IQH3/IQH4 Series

RFID Read/Write Devices for IDENTControl, 13.56 MHz, ISO 15693

Manual

IDENTControl

Your automation, our passion.

PEPPERL+FUCHS

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 Elektroindustrie (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

2026-05

2026-05

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 dismounting lies with the plant operator.

Only appropriately trained and qualified personnel may carry out mounting, installation, commissioning, operation, maintenance, and dismounting 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 for the identification of warning messages and of informative messages.

Warning Messages

You will find warning messages, whenever dangers may arise from your actions. It is mandatory that you observe these warning messages for your personal safety and in order to avoid property damage.

Depending on the risk level, the warning messages are displayed 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 Symbols

Note

This symbol brings important information to your attention.

Tip

This symbol brings an useful tip to your attention.

Example

This symbol provides an example of the issue described.

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 $_{bin}$.

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.

Note

In the TIA Portal from Siemens, hexadecimal numbers are displayed as follows:

Example:

16#00 (equivalent to 0x00)

1.6 Terms and Abbreviations

Terms

Tag	Mobile data carrier containing user data and a unique number
Fixed code	Unique and unalterable number of a tag
IDENTControl	RFID evaluation unit from Pepperl+Fuchs; interface to the higher-level controller; controls communication of the connected read/write devices
IQC	Pepperl+Fuchs-specific designation of tags with 13.56 MHz
ISO/IEC 15693	Standard for data transmission for a 13.56 MHz RFID system
Tag	Read/write tag; tag
AFI	Identifier for the application family; represents the application type that is replaced by the tag; enables only the relevant ones to be addressed from a variety of tags
DSFI	Identifier for the data format; indicates how the data is structured in the tag memory

Abbreviations

AFI	**A**pplication **F**amily **I**dentifier
FCC	**F**ederal **C**ommunications **C**ommission
HF	**H**igh **F**requency
IC	**I**ndustry **C**anada
ISO	**I**nternational **S**tandardisation **O**rganisation
RFID	**R**adio **F**requency **Id**entification
RSSI	**R**eceived **S**ignal **S**trength **I**ndicator
PLC	**P**rogrammable **L**ogic **C**ontroller
UID	**U**nique **I**tem **Id**entifier
DSFID	**D**ata **S**torage **F**ormat **I**dentifier

2026-05

2 Cybersecurity

The device is secure developed according to IEC 62443-4-1 for the application area defined here.

To ensure cybersecure operation and to protect the device, implement the following measures.

Security Context

The RFID read/write device is intended for operation on an IDENTControl RFID evaluation unit from Pepperl+Fuchs.

The Communication is realized via a proprietary, unencrypted, serial protocol.

No physical protection against unauthorized access is required.

Maintenance and Management

Check the website regularly for security advisories publications: https://www.pepperl-fuchs.com/cybersecurity.

Report incidents to the manufacturer.

Decommissioning

Reset the device to factory setting.

The device stores certain operating data permanently.

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 FCC Compliance Statement

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

1. This device may not cause harmful interference, and
2. This device must accept any interference received, including interference that may cause undesired operation.

Attention:

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

Note

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at their own expense.

To comply with FCC Part 15 rules in the United States, the system must be professionally installed to ensure compliance with the Part 15 certification. It is the responsibility of the operator and professional installer to ensure that only certified systems are deployed in the United States. The use of the system in any other combination (such as co-located antennas transmitting the same information) is expressly forbidden.

To comply with FCC RF exposure compliance requirements, the antennas used for this transmitter must be installed to provide a minimum separation distance from all persons as specified in the table below and must not be co-located or operated in conjunction with any other antenna or transmitter.

Type/Design	Minimum separation distance
IQH3-*-V1	20 cm
IQH4-*-V1	20 cm

2026-05

3.4 IC Compliance Statement

This device contains license-exempt transmitter(s)/receiver(s) that comply with Innovation, Science and Economic Development Canada's license-exempt RSS(s). Operation is subject to the following two conditions:

1. This device may not cause interference.
2. This device must accept any interference, including interference that may cause undesired operation of the device.

L'émetteur/récepteur exempt de licence contenu dans le présent appareil est conforme aux CNR d'Innovation, Sciences et Développement économique Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes:

1. L'appareil ne doit pas produire de brouillage.
2. L'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

To comply with IC RF exposure compliance requirements, the antennas used for this transmitter must be installed to provide a minimum separation distance from all persons as specified in the table below and must not be co-located or operated in conjunction with any other antenna or transmitter.

Type/Design	Minimum separation distance
IQH3-*-V1	20 cm
IQH4-*-V1	20 cm

3.5 UL Information

IQH3 Series

This device is for indoor use only.

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

The ambient temperature range is -20 °C ... +70 °C for operation with non-transmission periods, or -25 °C … +55 °C for continuous transmission mode. 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.

Nominal power supply voltage is 24 V $_{DC}$, voltage range is 20 V … 30 V $_{DC}$. The supply must be an LEC (Limited Energy Circuit), LPS (Limited Power Source), or Class 2. Overvoltage Category II applies. This is ensured when a Pepperl+Fuchs IDENTControl control interface is used.

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

The enclosure type is 1.
The products are intended for use in or with industrial machinery applications as defined in the Electrical Standard for Industrial Machinery, NFPA 79.

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!

Connection cable specifications

The maximum permissible temperature of the connection cable must be at least +70 °C. The minimum diameter of the connection cable must be 22 AWG or 0.34 mm^2.

IQH4 Series

The ambient temperature range is -25 °C ... +70 °C for operation with non-transmission periods.

Nominal power supply voltage is 24 V $_{DC}$, voltage range is 18 V … 30 V $_{DC}$. The supply must be CLASS 2.

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

Caution!

Connection cable specifications

The device must be connected exclusively using any R/C cord (type CYJV/7 or CYJV2/8) with appropriate ratings.

2026-05

PEPPERL+FUCHS

4 Product Description

4.1 Product Description

Functions

The IQH3/IQH4 series read/write devices are designed for reading and writing passive tags in the frequency range of 13.56 MHz. Tags complying with the ISO 15693 standard are supported. The read/write devices are connected to an IDENTControl evaluation unit using an integrated M12 plug.

Use and Application

The IQH3 series read/write device has an extended sensing range of up to 30 cm and is therefore especially suitable for applications in which tags must be reliably identified at greater distance or at higher passing speed.

IQH4 series read/write devices are available in different housing designs and have a sensing range of up to 25 cm, depending on the design. This makes them suitable for applications with limited installation space, for example.

Figure 4.1 IQH3-FP-V1

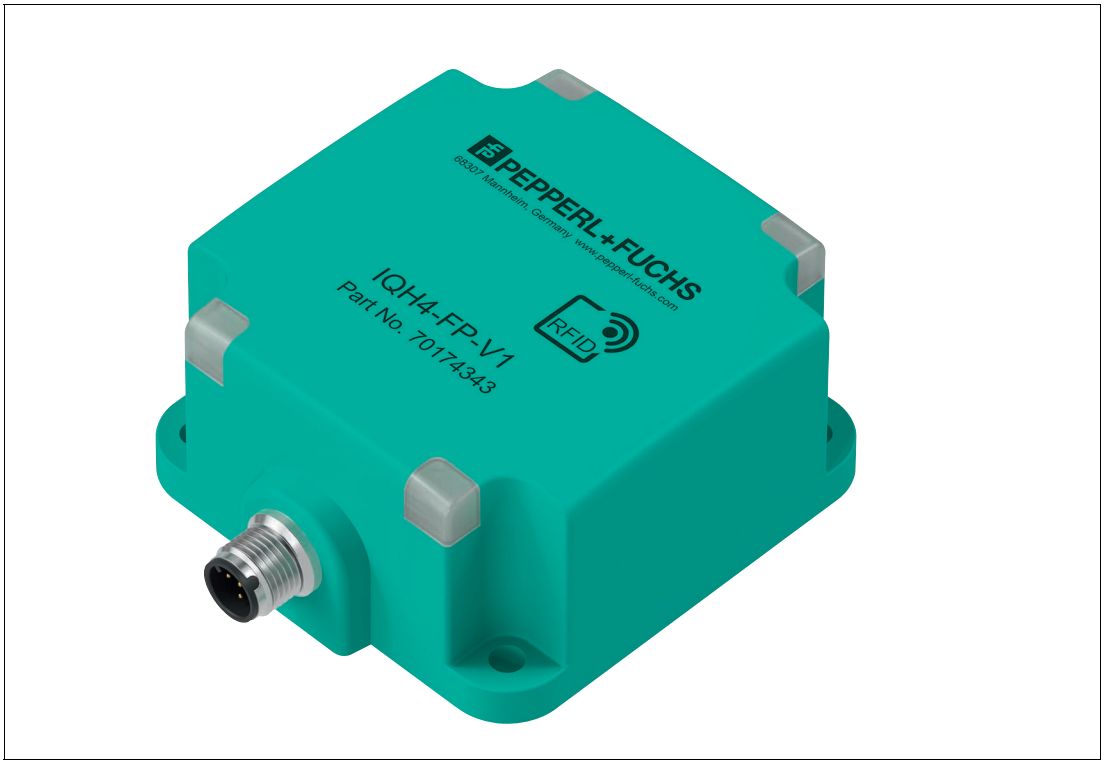


Figure 4.2 IQH4-FP-V1

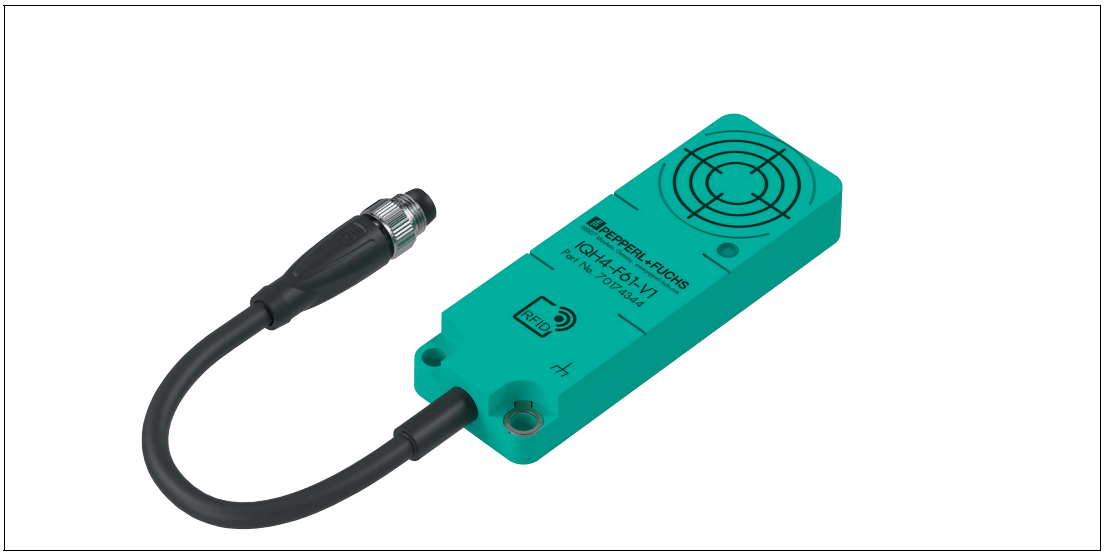


Figure 4.3 IQH4-F61-V1

2026-05

Figure 4.4 IQH4-18GM-V1

Figure 4.5 IQH4-F198-V1

Figure 4.6 IQH4-F198-M-V1

4.2 Dimensions

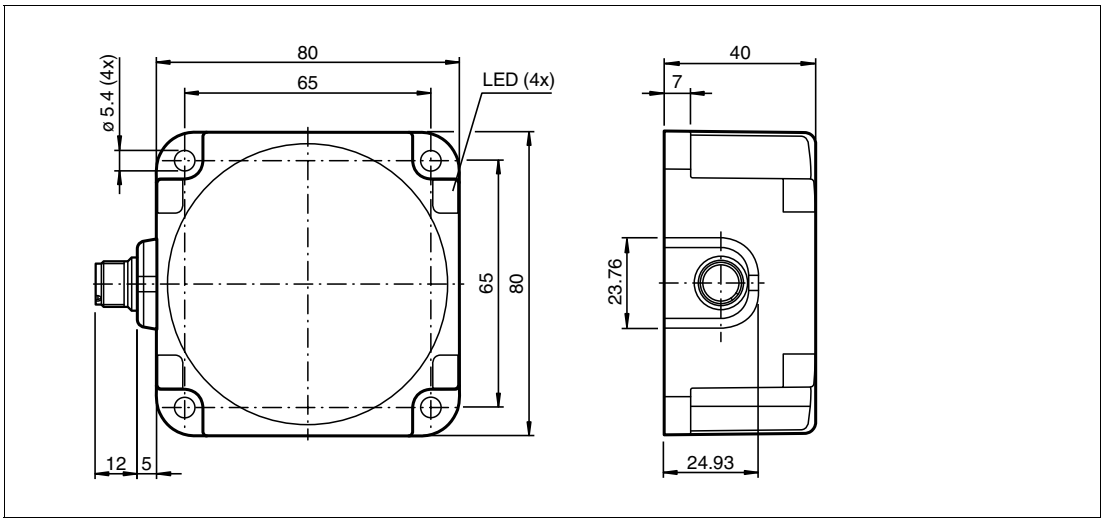


Figure 4.7 IQH3-FP-V1 / IQH4-FP-V1

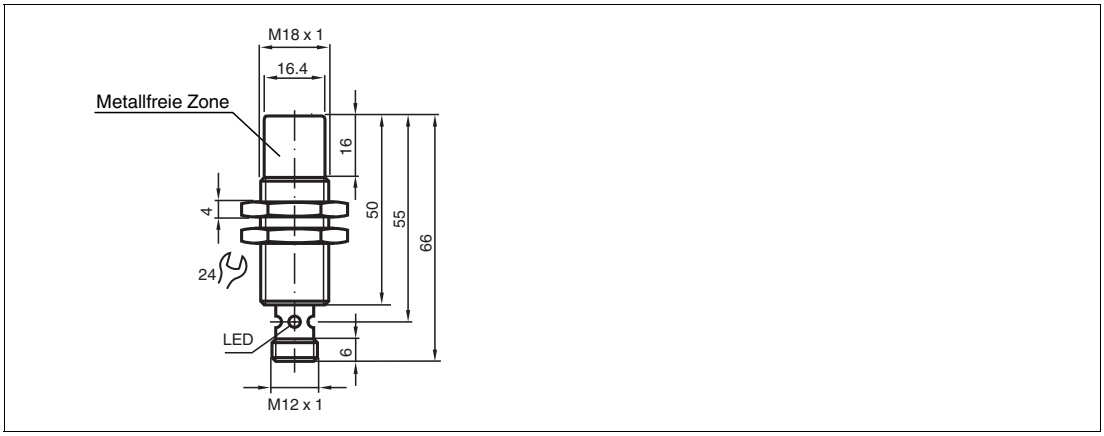


Figure 4.8 IQH4-18GM-V1

2026-05

PEPPERL+FUCHS

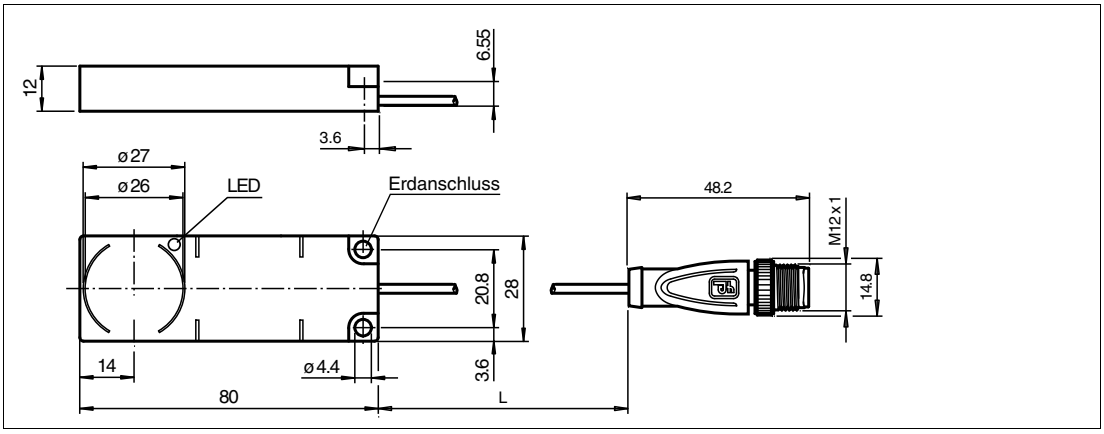


Figure 4.9 IQH4-F61-V1

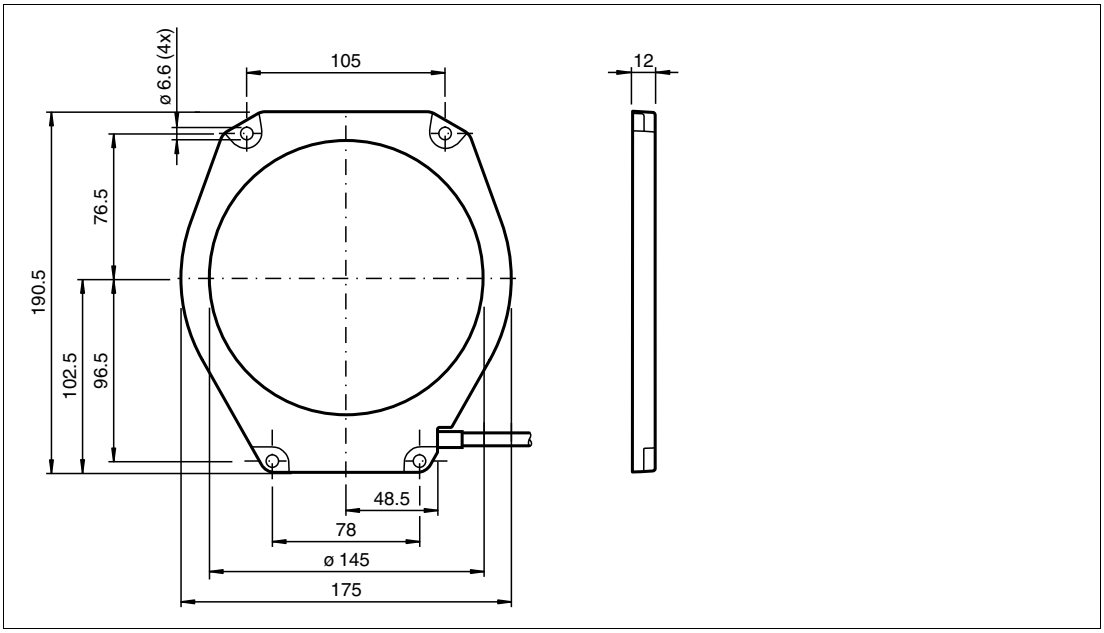


Figure 4.10 IQH4-F198-V1

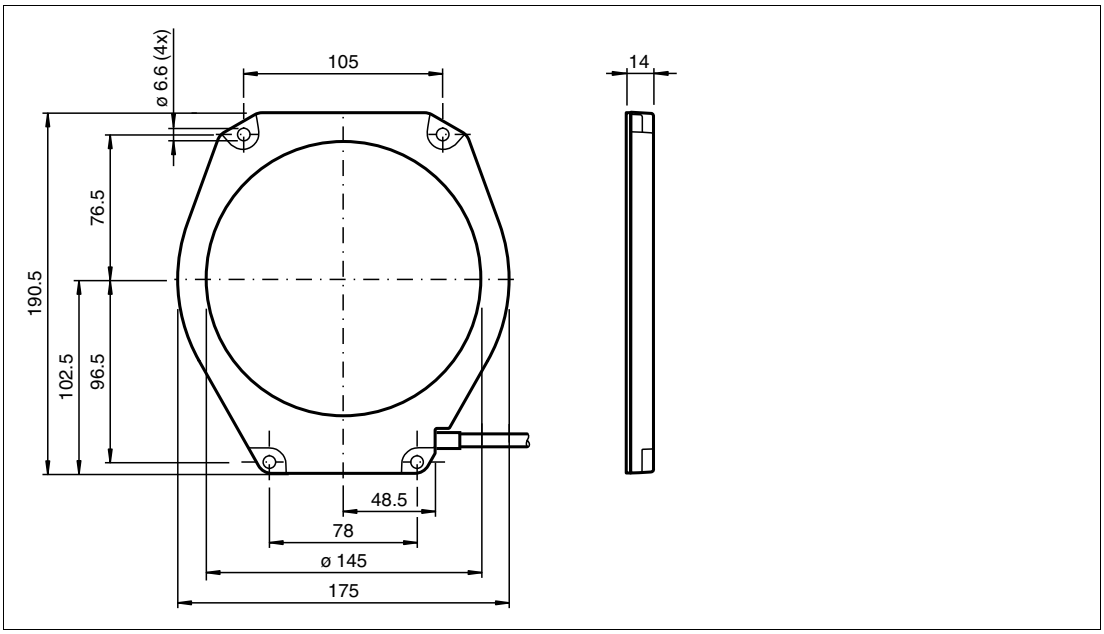


Figure 4.11 IQH4-F198-M-V1

4.3 Indicators

IQH3 Series LEDs

The IQH3-FP-V1 RFID read/write device has four LEDs for indicating the operating state. The various indicators denote:

LED	Description
Green	On: Device is ready for operation
Blue	On: Write/read attempt is being performed
Yellow	On: Write/read attempt was successful
Red	On: Interference from external signals in the usage spectrum, detection range may be limited Slow flashing (1 Hz): Metal interference Fast flashing (10 Hz): Excess temperature

IQH4 Series LEDs

The IQH4-18GM-V1, IQH4-F61-V1, IQH4-F198-V1, and IQH4-F198-M-V1 RFID read/write devices have a dual LED for indicating the operating state. The IQH4-FP-V1 read/write device has four LEDs, with two identical LEDs arranged diagonally on the device. The various indicators denote:

LED	Description
Green	On: Device is ready for operation Fast flashing (10 Hz): Write/read attempt is being performed Slow flashing (1 Hz): Status message
Orange	On: Write/read attempt was successful

4.4 Electrical Connection

IQH3 Series

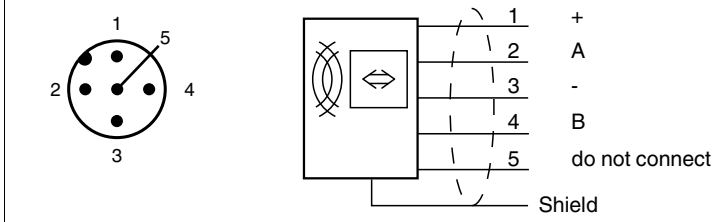


Figure 4.12

IQH4 Series

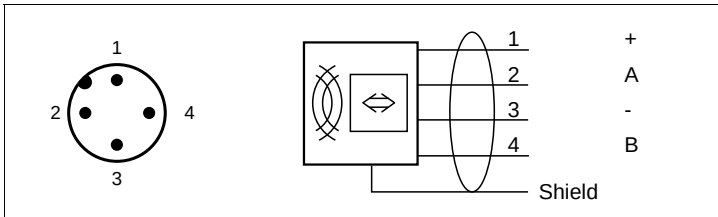


Figure 4.13

The read/write devices are connected to an IDENTControl evaluation unit via an M12 x 1 connector.

2026-05

4.5 Accessories

4.5.1 IDENTControl

The read/write devices are connected to Pepperl+Fuchs IDENT*Control* evaluation units. A selection of available IDENTControl evaluation units can be found online at www.pepperl-fuchs.com on the product page for the respective read/write device.

4.5.2 RFID Tags

The read/write devices can access any RFID tag that is compliant with the ISO15693 standard. A selection of available RFID tags from Pepperl+Fuchs can be found online at www.pepperl-fuchs.com on the product page for the respective read/write device.

4.5.3 Connection Cable

Compatible shielded connection cables are available for connecting the read/write devices. A selection of available connection cables from Pepperl+Fuchs can be found online at www.pepperl-fuchs.com on the product page for the respective read/write device.

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 Unpacking

Check the product for damage while unpacking. In the event of damage to the product, inform the post office or parcel service and notify the supplier.

Check the package contents against your purchase order and the shipping documents for:

- Delivery quantity
- Device type and version in accordance with the type label
- Any accessories ordered

Retain the original packaging in case you have to store or ship the device again at a later date.

Should you have any questions, please contact Pepperl+Fuchs.

5.3 Mounting

Note

The read/write devices are intended for wall mounting or mounting on brackets in indoor spaces. Mount the devices on a flat surface.

Attach the devices using only the existing mounting holes in the housing.

Caution!

Hot surfaces

Risk of burns when handling the read/write devices! Allow the devices to cool for at least half an hour after they have been switched off before touching them.

To attach IQH3-FP-V1, IQH4-FP-V1, IQH4-F198-V1, and IQH4-F198-M-V1 read/write devices, use four screws (diameter 4 mm).

Use two screws (diameter 4 mm) to attach the IQH4-F61-V1 read/write device.

To attach the read/write devices, use suitable flat washers and mounting materials suitable for the type of mounting surface.

The tightening torque of the screws depends on the type of mounting. We recommend using a tightening torque of 1.8 Nm.

Use a maximum tightening torque of 2.4 Nm to prevent damage to the plastic housing.

Use suitable mounting materials, such as nuts or mounting aids, to secure the IQH4-18GM-V1 read/write device.

Note

Do not route the connection cable of the read/write device and the IDENTControl evaluation unit into the raceway area of the tag. Maintain a distance of at least 10 cm.

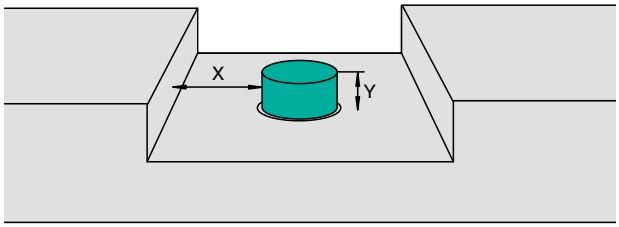
If the distance is too small, then in rare cases the tag may be incorrectly detected by inductive couplings on the connection cable.

2026-05

5.4 Minimum Distances

For flush-mounting in metal surfaces, a lateral minimum distance X to metal must be observed.

For flush-mounting in metal, a protrusion Y above the mounting surface must be observed.



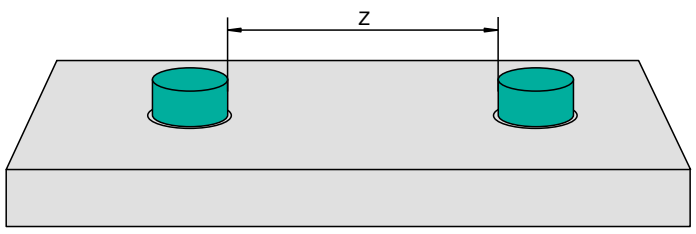
Read/write device	X	Y
IQH4-18GM-V1	XX	XX
IQH4-F61-V1	XX	XX
IQH4-FP-V1	XX	XX
IQH4-F198-V1	XX	XX
IQH4-F198-M-V1	XX	XX
IQH3-FP-V1	XX	XX

During simultaneous operation of several read/write devices, only one device may ever communicate with a tag at any given time. When arranging the read/write devices, make sure that the sensing ranges do not overlap. You can increase or decrease the sensing range by changing the transmitter radiated power accordingly. Determine the sensing range of each device at the mounting location.

If you want to transmit with just one read/write device, use the multiplex mode of the IDENT-Control evaluation unit. Multiplex mode allows chronologically exclusive access to tags, and prevents mutual interference from read/write devices. For a precise description, see the manual for your evaluation unit.

When mounting several RFID read/write devices next to each other, a minimum distance Z must be observed. This distance prevents the read/write devices from interfering with each other.

When mounting multiple read/write devices, keep to the following minimum distances:



Read/write device	Z
IQH4-18GM-V1	XX
IQH4-F61-V1	XX
IQH4-FP-V1	XX

Read/write device	Z
IQH4-F198-V1	XX
IQH4-F198-M-V1	XX
IQH3-FP-V1	Multiplex on: ≥ 100 mm, multiplex off: ≥ 750 mm

5.5 Main and Secondary Fields

The RFID read/write devices have a main field and secondary fields. However, the secondary fields should only be used for configuration in exceptional cases, as the writing/reading distances are very limited. Precise details of the geometries of the secondary fields cannot be provided, as the values depend heavily on the operating distance and the application.

In dynamic operation, it must be taken into account that the presence of the tag may be temporarily lost when the transition from the main field to the secondary field is made across gaps between the fields.

A schematic depiction of the typical main and secondary fields of the three device versions is shown below.

Legend	Designation
1	Main field
2	Secondary field
3	Gap between fields

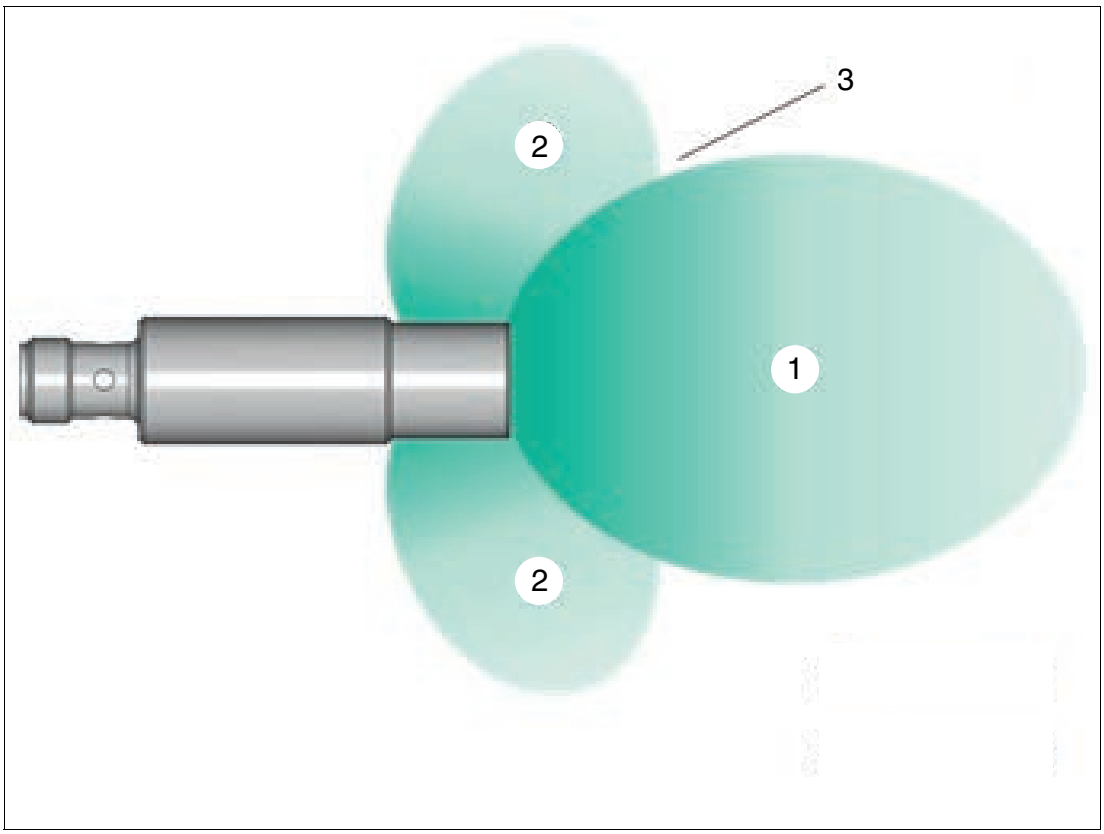


Figure 5.1 Schematic depiction of the main and secondary fields with IQH4-18GM-V1

