

EtherCAT®

CANopen®

Kuhnke FIO I/O System

Product Manual: OC Modules

12/03/2024

Table of Contents

- 1 Preface3
 - 1.1 Legal Notice.....3
 - 1.2 About this Manual.....3
- 2 OC Modules.....4
 - 2.1 Generalities4
 - 2.1.1 Kendrion Kuhne OC Technology4
 - 2.1.2 Numeric Values4
 - 2.1.3 CoE – CANopen over EtherCAT5
 - 2.2 FIO OC AI4.....6
 - 2.2.1 Function.....6
 - 2.2.2 Object dictionary: Extended OC Process Data Objects.....7
 - 2.3 FIO OC Counter/Encoder15
 - 2.3.1 Function.....15
 - 2.3.2 Object dictionary: Extended OC Process Data Objects.....16
 - 2.4 Use with CODESYS Safety24
 - 2.4.1 Prerequisites24
 - 2.4.2 Applicability24
 - 2.4.3 Other Applicable Documents24
 - 2.4.4 Example Application.....25
- 3 Appendix.....32
 - 3.1 Order Data32
 - 3.1.1 Modules.....32
 - 3.1.2 Accessories32

1 Preface

1.1 Legal Notice

Contact Details

Kendrion Kuhnke Automation GmbH
Industrial Control Systems
Lütjenburger Str. 101
D-23714 Malente
Germany

Support (phone) +49 4523 402-300
Support (email) controltechnology-ics@kendrion.com
Switchboard +49 4523 402-0
Sales (email) sales-ics@kendrion.com
Internet www.kendrion.com

Document History

Modification History

Date	Comments / Modifications
12.03.2024	New document structure created according to module groups

1.2 About this Manual

This technical information is primarily directed to system designers, project engineers and device developers. It does not contain any availability information. We reserve the rights for errors, omissions and modifications. Pictures are similar.

This product manual extends the system, installation and safe handling information provided by the Kuhnke FIO System Manual. This product manual only applies in conjunction with the system manual.

2 OC Modules

2.1 Generalities

The OC modules are comprised of a group of Kuhnke FIO modules expanded by process data objects that allow their integration in safety applications.

2.1.1 Kendrion Kuhne OC Technology

Apart from the normal EtherCAT process data, Kuhnke's FIO OC modules can also use "safeguarded" OC data containers for communication. Various mechanisms are deployed to safeguard communication by recognising data loss, interruptions, changed sequences or data corruption. These mechanisms provide sensor values in extra OC data containers. In other words, they additionally safeguard the process data within the EtherCAT communication processes.

OC is short for **One Channel**, which is to say that these modules are standard, but single-channel modules not containing a two-channel safety architecture to enable their functional safety feature.

Nevertheless, Kuhnke's FIO OC modules still allow your safety-related application to make use of two non safety-related signals. They become part of a two-channel input architecture combined to standard components of diverse technologies. A safe control unit (Kuhnke FIO Safety PLC) can then process both signals to generate a safety-related result.

Extended OC Process Data Objects

The PDO mapping contains a record dedicated to the process data objects you need for an OC data container. These objects are made up as follows:

Variable	Data type	Explanation
CycleCount	UDINT	Cycle count: increments with every communication cycle of the module.
Timestamp	UDINT	Time stamp: generated by the module
SenderId	UINT	Sender identification: equivalent to an EtherCAT not address with a channel number affixed to it. Example EtherCAT node address: 1003 Channel: 2 Resulting SenderId: 10032
Value	DINT	Input value as Field Value Inc
Crc32	UDINT	Checksum: calculated by the module from the above data.

2.1.2 Numeric Values

As a general rule, numeric values are shown as decimals.

A prefixed 0x marks hexadecimal values (example: 0xFFFF)

A prefixed 0b marks binary values (example: 0b01010011)

Objects from the object dictionary generally show as hexadecimal value.

2.1.3 CoE – CANopen over EtherCAT

Communication protocol CANopen is based on CAN and designed to interconnect automation devices. Communication profiles for various device classes harmonise device operations and simplify their handling.

EtherCAT features the same communication mechanisms as CANopen, i.e. an object dictionary, process data objects (PDOs), service data objects (SDOs), and a similar network management methodology.

The object dictionary describes the objects available to the EtherCAT slave. It distinguishes between objects with read access (read), write access (write) and read/write access (read/write). The dictionary also classifies these objects as (mappable) process data objects, if so.


A cyclic process exchanges the process data objects (PDOs) and the input and output data they normally carry. Depending on your EtherCAT slave, you may be able to add (map) further object dictionary variables.

Service data objects (SDOs) provide options like setting the parameters of EtherCAT slaves and adding them to the startup parameters. They will then be automatically transferred to the EtherCAT slave as the EtherCAT bus starts up. Setting up the parameters is as easy as that. And if you have to replace your EtherCAT slave, you can simply use another EtherCAT slave of the same type.

2.2 FIO OC AI4

2.2.1 Function

Module Kuhnke FIO OC AI4 is very similar to module Kuhnke FIO AI4 / AO4, except that it has no analogue outputs. Both modules share the same configuration of analogue inputs. The process image of FIO OC AI4 additionally has OC data containers for each of the analogue input channels.

