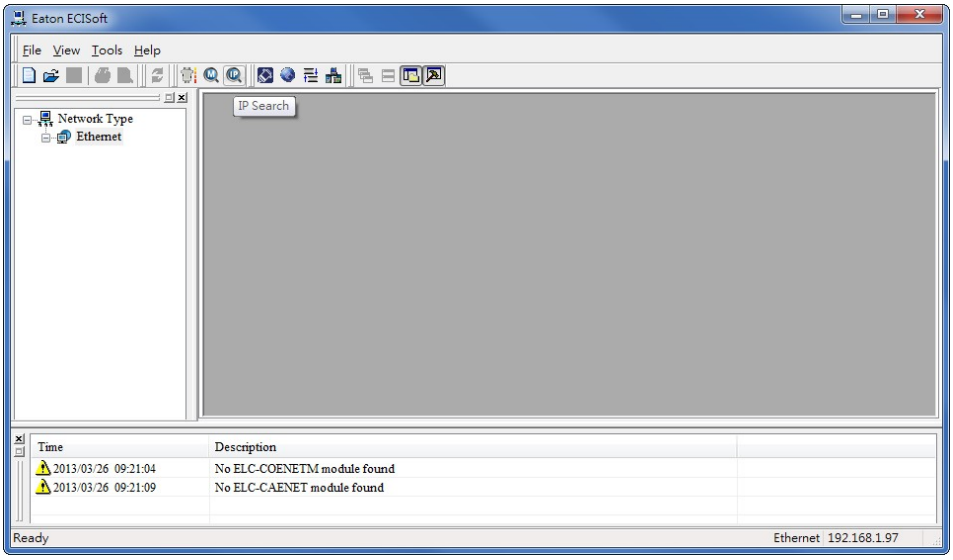


4.6.4 SEARCHING BY AN IP ADDRESS

(1) Select **Ethernet** in the **Type** drop-down list box, type an IP address in the **IP Address** box, and click **OK**.



(2) Click **Search** on the toolbar to start searching for the designated IP address.



(3) The model found will be displayed in the right hand side window. Double-click the device to enter the setup page.

4.7 Data Exchange

An Eaton Ethernet master can read/write data from/into a slave by means of instructions. It can also read/write data from/into a slave by means of tables. The number of data exchanges that models provide is different. Please refer to section B.1 for more information about the number of data exchanged.

(1) Enable:

Users can enable or disable a data exchange. After a data exchange is enabled, the data will be exchanged.

(2) Enable Condition:

You can select **Always Enable** or **Program Control**. If **Always Enable** is selected, ELC-COENETM will execute data exchange continuously until the setting in ECISoft is changed. If **Program Control** is selected, ELC-COENETM will execute data exchange according to the program setting. The internal registers in different models used to enable data exchanges are different. Please refer to section B.2 for more information.

(In ELC-COENETM, the data exchange is executed if CR#13=2, and the data exchanged is not executed if CR#13=0.)

(3) Station Address-IP Address:

You have to type the IP address of a slave. If the IP address of a slave is 192.168.0.1, and the station number of the slave is 1, you can type 1 in the first **Station Address** cell, select the box in the first **Enable** cell, and type 192.168.0.1 in the first **IP Address** cell.

(4) Master Device, Slave Device, and Quantity:

Reading (←): Initial reception register in a master←Initial transmission register in a slave

Writing (→): Initial transmission register in a master→ Initial reception register in a slave

If a data exchange is enabled, the Ethernet ELC will write data, and then read data.

Quantity: A slave station can send 100 pieces of data at most and receive 100 pieces of data at most simultaneously.

※ If a device which is not an Eaton ELC is connected, users can type a hexadecimal four-digit MODBUS absolute position in the **Slave Device** cell.

4.8 EtherNet/IP List

EtherNet/IP is a communication protocol defined by ODVA, and is different from the Ethernet mentioned in the previous sections. ELC2-PE series ELCs supports the EtherNet/IP slave communication protocol. The EtherNet/IP objects which are supported are described below.

4.8.1 ETHERNET/IP INFORMATION SUPPORTED BY ELC2-PE SERIES ELCS

(1) Object list

Object Name	Class Code	# of Instance
Identity	0x01	7
Message Router	0x02	NA
Assembly	0x04	7
Connection Manager	0x06	NA
X input	0x64	256
Y output	0x65	256
T Timer	0x66	256
M Relay	0x67	4096
C Counter	0x68	256
D Register	0x69	12000
TCP/IP Interface	0xF5	6
Ethernet Link	0xF6	3

(2) Data types

8-bit	16-bit	32-bit	64-bit
USINT	WORD	UDINT	ULINT

SINT	UINT	DWORD	LINT
BYTE	INT	DINT	

(3) Error codes

Value	Name	Description
0	Success	Success
0x01	Connection Failure	The forwarding function can not be enabled.
0x04	Path Segment Error	The segment type is not supported. (ref. V1 C-1.4)
0x05	Path Destination Unknown	The instance is not supported.
0x08	Service Not Supported	The service (Get or Set) is not supported.
0x09	Invalid Attribute Value	The value written is incorrect.
0x0E	Attribute Not Settable	The setting of the attribute is not allowed.
0x13	Not Enough Data	The length of the data written is too short.
0x14	Attribute Not Supported	The attribute is not supported.
0x15	Too Much Data	The length of the data written is too long.
0x16	Object Not Exist	The object is not supported.
0x20	Invalid Parameter	The service parameter is not supported. (ref. V1 5-2.3.1)
0x26	Path Size Invalid	Incorrect item length

4.8.2 ETHERNET/IP OBJECTS SUPPORTED BY ELC2-PE SERIES ELCS

(1) Identity Object (0x01)

Instance: 0x01

Attribute	Name	Access	Data Type	Value
0x01	Vendor ID	Get	UINT	68
0x02	Device Type	Get	UINT	14 (Programmable Logic Controller)
0x03	Product Code	Get	UINT	0x8013
0x04	Revision	Get	STRUCT of:	1.00
	Major		USINT	0x01
	Minor		USINT	0x00
0x05	Status	Get	WORD	0 (Owned)
0x06	Serial Number	Get	UDINT	
0x07	Product Name	Get	SHORT_STRING	ELC2-PE

(2) Message Router (0x02)

Instance: 0x01

Attribute	Name	Access	Data Type	Value
0x01	Not Support	NA	NA	NA

(3) Assembly (0x04)

Explicit message

Conformance Test is not supported.

Instance	Attribute	Name	Access	Data Type	Data
0x65	0x03	D Block 1	Set	10 words	D500~D509
0x66		D Block 2	Set	30 words	D510~D539
0x67		D Block 3	Set	60 words	D540~D599
0x68		D Block 4	Set	100 words	D600~D699
0x69		D Block 5	Set	100 words	D700~D799
0x6A		D Block 6	Set	100 words	D800~D899
0x6B		D Block 7	Set	100 words	D900~D999

(4) X input (0x64)

Instance	Attribute	Name	Access	Data Type
1	0x64	X0	Get	BYTE
2	0x64	X1	Get	BYTE
……				
256	0x64	X377	Get	BYTE

(5) Y output (0x65)

Instance	Attribute	Name	Access	Data Type
1	0x64	Y0	Set	BYTE (0x00 or 0x01)
2	0x64	Y1	Set	BYTE (0x00 or 0x01)
……				
256	0x64	Y377	Set	BYTE (0x00 or 0x01)

(6) T timer (0x66)

Instance	Attribute	Name	Access	Data Type
1	0x64	T0	Set	INT
2	0x64	T1	Set	INT
……				
256	0x64	T255	Set	INT

Instance	Attribute	Name	Access	Data Type
1	0x65	T0	Set	BYTE (0x00 or 0x01)
2	0x65	T1	Set	BYTE (0x00 or 0x01)
……				
256	0x65	T255	Set	BYTE (0x00 or 0x01)

(7) M Relay (0x67)

Instance	Attribute	Name	Access	Data Type
1	0x64	M0	Set	BYTE
2	0x64	M1	Set	BYTE
……				
4096	0x64	M4095	Set	BYTE

