AXL F BK EC - TUNNEL

AXL F BK EC bus coupler: Access to PDI objects and startup parameterization

Application note
105646_en_02

Make sure you always use the latest documentation.
It can be downloaded at phoenixcontact.net/products.

1 About this document

This document describes how to access the PDI objects of Axioline F modules from the AXL F BK EC bus coupler using tunnel objects. It also explains how to use the startup parameterization option.

Example

The following bus configuration is used here as an example:

AXL F BK EC	AXL F DI32/1 1F	AXL F DO32/1 1H	**AXL F AO8 1H**

Write and read access is explained using the example of the AXL F AO8 1H module.

The TwinCAT® software is used for explanatory purposes. However, the steps described here can also be applied to other tools.

2 Tunnel objects of the AXL F BK EC bus coupler

Parameter and diagnostic data as well as other information is transmitted via the PDI channel of the Axioline F station.

You can access PDI objects of the modules of a station via EtherCAT®.

Objects $20nn_{hex}$ for reading and objects $30nn_{hex}$ for writing are available, with which a tunnel method can be implemented. For each connected module, one object is available for reading and one for writing. “nn” refers to the module position in the Axioline F local bus, starting with 00.

Module after the bus coupler	nn	Index of the PDI Read Tunnel object	Index of the PDI Write Tunnel object
1st module	00	3000	2000
2nd module	01	3001	2001
...	...	...	...
63rd module	62	3062	2062

For the structure of the tunnel objects, please refer to the data sheet for the bus coupler.
For the PDI objects of the individual Axioline F modules, please refer to the module-specific documentation.

3 Tunnel objects in the software

- Open the project in the TwinCAT® System Manager.
- Select the AXL F BK EC EtherCAT® bus coupler in the project tree.
- Open the “CoE-Online” tab.

In order to be able to view all CoE objects, they must be imported first.

- Click on “Advanced ...”.
- In the window that opens, first select “Online - via SDO Information” and then “All Objects”. Confirm your selection with “OK”.

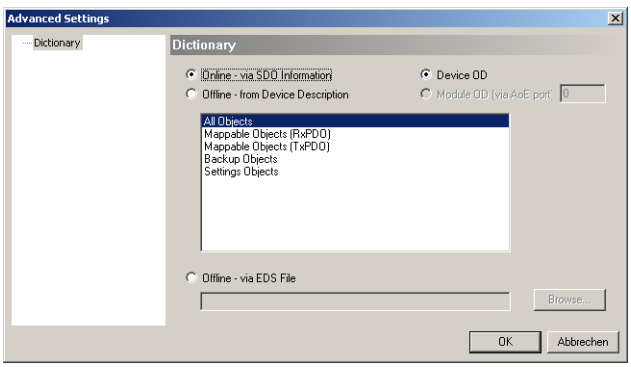


Figure 1 Reading all objects

A list including all CoE objects is now displayed under “CoE-Online”.

