8903/CB
CANopen Communications Interface

Technical Manual

HA469262U001 Issue 2

Compatible with Version 1.x Software

Safety Information

WARNING!
During commissioning, remove the fuses (or trip the circuit breaker) on your 3-phase supply.
Make sure the power is OFF, and that it cannot be switched on accidentally whilst you are working.

Please read this information BEFORE installing the equipment.

Intended Users

This manual is to be made available to all persons who are required to install, configure or service equipment described herein, or any other associated operation.

The information given is intended to highlight safety issues, EMC issues, and to enable the user to obtain maximum benefit from the equipment

Application Area

The equipment described is intended for industrial motor speed control.

Personnel

Installation, operation and maintenance of the equipment should be carried out by qualified personnel. A qualified person is someone who is technically competent and familiar with all safety information and established safety practices; with the installation process, operation and maintenance of this equipment; and with all the hazards involved.

REFER TO YOUR MAIN PRODUCT MANUAL FOR SPECIFIC SAFETY INFORMATION ABOUT THE DEVICE YOU ARE CONTROLLING

Acknowledgements

DSE (Drives System Explorer) is a registered trademark of SSD Drive Inc.

SyCon (System Configurator) is a registered trademark of Hilscher GmbH.

WARRANTY

Parker SSD Drives warrants the goods against defects in design, materials and workmanship for the period of 12 months from the date of delivery on the terms detailed in Parker SSD Drives Standard Conditions of Sale IA058393C.

Parker SSD Drives reserves the right to change the content and product specification without notice.

Contents

Contents *Page*

8903/CB CANopen TechCard 1

Contents

Contents *Page*

8903/CB CANopen TechCard

System Overview

Product Features

- Suitable for use with drive models:

  890CD Common Bus Drive using 890 firmware version 1.3 onwards

  890SD Standalone Drive using 890 firmware version 1.3 onwards

- Easy plug-in installation
- CANopen Port
- LED's to indicate board and communications status
- Hardware or software-selectable Slave Address and Baudrate

Product Code

Part Number: 8903/CB - CANopen TechCard

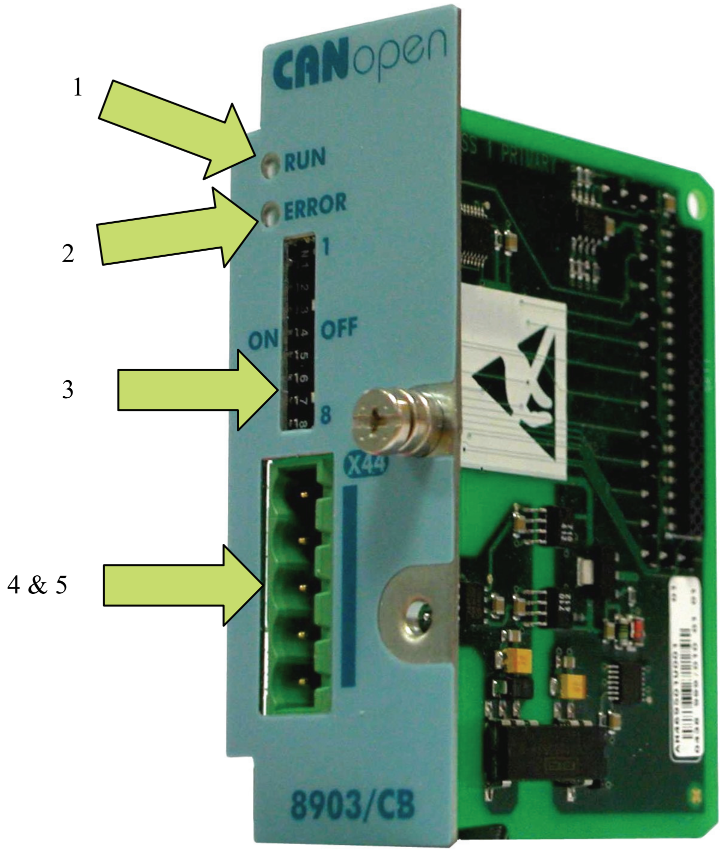


Figure 1. CANopen TechCard

1	Run LED	4	X44 CANopen port
2	Error LED	5	Connector (not shown)
3	Node Address and Baudrate Switches		

2

Installation

WARNING!
Disconnect all sources of power before attempting installation.

To Remove the Control Board

1. Remove the blank covers, each secured by a single screw (1), that fit over the TechCard holes.
2. Remove the top and bottom screws from the blue handles of the Control Board (2).
3. Pull gently on the handles and slide the Control Board out of the drive.

Note: *Save the blank cover and screw for future use. The drive should not be operated without a TechCard or blank cover. When fitted, these maintain the drive's IP20 rating.*

Caution
This Option contains ESD (Electrostatic Discharge) sensitive parts. Observe static control precautions when handling, installing and servicing this Option.

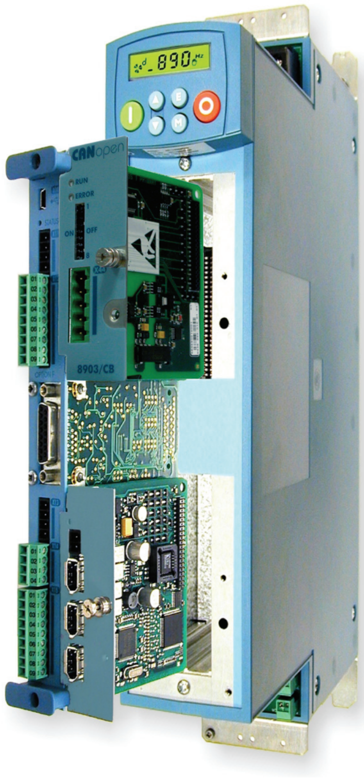


Figure 2. 890 showing Control Board withdrawn with Options fitted

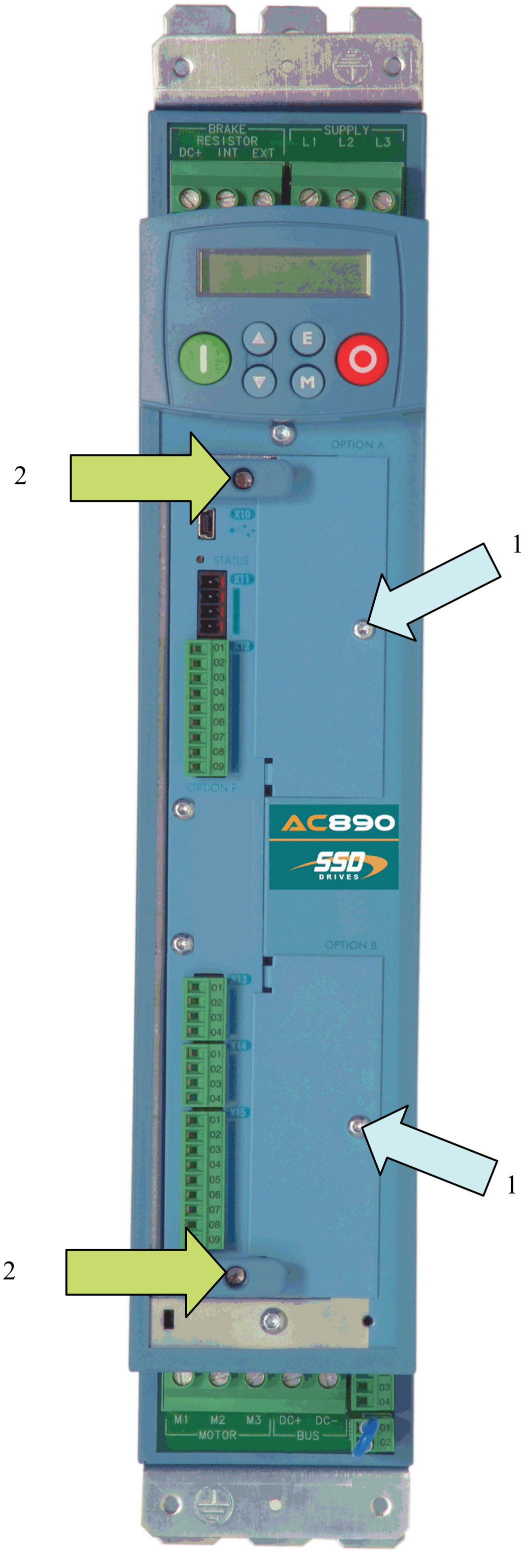


Figure 3. Front of 890 drive showing Control Board fitted

Fitting the TechCard

The TechCard fits onto the Control Board.

1. Insert the connector into the TechCard as shown. The legs of the connector will protrude through into the connector on the other side of the TechCard.

2. Press the assembly into the **TOP** connector (adjacent to terminals X10, X11 and X12) on the Control Board. Ensure that the front panel of the TechCard overlaps the front of the Control Board. Ease the connector at the TechCard so that the two pcb's are parallel when viewed on edge.

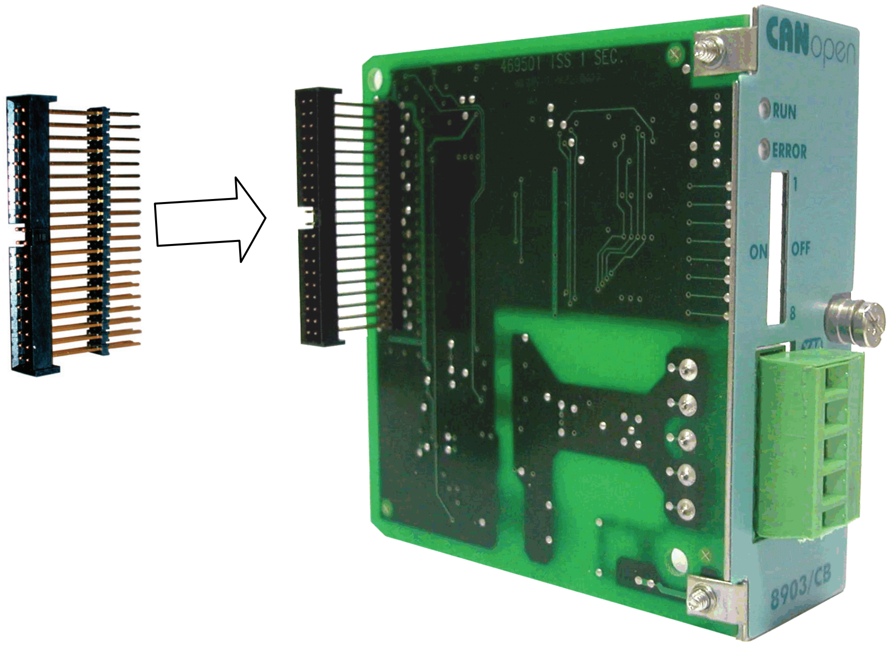


Figure 4. Fitting the connector to the TechCard

Re-fitting the Control Board

1. Slide the board into the drive, engaging the edges of the boards into the slots. Push until the back edge of the Control Board pcb locates with the connectors in the drive.
2. Tighten in position using the top and bottom screws in the blue handles of the Control Board.
3. Screw the TechCard in position using the captive screw on the front of the TechCard.

