A8NECT_2P

Instruction manual
EtherCAT Option Board

SDD-7688-014
Version 1.07

Important user information

This document is intended to provide a good understanding of the functionality offered by the A8NECT_2P EtherCAT Option Board. The document only describes the features that are specific to the option board. For general information regarding the inverter, consult the inverter design guides.

The reader of this document is expected to be familiar with high level software design, and communication systems in general. The use of advanced EtherCAT-specific functionality may require in-depth knowledge in EtherCAT networking internals and/or information from the official EtherCAT specifications. In such cases, the people responsible for the implementation of this product should either obtain the EtherCAT specification to gain sufficient knowledge or limit their implementation in such a way that this is not necessary.

Liability

Every care has been taken in the preparation of this manual. Please inform HMS Industrial Networks AB of any inaccuracies or omissions. The data and illustrations found in this document are not binding. We, HMS Industrial Networks AB, reserve the right to modify our products in line with our policy of continuous product development.

The information in this document is subject to change without notice and should not be considered as a commitment by HMS Industrial Networks AB. HMS Industrial Networks AB assumes no responsibility for any errors that may appear in this document.

There are many applications of this product. Those responsible for the use of this device must ensure that all the necessary steps have been taken to verify that the applications meet all performance and safety requirements including any applicable laws, regulations, codes, and standards.

HMS Industrial Networks AB will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features, timing, or functional side effects found outside the documented scope of this product. The effects caused by any direct or indirect use of such aspects of the product are undefined, and may include e.g. compatibility issues and stability issues.

The examples and illustrations in this document are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular implementation, HMS Industrial Networks AB cannot assume responsibility for actual use based on these examples and illustrations.

Intellectual property rights

HMS Industrial Networks AB has intellectual property rights relating to technology embodied in the product described in this document. These intellectual property rights may include patents and pending patent applications in the US and other countries.

Trademark acknowledgements

Anybus® is a registered trademark of HMS Industrial Networks AB.

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

All other trademarks are the property of their respective holders.

Warning:	This is a class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.
ESD Note:	This product contains ESD (Electrostatic Discharge) sensitive parts that may be damaged if ESD control procedures are not followed. Static control precautions are required when handling the product. Failure to observe this may cause damage to the product.

Thank you for choosing this Mitsubishi Inverter plug-in option for the Mitsubishi FR-A800/FR-F800 Series Inverter. This Instruction Manual gives handling information and precautions for use of this equipment. Incorrect handling may cause an unexpected failure or damage. In order to ensure optimal performance, please read this manual carefully prior to use of the equipment.

Please forward this manual to the end user of the equipment.

This section is specifically about safety matters

Do not attempt to install, operate, maintain or inspect this product until you have read through this Instruction Manual and any related documents carefully, and can use the equipment correctly. Do not use this product until you have a full working knowledge of the equipment, safety information and instructions.

In this Instruction Manual, the safety instruction levels are classified into “WARNING” and “CAUTION” levels.

⚠ WARNING	Assumes that incorrect handling may cause hazardous conditions, resulting in death or severe injury.
⚠ CAUTION	Assumes that incorrect handling may cause hazardous conditions, resulting in moderate or slight injury, or may cause physical damage only.

Please note that even the ⚠ CAUTION level may lead to a serious consequence depending on conditions. Please be sure to follow the instructions of both levels as they are critical to personnel safety.

SAFETY INSTRUCTIONS

1. Electric shock prevention

⚠ WARNING

• Do not open any cover on the inverter while power is on or while the inverter is running, as an electrical shock may result.

• Do not operate the inverter with any cover or wiring cover removed, as accidental contact with exposed high-voltage terminals and internal components may occur, resulting in an electrical shock.

• If power is off do not remove any cover except when necessary for wiring or periodic inspection. While any cover is removed, accidental contact with exposed high-voltage terminals and internal components may occur, resulting in an electrical shock.

• Prior to starting wiring or inspection, confirm that input power to the inverter has been switched off via observation of the inverter's display panel. Additionally, wait for at least 10 minutes after removal of input power, and then confirm that all residual voltage has been dissipated by using a voltage meter. Internal DC bus capacitors may contain high voltages for several minutes after removal of input power, resulting in a dangerous situation should anything come into contact with them.

• All personnel involved in the installation or inspection of this equipment should be fully competent to perform the required work.

• Always install plug-in options prior to wiring main power.

• Do not touch the plug-in option with wet hands.

• Do not subject the cables to scratches, excessive stress, heavy loads or pinching.

2. Injury prevention

⚠ CAUTION

• To prevent explosions or similar damage, apply only the voltages specified in the instruction manual to each terminal.

• To prevent explosions or similar damage, ensure that all cables are properly connected to the correct terminals.

• To prevent explosions or similar damage, observe all wiring polarity indicators.

• To prevent burns from hot components, do not touch the inverter while power is on, or for some time after power is removed.

• Do not subject the cables to scratches, excessive stress, heavy loads or pinching.

3. Additional Instructions

Please note the following points to prevent equipment damage, injury or electrical shock.

1. Transportation and mounting

⚠ CAUTION

- Do not install or operate the plug-in option if it is damaged or has parts missing.
- Do not stand or rest heavy objects on the equipment.
- Check that the mounting orientation is correct.
- Prevent conductive items such as screws and metal fragments, or flammable substances such as oil from entering the inverter.

2. Trial run

⚠ CAUTION

- To prevent unexpected equipment movement, confirm and adjust all required parameters prior to starting operation.

3. Usage

⚠ WARNING

- Do not modify the equipment.
- Do not remove any inverter or option parts unless specifically instructed to do so in this manual.

⚠ CAUTION

- Performing a “parameter clear” or “all parameter clear” will reset all inverter parameters to their factory default settings.

After performing one of these operations, remember to reenter any custom parameter values prior to starting operation.

- To prevent damage from electric discharge, always touch a grounded piece of metal prior to touching any equipment.

4. Maintenance, inspection and parts replacement

⚠ CAUTION
• Do not perform hi-pot tests on the equipment.

5. Disposal

⚠ CAUTION
• Contact the local or state environmental agency in your area for details on the disposal of electrical components and packaging.

6. General instruction

For clarity purposes, illustrations in this manual may be drawn with covers or safety guards removed. Ensure all covers and safety guards are properly installed prior to starting operation.

Table of Contents

P. About this document

For more information, documentation etc., please visit the Mitsubishi Electric web site, see section P.2 below.

P.1 Related Documents

Beckhoff. (u.d.). EtherCAT Slave Controller Section II - Register Description. *2.7*.
Mitsubishi Electric. (u.d.). Installation guideline of Mitsubishi inverter drive. -.
Mitsubishi Electric. (u.d.). Instruction manual of Mitsubishi inverter drive. -.

P.2 Download

The following web sites are available for downloads:

Item	Web site	Region
ESI-file	http://www.mitsubishi-automation.com	EU
ESI-file	https://us.mitsubishielectric.com/fa/en/my-mitsubishi/search-results?q=@uri&f=@owsPrimaryContentType=Network Configuration Files&h=	Americas
Firmware manager II	http://www.anybus.com/support/support.asp?PID=410&ProductType=Support%20Tools	-

P.3 Document History

Version	History	Author	Date
1.00	Initial version	JoG	2015-05-06
1.01	Updates after first review	DaR, JoG	2015-06-18
1.02	Updates after second review	DaR, JoG	2015-07-06
1.03	Minor corrections	DaR	2015-07-16
1.04	Section 5.8.2.20 and 5.8.2.21: Added information that sub-index 5 is only present when user process data mapping is enabled.	JoG	2015-08-18
1.05	Added information about support also for FR-F800.	DaR	2015-09-08
1.06	Section 4.2: Added information that parameter 1305 can't be changed using SDO download requests. Sections 5.8.2.10, 5.8.2.15, 5.8.2.17, 5.8.2.26.10, 5.8.2.26.16, 5.8.2.26.19, 5.8.2.26.20, 5.8.2.26.21: Added information that the following objects aren't supported on the FR-F800 drives: 0x1604, 0x1A04, 0x604A, 0x6071, 0x6077, 0x6087 and 0x6088.	JoG	2015-11-19
1.07	Added section 1.6 – Product Identification Section 2.4: Updated overlay picture.	DaR	2015-11-20

P.4 Conventions & Terminology

The following conventions are used throughout this manual:

- Numbered lists provide sequential steps.
- Bulleted lists provide information, not procedural steps.
- The term ‘module’ refers to the option board.
- Hexadecimal values are written in the format 0xNNNN, where NNNN is the hexadecimal value.

P.5 Support

MITSUBISHI ELECTRIC EUROPE EUROPE B.V.
German Branch
Mitsubishi-Electric-Platz 1
D-40882 Ratingen
Phone: +49 (0) 21 02 / 486-0
Hotline: +49 2102 1805 000-765 /-766
Fax: +49 (0) 21 02 / 4 86-1 12 0
e-mail: megfa-mail@meg.mee.com
www.mitsubishi-automation.com

MITSUBISHI ELECTRIC USA AUTOMATION
500 Corporate Woods Parkway
Vernon Hills, Illinois 60061
Phone: +1 (0) 847 / 478 21 00
Fax: +1 (0) 847 / 478 03 27

MITSUBISHI ELECTRIC JAPAN CORPORATION
Tokyo Bldg.
2-7-3 Marunouchi Chiyoda-Ku
Tokyo 100-8310
Phone: +81 (0) 3 / 32 18 31 76
Fax: +81 (0) 3 / 32 18 24 22

Please refer to (Mitsubishi Electric) for other region contact addresses.

1 Preoperation instructions

1.1 History

The FR-A800/FR-F800 series from Mitsubishi Electric (ME), is two of frequency inverter families. The communication modules, option boards, enabling communication on different industrial networks, are developed and produced by HMS Industrial Networks.

Examples of applications for the frequency inverters are:

- Lifting equipment
- Warehouse systems
- Extruders
- Centrifuges

1.2 Product overview

The A8NECT_2P EtherCAT Option Board for EtherCAT allows information to be transferred seamlessly between an FR-A800/FR-F800 inverter and an EtherCAT network with minimal configuration requirements. The option board installs directly onto the inverter's control board, and presents two 100BASE-TX Ethernet ports for connection to the EtherCAT network.

The option board is connected directly to the control board of the inverter and communicates to the inverter via a built-in communication port. Note that when the inverter's network communication port is used by the A8NECT_2P EtherCAT Option Board, it is unavailable for use by any other network.

Before using the interface, please familiarize yourself with the product and be sure to thoroughly read the instructions and precautions contained in this manual. In addition, please make sure that this instruction manual is delivered to the end user of the product, and keep this instruction manual in a safe place for future reference or unit inspection.

1.3 Features

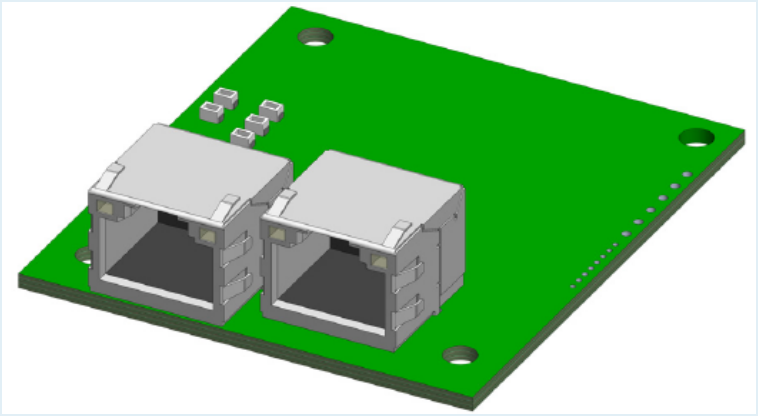
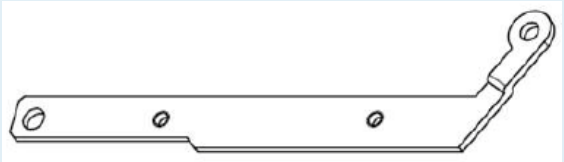
- CANopen over EtherCAT application layer
- Drive parameters accessible as Service Data Objects (SDOs)
- Monitor parameters accessible as Process Data Objects (PDOs)
- Drive operation according to CANopen DS402 drive profile
    - Supports velocity mode and torque mode
- Process data watchdog functionality
- Firmware upgrade using File access over EtherCAT supported

1.4 Unpacking and product confirmation

1.4.1 Shipment confirmation

Check the enclosed items. Confirm that the correct quantity of each item was received, and that no damage occurred during shipment.

1.4.2 Component overview

Item	No. of pcs	
PCB board	1	
M3 x 6 mm screw	3	
Board spacer	2	
LED cover	1	
PE plate	1	

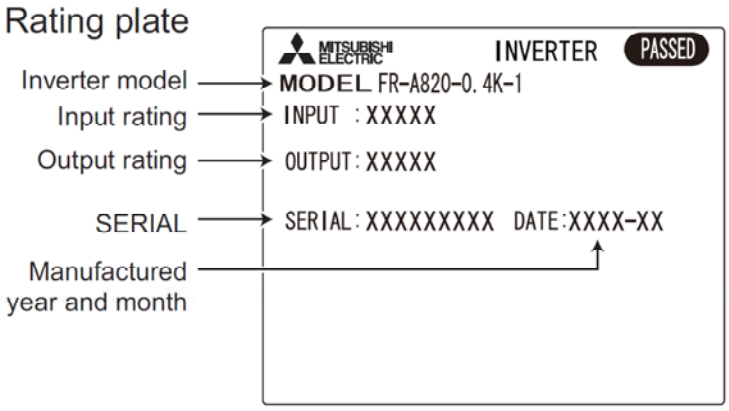
1.5 Environmental specifications

Item	Specification
Operating temperature	-10° to +50° Celsius (ambient of the drive, non-freezing)
Storage temperature	-40° to +65° Celsius
Relative humidity	93% non-condensing
Vibration	Max acceleration amplitude: 10 m/s^2 at 9 - 200 Hz Max displacement amplitude: 3 mm at 2 - 9 Hz
Grounding	Connected to inverter frame ground through the PE plate / isolated from inverter control power common
Power supply	Supplied from inverter
Cooling method	Self-cooled
Communication speed	100 Mbit

The A8NECT_2P interface is lead-free / RoHS-compliant.

1.6 Product Identification

The A8NECT_2P can be used with the inverter models listed below which have the following SERIAL number or later. Check the SERIAL number indicated on the inverter rating plate or package.



x x x xxxxxx
Symbol Year Month Control number
SERIAL

The SERIAL consists of one symbol, two characters indicating the production year and month, and six characters indicating the control number.
The last digit of the production year is indicated as the Year, and the Month is indicated by 1 to 9, X (October), Y (November), or Z (December).

FR-A800 series

Model	Country of origin indication	SERIAL number
FR-A820-00046(0.4K) to 04750(90K) FR-A840-00023(0.4K) to 06830(280K) FR-A842-07700(315K) to 12120(500K) FR-A846-00023(0.4K) to 03610(132K)	MADE in Japan	x5Yxxxxxx or later
	MADE in China	x5Zxxxxxx or later

FR-F800 series

Model	Country of origin indication	SERIAL number
FR-F820-00046(0.75K) to 04750(110K) FR-F840-00023(0.75K) to 06830(315K) FR-F842-07700(355K) to 12120(560K)	MADE in Japan	x5Yxxxxxx or later
	MADE in China	x5Zxxxxxx or later

2 Installation

2.1 Pre-installation instructions

Important! Make sure that the inverter's power is off.

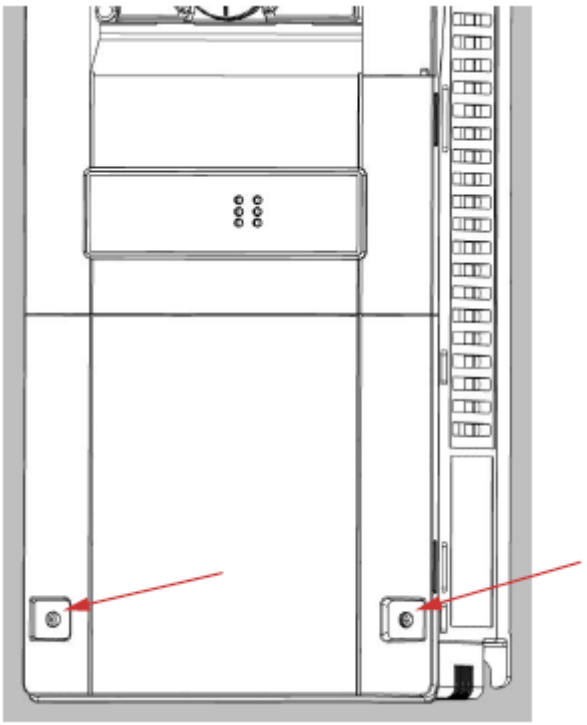
	⚠ WARNING
⚠	To avoid damage to the inverter or plug-in option board, never install or remove a plug-in option board while the inverter's input power is on.

Physical installation of the option board is a two-step process. First, the board will be mounted onto an available option connector on the inverter's control board. Second, the board will be connected to the EtherCAT network using a network cable.

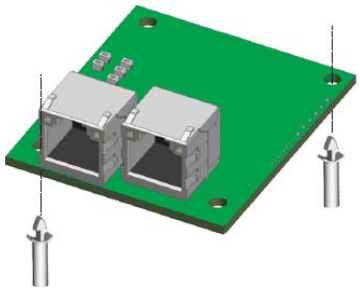
2.2 Installation procedure

1. Make sure that power is off. After switching off the power, wait for at least 10 minutes prior to proceeding with the installation.

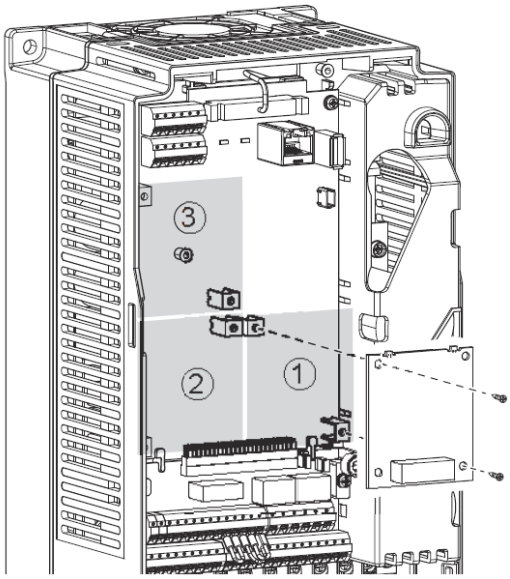
2. Remove both lids of the FR-A800/FR-F800
   - Unscrew the two screws in the bottom corners of the inverter.
   - Remove the lid covering the front of the inverter.
   - Unscrew the screw in the bottom right corner of the lid covering the upper from of the inverter.
   - Remove the lid.



3. Put the included board spacers in the holes at the right top and left bottom corners of the PCB.



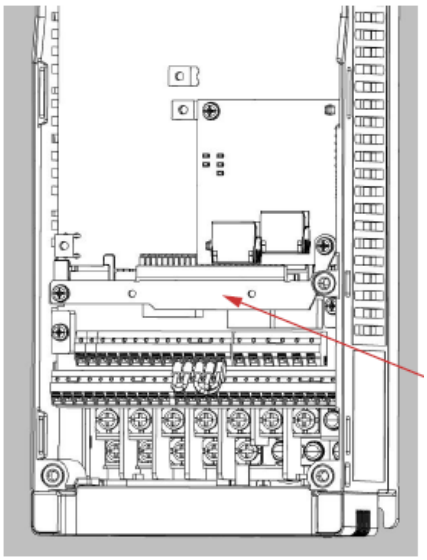
4. Position the A8NECT_2P EtherCAT option board at the option slot 1 as shown in the image. This is the only position that will allow network connectivity.



5. Fasten the option board by tightening the included screws at the top left and bottom right corners. The PE plate is attached along with the screw in the bottom right corner.

   Note: Over-tightening the screws will damage the board.

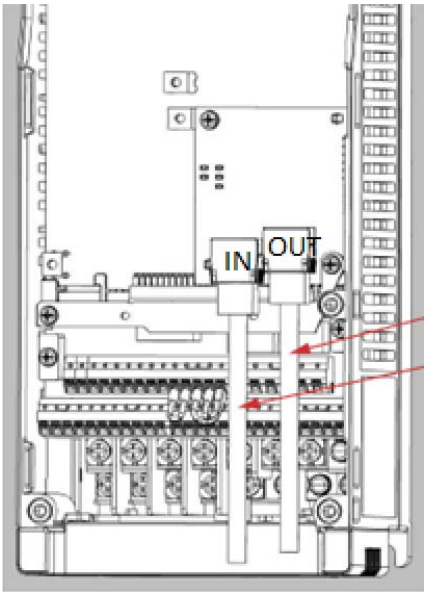
6. Fasten the other end of the PE plate with another screw as shown in the picture.



7. Attach the network cable that is coming upstream from the EtherCAT master or a slave to the EtherCAT IN port. If more slaves shall be present on the network connect the cable from the EtherCAT OUT port to the next slave.

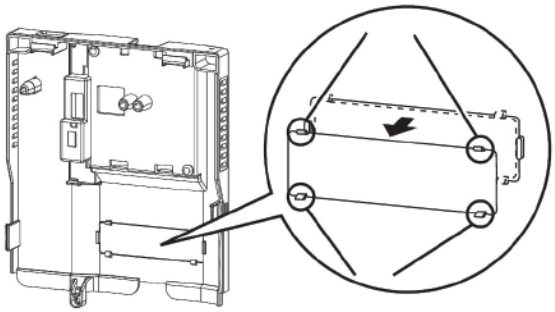
   See section 2.3.1 for more information about EtherCAT port assignment.

   Note: The shield of the RJ45 connector is not connected directly to PE.

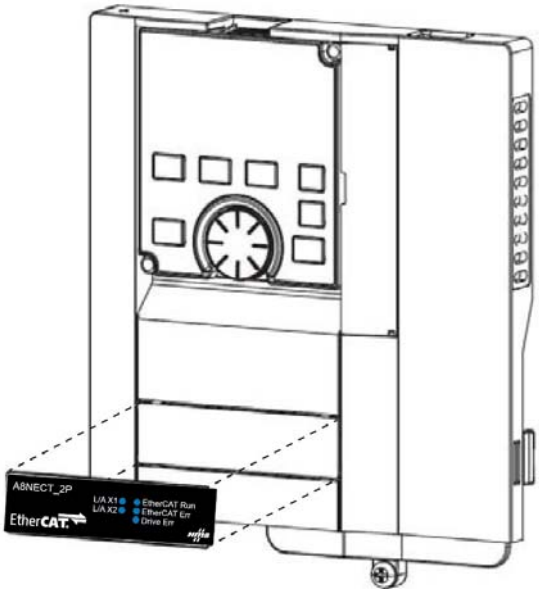


8. To fit the LED cover on the front cover of the drive, do as follows:

   - Cut the bridges, using nippers, on the upper front cover.



   - Snap the LED cover into the front cover of the drive.



9. Fasten both front covers, top front cover first.

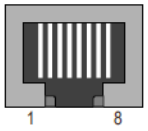
The option board is now mounted and power can be applied.

Removal

1. Remove power from the inverter.
2. Remove both lids of the FR-A800/FR-F800.
3. Remove the network cable(s).
4. Remove the screws.
5. Carefully remove the option board by lifting it straight up.
6. Remove the board spacers.
7. Replace the lids.

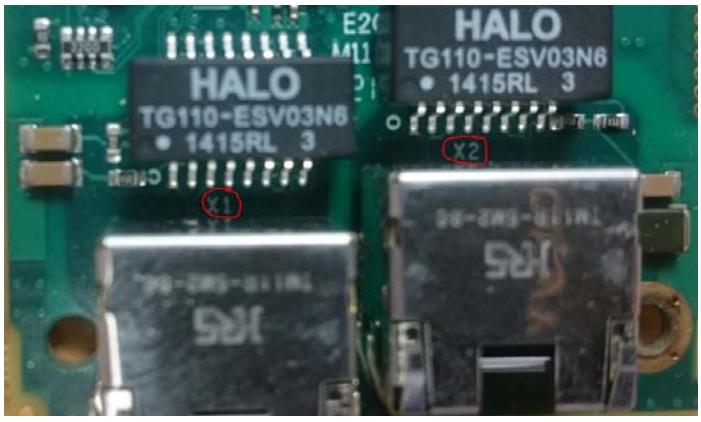
2.3 Network connectors (RJ45)

The option board provides connection to EtherCAT through two RJ45 connectors.

Pin	Name	Description	
1	TX+	Transmit positive	
2	TX-	Transmit negative	
3	RX+	Receive positive	
4	NC	Terminated with a 50-75Ω resistor	
5	NC	Terminated with a 50-75Ω resistor	
6	RX-	Receive negative	
7	NC	Terminated with a 50-75Ω resistor	
8	NC	Terminated with a 50-75Ω resistor	
Housing	Shield	Filter connection to PE	

2.3.1 EtherCAT port assignment

For the EtherCAT network to work properly it is important to connect the option board correctly. Slaves with two ports are connected in a chain starting with the master and ending with the last slave. Other topologies can be created using slaves with more than two ports.



The RJ45 ports are marked X1 and X2 on the PCB according to the picture.

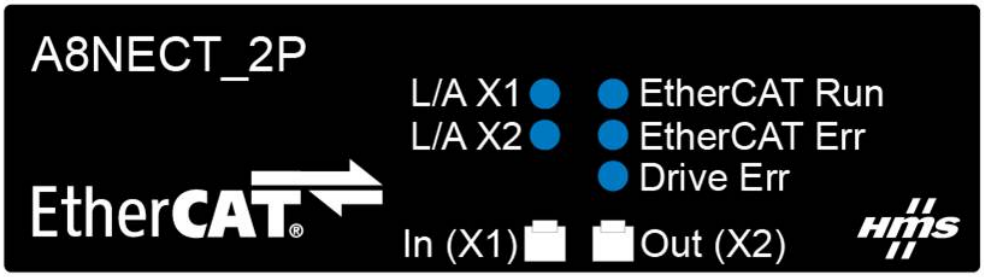
The ports are used on EtherCAT in the following way:

PCB port marking	EtherCAT port
X1	EtherCAT IN
X2	EtherCAT OUT

The EtherCAT IN port shall be connected in the upstream direction towards the master.

The EtherCAT OUT port shall be connected in the downstream direction away from the master unless the option board is the last slave on the EtherCAT network.

2.4 LED indicators



LED	State	Status
EtherCAT Run	Off	The option board is in EtherCAT state INIT
	Blinking green	The option board is in EtherCAT state PRE-OPERATIONAL
	Flashing green, 1 flash	The option board is in EtherCAT state SAFE-OPERATIONAL
	Solid green	The option board is in EtherCAT state OPERATIONAL
	Flickering green	The option board is in EtherCAT state BOOTSTRAP
	Solid red	Major unrecoverable error detected. Option board needs to be power cycled.
EtherCAT Err	Off	No EtherCAT related error
	Blinking red	Invalid EtherCAT configuration. State change commanded by master is impossible due to register or object settings.
	Flashing red, 1 flash	Unsolicited state change. The option board has changed the EtherCAT state autonomously.
	Flashing red, 2 flashes	Process data watchdog. Process data has not been updated within the configured timeout or updated before transition to state OPERATIONAL.
	Solid red	Major unrecoverable error detected. Option board needs to be power cycled.
Drive Err	Off	Communication with inverter is working without problems
	Flashing red, 2 flashes	Invalid process data parameter mapped
	Flashing red, 3 flashes	Too many process data parameters mapped
	Flashing red, 4 flashes	The option board has not been able to initialize the main unit correctly and generated a major unrecoverable event.
	Solid red	Error in communication with inverter
L/A X1	Off	No link on port X1 (EtherCAT IN)
	Solid green	Link, no activity on port X1 (EtherCAT IN)
	Flickering green	Link and activity on port X1 (EtherCAT IN)
L/A X2	Off	No link on port X2 (EtherCAT OUT)
	Solid green	Link, no activity on port X2 (EtherCAT OUT)
	Flickering green	Link and activity on port X2 (EtherCAT OUT)

3 Getting started

This chapter describes how to install the option board and how to implement a small example configuration. Most of the configuration of parameters and data exchange is done in the master tool, but it is also possible to change parameters via the handheld parameter unit.

3.1 Physical installation

The physical installation is described in section 2.2 Installation procedure.

3.2 Inverter firmware version

For the A8NECT_2P module to work properly the firmware version installed on the inverter must be 8485/5 or higher for an FR A800 series inverter and 8463/5 or higher for an FR F800 inverter.

Firmware version can be checked by reading inverter parameters 168 and 169.

3.3 Download ESI file

Each device on EtherCAT is associated with an EtherCAT Slave Interface (ESI) file in XML format, which holds a description of the device and its functions.

Download the appropriate ESI file from the Mitsubishi Electric website, see section P.2 Download

The following information from the ESI file is used to match it against a product on the EtherCAT network by the configuration tool:

- Vendor ID
- Product code
- Revision

Therefore it is important to make sure the downloaded ESI file matches the information provided by the option board.

3.4 Inverter setup

To use EtherCaT communication inverter operation mode has to be changed to network. One way to do this is by leaving the default setting of parameter 79 (0), and setting parameter 340 to 1. After restarting the inverter it should operate in network mode. For detailed settings, please refer to inverter instruction manual.

3.5 TwinCAT 3.1 Configuration example

This chapter shows how to set up a simple configuration containing the A8NECT_2P module in TwinCAT 3.1 from Beckhoff.

