DOWNLOADED FROM WWW.SCATTS.CO.UK

Axioline F bus coupler for EtherNet/IP™: object classes, messages, and services

User manual

PHOENIX CONTACT
INSPIRING INNOVATIONS

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

User manual

Axioline F bus coupler for EtherNet/IP™: object classes, messages, and services

2016-11-11

Designation: UM EN AXL F BK EIP (EF) - OBJECTS

Revision: 02

This user manual is valid for:

Designation	Order No.
AXL F BK EIP	2688394
AXL F BK EIP EF	2702782

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

Please observe the following notes

User group of this manual

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

- Qualified electricians or persons instructed by them, who are familiar with applicable standards and other regulations regarding electrical engineering and, in particular, the relevant safety concepts.
- Qualified application programmers and software engineers, who are familiar with the safety concepts of automation technology and applicable standards.

Explanation of symbols used and signal words

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety measures that follow this symbol to avoid possible injury or death.

There are three different categories of personal injury that are indicated with a signal word.

DANGER This indicates a hazardous situation which, if not avoided, will result in death or serious injury.

WARNING This indicates a hazardous situation which, if not avoided, could result in death or serious injury.

CAUTION This indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

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

This symbol and the accompanying text provide the reader with additional information or refer to detailed sources of information.

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

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

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.

This manual, including all illustrations contained herein, is copyright protected. Any changes to the contents or the publication of extracts of this document is prohibited.

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.

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

Table of contents

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1 EtherNet/IP™ - object classes, messages, and services

1.1 General information

The information in this user manual apply in general for both bus couplers. Any differences will be indicated.

The bus coupler maps the I/O device connected to the standard or user-defined CIP objects via the local bus.

The bus coupler supports the Common Industrial Protocol (CIP). EtherNet/IP™ uses the Common Industrial Protocol (CIP) as the application layer. IP and TCP or UDP are used for the network and transport layers. CIP and EtherNet/IP™ are standardized by the ODVA on a manufacturer-neutral basis. The Common Industrial Protocol is an object-oriented protocol with two different types of communication between a controller and terminal devices. The following table describes the two communication types.

Table 1-1 Connection types

Connection type	Description
Explicit messaging	Explicit messaging is based on the principle of “request/response”. This means that a controller or engineering system sends a request and the terminal device responds. For example, explicit messaging can be used for configuration and/or diagnostics.
Implicit messaging	Implicit messaging is used for the cyclic transmission of I/O data. That means, for example, that a terminal device sends an analog value which is present at a terminal device input. The time for a transmission cycle can be set via the requested packet interval (RPI).

1.2 CIP classes and instance services

The bus coupler supports the following classes and instance services:

Table 1-2 Supported classes and instance services

Service code		Service name
dec	hex	
01	01	Get_Attribute_All
02	02	Set_Attribute_All
05	05	Reset
09	09	Delete
14	0E	Get_Attribute_Single
16	10	Set_Attribute_Single

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.3 CIP object classes

The bus couplers AXL F BK EIP and AXL F BK EIP EF support the following CIP object classes:

Table 1-3 Supported CIP object classes

Class code		Object type	on page
dec	hex		
01	01	Identity object	9
02	02	Router object	12
04	04	Assembly object	13
55	37	File object	14
100	64	Configuration object	22
102	66	Axioline F module object	25
103	67	Axioline F diagnostics object	28
105	69	PDI object	32
245	F5	TCP/IP interface object	33
246	F6	Ethernet link object	36

The AXL F BK EIP EF bus coupler additionally supports the following CIP object classes:

Tabelle 1-4 Supported CIP object classes

Class code		Object type	on page
dec	hex		
71	47	Device level ring object	22
72	48	Quality of service objekt	20

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.4 Identity object (class code 01_{hex})

The “Identity object” is required by all devices and provides the device ID and general information on the device.

1.4.1 Class attributes

Table 1-5 Identity object - class attributes

Attribute	Name	Access	Data type	Value
1	Revision	Get	UINT	1
2	Max instance	Get	UINT	1
6	Max class attribute	Get	UINT	7
7	Max instance attribute	Get	UINT	7

1.4.2 Instance attributes

Table 1-6 Identity object - instance attributes

Attribute	Name	Access	Data type	Value
1	Vendor ID	Get	UINT	562
2	Device type	Get	UINT	12 = Communication adapter
3	Product code	Get	UINT	8168 The “product code” is used in electronic data sheet format for clear identification of the product type (“device type”).
4	Revision	Get	STRUCT OF	Through implementing additional functions, the “major revision” value is increased. The “minor revision” value is increased when minor changes are incorporated.
	Major revision		USINT	1
	Minor revision		USINT	10

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

Table 1-6 Identity object - instance attributes [...]

Attribute	Name	Access	Data type	Value		
5	Device status	Get	UINT	Bit 0	Owned	0 = Not used 1 = Assigned
				Bit 1	Reserved	0
				Bit 2	Configured	1 = Configured
				Bit 3	Reserved	0
				Bit 4 ... 7	Extended device status	0 = Self-test 1 = Executing firmware update 2 = At least one faulty I/O connection 3 = No I/O connection established 4= Non-volatile configuration is incorrect 5= Major fault, bit 10 or 11 is True 6 = At least one I/O connection is in Run mode 7 = At least one I/O connection is established, all are in Idle mode 8 ... 15 = Reserved
				Bit 8	Minor recoverable fault	0 = No errors 1 = Minor recoverable error (e.g., short circuit of an output)
				Bit 9	Minor unrecoverable fault	0 = No errors 1 = Minor unrecoverable fault
				Bit 10	Major recoverable fault	0 = No errors 1 = Major recoverable fault (e.g., loss of +24 V DC)
				Bit 11	Major unrecoverable fault	0 = No errors 1 = Major unrecoverable fault (e.g., checksum, A/D)
				Bit 12 ... 15	Reserved	0
6	Serial number	Get	UDINT	The serial number is coded in the product during manufacturing and has a once-off guarantee in the Phoenix Contact product groups.		
7	Product name	Get	STRUCT OF			
	Length		USIGN	12		
	Name		STRING	AXL F BK EIP		

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.4.3 Services

Table 1-7 Identity object - available services

Service code		Class	Instance	Service name
dec	hex			
05	05	yes	yes	Reset (type 0, 1)
14	0E	yes	yes	Get_Attribute_Single

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.5 Router object (class code 02_{hex})

The “Router object” provides a messaging connection point through which a client may address a service to any object class or instance in a physical device.

1.5.1 Class attributes

Table 1-8 Router object - class attributes

Attribute	Name	Access	Data type	Value
1	Revision	Get	UINT	1
2	Max instance	Get	UINT	1
6	Max class identifier	Get	UINT	7
7	Max instance attribute	Get	UINT	2

1.5.2 Instance attributes

Table 1-9 Router object - instance attributes

Attribute	Name	Access	Data type	Value
1	Router class list	Get	ARRAY	0C 00 01 00 02 00 04 00 06 00 67 00 F4 00 F5 00 F6 00 64 00 66 00 69 00 37 00
2	Max connections	Get	UINT	32

1.5.3 Services

Table 1-10 Router object - available services

Service code		Class	Instance	Service name
dec	hex			
14	0E	yes	yes	Get_Attribute_Single

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.6 Assembly object (class code 04_{hex})

The “Assembly object” combines attributes of multiple objects to allow data to or from each object to be sent or received via a single connection.

