Network-specific properties of the Axioline E EtherCAT® IO-Link devices

User manual
UM EN AXL E EC IOL

PHOENIX CONTACT
INSPIRING INNOVATIONS

User manual

Network-specific properties of the Axioline E EtherCAT® IO-Link devices

UM EN AXL E EC IOL, Revision 02 2019-08-23

This user manual is valid for the following Axioline E EtherCAT® IO-Link devices:

Description	Order No.
AXL E EC IOL8 DI4 M12 6M	2701531
AXL E EC IOL8 DI4 M12 6P	2701524

107945_en_02

PHOENIX CONTACT GmbH & Co. KG • Flachsmarktstraße 8 • 32825 Blomberg • Germany
phoenixcontact.com

Table of contents

1 For your safety

Read this user manual carefully and keep it for future reference.

1.1 Labeling of warning notes

This symbol indicates hazards that could lead to personal injury.
There are three signal words indicating the severity of a potential injury.

DANGER
Indicates a hazard with a high risk level. If this hazardous situation is not avoided, it will result in death or serious injury.

WARNING
Indicates a hazard with a medium risk level. If this hazardous situation is not avoided, it could result in death or serious injury.

CAUTION
Indicates a hazard with a low risk level. If this hazardous situation is not avoided, it could result in minor or moderate injury.

This symbol together with the **NOTE** signal word alerts the reader to a situation which may cause damage or malfunction to the device, hardware/software, or surrounding property.

Here you will find additional information or detailed sources of information.

1.2 Qualification of users

The use of products described in this user manual is oriented exclusively to:

- Qualified electricians or persons instructed by them. The users must be familiar with the relevant safety concepts of automation technology as well as applicable standards and other regulations.
- Qualified application programmers and software engineers. The users must be familiar with the relevant safety concepts of automation technology as well as applicable standards and other regulations.

1.3 Product changes

Changes or modifications to hardware and software of the device are not permitted.

Incorrect operation or modifications to the device can endanger your safety or damage the device. Do not repair the device yourself. If the device is defective, please contact Phoenix Contact.

1.4 Security in the network

NOTE: : Risk of unauthorized network access

Connecting devices to a network via Ethernet always entails the risk of unauthorized access to the network.

Therefore, please check your application for any option of deactivating active communication channels . Setting passwords to prevent third parties from accessing the controller without authorization and modifying the system.

Because of the controller's communication interfaces, we advise against using the controller in safety-critical applications without additional security appliances.
Please take additional protective measures according to the IT security requirements and the standards applicable to your application (for instance virtual networks (VPN) for remote maintenance access, firewalls, etc.) for protection against unauthorized network access.

On first request, you shall release Phoenix Contact and the companies associated with Phoenix Contact GmbH & Co. KG, Flachsmarktstraße 8, 32825 Blomberg in accordance with §§ 15 ff. AktG or German Stock Corporation Act (hereinafter collectively referred to as "Phoenix Contact") from all third-party claims that are made due to improper use.

For the protection of networks for remote maintenance via VPN, Phoenix Contact offers the mGuard product series security appliances which you can find described in the latest Phoenix Contact catalog (phoenixcontact.net/products).

Additional measures for protection from unauthorized network access can be found in the AH EN INDUSTRIAL SECURITY application note. The application note can be downloaded at phoenixcontact.net/products.

2 EtherCAT®

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

The CAN application layer over EtherCAT® (CoE) mailbox protocol is the basis for the Device Profile. It enables the parameterization of EtherCAT® devices via the object dictionary. The object dictionary is accessed via CoE using Service Data Object (SDO) services. The objects implemented on the device are described in the “EtherCAT® object dictionary” section.

3 EtherCAT®: Modular Device Profile (MDP)

The Axioline E EtherCAT® IO-Link device operates on the basis of the Modular Device Profile (ETG.5001) and is identified as profile implementation “5001” (Modular Device Profile).

Furthermore, the device essentially conforms to specification ETG.5001.3 Annex K (IO-Link master) and corresponds to “Modular Device Profile 6220” from a functional point of view. However, in contrast to ETG.5001.3 Annex K, which prescribes the synchronization of the status machine of the EtherCAT® slave with the status machine of the IO-Link master, conflicts between the states of both status machines are intercepted with Axioline E's own failsafe mechanism (EtherCAT® slave is switched to PRE-OP).

The CAN application layer over EtherCAT® (CoE) mailbox protocol is the basis for the Modular Device Profile (MDP) and enables the parameterization of EtherCAT® devices via the object dictionary.

The object dictionary is accessed via CoE using Service Data Object (SDO) services. The objects implemented on the device are described in the “EtherCAT® object dictionary” section.

The Axioline E EtherCAT® IO-Link master has the following structure:

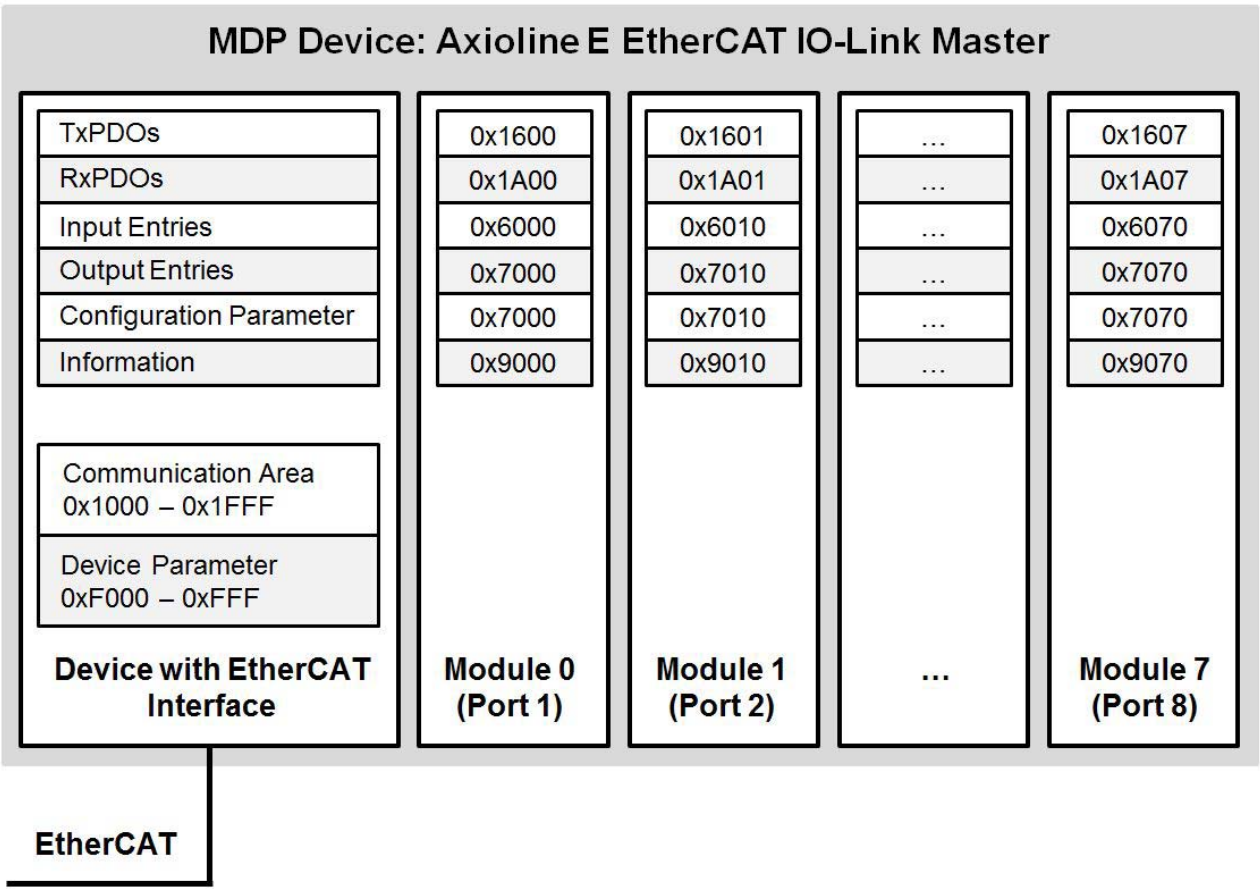


Figure 1: Structure of the Axioline E EtherCAT® IO-Link master

4 EtherCAT® communication methods

The EtherCAT® protocol offers two communication methods:

– Mailbox method
– Buffered method

The mailbox method is used for the acyclic transmission of commands to slaves. If a slave receives a mailbox message, this must be processed first before anything else can be processed.

In contrast, the buffered method enables the master and slave to use a shared data area. The master and slave can read or write data in this area (buffer) at any time.

The Sync Manager manages the data exchange for both methods to ensure that there are no data collisions.
A detailed description of all Sync Manager registers can be found in the “EtherCAT Slave Controller” description from the EtherCAT® user organization (www.EtherCAT.org).

5 EtherCAT® synchronization

There are two modes for synchronizing the application which can be selected in the engineering system.

– FreeRun (process data update by internal device application cycle)
– SM Synchronous (process data update when an SM event occurs)

5.1 FreeRun

In this mode, the EtherCAT® communication system and the I/Os operate asynchronously. The I/Os are in Auto-Run mode and run with the minimum possible cycle time for the current module configuration. This mode is set by default in the device.

5.2 SM Synchronous

In this mode, the EtherCAT® communication system and the I/Os operate synchronously. The I/Os are exchanged between the EtherCAT® bus and the device on every Sync Manager Event (bus cycle). It is possible to synchronize on SM2 (Sync Manager for data from the master to the slave) or on SM3 (Sync Manager for data from the slave to the master). If the device is configured for input and output data, it makes sense to use SM2. If only input data (data from the slave to the master) is configured, synchronization can also be carried out on SM3. The settings must be written to subobject 1 of objects $1C12_{hex}$ (SM2) and $1C13_{hex}$ (SM3) in the PRE-Operational state.

