

IC-KP2-2HB40-2V1D

**IDENTControl
RFID Evaluation Unit
with Ethernet Interface**

Manual



IDENTControl

With regard to the supply of products, the current issue of the following document is applicable:
The General Terms of Delivery for Products and Services of the Electrical Industry, published
by the Central Association of the Electrical Industry (Zentralverband Elektrotechnik und Elek-
troindustrie (ZVEI) e.V.) in its most recent version as well as the supplementary clause:
"Expanded reservation of proprietorship"

Worldwide

Pepperl+Fuchs Group
Lilienthalstr. 200
68307 Mannheim
Germany
Phone: +49 621 776 - 0
E-mail: info@de.pepperl-fuchs.com

North American Headquarters

Pepperl+Fuchs Inc.
8093 Darrow Road
Twinsburg, Ohio 44087
+1 330 425-3555
sales@us.pepperl-fuchs.com
United States of America
E-mail: sales@us.pepperl-fuchs.com

Asia Headquarters

Pepperl+Fuchs Pte. Ltd.
P+F Building
18 Ayer Rajah Crescent
Singapore 139942
Phone: +65 6779-9091
E-mail: sales@sg.pepperl-fuchs.com
<https://www.pepperl-fuchs.com>

1	Introduction	5
1.1	Content of this Document	5
1.2	Target Group, Personnel	6
1.3	Intended Use	6
1.4	Symbols Used	6
1.5	Definitions	7
1.6	Abbreviations and Terms	8
2	Cybersecurity	9
3	Certificates and Approvals	12
3.1	General	12
3.2	CE Marking	12
3.3	UL Information	12
4	Product Description	13
4.1	General	13
4.2	Interfaces and Connections	13
4.3	Indicators and Operating Elements	14
5	Installation	18
5.1	Storage and Transportation	18
5.2	Installation	18
5.3	Connection	19
5.3.1	Voltage Supply	19
5.3.2	Network Connection	19
5.3.3	RFID Read/Write Devices and External Triggers	20
5.3.4	Grounding Connection	20
6	Commissioning and Operation	22
6.1	Web Interface	22
6.2	REST API	23
6.3	Process Interfaces	23
6.4	Easy Mode	23
6.4.1	Command Overview	24
6.4.2	Basic Structure of the Process Data	26
6.4.2.1	Output Process Data (PLC -> Device)	26
6.4.2.2	Input Process Data (Device -> PLC)	27
6.4.2.3	Examples	28
6.4.2.4	Flowcharts	37
6.4.2.5	Timing	41

6.5	Expert Mode	42
6.5.1	Basic Command Sequence	42
6.5.1.1	Telegram Structure	42
6.5.1.2	Handshake Procedure.....	44
6.5.2	Command Overview	46
6.5.3	Parameter Overview	49
6.5.4	Command Examples	52
6.5.4.1	Single Read Commands.....	52
6.5.4.2	Enhanced Read Commands	56
6.5.4.3	Single Write Commands	59
6.5.4.4	Enhanced Write Commands.....	63
6.5.4.5	Configuration Commands – Read Parameters	66
6.5.4.6	Configuration Commands – Write Parameters.....	67
6.5.4.7	System Commands	68
6.6	Error and Status Messages	69
6.7	PROFINET	70
7	Service and Maintenance.....	74
8	Troubleshooting	75
9	Decommissioning and Disposal	76
10	Appendix.....	77
10.1	ASCII Table.....	77

1 Introduction

1.1 Content of this Document

This document contains information that you need in order to use your product throughout the applicable stages of the product life cycle. These can include the following:

- Product identification
- Delivery, transport, and storage
- Mounting and installation
- Commissioning and operation
- Maintenance and repair
- Troubleshooting
- Dismounting
- Disposal



Note

For full information on the product, refer to the further documentation on the Internet at www.pepperl-fuchs.com.



Note

For specific device information such as the year of construction, scan the QR code on the device. As an alternative, enter the serial number in the serial number search at www.pepperl-fuchs.com/device-information.

The documentation consists of the following parts:

- Present document
- Datasheet

Additionally, the following parts may belong to the documentation, if applicable:

- EU-type examination certificate
- EU declaration of conformity
- Attestation of conformity
- Certificates
- Control drawings
- Instruction manual
- Functional safety manual
- Cybersecurity documentation
- Other documents

1.2 Target Group, Personnel

Responsibility for planning, assembly, commissioning, operation, maintenance, and dismantling lies with the plant operator.

Only appropriately trained and qualified personnel may carry out mounting, installation, commissioning, operation, maintenance, and dismantling of the product. The personnel must have read and understood the instruction manual and the further documentation.

Prior to using the product make yourself familiar with it. Read the document carefully.

1.3 Intended Use

Always operate the device as described in these instructions. Only in this way, the safe function of the device and the connected systems is guaranteed.

The protection of operating personnel and plant is only given if the device is used in accordance with its intended use.

1.4 Symbols Used

This document contains symbols used to identify warning messages and informative messages.

Warning messages

Warnings are provided whenever your actions may result in hazards. Always observe these warning messages to ensure your personal safety and to prevent property damage.

Depending on the level of risk, warnings are presented in descending order as follows:



Danger!

This symbol indicates an imminent danger.

Non-observance will result in personal injury or death.



Warning!

This symbol indicates a possible fault or danger.

Non-observance may cause personal injury or serious property damage.



Caution!

This symbol indicates a possible fault.

Non-observance could interrupt the device and any connected systems and plants, or result in their complete failure.

Informative messages



Note

This symbol brings important information to your attention.



Action

This symbol indicates a paragraph with instructions. You are prompted to perform an action or a sequence of actions.

1.5

Definitions

The data representation format varies in the following documentation. The following formats are used:

ASCII

Example: "A"; "B"; "1"; "2"

An ASCII character corresponds to one byte. $2^8 = 256$ different characters can be represented.

Binary

Example: 1001_{bin}

Binary numbers are indicated by the suffix $_{\text{bin}}$.

DECIMAL

Example: 1234

Decimal numbers are shown without any additional notation.

HEX

Example: 16#41; 16#42; 16#31; 16#32

The hexadecimal representation of a byte consists of two digits (e.g., 16#41). The left portion is the high-order nibble (16#4x). The right portion is the low-order nibble (16#x1). The value range extends from 16#00 to 16#FF.

1.6 Abbreviations and Terms

Abbreviations

AFI	A pplication F amily I dentifier
CRC	C yclic R edundancy C heck
EIRP	E quivalent I sotopically R adiated P ower
EPC	E lectronic P roduct C ode
ERP	E ffective R adiated P ower
FCC	F ederal C ommunications C ommission
FHSS	F requency H opping S pread S pectrum
IC	I ndustry C anada
ISO	I nternational S tandardisation O rganisation
MDID	M ask D esigner I dentifier
PC	P rotocol C ontrol
RFID	R adio F requency I dentification
PLC	P rogrammable L ogic C ontroller
Tag	Read/write tag; tag
TID	T ag I dentification
TMN	T ag M odel N umber
UHF	U ltra- H igh F requency
UII	U nique I tem I dentifier

Terms

Tag	Mobile data carrier containing user data and a unique number
Fixed code	Unique and unalterable number of a tag
IPC	Pepperl+Fuchs-specific designation of an LF tag
IQC	Pepperl+Fuchs-specific designation of an HF tag
IUC	Pepperl+Fuchs-specific designation of an UHF tag
ISO/IEC 15693	Standard for data transmission in an HF RFID system
ISO/IEC 18000-63	Standard for data transmission in an UHF RFID system
PLC	Programmable logic controller; device for controlling a machine or plant

2 Cybersecurity

General

- Regularly check the manufacturer portal for cybersecurity alerts at <https://www.pepperl-fuchs.com/cybersecurity>.
- Report all cybersecurity incidents to the manufacturer.
- To protect devices, machines, systems, technical infrastructure and networks against cyber threats, users must have a comprehensive cybersecurity concept in place, which must be continuously reviewed and maintained. General recommendations can be found at <https://www.bsi.bund.de/dok/6603576>.

Security Context

The device is designed for operation in an automation network. This is a secure network with known and trusted participants, which is (physically or logically) separated from the corporate network.

Do not connect the device directly to the internet. Operate the device in a protected network area. Use a firewall to connect the internal protected network to external networks. Configure this with restrictive rules.

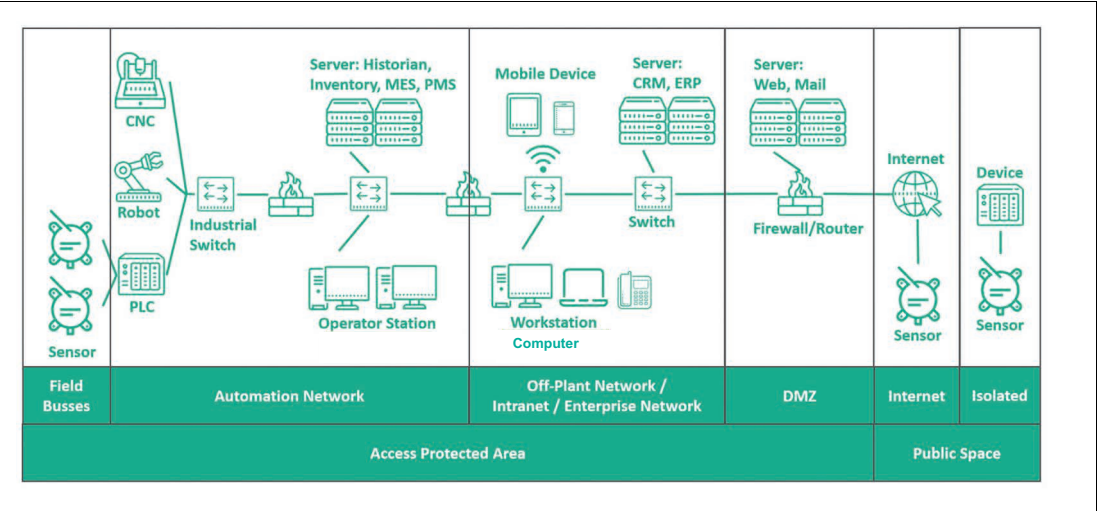


Figure 2.1 Example of the integration of devices and software into a system

Available Protocols and Services

Protocol/ Service	Protocol/ Port Number	Factory settings	Service status configu- rable	Notes
DHCP	UDP/68	Enabled	Yes	The service status depends on the setting of the rotary switch.
PROFINET	- UDP/34964 - UDP/49152	Enabled	Yes	The service status depends on the setting of the rotary switch.
Profinet DCP		Enabled	No	The service status depends on the setting of the rotary switch.
EtherNet/IP	- TCP/44818 - UDP/44818 - UDP/2222	Enabled	Yes	The service status depends on the setting of the rotary switch.
HTTP (REST API)	TCP/80	Disabled	Yes	
HTTPS (REST API)	TCP/443	Enabled	Yes	Self-generated certificates The device can generate self-signed certificates that are valid for the device's IP address. If the IP address changes, a new certificate is generated. Encryptions TLSv1.2 is supported. The supported encryption suites for key pairs with ECC are: - TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 The following elliptic curves are supported: - secp256r1 - bp256r1 - Curve25519 For RSA key pairs: - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (rsa 2048) - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (rsa 2048) Recommendation: If possible, use ECC key pairs with CHACHA20_POLY1305-bulk encryption to achieve the highest transfer rate. All modern clients support this. Browser: Google Chrome, Microsoft Edge, Apple Safari
SNMP	UDP/161	Enabled	Yes	
NTP Client	UDP/123	Disabled	Yes	

2026-04

Protocol/ Service	Protocol/ Port Number	Factory settings	Service status configu- rable	Notes
LLDP		Enabled	No	
ICMP		Enabled	No	

Hardening of the Device

- Physically protect the device and the application against unauthorized access.
- Disable any unused services.
- Ensure that all passwords and user-generated private keys are protected from unauthorised access.
- Avoid reusing passwords for different users and systems.
- Replace passwords, certificates and keys immediately as soon as you identify potential security vulnerabilities.
- Renew certificates regularly.
- Change passwords regularly.
- Use a suitable password manager to create and store passwords.

Security Functions

- The device has protection against brute force attacks to prevent unauthorised access through password guessing attempts.
- The device implements a session timeout to automatically disconnect connections after a specified period of inactivity.
- The device is assigned a unique default password, which is printed on a sticker on the device.
- The device uses signed firmware to ensure that only software from trusted sources is installed on the device, thereby preventing tampering or unauthorised modifications to the firmware.

User Profiles and Rights

The device's web interface has the following user roles and rights.

User profile	Username	Rights	Factory settings	Remark
Administrator	admin	No restrictions	Enabled	Standard user
Beobachter	observer	Read only	Disabled	Configurable by administrator

- No further user profiles can be defined or users created.
- The rights and names of existing users cannot be changed.

Firmware

Only use firmware from trusted sources. Keep the software up to date. Always use the latest firmware/software version for the device. Check regularly at <https://www.pepperl-fuchs.com/de-de>, whether security updates are available for the products and install them. After the release of a new version, older versions will no longer be supported or maintained.

Decommissioning

Reset the device to factory settings.

3 Certificates and Approvals

3.1 General

This manual applies to several types and variants of the device. The applicable approvals can be found in the relevant data sheet for your device. Further approvals are available on request.

3.2 CE Marking

This product was developed and manufactured in line with the applicable European standards and directives.



Note

A Declaration of Conformity can be requested from the manufacturer or downloaded from www.pepperl-fuchs.com.

3.3 UL Information

This device may be operated at altitudes of up to 5000 m.

The ambient temperature range for operation is -25 °C to +70 °C. The pollution degree is 2.

The maximum relative humidity is 80 % at temperatures up to 31 °C, decreasing linearly to 50 % relative humidity at 40 °C.

The nominal supply voltage is 24 V_{DC}; the voltage range is 18 V to 32 V_{DC}. The supply must be an LEC (Limited Energy Circuit), LPS (Limited Power Source), or Class 2. Overvoltage Category II applies.

Degree of protection (IP67) is not included in the UL approval. The degree of protection is tested by Pepperl+Fuchs SE.

External circuits intended to be connected to this unit must be galvanically isolated from the mains supply or hazardous live voltages by reinforced or double insulation and must meet the limits of Clauses 6.3 and 9.4 of UL 61010-1.



Caution!

Cable specifications

The maximum permissible temperature of the connection cable must be at least +90 °C.

The minimum cross-section of the connection cable must be 22 AWG or 0.34 mm².

Connection cables with cULus / cURus approval (PVVA/7, PVVA2/8, CYJV/7, CYJV2/8) must be used.

The rated current of the connection cable must be at least 1 A (per pin/wire).

4 Product Description

4.1 General

The IDENTControl is an RFID evaluation unit that forms part of an RFID identification system from Pepperl+Fuchs. The system consists of the evaluation unit itself as well as the associated RFID read/write devices, which are available in different designs and operating frequencies.

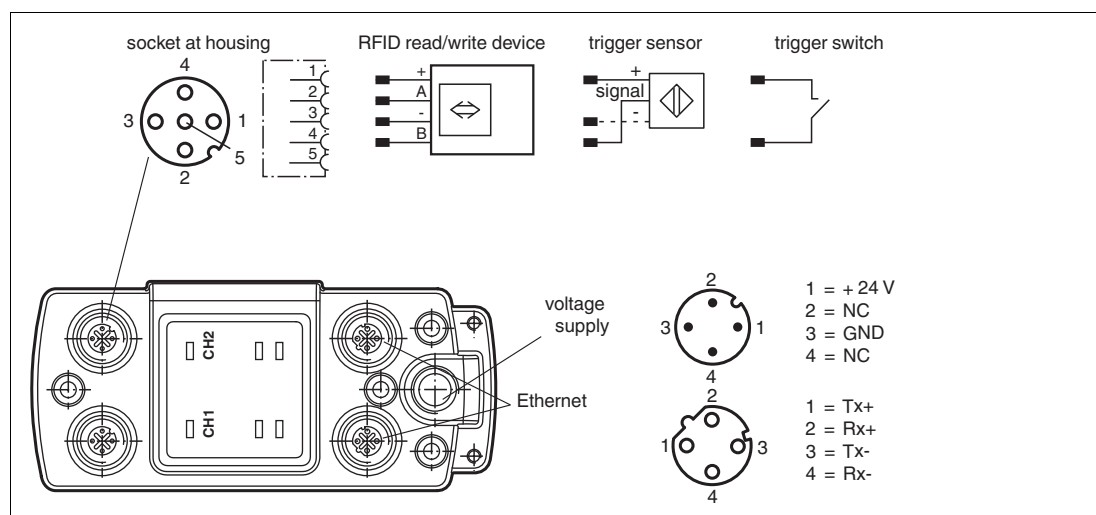
The IDENTControl provides standardized connection, configuration, and data processing and serves as the interface to higher-level systems. For this purpose, the IDENTControl features a multiprotocol Ethernet interface that enables connection to different network protocols such as programmable logic controllers (OT systems) or other IT applications.

4.2 Interfaces and Connections

The IDENTControl has the following connections:

- 2 × M12 female connectors (A-coded) for connecting RFID read/write devices or for use as inputs (e.g., for connecting external trigger sensors) or outputs (e.g., for connecting external signaling devices).
- 2 × M12 female connectors (D-coded) for Ethernet, configured as a switch.
- 1 × M12 male connector (A-coded) for connecting the power supply.

Electrical Connection



4.3 Indicators and Operating Elements

LEDs

The device has 8 LEDs on the front panel that indicate the device status.

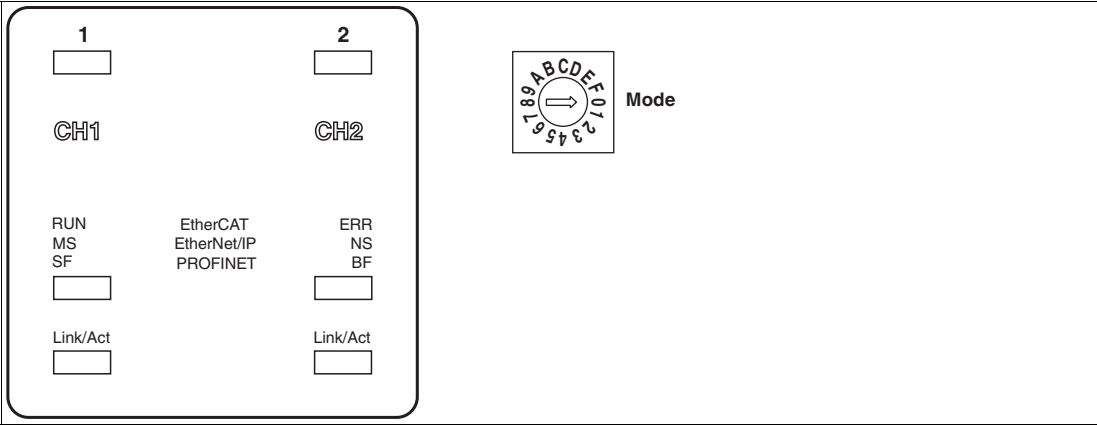


Figure 4.1

LED	Function	Meaning
1 2	Status indication of the RFID read/write devices	- Off: No read/write operation active. - Green: Read/write operation in progress. - Yellow: Read/write operation completed successfully.
CH1 CH2	Status of the RFID read/write device connection	- Off: No RFID read/write device connected or connection interrupted. - Green: Connected RFID read/write device ready for operation. - Red: Configuration error of the connected RFID read/write device (incorrect tag type configured).
Link/Act	Connection/network activity for the respective Ethernet port	- Off: No network connection or communication - Green: Network connection established - Yellow: Communication active
EtherCAT		
RUN	Not yet implemented	
ERR	Not yet implemented	
EtherNet/IP		

LED	Function	Meaning
MS	Module Status	- Green: IDENTControl operating normally. - Flashing green (1 Hz): The IDENTControl has not yet been configured. - Flashing green/red/green rapidly: IDENTControl performs a self-test after power-on. During the self-test, the following sequence is displayed: - NS LED off - MS LED lights green for approx. 250 ms, then red for approx. 250 ms, then green again (remains in this state until the self-test is complete). - NS LED lights green for approx. 250 ms, then red for approx. 250 ms, then turns off (remains in this state until the self-test is complete). - Flashing sequence red/green/off: The flashing sequence visually identifies the IDENTControl. The scanner can start the flashing sequence via Identity Object 1 of the IDENTControl. The MS LED and NS LED perform the flashing sequence simultaneously. - Flashing red (1 Hz): IDENTControl has detected a serious, recoverable fault. (e.g., incorrect or inconsistent configuration). - Red: IDENTControl has detected a serious, non-recoverable fault. - Off: The IDENTControl is switched off.

LED	Function	Meaning
NS	Network Status	- Green: An IP address is configured, at least one CIP connection is established, and no Exclusive Owner connection has timed out. - Flashing green (1 Hz): An IP address is configured, but no CIP connections are established, and no Exclusive Owner connection has timed out. - Flashing sequence red/green/off: The flashing sequence visually identifies the IDENTControl. The scanner can start the flashing sequence via Identity Object 1 of the IDENTControl. The MS LED and NS LED perform the flashing sequence simultaneously. - Flashing green/red/off rapidly: IDENTControl performs a self-test after power-on. See the description of the MS LED in self-test status. - Flashing red (1 Hz): An IP address is configured, and an Exclusive Owner connection for which this IDENTControl is the target has timed out. The NS LED turns solid green again only after all timed-out Exclusive Owner connections are restored. - Red: IDENTControl has detected that its IP address is already in use. - Off: The IDENTControl has no IP address (or is switched off).
PROFINET		
SF	System Failure	- Off: No error. - Flashing red (1 Hz, 3 s): DCP signal service is being initiated via the bus. - Red: Generic, extended, or channel diagnostics present; system fault.
BF	Bus Failure	- Off: No error. - Flashing red (2 Hz): No data exchange. - Red: No configuration, slow physical connection, or no physical connection.

Table 4.1

Operating Elements

A rotary switch for setting the communication protocol is located on the back of the device.

Switch position	Meaning
0	EtherNet/IP Network settings from non-volatile memory After resetting to factory settings: DHCP All network settings can be changed using the web interface or other software. All the applied network settings are stored in non-volatile memory.
1	EtherNet/IP BOOTP With this rotary switch setting, the device always boots in BOOTP mode. (Network settings in non-volatile memory are ignored and overwritten.) All network settings can be changed using the web interface or other software. All the applied network settings are stored in non-volatile memory.
2	EtherNet/IP DHCP With this rotary switch setting, the device always boots in DHCP mode. (Network settings in non-volatile memory are ignored and overwritten.) All network settings can be changed using the web interface or other software. All the applied network settings are stored in non-volatile memory.
3	EtherNet/IP Static IP: 169.254.10.12, subnet mask: 255.255.0.0 The device always boots with IP: 169.254.10.12. (Network settings in non-volatile memory are ignored and overwritten.)
4	PROFINET DCP
5	Not used
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	Reset to factory settings To reset to factory settings, disconnect the supply voltage, select the switch position, and reconnect the supply voltage. The ERR/NS/BF LED lights up red. When it starts to flash, the process is complete. Disconnect the supply voltage and select the desired communication protocol using the rotary switch.

Table 4.2

5 Installation

5.1 Storage and Transportation

Keep the original packaging. Always store and transport the device in the original packaging. Store the device in a clean and dry environment. The permitted ambient conditions must be considered, see datasheet.