2026-05

PEPPERL+FUCHS

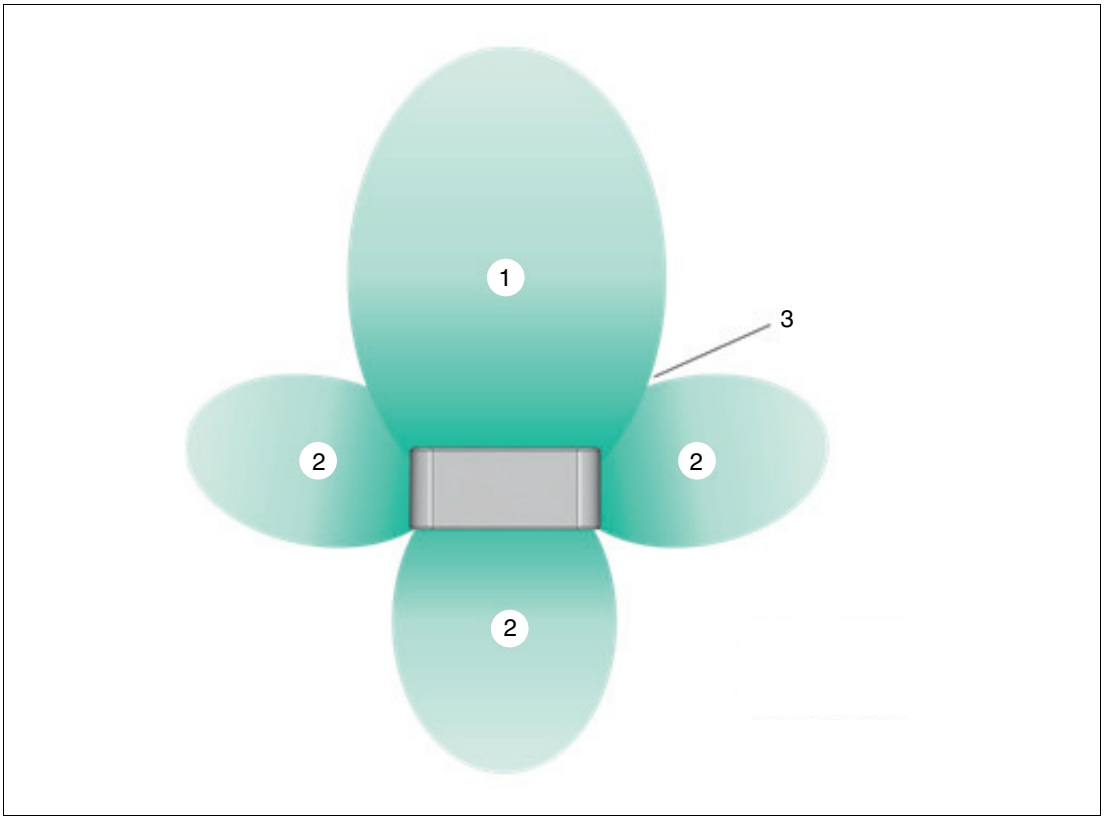


Figure 5.2 Schematic depiction of the main and secondary fields with IQH4-F61-V1

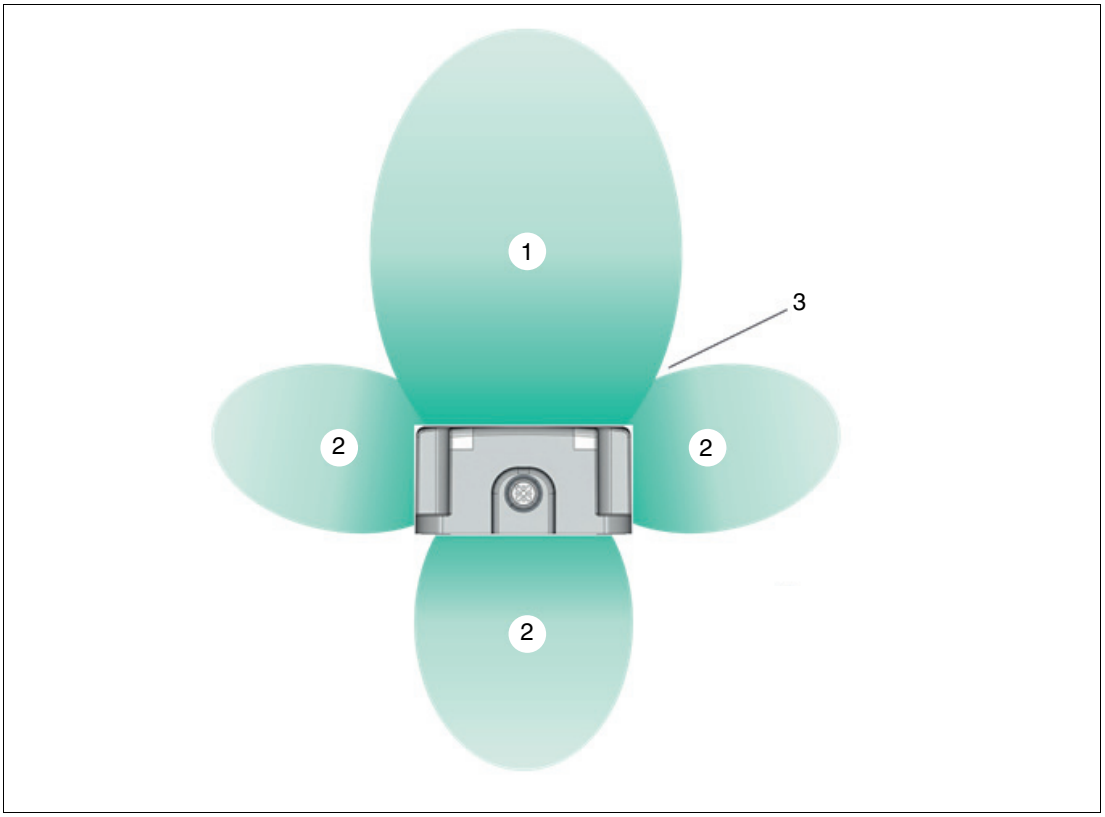


Figure 5.3 Schematic depiction of the main and secondary fields with IQH4-FP-V1

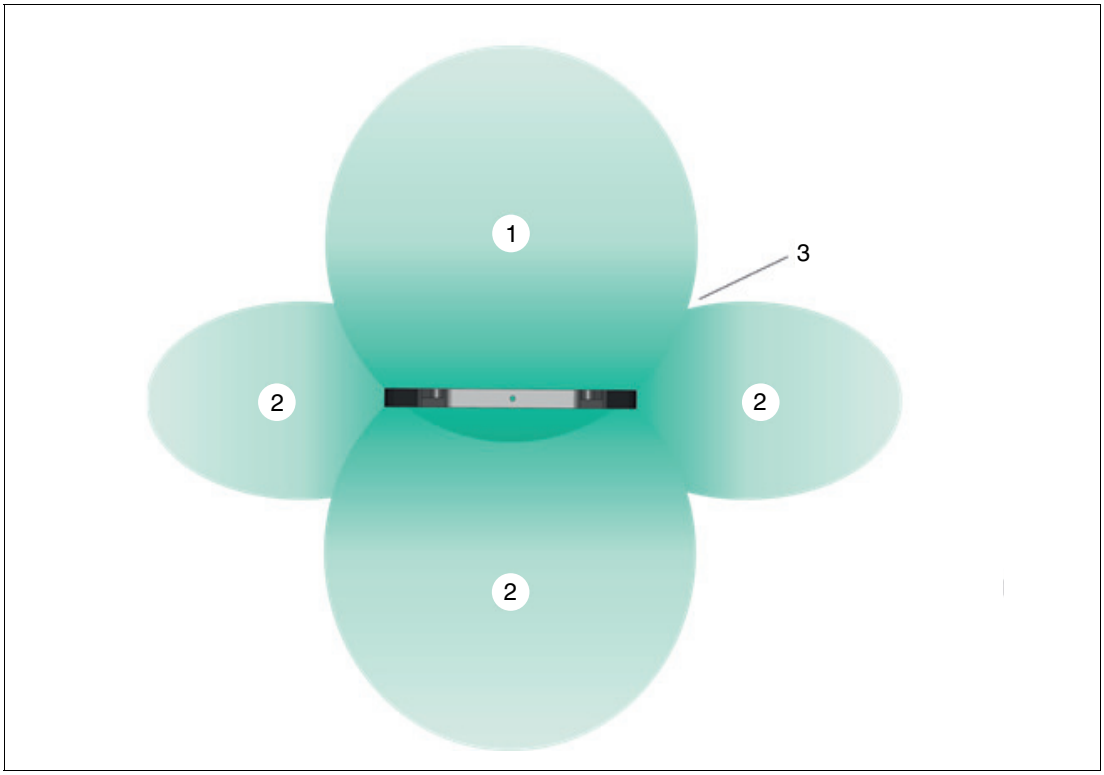


Figure 5.4 Schematic depiction of the main and secondary fields with IQH4-F198-V1

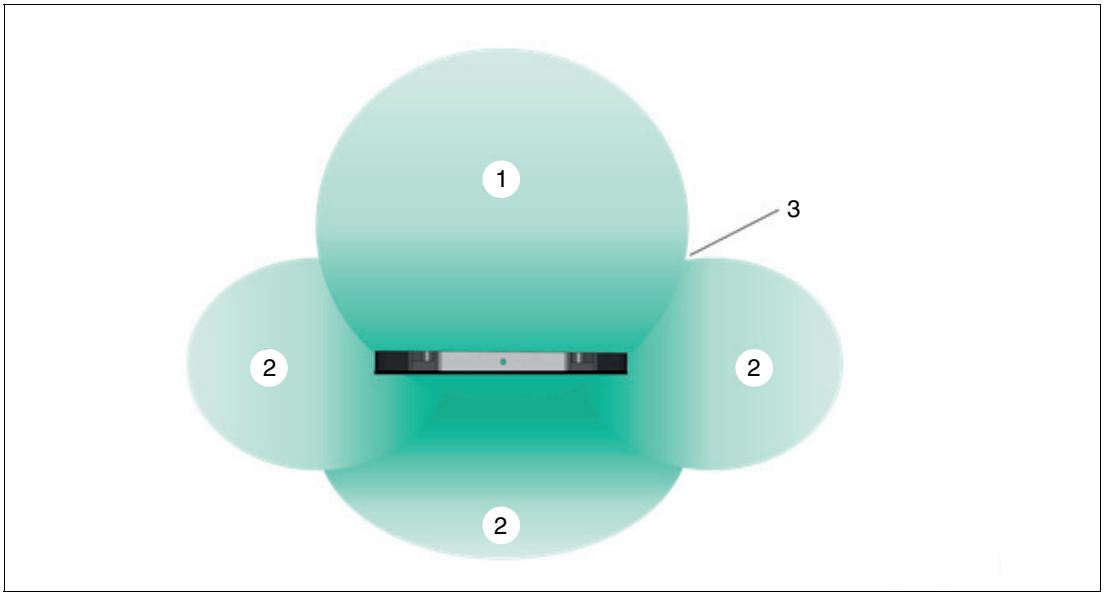


Figure 5.5 Schematic depiction of the main and secondary fields with IQH4-F198-M-V1

If no shielding measures have been taken, the RFID read/write device can also detect tags via one of the secondary fields.

Legend	Designation
1	Main field
2	Secondary field
3	Conveying direction
4	Tag
5	Metal shielding

2026-05

PEPPERL+FUCHS

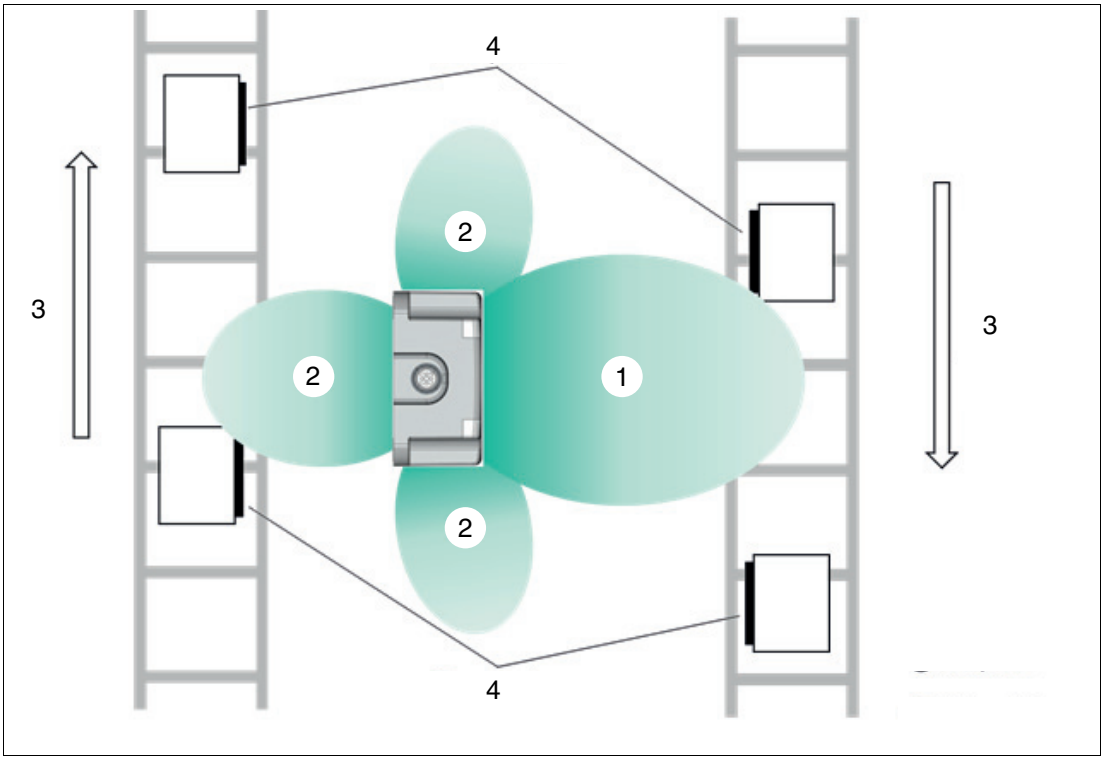


Figure 5.6 Schematic depiction of secondary fields without shielding

Metal shielding is required to prevent unwanted detection of tags via the secondary field.

Typical main and secondary fields are shown below, but with metal shielding. The metal shielding prevents tags from being detected via the secondary field of the RFID read/write device.

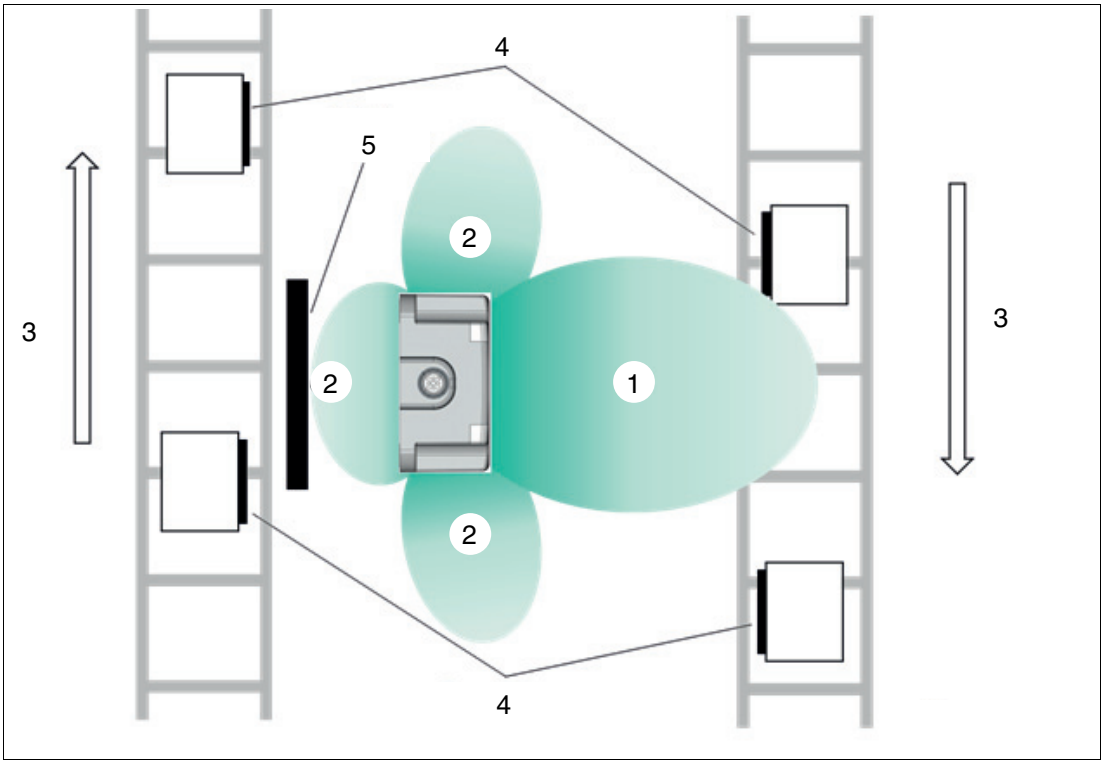


Figure 5.7 Schematic depiction of secondary fields with shielding

5.6 Ambient Temperature Range

The ambient temperature at which the devices can be operated depends on various factors, such as the operating mode or the selected transmitter radiated power.

By adjusting the operating mode or the transmitter radiated power (PT parameter), the devices can operate permanently at higher ambient temperatures, as the inherent heating of the devices is reduced.

Operation with Pauses in Transmission

PT	1	2	3	4
IQH3-FP-V1	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)
IQH4-18GM-V1	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)
IQH4-F61-V1	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)
IQH4-FP-V1	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)
IQH4-F198-[M]-V1	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)

Continuous Transmission Mode

PT	1	2	3	4
IQH3-FP-V1	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 65 °C (-13 °F- 149 °F)	-25 °C- 55 °C (-13 °F- 131 °F)
IQH4-18GM-V1	-25 °C- 60 °C (-13 °F- 140 °F)	-25 °C- 60 °C (-13 °F- 140 °F)	-25 °C- 50 °C (-13 °F- 122 °F)	-25 °C- 50 °C (-13 °F- 122 °F)
IQH4-F61-V1	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 65 °C (-13 °F- 149 °F)	-25 °C- 65 °C (-13 °F- 149 °F)
IQH4-FP-V1	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)
IQH4-F198-[M]-V1	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 65 °C (-13 °F- 149 °F)	-25 °C- 65 °C (-13 ... 149 °F)

Continuous Transmission Mode, with Forced Convection

PT	1	2	3	4
IQH3-FP-V1	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)
IQH4-18GM-V1	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 65 °C (-13 °F- 149 °F)	-25 °C- 65 °C (-13 °F- 149 °F)
IQH4-F61-V1	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)
IQH4-FP-V1	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)
IQH4-F198-[M]-V1	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)	-25 °C- 70 °C (-13 °F- 158 °F)

2026-05

6 Commissioning and Operation

6.1 Operation via the Communication Interface

This section shows you how to operate the RFID read/write device using an IDENTControl evaluation unit with serial interface. The examples include the syntax for coding the commands and parameters via the RS-232, Ethernet TCP/IP and PROFIBUS/PROFINET interfaces. Further details about these codes and the factory settings for your IDENTControl evaluation unit can be found in the corresponding manual.

Example:

In the examples below, the RFID read/write device is connected to channel 1 of the evaluation unit.

Reading Tags

Single Read Fixcode

Send the Single Read Fixcode command to the RFID read/write device.

	Serial	Ethernet	PROFIBUS/PROFINET
Command:	SF1	.00.04.1D.02	.1D.02
Confirmation:	-	.00.06.1D.02.FF.0E	.1D.02.FF.0E
Response:	.35.31	.00.06.1D.02.05.0F	.1D.02.05.0F

Table 6.1 Single Read Fixcode, no tag in the sensing range

Move a tag into the sensing range of the RFID read/write device. When the tag has been detected and the UID has been read out, the orange LED on the RFID read/write device lights up. The UID is displayed in the terminal program.

	Serial	Ethernet	PROFIBUS/PROFINET
Response:	.30.31.E0.08.01.48.82.ED.9B.AC	.00.0E.1D.02.00.10.E0.08.01.48.82.ED.9B.AC	.1D.02.00.10.E0.08.01.48.82.ED.9B.AC

Table 6.2 Single Read Fixcode, tag is entering the sensing range

Parameterizing the RFID Read/Write Device

Requesting and Setting the Transmitter Radiated Power

Read the transmitter radiated power of the RFID read/write device using the Read Parameter PT command:

	Serial	Ethernet	PROFIBUS/PROFINET
Command:	RP1QPT.00.00	.00.0A.BE.02.00.51.50.54.00.00	.BE.03.00.51.50.54.00.00
Confirmation:	-	.00.06.BE.02.FF.33	.BE.02.FF.33
Response:	.30.31.00.04	.00.0A.BE.02.00.34.00.02.00.04	.BE.02.00.34.00.02.00.04

Table 6.3 Read Parameter, **Power** mode

The transmitter radiated power of the RFID read/write device is set to **Power** mode.

Change the transmitter radiated power of the RFID read/write device to **Eco** mode using the Write Parameter PT command:

	Serial	Ethernet	PROFIBUS/PROFINET
Command:	WP1QPT.00.02.00.02	.00.0C.BF.02.00.51.50.54.00.02.00.02	.BF.02.00.51.50.54.04.00.02.00.02
Confirmation:	-	.00.06.BF.02.FF.35	.BF.02.FF.35
Response:	.30.31	.00.06.BF.02.00.36	.BF.02.00.36

Table 6.4 Write Parameter, **Eco** mode

6.2 13.56 MHz ISO15693 Tags

The tags of an RFID system with 13.56 MHz offer significantly quicker access to the data than a comparable RFID system based on a working frequency of 125 kHz. The 13.56 MHz system is standardized through ISO15693. A great variety of tags from different manufacturers using different RFID chips is supported.

Parameterization of the associated tag type is recommended to set the RFID read/write device to the tag being used. On delivery of the RFID read/write device, tag type 20 is preset. This setting guarantees access to the read-only code of ISO15693-compliant tags. The following table shows the tag types specified and recommended for the RFID read/write device.

13.56 MHz/ISO 15693 Tag Types

Pepperl +Fuchs designation	Chip type	Manufacturer	Size of read/write memory [byte]	Size of memory block [byte]
IQC20	All tags in accordance with ISO15693	-	Depending on tag	Depending on tag
IQC21	I-Code SLI(X)	NXP	112	4
IQC22	Tag-it HF-I Plus	Texas Instruments	256	4
IQC23	my-D SRF55V02P	Infineon	224	4
IQC24	my-D SRF55V10P	Infineon	992	4
IQC27	EM4135	EM Microelectronic	288	8
IQC31	Tag-it HFI standard	Texas Instruments	32	4
IQC32	Tag-it HFI pro	Texas Instruments	32	4
IQC33	FRAM MB89R118	Fujitsu	2000	8
IQC34	FRAM MB89R119	Fujitsu	232	4
IQC35	I-Code SLI-S	NXP	160	4
IQC36	I-Code SLI-L	NXP	32	4
IQC37	FRAM MB89R112	Fujitsu	8192	32
IQC38	EM4233	EM Microelectronic	208	4
IQC50	I-Code SLIX2	NXP	316	4
IQC51	I-Code 3	NXP	300	4

Table 6.5 13.56 MHz tag types in accordance with ISO 15693

2026-05

All ISO15693-compliant tags have a unique 8-byte read-only code. The read-only code is determined by the chip manufacturer. The user can only read it and it cannot be changed. In addition, the tags have a memory for user data. This can be written with application-specific data and read. The size of the memory for user data differs according to the tag type.

The memory is divided into blocks with a length of 4 bytes. There are also exceptions with a block length of 8 bytes or 32 bytes.

The read and write commands use the "Number of bytes" and "Start address" parameters. This defines how many bytes are accessed in the memory of the user data and from which memory address this starts. If the tag type used, for example, has a block length of 4 bytes, the values of the "Number of bytes" and "Start address" parameters must be a multiple of 4. In the case of a block length of 8 or 32 bytes, they are multiples of 8 or 32.