The recommended setting is $1C12_{hex}$: $1 = 01_{hex}$ and $1C13_{hex}$: $1 = 22_{hex}$.
For detailed information about this, please refer to specification ETG1020.

So as not to have to repeat the settings in the object dictionary every time, the settings can be stored in the startup parameters.

6 EtherCAT® object dictionary (CoE objects)

The device object dictionary contains objects which can be addressed via SDO services. The device supports standard objects and manufacturer-specific objects. The standard objects are described in ETG standard ETG.1000.6 (Application Layer Protocol Specification). Modular Device Profile-specific objects, which are described in ETG standard ETG.5001.3, are supported. Manufacturer-specific objects, which are described in detail below, are also supported.

The objects are addressed using a combination of index and subindex. Subindex 0 indicates the number of subindices or the number of the highest subindex.

The following applies for the tables below:

Length = Length of the elements in bytes
Rights = Access rights
RO = Read only
R/W = Read and write (Read/Write)

Index (hex)	Object name	Data type	Rights
Device Identity Objects (1)			
1000	Device Type	UINT32	RO
1008	Manufacturer Device Name	STRING	RO
1009	Manufacturer Hardware Version	STRING	RO
100A	Manufacturer Software Version	STRING	RO
1018	Identity	IDENTITY	RO
Diagnosis Objects			
10F3	Diagnosis History	RECORD	RO
10F8	Timestamp Object	UINT64	RO
PDO Mapping Objects			
1600 ... 1607	RxPDO Mapping IO-L Outputs Port 1…8	PDO Mapping	R/W
1608	RxPDO Mapping IOLM_Control	PDO Mapping	RO
1A01 ... 1A07	TxPDO Mapping IO-L Inputs Port 1...8	PDO Mapping	RW
1A08	New Message Available	PDO Mapping	RO
1A09	Timestamp	PDO Mapping	RO
1A81	TxPDO Mapping IOLS_Status	PDO Mapping	RO
1A82	TxPDO Mapping IOLM_Status	PDO Mapping	RO
1B02	TxPDO Alignment	PDO Mapping	R/W
Sync Manager Objects			
1C00	Sync Manager Communication Type	UINT8	RO
1C10	Sync Manager 0 PDO Assignment (Mail OUT control register)	UINT16	RO
1C11	Sync Manager 1 PDO Assignment (Mail IN control register)	UINT16	RO
1C12	Sync Manager 2 PDO Assignment (process data output control register)	UINT16	RW
1C13	Sync Manager 3 PDO Assignment (process data input control register)	UINT16	R/W
1C32	Sync Parameter of SM2	SYNC_PAR	R/W
1C33	Sync Parameter of SM3	SYNC_PAR	R/W

Index (hex)	Object name	Data type	Rights
Device Identity Objects (2)			
2001	Component Name	STRING	RO
2002	Vendor Name	STRING	RO
2003	Vendor URL	STRING	RO
2004	Order Number	STRING	RO
2005	Manufacturing Date	STRING	RO
2006	QS Date	STRING	RO
2007	Installation Location	STRING	R/W
2008	Operational Hours	UINT32	RO
2009	Service Date	STRING	RO
200A	Equipment Ident	STRING	R/W
Safe State Objects			
2100 ... 2170	Safe State Mode IOL-Port 1 ... 8	UINT8	R/W
2180	Safe State Mode IOLM_Control (DO)	UINT8	R/W
2182	Safe State Values IOLM_Control (DO)	UINT8	R/W
IO-Link Port Parameter Objects			
2F00	IO-Link Parameter Port 1…8	RECORD	RO
Reset Object			
2F00	Reset to Factory Settings	UINT8	R/W
IO-Link Master Status & Control Objects			
3000	IOLM_Control	UINT8	RW
3001	IOLM_Status	UINT8	RO
IO-Link Process Data Objects			
6000 ... 6070	IO-Link Inputs Port 1…8	STRING	RO
7000 ... 7070	IO-Link Outputs Port 1…8	STRING	RO
IO-Link Configuration Objects (Expected Configuration of the IO-Link Devices)			
8000 ... 8070	IO-Link Configuration Data Port 1…8	RECORD	RO
IO-Link Information Objects (Detected Configuration of the IO-Link Devices)			
9000 ... 9070	IO-Link Information Data Port 1…8	RECORD	RO
IO-Link Device Objects			
F000	Modular Device Profile	RECORD	RO
F020	Module Address List	UINT8	RO
F030	Configured Module Ident List	UINT32	R/W
F050	Detected Module Ident List	UINT32	RO
F100	IO-Link Port Status	UINT8	RO

7 CoE: device rating plate

The device has objects for identification. They contain information about the manufacturer and device and make up the device rating plate.

The objects that make up the device rating plate are described below.

Index (hex)	Subindex	Object name	Data type	Rights	Meaning/value
1000	-	Device Type	UINT32	RO	5001 (Modular Device Profile)
1008	-	Manufacturer Device Name	STRING	RO	Product name
1009	-	Manufacturer Hard-ware Version	STRING	RO	Hardware version, version ID
100A	-	Manufacturer Software Version	STRING	RO	Firmware version, version ID
1018	Identity				
	00	Number of Entries	UINT8	RO	Number of object subindices
	01	Vendor ID	UINT32	RO	00000084_{hex} (132)
	02	Product Code	UINT32	RO	Order No.
	03	Revision Number	UINT32	RO	Device revision, version ID
	04	Serial Number	UINT32	RO	Serial number
2001	-	Component Name	STRING	RO	EtherCAT® IO-Link gateway
2002	-	Vendor Name	STRING	RO	Phoenix Contact
2003	-	Vendor URL	STRING	RO	www.phoenixcontact.com
2004	-	Order Number	STRING	RO	Order No.
2005	-	Manufacturing Date	STRING	RO	YYYY-MM-DD T HH:MM:SS Z (date and time of manufacture)
2006	-	QS Date	STRING	RO	YYYY-MM-DD T HH:MM:SS Z (date and time of final manufacturing test)
2007	-	Installation Location	STRING	R/W	User-defined device installation location
2008	-	Operational Hours	UINT32	RO	Operating hours counter
2009	-	Service Date	STRING	RO	YYYY-MM-DD T HH:MM:SS Z (date and time of a service)
200A	-	Equipment Ident	STRING	R/W	User-defined device name

8 CoE: IO-Link port configuration

The Axioline E EtherCAT® IO-Link master is based on the Modular Device Profile (ETG.5001). A distinction is made between modules and slots. A module corresponds to the configuration of an IO-Link device with fixed process data lengths. The term module is used as a synonym for an IO-Link device.

A slot is an IO-Link port of the IO-Link master. Consequently, a module is therefore inserted into a slot.

Just one module and therefore only one Rx/TxPDO can be assigned to each port of the Axioline E EtherCAT® IO-Link master. Only one data object (6000_{hex}, 7000_{hex}) can be assigned to an Rx/TxPDO. The data object consists of individual subobjects.

The first subobject of every object (subindex 01) always contains the size and number of subobjects present. Each object can have a maximum of 32 subindices and can therefore be a maximum of 32 bytes in size. This corresponds to the maximum IO-Link process data length.

If the exact IO-Link process data length of the connected device is not available in the modules of the device description file, select the next largest constellation.

8.1 Configuration of the operating mode

The “PHOENIX_CONTACT_AXL_E_EC_IOL8_Modules-XXX” ESI file contains all modules supported by the device. A module is assigned to a slot using a configuration tool (e.g., TwinCAT®). If a module is added to a slot here, all associated startup commands (startup parameters) are generated automatically by the configuration tool and the necessary entries are created in the PDO list. The following values are available as operating modes:

Hex	Description
00	Deactivated
01	Digital_IN (function as digital input)
02	Digital_OUT (function as digital output)
03	IO-Link
04	DI with IO-Link (no cyclic IO-Link communication, also referred to as SIO mode)

Depending on the slot configuration, these values are also entered in object 0x80n0:28 (Master Control) as a startup command.

A possible port configuration in the TwinCAT® engineering tool could look like this:

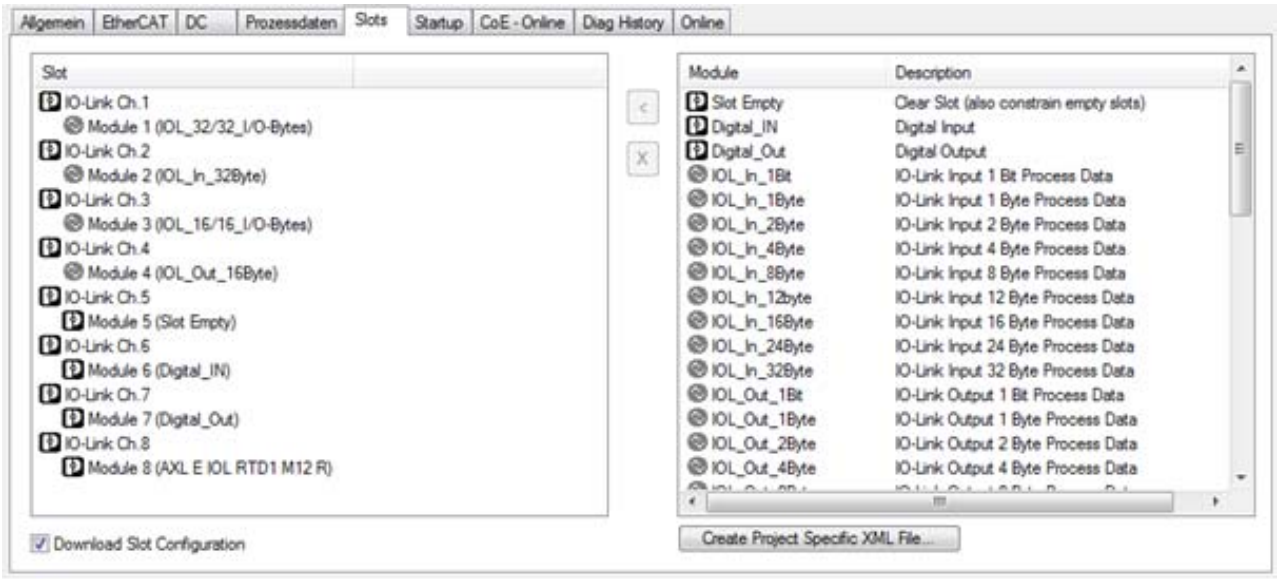


Figure 2: Possible port configuration in the TwinCAT® engineering tool

For further information about the structure of the $80n0_{hex}$ objects, please refer to the “IO-Link device configuration (set)” section.

