IQT3-FP-IO-V1

IO-Link RFID read/write device 13.56 MHz, ISO 15693

Manual

IO-Link

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.

1600 Enterprise Parkway

Twinsburg, Ohio 44087

USA

Phone: +1 330 425-3555

E-mail: sales@us.pepperl-fuchs.com

Asia Headquarters

Pepperl+Fuchs Pte. Ltd.

P+F Building

18 Ayer Rajah Crescent

Singapore 139942

Phone: +65 6779-9091

E-mail: sales@sg.pepperl-fuchs.com

https://www.pepperl-fuchs.com

1 Introduction 5
1.1 Content of this Document 5
1.2 Target Group, Personnel 5
1.3 Intended Use 5
1.4 Symbols Used 6
1.5 Definitions 6

2 Certificates and approvals 9
2.1 Declaration of Conformity (RE Directive 2014/53/EU) 9
2.2 FCC Information 9
2.3 IC Information 10
2.4 UL Information 10
2.5 Other Country-Specific Approvals 10

3 Product Description 11
3.1 Product Description 11
3.2 Dimensions 11
3.3 Indicators 12
3.4 Electrical Connection 12
3.5 IO-Link Interface Properties 12
3.6 Accessories 13
3.6.1 IO-Link master 13
3.6.2 Read/write tags 13
3.6.3 Connection Cable 13

4 Installation 14
4.1 Storage and Transportation 14
4.2 Unpacking 14
4.3 Mounting and Connection 14
4.4 Minimum Distances 15

5 Commissioning 16
5.1 Operating Modes 16

6 Operation 17
6.1 Read/Write Tags 13.56 MHz ISO15693 17
6.2 Sensing Range 18

6.3 Multiple Tags in Sensing Range 18

7 Easy Mode 19
7.1 Command Overview 19
7.2 Basic Structure of the Process Data 19
7.2.1 Output Process Data (PLC -> Device) 20
7.2.2 Input Process Data (Device -> PLC) 21
7.2.3 Flow Diagrams 25
7.2.4 Timing 29
7.2.5 Interruption of IO-Link Communication 30
7.3 Easy Mode with PACTware 30

8 ExpertMode 38
8.1 Basic Command Process 38
8.2 Legend 38
8.3 Structure of OUTPUT telegram 39
8.4 Structure of INPUT telegram 39
8.5 Handshake Procedure 40
8.6 Overview of Commands 42
8.6.1 Read/Write Commands 43
8.6.2 System Commands 69
8.6.3 HF Configuration Commands 72
8.6.3.1 Basic Command Structure 73
8.6.3.2 Parameter Overview 76
8.7 Error / Status Messages 98

9 Appendix 99
9.1 Fault Repair 99
9.2 ASCII table 100

PEPPERL+FUCHS

1 Introduction

1.1 Content of this Document

This document contains information required to use the product in the relevant phases of the product life cycle. This may include information on 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.

The documentation comprises the following parts:

- This document
- Datasheet
- IO-Link parameter datasheet

The documentation may also comprise the following parts, if applicable:

- EU-type examination certificate
- EU declaration of conformity
- Attestation of conformity
- Certificates
- Control drawings
- Instruction manual
- Functional safety handbook
- 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.

Action

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

1.5 Definitions

Data is displayed in various ways in the following documentation. The following formats are used:

ASCII

For example: "A"; "B"; "1"; "2"

Each ASCII character corresponds to one byte. 2^8= 256 different characters can be displayed.

Binary

For example: 1001_{bin}

Binary numbers are identified by a $_{bin}$.

DECIMAL

For example: 1234

Decimal numbers are shown without additional identification.

HEX

For example: 0x41; 0x42; 0x31; 0x32

The hexadecimal representation of a byte consists of two digits (e.g., 0x41). The section on the left side is the higher nibble (0x4). The lower nibble is on the right side (0x1). The range of values is between 0x00 … 0xFF.

Note

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

For example:

16#00 (equivalent to 0x00)

Terms

COM Mode	The IO-Link data transfer rate
Read/write tags	Mobile data memory with user data and unique number
Device ID	Identification number of the device
Easy Mode	Communication protocol for simple data access of a read/write device; no function block required
Expert Mode	Communication protocol for high-performance data access of the RFID read/write device; the use of a function block is required
Read-only code	Unique and unchangeable number of a read/write tag; Unique Identifier UID
IODD	IO Device Description; file with information about IO-Link parameters of an IO-Link-enabled device
IO-Link	Communication system for the connection of intelligent sensors and actuators via point-to-point communication
IO-Link master	Interface to higher-level control; controls the communication to connected IO-Link devices
IO-Link device	Intelligent sensor or actuator for connecting to an IO-Link master; has device-specific IO-Link parameters
IO-Link parameters	Device-specific information about an IO-Link device; parameters are stored in an IODD; acyclic change of the parameters
IO-Link protocol	Version of the supported IO-Link communication; V1.0 or V1.1
IQC	Pepperl+Fuchs-specific designation of a 13.56 MHz read/write tag
ISO/IEC 15693	Standard for data transfer for a 13.56 MHz RFID system
ISDU	Indexed Service Data Unit
PACTware	Parameterization software for access to IO-Link parameters
Port type	Type of IO-Link port
RSSI	Indicator for the receive field strength
SIO Mode	Standard IO mode; mode for conventional signal transmission without IO-Link data[1]
PLC	Programmable logic controller; device for controlling a machine or plant
UID	See read-only code
Vendor ID	Identification number of the device manufacturer, Pepperl+Fuchs: 0x01

1. Not supported

Abbreviations

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
Tag	Read/write tag; tag
UID	**U**nique **I**dentifier

2 Certificates and approvals

2.1 Declaration of Conformity (RE Directive 2014/53/EU)

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.

The product manufacturer, Pepperl+Fuchs SE, 68307 Mannheim, Germany, has a certified quality assurance system that conforms to ISO 9001.

ISO**9001**

2.2 FCC Information

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.

FCC Notice

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.

FCC Exposure Information

To comply with FCC RF exposure compliance requirements, the antennas used for this transmitter must be installed to provide a separation distance of at least 20 cm from all persons and must not be co-located or operated in conjunction with any other antenna or transmitter.

2.3 IC Information

This device complies with Industry Canada license-exempt RSS standard(s) and with part 15 of the FCC Rules. Operation is subject to the following two conditions:

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

Le présent appareil est conforme aux CNR d'Industrie 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, et
2. l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

IC Exposure Information

To comply with IC RF exposure compliance requirements, the antennas used for this transmitter must be installed to provide a separation distance of at least 20 cm from all persons and must not be co-located or operated in conjunction with any other antenna or transmitter.

2.4 UL Information

Technical Data and Environmental Conditions

This device is for indoor use only.

This device may be operated in altitudes up to 5000 m.

The ambient temperature range is from -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 % for 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 18 V ... 30 V_{DC}. Supply must be LEC (Limited Energy Circuit), LPS (Limited Power Source) or CLASS 2. The Overvoltage Category II is applied.

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.

Protection class IP67 is not included in the UL approval. The protection class is tested by Pepperl+Fuchs SE.

The ext. circuits are intended to be connected to this unit shall be galv. separated from mains supply or hazardous live voltage by reinforced or double insulation and meet the limits of clauses 6.3 and 9.4 of UL 61010-1.

2.5 Other Country-Specific Approvals

All currently valid approvals can be found in the datasheet for your device at www.pepperl-fuchs.com.

3 Product Description

3.1 Product Description

Use and Application

This device is an RFID read/write device with an IO-Link communication interface (IO-Link device). The device supports tags in the frequency range of 13.56 MHz according to the ISO15693 standard and is connected to an IO-Link master via an integrated M12 plug.

Figure 3.1

3.2 Dimensions

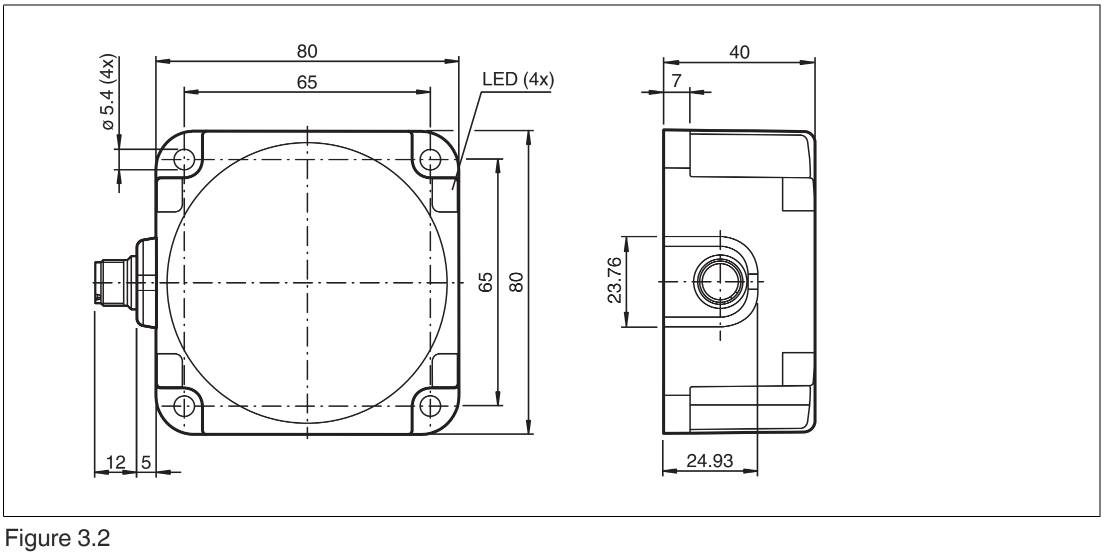


Figure 3.2

3.3 Indicators

The RFID read/write device has LEDs to indicate the operating state. The various indicators denote:

LED	Description
Green	On: Device is ready for operation, no IO-Link communication Flashing: IO-Link communication active
Blue	On: Write/read attempt is being performed
Yellow	On: Write/read attempt was successful
Red	On: Interference from external signals in the sensing range, range may be limited. Slow flashing (1 Hz): Metal interference Fast flashing (10 Hz): Excess temperature

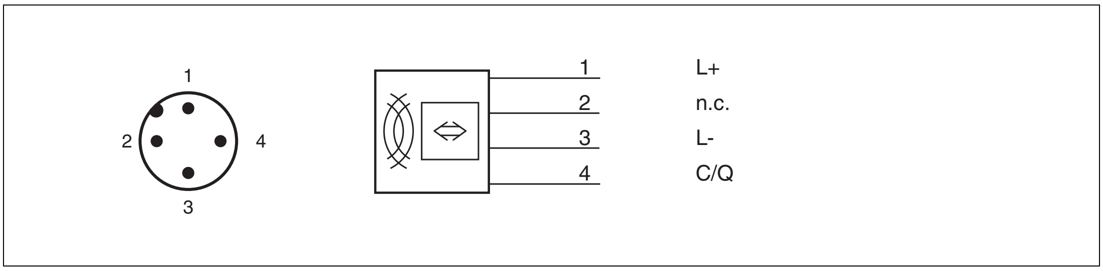
3.4 Electrical Connection

Caution!

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 mm².



Pin 1	L+	+24 V
Pin 2	n.c.	Not connected
Pin 3	L-	0 V/GND
Pin 4	C/Q	C/Q

The RFID read/write device is connected to an IO-Link master via a point-to-point connection. According to the IO-Link installation instructions, the length of the connection line should not exceed 20 meters. The RFID read/write device is supplied with power via the IO-Link master. For technical details, refer to the product datasheet.

3.5 IO-Link Interface Properties

IO-Link version:	1.1
Data transfer rate	COM3 (230.4 kbits/s)
Min. cycle time	4 ms
Process data	Input 32 bytes Output 32 bytes
SIO mode support	No
Compatible master port type	Class A Class B
Device ID	4195073 (0x400301)
Vendor ID	1 (0x0001)

3.6 Accessories

3.6.1 IO-Link master

You can connect the RFID read/write device to any IO-Link master, as long as it supports IO-Link standard V1.1 and has a sufficient power supply. Below are some of the available IO-Link masters from Pepperl+Fuchs:

Model number	Description
IO-Link-Master02-USB	IO-Link master with USB interface for connecting to higher-level devices (for example PC)
ICE1-8IOL-G60L-V1D	Ethernet IO-Link module with 8 IO-Link ports for connection to a higher-level system via PROFINET and EtherNet/IP
ICE1-8IOL-G30L-V1D	Ethernet IO-Link module with 8 IO-Link ports for connection to a higher-level system via PROFINET and EtherNet/IP
ICE1-8IOL-S2-G60L-V1D	EtherNet/IP IO-Link master with PROFINET S2 redundancy
ICE2-8IOL-G65L-V1D	EtherNet/IP IO-Link master with 8 inputs/outputs
ICE3-8IOL-G65L-V1D	PROFINET/IO IO-Link master with 8 inputs/outputs

Table 3.1

Note

Check the performance of the power supply at the connector of the master and the cable cross-section of the connecting cable to ensure stable communication.

3.6.2 Read/write tags

The RFID read/write device can access any read/write tag that is compliant with the standard ISO15693. For an overview of possible read/write tags . The following read/write tags from Pepperl+Fuchs can be used, for example:

- IQC21-16 50pcs
- IQC21-30 25pcs
- IQC21-50 25pcs
- IQC33-30 25pcs
- IQC33-50 25pcs
- IQC22-C1 10pcs

3.6.3 Connection Cable

To connect the RFID read/write device to an IO-Link master, you can use unshielded, 3- or 4-wire cables with an M12 plug and a maximum length of 20 m. For example, you can use the following connection cables from Pepperl+Fuchs:

- V1-G-2M-PUR-V1-W
- V1-G-5M-PUR-V1-W
- V1-G-10M-PUR-V1-W
- V1-G-20M-PUR-V1-W

You can find further suitable accessories on our website http://www.pepperl-fuchs.com.

Note

Enter the order designation of your RFID read/write device in the product search. The product detail page contains a list of the related products.

4 Installation

4.1 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.

4.2 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.

4.3 Mounting and Connection

Note

The RFID read/write device is intended for wall mounting and mounting on brackets in indoor spaces.

Mount the device on a flat surface and secure it only with the holes provided on the housing.

Warning!

Processes started in an uncontrolled manner jeopardize the plant

Before commissioning, ensure that there are no risks involved in using the device that may endanger the plant.

Caution!

Hot surfaces

Risk of burns when handling the read/write device! Allow the device to cool for at least half an hour after it has been switched off before touching it.

To attach the read/write device, use 4 screws with a diameter of 4 mm, suitable washers and mounting materials that are 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.

Note

Do not route the connection cable of the read/write device and the IO-Link master into the raceway area of the read/write tag. Keep a distance of at least 10 cm.

If the distance is too small, it can occur in rare cases that the read/write tag is incorrectly detected by inductive couplings on the connection cable.

4.4 Minimum Distances

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 transmit power accordingly. Determine the sensing range of each device at the mounting location.

When mounting multiple read/write devices, keep a **minimum distance of 750 mm**.

5 Commissioning

5.1 Operating Modes

The device supports two operating modes:

- **Easy mode**
  Easy mode enables simplified commissioning with a limited range of functions. This is the preferred operating mode for standard applications.
- **Expert mode**
  The entire set of commands is available in Expert mode. To use Expert mode, a function block is required to integrate into the PLC.

6 Operation

6.1 Read/Write Tags 13.56 MHz ISO15693

The read/write tags of an RFID system with 13.56 MHz offer significantly quicker access to the data than a comparable RFID system based on an operating frequency of 125 kHz. The 13.56 MHz system is standardized through ISO15693. A great variety of read/write 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 the read/write tag type 20 is preset. This setting guarantees access to the read-only code of ISO15693-compliant read/write tags.

The following table shows the read/write tag types specified and recommend for the RFID read/write device.

13.56 MHz/ISO 15693 Tag Types

Pepperl+ Fuchs designation	Chip type	Manufacturer	Length of read-only code [byte]	Size of read/write memory [byte]	Size of memory block [byte]
IQC20	All read/write tags in accordance with ISO15693	-	8	Depending on the read/write tags	Depending on the read/write tags
IQC21	I-Code SLI(X)	NXP	8	112	4
IQC22	Tag-it HF-I Plus	Texas Instruments	8	256	4
IQC23	my-D SRF55V02P	Infineon	8	224	4
IQC24	my-D SRF55V10P	Infineon	8	992	4
IQC27[1]	EM4135	EM Microelectronic	8	288	8
IQC31	Tag-it HFI standard	Texas Instruments	8	32	4
IQC32	Tag-it HFI pro	Texas Instruments	8	32	4
IQC33[1]	FRAM MB89R118	Fujitsu	8	2000	8
IQC34	FRAM MB89R119	Fujitsu	8	232	4
IQC35	I-Code SLI-S	NXP	8	160	4
IQC36	I-Code SLI-L	NXP	8	32	4
IQC37[2]	FRAM MB89R112	Fujitsu	8	8192	32
IQC38	EM4233	EM Microelectronic	8	208	4
IQC50	I-Code SLIX2	NXP	8	316	4

Table 6.1 13.56 MHz tag types in accordance with ISO 15693

1. Exception: Block size = 8 bytes
'Number of bytes' must be a multiple of 8
2. Exception: Block size = 32 bytes
'Number of bytes' must be a multiple of 32

All ISO15693-compliant read/write 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 read/write tags have a memory area for user data. This can be written with application-specific data and read. The size of the memory for the user data differs according to the read/write tag type.

The memory area is divided into blocks with a length of 4 bytes. There are also exceptions with a block length of 8 bytes or 32 bytes. As a result of the block length of 32 bytes, Easy Mode does not support the tag type IQC37.

The read and write commands use the "Number bytes" and "Start address" parameters. This defines how many bytes are accessed in the memory area 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 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.

6.2 Sensing Range

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 transmit 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 transmit power PT > 1. Otherwise, there is a risk that a tag will be damaged in the close range. Reduce the transmit power of the read/write device to PT = 1 to meet the limit value at small minimum distances.

6.3 Multiple Tags in Sensing Range

The behavior when identifying tags in the sensing range depends on the operating mode selected.

Easy mode

All detected read/write tags are transmitted. Filtering for individual tags is not possible.

Expert mode

Each read or write command can access one, several, or all tags in a sensing range. Filter masks that are managed with the **TI** parameter are used for control. These commands allow you to detect specific tags in the sensing range. See "Setting a Tag ID Filter (TI) Filter" on page 94.

Note

For multitag applications, use the same tag types.

For multi-tag operation, the IO-Link parameter "Input Representation" must be set to "Long-Form".