5.2 Installation



Note

The installation recommendations provided in this document assume the most favorable ambient conditions. Pepperl+Fuchs does not guarantee fault-free operation in environments outside the system.

Mounting the IDENTControl RFID evaluation unit

IC-KP2-2HB40-2V1D

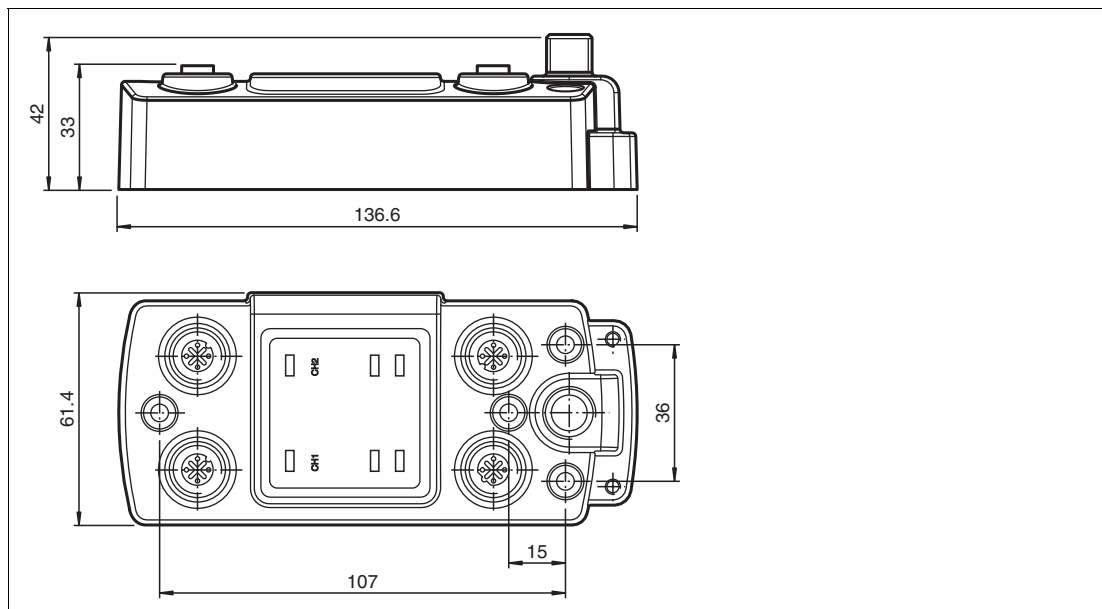


Figure 5.1

The IDENTControl has four 5 mm holes for mounting the device. M4 screws with a length >20 mm are recommended for this purpose. Alternatively, DIN rail mounting is possible using mounting accessories that are available separately (not included in the scope of delivery).

Before connection and final installation, set the desired communication protocol (see chapter 4.3).



Note

The rotary switch for this is located on the underside of the IDENTControl. After the IDENTControl has been installed, this rotary switch is no longer freely accessible.

5.3

Connection

**Warning!**

Incorrect electrical connection

Damage to the device or plant caused by incorrect electrical connection.

Check all connections in the plant before commissioning the device.

**Note**

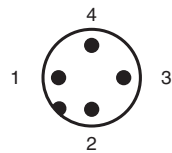
Connect the device in the following order:

1. Grounding connection
2. Network connections
3. RFID read/write devices
4. Supply voltage

5.3.1

Voltage Supply

Connect the voltage supply via an M12 female connector (A-coded). A connector with the following pin assignment is located on the housing:

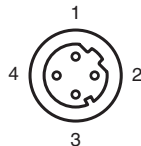


- | | |
|----------|--------|
| 1 | + 24 V |
| 2 | NC |
| 3 | GND |
| 4 | NC |

5.3.2

Network Connection

Connect the IDENTControl to the network using an M12 connector (D-coded). Two ports are available for this purpose, both of which are located in the same network (Ethernet switch).



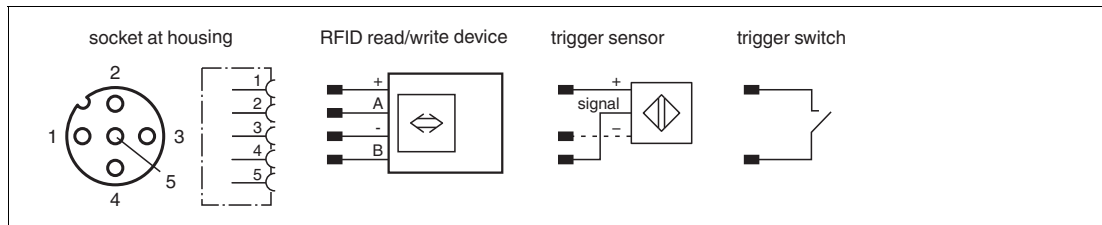
- | | |
|----------|-----|
| 1 | Tx+ |
| 2 | Rx+ |
| 3 | Tx- |
| 4 | Rx- |

5.3.3 RFID Read/Write Devices and External Triggers

Suitable RFID read/write devices can be connected to the IDENTControl via an M12 connector (A-coded).

Alternatively, the connections can also be used to connect a trigger sensor or switch. The trigger can be assigned to an RFID read/write device. The trigger sensor must be sourcing (PNP).

The connections can also be used as outputs, for example to connect external signaling devices. The output can be switched depending on a successful read/write event.



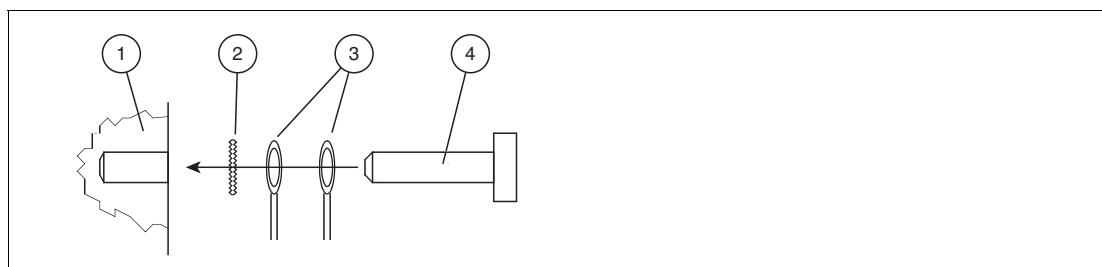
5.3.4 Grounding Connection

Connect the IDENTControl to ground using a screw at one of the two grounding connections on the housing.

Note

To ensure reliable grounding, the toothed lock washer must be installed between the crimp connectors and the housing.

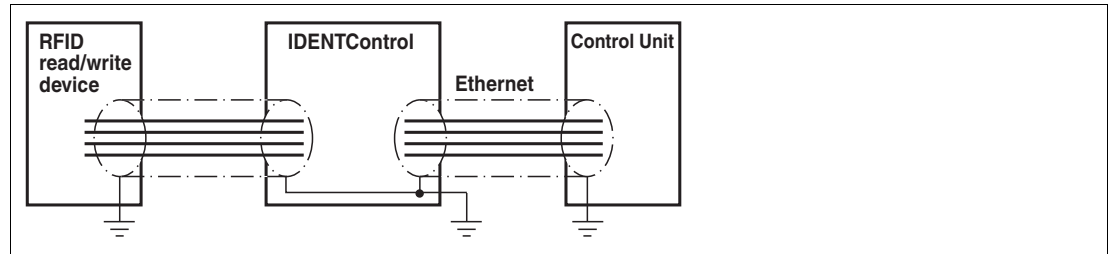
Use a protective earth conductor with a conductor cross section of at least 4 mm².



- 1 Housing
- 2 Serrated lock washer
- 3 Crimp connector
- 4 Lock screw

EMC Concept

The IDENTControl's excellent emissions and immunity performance is based on its continuous shielding concept. This concept uses the principle of the Faraday cage. Interference is intercepted by the shield and safely dissipated via the protective earth connections.



The cable shielding is used to discharge electromagnetic interference. When shielding a cable, you must connect both sides of the shield to ground with low resistance and low inductance.



Note

If cables with double shields are used, e.g. wire mesh and metalized foil, the both shields must be connected together, with low resistance, at the ends when making up the cable.

Power supply cables are the source of much interference, e.g. from the supply lines of 3-phase electric motors. For this reason, the parallel laying of power supply cables with data and signal cables should be avoided, particularly in the same cable duct.

The continuous shielding concept is complemented by the die-cast metal housing of the IDENTControl and the metal housing of the RFID read/write devices.

To ensure that the shielding is not interrupted by the metal housing, the shields must be connected to ground with low impedance and low inductance. All electronics and cabling are thus located within a Faraday cage.



Note

Not all RFID read/write devices that can be connected to the IDENTControl have a grounding connection. In such cases, grounding of the RFID read/write device can be omitted.



Warning!

To prevent device malfunctions caused by electromagnetic interference, use cables with continuous shielding.

6 Commissioning and Operation

6.1 Web Interface

The device can be conveniently configured and operated via an integrated web interface. The selected communication protocol is irrelevant in this context.

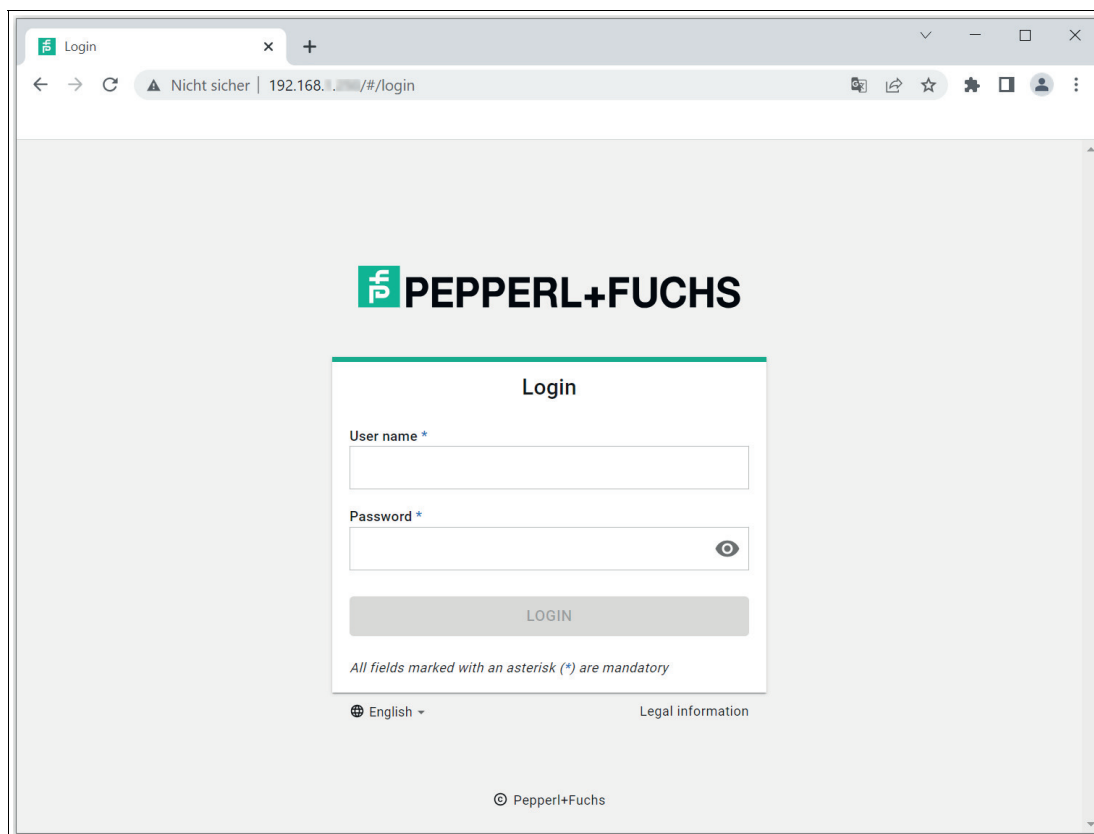


Figure 6.1



Logging in via the web interface

1. Make sure that your computer is on the same network as the device.
2. Enter the current IP address in the address bar of your browser (see "Operating Elements" on page 17).



Note

For security reasons, each device is delivered with a unique password, which can be found in the accompanying documentation or on the label attached to the device.

- Username: admin
- Password: see label on the device ("Factory Password")

3. Enter the login credentials.

The additional interface pages are continuously expanded and can be operated intuitively using the help texts.

6.2 REST API

The device has a REST API (Representational State Transfer) for programming. An interface description is available on Swagger Hub at <https://pepperl-fuchs.com/openapi>.

6.3 Process Interfaces

General

This device supports various process interfaces that can be selected using the rotary switch on the rear of the device (see "Operating Elements" on page 17).

The device can also be operated in two operating modes, which are available for all process interfaces:

- **Easy Mode**
Easy Mode enables simplified commissioning with limited functionality. This is the preferred operating mode for standard applications.
- **Expert Mode**
In Expert Mode, the full command set is available. To use the full range of functions, a function block is required for integration into the PLC.

6.4 Easy Mode

The device uses the Easy Mode operating mode for data transmission to a higher-level system. When this mode is used, the device can be commissioned on a controller without an additional function block. This reduces the effort required for commissioning the device.

When Easy Mode is used, a distinction is made between parameter data and process data. The parameter data is configured in the controller via the device description file and is transferred when the device starts. The file contains separate data for each channel for configuring the IDENTControl and the connected RFID read/write devices. The process data is transmitted cyclically. The process data is divided into input data and output data. The length of the process data can be configured for each channel via the device description file.

Length of process data	PROFINET	EtherNet/IP
32 bytes	32 bytes in/out Easy Mode	32 bytes in/out Easy Mode
64 bytes	64 bytes in/out Easy Mode	64 bytes in/out Easy Mode
128 bytes	128 bytes in/out Easy Mode	128 bytes in/out Easy Mode
256 bytes	256 bytes in/out Easy Mode	256 bytes in/out Easy Mode

Table 6.1 Overview of process data sizes

6.4.1 Command Overview

Easy Mode supports the following read and write commands via the process image:

Read Commands

Command	Description	System
Autostart	Automatic execution of a read command according to the parameter settings in the device description file. A read command can be started automatically without being triggered via the process image. Autostart is disabled in the factory setting.	IPH*, IQH*, IUH*
Memory area	Memory area (UII/EPC/UID) + user data Reads the user data according to the following settings: "Number of bytes" and "Start address". With IQH3 and IUH* systems, the UID or UII/EPC is prepended when the long-form data format is used. This part is omitted when the short-form data format is used.	IPH*, IQH*, IUH*
	UII/EPC Reads the UII/EPC including Protocol Control (PC).	IUH*
	(UII/EPC) + TID Reads the TID. When the long-form data format is used, the UII/EPC is prepended. This part is omitted when the short-form data format is used.	IUH*
	UID Reads the UID.	IQH*
	Fixed code Reads the fixed code.	IPH*
Number of bytes	The number of bytes to be read from the user data.	IPH*, IQH*, IUH*
Start address	The start address for accessing the user data.	IPH*, IQH*, IUH*

Table 6.2

Write Commands

Command	Description	System
Memory area	User data Writes the user data according to the following settings: "Number of bytes" and "Start address".	IPH*, IQH*, IUH*
	UII/EPC (incl. PC) Writes the UII/EPC including Protocol Control (PC). The correct PC content must be calculated by the user.	IUH*
	EPC (excl. PC) Writes the UII/EPC including Protocol Control (PC). The PC content is calculated automatically.	IUH*
Number of bytes	The number of bytes to be written to the user data or UII/EPC.	IPH*, IQH*, IUH*
Start address	The start address for accessing the user data.	IPH*, IQH*, IUH*

Table 6.3

IPH* parameters

Parameter	Description
Tag type (CT)	Selects the tag type.

Table 6.4

IQH1* parameters

Parameter	Description
Tag type (CT)	Selects the tag type.

Table 6.5

IQH3* parameters

Parameter	Description
Transmitter radiated power (PT)	Selects the transmitter radiated power level.
Abort criterion for the search algorithm (NT)	Abort criterion for the read/write command.
Number of attempts (TA)	Number of attempts to access a tag.
Expected number of tags (QW)	Sets the time slots for accessing tags.
Tag loss smoothing (E5)	Sets the number of unsuccessful attempts to access a tag before the tag is reported as having left the detection zone.
Input representation	Chooses between short-form and long-form data format.
Tag type (CT)	Selects the tag type.
Overtemperature handling (TO)	Configures behavior in the event of overtemperature.

Table 6.6

IUH-F19* parameters

Parameter	Description
Transmitter radiated power (PT)	Selects the transmitter radiated power level.
Abort criterion for the search algorithm (NT)	Abort criterion for the read/write command.
Number of attempts (TA)	Number of attempts to access a tag.
Expected number of tags (QW)	Sets the time slots for accessing tags.
Tag loss smoothing (E5)	Sets the number of unsuccessful access attempts before the tag is reported as having left the sensing range.
Antenna polarization (AP)	Selects the antenna polarization.
Input representation	Chooses between short-form and long-form data format.
Number of channels (NC)	Configures the transmit channels used (depending on the country variant used).
Transmit channels (CD)	Configures the transmit channels used (depending on the country variant used).

Table 6.7

For a detailed description of the parameters, refer to the manuals for the relevant RFID read/write devices.

6.4.2 Basic Structure of the Process Data

The device process data is exchanged cyclically between a higher-level system (e.g., a PLC) and the device. A distinction is made between the output process data and the input process data. The process data for the outputs is transmitted from the PLC towards the device. The process data for the inputs is transmitted from the device towards the PLC.

The length of the process data fields depends on the communication module selected in the PLC.

6.4.2.1 Output Process Data (PLC -> Device)

Byte	Content							
0	0	0	0	0	0	0	Write	Read
1	16#00							
2	16#00							
3	16#00							
4	User data for write job or 16#00 for read job							
...	User data for write job or 16#00 for read job							
255	User data for write job or 16#00 for read job							

Table 6.8

Byte 0

This byte contains the control bits for starting a read job or write job. If the autostart function is enabled, the control bits have no effect.

Read: As soon as this bit is set (TRUE), a read job is started according to the configuration defined by the parameters in the device description file. The read job is executed continuously. To cancel the read job, reset the bit (FALSE).

Write: As soon as this bit is set (TRUE), a write job is started according to the configuration defined by the parameters in the device description file. The write job is executed continuously. To cancel the write job, reset the bit (FALSE). Note that both bits must not be set at the same time. The remaining bits have no function.

Byte 1/2/3

These bytes are not used when Easy Mode is used. Set the value 16#00.

Byte 4 ... 255

When executing a read job, these bytes have no function and are set to the value 16#00. When executing a write job, this area contains the user data to be written to the tag.

6.4.2.2 Input Process Data (Device -> PLC)

Byte	Content								
0	0	0	0	0	Tag present bit	Error	Job active	Write successful	Read successful
1	Length information								
2	RSSI value/16#00								
3	Transmitter radiated power in dBm (IUH* only)/16#00								
4	Data for read/write job / error code								
5	Data for read/write job / error information								
...	Data for read/write job / error information								
255	Data for read/write job / error information								

Table 6.9

- Byte 0** This byte contains the control bits for signaling the execution status of the read or write job.
- Read successful:** This bit signals that data was read from a tag successfully. If a tag enters the detection zone and the data was read successfully, this bit is set. While the tag remains within the detection zone, the bit remains set. As soon as the tag leaves this zone, the bit is reset. If multiple tags are present simultaneously within the sensing range, a positive edge indicates the transmission of another tag.
- Write successful:** This bit signals that data was written to a tag successfully. If a tag enters the detection zone and the data was written successfully, this bit is set. While the tag remains within the detection zone, the bit remains set. As soon as the tag leaves this zone, the bit is reset. If multiple tags are present simultaneously within the sensing range, a positive edge indicates that another tag was written to successfully.
- Job active:** This bit is set while the read or write job is being executed. As soon as the job is complete, the bit is reset.
- Error:** If an error occurs during execution of a read or write job, or if a parameter setting is incorrect, this is signaled by this bit. At the same time, the process data contains additional error information in the form of an error code and an error description. For a detailed description of the errors, see chapter 6.6.
- Tag present bit:** This bit is set as long as at least one tag is present within the sensing range during an active read or write job.
- Byte 1** This byte contains the number of bytes transferred (starting from byte 4). If a tag enters the detection zone and the data was read successfully (Read successful = TRUE), this byte indicates the length of the read data (starting from byte 4). If an error occurs during execution of a job (Error = TRUE), this byte contains a length specification for the error information.
- Byte 2** This byte contains the RSSI value.
- Byte 3** This byte contains the configured transmitter radiated power in dBm with which the tag was read or written.
- Byte 4** Data for read/write job / error code.
- Byte 5** Data for read/write job / error information.
- ...** Data for read/write job / error information.
- Byte 255** Data for read/write job / error information.

6.4.2.3

Examples

Output Process Data (PLC -> Device)

Example: Write to user memory

Name	Address	Displ...	Monitor value	Modify value
StartRead	%A0.0	Bool	☐ FALSE	
StartWrite	%A0.1	Bool	☒ TRUE	TRUE
ControlByte_OutData	%AB0	Bin	2#0000_0010	
Unused_1	%AB1	Hex	16#00	
Unused_2	%AB2	Hex	16#00	
Unused_3	%AB3	Hex	16#00	
WriteData_Byte_1	%AB4	Hex	16#01	16#01
WriteData_Byte_2	%AB5	Hex	16#02	16#02
WriteData_Byte_3	%AB6	Hex	16#03	16#03
WriteData_Byte_4	%AB7	Hex	16#04	16#04
WriteData_Byte_5	%AB8	Hex	16#05	16#05
WriteData_Byte_6	%AB9	Hex	16#06	16#06
WriteData_Byte_7	%AB10	Hex	16#07	16#07
WriteData_Byte_8	%AB11	Hex	16#08	16#08
WriteData_Byte_9	%AB12	Hex	16#00	

Output data: Start write job

Byte 0: Control byte; 16#02 (000_0010bin)

Byte 4 ... 11: Write data; the "Start Write" bit is set to 1, which starts the write job. At the same time, the data to be written must be transferred to the process data.

Name	Address	Displa...	Monitor value	Modify value
ReadValid	%E0.0	Bool	☐ FALSE	
WriteValid	%E0.1	Bool	☐ FALSE	
TaskActive	%E0.2	Bool	☒ TRUE	
Error	%E0.3	Bool	☐ FALSE	
TagPresent	%E0.4	Bool	☐ FALSE	
ControlByte_InData	%EB0	Bin	2#0000_0100	
Length	%EB1	DEC	0	
RSSI	%EB2	DEC	0	
PowerTransmit_dBm	%EB3	DEC	0	
ReadData_Byte_1	%EB4	Hex	16#00	
ReadData_Byte_2	%EB5	Hex	16#00	
ReadData_Byte_3	%EB6	Hex	16#00	
ReadData_Byte_4	%EB7	Hex	16#00	
ReadData_Byte_5	%EB8	Hex	16#00	
ReadData_Byte_6	%EB9	Hex	16#00	
ReadData_Byte_7	%EB10	Hex	16#00	
ReadData_Byte_8	%EB11	Hex	16#00	
ReadData_Byte_9	%EB12	Hex	16#00	

Input data: No tag

Byte 0: Control byte; 16#04 (0000_0100bin); the "Job active" bit is set to 1, thereby signaling that job execution has been activated. The "Write successful" bit is set to 0. There is no tag in the detection zone.