8.2 Configuration of port parameters

The objects below describe basic startup parameters, such as the “data storage mechanism” of IO-Link devices. The parameters can be set for every port.

Index (hex)	Sub-index (hex)	Object name	Data type	Rights	Meaning/value	
2800	**IO-Link Parameter Port 1**					
	00	Number of entries	UINT8	RO	Number of object subindices	
	01	Data Storage	UINT8	R/W	The “data storage mechanism” enables parameters to be exchanged between the master and device. Connected devices must support at least IO-Link specification v1.1 in order to use the function.	
					Hex	**Description**
					00	**Deactivated (default)**
					01	Download only The parameter data is sent from the IO-Link master to the device. In the event of an inconsistency between the parameter data of the IO-Link device and IO-Link master, the data from the IO-Link master is taken as the default. As such it is possible to swap the IO-Link device. Upload is not possible in this mode.
					02	Upload only The parameter data is sent from the IO-Link device to the IO-Link master. In the event of an inconsistency between the parameter data of the IO-Link device and the IO-Link master, the data from the device is taken as the default. As such it is possible to swap the IO-Link master. Download is not possible in this mode.
					03	Download and upload The parameter data is stored in the IO-Link master as well as the device. In the event of an inconsistency between the parameter data of the IO-Link device and the IO-Link master, the data from the IO-Link master is taken as the default. As such it is possible to swap the IO-Link device or the IO-Link master.
					04	Deactivated and deleted The data storage mechanism is deactivated and the master deletes all stored parameters for the respective port.

Index (hex)	Sub-index (hex)	Object name	Data type	Rights	Meaning/value	
	02	Device Parameter	ARRAY [0…15] OF BYTE	R/W	Direct Parameter Page 2 (DPP2) Device Parameter Page 2 describes the area between IO-Link objects 10_{hex} … $1F_{hex}$. This is an optional manufacturer-specific area of the IO-Link device data.	
					Element	**Description**
					[0]	DPP2 object 10_{hex}
					[15]	DPP2 object $1F_{hex}$
	03	Analog Converter Parameter	UINT16	R/W	Special object for Phoenix Contact Axioline E IO-Link/analog converters. The settings in this object describe ISDU object 80_{hex} of the converter. For the exact structure of the object, please refer to the corresponding device data sheet.	
2810	**IO-Link Parameter Port 2**					
	00	Number of entries	UINT8	RO	Number of object subindices	
...	...	...	...	...	...	
2870	**IO-Link Parameter Port 8**					
...	...	...	...	...	...	

The respective IO-Link port must be in IO-Link mode (corresponds to modules: IOL_... or AXL E IOL …) so that the settings in these objects take effect.

8.3 Port status

Object $F100_{hex}$ (IO-Link Port Status) contains status data for every port, represented by a subindex. The IO-Link status (e.g., port inactive, SIO mode Digital_In, etc.) and an error code (if one exists) are stored here. The object can be read cyclically or even mapped to the process data as an option.

The object has the following structure:

Index (hex)	Subindex (hex)	Object name	Data type	Rights	Meaning/value	
F100	**IO-Link Port Status**					
	00	Number of entries	UINT8	RO	Number of object subindices	
	01	IO-Link Port 1 Status	UINT8	R/W	Status byte of the first IO-Link port	
					Bit	**Meaning**
					0 ... 3	IO-Link state 0 = Port inactive 1 = Port operating as Digital_In 2 = Port operating as Digital_Out 3 = Communication in OP state 4 = Communication in STOP state
					4 ... 7	ErrorCode 0 = No error 3 = Invalid Device ID 10 = No device detected
	02	IO-Link Port 2 Status	UINT8	R/W	Status byte of the second IO-Link port	
	...	...	...	...	...	
	08	IO-Link Port 8 Status	UINT8	R/W	Status byte of the eighth IO-Link port	

8.4 CoE: IO-Link modul identification

Module identification is used to compare the configured modules with the modules that are actually connected.

The Axioline E EtherCAT® IO-Link master performs this comparison by checking the Vendor ID and Device ID. The configuration data for the Vendor ID and Device ID is located in objects 8000_{hex} to 8070_{hex} and is written when the engineering tool is started up. The Axioline E EtherCAT® IO-Link master provides the information about the connected modules in objects 9000_{hex} to 9070_{hex}.

A check is then only performed if subobjects 0x80n0:04 (Device ID) and 0x80n0:05 (Vendor ID) have been previously configured to a value other than zero for the respective port (n).

If no check is performed because no Device ID or Vendor ID has been entered, and the configured data lengths of a module do not match, only the amount of data that was previously configured is transmitted.

Example:

A real IO-Link device with 16 bytes of IN process data is connected. However, only eight bytes have been configured, so only eight bytes will be transmitted.

The Axioline E EtherCAT® IO-Link master is identified as MDP profile implementation “5001”. Objects $F030_{hex}$ (Configured Module Ident List) and $F050_{hex}$ (Detected Module Ident List), which are used by an MDP device for module identification, must therefore be available.

Since the Axioline E EtherCAT® IO-Link master operates in accordance with IO-Link master profile implementation “6220”, these objects are not evaluated by the device because identification using the Vendor ID and Device ID is sufficient. The module numbers are therefore not compared.

The entries of object $F030_{hex}$ correspond to the entries in subindices 0x8nn0:0A of the configured IO-Link devices, which (having been read from the ESI file) are written to subobject 8xx0:0A during startup.

The entries of object $F050_{hex}$ correspond to the entries in subindices 0x9nn0:0A of the detected IO-Link devices, both are filled with zeros.

8.5 IO-Link device configuration (set)

Objects 8000_{hex} to 8070_{hex} contain the device-specific configuration data manually specified for each IO-Link port, such as the Device ID, Vendor ID, process data length, etc.

During startup, state change from PREOP to SAFEOP, the Axioline E EtherCAT® IO-Link master writes all configuration data to the Axioline E EtherCAT® IO-Link master.

Index (hex)	Sub-index (hex)	Object name	Data type	Rights	Meaning/value	
8000	**IO-Link Configuration Data Port 1**					
	00	Number of entries	UINT8	RO	Number of object subindices	
	04	Device ID	UINT32	R/W	Device ID of the IO-Link device A check is only performed if the value is != 0.	
	05	Vendor ID	UINT32	R/W	Vendor ID of the IO-Link device A check is only performed if the value is != 0.	
	0A	Module Ident	UINT32	R/W	Module identification number Refer to the ESI files for known numbers. The device does not evaluate this entry.	
	24	Process Data IN Length	UINT8	R/W	Amount and structure of IN process data for the device	
					Bit	**Meaning**
					0 ... 4	Process data length
					5	Reserved
					6	SIO indicator (device supports Standard IO mode)
					7	Byte indicator (length interpreted as byte + 1)
	25	Process Data OUT Length	UINT8	R/W	Amount and structure of OUT process data for the device	
					Bit	**Meaning**
					0 ... 4	Process data length
					5	Reserved
					6	SIO indicator (device supports Standard IO mode)
					7	Byte indicator (length interpreted as byte + 1)
	28	Master Control	UINT16	R/W	Operating mode of the IO-Link port	
					Hex	**Meaning**
					00	Deactivated
					01	Digital_In (function as digital input)
					02	Digital_Out (function as digital output)
					03	IO-Link
					04	DI with IO-Link (no cyclic IOL communication, also referred to as SIO mode)
8010	**IO-Link Configuration Data Port 2**					
...	00	Number of entries	UINT8	RO	Number of object subindices	
	04	Device ID	UINT32	R/W	Device ID of the IO-Link device	
...	...	...	...	...	...	
8070	**IO-Link Configuration Data Port 8**					
...	...	...	...	...	...	

8.6 IO-Link device information (Detected)

Objects 9000_{hex} to 9070_{hex} contain the device-specific data read for each IO-Link port, such as the Device ID, Vendor ID, process data length, etc.

The Axioline E EtherCAT® IO-Link master reads all this data required for startup of the connected device and transmits it to the intended objects.

Index (hex)	Sub-index (hex)	Object name	Data type	Rights	Meaning/value
9000	**IO-Link Information Data Port 1**				
	00	Number of entries	UINT8	RO	Number of object subindices
	01	Fixed Station Address	UINT16	RO	First IO-Link port
	05	Vendor ID	UINT32	RO	Vendor ID indicated by the device Structure corresponds to 0x8000:05
	06	Product Code	UINT32	RO	Product code indicated by the device
	07	Revision Number	UINT32	RO	Revision number indicated by the device
	08	Serial Number	UINT32	RO	Serial number indicated by the device
	0A	Module Ident	UINT32	R/W	Structure corresponds to 0x8000:10
	20	IO-Link-Revision	UINT8	RO	IO-Link revision indicated by the device
					Bit / **Meaning**
					0 ... 3 / Minor revision
					4 ... 7 / Major revision
	22	Cycle Time (ms)	UINT8	RO	Cycle time in milliseconds indicated by the device
					Bit / **Meaning**
					0 ... 5 / Multiplier
					6 ... 7 / Time base
	24	Process Data IN Length	UINT8	RO	IN process data length indicated by the device Structure corresponds to 0x8000:36
	25	Process Data OUT Length	UINT8	R/W	OUT process data length indicated by the device Structure corresponds to 0x8000:37
	28	Master Control	UINT16	R/W	Operating mode of the IO-Link port
					Hex / **Meaning**
					00 / Deactivated
					01 / Digital_In (function as digital input)
					02 / Digital_Out (function as digital output)
					03 / IO-Link
					04 / DI with IO-Link (no cyclic IOL communication, also referred to as SIO mode)
9010	**IO-Link Information Data Port 2**				
...	00	Number of entries	UINT8	RO	Number of object subindices
	01	Fixed Station Address	UINT16	RO	Second IO-Link port
...	...	...	...	...	...
9070	**IO-Link Information Data Port 8**				
...	...	...	...	...	...