	Information
	<i>Please note the product description of module Kuhnke FIO AI4 / AO4 in product manual Kuhnke FIO Analogue I/O Modules and the Kuhnke FIO System Manual.</i>

2.2.2 Object dictionary: Extended OC Process Data Objects

0x1a0c One Channel Field Value AI0 (Record)

Sub	0x00
Name	SubIndex 000
Data type	UNSIGNED8
Access	R/O
Default value	5
PDO mapping	No

Sub	0x01
Name	Mapping entry 1
Data type	UNSIGNED32
Access	R/O
Default value	0x40000120
PDO mapping	No

Sub	0x02
Name	Mapping entry 2
Data type	UNSIGNED32
Access	R/O
Default value	0x40000220
PDO mapping	No

Sub	0x03
Name	Mapping entry 3
Data type	UNSIGNED32
Access	R/O
Default value	0x40000310
PDO mapping	No

Sub	0x04
Name	Mapping entry 4
Data type	UNSIGNED32
Access	R/O
Default value	0x40000420
PDO mapping	No

Sub	0x05
Name	Mapping entry 5
Data type	UNSIGNED32
Access	R/O
Default value	0x40000520
PDO mapping	No

0x1a0d One Channel Field Value AI1 (Record)

Sub	0x00
Name	SubIndex 000
Data type	UNSIGNED8
Access	R/O
Default value	5
PDO mapping	No

Sub	0x01
Name	Mapping entry 1
Data type	UNSIGNED32
Access	R/O
Default value	0x40010120
PDO mapping	No

Sub	0x02
Name	Mapping entry 2
Data type	UNSIGNED32
Access	R/O
Default value	0x40010220
PDO mapping	No

Sub	0x03
Name	Mapping entry 3
Data type	UNSIGNED32
Access	R/O
Default value	0x40010310
PDO mapping	No

Sub	0x04
Name	Mapping entry 4
Data type	UNSIGNED32
Access	R/O
Default value	0x40010420
PDO mapping	No

Sub	0x05
Name	Mapping entry 5
Data type	UNSIGNED32
Access	R/O
Default value	0x40010520
PDO mapping	No

0x1a0e One Channel Field Value AI2 (Record)

Sub	0x00
Name	SubIndex 000
Data type	UNSIGNED8
Access	R/O
Default value	5
PDO mapping	No

Sub	0x01
Name	Mapping entry 1
Data type	UNSIGNED32
Access	R/O
Default value	0x40020120
PDO mapping	No

Sub	0x02
Name	Mapping entry 2
Data type	UNSIGNED32
Access	R/O
Default value	0x40020220
PDO mapping	No

Sub	0x03
Name	Mapping entry 3
Data type	UNSIGNED32
Access	R/O
Default value	0x40020310
PDO mapping	No

Sub	0x04
Name	Mapping entry 4
Data type	UNSIGNED32
Access	R/O
Default value	0x40020420
PDO mapping	No

Sub	0x05
Name	Mapping entry 5
Data type	UNSIGNED32
Access	R/O
Default value	0x40020520
PDO mapping	No

0x1a0f One Channel Field Value AI3 (Record)

Sub	0x00
Name	SubIndex 000
Data type	UNSIGNED8
Access	R/O
Default value	5
PDO mapping	No

Sub	0x01
Name	Mapping entry 1
Data type	UNSIGNED32
Access	R/O
Default value	0x40030120
PDO mapping	No

Sub	0x02
Name	Mapping entry 2
Data type	UNSIGNED32
Access	R/O
Default value	0x40030220
PDO mapping	No

Sub	0x03
Name	Mapping entry 3
Data type	UNSIGNED32
Access	R/O
Default value	0x40030310
PDO mapping	No

Sub	0x04
Name	Mapping entry 4
Data type	UNSIGNED32
Access	R/O
Default value	0x40030420
PDO mapping	No

Sub	0x05
Name	Mapping entry 5
Data type	UNSIGNED32
Access	R/O
Default value	0x40030520
PDO mapping	No

0x4000 AI0 OC Fieldvalue (Record)

Sub	0x00
Name	Highest sub-index supported
Data type	UNSIGNED8
Access	R/O
Default value	5
PDO mapping	No

Sub	0x01
Name	CycleCnt
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI0OCFieldvalue.CycleCnt

Sub	0x02
Name	Timestamp
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI0OCFieldvalue.Timestamp

Sub	0x03
Name	SenderId
Data type	UNSIGNED16
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI0OCFieldvalue.SenderId

Sub	0x04
Name	Value
Data type	INTEGER32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI0OCFieldvalue.Value

Sub	0x05
Name	Crc32
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI0OCFieldvalue.Crc32

0x4001 AI1 OC Fieldvalue (Record)

Sub	0x00
Name	Highest sub-index supported
Data type	UNSIGNED8
Access	R/O
Default value	5
PDO mapping	No

Sub	0x01
Name	CycleCnt
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI1OCFieldvalue.CycleCnt

Sub	0x02
Name	Timestamp
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI1OCFieldvalue.Timestamp

Sub	0x03
Name	SenderId
Data type	UNSIGNED16
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI1OCFieldvalue.SenderId