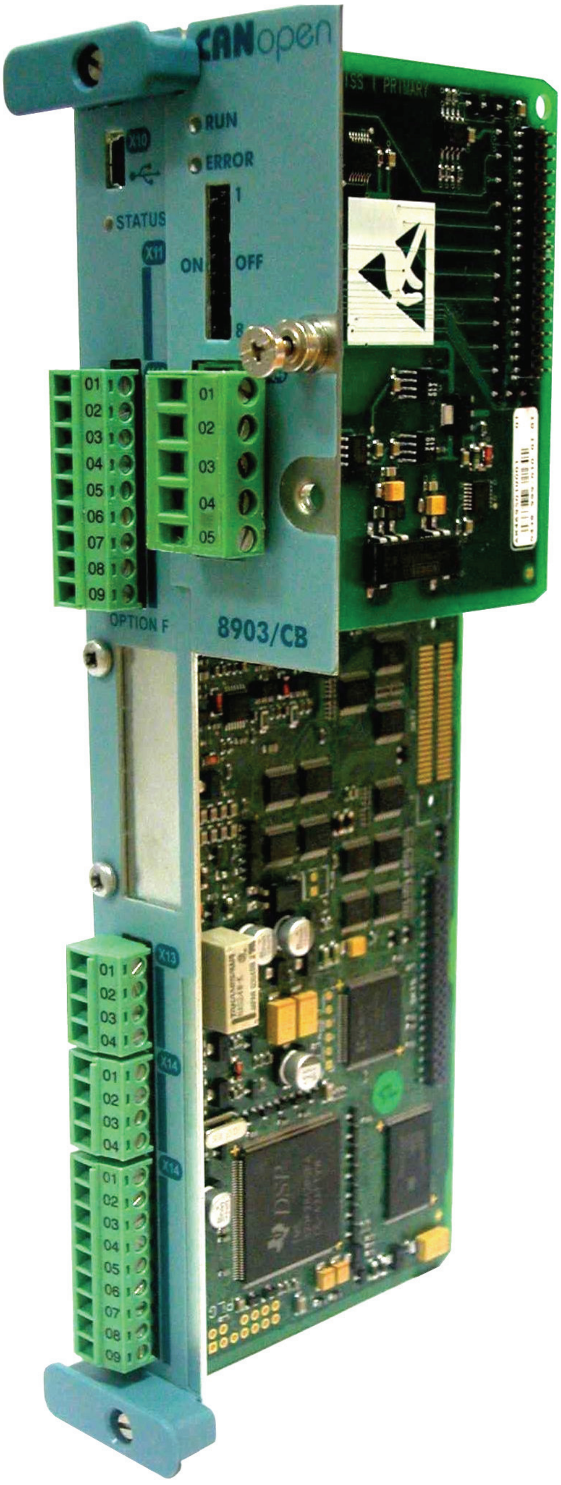


Figure 5. 890 Control Board with TechCard fitted

Wiring the System

Terminal X44

Pin	Connection
1	GND
2	CAN-
3	Screen (N/C)
4	CAN+
5	V+ (N/C)

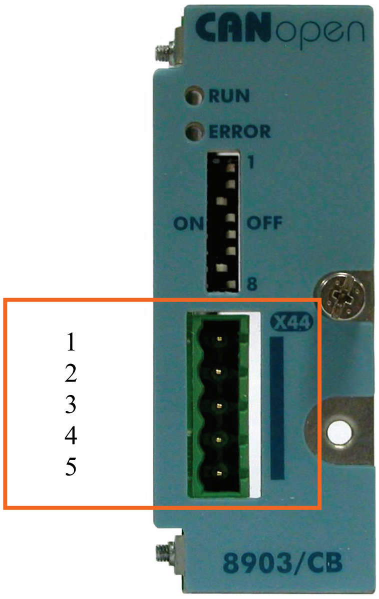


Figure 6. Terminal X44

Note: *It is possible to make serial communications operate without adhering to the following recommendations, however, the recommendations will promote greater reliability.*

Cable Specification

The media for CANopen is a shielded copper cable consisting of one twisted pair and two optional cables for an external power supply. As standard, the CANopen option does not use the external power supply. The user organisation (CiA) has specified ISO/DIS 11898 as the standard bus cable.

Maximum Cable Lengths

The maximum cable length depends on the baud rate selected:

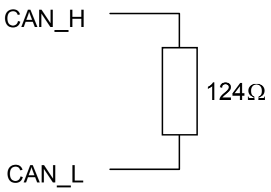
Data Rate	Maximum Distance
125 kBit/s	500 metres
250 kBit/s	250 metres
500 kBit/s	100 metres
1 Mbit/s	25 metres

Terminators

- If the drive is at the end of the trunk it must have a terminating resistor.
- All other drives in the system should not have a terminator.

Connect terminating resistors to the last drive as shown opposite. (resistor is ±1%, minimum ¼ Watt).

The CANopen specification recommends 124Ω, but it should be chosen to equal as closely as possible the characteristic impedance of the cable.



IMPORTANT: Failing to fit terminating resistors correctly may result in unreliable operation.

Initial Power-on Checks

Understanding the Status LED Indications

Table 1: CANopen RUN LED - Green

RUN LED	State	Description
Flickering	AutoBaud/LSS	Auto Baudrate detection in progress or LSS services in progress.
Single Flash	STOPPED	The Device is in STOPPED state
Blinking	PRE-OPERATIONAL	The Device is in PRE-OPERATIONAL state
On	OPERATIONAL	The Device is in OPERATIONAL state

Table 2: CANopen ERROR LED - Red

ERROR LED	State	Description
Off	No error	The Device is in working condition
Single Flash	Warning limit reached	At least one of the error counters of CAN controller has reached or exceeded the warning level (too many error frames)
Flickering	Autobaud/LSS	Auto Baudrate detection in progress or LSS services in progress.
Double Flash	Error Control Event	A guard event or heartbeat event has occurred
Triple Flash	Sync Error	The SYNC message has not been received within the configured communication cycle period time out
On	Bus Off	The CAN controller bus is off

The LED states and flash rates are as defined in the CiA DR-303-3 Indicator Specification.

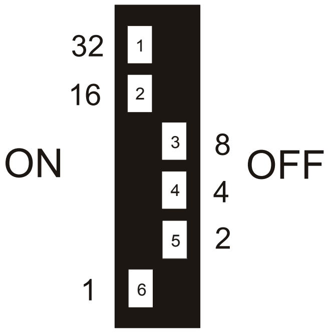
Setting the Node Address and Baudrate

The 890CD and 890SD drives are configured identically.

Node Address

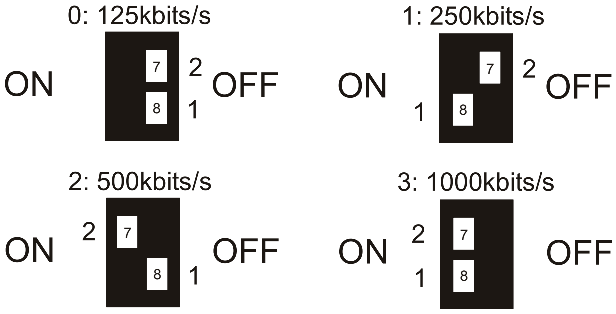
The node address is set using switches 1 to 6. Set a value between 1 and 63.

For example, 49 in binary is:



Baudrate

The CANopen baudrate is set using switches 7 and 8.



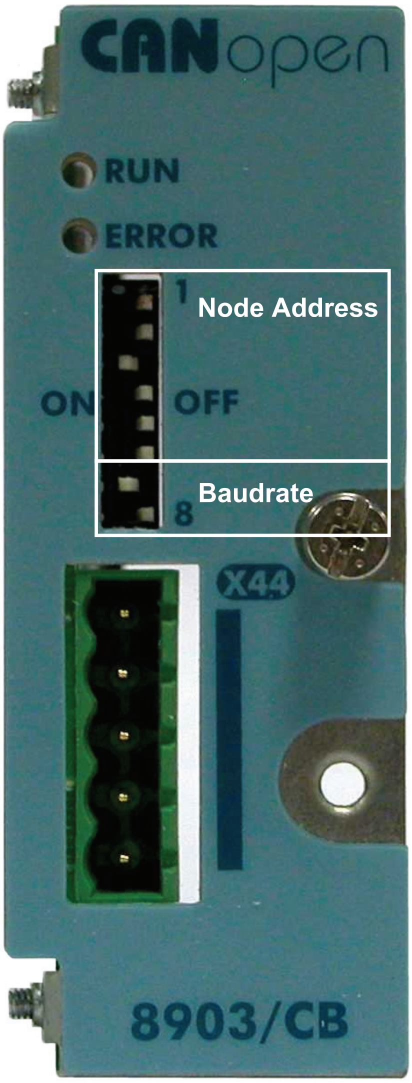


Figure 7. Setting the Node Address and Baudrate

Note: *If all switches are set to ON, the node address and baudrate are set by the MMI or the DSE Configuration Tool.*

8

Configuring the Drive

The CANopen MMI View

The CANopen TechCard correctly installed, the CANOPEN function block will contain the following parameter names when viewed using the MMI. These are read-only parameters.

Parameter Descriptions

BAUDRATE *Read Only* *Range: Enumerated – see below*

The CANopen baudrate being used.

0: 125K
1: 250K
2: 500K
3: 1000K

NODE ADDRESS *Read Only* *Range: 1 to 63*

The CANopen node address being used.

STATUS RUN *Read Only* *Range: Enumerated - see below*

Displays the CANopen running state.

Enumerated Value :

0 : STOPPED
1 : PRE-OPERATIONAL
2 : OPERATIONAL

STATUS ERROR *Read Only* *Range: Enumerated - see below*

Displays the CANopen error state.

Enumerated Value :

0 : NO ERROR
1 : WARNING LIMIT
2 : AUTOBAUD OR LSS
3 : CONTROL EVENT
4 : SYNC. ERROR
5 : BUS OFF

HARDWARE *Read Only* *Range: FALSE / TRUE*

The method being used to set the node address and baudrate. If all the Node Address and Baudrate Switches are set to ON, then the method is set by MMI or the DSE Configuration Tool, otherwise it is by hardware i.e. by the switches.

Enumerated Value : Hardware

0 : FALSE	Baudrate set by MMI or the DSE Configuration Tool
1 : TRUE	Baudrate set by hardware

BAUDRATE SOFT *Read/Write* *Range: Enumerated – see below*

The Baudrate set by software, either by the MMI or by the DSE Configuration Tool. (Functional when all the Node Address and Baudrate Switches are set to ON).

Enumerated Value : Baudrate Soft

0: 125K
1: 250K
2: 500K
3: 1000K

ADDRESS SOFT *Read/Write* *Range: 1 to 63*

Sets the address set by software, either by the MMI or by the DSE Configuration Tool. (Functional when all the Node Address and Baudrate Switches are set to ON).

Configuring the CANopen System

To configure the CANopen system, complete the steps below. Our example is shown using a PLC configured using SyCon® System Configurator by Hilscher GmbH (http://www.hilscher.com/) For other systems, refer to the manufacturer's instructions.

Step 1: Configuring the CANopen TechCard using DSE 890

You can configure your CANopen TechCard using DSE 890. Follow the instructions below.

Step 1.1: Inserting a CANOPEN Function Block

Display your configuration page. Click on the Block menu at the top of the screen.

1. Move the cursor down to select "890 Comms" and select "CANopen".

2. Click to select the CANopen block. Move this to where you want on the screen then click again to place the block.

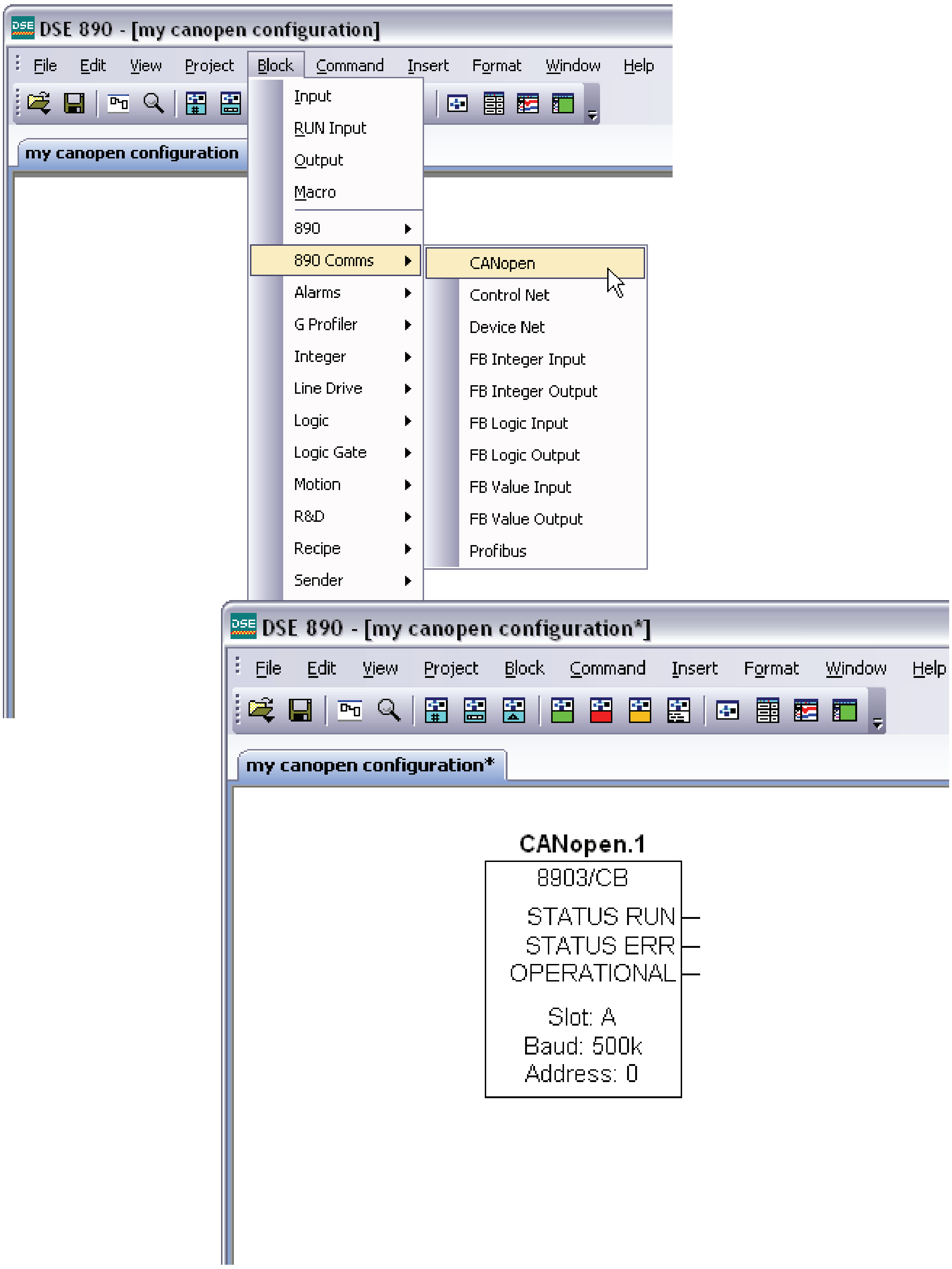


Figure 8. Configuration showing CANopen function block

Step 1.2: Attaching Fieldbus Connectors

Six fieldbus connector types are available:

FB Logic Input
FB Logic Output
FB Integer Input
FB Integer Output
FB Value Input
FB Value Output

Input connector : the data is sent from PLC → 890

Output connector : the data is sent from 890 → PLC

The fieldbus connectors must be added before they will appear in the CANopen function block.

Note: *The function block and connectors can be renamed by using the right mouse button and selecting Rename Block.*

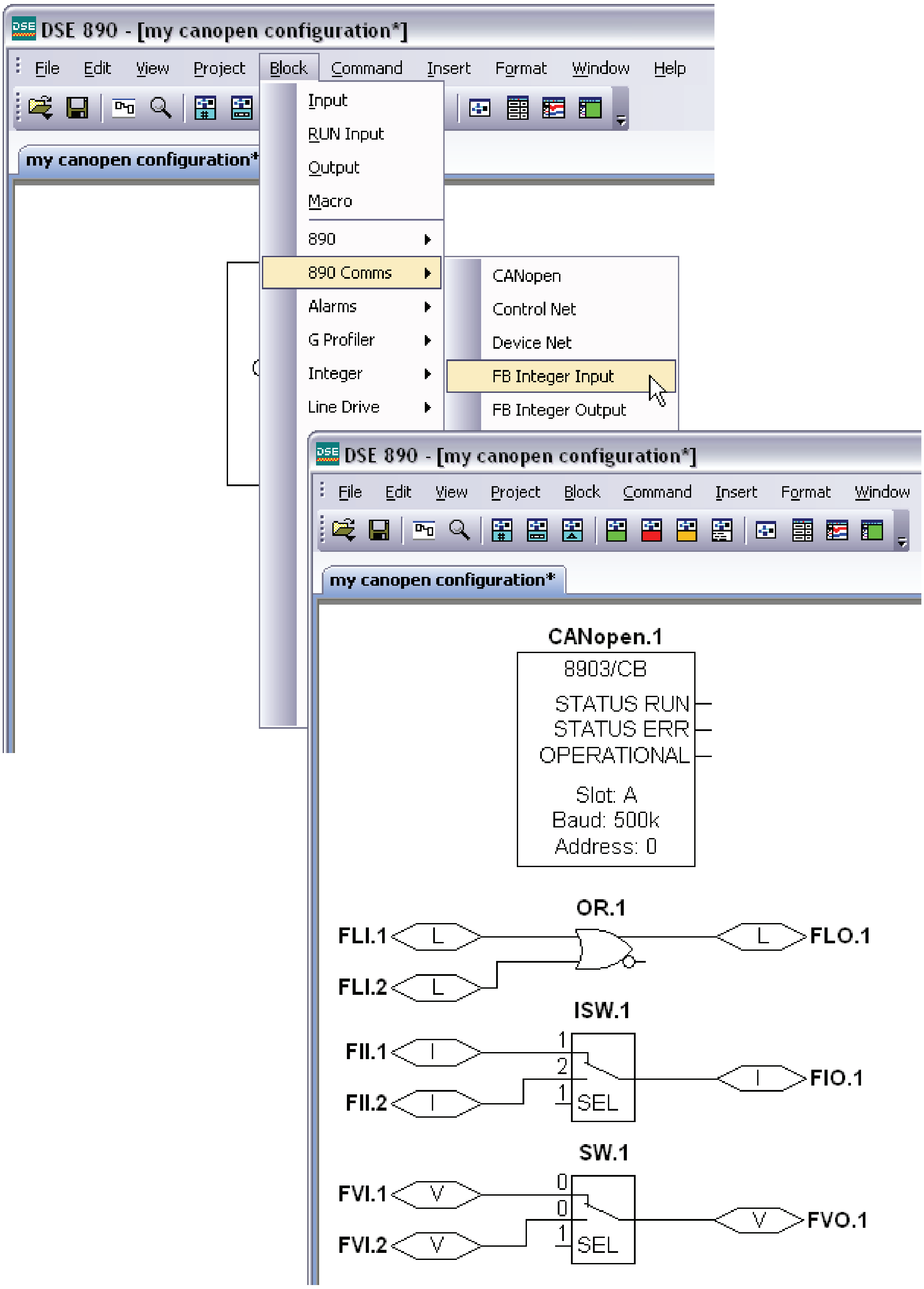
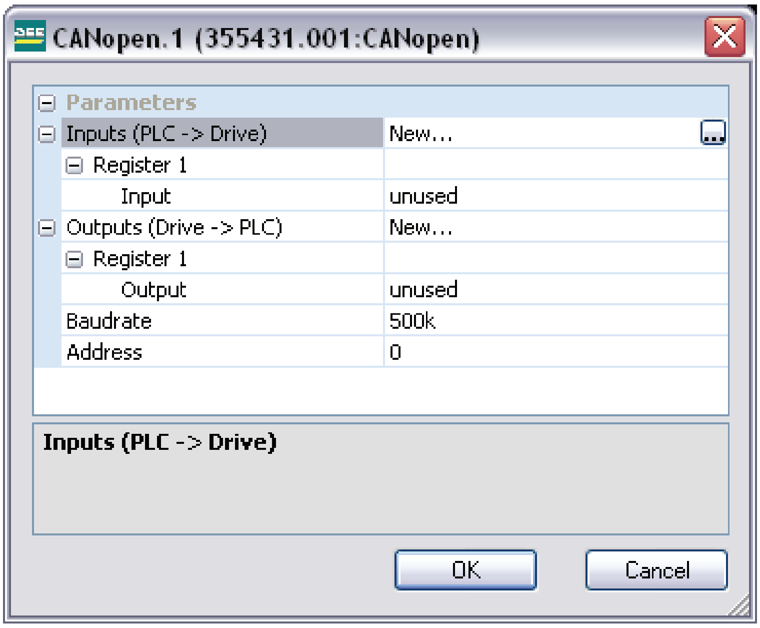


Figure 9. Configuration showing CANopen function block and Fieldbus Connectors

Step 1.3 : Configuring the Fieldbus Connectors

Double-click on the function block to display the dialog below. The fieldbus connectors (inputs and outputs) are assignable in the function block along with their data type to/from the PLC. The option slot and Address can also be selected.



CANopen.1 (355431.001:CANopen)

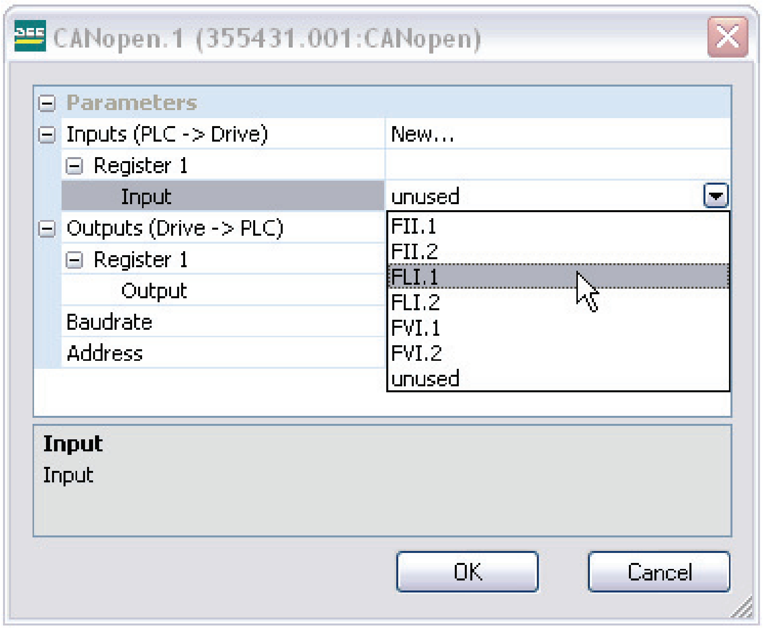
Parameters	
Inputs (PLC -> Drive)	New...
Register 1	
Input	unused
Outputs (Drive -> PLC)	New...
Register 1	
Output	unused
Baudrate	500k
Address	0

Inputs (PLC -> Drive)

To configure the input and output connectors you have placed in the configuration:

1. Expand the **Inputs** and **Outputs** trees to reveal the registers. By default the trees each have one register. To add more registers click on ... adjacent to **New…**

2. Select the drop-down menu adjacent to **Input** to choose the required input/output connector on the Register. For example below, Register 1 "Input" is shown with the possible fieldbus selections that have been placed in the configuration: FII.1 (**F**ieldbus **I**nteger **I**nput 1), FLI.1 (**F**ieldbus **L**ogic **I**nput **1**), FVI.1 (**F**ieldbus **V**alue **I**nput **1**) etc.



CANopen.1 (355431.001:CANopen)

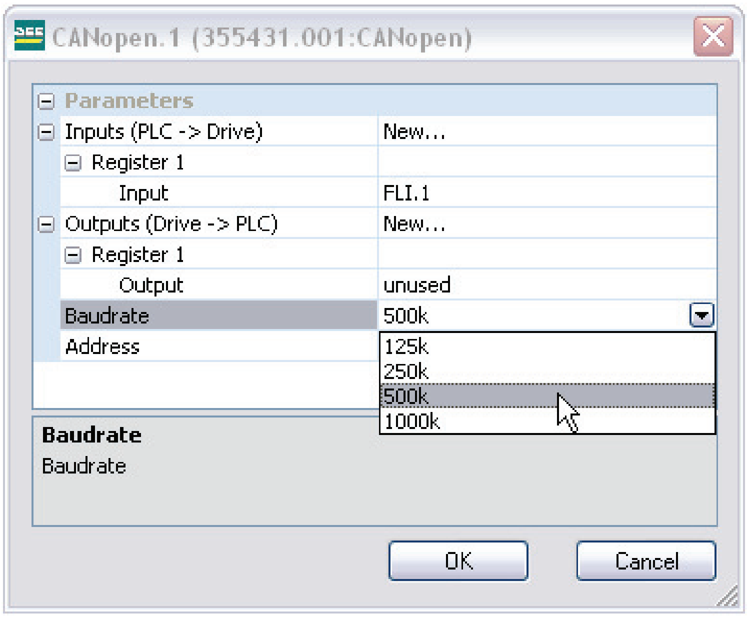
Parameters	
Inputs (PLC -> Drive)	New...
Register 1	
Input	unused
Outputs (Drive -> PLC)	
Register 1	
Output	
Baudrate	
Address	