6.3 Sensing Range

IQH3 Series

The read/write device is classified with a sensing range of up to 30 cm. The range of the device depends on the tags to be identified and may therefore vary. You can increase or decrease the sensing range by changing the transmitter radiated power accordingly.

Note

To comply with the limit value of the magnetic field strength of max. 10 A/m in accordance with ISO 15693-1, a minimum distance of 50 mm must be maintained between the tag and the read/write device at a transmitter radiated power PT > 1. Otherwise, there is a risk that a tag will be damaged in the near range. Reduce the transmitter radiated power of the read/write device to PT = 1 to meet the limit value at small minimum distances.

IQH4 Series

The read/write devices are classified with a sensing range of up to 30 cm. The detection range of the devices varies depending on the design and is dependent on the tags to be identified. You can increase or decrease the sensing range by changing the transmitter radiated power (PT parameter).

6.4 Multiple Tags in Sensing Range

Each read or write command can access one, several, or all tags in the sensing range. To respond to specific tags in the sensing range, you can set a filter to the tag UIDs via the **TI** parameter. See "Tag ID Filter (TI)" on page 107.

Note

For applications in the multi-frame protocol, always use tags of the same tag type.

6.5 Command Overview

The commands in the list are described in detail on the following pages.

Below, the command syntax for IQH3/IQH4 series read/write devices is used in conjunction with an IC-KP2-xHB40-2V1D IDENTControl evaluation unit.

Read/Write Commands

Command name	Command abbreviation	Command code	Command description	System
Single Read Fixcode	**SF**	16#01	Single read access to the Fixcode (UID)	IQH3, IQH4
Enhanced Read Fix-code	**EF**	16#1D	Continuous read access to the Fix-code (UID)	IQH3, IQH4
Single Read 4-Byte Blocks	**SR**	16#10	Single read access to the user memory	IQH3, IQH4
Enhanced Read 4-Byte Blocks	**ER**	16#19	Continuous read access to the user memory	IQH3, IQH4
Single Write 4-Byte Blocks	**SW**	16#40	Single write access to the user memory	IQH3, IQH4
Enhanced Write 4-Byte Blocks	**EW**	16#1A	Continuous write access to the user memory	IQH3, IQH4
Single Write 4-Byte Blocks with Lock	**SL**	16#47	Single write access to the user memory with write protection	IQH3, IQH4
Enhanced Write 4-Byte Blocks with Lock	**EL**	16#48	Continuous write access to the user memory with write protection	IQH3, IQH4
Single Read Configu-ration	**SG**	16#61	Single read access to the configuration area	IQH3, IQH4
Enhanced Read Con-figuration	**EG**	16#68	Continuous read access to the config-uration area	IQH3, IQH4
Single Write Configura-tion	**SC**	16#12	Single write access to AFI/DSFID in the configuration area	IQH3, IQH4
Enhanced Write Con-figuration	**EC**	16#66	Continuous write access to AFI/DSFID in the configuration area	IQH3, IQH4
Fill Transponder	**S#**	16#AA	Write access to the user memory with fill characters	IQH3, IQH4
Selective Single Read Fixcode	**XSF**	16#E8	Single read access to the fixed code (UID) of a selected tag	IQH3[1], IQH4
Selective Enhanced Read Fixcode	**XEF**	16#E9	Continuous read access to the fixed code (UID) of a selected tag	IQH3[1]<Default ¬[1] Font>, IQH4

2026-05

Command name	Command abbreviation	Command code	Command description	System
Selective Single Read 4-Byte Blocks	**XSR**	16#E0	Single read access to the user memory of a selected tag	IQH3[1]<Default ¬[1] Font>, IQH4
Selective Enhanced Read 4-Byte Blocks	**XER**	16#D9	Continuous read access to the user memory of a selected tag	IQH3[1]<Default ¬[1] Font>, IQH4
Selective Single Write 4-Byte Blocks	**XSW**	16#E3	Single write access to the user memory of a selected tag	IQH3[1]<Default ¬[1] Font>, IQH4
Selective Enhanced Write 4-Byte Blocks	**XEW**	16#DC	Continuous write access to the user memory of a selected tag	IQH3[1]<Default ¬[1] Font>, IQH4
Selective Single Write 4-Byte Blocks with Lock	**XSL**	16#E4	Single write access to the user memory with write protection of a selected tag	IQH3[1]<Default ¬[1] Font>, IQH4
Selective Enhanced Write 4-Byte Blocks with Lock	**XEL**	16#E5	Continuous write access to the user memory with write protection of a selected tag	IQH3[1]<Default ¬[1] Font>, IQH4
Selective Single Read Configuration	**XSG**	16#DD	Single read access to AFI/DSFID of a selected tag	IQH3[1]<Default ¬[1] Font>, IQH4
Selective Enhanced Read Configuration	**XEG**	16#D6	Continuous read access to AFI/DSFID of a selected tag	IQH3[1]<Default ¬[1] Font>, IQH4
Selective Single Write Configuration	**XSC**	16#E1	Single write access to AFI/DSFID of a selected tag	IQH3[1]<Default ¬[1] Font>, IQH4
Selective Enhanced Write Configuration	**XEC**	16#DA	Continuous write access to AFI/DSFID of a selected tag	IQH3[1]<Default ¬[1] Font>, IQH4
Selective Fill Tag	**XS#**	16#E7	Write access to the user memory with fill characters of a selected tag	IQH3, IQH4

1. from SW Rev. 26

Configuration Commands

Command name	Command abbreviation	Command code	Command description	System
Read Parameter	**RP**	16#BE	Read device parameters	IQH3, IQH4
Write Parameter	**WP**	16#BF	Write device parameters	IQH3, IQH4

6.6 Read/Write Commands

When writing the following commands and responses, the IF parameter of the read/write device is set to 16#01, so that the Information Frame Mode is activated. This means that, for each tag read, a further telegram with the status value 16#0B is transmitted that contains the RSSI value for the signal strength of the tag response.

In addition, the parameter QV (Reply Frame Mode) is set to 16#4D, so that the multi-frame protocol is enabled. In the multi-frame protocol, each tag that responded is output. The response also contains the UID of the tag and additional length information for the UID. The response ends with an end telegram with the status 16#0F, containing the number of tags that have responded.

Single Read Fixcode (SF)

The SF command has the command code 16#01 and performs a single read operation on the UID of a tag within the detection zone.

The information is transmitted for each tag read in a separate data telegram with the status value 16#00. In addition, for each tag read, a further telegram with the status value 16#0B is transmitted that contains the RSSI value for the signal strength of the tag response.

The end of the command execution is indicated by an end telegram. The end telegram has the status value 16#0F and contains the number of tags identified during execution of the command.

Command Telegram

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#06							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#03							
Byte 5	Command	16#01							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#01							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#01							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#0F (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#01							
Byte 6	Status	16#0F							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 7	Data	<Number of Tags Byte 1>							
Byte 8	Data	<Number of Tags Byte 2>							
Byte 9	Data	<Number of Tags Byte 3>							
Byte 10	Data	<Number of Tags Byte 4>							
…	Data	16#00							
Byte 255	Data	16#00							

Enhanced Buffered Read Fixcode (EF)

The EF command has the command code 16#1D and performs a continuous read operation on the UID of a tag within the detection zone.

The information is transmitted for each tag read in a separate data telegram with the status value 16#00. In addition, for each tag read, a further telegram with the status value 16#0B is transmitted that contains the RSSI value for the signal strength of the tag response.

A tag leaving the detection zone is indicated by a telegram containing the UID of the tag. This telegram has the status value 16#05.

A Quit command stops the command execution.

Command Telegram

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#06							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#03							
Byte 5	Command	16#1D							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#1D							
Byte 6	Status	16#00							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#1D							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#05 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#1D							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 6	Status	16#05							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Single Read 4-Byte Blocks (SR)

The SR command has the command code 16#10 and performs a single read access to the user data area of a tag within the sensing range.

The <Start Address> parameter specifies the start address within the user data area. The value of <Start Address> is based on bytes. Only multiples of the memory block size can be parameterized. <Number of Bytes> defines the number of bytes to be read. The number of bytes must be a multiple of the memory block size.

The information is transmitted for each tag read in a separate data telegram with the status value 16#00. In addition, for each tag read, a further telegram with the status value 16#0B is sent that contains the RSSI value for the signal strength of the tag response.

The end of the command execution is indicated by an end telegram. The end telegram has the status value 16#0F and contains the number of tags that were identified during execution of the command.

Command Telegram

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0A							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#07							
Byte 5	Command	16#10							
Byte 6	Command parameter	<Start Address (High Byte)>							
Byte 7	Command parameter	<Start Address (Low Byte)>							
Byte 8	Command parameter	<Number of Bytes (High Byte)>							
Byte 9	Command parameter	<Number of Bytes (Low Byte)>							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#17							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#14							
Byte 5	Command	16#10							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
Byte 17	Data	16#00							
Byte 18	Data	16#04							
Byte 19	Data	<User Data Byte 1>							
Byte 20	Data	<User Data Byte 2>							
Byte 21	Data	<User Data Byte 3>							
Byte 22	Data	<User Data Byte 4>							
...	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#10							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#0F (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#10							
Byte 6	Status	16#0F							
Byte 7	Data	<Number of Tags Byte 1>							
Byte 8	Data	<Number of Tags Byte 2>							
Byte 9	Data	<Number of Tags Byte 3>							
Byte 10	Data	<Number of Tags Byte 4>							
…	Data	16#00							
Byte 255	Data	16#00							

Enhanced Read 4-Byte Blocks (ER)

The ER command has the command code 16#19 and performs a continuous read access to the user data area of a tag within the detection zone.

The <Start Address> parameter specifies the start address within the user data area. The value of <Start Address> is based on bytes. Only multiples of the memory block size can be parameterized. <Number of Bytes> defines the number of bytes to be read. The number of bytes must be a multiple of the memory block size.

The information is transmitted for each tag read in a separate data telegram with the status value 16#00. In addition, for each tag read, a further telegram with the status value 16#0B is transmitted, which contains information on the signal strength of the tag response (RSSI value).

A tag leaving the detection zone is indicated by a telegram containing the UID of the tag. This telegram has the status value 16#05.

A Quit command stops the command execution.

2026-05

Command Telegram

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0A							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#07							
Byte 5	Command	16#19							
Byte 6	Command parameter	<Start Address (High Byte)>							
Byte 7	Command parameter	<Start Address (Low Byte)>							
Byte 8	Command parameter	<Number of Bytes (High Byte)>							
Byte 9	Command parameter	<Number of Bytes (Low Byte)>							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#17							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#14							
Byte 5	Command	16#19							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
Byte 17	Data	16#00							
Byte 18	Data	16#04							
Byte 19	User Data	<User Data Byte 1>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 20	User Data	<User Data Byte 2>							
Byte 21	User Data	<User Data Byte 3>							
Byte 22	User Data	<User Data Byte 4>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#19							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#05 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#19							
Byte 6	Status	16#05							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
...	Data	16#00							
Byte 255	Data	16#00							

Single Write 4-Byte Blocks (SW)

The SW command has the command code 16#40 and performs a single write operation to the user data area of a tag within the sensing range.

The <Start Address> parameter specifies the start address within the user data area. The value of <Start Address> is based on bytes. Only multiples of the memory block size can be parameterized. <Number of Bytes> defines the number of bytes to be written. The number of bytes must be a multiple of the memory block size.

A separate data telegram with the status value 16#00 indicates a successful write operation for each tag written. The data telegram contains the UID of the tag to which the user data area was written. In addition, for each tag written, a further telegram with the status value 16#0B is transmitted, which contains information on the signal strength of the tag response (RSSI value).

The end of the command execution is indicated by an end telegram. The end telegram has the status value 16#0F and contains the number of tags that were written during execution of the command.

Command Telegram (4 Byte User Data)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0E							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0B							
Byte 5	Command	16#40							
Byte 6	Command parameter	16#00 <Start Address (High Byte)>							
Byte 7	Command parameter	16#00 <Start Address (Low Byte)>							
Byte 8	Command parameter	16#00 <Number of Bytes (High Byte)>							
Byte 9	Command parameter	16#04 <Number of Bytes (Low Byte)>							
Byte 10	Command parameter	<Data Byte 1>							
Byte 11	Command parameter	<Data Byte 2>							
Byte 12	Command parameter	<Data Byte 3>							
Byte 13	Command parameter	<Data Byte 4>							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#40							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#40							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

2026-05

Response Telegram 3 – Status 16#0F (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#40							
Byte 6	Status	16#0F							
Byte 7	Data	<Number of Tags Byte 1>							
Byte 8	Data	<Number of Tags Byte 2>							
Byte 9	Data	<Number of Tags Byte 3>							
Byte 10	Data	<Number of Tags Byte 4>							
…	Data	16#00							
Byte 255	Data	16#00							

Enhanced Write 4-Byte Blocks (EW)

The EW command has the command code 16#1A and performs a continuous write operation to the user data area of a tag within the sensing range.

The <Start Address> parameter specifies the start address within the user data area. The value of <Start Address> is based on bytes. Only multiples of the memory block size can be parameterized. <Number of Bytes> defines the number of bytes to be written. The number of bytes must be a multiple of the memory block size.

A separate data telegram with the status value 16#00 indicates a successful write operation for each tag written. The data telegram contains the UID of the tags to which the user data area was written. In addition, for each tag written, a further telegram with the status value 16#0B is transmitted, which contains information on the signal strength of the tag response (RSSI value).

A tag leaving the detection zone is indicated by a telegram containing the UID of the tag. This telegram has the status value 16#05.

A Quit command stops the command execution.

Command Telegram (4 Byte User Data)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0E							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0B							
Byte 5	Command	16#1A							
Byte 6	Command parameter	16#00 <Start Address (High Byte)>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 7	Command parameter	16#00 <Start Address (Low Byte)>							
Byte 8	Command parameter	16#00 <Number of Bytes (High Byte)>							
Byte 9	Command parameter	16#04 <Number of Bytes (Low Byte)>							
Byte 10	Command parameter	<Data Byte 1>							
Byte 11	Command parameter	<Data Byte 2>							
Byte 12	Command parameter	<Data Byte 3>							
Byte 13	Command parameter	<Data Byte 4>							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#1A							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#1A							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#05 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#1A							
Byte 6	Status	16#05							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Single Write 4-Byte Blocks with Lock (SL)

The SL command has the command code 16#47 and performs a single write operation to the user data area of a tag within the detection zone. The data is write-protected at the end of the write process, provided the tag offers this function. This applies to tag types 21, 22, 24, 33, 50, and 51. The write protection is permanent and cannot be undone. Write protection is only activated for memory blocks involved in the writing process. Data can continue to be written to all other memory blocks.

The <Start Address> parameter specifies the start address within the user data area. The value of <Start Address> is based on bytes. Only multiples of the memory block size can be parameterized. <Number of Bytes> defines the number of bytes to be written. The number of bytes must also be a multiple of the memory block size.

A separate data telegram with the status value 16#00 indicates a successful write operation for each tag written. The data telegram contains the UID of the tag to which the user data area was written. In addition, for each tag written, a further telegram with the status value 16#0B is transmitted, which contains information on the signal strength of the tag response (RSSI value).

The end of the command execution is indicated by an end telegram. The end telegram has the status value 16#0F and contains the number of tags that were written during execution of the command.

Command Telegram (4 Byte User Data)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0E							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0B							
Byte 5	Command	16#47							
Byte 6	Command parameter	16#00 <Start Address (High Byte)>							
Byte 7	Command parameter	16#00 <Start Address (Low Byte)>							
Byte 8	Command parameter	16#00 <Number of Bytes (High Byte)>							
Byte 9	Command parameter	16#04 <Number of Bytes (Low Byte)>							
Byte 10	Command parameter	<Data Byte 1>							
Byte 11	Command parameter	<Data Byte 2>							
Byte 12	Command parameter	<Data Byte 3>							
Byte 13	Command parameter	<Data Byte 4>							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#47							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#47							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#0F (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#47							
Byte 6	Status	16#0F							
Byte 7	Data	<Number of Tags Byte 1>							
Byte 8	Data	<Number of Tags Byte 2>							
Byte 9	Data	<Number of Tags Byte 3>							
Byte 10	Data	<Number of Tags Byte 4>							
…	Data	16#00							
Byte 255	Data	16#00							

Enhanced Write 4-Byte Blocks with Lock (EL)

The EL command has the command code 16#48 and performs a continuous write operation to the user data area of a tag within the detection zone. The data is write-protected at the end of the write process, provided the tag offers this function. This applies to tag types 21, 22, 24, 33, 50, and 51. The write protection is permanent and cannot be undone. Write protection is only activated for memory blocks involved in the writing process. Data can continue to be written to all other memory blocks.

The <Start Address> parameter specifies the start address within the user data area. The value of <Start Address> is based on bytes. Only multiples of the memory block size can be parameterized. <Number of Bytes> defines the number of bytes to be written. The number of bytes must also be a multiple of the memory block size.

A separate data telegram with the status value 16#00 indicates a successful write operation for each tag written. The data telegram contains the UID of the tags to which the user data area was written. In addition, for each tag written, a further telegram with the status value 16#0B is transmitted, which contains information on the signal strength of the tag response (RSSI value).

A tag leaving the detection zone is indicated by a telegram containing the fixed code of the tag. This telegram has the status value 16#05. A Quit command stops the command execution.

Command Telegram (4 Byte User Data)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0E							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0B							
Byte 5	Command	16#48							
Byte 6	Command parameter	16#00 <Start Address (High Byte)>							
Byte 7	Command parameter	16#00 <Start Address (Low Byte)>							
Byte 8	Command parameter	16#00 <Number of Bytes (High Byte)>							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 9	Command parameter	16#04 <Number of Bytes (Low Byte)>							
Byte 10	Command parameter	<Data Byte 1>							
Byte 11	Command parameter	<Data Byte 2>							
Byte 12	Command parameter	<Data Byte 3>							
Byte 13	Command parameter	<Data Byte 4>							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#48							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#48							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#05 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#48							
Byte 6	Status	16#05							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Single Get Configuration (SG)

The SG command has the command code 16#61 and performs a single read operation on the configuration area of a tag within the detection zone. This can be used, for example, to read the AFI or DSFID of the tag.

The <Config Address> parameter defines access to the configuration area and has the value 16#00 or 16#30.

The information is transmitted for each tag read in a separate data telegram with the status value 16#00. In addition, for each tag read, a further telegram with the status value 16#0B is sent that contains the RSSI value for the signal strength of the tag response.

2026-05

The end of the command execution is indicated by an end telegram. The end telegram has the status value 16#0F and contains the number of tags that were identified during execution of the command.