2023-11

PEPPERL+FUCHS

7 Easy Mode

The RFID read/write device uses the "Easy Mode" communication protocol on the basis of IO-Link for data transfer to a higher-level system. If this protocol is used, the RFID read/write device can be commissioned without an additional function block on a control system. This makes it easier to commission the read/write device.

When "Easy Mode" is used, there is a distinction between parameter and process data. The parameter data is IO-Link parameters being transferred acyclically. This is data for the configuration of the read/write jobs, for parameterizing the device properties, e.g., transmit power and service data, e.g., operating hours meter. The process data is transmitted in cycles. The process data is divided into input and output data. It has a length of 32 bytes and contains the control values for the execution of the read and write commands and the associated values.

The IO-Link parameters for setting the RFID read/write device are defined by a device-specific IODD file. The IO-Link parameters are set using suitable configuration software. During this process, the IO-Link parameters are saved in a non-volatile memory in the RFID read/write device.

7.1 Command Overview

The Easy Mode supports the following read and write commands:

Command	Description
Read the read-only code (UID)	Read-out of the 8 byte read-only code; no setting of the "Number of bytes" or "Start Address" parameters required
Read the user memory	Read-out of user data; setting of the "Number of bytes" and "Start Address" parameters is required
Write the user memory	Writing of user data; setting of the "Number of bytes" and "Start Address" parameters is required
Auto-start	Automatic execution of a read task for read-only code or user data after device start

7.2 Basic Structure of the Process Data

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

A detailed description of the process data can be found in the IODD or on the IO-Link parameter datasheet of the device. You can find the IODD and the parameter datasheet on the product details page at www.pepperl-fuchs.com.

The input and output data of the process data have a fixed length of 32 bytes.

Note

Output process data is ignored by the device if the "valid" flag is not set. An unset "valid" flag indicates invalid process data.

7.2.1 Output Process Data (PLC -> Device)

Byte	Content							
0	0	0	0	0	0	0	Start write	Start read
1	0x00							
2	0x00							
3	0x00							
4	User data for write job or 16#00 for read job							
etc.	User data for write job or 16#00 for read job							
31	User data for write job or 16#00 for read job							

Table 7.1

Byte 0 This byte contains the control bits for starting a read job or write job. The control bits will have no effect if the auto-start function is activated.
Start Read: As soon as this bit is set (TRUE), a read job is started using the configuration set by the parameter "Read Task" in the IODD file. The read job runs continuously. To cancel the read job, reset the bit (FALSE).
Start Write: As soon as this bit is set (TRUE), a write job is started. The write job transfers the user data, which must be stored starting from byte 4, according to the configuration set by the parameter "Write Task" in the IODD file. The write job runs continuously. To cancel the write job, reset the bit (FALSE).
Note that both bits cannot be set simultaneously. The remaining bits have no significance.

Byte 1/2/3 These bytes are not used in Easy Mode. Set the value 0x00.

Byte 4 … 31 When a read job is being executed, these bytes have no significance and are set with the value 0x00. When a write job is being executed, the user data to be written to the read/write tag is stored in this area.

For example: Writing user memory in TIA Portal

Name	Address	Displa...	Monitor value
"ControlByte_Out"	%QB0	Bin	2#0000_0010
"Unused_1"	%QB1	Hex	16#00
"Unused_2"	%QB2	Hex	16#00
"Unused_3"	%QB3	Hex	16#00
"WriteData_1"	%QB4	Hex	16#01
"WriteData_2"	%QB5	Hex	16#02
"WriteData_3"	%QB6	Hex	16#03
"WriteData_4"	%QB7	Hex	16#04
"WriteData_5"	%QB8	Hex	16#05
"WriteData_6"	%QB9	Hex	16#06
"WriteData_7"	%QB10	Hex	16#07
"WriteData_8"	%QB11	Hex	16#08

Output process data

Byte 0:	0000 0010_{bin}; the "Start write" bit is set to 1; a write job is executed.
Byte 1/2/3:	16#00; not used
Byte 4 … Byte 11:	Write data

2023-11

PEPPERL+FUCHS

7.2.2 Input Process Data (Device -> PLC)

Byte	Content							
0	0	0	0	Tag present	Error	Job active	Write successful	Read successful
1	Length specification							
2	RSSI value							
3	Reserved							
4	Data for read/write job / error code							
5	Data for read/write job / error information							
etc.	Data for read/write job / error information							
31	Data for read/write job / error information							

Table 7.2

Byte 0 This byte contains the control bits to indicate the status of executing the read or write job.
Read successful: This bit indicates successful reading of data from a read/write tag. This bit is set if a read/write tag enters the sensing range and the data has been read successfully. The bit remains set while the read/write tag is within the sensing range. As soon as the read/write tag has left this area, the bit is reset again. A positive edge change is triggered if there are multiple read/write tags in the sensing range at the same time[1] Signals the transfer of another read/write tag.
Write successful: This bit indicates successful writing of data to a read/write tag. This bit is set if a read/write tag enters the sensing range and the data has been written successfully. The bit remains set while the read/write tag is within the sensing range. As soon as the read/write tag has left this area, the bit is reset again. If several read/write tags are within the sensing range at the same time, a positive edge change indicates that another read/write tag has been successfully written.
Job active: This bit is set while the read or write job is being executed. As soon as the job is finished, this bit is reset again.
Error: If an error occurs during execution of a read or write job or if a parameter has not been set correctly, the error bit is set. At the same time, additional error information in the form of an error code is located in the process data.
Tag present: Bit is set if one or more read/write tags are in the sensing range. If there is no read/write tag in the sensing range, this bit has the value FALSE.

Byte 1 This byte contains the number of transferred bytes. If a read/write tag enters the sensing range and the data has been read successfully (Read successful = TRUE), this byte indicates the length of the read data. If an error occurs during execution of a job (Error = TRUE), the byte contains a length specification for the error information.

Byte 2 The RSSI value indicates the signal strength of the tag response as a percentage. The value range is between 0 and 100. 0 % = weak signal, 100 % = strong signal.

Byte 3 Reserved.

Byte 4/5 If a read/write tag is successfully accessed, the length of the read-only code UID is located at this position.[2]

Byte 6 … Byte 31 Starting at byte 6, the UID of the read/write tag from which the information was read or to which the information was written. The UID is always included in the response to a read or write job. This information ensures unique assignment of the read/write tag. When a user memory read job is performed, the length specification (2 bytes) of the read-in user memory and the read-in subarea of the user memory are connected.

1. Positive edge change: Change from 0 to 1

2. Applies only to the long-form data format. In the short-form data format, the UID or the user memory of the tag is transferred directly from byte 4.

If an error occurs during execution of a read or write job ("Error" bit = TRUE), byte 4 contains an error code. In the fault state, an error message is transmitted in plain text (ASCII) starting at byte 5. This provides a possible cause of the fault.

For example: Read read-only code (UID), long-form data format

Name	Address	Displ...	Monitor value
"ControlByte_In"	%IB0	Bin	2#0001_0101
"Length"	%IB1	DEC	10
"RSSI"	%IB2	DEC	97
"Reserved"	%IB3	DEC	0
"Length_UID_HighByte"	%IB4	Hex	16#00
"Length_UID_LowByte"	%IB5	Hex	16#08
"UID_1"	%IB6	Hex	16#E0
"UID_2"	%IB7	Hex	16#04
"UID_3"	%IB8	Hex	16#01
"UID_4"	%IB9	Hex	16#50
"UID_5"	%IB10	Hex	16#D3
"UID_6"	%IB11	Hex	16#23
"UID_7"	%IB12	Hex	16#74
"UID_8"	%IB13	Hex	16#BA

Byte 0: 0001 0101$_{bin}$; the "Job active" bit is set to 1, which indicates that a read job is activated. The "Read successful" bit is set to 1. A read/write tag is therefore located in the sensing range and the data has been read. The "Tag present" bit is also set because there is at least one tag in the sensing zone.

Byte 1: 10; the byte indicates the length of the transmitted information. This telegram is used to transmit information with a length of 10 bytes. The information starts at byte 4.

Byte 2: 97; RSSI value

Byte 3: 0; Reserved

Byte 4/5: 16#0008; length specification of the UID. The information has a length of 2 bytes. The length of the UID is always 8 bytes.

Byte 8 … Byte 13: UID of the read/write tag that was read

For example: Read read-only (UID), short-form data format

Name	Address	Displ...	Monitor value
"ControlByte_In"	%IB0	Bin	2#0001_0101
"Length"	%IB1	DEC	8
"RSSI"	%IB2	DEC	98
"Reserved"	%IB3	Hex	16#00
"ReadData_1"	%IB4	Hex	16#E0
"ReadData_2"	%IB5	Hex	16#04
"ReadData_3"	%IB6	Hex	16#01
"ReadData_4"	%IB7	Hex	16#50
"ReadData_5"	%IB8	Hex	16#D3
"ReadData_6"	%IB9	Hex	16#23
"ReadData_7"	%IB10	Hex	16#74
"ReadData_8"	%IB11	Hex	16#BA

Byte 0: 0001 0101$_{bin}$; the "Job active" bit is set to 1, which indicates that a read job is activated. The "Read successful" bit is set to 1. A read/write tag is therefore located in the sensing range and the data has been read. The "Tag present" bit is also set because there is at least one tag in the sensing zone.

Byte 1: 8; the byte indicates the length of the transmitted information. This telegram is used to transmit information with a length of 8 bytes. The information starts at byte 4.

Byte 2: 98; RSSI value

Byte 3: 16#00; reserved

Byte 4 … Byte 11: UID of the read/write tag that was read

For example: Read user memory, long form data format

Name	Address	Displ...	Monitor value
"ControlByte_In"	%IB0	Bin	2#0001_0101
"Length"	%IB1	DEC	20
"RSSI"	%IB2	DEC	97
"Reserved"	%IB3	DEC	0
"Length_UID_HighByte"	%IB4	Hex	16#00
"Length_UID_LowByte"	%IB5	Hex	16#08
"UID_1"	%IB6	Hex	16#E0
"UID_2"	%IB7	Hex	16#04
"UID_3"	%IB8	Hex	16#01
"UID_4"	%IB9	Hex	16#50
"UID_5"	%IB10	Hex	16#D3
"UID_6"	%IB11	Hex	16#23
"UID_7"	%IB12	Hex	16#74
"UID_8"	%IB13	Hex	16#BA
"Length_Data_HighByte"	%IB14	Hex	16#00
"Length_Data_LowByte"	%IB15	Hex	16#08
"ReadData_1"	%IB16	Hex	16#01
"ReadData_2"	%IB17	Hex	16#02
"ReadData_3"	%IB18	Hex	16#03
"ReadData_4"	%IB19	Hex	16#04
"ReadData_5"	%IB20	Hex	16#05
"ReadData_6"	%IB21	Hex	16#06
"ReadData_7"	%IB22	Hex	16#07
"ReadData_8"	%IB23	Hex	16#08

Byte 0: 0001 0101$_{bin}$; The "Job active" bit is set to 1 and signals an active read job. The "Read successful" bit is set to 1. A read/write tag is located in the sensing range and the data has been read. The "Tag present" bit is also set because there is at least one tag in the sensing zone.

Byte 1: 20; the byte indicates the length of the transmitted information. This telegram is used to transmit information with a length of 20 bytes. The information starts at byte 4.

Byte 2: 97; RSSI value

Byte 3: 0; Reserved

Byte 4/5: 16#0008; length specification of the UID. The information has a length of 2 bytes. The length of the UID is always 8 bytes.

Byte 6 … Byte 13: UID of the identified read/write tag

Byte 14/15: 16#0008; length specification for the read-in user memory

Byte 16 … Byte 23: User memory section read in

For example: Read user memory, short-form data format

Name	Address	Displ...	Monitor value
"ControlByte_In"	%IB0	Bin	2#0001_0101
"Length"	%IB1	DEC	8
"RSSI"	%IB2	DEC	97
"Reserved"	%IB3	Hex	16#00
"ReadData_1"	%IB4	Hex	16#01
"ReadData_2"	%IB5	Hex	16#02
"ReadData_3"	%IB6	Hex	16#03
"ReadData_4"	%IB7	Hex	16#04
"ReadData_5"	%IB8	Hex	16#05
"ReadData_6"	%IB9	Hex	16#06
"ReadData_7"	%IB10	Hex	16#07
"ReadData_8"	%IB11	Hex	16#08

Byte 0: 0001 0101$_{bin}$; The "Job active" bit is set to 1 and signals an active read job. The "Read successful" bit is set to 1. A read/write tag is located in the sensing range and the data has been read. The "Tag present" bit is also set because there is at least one tag in the sensing zone.

Byte 1: 8; the byte indicates the length of the transmitted information. This telegram is used to transmit information with a length of 8 bytes. The information starts at byte 4.

Byte 2: 97; RSSI value

Byte 3: 16#00; reserved

Byte 4 … Byte 11: User memory section read in

For example: Write user memory, long form data format

Name	Address	Displ...	Monitor value
"ControlByte_In"	%IB0	Bin	2#0001_0110
"Length"	%IB1	DEC	10
"RSSI"	%IB2	DEC	98
"Reserved"	%IB3	DEC	0
"Length_UID_HighByte"	%IB4	Hex	16#00
"Length_UID_LowByte"	%IB5	Hex	16#08
"UID_1"	%IB6	Hex	16#E0
"UID_2"	%IB7	Hex	16#04
"UID_3"	%IB8	Hex	16#01
"UID_4"	%IB9	Hex	16#50
"UID_5"	%IB10	Hex	16#D3
"UID_6"	%IB11	Hex	16#23
"UID_7"	%IB12	Hex	16#74
"UID_8"	%IB13	Hex	16#BA

Byte 0: 0001 0110_{bin}; The "Job active" bit is set to 1 and signals an active write job. The "Write successful" bit is set to 1. A read/write tag is located in the sensing range and the data has been written successfully. The "Tag present" bit is also set because there is at least one tag in the sensing range.

Byte 1: 10; the byte indicates the length of the transmitted information. This telegram is used to transmit information with a length of 10 bytes. The information starts at byte 4.

Byte 2: 98; RSSI value

Byte 3: 0; Reserved

Byte 4/5: 16#0008; length specification of the UID. The information has a length of 2 bytes. The length of the UID is always 8 bytes.

Byte 6 … Byte 13: UID of the read/write tag to which the data was written.

For example: Write user memory, short-form data format

Name	Address	Displ...	Monitor value
"ControlByte_In"	%IB0	Bin	2#0001_0110
"Length"	%IB1	DEC	08
"RSSI"	%IB2	DEC	98
"Reserved"	%IB3	DEC	0
"UID_1"	%IB4	Hex	16#E0
"UID_2"	%IB5	Hex	16#04
"UID_3"	%IB6	Hex	16#01
"UID_4"	%IB7	Hex	16#50
"UID_5"	%IB8	Hex	16#D3
"UID_6"	%IB9	Hex	16#23
"UID_7"	%IB10	Hex	16#74
"UID_8"	%IB11	Hex	16#BA

Byte 0: 0001 0110_{bin}; The "Job active" bit is set to 1 and signals an active write job. The "Write successful" bit is set to 1. A read/write tag is located in the sensing range and the data has been written successfully. The "Tag present" bit is also set because there is at least one tag in the sensing zone.

Byte 1: 8; the byte indicates the length of the transmitted information. This telegram is used to transmit information with a length of 8 bytes. The information starts at byte 4.

Byte 2: 98; RSSI value

Byte 3: 0; Reserved

Byte 4 … Byte 11: UID of the read/write tag to which the data was written.

7.2.3 Flow Diagrams

Read Job without Auto-Start Function

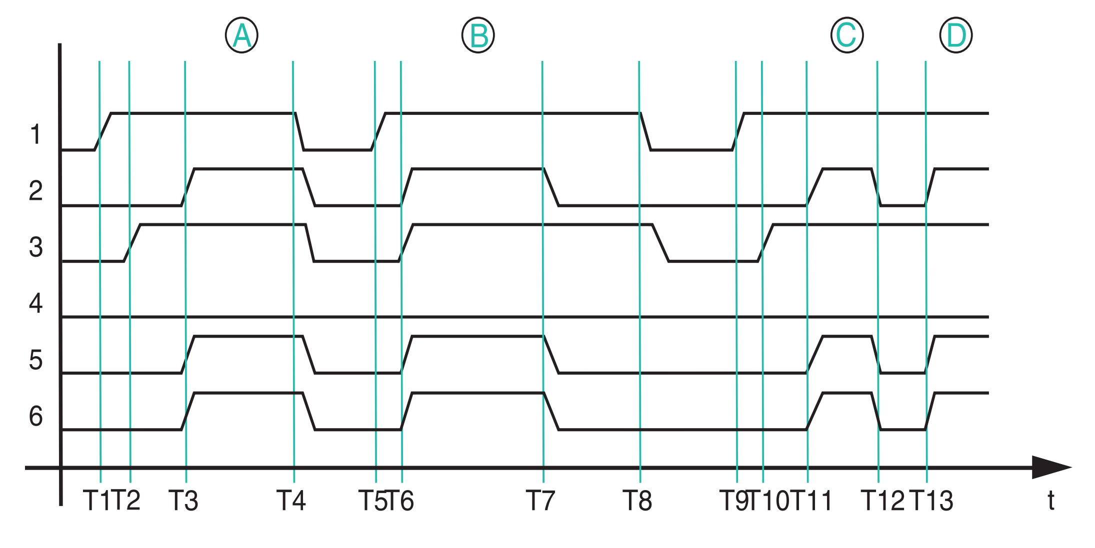


Figure 7.1 Timing sequence of bits in byte 0

1 Start read
2 Read successful
3 Job active
4 Error
5 Tag present
6 Data (input)

If the auto-start function is not used, the read job is started with the "Start read" bit. The read job is performed until the "Start read" bit is reset to FALSE.