Sub	0x04
Name	Value
Data type	INTEGER32
Access	R/O
Default value	
PDO mapping	Optional, TPDO only
Access name	AI1OCFieldvalue.Value

Sub	0x05
Name	Crc32
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI1OCFieldvalue.Crc32

0x4002 AI2 OC Fieldvalue (Record)

Sub	0x00
Name	Highest sub-index supported
Data type	UNSIGNED8
Access	R/O
Default value	5
PDO mapping	No

Sub	0x01
Name	CycleCnt
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI2OCFieldvalue.CycleCnt

Sub	0x02
Name	Timestamp
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI2OCFieldvalue.Timestamp

Sub	0x03
Name	SenderId
Data type	UNSIGNED16
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI2OCFieldvalue.SenderId

Sub	0x04
Name	Value
Data type	INTEGER32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI2OCFieldvalue.Value

Sub	0x05
Name	Crc32
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI2OCFieldvalue.Crc32

0x4003 AI3 OC Fieldvalue (Record)

Sub	0x00
Name	Highest sub-index supported
Data type	UNSIGNED8
Access	R/O
Default value	5
PDO mapping	No

Sub	0x01
Name	CycleCnt
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI3OCFieldvalue.CycleCnt

Sub	0x02
Name	Timestamp
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI3OCFieldvalue.Timestamp

Sub	0x03
Name	SenderId
Data type	UNSIGNED16
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI3OCFieldvalue.SenderId


Sub	0x04
Name	Value
Data type	INTEGER32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI3OCFieldvalue.Value

Sub	0x05
Name	Crc32
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	AI3OCFieldvalue.Crc32

2.3 FIO OC Counter/Encoder

2.3.1 Function

Module Kuhnke FIO OC Counter/Encoder is very similar to module Kuhnke FIO Counter/Encoder. The both share the same configuration. The process image of FIO OC Counter/Encoder additionally has OC data containers for the positions and speeds of the encoder inputs.

	Information
	<i>Please note the product description of module Kuhnke FIO Counter/Encoder in product manual Kuhnke FIO Counter Modules and the Kuhnke FIO System Manual.</i>

2.3.2 Object dictionary: Extended OC Process Data Objects

0x1a08 One Channel Position Encoder 1 (Record)

Sub	0x00
Name	SubIndex 000
Data type	UNSIGNED8
Access	R/O
Default value	5
PDO mapping	No

Sub	0x01
Name	Mapping entry 1
Data type	UNSIGNED32
Access	R/O
Default value	0x40000120
PDO mapping	No

Sub	0x02
Name	Mapping entry 2
Data type	UNSIGNED32
Access	R/O
Default value	0x40000220
PDO mapping	No

Sub	0x03
Name	Mapping entry 3
Data type	UNSIGNED32
Access	R/O
Default value	0x40000310
PDO mapping	No

Sub	0x04
Name	Mapping entry 4
Data type	UNSIGNED32
Access	R/O
Default value	0x40000420
PDO mapping	No

Sub	0x05
Name	Mapping entry 5
Data type	UNSIGNED32
Access	R/O
Default value	0x40000520
PDO mapping	No

0x1a09 One Channel Speed Encoder 1 (Record)

Sub	0x00
Name	SubIndex 000
Data type	UNSIGNED8
Access	R/O
Default value	5
PDO mapping	No

Sub	0x01
Name	Mapping entry 1
Data type	UNSIGNED32
Access	R/O
Default value	0x40010120
PDO mapping	No

Sub	0x02
Name	Mapping entry 2
Data type	UNSIGNED32
Access	R/O
Default value	0x40010220
PDO mapping	No

Sub	0x03
Name	Mapping entry 3
Data type	UNSIGNED32
Access	R/O
Default value	0x40010310
PDO mapping	No

Sub	0x04
Name	Mapping entry 4
Data type	UNSIGNED32
Access	R/O
Default value	0x40010420
PDO mapping	No

Sub	0x05
Name	Mapping entry 5
Data type	UNSIGNED32
Access	R/O
Default value	0x40010520
PDO mapping	No

0x1a0a One Channel Position Encoder 2 (Record)

Sub	0x00
Name	SubIndex 000
Data type	UNSIGNED8
Access	R/O
Default value	5
PDO mapping	No

Sub	0x01
Name	Mapping entry 1
Data type	UNSIGNED32
Access	R/O
Default value	0x40020120
PDO mapping	No

Sub	0x02
Name	Mapping entry 2
Data type	UNSIGNED32
Access	R/O
Default value	0x40020220
PDO mapping	No

Sub	0x03
Name	Mapping entry 3
Data type	UNSIGNED32
Access	R/O
Default value	0x40020310
PDO mapping	No

Sub	0x04
Name	Mapping entry 4
Data type	UNSIGNED32
Access	R/O
Default value	0x40020420
PDO mapping	No

Sub	0x05
Name	Mapping entry 5
Data type	UNSIGNED32
Access	R/O
Default value	0x40020520
PDO mapping	No

0x1a0b One Channel Speed Encoder 2 (Record)

Sub	0x00
Name	SubIndex 000
Data type	UNSIGNED8
Access	R/O
Default value	5
PDO mapping	No