Command Telegram

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#07							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#04							
Byte 5	Command	16#61							
Byte 6	Command parameter	16#00 <Config Address>							
Byte ...	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#21							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#1E							
Byte 5	Command	16#61							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
Byte 17	Data	16#00							
Byte 18	Data	16#0E							
Byte 19	Data	<Info Flags>							
Byte 20	Data	<UID Byte 8>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 21	Data	<UID Byte 7>							
Byte 22	Data	<UID Byte 6>							
Byte 23	Data	<UID Byte 5>							
Byte 24	Data	<UID Byte 4>							
Byte 25	Data	<UID Byte 3>							
Byte 26	Data	<UID Byte 2>							
Byte 27	Data	<UID Byte 1>							
Byte 28	Data	<DSFID>							
Byte 29	Data	<AFI>							
Byte 30	Data	<VICC Memory Size (Number of Blocks -1)>							
Byte 31	Data	<VICC Memory Size (Block Size -1)>							
Byte 32	Data	<IC Reference>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#61							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#0F (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#61							
Byte 6	Status	16#0F							
Byte 7	Data	<Number of Tags Byte 1>							
Byte 8	Data	<Number of Tags Byte 2>							
Byte 9	Data	<Number of Tags Byte 3>							
Byte 10	Data	<Number of Tags Byte 4>							
…	Data	16#00							
Byte 255	Data	16#00							

Enhanced Get Configuration (EG)

The EG command has the command code 16#68 and performs a continuous read operation on the configuration area of a tag within the detection zone. This can be used, for example, to read the AFI or DSFID of the tag.

The <Config Address> parameter defines access to the configuration area and has the value 16#00 or 16#30.

The information is transmitted for each tag read in a separate data telegram with the status value 16#00. In addition, for each tag read, a further telegram with the status value 16#0B is transmitted, which contains information on the signal strength of the tag response (RSSI value).

A tag leaving the detection zone is indicated by a telegram containing the UID of the tag. This telegram has the status value 16#05.

A Quit command stops the command execution.

Command Telegram

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#07							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#04							
Byte 5	Command	16#68							
Byte 6	Command parameter	16#00 <Config Address>							
Byte ...	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#21							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#1E							
Byte 5	Command	16#68							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
Byte 17	Data	16#00							
Byte 18	Data	16#0E							
Byte 19	Data	<Info Flags>							
Byte 20	Data	<UID Byte 8>							
Byte 21	Data	<UID Byte 7>							
Byte 22	Data	<UID Byte 6>							
Byte 23	Data	<UID Byte 5>							
Byte 24	Data	<UID Byte 4>							
Byte 25	Data	<UID Byte 3>							
Byte 26	Data	<UID Byte 2>							
Byte 27	Data	<UID Byte 1>							
Byte 28	Data	<DSFID>							
Byte 29	Data	<AFI>							
Byte 30	Data	<VICC Memory Size (Number of Blocks -1)>							
Byte 31	Data	<VICC Memory Size (Block Size -1)>							
Byte 32	Data	<IC Reference>							
…	Data	16#00							
Byte 255	Data	16#00							

2026-05

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#68							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#05 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#68							
Byte 6	Status	16#05							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Single Set Configuration (SC)

The SC command has the command code 16#12 and performs a single write operation to the AFI or DSFID in the configuration area of a tag within the detection zone.

The <Config Address> parameter specifies whether the AFI (16#00 or 16#30) or the DSFID (16#01 or 16#31) should be written. The <Lock> parameter specifies whether the AFI or DSFID should be written without (16#00 or 16#30) or with write protection (16#01 or 16#31).

If the AFI or DSFID is already write-protected, a rewrite or write-protect operation has a response with status 5.

A separate data telegram with the status value 16#00 indicates a successful write operation for each tag written. The data telegram contains the UID of the tag to which the configuration area was written. In addition, for each tag written, a further telegram with the status value 16#0B is transmitted, which contains information on the signal strength of the tag response (RSSI value).

The end of the command execution is indicated by an end telegram. The end telegram has the status value 16#0F and contains the number of tags that were written during execution of the command.

Command Telegram

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#12							
Byte 6	Command parameter	16#00 <Config Address>							
Byte 7	Command parameter	<Lock>							
Byte 8	Command parameter	<DSFID / AFI>							
Byte 9	Command parameter	16#00 (always)							
Byte 10	Command parameter	16#00 (always)							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#12							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#12							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#0F (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 5	Command	16#12							
Byte 6	Status	16#0F							
Byte 7	Data	<Number of Tags Byte 1>							
Byte 8	Data	<Number of Tags Byte 2>							
Byte 9	Data	<Number of Tags Byte 3>							
Byte 10	Data	<Number of Tags Byte 4>							
…	Data	16#00							
Byte 255	Data	16#00							

Enhanced Set Configuration (EC)

The EC command has the command code 16#66 and performs a continuous write operation to the AFI or DSFID in the configuration area of a tag within the detection zone.

The <Config Address> parameter specifies whether the AFI (16#00 or 16#30) or the DSFID (16#01 or 16#31) should be written. The <Lock> parameter specifies whether the AFI or DSFID should be written without (16#00 or 16#30) or with write protection (16#01 or 16#31). The read/write device continuously attempts to write the AFI or DSFID to the configuration area.

If the AFI or DSFID is already write-protected, a rewrite or write-protect operation has a response with status 5.

A separate data telegram with the status value 16#00 indicates a successful write operation for each tag written. The data telegram contains the UID of the tag to which the configuration area was written. In addition, for each tag written, a further telegram with the status value 16#0B is transmitted, which contains information on the signal strength of the tag response (RSSI value).

A tag leaving the detection zone is indicated by a telegram containing the UID of the tag. This telegram has the status value 16#05.

A Quit command stops the command execution.

Command Telegram

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#66							
Byte 6	Command parameter	16#00 <Config Address>							
Byte 7	Command parameter	<Lock>							
Byte 8	Command parameter	<DSFID / AFI>							
Byte 9	Command parameter	16#00 (always)							
Byte 10	Command parameter	16#00 (always)							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

2026-05

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#66							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#66							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#05 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#66							
Byte 6	Status	16#05							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Fill Tag (S#)

The S# command has the command code 16#AA and performs a single write operation to the user data area of a tag within the detection zone.

The <Start Address> parameter specifies the start address within the user data area. The value of <Start Address> is based on bytes. Only multiples of the memory block size can be parameterized. <Number of Bytes> defines the number of bytes to be written. The number of bytes must also be a multiple of the memory block size. The <Fill Sign> parameter specifies the fill character to be written.

A separate data telegram with the status value 16#00 indicates a successful write operation for each tag written. The data telegram contains the UID of the tag to which the user data area was written. In addition, for each tag written, a further telegram with the status value 16#0B is transmitted, which contains information on the signal strength of the tag response (RSSI value).

The end of the command execution is indicated by an end telegram. The end telegram has the status value 16#0F and contains the number of tags that were written during execution of the command.

Note

If <Number of Bytes> is set to 16#0000, the tag is filled from <Start Address> to the end of the tag.

Command Telegram

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#AA							
Byte 6	Command parameter	16#00 <Start Address (High Byte)>							
Byte 7	Command parameter	16#00 <Start Address (Low Byte)>							
Byte 8	Command parameter	16#00 <Number of Bytes (High Byte)>							
Byte 9	Command parameter	16#04 <Number of Bytes (Low Byte)>							
Byte 10	Command parameter	<Fill Sign>							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#AA							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#AA							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#0F (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#AA							
Byte 6	Status	16#0F							
Byte 7	Data	<Number of Tags Byte 1>							
Byte 8	Data	<Number of Tags Byte 2>							
Byte 9	Data	<Number of Tags Byte 3>							
Byte 10	Data	<Number of Tags Byte 4>							
…	Data	16#00							
Byte 255	Data	16#00							

Selective Single Read Fixcode (XSF)

The XSF command has the command code 16#E8. In principle, the command works exactly like the SF command (16#01), but is only executed for one selected tag.

The parameters <UID Mask Byte 1> to <UID Mask Byte 8> are used to specify the UID of the selected tag. <Length UID Mask in Bit> specifies the length of the UID in bits.

2026-05

Command Telegram (8 Byte UID Mask)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#10							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0D							
Byte 5	Command	16#E8							
Byte 6	Command parameter	16#00 <Length UID Mask in Bit (High Byte)>							
Byte 7	Command parameter	16#40 <Length UID Mask in Bit (Low Byte)>							
Byte 8	Command parameter	<UID Mask Byte 1>							
Byte 9	Command parameter	<UID Mask Byte 2>							
Byte 10	Command parameter	<UID Mask Byte 3>							
Byte 11	Command parameter	<UID Mask Byte 4>							
Byte 12	Command parameter	<UID Mask Byte 5>							
Byte 13	Command parameter	<UID Mask Byte 6>							
Byte 14	Command parameter	<UID Mask Byte 7>							
Byte 15	Command parameter	<UID Mask Byte 8>							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#E8							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#E8							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#0F (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#E8							
Byte 6	Status	16#0F							
Byte 7	Data	<Number of Tags Byte 1>							
Byte 8	Data	<Number of Tags Byte 2>							
Byte 9	Data	<Number of Tags Byte 3>							
Byte 10	Data	<Number of Tags Byte 4>							
…	Data	16#00							
Byte 255	Data	16#00							

2026-05

Selective Enhanced Read Fixcode (XEF)

The XEF command has the command code 16#E9. In principle, the command works exactly like the EF command (16#1D), but is only executed for one selected tag.

The parameters <UID Mask Byte 1> to <UID Mask Byte 8> are used to specify the UID of the selected tag. <Length UID Mask in Bit> specifies the length of the UID in bits.

Command Telegram (8 Byte UID Mask)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#10							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0D							
Byte 5	Command	16#E9							
Byte 6	Command parameter	16#00 <Length UID Mask in Bit (High Byte)>							
Byte 7	Command parameter	16#40 <Length UID Mask in Bit (Low Byte)>							
Byte 8	Command parameter	<UID Mask Byte 1>							
Byte 9	Command parameter	<UID Mask Byte 2>							
Byte 10	Command parameter	<UID Mask Byte 3>							
Byte 11	Command parameter	<UID Mask Byte 4>							
Byte 12	Command parameter	<UID Mask Byte 5>							
Byte 13	Command parameter	<UID Mask Byte 6>							
Byte 14	Command parameter	<UID Mask Byte 7>							
Byte 15	Command parameter	<UID Mask Byte 8>							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#E9							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#E9							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#05 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#E9							
Byte 6	Status	16#05							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Selective Single Read 4-Byte Blocks (XSR)

The XSR command has the command code 16#E0. In principle, the command works exactly like the SR command (16#10), but is only executed for one selected tag.

The parameters <UID Mask Byte 1> to <UID Mask Byte 8> are used to specify the UID of the selected tag. <Length UID Mask in Bit> specifies the length of the UID in bits.

Command Telegram (8 Byte UID Mask, 4 Byte User Data)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#14							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#11							
Byte 5	Command	16#E0							
Byte 6	Command parameter	16#00 <Length UID Mask in Bit (High Byte)>							
Byte 7	Command parameter	16#40 <Length UID Mask in Bit (Low Byte)>							
Byte 8	Command parameter	<UID Mask Byte 1>							
Byte 9	Command parameter	<UID Mask Byte 2>							
Byte 10	Command parameter	<UID Mask Byte 3>							
Byte 11	Command parameter	<UID Mask Byte 4>							
Byte 12	Command parameter	<UID Mask Byte 5>							
Byte 13	Command parameter	<UID Mask Byte 6>							
Byte 14	Command parameter	<UID Mask Byte 7>							
Byte 15	Command parameter	<UID Mask Byte 8>							
Byte 16	Command parameter	<Start Address (High Byte)>							
Byte 17	Command parameter	<Start Address (Low Byte)>							
Byte 18	Command parameter	<Number of Bytes (High Byte)>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 19	Command parameter	<Number of Bytes (Low Byte)>							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#17							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#14							
Byte 5	Command	16#E0							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
Byte 17	Data	16#00							
Byte 18	Data	16#04							
Byte 19	Data	<User Data Byte 1>							
Byte 20	Data	<User Data Byte 2>							
Byte 21	Data	<User Data Byte 3>							
Byte 22	Data	<User Data Byte 4>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#E0							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#0F (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#E0							
Byte 6	Status	16#0F							
Byte 7	Data	<Number of Tags Byte 1>							
Byte 8	Data	<Number of Tags Byte 2>							
Byte 9	Data	<Number of Tags Byte 3>							
Byte 10	Data	<Number of Tags Byte 4>							
…	Data	16#00							
Byte 255	Data	16#00							

Selective Enhanced Read 4-Byte Blocks (XER)

The XER command has the command code 16#D9. In principle, the command works exactly like the ER command (16#19), but is only executed for one selected tag.

The parameters <UID Mask Byte 1> to <UID Mask Byte 8> are used to specify the UID of the selected tag. <Length UID Mask in Bit> specifies the length of the UID in bits.

Command Telegram (8 Byte UID Mask, 4 Byte User Data)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#14							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#11							
Byte 5	Command	16#D9							
Byte 6	Command parameter	16#00 <Length UID Mask in Bit (High Byte)>							
Byte 7	Command parameter	16#40 <Length UID Mask in Bit (Low Byte)>							
Byte 8	Command parameter	<UID Mask Byte 1>							
Byte 9	Command parameter	<UID Mask Byte 2>							
Byte 10	Command parameter	<UID Mask Byte 3>							
Byte 11	Command parameter	<UID Mask Byte 4>							
Byte 12	Command parameter	<UID Mask Byte 5>							
Byte 13	Command parameter	<UID Mask Byte 6>							
Byte 14	Command parameter	<UID Mask Byte 7>							
Byte 15	Command parameter	<UID Mask Byte 8>							
Byte 16	Command parameter	<Start Address (High Byte)>							
Byte 17	Command parameter	<Start Address (Low Byte)>							
Byte 18	Command parameter	<Number of Bytes (High Byte)>							
Byte 19	Command parameter	<Number of Bytes (Low Byte)>							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#17							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#14							
Byte 5	Command	16#D9							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
Byte 17	Data	16#00							
Byte 18	Data	16#04							
Byte 19	Data	<User Data Byte 1>							
Byte 20	Data	<User Data Byte 2>							
Byte 21	Data	<User Data Byte 3>							
Byte 22	Data	<User Data Byte 4>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#D9							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#05 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#D9							
Byte 6	Status	16#05							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Selective Single Write 4-Byte Blocks (XSW)

The XSW command has the command code 16#E3. In principle, the command works exactly like the SW command (16#40), but is only executed for one selected tag.

The parameters <UID Mask Byte 1> to <UID Mask Byte 8> are used to specify the UID of the selected tag. <Length UID Mask in Bit> specifies the length of the UID in bits.

Command Telegram (8 Byte UID Mask, 4 Byte User Data)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#18							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#15							
Byte 5	Command	16#E3							
Byte 6	Command parameter	16#00 <Length UID Mask in Bit (High Byte)>							
Byte 7	Command parameter	16#40 <Length UID Mask in Bit (Low Byte)>							
Byte 8	Command parameter	<UID Mask Byte 1>							
Byte 9	Command parameter	<UID Mask Byte 2>							
Byte 10	Command parameter	<UID Mask Byte 3>							
Byte 11	Command parameter	<UID Mask Byte 4>							
Byte 12	Command parameter	<UID Mask Byte 5>							
Byte 13	Command parameter	<UID Mask Byte 6>							
Byte 14	Command parameter	<UID Mask Byte 7>							
Byte 15	Command parameter	<UID Mask Byte 8>							
Byte 16	Command parameter	16#00 <Start Address (High Byte)>							
Byte 17	Command parameter	16#00 <Start Address (Low Byte)>							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 18	Command parameter	16#00 <Number of Bytes (High Byte)>							
Byte 19	Command parameter	16#04 <Number of Bytes (Low Byte)>							
Byte 20	Command parameter	<Data Byte 1>							
Byte 21	Command parameter	<Data Byte 2>							
Byte 22	Command parameter	<Data Byte 3>							
Byte 23	Command parameter	<Data Byte 4>							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#E3							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#E3							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#0F (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#E3							
Byte 6	Status	16#0F							
Byte 7	Data	<Number of Tags Byte 1>							
Byte 8	Data	<Number of Tags Byte 2>							
Byte 9	Data	<Number of Tags Byte 3>							
Byte 10	Data	<Number of Tags Byte 4>							
…	Data	16#00							
Byte 255	Data	16#00							

Selective Enhanced Write 4-Byte Blocks (XEW)

The XEW command has the command code 16#DC. In principle, the command works exactly like the EW command (16#1A), but is only executed for one selected tag.

The parameters <UID Mask Byte 1> to <UID Mask Byte 8> are used to specify the UID of the selected tag. <Length UID Mask in Bit> specifies the length of the UID in bits.

Command Telegram (8 Byte UID Mask, 4 Byte User Data)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#18							
Byte 2	Fragmentation Counter	16#00							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#15							
Byte 5	Command	16#DC							
Byte 6	Command parameter	16#00 <Length UID Mask in Bit (High Byte)>							
Byte 7	Command parameter	16#40 <Length UID Mask in Bit (Low Byte)>							
Byte 8	Command parameter	<UID Mask Byte 1>							
Byte 9	Command parameter	<UID Mask Byte 2>							
Byte 10	Command parameter	<UID Mask Byte 3>							
Byte 11	Command parameter	<UID Mask Byte 4>							
Byte 12	Command parameter	<UID Mask Byte 5>							
Byte 13	Command parameter	<UID Mask Byte 6>							
Byte 14	Command parameter	<UID Mask Byte 7>							
Byte 15	Command parameter	<UID Mask Byte 8>							
Byte 16	Command parameter	16#00 <Start Address (High Byte)>							
Byte 17	Command parameter	16#00 <Start Address (Low Byte)>							
Byte 18	Command parameter	16#00 <Number of Bytes (High Byte)>							
Byte 19	Command parameter	16#04 <Number of Bytes (Low Byte)>							
Byte 20	Command parameter	<Data Byte 1>							
Byte 21	Command parameter	<Data Byte 2>							
Byte 22	Command parameter	<Data Byte 3>							
Byte 23	Command parameter	<Data Byte 4>							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#DC							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#DC							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#05 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#DC							
Byte 6	Status	16#05							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Selective Single Write 4-Byte Blocks with Lock (XSL)

The XSL command has the command code 16#E4. In principle, the command works exactly like the SL command (16#47), but is only executed for one selected tag.

The parameters <UID Mask Byte 1> to <UID Mask Byte 8> are used to specify the UID of the selected tag. <Length UID Mask in Bit> specifies the length of the UID in bits.