9 CoE: IO-Link process data

9.1 PDO mapping objects

An RxPDO and a TxPDO are each assigned to a port of an Axioline E EtherCAT® IO-Link master. The indices consist of the following:

– For RxPDO: Index = 1600_{hex} + (IOL port number - 1)
– For TxPDO: Index = $1A00_{hex}$ + (IOL port number - 1)

Therefore, for example, index 1600_{hex} contains the output data of the module in port 1, index 1601_{hex} contains the output data of the module in port 2, etc.

The RxPDO and TxPDO reference all entries in the corresponding 0x6000 and 0x7000 objects of the device respectively. Which objects are included depends on the module configuration performed for the respective port (slot).

The RxPDO with the number 1608_{hex} contains the object with the control data for the IO-Link master (3000_{hex}), the TxPDO with the number $1A82_{hex}$ contains the object with the status data for the IO-Link master (3001_{hex}).

If no module is assigned to a port, the corresponding Rx/TxPDO and objects do not exist.
Unused ports should always be occupied with the “Empty Slot” placeholder module, as otherwise display errors and possibly even configuration errors can occur.

Index (hex)	Subindex (hex)	Object name	Data type	Rights	Meaning/value
1600	**RxPDO Mapping IO-L Outputs Port 1**				
	00	Number of entries	UINT8	RO	Number of object subindices
	01	Subindex 01	UINT32	R/W	$7000ppss_{hex}$ – pp: subindex of the RxPDO – ss: size of the RxPDO subindex in bits
	02	Subindex 02	UINT32	R/W	$7000ppss_{hex}$ – pp: subindex of the RxPDO – ss: size of the RxPDO subindex in bits
	...	...	...	...	...
	20	Subindex 32	UINT32	R/W	$7000ppss_{hex}$ – pp: subindex of the RxPDO – ss: size of the RxPDO subindex in bits
1601	**RxPDO Mapping IO-L Outputs Port 2**				
	00	Number of entries	UINT8	RO	Number of object subindices
	01	Subindex 01	UINT32	R/W	$7010ppss_{hex}$ – pp: subindex of the RxPDO – ss: size of the RxPDO subindex in bits
...	...	...	...	...	...

Index (hex)	Subindex (hex)	Object name	Data type	Rights	Meaning/value
1607	**RxPDO Mapping IO-L Output Port 8**				
...	...	...	...	...	...
1608	**RxPDO Mapping IOLM_Control**				
	00	Number of entries	UINT8	RO	Number of object subindices
	01	Subindex 01	UINT32	RO	0x3000:01 (COM control)
	02	Subindex 02	UINT32	RO	0x3000:02 (reserved)
	03	Subindex 03	UINT32	RO	0x3000:03 (Digital Outputs C/Q)
	04	Subindex 04	UINT32	RO	0x3000:04 (reserved)
1A00	**TxPDO Mapping IO-L Inputs Port 1**				
	00	Number of entries	UINT8	RO	Number of object subindices
	01	Subindex 01	UINT32	R/W	$6000ppss_{hex}$ – pp: subindex of the TxPDO – ss: size of the TxPDO subindex in bits, starting with the first RxPDO of this slave
	02	Subindex 02	UINT32	R/W	$6000ppss_{hex}$ – pp: subindex of the TxPDO – ss: size of the TxPDO subindex in bits
	...	...	...	...	...
	20	Subindex 32	UINT32	R/W	$6000ppss_{hex}$ – pp: subindex of the TxPDO – ss: size of the TxPDO subindex in bits
1A01	**TxPDO Mapping IO-L Inputs Port 2**				
	00	Number of entries	UINT8	RO	Number of object subindices
	01	Subindex 01	UINT32	R/W	$6010ppss_{hex}$ – pp: subindex of the TxPDO – ss: size of the TxPDO subindex in bits
...	...	...	...	...	...
1A07	**TxPDO Mapping IO-L Inputs Port 8**				
...	...	...	...	...	...
1A08	**New Message Available**				
	00	Number of entries	UINT8	RO	Number of object subindices
	01	Subindex 01	UINT32	R/W	0x10F3:04, 1
	02	Subindex 02	UINT32	R/W	0x1000:00, 7
1A09	**Timestamp**				
	00	Number of entries	UINT8	RO	Number of object subindices
	01	Subindex 01	UINT32	RO	0x10F8:00, 64

Index (hex)	Subindex (hex)	Object name	Data type	Rights	Meaning/value
1A81	**TxPDO Mapping IOLS_Status**				
	00	Number of entries	UINT8	RO	Number of object subindices
	01	Subindex 01	UINT8	RO	0xF100:01 (state of IO-Link)
	...	...	...	...	...
	01	Subindex 01	UINT8	RO	0xF100:08 (state of IO-Link)
1A82	**TxPDO Mapping IOLM_Status**				
	00	Number of entries	UINT8	RO	Number of object subindices
	01	Subindex 01	UINT32	RO	0x3001:01 (COM states)
	02	Subindex 02	UINT32	RO	0x3002:02 (PD valid states)
	03	Subindex 03	UINT32	RO	0x3003:03 (digital inputs C/Q)
	04	Subindex 04	UINT32	RO	0x3004:04 (digital inputs pin 2)
1B02	**TxPDO Alignment**				
	00	Number of entries	UINT8	R/W	Number of object subindices
	01	Subindex 01	UINT32	RO	0x0000:00, n (where n can be variable); it is possible that this subindex does not exist if the process data is already aligned.

9.2 Control and status objects

The device has a total of 32 control and status bits.These control and status bits make it possible to set the IO-Link ports that are in "Digital_Out" mode or read out the states of the IO-Link ports that are in "Digital_In" mode.

The objects are always mapped to the cyclic PDO data as TxPDO IOLM_Status/RxPDO IOLM_Control. Furthermore, the IOLM_Control object can be written in the PREOP state by means of SDO access.

The object (index 3000) is mapped in the cyclic output process data under the variable name RxPDO IOLM_Control.

The object (index 3001) is mapped in the cyclic input process data under the variable name TxPDO IOLM_Status.

Index (hex)	Subindex (hex)	Object name	Data type	Rights	Meaning/value	
3000	**IOLM_Control**					
	00	Number of entries	UINT8	RO	Number of object subindices	
	01	ComControl	UINT8	R/W	Enables one or more IO-Link ports **that are operating in Digital_In mode** to temporarily (as long as the corresponding bit is set) switch to IO-Link mode.	
					Bit	**Meaning**
					0	Set port 1 to IO-Link mode
					...	...
					7	Set port 8 to IO-Link mode
	02	Reserved	UINT8	R/W	Reserved	
	03	DO	UINT8	R/W	Enables the IO-Link port to be controlled (set or reset) in Digital_Out (DO) mode.	
	04	Reserved	UINT8	R/W	Reserved	
3001	**IOLM_Status**					
	00	Number of entries	UINT8	RO	Number of object subindices	
	01	COM-States	UINT8	RO	Provides information as to whether the corresponding port or ports have established communication with the IO-Link device.	
					Bit	**Description**
					0	Port 1 IO-Link communication established
					...	...
					7	Port 8 IO-Link communication established
	02	PD-Valid States	UINT8	RO	Provides information as to whether the data or corresponding port is marked as valid or invalid.	
					Bit	**Description**
					0	Port 1 IO-Link input data qualifier
					...	...
					7	Port 8 IO-Link input data qualifier
	03	Digital Inputs C/Q	UINT8	RO	Indicates the state of the digital inputs (pin 4) (X01 ... X08)	
					Bit	**Description**
					0	Input 1 (port 1) pin 4
					...	...
					7	Input 8 (port 1) pin 4
	04	Digital Inputs (Pin 2)	UINT8	RO	Indicates the state of the hardwired digital inputs (pin 2) at type A ports X01 ... X04)	
					Bit	**Description**
					0	Input 1 (port 1) pin 2
					...	...
					3	Input 4 (port 4) pin 2

10 Process data objects

Objects 6000_{hex} to $6FFF_{hex}$ contain the input data and objects 7000_{hex} to $7FFF_{hex}$ contain the output data of the connected IO-Link devices. One object, which can contain several subobjects, is created for each module (IO-Link device). The index of the objects is as follows:

– For input data: Index = 6000_{hex} + (number of the IO-Link port - 1) x 0010_{hex}
– For output data: Index = 7000_{hex} + (number of the IO-Link port - 1) x 0010_{hex}

Therefore, for example, index 7000_{hex} (and its subindices) contains the output data of the module in port 1, index 7010_{hex} (and its subindices) contains the output data of the module in port 2, etc. The number of subobjects each index contains depends on the module configuration performed for the respective port (slot).