Name	Address	Displa...	Monitor value	Modify value
"ReadValid"	%E0.0	Bool	☐ FALSE	
"WriteValid"	%E0.1	Bool	☒ TRUE	
"TaskActive"	%E0.2	Bool	☒ TRUE	
"Error"	%E0.3	Bool	☐ FALSE	
"TagPresent"	%E0.4	Bool	☒ TRUE	
"ControlByte_InData"	%EB0	Bin	2#0001_0110	
"Length"	%EB1	DEC	8	
"RSSI"	%EB2	DEC	29	
"PowerTransmit_dBm"	%EB3	DEC	0	
"ReadData_Byte_1"	%EB4	Hex	16#E0	
"ReadData_Byte_2"	%EB5	Hex	16#04	
"ReadData_Byte_3"	%EB6	Hex	16#01	
"ReadData_Byte_4"	%EB7	Hex	16#50	
"ReadData_Byte_5"	%EB8	Hex	16#E9	
"ReadData_Byte_6"	%EB9	Hex	16#DE	
"ReadData_Byte_7"	%EB10	Hex	16#0E	
"ReadData_Byte_8"	%EB11	Hex	16#7E	
"ReadData_Byte_9"	%EB12	Hex	16#00	

Input data: Data written; short-form data format

Byte 0: Control byte; 16#16 (0001_0101bin); the "Write successful" bit is set to 1, thereby signaling successful write access to a tag. The "Job active" bit is set to 1 because job execution has been activated. The "Tag-Present" bit is set to 1 because at least one tag is located in the detection zone.

Byte 1: Length information; 16#08 (8 dec); the length of the transmitted information is 8 bytes.

Byte 2: RSSI value; 16#1D (29dec); the RSSI value for tag access is 29.

Byte 3: Transmitter radiated power in dBm; this information is not supported by all systems. When the IQH3 RFID read/write device is used, this byte has the value 16#00 (0 dec).

Byte 4 ... 11: UID; UID of the tag that was successfully written.

Name	Address	Displa...	Monitor value	Modify value
ReadValid	%E0.0	Bool	☐ FALSE	
WriteValid	%E0.1	Bool	☒ TRUE	
TaskActive	%E0.2	Bool	☒ TRUE	
Error	%E0.3	Bool	☐ FALSE	
TagPresent	%E0.4	Bool	☒ TRUE	
ControlByte_InData	%EB0	Bin	2#0001_0110	
Length	%EB1	DEC	10	
RSSI	%EB2	DEC	29	
PowerTransmit_dBm	%EB3	DEC	0	
ReadData_Byte_1	%EB4	Hex	16#00	
ReadData_Byte_2	%EB5	Hex	16#08	
ReadData_Byte_3	%EB6	Hex	16#E0	
ReadData_Byte_4	%EB7	Hex	16#04	
ReadData_Byte_5	%EB8	Hex	16#01	
ReadData_Byte_6	%EB9	Hex	16#50	
ReadData_Byte_7	%EB10	Hex	16#E9	
ReadData_Byte_8	%EB11	Hex	16#DE	
ReadData_Byte_9	%EB12	Hex	16#0E	
ReadData_Byte_10	%EB13	Hex	16#7E	
ReadData_Byte_11	%EB14	Hex	16#00	

Input data: Data written;
long-form data format

Byte 0: Control byte; 16#16 (0001_0110bin); the "Write successful" bit is set to 1, thereby signaling successful write access to a tag. The "Job active" bit is set to 1 because job execution has been activated. The "Tag-Present" bit is set to 1 because at least one tag is located in the detection zone.

Byte 1: Length information; 16#0A (10 dec); the length of the transmitted information is 10 bytes.

Byte 2: RSSI value; 16#1D (29dec); the RSSI value for tag access is 29.

Byte 3: Transmitter radiated power in dBm; 16#00 (0dec); this information is not supported by all systems. When the IQH3 RFID read/write device is used, this byte has the value 16#00 (0 dec).

Byte 4 ... 5: UID length; 16#0008 (8dec); length of the UID of the programmed tag.

Byte 6 ... 13: UID; 16#E0...; UID (fixed code) of the programmed tag.

Output data: End write job

Byte 0: Control byte; 16#00 (0000_0000bin); the "Start Write" bit is reset to 0, thereby ending the write job.

Name	Address	Displa...	Monitor value	Modify value
StartRead	%A0.0	Bool	☐ FALSE	
StartWrite	%A0.1	Bool	☐ FALSE	FALSE
ControlByte_OutData	%AB0	Bin	2#0000_0000	
Unused_1	%AB1	Hex	16#00	
Unused_2	%AB2	Hex	16#00	
Unused_3	%AB3	Hex	16#00	
WriteData_Byte_1	%AB4	Hex	16#01	16#01
WriteData_Byte_2	%AB5	Hex	16#02	16#02
WriteData_Byte_3	%AB6	Hex	16#03	16#03
WriteData_Byte_4	%AB7	Hex	16#04	16#04
WriteData_Byte_5	%AB8	Hex	16#05	16#05
WriteData_Byte_6	%AB9	Hex	16#06	16#06
WriteData_Byte_7	%AB10	Hex	16#07	16#07
WriteData_Byte_8	%AB11	Hex	16#08	16#08
WriteData_Byte_9	%AB12	Hex	16#00	

Name	Address	Displa...	Monitor value	Modify value
"ReadValid"	%E0.0	Bool	☐ FALSE	
"WriteValid"	%E0.1	Bool	☐ FALSE	
"TaskActive"	%E0.2	Bool	☐ FALSE	
"Error"	%E0.3	Bool	☐ FALSE	
"TagPresent"	%E0.4	Bool	☐ FALSE	
"ControlByte_InData"	%EB0	Bin	2#0000_0000	
"Length"	%EB1	DEC	0	
"RSSI"	%EB2	DEC	0	
"PowerTransmit_dBm"	%EB3	DEC	0	
"ReadData_Byte_1"	%EB4	Hex	16#00	
"ReadData_Byte_2"	%EB5	Hex	16#00	
"ReadData_Byte_3"	%EB6	Hex	16#00	
"ReadData_Byte_4"	%EB7	Hex	16#00	
"ReadData_Byte_5"	%EB8	Hex	16#00	
"ReadData_Byte_6"	%EB9	Hex	16#00	
"ReadData_Byte_7"	%EB10	Hex	16#00	
"ReadData_Byte_8"	%EB11	Hex	16#00	
"ReadData_Byte_9"	%EB12	Hex	16#00	

Input data: Write job ended
Byte 0: Control byte; 16#00 (0000_0000bin); the "Job active" bit is reset to 0, thereby signaling the end of job execution.

Input Process Data (Device -> PLC)
Example: Read user memory without autostart

Name	Address	Displ...	Monitor value	Modify value
"StartRead"	%A0.0	Bool	☒ TRUE	TRUE
"StartWrite"	%A0.1	Bool	☐ FALSE	
"ControlByte_OutData"	%AB0	Bin	2#0000_0001	
"Unused_1"	%AB1	Hex	16#00	
"Unused_2"	%AB2	Hex	16#00	
"Unused_3"	%AB3	Hex	16#00	
"WriteData_Byte_1"	%AB4	Hex	16#00	
"WriteData_Byte_2"	%AB5	Hex	16#00	
"WriteData_Byte_3"	%AB6	Hex	16#00	
"WriteData_Byte_4"	%AB7	Hex	16#00	
"WriteData_Byte_5"	%AB8	Hex	16#00	
"WriteData_Byte_6"	%AB9	Hex	16#00	
"WriteData_Byte_7"	%AB10	Hex	16#00	
"WriteData_Byte_8"	%AB11	Hex	16#00	
"WriteData_Byte_9"	%AB12	Hex	16#00	

Output data: Start read job
Byte 0: Control byte; 16#01 (0000_0001bin); the "Start Read" bit is set to 1, thereby starting the read job.

Name	Address	Displa...	Monitor value	Modify value
"ReadValid"	%E0.0	Bool	☐ FALSE	
"WriteValid"	%E0.1	Bool	☐ FALSE	
"TaskActive"	%E0.2	Bool	☒ TRUE	
"Error"	%E0.3	Bool	☐ FALSE	
"TagPresent"	%E0.4	Bool	☐ FALSE	
"ControlByte_InData"	%EB0	Bin	2#0000_0100	
"Length"	%EB1	DEC	0	
"RSSI"	%EB2	Hex	16#00	
"PowerTransmit_dBm"	%EB3	Hex	16#00	
"ReadData_Byte_1"	%EB4	Hex	16#00	
"ReadData_Byte_2"	%EB5	Hex	16#00	
"ReadData_Byte_3"	%EB6	Hex	16#00	
"ReadData_Byte_4"	%EB7	Hex	16#00	
"ReadData_Byte_5"	%EB8	Hex	16#00	
"ReadData_Byte_6"	%EB9	Hex	16#00	
"ReadData_Byte_7"	%EB10	Hex	16#00	
"ReadData_Byte_8"	%EB11	Hex	16#00	
"ReadData_Byte_9"	%EB12	Hex	16#00	

Input data: No tag

Byte 0: Control byte; 16#04 (0000_0100bin); the "Job active" bit is set to 1, thereby signaling that job execution has been activated. The "Read successful" bit is set to 0. There is no tag in the detection zone.

Name	Address	Displa...	Monitor value	Modify value
"ReadValid"	%E0.0	Bool	☒ TRUE	
"WriteValid"	%E0.1	Bool	☐ FALSE	
"TaskActive"	%E0.2	Bool	☒ TRUE	
"Error"	%E0.3	Bool	☐ FALSE	
"TagPresent"	%E0.4	Bool	☒ TRUE	
"ControlByte_InData"	%EB0	Bin	2#0001_0101	
"Length"	%EB1	DEC	8	
"RSSI"	%EB2	DEC	29	
"PowerTransmit_dBm"	%EB3	DEC	0	
"ReadData_Byte_1"	%EB4	Hex	16#31	
"ReadData_Byte_2"	%EB5	Hex	16#32	
"ReadData_Byte_3"	%EB6	Hex	16#33	
"ReadData_Byte_4"	%EB7	Hex	16#34	
"ReadData_Byte_5"	%EB8	Hex	16#35	
"ReadData_Byte_6"	%EB9	Hex	16#36	
"ReadData_Byte_7"	%EB10	Hex	16#37	
"ReadData_Byte_8"	%EB11	Hex	16#38	
"ReadData_Byte_9"	%EB12	Hex	16#00	

Input data: Data read; short-form data format

Byte 0: Control byte; 16#15 (0001_0101bin); the "Read successful" bit is set to 1, thereby signaling successful read access to a tag. The "Job active" bit is set to 1 because job execution has been activated. The "TagPresent" bit is set to 1 because at least one tag is located in the detection zone.

Byte 1: Length information; 16#08 (8 dec); the length of the transmitted information is 8 bytes.

Byte 2: RSSI value; 16#1D (29dec); the RSSI value for tag access is 29.

Byte 3: Transmitter radiated power in dBm; this information is not supported by all systems. When the IQH3 RFID read/write device is used, this byte has the value 16#00 (0 dec).

Byte 4 ... 11: Read data; data read from the tag's user memory.

Name	Address	Displa...	Monitor value	Modify value
"ReadValid"	%E0.0	Bool	 TRUE	
"WriteValid"	%E0.1	Bool	 FALSE	
"TaskActive"	%E0.2	Bool	 TRUE	
"Error"	%E0.3	Bool	 FALSE	
"TagPresent"	%E0.4	Bool	 TRUE	
"ControlByte_InData"	%EB0	Bin	2#0001_0101	
"Length"	%EB1	DEC	20	
"RSSI"	%EB2	DEC	88	
"PowerTransmit_dBm"	%EB3	DEC	0	
"ReadData_Byte_1"	%EB4	Hex	16#00	
"ReadData_Byte_2"	%EB5	Hex	16#08	
"ReadData_Byte_3"	%EB6	Hex	16#E0	
"ReadData_Byte_4"	%EB7	Hex	16#04	
"ReadData_Byte_5"	%EB8	Hex	16#01	
"ReadData_Byte_6"	%EB9	Hex	16#50	
"ReadData_Byte_7"	%EB10	Hex	16#E9	
"ReadData_Byte_8"	%EB11	Hex	16#DE	
"ReadData_Byte_9"	%EB12	Hex	16#0E	
"ReadData_Byte_10"	%EB13	Hex	16#7E	
"ReadData_Byte_11"	%EB14	Hex	16#00	
"ReadData_Byte_12"	%EB15	Hex	16#08	
"ReadData_Byte_13"	%EB16	Hex	16#31	
"ReadData_Byte_14"	%EB17	Hex	16#32	
"ReadData_Byte_15"	%EB18	Hex	16#33	
"ReadData_Byte_16"	%EB19	Hex	16#34	
"ReadData_Byte_17"	%EB20	Hex	16#35	
"ReadData_Byte_18"	%EB21	Hex	16#36	
"ReadData_Byte_19"	%EB22	Hex	16#37	
"ReadData_Byte_20"	%EB23	Hex	16#38	
"ReadData_Byte_21"	%EB24	Hex	16#00	

Input data: Data read; long-form data format

Byte 0: Control byte; 16#15 (0001_0101bin); the "Read successful" bit is set to 1, thereby signaling successful read access to a tag. The "Job active" bit is set to 1 because job execution has been activated. The "TagPresent" bit is set to 1 because at least one tag is located in the detection zone.

Byte 1: Length information; 16#14 (20 dec); the length of the transmitted information is 20 bytes.

Byte 2: RSSI value; 16#58 (88dec); the RSSI value for tag access is 88.

Byte 3: Transmitter radiated power in dBm; 16#00 (0dec); this information is not supported by all systems. When the IQH3 RFID read/write device is used, this byte has the value 16#00 (0 dec).

Byte 4 ... 5: UID length; 16#0008 (8dec); length of the UID of the read tag.

Byte 6 ... 13: UID; 16#E0...; UID (fixed code) of the read tag.

Byte 14 ... 15: Length of read data; 16#08 (8dec; length of the data read from the user memory.

Byte 16 ... 23: User memory data; data read from the tag's user memory.

Name	Address	Displ...	Monitor value	Modify value
"StartRead"	%A0.0	Bool	☐ FALSE	FALSE
"StartWrite"	%A0.1	Bool	☐ FALSE	
"ControlByte_OutData"	%AB0	Bin	2#0000_0000	
"Unused_1"	%AB1	Hex	16#00	
"Unused_2"	%AB2	Hex	16#00	
"Unused_3"	%AB3	Hex	16#00	
"WriteData_Byte_1"	%AB4	Hex	16#00	
"WriteData_Byte_2"	%AB5	Hex	16#00	
"WriteData_Byte_3"	%AB6	Hex	16#00	
"WriteData_Byte_4"	%AB7	Hex	16#00	
"WriteData_Byte_5"	%AB8	Hex	16#00	
"WriteData_Byte_6"	%AB9	Hex	16#00	
"WriteData_Byte_7"	%AB10	Hex	16#00	
"WriteData_Byte_8"	%AB11	Hex	16#00	
"WriteData_Byte_9"	%AB12	Hex	16#00	

Output data: End of read job

Byte 0: Control byte; 16#00 (0000_0000bin); the "Start Read" bit is reset to 0, thereby ending the read job.

Name	Address	Displa...	Monitor value	Modify value
"ReadValid"	%E0.0	Bool	☐ FALSE	
"WriteValid"	%E0.1	Bool	☐ FALSE	
"TaskActive"	%E0.2	Bool	☐ FALSE	
"Error"	%E0.3	Bool	☐ FALSE	
"TagPresent"	%E0.4	Bool	☐ FALSE	
"ControlByte_InData"	%EB0	Bin	2#0000_0000	
"Length"	%EB1	DEC	0	
"RSSI"	%EB2	DEC	0	
"PowerTransmit_dBm"	%EB3	DEC	0	
"ReadData_Byte_1"	%EB4	Hex	16#00	
"ReadData_Byte_2"	%EB5	Hex	16#00	
"ReadData_Byte_3"	%EB6	Hex	16#00	
"ReadData_Byte_4"	%EB7	Hex	16#00	
"ReadData_Byte_5"	%EB8	Hex	16#00	
"ReadData_Byte_6"	%EB9	Hex	16#00	
"ReadData_Byte_7"	%EB10	Hex	16#00	
"ReadData_Byte_8"	%EB11	Hex	16#00	
"ReadData_Byte_9"	%EB12	Hex	16#00	

Input data: Read job ended

Byte 0: Control byte; 16#00 (0000_0000bin); the "Job active" bit is reset to 0, thereby signaling the end of job execution.

Example: Read UID without autostart

Name	Address	Displ...	Monitor value	Modify value
'StartRead'	%A0.0	Bool	 TRUE	TRUE
'StartWrite'	%A0.1	Bool	 FALSE	
'ControlByte_OutData'	%AB0	Bin	2#0000_0001	
'Unused_1'	%AB1	Hex	16#00	
'Unused_2'	%AB2	Hex	16#00	
'Unused_3'	%AB3	Hex	16#00	
'WriteData_Byte_1'	%AB4	Hex	16#00	
'WriteData_Byte_2'	%AB5	Hex	16#00	
'WriteData_Byte_3'	%AB6	Hex	16#00	
'WriteData_Byte_4'	%AB7	Hex	16#00	
'WriteData_Byte_5'	%AB8	Hex	16#00	
'WriteData_Byte_6'	%AB9	Hex	16#00	
'WriteData_Byte_7'	%AB10	Hex	16#00	
'WriteData_Byte_8'	%AB11	Hex	16#00	
'WriteData_Byte_9'	%AB12	Hex	16#00	

Output data: Start read job
Byte 0: Control byte; 16#01 (0000_0001bin); the "Start Read" bit is set to 1, thereby starting the read job.

Name	Address	Displa...	Monitor value	Modify value
"ReadValid"	%E0.0	Bool	 FALSE	
"WriteValid"	%E0.1	Bool	 FALSE	
"TaskActive"	%E0.2	Bool	 TRUE	
"Error"	%E0.3	Bool	 FALSE	
"TagPresent"	%E0.4	Bool	 FALSE	
"ControlByte_InData"	%EB0	Bin	2#0000_0100	
"Length"	%EB1	DEC	0	
"RSSI"	%EB2	DEC	0	
"PowerTransmit_dBm"	%EB3	DEC	0	
"ReadData_Byte_1"	%EB4	Hex	16#00	
"ReadData_Byte_2"	%EB5	Hex	16#00	
"ReadData_Byte_3"	%EB6	Hex	16#00	
"ReadData_Byte_4"	%EB7	Hex	16#00	
"ReadData_Byte_5"	%EB8	Hex	16#00	
"ReadData_Byte_6"	%EB9	Hex	16#00	
"ReadData_Byte_7"	%EB10	Hex	16#00	
"ReadData_Byte_8"	%EB11	Hex	16#00	
"ReadData_Byte_9"	%EB12	Hex	16#00	

Input data: No tag
Byte 0: Control byte; 16#04 (0000_0100bin); the "Job active" bit is set to 1, thereby signaling that job execution has been activated. The "Read successful" bit is set to 0. There is no tag in the detection zone.

Name	Address	Displa...	Monitor value	Modify value
ReadValid	%E0.0	Bool	☒ TRUE	
WriteValid	%E0.1	Bool	☐ FALSE	
TaskActive	%E0.2	Bool	☒ TRUE	
Error	%E0.3	Bool	☐ FALSE	
TagPresent	%E0.4	Bool	☒ TRUE	
ControlByte_InData	%EB0	Bin	2#0001_0101	
Length	%EB1	DEC	10	
RSSI	%EB2	DEC	88	
PowerTransmit_dBm	%EB3	DEC	0	
ReadData_Byte_1	%EB4	Hex	16#00	
ReadData_Byte_2	%EB5	Hex	16#08	
ReadData_Byte_3	%EB6	Hex	16#E0	
ReadData_Byte_4	%EB7	Hex	16#04	
ReadData_Byte_5	%EB8	Hex	16#01	
ReadData_Byte_6	%EB9	Hex	16#50	
ReadData_Byte_7	%EB10	Hex	16#E9	
ReadData_Byte_8	%EB11	Hex	16#DE	
ReadData_Byte_9	%EB12	Hex	16#0E	
ReadData_Byte_10	%EB13	Hex	16#7E	
ReadData_Byte_11	%EB14	Hex	16#00	

Input data: Data read; long-form data format

Byte 0: Control byte; 16#15 (0001_0101bin); the "Read successful" bit is set to 1, thereby signaling successful read access to a tag. The "Job active" bit is set to 1 because job execution has been activated. The "TagPresent" bit is set to 1 because at least one tag is located in the detection zone.

Byte 1: Length information; 16#0A (10 dec); the length of the transmitted information is 10 bytes.

Byte 2: RSSI value; 16#58 (88dec); the RSSI value for tag access is 88.

Byte 3: Transmitter radiated power in dBm; 16#00 (0dec); this information is not supported by all systems. When the IQH3 RFID read/write device is used, this byte has the value 16#00 (0 dec).

Byte 4 ... 5: UID length; 16#0008 (8dec); length of the UID of the read tag.

Byte 6 ... 13: UID; 16#E0...; UID (fixed code) of the read tag.

Output data: End of read job

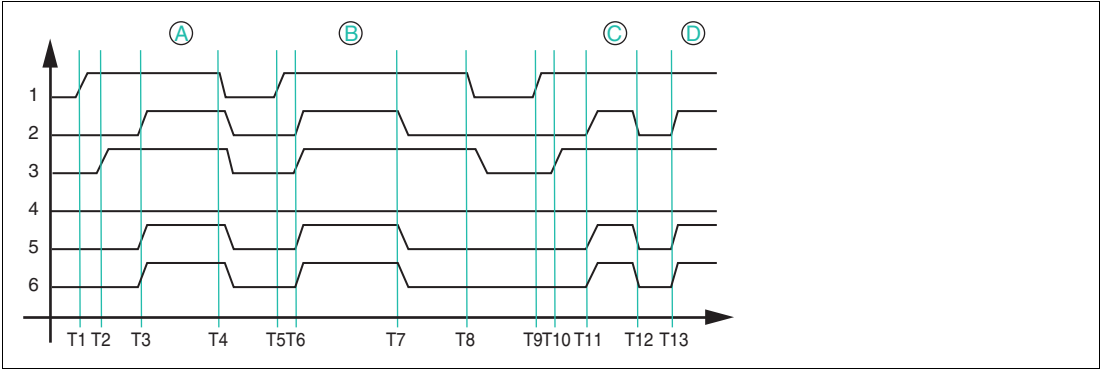
Byte 0: Control byte; 16#00 (0000_0000bin); the "Start Read" bit is reset to 0, thereby ending the read job.