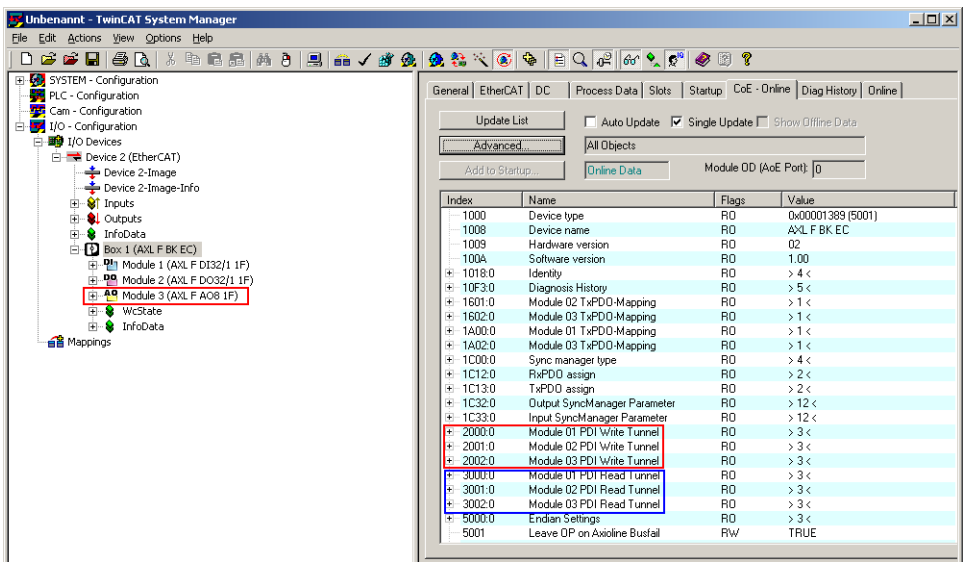


Figure 2 List including the CoE objects of the bus coupler

For the relevant third module, write access is carried out via index 2002 and read access via index 3002.

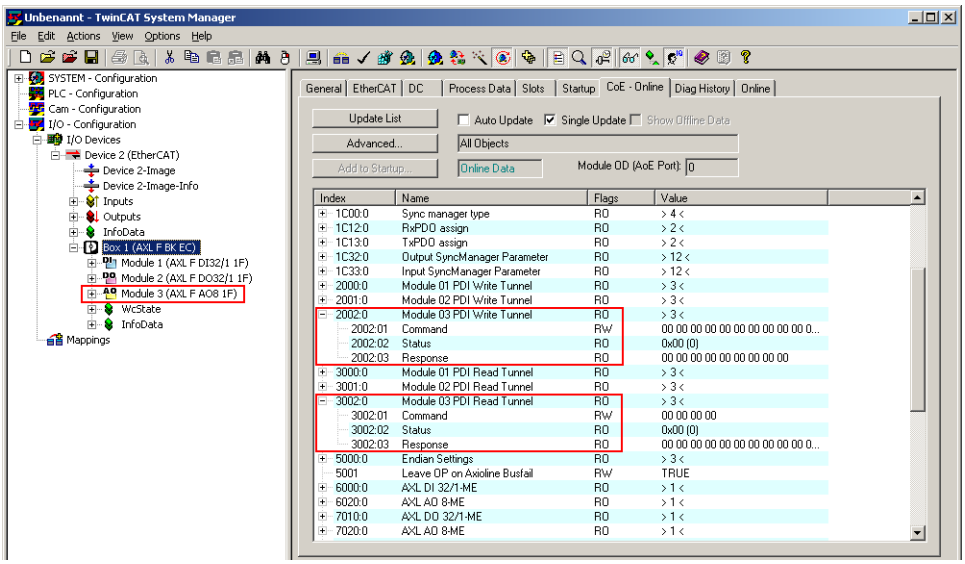


Figure 3 Tunnel objects of the AXL F AO8 1H module

4 Writing to PDI objects with PDI Write Tunnel

This CoE object allows writing to any PDI object where write access is permitted. In the following, write access is shown using a parameterization example.

Parameterization example

Please refer to the module-specific data sheet for parameterization options.

– The AXL F AO8 1H module needs to be parameterized. It is the third module after the bus coupler (nn = 02).
– Channel 1 needs to be parameterized within a range of -5 V ... +5 V (code 0003_{hex}).
– The parameterization of channel 1 is performed via the PDI object 0080_{hex} (ParaTable), subindex 1.

Procedure

• In index 2002, double-click subindex 1, Command.

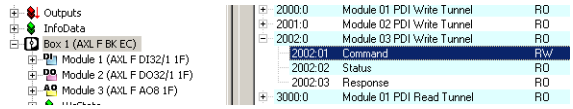


Figure 4 Command of PDI Write Tunnel for the AXL F AO8 1H module

The “Set Value Dialog” box opens.

• In the “Binary” line, enter the octet string for the parameterization.

A binary code is not required, even if the parameterization is entered under “Binary”.