(8) C counter (0x68)

Instance	Attribute	Name	Access	Data Type
1	0x64	C0	Set	INT
2	0x64	C1	Set	INT
……				
200	0x64	C199	Set	INT

Instance	Attribute	Name	Access	Data Type
201	0x64	C200	Set	DINT
202	0x64	C201	Set	DINT

……				
256	0x64	C255	Set	DINT

Instance	Attribute	Name	Access	Data Type
1	0x65	C0	Set	BYTE (0x00 or 0x01)
2	0x65	C1	Set	BYTE (0x00 or 0x01)
……				
256	0x65	C255	Set	BYTE (0x00 or 0x01)

(9) D Register (0x69)

Instance	Attribute	Name	Access	Data Type
1	0x64	D0	Set	INT
2	0x64	D1	Set	INT
……				
12000	0x64	D11999	Set	INT

(10) TCP/IP Interface Object (0xF5)

Instance: 0x01

Attribute	Name	Access	Data Type	Value
0x01	Status	Get	DWORD	0x00000001UL
0x02	Configuration Capability	Get	DWORD	0x00000014UL (DHCP client, Configuration Settable)
0x03	Configuration Control	Get	DWORD	Static IP: 0U DHCP: 0x02U
0x04	Physical Link Object:	Get	STRUCT of:	
	Path Size		UINT	
	Path		Padded EPATH	
0x05	Interface Configuration:	Set	STRUCT of:	
	IP Address		UDINT	
	Network Mask		UDINT	
	Gateway Address		UDINT	
	Name Server		UDINT	
	Name Server 2		UDINT	
	Domain Name		STRING	
0x06	Host Name	Get	STRING	ELC2-PE

(11) Ethernet Link Object (0xF6)

Instance: 0x01

Attribute	Name	Access	Data Type	Value
0x01	Interface Speed	Get	UDINT	10 or 100 Mbps
0x02	Interface Flag	Get	UDINT	Bit 0: Link Status Bit 1: Half/Full Duplex
0x03	MAC Address	Get	USINT[6]	

Communications

This chapter contains information regarding the built-in communications ports of the ELC processors.

This Chapter Contains

5 ELC processor communications

This chapter describes the communications capabilities that are built in to the ELC processor modules.

A.1 Communication Ports

The communication interfaces which are supported are listed below. (All the ports can carry out communication simultaneously.):

COM port	COM1	COM2	COM3
ELC-PA/PV ELCB-PB ELC2-PB/PV	RS-232	RS-485	-
ELCM-PH/PA ELC2-PH	RS-232	RS-485	RS-485
ELC2-PE	USB	RS-485	RS-485
ELC2-PA	RS-232	RS-485	USB

Slave/Master mode (Both support ASCII/RTU):

COM port	COM1	COM2	COM3
ELC-PA/PV ELCB-PB	Slave	Slave/Master	-
ELCM-PH/PA ELC2-PH	Slave/Master	Slave/Master	Slave/Master
ELC2-PB/PV	Slave/Master	Slave/Master	-
ELC2-PE	Slave	Slave/Master	Slave/Master
ELC2-PA	Slave/Master	Slave/Master	Slave

Communication port summary:

COM / Parameter	COM1	COM2	COM3)
Baud rate	ELCB-PB, ELC-PA/PV, ELC2-PV:110~115200 bps In ELCM-PH/PA, ELC2-PB/PH/PA/PE, COM1: 110~115200 bps, COM2/COM3: 110~921,000 bps.		
Data length	7~8bits		
Parity	Even/Odd/None parity check		
Stop bits length	1~2 bits		
Configuration Register	D1036	D1120	D1109
communication mode	ASCII/RTU Mode		
Data length for access	100 registers (Both of ASCII/RTU)		

A.2 Configuration of the communication ports

Configuring the operational mode for the ports includes

- Selecting master or slave operation
- Selecting ASCII or RTU transmission mode
- Selecting data packet format: Data Length, Parity, and stop bit length

Note: For a Modbus serial network to operate properly, only one device can be a master and all devices must use the same transmission mode and data packet format.

SELECTING MASTER OR SLAVE OPERATION.

For the ELC-PA and the ELCB-PB models COM1 is always a slave port. COM2 is automatically set to master functionality if a message instruction is entered in the program, otherwise COM2 will function as a slave port.

SELECTING TRANSMISSION MODE.

The transmission mode for the ports is set using dedicated M bits as described in the chart below

Bit \ Value	0	1
M1139	COM1 = ASCII	COM1 = RTU
M1143	COM2 = ASCII	COM2 = RTU
M1320	COM3 = ASCII	COM3 = RTU

Default for all ports is ASCII transmission mode. If RTU transmission mode is desired for a port, the corresponding M bit should be set at the beginning of the program.

SELECTING DATA PACKET FORMAT.

The data packet format for the ports is set using dedicated D registers as described in the chart below

Register	Contents
D1036	COM1 data packet format
D1120	COM2 data packet format
D1109	COM3 data packet format

The table below defines the bit fields in these D registers:

	CONTENT		
b0	Data Length	0: 7 data bits, 1: 8 data bits	
b1 b2	Parity	00:None 01:Odd 11: Even	
b3	Stop bits	0: 1 bit, 1: 2 bits	
b4 b5 b6 b7	Baud rate	0001 (H1): 110 0010 (H2): 150 0011 (H3): 300 0100 (H4): 600 0101 (H5): 1200 0110 (H6): 2400 0111 (H7): 4800 1000 (H8): 9600 1001 (H9): 19200 1010 (HA): 38400 1011 (HB): 57600 1100 (HC): 115200 1101 (HD): 500000(COM2/COM3) 1110 (HE): 312500(COM2/COM3) 1111 (HF): 921000(COM2/COM3)	
b8	Select start bit	0: None	1: D1124(COM2)
b9	Select the 1st end bit	0: None	1: D1125(COM2)
b10	Select the 2nd end bit	0: None	1: D1126(COM2)
b11~b15	Reserved (leave at 0)		

Default data packet format for all ports is: 7 data bits, 1 stop bit, Even parity, 9600 baud. If a different data packet format is desired, the appropriate D register value should be loaded at the start of the user program.

A.3 Communication Protocol ASCII transmission mode

This section describes the messaging for a port that has been configured to support Modbus ASCII Transmission mode.
An ASCII transmission message Frame is shown below

Field name	Content	Explanation
Start bit	STX	Start bit ':' (3AH)
Slave address	ADR 1	Address consists of 2 ASCII codes
	ADR 0	
Function code	CMD 1	Command code consists of 2 ASCII codes
	CMD 0	
Data	DATA (0)	Data content consist of 2n ASCII codes, n≤205
	DATA (1)	
	…….	
	DATA (n-1)	
LRC checksum	LRC CHK 1	LRC checksum consists of 2 ASCII codes
	LRC CHK 0	
Stop bit	END1	Stop bit consists of 2 ASCII codes END1 = CR (0DH), END0 = LF (0AH)
	END0	

Corresponding table for Hexadecimal value and ASCII codes

ASCII	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"
Hex	30H	31H	32H	33H	34H	35H	36H	37H
ASCII	"8"	"9"	"A"	"B"	"C"	"D"	"E"	"F"
Hex	38H	39H	41H	42H	43H	44H	45H	46H

ADR（MODBUS ADDRESS）

The target address for the Modbus message. Valid addresses are in the range of 0…31. An address of 0 indicates a broadcast to all devices.
For Example, if the target address is 16 Decimal (10 Hex) the ADR values are:
(ADR 1, ADR 0)='1','0'⇨'1'=31H, '0' = 30H

CMD (FUNCTION CODE) AND DATA (DATA CHARACTERS)

The content of access data depends on the function code. For descriptions of supported function codes, please refer to **A.6** in this chapter.