Index (hex)	Subindex (hex)	Object name	Data type	Rights	Meaning/value
6000	**IO-Link Input Port 1**				
	00	Number of entries	UINT8	RO	Depends on the slot configuration (max. 32)
	01	IOL Port (1 Byte in)	OCTET-STRING	RO	Represents TxPDO 1 of the first IO-Link device (IO-Link node address is defined in object 0xF020:01)
	02	IOL Port (2 Byte in)	OCTET-STRING	RO	Represents TxPDO 2 of the first IO-Link device (IO-Link node address is defined in object 0xF020:01)
	...	...	...	...	...
	20	IOL Port (32 Byte in)	OCTET-STRING	RO	Represents TxPDO 32 of the first IO-Link device (IO-Link node address is defined in object 0xF020:01)
6010	**IO-Link Input Port 2**				
	00	Number of entries	UINT8	RO	Depends on the slot configuration performed (max. 32)
	01	IOL Port (1 Byte in)	OCTET-STRING	RO	Represents TxPDO 1 of the first IO-Link device (IO-Link node address is defined in object 0xF020:01)
	...	...	...	...	...
6070	**IO-Link Input Port 8**				
...	...	...	...	...	...
7000	**IO-Link Output Port 1**				
	00	Number of entries	UINT8	RO	Depends on the slot configuration performed (max. 32)
	01	IOL Port (1 Byte in)	OCTET-STRING	RO	Represents RxPDO 1 of the first IO-Link device (IO-Link node address is defined in object 0xF020:01)
	02	IOL Port (2 Byte in)	OCTET-STRING	RO	Represents RxPDO 2 of the first IO-Link device (IO-Link node address is defined in object 0xF020:01)
	...	...	...	...	...
	20	IOL Port (32 Byte in)	OCTET-STRING	RO	Represents RxPDO 32 of the first IO-Link device (IO-Link node address is defined in object 0xF020:01)
7010	**IO-Link Outputs Port 2**				
	00	Number of entries	UINT8	RO	Depends on the slot configuration performed (max. 32)
	01	IOL Port (1 Byte out)	OCTET-STRING	RO	Represents RxPDO 1 of the second IO-Link device (IO-Link node address is defined in object 0xF020:01)
...	...	...	...	...	... ...
7070	**IO-Link Outputs Port 8**				
	...	...	...	...	... ...

11 CoE: IO-Link substitute value behavior

The Axioline E EtherCAT® IO-Link master monitors cyclic communication with the controller and responds to possible errors, e.g., communication abort/failure. If EtherCAT® communication fails (EtherCAT® state other than Operational), all IO-Link ports of the device are set to the parameterized substitute values.

The substitute value behavior for ports in IO-Link mode (corresponds to modules: IOL_... or AXL E IOL …) is set for each port via objects 2100_{hex} to 2170_{hex}.

The substitute value behavior of IO-Link ports in “Digital_Out” mode is set via the following objects:

2180_{hex}: Safe State Mode IOLM_Control (DO), sets the behavior globally for all ports

2182_{hex}: Safe State Values IOL_Control (DO), the behavior for each port can be set using the substitute value pattern. This is only valid if “substitute values” is selected in the previous parameter.

Index (hex)	Sub-index	Object name	Data type	Rights	Meaning/value
2100	-	Safe State Mode IOL-Port 1	UINT8	R/W	00_{hex} = Set data to “0”
					01_{hex} = Set data to “1”
					02_{hex} = Hold last value
					03_{hex} = IO-Link master command (default)
					The “IO-Link master command” option allows IO-Link-specific mechanisms to be used for valid/invalid OUT process data.
2110	-	Safe State Mode IOL-Port 2	UINT8	R/W	See above
....	...	...	...	...	...
2170	-	Safe State Mode IOL-Port 8	UINT8	R/W	See above
2180	-	Safe State Mode IOLM_Control (DO)	UINT8	R/W	00_{hex} = Set all outputs to “0”
					01_{hex} = Set all outputs to “1”
					02_{hex} = All outputs hold the last value.
					03_{hex} = Set substitute value pattern
					This option ensures that the substitute value pattern set in object 2182 is applied.
2182	-	Safe State Values IOL_Control (DO)	UINT8	R/W	**Example:** port 2, 5, and 8 are operating in “Digital_Out” mode and should be set in the event of an error.

Port	1	2	3	4	5	6	7	8
Bit	0	1	2	3	4	5	6	7
Substitute value	0	1	0	0	1	0	0	1

If the bit pattern is converted into hex, the resulting value is 92_{hex}. This value must be entered in this object.

The CoE objects for configuring the substitute value behavior can only be set in the EtherCAT® “Pre-Operational” state.

12 Diagnostic strategy

12.1 Mechanisms

Different mechanisms are used to diagnose the devices.

Mechanism		Diagnostic
EtherCAT® state machine		EtherCAT®- system diagnostics
EtherCAT® hardware watchdog		
Emergency messages		Signaling errors to the master
Diagnostic objects in the CoE object dictionary		Extended diagnosis, eg B. peripheral errors
	3001_{hex}	IO status value*
Diagnosic history object		64 diagnostic messages can be stored
	$10F3_{hex}$	Diagnosis history
	10F8hex	Timestamp Object

* see table in chapter 9.2

The device has a state machine called the EtherCAT® State Machine (ESM).

The EtherCAT® master sends state change requests to the AL Control register of the slave. The slave indicates the current state in the AL Status register and also provides additional error codes in the AL Status Code register in the event of errors.

AL Control and AL Status register

If the master writes to the AL Control register, then the corresponding state transition is triggered in the device state machine by the slave. The AL Status register reflects the current state of the slave.

Index (hex)	Object name	Data type	Rights	Meaning/value	
0120	AL Control	UINT16	R/W	Bit 0 ... bit 3	**State (AL status requested by master)**
					01_{hex} = Init (I)
					02_{hex} = Pre-Operational (P)
					03_{hex} = Bootstrap (B)
					04_{hex} = Safe-Operational (S)
					08_{hex} = Operational (O)
				Bit 4	**Acknowledge (master acknowledge bit)**
					00_{hex} = Parameter change of the AL Status register will be unchanged.
					01_{hex} = Parameter change of the AL Status register will be reset.
				Bit 5 ... bit 7	Reserved
					00_{hex} = Shall be zero
0130	AL Status	UINT16	RO	Bit 0 ... bit 3	**AL Status (AL status requested by master)**
					01_{hex} = Init (I)
					02_{hex} = Pre-Operational (P)
					03_{hex} = Bootstrap (B)
					04_{hex} = Safe-Operational (S)
					08_{hex} = Operational (O)
				Bit 4	**Change (error flag, master acknowledge bit)**
					00_{hex} = Confirmation of state in AL Control register
					01_{hex} = A change has happened or an error occurred.
				Bit 5 ... bit 7	Reserved
				Bit 8 ... bit 15	**Application-specific**
					Reserved

AL Status Code register

If the state transition requested by the master is not possible, then the slave sets an error flag in the AL Status register (bit 4) and writes an error code to the AL Status Code register.

AL Status Code (hex)	Description	State or transition	Resulting state
0000	No error	Any	Current state
0001	Unspecified error	Any	Any + E
0011	Invalid requested state change	I→S, I→O, P→O O→B, S→B, P→B	Current state + E
0012	Unknown requested state	Any	Current state + E
0015	Invalid Mailbox Configuration for Bootstrap	I→B	I + E
0017	Invalid Sync Manager Configuration	P→S, S→O	Current state + E
001B	Sync Manager Watchdog	O, S	S + E
001D	Invalid output configuration	O, S P→S	S + E P + E
001E	Invalid input configuration	O, S, P→S	P + E

12.2 EtherCAT®: Emergency messages

Emergency messages are messages that are actively sent from the device to the EtherCAT® master if certain events/problems occur. They are an unacknowledged service which is based on CoE.

This takes the form of messages which are specified in ETG.1000.6.

An emergency message has the following structure:

CoE emergency message		
2 bytes	**1 byte**	**5 bytes**
Error code	Error register	Diagnostic data

The following emergency messages are supported by the device:

Error code (hex)	Error register	Diagnostic data (hex)	Localization	Meaning
1001 ... 1008	Bit D1 set	00, 09, **0x**, 00, 00, **x** = 1 ... 8	Channel level	General error
1011 ... 1018	Bit D1 set	00, 0A, **0x**, 00, 00, **x** = 1 ... 8	Channel level	Parameter missing
1021 ... 1028	Bit D2 set	00, 0B, **0x**, 00, 00, **x** = 1 ... 8	Channel level	Parameterization error
1031 ... 1038	Bit D1 set	00, 0E, **0x**, 00, 00, **x** = 1 ... 8	Channel level	Upper limit exceeded
1041 ... 1048	Bit D1 set	00, 0F, **0x**, 00, 00, **x** = 1 ... 8	Channel level	Lower limit not reached
1051 ... 1058	Bit D1 set	00, 11, **0x**, 00, 00, **x** = 1 ... 8	Channel level	Simulation active
1061 ... 1068	Bit D1 set	00, 12, **0x**, 00, 00, **x** = 1 ... 8	Channel level	Ground fault
2301 ... 2308	Bit D1 set	00, 07, **0x**, 00, 00, **x** = 1 ... 8	Channel level	Short circuit at pin 4
3001 ... 3008	Bit D2 set	00, 01, **0x**, 00, 00, **x** = 1 ... 8	Channel level	Sensor supply overload
3010	Bit D2 set	00, 02, 00, 00, 00	Device level	Overvoltage of the sensor supply
3020	Bit D2 set	00, 03, 00, 00, 00	Device level	Undervoltage of the sensor supply
3030	Bit D2 set	00, 04, 00, 00, 00	Device level	Overvoltage of the actuator supply
3031 ... 3038	Bit D2 set	00, 09, **0x**, 00, 00, **x** = 1 ... 8	Channel level	Overvoltage of the sensor supply
3040	Bit D2 set	00, 05, 00, 00	Device level	Undervoltage of the actuator supply
3041 ... 3048	Bit D2 set	00, 10, **0x**, 00, 00, **x** = 1 ... 8	Channel level	Undervoltage of the sensor supply
3050	Bit D2 set	00, 0D, 00, 00, 00	Device level	Short circuit of the actuator supply
3061 ... 3068	Bit D3 set	00, 0C, **0x**, 00, 00, **x** = 1 ... 8	Channel level	Supply voltage error
4001 ... 4008	Bit D3 set	00, 06, **0x**, 00, 00, **x** = 1 ... 8	Channel level	Overtemperature
5001 ... 5008	Bit D2 set	00, 10, **0x**, 00, 00, **x** = 1 ... 8	Channel level	Fuse blown
8101 ... 8108	Bit D7 set	00, 08, **0x**, 00, 00, **x** = 1 ... 8	Channel level	Short circuit

The device sends emergency messages for an incoming and a going error. After the problem has been solved, an emergency message with the error code 0000_{hex} (Reset error) is sent by the slave.