Input

Input

3. Set up all the input/output registers in a similar way.

4. The Baudrate can be selected to be either 125k, 250k, 500k or 1000k.

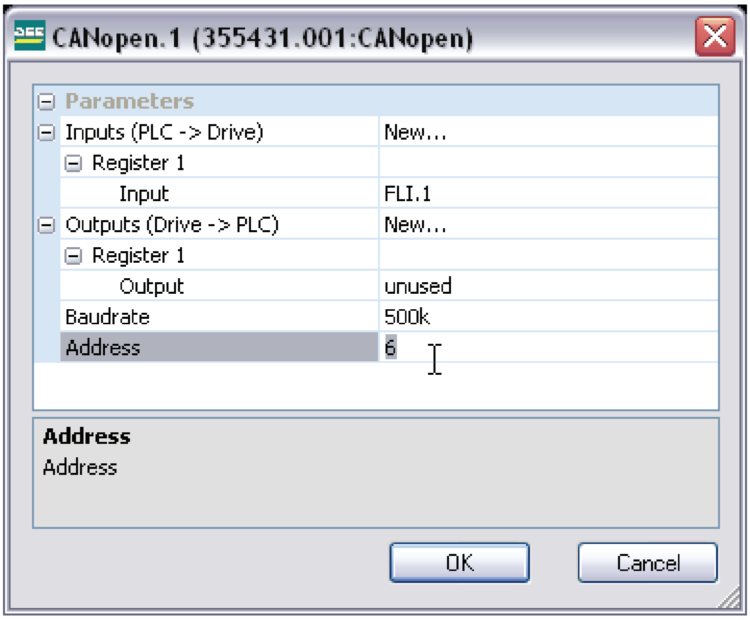


CANopen.1 (355431.001:CANopen)

Parameters	
Inputs (PLC -> Drive)	New...
Register 1	
Input	FLI.1
Outputs (Drive -> PLC)	New...
Register 1	
Output	unused
Baudrate	500k
Address	

Note: *The Baudrate set in DSE 890 will only by used if all switches on the TechCard are set to ON.*

5. The Address can be selected in the range 0 – 127.



CANopen.1 (355431.001:CANopen)

Parameters	
Inputs (PLC -> Drive)	New...
Register 1	
Input	FLI.1
Outputs (Drive -> PLC)	New...
Register 1	
Output	unused
Baudrate	500k
Address	6

Note: *The Address set in DSE 890 will only by used if all switches on the TechCard are set to ON. If the Address is set to zero and the switches on the CANopen TechCard are all set to ON, the option is disabled and will not appear on the network.*

FB Input and Output Data Types

Data Type	Description	Range
LOGIC	Logic	False (F) and True (T)
INTEGER	32-bit signed integer	-2,147,483,648 to 2,147,483,647
VALUE	32-bit fixed point value	-32768.0 to 32767.9999

CANopen Data Types

Data Type	Description	Range
Boolean	8-bit Boolean	False (0x00) and True (0x01)
Integer8	8-bit signed integer	-128 to 127
Integer16	16-bit signed integer	-32,768 to 32,767
Integer32	32-bit signed integer	-2,147,438,648 to 2,147,483,647
Unsigned8	8-bit unsigned integer	0 to 255
Unsigned16	16-bit unsigned integer	0 to 65,535
Unsigned32	32-bit unsigned integer	0 to 4,294,967,295
Real32	32-bit IEEE-754 floating-point value	1.19209290e-38 to 3.4028235e+38

Conversion of DSE Type < > CANopen Type

Each FB Input, regardless of type, can be written to over CANopen using any of the CANopen data types. FB Outputs can be similarly both read and written. The selection of the CANopen data type is not part of the DSE configuration, as it is with other Fieldbusses, but depends on which Sub-Index is used for access.

Sub-Index	CANopen Data Type
1	Boolean
2	Integer8
3	Integer16
4	Integer32
5	Unsigned8
6	Unsigned16
7	Unsigned32
8	Real32

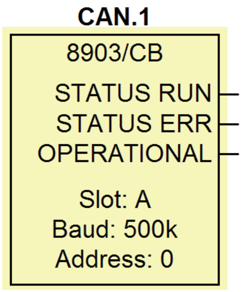
The conversion between the DSE type and the CANopen type is performed automatically (refer to DSE/CANopen Conversion Rules, page 30).

Some recommended PLC type assignments to fieldbus connectors are given in the table below:

Fieldbus Connector	CANopen Type
LOGIC	Boolean
INTEGER	Integer32
VALUE	Real32

CANopen Status Information

The CANopen function block in DSE 890 provides status information about the CANopen network interface.



When online, the *actual* Baudrate or Address in use can be found by clicking the right mouse button over the "Baud" or “Address:” text and selecting **Get**. This may be different to that set in the function block configuration if the switches on the TechCard are not all set in the ON position.

The function block also provides three status outputs that can be wired to: STATUS RUN, STATUS ERR and OPERATIONAL.

For example, the OPERATIONAL output could be ANDed with the motor START causing the drive to stop if the PLC connection is lost.

OPERATIONAL
Logic value: True (T) indicates that the CANopen interface is in the Data Exchange state.

The STATUS RUN and STATUS ERR outputs could be used with the LOGIC::LOOKUP function block to determine a particular state.

STATUS RUN
Enumerated value: Status Run
0: STOPPED
1: PRE-OPERATIONAL
2: OPERATIONAL

STATUS ERR
Enumerated value: Status Run
0: NO ERROR
1: WARNING LI MIT
2 : AUTOBAUD OR LSS
3 : CONTROL EVENT
4 : SYNC. ERROR
5 : BUS OFF

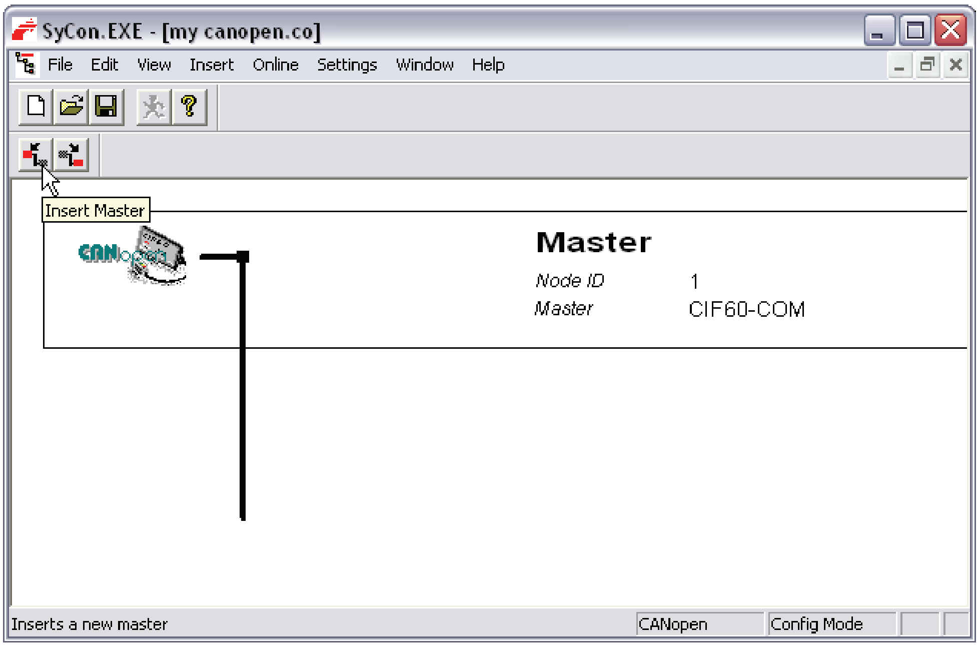
Step 2: Configuring the PLC/SCADA Supervisor

Note: *Our example is shown using a PLC configured using SyCon® System Configurator by Hilscher GmbH (http://www.hilscher.com/) For other systems, refer to the manufacturer's instructions.*

Step 2.1: Creating a Project

1. Copy the EDS file called "ssd890.eds" into the directory called "C:\Program Files\Hilscher\SyCon\Fieldbus\CANopen\EDS".
   Copy the files called "ssd890_d.dib", "ssd890_r.dib" and "ssd890_s.dib" into the directory called "C:\Program Files\Hilscher\SyCon\Fieldbus\CANopen\BMP".

2. Create a project selecting the CANopen. Click on "Insert Master" to add the required master.

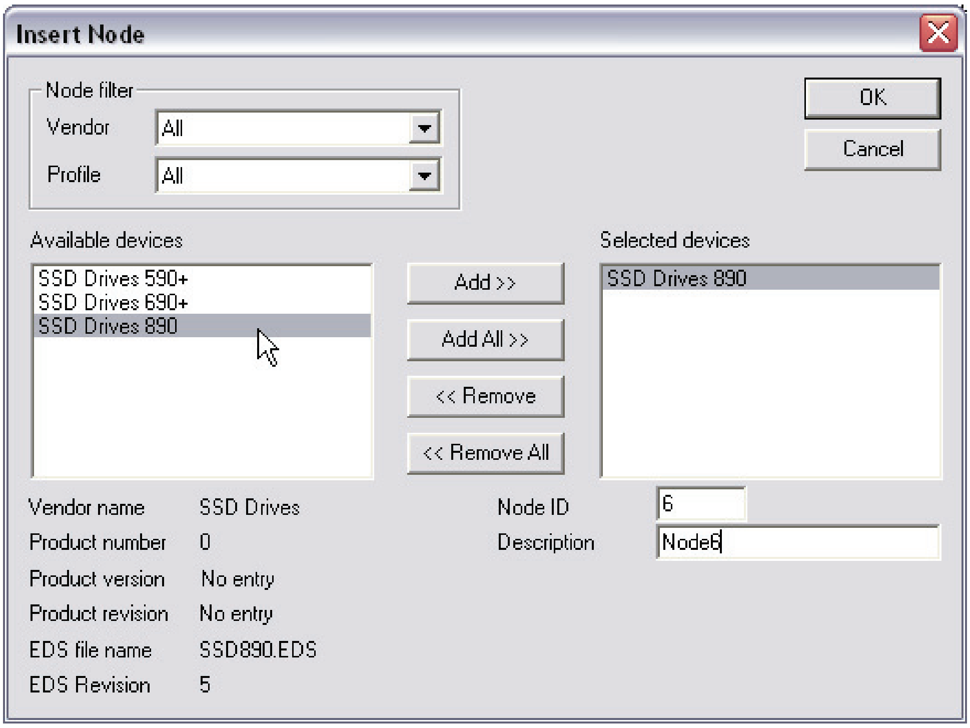

SyCon.EXE - [my canopen.co]
File Edit View Insert Online Settings Window Help
Insert Master
CANopen
Inserts a new master
CANopen
Config Mode

Master

Node ID 1

Master CIF60-COM

3. Click on the Insert Node to add the 890 Drives and assign a Node ID.



Vendor: All

Profile: All

Vendor name SSD Drives

Product number 0

Product version No entry

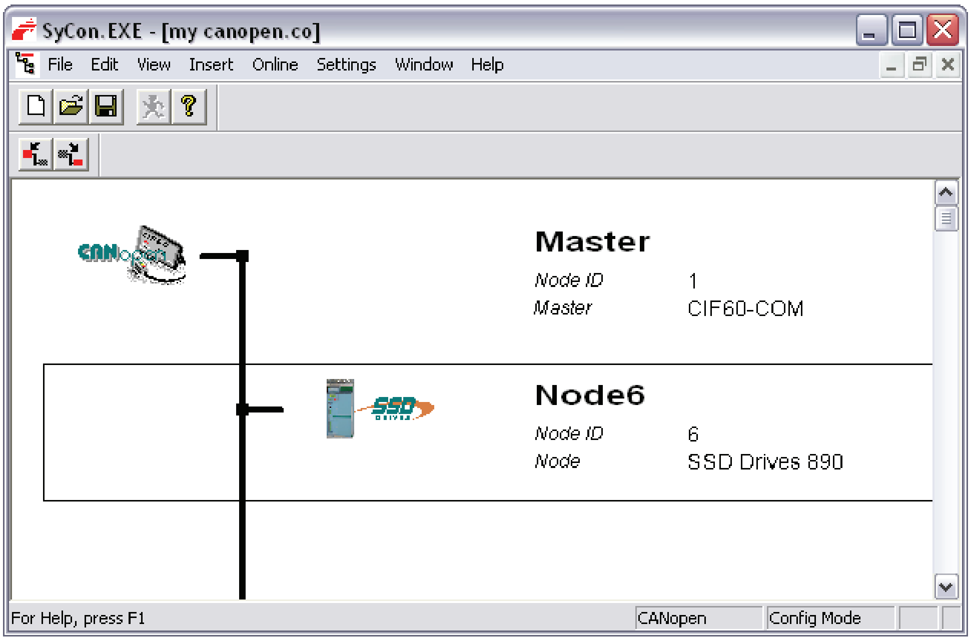
Product revision No entry

EDS file name SSD890.EDS

EDS Revision 5

Node ID 6

Description Node6



SyCon.EXE - [my canopen.co]