Command Telegram (8 Byte UID Mask, 4 Byte User Data)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#18							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#15							
Byte 5	Command	16#E4							
Byte 6	Command parameter	16#00 <Length UID Mask in Bit (High Byte)>							
Byte 7	Command parameter	16#40 <Length UID Mask in Bit (Low Byte)>							
Byte 8	Command parameter	<UID Mask Byte 1>							
Byte 9	Command parameter	<UID Mask Byte 2>							
Byte 10	Command parameter	<UID Mask Byte 3>							
Byte 11	Command parameter	<UID Mask Byte 4>							
Byte 12	Command parameter	<UID Mask Byte 5>							
Byte 13	Command parameter	<UID Mask Byte 6>							
Byte 14	Command parameter	<UID Mask Byte 7>							
Byte 15	Command parameter	<UID Mask Byte 8>							
Byte 16	Command parameter	16#00 <Start Address (High Byte)>							
Byte 17	Command parameter	16#00 <Start Address (Low Byte)>							
Byte 18	Command parameter	16#00 <Number of Bytes (High Byte)>							
Byte 19	Command parameter	16#04 <Number of Bytes (Low Byte)>							
Byte 20	Command parameter	<Data Byte 1>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 21	Command parameter	<Data Byte 2>							
Byte 22	Command parameter	<Data Byte 3>							
Byte 23	Command parameter	<Data Byte 4>							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#E4							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 5	Command	16#E4							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#0F (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#E4							
Byte 6	Status	16#0F							
Byte 7	Data	<Number of Tags Byte 1>							
Byte 8	Data	<Number of Tags Byte 2>							
Byte 9	Data	<Number of Tags Byte 3>							
Byte 10	Data	<Number of Tags Byte 4>							
…	Data	16#00							
Byte 255	Data	16#00							

Selective Enhanced Write 4-Byte Blocks with Lock (XEL)

The XEL command has the command code 16#E5. In principle, the command works exactly like the SL command (16#48), but is only executed for one selected tag.

The parameters <UID Mask Byte 1> to <UID Mask Byte 8> are used to specify the UID of the selected tag. <Length UID Mask in Bit> specifies the length of the UID in bits.

Command Telegram (8 Byte UID Mask, 4 Byte User Data)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#18							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#15							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 5	Command	16#E5							
Byte 6	Command parameter	16#00 <Length UID Mask in Bit (High Byte)>							
Byte 7	Command parameter	16#40 <Length UID Mask in Bit (Low Byte)>							
Byte 8	Command parameter	<UID Mask Byte 1>							
Byte 9	Command parameter	<UID Mask Byte 2>							
Byte 10	Command parameter	<UID Mask Byte 3>							
Byte 11	Command parameter	<UID Mask Byte 4>							
Byte 12	Command parameter	<UID Mask Byte 5>							
Byte 13	Command parameter	<UID Mask Byte 6>							
Byte 14	Command parameter	<UID Mask Byte 7>							
Byte 15	Command parameter	<UID Mask Byte 8>							
Byte 16	Command parameter	16#00 <Start Address (High Byte)>							
Byte 17	Command parameter	16#00 <Start Address (Low Byte)>							
Byte 18	Command parameter	16#00 <Number of Bytes (High Byte)>							
Byte 19	Command parameter	16#04 <Number of Bytes (Low Byte)>							
Byte 20	Command parameter	<Data Byte 1>							
Byte 21	Command parameter	<Data Byte 2>							
Byte 22	Command parameter	<Data Byte 3>							
Byte 23	Command parameter	<Data Byte 4>							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#E5							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#E5							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#05 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#E5							
Byte 6	Status	16#05							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Selective Single Get Configuration (XSG)

The XSG command has the command code 16#DD. In principle, the command works exactly like the SG command (16#61), but is only executed for one selected tag.

The parameters <UID Mask Byte 1> to <UID Mask Byte 8> are used to specify the UID of the selected tag. <Length UID Mask in Bit> specifies the length of the UID in bits.

Command Telegram (8 Byte UID Mask)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#DD							
Byte 6	Command parameter	16#00 <Length UID Mask in Bit (High Byte)>							
Byte 7	Command parameter	16#40 <Length UID Mask in Bit (Low Byte)>							
Byte 8	Command parameter	<UID Mask Byte 1>							
Byte 9	Command parameter	<UID Mask Byte 2>							
Byte 10	Command parameter	<UID Mask Byte 3>							
Byte 11	Command parameter	<UID Mask Byte 4>							
Byte 12	Command parameter	<UID Mask Byte 5>							
Byte 13	Command parameter	<UID Mask Byte 6>							
Byte 14	Command parameter	<UID Mask Byte 7>							
Byte 15	Command parameter	<UID Mask Byte 8>							
Byte 16	Command parameter	16#00 <Config Address>							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

2026-05

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#21							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#1E							
Byte 5	Command	16#DD							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
Byte 17	Data	16#00							
Byte 18	Data	16#0E							
Byte 19	Data	<Info Flags>							
Byte 20	Data	<UID Byte 8>							
Byte 21	Data	<UID Byte 7>							
Byte 22	Data	<UID Byte 6>							
Byte 23	Data	<UID Byte 5>							
Byte 24	Data	<UID Byte 4>							
Byte 25	Data	<UID Byte 3>							
Byte 26	Data	<UID Byte 2>							
Byte 27	Data	<UID Byte 1>							
Byte 28	Data	<DSFID>							
Byte 29	Data	<AFI>							
Byte 30	Data	<VICC Memory Size (Number of Blocks -1)>							
Byte 31	Data	<VICC Memory Size (Block Size -1)>							
Byte 32	Data	<IC Reference>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#DD							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#0F (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#DD							
Byte 6	Status	16#0F							
Byte 7	Data	<Number of Tags Byte 1>							
Byte 8	Data	<Number of Tags Byte 2>							
Byte 9	Data	<Number of Tags Byte 3>							
Byte 10	Data	<Number of Tags Byte 4>							
…	Data	16#00							
Byte 255	Data	16#00							

Selective Enhanced Get Configuration (XEG)

The XEG command has the command code 16#D6. In principle, the command works exactly like the EG command (16#68), but is only executed for one selected tag.

The parameters <UID Mask Byte 1> to <UID Mask Byte 8> are used to specify the UID of the selected tag. <Length UID Mask in Bit> specifies the length of the UID in bits.

2026-05

Command Telegram (8 Byte UID Mask)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#D6							
Byte 6	Command parameter	16#00 <Length UID Mask in Bit (High Byte)>							
Byte 7	Command parameter	16#40 <Length UID Mask in Bit (Low Byte)>							
Byte 8	Command parameter	<UID Mask Byte 1>							
Byte 9	Command parameter	<UID Mask Byte 2>							
Byte 10	Command parameter	<UID Mask Byte 3>							
Byte 11	Command parameter	<UID Mask Byte 4>							
Byte 12	Command parameter	<UID Mask Byte 5>							
Byte 13	Command parameter	<UID Mask Byte 6>							
Byte 14	Command parameter	<UID Mask Byte 7>							
Byte 15	Command parameter	<UID Mask Byte 8>							
Byte 16	Command parameter	16#00 <Config Address>							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#21							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#1E							
Byte 5	Command	16#D6							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
Byte 17	Data	16#00							
Byte 18	Data	16#0E							
Byte 19	Data	<Info Flags>							
Byte 20	Data	<UID Byte 8>							
Byte 21	Data	<UID Byte 7>							
Byte 22	Data	<UID Byte 6>							
Byte 23	Data	<UID Byte 5>							
Byte 24	Data	<UID Byte 4>							
Byte 25	Data	<UID Byte 3>							
Byte 26	Data	<UID Byte 2>							
Byte 27	Data	<UID Byte 1>							
Byte 28	Data	<DSFID>							
Byte 29	Data	<AFI>							
Byte 30	Data	<VICC Memory Size (Number of Blocks -1)>							
Byte 31	Data	<VICC Memory Size (Block Size -1)>							
Byte 32	Data	<IC Reference>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#D6							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

2026-05

Response Telegram 3 – Status 16#05 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#D6							
Byte 6	Status	16#05							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Selective Single Set Configuration (XSC)

The XSC command has the command code 16#E1. In principle, the command works exactly like the SC command (16#12), but is only executed for one selected tag.

The parameters <UID Mask Byte 1> to <UID Mask Byte 8> are used to specify the UID of the selected tag. <Length UID Mask in Bit> specifies the length of the UID in bits.

Command Telegram (8 Byte UID Mask)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#15							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#12							
Byte 5	Command	16#E1							
Byte 6	Command parameter	16#00 <Length UID Mask in Bit (High Byte)>							
Byte 7	Command parameter	16#40 <Length UID Mask in Bit (Low Byte)>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 8	Command parameter	<UID Mask Byte 1>							
Byte 9	Command parameter	<UID Mask Byte 2>							
Byte 10	Command parameter	<UID Mask Byte 3>							
Byte 11	Command parameter	<UID Mask Byte 4>							
Byte 12	Command parameter	<UID Mask Byte 5>							
Byte 13	Command parameter	<UID Mask Byte 6>							
Byte 14	Command parameter	<UID Mask Byte 7>							
Byte 15	Command parameter	<UID Mask Byte 8>							
Byte 16	Command parameter	16#00 <Config Address>							
Byte 17	Command parameter	<Lock>							
Byte 18	Command parameter	<DSFID / AFI>							
Byte 19	Command parameter	16#00 (always)							
Byte 20	Command parameter	16#00 (always)							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#E1							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

2026-05

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#E1							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#0F (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#E1							
Byte 6	Status	16#0F							
Byte 7	Data	<Number of Tags Byte 1>							
Byte 8	Data	<Number of Tags Byte 2>							
Byte 9	Data	<Number of Tags Byte 3>							
Byte 10	Data	<Number of Tags Byte 4>							
…	Data	16#00							
Byte 255	Data	16#00							

Selective Enhanced Set Configuration (XEC)

The XEC command has the command code 16#DA. In principle, the command works exactly like the EC command (16#66), but is only executed for one selected tag.

The parameters <UID Mask Byte 1> to <UID Mask Byte 8> are used to specify the UID of the selected tag. <Length UID Mask in Bit> specifies the length of the UID in bits.

Command Telegram (8 Byte UID Mask)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#15							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#12							
Byte 5	Command	16#DA							
Byte 6	Command parameter	16#00 <Length UID Mask in Bit (High Byte)>							
Byte 7	Command parameter	16#40 <Length UID Mask in Bit (Low Byte)>							
Byte 8	Command parameter	<UID Mask Byte 1>							
Byte 9	Command parameter	<UID Mask Byte 2>							
Byte 10	Command parameter	<UID Mask Byte 3>							
Byte 11	Command parameter	<UID Mask Byte 4>							
Byte 12	Command parameter	<UID Mask Byte 5>							
Byte 13	Command parameter	<UID Mask Byte 6>							
Byte 14	Command parameter	<UID Mask Byte 7>							
Byte 15	Command parameter	<UID Mask Byte 8>							
Byte 16	Command parameter	16#00 <Config Address>							
Byte 17	Command parameter	<Lock>							
Byte 18	Command parameter	<DSFID / AFI>							
Byte 19	Command parameter	16#00 (always)							
Byte 20	Command parameter	16#00 (always)							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#DA							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#DA							
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#05 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#DA							
Byte 6	Status	16#05							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Selective Fill Tag (XS#)

The XS# command has the command code 16#E7. In principle, the command works exactly like the S# command (16#AA), but is only executed for one selected tag.

The parameters <UID Mask Byte 1> to <UID Mask Byte 8> are used to specify the UID of the selected tag. <Length UID Mask in Bit> specifies the length of the UID in bits.

Command Telegram (8 Byte UID Mask)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#15							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#12							
Byte 5	Command	16#E7							
Byte 6	Command parameter	16#00 <Length UID Mask in Bit (High Byte)>							
Byte 7	Command parameter	16#40 <Length UID Mask in Bit (Low Byte)>							
Byte 8	Command parameter	<UID Mask Byte 1>							
Byte 9	Command parameter	<UID Mask Byte 2>							
Byte 10	Command parameter	<UID Mask Byte 3>							
Byte 11	Command parameter	<UID Mask Byte 4>							
Byte 12	Command parameter	<UID Mask Byte 5>							
Byte 13	Command parameter	<UID Mask Byte 6>							
Byte 14	Command parameter	<UID Mask Byte 7>							
Byte 15	Command parameter	<UID Mask Byte 8>							
Byte 16	Command parameter	16#00 <Start Address (High Byte)>							
Byte 17	Command parameter	16#00 <Start Address (Low Byte)>							
Byte 18	Command parameter	16#00 <Number of Bytes (High Byte)>							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 19	Command parameter	16#04 <Number of Bytes (Low Byte)>							
Byte 20	Command parameter	<FillSign>							
…	Command parameter	16#00							
Byte 255	Command parameter	16#00							

Response Telegram 1 – Status 16#00 (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#E7							
Byte 6	Status	16#00							
Byte 7	Data	16#00 <Length UID in Byte (High Byte)>							
Byte 8	Data	16#08 <Length UID in Byte (Low Byte)>							
Byte 9	Data	<UID Byte 1>							
Byte 10	Data	<UID Byte 2>							
Byte 11	Data	<UID Byte 3>							
Byte 12	Data	<UID Byte 4>							
Byte 13	Data	<UID Byte 5>							
Byte 14	Data	<UID Byte 6>							
Byte 15	Data	<UID Byte 7>							
Byte 16	Data	<UID Byte 8>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 2 – Status 16#0B (Information Frame Mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#E7							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 6	Status	16#0B							
Byte 7	Data	16#01 <Info Type>							
Byte 8	Data	<RSSI>							
…	Data	16#00							
Byte 255	Data	16#00							

Response Telegram 3 – Status 16#0F (Multi-Frame Protocol)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#E7							
Byte 6	Status	16#0F							
Byte 7	Data	<Number of Tags Byte 1>							
Byte 8	Data	<Number of Tags Byte 2>							
Byte 9	Data	<Number of Tags Byte 3>							
Byte 10	Data	<Number of Tags Byte 4>							
…	Data	16#00							
Byte 255	Data	16#00							

6.7 Configuration Commands

The "Read Parameter" and "Write Parameter" configuration commands can be used to read or change the parameters of the device.

A system code is required to access the HF parameters on the device. This distinguishes between other systems in which parameters can also be changed. For IQH3/IQH4 series devices, the system code is "Q" (16#51).

Read Parameter (RP)

The "Read Parameter" (RP) command has the command code 16#BE and reads out a device parameter. The response to the configuration command is a status message from the read/write device and the corresponding data.

Command Telegram

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							

2026-05

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	<Telegram Length (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<Telegram Length (Low Byte)>							
Byte 5	Command	16#BE							
Byte 6	System code	<System Code> 16#51 "Q"							
Byte 7	Parameter Name (High Byte)	<Parameter Name (High Byte)>							
Byte 8	Parameter Name (Low Byte)	<Parameter Name (Low Byte)>							
Byte 9	Length Parameter (High Byte)	<Parameter Length (High Byte)>							
Byte 10	Length Parameter (Low Byte)	<Parameter Length (Low Byte)>							
Byte 11	Command parameter	<Parameter Data Byte 1>							
Byte 12	Command parameter	<Parameter Data Byte 2>							
…	Command parameter	…							
…	Command parameter	<Parameter Data Byte X>							
…	Not relevant	16#00							
Byte 255	Not relevant	16#00							

Response Telegram

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	<Frame Length>			
Byte 1	Frame Length	<Frame Length>							
Byte 2	Fragmentation Counter	<Fragmentation Counter>							
Byte 3	Telegram Length (High Byte)	<Telegram Length (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<Telegram Length (Low Byte)>							
Byte 5	Command	16#BE							
Byte 6	Status	<Status>							
Byte 7	Data	<Parameter Data Byte 1>							
Byte 8	Data	<Parameter Data Byte 2>							
…	Data	…							
…	Data	<Parameter Data Byte X>							
…	Not relevant	16#00							
Byte 255	Not relevant	16#00							

Write Parameter (WP)

The "Write Parameter" (WP) command has the command code 16#BF. The command can be used to change device parameters. The response to the configuration command is a status message from the read/write device.