The value of the corresponding bit in the error register is then 0 if no error is pending for this bit. The value of Diagnostic data does not change.

12.3 CoE: IO-Link device diagnostics (events)

Object $10F3_{hex}$ is available for the diagnostics function. Up to 64 diagnostic messages can be stored in a ring buffer in this object. All events that have triggered an emergency message in the device are recorded.

These are:

– **EtherCAT® system diagnostics**, which are generated by the slave stack (ESM) of the Axioline E EtherCAT® IO-Link master (info, warning, error). A list of all supported emergency messages can be found in the “EtherCAT®: emergency messages” section.

– **IO-Link events**, which are sent from the connected IO-Link device to the master.

In addition, a timestamp is stored for each diagnostic message in object $10F8_{hex}$ (“Timestamp Object”). The “Diagnosis History” object has two operating modes: overwrite mode and acknowledgment mode.

In overwrite mode, new messages overwrite the old ones, i.e., if the ring buffer is full, existing diagnostic messages are overwritten even if they have not been acknowledged yet.

In acknowledgment mode, newer messages are rejected and discarded as soon as the ring buffer is filled completely with unacknowledged messages.

The mode is selected via 0x10F3:SI5:Bit4.

Diagnostic messages are acknowledged via 0x10F3:SI3.

12.4 Diagnostics objects

Index (hex)	Sub-index	Object name	Data type	Rights	Meaning/value	
10F3	**Diagnosis History**		Record		Diagnostic statistics	
	00	Number of entries	UINT8	RO	Number of object subindices	
	01	Maximum messages	UINT8	RO	Maximum number of messages (64)	
	02	Newest message	UINT8	RO	Subindex of the newest diagnostic message (0, 6 ... 69) 0 = No message available	
	03	Newest acknowledged message	UINT8	R/W	Newest acknowledged message	
					In overwrite mode (SI5, bit 4 = 0)	
					Read	
					0	All messages have been deleted.
					Write	
					0	Delete all messages
					6 ... 69	Acknowledge pending message. If no message is available, SDO abort code 06090030_{hex} is sent.
					In acknowledgment mode (SI5, bit 4 = 1)	
					Read	
					0	No messages acknowledged as yet
					! = 0	Subindex of the last acknowledged message
					Write	
					0	Delete all acknowledged messages
					6 ... 69	Acknowledge pending message. If no message is available, SDO abort code 06090030_{hex} is sent.

Index (hex)	Sub-index	Object name	Data type	Rights	Meaning/value	
	04	New message available	BIT	RO	New message available	
					In overwrite mode (SI5, bit 4 = 0)	
					0	Newest messages read
					1	Newest messages not read
					In acknowledgment mode (bit 4 = 1)	
					0	No unacknowledged messages
					1	Messages available for acknowledgment
	05	Flags	UINT16	R/W	Settings for sending and storing diagnostic messages	
					Bit	Description
					0	Send emergency messages 0 = Deactivated 1 = Activated (default)
					1	Store/send info messages 0 = Activated (default) 1 = Deactivated
					2	Store/send warning messages 0 = Activated (default) 1 = Deactivated
					3	Store/send error messages 0 = Activated (default) 1 = Deactivated
					4	Mode selection 0 = Overwrite mode 1 = Acknowledgment mode
					5	Overwrite information (RO) In overwrite mode: 1 = Unacknowledged messages have been overwritten In acknowledgment mode: 1 = Message buffer is full with unacknowledged messages and a new message has been rejected
					6 ... 15	Reserved
	6 ... 255	Diagnosis message	OCTET STRING	RO	The first message is stored in subindex 6, the next in 7, and so on. As soon as the buffer is full, the process starts again from subindex 6 and the old message is overwritten. For the exact structure of diagnostic messages, please refer to the “Structure of a diagnostic message” section.	
10F8	-	Timestamp Object	UINT64	RO	The object contains the current local time of the device and is specified in ns.	

*** SI =** Subindex

Object $1A08_{hex}$ (New Message Available) contains a status bit when a new event occurs and can be integrated into the process data as an option. For more detailed information, please refer to the “PDO Mapping objects” section.

For further information on the $10F3_{hex}$ Diagnosis History object, see ETG.1020.

12.5 Structure of a diagnostic message

Parameter	Data type	Description		
Diag Code	UINT32	Diagnostic code for unique identification of the diagnostic message		
		Bit 0 … 15	**Bit 16 … 31**	**Description**
		E800	Emergency code as per specification	The Axioline E EtherCAT® IO-Link master is informed about a diagnostic event.
Flags	UINT16	**Bit**		**Description**
		0 ... 3		**Type of diagnostics** 0 = Info message 1 = Warning message 2 = Error message Other = Reserved
		4		Time stamp
		5 ... 7		Reserved
		8 ... 15		Number of parameters in this diagnostic message
Text ID	UINT16	Text ID as reference for the diagnostic text defined in the ESI file		
		Bit		**Description**
		0		No Text ID available
		1 ... 65535		Text ID as reference in the ESI file
Time stamp	UINT64	Time stamp in ns from the Timestamp Object ($10F8_{hex}$) at the moment when the event was detected		
Parameter Flags 1	UINT16	Parameter 1Flags1:		
		0005_{hex}		Data type of parameter 1: UINT8
Parameter 1	UINT8	Port number		
Parameter Flags 2	UINT16	Parameter Flags 2:		
		0007_{hex}		Data type of parameter 2: UINT32
Parameter 3	UINT32	Event code; see “Event codes” section		
Parameter Flags 3	UINT16	Parameter Flags 3:		
		0005_{hex}		Data type of parameter 3: UINT8

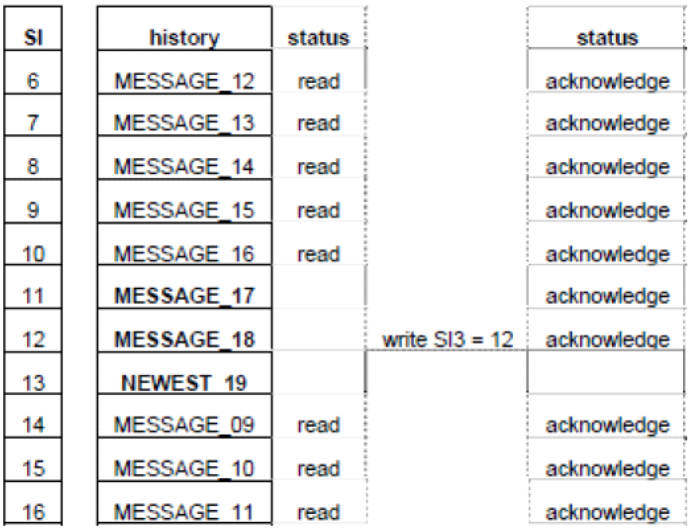
Parameter	Data type	Description	
Parameter 3	UINT8	Event qualifier Provides additional information about the event code	
		Bit	**Description**
		0 ... 2	**Instance (source of the event)** 0 = Unknown 1 ... 3 = Reserved 4 = Application 5 ... 7 = Reserved
		3	**Source** 0 = Device 1 = Master
		4 ... 5	**Type** 0 = Reserved 1 = Single-shot event 2 = Outgoing event 3 = Incoming event

12.6 Acknowledging diagnostic messages

In overwrite mode (SI5, bit 4 = 0)

If the subindex of the newest acknowledged message is written to the “Newest Acknowledged Message” field (0x10F3:SI3), all older diagnostic messages will likewise be acknowledged. There is no check performed to determine whether these messages have already been read (indicated by 0x10F3:SI5:Bit 5 = 0).

When a subindex is written that does not contain a diagnostic message, SDO abort code 06090030_{hex} is returned.



SI	history	status		status
6	MESSAGE_12	read		acknowledge
7	MESSAGE_13	read		acknowledge
8	MESSAGE_14	read		acknowledge
9	MESSAGE_15	read		acknowledge
10	MESSAGE_16	read		acknowledge
11	MESSAGE_17			acknowledge
12	MESSAGE_18		write SI3 = 12	acknowledge
13	NEWEST_19			
14	MESSAGE_09	read		acknowledge
15	MESSAGE_10	read		acknowledge
16	MESSAGE_11	read		acknowledge

Figure 3: Example

In acknowledgment mode (SI5, bit 4 = 1)

If the subindex of the newest acknowledged message is written to the “Newest Acknowledged Message” field (0x10F3:SI3), all older diagnostic messages will likewise be acknowledged. There is no check performed to determine whether these messages have already been read (indicated by 0x10F3:SI5:Bit 5 = 0).

When a subindex is written that does not contain a diagnostic message, SDO abort code 06090030_{hex} is returned.

12.7 Deleting diagnostic messages

In overwrite mode (SI5, bit 4 = 0)

If a zero is written to the “Newest Acknowledged Message” field (0x10F3:SI3), all messages will be deleted without performing a check.

In acknowledgment mode (SI5, bit 4 = 1)

If a zero is written to the “Newest Acknowledged Message” field (0x10F3:SI3), all messages that have already been acknowledged will be deleted. All messages that have not yet been acknowledged will be moved up in the buffer, as illustrated in the example below.