Master

Node ID 1

Master CIF60-COM

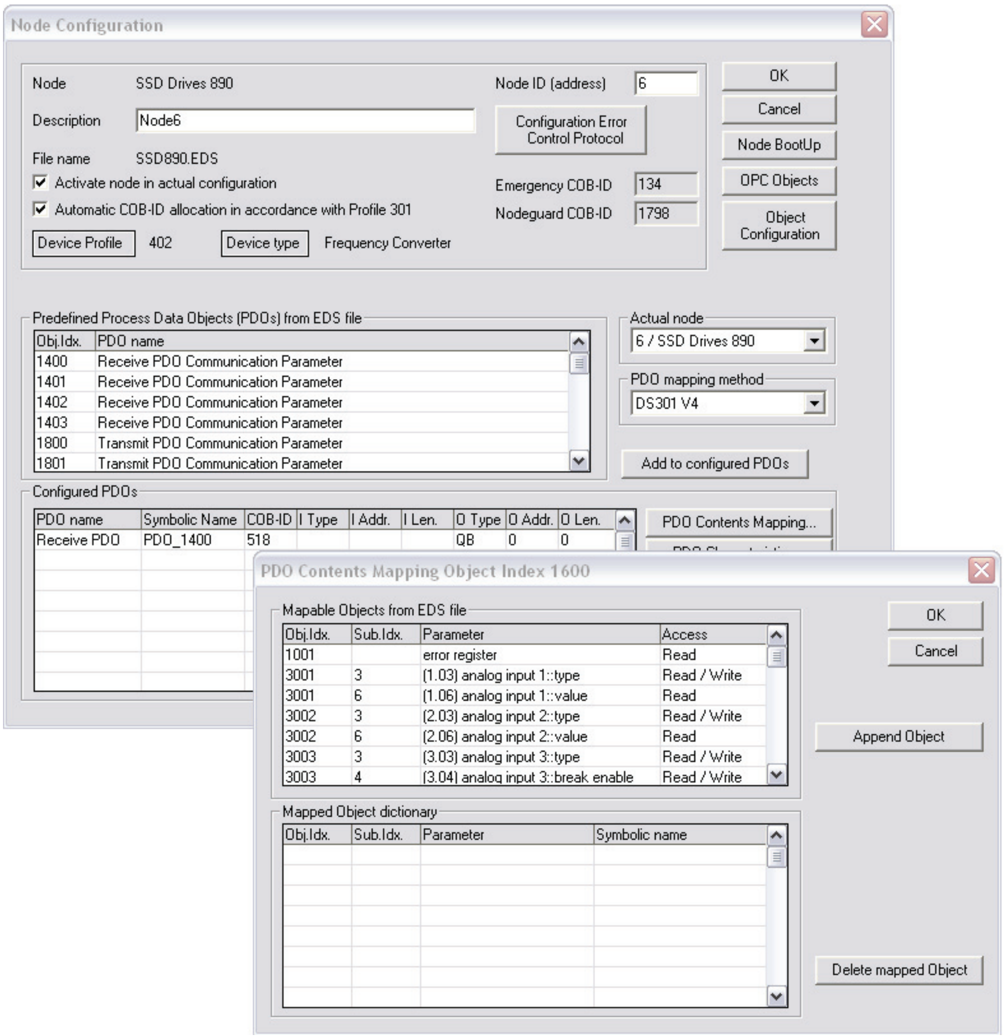
Node6

Node ID 6

Node SSD Drives 890

For Help, press F1 | CANopen | Config Mode

4. Double-click on the created node to allow configuration of the Receive PDOs and Transmit PDOs.



Node Configuration

Node: SSD Drives 890

Description: Node6

File name: SSD890.EDS

☑ Activate node in actual configuration

☑ Automatic COB-ID allocation in accordance with Profile 301

Device Profile: 402

Device type: Frequency Converter

Node ID (address): 6

Emergency COB-ID: 134

Nodeguard COB-ID: 1798

Predefined Process Data Objects (PDOs) from EDS file

Obj.Idx.	PDO name
1400	Receive PDO Communication Parameter
1401	Receive PDO Communication Parameter
1402	Receive PDO Communication Parameter
1403	Receive PDO Communication Parameter
1800	Transmit PDO Communication Parameter
1801	Transmit PDO Communication Parameter

Actual node: 6 / SSD Drives 890

PDO mapping method: DS301 V4

Configured PDOs

PDO name	Symbolic Name	COB-ID	I Type	I Addr.	I Len.	O Type	O Addr.	O Len.
Receive PDO	PDO_1400	518				QB	0	0

PDO Contents Mapping Object Index 1600

Mapable Objects from EDS file

Obj.Idx.	Sub.Idx.	Parameter	Access
1001		error register	Read
3001	3	(1.03) analog input 1::type	Read / Write
3001	6	(1.06) analog input 1::value	Read
3002	3	(2.03) analog input 2::type	Read / Write
3002	6	(2.06) analog input 2::value	Read
3003	3	(3.03) analog input 3::type	Read / Write
3003	4	(3.04) analog input 3::break enable	Read / Write

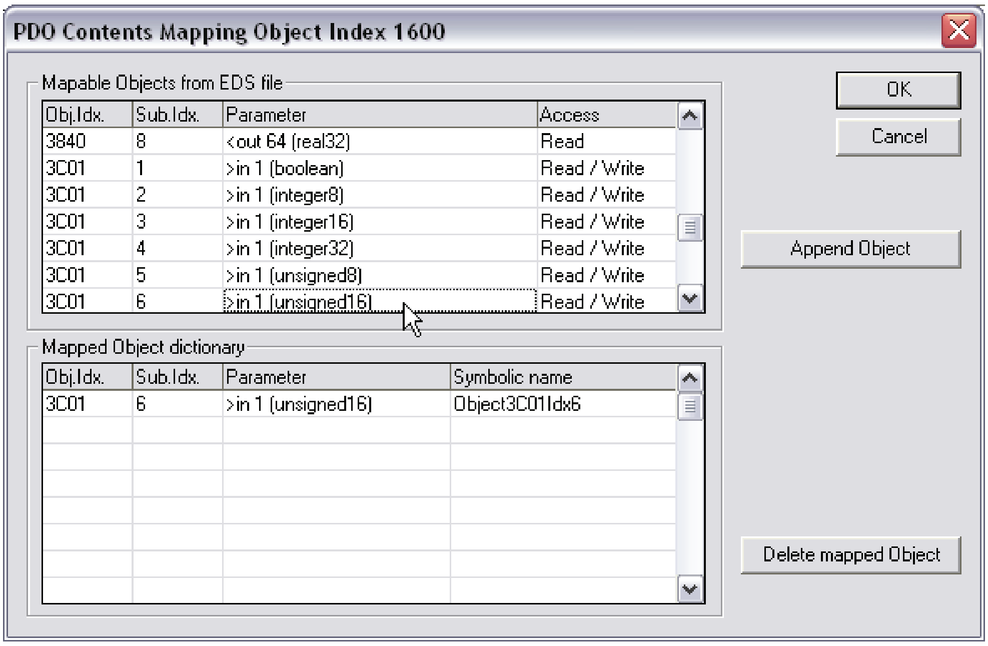
Mapped Object dictionary

Obj.Idx.	Sub.Idx.	Parameter	Symbolic name

5. Two kinds of parameter may be selected:

A : User-defined Input and Output Registers
These are the registers as declared in the DSE 890 configuration.

Note: *Input Registers have Index 3c00h + input register number. Output Registers have Index 3800h + output register number.*



PDO Contents Mapping Object Index 1600

Mapable Objects from EDS file

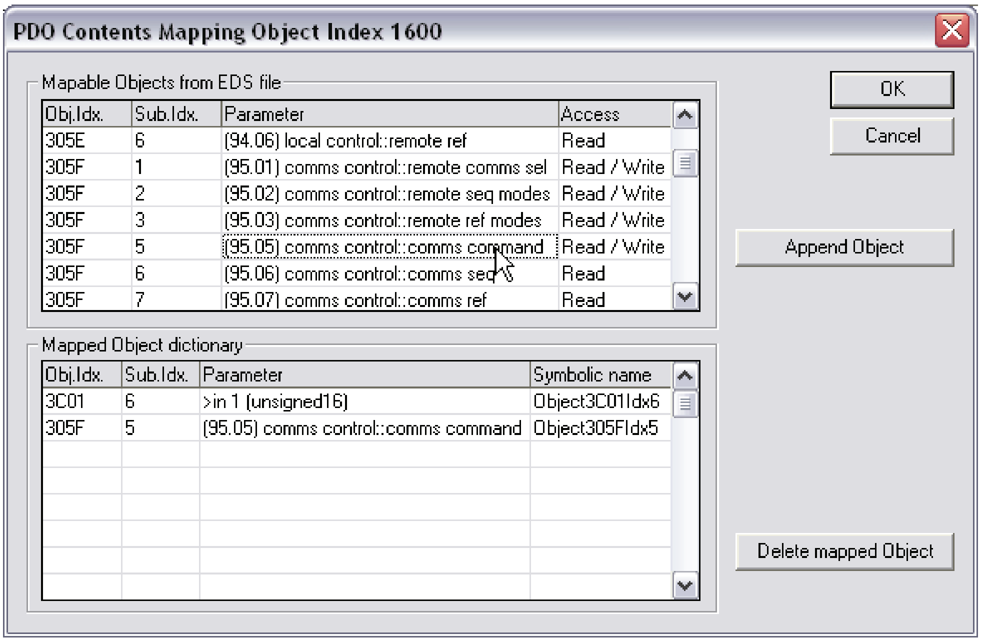
Obj.Idx.	Sub.Idx.	Parameter	Access
3840	8	<out 64 (real32)	Read
3C01	1	>in 1 (boolean)	Read / Write
3C01	2	>in 1 (integer8)	Read / Write
3C01	3	>in 1 (integer16)	Read / Write
3C01	4	>in 1 (integer32)	Read / Write
3C01	5	>in 1 (unsigned8)	Read / Write
3C01	6	>in 1 (unsigned16)	Read / Write

Mapped Object dictionary

Obj.Idx.	Sub.Idx.	Parameter	Symbolic name
3C01	6	>in 1 (unsigned16)	Object3C01Idx6

As described previously, each register can be accessed using any of the CANopen data types by selecting the correct Sub-Index. The example above shows FB Input Register being added to the RxPDO to be sent as Unsigned16 data.

B : Fixed Parameters
These are Drive Parameters that are always present in the 890. They can be found in the Motor Control macro block in the DSE 890 Configuration.



PDO Contents Mapping Object Index 1600

Mapable Objects from EDS file

Obj.Idx.	Sub.Idx.	Parameter	Access
305E	6	(94.06) local control::remote ref	Read
305F	1	(95.01) comms control::remote comms sel	Read / Write
305F	2	(95.02) comms control::remote seq modes	Read / Write
305F	3	(95.03) comms control::remote ref modes	Read / Write
305F	5	(95.05) comms control::comms command	Read / Write
305F	6	(95.06) comms control::comms seq	Read
305F	7	(95.07) comms control::comms ref	Read

Mapped Object dictionary

Obj.Idx.	Sub.Idx.	Parameter	Symbolic name
3C01	6	>in 1 (unsigned16)	Object3C01Idx6
305F	5	(95.05) comms control::comms command	Object305FIdx5

The example above shows the Comms Command parameter (reference 95.5) being added to the RxPDO. The data type is fixed depending on the selected parameter.