1.6.1 Class attributes

Table 1-11 Assembly object - class attributes

Attribute	Name	Access	Data type	Value
1	Revision	Get	UINT	2
2	Max instance	Get	UINT	102
6	Max class attribute	Get	UINT	7
7	Max instance attribute	Get	UINT	4

1.6.2 Instance attributes

Table 1-12 Assembly object - instance attributes (instance 110)

Attribute	Name	Access	Data type	Description
3	Data	Get, Set	ARRAY OF BYTE	Process data
4	Size	Get	UINT	Byte count in attribute 3

Attributes of instance 100

Instance 100 is used to generate the consumed I/O data.
It comprises a variable quantity of output process data. The count is dependent on the station setup.

Attributes of instance 110

Instance 110 is used to generate the produced I/O data.
It comprises a variable quantity of input process data. The count is dependent on the station setup.

1.6.3 Services

Table 1-13 Assembly object - available services

Service code		Class	Instance	Service name
dec	hex			
14	0E	yes	yes	Get_Attribute_Single
16	10	no	yes	Set_Attribute_Single

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.7 File object (class code 37_{hex})

1.7.1 Class attributes

Table 1-14 File object - class attributes

Attri-bute	Name	Access	Data type	Value
1	Class revision	Get	UINT	1
2	Max instance number	Get	UINT	200
3	Number of instances	Get	UINT	2
6	Max class attribute	Get	UINT	32
7	Max instance attribute	Get	UINT	11
32	Directory	Get	STRUCT OF	List of all instances and file names that are assigned to this device and the corresponding “Instance number”
	Instance number		UINT	(1), see 1.7.5 on page 17
	Instance name		STRING	
	File name		STRING	

1.7.2 Class services

Table 1-15 File object - available class services

Service code		Service name	Description
dec	hex		
14	0E	Get_Attribute_Single	Returned data for the above-listed class attributes
1	01	Get_Attribute_All	Returns all of the above-listed class attributes

The optional Create/Delete services are not supported.

1.7.3 Instance attributes

Table 1-16 File object - instance attributes

Attri-bute	Name	Access	Data type	Value/description
1	State	Get	USINT	(2), see 1.7.5 on page 17
2	Instance name	Get	STRING	(1), see 1.7.5 on page 17
3	File format version	UINT		(1), see 1.7.5 on page 17
4	File name	Get		(1), see 1.7.5 on page 17
5	File revision	Get	STRUCT OF	(1), see 1.7.5 on page 17
	Major revision		USINT	
	Minor revision		USINT	

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

Table 1-16 File object - instance attributes [...]

Attribute	Name	Access	Data type	Value/description
6	File size	Get	UDINT	Size of file in bytes
7	File checksum	Get	UINT	Two's complement of the 16-bit total of all bytes
8	Invocation method	Get	USINT	(3), see 1.7.5 on page 17
9	File save parameters	Get	BYTE	0
10	File access rule	Get/Set	USINT	(1), see 1.7.5 on page 17 0 = Access rights = Read/write 1 = Access right = Read only
11	File encoding format	Get	USINT	(1), see 1.7.5 on page 17 0 = No encryption 1 = Encryption

1.7.4 Instance services

Table 1-17 File object - available instance services

Service code		Service name	Description
dec	hex		
14	0E	Get_Attribute_Single	Read instance attribute
1	01	Get_Attribute_All	Read all instance attributes
16	10	Set_Attribute_Single	Write instance attribute
75	4B	Initiate_Upload	Starts the upload process. The request contains the maximum file size which can be accepted during upload. The response contains the actual file size.
79	4F	Upload_Transfer	Upload of file data. Each request contains the transfer number, which is increased with each subsequent transfer. The response includes the transfer number, transmission type, the data to be transmitted, and the checksum in the last packet. The transmission type indicates whether the first, middle or last packet is involved, as well as whether it is the only packet or if the transmission should be canceled.

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

Table 1-17 File object - available instance services [...]

Service code		Service name	Description
dec	hex		
76	4C	Initiate_Download	Starts the download of a file to the device. The request includes the download size to be transmitted, the version of the instance format, the file version, and the file name. The response includes the following values: Size of burned data: byte number before it was retentively saved Burn duration: number of seconds which should be reserved for the retentive saving. Transmission size: number of bytes which were sent with each "Download Transfer" request.
80	50	Download_Transfer	Download of file data to the device. The request contains the transfer number, which is increased with each subsequent transfer. The request also includes the transmission type, the data to be transmitted, and the checksum of the last transmission. The transfer type indicates whether the first, middle or last packet is involved, as well as whether it is the only packet or if the transmission should be canceled. The response includes the corresponding transfer number.
81	51	Clear_File	Deletes content of the file. The status of the file instance is set to “File is empty” and the file size is set to 0.

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.7.5 Values of the class and instance attributes

(1)

Instance	State	Instance name	Instance format version	File name	Version	Invocation method	Type	Encoded
3	2	Configuration file	1	config.svc	1.0	2	Read/Write	False
200	2	EDS and icon file	1	EDS.gz	1.0	0	Read only	True

(2) **State**

0	Does not exist
1	File is empty (no file downloaded)
2	File downloaded
3	Uploaded started
4	Download started
5	Upload running
6	Download running
7	File saving
8 ... 255	Reserved

(3) **Invocation method**

This value defines what should happen once the file has been downloaded.
Possible options include:

0	No action
1	Reset via “Identity object”
2	Voltage reset

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.8 Device level ring object (Class code 47_{hex})

This object is only supported by the AXL F BK EIP EF bus coupler.

Device level ring = DLR

The “Device level ring object” provides the configuration of the DLR protocol.

1.8.1 Class attributes

Tabelle 1-18 Device level ring object - class attributes

Attribute	Name	Access	Data type	Value
1	Revision	Get	UINT	3
2	Max instance number	Get	UINT	1
3	Number of instances	Get	UINT	1
6	Max class attribute	Get	UINT	7
7	Max instance attribute	Get	UINT	12

1.8.2 Instance attributes

Tabelle 1-19 Device level ring object - instance attributes

Attri-bute	Name	Access	Data type	Value/note	
1	Network topology	Get	USINT	0 = Linear topology 1 = Ring topology	
2	Network status	Get	USINT	(1), see 1.8.4 on page 19	
10	Active supervisor address	Get	STRUCT of:	(2), see 1.8.4 on page 19	
	Supervisor IP address		UDINT	0.0.0.0	
	Supervisor MAC adress		ARRAY of 6 USINTs	00 00 00 00 00 00	
12	Capability flags	Get	DWORD	(3), see 1.8.4 on page 19	
				Bit 0	Announced-based ring node*
				Bit 1	Beacon-based ring node
				Bit 2 ... 4	Reserved (= 0)
				Bit 5	Supervisor capable*
				Bit 6	Redundant gateway capable*
				Bit 7	Flush_Table frame capable
				Bit 8 ... 31	Reserved (= 0)

* This function is not supported by the AXL F BK EIP EF.