Name	Address	Displa...	Monitor value	Modify value
StartRead	%A0.0	Bool	☐ FALSE	FALSE
StartWrite	%A0.1	Bool	☐ FALSE	
ControlByte_OutData	%AB0	Bin	2#0000_0000	
Unused_1	%AB1	Hex	16#00	
Unused_2	%AB2	Hex	16#00	
Unused_3	%AB3	Hex	16#00	
WriteData_Byte_1	%AB4	Hex	16#00	
WriteData_Byte_2	%AB5	Hex	16#00	
WriteData_Byte_3	%AB6	Hex	16#00	
WriteData_Byte_4	%AB7	Hex	16#00	
WriteData_Byte_5	%AB8	Hex	16#00	
WriteData_Byte_6	%AB9	Hex	16#00	
WriteData_Byte_7	%AB10	Hex	16#00	
WriteData_Byte_8	%AB11	Hex	16#00	
WriteData_Byte_9	%AB12	Hex	16#00	

Name	Address	Displa...	Monitor value	Modify value
"ReadValid"	%E0.0	Bool	☐ FALSE	
"WriteValid"	%E0.1	Bool	☐ FALSE	
"TaskActive"	%E0.2	Bool	☐ FALSE	
"Error"	%E0.3	Bool	☐ FALSE	
"TagPresent"	%E0.4	Bool	☐ FALSE	
"ControlByte_InData"	%EB0	Bin	2#0000_0000	
"Length"	%EB1	DEC	0	
"RSSI"	%EB2	DEC	0	
"PowerTransmit_dBm"	%EB3	DEC	0	
"ReadData_Byte_1"	%EB4	Hex	16#00	
"ReadData_Byte_2"	%EB5	Hex	16#00	
"ReadData_Byte_3"	%EB6	Hex	16#00	
"ReadData_Byte_4"	%EB7	Hex	16#00	
"ReadData_Byte_5"	%EB8	Hex	16#00	
"ReadData_Byte_6"	%EB9	Hex	16#00	
"ReadData_Byte_7"	%EB10	Hex	16#00	
"ReadData_Byte_8"	%EB11	Hex	16#00	
"ReadData_Byte_9"	%EB12	Hex	16#00	

Input data: Read job ended
 Byte 0: Control byte; 16#00 (0000_0000bin); the "Job active" bit is reset to 0, thereby signaling the end of job execution.

6.4.2.4 **Flowcharts**
Read Job without Autostart Function



- 1 Read
- 2 Read successful
- 3 Job active
- 4 Error
- 5 Tag in the sensing range
- 6 Data

If the autostart function is not used, the read job is started by the "Read" bit. The read job is executed until the "Read" bit is reset to FALSE.

- T1: Start of the read job by setting the "Read" bit to TRUE
- T2: The read job is executed and indicated by the "Job active" bit ("Job active" = TRUE)
- T3: Tag A enters the sensing range; "Read successful" and "Tag in detection zone" are set to TRUE, and the read data is located in the input field of the process data.

T4: Cancellation of the read job by resetting the "Read" bit to FALSE while the tag is in the detection zone; the "Job active," "Read successful," and "Tag in sensing range" bits are set to FALSE, and the process data is filled with 16#00.

T5: Start of the read job by setting the "Read" bit to TRUE; at the time of startup, tag B is already in the detection zone.

T6: The read job is executed ("Job active" = TRUE), and the data is read successfully ("Read successful" and "Tag in detection zone" = TRUE); the read data is located in the input field of the process data.

T7: The tag leaves the detection zone ("Read successful" and "Tag in sensing range" = FALSE); the area of the input field containing the read process data is set to 16#00.

T8: Cancellation of the read job ("Read" = FALSE); the "Job active" bit is reset; the "Job active," "Read successful," and "Tag in sensing range" bits are set to FALSE, and the process data is filled with 16#00.

T9: Start of the read job by setting the "Read" bit to TRUE; at the time of startup, no tag is in the detection zone; the read job remains active continuously.

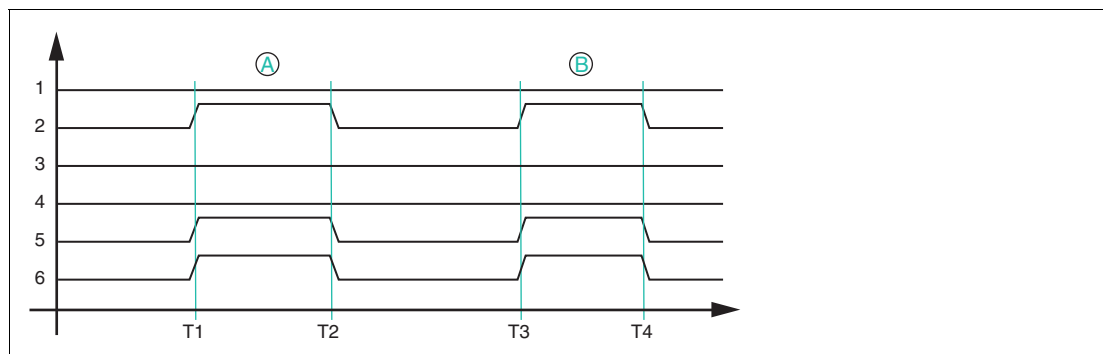
T10: The read job is executed ("Job active" = TRUE).

T11: Tag C enters the detection zone and the data is read ("Read successful" and "Tag in sensing range" = TRUE); the read data is located in the input field of the process data.

T12: Tag C leaves the detection zone ("Read successful" and "Tag in sensing range" = FALSE).

T13: Tag D enters the sensing range; behavior is identical to T11.

Read Job with Autostart Function



- 1 Read
- 2 Read successful
- 3 Job active
- 4 Error
- 5 Tag in the sensing range
- 6 Data

With the autostart function, read access is performed continuously, and the "Job active" bit of the input process data is set continuously.

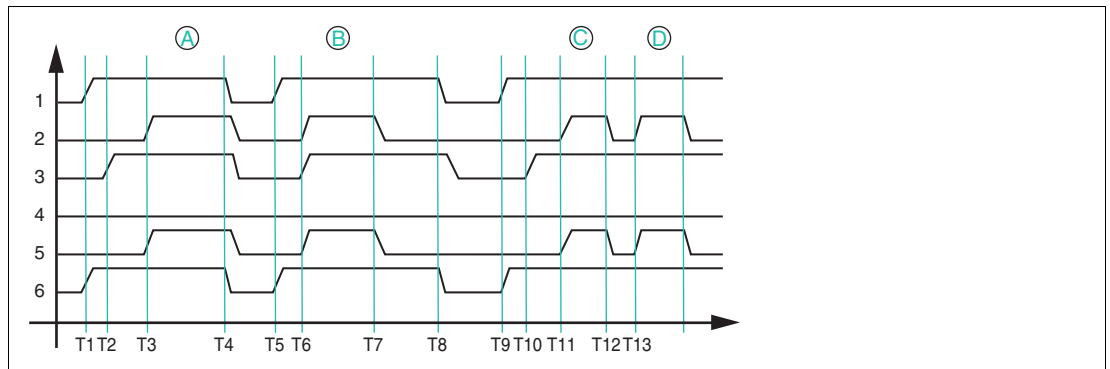
T1: Tag A enters the detection zone; "Read successful" and "Tag in sensing range" are set to TRUE, and the read data is located in the input field of the process data.

T2: Tag A leaves the detection zone; "Read successful" and "Tag in sensing range" are reset to FALSE; the area containing the previously read data is filled with 16#00.

T3: Tag B enters the detection zone; behavior is identical to T1.

T4: Tag B leaves the detection zone; behavior is identical to T2.

Write Job



- 1 Write
- 2 Write successful
- 3 Job active
- 4 Error
- 5 Tag in the sensing range
- 6 Data

Execution of a write job cannot be performed via the autostart function. To start a write job, set the "Write" bit to TRUE.

T1: Start of the write job by setting the "Write" bit to TRUE; at the same time, the user data to be written to the tag must be transferred to the output field of the process data.

T2: The write job is active ("Job active" = TRUE), and no tag is present in the detection zone ("Write successful" and "Tag in sensing range" = FALSE).

T3: Tag A enters the detection zone, and the data is written successfully ("Write successful" and "Tag in sensing range" = TRUE).

T4: Cancellation of the write job by resetting the "Write" bit to FALSE; the "Job active," "Write successful," and "Tag in sensing range" bits are reset to FALSE, and the user data is reset by the user to 16#00.

T5: Start of the write job by setting the "Write" bit to TRUE while simultaneously transferring the data to be written to the output field of the process data; at the time of startup, tag B is in the sensing range.

T6: The write job is active ("Job active" = TRUE), and tag B is written successfully ("Write successful" and "Tag in sensing range" = TRUE).

T7: Tag B leaves the sensing range ("Write successful" and "Tag in sensing range" = FALSE); the write job remains active ("Job active" = TRUE).

T8: Behavior is identical to T4.

T9: Start of the write job by setting the "Write" bit to TRUE; at the same time, the user data to be written to the tag is transferred to the output field of the process data.

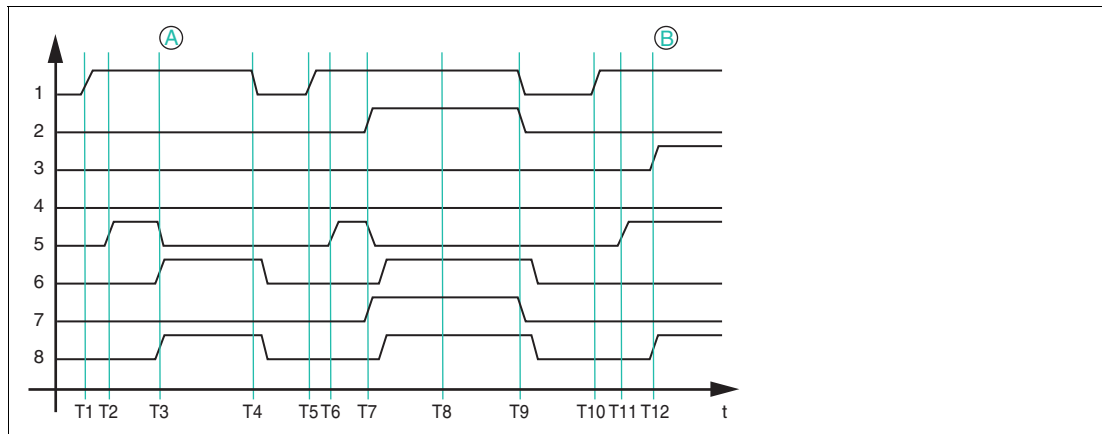
T10: The write job is active ("Job active" = TRUE), and no tag is present in the sensing range ("Write successful" and "Tag in sensing range" = FALSE).

T11: Tag C enters the sensing range, and the data is written successfully ("Write successful" and "Tag in sensing range" = TRUE); the write job remains active ("Job active" = TRUE).

T12: Behavior is identical to T7.

T13: Tag D enters the sensing range; behavior is identical to T11.

Error State



- 1 Read
- 2 Write
- 3 Read successful
- 4 Write successful
- 5 Job active
- 6 Error
- 7 Tag in the sensing range
- 8 Data (input)

If an error occurs during execution of a read or write job, this condition is signaled by the "Error" bit. At the same time, an error message is transmitted in the input process data.

T1: Start of the read job for 8 bytes of user memory by setting the "Read" bit to TRUE.

T2: The read job is active and is being executed ("Job active" = TRUE).

T3: A tag enters the detection zone, and the "Error" bit is set to TRUE. At the same time, the "Job active" bit is reset to FALSE, and error code 16#04 and the text "invalid command" are entered in the input field of the process data. This indicates that the read job configured via the device description file does not match the properties of the tag. The cause is the number of bytes to be read. A maximum of 4 bytes can be read from this tag during a read access operation. In this example, the value is set to 8. To rectify the error, correct the value in the device description file.

T4: Cancellation of the read job by resetting the "Read" bit to FALSE; the "Error" bit of the error message in the process data remains unchanged.

T5: Start of a new read job by setting the "Read" bit to TRUE; at the same time, the "Error" bit is reset to FALSE, and the error message in the input data is cleared.

T6: The read job is active and is being executed ("Job active" = TRUE).

T7: In addition, a write job is started by setting the "Write" bit to TRUE. The "Error" bit is set and "Job active" is reset to FALSE. An error message with error code 16#04 and the text "read AND write" is transmitted in the input field of the process data. This indicates that a read and a write job were triggered simultaneously. This is not permitted for the device.

T8: The error state remains active ("Error" = TRUE) because the "Read" and "Write" bits are set.

T9: The "Read" and "Write" bits are reset to FALSE.

T10: Start of the read job by setting the "Read" bit to TRUE; at the same time, the "Error" bit is reset to FALSE, and the error message in the input data is cleared.

T11: The read job is active and is being executed ("Job active" = TRUE).

T12: A tag enters the detection zone and is read successfully ("Read successful" and "Tag in sensing range" = TRUE).

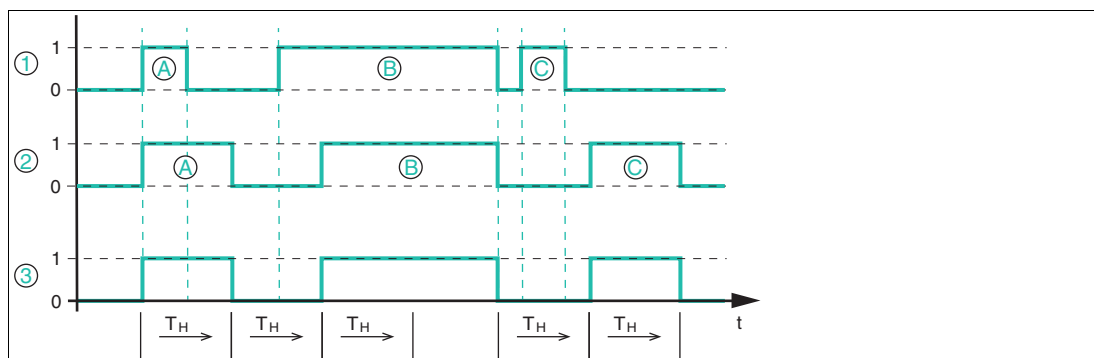
2026-04

6.4.2.5 Timing

In Easy Mode, a complex handshake procedure is not used for data transmission. Instead, telegrams are written to the input field of the process data (device → PLC) and remain there for a defined hold time. During this hold time, the device cannot modify the input data field.

The minimum hold time is 50 ms.

If a new telegram is generated by the device during the hold time (e.g., a new tag is read or a tag leaves the sensing range), it is written to the input field of the process data only after the 50 ms have elapsed. If no new telegram is generated during the hold time, the data in the input field of the process data remains unchanged.



- 1 Tag in the field
- 2 Data
- 3 Read successful

The figure above shows the basic timing sequence of data transmission as a function of the presence of a tag in the device's sensing range.

" T_H " corresponds to the device hold time of at least 50 ms.

The device is triggered either via the autostart function or by the "Read" bit. As a result, a read job is executed continuously on the device.

At the beginning, tag A enters the device's sensing range, and the signal state of the "Read successful" bit in the input field of the process data changes from FALSE to TRUE. The dwell time of the tag in the sensing range is less than 50 ms, and it leaves shortly after entering. The input field of the process data with the information for tag A remains valid for the period " T_H " (= 50 ms). Only after this period has elapsed is the input field of the process data updated again; it now contains the information "Read successful" = FALSE (no tag) and signals that the tag has left the sensing range. This telegram also remains in the input field of the process data for the hold time " T_H ".

Tag B enters the sensing range before the hold time of the previous telegram has elapsed. The input field of the process data is not updated until the 50 ms hold time has elapsed, and the signal state of the "Read successful" bit changes from FALSE to TRUE. At the same time, the read data is written to the input field of the process data. Tag B has a dwell time of more than 50 ms ($> T_H$) in the device's sensing range. During this period, the input field of the process data remains unchanged, and the signal state of the "Read successful" bit remains TRUE.

Tag B leaves the sensing range, and the signal state of the "Read successful" bit in the input field of the process data changes from TRUE to FALSE. Before the hold time " T_H " has elapsed, tag C enters the sensing range. The input field of the process data remains unchanged for the duration of the hold time and, after " T_H " has elapsed, the signal state of "Read successful" changes from FALSE to TRUE. This signals the presence of tag C, and the read data of this tag is transmitted.

Tag C leaves the sensing range before the hold time " T_H " has elapsed. After the hold time of the previous information has elapsed (tag B has left the sensing range), the process image is changed accordingly. The signal state of "Read successful" changes from FALSE to TRUE.

6.5 Expert Mode

6.5.1 Basic Command Sequence

The telegram length is configured in the device description file, in the same way as in Easy Mode (see chapter 6.4). The data is first combined to form a telegram. This telegram may be significantly larger than the configured length. Transmission of the telegram is performed step by step using individual fragments. The size of a fragment is a maximum of 256 bytes. The data transmission is controlled via a handshake procedure.

6.5.1.1 Telegram Structure

Legend

D_S	Delete bit; clears the device's internal buffer
U_M	Update Master Bit
U_S	Update Slave Bit
Frame Length	Number of valid bytes within the telegram
Fragmentation Counter	Number of fragments still to be transmitted
Telegram Length	Telegram length including all fragments
Command	Command code
Status	Command execution status
Parameter/Data	Parameters or data
Byte N	Number of bytes within the telegram; max. 255

OUTPUT telegram

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
0	ControlByte / frame length	D _S	U _M	U _S	0	<FrameLength>			
1	Frame length	<Frame Length>							
2	Fragmentation counter	<Fragmentation Counter>							
3	Telegram length (high byte)	<Telegram Length (High Byte)>							
4	Telegram length (low byte)	<Telegram Length (Low Byte)>							
5	Command	<Command>							
6	Parameter/data	<Data byte 1>							
7	Parameter/data	<Data byte 2>							
8	Parameter/data	...							
...	Parameter/data	<Data Byte X>							
...	Parameter/data	16#00							
...	Parameter/data	...							
N	Parameter/data	16#00							

Table 6.10

The value of <Frame Length> depends on how many <Data Byte> data values must be transmitted to execute a command. It defines the length of the fragment up to and including <Data Byte X>. If no additional command parameters are required to execute the command, the length of the fragment extends up to <Command> and has the value 16#06.

The <Fragmentation Counter> has the value 16#00, since the command can be transmitted in a single fragment by the controller.

The <Telegram Length> specifies the length of the telegram starting from the <Telegram Length> field itself and including <Data Byte X>. If no additional command parameters are transmitted, the telegram ends at <Command>, and <Telegram Length> has the value 16#03.

The <Command> byte defines the command to be executed. Different commands are executed depending on the value of <Command>. The following classes of commands are available:

- **Read/Write commands:** Access to one or more tags in the detection zone
- **System commands:** Execution of device settings; no access to tags
- **Filter commands:** Configuration of filters for tag access
- **Configuration commands:** Configuration of the properties of the connected RFID read/write device

The <Data Byte> fields are used to transmit the data required for execution of a command. This may include additional command parameters (e.g., start address) or user data to be written to a tag.

Unused areas within the telegram frame are set to the value 16#00.

INPUT Telegram

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
0	Control byte / frame length	D _S	U _M	U _S	0	<FrameLength>			
1	Frame Length	<Frame Length>							
2	Fragmentation Counter	<Fragmentation Counter>							
3	Telegram Length (High Byte)	<Telegram Length (High Byte)>							
4	Telegram Length (Low Byte)	<Telegram Length (Low Byte)>							
5	Command	<Command>							
6	Status	<Status>							
7	Parameter/data	<Data Byte 1>							
8	Parameter/data	<Data Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<Data Byte X>							
...	Parameter/data	16#00							
...	Parameter/data	...							
N	Parameter/data	16#00							

Table 6.11

The value of <Frame Length> depends on how many data values (<Data Byte>) are returned by the device in the command response. It specifies the length of the fragment up to and including <Data Byte X>. If no additional data values are included in the command response, the fragment length extends up to <Status> and has the value 16#07.

Since the command response can be transmitted in a single fragment by the controller, the <Fragmentation Counter> has the value 16#00.

The <Telegram Length> specifies the length of the telegram starting from the <Telegram Length> field itself and including <Data Byte X>. If no additional response parameters are transmitted, the telegram ends at <Status>, and <Telegram Length> has the value 16#04.

The <Command> byte is a mirror of the command code from the command in the response.

The value of <Status> indicates the command execution status. Error states are indicated by corresponding status values.

2026-04

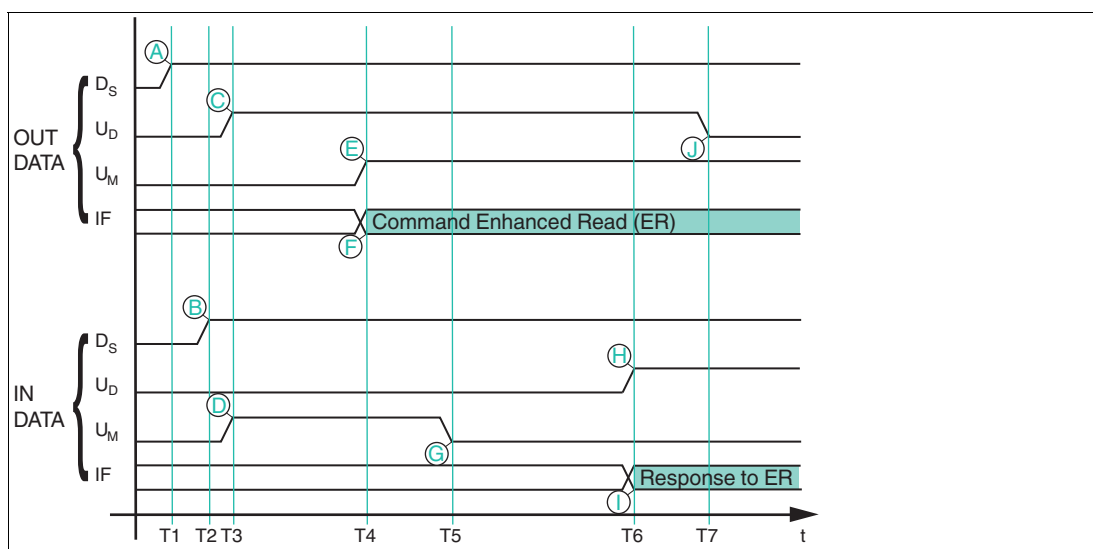
The <Data Byte> bytes contain the data returned by the device in response to the command. This may include, for example, data read from a tag or parameter values of the connected RFID read/write device.

Unused areas within the response telegram are set to 16#00.

6.5.1.2 Handshake Procedure

To ensure continuous, loss-free data transmission, the data flow between a PLC and the IDENTControl must be synchronized. The input data and output data are transmitted cyclically. Control of the data flow via the software is also referred to as "handshaking". The control bits required for this are contained in the control byte (see table "INPUT Telegram" on page 43 and see table "OUTPUT telegram" on page 42).

The following handshake procedure is used to transmit telegrams safely and quickly between the PLC and the device:



D_S Delete bit; deletes internal memory of the device

U_M Update master bit

U_D Update device bit

T1 The PLC changes the delete bit D_S in the output process data to High (A), which deletes the FIFO memory in the device.

T2 The IDENTControl changes the delete bit in the input process data (B) in response and deletes the entire contents of the FIFO memory

T3 The PLC mirrors the inverted state of U_S-INPUT from the input field to the output field (C). Likewise, the device mirrors the inverted state of U_M-OUTPUT to the input field (D). Both communication partners are indicating that they are ready to receive a telegram.

T4 The PLC enters an Enhanced Read command (ER) in IF-OUT (Ident Frame) (F). At the same time, the PLC applies U_M-INPUT in U_M-OUTPUT (E) and thus indicates the validity of a new telegram.

T5 The device mirrors the inverted state of U_M-OUTPUT to U_M-INPUT (G). This informs the control panel that the telegram has been received.