T1: Starting the read job by setting the "Start read" bit to TRUE

T2: Read job is executed and indicated through the "Job active" bit ("Job active" = TRUE)

T3: Read/write tag A enters the sensing range; "Read successful" and "Tag present" are set to TRUE and the read data is located in the input field of the process data

T4: Read job is canceled by resetting the "Start read" bit to FALSE while the read/write tag is located in the sensing range; the "Job active" bit, the "Read successful" bit and the "Tag present" are set to FALSE and the process data is filled with 0x00

T5: Start the read job by setting the "Start read" bit to TRUE; at the time of the start, a read/write tag B is already located in the sensing range

T6: Read job is being executed ("Job active" = TRUE) and the data is read successfully ("Read successful" and "Tag present" = TRUE); the read data is located in the input field of the process data

T7: Read/write tag leaves the sensing range ("Read successful" and "Tag present" = FALSE); the area of the input field with the read process data is set to the value 0x00

T8: Cancellation of the read job ("Start read" = FALSE); the "Job active" bit is reset

T9: Start the read job by setting the "Start read" bit to TRUE; at the time of the start, no read/write tag is located in the sensing range; read job remains permanently active

T10: Read job is being executed ("Job active" = TRUE)

T11: T11: Read/write tag C enters the sensing range and the data is read ("Read successful" and "Tag present" = TRUE); read data is located in the input field of the process data

T12: Read/write tag C leaves the sensing range ("Read successful" and "Tag present" = FALSE)

T13: Read/write tag D enters the sensing range

Read Job with Auto-Start Function

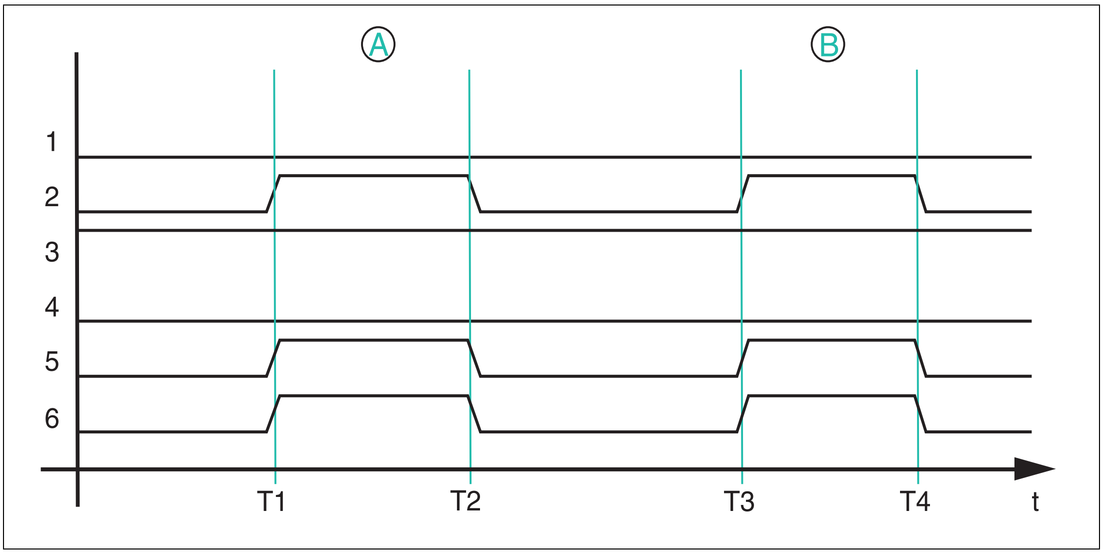


Figure 7.2 Timing sequence of bits in byte 0

1 Start read

2 Read successful

3 Job active

4 Error

5 Tag present

6 Data (input)

If the auto-start function is used, read access runs continuously; the "Job active" bit of the input process data is set permanently.

T1: Read/write tag A enters the sensing range; "Read successful" and "Tag present" are set to TRUE and the read data is located in the input field of the process data

T2: Read/write tag A leaves the sensing range; "Read successful" is reset to FALSE; the area with the previously read data is filled with 0x00

T3: Read/write tag B enters the sensing range; behavior same as T1

T4: Read/write tag B leaves the sensing range; behavior same as T2

Write Job

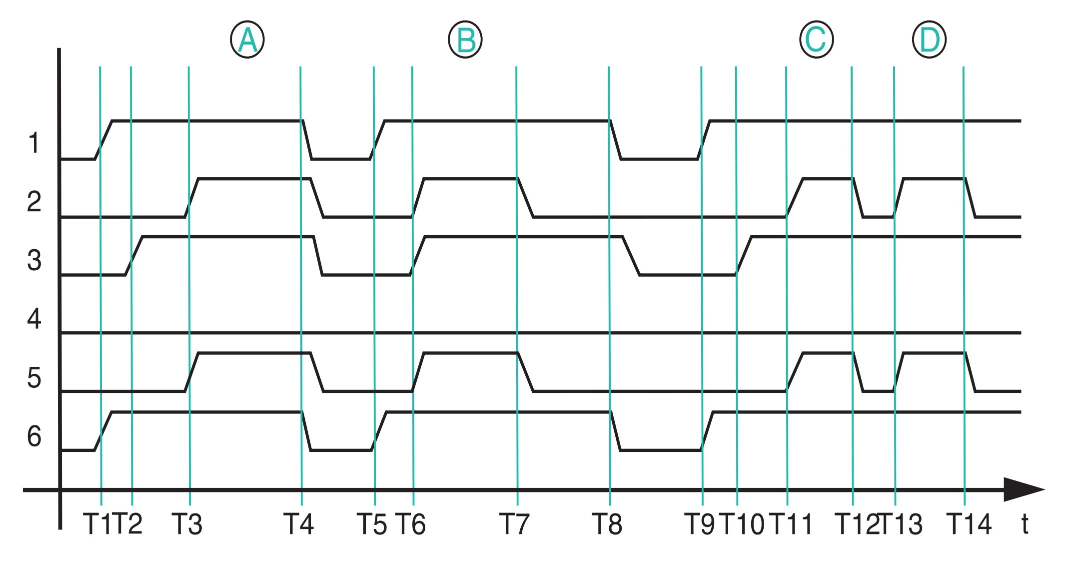


Figure 7.3 Timing sequence of bits in byte 0

1 Start write
2 Write successful
3 Job active
4 Error
5 Tag present
6 Data (input)

A write job cannot be executed using the auto-start function. To start a write job, set the "Start write" bit to TRUE.

T1: Start the write job by setting the "Start write" bit to TRUE; at the same time, the usable data to be written to the read/write tag is transmitted to the output field of the process data

T2: Write job is active ("Job active" = TRUE), and no read/write tag is located in the sensing range ("Write successful" = FALSE)

T3: Read/write tag A enters the sensing range and the data is written successfully ("Write successful" and "Tag present" = TRUE)

T4: The write job is canceled by resetting the "Start write" bit to FALSE; the "Job active," "Write successful" and "Tag present" bits are reset to FALSE and the usable data is reset by the user to the value 0x00

T5: The write job is started by setting the "Start write" bit to TRUE and at the same time transferring the data to be written to the output field of the process data; at the time of the start, read/write tag B is located in the sensing range

T6: The write job is active ("Job active" = TRUE) and read/write tag B is written successfully ("Write successful" and "Tag present" = TRUE)

T7: Read/write tag B leaves the sensing range ("Write successful" and "Tag present" = FALSE); the write job remains active ("Job active" = TRUE)

T8: The write job is canceled by resetting the "Start write" bit to FALSE; the "Job active," "Write successful" and "Tag present" bits are reset to FALSE and the usable data is reset by the user to the value 0x00

T9: The write job is started by setting the "Start write" bit to TRUE; at the same time, the usable data to be written on the read/write tag is transmitted to the output of the process data

T10: Write job is active ("Job active" = TRUE), and no read/write tag is located in the sensing range ("Write successful" = FALSE)

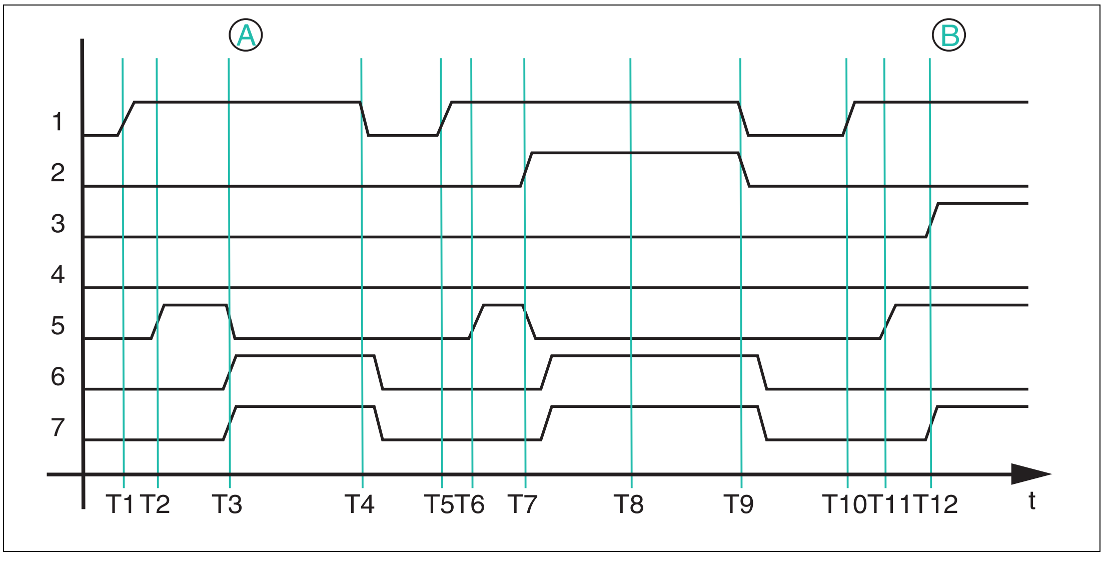
T11: Read/write tag C enters the sensing range and the data is successfully written ("Write successful" and "Tag present" = TRUE); write job is still active ("Job active" = TRUE)

T12: Read/write tag C leaves the sensing range ("Write successful" and "Tag present" = FALSE); write job is still active ("Job active" = TRUE)

T13: Read/write tag D enters the sensing range and the data is successfully written ("Write successful" and "Tag present" = TRUE); write job is still active ("Job active" = TRUE)

T14: Read/write tag D leaves the sensing zone ("Read successful" and "Tag present" = FALSE)

Fault State



1 Start read
2 Start write
3 Read successful
4 Write successful
5 Job active
6 Error
7 Data (input)

If an error occurs during a read or write job, this status will be indicated by the "Error" bit. An error message is transmitted in the area of the input process data at the same time.

T1: Starting the read job by setting the "Start read" bit to TRUE

T2: Read job is active and being executed ("Job active" = TRUE)

T3: Read/write tag A with 4 bytes of memory enters the sensing range and the "Error" bit is set to TRUE. The "Job active" bit is reset to FALSE and at the same time the error code 0x04 and the text "invalid command" are entered in the input process data. This indicates that the read job set by the IODD file parameters is not suitable for the properties of the read/write tag. This is due to the number of bytes to be read. A maximum of 4 bytes can be read to access the read/write tag. In this example, the value is set to 8. To rectify the error, correct the value in the IODD file.

T4: The read job is started by setting the "Start read" bit to FALSE; at the same time, the "Error" bit is reset to FALSE and the error message in the input data is deleted

T5: Start a new read job by setting the "Start read" bit to TRUE

T6: Read job is active and being executed ("Job active" = TRUE)

T7: A write job is started by setting the "Start write" bit to TRUE. The "Error" bit is set and "Job active" is reset to FALSE. An error message with the error code 0x04 and the text "read AND write set" is transmitted to the input field of the process data. This indicates that a read and a write job has been activated simultaneously. This is not permitted for the device.

T8: The fault state remains active ("Fault" = TRUE), because the "Start read" and "Start write" bits are set

T9: The "Start read" and "Start write" bits are reset to FALSE; at the same time, the "Fault" bit is reset to FALSE and the error message is deleted in the input data

T10: Starting the read job by setting the "Start read" bit to TRUE

T11: Read job is active and being executed ("Job active" = TRUE)

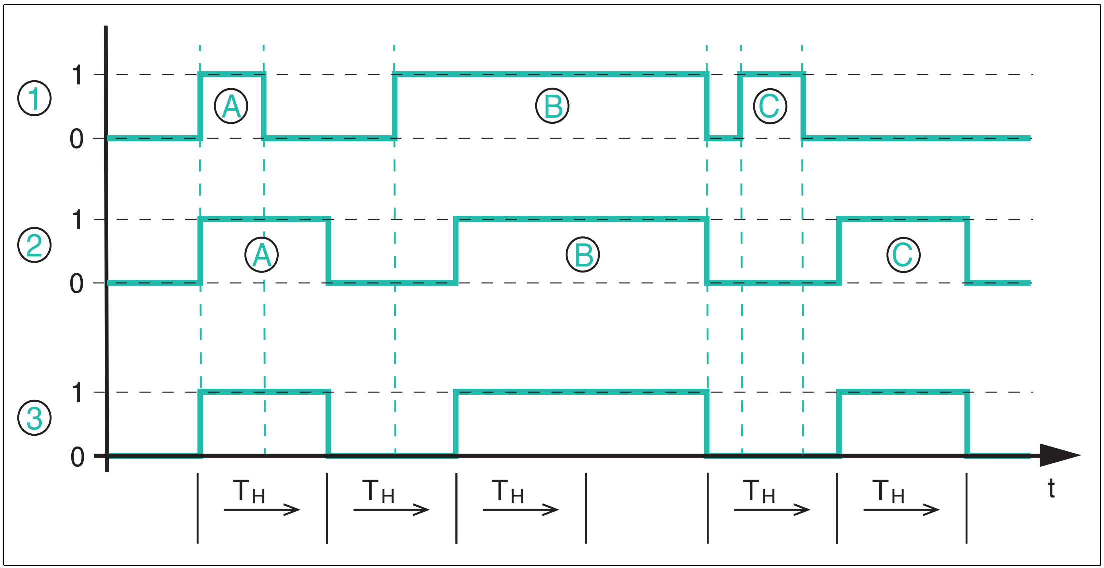
T12: A read/write tag enters the sensing range and is read successfully ("Read successful" = TRUE)

7.2.4 Timing

The device does not use the complex handshake procedure for data transmission in Easy Mode. The telegrams are set in the input process data and remain there for a defined hold time. The device cannot change the input process data within this hold time.

The hold time is ten times the set cycle time. The hold time is at least 40 ms long because the shortest possible cycle time is 4 ms.

The device can generate a new telegram within the hold time if a new read/write tag is read or a read/write tag leaves the sensing range. This telegram is only set in the input process data after the 40 ms have elapsed. If no new telegram occurs within the hold time, the input process data remains unchanged.



1 Read/write tag in field

2 Data

3 Read successful

The diagram shows the principle chronological sequence of the data transfer depending on the presence of a read/write tag within the device sensing range.

"T_H" corresponds to the minimum hold time of the device of 40 ms.

The device is activated via the auto-start function or the "Start read" bit. The device executes a read job permanently.

At the beginning, read/write tag A enters the sensing range of the device and the "Read successful" bit in the input process data changes the signal state to "TRUE" (1). The read/write tag stays in the sensing range for fewer than 40 ms and leaves it shortly after entering. The input process data containing information about read/write tag A is retained for the time "T_H" (= 40 ms). The input process data is only updated again once this time span has expired and then contains the information "Read successful" = FALSE (no read/write tag) and indicates that the read/write tag has left the sensing range. This telegram also remains in the input process data for the hold time of "T_H."

Read/write tag B enters the sensing range before the hold time of the previous telegram has expired. The input process data is only updated once the hold time of 40 ms has expired, and the "Read successful" bit then changes to "TRUE." The read-in data is simultaneously set in the input process data. Read/write tag B remains within the device sensing range for more than 40 ms (> "T_H"). During this time span, the input process data remains unchanged and the "Read successful" bit continues to have the signal state "TRUE."

Read/write tag B leaves the sensing range and the signal state of the "Read successful" bit changes from 1 to 0 in the input process data. Read/write tag C enters the sensing range before the hold time "T_H" expires. The input process data remains unchanged during the hold time span. It changes the signal state of "Read successful" to "TRUE" once "T_H" expires. The presence of the read/write tag C is indicated and this tag's read-in data is transmitted.

Read/write tag C leaves the sensing range before the hold time "T_H" expires. Once the hold time of the previous information has expired (read/write tag B has left the sensing range), the input process data is modified accordingly. The signal state of "Read successful" changes to "TRUE."

7.2.5 Interruption of IO-Link Communication

If IO-Link communication is interrupted, the device continues to operate normally. If a read job is active at the time of cancellation, the received data is stored in the device. When the device communicates via IO-Link again, this data is transferred to the IO-Link master.

Note

The device cache is designed for small amounts of data to compensate for short interruptions. Data may be lost if IO-Link communication is interrupted for a long time.

7.3 Easy Mode with PACTware

You can commission the RFID read/write device using the IO-Link master "IO-Link-Master02-USB."

Commissioning with PACTware

Note

Use PACTware version 5.0 software to operate the system.

You can use the "IODD Interpreter DTM" software to integrate the IODD files into the PACTware on a PC.

You can find the software, the IODD file and the driver on the Pepperl+Fuchs homepage.

1. Connect the RFID read/write device to the IO-Link master.
2. Connect the IO-Link master to a power supply.
3. Connect the IO-Link master to a PC using a USB cable.
4. Install the two software packages on your PC.

5. Install the **IO-Link USB Master DTM 2.0** driver.
6. Import the IODD file for the RFID read/write device with the **IODD DTM Configurator** program.

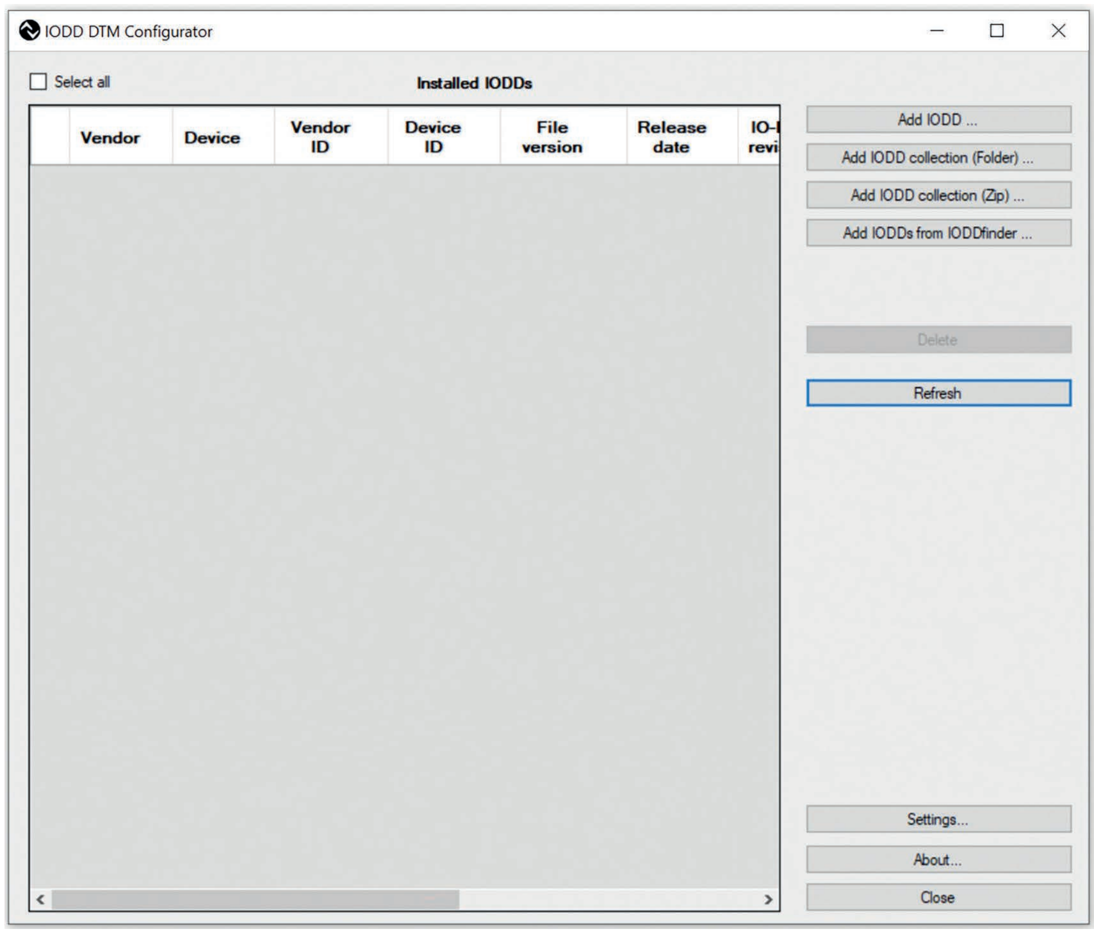


Figure 7.4

7. You can add IODDs using the IODDfinder in several ways:
   - "Add IODD" adds a single IODD.
   - "Add IODD Collection (Folder)" adds an entire folder that can contain multiple IODDs.
   - "Add IODD Collection (*.zip)" adds an archived folder that can contain multiple IODDs.[1]
   - "Add IODD from IODDfinder" allows full access to the IODDfinder database ioddfinder.io-link.com.
8. You can use the filter mask to narrow down the manufacturer and device. Enter "Pepperl+Fuchs" as the manufacturer and "IQT3-FP-IO-V1" as the device.
9. Select the required IODD file version of the device (highlighted in blue)
10. Click "Add selected IODD."