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.8.3 Services

Tabelle 1-20 Device level ring object - available services

Service code		Class	Instance	Service name
dec	hex			
1	01	yes	yes	Get_Attribute_All
14	0E	yes	yes	Get_Attribute_Single

1.8.4 Values of the instance attributes

(1) **Network status**

0	Normal	Normal operation both in ring topology mode and in linear topology mode.
1	Ring fault	Ring fault. A ring fault has been detected. Only valid if the network topology is a ring.
2	Unexpected loop detected	A ring loop has been detected. Only valid if the network topology is a line.
3	Partial network fault	Partial network fault. A network error has been detected in only one direction. Only valid if the network topology is a ring and the node is the active ring supervisor. (not supported by the AXL F BK EIP EF)
4	Rapid fault/Restore cycle	A series of rapid ring faults/restore cycles has been recorded. Like the partial network fault status, the supervisor nodes remain in a state where forwarding to a ring port is blocked. The condition must be explicitly cleared using the “Clear rapid faults” service.

(2) **Active supervisor address**

This attribute contains the IP address and/or the Ethernet MAC address of the active ring supervisor. The initial values of the IP address and MAC address must be 0 until the active ring supervisor is determined.

(3) **Capability flags**

These flags describe the DLR capabilities of the device.

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.9 Quality of service object (class code 48_{hex})

This object is only supported by the AXL F BK EIP EF bus coupler.

Quality of Service (QoS) affects the forwarding and handling of data streams and results in individual data streams being given differential treatment (usually preferential). QoS can be used, e.g., to guarantee a transmission bandwidth for individual data streams. The bus coupler uses QoS in connection with prioritization.

1.9.1 Class attributes

Tabelle 1-21 Quality of service object - class attributes

Attribute	Name	Access	Data type	Value
1	Revision	Get	UINT	1
2	Max instance number	Get	UINT	1
3	Number of instances	Get	UINT	1
6	Max class attribute	Get	UINT	7
7	Max instance attribute	Get	UINT	8

1.9.2 Instance attributes

Tabelle 1-22 Quality of service object - Instanzattribute

Attri-bute	Name	Access	Data type	Value/note
4	DSCP urgent	Get, Set	USINT	DSCP value for CIP transport class 0/1, urgent messages
5	DSCP scheduled	Get, Set	USINT	DSCP value for CIP transport class 0/1, planned messages
6	DSCP high	Get, Set	USINT	DSCP value for CIP transport class 0/1, high-priority messages
7	DSCP low	Get, Set	USINT	DSCP value for CIP transport class 0/1, low-priority messages
8	DSCP explicit	Get, Set	USINT	DSCP value for CIP explicit messages (transport class 2/3 and UCMM) and all other EtherNet/IP™ encapsulation messages

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.9.3 Services

Tabelle 1-23 Quality of service object - available services

Service code		Class	Instance	Service name
dec	hex			
1	01	yes	yes	Get_Attribute_All
14	0E	yes	yes	Get_Attribute_Single
16	10	no	yes	Set_Attribute_Single

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.10 Configuration object (class code 64_{hex})

The “Configuration object” allows you to configure what data you want in each I/O connection. In addition, you gain access to operational parameters such as status information via the object.

1.10.1 Class attributes

Table 1-24 Configuration object - class attributes

Attribute	Name	Access	Data type	Value
1	Revision	Get	UINT	1
2	Max object instance	Get	UINT	1
6	Max class identifier	Get	UINT	7
7	Max instance attribute	Get	UINT	25

1.10.2 Instance attributes

Table 1-25 Configuration object - instance attributes

Attribute	Name	Access	Data type	Value/note
1	Reload SVC	Set	USINT	1 = Read parameter file config.svc again
2	NetFail Ctrl	Set	USINT	1 = Set NetFail 2 = Acknowledge NetFail
3	Add DiagData to Consumed PD (8 bytes)	Set	USINT	1 = The first 8 bytes of the produced I/O data contain the Axioline F diagnostics register, as in the diagnostics object class attributes from 100 to 103. (Adds the produced 8 bytes of data).
4	Produced size	Get	UINT	Data size Determine produced data for entire station
5	Consumed size	Get	UINT	Data size Determine consumed data for entire station
6	HW Diag Ctrl	Set	USINT	1 = Reset the UL monitor (6), see 1.10.4 on page 24
7	Confirm startup parameterization	Set	USINT	1 = Acknowledge startup parameterization (7), see 1.10.4 on page 24
8	P&P mode	Get/Set	UINT	Select Plug and Play mode 0 = Disabled 1 = Enabled

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

Table 1-25 Configuration object - instance attributes [...]

Attribute	Name	Access	Data type	Value/note
10	BootP/DHCP Operation	Get/Set	USINT	Select BootP/DHCP mode 0 = Static IP 1 = Activate BootP 2 = Activate DHCP 3 = Activate DCP
11	Device name	Get	ARRAY_OF_BYTE (125 bytes)	AXL F BK EIP
12	Description	Get	ARRAY_OF_BYTE (125 bytes)	EtherNet/IP™ bus terminal
13	Installation location	Get	ARRAY_OF_BYTE (125 bytes)	Unknown
14	Contact	Get	ARRAY_OF_BYTE (125 bytes)	Unknown
15	Boot loader version	Get	UINT32	3
16	Firmware version	Get	ARRAY_OF_BYTE (6 bytes)	1.10
17	Firmware status	Get	UINT32	-
18	Hardware version	Get	UINT16	03
19	Firmware date	Get	ARRAY_OF_BYTE (6 bytes)	160315
20	Hardware date	Get	ARRAY_OF_BYTE (6 bytes)	280315
21	Serial number	Get	ARRAY_OF_BYTE (19 bytes)	1234567890
22	MAC address	Get	ARRAY_OF_BYTE (17 bytes)	00:A0:45:XX:XX:XX
23	Order No.	Get	ARRAY_OF_BYTE (19 bytes)	2688394 2702782
24	Designation	Get	ARRAY_OF_BYTE (30 bytes)	AXL F BK EIP AXL F BK EIP EF
25	Manufacturer name	Get	ARRAY_OF_BYTE (19 bytes)	Phoenix Contact
26	Manufacturer ID	Get	ARRAY_OF_BYTE (10 bytes)	FFFFFF00
27	Use last IP address for BootP/DHCP	Get/Set	USINT	0 When setting this attribute, the last valid IP address for BootP/DHCP is suggested. It is only applied following the reset.

The attributes 11 ... 26 map the electronic rating plate. This contains the basic information on the module.

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.10.3 Services

Table 1-26 Configuration object - available services

Service code		Class	Instance	Service name
dec	hex			
14	0E	yes	yes	Get_Attribute_Single
16	10	no	yes	Set_Attribute_Single

Changing the “configuration object” will cause the RECEIVED and GENERATED size of the POLL connection to be changed. These values are retained in non-volatile memory and may only be set when the I/O connection is not in the RUNNING state.

1.10.4 Values of the instance attributes

(6) **HW Diag Ctrl**

Reset the UL monitor. A monitoring function monitors the supply of the U_L communications power. Under/overshooting the specified voltage range is reported in diagnostics object (67_{hex}) in class attribute 100.

Undervoltage	
Bit 12	Set as long as the communications power is too low.
Bit 14	Still set after a brief undervoltage.

Surge voltage	
Bit 13	Set as long as the communications power is too high.
Bit 15	Still set after a brief surge voltage.

Very brief voltage disturbances are also registered with bits 14 and 15. Both bits remain set until the UL monitor has been acknowledged with the “HW Diag Ctrl” attribute.

(7) **Confirm startup parameterization**

Acknowledge startup parameterization. The acknowledgement is applied straight away. The parameterization is permanently acknowledged, i.e., a successful restart does not result in another message.

