The Drive & Control Company

Rexroth
Bosch Group

IndraControl S20 bus coupler

for EtherNet/IP™ S20-EIP-BK

Application Description
R911377106

Edition 02

Title IndraControl
S20 bus coupler
for EtherNet/IP™ S20-EIP-BK

Type of Documentation Application Description

Document Typecode DOK-CONTRL-S20*EIP*BK*-AP02-EN-P

Internal File Reference 107495_en_01, R911377106_02.pdf

Record of revision

Edition	Release date	Note
02	2016-10	First edition

Editorial department Engineering automation systems control hardware, SB

Table of contents

Table of contents

1 Use of the safety instructions

1.1 Structure of the safety instructions

The safety instructions are structured as follows:

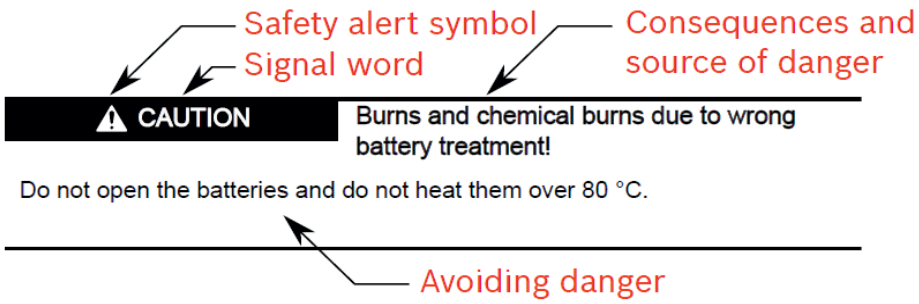

Abb. 1-1 Structure of the safety instructions

1.2 Explaining signal words and safety alert symbol

The safety instructions in this documentation contain specific signal words (danger, warning, caution, notice) and, if necessary, a safety alert symbol (according to ANSI Z535.6-2006).

The signal word is used to draw attention to the safety instruction and also provides information on the severity of the hazard.

The safety alert symbol (a triangle with an exclamation point), which precedes the signal words danger, warning and caution is used to alert the reader to personal injury hazards.

⚠ DANGER

In case of non-compliance with this safety instruction, death or serious injury **will** occur.

⚠ WARNING

In case of non-compliance with this safety instruction, death or serious injury **can** occur.

⚠ CAUTION

In case of non-compliance with this safety instruction, minor or moderate injury can occur.

NOTICE

In case of non-compliance with this safety instruction, material damage can occur.

Use of the safety instructions

1.3 Symbols used

Hints are represented as follows:

This is an information.

Tips are represented as follows:

This is a tip for the user.

1.4 Signal graphic explanation on the device

Prior to the installation and commissioning of the device, refer to the device documentation.

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

2.1 General information

The bus coupler maps the I/O devices 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.

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).

Fig. 2-1 Connection types

2.2 CIP class and instance services

The module supports the following class 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

Fig. 2-2 Supported class and instance services

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

2.3 CIP object classes

The module supports the following CIP object classes:

Class code		Object type	On page
dec	hex		
01	01	Identity object	7
02	02	Router object	9
04	04	Assembly object	10
55	37	File object	11
100	64	Configuration object	14
102	66	Module object	17
103	67	Diagnostics object	19
105	69	PDI object	22
245	F5	TCP/IP interface object	23
246	F6	Ethernet link object	26

Fig. 2-3 Supported CIP object classes

2.4 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.

2.4.1 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

Fig. 2-4 Identity object - class attributes

2.4.2 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

Fig. 2-5 Identity object - instance attributes

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

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
				Bits 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 fault 1 = Minor recoverable fault (e.g., short circuit of an output)
				Bit 9	Minor unrecoverable fault	0 = No fault 1 = Minor unrecoverable fault
				Bit 10	Major recoverable fault	0 = No fault 1 = Major recoverable fault (e.g., loss of +24 V DC)
				Bit 11	Major unrecoverable fault	0 = No fault 1 = Major non-recoverable fault (e.g., checksum, A/D)
				Bits 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 Bosch Rexroth product groups.		
7	Product name	Get	STRUCT OF			
	Length		USIGN	12		
	Name		STRING	S20-EIP-BK		

Fig. 2-5 Identity object - instance attributes [...]

2.4.3 Services

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

Fig. 2-6 Identity object - available services

2.5 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.

2.5.1 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

Fig. 2-7 Router object - class attributes

2.5.2 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

Fig. 2-8 Router object - instance attributes

2.5.3 Services

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

Fig. 2-9 Router object - available services

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

2.6 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.