CANopen Overview

Introduction

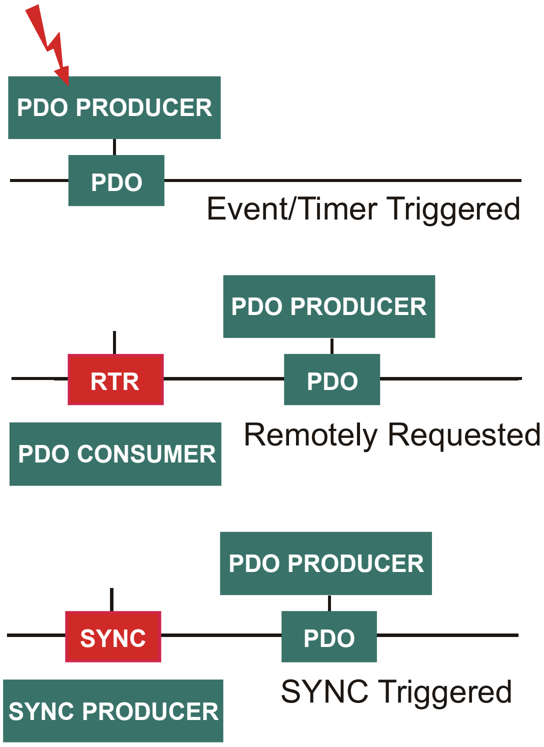
CANopen is a CAN-based higher layer protocol. It was developed as a standardised embedded network with highly flexible configuration capabilities. CANopen was designed for motion-oriented machine control networks, such as handling systems. By now it is used in many various fields, such as medical equipment, off-road vehicles, maritime electronics, public transportation, building automation, etc.

The CANopen application layer and communication profile (EN 50325-4; CiA 301) supports direct access to device parameters and transmission of time-critical process data. The CANopen network management services simplify project design, system integration, and diagnostics. In each decentralised control application, different communication services and protocols are required. CANopen defines all these services and protocols as well as the necessary communication objects.

Process Data Object (PDO)

Process Data Objects (PDOs) are mapped to a single CAN frame using up to 8 bytes of the data field to transmit application objects. Each PDO has a unique identifier and is transmitted by only one node, but it can be received by more than one (producer/consumer communication).

PDO Transmissions



PDO transmissions may be driven by an internal event, by an internal timer, by remote requests and by the Sync message received:

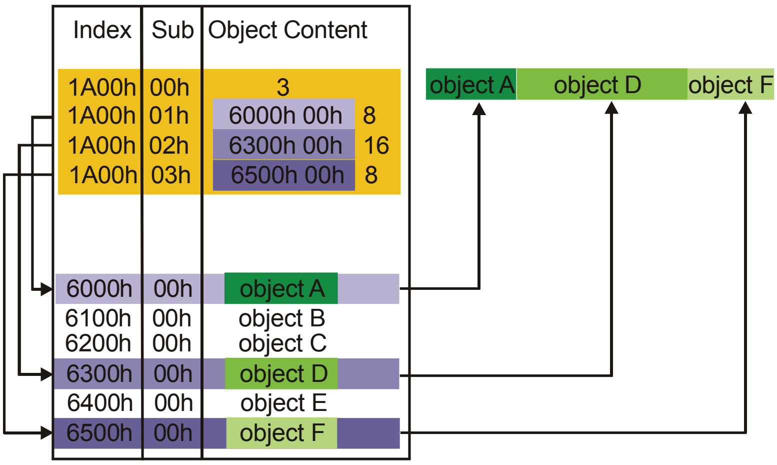
- Event- or timer-driven: An event (specified in the device profile) triggers message transmission. An elapsed timer additionally triggers the periodically transmitting nodes.

- Remotely requested: Another device may initiate the transmission of an asynchronous PDO by sending a remote transmission request (remote frame).

- Synchronous transmission: In order to initiate simultaneous sampling of input values of all nodes, a periodically transmitted Sync message is required. Synchronous transmission of PDOs takes place in cyclic and acyclic transmission mode. Cyclic transmission means that the node waits for the Sync message, after which it sends its measured values. Its PDO transmission type number (1 to 240) indicates the Sync rate it listens to (how many Sync messages the node waits before the next transmission of its values). Acyclically transmitted synchronous PDOs are triggered by a defined application-specific event. The node transmits its values with the next Sync message but will not transmit again until another application-specific event has occurred.

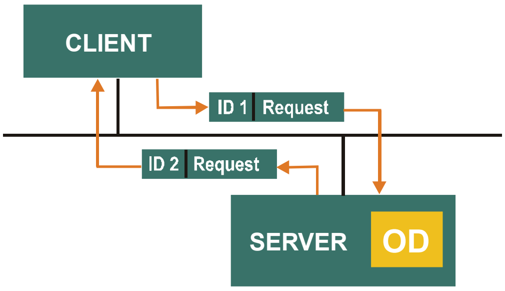
PDO Mapping

The default mapping of application objects as well as the supported transmission mode are described in the Object Dictionary for each PDO. PDO identifiers should have high priority to guarantee a short response time. PDO transmission is not confirmed. The PDO mapping defines which application objects are transmitted within a PDO. It describes the sequence and length of the mapped application objects. A device that supports variable mapping of PDOs must support this during the pre-operational state.



Index	Sub	Object Content
1A00h	00h	3
1A00h	01h	6000h 00h 8
1A00h	02h	6300h 00h 16
1A00h	03h	6500h 00h 8
6000h	00h	object A
6100h	00h	object B
6200h	00h	object C
6300h	00h	object D
6400h	00h	object E
6500h	00h	object F

Service Data Object (SDO)

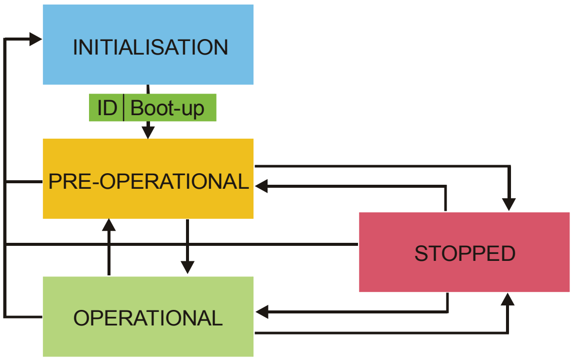


A Service Data Object (SDO) reads from entries or writes to entries of the Object Dictionary. The SDO transport protocol allows transmitting objects of any size. The first byte of the first segment contains the necessary flow control information including a toggle bit to overcome the well-known problem of doubly received CAN frames. The next three byte of the first segment contain index and sub-index of the Object Dictionary entry to be read or written. The last four byte of the first segment are available for user data. The second and the following segments (using the very same CAN identifier) contain the control byte and up to seven byte of user data. The receiver confirms each segment or a block of segments, so that a peer-to-peer communication (client/server) takes place.

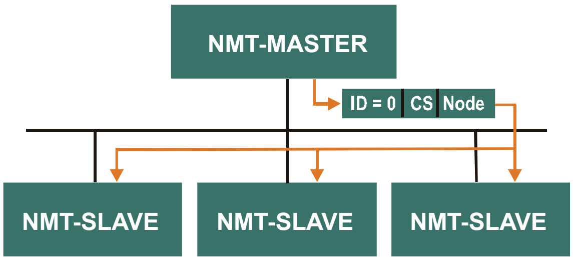
Network Management (NMT)

The Network Management objects include Boot-up message, Heartbeat protocol, and NMT message.

Boot-up message, and Heartbeat protocol are implemented as single CAN frames with 1-byte data field.



NMT Message

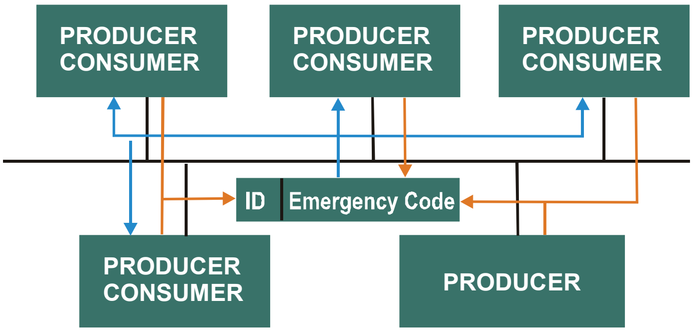


The NMT message is mapped to a single CAN frame with a data length of 2 byte. Its identifier is 0. The first byte contains the command specifier and the second contains the Node-ID of the device that must perform the command (in the case of Node-ID 0 all nodes have to perform the command). The NMT message transmitted by the NMT master forces the nodes to transit to another NMT state. The CANopen state machine specifies the states Initialisation, Pre-Operational, Operational and Stopped. After power-on, each CANopen device is in the state Initialisation and automatically transits to the state Pre-operational. In this state, transmission of SDOs is allowed. If the NMT master has set one or more nodes into the state Operational, they are allowed to transmit and to receive PDOs. In the state Stopped no communication is allowed except that of NMT objects.

Boot-up Message

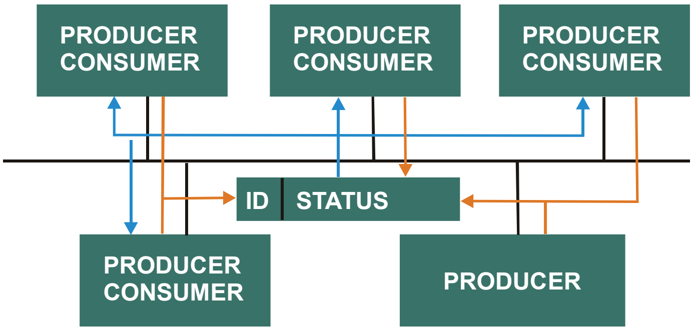
A device sends the Boot-up message to indicate to the NMT master that it has reached the state Pre-operational. This occurs whenever the device initially boots-up but also after a power-out during operation. The Boot-up message has the same identifier as the Heartbeat object, however, its data content is zero.

Synchronisation Object (Sync)



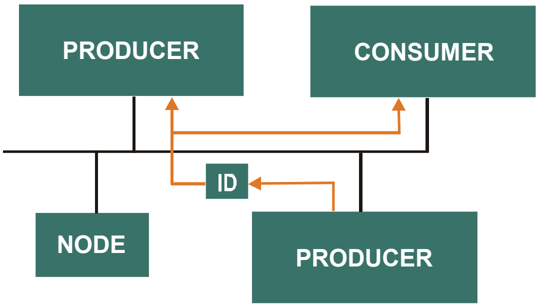
The Sync Object is broadcast periodically by the Sync Producer. The time period between Sync messages is defined by the Communication Cycle Period, which may be reset by a configuration tool to the application devices during the boot-up process. There can be a time jitter in transmission by the Sync Producer due to some other objects with higher prior identifiers or by one frame being transmitted just before the Sync message. The Sync message is mapped to a single CAN frame with the identifier 128 by default. The Sync message does not carry any data.

Emergency Object (Emcy)



The Emergency message is triggered by the occurrence of a device internal error situation and are transmitted from an Emergency producer on the concerned application device. This makes them suitable for interrupt type error alerts. An Emergency message is transmitted only once per 'error event'. As long as no new errors occurs on a device, no further Emergency message can be transmitted. Zero or more Emergency consumers may receive these. The reaction of the Emergency consumer is application-specific. CANopen defines several Emergency Error Codes to be transmitted in the Emergency message, which is a single CAN frame with 8 data byte.

Error Control: Heartbeat Protocol



The Heartbeat protocol is for error control purposes and signals the presence of a node and its state. The Heartbeat message is a periodic message of the node to one or several other nodes. It indicates that the sending node is still working properly.

Besides Heartbeat protocol there exists an old and out-dated error control services, which is called Node and Life Guarding protocol. It is not recommend to use Life Guarding.

Electronic Data Sheet (EDS)

An EDS file is delivered with every CANopen device. It contains all relevant information required by a configuration tool to allow the device to be integrated into a network.

Object Dictionary

The object dictionary represents the complete access to the application program of the device in terms of application data as well as in term of configuration parameters. The object dictionary gains access:

- to all data types used in the device,
- to the communication parameters (to configure the device in terms of communication), and
- to the application data and configuration parameters.

Index	Description
0000h	reserved
0001h - 025Fh	Data types
0260h - 0FFFh	reserved
1000h - 1FFFh	Communication object area
2000h - 5FFFh	Manufacturer specific area
6000h - 9FFFh	Device profile specific area
A000h - BFFFh	Interface profile specific area
C000h - FFFh	reserved

The object dictionary is divided into two parts:

- Communication Profile Area (Index 1000h to 1BFFh)
- Manufacturer Specific Profile Area (Index 2000h to 5FFFh)

Communication Profile Area

*PDO Mapping allowed. [1] Saved in non-volatile memory using Index 1010h.