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.11 Axioline F module object (class code 66_{hex})

The “Axioline F module object” allows you to monitor the Axioline F modules connected to the bus coupler.

1.11.1 Class attributes

The class attributes 100 and 101 map the currently loaded bus configuration frame of the connected devices.

When Plug and Play mode is activated, the bus configuration frame that is physically present is sent. When Plug and Play mode is deactivated, the stored reference configuration is sent.

Any differences between the stored reference configuration and the bus configuration that is physically present are indicated by diagnostic object (67_{hex}), attributes 100 to 103.

The two “Config arrays” (device type and PD length) can be used to monitor the bus configuration in the user application. The entire table can always be read. The “2 Max object instance” class attribute specifies how many devices are actually available.

Table 1-27 Axioline F module object - class attributes

Attribute	Name	Access	Data type	Value/note
1	Revision	Get	UINT	1
2	Max object instance	Get	UINT	Number of Axioline F modules
6	Max class identifier	Get	UINT	101
7	Max instance attribute	Get	UINT	4
100	Config array device type code	Get	ARRAY_OF_USINT	Number of Axioline F modules * 8 bytes (100), see 1.11.3 on page 26
101	Config array PD length	Get	ARRAY_OF_USINT	Number of Axioline F modules * 1 byte (101), see 1.11.3 on page 26

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.11.2 Attributes of instances 1 ... (number of Axioline F modules)

Table 1-28 Axioline F module object - instance attributes of instances 1 ... (number of Axioline F)

Attribute	Name	Access	Data type	Value
1	Axioline F Device Type Code (Config)	Get	ARRAY_OF_USINT (8 bytes)	Configured “DeviceType”
2	Axioline F PD length, number of bytes (Config)	Get	USINT	Configured process data length in bytes
3	Axioline F DeviceType Code (Current)	Get	ARRAY_OF_USINT (8 bytes)	Current “DeviceType”
4	Axioline F PD length, number of bytes (current)	Get	USINT	Current process data length in bytes

Services

Table 1-29 Axioline F module object - available services

Service code		Class	Instance	Service name
dec	hex			
14	0E	yes	yes	Get_Attribute_Single
16	10	no	no	Set_Attribute_Single

1.11.3 Values of class attributes

(100)

Config array DeviceType code

The “DeviceType” is a manufacturer-specific module identification. It can be used to replace and operate modules of the same type within a bus configuration. For example, a 16-channel output module with screw connection technology can be replaced by a module with spring-cage connection technology even though it does not have the same order number. On the other hand, a different functionality (e.g., 32 channels instead of 16) is indicated by a different “DeviceType”. The “DeviceType” acts as a uniquely assigned ID, but the module functionality cannot be directly derived from it (e.g., by evaluating a specific bit). Should this be necessary, use the relevant PDI objects for this (see module-specific data sheet).

Table 1-30 Config array device type code

Byte	Content	
0 ... 7	DeviceType	1st device
8 ... 15	DeviceType	2nd device
…	…	…
496 ... 503	DeviceType	63rd device

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

(101) **Config array PD length**

This class attribute specifies the number of bytes assigned by each device in the process data channel. This information can be used to dynamically adapt the user application to changes in the bus configuration. The offset for the relevant device in the process data table can therefore be calculated in the user application.

Table 1-31 Config array PD length

Byte	Content	
0	Amount of process data	1st device
1	Amount of process data	2nd device
…	…	…
62	Amount of process data	63rd device

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.12 Axioline F diagnostics object (class code 67_{hex})

This object is also referred to as a diagnostics object in the following section.

1.12.1 Class attributes

Table 1-32 Axioline F diagnostics object - class attributes

Attribute	Name	Data type	Access	Value
1	Revision	UINT	Get	1
2	Max object instance	UINT	Get	0 ... 63
6	Max class identifier	UINT	Get	103
7	Max instance attribute	UINT	Get	4
100	General status	UINT	Get	(100), see “General status” on page 28
101	Diagnosis status	UINT	Get	(101), see “Diagnosis status” on page 29
102	Diagnosis parameter 1	UINT	Get	
103	Diagnosis parameter 2	UINT	Get	

(100) General status

Table 1-33 General status

Bit	Code (hex)	Meaning	
0	0001	1	An error occurred in the local bus (e.g., a bit in the diagnostic register is set)
		0	No error
1	0002	1	A NetFail error occurred, active substitute values
		0	No error
2	0004	1	Active bus configuration does not match the reference configuration
		0	No error
3	0008	1	Startup parameterization is faulty
		0	No error
4	0010	1	Plug and Play mode is activated.
		0	Plug and Play mode is deactivated.
5	0020	Reserved	
6	0040	Reserved	
7	0080	Reserved	
8	0100	1	Overtemperature of the power supply
		0	Normal temperature
9	0200	1	Overtemperature of the logic PCB
		0	Normal temperature

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

Table 1-33 General status

Bit	Code (hex)	Meaning	
10	0400	Reserved	
11	0800	Reserved	
12	1000	1	Undervoltage active (voltage is below the specified range)
		0	Voltage OK
13	2000	1	Undervoltage fault detected (acknowledge via service $4B_{hex}$ “Acknowledge_Under_Voltage”)
		0	No fault detected
14	4000	1	Surge voltage active (voltage is above the specified range)
		0	Voltage OK
15	8000	1	Overvoltage fault detected (acknowledge via service $4C_{hex}$ “Acknowledge_Over_Voltage”)
		0	No fault detected

(101) **Diagnosis status**

A local bus master state is assigned to each bit in the diagnostic status attribute. The states in the error bits (F_PF_BIT, F_BUS_BIT, F_CTRL_BIT) are described in greater detail using the two diagnostic parameter attributes 102 and 103. The diagnostic parameter attributes are always written to if one of the above-mentioned error bits is set. Otherwise, they have the value 0000_{hex}.

Table 1-34 Diagnosis status

Bit	Name	Meaning	
0	F_PW_BIT	I/O warning	The device detected a warning at the I/Os
1	F_PF_BIT	I/O error	The device detected an I/O error (peripheral fault)
2	F_BUS_BIT	Bus error	A bus error occurred.
3	F_CTRL_BIT	Controller error	The driver detected an internal error.
4			Reserved
5	F_RUN_BIT	Run	Data cycles are being exchanged, output data is enabled.
6	F_ACTIVE_BIT	Active	The configuration is active, PDI to the devices is possible, output data is disabled (substitute value behavior).
7	F_READY_BIT	Ready	The local bus master is ready for operation, no data exchange over the bus.
8	F_BD_BIT	Bus different	A device which does not belong to the current configuration has been detected at the last interface.
9	F_BASP_BIT	SYS_FAIL	The controller is in the STOP state or no application program has been loaded. Output data is blocked (substitute value behavior)
10	F_FORCE_BIT	Force mode	Force mode (startup tool) is active.
11 ... 15			Reserved

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

Operating indicators

The Ready, Active, and Run operating indicators show the current state of the system. The diagnostic parameter attributes are not used.

After initialization the driver is ready for operation. The Ready indicator bit is set (F_READY_BIT = 1).

If the driver has been configured and a configuration frame has been activated without errors, the system indicates it is active. The Ready and Active indicator bits are set (F_READY_BIT = 1, F_ACTIVE_BIT = 1).