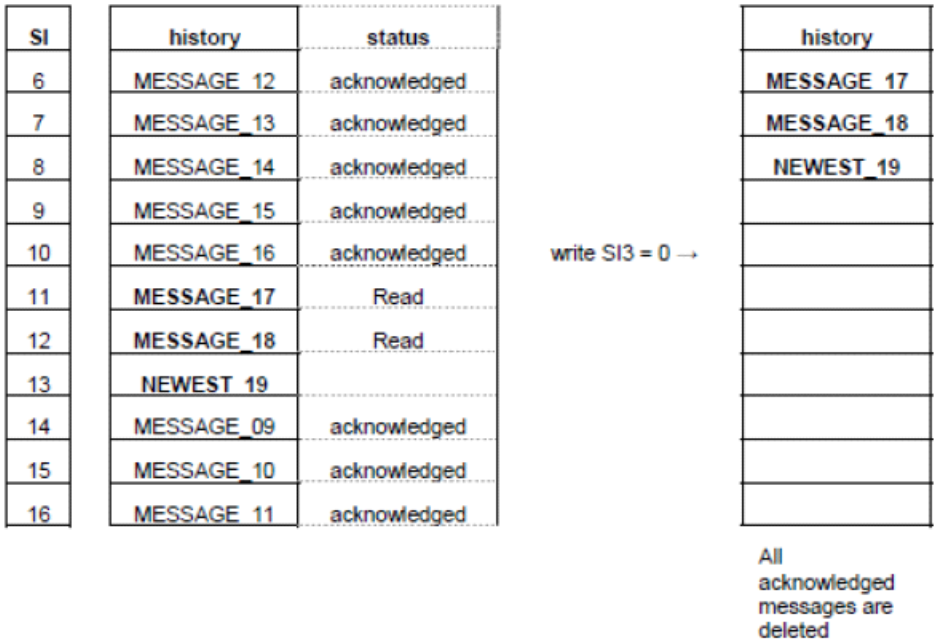


Figure 4: Example

12.8 Event codes

The table below provides an overview of possible event codes according to the IO-Link specification (extract from document: IOL-Interface-Spec_10002_V111_Oct11).

For details of which event codes are supported by the connected IO-Link device, please refer to the associated device documentation.

Event codes (hex)	Definition	Device Status Value	Type
0000	No malfunction	0	Notification
1000	General malfunction – unknown error	4	Error
1001 … 17FF	Reserved		
1800 … 18FF	Manufacturer/ vendor specific		
1900 … 3FFF	Reserved		
4000	Temperature fault – Overload	4	Error
4001 … 420F	Reserved		
4210	Device temperature over-run – Clear source of heat	2	Warning
4211 … 421F	Reserved		
4220	Device temperature under-run – Insulate Device	2	Warning
4221 … 4FFF	Reserved		
5000	Device hardware fault – Device exchange	4	Error
5001 … 500F	Reserved		
5010	Component malfunction – Repair or exchange	4	Error
5011	Non volatile memory loss – Check batteries	4	Error
5012	Batteries low – Exchange batteries	2	Warning
5013 … 50FF	Reserved		
5100	General power supply fault – Check availability	4	Error
5101	Fuse blown/open – Exchange fuse	4	Error
5102 … 510F	Reserved		
5110	Primary supply voltage over-run – Check tolerance	2	Warning
5111	Primary supply voltage under-run – Check tolerance	2	Warning
5112	Secondary supply voltage fault (Port Class B) – Check tolerance	2	Warning
5113 … 5FFF	Reserved		
6000	Device software fault – Check firmware revision	4	Error
6001 … 631F	Reserved		
6320	Parameter error – Check data sheet and values	4	Error
6321	Parameter missing – Check data sheet	4	Error
6322 … 634F	Reserved		
6350	Parameter changed – Check configuration	4	Error
6351 … 76FF	Reserved		
7700	Wire break of a subordinate device – Check installation	4	Error
7701 … 770F	Wire break of subordinate device 1 …device 15 – Check installation	4	Error
7710	Short circuit – Check installation	4	Error
7711	Ground fault – Check installation	4	Error
7712 … 8BFF	Reserved		

Event codes (hex)	Definition	Device Status Value	Type
8C00	Technology specific application fault – Reset Device	4	Error
8C01	Simulation active – Check operational mode	3	Warning
8C02 ... 8C0F	Reserved		
8C10	Process variable range over-run – Process Data uncertain	2	Warning
8C11 ... 8C1F	Reserved		
8C20	Measurement range over-run – Check application	4	Error
8C21 ... 8C2F	Reserved		
8C30	Process variable range under-run – Process Data uncertain	2	Warning
8C31 ... 8C3F	Reserved		
8C40	Maintenance required – Cleaning	1	Notification
8C41	Maintenance required – Refill	1	Notification
8C42	Maintenance required – Exchange wear and tear parts	1	Notification
8C43 ... 8C9F	Reserved		
8CA0 ... 8DFF	Manufacturer/ vendor specific		
8E00 ... AFFF	Reserved		
B000 ... BFFF	Reserved for profiles		
C000 ... FEFF	Reserved		
FF00 ... FFFF	SDCI specific Event codes		

13 AoE: IO-Link device parameterization

The AoE (ADS over EtherCAT®) protocol enables SDO access for the Axioline E EtherCAT® IO-Link master to IO-Link device parameters, known as ISDUs (Indexed Service Data Units).

Not all EtherCAT® masters support the AoE protocol. Please first make sure that your EtherCAT® master is suitable for this type of communication.

Communication is via ADS (Automation Device Specification). The Axioline E EtherCAT® IO-Link master provides appropriate ADS services for reading and writing IO-Link device parameters (ISDUs). The engineering tool then forwards ADS commands to the IO-Link master via AoE (client/server principle).

An AMS NetID (for unique identification of the Axioline E EtherCAT® IO-Link master) and the port number (for unique identification of the IO-Link master and the port) are mandatory for addressing.

IndexGroup $F302_{hex}$ is used for an ADS command.

The TwinCAT® engineering tool provides ready-made ADS services in the form of blocks or function libraries. For further information, please refer to the associated documentation.

13.1 AMS NetID

The AMS NetID is used for unique identification of the EtherCAT® slave (Axioline E EtherCAT® device) and is assigned by the engineering tool, as illustrated in this example using TwinCAT®:

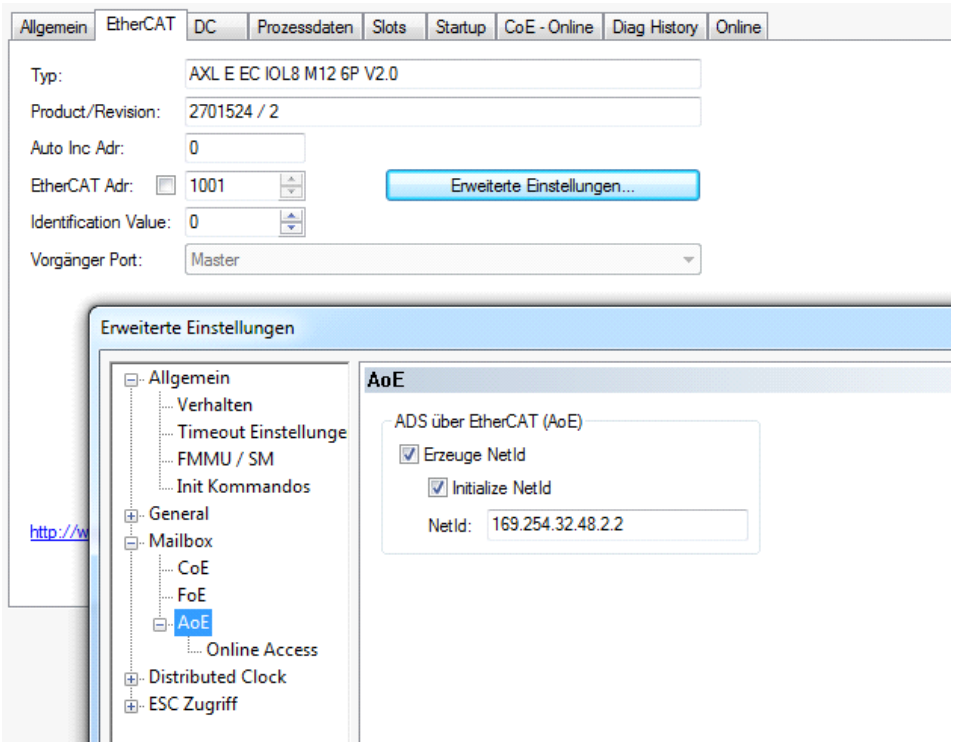


Figure 5: Example in the TwinCAT® engineering tool

13.2 Port number

The port number is used for unique identification of the IO-Link master and its ports. The following applies:

Value (hex)	Description
1000	IO-Link master (the Axioline E device itself)
1001	Port 1
...	...
1008	Port 8

13.3 AoE services

Like CoE, the IndexGroup specified for an ADS command is $F302_{hex}$ for the IO-Link parameter channel. The addressing of the IO-Link ISDU object is coded in the AoE IndexOffset with index and subindex.

The table below provides an overview.

CoE service	AoE service	AoE port	AoE IndexGroup	AoE IndexOffset		AoE data
SDO Upload	Read	EtherCAT® slave address (NetID+PortNo)	$F302_{hex}$	**Bit**	**Description**	Response: read data
				16 ... 31	ISDU index	
				8	0	
				0 ... 7	ISDU subindex	
SDO Download	Write	EtherCAT® slave address (NetID+PortNo)	$F302_{hex}$	**Bit**	**Description**	Request: data to be written
				16 ... 31	ISDU index	
				8	0	
				0 ... 7	ISDU subindex	

14 EtherCAT®: SDO abort codes

The table below provides information about possible error messages during SDO access:

Abort code (hex)	Description
0503 0000	Toggle bit not alternated
0504 0000	SDO protocol timed out
0504 0001	Client/server command specifier not valid or unknown
0504 0005	Out of memory
0601 0000	Unsupported access to an object
0601 0001	Attempt to read to a write only object
0601 0002	Attempt to write to a read only object
0601 0003	Subindex cannot be written, SIO must be 0 for write access
0601 0004	SDO complete access not supported for objects of variable length such as ENUM object types
0601 0005	Object length exceeds mailbox size
0601 0006	Object mapped to RxPDO, SDO download blocked
0602 0000	Object does not exist in the object dictionary
0604 0041	Object cannot be mapped into the PDO
0604 0042	The number and length of the objects to be mapped would exceed PDO length
0604 0043	General parameter incompatibility reason
0604 0047	General internal incompatibility in the device
0606 0000	Access failed due to a hardware error
0607 0010	Data type does not match, length of service parameter does not match
0607 0012	Data type does not match, length of service parameter too high
0607 0013	Data type does not match, length of service parameter too low
0609 0011	Subindex does not exist
0609 0030	Value range of parameter exceeded (only for write access)
0609 0031	Value of parameter written too high
0609 0032	Value of parameter written too low
0609 0036	Maximum value is less than minimum value
0800 0000	General error
0800 0020	Data cannot be transferred or stored to the application
0800 0021	Data cannot be transferred or stored to the application because of local control
0800 0022	Data cannot be transferred or stored to the application because of the present device state
0800 0023	Object dictionary dynamic generation fails or no object dictionary is present

15 Startup

15.1 Delivery state

The device has the following retentive objects:

– 2007_{hex}: Installation location
– $200A_{hex}$: Equipment identification

These objects are empty when delivered.