“Command” has a maximum length of 250 bytes. However, only the user data needs to be entered.
Set the following values for “Command”:

Data for the PDI write request			
Byte 0	Subslot	00	No subslot
Byte 1, 2	PDI object index	0080	ParaTable PDI object
Byte 3	PDI object subindex	01	Subindex 1 = channel 1
Byte 4	Length of the data to be written	00	Indication is not necessary
Bytes 5 ... n	User data	0003	-5 V ... +5 V range

This results in the following octet string for the parameterization: 00 00 80 01 00 00 03

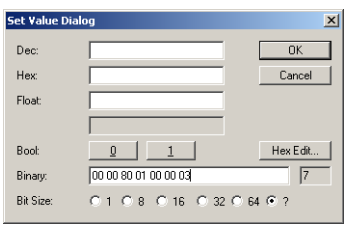


Figure 5 Parameterization value

• Confirm your entry with “OK”.

Result

Subindex 02 (Status) and subindex 03 (Response) contain the result.

– Positive write access

If the status is 01_{hex}, the write access has been completed successfully. The response is in this case irrelevant.

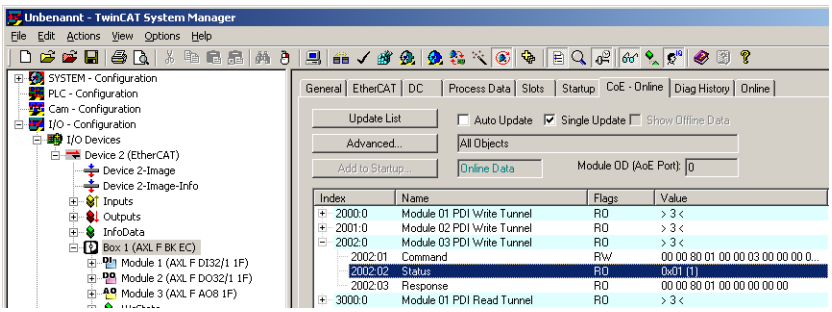


Figure 6 Write access successful

– Negative write access

An error has occurred if the status is 03_{hex}. The response contains the mirrored command and the error code. Apart from including the error code in the response, the error will also be indicated in the message window.

The diagnostic code included in the response is reported by the module or the Axioline F system.

Diagnostics in the message window are reported by the EtherCAT® system, including the diagnostics code and plain text.

The two messages do not have to be identical.

For the meaning of the error code included in the response, please refer to the module-specific data sheet or the AXL F SYS DIAG user manual.

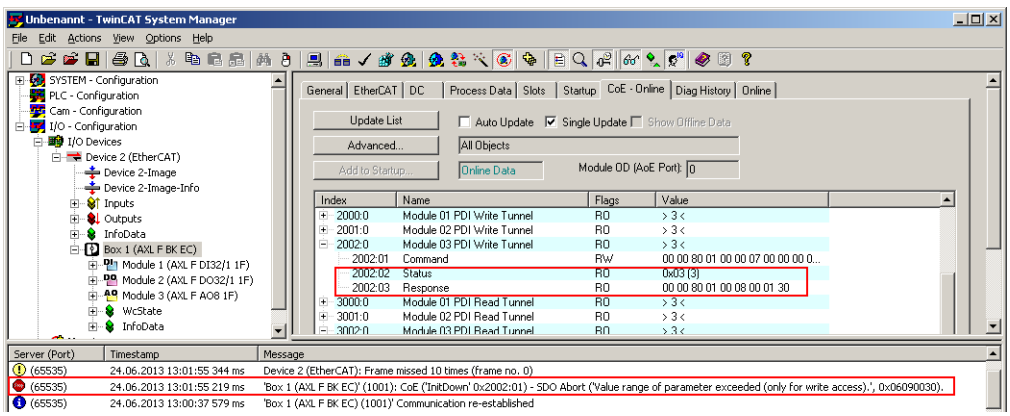


Figure 7 Write access not successful

Diagnostic messages

– Message in the response

00	00 80	01	00	08	00	01	30
Slot	ParaTable PDI object index	Subin- dex	Length (= 0)	Error class	Error code	Additional code	

In the example, the following error has occurred:
A reserved bit or reserved code was used during parameterization.
Additional code: 01 = channel 1, 30 = value is out of range

– Message in the message window

SDI Abort (Value range of parameter exceeded (only for write access)., 0x06090030.

5 Reading PDI objects with PDI Read Tunnel

This CoE object allows reading to any PDI object where read access is permitted.