3.5.1 Installing the ESI file

1. Download the ESI file according to section 3.3.
2. Copy the ESI file to the following location:
   %TwinCAT install directory%\3.1\Config\Io\EtherCAT

3.5.2 Installing TwinCAT driver on a network adapter

TwinCAT requires a special driver to be installed on the network adapter that is going to be used for EtherCAT communication. This section describes how to install this driver.

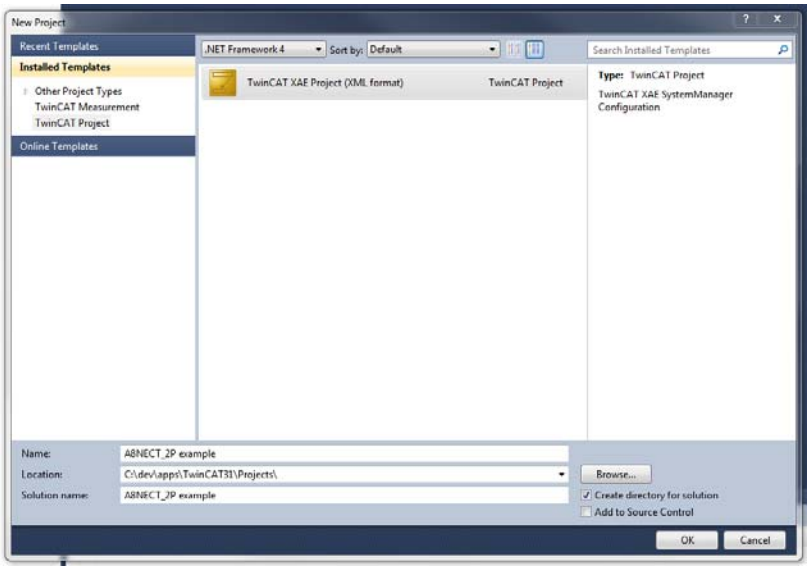
This step only needs to be performed once, directly after TwinCAT has been installed.

1. Run %TwinCAT install directory%\3.1\System\TcRteInstall.exe.
   If there already is an adapter under “Installed and ready to use devices” the remaining steps does not need to be performed.
2. Select an adapter under “Compatible devices” and press “Install”. If there isn’t an adapter under “Compatible devices” a new network adapter needs to be installed on the computer. Beckhoff provides a list with compatible network adapters here:
   http://infosys.beckhoff.com/english.php?content=content/1033/tcsystemmanager/reference/ethercat/html/ethercat_supnetworkcontroller.htm

Driver can be installed on incompatible devices as well but those devices can be used for demo mode only, where real time communication is not guaranteed.

3.5.3 Creating the TwinCAT project

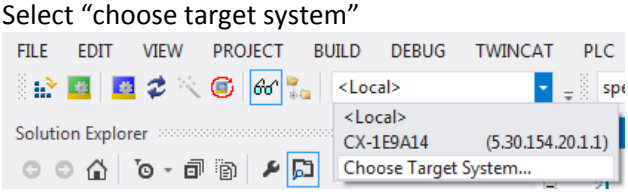
1. Start TwinCAT XAE
2. Create a new TwinCAT XAE project and give it an appropriate name.



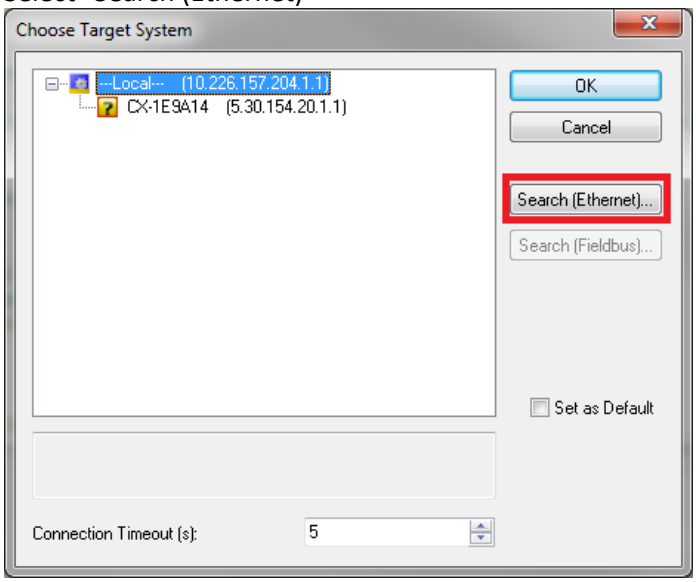
3.5.4 Selecting target system

This guides focuses mainly on configuration variant where the PC is EtherCaT master and is directly connected to the inverter. In that case default settings are enough and this step can be skipped. If however system configuration with Beckhoff PLC as EtherCaT master is used target system needs to be changed.

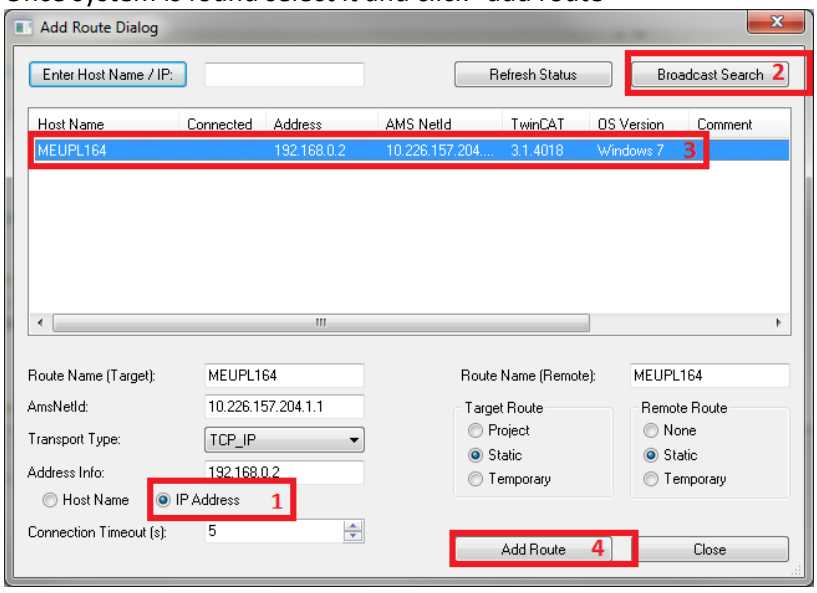
1. Select “choose target system”



2. Select “Search (Ethernet)”

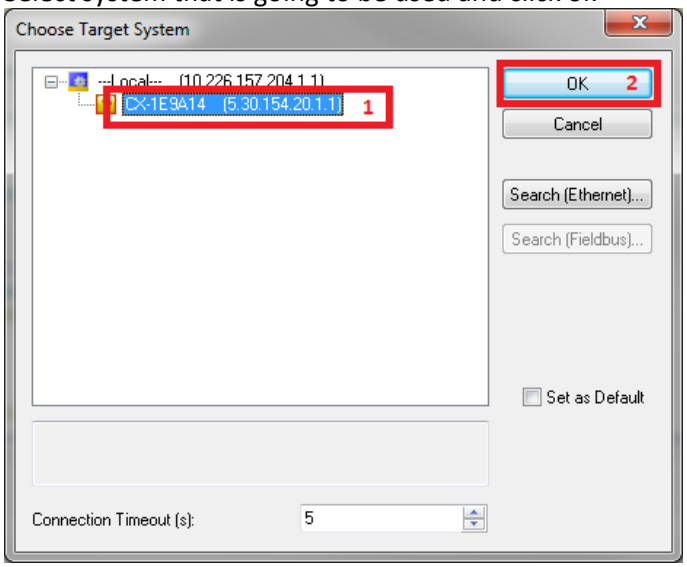


3. Select IP address under address info, then click “broadcast search”
4. Once system is found select it and click “add route”



5. When asked input username and password for target system

6. Select system that is going to be used and click ok

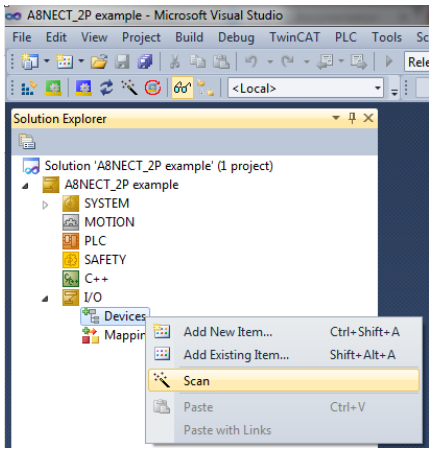


3.5.5 Online TwinCAT configuration

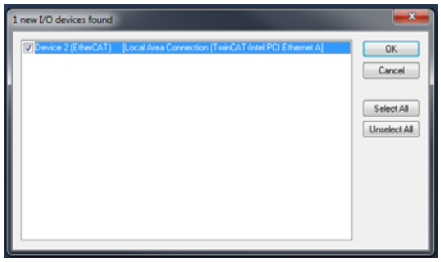
If the EtherCAT network is attached to the computer where the TwinCAT runtime is running it is possible to do an online configuration. During an online configuration TwinCAT will scan the EtherCAT network and automatically add all the present devices to the configuration.

If the EtherCAT network is not connected, an offline configuration needs to be done. See section 3.5.6 for information on how to do this.

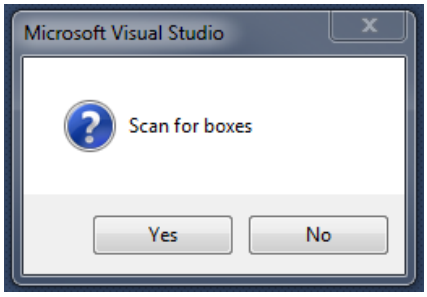
1. Make sure the EtherCAT network is connected.
2. Right click I/O->Devices and press “Scan”.



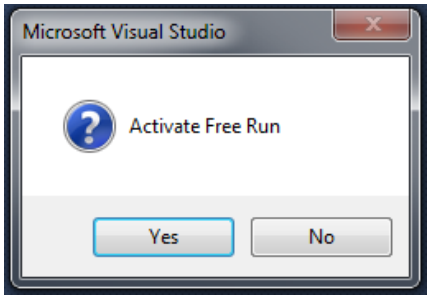
3. Select the network adapter to use.



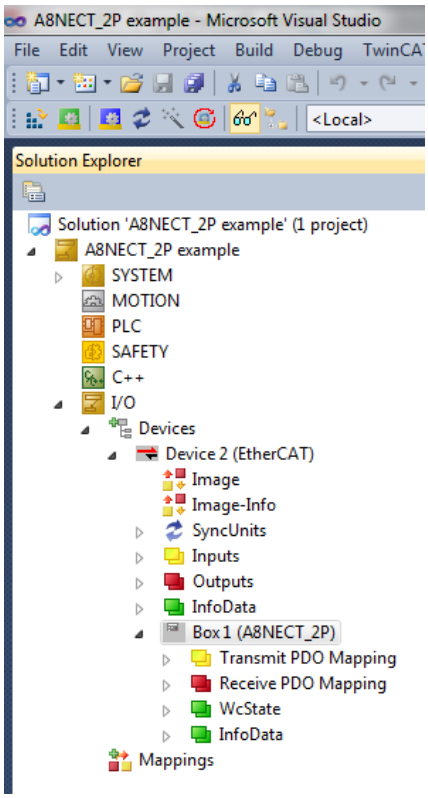
4. Answer “Yes” to the question “Scan for boxes”.



5. Answer “No” to the question “Active Free Run”



6. The A8NECT_2P is now detected by TwinCAT and can be used.

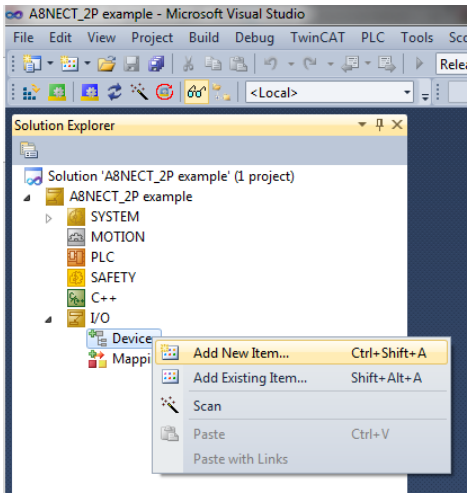


3.5.6 Offline TwinCAT configuration

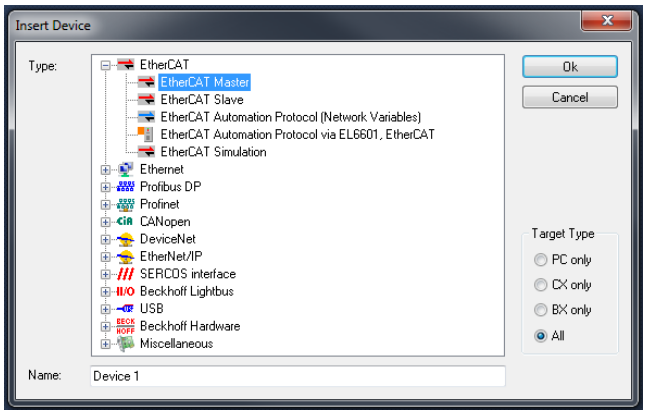
An offline configuration can be created even if the EtherCAT network is not connected.

If the EtherCAT network is connected it is recommended to do an online configuration instead, see section 3.5.5

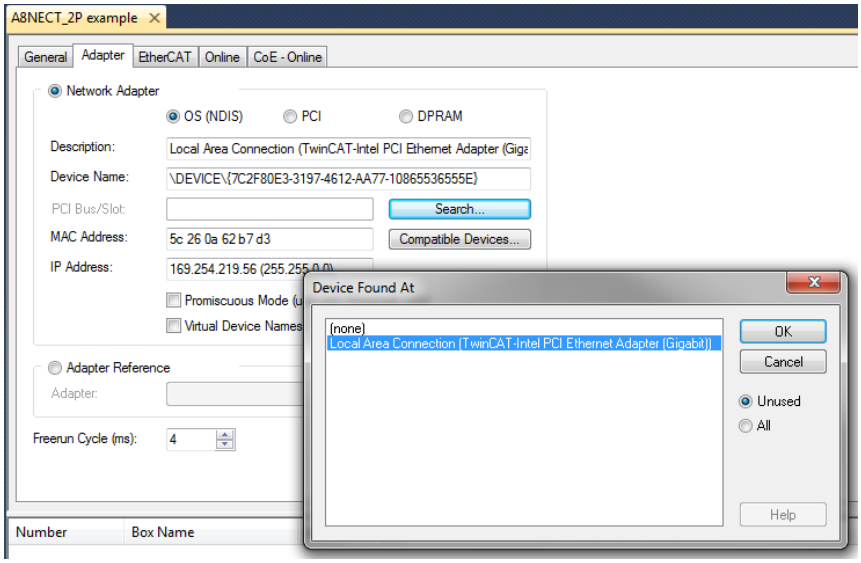
1. Right click I/O->Devices and select “Add new item”.



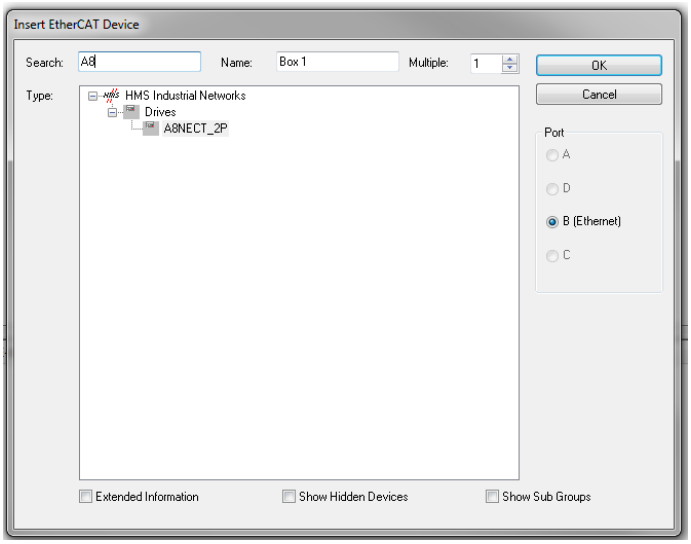
2. Add an “EtherCAT Master”.



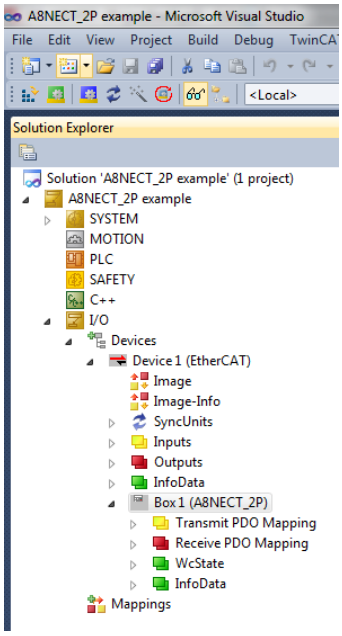
3. Configure the network adapter to use by double-clicking on the added EtherCAT master (Device 1 in this example) and go to the “Adapter” tab. Press the “Search” button and select the desired network adapter.



4. Right click the EtherCAT master and select “Add New Item…”, find the A8NECT_2P device and add it.



5. The A8NECT_2P is now added to the configuration and can be used.

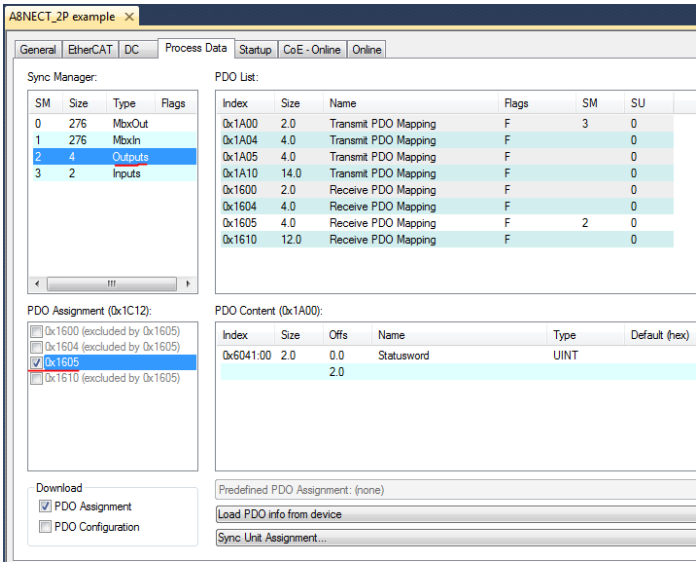


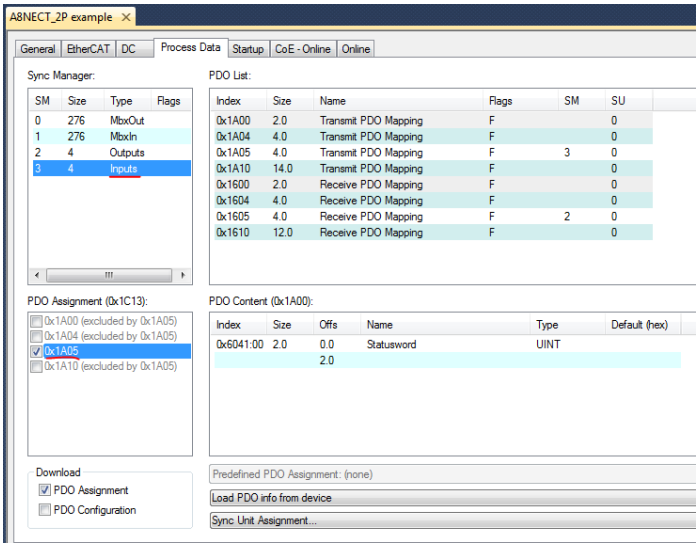
3.5.7 Configuring the A8NECT_2P

By double-clicking on the A8NECT_2P device it can be configured in a number of different ways.

3.5.7.1 Process data configuration

On the “Process Data” tab the process data configuration of the device is done. In this example the PDO configuration 0x1605 for output data and PDO configuration 0x1A05 for input data is selected. These are the PDOs used for velocity control according to the DS402 drive profile. The output PDO configuration contains the control word and target velocity while the input PDO configuration contains the status word and actual velocity.

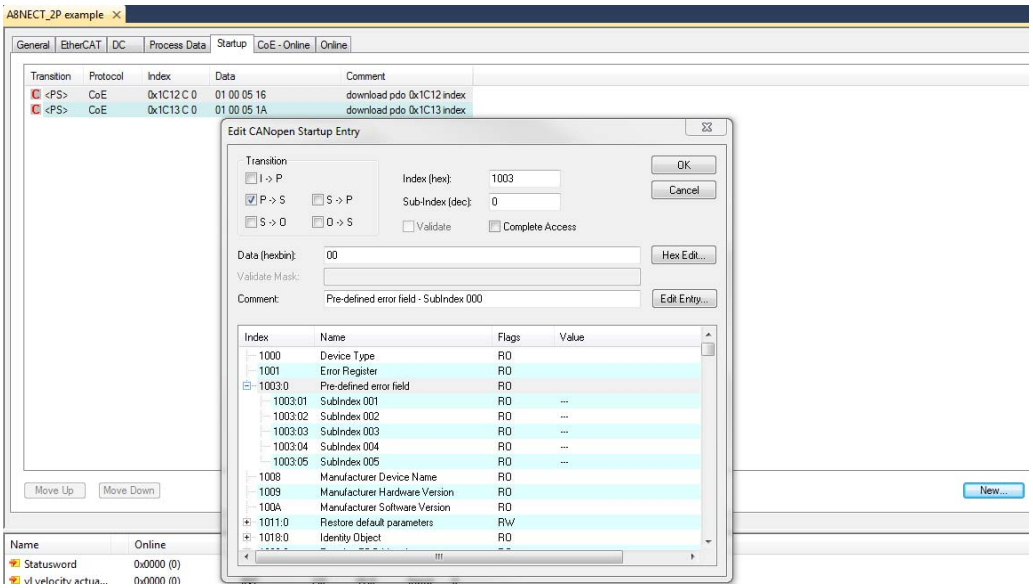




3.5.7.2 Startup commands

On the “Startup” tab it is possible to create device specific startup commands. This can be useful e.g. if a specific parameter in the drive shall be set to a certain value during startup.

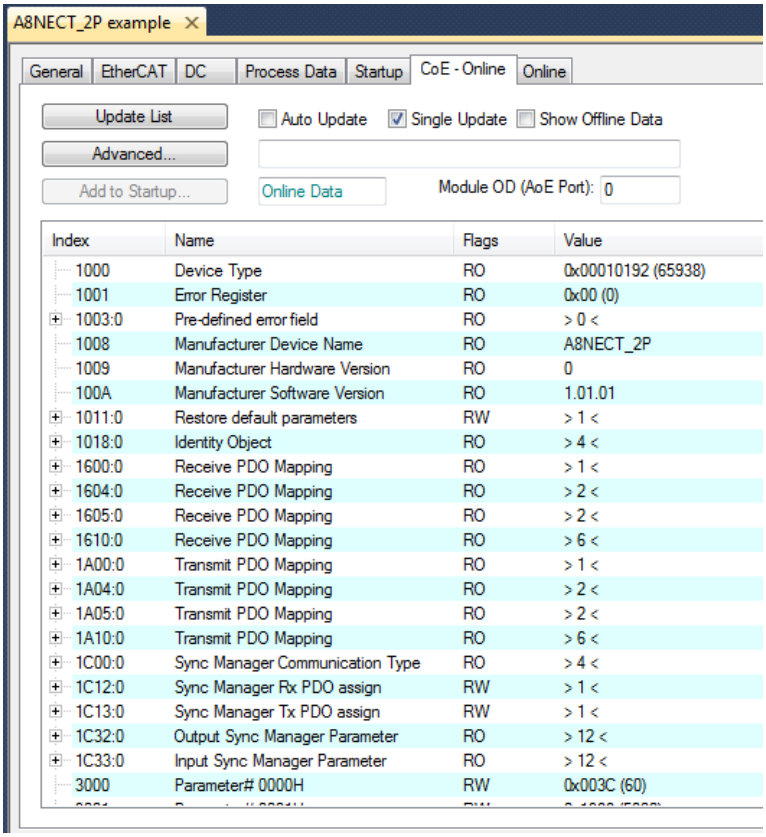
In this example a startup command to clear the error history in object 0x1003 is added on the transition from EtherCAT state PRE-OPERATIONAL to SAFE-OPERATIONAL.



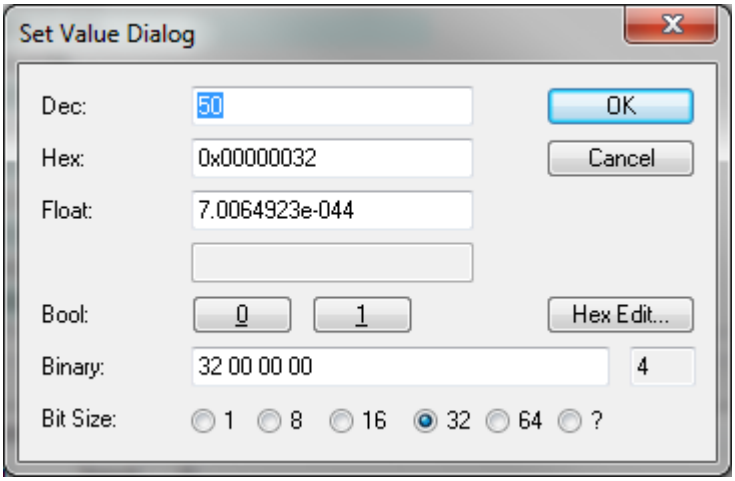
3.5.7.3 Object dictionary

The CANopen over EtherCAT object dictionary can be viewed on the “CoE – Online” tab.

If the EtherCAT network is connected the values of the objects can be shown. It is also possible to write objects here.



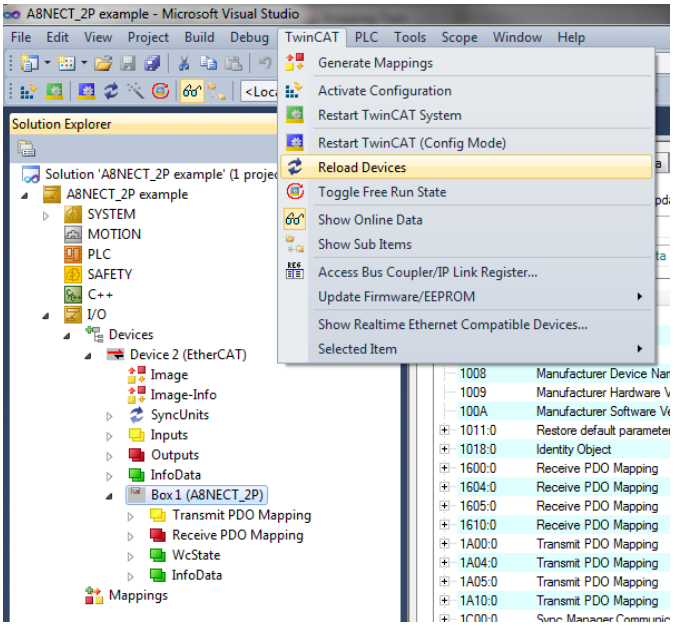
The CoE – online tab allows user to read and write inverter parameters as well. Object index is hexadecimal number. For inverter parameters index = parameter number + 0x3000. (for example pr. 80 Motor capacity has index 3050). To modify the value double click on the object and edit the value in dialog window.



3.5.7.4 Using the new configuration

Since the process data mapping was changed in section 3.5.7.1 and a startup command was added in 3.5.7.2 it is necessary to reload the devices to make TwinCAT use the new configuration.

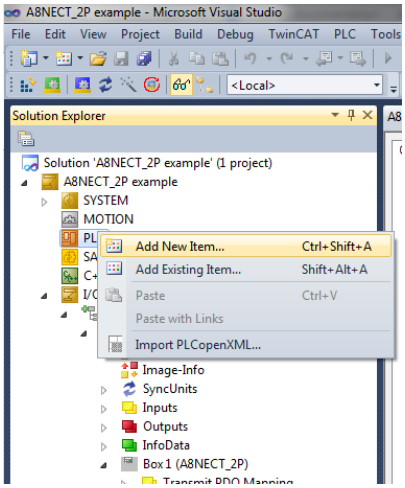
This is done by selecting “TwinCAT” in the menu and clicking “Reload Devices”.



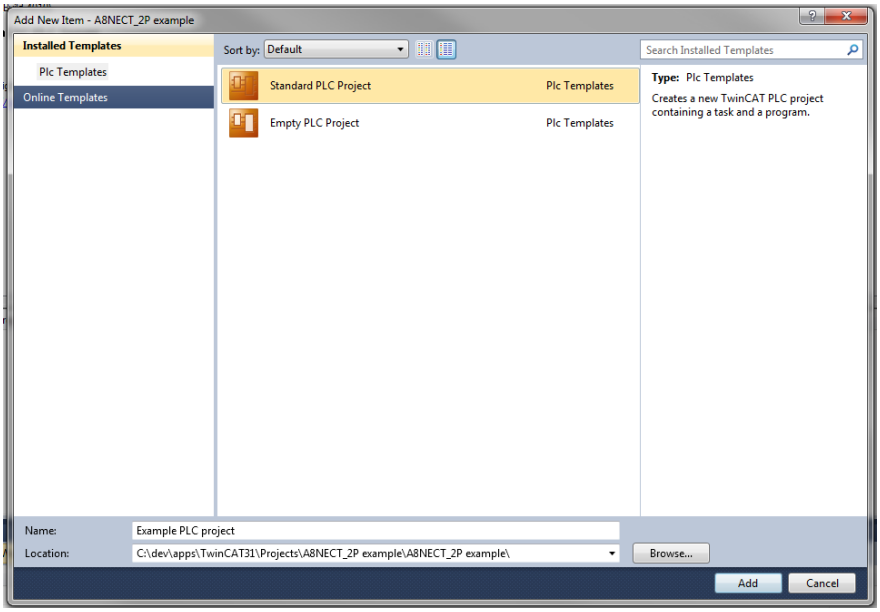
3.5.8 Creating a PLC project

This chapter describes how to create a PLC project that can use the process data of the A8NECT_2P.