T6 The device has processed the ER and enters the response to the command in the input field (I). In the same telegram, U_S-OUTPUT is mirrored in U_S-INPUT (H).

T7 The PLC has received the changed U_S-INPUT and mirrors the inverted state in U_S-OUTPUT (J). Only now can the device send another telegram.

Example Implementation in the Controller

Slave delete bit D_S :

After the device is ready for operation (process data communication = OK), this instruction must be executed once. This deletes the internal telegram buffer. If an internal device error has occurred, the internal telegram buffer should also be deleted.

```
DS_OUTPUT := NOT DS_INPUT
```

Slave update bit U_S :

If the U_S bits in the input field and output field have the same value, new valid data is present in the input data field. The device does not write new read data to the input data field until the controller has read the input data, that is, until the U_S bit in the input field and output field has an inverted signal state.

To avoid blocking transmission of the read data, the inverted state of the U_S bit from the input data field must be transferred to the U_S bit of the output data field in every cycle.

```
US_OUTPUT := NOT US_INPUT (* copy the inverted update bit from the input telegram to the output telegram *)
```

Master update bit U_M :

Before a command is sent, it must be checked whether the device (slave) is ready to receive new telegrams. This is the case when the U_M bit in the input data field and the output data field has an inverted signal state.

The command parameters must then be transferred to the corresponding positions within the output data field.

As soon as the U_M bit in the output data field is set to the same signal state as the U_M bit in the input data field, the controller transmits the new command to the device.

```
OUTPUT[1..x] := new telegram
```

```
IF (UM_OUTPUT <> UM_INPUT) then (* check whether the slave can receive new data *)
```

```
UM_OUTPUT := UM_INPUT (* slave is ready to receive; copy update bit *)
```

```
End_IF
```

6.5.2 Command Overview

The IDENTControl supports the following commands. For a detailed description of the individual commands, refer to the manuals for the respective RFID read/write device used.

Read/Write Commands

Abbreviation	Command code	Command description	System
SF	16#01	Single Read Fixcode Single read access to fixed code, UID or TID	IPH, IQH*, IUH*
EF	16#1D	Enhanced Read Fixcode Continuous read access to fixed code, UID, or TID	IPH, IQH*, IUH*
SX	16#1F	Single Write Fixcode Single write access to fixed code	IPH
EX	16#24	Enhanced Write Fixcode Continuous write access to fixed code	IPH
SN	16#D2	Single Read UII Single read access to UII/EPC	IUH*
EN	16#D3	Enhanced Read UII Continuous read access to UII/EPC	IUH*
SS	16#0A	Single Read Special Fixcode Single read access to EPC/UII This command is implemented for compatibility reasons and should no longer be used in new applications.	IUH*
ES	16#71	Enhanced Read Special Fixcode Continuous read access to UII/EPC This command is implemented for compatibility reasons and should no longer be used in new applications.	IUH*
#SU	16#CE	Single Write UII Single write access to UII/EPC	IUH*
SP	16#0D	Single Program Special Fixcode Single write access to the EPC/UII This command is implemented for compatibility reasons and should no longer be used in new applications.	IUH*
EP	16#75	Enhanced Program Special Fixcode This command is implemented for compatibility reasons and should no longer be used in new applications.	IUH1
#SR	16#49	Single Read 2-Byte Words Single read access to user memory	IUH*
#ER	16#4B	Enhanced Read 2-Byte Words Continuous read access to user memory	IUH*
#SW	16#4A	Single Write 2-Byte Words Single write access to user memory	IUH*
#EW	16#4C	Enhanced Write 2-Byte Words Continuous write access to user memory	IUH*
SR	16#10	Single Read 4-Byte Blocks Single read access to user memory	IPH, IQH*, IUH*
ER	16#19	Enhanced Read 4-Byte Blocks Continuous read access to user memory	IPH, IQH*, IUH*
SW	16#40	Single Write 4-Byte Blocks Single write access to user memory	IPH, IQH*, IUH*

2026-04

Abbrevi- ation	Command code	Command description	System
EW	16#1A	Enhanced Write 4-Byte Blocks Continuous write access to user memory	IPH, IQH*, IUH*
SL	16#47	Single Write 4-Byte Blocks with Lock Single write access to user memory with write protection	IQH*
EL	16#48	Enhanced Write 4-Byte Blocks with Lock Continuous write access to user memory with write protection	IQH*
SG	16#61	Single Read Configuration	IPH, IQH*
EG	16#68	Enhanced Read Configuration	IPH, IQH*
SC	16#12	Single Write Configuration	IPH, IQH*
EC	16#66	Enhanced Write Configuration	IPH, IQH*
X#SR	16#DF	Selective Single Read 2-Byte Words	IUH1
X#ER	16#D8	Selective Enhance Read 2-Byte Words	IUH1
X#SW	16#E2	Selective Single Write 2-Byte Words	IUH1
X#EW	16#DB	Selective Enhanced Write 2-Byte Words	IUH1
XSR	16#E0	Selective Single Read 4-Byte Blocks	IQH3, IQH4, IUH1
XER	16#D9	Selective Enhanced Read 4-Byte Blocks	IQH3, IQH4, IUH1
XSW	16#E3	Selective Single Write 4-Byte Blocks	IQH3, IQH4, IUH1
XEW	16#DC	Selective Enhanced Write 4-Byte Blocks	IQH3, IQH4, IUH1
XSL	16#E4	Selective Single Write 4-Byte Blocks with Lock	IQH3, IQH4
XEL	16#E5	Selective Enhances Write 4-Byte Blocks with Lock	IQH3, IQH4
XSN	16#DE	Selective Single Read Ull	IUH1
XEN	16#D7	Selective Enhanced Read Ull	IUH1
X#SU	16#E6	Selective Single Write Ull	IUH1
XSG	16#DD	Selective Single Read Configuration	IQH3, IQH4
XEG	16#D6	Selective Enhanced Read Configuration	IQH3, IQH4
XSC	16#E1	Selective Single Write Configuration	IQH3, IQH4
XEC	16#DA	Selective Enhanced Write Configuration	IQH3, IQH4
XSF	16#E8	Selective Single Read Fixcode	IQH3, IQH4, IUH1
XEF	16#E9	Selective Enhanced Read Fixcode	IQH3, IQH4, IUH1
XS#	16#E7	Selective Fill Tag	IQH3, IQH4
PM	16#18	Set Password Mode	IPH
PC	16#41	Change Password	IPH
PS	16#42	Set Password	IPH
S#	16#AA	Fill Tag	IQH*
KI	16#B9	Kill Tag	IUH*
LO	16#D5	Lock Tag	IUH*

System Commands

Abbreviation	Command code	Command description	System
VE	16#03	Version Reads the firmware version of the connected RFID read/write device	IPH, IQH*, IUH*
CT	16#04	Change Tag Set the tag type	IPH, IQH*
RS	16#16	Reset Power cycle of the connected RFID read/write device	IPH, IQH*, IUH*
GS	16#95	Get State Info Reads status information from the RFID read/write devices	IPH, IQH1
RD	16#B4	Reset to Default	IDENTControl
QU	16#02	Quit Aborts an enhanced command	IPH, IQH*, IUH*
CS	16#17	Command Store Stores a command that is executed automatically after a restart	IDENTControl

Filter Commands

Abbreviation	Command code	Command description	System
FI	16#CA	Filter Initialization	IUH*
MF	16#CB	Filter Mode	IUH*

In the factory setting, every read/write command accesses all detected tags within the detection zone. The filters can be used to make a selection. This makes it possible to specify that read/write commands access only a single tag or multiple tags. To parameterize access to specific tags, the commands "Set Filter Mask" (FI) and "Activate Filter" (MF) must be used. The "Set Filter Mask" (FI) command sets filter masks or filter conditions in the device. Filtering is activated by the "Activate Filter" (MF) command.

These commands cause the filter settings to be stored persistently and remain in effect even after a restart. Frequent changes to these settings (> 100,000 changes) should be avoided. For regular selective reading or writing, the corresponding selective read/write commands must be used.

Configuration Commands

Abbreviation	Command code	Command description	System
RP	16#BE	Read Parameter Reads device parameters	IQH3/4, IUH*
WP	16#BF	Write Parameter Writes device parameters	IQH3/4, IUH*

Service Commands

Abbreviation	Command code	Command description	System
TH	16#90	Test Head Creates a Test Function (TF) command and sends it to the RFID read/write device	IDENTControl
GF	16#9E	Get Frequency	IPH
LE	16#CC	LED	IDENTControl
IV	16#BD	Inventory	IQH*
TI	16#BC	Set Tag ID System	IQH*

6.5.3 Parameter Overview

The following parameters can be read or written using the configuration commands Read Parameters (RP) and Write Parameters (WP):

Abbreviation	Parameter description	Read/write	System
PT 16#5054	Power Transmit Sets the transmitter radiated power of the device	Read + write	IQH3, IQH4, IUH*
NT 16#4E54	Number of Tags Abort criterion for the read/write command	Read + write	IQH3, IQH4, IUH*
AP 16#4150	Antenna polarization Sets the polarization plane of the device	Read + write	IUH*
TA 16#5441	Tries Allowed Number of attempts to access a tag	Read + write	IQH3, IQH4, IUH*
IF 16#4946	Information Outputs additional information about tag access	Read + write	IQH3, IQH4, IUH*
RC 16#5243	Region Code Country code	Read	IUH*
CD 16#4344	Transmit channels Configures the transmit channels used	Read + write	IUH*
NC 16#4E43	Number of Channels Configures the transmit channels used	Read + write	IUH*
E5 16#4535	Enhanced Status 5 Sets the number of unsuccessful attempts to access a tag before a Status 5 message is issued	Read + write	IQH3, IQH4, IUH*
MB 16#4D42	Memory Bank Sets the memory bank for the Read/Write 2-byte Words and 4-byte Blocks commands	Read + write	IUH*
MF 16#4D46	Measure Reflexion Measures the reflected transmitter radiated power	Read	IUH
QW 16#5157	Q Value Configures the time slots for accessing tags	Read + write	IQH3, IQH4, IUH*
RD 16#5244	Reset to Default Resets to factory setting	Write	IQH3, IQH4, IUH*
SM 16#534D	Sensing Mode Configures transmit pauses	Read + write	IUH*

Abbreviation	Parameter description	Read/write	System
FA 16#4641	Filter Mode Reads the filter mode (written via the MF command)	Read	IUH*
FL 16#464C	Filter List Reads the filter mask (written via the FI command)	Read	IUH*
QV 16#5156	Protocol Mode Configures the protocol mode	Read + write	IQH3, IQH4, IUH*
SE 16#5345	Session Sets the session flags	Read + write	IUH*
FB 16#4642	Search Mode Configures the air interface search algorithm	Read + write	IUH*
FR 16#4652	Sequence Mode Configures the behavior for write access (Service)	Read + write	IQH3, IQH4, IUH*
EF 16#4546	Expected Fixcode Size Configures the fixed code size for tag access	Read + write	IUH*
MD 16#4D44	Retrieve Additional Information Outputs additional information about tag access (single frame)	Read	IUH
AF 16#4146	Configures the Application Family Identifier (AFI) request filter for tag access	Read + write	IQH3, IQH4
DN 16#444E	Device Serial Number (Service)	Read	IQH4, IUH1
ER 16#4552	Error Message (Service)	Read + write	IQH4, IUH1
GT 16#4745	Get Tag Type (Service)	Read	IQH*
ND 16#4E44	Noise Detection Level Configures the interference signal threshold indicated by the red LED	Read + write	IQH3, IQH4
OH 16#4F48	Operating Hours Reads the operating hours counter	Read	IQH3, IQH4, IUH1
PI 16#5049	Parameter Info Reads the parameter value range (Service)	Read	IUH
PR 16#5052	Production Interface Reads the production interface status (Service)	Read	IQH3, IQH4, IUH1
RV 16#5256	SVN Revision Reads the software revision number (Service)	Read	IQH3, IQH4, IUH*
SN 16#534E	Get Serial Number Reads the microcontroller serial number (Service)	Read	IQH3, IQH4, IUH*
T2 16#5432	Tag for Application	Read + write	IQH3, IQH4, IUH1
T3 16#5433	Tag for Function	Read + write	IQH3, IQH4, IUH1
T4 16#5434	Tag for Location	Read + write	IQH3, IQH4, IUH1

Abbreviation	Parameter description	Read/write	System
T5 16#5435	Tag for Installation	Read + write	IQH3, IQH4, IUH1
TE 16#5445	Temperature Reads the device temperature	Read	IQH3, IQH4, IUH*
TF 16#5446	Test Function Reads internal diagnostic data (e.g., voltage) (service)	Read	IUH
TI 16#5449	Tag ID Code Configures the tag filter	Read + write	IQH3, IQH4, IUH
TO 16#544F	Overtemperature Handling Configures the device behavior in the event of excess temperature	Read + write	IQH3, IQH4
UB 16#5542	Host UART Baud Rate (Service)	Read + write	IQH3, IQH4, IUH1
UT 16#5554	Host UART Timeout (Service)	Read + write	IQH3, IQH4, IUH1
VO 16#564F	Voltages Reads internal voltage and current values	Read	IQH3, IQH4, IUH1
DR 16#4452	Data Rate	Read + write	IQH3, IQH4
CT 16#4354	Change Tag Sets the tag type	Read + write	IQH3, IQH4
ST 16#5354	Status Frontend Reads the frontend status	Read	IQH3, IQH4
BB 16#4242	Back to Box Resets the read/write device to the delivery state, except for the operating hours counter	Write	IQH3, IQH4, IUH1
DS 16#4453	Data Swap Swaps blocks in block units	Read + write	IQH3, IQH4
LM 16#4C4D	LED Masking (Service)	Read + write	IQH3, IQH4

The parameters are stored in non-volatile memory in the device.

The parameters are not checked by the IDENTControl. A description of the parameters, including value ranges and default settings, is provided in the manual for the relevant RFID read/write device. Only an example of a Read Parameters or Write Parameters command is shown here (using Power Transmit as an example).

6.5.4 Command Examples



Note

Ensure that the selected command is suitable for the connected RFID read/write device (IPH, IQH, IUH); see the Command Overview table.

For an explanation of the commands, refer to the Command Overview table (including information on non-preferred commands).

For an explanation of the control byte, see chapter 6.5.1.2.

The exact content of the command and response telegrams depends in part on the system used. For more detailed explanations, refer to the relevant manual for the connected RFID read/write device. Only the short-form data format is shown here as an example.

6.5.4.1 Single Read Commands

Command Telegram (PLC → IDENTControl)

Byte	Content	SF	SN	SR	#SR	SS	SG (IQH*)	XSF (IUH1; 2 Byte UII / EPC Mask)	XSN (IUH1; 2 Byte UII / EPC Mask)	XSR (IUH1; 2 Byte UII / EPC Mask, 4 Byte User Memory)	X#SR (IUH1; 2 Byte UII / EPC Mask, 2 Byte User Memory)	XSG (IQH3, IQH4; 2 Byte UID Mask)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0
1	Frame Length	16#06	16#06	16#0A	16#0A	16#07	16#07	16#0A	16#0A	16#0E	16#0E	16#0B
2	Frag-mentation Counter	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
3	Tele-gram Length (High Byte)	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
4	Tele-gram Length (Low Byte)	16#03	16#03	16#07	16#07	16#04	16#04	16#07	16#07	16#0B	16#0B	16#08
5	Com-mand	16#01	16#D2	16#10	16#49	16#0A	16#61	16#E8	16#DE	16#E0	16#DF	16#DD
6	Com-mand param-e-ters	16#00	16#00	<Start Address (High Byte)>	<Start Address (High Byte)>	16#00 <Fix Length>	16#00 <Config Address>	16#00 <Length UII / EPC Mask (High Byte)>	16#00 <Length UII / EPC Mask (High Byte)>	16#00 <Length UII / EPC Mask (High Byte)>	16#00 <Length UII / EPC Mask (High Byte)>	16#00 <Length UID Mask (High Byte)>
7	Com-mand param-e-ters	16#00	16#00	<Start Address (Low Byte)>	<Start Address (Low Byte)>	16#00	16#00	16#10 <Length UII / EPC Mask (Low Byte)>	16#10 <Length UII / EPC Mask (Low Byte)>	16#10 <Length UII / EPC Mask (Low Byte)>	16#10 <Length UII / EPC Mask (Low Byte)>	16#10 <Length UID Mask (Low Byte)>
8	Com-mand param-e-ters	16#00	16#00	<Num-ber of Bytes (High Byte)>	<Num-ber of Bytes (High Byte)>	16#00	16#00	<UII / EPC Mask Byte 1>	<UII / EPC Mask Byte 1>	<UII / EPC Mask Byte 1>	<UII / EPC Mask Byte 1>	<UID Mask Byte 1>
9	Com-mand param-e-ters	16#00	16#00	<Num-ber of Bytes (Low Byte)>	<Num-ber of Bytes (Low Byte)>	16#00	16#00	<UII / EPC Mask Byte 2>	<UII / EPC Mask Byte 2>	<UII / EPC Mask Byte 2>	<UII / EPC Mask Byte 2>	<UID Mask Byte 2>
10	Com-mand param-e-ters	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00 <Start Address (High Byte)>	16#00 <Start Address (High Byte)>	16#00 <Config Address>
11	Com-mand param-e-ters	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00 <Start Address (Low Byte)>	16#00 <Start Address (Low Byte)>	16#00

2026-04

Byte	Content	SF	SN	SR	#SR	SS	SG (IQH*)	XSF (IUH1; 2 Byte Ull / EPC Mask)	XSN (IUH1; 2 Byte Ull / EPC Mask)	XSR (IUH1; 2 Byte Ull / EPC Mask, 4 Byte User Memory)	X#SR (IUH1; 2 Byte Ull / EPC Mask, 2 Byte User Memory)	XSG (IQH3, IQH4; 2 Byte UID Mask)
12	Com- mand param- eters	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00 <Num- ber of Bytes (High Byte)>	16#00 <Num- ber of Bytes (High Byte)>	16#00
13	Com- mand param- eters	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#04 <Num- ber of Bytes (Low Byte)>	16#02 <Num- ber of Bytes (Low Byte)>	16#00
...	Com- mand param- eters	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
255	Com- mand param- eters	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Response telegram 1 – Status 16#00(IDENTControl →) SPS)

Byte	Content	SF (IQH*)	SN (IUH*; 12 Byte Ull / EPC)	SR (IQH*; 4 Byte User Memory)	#SR (IUH*; 2 Byte User Memory)	SS (IUH*; 12 Byte Ull / EPC)	SG (IQH*)	XSF (IUH1)	XSN (IUH1)	XSR (IUH1; 4 Byte User Memory)	X#SR (IUH1; 2 Byte User Memory)	XSG (IQH3, IQH4)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0
1	Frame Length	16#0F	16#15	16#0B	16#09	16#15	16#15	16#13	16#15	16#0B	16#09	16#15
2	Frag- menta- tion Counter	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
3	Tele- gram Length (High Byte)	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
4	Tele- gram Length (Low Byte)	16#0C	16#12	16#08	16#06	16#12	16#12	16#10	16#12	16#08	16#06	16#12
5	Com- mand	16#01	16#D2	16#10	16#49	16#0A	16#61	16#E8	16#DE	16#E0	16#DF	16#DD
6	Status	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
7	Data	16#E0 <UID Byte 1>	<PC- Word (High Byte)>	<Data Byte 1>	<Data Byte 1>	<PC- Word (High Byte)>	<Info Flags>	16#E2 <TID Byte 1>	<PC- Word (High Byte)>	<Data Byte 1>	<Data Byte 1>	<Info Flags>
8	Data	<UID Byte 2>	<PC- Word (Low Byte)>	<Data Byte 2>	<Data Byte 2>	<PC- Word (Low Byte)>	<UID Byte 8>	<TID Byte 2>	<PC- Word (Low Byte)>	<Data Byte 2>	<Data Byte 2>	<UID Byte 8>
9	Data	<UID Byte 3>	<EPC / Ull Byte 1>	<Data Byte 3>	16#00	<EPC / Ull Byte 1>	<UID Byte 7>	<TID Byte 3>	<EPC / Ull Byte 1>	<Data Byte 3>	16#00	<UID Byte 7>
10	Data	<UID Byte 4>	<EPC / Ull Byte 2>	<Data Byte 4>	16#00	<EPC / Ull Byte 2>	<UID Byte 6>	<TID Byte 4>	<EPC / Ull Byte 2>	<Data Byte 4>	16#00	<UID Byte 6>
11	Data	<UID Byte 5>	<EPC / Ull Byte 3>	16#00	16#00	<EPC / Ull Byte 3>	<UID Byte 5>	<TID Byte 5>	<EPC / Ull Byte 3>	16#00	16#00	<UID Byte 5>
12	Data	<UID Byte 6>	<EPC / Ull Byte 4>	16#00	16#00	<EPC / Ull Byte 4>	<UID Byte 4>	<TID Byte 6>	<EPC / Ull Byte 4>	16#00	16#00	<UID Byte 4>

Byte	Content	SF (IQH*)	SN (IUH*; 12 Byte UII / EPC)	SR (IQH*; 4 Byte User Memory)	#SR (IUH*; 2 Byte User Memory)	SS (IUH*; 12 Byte UII / EPC)	SG (IQH*)	XSF (IUH1)	XSN (IUH1)	XSR (IUH1; 4 Byte User Memory)	X#SR (IUH1; 2 Byte User Memory)	XSG (IQH3, IQH4)
13	Data	<UID Byte 7>	<EPC / UII Byte 5>	16#00	16#00	<EPC / UII Byte 5>	<UID Byte 3>	<TID Byte 7>	<EPC / UII Byte 5>	16#00	16#00	<UID Byte 3>
14	Data	<UID Byte 8>	<EPC / UII Byte 6>	16#00	16#00	<EPC / UII Byte 6>	<UID Byte 2>	<TID Byte 8>	<EPC / UII Byte 6>	16#00	16#00	<UID Byte 2>
15	Data	16#00	<EPC / UII Byte 7>	16#00	16#00	<EPC / UII Byte 7>	<UID Byte 1>	<TID Byte 9>	<EPC / UII Byte 7>	16#00	16#00	<UID Byte 1>
16	Data	16#00	<EPC / UII Byte 8>	16#00	16#00	<EPC / UII Byte 8>	<DSFID>	<TID Byte 10>	<EPC / UII Byte 8>	16#00	16#00	<DSFID>
17	Data	16#00	<EPC / UII Byte 9>	16#00	16#00	<EPC / UII Byte 9>	<AFI>	<TID Byte 11>	<EPC / UII Byte 9>	16#00	16#00	<AFI>
18	Data	16#00	<EPC / UII Byte 10>	16#00	16#00	<EPC / UII Byte 10>	<VICC Memory Size (Number of Blocks - 1)>	<TID Byte 12>	<EPC / UII Byte 10>	16#00	16#00	<VICC Memory Size (Number of Blocks - 1)>
19	Data	16#00	<EPC / UII Byte 11>	16#00	16#00	<EPC / UII Byte 11>	<VICC Memory Size (Block Size - 1)>	16#00	<EPC / UII Byte 11>	16#00	16#00	<VICC Memory Size (Block Size - 1)>
20	Data	16#00	<EPC / UII Byte 12>	16#00	16#00	<EPC / UII Byte 12>	<IC Reference>	16#00	<EPC / UII Byte 12>	16#00	16#00	<IC Reference>
21	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
...	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
255	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Response telegram 2 – Status 16#0B (IDENTControl → SPS)