2.6.1 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

Fig. 2-10 Assembly object - class attributes

2.6.2 Instance attributes

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

Fig. 2-11 Assembly object - instance attributes (instance 110)

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

2.6.3 Services

Service code		Class	Instance	Service name
dec	hex			
14	0E	Yes	Yes	Get_Attribute_Single
16	10	No	Yes	Set_Attribute_Single

Fig. 2-12 Assembly object - available services

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

2.7.1 Class attributes

Attribute	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, which are assigned to this device and the corresponding “Instance number”
	Instance number		UINT	(1), see 2.7.5 on page 13
	Instance name		STRING	
	File name		STRING	

Fig. 2-13 File object - class attributes

2.7.2 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

Fig. 2-14 File object - available class services

The optional Create/Delete services are not supported.

2.7.3 Instance attributes

Attribute	Name	Access	Data type	Value/description
1	State	Get	USINT	(2), see 2.7.5 on page 13
2	Instance name	Get	STRING	(1), see 2.7.5 on page 13
3	File format version	UINT		(1), see 2.7.5 on page 13
4	File name	Get		(1), see 2.7.5 on page 13
5	File revision	Get	STRUCT OF	(1), see 2.7.5 on page 13
	Major revision		USINT	
	Minor revision		USINT	
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 2.7.5 on page 13

Fig. 2-15 File object - instance attributes

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

Attribute	Name	Access	Data type	Value/description
9	File save parameters	Get	BYTE	0
10	File access rule	Get/set	USINT	(1), see 2.7.5 on page 13 0 = Access rights = Read/write 1 = Access right = Read only
11	File encoding format	Get	USINT	(1), see 2.7.5 on page 13 0 = No encryption 1 = Encryption

Fig. 2-15 File object - instance attributes [...]

2.7.4 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 by a client 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.
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 checksums 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.

Fig. 2-16 File object - available instance services

2.7.5 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	Upload started
4	Download started
5	Upload running
6	Download running
7	File saving
8 ... 255	Reserved

(3) **Invocation method**

The value defines what should happen once the file has been downloaded. Potential options include:

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

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

2.8 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.

2.8.1 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

Fig. 2-17 Configuration object - class attributes

2.8.2 Instance attributes

Attri-bute	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 S20 diagnostic register, as in the diagnostics object the 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 2.8.4 on page 16
7	Confirm startup parametrization	Set	USINT	1 = Acknowledge startup parameterization (7), see 2.8.4 on page 16
8	P&P mode	Get/Set	UINT	Select Plug and Play mode 0 = Disabled 1 = Enabled
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)	S20-EIP-BK

Fig. 2-18 Configuration object - instance attributes

Attri-bute	Name	Access	Data type	Value/note
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)	xx xx xx xx xx xx xx x
22	MAC address	Get	ARRAY_OF_BYTE (17 bytes)	00:60:34:XX:XX:XX
23	Material number	Get	ARRAY_OF_BYTE (19 bytes)	R911173904
24	Order code	Get	ARRAY_OF_BYTE (30 bytes)	S20-EIP-BK
25	Manufacturer name	Get	ARRAY_OF_BYTE (19 bytes)	Bosch Rexroth AG
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.

Fig. 2-18 Configuration object - instance attributes [...]

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

2.8.3 Services

Service code		Class	Instance	Service name
dec	hex			
14	0E	Yes	Yes	Get_Attribute_Single
16	10	No	Yes	Set_Attribute_Single

Fig. 2-19 Configuration object - available services

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

2.8.4 Values of the instance attributes

(6) HW Diag Ctrl

Reset the UL monitor. A monitoring function monitors the supply of the communications power U_L. 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 parametrization

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

2.9 Module object (class code 66_{hex})

The “Module object” allows you to monitor the S20 modules connected to the bus coupler.

2.9.1 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 diagnostics 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.

Attribute	Name	Access	Data type	Value/note
1	Revision	Get	UINT	1
2	Max object instance	Get	UINT	Number of S20 modules
6	Max class identifier	Get	UINT	101
7	Max instance attribute	Get	UINT	4
100	Config array DeviceType code	Get	ARRAY_OF_USINT	Number of S20 modules * 8 bytes (100), see 2.9.3 on page 18
101	Config array PD length	Get	ARRAY_OF_USINT	Number of S20 modules * 1 byte (101), see 2.9.3 on page 18

Fig. 2-20 Module object - class attributes

2.9.2 Attributes of instances 1 ... (number of S20 modules)

Attribute	Name	Access	Data type	Value
1	DeviceType code (config)	Get	ARRAY_OF_USINT (8 bytes)	Configured “DeviceType”
2	PD length, number of bytes (config)	Get	USINT	Configured process data length in bytes
3	DeviceType code (current)	Get	ARRAY_OF_USINT (8 bytes)	Current “DeviceType”
4	PD length, number of bytes (current)	Get	USINT	Current process data length in bytes

Fig. 2-21 Module object - instance attributes of instances 1 ... (number of S20 modules)

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

Services

Service code		Class	Instance	Service name
dec	hex			
14	0E	Yes	Yes	Get_Attribute_Single
16	10	No	No	Set_Attribute_Single

Fig. 2-22 Module object - available services

2.9.3 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 material 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).