1. For example IODD download from the Pepperl+Fuchs homepage

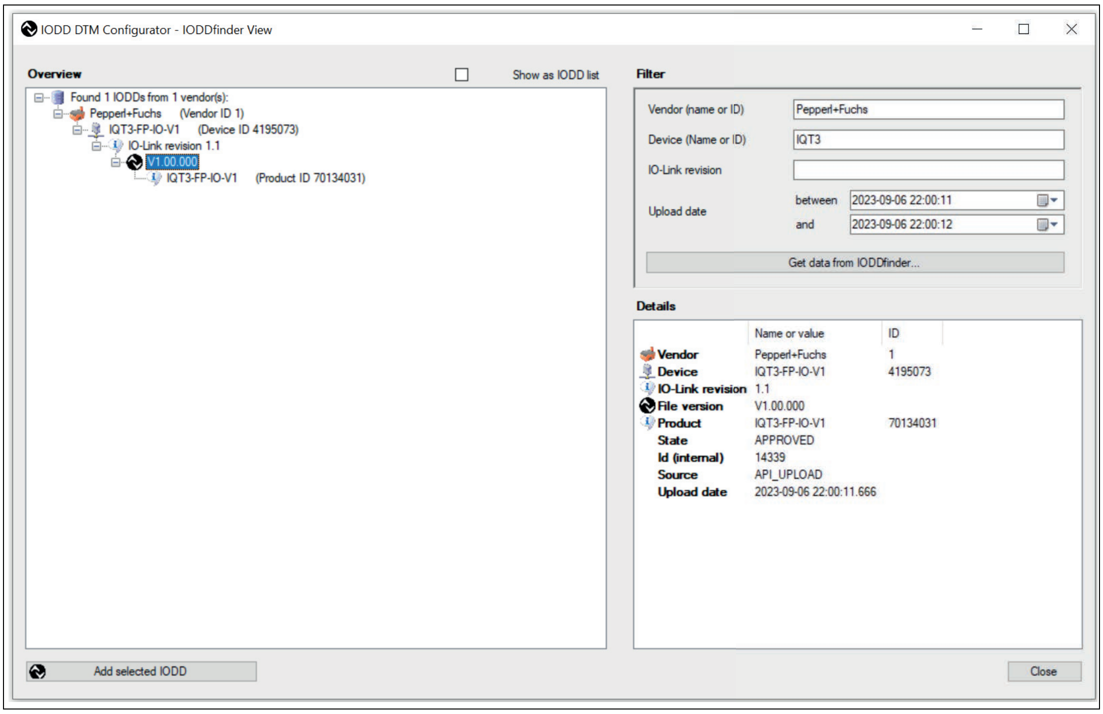


Figure 7.5

↳ Successfully added IODDs appear in "Installed IODDs."

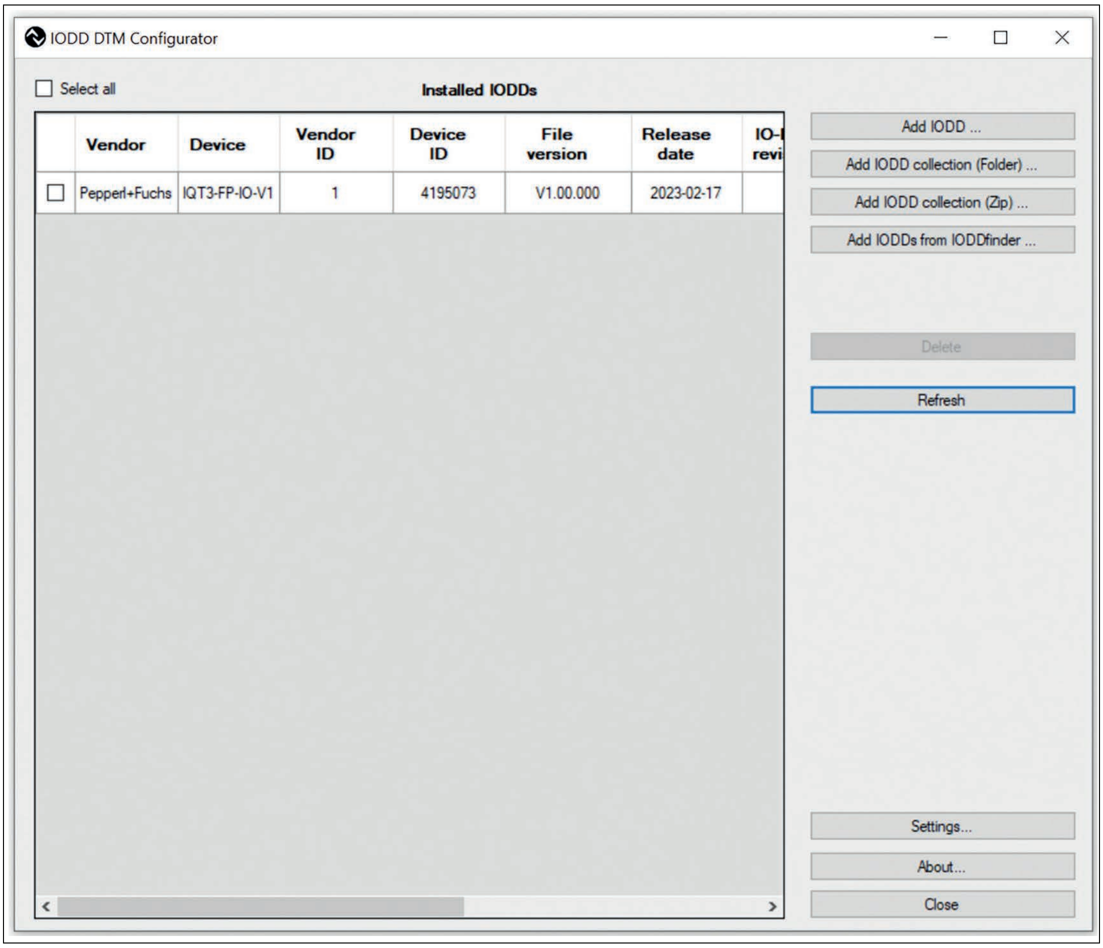


Figure 7.6

↳ Close the **IODD DTM Configurator** program.

2023-11

PEPPERL+FUCHS

11. Start PACTware.
12. Right-click on the "HOST PC."
13. Select the "Add device" menu item.

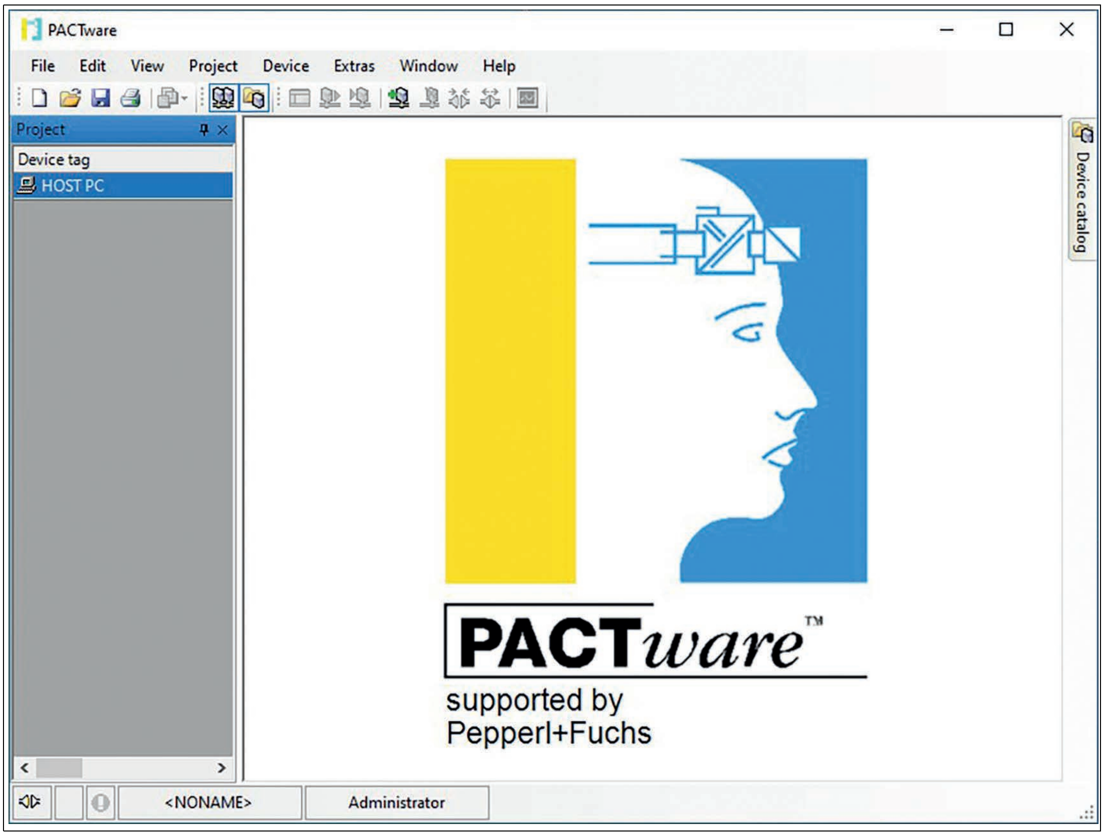


Figure 7.7

↳ The "Device for" window opens.

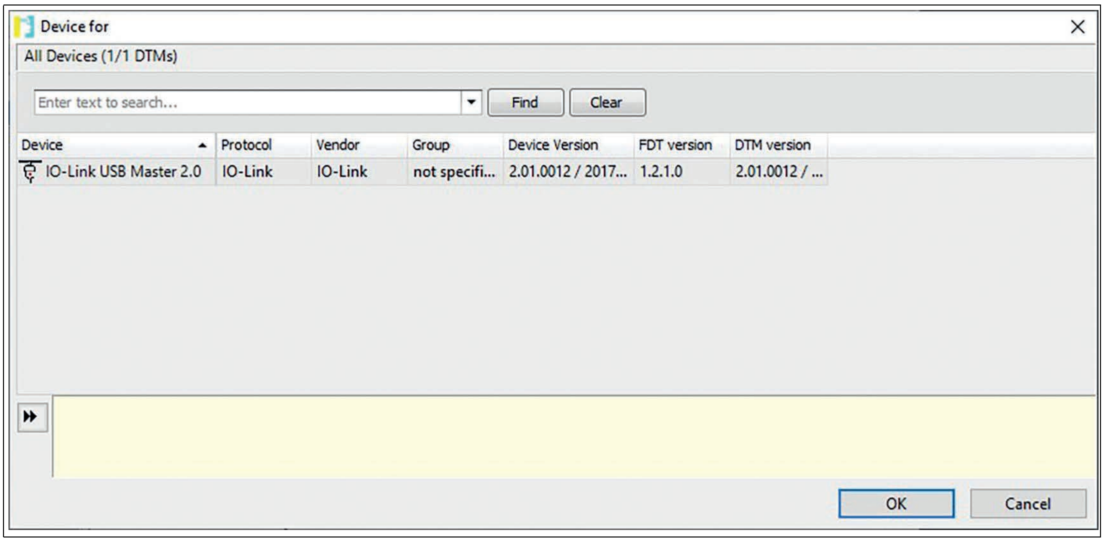


Figure 7.8

14. Select IO-Link USB Master 2.0.

Note

If you are using a different IO-Link master, select it.

15. Confirm with "OK."

↳ The IO-Link master appears in the menu on the left under your project.

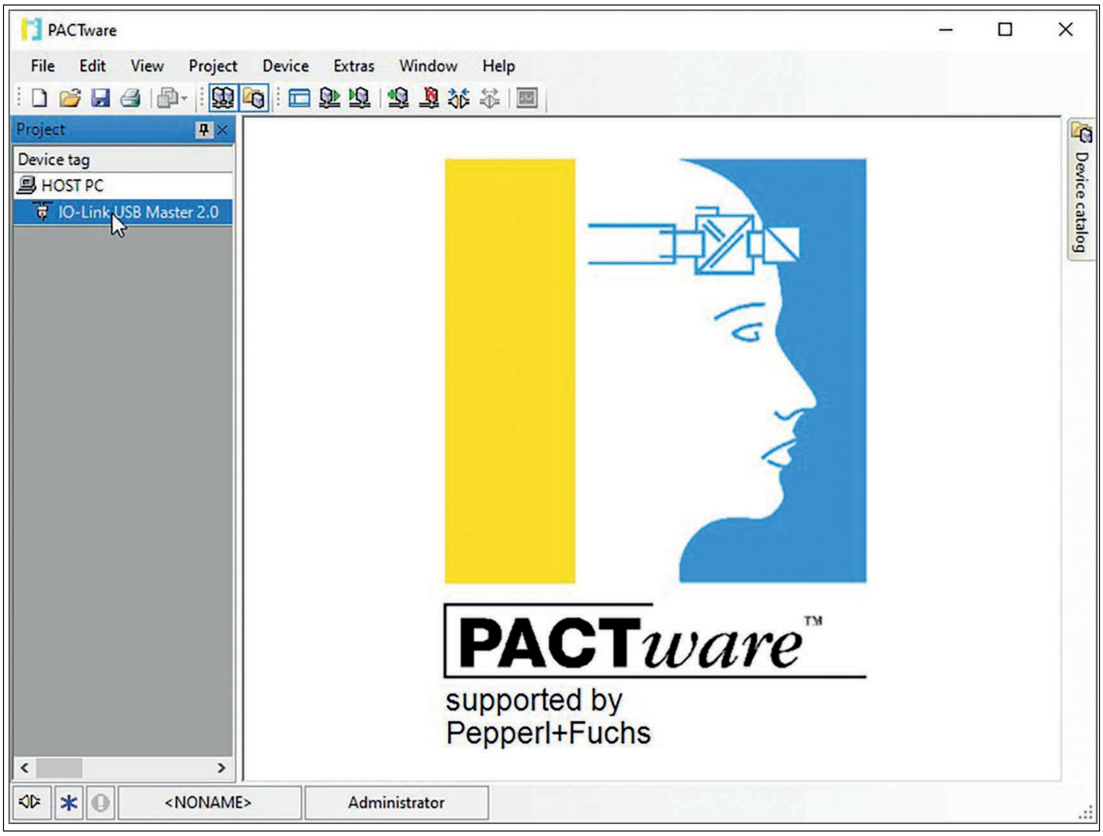


Figure 7.9

16. Right-click on the IO-Link master.
17. Select the "Add device" menu item.

↳ The "Device for" window opens.

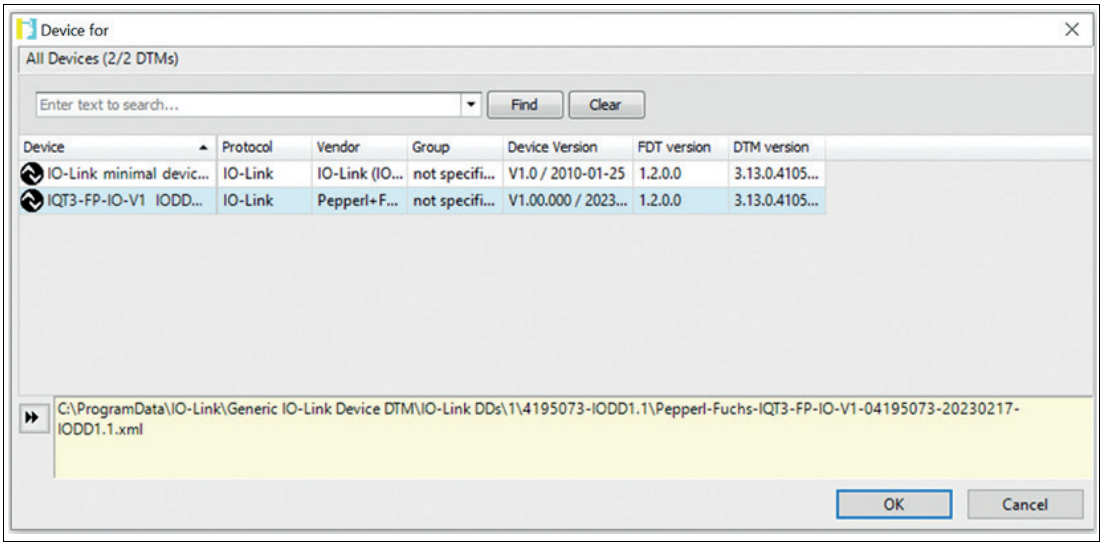


Figure 7.10

18. Select the required RFID read/write device.
19. Confirm with "OK."
20. Double-click the IO-Link device.

↳ The Parameters menu opens.