Sub	0x01
Name	Mapping entry 1
Data type	UNSIGNED32
Access	R/O
Default value	0x40030120
PDO mapping	No

Sub	0x02
Name	Mapping entry 2
Data type	UNSIGNED32
Access	R/O
Default value	0x40030220
PDO mapping	No

Sub	0x03
Name	Mapping entry 3
Data type	UNSIGNED32
Access	R/O
Default value	0x40030310
PDO mapping	No

Sub	0x04
Name	Mapping entry 4
Data type	UNSIGNED32
Access	R/O
Default value	0x40030420
PDO mapping	No

Sub	0x05
Name	Mapping entry 5
Data type	UNSIGNED32
Access	R/O
Default value	0x40030520
PDO mapping	No

0x4000 Enc01 OC Position (Record)

Sub	0x00
Name	Highest sub-index supported
Data type	UNSIGNED8
Access	R/O
Default value	5
PDO mapping	No

Sub	0x01
Name	CycleCnt
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc01OCPosition.CycleCnt

Sub	0x02
Name	Timestamp
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc01OCPosition.Timestamp

Sub	0x03
Name	SenderId
Data type	UNSIGNED16
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc01OCPosition.SenderId

Sub	0x04
Name	Value
Data type	INTEGER32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc01OCPosition.Value

Sub	0x05
Name	Crc32
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc01OCPosition.Crc32

0x4001 Enc01 OC Speed (Record)

Sub	0x00
Name	Highest sub-index supported
Data type	UNSIGNED8
Access	R/O
Default value	5
PDO mapping	No

Sub	0x01
Name	CycleCnt
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc01OCSpeed.CycleCnt

Sub	0x02
Name	Timestamp
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc01OCSpeed.Timestamp

Sub	0x03
Name	SenderId
Data type	UNSIGNED16
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc01OCSpeed.SenderId

Sub	0x04
Name	Value
Data type	INTEGER32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc01OCSpeed.Value

Sub	0x05
Name	Crc32
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc01OCSpeed.Crc32

0x4002 Enc02 OC Position (Record)

Sub	0x00
Name	Highest sub-index supported
Data type	UNSIGNED8
Access	R/O
Default value	5
PDO mapping	No

Sub	0x01
Name	CycleCnt
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc02BCPosition.CycleCnt

Sub	0x02
Name	Timestamp
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc02BCPosition.Timestamp

Sub	0x03
Name	SenderId
Data type	UNSIGNED16
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc02BCPosition.SenderId

Sub	0x04
Name	Value
Data type	INTEGER32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc02BCPosition.Value

Sub	0x05
Name	Crc32
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc02BCPosition.Crc32

0x4003 Enc02 OC Speed (Record)

Sub	0x00
Name	Highest sub-index supported
Data type	UNSIGNED8
Access	R/O
Default value	5
PDO mapping	No

Sub	0x01
Name	CycleCnt
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc02BCSpeed.CycleCnt

Sub	0x02
Name	Timestamp
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc02BCSpeed.Timestamp

Sub	0x03
Name	SenderId
Data type	UNSIGNED16
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc02BCSpeed.SenderId

Sub	0x04
Name	Value
Data type	INTEGER32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc02BCSpeed.Value

Sub	0x05
Name	Crc32
Data type	UNSIGNED32
Access	R/O
PDO mapping	Optional, TPDO only
Access name	Enc02BCSpeed.Crc32

2.4 Use with CODESYS Safety

2.4.1 Prerequisites

Know how to programme and configure EtherCAT.

2.4.2 Applicability

Software

Note on software releases: Note that safety applications set requirements as to the software releases you use. Refer to the release documentation of the Safety Packages for details.

- CODESYS V3.5 SP16 Patch40 (32-Bit)
 - CODESYS Safety 1.6.1 on V3.5.16.40 for Kendrion.Package
- CODESYS V3.5 SP17 Patch40 (32-Bit)
 - CODESYS_3.5SP17Patch40Hotfix1.Package
 - CODESYS Safety V1.6.1 on V3.5.17.41 for Kendrion.ackage
- CODESYS Bibliothek KICS_Safety_Library

Hardware

- CODESYS control unit with an EtherCAT master of a CODESYS version matching that of the software
- Kuhnke FIO Bus Coupler
- Kuhnke FIO Safety PLC
- Kuhnke FIO OC