For example, the following two messages demonstrate reading the 8 timer words T20~T27 from an ELC at Modbus address 1
The values of T20~T27 are mapped to Modbus Registers: H0614~H061B
Command message:
": 01 03 06 14 00 08 DA CR LF"

Field name	ASCII	Hex
STX	:	3A
Slave Address	01	30 31
Function code	03	30 33
Starting Address High	06	30 36
Starting Address Low	14	31 34
Number of Points High	00	30 30
Number of Points Low	08	30 38
LRC checksum	DA	44 41
END	CR LF	0D 0A

Response message:
“: 01 03 10 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 C8 CR LF”

Field name	ASCII	Hex
STX	:	3A
Slave Address	01	30 31
Command code	03	30 33
Bytes Count	10	31 30
Data Hi (T20)	00	30 30
Data Lo (T20)	01	30 31
Data Hi (T21)	00	30 30
Data Lo (T21)	02	30 32
Data Hi (T22)	00	30 30
Data Lo (T22)	03	30 33
Data Hi (T23)	00	30 30
Data Lo (T23)	04	30 34
Data Hi (T24)	00	30 30
Data Lo (T24)	05	30 35
Data Hi (T25)	00	30 30
Data Lo (T25)	06	30 36
Data Hi (T26)	00	30 30
Data Lo (T26)	07	30 37
Data Hi (T27)	00	30 30
Data Lo (T27)	08	30 38
Check sum(LRC)	C8	43 38
END	CR LF	0D 0A

LRC CHK（CHECK SUM）

The LRC (Longitudinal Redundancy Check) is calculated by summing up, module 256, the values of the bytes from ADR1 to the last data character, then calculating the hexadecimal representation of the 2’s-complement negation of the sum. The ELC supports LRC CHK per the Modbus Serial specification

A.4 Communication Protocol RTU transmission mode

This section describes the messaging for a port that has been configured to support Modbus RTU Transmission mode

An RTU transmission message Frame is shown below

START	No data input ≥ 10 ms (refer to end time list)
Slave Address	Slave Address: the 8-bit binary Modbus address
Function code	Command Code: the 8-bit binary Modbus function code
DATA (n-1)	Data Contents: n × 8-bit BIN data, n≦202
…….	
DATA 0	
CRC CHK Low	CRC Checksum: The 16-bit CRC checksum is composed of 2 8-bit binary codes
CRC CHK High	
END	No data input ≥ 10 ms (refer to end time list)

ADDRESS (SLAVE ADDRESS)

The target address for the Modbus message. Valid addresses are in the range of 0~254. An address of 0 indicates a broadcast to all devices.

CMD (FUNCTION CODE) AND DATA

The content of access data depends on the function code. For descriptions of supported function codes, please refer to **A.6** in this chapter.

For example, the following two messages demonstrate reading the 8 timer words T20~T27 from an ELC at Modbus address 1

The values of T20~T27 are mapped to Modbus Registers: H0614~H061B

Command message:

" 01 03 06 14 00 08 04 80"

Sent message:

Field Name	Example (Hex)
START	No data input ≥ end time
Slave Address	01
Function code	03
Starting Address	06
	14
Number of Points	00
	08
CRC CHK Low	04
CRC CHK High	80
END	No data input ≥ end time

ELC→PC

" 01 03 10 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 72 98"

Feedback message:

Field Name	Example (Hex)
START	No data input ≥ end time
Slave Address	01
Function code	03
Bytes Count	10
Data Hi (T20)	00
Data Lo (T20)	01
Data Hi (T21)	00
Data Lo (T21)	02
Data Hi (T22)	00
Data Lo (T22)	03

Field Name	Example (Hex)
Data Hi (T23)	00
Data Lo (T23)	04
Data Hi (T24)	00
Data Lo (T24)	05
Data Hi (T25)	00
Data Lo (T25)	06
Data Hi (T26)	00
Data Lo (T26)	07
Data Hi (T27)	00
Data Lo (T27)	08
CRC CHK Low	72
CRC CHK High	98
END	No data input ≥ end time

CRC CHK (CHECK SUM)

The CRC Check starts from “Slave Address” and ends in “The last data content.” Calculation of the CRC is performed per the Modbus serial specification

END TIME LIST

Baud rate	110	150	300	600	1200	2400	4800	9600	19200
End time (ms)	381	280	140	70	35	18	9	5	3
Baud rate	38400	57600	32500	115200 or more					
End time (ms)	2	1	2	1					

A.5 ELC Modbus Address mapping

This section describes how the native ELC data elements map to Modbus addresses

- Internal Device Communication Address

Device	Range		Type	ELC communication address (Hex)	Modbus communication address (Dec)
S	000~255		bit	0000~00FF	000001~000256
S	246~511		bit	0100~01FF	000247~000512
S	512~767		bit	0200~02FF	000513~000768
S	768~1023		bit	0300~03FF	000769~001024
X	000~377 (Octal)		bit	0400~04FF	101025~101280
Y	000~377 (Octal)		bit	0500~05FF	001281~001536
T	000~255		bit	0600~06FF	001537~001792
			word	0600~06FF	401537~401792
M	000~255		bit	0800~08FF	002049~002304
M	256~511		bit	0900~09FF	002305~002560
M	512~767		bit	0A00~0AFF	002561~002816
M	768~1023		bit	0B00~0BFF	002817~003072
M	1024~1279		bit	0C00~0CFF	003073~003328
M	1280~1535		bit	0D00~0DFF	003329~003584
M	1536~1791		bit	B000~B0FF	045057~045312
M	1792~2047		bit	B100~B1FF	045313~045568
M	2048~2303		bit	B200~B2FF	045569~045824
M	2304~2559		bit	B300~B3FF	045825~046080
M	2560~2815		bit	B400~B4FF	046081~046336
M	2816~3071		bit	B500~B5FF	046337~046592
M	3072~3327		bit	B600~B6FF	046593~046848
M	3328~3583		bit	B700~B7FF	046849~047104
M	3584~3839		bit	B800~B8FF	047105~047360
M	3840~4095		bit	B900~B9FF	047361~047616
C	0~199	16-bit	bit	0E00~0EC7	003585~003784
			word	0E00~0EC7	403585~403784
	200~255	32-bit	bit	0EC8~0EFF	003785~003840
			Dword	0EC8~0EFF	403785~403840
D	000~256		word	1000~10FF	404097~404352
D	256~511		word	1100~11FF	404353~404608
D	512~767		word	1200~12FF	404609~404864
D	768~1023		word	1300~13FF	404865~405120

Device	Range	Type	ELC communication address (Hex)	Modbus communication address (Dec)
D	1024~1279	word	1400~14FF	405121~405376
D	1280~1535	word	1500~15FF	405377~405632
D	1536~1791	word	1600~16FF	405633~405888
D	1792~2047	word	1700~17FF	405889~406144
D	2048~2303	word	1800~18FF	406145~406400
D	2304~2559	word	1900~19FF	406401~406656
D	2560~2815	word	1A00~1AFF	406657~406912
D	2816~3071	word	1B00~1BFF	406913~407168
D	3072~3327	word	1C00~1CFF	407169~407424
D	3328~3583	word	1D00~1DFF	407425~407680
D	3584~3839	word	1E00~1EFF	407681~407936
D	3840~4095	word	1F00~1FFF	407937~408192
D	4096~4351	word	9000~90FF	436865~437120
D	4352~4607	word	9100~91FF	437121~437376
D	4608~4863	word	9200~92FF	437377~437632
D	4864~5119	word	9300~93FF	437633~437888
D	5120~5375	word	9400~94FF	437889~438144
D	5376~5631	word	9500~95FF	438145~438400
D	5632~5887	word	9600~96FF	438401~438656
D	5888~6143	word	9700~97FF	438657~438912
D	6144~6399	word	9800~98FF	438913~439168
D	6400~6655	word	9900~99FF	439169~439424
D	6656~6911	word	9A00~9AFF	439425~439680
D	6912~7167	word	9B00~9BFF	439681~439936
D	7168~7423	word	9C00~9CFF	439937~440192
D	7424~7679	word	9D00~9DFF	440193~440448
D	7680~7935	word	9E00~9EFF	440449~440704
D	7936~8191	word	9F00~9FFF	440705~440960
D	8192~8447	word	A000~A0FF	440961~441216
D	8448~8703	word	A100~A1FF	441217~441472
D	8704~8959	word	A200~A2FF	441473~441728
D	8960~9215	word	A300~A3FF	441729~441984
D	9216~9471	word	A400~A4FF	441985~442240
D	9472~9727	word	A500~A5FF	442241~442496
D	9728~9983	word	A600~A6FF	442497~442752