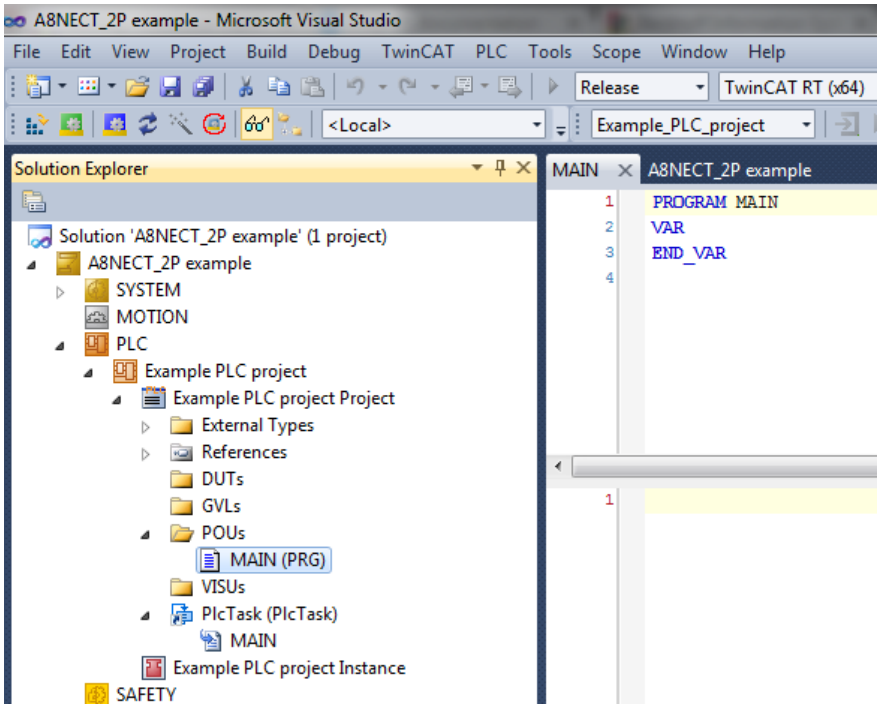
1. Right click “PLC” and select “Add New Item…”.



2. Select “Standard PLC Project” and give it a suitable name.



3. Expand PLC->”PLC project name”->”PLC project name” Project->POUs and double click on MAIN(PRG) to open the main PLC program.

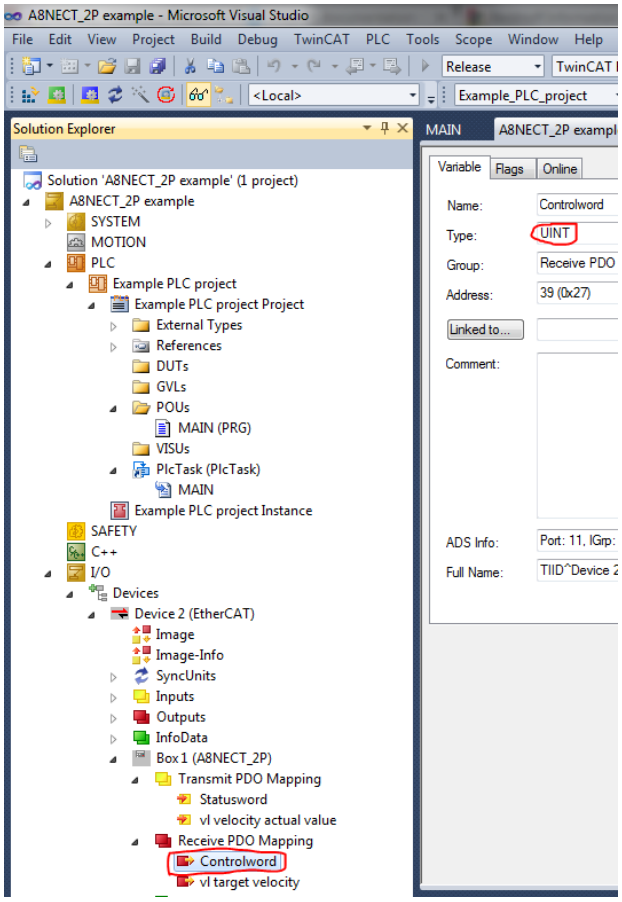


4. Define the variables that the PLC program will use. By using the keywords AT %Q* a variable that can be linked to output data is created, and by using the keywords AT %I* a variable that can be linked to input data is created.

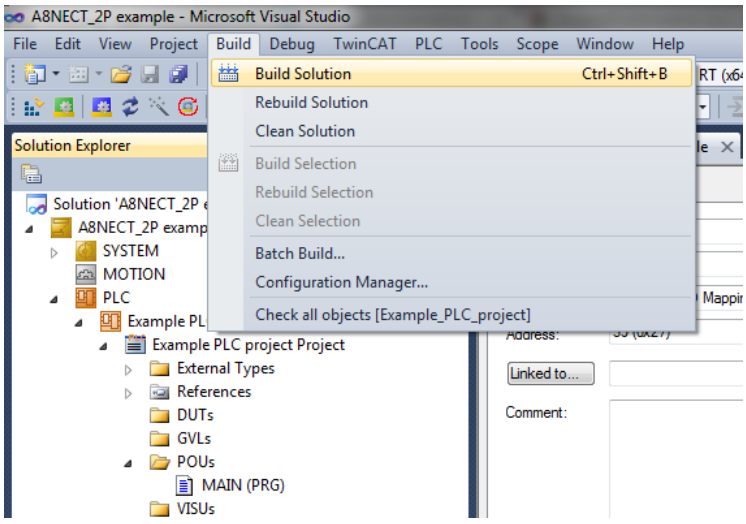
   In this example variables for controlword, target velocity, statusword and actual velocity has been created.

```
MAIN | A8NECT_2P example
1  PROGRAM MAIN
2  VAR
3      iControlWord    AT %Q* : UINT;
4      iTargetVelocity AT %Q* : INT := 200;
5
6      iStatusWord     AT %I* : UINT;
7      iActualVelocity AT %I* : INT;
8  END_VAR
9
```

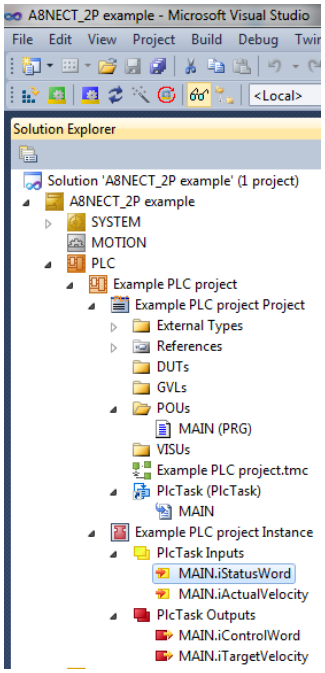
   To check what data type the process data mapped objects have, double-click on the object under the A8NECT_2P and check the “Type” field.



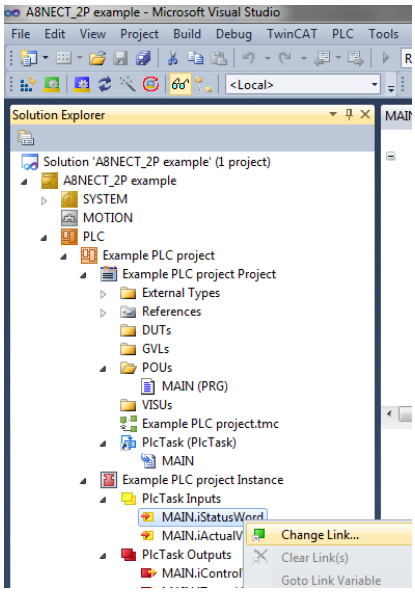
5. Build the PLC project.



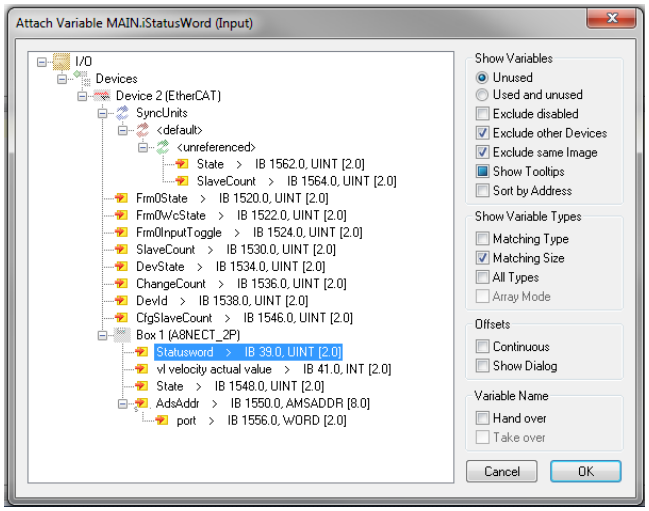
6. The new variables are created under “PlcTask Inputs” and “PlcTask Outputs”.



7. These variables needs to be linked to the process data mapped objects of the A8NECT_2P. Do this by right-clicking on the variable and select “Change Link”.



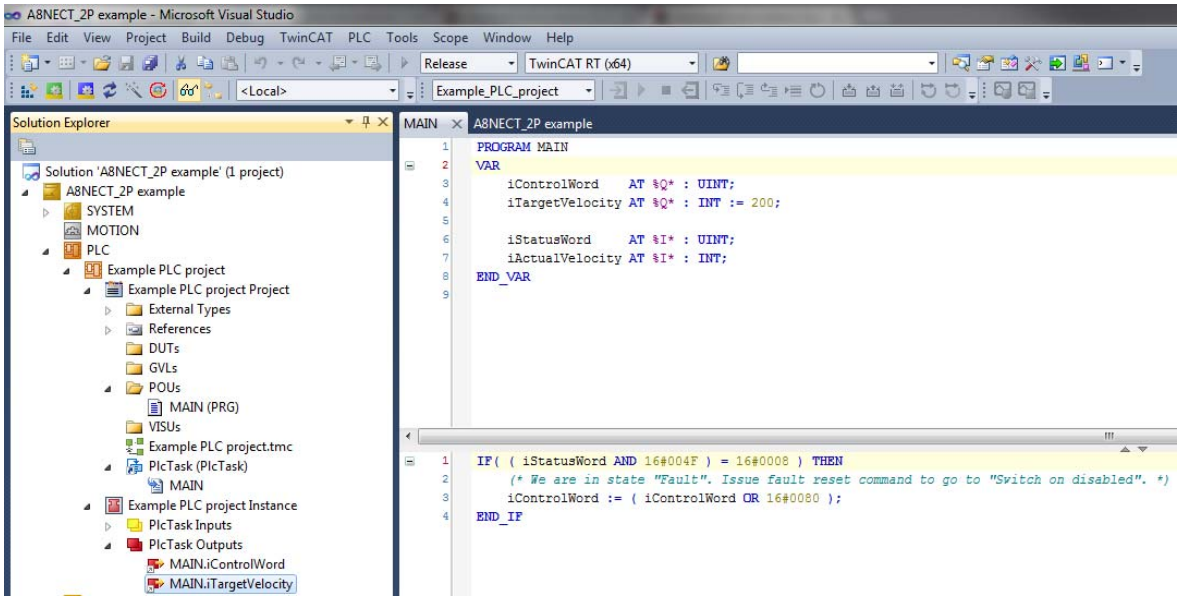
8. Find the process data mapped object corresponding to this variable.



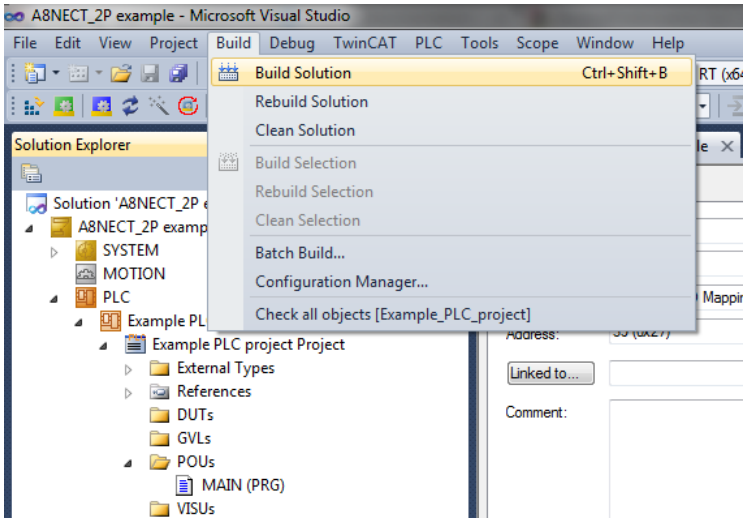
Repeat for all variables that needs to be linked to the process data of the A8NECT_2P.

9. Create code for the PLC project.

   In this example the fault bit is checked in the status word, and if it is set, the fault reset bit is set in the control word.



10. Build the PLC project.



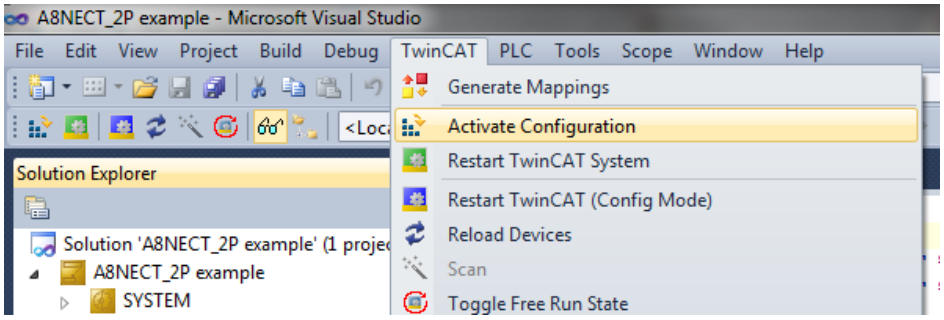
11. The PLC project is now created and can be used.

3.5.9 Starting communication

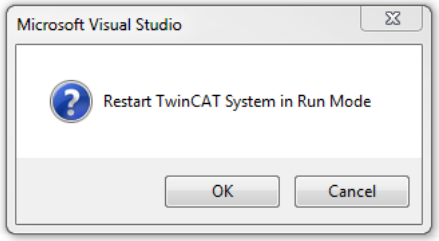
The A8NECT_2P device has now been configured and a PLC program using the process data of the A8NECT_2P has been created.

It is now time to start the EtherCAT network and start the PLC program. The EtherCAT network must now be connected if it has not been done before.

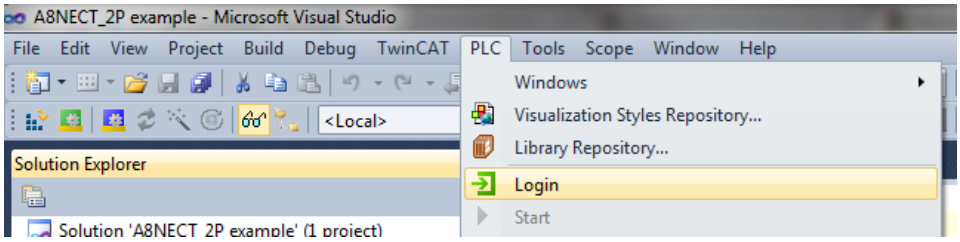
1. Activate the entire TwinCAT configuration.



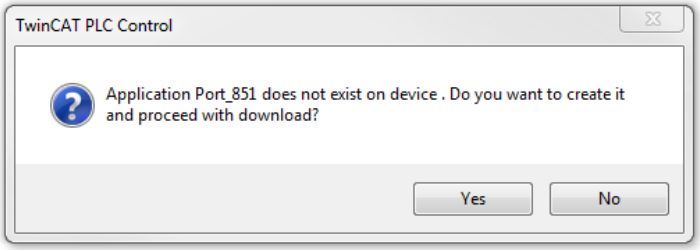
2. When asked if Run Mode should be started press “OK”.



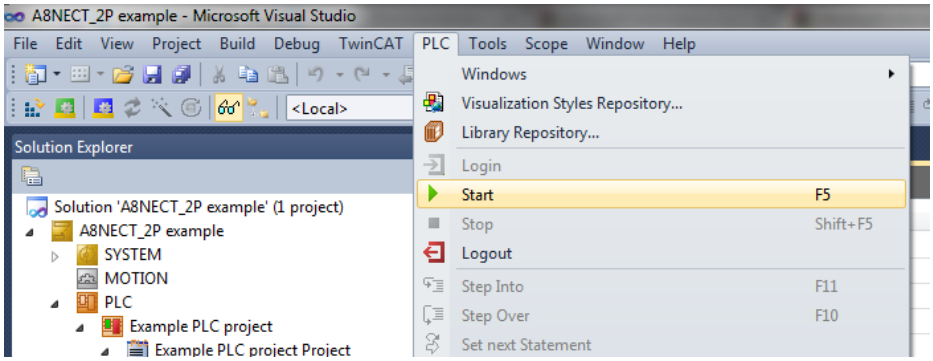
3. TwinCAT will now enter Run Mode. It is time to log in to the PLC and start the PLC program.



4. Press “Yes” when asked if application port should be created.



5. Start the PLC program.



6. The PLC program is now running and process data is exchanged with the A8NECT_2P device.

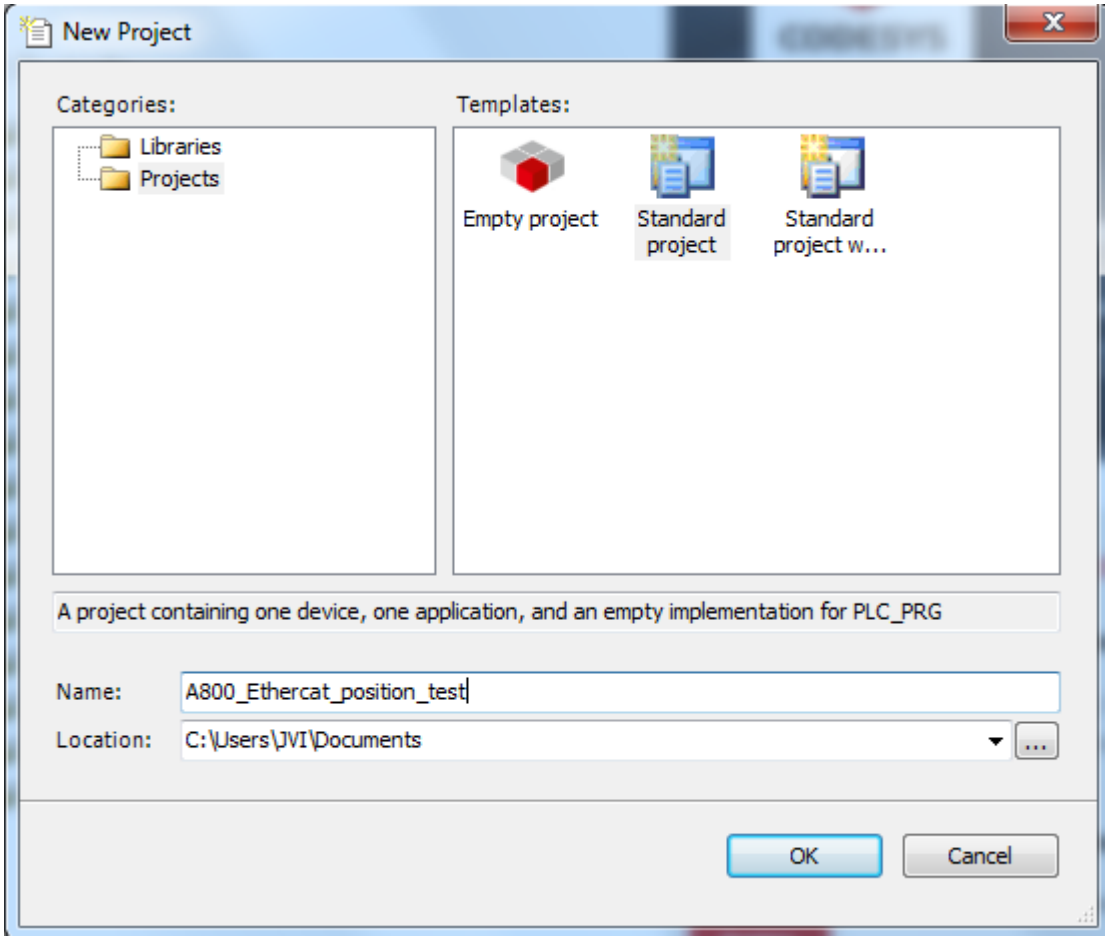
3.6 CoDeSys 3.5 Configuration example

This chapter shows how to setup a simple configuration containing the A8NECT_2P module in CoDeSys 3.5. Example hardware is iX TxB SoftMotion panel.

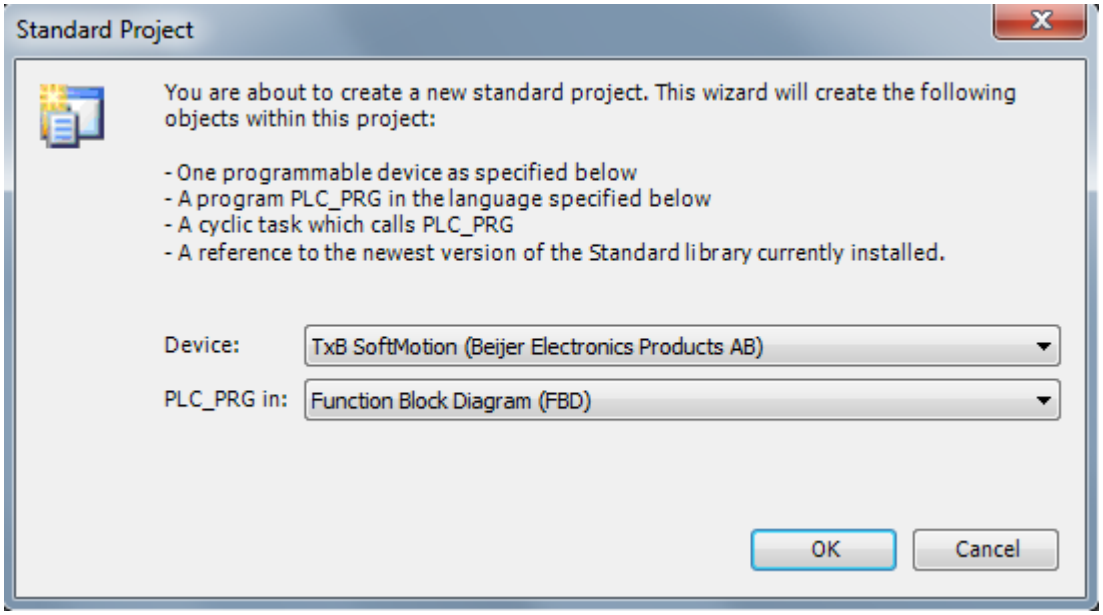
3.6.1 Creating the CoDeSys project

Approach with CoDeSys differs from TwinCAT. Version 3.5 SP7 patch 2 is used in this example but older version works as well.

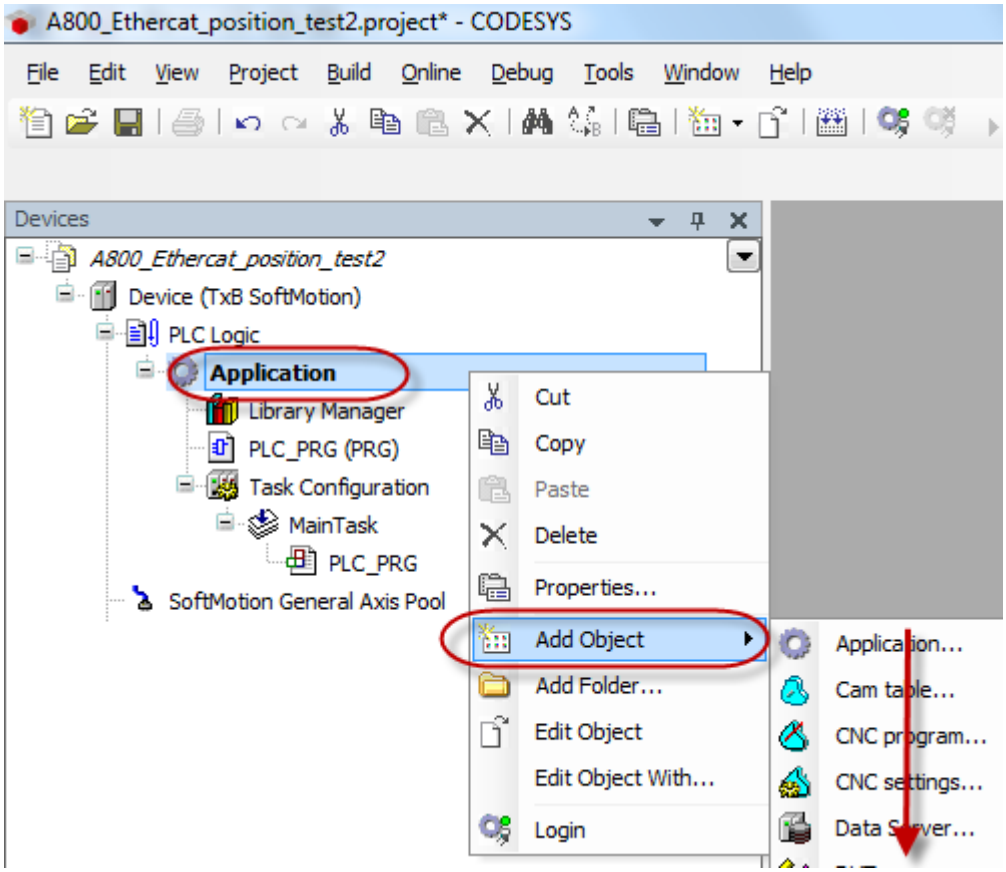
1. Start CoDeSys
2. Start new project. Name it and choose for example Standard project Template.



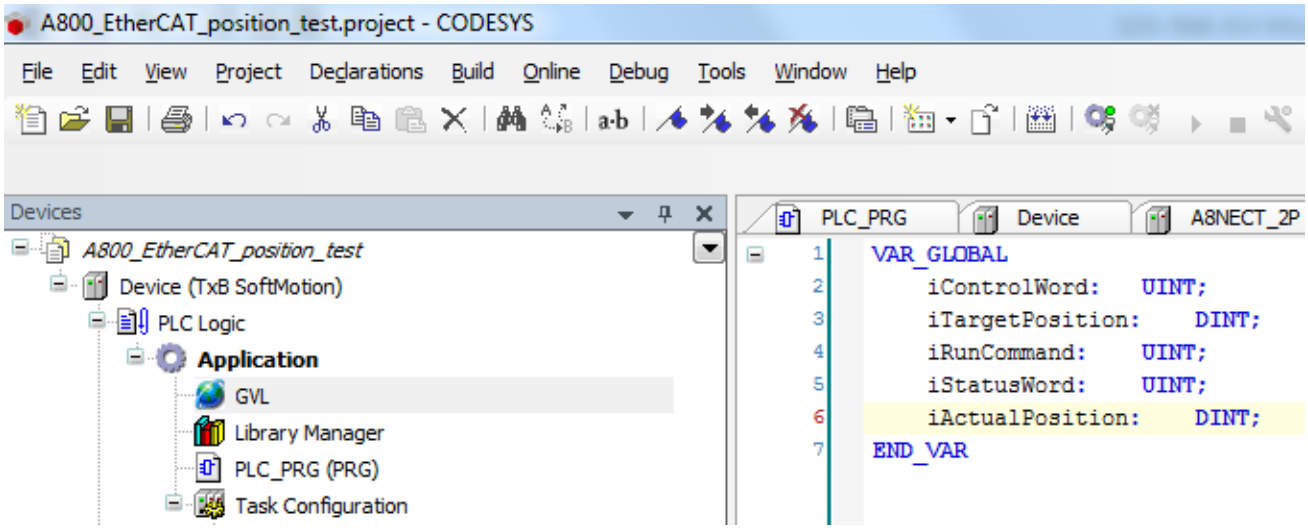
3. Select the Device and PLC programming editor type.