Virtual Devices

- External Communication Monitoring V1.0.0.0

2.4.3 Other Applicable Documents

CODESYS Safety User Guide

Instruction Manual: Kuhnke FIO Safety PLC

System Manual: Kuhnke FIO I/O System

Product Manual: Kuhnke FIO System Coupler

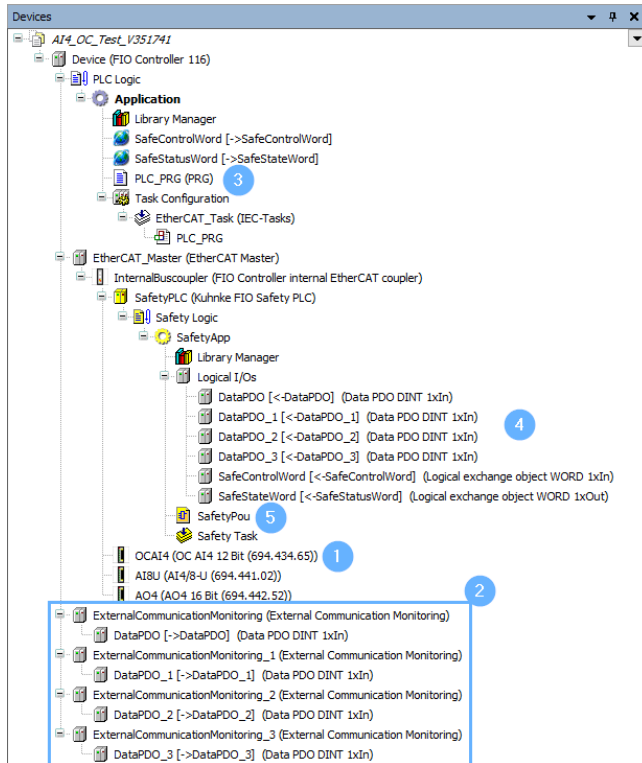
Product Manual: Kuhnke FIO Analogue I/O Modules

Product Manual: Kuhnke FIO Counter Modules

2.4.4 Example Application

The example application below uses a Kuhnke FIO OC AI4. In this case, all analogue input channels are transferred to the safety application and monitored there.

Overview



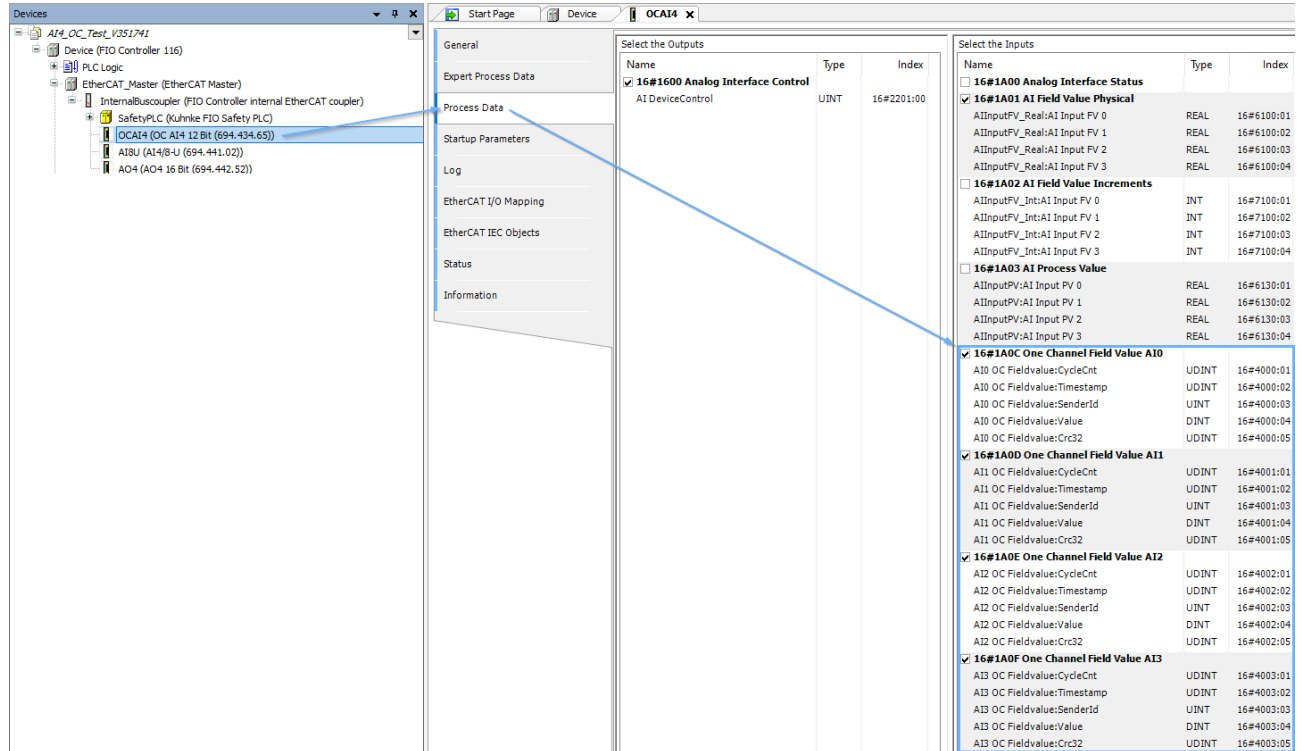
Working steps

- (1) Kuhnke FIO OC AI4 12-Bit
Analogue input module based on Kendrion Kuhnke's OC technology
- (2) External Communication Monitoring
These modules hold the safeguarded OC data containers for logic data interchange with the safety application
- (3) PLC_PRG
PLC_PRG converts the OC variables into the variables of module External Communication Monitoring.
- (4) Data PDO DINT
Logic data interchange unit containing the OC data containers inside the safety application.
- (5) SafetyPOU
SafetyPOU checks the data received from logic data interchange. Every OC data container needs a separate function block instance of type SF01_ECM.