In addition the Run indicator bit is set (F_READY_BIT = 1, F_ACTIVE_BIT = 1 and F_RUN_BIT = 1) when data exchange is started.

Error indicators

The PF, BUS, and CTRL error indicators report an error, PW reports a warning. Errors which are indicated with BUS or CTRL will cause the bus to be disconnected. The Run indicator bit is reset (F_RUN_BIT = 0).

Further information on the error cause is provided by the two diagnostic parameter attributes.

If several error bits are set to 1 at the same time, the values in the parameter attributes represent the error with the highest priority.

Table 1-35 Error indicators

Message	Priority
CTRL	1 (highest priority)
BUS	F_PF_BIT
PF	F_BUS_BIT
PW	4 (lowest priority)

If there are I/O errors at several devices, the parameter attributes show the message that occurred first. If this message has been removed, the next pending message with the lowest device number is shown.

If there are I/O warnings from several devices, the warnings are shown in the same way as the I/O errors.

After an error has been removed or disappears (e.g., elimination of an interrupt) the bus is started automatically and output data is enabled in the default setting. The Run indicator bit is set again (F_RUN_BIT = 1).

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.12.2 Instance attributes

The error state of a connected module can be respectively determined via the instances of the diagnostics object.

Table 1-36 Axioline F diagnostics object - instance attributes

Attribute	Name	Data type	Access
1	Error count	UINT	Get
2	Priority	USINT	Get
3	Channel/Group/Module	USINT	Get
4	Error code	UINT	Get

1.12.3 Services

Table 1-37 Axioline F diagnostics object - available services

Service code		Class	Instance	Service name
dec	hex			
14	0E	yes	yes	Get_Attribute_Single
75	4B	yes	no	Acknowledge_Under_Voltage
75	4C	yes	no	Acknowledge_Over_Voltage

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.13 PDI object (class code 69_{hex})

The “PDI object” enables access to PDI objects of the Axioline F modules.
You can respectively address a connected module via an instance of the object.

For more detailed information and examples, please refer to Section “PDI objects” on page 39

1.13.1 Class attributes

Table 1-38 PDI object - class attributes

Attribute	Name	Access	Data type	Value
1	Revision	Get	UINT	1
2	Max object instance	Get	UINT	0 ... 63 (number of connected modules)
6	Max class identifier	Get	UINT	7
7	Max instance attribute	Get	UINT	$FFFF_{hex}$

1.13.2 Attributes of instances 1 ... (number of Axioline F modules)

Table 1-39 PDI object - instance attributes of instances 1 ... (number of Axioline F modules)

Attribute	Name	Access	Data type	Value
Index of the PDI object	AXL PDI index	Get/Set	ARRAY_OF_BYTE (256 bytes)	

1.13.3 Services

Table 1-40 PDI object - available services

Service code		Class	Instance	Service name
dec	hex			
14	0E	yes	yes	Get_Attribute_Single
16	10	no	yes	Set_Attribute_Single
75	4B	no	yes	PDI_Read_Object
75	4C	no	yes	PDI_Write_Object

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.14 TCP/IP object (class code: $F5_{hex}$)

1.14.1 Class attributes

Table 1-41 TCP/IP object - class attributes

Attribute	Name	Access	Data type	Value
1	Revision	Get	UINT	3
2	Max instance	Get	UINT	1
3	Number of object instances	Get	UINT	1

1.14.2 Instance attributes

Table 1-42 TCP/IP object - instance attributes

Attribute	Name	Access	Data type	Value
1	Status	Get	DWORD	(1), see 1.14.4 on page 34
2	Configuration capability	Get	DWORD	(2), see 1.14.4 on page 34
3	Configuration control	Get/Set	DWORD	(3), see 1.14.4 on page 34
4	Physical link object	Get	STRUCT OF	00
			UINT	
			Padded EPATH	
5	Interface configuration	Get/Set	STRUCT OF	Dependent upon configuration
	IP address		UDINT	
	Network mask		UDINT	
	Gateway address		UDINT	
	Name server		UDINT	
	Name server 2		UDINT	
	Domain name		STRING	
6	Host name	Get/Set	STRING	Dependent upon configuration
10	SelectAcd	Get/Set	BOOL	Activates/deactivates use of ACD 0 = Activate ACD (default) 1 = Deactivate ACD When the value of SelectAcd is changed by a Set_Attribute service, the new value of SelectAcd shall not be applied until the device executes a restart.
11	Last conflict detection	Get/Set	STRUCT OF	(11), see 1.14.4 on page 34
	AcdActivity		USINT	
	RemoteMAC		Array of 6 USINT	
	ArpPdu		Array of 28 USINT	

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.14.3 Services

Table 1-43 TCP/IP object - available services

Service code		Class	Instance	Service name
dec	hex			
1	01	no	yes	Get_Attribute_All
14	0E	yes	yes	Get_Attribute_Single
16	10	no	yes	Set_Attribute_Single

1.14.4 Values of the instance attributes

(1) Status

Bit	Call	Definition
0 ... 3	Interface configuration status	0 = Not configured 1 = Valid configuration 2 ...15 = Reserved
4	Mcast pending	Not used.
5	Interface configuration pending	1 = Change to parameters in the “Interface configuration” attribute
6	AcdStatus	1 = IP address conflict
7	AcdFault	1 = IP address conflict
8 ... 31	Reserved	Reserved (set to 0)

(2) Configuration capability

This attribute indicates whether the device has optional network capabilities.

Bit	Call	Definition
0	BootP client	1 = Device can receive its configuration via BootP.
1	DNS client	0 = Not supported
2	DHCP client	1 = Device can receive the network configuration via DHCP
3	DHCP DNS update	1 = Device can send its name in a DHCP request
4	Configuration settable	1 = “Interface configuration” attribute can be set
5	Hardware configurable	1 = IP address can be set via a rotary coding switch
6	Interface configuration change requires reset	1 = Each change to the “Interface configuration” attribute is only applied following a reset
7	AcdCapable	1 = Device has ACD functionality
8 ... 31	Reserved	Reserved (set to 0)

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

(3) **Configuration control**

This attribute directs the device as to how it should determine its own network configuration.

Bit	Call	Definition
0 ... 3	Startup configuration	0 = Static IP configuration 1 = BootP 2 = DHCP 3 ... 15 = Reserved
4	DNS enable	Set to 0, as not supported.
5 ... 31	Reserved	Set to 0.

(11) **Last conflict detection**

The “LastConflictDetected” attribute is a diagnostic attribute presenting information about the ACD state when the last IP Address conflict was detected. This attribute shall be updated by the device whenever an incoming ARP packet is received that represents a conflict with the device's IP address as described in IETF RFC 5227.

To reset this attribute, the Set_Attribute_Single service is called by an attribute value which contains zeros. Use of values other than 0 leads to an error reaction (status code 09_{hex}, Invalid Attribute Value).

AcdActivity

The ACD contains the state of the ACD algorithm when the last IP address conflict was detected. The ACD activities are defined in the following table.

Value	ACD mode	Description
0	NoConflictDetected (default)	No conflict has been detected since this attribute was last cleared.
1	ProbeIpv4Address	Last conflict detected during the ProbeIpv4Address state.
2	OngoingDetection	Last conflict detected during OngoingDetection state or subsequent DefendWithPolicyB state.
3	SemiActiveProbe	Last conflict detected during SemiActiveProbe state or subsequent DefendWithPolicyB state.