Device	Range	Type	ELC communication address (Hex)	Modbus communication address (Dec)
D	9984~10239	word	A700~A7FF	442753~443008
D	10234~10495	word	A800~A8FF	443009~443246
D	10496~10751	word	A900~A9FF	443247~443502
D	10752~11007	word	AA00~AAFF	443503~443758
D	11008~11263	word	AB00~ABFF	443759~444014
D	11264~11519	word	AC00~ACFF	444015~444270
D	11520~11775	word	AD00~ADFF	444271~444526
D	11776~11999	word	AE00~AEDF	444527~444750
R	0 ~ 9999	word	C000~E70F	449153~459152

- **Devices which are supported are listed below.**

Device / Model	X	Y	M	S	T	C	D
ELCB-PB	X0~X177	Y0~Y177	M0~M1535	S0~S127	T0~T127	C0~C127, C232~C255	D0~D599, D1000~D1999
ELC-PA	X0~X177	Y0~Y177	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D4999
ELC-PV	X0~X377	Y0~X377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999
ELCM-PH ELCM-PA ELC2-PH ELC2-PA	X0~X377	Y0~X377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999
ELC2-PB	X0~X377	Y0~X377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D4999
ELC2-PV ELC2-PE	X0~X377	Y0~X377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999

A.6 Function Code support (Slave Mode)

The following function codes are supported by the ELC in slave mode:

CMD(Hex)	Explanation	Device
01 (01 H)	Read Coil Status of Contact	S, Y, M, T, C
02 (02 H)	Read Input Status	S, X, Y, M,T, C
03 (03 H)	Read content of register	T, C, D
05 (05 H)	Force single Coils	S, Y, M, T, C
06 (06 H)	Preset single register	T, C, D
15 (0F H)	Force Multiple Coils	S, Y, M, T, C
16 (10 H)	Preset Multiple Register	T, C, D

17 (11 H)	Retrieve information of Slave	None
23 (17 H)	Simultaneous data read/write in a polling of EASY ELC LINK	None

COMMAND CODE：01, READ COIL STATUS OF CONTACT

Number of Points (max) = 255 (Decimal) = FF (Hex)

Example：Read contacts T20~T56 from Slave ID#1

Master→ELC ":01 01 06 14 00 25 BF CR LF"

Sent message:

Field Name	ASCII
STX	3A (Hex)
Slave Address	01
Command code	01
Bytes Count	05
Data（Coils T27…T20）	CD
Data（Coils T35…T38）	6B
Data（Coils T43…T36）	B2
Data（Coils T51…T44）	0E
Data（Coils T56…T52）	1B
Error Check（LRC）	E6
ETX 1	0D (Hex)
ETX 0	0A (Hex)

Assume Number of Points in sent message is **n** (Dec), quotient of **n**/8 is **M** and the remainder is **N.**

When **N** = 0, Bytes Count in feedback message will be **M**; when N≠0, Bytes Count will be **M+1**.

ELC→Master ":01 01 05 CD 6B B2 0E 1B D6 CR LF"

Feedback message:

Field Name	ASCII
STX	3A (Hex)
Slave Address	01
Command code	01
Bytes Count	05
Data (Coils T27…T20)	CD
Data (Coils T35…T38)	6B

Field Name	ASCII
Data (Coils T43…T36)	B2
Data (Coils T51…T44)	0E
Data (Coils T56…T52)	1B
Error Check (LRC)	E6
END 1	0D (Hex)
END 0	0A (Hex)

COMMAND CODE：02, READ INPUT STATUS

Example: Read status of contact Y024~Y070 from Slave ID#01

Master→ELC ": 01 02 05 14 00 25 BF CR LF"

Sent message:

Field Name	ASCII
STX	3A (Hex)
Slave Address	01
Command code	02
Starting Address Hi	05
Starting Address Lo	14
Number of Points Hi	00
Number of Points Lo	25
Error Check (LRC)	BF
END 1	0D (Hex)
END 0	0A (Hex)

Assume Number of Points in sent message is **n** (Dec), quotient of **n**/8 is **M** and the remainder is **N.**

When **N** = 0, Bytes Count in feedback message will be **M**; when N≠0, Bytes Count will be **M+1**.

ELC→Master ": 01 01 05 CD 6B B2 0E 1B E5 CR LF"

Feedback message:

Field Name	ASCII
STX	3A (Hex)
Slave Address	01
Command code	02
Bytes Count	05
Data (Coils Y033…Y024)	CD
Data (Coils Y043…Y034)	6B
Data (Coils Y053…Y044)	B2

Field Name	ASCII
Data (Coils Y063…Y054)	0E
Data (Coils Y070…Y064)	1B
Error Check (LRC)	E5
END 1	0D (Hex)
END 0	0A (Hex)

COMMAND CODE：03, READ MULTIPLE REGISTER (T, C, D)

What is the maximum number of registers that can be read?

Example: Read coils T20~T27 from Slave ID#01

Master→ELC ": 01 03 06 14 00 08 DA CR LF"

Sent message:

Field Name	ASCII
STX	3A (Hex)
Slave Address	01
Command code	03
Starting Address Hi	06
Starting Address Lo	14
Number of Points Hi	00
Number of Points Lo	08
Error Check (LRC)	DA
END 1	0D
END 0	0A (Hex)

ELC→Master

":01 03 10 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 B8 CR LF"

Feedback message:

Field Name	ASCII
STX	3A (Hex)
Slave Address	01
Command code	03
Bytes Count	10

Field Name	ASCII
Data Hi (T20)	00
Data Lo (T20)	01
Data Hi (T21)	00
Data Lo (T21)	02
Data Hi (T22)	00
Data Lo (T22)	03
Data Hi (T23)	00
Data Lo (T23)	04
Data Hi (T24)	00
Data Lo (T24)	05
Data Hi (T25)	00
Data Lo (T25)	06
Data Hi (T26)	00
Data Lo (T26)	07
Data Hi (T27)	00
Data Lo (T27)	08
Error Check (LRC)	C8
END 1	0D (Hex)
END 0	0A (Hex)

COMMAND CODE：05, WRITE SINGLE COIL

A data value of FF00 (Hex) turns the coil on. A data value of 0000 (Hex) turns the coil off. All other data values are invalid and will have no effect.

Example: Force coil Y0 ON

Master→ELC ": 01 05 05 00 FF 00 F6 CR LF"

Sent message:

Field Name	ASCII
Heading	3A (Hex)
Slave Address	01
Command code	05
Coil Address Hi	05
Coil Address Lo	00
Force Data Hi	FF
Force Data Lo	00
Error Check (LRC)	F6

END 1	0D (Hex)
END 0	0A (Hex)

ELC→Master “: 01 05 05 00 FF 00 F6 CR LF”

Feedback message:

Field Name	ASCII
Heading	3A (Hex)
Slave Address	01
Command code	05
Coil Address Hi	05
Coil Address Lo	00
Force Data Hi	FF
Force Data Lo	00
Error Check (LRC)	F6
END 1	0D (Hex)
END 0	0A (Hex)

COMMAND CODE：06, WRITE SINGLE REGISTER

Example 1: Setting Register T0 to 12 34

Master→ELC “: 01 06 06 00 12 34 AD CR LF”

Sent message:

Field Name	ASCII
Heading	3A (Hex)
Slave Address	01
Command code	06
Register Address Hi	06
Register Address Lo	00
Preset Data Hi	12
Preset Data Lo	34
Error Check (LRC)	AD
END 1	0D (Hex)
END 0	0A (Hex)

ELC→Master “: 01 06 06 00 12 34 AD CR LF”

Feedback message:

Field Name	ASCII
Heading	3A (Hex)
Slave Address	01
Command code	06
Register T0 Address Hi	06
Register T0 Address Lo	00
Preset Data Hi	12
Preset Data Lo	34
Error Check (LRC)	AD
END 1	0D (Hex)
END 0	0A (Hex)

Example 2:

Case 1: C0

Master→ELC : 01 06 0E 00 12 34 AF CR LF

Case 2: C232 (32bit) (This is an exception of the function code.)