Byte	Content	SF (IQH*)	SN (IUH*)	SR (IQH*)	#SR (IUH*)	SS (IUH*)	SG (IQH*)	XSF (IUH1)	XSN (IUH1)	XSR (IUH1; 4 Byte User Memory)	X#SR (IUH1)	XSG (IQH*)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0
1	Frame Length	16#09	16#0C	16#09	16#0C	16#0C	16#09	16#0C	16#0C	16#0C	16#0C	16#09
2	Fragmentation Counter	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
3	Telegram Length (High Byte)	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
4	Telegram Length (Low Byte)	16#06	16#09	16#06	16#09	16#09	16#06	16#09	16#09	16#09	16#09	16#06
5	Command	16#01	16#D2	16#10	16#49	16#0A	16#61	16#E8	16#DE	16#E0	16#DF	16#DD
6	Status	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B
7	Data	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>
8	Data	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>
9	Data	16#00	<Transmission Channel>	16#00	<Transmission Channel>	<Transmission Channel>	16#00	<Transmission Channel>	<Transmission Channel>	<Transmission Channel>	<Transmission Channel>	16#00

2026-04

Byte	Content	SF (IQH*)	SN (IUH*)	SR (IQH*)	#SR (IUH*)	SS (IUH*)	SG (IQH*)	XSF (IUH1)	XSN (IUH1)	XSR (IUH1; 4 Byte User Mem- ory)	X#SR (IUH1)	XSG (IQH*)
10	Data	16#00	<Trans- mission Power (High Byte)>	16#00	<Trans- mission Power (High Byte)>	<Trans- mission Power (High Byte)>	16#00	<Trans- mission Power (High Byte)>	<Trans- mission Power (High Byte)>	<Trans- mission Power (High Byte)>	<Trans- mission Power (High Byte)>	16#00
11	Data	16#00	<Trans- mission Power (Low Byte)>	16#00	<Trans- mission Power (Low Byte)>	<Trans- mission Power (Low Byte)>	16#00	<Trans- mission Power (Low Byte)>	<Trans- mission Power (Low Byte)>	<Trans- mission Power (Low Byte)>	<Trans- mission Power (Low Byte)>	16#00
...	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
255	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Response telegram 3 – Status 16#0F (IDENTControl → SPS)

Byte	Content	SF (IQH*)	SN (IUH*)	SR (IQH*)	#SR (IUH*)	SS (IUH*)	SG (IQH*)	XSF (IUH1)	XSN (IUH1)	XSR (IUH1; 4 Byte User Mem- ory)	X#SR (IUH1)	XSG (IQH*)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0
1	Frame Length	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B
2	Frag- menta- tion Counter	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
3	Tele- gram Length (High Byte)	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
4	Tele- gram Length (Low Byte)	16#08	16#08	16#08	16#08	16#08	16#08	16#08	16#08	16#08	16#08	16#08
5	Com- mand	16#01	16#D2	16#10	16#49	16#0A	16#61	16#E8	16#DE	16#E0	16#DF	16#DD
6	Status	16#0F	16#0F	16#0F	16#0F	16#0F	16#0F	16#0F	16#0F	16#0F	16#0F	16#0F
7	Data	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30
8	Data	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30
9	Data	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30
10	Data	16#31	16#31	16#31	16#31	16#31	16#31	16#31	16#31	16#31	16#31	16#31
...	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
255	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

6.5.4.2 Enhanced Read Commands

Command Telegram (PLC → IDENTControl)

Byte	Content	EF	EN	ER	#ER	ES	EG (IQH*)	XEF (IUH1; 2 Byte UII / EPC Mask)	XEN (IUH1; 2 Byte UII / EPC Mask)	XER (IUH1; 2 Byte UII / EPC Mask, 4 Byte User Memory)	X#ER (IUH1; 2 Byte UII / EPC Mask, 2 Byte User Memory)	XEG (IQH3, IQH4; 2 Byte UID Mask)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0
1	Frame Length	16#06	16#06	16#0A	16#0A	16#07	16#07	16#0A	16#0A	16#0E	16#0E	16#0B
2	Frag-mentation Counter	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
3	Tele-gram Length (High Byte)	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
4	Tele-gram Length (Low Byte)	16#03	16#03	16#07	16#07	16#04	16#04	16#07	16#07	16#0B	16#0B	16#08
5	Com-mand	16#1D	16#D3	16#19	16#4B	16#71	16#68	16#E9	16#D7	16#D9	16#D8	16#D6
6	Com-mand param-e-ters	16#00	16#00	<Start Address (High Byte)>	<Start Address (High Byte)>	16#00 <Fix Length>	16#00 <Config Address>	16#00 <Length UII / EPC Mask (High Byte)>	16#00 <Length UII / EPC Mask (High Byte)>	16#00 <Length UII / EPC Mask (High Byte)>	16#00 <Length UII / EPC Mask (High Byte)>	16#00 <Length UID Mask (High Byte)>
7	Com-mand param-e-ters	16#00	16#00	<Start Address (Low Byte)>	<Start Address (Low Byte)>	16#00	16#00	16#10 <Length UII / EPC Mask (Low Byte)>	16#10 <Length UII / EPC Mask (Low Byte)>	16#10 <Length UII / EPC Mask (Low Byte)>	16#10 <Length UII / EPC Mask (Low Byte)>	16#10 <Length UID Mask (Low Byte)>
8	Com-mand param-e-ters	16#00	16#00	<Num-ber of Bytes (High Byte)>	<Num-ber of Bytes (High Byte)>	16#00	16#00	<UII / EPC Mask Byte 1>	<UII / EPC Mask Byte 1>	<UII / EPC Mask Byte 1>	<UII / EPC Mask Byte 1>	<UID Mask Byte 1>
9	Com-mand param-e-ters	16#00	16#00	<Num-ber of Bytes (Low Byte)>	<Num-ber of Bytes (Low Byte)>	16#00	16#00	<UII / EPC Mask Byte 2>	<UII / EPC Mask Byte 2>	<UII / EPC Mask Byte 2>	<UII / EPC Mask Byte 2>	<UID Mask Byte 2>
10	Com-mand param-e-ters	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00 <Start Address (High Byte)>	16#00 <Start Address (High Byte)>	16#00 <Config Address>
11	Com-mand param-e-ters	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00 <Start Address (Low Byte)>	16#00 <Start Address (Low Byte)>	16#00
12	Com-mand param-e-ters	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00 <Num-ber of Bytes (High Byte)>	16#00 <Num-ber of Bytes (High Byte)>	16#00
13	Com-mand param-e-ters	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#04 <Num-ber of Bytes (Low Byte)>	16#02 <Num-ber of Bytes (Low Byte)>	16#00
...	Com-mand param-e-ters	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
255	Com-mand param-e-ters	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

2026-04

Response telegram 1 – Status 16#00(IDENTControl →) SPS)

Byte	Content	EF (IQH*)	EN (IUH*; 12 Byte Ull / EPC)	ER (IQH*; 4 Byte User Memory)	#ER (IUH*; 2 Byte User Memory)	ES (IUH*; 12 Byte Ull / EPC)	EG (IQH*)	XEF (IUH1)	XEN (IUH1)	XSR (IUH1; 4 Byte User Memory)	X#ER (IUH1; 2 Byte User Memory)	XEG (IQH3, IQH4)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0
1	Frame Length	16#0F	16#15	16#0B	16#09	16#15	16#15	16#13	16#15	16#0B	16#09	16#15
2	Frag-mentation Counter	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
3	Tele-gram Length (High Byte)	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
4	Tele-gram Length (Low Byte)	16#0C	16#12	16#08	16#06	16#12	16#12	16#10	16#12	16#08	16#06	16#12
5	Com-mand	16#1D	16#D3	16#19	16#4B	16#71	16#68	16#E9	16#D7	16#D9	16#D8	16#D6
6	Status	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
7	Data	16#E0 <UID Byte 1>	<PC-Word (High Byte)>	<Data Byte 1>	<Data Byte 1>	<PC-Word (High Byte)>	<Info Flags>	16#E2 <TID Byte 1>	<PC-Word (High Byte)>	<Data Byte 1>	<Data Byte 1>	<Info Flags>
8	Data	<UID Byte 2>	<PC-Word (Low Byte)>	<Data Byte 2>	<Data Byte 2>	<PC-Word (Low Byte)>	<UID Byte 8>	<TID Byte 2>	<PC-Word (Low Byte)>	<Data Byte 2>	<Data Byte 2>	<UID Byte 8>
9	Data	<UID Byte 3>	<EPC / Ull Byte 1>	<Data Byte 3>	16#00	<EPC / Ull Byte 1>	<UID Byte 7>	<TID Byte 3>	<EPC / Ull Byte 1>	<Data Byte 3>	16#00	<UID Byte 7>
10	Data	<UID Byte 4>	<EPC / Ull Byte 2>	<Data Byte 4>	16#00	<EPC / Ull Byte 2>	<UID Byte 6>	<TID Byte 4>	<EPC / Ull Byte 2>	<Data Byte 4>	16#00	<UID Byte 6>
11	Data	<UID Byte 5>	<EPC / Ull Byte 3>	16#00	16#00	<EPC / Ull Byte 3>	<UID Byte 5>	<TID Byte 5>	<EPC / Ull Byte 3>	16#00	16#00	<UID Byte 5>
12	Data	<UID Byte 6>	<EPC / Ull Byte 4>	16#00	16#00	<EPC / Ull Byte 4>	<UID Byte 4>	<TID Byte 6>	<EPC / Ull Byte 4>	16#00	16#00	<UID Byte 4>
13	Data	<UID Byte 7>	<EPC / Ull Byte 5>	16#00	16#00	<EPC / Ull Byte 5>	<UID Byte 3>	<TID Byte 7>	<EPC / Ull Byte 5>	16#00	16#00	<UID Byte 3>
14	Data	<UID Byte 8>	<EPC / Ull Byte 6>	16#00	16#00	<EPC / Ull Byte 6>	<UID Byte 2>	<TID Byte 8>	<EPC / Ull Byte 6>	16#00	16#00	<UID Byte 2>
15	Data	16#00	<EPC / Ull Byte 7>	16#00	16#00	<EPC / Ull Byte 7>	<UID Byte 1>	<TID Byte 9>	<EPC / Ull Byte 7>	16#00	16#00	<UID Byte 1>
16	Data	16#00	<EPC / Ull Byte 8>	16#00	16#00	<EPC / Ull Byte 8>	<DSFID>	<TID Byte 10>	<EPC / Ull Byte 8>	16#00	16#00	<DSFID>
17	Data	16#00	<EPC / Ull Byte 9>	16#00	16#00	<EPC / Ull Byte 9>	<AFI>	<TID Byte 11>	<EPC / Ull Byte 9>	16#00	16#00	<AFI>
18	Data	16#00	<EPC / Ull Byte 10>	16#00	16#00	<EPC / Ull Byte 10>	<VICC Memory Size (Number of Blocks -1)>	<TID Byte 12>	<EPC / Ull Byte 10>	16#00	16#00	<VICC Memory Size (Number of Blocks -1)>
19	Data	16#00	<EPC / Ull Byte 11>	16#00	16#00	<EPC / Ull Byte 11>	<VICC Memory Size (Block Size -1)>	16#00	<EPC / Ull Byte 11>	16#00	16#00	<VICC Memory Size (Block Size -1)>
20	Data	16#00	<EPC / Ull Byte 12>	16#00	16#00	<EPC / Ull Byte 12>	<IC Reference>	16#00	<EPC / Ull Byte 12>	16#00	16#00	<IC Reference>
21	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Byte	Content	EF (IQH*)	EN (IUH*; 12 Byte UII / EPC)	ER (IQH*; 4 Byte User Memory)	#ER (IUH*; 2 Byte User Memory)	ES (IUH*; 12 Byte UII / EPC)	EG (IQH*)	XEF (IUH1)	XEN (IUH1)	XSR (IUH1; 4 Byte User Memory)	X#ER (IUH1; 2 Byte User Memory)	XEG (IQH3, IQH4)
...	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
255	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Response telegram 2 – Status 16#0B (IDENTControl → SPS)

Byte	Content	EF (IQH*)	EN (IUH*)	ER (IQH*)	#ER (IUH*)	ES (IUH*)	EG (IQH*)	XEF (IUH1)	XEN (IUH1)	XER (IUH1; 4 Byte User Memory)	X#ER (IUH1)	XEG (IQH*)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0
1	Frame Length	16#09	16#0C	16#09	16#0C	16#0C	16#09	16#0C	16#0C	16#0C	16#0C	16#09
2	Fragmentation Counter	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
3	Telegram Length (High Byte)	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
4	Telegram Length (Low Byte)	16#06	16#09	16#06	16#09	16#09	16#06	16#09	16#09	16#09	16#09	16#06
5	Command	16#1D	16#D3	16#19	16#4B	16#71	16#68	16#E9	16#D7	16#D9	16#D8	16#D6
6	Status	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B
7	Data	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>
8	Data	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>
9	Data	16#00	<Transmission Channel>	16#00	<Transmission Channel>	<Transmission Channel>	16#00	<Transmission Channel>	<Transmission Channel>	<Transmission Channel>	<Transmission Channel>	16#00
10	Data	16#00	<Transmission Power (High Byte)>	16#00	<Transmission Power (High Byte)>	<Transmission Power (High Byte)>	16#00	<Transmission Power (High Byte)>	<Transmission Power (High Byte)>	<Transmission Power (High Byte)>	<Transmission Power (High Byte)>	16#00
11	Data	16#00	<Transmission Power (Low Byte)>	16#00	<Transmission Power (Low Byte)>	<Transmission Power (Low Byte)>	16#00	<Transmission Power (Low Byte)>	<Transmission Power (Low Byte)>	<Transmission Power (Low Byte)>	<Transmission Power (Low Byte)>	16#00
...	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
255	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

6.5.4.3

Single Write Commands

Command Telegram (PLC → IDENTControl)

Byte	Content	SX (IPH; 5-byte fixed code, can be over-written)	#SU (IUH*; 2 Byte PC + 12 Byte UII / EPC)	SW (IQH3, IQH4; 4 Byte User Memory)	#SW (IUH*)	SP (IUH*)	SC (IQH*)	X#SU (IUH1; 2 Byte UII / EPC Mask, 2 Byte PC + 12 Byte UII / EPC)	XSW (IUH1; 2 Byte UII / EPC Mask, 4 Byte User Memory)	X#SW (IUH1; 2 Byte UII / EPC Mask, 2 Byte User Memory)	XSC (IQH3, IQH4; 2 Byte UID Mask)	SL (IQH*; 4 Byte User Memory)	XSL (IQH3, IQH4; 2 Byte UID Mask, 4 Byte User Memory)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0
1	Frame Length	16#0F	16#14	16#0E	16#0C	16#15	16#0B	16#18	16#12	16#10	16#0F	16#0E	16#12
2	Fram- menta- tion Counter	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
3	Tele- gram Length (High Byte)	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
4	Tele- gram Length (Low Byte)	16#0C	16#11	16#0B	16#09	16#12	16#08	16#15	16#0F	16#0D	16#0C	16#0B	16#0F
5	Com- mand	16#1F	16#CE	16#40	16#4A	16#0D	16#12	16#E6	16#E3	16#E2	16#E1	16#47	16#E4
6	Com- mand param- eters	16#31 <Fix- code Type (High Byte)>	16#30 <PC (High Byte)>	16#00 <Start Address (High Byte)>	16#00 <Start Address (High Byte)>	16#30	<config Address >	16#00 <Length UII / EPC Mask (High Byte)>	16#00 <Length UII / EPC Mask (High Byte)>	16#00 <Length UII / EPC Mask (High Byte)>	16#00 <Length UID Mask (High Byte)>	16#00 <Start Address (High Byte)>	16#00 <Length UID Mask (High Byte)>
7	Com- mand param- eters	16#31 <Fix- code Type (Low Byte)>	16#00 <PC (Low Byte)>	16#00 <Start Address (Low Byte)>	16#00 <Start Address (Low Byte)>	<PC- Word Address (High Byte)>	<Lock>	16#10 <Length UII / EPC Mask (Low Byte)>	16#10 <Length UII / EPC Mask (Low Byte)>	16#10 <Length UII / EPC Mask (Low Byte)>	16#10 <Length UID Mask (Low Byte)>	16#00 <Start Address (Low Byte)>	16#10 <Length UID Mask (Low Byte)>
8	Com- mand param- eters	16#00 <Fix- code Length (High Byte)>	<EPC / UII Byte 1>	16#00 <Num- ber of Bytes (High Byte)>	16#00 <Num- ber of Bytes (High Byte)>	<PC- Word (Low Byte)>	<DSFID / AFI>	<UII / EPC Mask Byte 1>	<UII / EPC Mask Byte 1>	<UII / EPC Mask Byte 1>	<UID Mask Byte 1>	16#00 <Num- ber of Bytes (High Byte)>	<UID Mask Byte 1>
9	Com- mand param- eters	16#00 <Fix- code Length (Low Byte)>	<EPC / UII Byte 2>	16#04 <Num- ber of Bytes (Low Byte)>	16#02 <Num- ber of Bytes (Low Byte)>	<EPC / UII Byte 1>	16#00 (always)	<UII / EPC Mask Byte 2>	<UII / EPC Mask Byte 2>	<UII / EPC Mask Byte 2>	<UID Mask Byte 2>	16#04 <Num- ber of Bytes (Low Byte)>	<UID Mask Byte 2>
10	Com- mand param- eters	<Fix- code Byte 1>	<EPC / UII Byte 3>	<Data Byte 1>	<Data Byte 1>	<EPC / UII Byte 2>	16#00 (always)	16#30 <PC (High Byte)>	16#00 <Start Address (High Byte)>	16#00 <Start Address (High Byte)>	16#00 <Config Address >	<Data Byte 1>	16#00 <Start Address (High Byte)>
11	Com- mand param- eters	<Fix- code Byte 2>	<EPC / UII Byte 4>	<Data Byte 2>	<Data Byte 2>	<EPC / UII Byte 3>	16#00	16#00 <PC (Low Byte)>	16#00 <Start Address (Low Byte)>	16#00 <Start Address (Low Byte)>	<Lock>	<Data Byte 2>	16#00 <Start Address (Low Byte)>
12	Com- mand param- eters	<Fix- code Byte 3>	<EPC / UII Byte 5>	<Data Byte 3>	16#00	<EPC / UII Byte 4>	16#00	<EPC / UII Byte 1>	16#00 <Num- ber of Bytes (High Byte)>	16#00 <Num- ber of Bytes (High Byte)>	<DSFID / AFI>	<Data Byte 3>	16#00 <Num- ber of Bytes (High Byte)>
13	Com- mand param- eters	<Fix- code Byte 4>	<EPC / UII Byte 6>	<Data Byte 4>	16#00	<EPC / UII Byte 5>	16#00	<EPC / UII Byte 2>	16#04 <Num- ber of Bytes (Low Byte)>	16#02 <Num- ber of Bytes (Low Byte)>	16#00 (always)	<Data Byte 4>	16#04 <Num- ber of Bytes (Low Byte)>
14	Com- mand param- eters	<Fix- code Byte 5>	<EPC / UII Byte 7>	16#00	16#00	<EPC / UII Byte 6>	16#00	<EPC / UII Byte 3>	<Data Byte 1>	<Data Byte 1>	16#00 (always)	16#00	<Data Byte 1>

2026-04

Byte	Content	SX (IPH; 5-byte fixed code, can be over- written)	#SU (IUH*; 2 Byte PC + 12 Byte UII / EPC)	SW (IQH3, IQH4; 4 Byte User Mem- ory)	#SW (IUH*)	SP (IUH*)	SC (IQH*)	X#SU (IUH1; 2 Byte UII / EPC Mask, 2 Byte PC + 12 Byte UII / EPC)	XSW (IUH1; 2 Byte UII / EPC Mask, 4 Byte User Mem- ory)	X#SW (IUH1; 2 Byte UII / EPC Mask, 2 Byte User Mem- ory)	XSC (IQH3, IQH4; 2 Byte UID Mask)	SL (IQH*; 4 Byte User Mem- ory)	XSL (IQH3, IQH4; 2 Byte UID Mask, 4 Byte User Mem- ory)
15	Com- mand param- eters	16#00	<EPC / UII Byte 8>	16#00	16#00	<EPC / UII Byte 7>	16#00	<EPC / UII Byte 4>	<Data Byte 2>	<Data Byte 2>	16#00	16#00	<Data Byte 2>
16	Com- mand param- eters	16#00	<EPC / UII Byte 9>	16#00	16#00	<EPC / UII Byte 8>	16#00	<EPC / UII Byte 5>	<Data Byte 3>	16#00	16#00		<Data Byte 3>
17	Com- mand param- eters	16#00	<EPC / UII Byte 10>	16#00	16#00	<EPC / UII Byte 9>	16#00	<EPC / UII Byte 6>	<Data Byte 4>	16#00	16#00	16#00	<Data Byte 4>
18	Com- mand param- eters	16#00	<EPC / UII Byte 11>	16#00	16#00	<EPC / UII Byte 10>	16#00	<EPC / UII Byte 7>	16#00	16#00	16#00	16#00	16#00
19	Com- mand param- eters	16#00	<EPC / UII Byte 12>	16#00	16#00	<EPC / UII Byte 11>	16#00	<EPC / UII Byte 8>	16#00	16#00	16#00	16#00	16#00
20	Com- mand param- eters	16#00	16#00	<EPC / UII Byte 12>	16#00	<EPC / UII Byte 9>	16#00	16#00	16#00	16#00	16#00	16#00	16#00
21	Com- mand param- eters	16#00	16#00	16#00	16#00	16#00	16#00	<EPC / UII Byte 10>	16#00	16#00	16#00	16#00	16#00
22	Com- mand param- eters	16#00	16#00	16#00	16#00	16#00	16#00	<EPC / UII Byte 11>	16#00	16#00	16#00	16#00	16#00
23	Com- mand param- eters	16#00	16#00	16#00	16#00	16#00	16#00	<EPC / UII Byte 12>	16#00	16#00	16#00	16#00	16#00
...	Com- mand param- eters	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
255	Com- mand param- eters	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Response telegram 1 – Status 16#00(IDENTControl → SPS)

Byte	Content	SX (IPH)	#SU (IUH*; 12 Byte UII / EPC)	SW (IQH3, IQH4)	#SW (IUH*; 12 Byte UII / EPC)	SP (IUH*; 12 Byte UII / EPC)	SC (IQH*)	X#SU (IUH1; 12 Byte UII / EPC)	XSW (IUH1; 12 Byte UII / EPC)	X#SW (IUH1; 12 Byte UII / EPC)	XSC (IQH3, IQH4)	SL (IQH*)	XSL (IQH3, IQH4)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0
1	Frame Length	16#07	16#15	16#0F	16#15	16#15	16#0F	16#15	16#15	16#15	16#0F	16#0F	16#0F
2	Frag- menta- tion Counter	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
3	Tele- gram Length (High Byte)	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
4	Tele- gram Length (Low Byte)	16#04	16#12	16#0C	16#12	16#12	16#0C	16#12	16#12	16#12	16#0C	16#0C	16#0C
5	Com- mand	16#1F	16#CE	16#40	16#4A	16#0D	16#12	16#E6	16#E3	16#E2	16#E1	16#47	16#E4