Index	Sub Index	Name	Type	Attr.	Default	Notes
1000h	00h	device type	Unsigned32	const	00010192h	Frequency converter
1001h*	00h	error register	Unsigned8	ro	00h	
1004h	00h	number of PDOs supported	Unsigned32	ro	00040004h	4 transmit and 4 receive
	01h	number of synch. PDOs	Unsigned32	ro	00040004h	All can be synchronous
	02h	number of asynch. PDOs	Unsigned32	ro	00040004h	All can be asynchronous
1005h	00h	COB-ID SYNC	Unsigned32	rw		[1] SYNC Consumer
1006h	00h	communications cycle period	Unsigned32	rw	00000000h	[1] Used by SYNC watchdog
1008h	00h	manufacturer device name	Vis-String	const	“SSD Drives 890"	Depends on host Drive
1009h	00h	manufacturer hardware version	Vis-String	const	“1.0”	
100Ah	00h	manufacturer software version	Vis-String	const	“1.7”	Main Firmware Version
100Ch	00h	guard time	Unsigned16	rw	0000h	[1]
100Dh	00h	lifetime factor	Unsigned8	rw	00h	[1]
100Fh	00h	Number of SDOs supported	Unsigned32	ro	4	
1014h	00h	COB-ID EMCY	Unsigned32	rw	00h	[1]
1015h	00h	Inhibit Time EMCY	Unsigned32	rw	00h	[1]
1018h		Identity Object	Identity			

Communication Profile Area

*PDO Mapping allowed. [1] Saved in non-volatile memory using Index 1010h.

Index	Sub Index	Name	Type	Attr.	Default	Notes
	00h	Number of entries	Unsigned8	ro	4	
	01h	Vendor ID	Unsigned32	ro	00000098h	SSD Drives
	02h	Product Code	Unsigned32	ro	00000890h	
	03h	Revision Number	Unsigned32	ro		
	04h	Serial Number	Unsigned32	ro		
1201h		Server SDO Parameter	SDO Parameter			
	00h	Number of Entries	Unsigned8	ro	3	
	01h	COB-ID Client -> Server	Unsigned32	rw		[1]
	02h	COB-ID Client -> Server	Unsigned32	rw		[1]
	03h	Node iD of the SDO Client	Unsigned8	rw		[1]
1202h		Server SDO Parameter	SDO Parameter			
	00h	Number of Entries	Unsigned8	ro	3	
	01h	COB-ID Client -> Server	Unsigned32	rw		[1]
	02h	COB-ID Client -> Server	Unsigned32	rw		[1]
	03h	Node iD of the SDO Client	Unsigned8	rw		[1]
1203h		Server SDO Parameter	SDO Parameter			
	00h	Number of Entries	Unsigned8	ro	3	
	01h	COB-ID Client -> Server	Unsigned32	rw		[1]
	02h	COB-ID Client -> Server	Unsigned32	rw		[1]
	03h	Node iD of the SDO Client	Unsigned8	rw		[1]
1400h		receive PDO1 parameter	PDO Parameter			
	00h	largest sub-index supported	Unsigned8	ro	5	
	01h	COB-ID	Unsigned32	rw	200h + nodeID	[1]
	02h	transmission type	Unsigned8	rw	254	[1]
	05h	event timer	Unsigned16	rw	0000h	[1]
1401h		receive PDO2 parameter	PDO Parameter			
	00h	largest sub-index supported	Unsigned8	ro	5	
	01h	COB-ID	Unsigned32	rw	300h + nodeID	[1]
	02h	transmission type	Unsigned8	rw	254	[1]
	05h	event timer	Unsigned16	rw	0000h	[1]
1402h		receive PDO3 parameter	PDO Parameter			
	00h	largest sub-index supported	Unsigned8	ro	5	
	01h	COB-ID	Unsigned32	rw	400h + nodeID	[1]
	02h	transmission type	Unsigned8	rw	254	[1]
	05h	event timer	Unsigned16	rw	0000h	[1]
1403h		receive PDO4 parameter	PDO Parameter			
	00h	largest sub-index supported	Unsigned8	ro	5	
	01h	COB-ID	Unsigned32	rw	500h + nodeID	[1]
	02h	transmission type	Unsigned8	rw	254	[1]
	05h	event timer	Unsigned16	rw	0000h	[1]
1600h		receive PDO1 mapping parameter	PDO Mapping			
	00h	number of mapped objects	Unsigned8	rw	0	[1] Maximum 4
	01h	1st mapped object	Unsigned32	rw	00000000h	[1]
	02h	2nd mapped object	Unsigned32	rw	00000000h	[1]
	03h	3rd mapped object	Unsigned32	rw	00000000h	[1]
	04h	4th mapped object	Unsigned32	rw	00000000h	[1]

Communication Profile Area

*PDO Mapping allowed. [1] Saved in non-volatile memory using Index 1010h.

Index	Sub Index	Name	Type	Attr.	Default	Notes
1601h		receive PDO2 parameter	PDO Parameter			
	00h	number of mapped objects	Unsigned8	rw	0	[1] Maximum 4
	01h	1st mapped object	Unsigned32	rw	00000000h	[1]
	02h	2nd mapped object	Unsigned32	rw	00000000h	[1]
	03h	3rd mapped object	Unsigned32	rw	00000000h	[1]
	04h	4th mapped object	Unsigned32	rw	00000000h	[1]
1602h		receive PDO3 parameter	PDO Parameter			
	00h	number of mapped objects	Unsigned8	rw	0	[1] Maximum 4
	01h	1st mapped object	Unsigned32	rw	00000000h	[1]
	02h	2nd mapped object	Unsigned32	rw	00000000h	[1]
	03h	3rd mapped object	Unsigned32	rw	00000000h	[1]
	04h	4th mapped object	Unsigned32	rw	00000000h	[1]
1603h		receive PDO4 parameter	PDO Parameter			
	00h	number of mapped objects	Unsigned8	rw	0	[1] Maximum 4
	01h	1st mapped object	Unsigned32	rw	00000000h	[1]
	02h	2nd mapped object	Unsigned32	rw	00000000h	[1]
	03h	3rd mapped object	Unsigned32	rw	00000000h	[1]
	04h	4th mapped object	Unsigned32	rw	00000000h	[1]
1800h		transmit PDO1 parameter	PDO Parameter			
	00h	largest sub-index supported	Unsigned8	ro	5	
	01h	COB-ID	Unsigned32	rw	180h + nodeID	[1]
	02h	transmission type	Unsigned8	rw	253	[1]
	03h	inhibit time	Unsigned16	rw	0000h	[1]
	05h	event timer	Unsigned16	rw	0000h	[1]
1801h		transmit PDO2 parameter	PDO Parameter			
	00h	largest sub-index supported	Unsigned8	ro	5	
	01h	COB-ID	Unsigned32	rw	280h + nodeID	[1]
	02h	transmission type	Unsigned8	rw	253	[1]
	03h	inhibit time	Unsigned16	rw	0000h	[1]
	05h	event timer	Unsigned16	rw	0000h	[1]
1802h		transmit PDO3 parameter	PDO Parameter			
	00h	largest sub-index supported	Unsigned8	ro	5	
	01h	COB-ID	Unsigned32	rw	380h + nodeID	[1]
	02h	transmission type	Unsigned8	rw	253	[1]
	03h	inhibit time	Unsigned16	rw	0000h	[1]
	05h	event timer	Unsigned16	rw	0000h	[1]
1803h		transmit PDO4 parameter	PDO Parameter			
	00h	largest sub-index supported	Unsigned8	ro	5	
	01h	COB-ID	Unsigned32	rw	480h + nodeID	[1]
	02h	transmission type	Unsigned8	rw	253	[1]
	03h	inhibit time	Unsigned16	rw	0000h	[1]
	05h	event timer	Unsigned16	rw	0000h	[1]

Communication Profile Area

*PDO Mapping allowed. [1] Saved in non-volatile memory using Index 1010h.

Index	Sub Index	Name	Type	Attr.	Default	Notes
1A00h		transmit PDO1 mapping parameter	PDO Mapping			
	00h	number of mapped objects	Unsigned8	rw	0	[1] Maximum 4
	01h	1st mapped object	Unsigned32	rw	00000000h	[1]
	02h	2nd mapped object	Unsigned32	rw	00000000h	[1]
	03h	3rd mapped object	Unsigned32	rw	00000000h	[1]
	04h	4th mapped object	Unsigned32	rw	00000000h	[1]
1A01h		transmit PDO2 parameter	PDO Parameter			
	00h	number of mapped objects	Unsigned8	rw	0	[1] Maximum 4
	01h	1st mapped object	Unsigned32	rw	00000000h	[1]
	02h	2nd mapped object	Unsigned32	rw	00000000h	[1]
	03h	3rd mapped object	Unsigned32	rw	00000000h	[1]
	04h	4th mapped object	Unsigned32	rw	00000000h	[1]
1A02h		transmit PDO3 parameter	PDO Parameter			
	00h	number of mapped objects	Unsigned8	rw	0	[1] Maximum 4
	01h	1st mapped object	Unsigned32	rw	00000000h	[1]
	02h	2nd mapped object	Unsigned32	rw	00000000h	[1]
	03h	3rd mapped object	Unsigned32	rw	00000000h	[1]
	04h	4th mapped object	Unsigned32	rw	00000000h	[1]
1A03h		transmit PDO4 parameter	PDO Parameter			
	00h	number of mapped objects	Unsigned8	rw	0	[1] Maximum 4
	01h	1st mapped object	Unsigned32	rw	00000000h	[1]
	02h	2nd mapped object	Unsigned32	rw	00000000h	[1]
	03h	3rd mapped object	Unsigned32	rw	00000000h	[1]
	04h	4th mapped object	Unsigned32	rw	00000000h	[1]

Manufacturer Specific Profile Area

Index	Sub Index	Name	Type	Attr.	Default	Notes
2014h		PDO1 transmit mask				
	00h	number of entries	Unsigned8	ro	2	
	01h	mask low	Unsigned32	rw	FFFFFFFFh	[1]
	02h	mask low	Unsigned32	rw	FFFFFFFFh	[1]
2015h		PDO2 transmit mask				
	00h	number of entries	Unsigned8	ro	2	
	01h	mask low	Unsigned32	rw	FFFFFFFFh	[1]
	02h	mask low	Unsigned32	rw	FFFFFFFFh	[1]
2016h		PDO3 transmit mask				
	00	number of entries	Unsigned8	ro	2	
	01h	mask low	Unsigned32	rw	FFFFFFFFh	[1]
	02h	mask low	Unsigned32	rw	FFFFFFFFh	[1]
2017h		PDO4 transmit mask				
	00h	number of entries	Unsigned8	ro	2	
	01h	mask low	Unsigned32	rw	FFFFFFFFh	
	02h	mask low	Unsigned32	rw	FFFFFFFFh	

Manufacturer Specific Profile Area

Area 3000h-37FFh corresponds to the fixed parameter area of 890

Index	Sub Index	Name	Data type	Access	Default	Notes
3001h		Analog input 1				
	06h	(1.6) Analog input 1::Value	Integer16	rw	0	
…	…	…	…	…	00000000	…
3015h		Fluxing				
	0Ah	(21.10) Fluxing::User freq. 1	Unsigned32	rw	41200000	
	0Bh	(21.11) Fluxing::User voltage 1	Unsigned32	rw	41200000	
	0Ch	(21.12) Fluxing::User freq. 2	Unsigned32	rw	41A00000	
	0Dh	(21.13) Fluxing::User voltage 2	Unsigned32	rw	41A00000	
	0Eh	(21.14)Fluxing::User freq. 3	Unsigned32	rw	41F00000	
	0Fh	(21.15) Fluxing::User voltage 3	Unsigned32	rw	41F00000	
…	…	…	…	…	…	…

Area 3800h-3BFFh corresponds to user-configurable Output Registers

Index	Sub Index	Name	Data type	Access	Default	Notes
3801h		< out 1				
	01h	< out 1 (boolean)	Unsigned8	ro	no default	0 = FALSE, 1 = TRUE
	02h	< out 1 (integer8)	Integer8	ro	no default	
	03h	< out 1 (integer16)	Integer16	ro	no default	
	04h	< out 1 (integer32)	Integer32	ro	no default	
	05h	< out 1 (unsigned8)	Unsigned8	ro	no default	
	06h	< out 1 (unsigned16)	Unsigned16	ro	no default	
	07h	< out 1 (unsigned32)	Unsigned32	ro	no default	
	08h	< out 1 (real32)	Real32	ro	no default	
…	…	…	…	…	…	…
3802h		< out 2				
	01h	< out 2 (boolean)	Unsigned8	ro	no default	0 = FALSE, 1 = TRUE
…	…	…	…	…	…	…

Area 3C00h-3FFFh corresponds to user-configurable Input Registers

Index	Sub Index	Name	Data type	Access	Default	Notes
3C01h		> in 1				
	01h	> in 1 (boolean)	Unsigned8	rw	no default	0 = FALSE, 1 = TRUE
	02h	> in 1 (integer8)	Integer8	rw	no default	
	03h	> in 1 (integer16)	Integer16	rw	no default	
	04h	> in 1 (integer32)	Integer32	rw	no default	
	05h	> in 1 (unsigned8)	Unsigned8	rw	no default	
	06h	> in 1 (unsigned16)	Unsigned16	rw	no default	
	07h	> in 1 (unsigned32)	Unsigned32	rw	no default	
	08h	> in 1 (real32)	Real32	rw	no default	
…	…	…	…	…	…	…
3C02h		> in 2				
	01h	> in 2 (boolean)	Unsigned8	rw	no default	0 = FALSE, 1 = TRUE
…	…	…	…	…	…	…

External Control of the Drive

Communications Command

When sequencing is in the Remote Comms mode, the sequencing of the Drive is controlled by writing to the COMMS COMMAND (PREF 95.09). If the Comms TIMEOUT feature is to be used, the hidden parameter (PREF 95.05) should be written to using a communications interface. This hidden parameter has the same format as COMMS COMMAND.