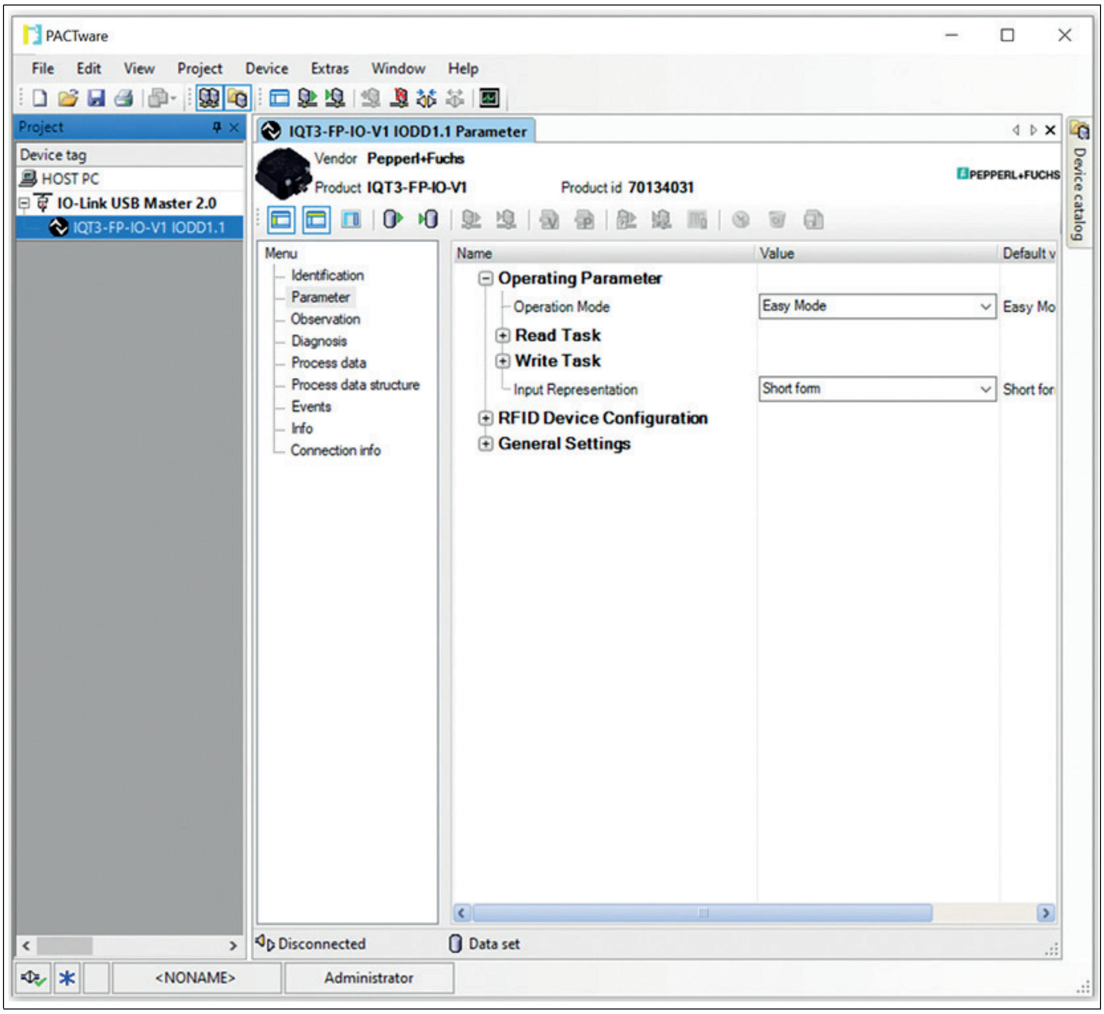


Figure 7.11

21. Right-click on the RFID read/write device and select the menu item "Connect."

22. Acknowledge the "Read from Device (Upload)?" dialog with "Yes."

↳ The parameter values displayed in PACTware are transferred to the device.

↳ A connection is established between the IO-Link master and the device.

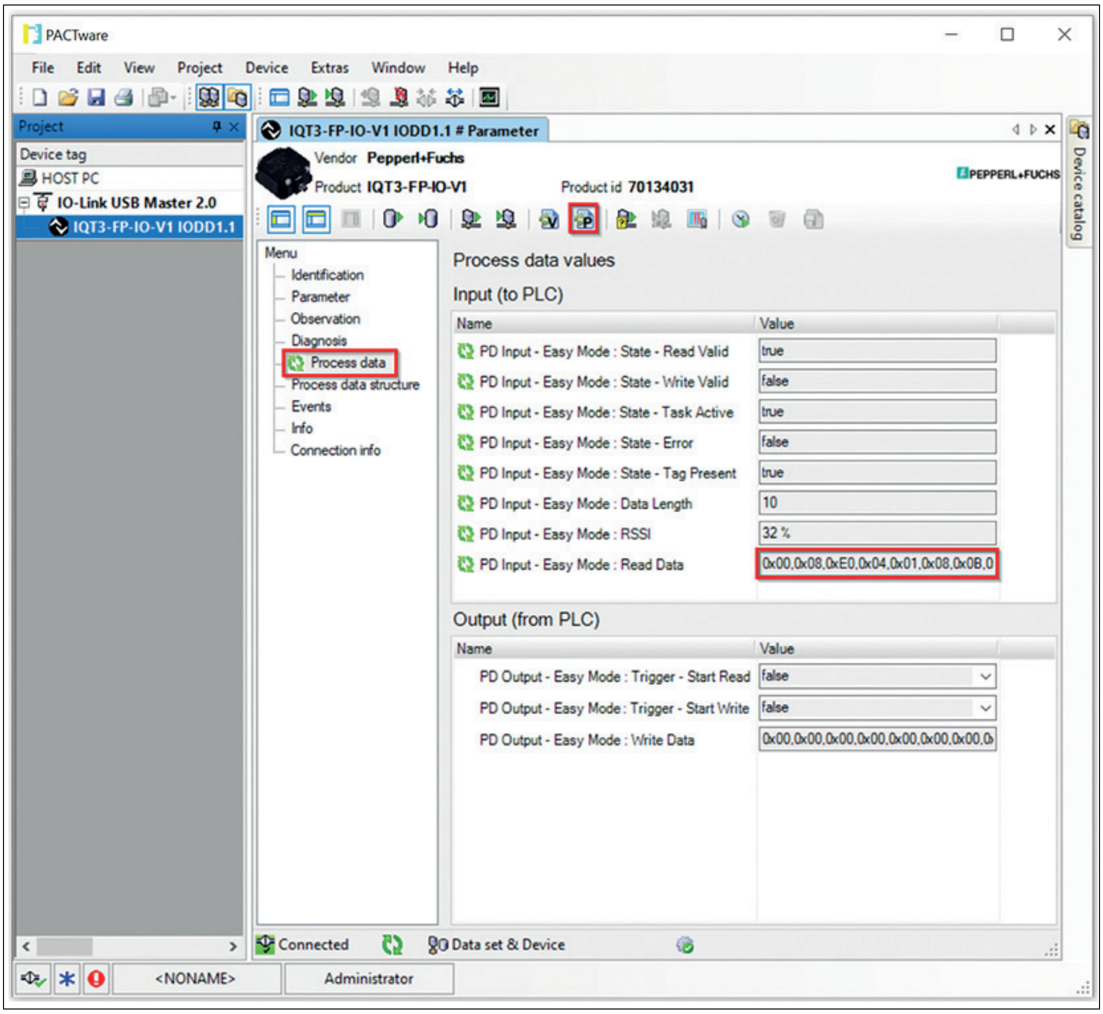


Figure 7.12

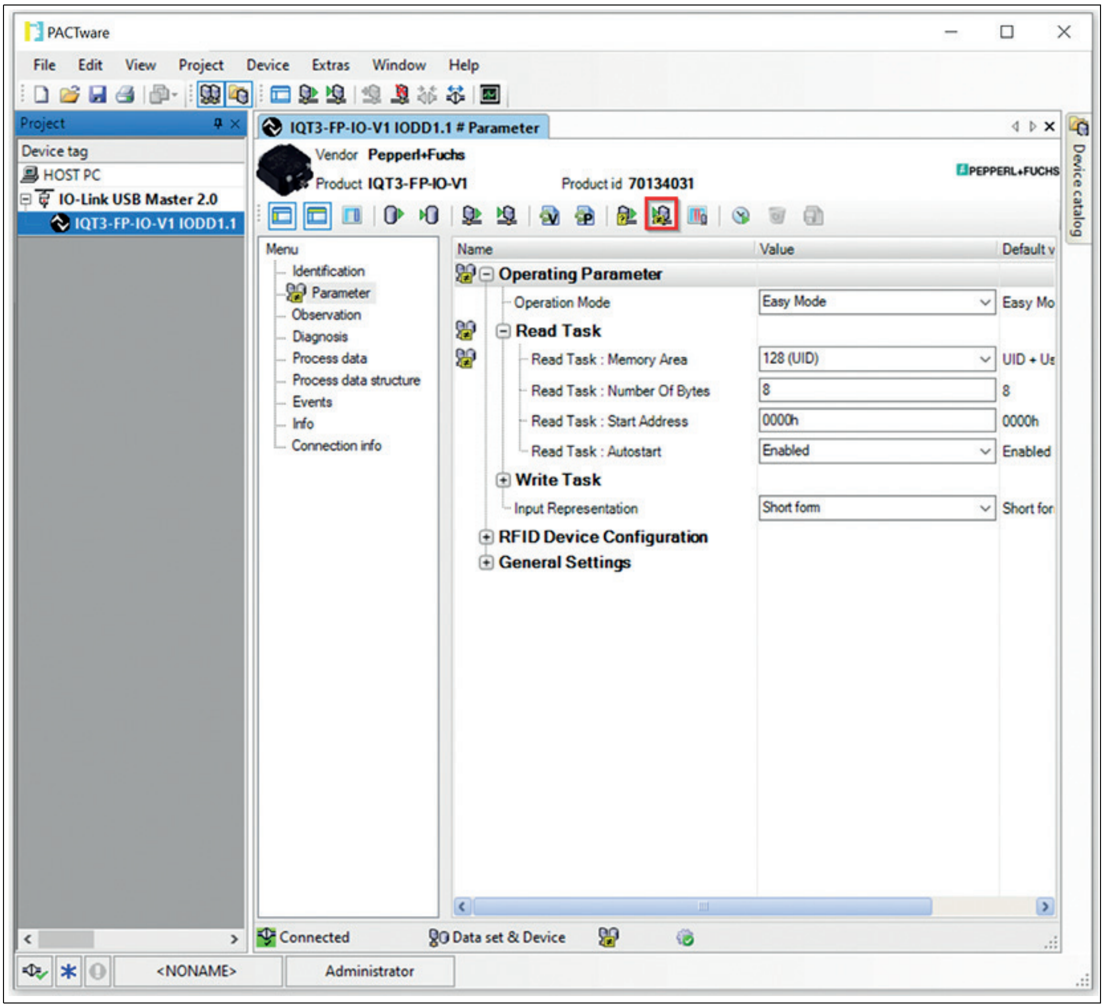
Tip

Activate the cyclic updating of the process data to display the data in PACTware.

As soon as a connection is established, the RFID read/write device automatically starts reading tags in the sensing range. The data is displayed in the input process data as per the set parameters under "Read Task."

1. Adjust the parameters of the device according to your application.
2. In the device selection menu, click on the Parameters entry.

   ↳ The Parameters menu opens.
3. Change the desired parameters and confirm the entry with the Enter key. Then click the "Write different values to device" icon.


PACTware
File
Edit
View
Project
Device
Extras
Window
Help
Project
Device tag
HOST PC
IO-Link USB Master 2.0
IQT3-FP-IO-V1 IODD1.1
IQT3-FP-IO-V1 IODD1.1 # Parameter
Vendor Pepperl+Fuchs
Product IQT3-FP-IO-V1
Product id 70134031
Menu
Identification
Parameter
Observation
Diagnosis
Process data
Process data structure
Events
Info
Connection info
Name
Value
Operating Parameter
Operation Mode
Easy Mode
Read Task
Read Task : Memory Area
128 (UID)
Read Task : Number Of Bytes
8
Read Task : Start Address
0000h
Read Task : Autostart
Enabled
Write Task
Input Representation
Short form
RFID Device Configuration
General Settings
Connected
Data set & Device
<NONAME>
Administrator
Device catalog
Figure 7.13

8 ExpertMode

8.1 Basic Command Process

As in Easy Mode, the length of the input and output process data is 32 bytes, see chapter 7. The commands are first combined to form a telegram. This telegram can be significantly longer than the set in/out length. The telegram is transmitted one after another using individual fragments. The maximum size of a fragment is 32 bytes. A handshake procedure controls the data transfer.

8.2 Legend

Name	Length	Meaning
<Number of Read/Write Tags>	4 bytes	The number of read/write tags identified when a single read or write command is executed. The number of read/write tags is encoded in ASCII format
<ByteAddress>	2 bytes	Start address for read/write access to the user memory of the read/write tag. The value must be a multiple of 4.
<Command>	1 byte	Command code Identifier of the command to be executed
<ControlByte>	4 bits	Control bits for implementing the handshake procedure or deleting the device memory
<FragmentationCounter>	1 byte	Fragmentation counter Number of fragments still to be transmitted
<FrameLength>	12 bits	Number of valid bytes within the telegram
<Length User Data>	2 bytes	Length of user data Specifies the length of the data read in from the user memory bank
<LengthParameter>	2 bytes	Parameter length Specifies the length of the parameter data set to be transmitted to the device.
<Number of Bytes>	2 bytes	Number of bytes to be accessed during execution of a read or write command. The value must be a multiple of 4.
<ParameterData>	Variable length	Parameter data Parameter data set that was read or transmitted.
<ParameterName>	2 bytes	Device parameter name The device parameter name determines the HF parameter to be accessed.
<Status>	1 byte	Status byte Status information about the execution state of the command.
<SystemCode>	1 byte	System identifier The system identifier for the HF system is "Q" (16#51).
<TelegramLength>	2 bytes	Length of telegram including all fragments
<UID>	8 bytes	Unique Identifier Unique serial number of a read/write tag
<User Data>	Variable length	Read-in data Data read in during read access to the user memory bank
<Write Data>	Variable length	Write data Data set to be transmitted to the read/write tag.

8.3 Structure of OUTPUT telegram

OUTPUT-Telegram

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
0	ControlByte / Frame Length	D_S	U_M	U_D	0	<FrameLength>			
1	Frame Length	<Frame Length>							
2	Fragmentation Counter	<Fragmentation Counter>							
3	Telegram Length (High Byte)[1]	<Telegram Length (High Byte)>							
4	Telegram Length (Low Byte)[1]	<Telegram Length (Low Byte)>							
5	Command[1]	<Command>							
6	Parameter / Data	<Data Byte 1>							
7	Parameter / Data	<Data Byte 2>							
8	Parameter / Data	...							
...	Parameter / Data	<Data Byte X>							
...	Parameter / Data	16#00							
...	Parameter / Data	...							
31	Parameter / Data	16#00							

Table 8.1

1. From the second fragment on, parameters / data are transmitted from byte 3.

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

<FragmentationCounter> has the value 16#00 because the command can be transmitted from the control panel via one fragment.

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

The <Command> byte specifies the command to be executed. Different commands are executed depending on the value in <Command>. The commands are classified as follows:

- **Read/Write Commands:** Access to one or more read/write tags in the sensing zone
- **System Commands:** Execution of device settings; no access to read/write tags
- **HF Configuration Commands:** Setting of the device HF properties

<Data Byte> is used to transfer the data required to execute a command. This can include additional command parameters (e.g., start address) or user data to be written to a read/write tag.

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

8.4 Structure of INPUT telegram

INPUT-Telegram

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

Byte	Content	Bit Number							
		7	6	5	4	3	2	1	0
2	Fragmentation Counter	<Fragmentation Counter>							
3	Telegram Length (High Byte)[1]	<Telegram Length (High Byte)>							
4	Telegram Length (Low Byte)[1]	<Telegram Length (Low Byte)>							
5	Command[1]	<Command>							
6	Status	<Status>							
7	Parameter / Data	<Data Byte 1>							
8	Parameter / Data	<Data Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<Data Byte X>							
...	Parameter / Data	0x00							
...	Parameter / Data	...							
31	Parameter / Data	0x00							

Table 8.2

1. From the second fragment on, parameters / data are transmitted from byte 3.

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

Because the command response can be transmitted from the controller via a fragment, the <Fragmentation Counter> has the value 16#00.

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

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

The value within <Status> indicates the status of the command execution. The corresponding status values indicate fault states. .

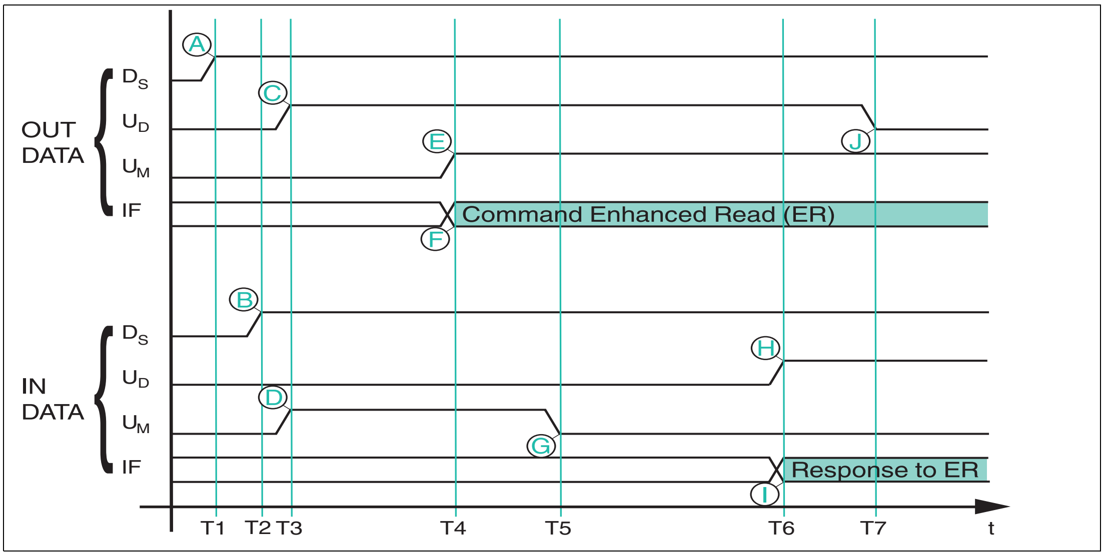
The data being transmitted from the device as a result of the command is returned in the <Data Byte> bytes. This could be data read from a read/write tag or parameter values for the UHF settings.

The unused areas within the response telegram are set to the value 16#00.