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

Fig. 2-23 Config array device type code

(101) Config array PD length

This class attribute specifies the number of bytes assigned to 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.

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

Fig. 2-24 Config array PD length

2.10 Diagnostics object (class code 67_{hex})

2.10.1 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 19
101	Diagnostic status	UINT	Get	(101), see “Diagnostic status” on page 20
102	Diagnostic parameter 1	UINT	Get	
103	Diagnostic parameter 2	UINT	Get	

Fig. 2-25 Diagnostics object - class attributes

(100) 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
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

Fig. 2-26 General status

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

Bit	Code (hex)	Meaning	
14	4000	1	Surge voltage active (voltage is above the specified range)
		0	Voltage OK
15	8000	1	Surge voltage fault detected (acknowledge via service $4C_{hex}$ “Acknowledge_Over_Voltage”)
		0	No error detected

Fig. 2-26 General status [...]

(101) Diagnostic 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}.

Bit	Name	Meaning	
0	F_PW_BIT	I/O warning	The device detected a warning at the I/Os (PW = peripheral warning).
1	F_PF_BIT	I/O error	The device detected an I/O error (PF = peripheral fault).
2	F_BUS_BIT	Bus error	A bus error has 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

Fig. 2-27 Diagnostic status

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.

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

Fig. 2-28 Error indicators

If there are I/O errors (PF = peripheral fault) 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 (PW = peripheral warning) 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).

2.10.2 Instance attributes

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

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

Fig. 2-29 Diagnostics object - instance attributes

2.10.3 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

Fig. 2-30 Diagnostics object - available services

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

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

The “PDI object” enables access to PDI objects of the S20 modules.
You can respectively address a connected module via an instance of the object.

For more detailed access information and examples, please refer to Chapter “PDI objects” on page 29.

2.11.1 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	$FFFF_{hex}$

Fig. 2-31 PDI object - class attributes

2.11.2 Attributes of instances 1 ... (number of S20 modules)

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

Fig. 2-32 PDI object - instance attributes of instances 1 ... (number of S20 modules)

2.11.3 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

Fig. 2-33 PDI object - available services

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

2.12.1 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

Fig. 2-34 TCP/IP object - class attributes

2.12.2 Instance attributes

Attribute	Name	Access	Data type	Value
1	Status	Get	DWORD	(1), see 2.12.4 on page 24
2	Configuration capability	Get	DWORD	(2), see 2.12.4 on page 24
3	Configuration control	Get/Set	DWORD	(3), see 2.12.4 on page 24
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 2.12.4 on page 24
	AcdActivity		USINT	
	RemoteMAC		Array of 6 USINT	
	ArpPdu		Array of 28 USINT	

Fig. 2-35 TCP/IP object - instance attributes

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

2.12.3 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

Fig. 2-36 TCP/IP object - available services

2.12.4 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 host 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)

(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 information on 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.

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

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

2.13.1 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

Fig. 2-37 Ethernet link object - class attributes

2.13.2 Instance attributes

Attribute	Name	Access	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 interface 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.
				Bits 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 octets. The recommended display format is “XX-XX-XX-XX-XX-XX”, starting with the first octet. 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.	

Fig. 2-38 Ethernet link object - instance attributes

Attri-bute	Name	Acces s	Data type	Description	
6	Interface control	Get/set	STRUC OF	Configuration of 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 lable	Get/set	SHORT_ STRING	Interface description Instance 1 = X1 Instance 2 = X2 Instance 3 = Internal	

Fig. 2-38 Ethernet link object - instance attributes [...]

2.13.3 Services

Service code		Class	Instance	Service name
dec	hex			
16	10	Yes	No	Set_Attribute_Single
14	0E	Yes	Yes	Get_Attribute_Single

Fig. 2-39 Ethernet link object - available services

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

3 PDI objects

3.1 Access to PDI objects

PDI stands for parameters, diagnostics, and information. The PDI channel is used in addition to the process data channel in the S20 system for the demand-oriented, acyclic transmission of parameter and diagnostic data as well as other information. Each S20 device has this channel and can use it independently of the process data.

Objects created in the S20 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. 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.

3.1.1 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.

PDI objects

3.1.2 Structure of the extended access procedure (as of 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”.

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

This service is used to read a PDI object to a connected S20 device.

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}

Fig. 3-1 PDI object - PDI_Read_Object service (request)

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}
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	

Fig. 3-2 PDI object - PDI_Read_Object service (positive 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} … $FFFF_{hex}$

Fig. 3-3 PDI object - PDI_Read_Object service (negative response)

PDI objects

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

This service is used to write a PDI object to a connected S20 device.