Command Telegram

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	<Telegram Length (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<Telegram Length (Low Byte)>							
Byte 5	Command	16#BF							
Byte 6	System code	<System Code> 16#51 "Q"							
Byte 7	Parameter Name (High Byte)	<Parameter Name (High Byte)>							
Byte 8	Parameter Name (Low Byte)	<Parameter Name (Low Byte)>							
Byte 9	Length Parameter (High Byte)	<Parameter Length (High Byte)>							
Byte 10	Length Parameter (Low Byte)	<Parameter Length (Low Byte)>							
Byte 11	Command parameter	<Parameter Data Byte 1>							
Byte 12	Command parameter	<Parameter Data Byte 2>							
…	Command parameter	…							
…	Command parameter	<Parameter Data Byte X>							
…	Not relevant	16#00							
Byte 255	Not relevant	16#00							

Response Telegram

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control Byte / Frame Length	D_S	U_M	U_D	0	<Frame Length>			
Byte 1	Frame Length	16#07							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#04							
Byte 5	Command	16#BF							
Byte 6	Status	16#00							
…	Not relevant	16#00							
Byte 255	Not relevant	16#00							

2026-05

6.7.1 Parameter Overview

The following parameters can be read or written using the Read Parameter **RP** and Write Parameter **WP** configuration commands:

Abbreviation	Command code	Parameter description	Read/write	Default value	System
AF	16#4146	AFI Filters by AFI for tag access	Read + write	16#00	IQH3, IQH4
BB	16#4242	Back to Box Resets all parameters to factory settings	Write	-	IQH3[1], IQH4
CT	16#4654	Tag Type (Service) Sets the tag type	Read + write	16#14	IQH3, IQH4
DA	16#4441	Device Address Device address setting	Read + write	16#FF	IQH3[1]<Default ¬[1] Font>, IQH4
DD	16#4444	Diagnosis Data (Service) Reads out the combined diagnostic data for parameters OH, ST, TE and VO	Read	-	IQH3, IQH4
DN	16#444E	Device Serial Number Reads the serial number of the device	Read	-	IQH3[1]<Default ¬[1] Font>, IQH4
DR	16#4452	Data Rate Transmission rate setting	Read + write	16#00	IQH3, IQH4
DS	16#4453	Data Swap Rotates data blocks in blocks	Read + write	16#01	IQH3[1]<Default ¬[1] Font>, IQH4
E5	16#4535	Tag Lost Smoothing Sets the number of unsuccessful attempts to access a tag before a Status 5 message is issued	Read + write	16#05	IQH3, IQH4
ED	16#4544	Standalone End Delimiter (Service) Selects the end delimiter	Read + write	16#xx	IQH3[1]<Default ¬[1] Font>, IQH4
FR	16#4652	Sequence Mode (Service) Sets the behavior for write access	Read + write	16#01	IQH3, IQH4
IF	16#4946	Information Frame Mode Outputs additional information about tag access	Read + write	16#00	IQH3, IQH4
LM	16#4C4D	LED Masking Sets the operating status display via LEDs	Read + write	16#0000	IQH3[1]<Default ¬[1] Font>, IQH4
MF	16#4D46	Reflected Power (Service) Measures the reflected transmitter radiated power	Read	-	IQH3, IQH4
ND	16#4E44	Noise Detection Level Sets the noise display threshold via LED	Read + write	16#32	IQH3[1]<Default ¬[1] Font>, IQH4

Abbreviation	Command code	Parameter description	Read/write	Default value	System
NT	16#4E54	Number of Tags to Find Sets the cancellation criterion for read/write commands	Read + write	16#FF	IQH3, IQH4
OH	16#4F48	Operating Hours Reads the operating hours counter	Read	-	IQH3, IQH4
PT	16#5054	Transmission Power Sets the transmitter radiated power of the device	Read + write	16#0004/ 16#0001	IQH3, IQH4
QV	16#5156	Reply Frame Mode Sets the protocol mode	Read + write	16#53	IQH3, IQH4
QW	16#5157	Q-Value Sets the time slots for accessing tags	Read + write	16#00	IQH3, IQH4
RC	16#5243	Region Code (Service) Reads the country code	Read	16#01	IQH3, IQH4
RD	16#5244	Reset to Default Resets some parameters to factory settings	Write	-	IQH3, IQH4
RL	16#524C	RSSI Lower Limit Sets the lower RSSI threshold	Read + write	16#00	IQH3[1]<Default ¬[1] Font>, IQH4
RU	16#5255	RSSI Upper Limit Sets the upper RSSI threshold	Read + write	16#64	IQH3[1]<Default ¬[1] Font>, IQH4
RV	16#5256	GIT Revision (Service) Reads the software revision number	Read	-	IQH3, IQH4
SN	16#534E	Serial Number (Service) Reads the serial number of the microcontroller	Read	-	IQH3, IQH4
ST	16#5354	Frontend State Reads the frontend status	Read	16#00	IQH3, IQH4
T2	16#5432	Application Tag Sets the application tag	Read + write	Your automation, our passion.	IQH3[1]<Default ¬[1] Font>, IQH4
T3	16#5433	Function Tag Sets the function tag	Read + write	***	IQH3[1]<Default ¬[1] Font>, IQH4
T4	16#5434	Location Tag Sets the location tag	Read + write	***	IQH3[1]<Default ¬[1] Font>, IQH4
T5	16#5435	Installation Tag Sets the installation tag	Read + write	***	IQH3[1]<Default ¬[1] Font>, IQH4
TA	16#5441	Tries Allowed Sets the number of access attempts for tags	Read + write	16#02	IQH3, IQH4
TE	16#5445	Temperature Reads the device temperature	Read	-	IQH3, IQH4
TI	16#5449	Tag ID Filter Sets the filter on the UID	Read + write	-	IQH3, IQH4

2026-05

Abbreviation	Command code	Parameter description	Read/write	Default value	System
TO	16#544F	Overtemperature Handling Sets the device behavior in the event of excess temperature	Read + write	16#00	IQH3, IQH4
UB	16#5542	Host Uart Baudrate (Service) Sets the baud rate	Read + write	16#00009600	IQH3[1]<Default ¬[1] Font>, IQH4
UT	16#5554	Host Uart Timeout (Service) Sets the timeout	Read + write	16#0064	IQH3[1]<Default ¬[1] Font>, IQH4
VO	16#564F	Voltages Reads internal voltage and current values	Read	-	IQH3, IQH4

1. from SW Rev. 26

The parameters are saved in the read/write device as nonvolatile.

6.7.2 Parameters

AFI (AF)

The AF parameter can be used to filter tags by AFI. Read/write commands are executed on tags with an AFI that matches the value of the filter.

Parameter abbreviation:	AF (16#4146)
Parameter length:	1 byte
Default:	16#00 = all tags respond
Value range:	16#00 ... 16#FF (0_{dec} ... 255_{dec})

Back to Box (BB)

The BB parameter can be used to reset all RFID read/write device settings to their factory setting.

Parameter abbreviation:	BB (16#4242)
Factory setting:	See chapter 6.7.1

Tag Type (CT)

This parameter sets the tag type with which the RFID read/write device communicates.

Parameter abbreviation:	CT (16#4654)
Parameter length:	1 byte
Default:	16#14 (20_{dec})
Value range:	16#00 ... 16#27, 16#32, 16#33 (20_{dec} ... 39_{dec}, 50_{dec}, 51_{dec})

For supported tag types, see chapter 6.2.

With the tag type "20", mixed operation of different tag types is possible.

Note

You can also set the tag type using the CT system command. This command is described in your IDENTControl evaluation unit manual. We recommend using the CT system command because the tag type is stored in the IDENTControl evaluation unit and sent to the RFID read/write device at startup.

Device Address (DA)

The DA parameter can be used to set the device address of the RFID read/write device.

Parameter abbreviation:	DA (16#4441)
Parameter length:	1 byte
Default:	16#FF
Value range:	16#00 … 16#FF

P+F Serial Number (DN)

The DN parameter can be used to read the serial number of the RFID read/write device.

Parameter abbreviation:	DN (16#444E)
Parameter length:	14 bytes

Data Rate (DR)

This parameter sets the data transmission rate between read/write device and tag.

Parameter abbreviation:	DR (16#4452)
Parameter length:	1 byte
Default:	16#00
Value range:	16#00 = normal = 26 kBit/s 16#01 = fast read mode = 53 kBit/s[1]

1. ISO/IEC 15693-2 X2

The "fast read mode" is supported only by tag types IQC21, IQC33, IQC37, IQC50 and IQC51 with a memory size of two blocks or more.

Data Swap (DS)

The DS parameter can be used to set the orientation of the data in a block of the user memory.

Parameter abbreviation:	DS (16#4453)
Parameter length:	1 byte
Default:	16#01
Value range:	16#00 = no byte-by-byte exchange of data in the user memory 16#01 = byte-by-byte exchange of blocks in the user memory (compatible with IQT1)

For a tag with a block size of 4 bytes, the data can be oriented as follows, for example.

2026-05

Normal orientation, DS = 16#00

Data block 0				Data block 1			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 0	Byte 1	Byte 2	Byte 2

Reverse orientation, DS = 16#01

Data block 0				Data block 1			
Byte 3	Byte 2	Byte 1	Byte 0	Byte 3	Byte 2	Byte 1	Byte 0

Tag Lost Smoothing (E5)

The E5 parameter sets the number of unsuccessful write/read attempts when executing an Enhanced command before the device outputs the telegram with the status value 5. Via the telegram with the status 5, the device indicates that a tag has left the sensing range or could no longer be identified. This parameter is only used when performing Enhanced commands. When executing Single commands, the parameter has no significance.

Status 0 and status A do not depend on this parameter, and are output immediately.

Parameter abbreviation:	E5 (16#4535)
Parameter length:	1 byte
Default:	16#05 = five unsuccessful read/write attempts
Value range:	16#00 ... 16#0A unsuccessful read/write attempts

Effects with reduced number of unsuccessful read/write attempts:

⊕ Faster response time in Enhanced mode.

⊖ Multiple status 5 messages in the event of unstable tag reading.

Information Frame Mode (IF)

The IF parameter allows you to output the RSSI value of the reading as additional information if the reading was successful and the multi-frame protocol is activated; see "Reply Frame Mode (QV)" on page 103.

Parameter abbreviation:	IF (16#4946)
Parameter length:	1 byte
Default:	16#00
Value range:	16#00 = Information Frame Mode deactivated 16#01 = Information Frame Mode activated

LED Masking (LM)

The LM parameter can be used to set which operating status is to be displayed via the LEDs of the RFID read/write device.

Parameter abbreviation:	LM (16#4C4D)
Parameter length:	2 bytes
Default:	16#0000
Value range[1]:	16#0000 = all operating statuses are output via LED 16#0001 = operating status "Interference from external signals in the usage spectrum" is suppressed 16#0002 = operating status "Detuning due to surrounding metal" is suppressed 16#0004 = operating status "Excess temperature warning" is suppressed

1. Binary combinations are possible

Note

The operating statuses "Excess temperature error" (ST = 16#0008) and "Undervoltage/overvoltage error" (ST = 16#0010) cannot be suppressed.

The operating status of the RFID read/write device can still be read out at any time via the ST parameter.

Noise Detection Level (ND)

The ND parameter defines the threshold (0–100%) from which the status message LED is activated due to a reduction in the read/write range (e.g. due to detuning).

For example, if the value is set to 50dec, the LED lights up as soon as the reading/writing distance of the device falls by 50 % from the optimum value.

Parameter abbreviation:	ND (16#4E44)
Parameter length:	1 byte
Default:	16#32 (50_{dec})
Value range:	16#00 … 16#64 (0_{dec} ... 100_{dec})

Number of Tags to Find (NT)

The NT parameter specifies the number of tags that the RFID read/write device searches for within the sensing range. Each command is repeated according to the Number of Attempts (TA); see "Tries Allowed (TA)" on page 107.

If the number of tags found during the repeat operations reaches or exceeds the specified NT value, all further read attempts are canceled. The command is canceled and the data is output.

Note

If the number of tags is set to 255 (= .FF), the function is deactivated. This parameter applies to Single commands and does not impact Enhanced commands.

Parameter abbreviation:	NT (16#4E54)
Parameter length:	1 byte
Default:	16#FF
Value range:	16#00 ... 16#14 16#FF = off

2026-05

Operating Hours (OH)

You can use the OH parameter to output for how long the write/read device is in operation or for how long a write/read command is executed.

Parameter abbreviation:	OH(16#4F48)
Parameter length:	8 bytes (byte 1–4: RFID read/write device operating time in hours; byte 5–8: read/write commands operating time in hours)

Transmission Power (PT)

The PT parameter can be used to set the transmitter radiated power of the RFID read/write device. The transmitter radiated power affects the detection range of a tag's detection zone. As the transmitter radiated power increases, the detection range for read and write access to a tag is increased. A transmitter radiated power level of PT1 is preset in the factory setting. It is possible to use up to four power levels (PT1 … PT4) in the device. This enables a ramp to be generated for a continuous increase in the transmitter radiated power. When used in conjunction with a "Single Read" or "Single Write" command and the definition of the cancellation criterion (NT parameter), the read or write operation is canceled as soon as the set number of tags is detected. Setting multiple transmitter radiated power levels increases the execution time of a read or write operation. This is due to the inventory runs that are executed for each power level. The NT cancellation criterion can be set to reduce the execution time.

Parameter abbreviation:	PT (16#5054)
Parameter length:	2 bytes
Default:	IQH3 –〉 16#0004 IQH4 –〉 16#0001
Value range:	16#0001 (Min) 16#0002 (Eco) 16#0003 (Normal) 16#0004 (Maximum)

Reply Frame Mode (QV)

The QV parameter specifies the output protocol as a single-frame protocol or a multi-frame protocol.

The single-frame protocol complies with the P+F standard for LF systems IPH and HF systems IQH1. If there is more than one tag in the sensing range, status A is output as a warning.

In the multi-frame protocol, each tag that responded is output. An output is issued which begins with status F, contains the number of responses, and stops the command.

Parameter abbreviation:	QV(16#5156)
Parameter length:	1 byte
Default:	16#53
Value range:	16#53 = single-frame protocol 16#4D = multi-frame protocol

Q-Value (QW)

The QW parameter defines the number of tags expected in the sensing range. This setting affects the time slot when communicating with a tag to avoid collisions due to simultaneous communication between multiple tags.

Parameter abbreviation:	QW (16#5157)
Parameter length:	1 byte
Default:	16#00
Value range:	16#00 = one tag 16#01 = two tags 16#02 = multiple tags (approx. 4) 16#03 = many tags (approx. 8) 16#04 = very many tags (approx. 16)

Note

The execution time for tag access is reduced by reducing the Q value. The Q value influences the execution time of write/read commands.

Reset to Default (RD)

The RD parameter is used to reset all RFID read/write device settings to their factory setting.

The following parameters are excluded: Device Address (DA), Info Frame Mode (IF)[1], Reply Frame Mode (QV)[1]<Default ¬[1] Font>, UART Baud Rate (UB), UART Timeout (UT), Application Specific Tag (T2), Function Tag (T3), Location Tag (T3), and Installation Tag (T4).

The BB parameter can be used to reset all settings.

Parameter abbreviation:	RD (16#5244)

Factory setting: see chapter 6.7.1.

RSSI Lower Limit (RL)

The RL parameter can be used to set the lower RSSI threshold. As soon as the RSSI value of a read tag falls below the set threshold, it is no longer output.

Parameter abbreviation:	RL (16#524C)
Parameter length:	1 byte
Default:	16#00
Value range:	16#00 … 16#64 (0_{dec} ... 100_{dec})

Legend	Designation
1	Zone A
2	RSSI threshold
3	Zone B
4	Zone C
5	Tag
6	Reading/writing distance

1. IQH3 from SW Rev. 26

2026-05

PEPPERL+FUCHS

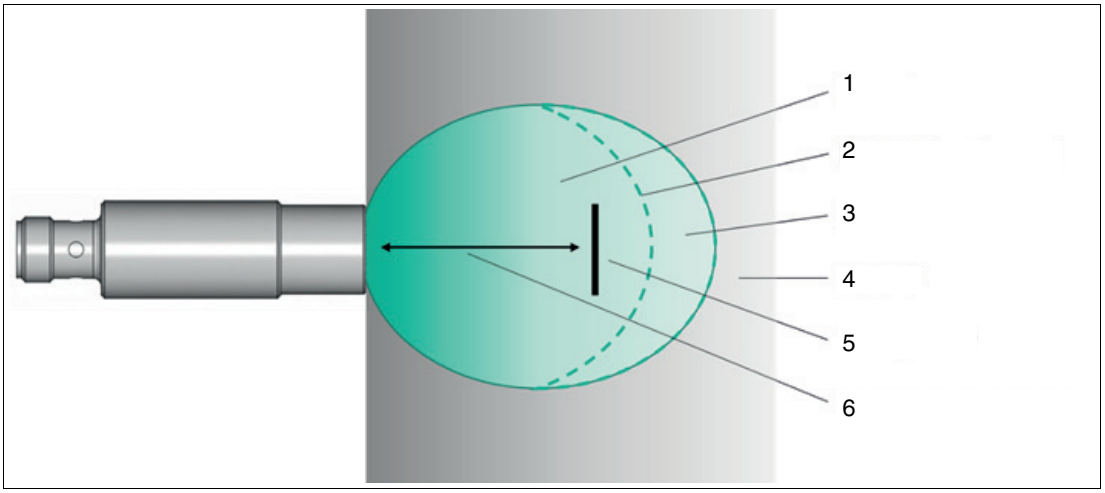


Figure 6.1 Schematic depiction of the reading range of the IQH4-18GM-V1, in relation to RSSI

Zone	RSSI	Note
A	RSSI tag ≥ RSSI threshold	The tag is within the sensing range and is output.
B	RSSI tag < RSSI threshold	The tag is within the maximum sensing range but is not output.
C	-	The tag is outside the sensing range.

RSSI Upper Limit (RU)

The RL parameter can be used to set the upper RSSI threshold. As soon as the RSSI value of a read tag exceeds the set threshold, it is no longer output.

Parameter abbreviation:	RU (16#5255)
Parameter length:	1 byte
Default:	16#64
Value range:	16#00 … 16#64 (0_{dec} ... 100_{dec})

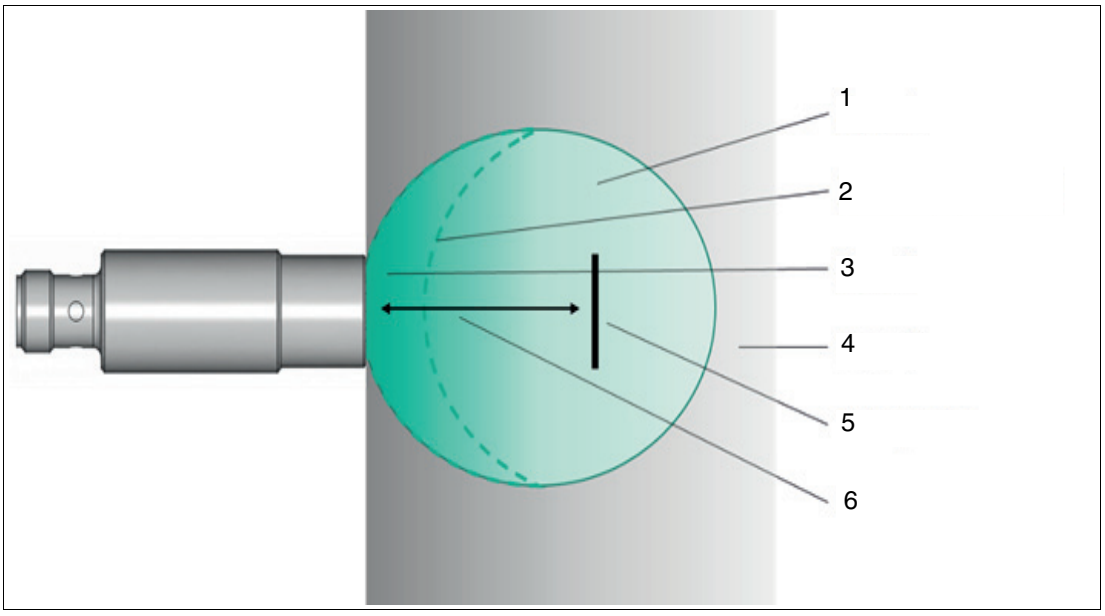


Figure 6.2 Schematic depiction of the reading range of the IQH4-18GM-V1, in relation to RSSI

Zone	RSSI	Note
A	RSSI tag ≤ RSSI threshold	The tag is within the sensing range and is output.
B	RSSI tag > RSSI threshold	The tag is within the sensing range but is not output.
C	-	The tag is outside the sensing range.

Frontend State (ST)

The ST parameter can be used to check the operating status of the RFID read/write device.

Parameter abbreviation:	ST (16#5354)
Parameter length:	2 bytes (byte 1: Current status; byte 2: Stored status since last read-out)
Default:	16#00
Value range[1]:	16#00 = no interference 16#01= interference from external signals in the usage spectrum, detection range may be limited. 16#02 = detuning due to surrounding metal 16#04 = excess temperature warning, internal device temperature 80 °C ... 85 °C 16#08 = excess temperature error, internal device temperature > 85 °C 16#10 = undervoltage/overvoltage error, voltage < 17.5 V DC or > 30.5 V DC

1. Binary combinations are possible

Application Specific Tag (T2)

The T2 parameter can be used to mark the device with an application tag in order to store application-specific information on the device.