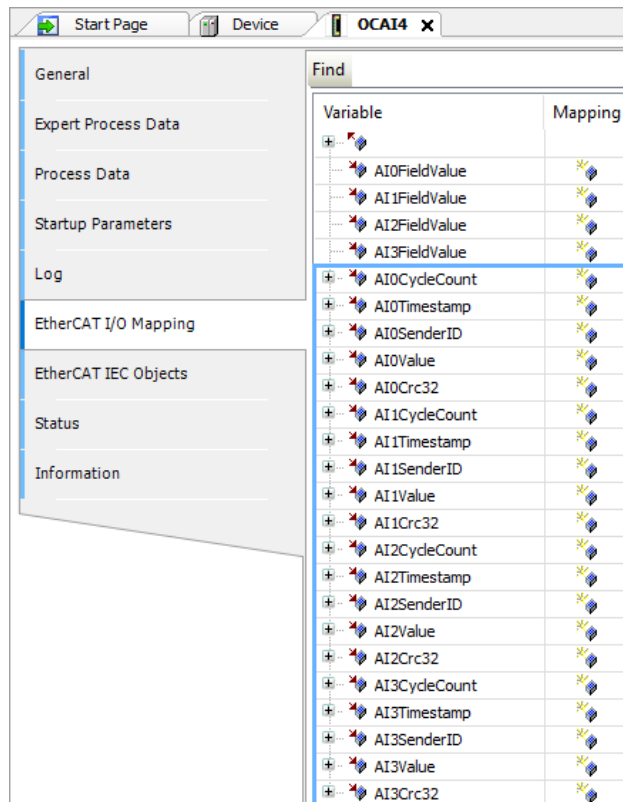
Configuring the Kuhnke FIO OC Module

This example uses module Kuhnke FIO OC AI4 12-Bit. Refer to product manual Kuhnke FIO Analogue I/O Modules for the standard analogue input configuration.

To make actual use of the OC technology, display the configuration of module Kuhnke FIO OC AI4, go to tab Process Data and enable the OC data containers you need.

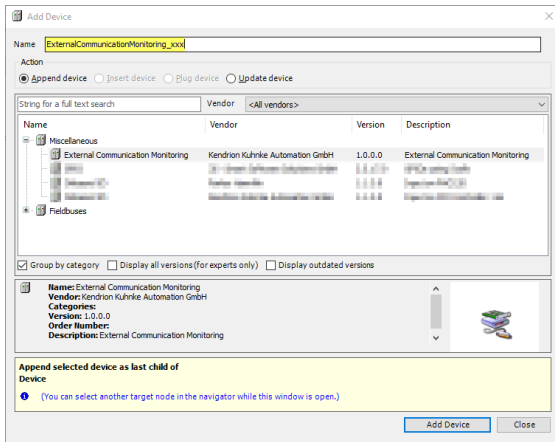


Now go to tab EtherCAT I/O Mapping and assign symbolic names to the process data, for example.



Appending the External Communication Monitoring Module(s)

You need a module of type External Communication Monitoring for every OC data container in your application. To add the modules to the Devices Explorer, right-click on the control unit and pick Add Device from the right-click menu.



Assign any name to the device and click on Add Device. Repeat the process until every OC data container has an External Communication Monitoring module to it.

Then display the associated configuration. Go to tab DataPDO I/O Mapping and assign a symbolic name to each of your OC data containers.

Sample symbolic name of the first OC data container:

Variable	Mapping	Channel	Address	Type
OC0CycleCount		CycleCount	%QD27	UDINT
OC0Timestamp		Timestamp	%QD28	TIME
OC0SenderID		SenderID	%QW58	UINT
OC0Value		Value	%QD30	DINT
OC0Crc32		CRC32	%QD31	UDINT

Copying the Data for the OC Data Containers

A standard POU (in PLC_PRG in this case) is used to copy the data. Using the previously assigned symbolic names, the resulting ST programme code looks as follows:

```
OC0CycleCount := AI0CycleCount;
OC0Timestamp := UDINT_TO_TIME(AI0Timestamp);
OC0SenderID := AI0SenderID;
OC0Value := AI0Value;
OC0Crc32 := AI0Crc32;

OC1CycleCount := AI1CycleCount;
OC1Timestamp := UDINT_TO_TIME(AI1Timestamp);
OC1SenderID := AI1SenderID;
OC1Value := AI1Value;
OC1Crc32 := AI1Crc32;

OC2CycleCount := AI2CycleCount;
OC2Timestamp := UDINT_TO_TIME(AI2Timestamp);
OC2SenderID := AI2SenderID;
OC2Value := AI2Value;
OC2Crc32 := AI2Crc32;

OC3CycleCount := AI3CycleCount;
OC3Timestamp := UDINT_TO_TIME(AI3Timestamp);
OC3SenderID := AI3SenderID;
OC3Value := AI3Value;
OC3Crc32 := AI3Crc32;
```

Logic Data Interchange Devices

Appending an External Communication Monitoring module already added the associated logic data interchange devices of type DataPDO DINT 1x below the Logic Devices level within the safety application. There is no more configuration than that.