RemoteMac

The IEEE 802.3-compliant MAC source address from the header of the received Ethernet packet, which was sent by a device which reports a conflict.

ArpPdu

The ARP Response PDU in binary format.

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

1.15 Ethernet link object (class code: $F6_{hex}$)

1.15.1 Class attributes

Table 1-44 Ethernet link object - class attributes

Attribute	Name	Access	Data type	Value
1	Revision	Get	UINT	3
2	Max instance	Get	UINT	3
3	Number of object instances	Get	UINT	3
6	Max class identifier	Get	UINT	7
7	Max instance attribute	Get	UINT	10

1.15.2 Instance attributes

Table 1-45 Ethernet link object - instance attributes

Attri-bute	Name	Acces s	Data type	Description	
1	Interface speed	Get	UDINT	Speed of the interface in Mbps If an interface is operated with 100 Mbps, the value of the attribute must be 100. The attribute is used to display the media bandwidth. If the interface is operated in full duplex mode, the attribute must not be duplicated.	
2	Interface flags	Get	DWORD	Bit 0	Link status This bit indicates whether an Ethernet-802.3 communication port is connected to an active network or not. 0 = Inactive link 1 = Active link
				Bit 1	Half/full duplex status 0 = Half duplex 1 = Full duplex Note that the value of the half/full duplex flag is undetermined if the link status flag is set to 0.
				Bit 2 ... 31	Not supported (set to 0)
3	Physical address	Get	ARRAY of 6 USINTs	MAC layer address of the interface The “Physical address” attribute is an array of octaves. The recommended display format is "XX-XX-XX-XX-XX-XX", starting with the first octave. Please note that the “Physical address” attribute cannot be set. The Ethernet address is to be set by the manufacturer and must be unique according to the requirements of IEEE 802.3.	

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

Table 1-45 Ethernet link object - instance attributes

Attri-bute	Name	Acces s	Data type	Description	
6	Interface control	Get/Set	STRUC OF	Configuration of the physical interface	
	Control bits		WORD	Interface control bits	
				Bit 0	Auto negotiation (set) 0 = Auto negotiation disabled 1 = Auto negotiation enabled
				Bit 1	Forced duplex mode (set) 0 = Half duplex 1 = Full duplex
				Bit 2 ... Bit 15	Reserved (set to 0)
	Forced interface speed		UINT	If the “Auto-negotiation” bit is not set, use the “Forced interface speed” bit to set the required transmission speed. 10 = 10 Mbps 100 = 100 Mbps	
7	Interface type	Get	USINT	Type of physical interface 2 = Twisted pair 10/100 base-T (permanent setting)	
10	Interface label	Get/Set	SHORT_ STRING	Interface description Instance 1 = X1 Instance 2 = X2 Instance 3 = internal	

1.15.3 Services

Table 1-46 Ethernet link object - available services

Service code		Class	Instance	Service name
dec	hex			
16	10	yes	no	Set_Attribute_Single
14	0E	yes	yes	Get_Attribute_Single

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

2 PDI objects

2.1 Access to PDI objects

PDI stands for parameter, diagnostics, and information. The PDI channel is used in addition to the process data channel in the Axioline F system for the demand-oriented, acyclic transmission of parameter and diagnostic data as well as other information. Each Axioline F device has this channel and can use it independently of the process data.

Objects created in the Axioline F device can be accessed via the PDI channel using services. These objects can be used, for example, to set measuring ranges, to specify the substitute value behavior of outputs in the event of a bus error or to read diagnostic details.

The objects are addressed via an object index (e.g., 0018_{hex}: DiagState). For detailed information on the objects present on a module, please refer to the module-specific documentation.

For direct access to the acyclic PDI channel, object class 69_{hex} is defined, in which a connected module can be respectively addressed via an instance of this object.

Two access versions are available.
The first access version is very easy to use; the length is automatically calculated. Access is via service codes $0E_{hex}$ and 10_{hex}. In this case, access to the subindex and subslot is not supported.

The second access version involves extended access via service code $4B_{hex}$ and $4C_{hex}$. In this case, there are no restrictions with regard to addressing submodules.

2.1.1 Structure of the simple access procedure

- Object class 69_{hex}
- Instance: Slot No.
- Attribute: PDI object index (e.g., 0018_{hex})
- Service $0E_{hex}$: Read object
- Service 10_{hex}: Write object

Advantage:	This access version is very easy to use; the length, etc. is automatically calculated.
Disadvantage:	Access to the subindex and subslot is not supported.

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

2.1.2 Structure of the extended access procedure (from firmware version 1.20)

– Object class 69_{hex}
– Instance: Slot No.
– Attribute: -
– Service $4B_{hex}$: Read object (with full access to PDI header)
– Service $4C_{hex}$: Write object (with full access to PDI header)

Advantage:	There are no restrictions with regard to addressing.
Disadvantage:	More effort required by the user.

Notes on services $4B_{hex}$ and $4C_{hex}$

The PDI service is fully integrated into CIP. A status other than 00_{hex} (success) is only output if the syntax of the CIP service does not match (e.g., incorrect instance).

If an error occurs on a PDI level (e.g., object not available), the status is 00_{hex} and the PDI error is reported in “Error Class”, “Error Code”, and “Additional Code”.

2.1.3 Structure of the PDI_Read_Object service ($4B_{hex}$)

This service is used to read a PDI object to a connected Axioline F device.

Table 2-1 PDI object - PDI_Read_Object service (request)

Service request parameters			
Name	**Data type**	**Description**	**Value range**
Subslot	USINT (1 byte)	Subslot number	00_{hex} ... FF_{hex}
Reserved	USINT (1 byte)		00_{hex}
Index	UINT (2 bytes)	PDI object index	0000_{hex} ... $FFFF_{hex}$
Subindex	USINT (1 byte)	PDI object subindex	0000_{hex} ... FF_{hex}

Table 2-2 PDI object - PDI_Read_Object service (positive response)

Positive response			
Name	**Data type**	**Description**	**Value range**
Subslot	USINT (1 byte)	Subslot number	00_{hex} ... FF_{hex}
Reserved	USINT (1 byte)		00_{hex}
Index	UINT (2 bytes)	PDI object index from request	0000_{hex} ... $FFFF_{hex}$
Subindex	USINT (1 byte)	PDI object subindex from request	00_{hex} ... FF_{hex}
Error class	USINT (1 byte)	Error class (see basic profile)	00_{hex}

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

Table 2-2 PDI object - PDI_Read_Object service (positive response)

Positive response			
Name	**Data type**	**Description**	**Value range**
Error code	USINT (1 byte)	Error code (see basic profile)	00_{hex}
Data length	USINT (1 byte)	Number of subsequent data bytes (PDI object length)	00_{hex} ... FF_{hex}
PDI object data	ARRAY OF USINT	Data content of the PDI object	

Table 2-3 PDI object - PDI_Read_Object service (negative response)

Negative response			
Name	**Data type**	**Description**	**Value range**
Subslot	USINT (1 byte)	Subslot number	00_{hex} … FF_{hex}
Reserved	USINT (1 byte)		00_{hex}
Index	UINT (2 bytes)	PDI object index from request	0000_{hex} … $FFFF_{hex}$
Subindex	USINT (1 byte)	PDI object subindex from request	00_{hex} … FF_{hex}
Error class	USINT (1 byte)	Error class (see basic profile)	00_{hex}
Error code	USINT (1 byte)	Error code (see basic profile)	00_{hex}
Reserved	USINT (1 byte)	-	00_{hex}
Additional code	UINT (2 bytes)	Additional code (see basic profile)	0000_{hex} … $FFFF_{hex}$

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

2.1.4 Structure of the PDI_Write_Object ($4C_{hex}$) service

This service is used to write a PDI object to a connected Axioline F device.