2026-04

Byte	Content	SX (IPH)	#SU (IUH*; 12 Byte UII / EPC)	SW (IQH3, IQH4)	#SW (IUH*; 12 Byte UII / EPC)	SP (IUH*; 12 Byte UII / EPC)	SC (IQH*)	X#SU (IUH1; 12 Byte UII / EPC)	XSW (IUH1; 12 Byte UII / EPC)	X#SW (IUH1; 12 Byte UII / EPC)	XSC (IQH3, IQH4)	SL (IQH*)	XSL (IQH3, IQH4)
6	Status	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
7	Data	16#00	<PC-Word (High Byte)>	16#E0 <UID Byte 1>	<PC-Word (High Byte)>	<PC-Word (High Byte)>	16#E0 <UID Byte 1>	<PC-Word (High Byte)>	<PC-Word (High Byte)>	<PC-Word (High Byte)>	16#E0 <UID Byte 1>	16#E0 <UID Byte 1>	16#E0 <UID Byte 1>
8	Data	16#00	<PC-Word (Low Byte)>	<UID Byte 2>	<PC-Word (Low Byte)>	<PC-Word (Low Byte)>	<UID Byte 2>	<PC-Word (Low Byte)>	<PC-Word (Low Byte)>	<PC-Word (Low Byte)>	<UID Byte 2>	<UID Byte 2>	<UID Byte 2>
9	Data	16#00	<EPC / UII Byte 1>	<UID Byte 3>	<EPC / UII Byte 1>	<EPC / UII Byte 1>	<UID Byte 3>	<EPC / UII Byte 1>	<EPC / UII Byte 1>	<EPC / UII Byte 1>	<UID Byte 3>	<UID Byte 3>	<UID Byte 3>
10	Data	16#00	<EPC / UII Byte 2>	<UID Byte 4>	<EPC / UII Byte 2>	<EPC / UII Byte 2>	<UID Byte 4>	<EPC / UII Byte 2>	<EPC / UII Byte 2>	<EPC / UII Byte 2>	<UID Byte 4>	<UID Byte 4>	<UID Byte 4>
11	Data	16#00	<EPC / UII Byte 3>	<UID Byte 5>	<EPC / UII Byte 3>	<EPC / UII Byte 3>	<UID Byte 5>	<EPC / UII Byte 3>	<EPC / UII Byte 3>	<EPC / UII Byte 3>	<UID Byte 5>	<UID Byte 5>	<UID Byte 5>
12	Data	16#00	<EPC / UII Byte 4>	<UID Byte 6>	<EPC / UII Byte 4>	<EPC / UII Byte 4>	<UID Byte 6>	<EPC / UII Byte 4>	<EPC / UII Byte 4>	<EPC / UII Byte 4>	<UID Byte 6>	<UID Byte 6>	<UID Byte 6>
13	Data	16#00	<EPC / UII Byte 5>	<UID Byte 7>	<EPC / UII Byte 5>	<EPC / UII Byte 5>	<UID Byte 7>	<EPC / UII Byte 5>	<EPC / UII Byte 5>	<EPC / UII Byte 5>	<UID Byte 7>	<UID Byte 7>	<UID Byte 7>
14	Data	16#00	<EPC / UII Byte 6>	<UID Byte 8>	<EPC / UII Byte 6>	<EPC / UII Byte 6>	<UID Byte 8>	<EPC / UII Byte 6>	<EPC / UII Byte 6>	<EPC / UII Byte 6>	<UID Byte 8>	<UID Byte 8>	<UID Byte 8>
15	Data	16#00	<EPC / UII Byte 7>	16#00	<EPC / UII Byte 7>	<EPC / UII Byte 7>	16#00	<EPC / UII Byte 7>	<EPC / UII Byte 7>	<EPC / UII Byte 7>	16#00	16#00	16#00
16	Data	16#00	<EPC / UII Byte 8>	16#00	<EPC / UII Byte 8>	<EPC / UII Byte 8>	16#00	<EPC / UII Byte 8>	<EPC / UII Byte 8>	<EPC / UII Byte 8>	16#00	16#00	16#00
17	Data	16#00	<EPC / UII Byte 9>	16#00	<EPC / UII Byte 9>	<EPC / UII Byte 9>	16#00	<EPC / UII Byte 9>	<EPC / UII Byte 9>	<EPC / UII Byte 9>	16#00	16#00	16#00
18	Data	16#00	<EPC / UII Byte 10>	16#00	<EPC / UII Byte 10>	<EPC / UII Byte 10>	16#00	<EPC / UII Byte 10>	<EPC / UII Byte 10>	<EPC / UII Byte 10>	16#00	16#00	16#00
19	Data	16#00	<EPC / UII Byte 11>	16#00	<EPC / UII Byte 11>	<EPC / UII Byte 11>	16#00	<EPC / UII Byte 11>	<EPC / UII Byte 11>	<EPC / UII Byte 11>	16#00	16#00	16#00
20	Data	16#00	<EPC / UII Byte 12>	16#00	<EPC / UII Byte 12>	<EPC / UII Byte 12>	16#00	<EPC / UII Byte 12>	<EPC / UII Byte 12>	<EPC / UII Byte 12>	16#00	16#00	16#00
21	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
...	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
255	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Response telegram 2 – Status 16#0B (IDENTControl → SPS)

Byte	Content	SX (IPH)	#SU (IUH*)	SW (IQH3, IQH4)	#SW (IUH*)	SP (IUH*)	SC (IQH*)	X#SU (IUH1)	XSW (IUH1)	X#SW (IUH1)	XSC (IQH3, IQH4)	SL (IQH*)	XSL (IQH3, IQH4)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0
1	Frame Length	16#09	16#0C	16#09	16#0C	16#0C	16#09	16#0C	16#0C	16#0C	16#09	16#09	16#09
2	Fragmentation Counter	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
3	Telegram Length (High Byte)	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
4	Telegram Length (Low Byte)	16#06	16#09	16#06	16#09	16#09	16#06	16#09	16#09	16#09	16#06	16#06	16#06

Byte	Content	SX (IPH)	#SU (IUH*)	SW (IQH3, IQH4)	#SW (IUH*)	SP (IUH*)	SC (IQH*)	X#SU (IUH1)	XSW (IUH1)	X#SW (IUH1)	XSC (IQH3, IQH4)	SL (IQH*)	XSL (IQH3, IQH4)
5	Command	16#1F	16#CE	16#40	16#4A	16#0D	16#12	16#E6	16#E3	16#E2	16#E1	16#47	16#E4
6	Status	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B
7	Data	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>
8	Data	16#FF <RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>
9	Data	16#00	<Transmission Channel>	16#00	<Transmission Channel>	<Transmission Channel>	16#00	<Transmission Channel>	<Transmission Channel>	<Transmission Channel>	16#00	16#00	16#00
10	Data	16#00	<Transmission Power (High Byte)>	16#00	<Transmission Power (High Byte)>	<Transmission Power (High Byte)>	16#00	<Transmission Power (High Byte)>	<Transmission Power (High Byte)>	<Transmission Power (High Byte)>	16#00	16#00	16#00
11	Data	16#00	<Transmission Power (Low Byte)>	16#00	<Transmission Power (Low Byte)>	<Transmission Power (Low Byte)>	16#00	<Transmission Power (Low Byte)>	<Transmission Power (Low Byte)>	<Transmission Power (Low Byte)>	16#00	16#00	16#00
...	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
255	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Response telegram 3 – Status 16#0F (IDENTControl → SPS)

Byte	Content	SX (IPH)	#SU (IUH*)	SW (IQH3, IQH4)	#SW (IUH*)	SP (IUH*)	SC (IQH*)	X#SU (IUH1)	XSW (IUH1)	X#SW (IUH1)	XSC (IQH3, IQH4)	SL (IQH*)	XSL (IQH3, IQH4)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0
1	Frame Length	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B
2	Fragmentation Counter	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
3	Telegram Length (High Byte)	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
4	Telegram Length (Low Byte)	16#08	16#08	16#08	16#08	16#08	16#08	16#08	16#08	16#08	16#08	16#08	16#08
5	Command	16#1F	16#CE	16#40	16#4A	16#0D	16#12	16#E6	16#E3	16#E2	16#E1	16#47	16#E4
6	Status	16#0F	16#0F	16#0F	16#0F	16#0F	16#0F	16#0F	16#0F	16#0F	16#0F	16#0F	16#0F
7	Data	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30
8	Data	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30
9	Data	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30	16#30
10	Data	16#31	16#31	16#31	16#31	16#31	16#31	16#31	16#31	16#31	16#31	16#31	16#31
11	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
...	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
255	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

6.5.4.4 Enhanced Write Commands

Command Telegram (PLC → IDENTControl)

Byte	Content	EX (IPH; 5-byte fixed code, can be overwritten)	EW (IQH3, IQH4; 4 Byte User Memory)	#EW (IUH*)	EP (IUH*)	EC (IQH*)	XEW (IUH1; 2 Byte UII / EPC Mask, 4 Byte User Memory)	X#EW (IUH1; 2 Byte UII / EPC Mask, 2 Byte User Memory)	XEC (IQH3, IQH4; 2 Byte UID Mask)	EL (IQH*; 4 Byte User Memory)	XEL (IQH3, IQH4; 2 Byte UID Mask, 4 Byte User Memory)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0
1	Frame Length	16#0F	16#0E	16#0C	16#15	16#0B	16#12	16#10	16#0F	16#0E	16#12
2	Fragmentation Counter	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
3	Telegram Length (High Byte)	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
4	Telegram Length (Low Byte)	16#0C	16#0B	16#09	16#12	16#08	16#0F	16#0D	16#0C	16#0B	16#0F
5	Command	16#24	16#1A	16#4C	16#75	16#66	16#DC	16#DB	16#DA	16#48	16#E5
6	Command parameters	16#31 <Fixcode Type (High Byte)>	16#00 <Start Address (High Byte)>	16#00 <Start Address (High Byte)>	16#30	<config Address>	16#00 <Length UII / EPC Mask (High Byte)>	16#00 <Length UII / EPC Mask (High Byte)>	16#00 <Length UID Mask (High Byte)>	16#00 <Start Address (High Byte)>	16#00 <Length UID Mask (High Byte)>
7	Command parameters	16#31 <Fixcode Type (Low Byte)>	16#00 <Start Address (Low Byte)>	16#00 <Start Address (Low Byte)>	<PC-Word (High Byte)>	<Lock>	16#10 <Length UII / EPC Mask (Low Byte)>	16#10 <Length UII / EPC Mask (Low Byte)>	16#10 <Length UID Mask (Low Byte)>	16#00 <Start Address (Low Byte)>	16#10 <Length UID Mask (Low Byte)>
8	Command parameters	16#00 <Fixcode Length (High Byte)>	16#00 <Number of Bytes (High Byte)>	16#00 <Number of Bytes (High Byte)>	<PC-Word (Low Byte)>	<DSFID / AFI>	<UII / EPC Mask Byte 1>	<UII / EPC Mask Byte 1>	<UID Mask Byte 1>	16#00 <Number of Bytes (High Byte)>	<UID Mask Byte 1>
9	Command parameters	16#00 <Fixcode Length (Low Byte)>	16#04 <Number of Bytes (Low Byte)>	16#02 <Number of Bytes (Low Byte)>	<EPC / UII Byte 1>	16#00 (always)	<UII / EPC Mask Byte 2>	<UII / EPC Mask Byte 2>	<UID Mask Byte 2>	16#04 <Number of Bytes (Low Byte)>	<UID Mask Byte 2>
10	Command parameters	<Fixcode Byte 1>	<Data Byte 1>	<Data Byte 1>	<EPC / UII Byte 2>	16#00 (always)	16#00 <Start Address (High Byte)>	16#00 <Start Address (High Byte)>	16#00 <Config Address>	<Data Byte 1>	16#00 <Start Address (High Byte)>
11	Command parameters	<Fixcode Byte 2>	<Data Byte 2>	<Data Byte 2>	<EPC / UII Byte 3>	16#00	16#00 <Start Address (Low Byte)>	16#00 <Start Address (Low Byte)>	<Lock>	<Data Byte 2>	16#00 <Start Address (Low Byte)>
12	Command parameters	<Fixcode Byte 3>	<Data Byte 3>	16#00	<EPC / UII Byte 4>	16#00	16#00 <Number of Bytes (High Byte)>	16#00 <Number of Bytes (High Byte)>	<DSFID / AFI>	<Data Byte 3>	16#00 <Number of Bytes (High Byte)>
13	Command parameters	<Fixcode Byte 4>	<Data Byte 4>	16#00	<EPC / UII Byte 5>	16#00	16#04 <Number of Bytes (Low Byte)>	16#02 <Number of Bytes (Low Byte)>	16#00 (always)	<Data Byte 4>	16#04 <Number of Bytes (Low Byte)>
14	Command parameters	<Fixcode Byte 5>	16#00	16#00	<EPC / UII Byte 6>	16#00	<Data Byte 1>	<Data Byte 1>	16#00 (always)	16#00	<Data Byte 1>
15	Command parameters	16#00	16#00	16#00	<EPC / UII Byte 7>	16#00	<Data Byte 2>	<Data Byte 2>	16#00	16#00	<Data Byte 2>
16	Command parameters	16#00	16#00	16#00	<EPC / UII Byte 8>	16#00	<Data Byte 3>	16#00	16#00	16#00	<Data Byte 3>

Byte	Content	EX (IPH; 5-byte fixed code, can be over-written)	EW (IQH3, IQH4; 4 Byte User Memory)	#EW (IUH*)	EP (IUH*)	EC (IQH*)	XEW (IUH1; 2 Byte UII / EPC Mask, 4 Byte User Memory)	X#EW (IUH1; 2 Byte UII / EPC Mask, 2 Byte User Memory)	XEC (IQH3, IQH4; 2 Byte UID Mask)	EL (IQH*; 4 Byte User Memory)	XEL (IQH3, IQH4; 2 Byte UID Mask, 4 Byte User Memory)
17	Command parameters	16#00	16#00	16#00	<EPC / UII Byte 9>	16#00	<Data Byte 4>	16#00	16#00	16#00	<Data Byte 4>
18	Command parameters	16#00	16#00	16#00	<EPC / UII Byte 10>	16#00	16#00	16#00	16#00	16#00	16#00
19	Command parameters	16#00	16#00	16#00	<EPC / UII Byte 11>	16#00	16#00	16#00	16#00	16#00	16#00
20	Command parameters	16#00	16#00	16#00	<EPC / UII Byte 12>	16#00	16#00	16#00	16#00	16#00	16#00
...	Command parameters	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
255	Command parameters	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Response Telegram 1 – Status 16#00 (IDENTControl → SPS)

Byte	Content	EX (IPH)	EW (IQH3, IQH4)	#EW (IUH*)	EP (IUH*)	EC (IQH*)	XEW (IUH1)	X#EW (IUH1)	XEC (IQH3, IQH4)	EL (IQH*)	XEL (IQH3, IQH4)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0
1	Frame Length	16#07	16#0F	16#15	16#15	16#0F	16#15	16#15	16#0F	16#0F	16#0F
2	Fragmentation Counter	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
3	Telegram Length (High Byte)	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
4	Telegram Length (Low Byte)	16#04	16#0C	16#12	16#12	16#0C	16#12	16#12	16#0C	16#0C	16#0C
5	Command	16#24	16#1A	16#4C	16#75	16#66	16#DC	16#DB	16#DA	16#48	16#E5
6	Status	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
7	Data	16#00	16#E0 <UID Byte 1>	<PC-Word (High Byte)>	<PC-Word (High Byte)>	16#E0 <UID Byte 1>	<PC-Word (High Byte)>	<PC-Word (High Byte)>	16#E0 <UID Byte 1>	16#E0 <UID Byte 1>	16#E0 <UID Byte 1>
8	Data	16#00	<UID Byte 2>	<PC-Word (Low Byte)>	<PC-Word (Low Byte)>	<UID Byte 2>	<PC-Word (Low Byte)>	<PC-Word (Low Byte)>	<UID Byte 2>	<UID Byte 2>	<UID Byte 2>
9	Data	16#00	<UID Byte 3>	<EPC / UII Byte 1>	<EPC / UII Byte 1>	<UID Byte 3>	<EPC / UII Byte 1>	<EPC / UII Byte 1>	<UID Byte 3>	<UID Byte 3>	<UID Byte 3>
10	Data	16#00	<UID Byte 4>	<EPC / UII Byte 2>	<EPC / UII Byte 2>	<UID Byte 4>	<EPC / UII Byte 2>	<EPC / UII Byte 2>	<UID Byte 4>	<UID Byte 4>	<UID Byte 4>
11	Data	16#00	<UID Byte 5>	<EPC / UII Byte 3>	<EPC / UII Byte 3>	<UID Byte 5>	<EPC / UII Byte 3>	<EPC / UII Byte 3>	<UID Byte 5>	<UID Byte 5>	<UID Byte 5>
12	Data	16#00	<UID Byte 6>	<EPC / UII Byte 4>	<EPC / UII Byte 4>	<UID Byte 6>	<EPC / UII Byte 4>	<EPC / UII Byte 4>	<UID Byte 6>	<UID Byte 6>	<UID Byte 6>
13	Data	16#00	<UID Byte 7>	<EPC / UII Byte 5>	<EPC / UII Byte 5>	<UID Byte 7>	<EPC / UII Byte 5>	<EPC / UII Byte 5>	<UID Byte 7>	<UID Byte 7>	<UID Byte 7>
14	Data	16#00	<UID Byte 8>	<EPC / UII Byte 6>	<EPC / UII Byte 6>	<UID Byte 8>	<EPC / UII Byte 6>	<EPC / UII Byte 6>	<UID Byte 8>	<UID Byte 8>	<UID Byte 8>
15	Data	16#00	16#00	<EPC / UII Byte 7>	<EPC / UII Byte 7>	16#00	<EPC / UII Byte 7>	<EPC / UII Byte 7>	16#00	16#00	16#00
16	Data	16#00	16#00	<EPC / UII Byte 8>	<EPC / UII Byte 8>	16#00	<EPC / UII Byte 8>	<EPC / UII Byte 8>	16#00	16#00	16#00

2026-04

Byte	Content	EX (IPH)	EW (IQH3, IQH4)	#EW (IUH*)	EP (IUH*)	EC (IQH*)	XEW (IUH1)	X#EW (IUH1)	XEC (IQH3, IQH4)	EL (IQH*)	XEL (IQH3, IQH4)
17	Data	16#00	16#00	<EPC / UII Byte 9>	<EPC / UII Byte 9>	16#00	<EPC / UII Byte 9>	<EPC / UII Byte 9>	16#00	16#00	16#00
18	Data	16#00	16#00	<EPC / UII Byte 10>	<EPC / UII Byte 10>	16#00	<EPC / UII Byte 10>	<EPC / UII Byte 10>	16#00	16#00	16#00
19	Data	16#00	16#00	<EPC / UII Byte 11>	<EPC / UII Byte 11>	16#00	<EPC / UII Byte 11>	<EPC / UII Byte 11>	16#00	16#00	16#00
20	Data	16#00	16#00	<EPC / UII Byte 12>	<EPC / UII Byte 12>	16#00	<EPC / UII Byte 12>	<EPC / UII Byte 12>	16#00	16#00	16#00
21	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
...	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
255	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Response telegram 2 – Status 16#0B (IDENTControl → SPS)

Byte	Content	EX (IPH)	EW (IQH3, IQH4)	#EW (IUH*)	EP (IUH*)	EC (IQH*)	XEW (IUH1)	X#EW (IUH1)	XEC (IQH3, IQH4)	EL (IQH*)	XEL (IQH3, IQH4)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0	16#x0
1	Frame Length	16#09	16#09	16#0C	16#0C	16#09	16#0C	16#0C	16#09	16#09	16#09
2	Framen- tation Counter	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
3	Telegram Length (High Byte)	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
4	Telegram Length (Low Byte)	16#06	16#06	16#09	16#09	16#06	16#09	16#09	16#06	16#06	16#06
5	Command	16#24	16#1A	16#4C	16#75	16#66	16#DC	16#DB	16#DA	16#48	16#E5
6	Status	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B	16#0B
7	Data	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>	16#01 <Info Type>
8	Data	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>	<RSSI>
9	Data	16#00	16#00	<Trans- mission Channel>	<Trans- mission Channel>	16#00	<Trans- mission Channel>	<Trans- mission Channel>	16#00	16#00	16#00
10	Data	16#00	16#00	<Trans- mission Power (High Byte)>	<Trans- mission Power (High Byte)>	16#00	<Trans- mission Power (High Byte)>	<Trans- mission Power (High Byte)>	16#00	16#00	16#00
11	Data	16#00	16#00	<Trans- mission Power (Low Byte)>	<Trans- mission Power (Low Byte)>	16#00	<Trans- mission Power (Low Byte)>	<Trans- mission Power (Low Byte)>	16#00	16#00	16#00
...	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
255	Data	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

6.5.4.5 Configuration Commands – Read Parameters

Command Telegram (PLC → IDENTControl)

Byte	Content	Example: parameter PT (IUH*)	Example: parameter NT (IUH*)	Example: parameter TA (IUH*)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0
1	Frame Length	16#0B	16#0B	16#0B
2	Fragmentation Counter	16#00	16#00	16#00
3	Telegram Length (High Byte)	16#00	16#00	16#00
4	Telegram Length (Low Byte)	16#08	16#08	16#08
5	Command	16#BE	16#BE	16#BE
6	System code	16#55 "U"	16#55 "U"	16#55 "U"
7	Parameter Name (High Byte)	16#50 "P"	16#4E "N"	16#54 "T"
8	Parameter Name (Low Byte)	16#54 "T"	16#54 "T"	16#41 "A"
9	Length Parameter (High Byte)	16#00	16#00	16#00
10	Length Parameter (Low Byte)	16#00	16#00	16#00
...	Command parameters	16#00	16#00	16#00
255	Command parameters	16#00	16#00	16#00

Response telegram – Status 16#00 (IDENTControl → SPS)

Byte	Content	Example: parameter PT (IUH*, 500mW)	Example: parameter NT (IUH*, 255 = Off)	Example: parameter TA (IUH*, 2 attempts)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0
1	Frame Length	16#09	16#08	16#08
2	Fragmentation Counter	16#00	16#00	16#00
3	Telegram Length (High Byte)	16#00	16#00	16#00
4	Telegram Length (Low Byte)	16#06	16#05	16#05
5	Command	16#BE	16#BE	16#BE
6	Status	16#00	16#00	16#00
7	Data	16#01	16#FF	16#02
8	Data	16#F4	16#00	16#00
...	Data	16#00	16#00	16#00
255	Data	16#00	16#00	16#00

6.5.4.6 Configuration Commands – Write Parameters

Command Telegram (PLC –? IDENTControl)

Byte	Content	Example: parameter PT (IUH*, 500mW)	Example: parameter NT (IUH*, 255 = Off)	Example: parameter TA (IUH*, 2 attempts)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0
1	Frame Length	16#0B	16#0B	16#0B
2	Fragmentation Counter	16#00	16#00	16#00
3	Telegram Length (High Byte)	16#00	16#00	16#00
4	Telegram Length (Low Byte)	16#08	16#08	16#08
5	Command	16#BF	16#BF	16#BF
6	System code	16#55 "U"	16#55 "U"	16#55 "U"
7	Parameter Name (High Byte)	16#50 "P"	16#4E "N"	16#54 "T"
8	Parameter Name (Low Byte)	16#54 "T"	16#54 "T"	16#41 "A"
9	Length Parameter (High Byte)	16#00	16#00	16#00
10	Length Parameter (Low Byte)	16#02	16#01	16#01
11	Command parameters	16#01	16#FF	16#02
12	Command parameters	16#F4	16#00	16#00
...	Command parameters	16#00	16#00	16#00
255	Command parameters	16#00	16#00	16#00