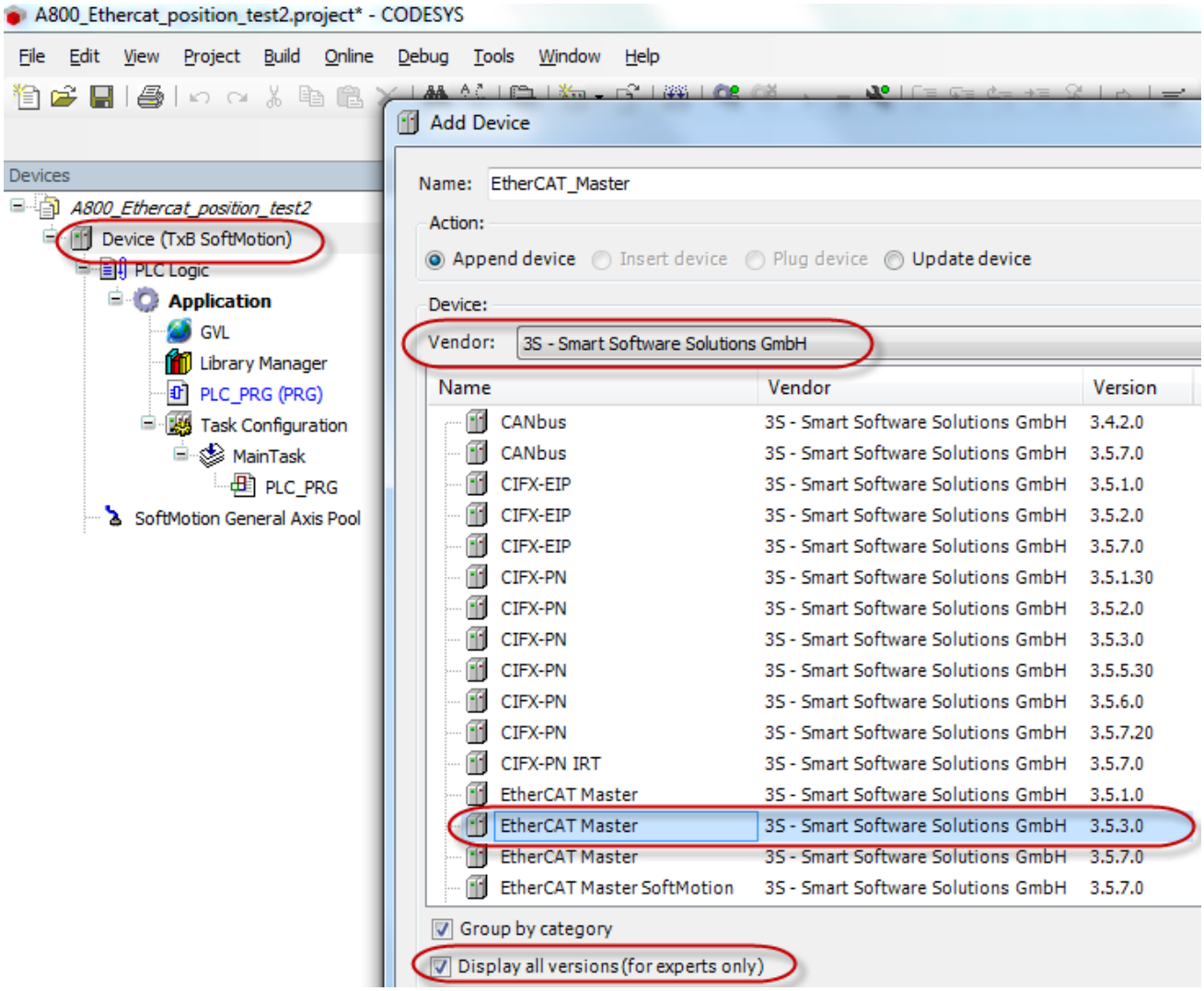
4. Click right mouse button on Application, select Add Object -> Global Variable List.



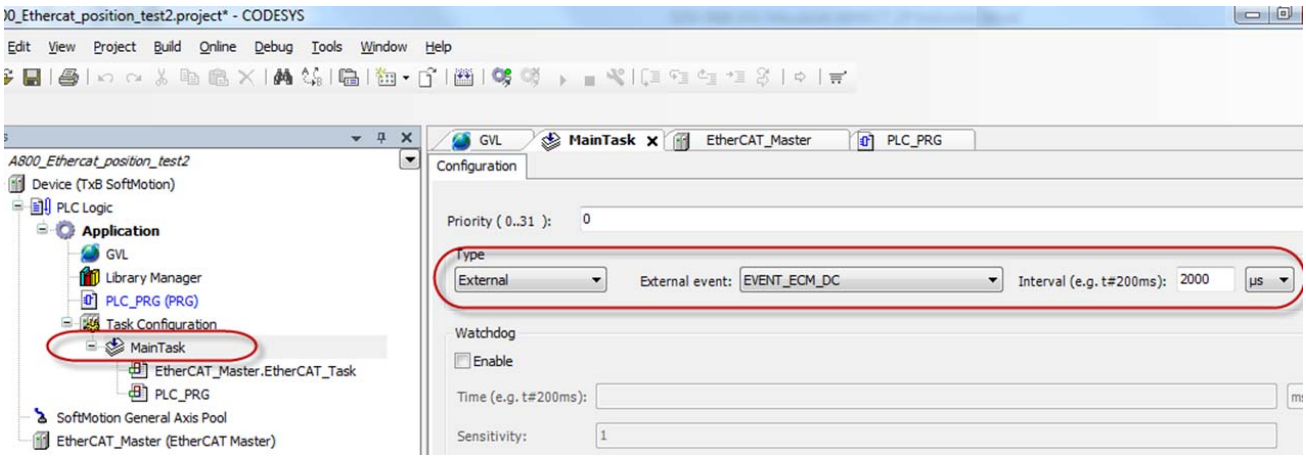
5. Open GVL and create following variables.



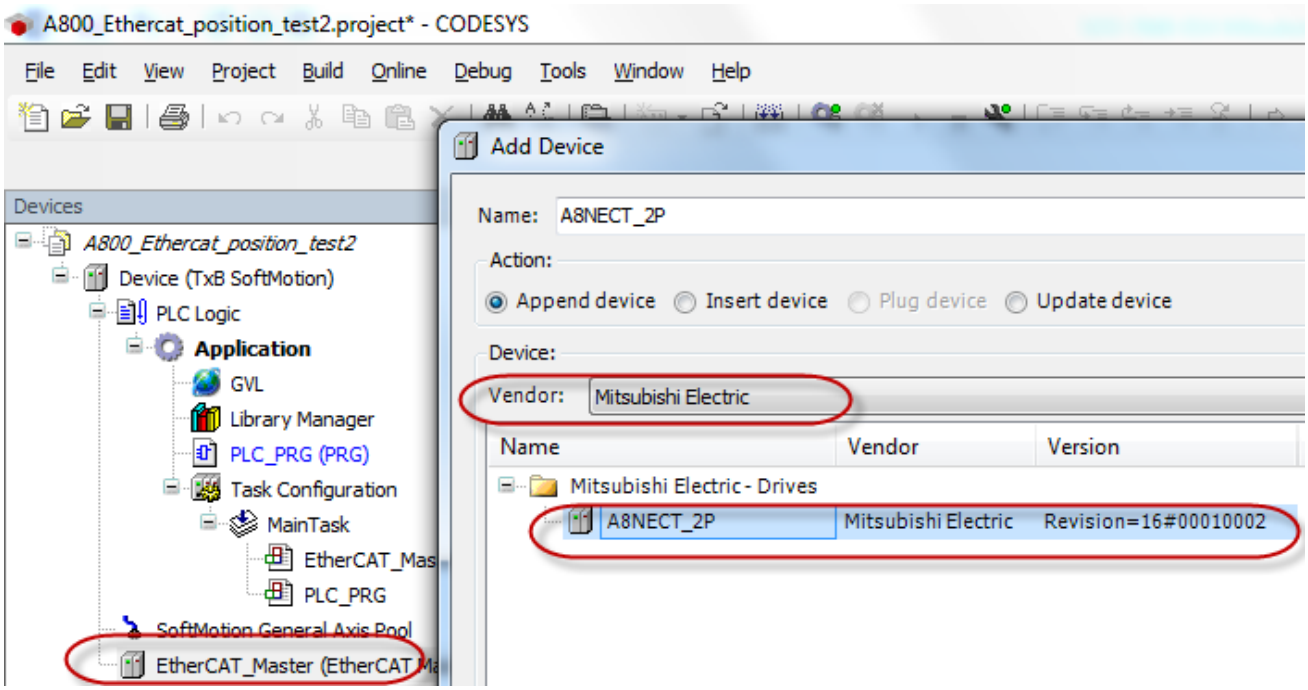
6. Click right mouse button on Device (TxB SoftMotion) and add EtherCAT Master device. Use vendor 3S, mark Display all versions to see all available versions if needed.



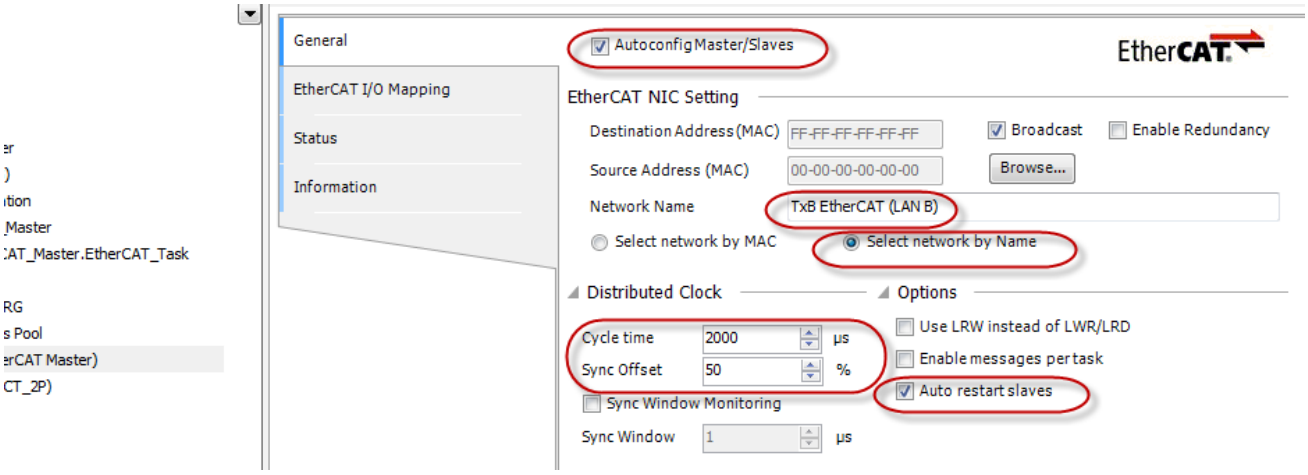
7. New Task comes automatically when EtherCAT master is selected. Change Task parameters according to following example.



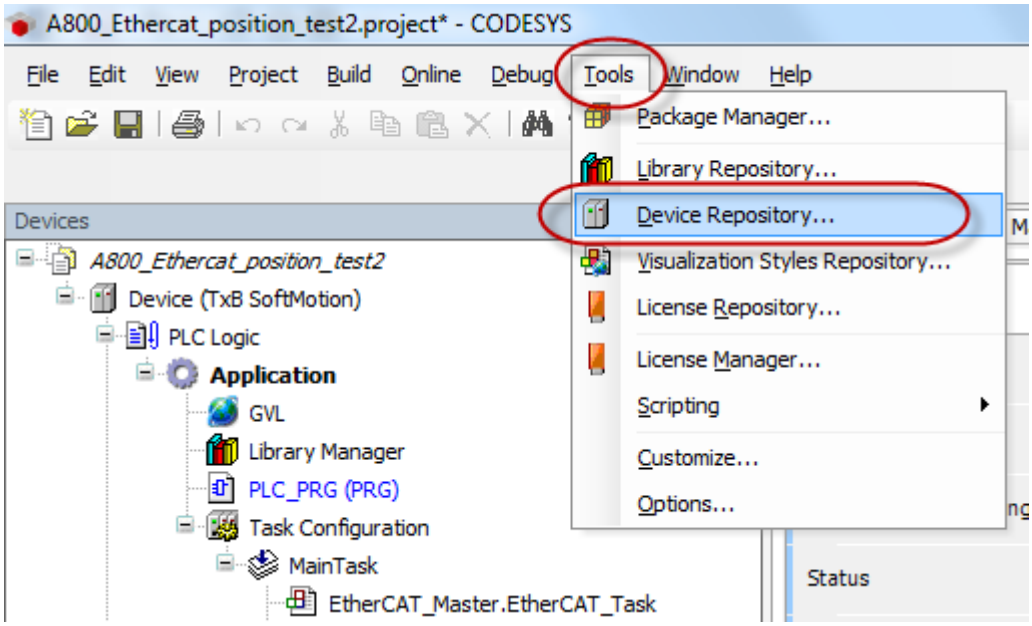
8. Click right mouse button on EtherCAT_Master and add device. Use vendor Mitsubishi Electric. If device is not found please follow section 3.5.1 for instructions on how to install the ESI file.

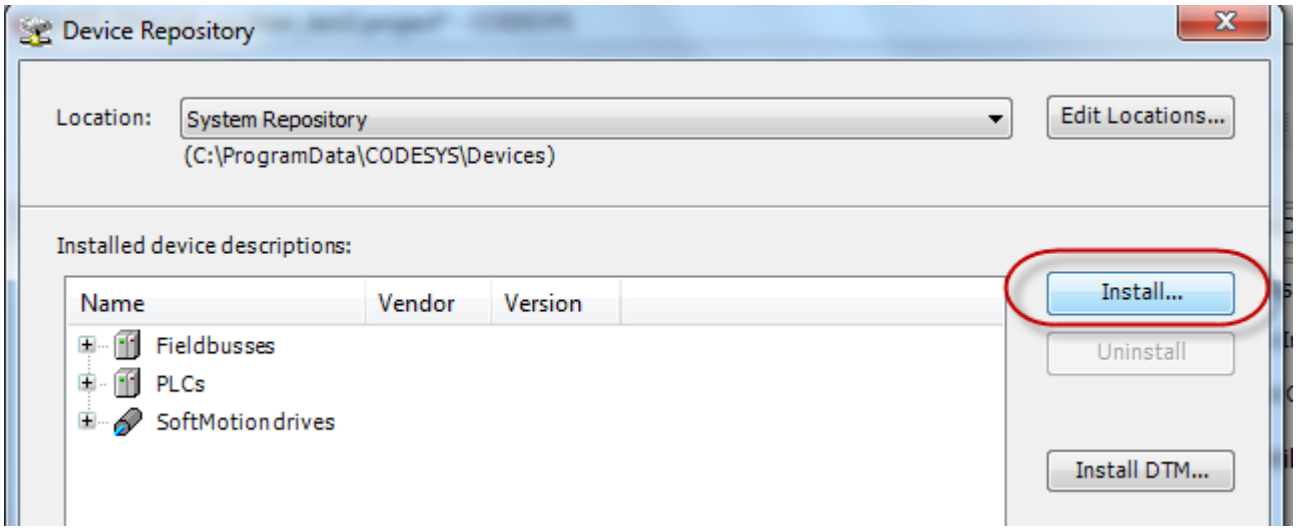


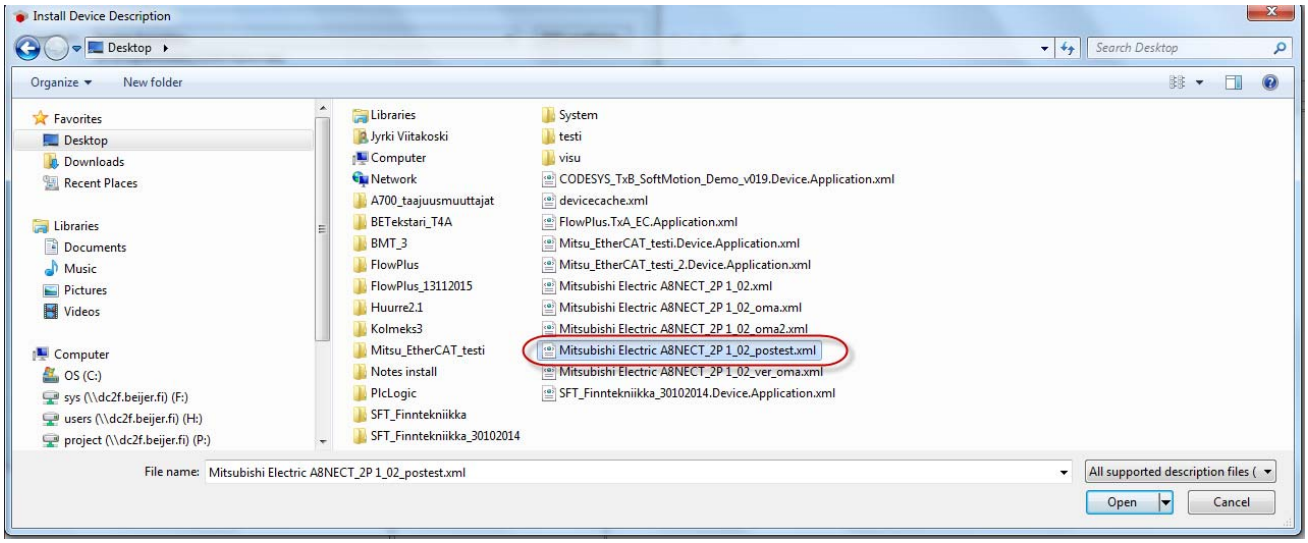
9. Double click EtherCAT Master and set parameters according to following example.



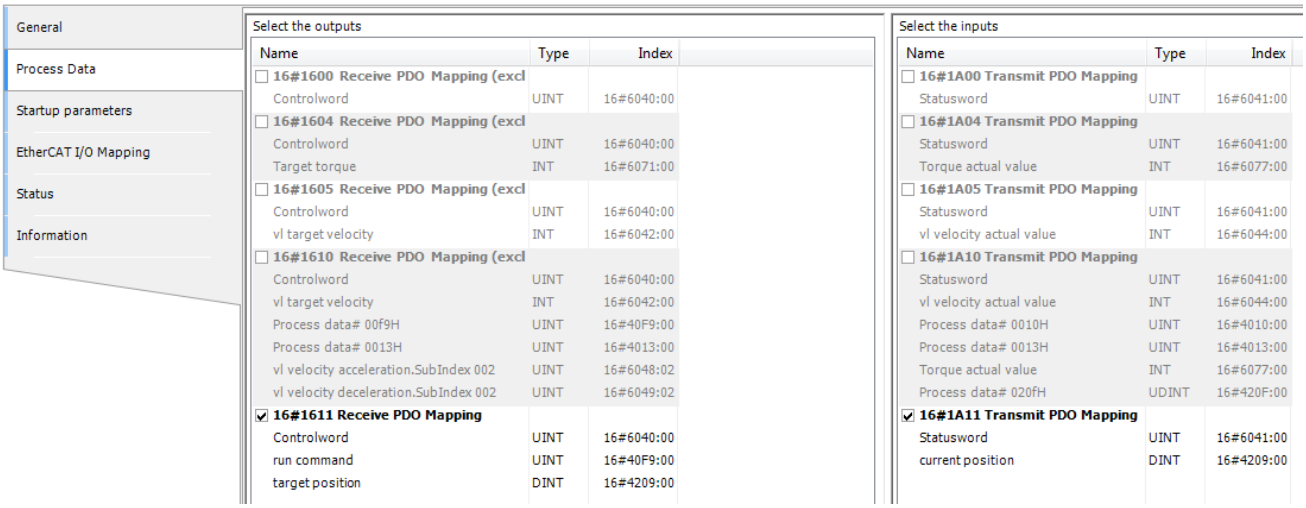
10. To be able to run inverter in positioning mode standard ESI file has to be modified. For instructions on modifying the ESI file and inverter parameter setting refer to section 3.7.3. New ESI file has to be changed in CoDeSys Device Repository according to following steps.



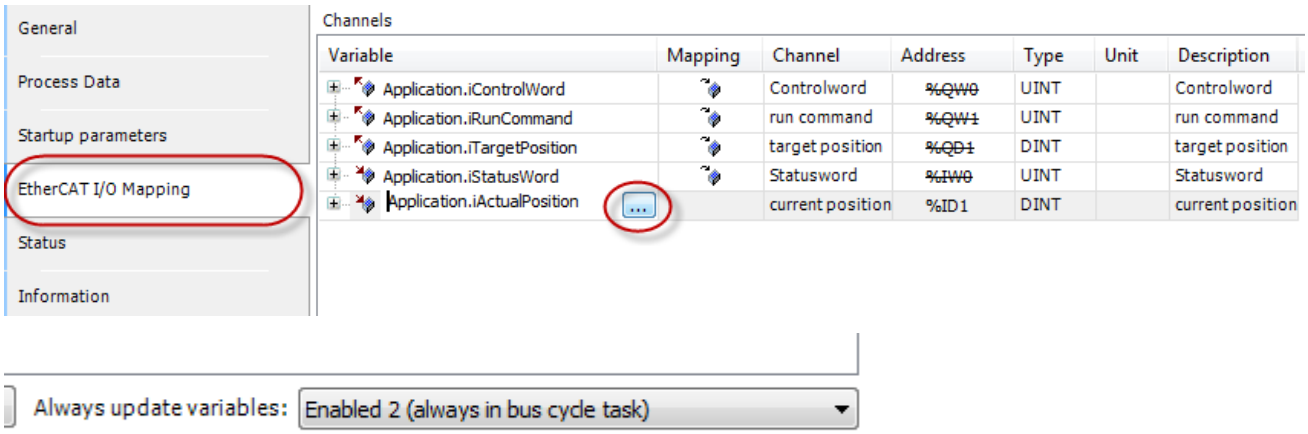




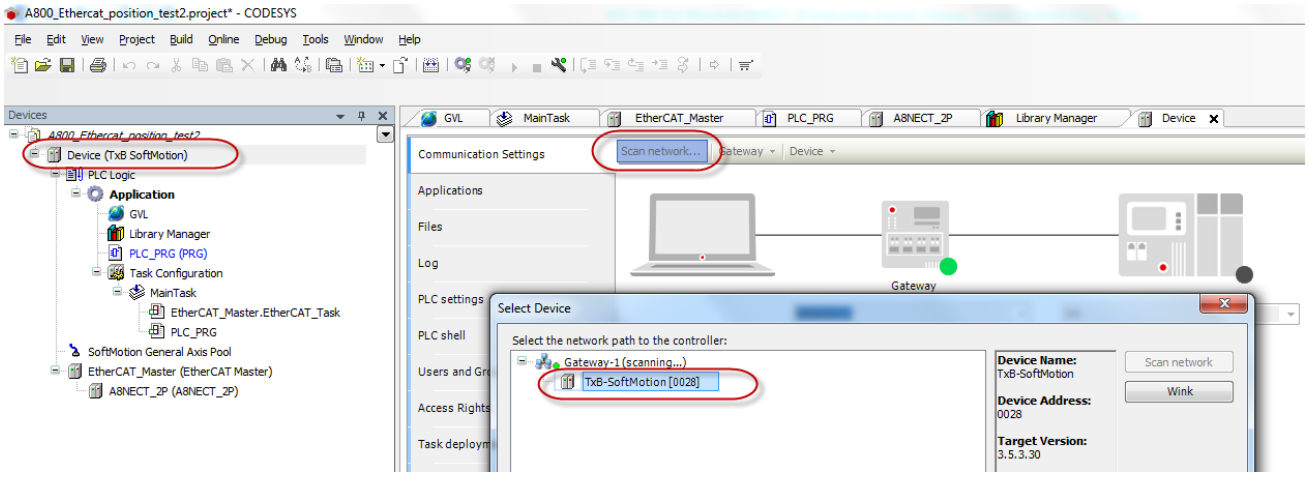
11. After ESI-file update new PDO Mappings should be available for use.



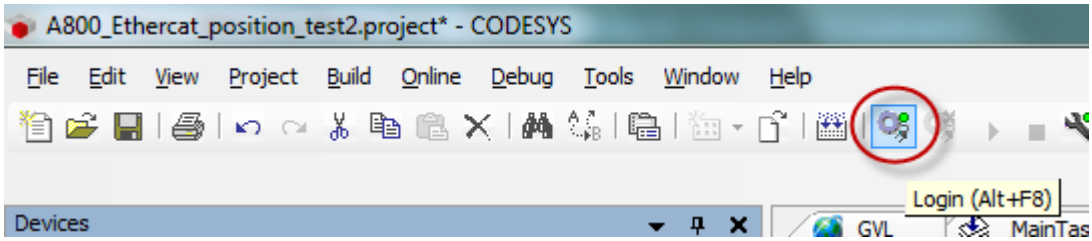
12. Pick correct Global Variables from GVL to EtherCAT I/O Mapping. Set Always update variables -> Enabled 2.



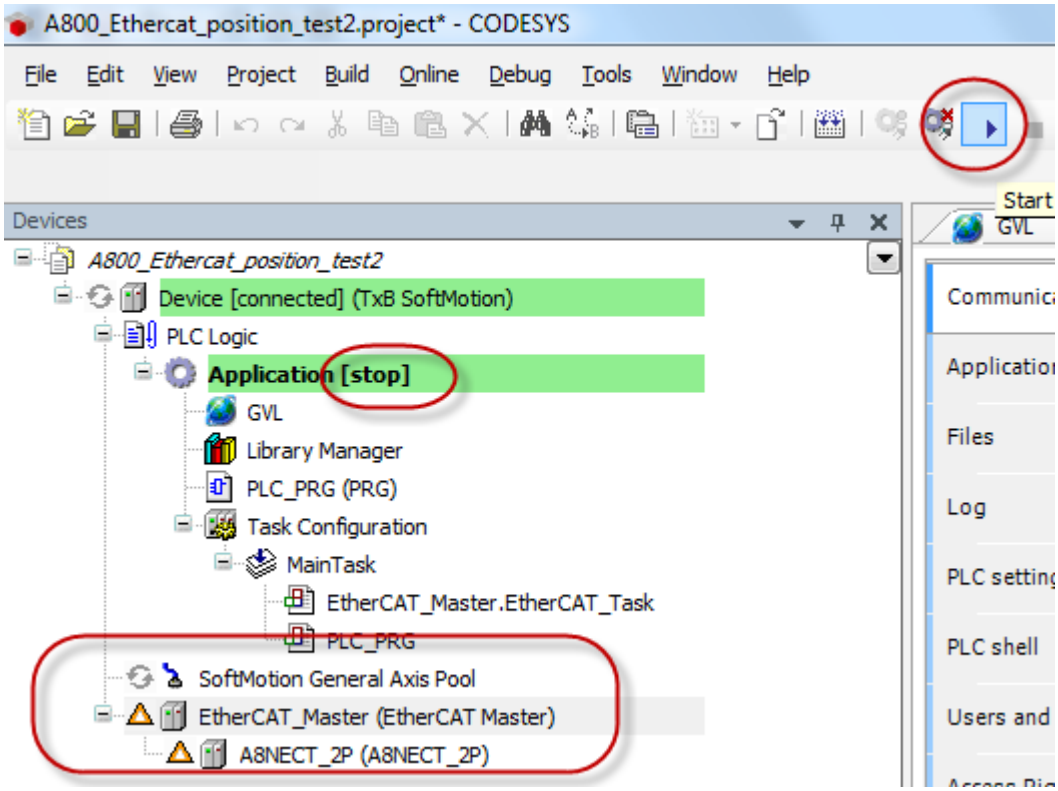
13. Double click Device (TxB SoftMotion) and Scan Network. If PLC is connected, you will find it as follows.

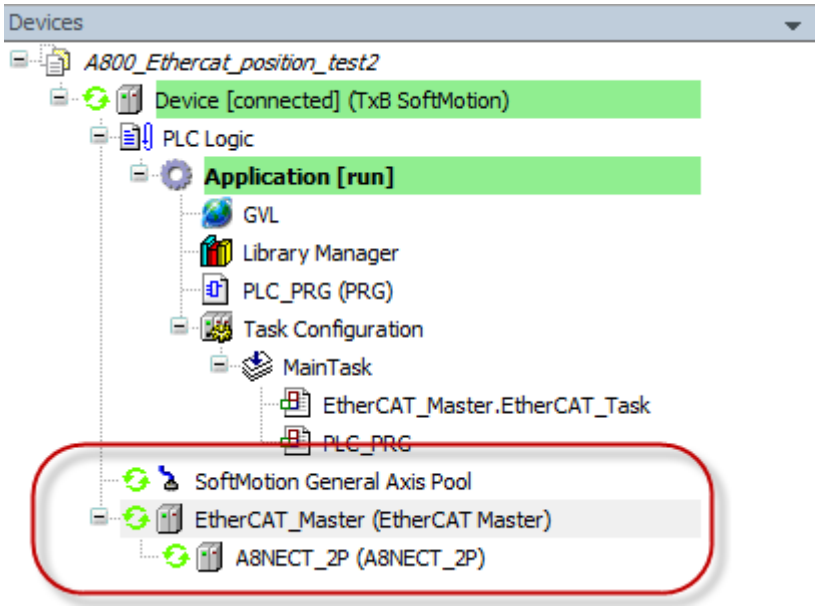


14. Login to PLC.

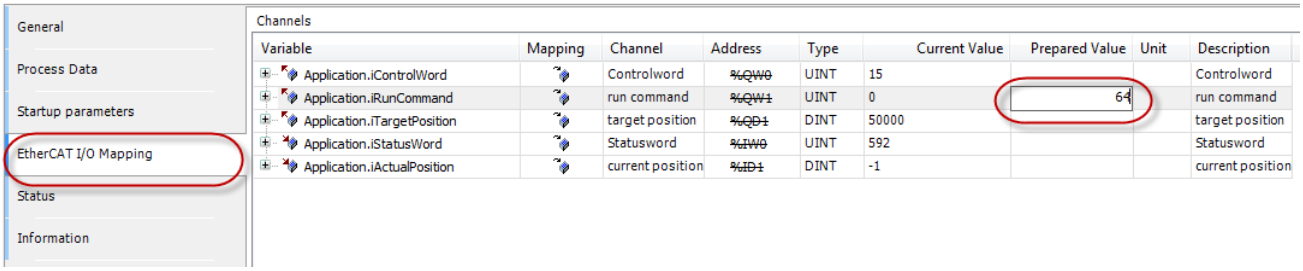


15. Start the PLC. EtherCAT communication indicators should change to green.





16. You are now able to test positioning by writing correct values to EtherCAT slave I/O mapping. CTRL + F7 writes new Prepared values. Please follow paragraph 3.7.3.5 with Controlword and RunCommand settings.