Table 2-4 PDI object - PDI_Write_Object service (request)

Service request parameters			
Name	**Data type**	**Description**	**Value range**
Subslot	USINT (1 byte)	Subslot number	00_{hex} ... FF_{hex}
Reserved	USINT (1 byte)		00_{hex}
Index	UINT (2 bytes)	PDI object index	0000_{hex} ... $FFFF_{hex}$
Subindex	USINT (1 byte)	PDI object subindex	00_{hex} ... FF_{hex}
Data length	USINT (1 byte)	Number of subsequent data bytes (PDI object length)	00_{hex} … FF_{hex}
PDI object data	ARRAY OF USINT	PDI object data (quantity dependent on “data length”)	

Table 2-5 PDI object - PDI_Write_Object service (positive response)

Positive response			
Name	**Data type**	**Description**	**Value range**
Subslot	USINT (1 byte)	Subslot number	00_{hex} … FF_{hex}
Reserved	USINT (1 byte)		00_{hex}
Index	UINT (2 bytes)	PDI object index from request	0000_{hex} … $FFFF_{hex}$
Subindex	USINT (1 byte)	PDI object subindex from request	00_{hex} … FF_{hex}
Error class	USINT (1 byte)	Error class (see basic profile)	00_{hex}
Error code	USINT (1 byte)	Error code (see basic profile)	00_{hex}
Reserved	USINT (1 byte)	-	00_{hex}

Table 2-6 PDI object - PDI_Write_Object service (negative response)

Negative response			
Name	**Data type**	**Description**	**Value range**
Subslot	USINT (1 byte)	Subslot number	00_{hex} … FF_{hex}
Reserved	USINT (1 byte)		00_{hex}
Index	UINT (2 bytes)	PDI object index from request	0000_{hex} … $FFFF_{hex}$
Subindex	USINT (1 byte)	PDI object subindex from request	00_{hex} … FF_{hex}
Error class	USINT (1 byte)	Error class (see basic profile)	00_{hex}
Error code	USINT (1 byte)	Error code (see basic profile)	00_{hex}
Reserved	USINT (1 byte)	-	00_{hex}
Additional code	UINT (2 bytes)	Additional code (see basic profile)	0000_{hex} … $FFFF_{hex}$

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

2.1.4.1 Parameter descriptions

Subslot	Specify a subslot if you wish to access a submodule (e.g., IO-Link). Not used at present (= 0).
PDI object index	See module-specific data sheet.
PDI object subindex	See module-specific data sheet.
Error class, error code	0000_{hex}: No error; $\neq 0000_{hex}$: An error occurred; negative response with error message ID
Additional code	More detailed information on the cause of the error. Should an error occur, the error message details comprise the error class, error code, and additional code.

For the meaning of this error code, please refer to the AXL F SYS DIAG user manual.

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

2.2 Examples of simple access

2.2.1 Read PDI object

Read the order number of the first module.

Read request (representation as CIP protocol)

Byte	Content (hex)	Parameter	Meaning
0	0E	Service	Read object
1	03	Path length	Three parameters follow
2	20	Segment class ID	Object class path
3	69	Class ID	PDI object
4	24	Segment instance ID	Instance path
5	01	Instance ID	Slot number of the first module
6	30(31)	Segment attribute ID	Path to attribute; depending on the PDI index, the length of the path can be 1 or 2 bytes (e.g., 31 5F FF).
7	0A	Attribute ID	PDI object index: order number (000A)

Response

The response always contains an even byte count. A byte with 00hex is added if necessary in order to round up to an even byte number.

- Positive response

Byte	Content (hex)	Parameter	Meaning
0	8E	Service	Response from read object service
1	00	Reserved	Fill byte
2	00	General status	No errors
3	00	Extended status size	Length of extended status
4 … 11	32 36 38 38 31 36 31 00	Read data	Order number (8 bytes, including zero termination) e.g., 2688161

– Negative response

Byte	Content (hex)	Parameter	Meaning
0	8E	Service	Response from read object service
1	00	Reserved	Fill byte
2	XX	General status	CIP-specific error message
3	00	Extended status size	Length of extended status

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

2.2.2 Write PDI object

All process data channels of the AXL F AI4 U 1H module should be parameterized.
In the physical bus configuration, the module is the eighth module.
The parameterization is performed via the ParaTable object (0080_{hex}).

Write request (representation as CIP protocol)

Byte	Content (hex)	Parameter	Meaning
0	10	Service	Write object
1	03	Path length	Three path parameters follow
2	20	Segment class ID	Object class path
3	69	Class ID	PDI object
4	24	Segment instance ID	Instance path
5	08	Instance ID	Slot number of the ninth module
6	30	Segment attribute ID	Path to attribute; depending on the PDI index, the length of the path can be 1 or 2 bytes (e.g., 31 FF 8D).
7	80	Attribute ID	PDI object index: Para-Table (parameter table)
8	00	Data byte 0	According to the module-specific data sheet: For each of the four channels: – Filter 30 Hz – 16-sample mean value – Measuring range 0 V... 5 V
9	02	Data byte 1	
10	00	Data byte 2	
11	02	Data byte 3	
12	00	Data byte 4	
13	02	Data byte 5	
14	00	Data byte 6	
15	02	Data byte 7	
16	00	Data byte 8	According to the module-specific data sheet: Data format IB IL
17	00	Data byte 9	
18	00	Data byte 10	According to the module-specific data sheet: Reserved
19	00	Data byte 11	

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

Response

- Positive response

Byte	Content (hex)	Parameter	Meaning
0	90	Service	Read response from object service
1	00	Reserved	Fill byte
2	00	General status	No errors
3	00	Extended status size	Length of extended status

– Negative response

Byte	Content (hex)	Parameter	Meaning
0	90	Service	Response from “read object” service
1	00	Reserved	Fill byte
2	1E	General status	CIP-specific error message
3	00	Extended status size	Length of extended status

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

2.3 Examples of extended access (from firmware version 1.20)

2.3.1 Read PDI object

Read the order number of the first module.