Example

– The firmware version should be read from the AXL F AO8 1H module. It is the third module after the bus coupler (nn = 02).
– The firmware version is read using the $000C_{hex}$ object FirmwareVersion.

Procedure

In order to read PDI objects, proceed in the same way as described for writing.

- In index 3002, double-click subindex 1, Command.

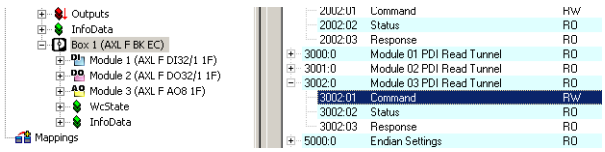


Figure 8 Command of PDI Read Tunnel for the AXL F AO8 1H module

The “Set Value Dialog” box opens.

- Enter the octet string in the “Binary” line.

Data for the PDI read request			
Byte 0	Subslot	00	No subslot
Byte 1, 2	PDI object index	000C	FirmwareVersion PDI object
Byte 3	PDI object subindex	00	No subindex

This results in the value 00 00 0C 00.

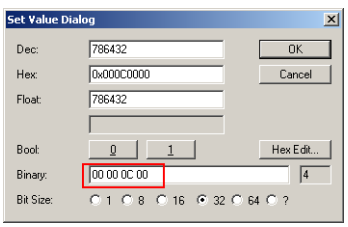


Figure 9 Parameter for reading

Result

Subindex 02 (Status) and subindex 03 (Response) contain the result.

– Positive read access

If the status is 01_{hex}, the read access has been completed successfully. The response includes the mirrored request and the read data.

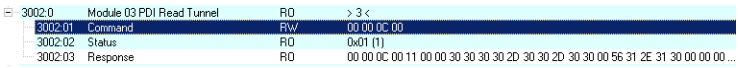


Figure 10 Read access successful; firmware version: 0000-00-00; V1.10

00	00 0C	00	11	00 00	30 30 30 30 2D 30 30 2D 30 30 00 56 31 2E 31 30 00 ...
Slot	FirmwareVersion PDI object index	Subin-dex	Length 17 bytes	-	0000-00-00 \0 V1.10 \0

– Negative read access

An error has occurred if the status is 03_{hex}. The response contains the mirrored command and the error code. Apart from including the error code in the response, the error will also be indicated in the message window. See also “– Negative write access” on page 5).

6 Startup parameterization

You can store a startup parameterization for the connected Axioline F modules. After replacing a module, the new module will start up with this parameterization.

Startup parameterization example

– The startup parameterization needs to be stored for the AXL F AO8 1H module. It is the third module after the bus coupler (nn = 02).
– Channel 1 needs to be parameterized within a range of -5 V ... +5 V (code 0003_{hex}).
– The parameterization of channel 1 is performed via the PDI object 0080_{hex} (ParaTable), subindex 1.

Procedure

• In order to store the startup parameterization, switch to the “Startup” tab.
• Click the “New...” button.

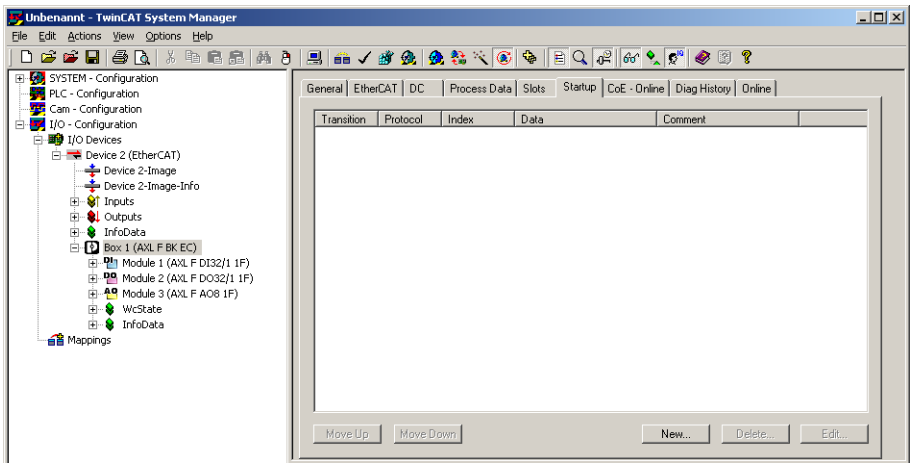


Figure 11 Startup window for storing the startup parameterization

• In the window that opens, specify under “Transition” the state where parameterization should be applied.