3.7 Quick setup guides

This section explains step by step how to configure the system to control the motor in three modes: velocity, torque and position mode. All examples build upon the section 3.5 TwinCAT 3.1 Configuration example but can be applied to CoDeSys as well.

3.7.1 Velocity mode

This example uses velocity mode that is part of CANopen DSP 402 drive profile

3.7.1.1 Inverter settings

Inverter must be configured to work in network operation mode and control mode must be set to speed control.

1. Set Pr.340 = 10, Pr. 79 = 0 – NET operation mode after power ON.
2. Set Pr.800 = 0 – speed control mode.
3. Acceleration and deceleration time can be set in Pr.7 and Pr.8 respectively.
4. Restart the inverter.

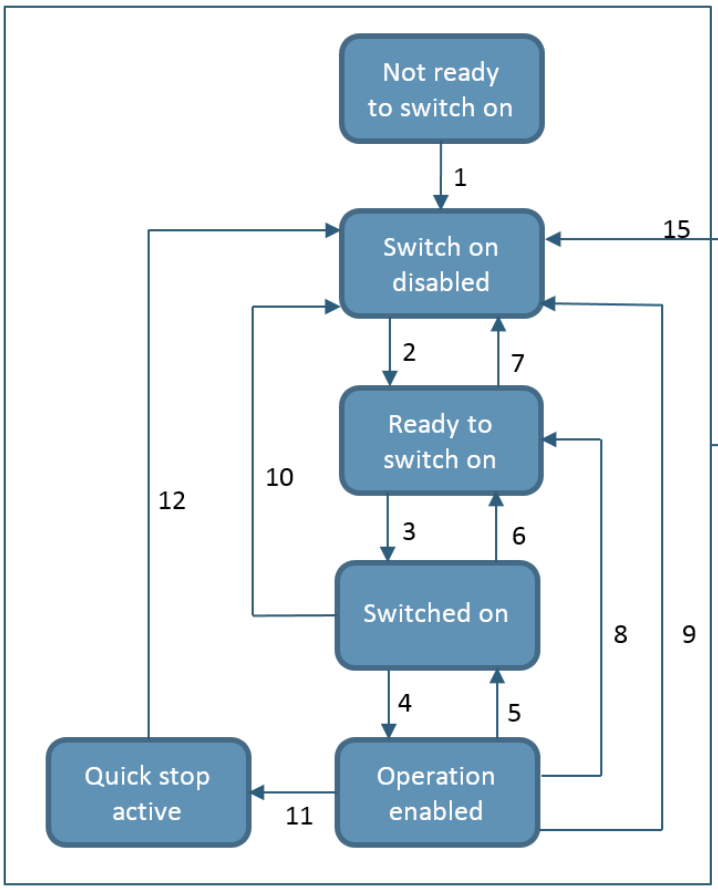
For further information about inverter parameters and speed control modes refer to inverter instruction manual.

3.7.1.2 EtherCAT master configuration

For master configuration follow section 3.5 TwinCAT 3.1 Configuration example as it uses velocity control mode as example.

3.7.1.3 Velocity mode operation example

Below is general state diagram. Transitions between states are dependent on value on controlword and current state is signaled by the value of status word.



Command	Bits of the controlword					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	X	1	1	0	2, 6, 8
Switch on	0	0	1	1	1	3
Switch on + Enable operation	0	1	1	1	1	3 + 4
Disable voltage	0	X	X	0	X	7, 9, 10, 12
Quick stop	0	X	0	1	X	7, 10, 11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4
Fault reset	0 -> 1	X	X	X	X	15

Bits of the statusword						State
Bit 6	Bit 5	Bit 3	Bit 2	Bit 1	Bit 0	
0	X	0	0	0	0	Not ready to switch on
1	X	0	0	0	0	Switch on disabled
0	1	0	0	0	1	Ready to switch on
0	1	0	0	1	1	Switched on
0	1	0	1	1	1	Operation enabled
0	0	0	1	1	1	Quick stop active
0	X	1	1	1	1	Fault reaction active
0	X	1	0	0	0	Fault

To perform operation system needs to be transferred to “enable operation” state. For detailed explanation of the CANopen state machine see section 5.8.2.26.1 DS402 state machine.

1. After the master configuration is finished and PLC is started system should be in “Switch on disabled” state – statusword bit 4 ,6, 9 ON.
   To transition to “ready to switch on” state set controlword value to 6 (bit 1, 2 ON).
   To write value to variable put it in “prepared value” field and select PLC-> “write values to all online applications”.

MAIN [Online]

TwinCAT_Device.testPLC1.MAIN

Expression	Type	Value	Prepared value	Address
iControlWord	UINT	0	6	%Q*
iTargetVelocity	INT	200		%Q*
iStatusWord	UINT	592		%I*
iActualVelocity	INT	0		%I*

2. Statusword should change to 561 (bit 0, 4, 5, 9 ON). To transition to “switch on” state set controlword value to 7 (bit 0, 1, 2 ON).

MAIN [Online]

TwinCAT_Device.testPLC1.MAIN

Expression	Type	Value	Prepared value	Address
iControlWord	UINT	6	7	%Q*
iTargetVelocity	INT	200		%Q*
iStatusWord	UINT	561		%I*
iActualVelocity	INT	0		%I*

3. Statusword should change to 563 (bit 0, 1 ,4, 5, 9 ON). Set target velocity to desired value in rpm. Sign of the velocity command determines the direction of the rotation, negative value results in reverse rotation. To enable operation set controlword value to 15 (bit 0, 1, 2, 3 ON).

MAIN [Online]

TwinCAT_Device.testPLC1.MAIN

Expression	Type	Value	Prepared value	Address
iControlWord	UINT	7	15	%Q*
iTargetVelocity	INT	200	600	%Q*
iStatusWord	UINT	563		%I*
iActualVelocity	INT	0		%I*

4. Statusword should change to 1591 (bit 0, 1, 2, 4, 5, 9, 10 is ON), motor is running with set frequency.

MAIN [Online]

TwinCAT_Device.testPLC1.MAIN

Expression	Type	Value	Prepared value
iControlWord	UINT	15	
iTargetVelocity	INT	600	
iStatusWord	UINT	1591	
iActualVelocity	INT	601	

3.7.2 Torque mode

This example uses torque mode that is part of CANopen DSP 402 drive profile

3.7.2.1 Inverter settings

Inverter must be configured to work in network operation mode and control mode must be set to torque control.

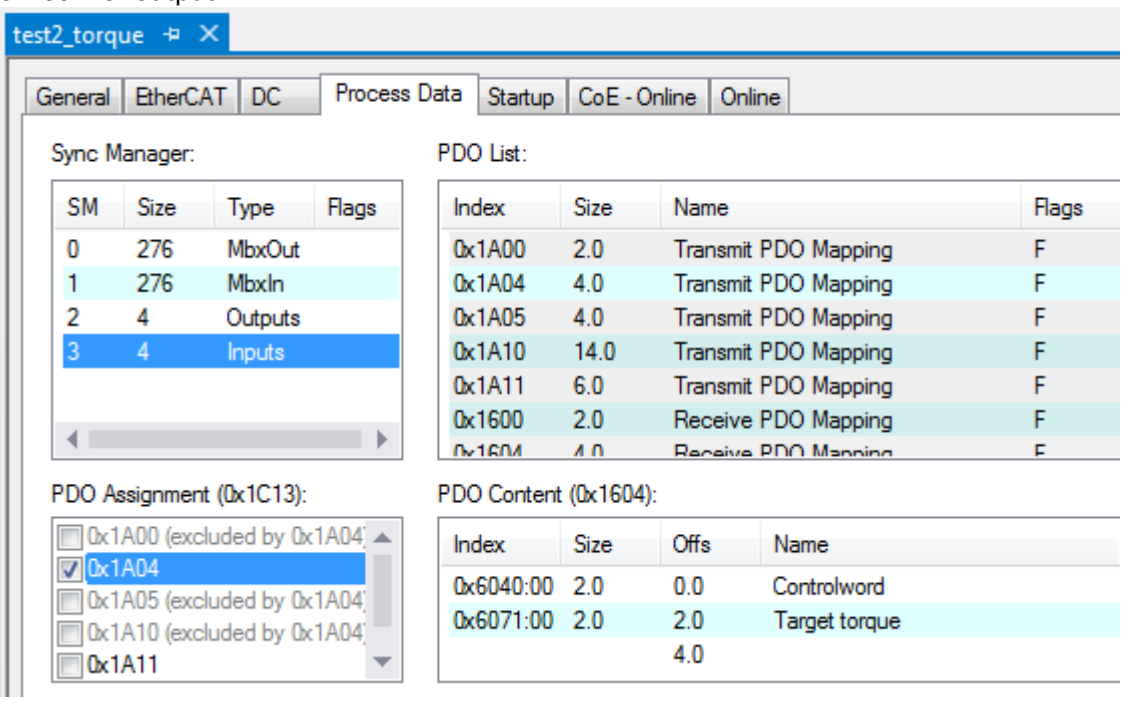
1. Inverter auto tuning must be performed before torque control mode can be activated. For details on how to perform Inverter offline auto tuning refer to section 5.13.2 of A800 instruction manual.
2. Set Pr.340 = 10, Pr. 79 = 0 – NET operation mode after power ON.
3. Set Pr.800 = 1 – torque control mode.
4. Set Pr.804 = 1 – torque command by parameter setting.
5. Set Pr.807 = 1 – forward and reverse speed limit set by Pr.808 and Pr.809 respectively.
6. Restart the inverter.

For further information about inverter parameters and torque control modes refer to inverter instruction manual.

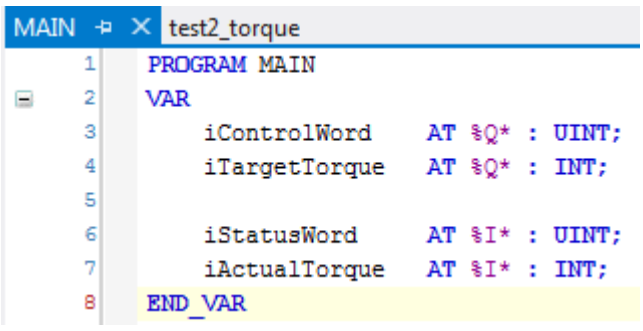
3.7.2.2 EtherCAT master configuration

For general master configuration follow section 3.5 TwinCAT 3.1 Configuration example. There are however two key differences when preparing project:

1. In process data configuration tab of the module properties select PDO 0x1A04 for Input and PDO 0x1604 for output.



2. Instead of variables defined in PLC project in the example enter following code:



After that perform same actions as described in example.

3.7.2.3 Torque mode operation example

To perform operation system needs to be transferred to “enable operation” state. For general state diagram see section 3.7.1.3 Velocity mode operation example. For detailed explanation of the CANopen state machine see section 5.8.2.26.1 DS402 state machine.

1. After the master configuration is finished and PLC is started system should be in “Switch on disabled” state – statusword bit 4 ,6, 9 ON.
   To transition to “ready to switch on” state set controlword value to 6 (bit 1, 2 ON).
   To write value to variable put it in “prepared value” field and select PLC-> “write values to all online applications”.

MAIN [Online] test2_torque

TwinCAT_Device.Untitled1.MAIN

Expression	Type	Value	Prepared value	Address
iControlWord	UINT	0	6	%Q*
iTargetTorque	INT	0		%Q*
iStatusWord	UINT	592		%I*
iActualTorque	INT	0		%I*

2. Statusword should change to 561 (bit 0, 4, 5, 9 ON). Set target torque to desired value (unit is 0,1% of rated torque). Sign of torque command determines the direction of the rotation, negative value results in reverse rotation. To transition to “switch on” state set controlword value to 7 (bit 0, 1, 2 ON).

MAIN [Online] test2_torque

TwinCAT_Device.Untitled1.MAIN

Expression	Type	Value	Prepared value	Address
iControlWord	UINT	6	7	%Q*
iTargetTorque	INT	0	100	%Q*
iStatusWord	UINT	561		%I*
iActualTorque	INT	0		%I*

3. Statusword should change to 563 (bit 0, 1 ,4, 5, 9 ON). To enable operation set controlword value to 15 (bit 0, 1, 2, 3 ON).

MAIN [Online] test2_torque

TwinCAT_Device.Untitled1.MAIN

Expression	Type	Value	Prepared value	Address
iControlWord	UINT	7	15	%Q*
iTargetTorque	INT	100		%Q*
iStatusWord	UINT	563		%I*
iActualTorque	INT	0		%I*

4. Statusword should change to 567 (bit 0, 1, 2, 4, 5, 9 is ON), motor is operating with set torque.

MAIN [Online] test2_torque

TwinCAT_Device.Untitled1.MAIN

Expression	Type	Value	Prepared value	Address
iControlWord	UINT	15		%Q*
iTargetTorque	INT	100		%Q*
iStatusWord	UINT	567		%I*
iActualTorque	INT	2		%I*

3.7.3 Position control

Note: By default module supports only velocity and torque control. Position control is possible, but requires special configuration and currently is not fully supported (in incremental positioning mode only forward movement is allowed)

This example describes how to configure the system to perform position control. There is no predefined PDOs for position control and it requires custom PDO mapping to be used.

3.7.3.1 Mapped items

For detailed explanation of custom PDO mapping and list of monitor items that can be mapped refer to section 4.2.2 User specific process data mapping.

In this example following items will be mapped to PDOs:

TxPDO:

- Status word
- Actual position value
- Output terminal status

RxPDO:

- control word
- Run command
- Target position
- Target velocity

Run command:

b15														b0
-	-	-	-	RES	STP (STOP)	CS	JOG	MRS	RT	RH	RM	RL	-	-	AU

Output terminal status:

b15														b0
—	—	—	—	—	—	—	—	SO	ABC2	ABC1	FU	OL	IPF	SU	RUN

, and set option board parameters on the inverter.

3.7.3.2 Modifying the ESI file

To use custom PDO mapping user needs to modify the A8NECT_2P ESI file. Following changes need to be made to the file:

After other RxPDO definitions:

```
<RxPdo Fixed="1">
  <Index>#x1611</Index>
  <Name>Receive PDO Mapping</Name>
  <Exclude>#x1600</Exclude>
  <Exclude>#x1604</Exclude>
  <Exclude>#x1605</Exclude>
  <Exclude>#x1610</Exclude>
  <Entry>
    <Index>#x6040</Index>
    <SubIndex>0</SubIndex>
    <BitLen>16</BitLen>
    <Name>Controlword</Name>
    <DataType>UINT</DataType>
  </Entry>
  <Entry>
    <Index>#x40F9</Index>
    <SubIndex>0</SubIndex>
    <BitLen>16</BitLen>
    <Name>run command</Name>
    <DataType>UINT</DataType>
  </Entry>
  <Entry>
    <Index>#x4209</Index>
    <SubIndex>0</SubIndex>
    <BitLen>32</BitLen>
    <Name>target position</Name>
    <DataType>DINT</DataType>
  </Entry>
  <Entry>
    <Index>#x6042</Index>
    <SubIndex>0</SubIndex>
    <BitLen>16</BitLen>
    <Name>vl target velocity</Name>
    <DataType>INT</DataType>
   </Entry>
</RxPdo>
```

After other TxPDO definitions:

```
<TxPdo Fixed="1">
  <Index>#x1A11</Index>
  <Name>Transmit PDO Mapping</Name>
  <Exclude>#x1A00</Exclude>
  <Exclude>#x1A04</Exclude>
  <Exclude>#x1A05</Exclude>
  <Exclude>#x1A10</Exclude>
  <Entry>
    <Index>#x6041</Index>
    <SubIndex>0</SubIndex>
    <BitLen>16</BitLen>
    <Name>Statusword</Name>
    <DataType>UINT</DataType>
  </Entry>
  <Entry>
    <Index>#x4209</Index>
    <SubIndex>0</SubIndex>
    <BitLen>32</BitLen>
    <Name>current position</Name>
    <DataType>DINT</DataType>
  </Entry>
  <Entry>
    <Index>#x4010</Index>
    <SubIndex>0</SubIndex>
```

```
    <BitLen>16</BitLen>
    <Name>Output terminal status</Name>
    <DataType>UINT</DataType>
  </Entry>
</TxPdo>
```

Save the file and use it instead of original ESI file for the rest of the example.

3.7.3.3 Inverter settings

Inverter must be configured to work in network operation mode and control mode must be set to torque control.

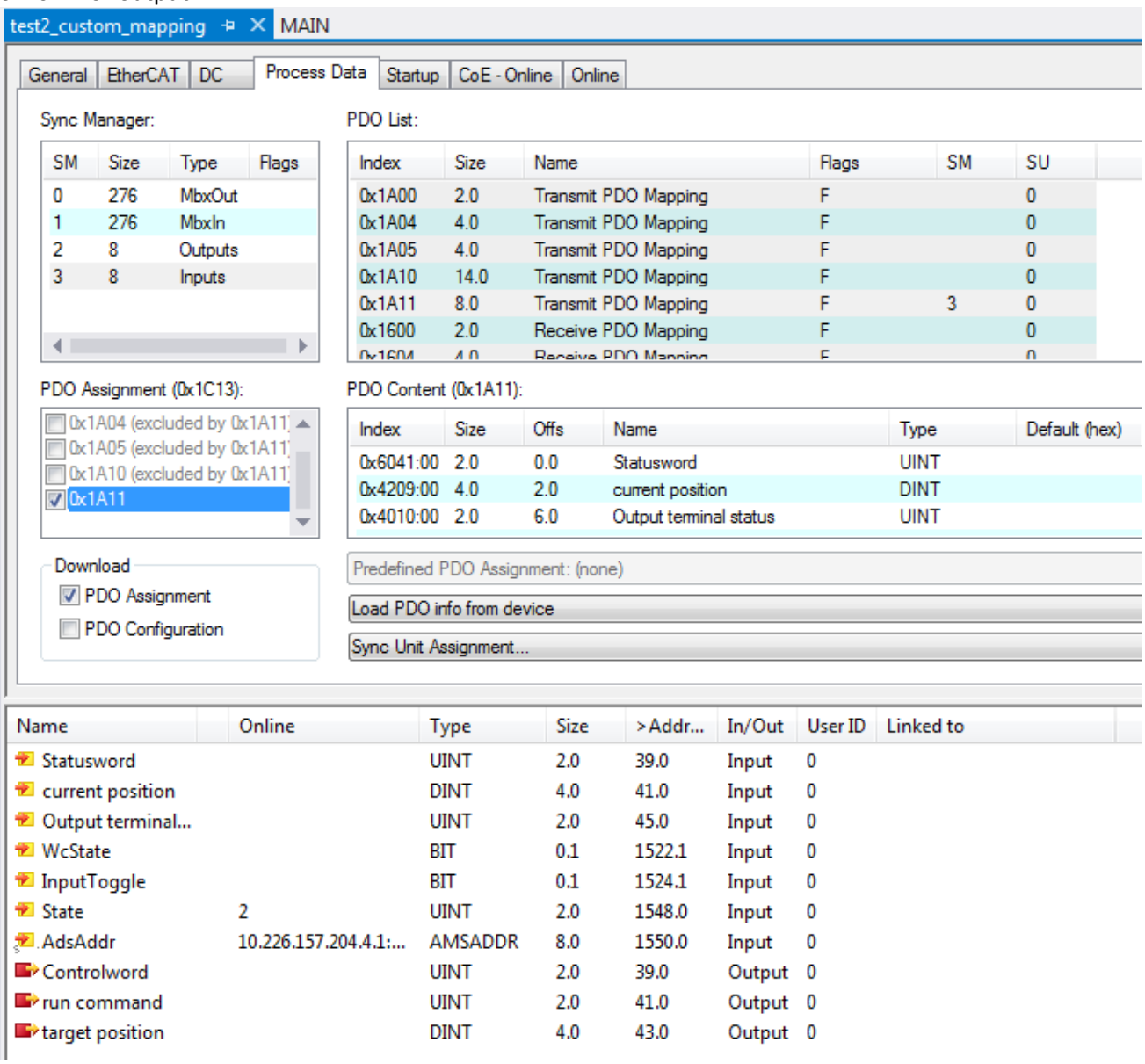
1. Inverter auto tuning must be performed before torque control mode can be activated. For details on how to perform Inverter offline auto tuning refer to section 5.13.2 of A800 instruction manual.
2. Set Pr.340 = 10, Pr. 79 = 0 – NET operation mode after power ON.
3. Set Pr.800 = 3 – position control mode.
4. Set Pr.419 = 0 – positioning by point tables.
5. Set Pr.1220 = 2 – Direct position command, direct maximum speed command.
6. Set acceleration and deceleration time in Pr.1222 and Pr.1223 respectively.
7. Set Pr.183 = 23 – assign LX (servo-ON) signal to the RT input terminal.
8. Set Pr.192 = 36 – assign in-position signal to the IPF output terminal
9. Set Pr.1282 = 4 – servo-ON position as the home position.
10. Set Pr.1225 = 0 for absolute positioning or 10 for incremental.
11. Set Pr.1300 = 1 – allow the use of custom PDO.
12. Set Pr.1306 = 1026 – map control word as first subindex in RxPDO
13. Set Pr.1307 = 249 – map run command as second subindex in RxPDO
14. Set Pr.1308 = 521 – map Target position as third subindex in RxPDO
15. Set Pr.1309 = 1029 – map target velocity as fourth subindex in RxPDO
16. Set Pr.1316 = 1027 – map status word as first subindex in TxPDO
17. Set Pr.1317 = 521 – map actual position as second subindex in TxPDO
18. Set Pr.1318 = 16 – map output terminal status as third subindex in TxPDO
19. Power cycle the inverter.

For further information about inverter parameters and position control modes refer to inverter instruction manual.

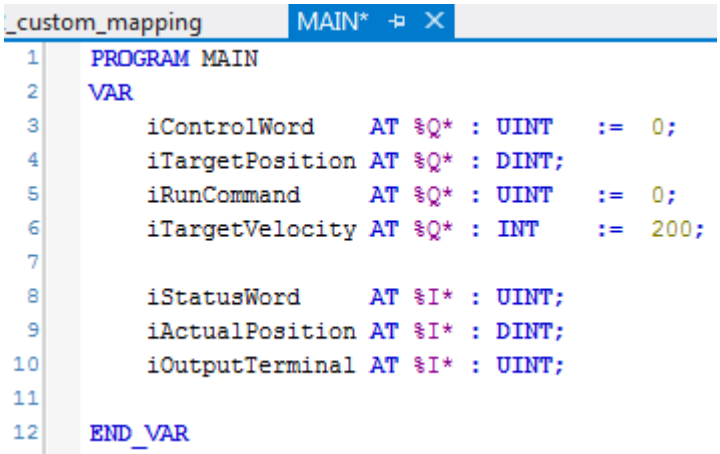
3.7.3.4 EtherCAT master configuration

For general master configuration follow section 3.5 TwinCAT 3.1 Configuration example using modified ESI file. There are however two key differences when preparing project:

1. In process data configuration tab of the module properties select PDO 0x1A11 for Input and PDO 0x1611 for output.



2. Instead of variables defined in PLC project in the example enter following code:



After that perform same actions as described in example.

3.7.3.5 Position control operation example

To perform operation system needs to be transferred to “enable operation” state. For general state diagram see section 3.7.1.3 Velocity mode operation example. For detailed explanation of the CANopen state machine see section 5.8.2.26.1 DS402 state machine.

1. After the master configuration is finished and PLC is started system should be in “Switch on disabled” state – statusword bit 4 ,6, 9 ON.
   To transition to “ready to switch on” state set controlword value to 6 (bit 1, 2 ON).
   To write value to variable put it in “prepared value” field and select PLC-> “write values to all online applications”.

MAIN [Online] test2_custom_mapping

TwinCAT_Device.Untitled1.MAIN

Expression	Type	Value	Prepared value	Address
iControlWord	UINT	0	6	%Q*
iTargetPosition	DINT	0		%Q*
iRunCommand	UINT	0		%Q*
iStatusWord	UINT	592		%I*
iActualPosition	DINT	0		%I*
iOutputTerminal	UINT	128		%I*

2. Statusword should change to 561 (bit 0, 4, 5, 9 ON). To transition to “switch on” state set controlword value to 7 (bit 0, 1, 2 ON).

MAIN [Online] test2_custom_mapping

TwinCAT_Device.Untitled1.MAIN

Expression	Type	Value	Prepared value	Address
iControlWord	UINT	6	7	%Q*
iTargetPosition	DINT	0		%Q*
iRunCommand	UINT	0		%Q*
iStatusWord	UINT	561		%I*
iActualPosition	DINT	0		%I*
iOutputTerminal	UINT	128		%I*

3. Statusword should change to 563 (bit 0, 1 ,4, 5, 9 ON). Write desired value to target position and turn LX signal ON by setting bit 6 of run command (write value 64).

MAIN [Online] test2_custom_mapping

TwinCAT_Device.Untitled1.MAIN

Expression	Type	Value	Prepared value	Address
iControlWord	UINT	7		%Q*
iTargetPosition	DINT	0	50000	%Q*
iRunCommand	UINT	0	64	%Q*
iStatusWord	UINT	563		%I*
iActualPosition	DINT	0		%I*
iOutputTerminal	UINT	128		%I*

4. Turn RH signal ON by setting bit 5 of run command (write value 96)

MAIN [Online] test2_custom_mapping

TwinCAT_Device.Untitled1.MAIN

Expression	Type	Value	Prepared value	Address
iControlWord	UINT	7		%Q*
iTargetPosition	DINT	50000		%Q*
iRunCommand	UINT	64	96	%Q*
iStatusWord	UINT	563		%I*
iActualPosition	DINT	-2		%I*
iOutputTerminal	UINT	149		%I*

5. To enable operation set controlword value to 15 (bit 0, 1, 2, 3 ON).

MAIN [Online] test2_custom_mapping

TwinCAT_Device.Untitled1.MAIN

Expression	Type	Value	Prepared value	Address
iControlWord	UINT	7	15	%Q*
iTargetPosition	DINT	50000		%Q*
iRunCommand	UINT	96		%Q*
iStatusWord	UINT	563		%I*
iActualPosition	DINT	-1		%I*
iOutputTerminal	UINT	149		%I*

6. After enabling operation positioning will begin. In-position signal should turn off (bit 2 of output terminal status). After in-position bit turns ON again positioning is completed.
7. To position motor to next point write value 7 to control word, then write new target position and write 15 to control word again.

3.8 Parameter settings

Parameters can be set at run time either using SDO download requests towards the option board or by using the handheld parameter unit. The values of some of the parameters will not be valid until the inverter has been reset.

For SDO download requests to work towards drive parameters the drive must be in the NET operation mode. See section 4.3 Operation mode setting for more information on how to change the operation mode.

3.8.1 Parameter unit

Parameter data can be seen, and in certain cases changed from the parameter unit.

For more information, please consult (Mitsubishi Electric).

4 Inverter settings

4.1 Inverter parameters

The inverter parameters are critical for overall operation of the end-to-end communication system. Some of these parameters must be set to specific values, and some may have multiple allowable settings depending on the desired operation of the overall application. It is important to understand the manner in which the parameters will impact successful communications with, and control of the inverter.

The inverter parameters can be changed either via the handheld parameter unit or via the SDO download of the option board. The handheld parameter unit is described in (Mitsubishi Electric), and parameter access via SDO is described in section 5.10.1.

Note: The scaling of the parameter values are different if they are accessed using the parameter unit or the SDO functionality, for further information see section 5.10.1.

4.2 Communication Option Parameters

The communication option parameters are stored in the inverter. At startup they are transferred from the inverter to the option board and can be changed using the parameter unit or by SDO download requests towards the option board.

Note that option board parameter 1305 can't be changed via SDO download request.

In the NET operation mode the option board parameters cannot be set from the parameter unit. To change the option board parameters from the parameter unit, first change the operation mode to PU (see section 4.3 for more information) or set Pr.77 to 2. Setting Pr.77 to 2 makes it possible to change the option board parameters in any operation mode.

No (dec)	No (hex)	CoE object index	Parameter name	Description
1300	0x514	0x3514	Option parameter 1	General settings. See section 4.2.1 for more information.
1301	0x515	0x3515	Option parameter 2	Reserved
1302	0x516	0x3516	Option parameter 3	Network type During startup the option board will verify the value of this parameter. If it differs from the network type of the option board, the option board will change the parameter value to the actual network type and clear all other option parameters.
1303	0x517	0x3517	Option parameter 4	Serial number (32 bits) Parameter 1303 is the low word and 1304 is the high word of the serial number. During startup the option board will verify the value of this parameter. If it differs from the serial number of the option board, the option board will change the parameter value to the actual serial number.
1304	0x518	0x3518	Option parameter 5	
1305	0x519	0x3519	Option parameter 6	Device ID This parameter works as the ID selector used for explicit device identification functionality on EtherCAT. The option board will continuously poll this parameter to detect any changes to it without the need for a power cycle. The CoE object corresponding to this parameter (0x3519) is read only, so this parameter cannot be changed using SDO download requests.