Read request (representation as CIP protocol)

Byte	Content (hex)	Parameter	Meaning
0	4B	Service	Read object
1	03	Path length	Three parameters follow
2	20	Segment class ID	Object class path
3	69	Class ID	PDI object
4	24	Segment instance ID	Instance path
5	01	Instance ID	Slot number of the first module
6	30(31)	Segment attribute ID	Attribute path
7	01	Attribute ID	Always set the attribute ID to 01.
8	00	Data byte 0	Subslot
9	00	Data byte 1	Reserved
10	00	Data byte 2	Index high
11	0A	Data byte 3	Index low
12	00	Data byte 4	Subindex

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

Response

- Positive response

Byte	Content (hex)	Parameter	Meaning
0	CB	Service	Read object
1	00	Reserved	Fill byte
2	00	General status	No errors
3	00	Extended status size	Length of extended status
4	00	Data byte 0	Subslot
5	00	Data byte 1	Reserved
6	00	Data byte 2	Index high
7	0A	Data byte 3	Index low
8	00	Data byte 4	Subindex
9	00	Data byte 5	Error class
10	00	Data byte 6	Error code
11	08	Data byte 7	Data length
12 … 19	32 36 38 38 31 36 31 00	Read data	Order number (8 bytes, including zero termination) e.g., 2688161

– Negative response

Byte	Content (hex)	Parameter	Meaning
0	CB	Service	Read object
1	00	Reserved	Fill byte
2	00	General status	No errors
3	00	Reserved	Fill byte
4	00	Data byte 0	Subslot
5	00	Data byte 1	Reserved
6	00	Data byte 2	Index high
7	80	Data byte 3	Index low
8	00	Data byte 4	Subindex
9	00	Data byte 5	Error class
10	00	Data byte 6	Error code
11	00	Data byte 7	Reserved
12	00	Data byte 8	Additional code
13	00	Data byte 9	Additional code

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

2.3.2 Write PDI object

Write request (representation as CIP protocol)

Byte	Content (hex)	Parameter	Meaning
0	4C	Service	Read object
1	03	Path length	Three parameters follow
2	20	Segment class ID	Object class path
3	69	Class ID	PDI object
4	24	Segment instance ID	Instance path
5	01	Instance ID	Slot number of the first module
6	30	Segment attribute ID	Attribute path
7	01	Attribute ID	Always set the attribute ID to 01.
8	00	Data byte 0	Subslot
9	00	Data byte 1	Reserved
10	00	Data byte 2	Index high
11	80	Data byte 3	Index low
12	00	Data byte 4	Subindex
13	0C	Data byte 5	Data length
8	00	Data byte 6	According to the module-specific data sheet: For each of the four channels: – Filter 30 Hz – 16-sample mean value – Measuring range 0 V... 5 V
9	02	Data byte 7	
10	00	Data byte 8	
11	02	Data byte 9	
12	00	Data byte 10	
13	02	Data byte 11	
14	00	Data byte 12	
15	02	Data byte 13	
16	00	Data byte 14	According to the module-specific data sheet: Data format IB IL
17	00	Data byte 15	
18	00	Data byte 16	According to the module-specific data sheet: Reserved
19	00	Data byte 17	

DOWNLOADED FROM WWW.SCATTS.CO.UK

DOWNLOADED FROM WWW.SCATTS.CO.UK

Response

- Positive response

Byte	Content (hex)	Parameter	Meaning
0	CB	Service	Read object
1	00	Reserved	Fill byte
2	00	General status	No errors
3	00	Extended status size	Length of extended status
4	00	Data byte 0	Subslot
5	00	Data byte 1	Reserved
6	00	Data byte 2	Index high
7	80	Data byte 3	Index low
8	00	Data byte 4	Subindex
9	00	Data byte 5	Error class
10	00	Data byte 6	Error code
11	00	Data byte 7	Reserved

– Negative response

Byte	Content (hex)	Parameter	Meaning
0	CC	Service	Read object
1	00	Reserved	Fill byte
2	00	General status	No errors
3	00	Extended status size	Length of extended status
4	00	Data byte 0	Subslot
5	00	Data byte 1	Reserved
6	00	Data byte 2	Index high
7	80	Data byte 3	Index low
8	00	Data byte 4	Subindex
9	00	Data byte 5	Error class
10	00	Data byte 6	Error code
11	00	Data byte 7	Reserved
12	00	Data byte 8	Additional code high byte
13	00	Data byte 9	Additional code low byte

DOWNLOADED FROM WWW.SCATTS.CO.UK

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

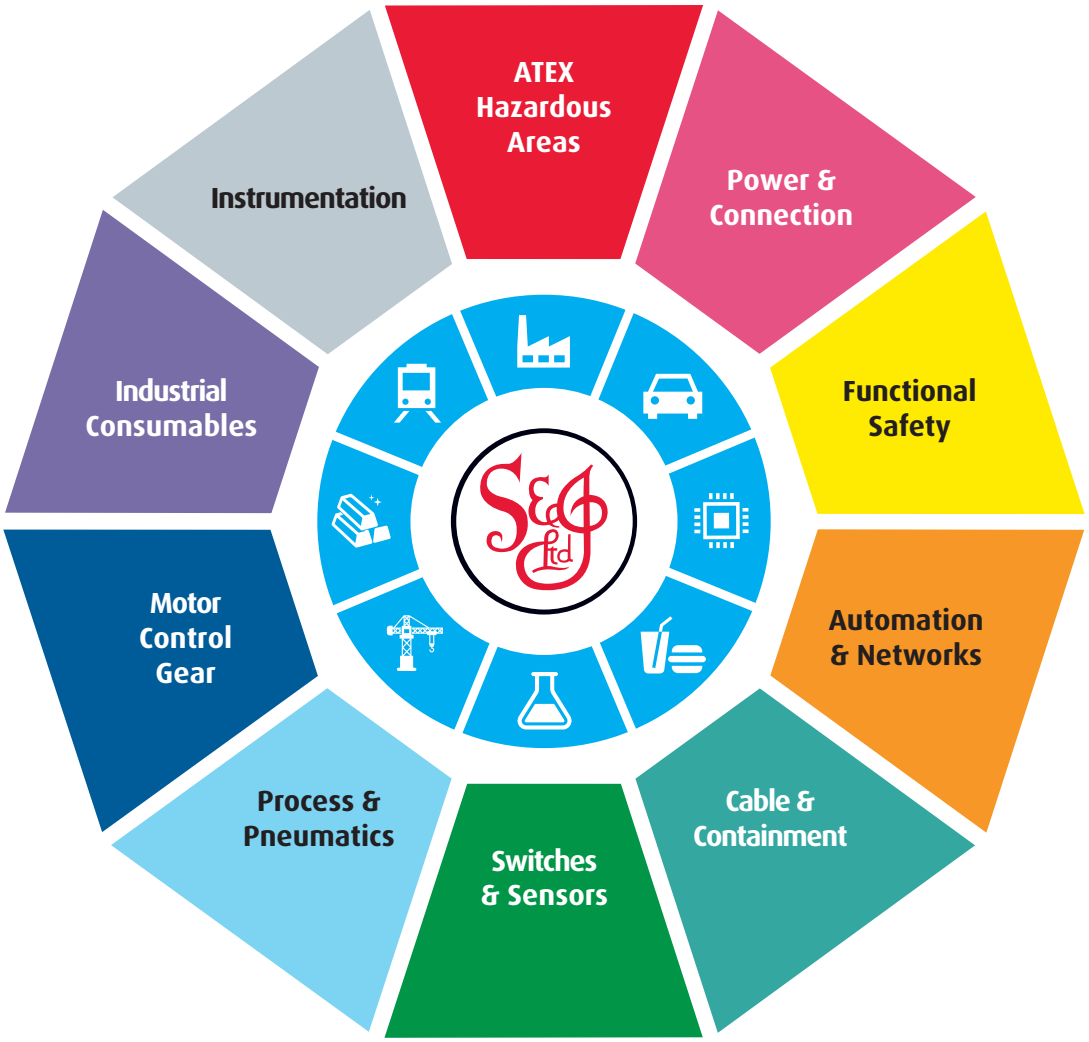
At Scattergood & Johnson Ltd, we pride ourselves on being a technical distributor to specialist industries.

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