8.5 Handshake Procedure

Data flow between a PLC and the read/write device must be synchronized to ensure continuous data transfer with no losses. The input and output process data is transmitted in cycles. Controlling the data flow via the software is referred to as "handshaking." The necessary control bits are contained in the control byte (see chapter 8.4 and see chapter 8.3).

The following handshake procedure transfers telegrams quickly and securely between the PLC and the device:



D_S Delete bit; deletes internal memory of the device

U_M Update master bit

U_D Update device bit

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

T2 The read/write station changes the delete bit in the input process data (**B**) in response and deletes the entire contents of the FIFO memory

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

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

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

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

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

Sample Implementation in the Controller

Device delete bit D_S:

Once the device is ready for operation (IO-Link communication = OK), this instruction must be executed once. The internal telegram memory is deleted as a result of this. The internal telegram memory should be deleted if an internal device fault has occurred.

`D_S_OUTPUT := NOT D_S_INPUT`

Device update bit U_D:

New, valid data will be in the input process data if U_D bits in the input and output process data have the same value. The device writes new read data to the input process data only once the PLC has read the input process data, i.e., the U_D bit in the input and output process data has an inverted signal state.

To prevent transmission of the read data from being blocked, the inverted state of the U_D bit must be transmitted from the input process data to the U_D bit of the output process data in each cycle.

`U_D_OUTPUT := NOT U_D_INPUT` (* copy the inverted update bit from the INPUT telegram to the OUTPUT telegram *)

Master update bit U_M:

The device must be ready to receive new telegrams before a command is sent. This is the case if the U_M bit in the input and output process data has an inverted signal state.

The command parameters must then be transmitted to the corresponding positions in the output process data.

The PLC transfers the new command to the device once the U_M bit in the output process data is set to the same signal state as the U_M bit in the input process data.

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

`IF (U_M_OUTPUT <> U_M_INPUT) then` (* check whether the device can receive new data *)

`U_M_OUTPUT := U_M_INPUT` (* device is ready to receive, transfer of update bit *)

`End_IF`

8.6 Overview of Commands

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

Read/Write Commands

Abbreviation	Command code	Command description
SF	16#01	Single Read Fixcode see "Single Read Fixcode (SF)" on page 43 One-time read access to the Fixcode (UID)
EF	16#1D	Enhanced Read Fixcode see "Enhanced Read Fixcode (EF)" on page 46 Permanent read access to the Fixcode (UID)
SR	16#10	Single Read Words see "Single Read Words (SR)" on page 49 One-time read access to the user data area
ER	16#19	Enhanced Read Words see "Enhanced Read Words (ER)" on page 53 Permanent read access to the user data area
SW	16#40	Single Write Words see "Single Write Words (SW)" on page 56 One-time write access to the user data area

Abbreviation	Command code	Command description
EW	16#1A	Enhanced Write Words see "Enhanced Write Words with Lock (EL)" on page 66 Permanent write access to the user data area
SL	16#47	Single Write Words with Locksee "Single Write Words with Locks (SL)" on page 63 One-time write access to the user data area with subsequent write protection
EL	16#48	Enhanced Write Words with Lock see "Enhanced Write Words (EW)" on page 59 Permanent write access to the user data area with subsequent write protection

System Commands

Abbreviation	Command code	Command description
QU	16#02	Quit see "Quit (QU)" on page 69 Cancels an active enhanced command
VE	16#03	Version see "Version (VE)" on page 71 Reads out the firmware version
CT	16#04	Change tag see "Change Tag (CT)" on page 72 Set the tag type

Configuration Commands

Abbreviation	Command code	Command description
RP	16#BE	Read parameter see "Read Parameter (RP)" on page 73 Reads out the device parameters
WP	16#BF	Write parameter see "Write Parameter (WP)" on page 75 Sets device parameters

8.6.1 Read/Write Commands

The following commands and responses are described using the long-form data format. The end of a single command is signaled back via a STATUS 16#0F telegram. When using the short-form data format, the ready-only code and the additional length information are omitted. The end of an enhanced command is not signaled back.

Single Read Fixcode (SF)

The "Single Read Fixcode" command has the command code 16#01 and performs a one-time read operation on the read-only code UID of a tag within the sensing zone. The information is transmitted for each tag 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:

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							
Byte 6	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.3

The <FrameLength> has the value 16#06 because no other command parameters are transferred and the fragment ends after <Command>. The <FragmentationCounter> has the value 16#00 because the command telegram can be transmitted within one fragment. The length of the command telegram in bytes (<TelegramLength>) is 16#03.

The command code <Command> for the "Single Read Fixcode" command is 16#01.

Note

Set the non-relevant bytes of the command fragment to the value 16#00.

Response data telegram, long-form data format:

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	Parameter/data	16#00							
Byte 8	Parameter/data	16#08							
Byte 9	Parameter/data	<UID Byte 1>							
Byte 10	Parameter/data	<UID Byte 2>							
Byte 11	Parameter/data	<UID Byte 3>							
Byte 12	Parameter/data	<UID Byte 4>							
Byte 13	Parameter/data	<UID Byte 5>							
Byte 14	Parameter/data	<UID Byte 6>							
Byte 15	Parameter/data	<UID Byte 7>							
Byte 16	Parameter/data	<UID Byte 8>							
Byte 17	Not relevant	16#00							

2023-11

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.4

The <FrameLength> has a value of 16#11. This specifies the length of the fragment up to and including the <UID Byte 8> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#0E and specifies the length of the response telegram starting from the telegram length itself up to and including the <UID Byte 8> byte.

The <Command> parameter has the value 16#01 and is mirrored within the data telegram. The <Status> parameter has the value 16#00 and therefore signals that access to the tag was successful.

The length specification for the read-only code always has a size of 2 bytes and the value 16#0008.

The read-in read-only code is used to uniquely identify the read-in tag.

All other bytes of the response fragment are not relevant and have a value of 16#00.

Response signal strength tag response, long-form data format:

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	Parameter/data	16#01							
Byte 8	Parameter/data	<RSSI value>							
Byte 9	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.5

The <FrameLength> is constant at the value 16#09. The fragment extends up to and including the <RSSI value> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#06.

The <Command> byte is mirrored and has the value as the value 16#01 as in the command telegram. The <Status> for the telegram that indicates the RSSI value is 16#0B. The first byte <Parameter / Data> has the value 16#01. The second byte <Parameter / Data> describes the <RSSI value>.

All other bytes of the response fragment are not relevant and have a value of 16#00.

End of command response, long-form data format:

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 7	Parameter/data	<Number of Read/Write Tags Byte 1>							
Byte 8	Parameter/data	<Number of Read/Write Tags Byte 2>							
Byte 9	Parameter/data	<Number of Read/Write Tags Byte 3>							
Byte 10	Parameter/data	<Number of Read/Write Tags Byte 4>							
Byte 11	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.6

The <FrameLength> has the value 16#0B in the response command end of the "Single Read Fixcode." The fragment extends up to and including <Number of Read/Write Tags Byte 4>. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#08.

The <Command> parameter has the value 16#01 and is mirrored within the data telegram. The <Status> for the telegram that indicates the command end is 16#0F.

The number of identified tags is transmitted within 4 bytes. The number is displayed in ASCII.

When identifying a tag, the <Number of Read/Write Tags> has the value "0001" (ASCII) or 16#30303031.

If no tag was detected during command execution, the data telegrams are omitted. The telegram is sent to signal the end of the command. <Number of Read/Write Tags> has the value "0000" (ASCII) or 16#30303030.

Enhanced Read Fixcode (EF)

The "Enhanced Read Fixcode" command has the command code 16#1D and performs a permanent read operation on the read-only code of a tag within the sensing zone. The information is transmitted for each tag 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 sensing zone is indicated by a telegram containing the Fixcode of the tag. This telegram has the status value 16#05. A Quit command stops the command execution.

Command:

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	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 4	Telegram length (low byte)	16#03							
Byte 5	Command	16#1D							
Byte 6	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.7

The <FrameLength> has the value 16#06 because no other command parameters have to be transferred and the fragment ends after <Command>. The <FragmentationCounter> has the value 16#00 because the command telegram can be transmitted within one fragment. The length of the command telegram in bytes (<TelegramLength>) is 16#03.

The command code <Command> for the Enhanced Read Fixcode command is 16#1D.

Note

Set the non-relevant bytes of the command fragment to the value 16#00.

Response data telegram, long-form data format:

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							
Byte 7	Length UID (High Byte)	16#00							
Byte 8	Length UID (Low Byte)	16#08							
Byte 9	Parameter/data	<UID Byte 1>							
Byte 10	Parameter/data	<UID Byte 2>							
Byte 11	Parameter/data	<UID Byte 3>							
Byte 12	Parameter/data	<UID Byte 4>							
Byte 13	Parameter/data	<UID Byte 5>							
Byte 14	Parameter/data	<UID Byte 6>							
Byte 15	Parameter/data	<UID Byte 7>							
Byte 16	Parameter/data	<UID Byte 8>							
Byte 17	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.8

The <FrameLength> has a value of 16#11. This specifies the length of the fragment up to and including the <Fixcode Byte 8> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#0E and specifies the length of the response telegram starting from the telegram length itself up to and including the <UID Byte 8> byte.

The <Command> parameter has the value 16#1D and is mirrored within the data telegram. The <Status> parameter has the value 16#00 and signals that access to the tag was successful. This is followed by a length specification for the read-only code. It is always 2 bytes in size and a value of 16#0008. This is followed by the read-in read-only code to uniquely identify the tag that has been read.

All other bytes of the response fragment are not relevant and have a value of 16#00.

Response signal strength tag response, long-form data format:

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	Parameter/data	16#01							
Byte 8	Parameter/data	<RSSI value>							
Byte 9	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.9

The <FrameLength> is constant at the value 16#09. The fragment extends up to and including the <RSSI value> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#06.

The <Command> byte is mirrored and has the value 16#1D as in the command telegram. The <Status> for the telegram that indicates the RSSI value is 16#0B. The first byte <Parameter / Data> has the value 16#01. The second byte <Parameter / Data> describes the <RSSI value>.

All other bytes of the response fragment are not relevant and have a value of 16#00.

Response tag has left sensing zone, long-form data format:

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#05							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 7	Length UID (High Byte)	16#00							
Byte 8	Length UID (Low Byte)	16#08							
Byte 9	Parameter/data	<UID Byte 1>							
Byte 10	Parameter/data	<UID Byte 2>							
Byte 11	Parameter/data	<UID Byte 3>							
Byte 12	Parameter/data	<UID Byte 4>							
Byte 13	Parameter/data	<UID Byte 5>							
Byte 14	Parameter/data	<UID Byte 6>							
Byte 15	Parameter/data	<UID Byte 7>							
Byte 16	Parameter/data	<UID Byte 8>							
Byte 17	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.10

The <FrameLength> has a value of 16#11. This specifies the length of the fragment up to and including the <UID Byte 8> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#0E and specifies the length of the response telegram starting from the telegram length itself up to and including the <UID Byte 8> byte.

The <Command> parameter has the value 16#1D and is mirrored within the data telegram. The <Status> parameter has the value 16#05 and signals that the tag has left the sensing zone. This is followed by a length specification for the read-only code. This always has a size of 2 bytes and the value 16#0008. This is followed by the read-in read-only code to uniquely identify the tag leaving the sensing range.

All subsequent bytes within the telegram have the value 16#00.

Single Read Words (SR)

The "Single Read Words" command has the command code 16#10 and performs a one-time read access to the user data area of a tag within the sensing range. The information is transmitted for each tag 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.

The amount of user data available on a tag depends on the chip type used and can vary in size. For more detailed information on tag types .

The <ByteAddress> parameter specifies the start address within the user data area. The value of <ByteAddress> 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.

Command:

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	Parameter/data	<ByteAddress (High Byte)>							
Byte 7	Parameter/data	<ByteAddress (Low Byte)>							
Byte 8	Parameter/data	<Number of Bytes (High Byte)>							
Byte 9	Parameter/data	<Number of Bytes (Low Byte)>							
Byte 10	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.11

The <FrameLength> has a value of 16#0A and extends up to and including the <Number of Bytes (Low Byte)> byte. The <FragmentationCounter> has the value 16#00 because the command telegram can be transmitted within one fragment. The command telegram has a length (<TelegramLength>) of 16#07 and ends with the byte <Number of Bytes (Low Byte)>.

The command code <Command> for the Enhanced Read Fixcode command is 16#10.

The <ByteAddress> parameter specifies the start address within the user data area from which the memory blocks are read. The <Number of Bytes> parameter is used to specify the number of bytes to be read.

Note

Set the non-relevant bytes of the command fragment to the value 16#00.

Response data telegram, long-form data format:

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	<FrameLength>							
Byte 2	Fragmentation counter	<FragmentationCounter>							
Byte 3	Telegram length (high byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram length (low byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#10							
Byte 6	Status	16#00							
Byte 7	Length UID (High Byte)	16#00							
Byte 8	Length UID (Low Byte)	16#08							
Byte 9	Parameter/data	<UID Byte 1>							
Byte 10	Parameter/data	<UID Byte 2>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 11	Parameter/data	<UID Byte 3>							
Byte 12	Parameter/data	<UID Byte 4>							
Byte 13	Parameter/data	<UID Byte 5>							
Byte 14	Parameter/data	<UID Byte 6>							
Byte 15	Parameter/data	<UID Byte 7>							
Byte 16	Parameter/data	<UID Byte 8>							
Byte 17	Parameter/data	<Length User Data (High Byte)>							
Byte 18	Parameter/data	<Length User Data (Low Byte)>							
Byte 19	Parameter/data	<User Data Byte 1>							
Byte 20	Parameter/data	<User Data Byte 2>							
Byte 21	Parameter/data	<User Data Byte 3>							
Byte 22	Parameter/data	<User Data Byte 4>							
etc.	Parameter/data	etc.							
etc.	Parameter/data	<User Data Byte X>							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.12

The length of the fragment <FrameLength> depends on the number of bytes read in from the identified tag. The <FrameLength> contains all bytes up to and including the byte <User Data Byte X>. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The value of <TelegramLength> depends on the number of bytes read in from the tag. The size of the telegram extends up to and including the byte <User Data Byte X>.

The <Command> parameter has the value 16#10 and is mirrored within the data telegram. The <Status> parameter has the value 16#00. This is followed by a length specification for the read-only code. It is always 2 bytes in size and a value of 16#0008. This is followed by the read-in read-only code to uniquely identify the tag that has been read. The read-only code is followed by a length specification of the read-in user data. It is always 2 bytes in size. The final piece of information is the user data read in from the identified tag up to and including <User Data Byte X>.

All other bytes of the response fragment are not relevant and have a value of 16#00.

Response signal strength tag response, long-form data format:

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#10							
Byte 6	Status	16#0B							
Byte 7	Parameter/data	16#01							
Byte 8	Parameter/data	<RSSI value>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 9	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.13

The <FrameLength> is constant at the value 16#09. The fragment extends up to and including the <RSSI value> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#06.

The <Command> byte is mirrored and has the value 16#10 as in the command telegram. The <Status> for the telegram that indicates the RSSI value is 16#0B. The first byte <Parameter / Data> has the value 16#01. The second byte <Parameter / Data> describes the <RSSI value>.

All other bytes of the response fragment are not relevant and have a value of 16#00.

End of command response, long-form data format:

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	Parameter/data	<Number of Read/Write Tags Byte 1>							
Byte 8	Parameter/data	<Number of Read/Write Tags Byte 2>							
Byte 9	Parameter/data	<Number of Read/Write Tags Byte 3>							
Byte 10	Parameter/data	<Number of Read/Write Tags Byte 4>							
Byte 11	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.14

The <FrameLength> is constant at the value 16#0B in the response for the end of the command of the "Single Read Words." The fragment extends up to and including <Number of Read/Write Tags Byte 4>. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#08.

The <Command> byte is mirrored and has the same value as the 16#10 command telegram. The <Status> for the telegram that indicates the command end is 16#0F.

The number of identified tags is transmitted within 4 bytes. The number is displayed in ASCII coding.

When identifying a tag, the <Number of Read/Write Tags> has the value "0001" (ASCII) or 16#30303031.

If no tag was detected while the command was executed, the data telegrams are omitted and only the telegram to indicate the end of the command is sent. <Number of Read/Write Tags> has the value "0000" (ASCII) or 16#30303030.

Enhanced Read Words (ER)

The command "Enhanced Read Words" has the command code 16#19 and performs a permanent read access to the user data area of a tag within the sensing zone. The information is transmitted for each tag 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 sensing zone is indicated by a telegram containing the read-only code of the tag. This telegram has the status value 16#05. A Quit command stops the command execution.

The amount of user data available on a tag depends on the chip type used and can vary in size. For more detailed information on tag types .

The <ByteAddress> parameter specifies the start address within the user data area. The value of <ByteAddress> 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.

Command:

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	Parameter/data	<ByteAddress (High Byte)>							
Byte 7	Parameter/data	<ByteAddress (Low Byte)>							
Byte 8	Parameter/data	<Number of Bytes (High Byte)>							
Byte 9	Parameter/data	<Number of Bytes (Low Byte)>							
Byte 10	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.15

The <FrameLength> has a value of 16#0A and extends up to and including the <Number of Bytes (Low Byte)> byte. The <FragmentationCounter> has the value 16#00 because the command telegram can be transmitted within one fragment. The command telegram has a length (<TelegramLength>) of 16#07 and ends with the byte <Number of Bytes (Low Byte)>.

The command code <Command> for the Enhanced Read Fixcode command is 16#19.

The <ByteAddress> parameter specifies the start address within the user data area from which the memory blocks are read. The <Number of Bytes> parameter is used to specify the number of bytes to be read.

Note

Set the non-relevant bytes of the command fragment to the value 16#00.

Response data telegram, long-form data format:

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	<FrameLength>							
Byte 2	Fragmentation counter	<FragmentationCounter>							
Byte 3	Telegram length (high byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram length (low byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#19							
Byte 6	Status	16#00							
Byte 7	Length UID (High Byte)	16#00							
Byte 8	Length UID (Low Byte)	16#08							
Byte 9	Parameter/data	<UID Byte 1>							
Byte 10	Parameter/data	<UID Byte 2>							
Byte 11	Parameter/data	<UID Byte 3>							
Byte 12	Parameter/data	<UID Byte 4>							
Byte 13	Parameter/data	<UID Byte 5>							
Byte 14	Parameter/data	<UID Byte 6>							
Byte 15	Parameter/data	<UID Byte 7>							
Byte 16	Parameter/data	<UID Byte 8>							
Byte 17	Parameter/data	<Length User Data (High Byte)>							
Byte 18	Parameter/data	<Length User Data (Low Byte)>							
Byte 19	Parameter/data	<User Data Byte 1>							
Byte 20	Parameter/data	<User Data Byte 2>							
Byte 21	Parameter/data	<User Data Byte 3>							
Byte 22	Parameter/data	<User Data Byte 4>							
etc.	Parameter/data	etc.							
etc.	Parameter/data	<User Data Byte X>							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.16

The length of the fragment <FrameLength> depends on the number of bytes read in from the identified tag. The <FrameLength> contains all bytes up to and including the byte <User Data Byte X>. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The value of <TelegramLength> depends on the number of bytes read in from the tag. The size of the telegram extends up to and including the byte <User Data Byte X>.