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”)	

Fig. 3-4 PDI object - PDI_Write_Object service (request)

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}

Fig. 3-5 PDI object - PDI_Write_Object service (positive 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} … $FFFF_{hex}$

Fig. 3-6 PDI object - PDI_Write_Object service (negative response)

3.1.4.1 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 meanings of the error codes, please refer to the DOK-CONTRL-S20*DIAG*ER-AP..-EN-P application description, MNR R911344826.

PDI objects

3.2 Examples of simple access

3.2.1 Read PDI object

Read the material 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: material number (000A)

Response

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

- Positive response

Byte	Content [hex]	Parameter	Meaning
0	8E	Service	Response from Read object service
1	00	Reserved	Filler 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	Material 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	Filler byte
2	XX	General status	CIP-specific error message
3	00	Extended status size	Length of extended status

3.2.2 Write PDI object

All process data channels of the S20-AI-4-U 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: IB IL data format
17	00	Data byte 9	
18	00	Data byte 10	According to the module-specific data sheet: reserved
19	00	Data byte 11	

PDI objects

Response

- Positive response

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

3.3 Examples of extended access (as of firmware version 1.20)

3.3.1 Read PDI object

Read the material 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

PDI objects

Response

- Positive response

Byte	Content [hex]	Parameter	Meaning
0	CB	Service	Read object
1	00	Reserved	Filler 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	Material number (8 bytes, including zero termination) e.g., 2688161

– Negative response

Byte	Content [hex]	Parameter	Meaning
0	CB	Service	Read object
1	00	Reserved	Filler byte
2	00	General status	No errors
3	00	Reserved	Filler 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

3.3.2 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: IB IL data format
17	00	Data byte 15	
18	00	Data byte 16	According to the module-specific data sheet: reserved
19	00	Data byte 17	

PDI objects

Response

- Positive response

Byte	Content [hex]	Parameter	Meaning
0	CB	Service	Read object
1	00	Reserved	Filler 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	Filler 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

4 Disposal

4.1 General information

Dispose the products according to the respective valid national standard.

4.2 Return

For disposal, our products can be returned free of charge. However, the products must be free of remains like oil and grease or other impurities.

Furthermore, the products returned for disposal must not contain any undue foreign substances or components.

Send the products free of charge to the following address:

Bosch Rexroth AG
Electric Drives and Controls
Bürgermeister-Dr.-Nebel-Straße 2
D-97816 Lohr am Main, Germany

4.3 Packaging

The packaging material consists of cardboard, plastics, wood or styrofoam. Packaging material can be recycled anywhere.

For ecological reasons, please do not return empty packages.

4.4 Batteries and accumulators

Batteries and accumulators can be labelled with this symbol.

The symbol indicating "separate collection" for all batteries and accumulators is the crossed-out wheeled bin.

The end user within the EU is legally obligated to return used batteries. Outside the validity of the EU Directive 2006/66/EC keep the stipulated directives.

Used batteries can contain hazardous substances, which can harm the environment or the health of the individual when they are stored incorrectly or disposed of.

After use, the batteries or accumulators contained in Rexroth products have to be disposed of according to the country-specific collection system.

Disposal

5 Service and support

Our worldwide service network provides an optimized and efficient support. Our experts offer you advice and assistance should you have any queries.
You can contact us **24/7**.

Service Germany

Our technology-oriented Competence Center in Lohr, Germany, is responsible for all your service-related queries for electric drive and controls.

Contact the **Service Hotline** and **Service Helpdesk** under:

Phone:	**+49 9352 40 5060**
Fax:	**+49 9352 18 4941**
E-mail:	service.svc@boschrexroth.de
Internet:	http://www.boschrexroth.com

Additional information on service, repair (e.g. delivery addresses) and training can be found on our internet sites.

Service worldwide

Outside Germany, please contact your local service office first. For hotline numbers, refer to the sales office addresses on the internet.

Preparing information

To be able to help you more quickly and efficiently, please have the following information ready:

- Detailed description of malfunction and circumstances
- Type plate specifications of the affected products, in particular type codes and serial numbers
- Your contact data (phone and fax number as well as your e-mail address)

Service and support

Notes

The Drive & Control Company

Rexroth
Bosch Group

Bosch Rexroth AG
Electric Drives and Controls
P.O. Box 13 57
97803 Lohr, Germany
Bgm.-Dr.-Nebel-Str. 2
97816 Lohr, Germany
Tel. +49 9352 18 0
Fax +49 9352 18 8400
www.boschrexroth.com/electrics

R911377106

DOK-CONTRL-S20*EIP*BK*-AP02-EN-P