Parameter abbreviation:	T2 (16#5432)
Parameter length:	Max. 32 bytes
Default:	Your automation, our passion.

Function Tag (T3)

The T3 parameter can be used to mark the device with a function tag in order to store function-specific information on the device.

Parameter abbreviation:	T3 (16#5433)
Parameter length:	Max. 32 bytes
Default:	***

Location Tag (T4)

The T4 parameter can be used to mark the device with a location tag to store location-specific information on the device.

Parameter abbreviation:	T4 (16#5434)
Parameter length:	Max. 32 bytes
Default:	***

2026-05

Installation Tag (T5)

The T5 parameter can be used to mark the device with an installation tag in order to save installation-specific information (e.g. date or details of initial commissioning) on the device.

Parameter abbreviation:	T5 (16#5435)
Parameter length:	Max. 32 bytes
Default:	***

Tries Allowed (TA)

The Tries Allowed (TA) parameter sets the number of access attempts during execution of a read/write operation on a tag.

Parameter abbreviation:	TA (16#5441)
Parameter length	1 byte
Default:	16#02
Value range:	16#01 … 16#0A

Effects of increased number of read/write attempts:

⊕ More reliable reading and writing

⊖ Longer execution time of a command

Note

To limit the increase in execution time caused by increasing tries allowed, it is recommended to parameterize the NT cancellation criterion. This will stop the command execution as soon as the set number of tags has been identified.

Temperature (TE)

The TE parameter allows you to read the internal temperature of the RFID read/write device.

Parameter abbreviation:	TE (16#5445)
Parameter length:	IQH3 –? 2 bytes (byte 1: Temperature on the power amplifier; byte 2: Temperature on the microcontroller) IQH4 –? 1 byte (temperature on the microcontroller)

Tag ID Filter (TI)

You can use the TI parameter to filter tags using the UID. Read/write commands are executed on tags with a UID that matches the filter. You can set the filter to the entire UID or to the leading bytes.

Parameter abbreviation:	TI (16#5449)
Parameter length:	0 ... 8 bytes

Overtemperature Handling (TO)

The TO parameter defines the behavior of the RFID read/write device in the event of excess temperature. The parameter can be used to protect the device in the event of excess temperature.

Parameter abbreviation	TO (16#544F)
Parameter length:	1 byte
Default:	16#00
Value range:	16#00 = Switch off 16#01 = Reduce Power 16#02 = Reduce Duty Cycle

Value	Meaning	Description
0	Switch off	The RFID read/write device switches off when the internal device temperature is above 85 °C. The device will switch back on when the internal temperature falls below 80 °C.
1	Reduce Power	The RFID read/write device gradually reduces the transmitter radiated power when the internal device temperature is between 80 °C and 85 °C. A reduced transmitter radiated power reduces inherent heating. The RFID read/write device switches off when the internal device temperature is above 85 °C.
2	Reduce Duty Cycle	The RFID read/write device pauses for approx. 0.5 seconds after a complete pass if the internal device temperature is between 80 °C and 85 °C. This reduces inherent heating. The length of the pause varies with the internal temperature and the duration of the executed read/write command The RFID read/write device switches off when the internal device temperature is above 85 °C.

Caution!

The RFID read/write device cannot identify tags during a pause.

It may be possible that data is not read if tags move quickly through the sensing range.

Voltages (VO)

The VO parameter can be used to read out the input voltage and other internal voltage and current values.

Parameter abbreviation:	VO (16#564F)
Parameter length:	14 bytes

6.7.3 Parameter Examples

Read Parameter

Command Telegram

Byte	Content	AF parameter example	CT parameter example	DN parameter example	DR parameter example	DS parameter example	E5 parameter example	IF parameter example	LM parameter example	ND parameter example	NT parameter example	OH parameter example	PT parameter example	QV parameter example
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	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	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	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	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	16#08
5	Command	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE
6	System code	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"
7	Parameter Name (High Byte)	16#41 "A"	16#43 "C"	16#44 "D"	16#44 "D"	16#44 "D"	16#45 "E"	16#49 "I"	16#4C "L"	16#4E "N"	16#4E "N"	16#4F "O"	16#50 "P"	16#51 "Q"
8	Parameter Name (Low Byte)	16#46 "F"	16#54 "T"	16#4E "N"	16#52 "R"	16#53 "S"	16#35 "5"	16#46 "F"	16#4D "M"	16#44 "D"	16#54 "T"	16#48 "H"	16#54 "T"	16#56 "V"
9	Length Parameter (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	16#00
10	Length Parameter (Low 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	16#00
11	Command parameter													
12	Command parameter													
13	Command parameter													
...														
255	Command parameter													

Byte	Content	QW parameter example	RL parameter example	RU parameter example	ST parameter example	T2 parameter example	T3 parameter example	T4 parameter example	T5 parameter example	TA parameter example	TE parameter example	TI parameter example	TO parameter example	VO parameter example
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	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	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	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	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	16#08
5	Command	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE
6	System code	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"
7	Parameter Name (High Byte)	16#51 "Q"	16#52 "R"	16#52 "R"	16#53 "S"	16#54 "T"	16#54 "T"	16#54 "T"	16#54 "T"	16#54 "T"	16#54 "T"	16#54 "T"	16#54 "T"	16#56 "V"
8	Parameter Name (Low Byte)	16#57 "W"	16#4C "L"	16#55 "U"	16#54 "T"	16#32 "2"	16#33 "3"	16#34 "4"	16#35 "5"	16#41 "A"	16#45 "E"	16#49 "I"	16#4F "O"	16#4F "O"
9	Length Parameter (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	16#00
10	Length Parameter (Low 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	16#00
11	Command parameter													
12	Command parameter													
13	Command parameter													
...														
255	Command parameter													

Response Telegram – Status 16#00

Byte	Content	AF parameter example	CT parameter example	DN parameter example	DR parameter example	DS parameter example	E5 parameter example	IF parameter example	LM parameter example	ND parameter example	NT parameter example	OH parameter example	PT parameter example	QV parameter example
0	Control Byte / Frame Length	16#00	16#x0	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#x0	16#00	16#x0	16#x0
1	Frame Length	16#08	16#08	16#15	16#08	16#08	16#08	16#08	16#09	16#08	16#08	16#0F	16#09	16#08

2026-05

PEPPERL+FUCHS

Byte	Content	AF parameter example	CT parameter example	DN parameter example	DR parameter example	DS parameter example	E5 parameter example	IF parameter example	LM parameter example	ND parameter example	NT parameter example	OH parameter example	PT parameter example	QV parameter example
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	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	16#00
4	Telegram Length (Low Byte)	16#05	16#05	16#12	16#05	16#05	16#05	16#05	16#06	16#05	16#05	16#0C	16#06	16#05
5	Command	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE
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	16#00
7	Data	16#00	16#14	16#34	16#00	16#01	16#05	16#01	16#00	16#32	16#FF	16#00	16#00	16#4C
8	Data			16#30					16#03			16#00	16#04	
9	Data			16#30								16#00		
10	Data			16#30								16#11		
11	Data			16#30								16#00		
12	Data			16#31								16#00		
13	Data			16#38								16#00		
14	Data			16#30								16#03		
15	Data			16#36										
16	Data			16#38										
17	Data			16#34										
18	Data			16#34										
19	Data			16#37										
20	Data			16#30										
21	Data													
22	Data													
23	Data													
24	Data													
25	Data													
26	Data													
27	Data													
28	Data													
29	Data													
30	Data													
31	Data													
32	Data													
33	Data													
34	Data													
35	Data													
...														
255	Data													

Byte	Content	QW parameter example	RL parameter example	RU parameter example	ST parameter example	T2 parameter example	T3 parameter example	T4 parameter example	T5 parameter example	TA parameter example	TE parameter example	TI parameter example	TO parameter example	VO parameter example
0	Control Byte / Frame Length	16#x0	16#x0	#	16#00	16#00	16#00	16#00	16#00	16#x0	16#x0	16#x0	16#x0	16#00
1	Frame Length	16#08	16#08	16#08	16#09	16#24	16#0a	16#0a	16#0a	16#08	16#08	16#0B	16#08	16#15
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	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	16#00
4	Telegram Length (Low Byte)	16#05	16#05	16#05	16#06	16#21	16#07	16#07	16#07	16#05	16#05	16#08	16#05	16#12
5	Command	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE	16#BE
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	16#00
7	Data	16#00	16#00	16#50	16#00	16#59	16#2a	16#2a	16#2a	16#02	16#20	16#E0	16#02	16#5D
8	Data				16#00	16#6f	16#2a	16#2a	16#2a			16#08		16#15
9	Data					16#75	16#2a	16#2a	16#2a			16#01		16#14
10	Data					16#72						16#48		16#71
11	Data					16#20								16#0C
12	Data					16#61								16#DF
13	Data					16#75								16#85
14	Data					16#74								16#E8
15	Data					16#6f								16#0C
16	Data					16#6d								16#B9
17	Data					16#61								16#00
18	Data					16#74								16#00
19	Data					16#69								16#02
20	Data					16#6f								16#96
21	Data					16#6e								
22	Data					16#2c								
23	Data					16#20								
24	Data					16#6f								
25	Data					16#75								
26	Data					16#72								
27	Data					16#20								
28	Data					16#70								
29	Data					16#61								
30	Data					16#73								
31	Data					16#73								
32	Data					16#69								
33	Data					16#6f								
34	Data					16#6e								
35	Data					16#2e								
...														
255	Data													

2026-05

Example	
AF parameter example	(16#00 = all tags respond)
CT parameter example	(16#14 = tag type 20)
DN parameter example	(Serial number 40000180684470)
DR parameter example	(16#00 = normal = 26 kBit/s)
DS parameter example	(16#01 = byte-by-byte exchange of blocks)
E5 parameter example	(16#05 = 5 unsuccessful read/write attempts)
IF parameter example	(16#01 = Information Frame Mode activated)
LM parameter example	(16#0003 = "Interference from external signals" and "Detuning due to surrounding metal" are suppressed)
ND parameter example	(16#32 = status display from 50% detection range reduction)
NT parameter example	(16#FF = Off)
OH parameter example	(16#0000001100000003 = 17 hours operating time & 3 hours of active read/write commands)
PT parameter example	(16#04 = Maximum)
QV parameter example	(16#4D = multi-frame protocol)
QW parameter example	(16#00 = one tag)
RL parameter example	(16#00 = lower RSSI threshold of 0)
RU parameter example	(16#50 = upper RSSI threshold of 80_{dec})
ST parameter example	(16#0000 = no interference)
T2 parameter example	(Your Automation, Our Passion.)
T3 parameter example	(***)
T4 parameter example	(***)
T5 parameter example	(***)
TA parameter example	(16#02 = 2 access attempts)
TE parameter example	(IQH4: 16#20 = 32° on microcontroller)
TI parameter example	(Filter by UID starting with 16#E0080148)
TO parameter example	(16#02 = Duty Cycle)
VO parameter example	(16#5D15 = 23829 mV input voltage; 16#14710CDF85E80CB900000296 = other internal voltage and current values)

Write Parameter

Command Telegram

Byte	Content	AF parameter example	BB parameter example	CT parameter example	DR parameter example	DS parameter example	E5 parameter example	IF parameter example	LM parameter example	ND parameter example	NT parameter example	PT parameter example	QV parameter example
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#0C	16#0B	16#0C	16#0C	16#0C	16#0C	16#0C	16#0D	16#0C	16#0C	16#0D	16#0C
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

Byte	Content	AF parameter example	BB parameter example	CT parameter example	DR parameter example	DS parameter example	E5 parameter example	IF parameter example	LM parameter example	ND parameter example	NT parameter example	PT parameter example	QV parameter example
4	Telegram Length (Low Byte)	16#09	16#08	16#09	16#09	16#09	16#09	16#09	16#0A	16#09	16#09	16#0A	16#09
5	Command	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF
6	System code	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"
7	Parameter Name (High Byte)	16#41 "A"	16#42 "B"	16#43 "C"	16#44 "D"	16#44 "D"	16#45 "E"	16#49 "I"	16#4C "L"	16#4E "N"	16#4E "N"	16#50 "P"	16#51 "Q"
8	Parameter Name (Low Byte)	16#46 "F"	16#42 "B"	16#54 "T"	16#52 "R"	16#53 "S"	16#35 "5"	16#46 "F"	16#4D "M"	16#44 "D"	16#54 "T"	16#54 "T"	16#56 "V"
9	Length Parameter (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
10	Length Parameter (Low Byte)	16#01	16#00	16#01	16#01	16#01	16#01	16#01	16#02	16#01	16#01	16#02	16#01
11	Command parameter	16#39		16#15	16#01	16#00	16#05	16#01	16#00	16#32	16#FF	16#00	16#4C
12	Command parameter								16#01			16#01	
13	Command parameter												
14	Command parameter												
15	Command parameter												
16	Command parameter												
17	Command parameter												
18	Command parameter												
19	Command parameter												
20	Command parameter												
21	Command parameter												
22	Command parameter												

2026-05

Byte	Content	AF parameter example	BB parameter example	CT parameter example	DR parameter example	DS parameter example	E5 parameter example	IF parameter example	LM parameter example	ND parameter example	NT parameter example	PT parameter example	QV parameter example
...													
255	Command parameter												

Byte	Content	QW parameter example	RD parameter example	RL parameter example	RU parameter example	T2 parameter example	T3 parameter example	T4 parameter example	T5 parameter example	TA parameter example	TI parameter example	TO parameter example
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#0C	16#0B	16#0C	16#0C	16#16	16#13	16#13	16#14	16#0C	16#0F	16#0C
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#09	16#08	16#09	16#09	16#13	16#10	16#10	16#11	16#09	16#0C	16#09
5	Command	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF
6	System code	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"	16#51 "Q"
7	Parameter Name (High Byte)	16#51 "Q"	16#52 "R"	16#52 "R"	16#52 "R"	16#54 "T"	16#54 "T"	16#54 "T"	16#54 "T"	16#54 "T"	16#54 "T"	16#54 "T"
8	Parameter Name (Low Byte)	16#57 "W"	16#44 "D"	16#55 "L"	16#4C "U"	16#32 "2"	16#33 "3"	16#34 "4"	16#35 "5"	16#41 "A"	16#49 "I"	16#4F "O"
9	Length Parameter (High Byte)	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
10	Length Parameter (Low Byte)	16#01	16#00	16#01	16#01	16#0B	16#08	16#08	16#09	16#01	16#04	16#01
11	Command parameter	16#04		16#14	16#50	16#41	16#46	16#4C	16#49	16#02	16#E0	16#02
12	Command parameter					16#70	16#75	16#6F	16#6E		16#08	
13	Command parameter					16#70	16#6E	16#63	16#73		16#01	
14	Command parameter					16#6C	16#63	16#61	16#74		16#48	
15	Command parameter					16#69	16#74	16#74	16#61			
16	Command parameter					16#63	16#69	16#69	16#6C			
17	Command parameter					16#61	16#6F	16#6F	16#6C			
18	Command parameter					16#74	16#6E	16#6E	16#61			
19	Command parameter					16#69			16#74			

Byte	Content	QW parameter example	RD parameter example	RL parameter example	RU parameter example	T2 parameter example	T3 parameter example	T4 parameter example	T5 parameter example	TA parameter example	TI parameter example	TO parameter example
20	Command parameter					16#6F			16#69			
21	Command parameter					16#6E			16#6F			
22	Command parameter								16#6E			
...												
255	Command parameter											

Example	
AF parameter example	(16#39 = 57_{dec})
BB parameter example	
CT parameter example	(16#15 = tag type 21)
DR parameter example	(16#01 = fast read mode)
DS parameter example	(16#00 = no data exchange)
E5 parameter example	(16#05 = 5 attempts until Status 5)
IF parameter example	(16#01 = Information Frame Mode activated)
LM parameter example	(16#0001 = operating status "Interference from external signals in the usage spectrum" is suppressed)
ND parameter example	(16#32 = threshold 50%)
NT parameter example	(16#FF = "Off")
PT parameter example	(16#0001 = "Min")
QV parameter example	(16#4D = multi-frame protocol)
QW parameter example	(16#04 = "very many tags")
RD parameter example	
RL parameter example	(16#14 = lower RSSI threshold 20_{dec})
RU parameter example	(16#50 = upper RSSI threshold 80_{dec})
T2 parameter example	("Application")
T3 parameter example	("Function")
T4 parameter example	("Location")
T5 parameter example	("Installation")
TA parameter example	(16#02 = 2 attempts)
TI parameter example	(16#E0080148 = UID starting with .E0.08.01.48
TO parameter example	(16#02 = "Reduce Duty Cycle")

Response Telegram – Status 16#00

Byte	Content	AF parameter example	BB parameter example	CT parameter example	DR parameter example	DS parameter example	E5 parameter example	IF parameter example	LM parameter example	ND parameter example	NT parameter example	PT parameter example	QV parameter example
0	Control Byte / Frame Length	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
1	Frame Length	16#07	16#07	16#07	16#07	16#07	16#07	16#07	16#07	16#07	16#07	16#07	16#07
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#04	16#04	16#04	16#04	16#04	16#04	16#04	16#04	16#04	16#04	16#04	16#04
5	Command	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF
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												
...													
255	Data												

Byte	Content	QW parameter example	RD parameter example	RL parameter example	RU parameter example	T2 parameter example	T3 parameter example	T4 parameter example	T5 parameter example	TA parameter example	TI parameter example	TO parameter example
0	Control Byte / Frame Length	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00
1	Frame Length	16#07	16#07	16#07	16#07	16#07	16#07	16#07	16#07	16#07	16#07	16#07
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#04	16#04	16#04	16#04	16#04	16#04	16#04	16#04	16#04	16#04	16#04
5	Command	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF	16#BF
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											
...												
255	Data											

7 Appendix

7.1 Error and Status Messages

Status/error	Description
0	The command has been executed without error
1	Excess temperature
2	Switch-on message, reset has been executed
3	Reserved
4	The command is incorrect or incomplete. The parameter is not in the valid area.
5	No tag in the sensing range.
6	Hardware error, e.g., error during self-test or read/write device defective
7	Internal device fault.
8	Reserved
9	The programmed tag type is not compatible with the read/write device
A	Single-frame protocol: multiple tags in the sensing range Multi-frame protocol: multiple tags in the sensing range with the same UID
B	Output of additional information
C	Reserved
D	Reserved
E	Internal memory overflow; execute reset
F	End of an output in the multi-frame protocol

7.2 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

PEPPERL+FUCHS

/ DOCT-9976