Master→ELC: 01 06 0E E8 12 34 56 78 EF CR LF

Case 3: D10

Master→ELC: 01 06 10 0A 12 34 99 CR LF

Case 4: D1000

Master→ELC: 01 06 13 E8 12 34 BA CR LF

COMMAND CODE：15, WRITE MULTIPLE COILS

Max contacts/coils: 255

Example: Set Coil Y007…Y000 = 1100 1101, Y011…Y010 = 01.

Master→ELC “: 01 0F 05 00 00 0A 02 CD 01 11 CR LF”

Sent message:

Field Name	ASCII
Heading	3A (Hex)
Slave Address	01
Command code	0F
Coil Address Hi	05
Coil Address Lo	00
Quantity of Coils Hi	00
Quantity of Coils Lo	0A

Field Name	ASCII
Byte Count	02
Force Data Hi	CD
Force Data Lo	01
Error Check (LRC)	11
END 1	0D (Hex)
END 0	0A (Hex)

ELC→PC ": 01 0F 05 00 00 0A E1 CR LF"

Feedback message:

Field Name	ASCII
Heading	3A (Hex)
Slave Address	01
Command code	0F
Register T0 Address Hi	05
Register T0 Address Lo	00
Preset Data Hi	00
Preset Data Lo	0A
Error Check (LRC)	E1
END 1	0D (Hex)
END 0	0A (Hex)

COMMAND CODE：16, WRITE MULTIPLE REGISTER

What is the maximum number of registers supported?

Example: Set register T0 to 00 0A, T1 to 01 02.

Master→ELC ": 01 10 06 00 00 02 04 00 0A 01 02 C6 CR LF"

Sent message:

Field Name	ASCII
Heading	3A (Hex)
Slave Address	01
Command code	10
Starting Address Hi	06
Starting Address Lo	00
Number of Register Hi	00
Number of Register Lo	02

Byte Count	04
Data Hi	00
Data Lo	0A
Data Hi	01
Data Lo	02
Error Check (LRC)	C6
END 1	0D (Hex)
END 0	0A (Hex)

ELC→Master ": 01 10 06 00 00 02 E7 CR LF"

Feedback message:

Field Name	ASCII
Heading	3A (Hex)
Slave Address	01
Command code	10
Starting Address Hi	06
Starting Address Lo	00
Number of Registers Hi	00
Number of Registers Lo	02
Error Check (LRC)	E7
END 1	0D (Hex)
END 0	0A (Hex)

COMMAND CODE：17, REPORT SLAVE ID

Returns a description of the ELCs present state

Sent message:

Field Name	ASCII
Heading	3A (Hex)
Slave Address	01
Command code	11
Error Check (LRC)	EE
END 1	0D (Hex)
END 0	0A (Hex)

Feedback message:

Field Name	ASCII
Heading	3A (Hex)
Slave Address	01
Command code	11
Byte Count	04
Slave ID	01
Run Indicator Status 00 = OFF FF = ON	FF
Data 0（D1001 HI）	40
Data 1（D1001 LOW）	10
Error Check (LRC)	9A
END 1	0D (Hex)
END 0	0A (Hex)

What is the “other information” in Data 0 and Data 1?

Exception response:

If the ELC is in slave mode and is unable to execute the Modbus function, it will return one of the following Exception codes:

Exception code:	Meaning:
01	Illegal function code: The function code received in the command message is not supported by the ELC.
02	Illegal device address: The device address received in the command message is not valid for the ELC.
03	Illegal device value: The device value received in the command message is not valid for the ELC.
07	Check Sum Error Check if the check Sum is correct Illegal command messages The command message is too short. Command message length is out of range.

A.7 Function Code Support (Master mode)

The ELC can function as a Modbus master device. Modbus messages are generated by the ELC based on the inclusion of Modbus message command. The table below describes the Modbus message commands and the corresponding Modbus Function codes.

Command	Function Code	Description
MODRD API=100	03H Read multiple registers	The ELC can read up to 6 register values
MODWR API=101	06H Write Single Register	The ELC can write a single register value
MODRW API=150	Selectable in the instruction.	The ELC can read the register values by H03/H04. The ELC can read the coil states by H02. The ELC can write the register values by H06/H10. The ELC can write the coil states by H05/H0F.

MEMO

Troubleshooting

This chapter provides information for troubleshooting during ELC operation.

This Chapter Contains

B Troubleshooting

B.1 Common Problems and Solutions

The following tables list common problems and troubleshooting procedures for the ELC system.

System Operation

Symptom	Troubleshooting and Corrective Actions
All LEDs are OFF	1. Check the power supply wiring. 2. Check if the power supply output power capacity is sufficiently rated for the ELC control units. 3. Check that the supply voltage is in the allowable range and is free of fluctuation. 4. If the power supply is shared with other devices, disconnect the power supply wiring to the other devices. If the LEDs on the ELC control unit now turn ON, the capacity of the power supply is insufficient. Add another power supply for the other devices or increase the capacity of the power supply. 5. If the problem can not be resolved with the instructions above, contact the Eaton support center.
ERROR LED is flashing	1. A flashing ERROR LED potentially indicates one of the following issues: • An invalid command, • A communication error, • An invalid operation, or missing instructions, The corresponding error code and error step are stored in special registers of the ELC. Fault code: D1004 Fault step: D1137 The Fault code can be read using ELCSoft. The fault codes are described in section B.2 of this document. 2. If the LED is flashing rapidly and it is not possible to connect ELCSoft to the ELC, a low 24VDC supply voltage is indicated. Check for possible 24VDC supply overload. 3. A steady ON LED indicates the program loop execution time is over the preset time (D1000 preset value). Check the program and the WDT (Watch Dog Timer). 4. If the problem can not be resolved with the instructions above, contact the Eaton support center

Symptom	Troubleshooting and Corrective Actions
Input does not turn on.	Check the corresponding input indicator LED. If the indicator is OFF: 1. Check the wiring of the input device 2. Check that power is present at the input terminals, and that the voltage level is within the specified range for the ELC Input. 3. If the power is present at the input terminal within the specified range but the input LED is off, a failure in the ELC's input circuit is indicated. Contact the Eaton support center. If the power is NOT present at the input terminal, a failure in the input device or input power supply is indicated. If the input LED indicator is ON: 1. Monitor the input condition using a programming tool. If the input monitored is OFF, a failure in the ELC's input circuit is indicated. Contact the Eaton support center. If the input monitored is ON, check the program flow and input address assignments.
Input does not turn off	Check the corresponding Input LED. If the indicator is ON 1. Check the wiring of the input device 2. If a two-wire sensor is used, check for excessive leakage current. 3. Check the voltage level at the input terminal. If it is below the input threshold but the Input LED is ON, a failure in the ELC input circuit is indicated. Contact the Eaton support center If the LED indicator is OFF 1. Monitor the input condition with a programming tool. If the input monitored is ON, a failure in the ELC's input circuit is indicated. Contact the Eaton support center 2. If the input monitored is off check the program flow and input address assignment.

Symptom	Troubleshooting and Corrective Actions
Output does not turn on.	Check the corresponding output LED. If the indicator is ON, 1. Check the wiring of the loads. 2. Check if power is properly supplied to the loads. 3. If the wiring is correct but power is not supplied to the load, there is potentially a fault with the ELC's output circuit. Contact the Eaton support center If the corresponding output LED is off, 1. Monitor the output condition using a programming tool. If the output monitored is turned Off, recheck your program 2. If the output monitored is on, there is a potential duplicate output instruction, or a potential fault in the ELC output circuit

B.2 Fault Code Table (Hex)

Errors detected in the ELC program will result in a flashing of the ERROR LED and M1004 = ON. The corresponding fault code (hex) can be read from special register D1004. The program address where the error occurred is stored in the data register D1137. If the error is a general loop error, the address stored in D1137 will be invalid.