No (dec)	No (hex)	CoE object index	Parameter name	Description
1306	0x51A	0x351A	Option parameter 7	User specific RxPDO data parameter #1. See section 4.2.2 for more information.
1307	0x51B	0x351B	Option parameter 8	User specific RxPDO data parameter #2. See section 4.2.2 for more information.
1308	0x51C	0x351C	Option parameter 9	User specific RxPDO data parameter #3. See section 4.2.2 for more information.
1309	0x51D	0x351D	Option parameter 10	User specific RxPDO data parameter #4. See section 4.2.2 for more information.
1310	0x51E	0x351E	Option parameter 11	User specific RxPDO data parameter #5. See section 4.2.2 for more information.
1311	0x51F	0x351F	Option parameter 12	User specific RxPDO data parameter #6. See section 4.2.2 for more information.
1312	0x520	0x3520	Option parameter 13	User specific RxPDO data parameter #7. See section 4.2.2 for more information.
1313	0x521	0x3521	Option parameter 14	User specific RxPDO data parameter #8. See section 4.2.2 for more information.
1314	0x522	0x3522	Option parameter 15	User specific RxPDO data parameter #9. See section 4.2.2 for more information.
1315	0x523	0x3523	Option parameter 16	User specific RxPDO data parameter #10. See section 4.2.2 for more information.
1316	0x524	0x3524	Option parameter 17	User specific TxPDO data parameter #1. See section 4.2.2 for more information.
1317	0x525	0x3525	Option parameter 18	User specific TxPDO data parameter #2. See section 4.2.2 for more information.
1318	0x526	0x3526	Option parameter 19	User specific TxPDO data parameter #3. See section 4.2.2 for more information.
1319	0x527	0x3527	Option parameter 20	User specific TxPDO data parameter #4. See section 4.2.2 for more information.
1320	0x528	0x3528	Option parameter 21	User specific TxPDO data parameter #5. See section 4.2.2 for more information.
1321	0x529	0x3529	Option parameter 22	User specific TxPDO data parameter #6. See section 4.2.2 for more information.
1322	0x52A	0x352A	Option parameter 23	User specific TxPDO data parameter #7. See section 4.2.2 for more information.
1323	0x52B	0x352B	Option parameter 24	User specific TxPDO data parameter #8. See section 4.2.2 for more information.
1324	0x52C	0x352C	Option parameter 25	User specific TxPDO data parameter #9. See section 4.2.2 for more information.
1325	0x52D	0x352D	Option parameter 26	User specific TxPDO data parameter #10. See section 4.2.2 for more information.
1326	0x52E	0x352E	Option parameter 27	User specific TxPDO data parameter #11. See section 4.2.2 for more information.
1327	0x52F	0x352F	Option parameter 28	User specific TxPDO data parameter #12. See section 4.2.2 for more information.
1328	0x530	0x3530	Option parameter 29	User specific TxPDO data parameter #13. See section 4.2.2 for more information.

No (dec)	No (hex)	CoE object index	Parameter name	Description
1329	0x531	0x3531	Option parameter 30	User specific TxPDO data parameter #14. See section 4.2.2 for more information.
1330-1343	0x532-0x53F	0x3532-0x353F	Option parameter 31-44	Reserved

4.2.1 Parameter Pr.1300, General settings

Setting bit	Description
0	User specific process data mapping enable. See section 4.2.2 for more information.
1-3	Reserved
4	Clear all option parameters. If this bit is set at power on all option board parameters will be set to default values. This bit is reset to 0 after completion.
5-15	Reserved

4.2.2 User specific process data mapping

If the user needs more parameters mapped to process data than what is offered in the standard PDO configurations it is possible to enable user specific process data configurations.

This is done in three steps:

1. Set bit 0 in parameter Pr.1300 to 1.
2. Add the parameters that shall be in user specific RxPDO configuration to parameters Pr.1306-1315. If e.g. two parameters are needed, set those parameter numbers to parameter Pr.1306 and Pr.1307 and set the rest to 0. See section 4.2.2.1 for more information on what parameters that can be added to user specific RxPDO configurations.
3. Add the parameters that shall be in user specific TxPDO configuration to parameters Pr.1316-1329. If e.g. two parameters are needed, set those parameter numbers to parameter Pr.1316 and Pr.1317 and set the rest to 0. See section 0 for more information on what parameters that can be added to user specific TxPDO configurations.
4. Power cycle the inverter otherwise changes will have no effect. Option board checks PDO mapping parameters at startup.

The option board will check these parameters and create two new PDO configuration objects if user specific process data mapping is enabled.

For output process data object 0x1611 is created and filled out with the CoE objects corresponding to the parameters added to parameters Pr.1306-1315.

For input process data object 0x1A11 is created and filled out with the CoE objects corresponding to the parameters added to parameters Pr.1316-1329.

4.2.2.1 Setting values that can be added to user specific RxPDO data

All monitor items with write access can be added to user specific RxPDO data. Consult “List of inverter monitored items” in Appendix4 “For customers using HMS network options” in (Mitsubishi Electric) for more information about these setting values. The monitor item number is then translated into a CoE object entry and mapped in object 0x1611. See section 5.10.1.2 for more information on parameter number translation.

It is also possible to add drive profile parameters to the user specific RxPDO data. The table below describes the setting values that should be written in parameters Pr.1306-1315 and what resulting CoE object will be mapped in object 0x1611.

Setting value	CoE object entry	Name	Data type	Size (bytes)
1026	0x6040:0	Controlword	UNSIGNED16	2
1029	0x6042:0	vl target velocity	SIGNED16	2
1042	0x6060:0	Modes of operation	SIGNED8	1
1045	0x6048:1	vl velocity acceleration (speed)	UNSIGNED32	4
1047	0x6048:2	vl velocity acceleration (time)	UNSIGNED16	2
1049	0x6049:1	vl velocity deceleration (speed)	UNSIGNED32	4
1051	0x6049:2	vl velocity deceleration (time)	UNSIGNED16	2
1061	0x6071:0	Target torque	SIGNED16	2

Important: The ESI file needs to be modified when PDO configuration object 0x1611 is enabled for EtherCAT masters to be able to use it. A new RxPdo-element containing the CoE object entry information corresponding to the setting values written to parameter Pr.1306-1315 needs to be added after the currently defined RxPdo-elements.

In this example the value 1026 (Controlword from the table above) was written to parameter Pr.1306, the value 19 (0x13) (meaning monitor item 19) was written to parameter Pr.1307 and parameter Pr.1308-1315 was set to 0. That means the following RxPdo-element needs to be added:

```
<RxPdo Fixed="1">
   <Index>#x1611</Index>
   <Name>Receive PDO Mapping</Name>
   <Entry>
      <Index>#x6040</Index>
      <SubIndex>0</SubIndex>
      <BitLen>16</BitLen>
      <Name>Controlword</Name>
      <DataType>UINT</DataType>
   </Entry>
   <Entry>
      <Index>#x4013</Index>
      <SubIndex>0</SubIndex>
      <BitLen>16</BitLen>
      <Name>Process data# 0013H</Name>
      <DataType>UINT</DataType>
   </Entry>
</RxPdo>
```

4.2.2.2 Setting values that can be added to user specific TxPDO data

All monitor items with read access can be added to user specific TxPDO data. Consult “List of inverter monitored items” in Appendix4 “For customers using HMS network options” in (Mitsubishi Electric) for more information about these setting values. The monitor item number is then translated into a CoE object entry and mapped in object 0x1A11. See section 5.10.1.2 for more information on parameter number translation.

It is also possible to add drive profile parameters to the user specific TxPDO data. The table below describes the setting value that should be written in parameters Pr.1316-1329 and what resulting CoE object will be mapped in object 0x1A11.

Setting value	CoE object entry	Name	Data type	Size (bytes)
1027	0x6041:0	Statusword	UNSIGNED16	2
1030	0x6043:0	vl velocity demand	SIGNED16	2
1031	0x6044:0	vl velocity actual value	SIGNED16	2
1042	0x6061:0	Modes of operation display	SIGNED8	1
1062	0x6077:0	Actual torque	SIGNED16	2

Important: The ESI file needs to be modified when PDO configuration object 0x1A11 is enabled for EtherCAT masters to be able to use it. A new TxPdo-element containing the CoE object entry information corresponding to the parameter numbers written to parameter Pr.1316-1329 needs to be added after the currently defined TxPdo-elements.

In this example the value 1027 (Statusword from the table above) was written to parameter Pr.1316, the value 1 (meaning monitor item 1) was written to parameter Pr.1317 and parameter Pr.1318-1329 was set to 0. That means the following TxPdo-element needs to be added:

```
<TxPdo Fixed="1">
   <Index>#x1A11</Index>
   <Name>Transmit PDO Mapping</Name>
   <Entry>
      <Index>#x6041</Index>
      <SubIndex>0</SubIndex>
      <BitLen>16</BitLen>
      <Name>Statusword</Name>
      <DataType>UINT</DataType>
   </Entry>
   <Entry>
      <Index>#x4001</Index>
      <SubIndex>0</SubIndex>
      <BitLen>16</BitLen>
      <Name>Process data# 0001H</Name>
      <DataType>UINT</DataType>
   </Entry>
</TxPdo>
```

4.3 Operation mode setting

To be able to write parameters in the inverter from EtherCAT the operation mode of the inverter needs to be set to NET operation mode (Refer to Pr.340 in the (Mitsubishi Electric)).

5 EtherCAT functionality

5.1 EtherCAT Slave Controller

The EtherCAT Slave Controller (ESC) handles all low level communication in an EtherCAT slave. The A8NECT_2P option board is using the HMS ESC developed by HMS.

The HMS ESC has 16 kB of RAM starting at address 0x1000 that can be accessed through sync managers.

5.1.1 ESC registers

The following registers are supported by the HMS ESC:

Address	Length (Byte)	Description	EtherCAT Access	Reset value
0x0000	1	Type	R	0xB0
0x0001	1	Revision	R	0x01
0x0002 to 0x0003	2	Build	R	0x0300
0x0004	1	FMMU supported	R	4
0x0005	1	SYNC managers supported	R	4
0x0006	1	RAM size (Kbyte)	R	0x10
0x0007	1	Port descriptor	R	0x0F
0x0008 to 0x0009	2	Features supported	R	0x00CC
0x0010 to 0x0011	2	Configured station address	RW	0
0x0012 to 0x0013	2	Configured station alias	R	EEPROM addr 0x0004
0x0100 to 0x0101	2	DL control	RW	0x0001
0x0102 to 0x0103	2	Extended DL Control	RW	0x0000
0x0108 to 0x0109	2	Physical Read/Write offset	RW	0
0x0110 to 0x0111	2	DL Status	R	0x5005
0x0120 to 0x0121	2	AL Control	RW	0x0001
0x0130 to 0x0131	2	AL Status	R	0x0001
0x0134 to 0x0135	2	AL Status code	R	0
0x0140	1	PDI Control	R	0x80
0x0141	1	ESC Configuration	R	0x36
0x0150 to 0x0153	4	PDI Configuration	R	0x00004600
0x0200 to 0x0201	2	ECAT interrupt mask	RW	0x0000
0x0204 to 0x0207	4	AL Event mask	R	0x00000331
0x0210 to 0x0211	2	ECAT Event Request	R	0x0000
0x0220 to 0x0223	4	AL Event Request	R	0x00000000
0x0300 to 0x0307	4x2	RX-Error counter[3:0]	RW(Writable with value 0 to clear counter)	0 Counter [3:2] not supported

Address	Length (Byte)	Description	EtherCAT Access	Reset value
0x0308 to 0x030B	4x1	Forwarded RW error counter [3:0]	RW(Writable with value 0 to clear counter)	0 Counter [3:2] not supported
0x0310 to 0x0313	4x1	Lost-Link counter[3:0]	RW(Writable with value 0 to clear counter)	0 Counter [3:2] not supported
0x0400 to 0x0401	2	Watchdog divider	RW	2498
0x0420 to 0x0421	2	Watchdog time Sync manager	RW	1000
0x0440 to 0x0441	2	Watchdog Sync manager status	R	0x0000
0x0500 to 0x050F	16	EEPROM interface	R/RW	EEPROM emulation
0x0600 to 0x063F	16x4	FMMU[15:0]	RW	4 FMMU supported
0x0800 to 0x083F	16x4	Sync-manager[15:0]	R/RW	4 SyncM supported
0x0900 to 0x090F	4x4	DC – Receive Times [3:0]	R/RW	[3:2] not supported
0x0910 to 0x0917	8	DC – System Time	R/RW	
0x0918 to 0x091F	8	DC – Receive Time EPU	R	
0x0920:0x0935	24	DC – Time Loop Control Unit	RW	
0x0980	1	DC – Cyclic Unit Control	RW	
0x0981 to 0x0983	3	DC – SYNC Out Unit	R/RW	
0x0984	1	DC – Activation status	R	
0x098E to 0x09A7	26	DC – SYNC Out Unit	R/RW	

A more detailed description of these registers can be found in (Beckhoff). This document can be downloaded from the Beckhoff homepage, www.beckhoff.com.

5.1.2 Supported EtherCAT commands

The HMS ESC supports the following EtherCAT commands:

- Auto increment physical read (APRD)
- Auto increment physical write (APWR)
- Configured address read (FPRD)
- Configured address write (FPWR)
- Configured address Read Write (FPRW)
- Broadcast read (BRD)
- Broadcast write (BWR)
- Logical read (LRD)
- Logical write (LWR)
- Logical Read Write (LRW)
- Auto increment physical read multiple write (ARMW)
- Configured read multiple write (FRMW)

5.1.3 EEPROM content

The EEPROM of the ESC contains information of the option board called the SII (Slave Information Interface) which is available to the master. The master uses this information to configure the option board during network start up.

In the HMS ESC the EEPROM is not an actual EEPROM, instead it is emulated in software, but the interface from the network is the same.

The SII is divided in two different areas, the Slave Information Interface area and the Slave Information Interface categories.

All addresses indicated in the tables are word (16-bit) addresses. In the case where the address is repeated on two rows this indicates two byte values residing on one address.

5.1.3.1 Slave Information Interface area

The slave information interface area contains the following information:

Address	Description	Data	Comment
0x0000	PDI Control	0x3680	Enhanced link detection on port 0 and 1, Distributed clocks SYNC Out unit enabled, On-chip bus.
0x0001	PDI Configuration	0x4600	
0x0002	SyncImpulseLen	0x01F4	5 µs
0x0003	PDI Configuration2	0x0000	
0x0004	Configured station alias	0x0000	
0x0005	Reserved	0x0000	
0x0006	Reserved	0x0000	
0x0007	Checksum	0x002B	
0x0008	Vendor ID	0x00000A1E	Mitsubishi Electric corporation
0x0009			
0x000A	Product Code	0x02000101 in case of FR-A800 inverter 0x02000201 in case of FR-F800 inverter	Mitsubishi Electric product codes for A8NECT_2P
0x000B			
0x000C	Revision Number	-	SW version of the option board in the format 0xXXXX.YYYY where XXXX is the major revision and YYYY is the minor revision.
0x000D			
0x000E	Serial number	-	Serial number of the option board that is assigned during production.
0x000F			
0x0010	Reserved	0x0000	
0x0011	Reserved	0x0000	
0x0012	Reserved	0x0000	
0x0013	Reserved	0x0000	
0x0014	Bootstrap mbx rx offset	0x2000	
0x0015	Bootstrap mbx rx size	0x0400	
0x0016	Bootstrap mbx tx offset	0x2800	

Address	Description	Data	Comment
0x0017	Bootstrap mbx tx size	0x0400	
0x0018	Standard mbx rx offset	0x2000	
0x0019	Standard mbx rx size	0x0114	
0x001A	Standard mbx tx offset	0x2800	
0x001B	Standard mbx tx size	0x0114	
0x001C	Mailbox protocol	0x000C	CoE and FoE
0x001D to0x003D	Reserved	0	
0x003E	Size	0x0002	Size of EEPROM in kBit-1
0x003F	Version	0x0001	This is version 1

The information stored at address 0x0004 (configured station alias) and address 0x0007 (checksum) is kept after power cycle.

Writing any other address than 0x0004 and 0x0007 will be successful, but the new data will be ignored. This is because the HMS ESC is using an emulated EEPROM.

5.1.3.2 Slave Information Interface categories

A number of slave information interface categories are also present in the EEPROM.

5.1.3.2.1 Structure category String

Address	Description	Data (LE)	Comment
0x0040	Category type	0x000A	10 STRINGS
0x0041	Category word size	0x0006	5 words for device name “A8NECT_2P” + 1 word for category type.
0x0042:LSB	Number of strings	0x01	
0x0042:MSB	Length of string 1	0x09	
0x0043-0x0047	String 1	“A8NECT_2P”	A pad byte is added since the total length of the string structure is odd.

5.1.3.2.2 Structure category General

Address	Description	Data (LE)	Comment
0x0048	Category type	0x001E	30 General
0x0049	Category word size	0x0010	16 words
0x004A:LSB	GroupIdx	0x00	Group Information (Not used)
0x004A:MSB	ImgIdx	0x00	Image name (Not used)
0x004B:LSB	OrderIdx	0x01	Device order number (Same as device name)
0x004B:MSB	NameIdx	0x01	Device Name Information (idx to string “A8NECT_2P” in string structure).
0x004C:LSB	Reserved	0x00	
0x004C:MSB	CoE details	B00100111	Enable SDO, Enable SDO info, Enable PDO Assign, Enable SDO complete access.
0x004D:LSB	FoE details	0x01	FoE enabled

Address	Description	Data (LE)	Comment
0x004D:MSB	E0E details	0x00	EoE disabled
0x004E:LSB	SoE Channels	0x00	Reserved
0x004E:MSB	DS402 Channels	0x00	Reserved
0x004F:LSB	SysmanClass	0x00	Reserved
0x004F:MSB	Flags	0x00	Bit 0: Enable safeop Bit 1: Enable notLRW
0x0050	CurrentOnEBus	0x0000	
0x0051	PAD_bytes	0x0000	Reserved
0x0052	Physical port info	0x0011	2x100base-TX
0x0053-0x0059	PAD_bytes	0x0000	Reserved

5.1.3.2.3 Structure category FMMU

Address	Description	Data (LE)	Comment
0x005A	Category type	0x0028	40 FMMU
0x005B	Category word size	0x0002	2 words
0x005C:LSB	FMMU0	0x01	Output I/O
0x005C:MSB	FMMU1	0x02	Input I/O
0x005D	FMMU2	0x03	Read Mailbox

5.1.3.2.4 Structure category SyncM

Address	Description	Data (LE)	Comment
0x005E	Category type	0x0029	41 SyncM
0x005F	Category word size	0x0010	16 words
0x0060	Physical Start Address	0x2000	
0x0061	Length	0x0114	
0x0062:LSB	Control Register	0x26	MBoxOut
0x0062:MSB	Status Register	0x00	Don't care
0x0063:LSB	Activate	0x01	Enable SyncM
0x0063:MSB	SM type	0x01	
0x0064	Physical Start Address	0x2800	
0x0065	Length	0x0114	
0x0066:LSB	Control Register	0x22	MBoxIn
0x0066:MSB	Status Register	0x00	Don't care
0x0067:LSB	Activate	0x01	Enable SyncM
0x0067:MSB	SM type	0x02	
0x0068	Physical Start Address	0x1000	
0x0069	Length	0xYYYY	YYYY == Output I/O data size
0x006A:LSB	Control Register	0x64	

Address	Description	Data (LE)	Comment
0x006A:MSB	Status Register	0x00	Don't care
0x006B:LSB	Activate	0x01	Enable SyncM
0x006B:MSB	SM type	0x03	
0x006C	Physical Start Address	0x1800	
0x006D	Length	0xZZZZ	ZZZZ == Input I/O data size
0x006E:LSB	Control Register	0x20	
0x006E:MSB	Status Register	0x00	Don't care
0x006F:LSB	Activate	0x01	Enable SyncM
0x006F:MSB	SM type	0x04	

5.2 Node addressing

There are a number of different addressing modes which can be applied when communicating with EtherCAT slaves. As a full EtherCAT slave device, the option board supports position addressing, node addressing and logical addressing.

5.2.1 Position addressing

Position addressing is used together with the Auto increment commands. See section 5.1.2 for more information about the supported commands. In this case the slave checks the address in the command and if the address equals 0 the slave will handle the command.

The slave always increments the address by 1 regardless if it is addressed or not. This means that in order to send a command to the second slave on the network the address 0xFFFF is added to the command. The first slave will increment this value by 1, causing the value (16-bit value) to overflow and turn 0. The second slave will receive the command with address 0 and therefore handle it.

5.2.2 Node addressing

During configuration the master can set the address of a slave by writing the address to ESC registers 0x0010 to 0x0011, "Configured station address". This address is then used together with the Configured address commands. See section 5.1.2 for more information about the supported commands.

5.2.3 Logical addressing

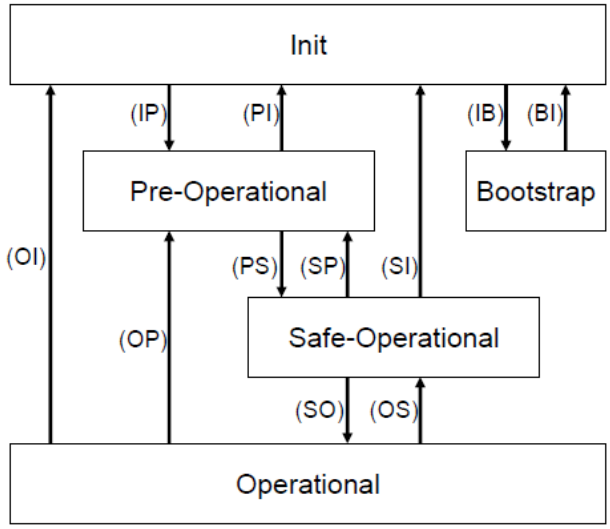
By using the fieldbus memory management units (FMMUs) the master can map a memory area of the slave ESC memory to a network wide memory area. See section 5.5 for more information about FMMUs.

The network wide memory address can then be used together with the Logical commands to read data of the slave. See section 5.1.2 for more information about the supported commands.

5.3 EtherCAT state machine

The EtherCAT master can control the state of the option board via the EtherCAT state machine.

The EtherCAT state machine is described in the following figure.



Depending on the state different functionality is enabled in the option board.

State	Description
Init	This is the state that is entered after power on. In this state the EtherCAT master has access to the registers in the EtherCAT slave controller and the EEPROM content.
Bootstrap	This state is not entered during normal boot up of the slave, it is a state that is used for firmware upgrade of an EtherCAT slave.
Pre-Operational	In this state the read and write mailboxes have been configured, meaning that mailbox communication such as CoE SDO functionality is enabled.
Safe-Operational	During transition to Safe-Operational the sync managers used for process data communication are configured, so in this state process data communication is started. The output process data is NOT VALID in this state.
Operational	Once this state is reached the output process data from the master is valid. Note that it is not possible to reach Operational without the master sending output process data during the Safe-Operational state.

5.4 Sync managers

Sync managers are used for reading and writing the user RAM of the ESC. A sync manager is mapped to a certain memory area of the user RAM and configured according to how that memory is going to be used.

Sync managers can be of two types, mailbox sync managers which are used for mailbox communication (e.g. SDO upload and download) and 3-buffered sync managers that are used for process data.

The option board features four Sync Managers:

- Sync Manager 0 Used for mailbox write transfers (Master to Slave). The option board has a configurable write mailbox size with default size of 276 bytes.
- Sync Manager 1 Used for mailbox read transfers (Slave to Master). The option has a configurable read mailbox size with default size of 276 bytes.
- Sync Manager 2 holds the RxPDO data. The size of sync manager 2 is the same as the size of the RxPDO data, so it changes depending on RxPDO configuration.
- Sync Manager 3 holds the TxPDO data. The size of sync manager 3 is the same as the size of the TxPDO data, so it changes depending on TxPDO configuration.

5.5 Fieldbus memory management units (FMMU)

FMMUs are used by the master to map any memory area within the ESC to a network wide memory area. One typical master usage of FMMUs is to map the memory of sync manager 2 (used for RxPDO data) to a network memory address (e.g. 0x10000000) and configure that FMMU to a write FMMU. Doing this will cause all data written to network memory address 0x100000000 to end up in sync manager 2.

The master will also map the memory of sync manager 3 into another FMMU with the same network memory address (logical address) 0x10000000 but configure that FMMU as a read FMMU. That means that when memory address 0x10000000 is read, the content of sync manager 3 is read.

This allows the master to use a single Logical Read Write (LRW) command for both writing the RxPDO data and reading the TxPDO data.

There are four FMMUs supported by the option board. The EtherCAT master can use the FMMUs freely for any purpose.

The default FMMU configuration looks like this:

- FMMU 0 is used for output process data.
- FMMU 1 is used for input process data.
- FMMU 2 is used for the read mailbox.

5.6 Watchdogs

5.6.1 ESC PDI watchdog

The PDI watchdog functionality is not supported by the HMS ESC.

5.6.2 Process data watchdog

If enabled, this watchdog monitors the RxPDO communication towards the option board. If the master doesn't update the RxPDO data within the specified time period, this will trigger a timeout condition in the module, causing it to shift from OPERATIONAL to SAFE-OPERATIONAL.

The sync manager watchdog is enabled by default in the ESI file, with a default time period of 100 ms. The sync manager watchdog can be disabled/enabled manually in the EtherCAT master configuration tool.

The process data watchdog timeout time is based on the content in registers 0x400 (watchdog divider) and 0x420 (watchdog time sync manager).

5.7 Device identification

The option board supports two methods for device identification:

- Configured station alias
- Requesting ID

Device identification is used primarily for two different use cases; prevention of cable swapping and hot connect applications.

5.7.1 Configured station alias

The configured station alias is changed from the EtherCAT master configuration tool by writing address 0x0004 in the ESC EEPROM. Changing the configured station alias also requires that the checksum at address 0x0007 is updated. Note that the configured station alias cannot be changed locally in the option board, it is only changed by the configuration tool.

The new value is not used until after power cycle or after a “Reload EEPROM” command has been issued towards the option board.

The valid range of the configured station alias is 1-65535. Value 0 means that the configured station alias is not configured. 0 is the default value.

The configured station alias can also be used as the device address used for communication with FPxx commands if bit 24 in the DL control ESC register is set to 1.

5.7.2 Requesting ID

The Device ID used for the requesting ID mechanism can be changed by writing option board parameter Pr.1305. See section 4.2 for more information on changing this parameter.

The valid range of the Device ID is 1-65535. Value 0 means that the Device ID is unconfigured.

The device ID is not an address used for communication, it is only a value used for identifying a slave uniquely on the network.

5.8 CANopen over EtherCAT

The option board uses CANopen over EtherCAT (CoE) as the primary application layer.

5.8.1 General information

The following CoE functionality is supported by the option board:

- SDO information
  - **Get OD List**
    Reads different types of object lists from the option board. The following list types are supported:
    - Number of objects in the different lists
    - All objects
    - RxPDO mappable objects
    - TxPDO mappable objects
  - **Get object description**
    Reads the details of a CoE object.
  - **Get entry description**
    Reads the details of a sub-index of a CoE object.
- SDO Download Expedited
  - Writes up to four octets to the slave.
- SDO Download Normal
  - Writes up to a negotiated number of octets to the slave.
- Download SDO segment
  - Writes additional data if the object size is greater than the negotiated number of octets.
- SDO Upload Expedited
  - Reads up to four octets from the slave.
- SDO Upload Normal
  - Reads up to a negotiated number of octets from the slave.
- Upload SDO segment
  - Reads additional data if the object size is greater than the negotiated number of octets.
- Abort SDO Transfer
  - Server abort of service in case of an erroneous condition.
- Complete access is supported for both SDO Download and SDO upload.

5.8.2 CoE object dictionary

Data in CANopen over EtherCAT is accessed through object entries where an object entry is a combination of an object index and a sub-index.

Each object in the object dictionary has one or more sub-indices that can be accessed with SDO download or SDO upload requests from the EtherCAT master.

The following objects are present in the option board:

Object index	Name	Comment
0x1000	Device Type	Described in 5.8.2.1.
0x1001	Error Register	Described in 5.8.2.2.
0x1003	Pre-defined error field	Described in 5.8.2.3.
0x1008	Manufacturer Device Name	Described in 5.8.2.4.
0x1009	Manufacturer Hardware version	Described in 5.8.2.5.
0x100A	Manufacturer Software version	Described in 5.8.2.6.
0x1011	Restore default parameters	Described in 5.8.2.7.
0x1018	Identity Object	Described in 5.8.2.8.
0x1600	Receive PDO Mapping	Described in 5.8.2.9.
0x1604	Receive PDO Mapping	Described in 5.8.2.10.
0x1605	Receive PDO Mapping	Described in 5.8.2.11.
0x1610	Receive PDO Mapping	Described in 5.8.2.12.
0x1611	Receive PDO Mapping	Described in 5.8.2.13. Only present when user specific process data mapping is enabled. See section 4.2.2 for more information
0x1A00	Transmit PDO Mapping	Described in 5.8.2.14.
0x1A04	Transmit PDO Mapping	Described in 5.8.2.15.
0x1A05	Transmit PDO Mapping	Described in 5.8.2.16.
0x1A10	Transmit PDO Mapping	Described in 5.8.2.17.
0x1A11	Transmit PDO Mapping	Described in 5.8.2.18. Only present when user specific process data mapping is enabled. See section 4.2.2 for more information
0x1C00	Sync Manager Communication Type	Described in 5.8.2.19.
0x1C12	Sync Manager Rx PDO assign	Described in 5.8.2.20.
0x1C13	Sync Manager Tx PDO assign	Described in 5.8.2.21.
0x1C32	Output Sync Manager Parameter	Described in 5.8.2.22.
0x1C33	Input Sync Manager Parameter	Described in 5.8.2.23.
0x3000-0x3FFF	Inverter parameters	Described in 5.8.2.24. Not all objects in this range are present.
0x4000-0x4FFF	Monitor items (process data parameters)	Described in 5.8.2.25. Not all objects in this range are present.
0x6040	Controlword	Described in 5.8.2.26.2.
0x6041	Statusword	Described in 5.8.2.26.3.
0x6042	vl target velocity	Described in 5.8.2.26.4.
0x6043	vl velocity demand	Described in 5.8.2.26.5.
0x6044	vl velocity actual value	Described in 5.8.2.26.6.
0x6046	vl velocity min max amount	Described in 5.8.2.26.7.
0x6048	vl velocity acceleration	Described in 5.8.2.26.8.