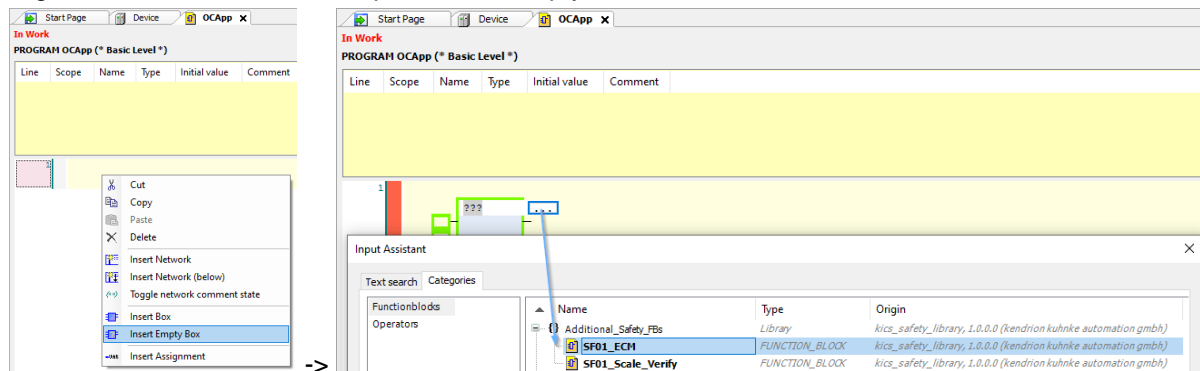
Using the OC Data Containers in the Safety Application

Before the safety application can make use of the data, it must first analyse the OC data containers. This is done by function block SF01_ECM, ECM being short for External Communication Monitoring. External communication monitoring targets a CRC32-verified data element consisting of a cycle counter, a timestamp, a sender ID and a value needed to check whether cyclic data element reception occurs within the set monitoring time.

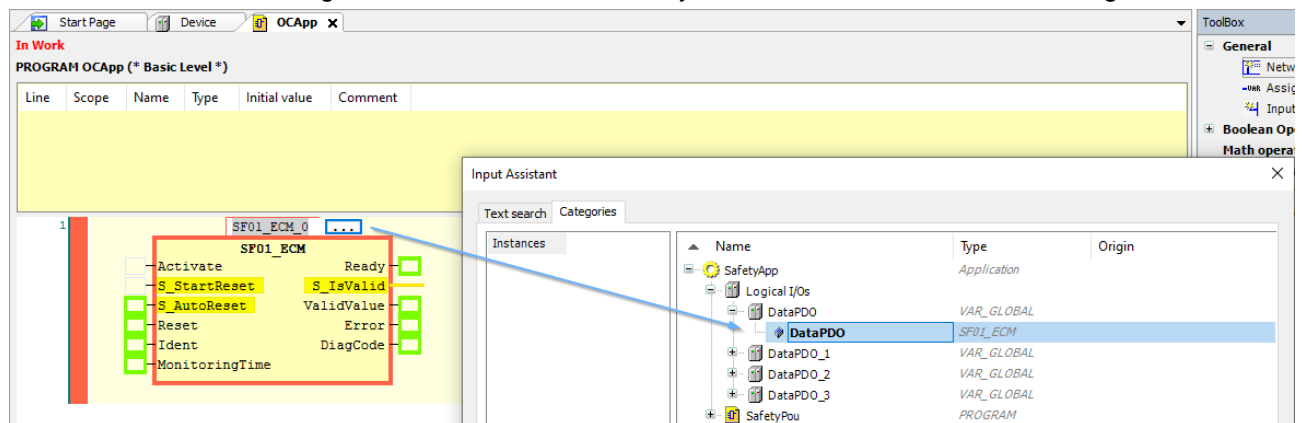
Run an existing program module in your safety application or create a new program module.

Add a new network to that module, as necessary.

Right-click on the network and pick "Insert empty box" from the menu



Now reference the resulting function block instance to any of the OC data containers in the logic I/Os.



To actually enable the function block, check that the following inputs are set:

- Activate: Can be set to a constant TRUE
- Reset: Link this input to any signal used for error acknowledgment. Our example also uses a signal from logic data interchange between the standard application and the safety application. This allows you to acknowledge safety application errors by means of a button on the machine visualisation display screen, for example.
- Ident: Enter the SenderID of the channel and be sure to use the same ID as in the OC data container. First of all, go to the declaration section of the POU, declare a constant of type WORD and enter the SenderID as the initial value. Then change the name of column Scope VAR to VAR_CONSTANT.

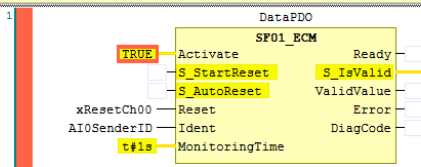
Line	Scope	Name	Type	Initial value	Comment
1	VAR_EXTERNAL	DataPDO	SF01_ECM		
2	VAR	AI0SenderID	WORD	10031	

Scope:
VAR
VAR_CONSTANT
VAR_EXTERNAL
VAR_EXTERNAL CONSTANT

- MonitoringTime: Set the monitoring time, as appropriate. The function block will return an error if the OC data container does not change within the set monitoring time.