Fault Code	Description	Action
0001	Operand bit device S exceeds the usage range	Check the D1137 (Error step number) Re-enter the instruction correctly
0002	Label P exceeds the usage range or duplicated	
0003	Operand KnSm exceeds the usage range	
0102	Interrupt pointer I exceeds the usage range or duplicated	
0202	Instruction MC exceeds the usage range	
0302	Instruction MCR exceeds the usage range	
0401	Operand bit device X exceeds the usage range	
0403	Operand KnXm exceeds the usage range	
0501	Operand bit device Y exceeds the usage range	
0503	Operand KnYm exceeds the usage range	
0601	Operand bit device T exceeds the usage range	
0604	Operand word device T register usage exceeds limit	
0801	Operand bit device M exceeds the usage range	
0803	Operand KnMm exceeds the usage range	
0B01	Operand K, H available range error	
0D01	DECO Misuse operand	
0D02	ENCO Misuse Operand	
0D03	DHSCS Misuse Operand	
0D04	DHSCR Misuse Operand	
0D05	PLSY Misuse Operand	
0D06	PWM Misuse Operand	
0D07	FROM / TO Misuse Operand	
0D08	PID Misuse Operand	
0D09	SPD Misuse Operand	
0D0A	DHSZ Misuse Operand	
0D0B	IST Misuse Operand	
0E01	Operand bit device C exceeds the usage range	
0E04	Operand word device C register usage exceeds limit	
0E05	DCNT misuse operand C	
0E18	BCD Conversion Error	
0E19	Division (divisor=0)	

Fault Code	Description	Action
0E1A	Device use is out of range (including index registers E, F)	Check the D1137 (Error step number) Re-enter the instruction correctly
0E1B	negative number after radical expression	
0E1C	FROM/TO communication error	
0F04	Operand word device D register usage exceeds limit	
0F05	DCNT Misuse Operand D	
0F06	SFTR Misuse Operand	
0F07	SFTL Misuse Operand	
0F08	REF Misuse Operand	
0F09	Improper use of operands of WSFR, WSFL instructions	
0F0A	Number of TTMR, STMR instruction exceed the range	
0F0B	Number of SORT instruction exceed the range	
0F0C	Number of TKY instruction exceed the range	
0F0D	Number of HKY instruction exceed the range	
1000	ZRST Misuse Operand	
10EF	E and F misuse operand or exceed the usage range	
2000	Usage exceed limit (MTR, ARWS, TTMR, PR, HOUR)	

Fault Code	Description	Action
C400	An unrecognized instruction code is being used	Select programming mode and correct the identified error
C401	Loop Error	
C402	LD / LDI continuously use more than 9 times	
C403	MPS continuously use more than 9 times	
C404	FOR-NEXT exceed 6 levels	
C405	STL / RET used between FOR and NEXT SRET / IRET used between FOR and NEXT MC / MCR used between FOR and NEXT END / FEND used between FOR and NEXT	
C407	STL continuously use more than 9 times	
C408	Use MC / MCR in STL, Use I / P in STL	
C409	Use STL/RET in subroutine or interrupt program	
C40A	Use MC / MCR in subroutine or interrupt program	
C40B	MC / MCR does not begin from N0 or discontinuously	
C40C	MC / MCR corresponding value N is different	
C40D	Use I / P incorrectly	

Fault Code	Description	Action
C40E	IRET does not follow by the last FEND command SRET does not follow by the last FEND command	
C40F	ELC program and data in parameters have not been initialized	Select programming mode and correct the identified error
C41B	Invalid RUN/STOP instruction to extension module	
C41C	The number of input/output points of I/O extension unit is larger than the specified limit	
C41D	Number of extension modules exceeds the range	
C41F	Failing to write data into memory	
C430	Initializing parallel interface error	
C440	Hardware error in high-speed counter	
C441	Hardware error in high-speed comparator	
C442	Hardware error in MCU pulse output	
C443	No response from extension unit	
C4EE	No END command in the program	
C4FF	Invalid instruction (no such instruction existing)	

B.3 Error Detection Addresses

Error Check Address	Description	Drop Latch	STOP → RUN	RUN → STOP
M1067	Program execution error flag	None	Reset	Latch
M1068	Execution error latch flag	None	Latch	Latch
D1067	Algorithm error code	None	Reset	Latch
D1068	Step value of algorithm errors	None	Latch	Latch

D1067 Error Code	Description
0E18	BCD Conversion Error
0E19	DIVISION (divisor=0)
0E1A	Floating Point exceeds the usage range
0E1B	The value of square root is negative

B.4 Table for Self-Detecting Abnormality

When you encounter abnormality using the product, you can analyze the problem first by doing the self detections below.

Abnormality	Possible cause	Suggested correction
Output point abnormality	Loosened terminal block	Check if the removable terminal block is loosened.
	The input counting specification may not match the pulse output frequency of the ELC model in use.	Check if the hardware is normal by low-frequency pulse counting.
Communication abnormality	The length of communication cable	Make sure the RS-232 cable is at least 3 meters long to ensure normal communication (specification unknown).
	Incorrect communication protocol or address setting	Broadcast from station 0 first by RS-232 to search for communication protocol and address and later confirm by RS-485 communication.
Extension module unable to work	Poor connection or MPU problem	Make sure the MPU is tightly connected to the extension module and compare to make sure whether the problem lies in the MPU or the extension module.
Counter (input point) abnormality	The applicable frequency exceeds the maximum bandwidth.	The frequency should be within the allowed ELC specifications.
ERROR LED flashes	Incorrect program syntax	Record the error code first, and write whether the ERROR LED should not flash anymore by syntactically correct program. Model with battery should be checked whether the time of RTC is correct (not being correct means the battery might once be unattached, causing reset of time).
L.V. LED On	Low input power supply	Make sure the power supply voltage is normal.
RUN LED Off after execution	There is no program inside the new ELC, resulting in misjudgment.	Write in the program first.

Installing a USB Driver in the ELC

This Chapter Contains

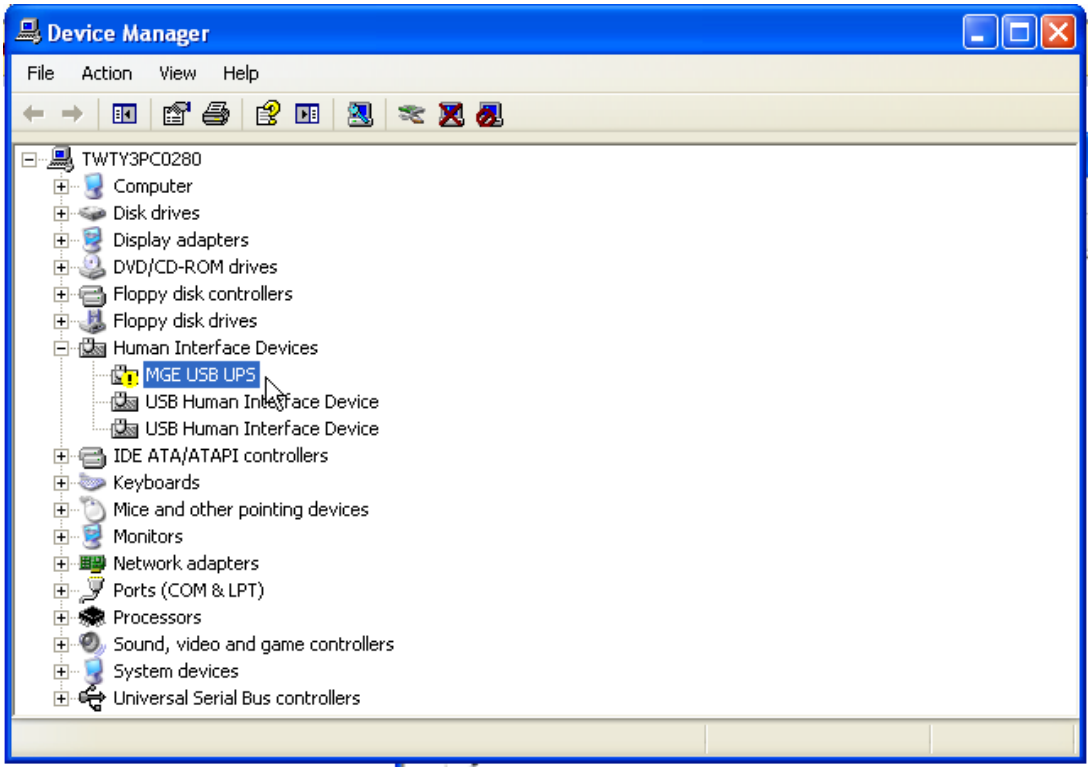
5.1 Installing the USB Driver for ELC2-PE

This section introduces the installation of the Eaton ELC2-PE USB driver in the computer. After the driver is installed, the USB interface can be used as the serial port (RS-232). Please use the standard USB cable. The length of the cable should be within five meters.