Object index	Name	Comment
0x6049	vl velocity deceleration	Described in 5.8.2.26.9.
0x604A	vl velocity quick stop	Described in 5.8.2.26.10.
0x604C	vl dimension factor	Described in 5.8.2.26.11.
0x605B	Shutdown option code	Described in 5.8.2.26.12.
0x605C	Disable operation option code	Described in 5.8.2.26.13.
0x6060	Modes of operation	Described in 5.8.2.26.14.
0x6061	Modes of operation display	Described in 5.8.2.26.15.
0x6071	Target torque	Described in 5.8.2.26.16.
0x6075	Motor rated current	Described in 5.8.2.26.18.
0x6076	Motor rated torque	Described in 5.8.2.26.18.
0x6077	Torque actual value	Described in 5.8.2.26.19.
0x6087	Torque slope	Described in 5.8.2.26.20.
0x6088	Torque profile type	Described in 5.8.2.26.21.
0x6402	Motor type	Described in 5.8.2.26.22.
0x6502	Supported drive modes	Described in 5.8.2.26.23.

5.8.2.1 Object 0x1000, Device Type

Object description	
Index	0x1000
Name	Device Type
Object code	VAR
Data type	UNSIGNED32

Entry description		
Sub-index 0	Name	Device Type
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x00010192 Bit 0-15: 0x0192 DS402 drive profile Bit 16-31: 0x0001 Frequency inverter

5.8.2.2 Object 0x1001, Error Register

Object description	
Index	0x1001
Name	Error Register
Object code	VAR
Data type	UNSIGNED8

Entry description		
Sub-index 0	Name	Error Register
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	0x00
	Details: Bit 0 set: Generic error Bit 1 set: Current error Bit 2 set: Voltage error Bit 3 set: Temperature error Bit 4 set: Communication error Bit 5 set: Device profile specific error (always 0) Bit 6 set: Reserved Bit 7 set: Manufacturer specific error See section 5.11 for more information on when the different bits are set.	

5.8.2.3 Object 0x1003, Pre-defined error field

This object logs fault reported by the inverter. The newest fault is always added to sub-index 0 and the older faults are shifted to one sub-index higher. If more than five faults occur the oldest fault is deleted.

Object description	
Index	0x1003
Name	Pre-defined error field
Object code	ARRAY
Data type	UNSIGNED32

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read and write This entry only accepts value 0x00 when written. Writing 0x00 to this entry clears the error history.
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	0x05
Sub-index 1	Name	Standard error field
	Access	Read only

Entry description		
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x00000000
Sub-index 2	Name	Standard error field
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x00000000
Sub-index 3	Name	Standard error field
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x00000000
Sub-index 4	Name	Standard error field
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x00000000
Sub-index 5	Name	Standard error field
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x00000000
Details for sub-index 1-5: When the inverter reports a fault the value 0x0000ee00 is set where ee is the EtherCAT event code. See section 5.11 for more information on how inverter faults are translated to EtherCAT event codes.		

5.8.2.4 Object 0x1008, Manufacturer Device Name

Object description	
Index	0x1008
Name	Manufacturer Device Name
Object code	VAR
Data type	VISIBLE_STRING

Entry description		
Sub-index 0	Name	Manufacturer Device Name
	Access	Read only
	PDO mapping	No
	Data type	VISIBLE_STRING
	Default value	A8NECT_2P

5.8.2.5 Object 0x1009, Manufacturer Hardware Version

Object description	
Index	0x1009
Name	Manufacturer Hardware Version
Object code	VAR
Data type	VISIBLE_STRING

Entry description		
Sub-index 0	Name	Manufacturer Hardware Version
	Access	Read only
	PDO mapping	No
	Data type	VISIBLE_STRING
	Default value	The value of the hardware version assigned during production represented as a string.

5.8.2.6 Object 0x100A, Manufacturer Software Version

Object description	
Index	0x100A
Name	Manufacturer Software Version
Object code	VAR
Data type	VISIBLE_STRING

Entry description		
Sub-index 0	Name	Manufacturer Software Version
	Access	Read only
	PDO mapping	No
	Data type	VISIBLE_STRING
	Default value	String in the format X.YY.ZZ where X is the option board major version, YY is the option board minor version and ZZ is the option board build.

5.8.2.7 Object 0x1011, Restore default parameters

Object description	
Index	0x1011
Name	Restore default parameters
Object code	ARRAY
Data type	UNSIGNED32

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	0x01
Sub-index 1	Name	Restore all default parameters
	Access	Read and write
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x00000001 Details: The master can write the value “load” (0x64616F6C) to this sub-index to initiate a reset to factory default. No other written value will be accepted. If the inverter doesn’t accept the factory default reset request the abort code 0x08000022 (invalid state) is sent as response. Factory default reset can only be performed in the pre-operational state. If successful, all non-volatile data in both the inverter and option board will be set to the default values after power cycle.

5.8.2.8 Object 0x1018, Identity Object

Object description	
Index	0x1018
Name	Identity Object
Object code	RECORD
Data type	IDENTITY

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	0x04
Sub-index 1	Name	Vendor ID
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x00000A1E
Sub-index 2	Name	Product Code
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x02000101 in case of FR-A800 inverter 0x02000201 in case of FR-F800 inverter
Sub-index 3	Name	Revision Number
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0xXXXXYYYY where XXXX is the major revision of the option board firmware and YYYY is minor revision of the option board firmware.
Sub-index 4	Name	Serial Number
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	Serial number of the option board assigned during production.

5.8.2.9 Object 0x1600, Receive PDO Mapping

Object description	
Index	0x1600
Name	Receive PDO Mapping
Object code	RECORD
Data type	PDO_MAPPING

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	0x01
Sub-index 1	Name	Mapped Object 001
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x60400010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x6040 Object index (Controlword)

5.8.2.10 Object 0x1604, Receive PDO Mapping

Object description	
Index	0x1604
Name	Receive PDO Mapping
Object code	RECORD
Data type	PDO_MAPPING
Note: This object is not supported on the F800 drive series.	

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	0x02
Sub-index 1	Name	Mapped Object 001
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x60400010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x6040 Object index (Controlword)
Sub-index 2	Name	Mapped Object 002
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x60710010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x6071 Object index (Target torque)

5.8.2.11 Object 0x1605, Receive PDO Mapping

Object description	
Index	0x1605
Name	Receive PDO Mapping
Object code	RECORD
Data type	PDO_MAPPING

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	0x02
Sub-index 1	Name	Mapped Object 001
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x60400010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x6040 Object index (Controlword)
Sub-index 2	Name	Mapped Object 002
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x60420010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x6042 Object index (vl target velocity)

5.8.2.12 Object 0x1610, Receive PDO Mapping

Object description	
Index	0x1610
Name	Receive PDO Mapping
Object code	RECORD
Data type	PDO_MAPPING

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	0x06
Sub-index 1	Name	Mapped Object 001
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x60400010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x6040 Object index (Controlword)
Sub-index 2	Name	Mapped Object 002
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x60420010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x6042 Object index (vl target velocity)
Sub-index 3	Name	Mapped Object 003
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x40F90010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x40F9 Object index (Input terminal)
Sub-index 4	Name	Mapped Object 004
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32

Entry description		
	Default value	0x40130010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x4013 Object index (Position pulse)
Sub-index 5	Name	Mapped Object 005
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x60480210 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x02 Sub-index Bits 16-31: 0x6048 Object index (vl velocity acceleration)
Sub-index 6	Name	Mapped Object 006
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x60490210 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x02 Sub-index Bits 16-31: 0x6049 Object index (vl velocity deceleration)

5.8.2.13 Object 0x1611, Receive PDO Mapping

This object is only present if user specific PDO mapping is enabled, see section 4.2.2 for more information.

Object description	
Index	0x1611
Name	Receive PDO Mapping
Object code	RECORD
Data type	PDO_MAPPING

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	Depends on the number of parameters configured in option board parameters Pr.1306-1315.
Sub-index 1-10	Name	Mapped Object 001-010
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	Depends on what parameters are configured in option board parameters Pr.1306-1315.

5.8.2.14 Object 0x1A00, Transmit PDO Mapping

Object description	
Index	0x1A00
Name	Transmit PDO Mapping
Object code	RECORD
Data type	PDO_MAPPING

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	0x01
Sub-index 1	Name	Mapped Object 001
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x60410010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x6041 Object index (Statusword)

5.8.2.15 Object 0x1A04, Transmit PDO Mapping

Object description	
Index	0x1A04
Name	Transmit PDO Mapping
Object code	RECORD
Data type	PDO_MAPPING
Note: This object is not supported on the F800 drive series.	

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	0x02
Sub-index 1	Name	Mapped Object 001
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x60410010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x6041 Object index (Statusword)
Sub-index 2	Name	Mapped Object 002
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x60770010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x6077 Object index (Torque actual value)

5.8.2.16 Object 0x1A05, Transmit PDO Mapping

Object description	
Index	0x1A05
Name	Transmit PDO Mapping
Object code	RECORD
Data type	PDO_MAPPING

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	0x02
Sub-index 1	Name	Mapped Object 001
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x60410010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x6041 Object index (Statusword)
Sub-index 2	Name	Mapped Object 002
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x60440010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x6044 Object index (vl velocity actual value)

5.8.2.17 Object 0x1A10, Transmit PDO Mapping

Object description	
Index	0x1A10
Name	Transmit PDO Mapping
Object code	RECORD
Data type	PDO_MAPPING

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	A800: 0x06 F800: 0x05
Sub-index 1	Name	Mapped Object 001
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x60410010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x6041 Object index (Statusword)
Sub-index 2	Name	Mapped Object 002
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x60440010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x6044 Object index (vl velocity actual value)
Sub-index 3	Name	Mapped Object 003
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x40100010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x4010 Object index (Output terminal)
Sub-index 4	Name	Mapped Object 004
	Access	Read only
	PDO mapping	No

Entry description		
	Data type	UNSIGNED32
	Default value	0x40130010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x4013 Object index (Position pulse)
Sub-index 5	Name	Mapped Object 005
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	A800: 0x60770010 Bits 0-7: 0x10 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x6077 Object index (Torque actual value) F800: 0x420F0020 Bits 0-7: 0x20 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x420F Object index (Position error)
Sub-index 6	Name	Mapped Object 006
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	A800: 0x420F0020 Bits 0-7: 0x20 Bit length of entry Bits 8-15: 0x00 Sub-index Bits 16-31: 0x420F Object index (Position error) F800: Sub-index does not exist.

5.8.2.18 Object 0x1A11, Transmit PDO Mapping

This object is only present if user specific PDO mapping is enabled, see section 4.2.2 for more information.

Object description	
Index	0x1A11
Name	Transmit PDO Mapping
Object code	RECORD
Data type	PDO_MAPPING

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	Depends on the number of parameters configured in option board parameters Pr.1316-1329.
Sub-index 1-14	Name	Mapped Object 001-014
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	Depends on what parameters are configured in option board parameters Pr.1316-1329.

5.8.2.19 Object 0x1C00, Sync Manager Communication Type

Object description	
Index	0x1C00
Name	Sync Manager Communication Type
Object code	ARRAY
Data type	UNSIGNED8

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	4
Sub-index 1	Name	Sync Manager 0
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	0x01 Value 0x01 means sync manager 0 is used for Mailbox write.
Sub-index 2	Name	Sync Manager 1
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	0x02 Value 0x02 means sync manager 1 is used for Mailbox read.
Sub-index 3	Name	Sync Manager 2
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	0x03 Value 0x03 means sync manager 2 is used for Process data out.
Sub-index 4	Name	Sync Manager 3
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	0x04 Value 0x04 means sync manager 3 is used for Process data in.

5.8.2.20 Object 0x1C12, Sync Manager Rx PDO assign

Object description	
Index	0x1C12
Name	Sync Manager Rx PDO assign
Object code	ARRAY
Data type	UNSIGNED16

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read in all states Write in state pre-operational
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	1 This value indicates how many PDO configuration objects that have been assigned to sync manager 2.
Sub-index 1	Name	Assigned PDO 001
	Access	Read in all states Write in state pre-operational Note: This sub-index can only be written when sub-index 0 equals 0.
	PDO mapping	No
	Data type	UNSIGNED16
	Default value	0x1600
Sub-index 2	Name	Assigned PDO 002
	Access	Read in all states Write in state pre-operational Note: This sub-index can only be written when sub-index 0 equals 0.
	PDO mapping	No
	Data type	UNSIGNED16
	Default value	0x0000
Sub-index 3	Name	Assigned PDO 003
	Access	Read in all states Write in state pre-operational Note: This sub-index can only be written when sub-index 0 equals 0.
	PDO mapping	No
	Data type	UNSIGNED16
	Default value	0x0000
Sub-index 4	Name	Assigned PDO 004
	Access	Read in all states Write in state pre-operational Note: This sub-index can only be written when sub-index 0 equals 0.

Entry description		
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	0x0000
Sub-index 5	Name	Assigned PDO 005
	Access	Read in all states Write in state pre-operational Note: This sub-index can only be written when sub-index 0 equals 0.
	PDO mapping	No
	Data type	UNSIGNED16
	Default value	0x0000
	This sub-index is only present when user specific process data mapping is enabled. See section 4.2.2 for more information.	

5.8.2.21 Object 0x1C13, Sync Manager Tx PDO assign

Object description	
Index	0x1C13
Name	Sync Manager Tx PDO assign
Object code	ARRAY
Data type	UNSIGNED16

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read in all states Write in state pre-operational
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	1 This value indicates how many PDO configuration objects that have been assigned to sync manager 3.
Sub-index 1	Name	Assigned PDO 001
	Access	Read in all states Write in state pre-operational Note: This sub-index can only be written when sub-index 0 equals 0.
	PDO mapping	No
	Data type	UNSIGNED16
	Default value	0x1A00
Sub-index 2	Name	Assigned PDO 002

Entry description		
	Access	Read in all states Write in state pre-operational Note: This sub-index can only be written when sub-index 0 equals 0.
	PDO mapping	No
	Data type	UNSIGNED16
	Default value	0x0000
Sub-index 3	Name	Assigned PDO 003
	Access	Read in all states Write in state pre-operational Note: This sub-index can only be written when sub-index 0 equals 0.
	PDO mapping	No
	Data type	UNSIGNED16
	Default value	0x0000
Sub-index 4	Name	Assigned PDO 004
	Access	Read in all states Write in state pre-operational Note: This sub-index can only be written when sub-index 0 equals 0.
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	0x0000
Sub-index 5	Name	Assigned PDO 005
	Access	Read in all states Write in state pre-operational Note: This sub-index can only be written when sub-index 0 equals 0.
	PDO mapping	No
	Data type	UNSIGNED16
	Default value	0x0000
	This sub-index is only present when user specific process data mapping is enabled. See section 4.2.2 for more information.	

5.8.2.22 Object 0x1C32, Output Sync Manager Parameter

Object description	
Index	0x1C32
Name	Output Sync Manager Parameter
Object code	RECORD
Data type	SYNC_PAR

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	12
Sub-index 1	Name	Synchronization Type
	Access	Read and write
	PDO mapping	No
	Data type	UNSIGNED16
	Default value	0x0000 Value 0x0000 means “Free run”.
Sub-index 2	Name	Cycle time
	Access	Read and write
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x001E8480 Cycle time in nanoseconds. The option board does not use this value.
Sub-index 3	Name	Shift Time
	Access	Read and write
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x00000000 Shift time in nanoseconds. The option board does not use this value.
Sub-index 4	Name	Synchronization Types supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED16
	Default value	0x0001 Only Free Run supported.
Sub-index 5	Name	Minimum Cycle Time
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32

Entry description		
	Default value	0x000186A0 Minimum cycle time in nanoseconds. The option board does not use this value.
Sub-index 6	Name	Calc and Copy Time
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x000001F4 Calc and copy time in nanoseconds. The option board does not use this value.
Sub-index 9	Name	Delay Time
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x00000000 Delay time in nanoseconds. The option board does not use this value.
Sub-index 12	Name	Cycle Time Too Small
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED16
	Default value	0x0000 Cycle time too small counter. The option board does not use this value.

5.8.2.23 Object 0x1C33, Input Sync Manager Parameter

Object description	
Index	0x1C33
Name	Input Sync Manager Parameter
Object code	RECORD
Data type	SYNC_PAR

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	12
Sub-index 1	Name	Synchronization Type
	Access	Read and write
	PDO mapping	No
	Data type	UNSIGNED16
	Default value	0x0000 Value 0x0000 means “Free run”.
Sub-index 2	Name	Cycle time
	Access	Read and write
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x001E8480 Cycle time in nanoseconds. The option board does not use this value.
Sub-index 3	Name	Shift Time
	Access	Read and write
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x00000000 Shift time in nanoseconds. The option board does not use this value.
Sub-index 4	Name	Synchronization Types supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED16
	Default value	0x0001 Only Free Run supported.
Sub-index 5	Name	Minimum Cycle Time
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32

Entry description		
	Default value	0x000186A0 Minimum cycle time in nanoseconds. The option board does not use this value.
Sub-index 6	Name	Calc and Copy Time
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x000001F4 Calc and copy time in nanoseconds. The option board does not use this value.
Sub-index 12	Name	Cycle Time Too Small
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED16
	Default value	0x0000 Cycle time too small counter. The option board does not use this value.

5.8.2.24 Object range 0x3000-0x3FFF, Inverter parameters

In this object range all network accessible inverter parameters are represented.

The inverter parameter number is translated into a CoE object by adding the offset 0x3000, meaning inverter parameter 1 can be accessed through CoE object index 0x3001.

Object description	
Index	Inverter parameter number + 0x3000
Name	“Parameter# nnnnH” where nnnn is the inverter parameter hexadecimal number
Object code	VAR
Data type	SIGNED16, UNSIGNED16, SIGNED32 or UNSIGNED32 depending on inverter parameter data type

Entry description		
Sub-index 0	Name	“Parameter# nnnnH” where nnnn is the inverter parameter hexadecimal number
	Access	Read only or Read and write depending on inverter parameter access rights
	PDO mapping	No
	Data type	SIGNED16, UNSIGNED16, SIGNED32 or UNSIGNED32 depending on inverter parameter data type
	Default value	Inverter parameter default value

5.8.2.25 Object range 0x4000-0x4FFF, Monitor items (process data parameters)

In this object range all monitor items of the inverter can be accessed from the EtherCAT master.

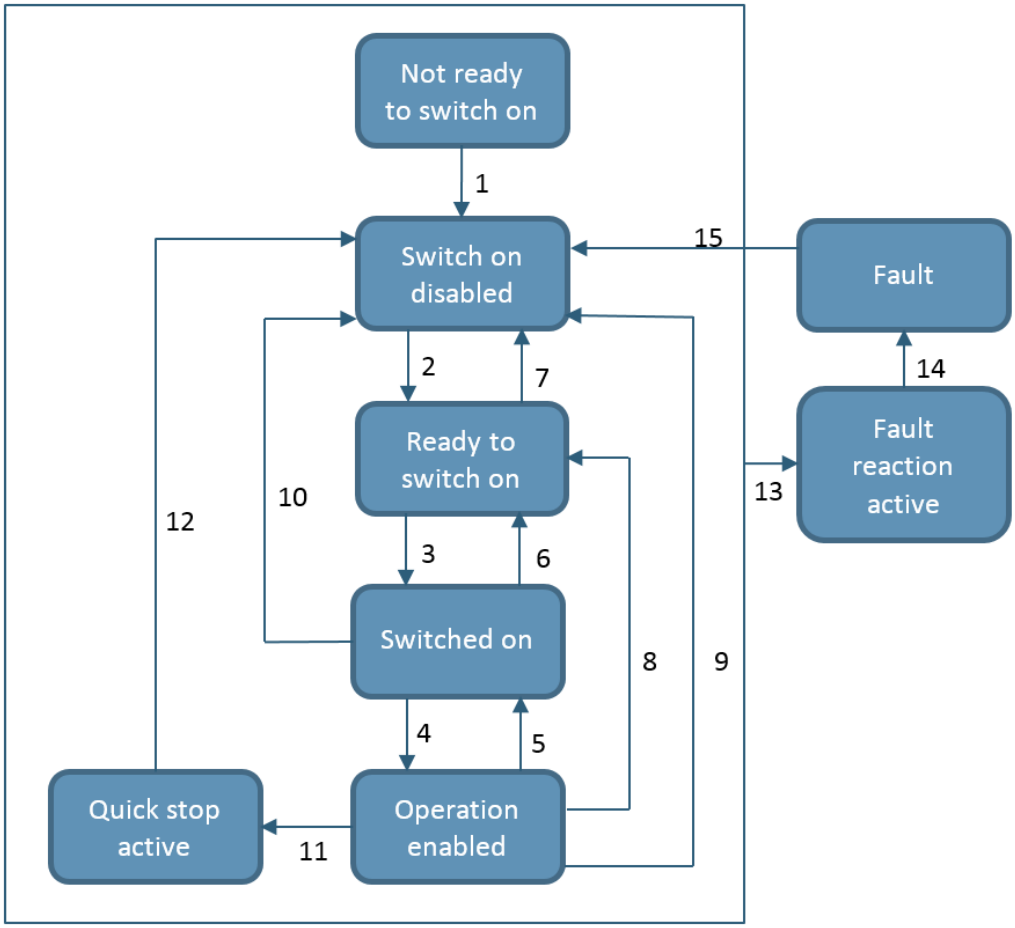
The monitor item number is translated into a CoE object by adding the offset 0x4000, meaning monitor item 1 can be accessed through CoE object index 0x4001.

Object description	
Index	Monitor item number + 0x4000
Name	“Process data# nnnnH” where nnnn is the monitor item hexadecimal number
Object code	VAR
Data type	SIGNED16, UNSIGNED16, SIGNED32 or UNSIGNED32 depending on monitor item data type

Entry description		
Sub-index 0	Name	“Process data# nnnnH” where nnnn is the monitor item hexadecimal number
	Access	Read only or Read and write depending on monitor item access rights
	PDO mapping	Read only monitor items are TxPDO mappable. Monitor items with read and write access can be mapped as both RxPDO and TxPDO data.
	Data type	SIGNED16, UNSIGNED16, SIGNED32 or UNSIGNED32 depending on monitor item data type

5.8.2.26 DS402 drive profile parameters (0x6000 to 0x6FFF)

5.8.2.26.1 DS402 state machine



Below is a table describing when the different state transitions takes place.

Transition	Event
1	Automatic transition
2	Shutdown command received
3	Switch on command received
4	Enable operation command received
5	Disable operation command received
6	Shutdown command received
7	Quick stop or Disable voltage command received
8	Shutdown command received
9	Disable voltage command received
10	Disable voltage or Quick stop command received
11	Quick stop command received
12	Automatic transition when the quick stop function is completed or Disable voltage command is received
13	Fault signal
14	Automatic transition
15	Fault reset command received

5.8.2.26.2 Object 0x6040, Controlword

Object description	
Index	0x6040
Name	Controlword
Object code	VAR
Data type	UNSIGNED16

Entry description		
Sub-index 0	Name	Controlword
	Access	Read and write
	PDO mapping	RxPDO
	Data type	UNSIGNED16
	Default value	0x0000

The different bits in the controlword are described below.

Bit	Name	Comments
0	Switch on (so)	
1	Enable voltage (ev)	
2	Quick stop (qs)	
3	Enable operation (eo)	
4	Enable ramp	Only used in velocity mode. Reserved in torque mode.
5	Unlock ramp	Only used in velocity mode. Reserved in torque mode.
6	Reference ramp	Only used in velocity mode. Reserved in torque mode.
7	Fault reset (fr)	
8-15	Reserved	Written value is ignored.

Command coding for the controlword is described below. The X below means the bit is ignored. The transitions refer to the state transitions in section 5.8.2.26.1.

Command	Bits of the controlword					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	X	1	1	0	2, 6, 8
Switch on	0	0	1	1	1	3
Switch on + Enable operation	0	1	1	1	1	3 + 4
Disable voltage	0	X	X	0	X	7, 9, 10, 12
Quick stop	0	X	0	1	X	7, 10, 11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4
Fault reset	0 -> 1	X	X	X	X	15

5.8.2.26.3 Object 0x6041, Statusword

Object description	
Index	0x6041
Name	Statusword
Object code	VAR
Data type	UNSIGNED16

Entry description		
Sub-index 0	Name	Statusword
	Access	Read only
	PDO mapping	TxPDO
	Data type	UNSIGNED16
	Default value	0x0250

The different bits in the statusword are described below.

Bit	Name	Comments
0	Ready to switch on (rtso)	
1	Switched on (so)	
2	Operation enabled (oe)	
3	Fault (f)	
4	Voltage enabled (ve)	
5	Quick stop (qs)	
6	Switch on disabled (sod)	
7	Warning (w)	Not used, always 0.
8	Manufacturer specific	Not used, always 0.
9	Remote (rm)	
10	Target reached	
11	Internal limit active (ila)	
12-13	Operation mode specific	Not used, always 0.
14-15	Manufacturer specific	Not used, always 0.

State coding for the statusword is described below. The X below means the bit is ignored. Bits not mentioned below does not affect the state. The states in the table below refer to the states in section 5.8.2.26.1.

Bits of the statusword						State
Bit 6	Bit 5	Bit 3	Bit 2	Bit 1	Bit 0	
0	X	0	0	0	0	Not ready to switch on
1	X	0	0	0	0	Switch on disabled
0	1	0	0	0	1	Ready to switch on
0	1	0	0	1	1	Switched on
0	1	0	1	1	1	Operation enabled
0	0	0	1	1	1	Quick stop active
0	X	1	1	1	1	Fault reaction active
0	X	1	0	0	0	Fault

5.8.2.26.4 Object 0x6042, *vl* target velocity

Object description	
Index	0x6042
Name	*vl* target velocity
Object code	VAR
Data type	SIGNED16

Entry description		
Sub-index 0	Name	*vl* target velocity
	Access	Read and write
	PDO mapping	RxPDO
	Data type	SIGNED16
	Default value	0
	Unit	rpm

5.8.2.26.5 Object 0x6043, *vl* velocity demand

Object description	
Index	0x6043
Name	*vl* velocity demand
Object code	VAR
Data type	SIGNED16

Entry description		
Sub-index 0	Name	*vl* velocity demand
	Access	Read only
	PDO mapping	TxPDO
	Data type	SIGNED16
	Default value	0
	Unit	rpm

5.8.2.26.6 Object 0x6044, *vl* velocity actual value

Object description	
Index	0x6044
Name	*vl* velocity actual value
Object code	VAR
Data type	SIGNED16

Entry description		
Sub-index 0	Name	*vl* velocity actual value
	Access	Read only
	PDO mapping	TxPDO
	Data type	SIGNED16
	Default value	0
	Unit	rpm

5.8.2.26.7 Object 0x6046, *vl* velocity min max amount

Object description	
Index	0x6046
Name	*vl* velocity min max amount
Object code	ARRAY
Data type	UNSIGNED32

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	2
Sub-index 1	Name	*vl* velocity min max amount.SubIndex 001
	Access	Read and write
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0
	Unit	rpm
	Note: This value specifies the minimum velocity.	
Sub-index 2	Name	*vl* velocity min max amount.SubIndex 002
	Access	Read and write
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	3600
	Unit	rpm
	Note: This value specifies the maximum velocity.	

5.8.2.26.8 Object 0x6048, *v/* velocity acceleration

Object description	
Index	0x6048
Name	*v/* velocity acceleration
Object code	RECORD
Data type	UNSIGNED32

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	2
Sub-index 1	Name	*v/* velocity acceleration.SubIndex 001
	Access	Read and write
	PDO mapping	RxPDO
	Data type	UNSIGNED32
	Default value	1800
	Unit	rpm
	Note: This value specifies the acceleration delta velocity in rpm.	
Sub-index 2	Name	*v/* velocity acceleration.SubIndex 002
	Access	Read and write
	PDO mapping	RxPDO
	Data type	UNSIGNED16
	Default value	5
	Unit	s
	Note: This value specifies the acceleration delta time in seconds.	

5.8.2.26.9 Object 0x6049, *v*/ velocity deceleration

Object description	
Index	0x6049
Name	*v*/ velocity deceleration
Object code	RECORD
Data type	UNSIGNED32

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	2
Sub-index 1	Name	*v*/ velocity deceleration.SubIndex 001
	Access	Read and write
	PDO mapping	RxPDO
	Data type	UNSIGNED32
	Default value	1800
	Unit	rpm
	Note: This value specifies the deceleration delta velocity in rpm.	
Sub-index 2	Name	*v*/ velocity deceleration.SubIndex 002
	Access	Read and write
	PDO mapping	RxPDO
	Data type	UNSIGNED16
	Default value	5
	Unit	s
	Note: This value specifies the deceleration delta time in seconds.	

5.8.2.26.10 Object 0x604A, *vl* velocity quick stop

Object description	
Index	0x604A
Name	*vl* velocity quick stop
Object code	RECORD
Data type	UNSIGNED32
Note: This object is not supported on the F800 drive series.	

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	2
Sub-index 1	Name	*vl* velocity quick stop.SubIndex 001
	Access	Read and write
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	1800
	Unit	Rpm
	Note: This value specifies the quick stop deceleration delta velocity in rpm.	
Sub-index 2	Name	*vl* velocity quick stop.SubIndex 002
	Access	Read and write
	PDO mapping	No
	Data type	UNSIGNED16
	Default value	5
	Unit	s
	Note: This value specifies the quick stop deceleration delta time in seconds.	