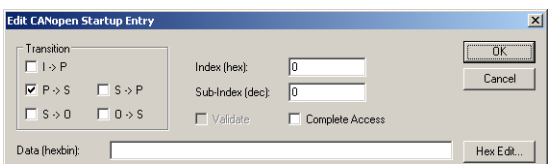


Figure 12 Transition specification in order to apply the parameterization

- Select the index 2002, Command, in the bottom window.
- Specify the parameterization in the “Data (hexbin)” field.
  In the example, the parameterization used is the same as before (see “Writing to PDI objects with PDI Write Tunnel” on page 4).
- Confirm your entry with “OK”.

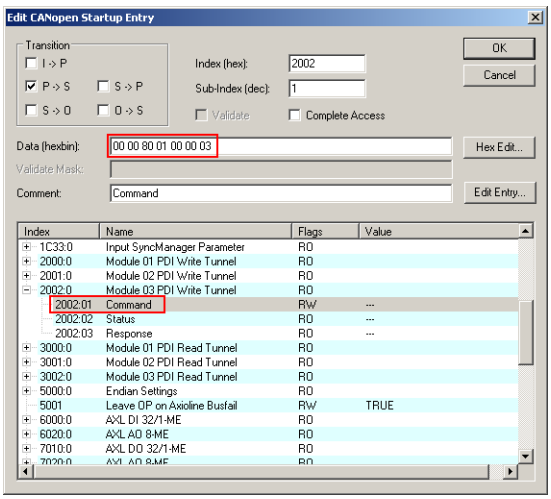


Figure 13 Specifying the startup parameterization

The startup parameterization has been stored and will be applied with the next startup of the module.

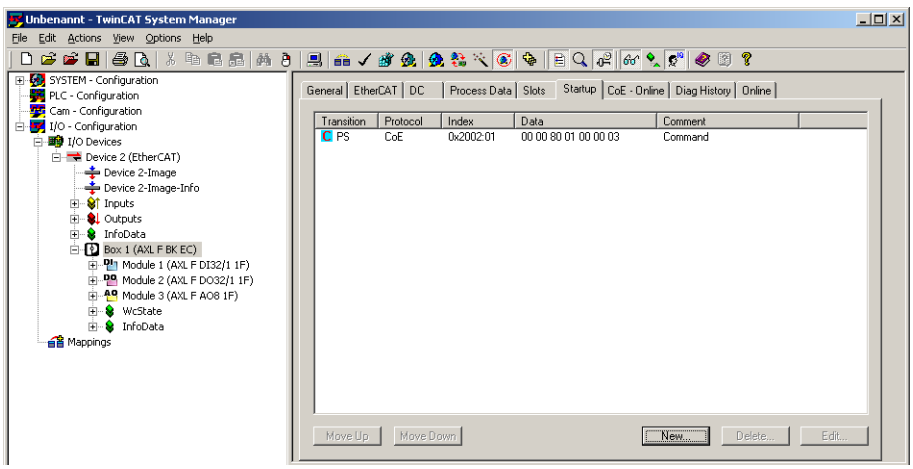


Figure 14 Parameter for reading

All changes made after the module startup will only be stored on the module. When you replace the module, the changes will not be applied.

- Repeat these steps if you want to store the startup parameterization for other modules.