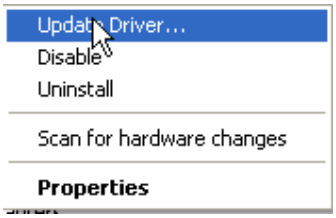
Installing the driver

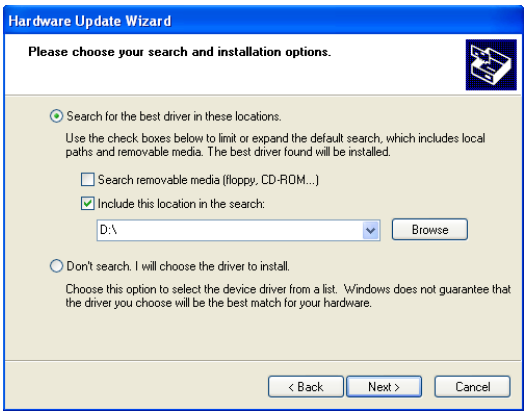
The personal computer and ELC2-PE are connected through a USB cable (A type) and a mini USB (B type) cable. The steps of installing the USB driver for ELC2-PE are described below.

Step 1: After a computer and ELC2-PE are connected, users can find **MGE USB UPS** in the **Device Manager** window.

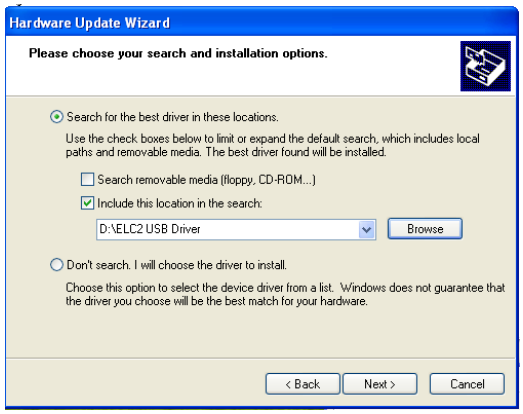


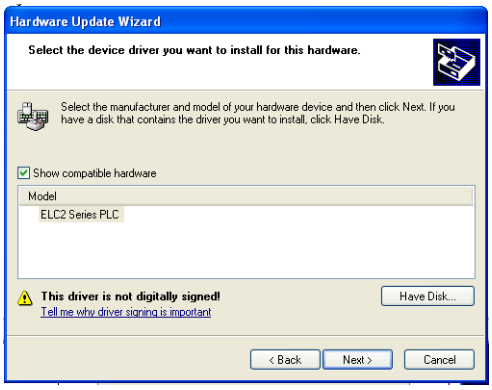
Step 2: Right-click **MGE USB UPS**, and select **Update Driver…** on the context menu to open the **Hardware Update Wizard** window.

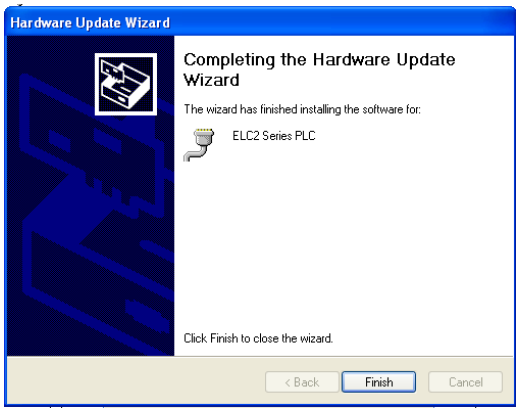




Step 3: Click **Browse** to specify the folder where the USB driver is saved, and then click **Next** to start the installation of the USB driver.

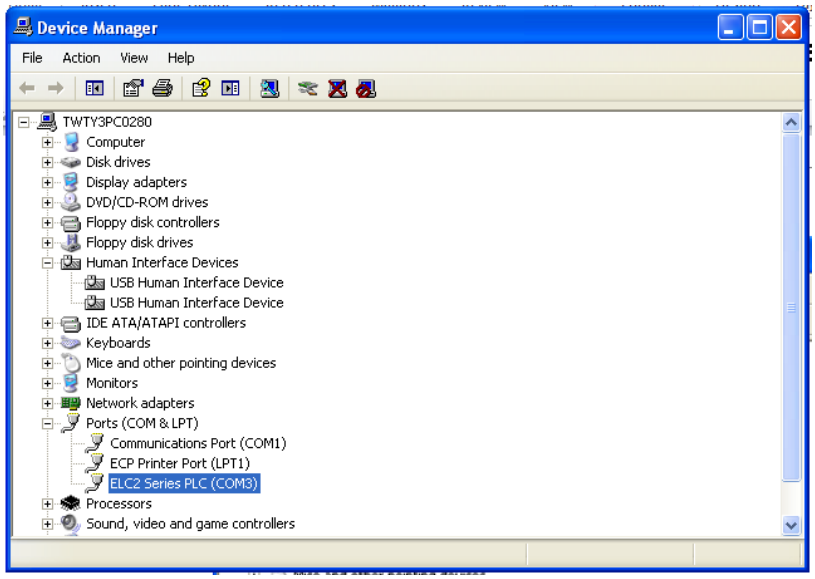




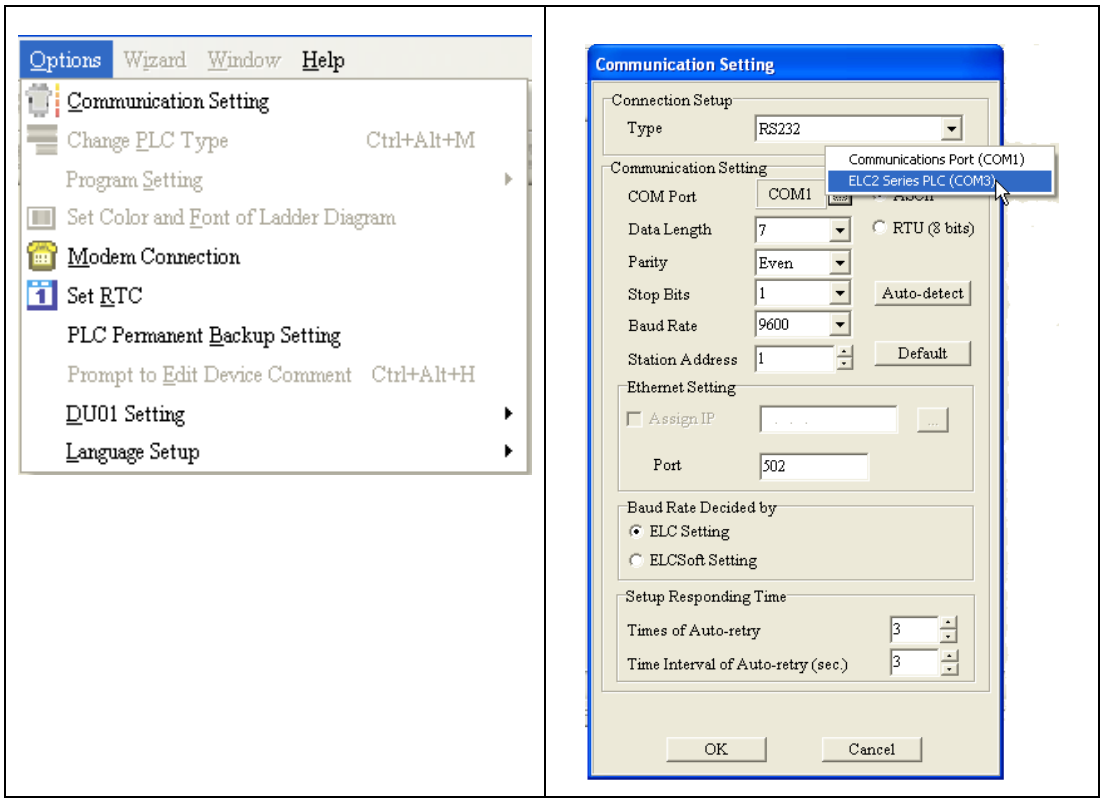


Step 4: After the USB driver is installed, the users can find **ELC2 series PLC** and the communication