The COMMS COMMAND parameter is a 16-bit word based on standard fieldbus drive profiles. Some bits are not implemented in this release (see "Supported" column of the table below).

Bit	Name	Description	Supported	Required Value
0	Switch On	OFF1 Operational	√	
1	(Not) Disable Voltage	OFF2 Coast Stop	√	
2	(Not) Quick Stop	OFF3 Fast Stop	√	
3	Enable Operation		√	
4	Enable Ramp Output	=0 to set ramp output to zero		1
5	Enable Ramp	=0 to hold ramp		1
6	Enable Ramp Input	=0 to set ramp input to zero		1
7	Reset Fault	Reset on 0 to 1 transition	√	
8				0
9				0
10	Remote	=1 to control remotely		1
11				0
12				0
13				0
14				0
15				0

Switch On

Replaces the RUN FWD, RUN REV and NOT STOP parameters of the SEQUENCING LOGIC function block. When Set (=1) is the same as :

RUN FWD	=	TRUE
RUN REV	=	FALSE
NOT STOP	=	FALSE

When Cleared (= 0) is the same as :

RUN FWD	=	FALSE
RUN REV	=	FALSE
NOT STOP	=	FALSE

(Not) Disable Voltage

ANDed with the NOT COAST STOP parameter of the SEQUENCING LOGIC function block. When both Set (=1) is the same as:

NOT COAST STOP = TRUE

When either or both Cleared (= 0) is the same as :

NOT COAST STOP = FALSE

(Not) Quick Stop

ANDed with the NOT FAST STOP parameter on the SEQUENCING LOGIC function block. When both Set (=1) is the same as:

NOT FAST STOP = TRUE

When either or both Cleared (= 0) is the same as :

NOT FAST STOP = FALSE

Enable Operation

ANDed with the DRIVE ENABLE parameter on the SEQUENCING LOGIC function block. When both Set (=1) is the same as:

DRIVE ENABLE = TRUE

When either or both Cleared (= 0) is the same as :

DRIVE ENABLE = FALSE

Enable Ramp Output, Enable Ramp, Enable Ramp Input

Not implemented. The state of these bits must be set (=1) to allow this feature to be added in the future.

Reset Fault

Replaces the REM TRIP RESET parameter on the SEQUENCING LOCIC function block. When Set (=1) is the same as:

REM TRIP RESET = TRUE

When Cleared (= 0) is the same as :

REM TRIP RESET = FALSE

Remote

Not implemented. It is intended to allow the PLC to toggle between local and remote. The state of this must be set (=1) to allow this feature to be added in the future.

Example Commands

047F hexadecimal to RUN

047E hexadecimal to STOP

Communications Status

The COMMS STATUS parameter (PREF 95.08) in the COMMS CONTROL function block monitors the sequencing of the Drive. It is a 16-bit word based on standard fieldbus drive profiles. Some bits are not implemented in the initial release and are set to 0 (see “Supported” column of the table below).

Bit	Name	Description	Supported
0	Ready To Switch On		√
1	Switched On	Ready for operation (refer control bit 0)	√
2	Operation Enabled	(refer control bit 3)	√
3	Fault	Tripped	√
4	(Not) Voltage Disabled	OFF 2 Command pending	√
5	(Not) Quick Stop	OFF 3 Command pending	√
6	Switch On Disable	Switch On Inhibited	√
7	Warning		
8	SP / PV in Range		
9	Remote	= 1 if Drive will accept Command Word	√
10	Setpoint Reached	= 1 if not ramping	√
11	Internal Limit Active	= 1 if current limit active or speed loop is in torque limit	√
12			
13			
14			
15			

Ready To Switch On

Same as the SWITCH ON ENABLE output parameter of the SEQUENCING LOGIC function block.

Switched On

Same as the SWITCHED ON output parameter of the SEQUENCING LOGIC function block.

Operation Enabled

Same as the RUNNING output parameter of the SEQUENCING LOGIC function block.

Fault

Same as the TRIPPED output parameter of the SEQUENCING LOGIC function block.

(Not) Voltage Disabled

If in Remote Comms mode, this is the same as Bit 1 of the COMMS COMMAND parameter. Otherwise it is the same as the NOT COAST STOP input parameter of the SEQUENCING LOGIC function block.

(Not) Quick Stop

If in Remote Comms mode, this is the same as Bit 2 of the COMMS COMMAND parameter. Otherwise it is the same as the NOT FAST STOP input parameter of the SEQUENCING LOGIC function block.

Switch On Disable

Set (=1) only when in START DISABLED state, refer to **Error! Reference source not found.**.

Remote

This bit is set (= 1) if the Drive is in Remote mode **AND** the parameter REMOTE COMMS SEL of the COMMS CONTROL function block is Set (= 1).

Setpoint Reached

This bit is set (=1) if the Reference Ramp is not ramping.

Internal Limit Active

This bit is set (=1) if, while in vector control mode, the speed limit has reached the torque limit; or, while in Volts/Hz mode, the open loop current limit is active.

DSE/CANopen Conversion Rules

The rules governing the conversion between 890 data types and CANopen data types are given below Note carefully that some conversions will result in rounding, limiting and truncation of the original value. Certain conversions are not supported, however if used then data space will be allocated in the buffer, but a data value of zero will be returned.

LOGIC Type Connector

	Data from CANopen	Data to 890
From BOOLEAN to LOGIC	False True	False True
From REAL 32 to LOGIC	Zero Non-zero	False True
From INTEGER 8 to LOGIC	Zero Non-zero	False True
From INTEGER 16 to LOGIC	Zero Non-zero	False True
From INTEGER 32 to LOGIC	Zero Non-zero	False True
From UNSIGNED 8 to LOGIC	Zero Non-zero	False True
From UNSIGNED 16 to LOGIC	Zero Non-zero	False True
From UNSIGNED 32 to LOGIC	Zero Non-zero	False True

	Data from 890	Data to CANopen
From LOGIC to BOOLEAN	False True	False True
From LOGIC to REAL 32	False True	0.0 1.0
From LOGIC to INTEGER 8	False True	0 1
From LOGIC to INTEGER 16	False True	0 1
From LOGIC to INTEGER 32	False True	0 1
From LOGIC to UNSIGNED 8	False True	0 1
From LOGIC to UNSIGNED 16	False True	0 1
From LOGIC to UNSIGNED 32	False True	0 1

INTEGER Type Connector

	Data from CANopen	Data to 890
From BOOLEAN to INTEGER	False True	0x0000 0000 0x0000 0001
From INTEGER 8 to INTEGER	-128 to 127	-128 to 127
From INTEGER 16 to INTEGER	-32,768 to 32,767	-32,768 to 32,767
From INTEGER 32 to INTEGER	-2,147,483,648 to 2,147,483,547	-2,147,483,648 to 2,147,483,547
From UNSIGNED 8 to INTEGER	0 to 255	0 to 255
From UNSIGNED 16 to INTEGER	0 to 65,535	0 to 65,535
From UNSIGNED 32 to INTEGER	0 to 4,294,967,295	0 to 2,147,483,647 limits apply
From REAL 32 to INTEGER	32-bit IEEE floating-point	-2,147,483,648 to 2,147,483,547 Fractional part rounded

	Data from 890	Data to CANopen
From INTEGER to BOOLEAN	Zero Non-zero	True False
From INTEGER to REAL 32	-2,147,483,648 to 2,147,483,647	32-bit IEEE floating-point
From INTEGER to INTEGER 8	-2,147,483,648 to 2,147,483,647	-128 to 127 limits apply
From INTEGER to INTEGER 16	-2,147,483,648 to 2,147,483,647	-32768 to 32767 limits apply
From INTEGER to INTEGER 32	-2,147,483,648 to 2,147,483,647	-2,147,483,648 to 2,147,483,647
From INTEGER to UNSIGNED 8	-2,147,483,648 to 2,147,483,647	0 to 255 limits apply
From INTEGER to UNSIGNED 16	-2,147,483,648 to 2,147,483,647	0 to 65,535 limits apply
From INTEGER to UNSIGNED 32	-2,147,483,648 to 2,147,483,647	0 to 2,147,483,647 limits apply

VALUE Type Connector

	Data from CANopen	Data to 890
From BOOLEAN to VALUE	False True	0.0 1.0
From REAL 32 to VALUE	32-bit IEEE floating-point	-32,768.0 to 32,767.9999
From INTEGER 8 to VALUE	-128 to 127	-128.0 to 127.0
From INTEGER 16 to VALUE	-32,768 to 32,767	-32,768.0 to 32,767.0
From INTEGER 32 to VALUE	-2,147,483,648 to 2,147,483,547	-32,768.0 to 32,767.0 limits apply
From UNSIGNED 8 to VALUE	0 to 255	0.0 to 255.0
From UNSIGNED 16 to VALUE	0 to 65,535	0.0 to 32,767.0 limits apply
From UNSIGNED 32 to VALUE	0 to 4,294,967,295	0.0 to 32,767.0 limits apply

	Data from 890	Data to CANopen
From VALUE to BOOLEAN	Zero Non-zero	False True
From VALUE to REAL 32	-32,768.0 to 32,767.9999	32-bit IEEE floating-point
From VALUE to INTEGER 8	-32,768.0 to 32,767.9999	-128 to 127 limits apply/ rounding applies
From VALUE to INTEGER 16	-32,768.0 to 32,767.9999	-32,768 to 32,767 limits apply/ rounding applies
From VALUE to INTEGER 32	-32,768.0 to 32,767.9999	-32768 to 32,767 limits apply/ rounding applies
From VALUE to UNSIGNED 8	-32,768.0 to 32,767.9999	0 to 255 limits apply/ rounding applies
From VALUE to UNSIGNED 16	-32,768.0 to 32,767.9999	0 to 32767 limits apply/ rounding applies
From VALUE to UNSIGNED 32	-32,768.0 to 32,767.9999	0 to 32767 limits apply/ rounding applies

Disposal

This product contains materials which are consignable waste under the Special Waste Regulations 1996 which complies with the EC Hazardous Waste Directive - Directive 91/689/EEC.

We recommend you dispose of the appropriate materials in accordance with the valid environmental control laws. The following table shows which materials can be recycled and which have to be disposed of in a special way.

Material	Recycle	Disposal
metal	yes	no
plastics material	yes	no
printed circuit board	no	yes

The printed circuit board should be disposed of in one of two ways:

1. High temperature incineration (minimum temperature 1200°C) by an incinerator authorised under parts A or B of the Environmental Protection Act
2. Disposal in an engineered land fill site that is licensed to take aluminium electrolytic capacitors. Do not dispose of in a land fill site set aside for domestic waste.

Packaging

During transport our products are protected by suitable packaging. This is entirely environmentally compatible and should be taken for central disposal as secondary raw material.

ISS.	MODIFICATION	ECN No.	DATE	DRAWN	CHK'D
1	Initial Issue (HA469262U001)	17320	13/03/06	CM	KJ
2	Small amendments. Company name change.	19768	09/01/07	CM	KJ

FIRST USED ON	MODIFICATION RECORD CANopen Communications Interface	
Parker SSD DRIVES	DRAWING NUMBER ZZ469262C001	SHT. 1 OF 1