Response telegram – Status 16#00 (IDENTControl → SPS)

Byte	Content	Example: parameter PT (IUH*, 500mW)	Example: parameter NT (IUH*, 255 = Off)	Example: parameter TA (IUH*, 2 attempts)
0	Control Byte / Frame Length	16#x0	16#x0	16#x0
1	Frame Length	16#07	16#07	16#07
2	Fragmentation Counter	16#00	16#00	16#00
3	Telegram Length (High Byte)	16#00	16#00	16#00
4	Telegram Length (Low Byte)	16#04	16#04	16#04
5	Command	16#BF	16#BF	16#BF
6	Status	16#00	16#00	16#00
...	Data	16#00	16#00	16#00
255	Data	16#00	16#00	16#00

6.5.4.7

System Commands

Command Telegram (PLC → IDENTControl)

Byte	Content	Example QU	Example CT (Tag Type 50)	Example RS	Example RD
0	Control Byte / Frame Length	16#x0	16#x0	16#x0	16#x0
1	Frame Length	16#06	16#07	16#06	16#06
2	Fragmentation Counter	16#00	16#00	16#00	16#00
3	Telegram Length (High Byte)	16#00	16#00	16#00	16#00
4	Telegram Length (Low Byte)	16#03	16#04	16#03	16#03
5	Command	16#02	16#04	16#16	16#B4
6	Command parameters	16#00	16#32	16#00	16#00
...	Command parameters	16#00	16#00	16#00	16#00
255	Command parameters	16#00	16#00	16#00	16#00

Response telegram – Status 16#00 (IDENTControl → SPS)

Byte	Content	Example QU	Example CT (Tag Type 50)	Example RS	Example RD
0	Control Byte / Frame Length	16#x0	16#x0	16#x0	16#x0
1	Frame Length	16#07	16#07	16#07	16#07
2	Fragmentation Counter	16#00	16#00	16#00	16#00
3	Telegram Length (High Byte)	16#00	16#00	16#00	16#00
4	Telegram Length (Low Byte)	16#04	16#04	16#04	16#04
5	Command	16#02	16#04	16#16	16#B4
6	Status	16#00	16#00	16#00	16#00
...	Data	16#00	16#00	16#00	16#00
255	Data	16#00	16#00	16#00	16#00

6.6 Error and Status Messages

Status	Meaning
16#00	The command was executed successfully.
16#02	Power-on message from a connected RFID read/write device.
16#04	Parameter error If this status message is received immediately after sending the command, a parameter is outside the valid range or the telegram structure is incorrect. This status message is also generated if an attempt is made to access a memory area that does not physically exist. Example: The amount of data to be read when executing the Read 4-Byte Blocks command exceeds the available memory.
16#05	The tag has left the detection zone. When executing an enhanced read or write command, this status message indicates that a tag has left the detection zone.
16#06	A command was sent to a channel to which no read/write device is connected, or the connection was interrupted during execution of a command.
16#0A	Short-form data format: Multiple tags are present in the detection zone. Long-form data format (IUH* only): Multiple tags with identical UII/EPC are present in the detection zone.
16#0B	Telegram with additional information This additional information (e.g., RSSI value) is transmitted in a telegram with this status.
16#0E	Buffer overflow The size of the internal telegram buffer has been exceeded. The device generated telegrams faster than they could be transmitted to the controller. The telegram buffer is cleared by inverting the Delete bit. In addition, the function of the handshake procedure must be checked.
16#0F	End message for execution of single commands The end of execution for single read or single write commands is indicated by this status message. This telegram contains the number of data carriers read or written during execution of the command.

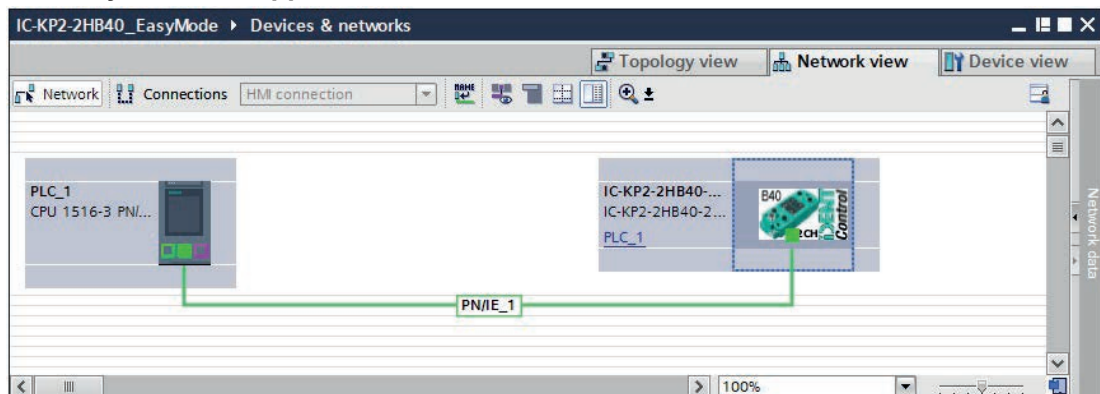
6.7 PROFINET

Integration of PROFINET devices into TIA follows a standardized procedure that is largely identical for different devices. To focus on the special features of the IDENTControl RFID evaluation unit, only the differing or device-specific configuration steps are explained below.

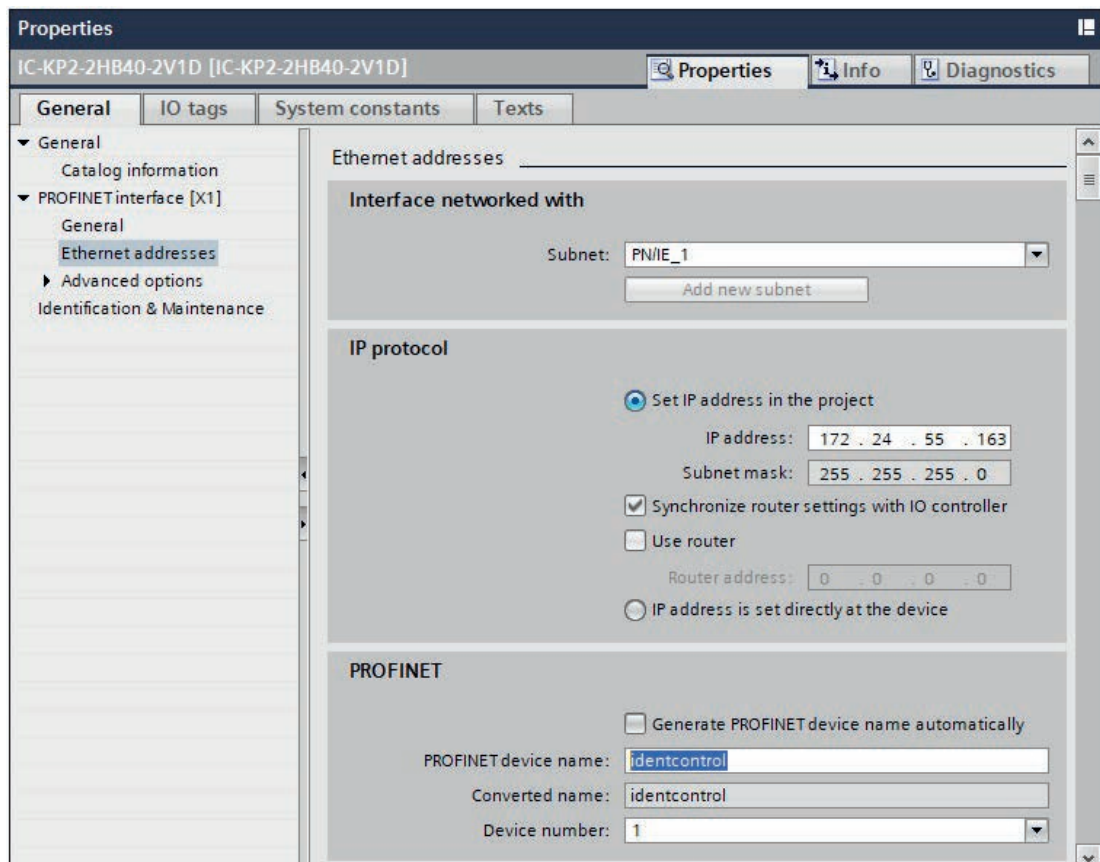


Integrating a PROFINET device into TIA

1. Add the device from the hardware catalog to the controller project and connect it to the PROFINET network:
To do this, go to the **"Additional Field Devices"** menu and then click on **"PROFINET IO"** > **"Ident Systems"** > **"Pepperl+Fuchs"** > **"IC-KP2-2HB40-2V1D"**.

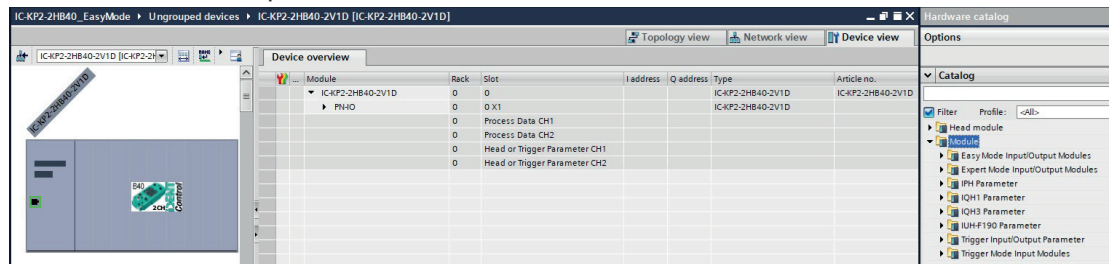


2. In the device parameters, set the network parameters and the PROFINET name.

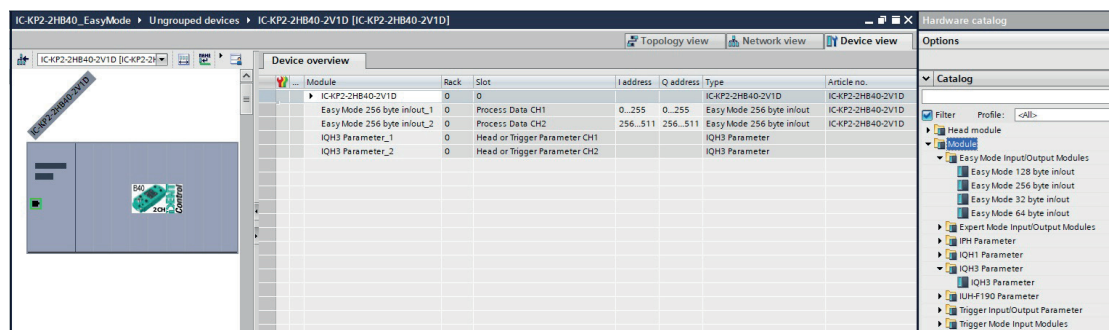


2026-04

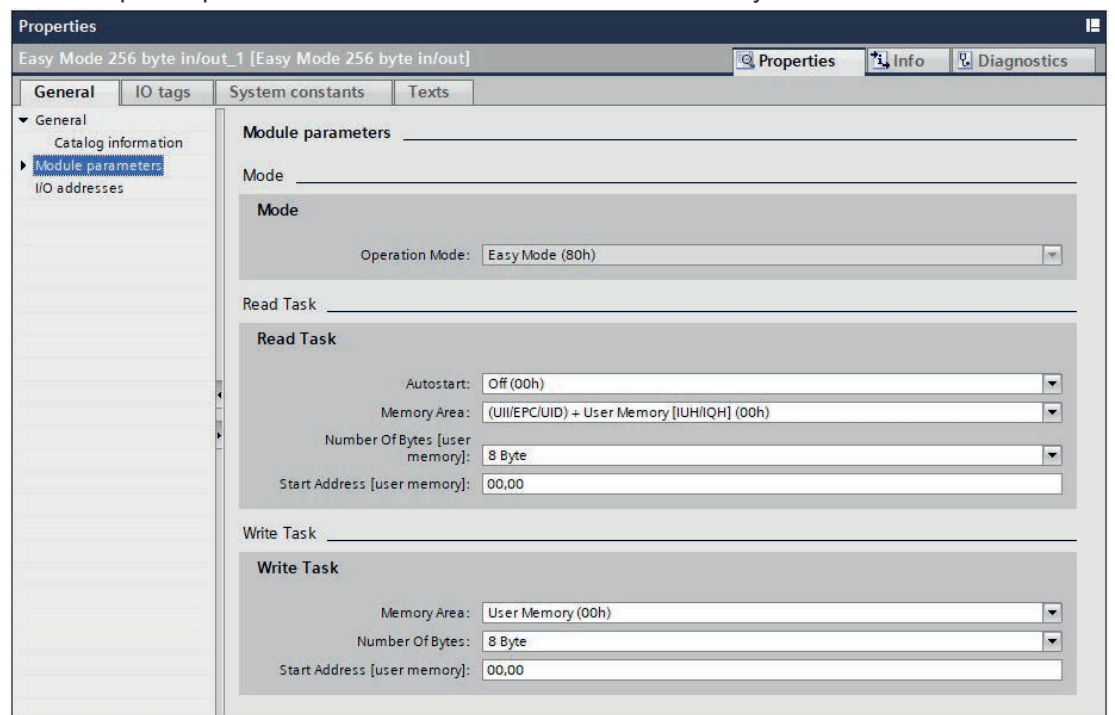
- Configure the device modules.
The device has 4 slots for modules. These are used to configure the process data for channels 1 and 2 as well as the parameters of the connected RFID read/write devices.



- Configure Easy Mode.
Several modules are available for Easy Mode, which differ in PROFINET telegram length. The modules are available with process data lengths of 32 bytes, 64 bytes, 128 bytes, and 256 bytes for the input and output data.
Slot 1 defines the operating mode for channel 1. Slot 2 defines the operating mode for channel 2.

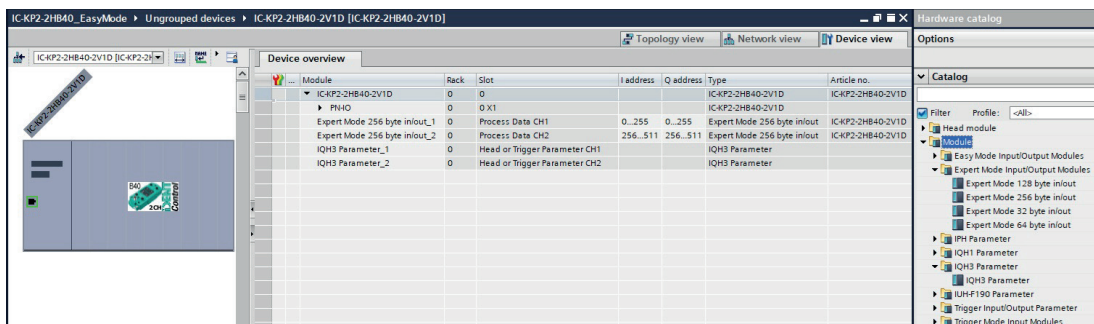


Module-specific parameters must be set for each module in Easy Mode.

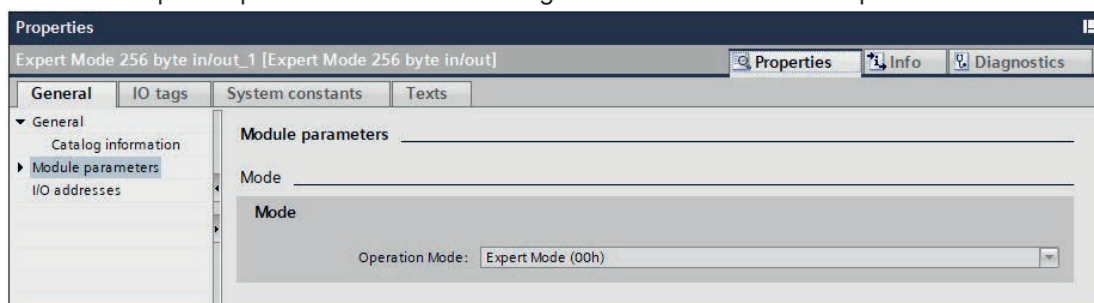


- Under the "Read Task" settings, the read operation to be executed is defined, and the autostart function is enabled or disabled.
- Under the "Write Task" setting, the parameters for the write access to be executed are defined.

5. Configure Expert Mode.
Several modules are available for Expert Mode, which differ in PROFINET telegram length. The modules are available with process data lengths of 32 bytes, 64 bytes, 128 bytes, and 256 bytes for the input and output data.
Slot 1 defines the operating mode for channel 1. Slot 2 defines the operating mode for channel 2.



No module-specific parameters need be configured for the modules in Expert Mode.



6. Configure the parameters of the connected RFID read/write devices.
The connected RFID read/write devices have device-specific parameters. These parameters are configured by the modules in slots 3 and 4. The module in slot 3 contains the settings for the RFID read/write device connected to channel 1. The RFID read/write device on channel 2 is parameterized by the module in slot 4.
Depending on the connected RFID read/write devices, the corresponding parameter modules must be used. The parameters configured by these modules depend on the device.

Properties

IQH3 Parameter_1 [IQH3 Parameter]

Properties Info Diagnostics

General IO tags System constants Texts

General

Catalog information

Module parameters

Module parameters

Transmission Powers (PT)

Power (PT): Maximum

Number Of Tags To Find (NT)

Number Of Tags To Find (NT): 255

Tries Allowed (TA)

Tries Allowed (TA): 2

Expected Number Of Tags (QW)

Q-Value (QW): 0 (exactly 1 Tag)

Tag Lost Smoothing (E5)

Enhanced Status 5 (E5): 5

Input Representation

Input Representation: Short Form

Tag Type (CT)

Tag Type (CT): 20 (IQC20, ISO 15693)

Over Temperature Handling (TO)

Over Temperature Handling (TO): Switch Off Carrier

7 Service and Maintenance

The device is designed and constructed to function stable over long periods of time. For this reason, regular cleaning or maintenance is unnecessary.

Regularly check the connections and connection cables.

The device is maintenance-free. If the device is defective

- disassemble the device
- send it to Pepperl+Fuchs for repair or
- replace the device.

Any unauthorised tampering or modification is dangerous and will void any warranty and manufacturer's liability.

8 Troubleshooting

Source of fault	Possible cause	Remedy
The PWR/ERR LED is off.	No power supply is present.	Ensure a 24 V DC power supply.
The PWR/ERR LED lights up yellow.	The device is still booting up.	Wait until the boot process is complete.
The PWR/ERR LED flashes green. (EtherNet/IP communication only)	Memory overflow.	Reduce the Data Hold Time.
The PWR/ERR LED lights up red. (PROFINET communication only)	No physical connection to the master, e.g., cable break.	Check the cable and repair it if necessary.
	The assigned device name is incorrect	Configure the same device name in the master and in the device.
The Link/Act LED is off.	No physical network connection, e.g., cable break.	Check the cable and repair it if necessary.
The CHx indication does not appear even though the RFID read/write device is connected to port 1.	The cable is faulty or not connected correctly.	Check the cable and repair it if necessary.
	The RFID read/write device is defective.	Check the RFID read/write device and repair it if necessary.
The LED in the RFID read/write device and the CHx indication on the display of the IDENTControl RFID evaluation unit are flashing.	The connected RFID read/write device does not support the configured tag type.	Configure a tag type that is supported by the RFID read/write device.
A read command (e.g., SR ...) returns Status 4 even though the syntax is correct.	An incorrect tag type (e.g., IPC02) is configured for the corresponding channel. Read commands work only with tags, not with read-only tags.	Use the CT... Command to set the correct tag type (e.g., IPC03) or "Autodetect."
The SG or EG (Get Configuration) command returns Status 4 even though the syntax is correct.	IPC03 is not configured for the corresponding channel. The configuration commands work only when tag type IPC03 is configured and not in Autodetect mode.	Use the CT... command to set tag type IPC03.
The website cannot be displayed.	The IP address is incorrect.	With DHCP server: Make sure that the IP address entered in the browser matches the new IP address assigned to the device. Without DHCP server: Check whether the IP address entered in the browser matches the fixed IP address configured in the device. Check the subnet mask in both cases.

Table 8.1

This table is updated and expanded as needed. The latest manual can be found online at: www.pepperl-fuchs.com

9 Decommissioning and Disposal

Decommissioning

For safe decommissioning, see chapter 2 cybersecurity.

Disposal

The device, built-in components, packaging, and any batteries contained within must be disposed in compliance with the applicable laws and guidelines of the respective country.

10 Appendix

10.1 ASCII table

hex	dec	ASCII	hex	dec	ASCII	hex	dec	ASCII	hex	dec	ASCII
00	0	NUL	20	32	Space	40	64	@	60	96	'
01	1	SOH	21	33	!	41	65	A	61	97	a
02	2	STX	22	34	"	42	66	B	62	98	b
03	3	ETX	23	35	#	43	67	C	63	99	c
04	4	EOT	24	36	\$	44	68	D	64	100	d
05	5	ENQ	25	37	%	45	69	E	65	101	e
06	6	ACK	26	38	&	46	70	F	66	102	f
07	7	BEL	27	39	'	47	71	G	67	103	g
08	8	BS	28	40	(	48	72	H	68	104	h
09	9	HT	29	41	)	49	73	I	69	105	i
0A	10	LF	2A	42	*	4A	74	J	6A	106	j
0B	11	VT	2B	43	+	4B	75	K	6B	107	k
0C	12	FF	2C	44	,	4C	76	L	6C	108	l
0D	13	CR	2D	45	-	4D	77	M	6D	109	m
0E	14	SO	2E	46	.	4E	78	N	6E	110	n
0F	15	SI	2F	47	/	4F	79	O	6F	111	o
10	16	DLE	30	48	0	50	80	P	70	112	p
11	17	DC1	31	49	1	51	81	Q	71	113	q
12	18	DC2	32	50	2	52	82	R	72	114	r
13	19	DC3	33	51	3	53	83	S	73	115	s
14	20	DC4	34	52	4	54	84	T	74	116	t
15	21	NAK	35	53	5	55	85	U	75	117	u
16	22	SYN	36	54	6	56	86	V	76	118	v
17	23	ETB	37	55	7	57	87	W	77	119	w
18	24	CAN	38	56	8	58	88	X	78	120	x
19	25	EM	39	57	9	59	89	Y	79	121	y
1A	26	SUB	3A	58	:	5A	90	Z	7A	122	z
1B	27	ESC	3B	59	;	5B	91	[	7B	123	{
1C	28	FS	3C	60	<	5C	92	\	7C	124	
1D	29	GS	3D	61	=	5D	93	]	7D	125	}
1E	30	RS	3E	62	>	5E	94	^	7E	126	~
1F	31	US	3F	63	?	5F	95	_	7F	127	DEL

Your automation, our passion.

- Industrial Sensors
- Industrial Communication and Interfaces
- Enterprise Mobility
- Products and Solutions for Hazardous Areas

www.pepperl-fuchs.com

Pepperl+Fuchs Quality

Download our latest policy here:

www.pepperl-fuchs.com/quality