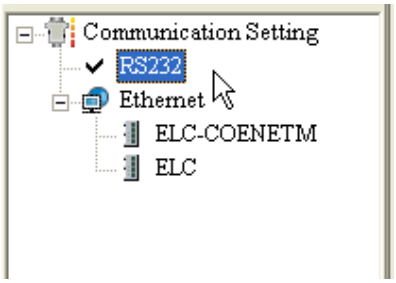
port assigned to it in the **Device Manger** window. The usage of this device is the same as that of RS-232.



Step 5: Click **Communication Setting** on the **Options** menu to open the **Communication Setting** window. Select **RS232** in the **Type** drop-down list box in the **Connection Setup** section, select the communication port assigned by the USB in the **Communication Setting** section, and click **OK**.



Step 6: After the communication setting is complete, the users can find that **RS232** in the communication work area is checked. They can download the program to ELC2-PE, upload the program from ELC2-PE through the USB, and use the online mode.



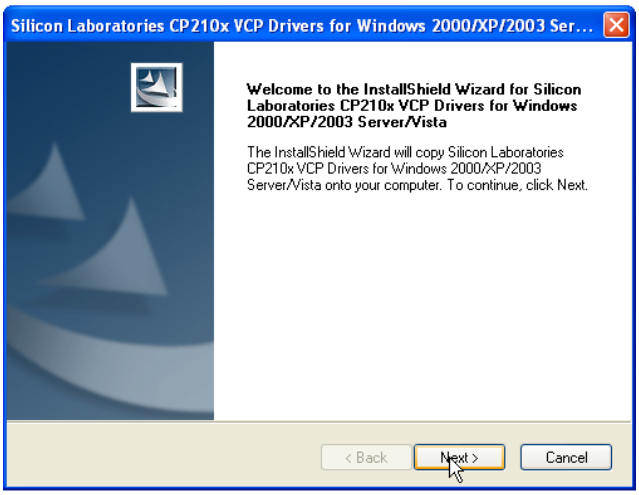
5.2 Installing the USB Driver for ELC2-PA

This section introduces the installation of the Eaton ELC2-PA USB driver in the computer. After the driver is installed, the USB interface can be used as the serial port (RS-232). Please use the standard USB cable. The length of the cable should be within five meters.

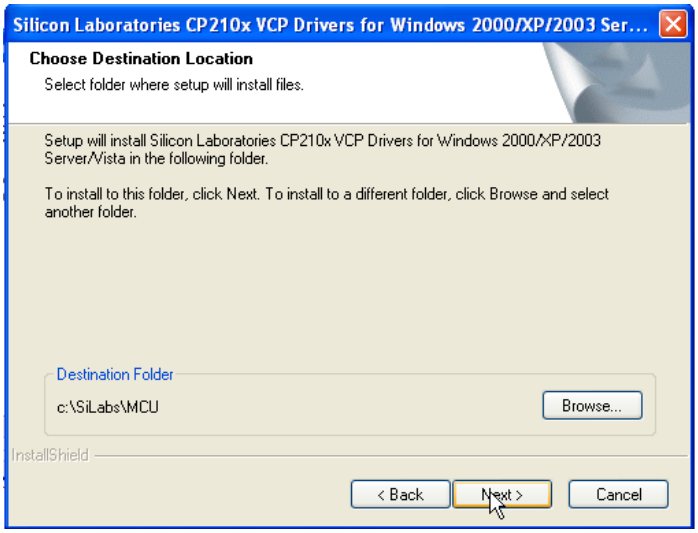
Installing the driver

The personal computer and ELC2-PA are connected through a USB (A type) cable and a mini USB (B type) cable. The steps of installing **Silicon Laboratories CP210x VCP Drivers for Windows 2000/XP/2003 Server/Vista** are described below.

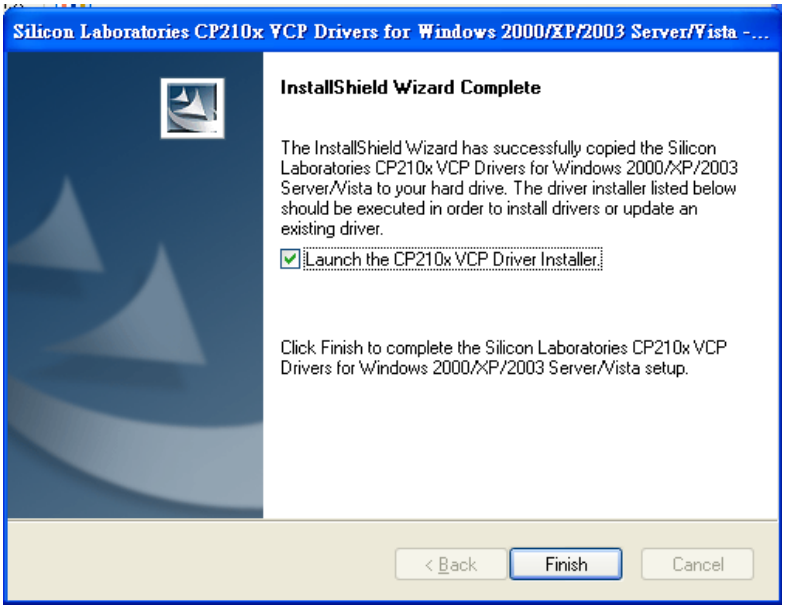
Step 1: Click **Next** in the **Silicon Laboratories CP210x VCP Drivers for Windows 2000/XP/2003 Server/Vista** window.



Step 2: Click **Next** in the **Silicon Laboratories CP210x VCP Drivers for Windows 2000/XP/2003 Server/Vista** window.



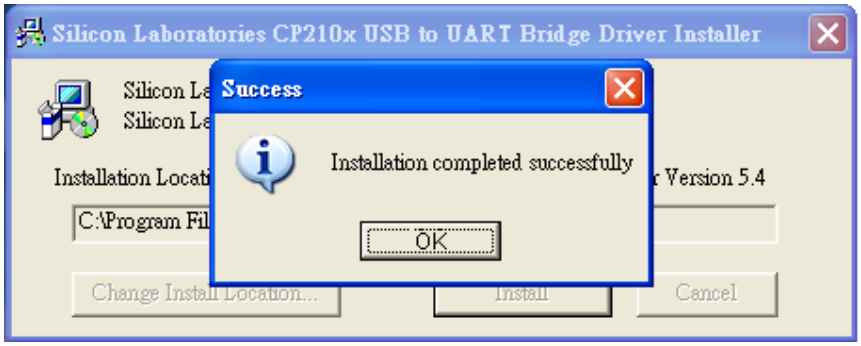
Step 3: Click **Finish** in the **Silicon Laboratories CP210x VCP Drivers for Windows 2000/XP/2003 Server/Vista** window.



Step 4: Click **Install** in the **Silicon Laboratories CP210x USB to UART Bridge Driver Installer** window which appears.

Step 5: After **Installed** is clicked, the **Scanning** window will appear.

Step 6: After the system is scanned, the **Success** window will appear. Users can click **OK**.



Step 7: After the driver is installed, the users can find **Silicon Labs CP210x USB to UART Bridge** and the communication port assigned to it in the **Device Manger** window.