Example:

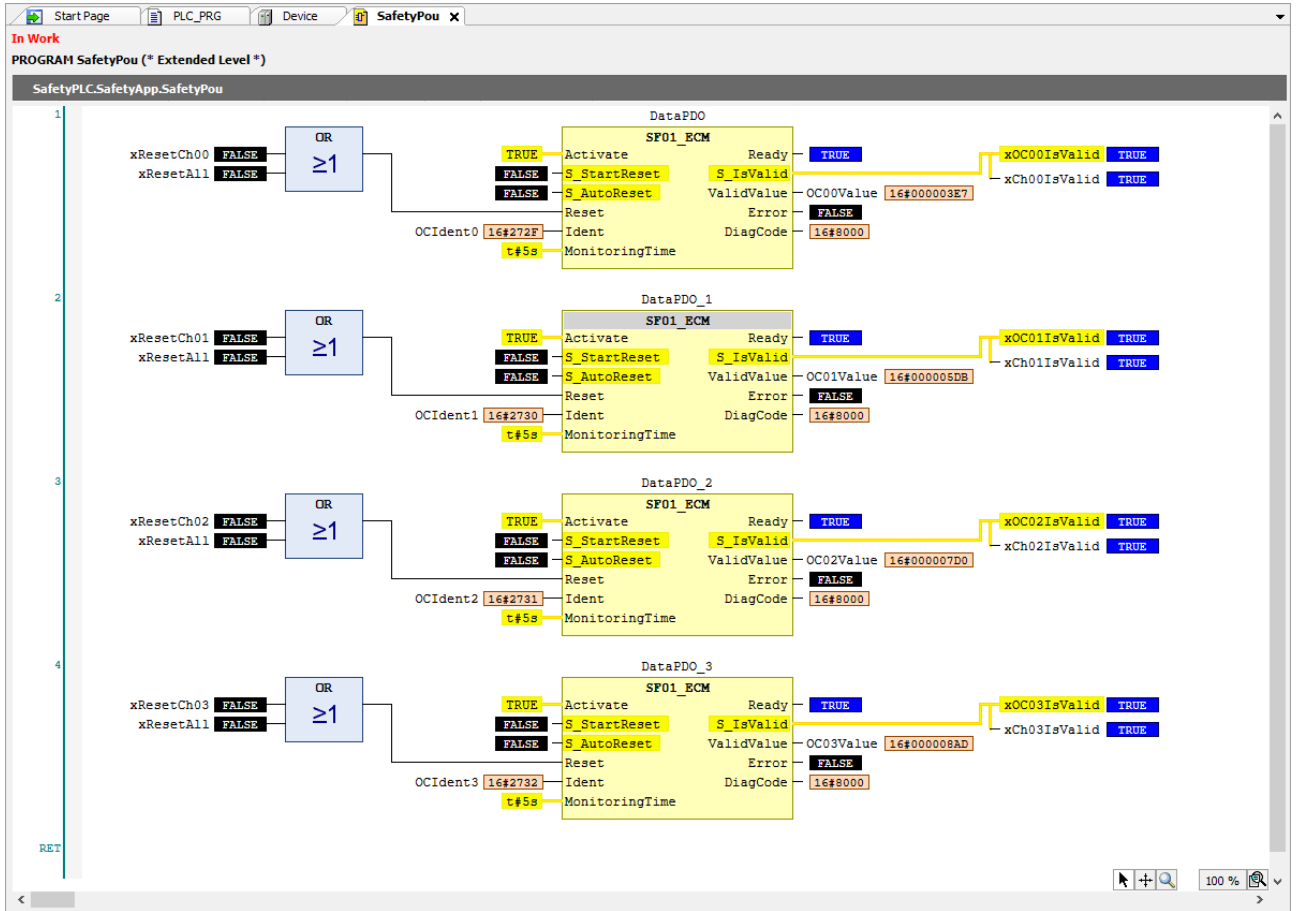
Line	Scope	Name	Type	Initial value
1	VAR_EXTERNAL	DataPDO	SF01_ECM	
2	VAR_CONSTANT	AI0SenderID	WORD	10031
3	VAR_EXTERNAL	xResetCh00	BOOL	



Your application defines how the outputs are used. In our example, status output S_IsValid was expanded by a variable called xOC00IsValid (type: SAFEBOOL) as well as variable xCH00IsValid (type: BOOL) needed for data exchange with the PLC.

Repeat the above steps to add further OC data containers to your safety application.

Considering all 4 of the analogue inputs, your application might look like this:



3 Appendix

3.1 Order Data

3.1.1 Modules

Kuhnke FIO OC AI4 12-Bit	694.434.65 / 200704
Kuhnke FIO OC Counter/Encoder.....	694.434.53 / 197674

3.1.2 Accessories

Kuhnke FIO Shield Terminal 2x8mm	694 412 03 / 196445
Kuhnke FIO Shield Terminal 1x14mm	694 412 04 / 196446
Kuhnke FIO Shield Terminal 4x8mm	694 412 05 / 196448
Kuhnke FIO Shield Terminal 2x14mm	694 412 06 / 197524

Kendrion Kuhnke Automation GmbH
Industrial Control Systems

Lütjenburger Str. 101
D-23714 Malente

Tel.: +49 4523 402 0
Fax: +49 4523 402 201