15.2 Restoring the default settings

To reset the objects to the delivery state, write the value “1” to CoE object $2F00_{hex}$. After powering on again, the device is in the default settings.

Index (hex)	Subindex	Object name	Type	Rights	Meaning/value
2F00	-	Reset to factory settings	UINT8	R/W	0 = Normal operation (default) 1 = Reset device Other = Reserved

15.3 Starting the firmware

Once you have connected the power, the firmware is started. After completion of the firmware boot process, the RDY LED either lights up green or flashes depending on the bus state. The current state of the internal state machine is indicated via the RUN LED.

15.4 Updating the firmware

It is possible to perform a firmware update via EtherCAT®. The File Access over EtherCAT® (FoE) mechanism is used for this, which is provided via your engineering system.

For detailed instructions for the firmware update with TwinCAT®, please refer to AH EN FIRMWARE UPDATE AXL E EC. The application note can be downloaded via the product at phoenixcontact.net/products.

16 Monitoring/substitute value behavior

The device monitors the connection for network communication. In the event of a connection timeout, the device switches its outputs to the previously parameterized safe state. For further information, please refer to the “Safe State Objects” section.

17 Diagnostic messages

The following diagnostic messages are indicated by the EtherNet/IP™ device:

Diagnostic messages	Status bit process data	Network-specific diagnostic mapping
Overtemperature device	-	Yes
Undervoltage U_S	-	Yes
Overload U_S	-	Yes
Overvoltage U_S	-	Yes
Undervoltage/Overload/ Short-circuit U_A	-	Yes
Overvoltage U_A	-	Yes
Overload/Short-circuit of an output	-	Yes
Cable break at IO-Link port	Port status/ Status module*	Yes

* Port status = 1; the IO-Link connection is active.
* Port status = 0; the IO-Link connection is interrupted.

The LED status can be found in the data sheet.

18 IO-Link master

IO-Link is an internationally standardized I/O technology (IEC 61131-9) for communicating with sensors and actuators. An IO-Link master is integrated in the EtherCAT® device. The IO-Link master establishes the connection between the IO-Link devices and the automation system. The device supports IO-Link specification v1.1.

19 Web-based management (WBM)

The device has a web server, which generates the required pages for web-based management and, depending on the requirements of the user, sends them to a standard web browser. Web-based management can be used to access static information (e.g., technical data, MAC address) or dynamic information (e.g., IP address, status information).

To access web-based management, proceed as follows for TwinCAT®, for example:

- Assign an IP address to the device.
  To do this, open the “Extended Settings” window on the “EtherCAT” tab in TwinCAT®.
  Open “Mailbox, EoE”.
- Confirm the settings with "OK".
- Reload the devices in TwinCAT®.
  You can now access the device via the web server.

20 Device description file (ESI)

In the case of EtherCAT®, a device description file is requ red for various configuration tools. This file is called EtherCAT® Slave Information (ESI). The Axioline E EtherCAT® IO-Link device has two ESI files. The modules together with their associated data objects are written to ESI file “PHOENIX_CONTACT_AXL_E_EC_IOL8_Modules-XXX.xml”, which is then referenced in the main ESI file called “PHOENIX_CONTACT_AXL_E_EC_IOL8_6X-XXX”. Both ESI files are required in order to ensure correct operation.

If several versions of the configuration file are available, make sure that you are working with the file version that corresponds to the firmware/hardware version used.

The current device description files can be downloaded via the product at phoenixcontact.net/products.

21 Data format

EtherCAT® uses Little Endian format. All variables, parameters, and data in this document are in Little Endian format (Intel), i.e., LSB/MSB.

Please observe the following notes

General terms and conditions of use for technical documentation

Phoenix Contact reserves the right to alter, correct, and/or improve the technical documentation and the products described in the technical documentation at its own discretion and without giving prior notice, insofar as this is reasonable for the user. The same applies to any technical changes that serve the purpose of technical progress.

The receipt of technical documentation (in particular user documentation) does not constitute any further duty on the part of Phoenix Contact to furnish information on modifications to products and/or technical documentation. You are responsible to verify the suitability and intended use of the products in your specific application, in particular with regard to observing the applicable standards and regulations. All information made available in the technical data is supplied without any accompanying guarantee, whether expressly mentioned, implied or tacitly assumed.

In general, the provisions of the current standard Terms and Conditions of Phoenix Contact apply exclusively, in particular as concerns any warranty liability.

Phoenix Contact reserves the right to register its own intellectual property rights for the product identifications of Phoenix Contact products that are used here. Registration of such intellectual property rights by third parties is prohibited.

Other product identifications may be afforded legal protection, even where they may not be indicated as such.

How to contact us

Internet

Up-to-date information on Phoenix Contact products and our Terms and Conditions can be found on the Internet at:
phoenixcontact.com

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

Subsidiaries

If there are any problems that cannot be solved using the documentation, please contact your Phoenix Contact subsidiary.
Subsidiary contact information is available at phoenixcontact.com.

Published by

PHOENIX CONTACT GmbH & Co. KG
Flachsmarktstraße 8
32825 Blomberg
GERMANY

Should you have any suggestions or recommendations for improvement of the contents and layout of our manuals, please send your comments to:
tecdoc@phoenixcontact.com

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

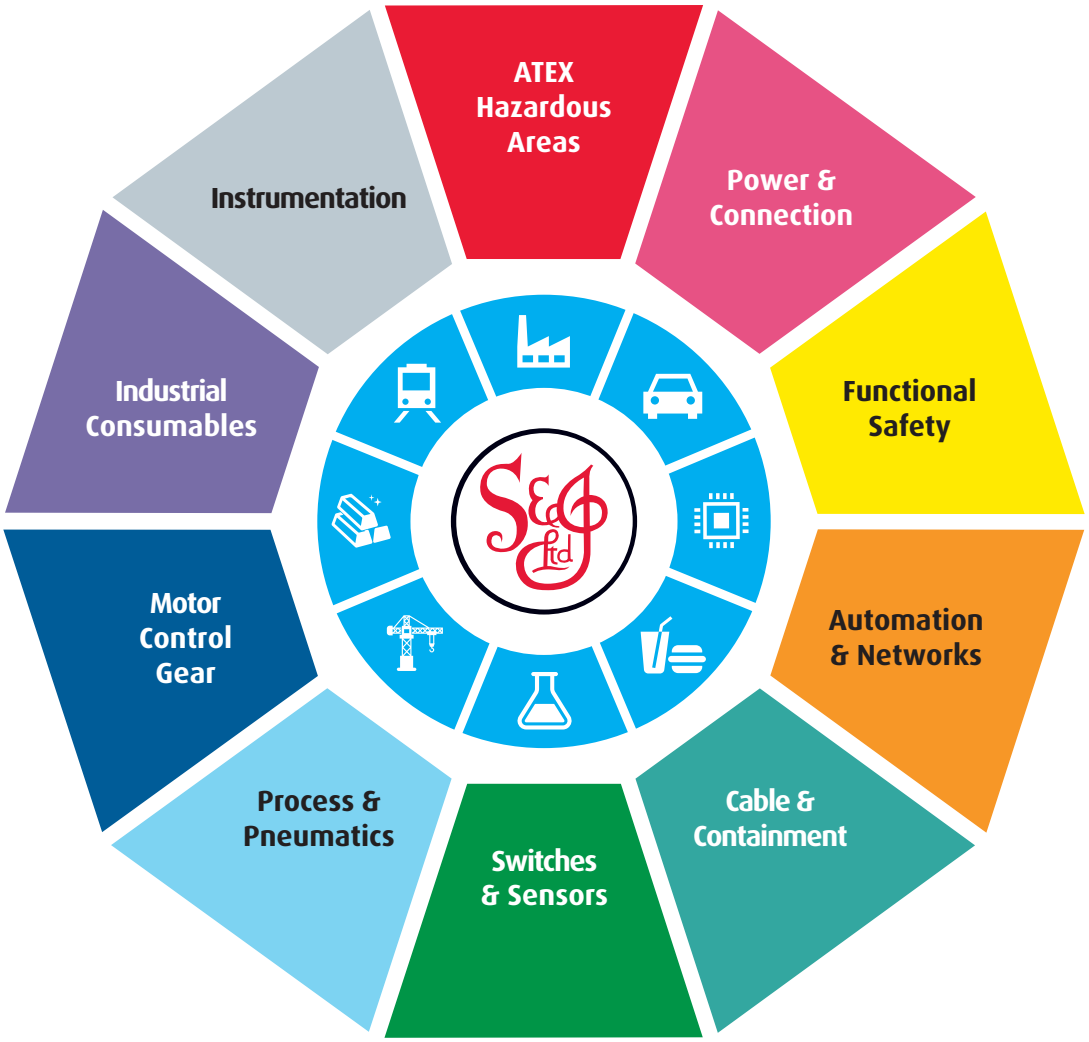
At Scattergood & Johnson Ltd, we pride ourselves on being a technical distributor to specialist industries.

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