5.8.2.26.11 Object 0x604C, *v/* dimension factor

Object description	
Index	0x604C
Name	*v/* dimension factor
Object code	ARRAY
Data type	SIGNED32

Entry description		
Sub-index 0	Name	Highest sub-index supported
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED8
	Default value	2
Sub-index 1	Name	*v/* dimension factor.SubIndex 001
	Access	Read and write
	PDO mapping	No
	Data type	SIGNED32
	Default value	1
Sub-index 2	Name	*v/* dimension factor.SubIndex 002
	Access	Read and write
	PDO mapping	No
	Data type	SIGNED32
	Default value	1

5.8.2.26.12 Object 0x605B, Shutdown option code

Object description	
Index	0x605B
Name	Shutdown option code
Object code	VAR
Data type	SIGNED16

Entry description		
Sub-index 0	Name	Shutdown option code
	Access	Read and write
	PDO mapping	No
	Data type	SIGNED16
	Default value	0

This object decides what action is performed if there is a transition from Operation Enabled state to Ready To Switch On state.

Value	Action
0	Disable drive function
1	Slow down with slow down ramp Disable drive function

5.8.2.26.13 Object 0x605C, Disable operation option code

Object description	
Index	0x605C
Name	Disable operation option code
Object code	VAR
Data type	SIGNED16

Entry description		
Sub-index 0	Name	Disable operation option code
	Access	Read and write
	PDO mapping	No
	Data type	SIGNED16
	Default value	1

This object decides what action is performed if there is a transition from Operation Enabled state to Switched On state.

Value	Action
0	Disable drive function
1	Slow down with slow down ramp Disable drive function

5.8.2.26.14 Object 0x6060, Modes of operation

Object description	
Index	0x6060
Name	Modes of operation
Object code	VAR
Data type	SIGNED8

Entry description		
Sub-index 0	Name	Modes of operation
	Access	Read and write
	PDO mapping	RxPDO
	Data type	SIGNED8
	Default value	-1

This object decides what mode of operation the drive shall operate in. The option board only supports the vendor specific operation mode, but the inverter can use both velocity and torque control is this mode.

Value	Mode of operation
-1	Vendor specific

5.8.2.26.15 Object 0x6061, Modes of operation display

Object description	
Index	0x6061
Name	Modes of operation display
Object code	VAR
Data type	SIGNED8

Entry description		
Sub-index 0	Name	Modes of operation display
	Access	Read only
	PDO mapping	TxPDO
	Data type	SIGNED8
	Default value	-1

This object shows what mode of operation the drive is operating in. The option board only supports the vendor specific operation mode, but the inverter can use both velocity and torque control is this mode.

Value	Mode of operation
-1	Vendor specific

5.8.2.26.16 Object 0x6071, Target torque

Object description	
Index	0x6071
Name	Target torque
Object code	VAR
Data type	SIGNED16
Note: This object is not supported on the F800 drive series.	

Entry description		
Sub-index 0	Name	Target torque
	Access	Read and write
	PDO mapping	RxPDO
	Data type	SIGNED16
	Default value	0
	Unit	10^{-3} * Rated torque

5.8.2.26.17 Object 0x6075, Motor rated current

Object description	
Index	0x6075
Name	Motor rated current
Object code	VAR
Data type	UNSIGNED32

Entry description		
Sub-index 0	Name	Motor rated current
	Access	Read and write
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	-
	Unit	mA

5.8.2.26.18 Object 0x6076, Motor rated torque

Object description	
Index	0x6076
Name	Motor rated torque
Object code	VAR
Data type	UNSIGNED32

Entry description		
Sub-index 0	Name	Motor rated torque
	Access	Read and write
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	-
	Unit	mNm

5.8.2.26.19 Object 0x6077, Torque actual value

Object description	
Index	0x6077
Name	Torque actual value
Object code	VAR
Data type	SIGNED16
Note: This object is not supported on the F800 drive series.	

Entry description		
Sub-index 0	Name	Torque actual value
	Access	Read only
	PDO mapping	TxPDO
	Data type	SIGNED16
	Default value	0
	Unit	10^{-3} * Rated torque

5.8.2.26.20 Object 0x6087, Torque slope

Object description	
Index	0x6087
Name	Torque slope
Object code	VAR
Data type	UNSIGNED32
Note: This object is not supported on the F800 drive series.	

Entry description		
Sub-index 0	Name	Torque actual value
	Access	Read and write
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0
	Unit	10^{-3} * Rated torque/s

5.8.2.26.21 Object 0x6088, Torque profile type

Object description	
Index	0x6088
Name	Torque profile type
Object code	VAR
Data type	SIGNED16
Note: This object is not supported on the F800 drive series.	

Entry description		
Sub-index 0	Name	Torque profile type
	Access	Read and write
	PDO mapping	No
	Data type	SIGNED16
	Default value	0

This object determines the torque profile type.

Value	Torque profile type
0	Linear ramp
1	$\sin^2$ ramp

5.8.2.26.22 Object 0x6402, Motor type

Object description	
Index	0x6402
Name	Motor type
Object code	VAR
Data type	UNSIGNED16

Entry description		
Sub-index 0	Name	Motor type
	Access	Read and write
	PDO mapping	No
	Data type	UNSIGNED16
	Default value	-

This object indicates the type of motor attached to and driven by the drive.

Value	Motor type
0	Non standard motor
1	Phase modulated DC motor
2	Frequency controlled DC motor
3	PM synchronous motor
4	FC synchronous motor
5	Switched reluctance motor
6	Wound rotor induction motor
7	Squirrel cage induction motor
8	Stepper motor
9	Micro-step stepper motor
10	Sinusoidal PM BL motor
11	Trapezoidal PM BL motor
12	AC synchronous reluctance sync
13	DC commutator PM
14	DC commutator wound field series
15	DC commutator wound field shunt
16	DC commutator wound field compound

5.8.2.26.23 Object 0x6502, Supported drive modes

Object description	
Index	0x6502
Name	Supported drive modes
Object code	VAR
Data type	UNSIGNED32

Entry description		
Sub-index 0	Name	Supported drive modes
	Access	Read only
	PDO mapping	No
	Data type	UNSIGNED32
	Default value	0x00010000

This object indicates what drive modes of operation that is supported. The option board only supports one vendor specific operation mode.

5.8.3 Emergency requests

Emergency requests are initiated by the EtherCAT slave to notify the master about unexpected events. It is up to the master to make sure emergency messages are read from the slave when created.

The option board will send out CoE emergency requests to notify the master about fault and error events. See section 5.11 for more information.

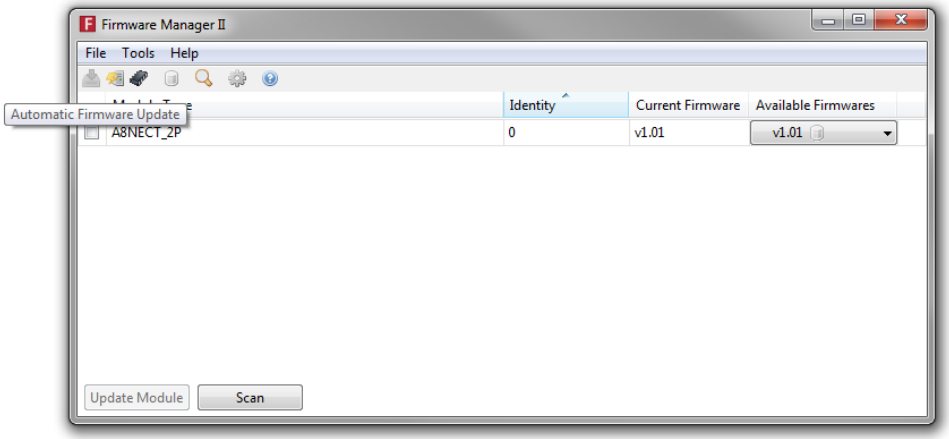
5.9 File access over EtherCAT

The option board supports the File access over EtherCAT (FoE) protocol for upgrade of the option board firmware.

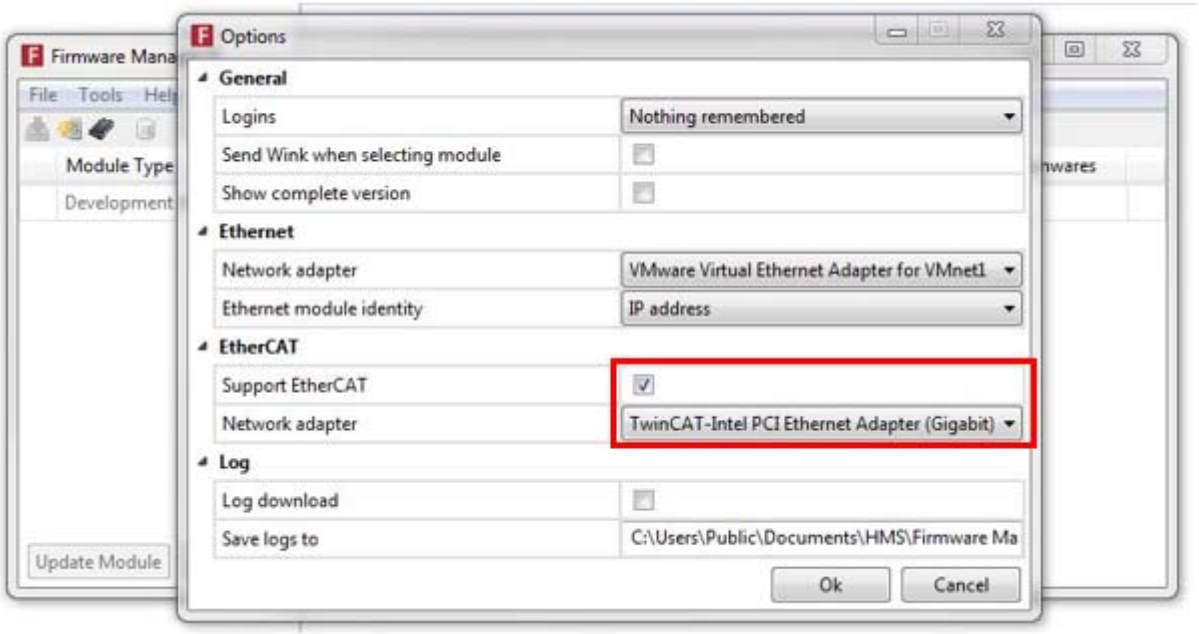
5.9.1 Firmware upgrade

To upgrade the option board firmware follow these steps:

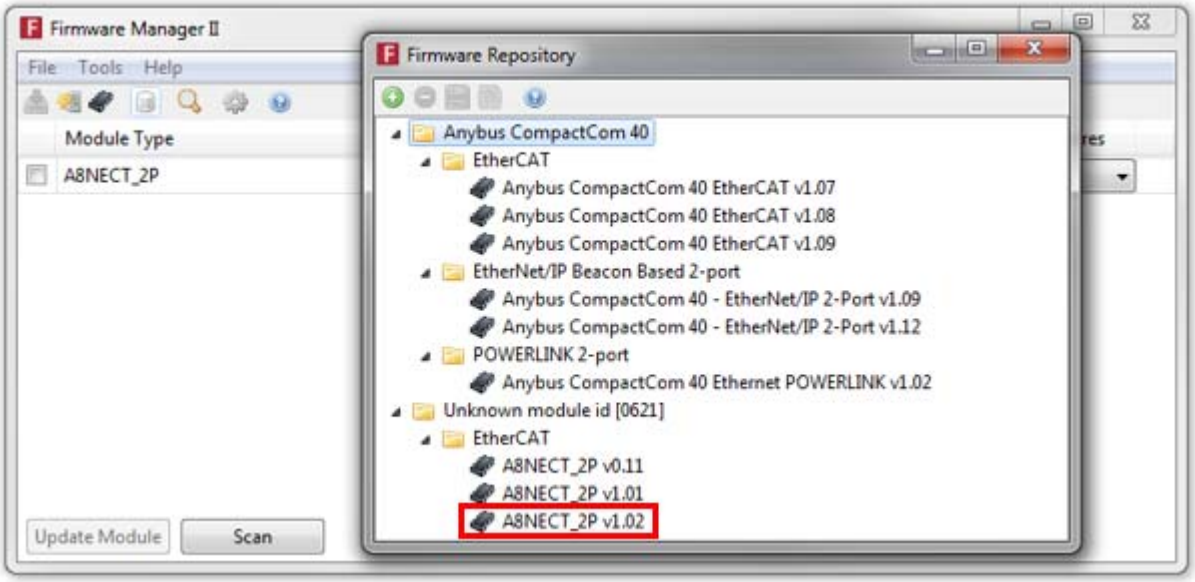
1. Download Firmware Manager II from the HMS Anybus website. See section P.2 Download
2. for the link.
3. Install Firmware Manager II and start it.
4. Make sure Firmware Manager II is in the “Automatic Firmware Update” mode.



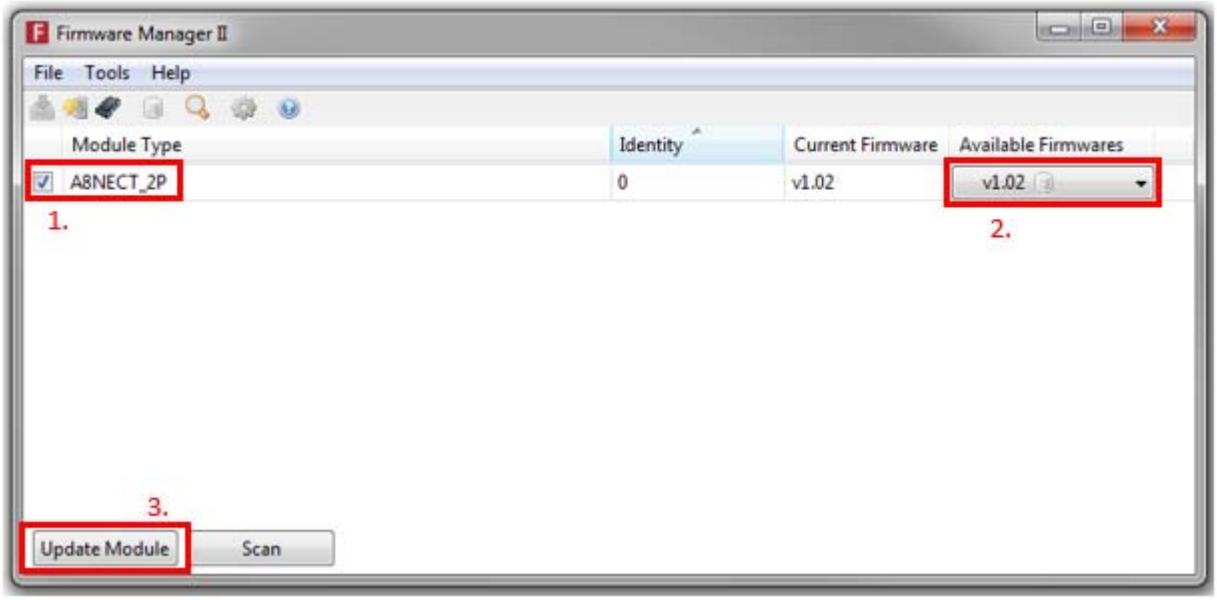
5. Go to Tools->Options and enable support for EtherCAT and select the network adapter used for EtherCAT communication.



6. Go to File->Firmware repository and import the .hiff firmware file provided by HMS.



7. Check the A8NECT_2P that shall be updated, select the correct firmware version and press “Update Module”.



8. The firmware upgrade process will now start. After a while the option board will be reset and start flashing itself. The firmware upgrade process is finished once all LEDs on the option board are off. **DO NOT POWER CYCLE THE DRIVE BEFORE FIRMWARE UPGRADE IS FINISHED.**
9. Power cycle the drive.

It is also possible to download the .hiff firmware file using any EtherCAT master with FoE support. In this case it is important that the file ending .hiff is present in the FoE download request, otherwise the request will be NAK:ed by the option board.

To upgrade the firmware using an EtherCAT master follow these steps:

1. Set the option board EtherCAT state to BOOTSTRAP.
2. Download the .hiff firmware file provided by HMS.
3. Change the EtherCAT state of the option board to INIT. This will trigger a reset of the option board and it will start flashing itself. The firmware upgrade process is finished once all LEDs on the option board are off. **DO NOT POWER CYCLE THE DRIVE BEFORE FIRMWARE UPGRADE IS FINISHED.**
4. Power cycle the drive.

5.10 Data exchange

5.10.1 Parameter data

5.10.1.1 Standard inverter parameters

All the parameters in the inverter can be accessed by the EtherCAT master as CANopen over EtherCAT objects. The master can send SDO upload or SDO download requests towards these objects to access the data.

Inverter parameters are presented in the object index range 0x3000-0x3FFF. The object index corresponding to the inverter parameter is calculated according to the following formula:

Object index = Inverter parameter number + 0x3000

The access rights of the parameter objects depends on the access rights of the inverter parameter according to the table below.

Inverter parameter access rights	CoE object access rights
Read only	Read only
Read and write	Read and write

The data type of the parameter depends on the data type of the inverter parameter according to the table below.

Inverter parameter data type	CoE object data type
Signed, 16 bits	SIGNED16
Unsigned, 16 bits	UNSIGNED16
Signed, 32 bits	SIGNED32
Unsigned, 32 bits	UNSIGNED32

A complete list of all inverter parameters can be found in (Mitsubishi Electric).

5.10.1.2 Monitor items (process data parameters)

All the monitor items in the inverter can be accessed by the EtherCAT master as CANopen over EtherCAT objects. The master can send SDO upload or SDO download requests towards these objects to access the data.

The monitor items can also be mapped as process data, meaning they will be updated cyclically. The direction of process data they can be mapped on depends on the access rights of the inverter monitor item.

Monitor items are presented in the object index range 0x4000-0x4FFF. The object index corresponding to the monitor item is calculated according to the following formula:

Object index = Monitor item number + 0x4000

The access rights of the monitor item objects depends on the access rights of the inverter monitor item according to the table below.

Monitor item access rights	CoE object access rights
Read only	Read only, TxPDO mappable
Read and write	Read and write, RxPDO- and TxPDO mappable

The data type of the monitor item object depends on the data type of the inverter monitor item according to the table below.

Monitor item data type	CoE object data type
Signed, 16 bits	SIGNED16
Unsigned, 16 bits	UNSIGNED16
Signed, 32 bits	SIGNED32
Unsigned, 32 bits	UNSIGNED32

A complete list of all monitor items can be found in “Appendix 4 For customers using HMS network options” in (Mitsubishi Electric).

5.10.1.3 Drive profile parameters

All drive profile parameters are presented in the profile specific CANopen over EtherCAT object range (0x6000-0x67FFF). The master can send SDO upload or SDO download requests towards these objects to access the data.

Some of the drive profile parameters can also be mapped on process data, meaning they will be updated cyclically.

The available drive profile parameter objects are described in detail in section 5.8.2.26.

5.10.2 Process data

Process data is exchanged between the EtherCAT master and the A8NECT_2P on a cyclic basis. RxPDO data is sent from the master to the A8NECT_2P and TxPDO data is sent from the A8NECT_2P to the EtherCAT master.

5.10.2.1 PDO configuration

What CoE objects that shall be sent as process data is determined by the PDO configuration objects and the PDO assignment objects.

The following PDO configuration objects are available for RxPDO data:

- 0x1600 - See section 5.8.2.9 for detailed information.
- 0x1604 - See section 5.8.2.10 for detailed information.
- 0x1605 - See section 5.8.2.11 for detailed information.
- 0x1610 - See section 5.8.2.12 for detailed information.
- 0x1611 - See section 5.8.2.13 for detailed information.

The following PDO configuration objects are available for TxPDO data:

- 0x1A00 - See section 5.8.2.14 for detailed information.
- 0x1A04 - See section 5.8.2.15 for detailed information.
- 0x1A05 - See section 5.8.2.16 for detailed information.
- 0x1A10 - See section 5.8.2.17 for detailed information.
- 0x1A11 - See section 5.8.2.18 for detailed information.

All PDO configuration objects are static meaning they cannot be changed.

These PDO configuration objects are assigned to the PDO assignment objects 0x1C12 (for RxPDO data) and 0x1C13 (for TxPDO data).

The PDO assignment objects decides what PDO configuration objects that are used for process data configuration, and the PDO configuration objects decide what CoE objects are transferred as process data.

For example:

PDO configuration object 0x1605 contains the objects 0x6040 (Controlword) and 0x6042 (vl target velocity). When object 0x1605 is assigned to object 0x1C12 this means that the data for object 0x6040 and 0x6042 will be transmitted as RxPDO data.

5.10.2.2 User specific PDOs

The user can customize the process data configuration objects 0x1611 and 0x1A11 to make a custom process data mapping. This process is described in section 4.2.2.

5.10.2.3 Response time

The worst case process data delay through the A8NECT_2P option board is 1011 µs. This delay value is identical in both directions (RxPDO and TxPDO data).

This means that the A8NECT_2P option board adds a worst case delay of 2022 µs to the total data exchange. To determine how long time is needed before the inverter has reacted on the RxPDO data, captured the status data and sent this back to the master as TxPDO the timing of the inverter is needed. Refer to (Mitsubishi Electric) for this information.

5.11 Diagnostics

The inverter can report a number of fault codes to the option board. How these fault codes are translated into EtherCAT specific error codes is described in the table below.

Main unit fault code	Main unit fault name	Main unit fault description	Option board fault	Option board error code	Bit(s) in Object 0x1001
0x10	E.OC1	OC During Acc	Current, device output side	0x23	0x03
0x11	E.OC2	Steady spd OC	Current, device output side	0x23	0x03
0x12	E.OC3	OC During Dec	Current, device output side	0x23	0x03
0x20	E.OV1	OV During Acc	Output Voltage	0x33	0x05
0x21	E.OV2	Steady spd OV	Output Voltage	0x33	0x05
0x22	E.OV3	OV During Dec	Output Voltage	0x33	0x05
0x30	E.THT	Inv. Ovrload	Current, device output side	0x23	0x03
0x31	E.THM	Motor Ovrload	Current, device output side	0x23	0x03
0x40	E.FIN	H/Sink O/Temp	Device Temperature	0x42	0x09
0x50	E.IPF	Inst. Pwr. Loss	Mains Voltage	0x31	0x05
0x51	E.UVT	Undervoltage	Mains Voltage	0x31	0x05
0x52	E.ILF	Input phase loss	Mains Voltage	0x31	0x05
0x60	E.OLT	Stall Prev STP	Current, device output side	0xF0	0x01
0x61	E.SOT	Loss of synchronization	Generic Error	0x10	0x01
0x62	E.LUP	Upper limit fault detection	Generic Error	0x10	0x01
0x63	E.LDN	Lower limit fault detection	Generic Error	0x10	0x01
0x70	E.BE	Br. Cct. Fault	Device Hardware	0x50	0x01
0x80	E.GF	Ground Fault	Current, device output side	0x23	0x03
0x81	E.LF	Output phase loss	Output Voltage	0x33	0x05
0x90	E.OHT	OH Fault	Temperature	0x40	0x09
0x91	E.PTC	PTC Activated	Temperature	0x40	0x09
0xA0	E.OPT	Option fault	Generic Error	0x10	0x01
0xA1	E.OP1	Option1 fault	Monitoring, Communication	0x81	0x11
0xA2	E.OP2	Option2 fault	Monitoring, Communication	0x81	0x11
0xA3	E.OP3	Option3 fault	Monitoring, Communication	0x81	0x11
0xA4	E.16	Sequence function user-defined abnormal	Generic Error	0x10	0x01
0xA5	E.17				
0xA6	E.18				
0xA7	E.19				
0xA8	E.20				
0xB0	E.PE	Corrupt memory	Device Hardware	0x50	0x01
0xB1	E.PUE	PU Leave out	Monitoring, Communication	0x81	0x11
0xB2	E.RET	Retry No Over	Additional Functions	0xF0	0x01
0xB3	E.PE2	PR Storage Alarm	Device Hardware	0x50	0x01
0xC0	E.CPU	CPU Fault	Device Hardware	0x50	0x01
0xC1	E.CTE	PU Short cct	Current, inside the device	0x22	0x03
0xC2	E.P24	24VDC short cct	Current, inside the device	0x22	0x03
0xC4	E.CDO	OC Detect level	Current, device output side	0x23	0x03
0xC5	E.IOH	Inrush overheat	Current, device input side	0x21	0x03

Main unit fault code	Main unit fault name	Main unit fault description	Option board fault	Option board error code	Bit(s) in Object 0x1001
0xC6	E.SER	VFD Comm error	Monitoring, Communication	0x81	0x11
0xC7	E.AIE	Analog in error	Additional functions	0xF0	0x01
0xC8	E.USB	USB Comm error	Monitoring, Communication	0x81	0x11
0xC9	E.SAF	Safety cct fault	Additional Functions	0xF0	0x01
0xCA	E.PBT	Internal circuit abnormality	Generic Error	0x10	0x01
0xD0	E.OS	Overspeed	Additional Functions	0xF0	0x01
0xD1	E.OSD	Excess spd deviation	Additional Functions	0xF0	0x01
0xD2	E.ECT	Enc. Signal loss	Additional Functions	0xF0	0x01
0xD3	E.OD	Excess pos fault	Additional Functions	0xF0	0x01
0xD5	E.MB1	Brake seq fault	Additional Functions	0xF0	0x01
0xD6	E.MB2	Brake seq fault	Additional Functions	0xF0	0x01
0xD7	E.MB3	Brake seq fault	Additional Functions	0xF0	0x01
0xD8	E.MB4	Brake seq fault	Additional Functions	0xF0	0x01
0xD9	E.MB5	Brake seq fault	Additional Functions	0xF0	0x01
0xDA	E.MB6	Brake seq fault	Additional Functions	0xF0	0x01
0xDB	E.MB7	Brake seq fault	Additional Functions	0xF0	0x01
0xDC	E.EP	Enc. Phase Fault	Additional Functions	0xF0	0x01
0xDE	E.MP	Magnetic pole position unknown	Generic Error	0x10	0x01
0xE1	E.IAH	Internal temperature anomaly	Generic Error	0x10	0x01
0xE4	E.LCI	4mA Input loss abnormality	Generic Error	0x10	0x01
0xE5	E.PCH	PID pre-charge abnormality	Generic Error	0x10	0x01
0xE6	E.PID	PID signal abnormality	Generic Error	0x10	0x01
0xF1	E.1	Fault 1 (opt slot 1)	Additional Modules	0x70	0x01
0xF2	E.2	Fault 2 (opt slot 2)	Additional Modules	0x70	0x01
0xF3	E.3	Fault 3 (opt slot 3)	Additional Modules	0x70	0x01
0xF5	E.5	Fault 5	Device Hardware	0x50	0x01
0xF6	E.6	Fault 6	Device Hardware	0x50	0x01
0xF7	E.7	Fault 7	Device Hardware	0x50	0x01
0xFB	E.11	Fault 11	Additional functions	0xF0	0x01
0xFD	E.13	Fault 13	Device Hardware	0x50	0x01

Whenever the inverter reports a new fault the following takes place:

- The bit corresponding to the error type is set in object 0x1001, Error Register. NOTE: Regardless of the error type, bit 0 is always set in case of an error. See section 5.8.2.2 for more information.
- The option board error code is added to object 0x1003, Pre-defined error field. See section 5.8.2.3 for more information.
- A CoE emergency request is sent to the master containing the following information:

Byte	Description	Value
0-1	Emergency error code	Same as corresponding entry in the 0x1003 object. (0xee00, where ee is the option board error code.)
2	Error register	Content of the 0x1001 Error register object.
3-7	Manufacturer specific	Not used. Set to 0.

When the inverter removes the fault, the corresponding bit will be removed from the Error Register object and an emergency with error code “error reset (0x0000) is sent to the EtherCAT master.

6 Troubleshooting

Check the following sections if the option board or inverter isn't working as intended.

6.1 Inverter operation panel display shows an error

Operation panel error code	Inverter Err LED on LED cover	EtherCAT Err LED on LED cover	Possible cause	Action
--	Off	Alternating green/red	Firmware upgrade is ongoing	Wait until all LEDs are off and power cycle the drive.
	Off	Solid red	The option board encountered a major unrecoverable error	Power cycle drive. If that does not help contact your local Mitsubishi Electric representative for further assistance.
	Off	Off	The option board is not mounted properly	Check if the option board is mounted properly and in the correct option slot.
			Firwmare upgrade has failed	Contact your local Mitsubishi Electric representative for further assistance.
	Solid red	Off	Communication between option board and inverter failed	Power cycle drive. If that does not help contact your local Mitsubishi Electric representative for further assistance.
E.OP1	Off	Off	The EtherCAT master has changed the EtherCAT state from OPERATIONAL to a lower state (e.g. PRE-OPERATIONAL etc.)	Check the master configuration to make sure it never changes the state of the option board when it is not supposed to.
	Off	Double flash red	Process data watchdog has timed out.	Make sure the master always updates the RxPDO data within the configured watchdog timeout.
	Flashing red, 4 flashes	On	Option board was unable to initialize the drive correctly	Make sure the option board has access to write the drive parameters residing in the drive and power cycle the drive.

6.2 Drive Err LED on LED cover is indicating an error

Drive Err LED indication	Possible cause	Action
Solid red	Communication has failed between drive and option board	Power cycle drive. If that does not help contact your local Mitsubishi Electric representative for further assistance.
Double flash red	Invalid parameter mapped to process data.	Check the process data configuration and make sure that all objects mapped to process data can be mapped in the direction they are currently mapped. E.g. mapping an object that only can be mapped as TxPDO data in an RxPDO will cause this error.
Triple flash red	Too many parameters mapped to process data	Decrease the number of parameters that are mapped as process data.
Quadruple flash red	Option board was unable to initialize the drive correctly	Make sure the option board has access to write the drive parameters residing in the drive and power cycle the drive.

7 Copyright notices