The <Command> parameter has the value 16#19 and is mirrored within the data telegram. The <Status> parameter has the value 16#00. This is followed by a length specification for the read-only code. It is always 2 bytes in size. This is followed by the read-in read-only code to uniquely identify the tag that has been read. The read-only code is followed by a length specification of the read-in user data. It is always 2 bytes in size and a value of 16#0008. The final piece of information is the user data read in from the identified tag up to and including <User Data Byte X>.

All other bytes of the response fragment are not relevant and have a value of 16#00.

Response signal strength tag response, long-form data format:

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	Parameter/data	16#01							
Byte 8	Parameter/data	<RSSI value>							
Byte 9	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.17

The <FrameLength> is constant at the value 16#09. The fragment extends up to and including the <RSSI value> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#06.

The <Command> byte is mirrored and has a value of 16#19 as in the command telegram. The <Status> for the telegram that indicates the RSSI value is 16#0B. The first byte <Parameter / Data> has the value 16#01. The second byte <Parameter / Data> describes the <RSSI value>.

All other bytes of the response fragment are not relevant and have a value of 16#00.

Response tag has left sensing zone, long-form data format:

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	Length UID (High Byte)	16#00							
Byte 8	Length UID (Low Byte)	16#08							
Byte 9	Parameter/data	<UID Byte 1>							
Byte 10	Parameter/data	<UID Byte 2>							
Byte 11	Parameter/data	<UID Byte 3>							
Byte 12	Parameter/data	<UID Byte 4>							
Byte 13	Parameter/data	<UID Byte 5>							
Byte 14	Parameter/data	<UID Byte 6>							
Byte 15	Parameter/data	<UID Byte 7>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 16	Parameter/data	<UID Byte 8>							
Byte 17	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.18

The <FrameLength> has a value of 16#11. This specifies the length of the fragment up to and including the <UID Byte 8> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#0E and specifies the length of the response telegram starting from the telegram length itself up to and including the <UID Byte 8> byte.

The <Command> parameter has the value 16#19 and is mirrored within the data telegram. The <Status> parameter has the value 16#05 and signals that the tag has left the sensing zone. This is followed by a length specification for the read-only code. It is always 2 bytes in size and a value of 16#0008. This is followed by the read-in read-only code to uniquely identify the tag leaving the sensing range.

All subsequent bytes within the telegram have the value 16#00.

Single Write Words (SW)

The "Single Write Words" command has the command code 16#40 and performs a one-time write operation to the user data area of a tag within the sensing range. A separate data telegram with the status value 16#00 indicates a successful write operation for each tag described. The data telegram contains the read-only code of the tag to which the user data area was written. In addition, for each tag described, 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 final telegram has the status value 16#0F and contains the number of tags that were written during execution of the command.

The amount of user data available on a read/write tag depends on the type of chip used and can vary in size. You can find more detailed information in the overview of tag types.

The <ByteAddress> parameter specifies the start address within the user data area. The value of <ByteAddress> 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.

Command:

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	<FrameLength>							
Byte 2	Fragmentation counter	<FragmentationCounter>							
Byte 3	Telegram length (high byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram length (low byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#40							
Byte 6	Parameter/data	<Byte Address (High Byte)>							
Byte 7	Parameter/data	<Byte Address (Low Byte)>							
Byte 8	Parameter/data	<Number of Bytes (High Byte)>							
Byte 9	Parameter/data	<Number of Bytes ((Low Byte)>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 10	Parameter/data	<Write Data Byte 1>							
Byte 11	Parameter/data	<Write Data Byte 2>							
Byte 12	Parameter/data	<Write Data Byte 3>							
Byte 13	Parameter/data	<Write Data Byte 4>							
etc.	Parameter/data	etc.							
etc.	Parameter/data	<Write Data Byte X>							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.19

The value of <FrameLength> depends on the number of bytes to be written. The fragment ends with the <Write Data Byte X> byte. The <FragmentationCounter> has the value 16#00 because the command telegram can be transmitted within one fragment. The length of the command in bytes (<TelegramLength>) depends on the number of bytes to be written. The telegram ends with the <Write Data Byte X> byte.

The command code <Command> for the "Single Write Words" command is 16#40.

The <ByteAddress> parameter specifies the start address within the user data area from which the data is written. The <Number of Bytes> parameter is used to specify the number of bytes to be written. This is followed by the information to be written to the tag <Write Data Byte>.

Note

Set the non-relevant bytes of the command fragment to the value 16#00.

Response data telegram, data successfully written, long-form data format:

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	Length UID (High Byte)	16#00							
Byte 8	Length UID (Low Byte)	16#08							
Byte 9	Parameter/data	<UID Byte 1>							
Byte 10	Parameter/data	<UID Byte 2>							
Byte 11	Parameter/data	<UID Byte 3>							
Byte 12	Parameter/data	<UID Byte 4>							
Byte 13	Parameter/data	<UID Byte 5>							
Byte 14	Parameter/data	<UID Byte 6>							
Byte 15	Parameter/data	<UID Byte 7>							
Byte 16	Parameter/data	<UID Byte 8>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 17	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.20

The <FrameLength> has a value of 16#11. This specifies the length of the fragment up to and including the <UID Byte 8> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#0E and specifies the length of the response telegram starting from the telegram length itself up to and including the <UID Byte 8> byte.

The <Command> parameter has the value 16#40 and is mirrored within the data telegram. The <Status> parameter has the value 16#00 and signals that access to the tag was successful. This is followed by a length specification for the read-only code. It is always 2 bytes in size and a value of 16#0008. This is followed by the read-in read-only code to uniquely identify the tag that has been read.

All other bytes of the response fragment are not relevant and have a value of 16#00.

Response signal strength tag response, long-form data format:

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	Parameter/data	16#01							
Byte 8	Parameter/data	<RSSI value>							
Byte 9	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.21

The <FrameLength> is constant at the value 16#09. The fragment extends up to and including the <RSSI value> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#06.

The <Command> byte is mirrored and has the value 16#40 as in the command telegram. The <Status> for the telegram that indicates the RSSI value is 16#0B. The first byte <Parameter / Data> has the value 16#01. The second byte <Parameter / Data> describes the <RSSI value>.

All other bytes of the response fragment are not relevant and have a value of 16#00.

End of command response, long-form data format:

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	Parameter/data	<Number of Read/Write Tags Byte 1>							
Byte 8	Parameter/data	<Number of Read/Write Tags Byte 2>							
Byte 9	Parameter/data	<Number of Read/Write Tags Byte 3>							
Byte 10	Parameter/data	<Number of Read/Write Tags Byte 4>							
Byte 11	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.22

The <FrameLength> is constant at the value 16#0B in the response for the end of the command of the "Single Read Words." The fragment extends up to and including <Number of Read/Write Tags Byte 4>. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#08.

The <Command> byte is mirrored and has the same value as in the 16#40 command telegram. The <Status> for the telegram that indicates the command end is 16#0F.

The number of identified tags is transmitted within 4 bytes. The number is displayed in ASCII coding.

When identifying a tag, the <Number of Read/Write Tags> has the value "0001" (ASCII) or 16#30303031.

If no tag was detected while the command was executed, the data telegrams are omitted and only the telegram to indicate the end of the command is sent. <Number of Read/Write Tags> has the value "0000" (ASCII) or 16#30303030.

Enhanced Write Words (EW)

The "Enhanced Write Words" command has the command code 16#1A and performs a permanent write operation to the user data area of a tag within the sensing range. A separate data telegram with the status value 16#00 indicates a successful write operation for each tag described. The data telegram contains the read-only code of the tags on which the user data area was written. In addition, for each tag described, 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 sensing zone is indicated by a telegram containing the read-only code of the tag. This telegram has the status value 16#05. A Quit command stops the command execution.

The amount of user data available on a tag depends on the chip type used and can vary in size. For more detailed information, see the overview of tag types.

The <ByteAddress> parameter specifies the start address within the user data area. The value of <ByteAddress> 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.

Command:

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	<FrameLength>							
Byte 2	Fragmentation counter	<Fragmentation Counter>							
Byte 3	Telegram length (high byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram length (low byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#1A							
Byte 6	Parameter/data	<Byte Address (High Byte)>							
Byte 7	Parameter/data	<Byte Address (Low Byte)>							
Byte 8	Parameter/data	<Number of Bytes (High Byte)>							
Byte 9	Parameter/data	<Number of Bytes ((Low Byte)>							
Byte 10	Parameter/data	<Write Data Byte 1>							
Byte 11	Parameter/data	<Write Data Byte 2>							
Byte 12	Parameter/data	<Write Data Byte 3>							
Byte 13	Parameter/data	<Write Data Byte 4>							
etc.	Parameter/data	etc.							
etc.	Parameter/data	<Write Data Byte X>							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.23

The value of <FrameLength> depends on the number of bytes to be written. The fragment ends with the <Write Data Byte X> byte. The <FragmentationCounter> has the value 16#00 because the command telegram can be transmitted within one fragment. The length of the command in bytes (<TelegramLength>) depends on the number of bytes to be written. The telegram ends with the <Write Data Byte X> byte.

The <Command> command code for the "Enhanced Write Words" command is 16#1A.

The <ByteAddress> parameter specifies the start address within the user data area from which the data is written. The <Number of Bytes> parameter is used to specify the number of bytes to be written. This is followed by the information to be written to the tag <Write Data Byte>.

i

Note

Set the non-relevant bytes of the command fragment to the value 16#00.

Reply data telegram data successfully written, long-form data format:

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	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 6	Status	16#00							
Byte 7	Parameter/data	16#00							
Byte 8	Parameter/data	16#08							
Byte 9	Parameter/data	<UID Byte 1>							
Byte 10	Parameter/data	<UID Byte 2>							
Byte 11	Parameter/data	<UID Byte 3>							
Byte 12	Parameter/data	<UID Byte 4>							
Byte 13	Parameter/data	<UID Byte 5>							
Byte 14	Parameter/data	<UID Byte 6>							
Byte 15	Parameter/data	<UID Byte 7>							
Byte 16	Parameter/data	<UID Byte 8>							
Byte 17	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.24

The <FrameLength> has a value of 16#11. This specifies the length of the fragment up to and including the <UID Byte 8> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#0E and specifies the length of the response telegram starting from the telegram length itself up to and including the <UID Byte 8> byte.

The <Command> parameter has the value 16#1A and is mirrored within the data telegram. The <Status> parameter has the value 16#00 and signals that access to the tag was successful. This is followed by a length specification for the read-only code. It is always 2 bytes in size and a value of 16#0008. This is followed by the read-in read-only code to uniquely identify the tag that has been read.

All other bytes of the response fragment are not relevant and have a value of 16#00.

Response signal strength tag response, long-form data format:

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#1A							
Byte 6	Status	16#0B							
Byte 7	Parameter/data	16#01							
Byte 8	Parameter/data	<RSSI value>							
Byte 9	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.25

The <FrameLength> is constant at the value 16#09. The fragment extends up to and including the <RSSI value> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#06.

The <Command> byte is mirrored and has the value of 16#1A as in the command telegram. The <Status> for the telegram that indicates the RSSI value is 16#0B. The first byte <Parameter / Data> has the value 16#01. The second byte <Parameter / Data> describes the <RSSI value>.

All other bytes of the response fragment are not relevant and have a value of 16#00.

Response tag has left sensing zone, long-form data format:

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	Length UID (High Byte)	16#00							
Byte 8	Length UID (Low Byte)	16#08							
Byte 9	Parameter/data	<UID Byte 1>							
Byte 10	Parameter/data	<UID Byte 2>							
Byte 11	Parameter/data	<UID Byte 3>							
Byte 12	Parameter/data	<UID Byte 4>							
Byte 13	Parameter/data	<UID Byte 5>							
Byte 14	Parameter/data	<UID Byte 6>							
Byte 15	Parameter/data	<UID Byte 7>							
Byte 16	Parameter/data	<UID Byte 8>							
Byte 17	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.26

The <FrameLength> has a value of 16#11. This specifies the length of the fragment up to and including the <UID Byte 8> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#0E and specifies the length of the response telegram starting from the telegram length itself up to and including the <UID Byte 8> byte.

The <Command> parameter has the value 16#1A and is mirrored within the data telegram. The <Status> parameter has the value 16#05 and signals that the tag has left the sensing zone. This is followed by a length specification for the read-only code. This always has a size of 2 bytes and the value 16#0008. This is followed by the read-in read-only code to uniquely identify the tag leaving the sensing range.

All subsequent bytes within the telegram have the value 16#00.

Single Write Words with Locks (SL)

The "Single Write Words with Lock" command has the command code 16#47 and performs a one-time write operation to the user data area of a tag within the sensing zone. The data is write-protected at the end of the write process, provided the tag offers this function. 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.

A separate data telegram with the status value 16#00 indicates a successful write operation for each tag described. The data telegram contains the read-only code of the tag to which the user data area was written. In addition, for each tag described, 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 final telegram has the status value 16#0F and contains the number of tags that were written during execution of the command.

The <ByteAddress> parameter specifies the start address within the user data area. The value of <ByteAddress> 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.

Command:

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	<FrameLength>							
Byte 2	Fragmentation counter	<Fragmentation Counter>							
Byte 3	Telegram length (high byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram length (low byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#47							
Byte 6	Parameter/data	<Byte Address (High Byte)>							
Byte 7	Parameter/data	<Byte Address (Low Byte)>							
Byte 8	Parameter/data	<Number of Bytes (High Byte)>							
Byte 9	Parameter/data	<Number of Bytes ((Low Byte)>							
Byte 10	Parameter/data	<Write Data Byte 1>							
Byte 11	Parameter/data	<Write Data Byte 2>							
Byte 12	Parameter/data	<Write Data Byte 3>							
Byte 13	Parameter/data	<Write Data Byte 4>							
etc.	Parameter/data	etc.							
etc.	Parameter/data	<Write Data Byte X>							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.27

The value of <FrameLength> depends on the number of bytes to be written. The fragment ends with the <Write Data Byte X> byte. The <FragmentationCounter> has the value 16#00 because the command telegram can be transmitted within one fragment. The length of the command in bytes (<TelegramLength>) depends on the number of bytes to be written. The telegram ends with the <Write Data Byte X> byte.

The command code <Command> for the "Single Write Words" command is 16#47.

The <ByteAddress> parameter specifies the start address within the user data area from which the data is written. The <Number of Bytes> parameter is used to specify the number of bytes to be written. This is followed by the information to be written to the tag <Write Data Byte>.

Note

Set the non-relevant bytes of the command fragment to the value 16#00.

Response data telegram, long-form data format:

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#47							
Byte 6	Status	16#00							
Byte 7	Parameter/data	16#00							
Byte 8	Parameter/data	16#08							
Byte 9	Parameter/data	<UID Byte 1>							
Byte 10	Parameter/data	<UID Byte 2>							
Byte 11	Parameter/data	<UID Byte 3>							
Byte 12	Parameter/data	<UID Byte 4>							
Byte 13	Parameter/data	<UID Byte 5>							
Byte 14	Parameter/data	<UID Byte 6>							
Byte 15	Parameter/data	<UID Byte 7>							
Byte 16	Parameter/data	<UID Byte 8>							
Byte 17	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.28

The <FrameLength> has a value of 16#11. This specifies the length of the fragment up to and including the <UID Byte 8> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#0E and specifies the length of the response telegram starting from the telegram length itself up to and including the <UID Byte 8> byte.

The <Command> parameter has the value 16#47 and is mirrored within the data telegram. The <Status> parameter has the value 16#00 and signals that access to the tag was successful. This is followed by a length specification for the read-only code. It is always 2 bytes in size and a value of 16#0008. This is followed by the read-in read-only code to uniquely identify the tag that has been read.

All other bytes of the response fragment are not relevant and have a value of 16#00.

Response signal strength tag response, long-form data format:

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	Parameter/data	16#01							
Byte 8	Parameter/data	<RSSI value>							
Byte 9	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.29

The <FrameLength> is constant at the value 16#09. The fragment extends up to and including the <RSSI value> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#06.

The <Command> byte is mirrored and has the value 16#47 as in the command telegram. The <Status> for the telegram that indicates the RSSI value is 16#0B. The first byte <Parameter / Data> has the value 16#01. The second byte <Parameter / Data> describes the <RSSI value>.

All other bytes of the response fragment are not relevant and have a value of 16#00.

End of command response, long-form data format:

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#47							
Byte 6	Status	16#0F							
Byte 7	Parameter/data	<Number of Read/Write Tags Byte 1>							
Byte 8	Parameter/data	<Number of Read/Write Tags Byte 2>							
Byte 9	Parameter/data	<Number of Read/Write Tags Byte 3>							
Byte 10	Parameter/data	<Number of Read/Write Tags Byte 4>							
Byte 11	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.30

The <FrameLength> is constant at the value 16#0B in the response for the end of the command of the "Single Read Words." The fragment extends up to and including <Number of Read/Write Tags Byte 4>. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#08.

The <Command> byte is mirrored and has the same value as in the 16#47 command telegram. The <Status> for the telegram that indicates the command end is 16#0F.

The number of identified tags is transmitted within 4 bytes. The number is displayed in ASCII coding.

When identifying a tag, the <Number of Read/Write Tags> has the value "0001" (ASCII) or 16#30303031.

If no tag was detected while the command was executed, the data telegrams are omitted and only the telegram to indicate the end of the command is sent. <Number of Read/Write Tags> has the value "0000" (ASCII) or 16#30303030.

Enhanced Write Words with Lock (EL)

The command "Enhanced Write Words with Lock" has the command code 16#48 and performs a permanent write operation to the user data area of a tag within the sensing zone. The data is write-protected at the end of the write process, provided the tag offers this function. Write protection is permanent and cannot be undone. Write protection is only activated for the memory blocks that have been written to. Data can continue to be written to all other memory blocks.

A separate data telegram with the status value 16#00 indicates a successful write operation for each tag described. The data telegram contains the read-only code of the tags on which the user data area was written. In addition, for each tag described, 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 sensing zone is indicated by a telegram containing the read-only code of the tag. This telegram has the status value 16#05. A Quit command stops the command execution.

The amount of user data available on a tag depends on the chip type used and can vary in size. For more detailed information, see the overview of tag types.

The <ByteAddress> parameter specifies the start address within the user data area. The value of <ByteAddress> 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.