SCATTERGOOD & JOHNSON LTD
ELECTRICAL ENGINEERING & FLUID CONTROL DISTRIBUTORS
Est.1899

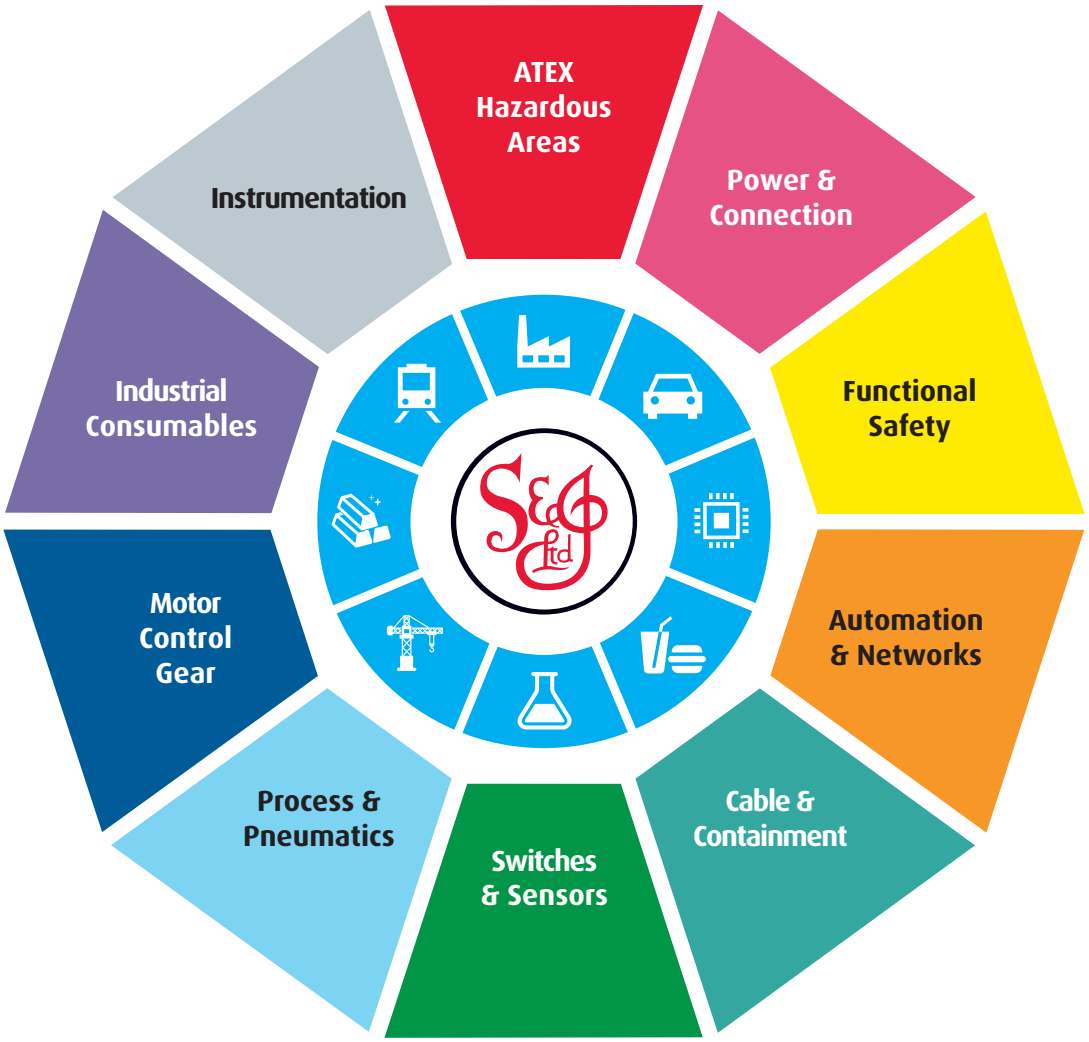
At Scattergood & Johnson Ltd, we pride ourselves on being a technical distributor to specialist industries.

Working with a range of quality product suppliers across a number of specialist markets, we are not your average 'box shifter' - we are your technical and supply chain partner.

We fully support every product we sell - for free! Our internal team and external sales engineers can answer any product or application question, no matter the complexity.

Backing up this technical ability is a range of 50,000+ products available from stock for nationwide next day delivery (same day if required!), or you can collect what you need from any of our trade counters around the UK.

Select your specialist interest below to learn more about how we can help.



Online, In Branch and On the Road - Scattergood & Johnson Ltd, there when you need us.

www.scatts.co.uk