Command:

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	<FrameLength>							
Byte 2	Fragmentation counter	<FragmentationCounter>							
Byte 3	Telegram length (high byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram length (low byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#48							
Byte 6	Parameter/data	<Byte Address (High Byte)>							
Byte 7	Parameter/data	<Byte Address (Low Byte)>							
Byte 8	Parameter/data	<Number of Bytes (High Byte)>							
Byte 9	Parameter/data	<Number of Bytes ((Low Byte)>							
Byte 10	Parameter/data	<Write Data Byte 1>							
Byte 11	Parameter/data	<Write Data Byte 2>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 12	Parameter/data	<Write Data Byte 3>							
Byte 13	Parameter/data	<Write Data Byte 4>							
etc.	Parameter/data	etc.							
etc.	Parameter/data	<Write Data Byte X>							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.31

The value of <FrameLength> depends on the number of bytes to be written. The fragment ends with the <Write Data Byte X> byte. The <FragmentationCounter> has the value 16#00 because the command telegram can be transmitted within one fragment. The length of the command in bytes (<TelegramLength>) depends on the number of bytes to be written. The telegram ends with the <Write Data Byte X> byte.

The <Command> command code for the "Enhanced Write Words" command is 16#48.

The <ByteAddress> parameter specifies the start address within the user data area from which the data is written. The <Number of Bytes> parameter is used to specify the number of bytes to be written. This is followed by the information to be written to the tag <Write Data Byte>.

Note

Set the non-relevant bytes of the command fragment to the value 16#00.

Response data telegram, data successfully written, long-form data format:

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	Length UID (High Byte)	16#00							
Byte 8	Length UID (Low Byte)	16#08							
Byte 9	Parameter/data	<UID Byte 1>							
Byte 10	Parameter/data	<UID Byte 2>							
Byte 11	Parameter/data	<UID Byte 3>							
Byte 12	Parameter/data	<UID Byte 4>							
Byte 13	Parameter/data	<UID Byte 5>							
Byte 14	Parameter/data	<UID Byte 6>							
Byte 15	Parameter/data	<UID Byte 7>							
Byte 16	Parameter/data	<UID Byte 8>							
Byte 17	Not relevant	16#00							
etc.	Not relevant	16#00							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 31	Not relevant	16#00							

Table 8.32

The <FrameLength> has a value of 16#11. This specifies the length of the fragment up to and including the <UID Byte 8> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#0E and specifies the length of the response telegram starting from the telegram length itself up to and including the <UID Byte 8> byte.

The <Command> parameter has the value 16#48 and is mirrored within the data telegram. The <Status> parameter has the value 16#00 and signals that access to the tag was successful. This is followed by a length specification for the read-only code. It is always 2 bytes in size and a value of 16#0008. This is followed by the read-in read-only code to uniquely identify the tag that has been read.

All other bytes of the response fragment are not relevant and have a value of 16#00.

Response signal strength tag response, long-form data format:

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#48							
Byte 6	Status	16#0B							
Byte 7	Parameter/data	16#01							
Byte 8	Parameter/data	<RSSI value>							
Byte 9	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.33

The <FrameLength> is constant at the value 16#09. The fragment extends up to and including the <RSSI value> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#06.

The <Command> byte is mirrored and has the value 16#48 as in the command telegram. The <Status> for the telegram that indicates the RSSI value is 16#0B. The first byte <Parameter / Data> has the value 16#01. The second byte <Parameter / Data> describes the <RSSI value>.

All other bytes of the response fragment are not relevant and have a value of 16#00.

Response tag has left sensing zone, long-form data format:

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	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#0E							
Byte 5	Command	16#48							
Byte 6	Status	16#05							
Byte 7	<Length UID (High Byte)>	16#00							
Byte 8	<Length UID (Low Byte)>	16#08							
Byte 9	Parameter/data	<UID Byte 1>							
Byte 10	Parameter/data	<UID Byte 2>							
Byte 11	Parameter/data	<UID Byte 3>							
Byte 12	Parameter/data	<UID Byte 4>							
Byte 13	Parameter/data	<UID Byte 5>							
Byte 14	Parameter/data	<UID Byte 6>							
Byte 15	Parameter/data	<UID Byte 7>							
Byte 16	Parameter/data	<UID Byte 8>							
Byte 17	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.34

The <FrameLength> has a value of 16#11. This specifies the length of the fragment up to and including the <UID Byte 8> byte. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The <TelegramLength> has the value 16#0E and specifies the length of the response telegram starting from the telegram length itself up to and including the <UID Byte 8> byte.

The <Command> parameter has the value 16#48 and is mirrored within the data telegram. The <Status> parameter has the value 16#05 and signals that the tag has left the sensing zone. This is followed by a length specification for the read-only code. It is always 2 bytes in size and a value of 16#0008. This is followed by the read-in read-only code to uniquely identify the tag leaving the sensing range.

All subsequent bytes within the telegram have the value 16#00.

8.6.2 System Commands

Quit (QU)

The "Quit" command has the command code 16#02 and stops the execution of an active command on the device. This terminates the enhanced read or enhanced write commands. Successful execution of the command is indicated by a telegram with the status value 16#00.

Command:

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	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#03							
Byte 5	Command	16#02							
Byte 6	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.35

The <FrameLength> has the value 16#06 because no other command parameters have to be transferred and the fragment ends after <Command>. The <FragmentationCounter> has the value 16#00 because no additional fragments are required for the transmission of the command telegram. The length of the command telegram in bytes (<TelegramLength>) is 16#03. The <Command> command code for the "Quit" command is 16#02.

End of command response:

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#02							
Byte 6	Status	16#00							
Byte 7	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.36

The length of the <FrameLength> fragment has a constant value of 16#07, because no further parameters are transmitted within the response. <FrameLength> contains all bytes up to and including <Status>. The <Fragmentation Counter> has the value 16#00 because all response data can be transmitted within one fragment. The value of <TelegramLength> is 16#04 and the size of the telegram extends up to and including <Status>.

The <Command> parameter has the value 16#02 and is mirrored within the data telegram. The <Status> parameter has the value 16#00.

All subsequent bytes within the data telegram have the value 16#00.

Version (VE)

The "VE" command has the command code 16#03 and reads out the device firmware version.

Command:

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#03							
Byte 6	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.37

The <FrameLength> has the value 16#06 because no other command parameters have to be transferred and the fragment ends after <Command>. The <Fragmentation Counter> has the value 16#00 because the command telegram can be transmitted within one fragment. The length of the command telegram in bytes (<TelegramLength>) is 16#03, because the command ends with the <Command> byte. The <Command> command code for the Version command is 16#03.

End of command response:

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	<Frame Length>							
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#03							
Byte 6	Status	16#00							
Byte 7	Version byte 1	16#XX							
etc.	etc.	16#XX							
Byte X	Version byte X	16#XX							
Byte 31	Not relevant	16#00							

Table 8.38

Change Tag (CT)

The command "CT" has the command code 16#04 and configures the tag type with which the read/write device is communicating. The delivery status is type "20."

Command:

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#04							
Byte 6	TagType	16#14							
Byte 7	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.39

End of command response:

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#04							
Byte 6	Status	16#00							
Byte 7	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.40

The tag type can also be set via a parameter, see "Change Tag (CT) Tag Type" on page 77.

8.6.3 HF Configuration Commands

The "Read Parameter" and "Write Parameter" configuration commands can be used to read or change the parameters of the device. This allows the behavior of the device to be adjusted using the radio interface.

All parameter values except the parameter "TI" are stored in a non-volatile memory and remain unchanged after a power interruption. The response to a configuration command is a status message from the read/write device. During the read operation, a status message and the corresponding data are received as the response.

A system code is required to access the HF parameters on the device. This distinguishes between other systems in which parameters can be changed. This device uses the system code "Q" (16#51).

8.6.3.1 Basic Command Structure

Read Parameter (RP)

The "Read Parameter" (RP) command has the command code 16#BE and is used to read a parameter from the HF settings.

Command:

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)	<TelegramLength (High Byte)>							
Byte 4	Telegram length (low byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#BE							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	<ParameterName (High Byte)>							
Byte 8	Parameter name (low byte)	<ParameterName (Low Byte)>							
Byte 9	Length parameter (high byte)	<ParameterLength (High Byte)>							
Byte 10	Length parameter (low byte)	<ParameterLength (Low Byte)>							
Byte 11	Parameter data byte 1	<Parameter Data Byte 1>							
Byte 12	Parameter data byte 2	<Parameter Data Byte 2>							
etc.	etc.	etc.							
etc.	Parameter data byte X	<Parameter Data Byte X>							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.41

The value of <FrameLength> depends on whether parameter values still need to be transmitted with the "Read Parameter" command. This determines the length of the fragment up to and including <Parameter Data Byte X>. For the vast majority of parameters, no additional parameter values are required when performing read access. In these cases, the fragment ends at <ParameterLength (Low Byte)> and the value is 16#0B.

<FragmentationCounter> has the value 16#00 because the command can be transmitted from the control panel via one fragment.

<TelegramLength> specifies the length of the telegram, starting from the telegram length itself and including the <Parameter Data Byte X> byte. If the command has no additional parameter values, the telegram ends at <ParameterLength (Low Byte)> and the <TelegramLength> for this command is 16#08.

The <Command> byte specifies the command to be executed. The <Command> byte has the value 16#BE to execute the "Read Parameter" command.

The <SystemCode> for the device is 16#51 "Q."

The <ParameterName> parameter specifies the parameter to be read. The value of <ParameterName> corresponds to the 2 characters of the parameter's short name. The entries are case-sensitive.

<ParameterLength> specifies the length of a parameter set within the "Read Parameter" command.

For some parameters, the "Read Parameter" command contains a <ParameterData> parameter set. The length depends on the associated parameter.

Note

Set the non-relevant bytes of the command fragment to the value 16#00.

Response:

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	<FragmentationCounter>							
Byte 3	Telegram length (high byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram length (low byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#BE							
Byte 6	Status	<Status>							
Byte 7	Parameter name (high byte)	<Parameter Data Byte 1>							
Byte 8	Parameter name (low byte)	<Parameter Data Byte 2>							
etc.	etc.	etc.							
etc.	Parameter data byte X	<Parameter Data Byte X>							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.42

The value of <FrameLength> depends on the size of the read parameter. <FrameLength> specifies the length of the fragment up to and including the <Parameter Data Byte X> byte.

<FragmentationCounter> has the value 16#00 because the command response can be transmitted from the device via one fragment.

<TelegramLength> specifies the length of the response telegram, starting from the telegram length itself and including the <Parameter Data Byte X> byte. The value of the telegram length for this command response depends on the length of the read parameter.

The <Command> byte contains the mirrored command code. When executing a "Read Parameter" command, the <Command> byte has the value 16#BE in the response.

The <Status> byte has the value 16#00. It indicates that the command was executed correctly and that this telegram contains the read parameter value. If <Status> has a different value, an error has occurred.

The read <Parameter Byte> parameter values follow. The number of parameter values is variable.

All other bytes of the response fragment are not relevant and have a value of 16#00.

Write Parameter (WP)

The "Write Parameter" (WP) command has the command code 16#BF. This command can be used to change the parameters of the HF setting.

Command:

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)	<TelegramLength (High Byte)>							
Byte 4	Telegram length (low byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#BF							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	<ParameterName (High Byte)>							
Byte 8	Parameter name (low byte)	<ParameterName (Low Byte)>							
Byte 9	Length parameter (high byte)	<LengthParameter (High Byte)>							
Byte 10	Length parameter (low byte)	<LengthParameter (Low Byte)>							
Byte 11	Parameter data byte 1	<Parameter Data Byte 1>							
Byte 12	Parameter data byte 2	<Parameter Data Byte 2>							
etc.	etc.	etc.							
etc.	Parameter data byte X	<Parameter Data Byte X>							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.43

The value of <FrameLength> depends on how many <Parameter Data Byte> parameter values are being transmitted with the "Write Parameter" command. This determines the length of the fragment up to and including <Parameter Data Byte X>.

<FragmentationCounter> has the value 16#00 because the command can be transmitted from the control panel via one fragment.

<TelegramLength> specifies the length of the telegram, starting from the telegram length itself and including the <Parameter Data Byte X> byte.

The <Command> byte specifies the command to be executed. The <Command> byte has the value 16#BF to execute the "Write Parameter" command.

The <SystemCode> for the device is 16#51 ("Q").

The <ParameterName> parameter specifies the parameter to be read. The value of <ParameterName> corresponds to the 2 characters of the parameter's short name.

<LengthParameter> is used to specify the length of a parameter set within the "Write Parameter" command.

The length of the <Parameter Data Byte> parameter set is variable and parameter-dependent.

Note

Set the non-relevant bytes of the command fragment to the value 16#00.

Response:

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#BF							
Byte 6	Status	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.44

The value of the <FrameLength> is 16#07. <FrameLength> specifies the length of the fragment up to and including the <Status> byte.

The <FragmentationCounter> has the value 16#00 because the command response can be transmitted from the device via one fragment.

<TelegramLength> specifies the length of the response telegram starting from the telegram length itself and including the <Status> byte. The value of the telegram length for this command response is 16#04.

The <Command> byte contains the mirrored command code. When executing a "Write Parameter" command, the <Command> byte has the value 16#BF in the response.

The "Write Parameter" command was successfully executed if the <Status> has the value 16#00. A different value for <Status> indicates an error.

All other bytes of the response fragment are not relevant and have a value of 16#00.

8.6.3.2 Parameter Overview

With the "Read Parameter" (RP) and "Write Parameter" (WP) commands, you can read/write the following parameters:

Parameter abbreviation	Parameter name/Function	Parameter is readable/writeable
CT 16#4354	See "Change Tag (CT) Tag Type" on page 77 Setting the tag type	Readable/writeable
DR 16#4452	See "Data Rate (DR) Transfer Rate" on page 79 Setting the transfer rate	Readable/writeable
E5 16#4535	See "Number of Unsuccessful Attempts until Status 5 "Enhanced Status 5 (E5)"" on page 80 Setting the number of unsuccessful attempts up to Status 5	Readable/writeable
NT 16#4E54	See "Number of Tags (NT) / Stop Criterion" on page 82 Setting the cancellation criterion for write/read commands	Readable/writeable
OH 16#4F48	See "Operating Hours (OH)" on page 84 Reading the operating hours	Readable
PT 16#5054	See ""Transmit Power" PT" on page 85 Adjust transmit power	Readable/writeable
QW 16#5157	See "Q Value QW" on page 87 Sets the number of time slots for accessing a read/write tag	Readable/writeable

Parameter abbreviation	Parameter name/Function	Parameter is readable/writeable
RD 16#5244	See "Resetting to Factory Setting "Reset to Default (RD)"" on page 89 Reset to Default	Writeable
ST 16#5354	See "Status Query "Status Frontend (ST)"" on page 90 Output of the status of the front end	Readable
TA 16#5441	See "Number of Tries Allowed (TA) Access Attempts" on page 91 Number of Attempts setting	Readable/writeable
TE 16#5445	See "Status Query "Temperature Output (TE)"" on page 93 Output of the internal temperature	Readable
TI 16#5449	See "Setting a Tag ID Filter (TI) Filter" on page 94 Setting a filter to the tag ID	Readable/writeable
TO 16#544F	See "Behavior in the Event of Excess Temperature "Excess Temperature Handling (TO)"" on page 95 Setting the behavior in case of excess temperature	Readable/writeable

Table 8.45

Change Tag (CT) Tag Type

The CT parameter sets the type of tag with which the read/write device communicates.

Parameter character	CT (16#4354)
Length of CT parameter value	1 byte
Factory setting	16#14
Value range	16#14 ... 16#26; 16#32

For supported read/write tags, please see see chapter 6.1.

For example: Command telegram to change the tag type to the value 16#15 (IQC21)

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#0C							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#09							
Byte 5	Command	16#BF							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#43 "C"							
Byte 8	Parameter name (low byte)	16#54 "T"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#01							
Byte 11	CT parameter	16#15							
Byte 12	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.46

For example: Command telegram to read out the tag type

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#BE							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#43 "C"							
Byte 8	Parameter name (low byte)	16#54 "T"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#00							
Byte 11	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.47

For example: Response telegram with the set tag type = 20 (16#14)

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#05							
Byte 5	Command	16#BE							
Byte 6	Status	16#00							
Byte 7	CT parameter	16#14							
Byte 8	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.48

Note

The tag type can also be changed using the CT system command see "Change Tag (CT)" on page 72.

Data Rate (DR) Transfer Rate

The DR parameter sets the transfer rate between the read/write device and read/write tag.

Parameter character	DR (16#4452)
Length of DR parameter value	1 byte
Factory setting	16#00
Value range	0 (16#00) = normal [26 kBit/s] 1 (16#01) = fast read mode in accordance with ISO/IEC 15693-2 X2 [53 kbps]

The "fast read mode" transfer rate is supported by

- Pepperl+Fuchs tag types IQC21,-33, -37, and -50, see chapter 6.1
- Tag with access to at least 2 memory blocks

For example: Command telegram to change the transfer rate to the value 16#01 (fast read 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#0C							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#09							
Byte 5	Command	16#BF							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#44 "D"							
Byte 8	Parameter name (low byte)	16#52 "R"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#01							
Byte 11	DR parameter	16#01							
Byte 12	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.49

For example: Command telegram to read the transfer rate

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#BE							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#44 "D"							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 8	Parameter name (low byte)	16#52 "R"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#00							
Byte 11	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.50

For example: Response telegram with the set transmission rate 16#00 (normal)

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#08							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#05							
Byte 5	Command	16#BE							
Byte 6	Status	16#00							
Byte 7	DR parameter	16#00							
Byte 8	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.51

Number of Unsuccessful Attempts until Status 5 "Enhanced Status 5 (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 16#05. Via the telegram with the status 16#05 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.

Parameter character	E5 (16#4535)
Length of E5 parameter value	1 byte
Factory setting	16#05 = 5 unsuccessful read/write attempts
Value range	16#00 ... 16#0A = 0 ... 10 unsuccessful read/write attempts

The value for the E5 parameter can be increased if communication between the tag and the device is unstable. This reduces the number of telegrams with status 16#05 received. In dynamic applications, parameters can be used to bridge gaps in the sensing range without receiving a status 16#05 message if there are minor interruptions in read/write tag communication. The sensing zone appears more homogeneous.

If a large number of read/write tags are detected at the same time during a dynamic application, the receipt of the telegram with status 16#05 can be delayed by increasing the parameter value of E5. This allows the status 16#00 telegrams with the read-in information of the tags to be transmitted first. The STATUS 16#05 telegrams are transmitted with a time delay.

Reducing the parameter value E5 shortens the reaction time of the system when a read/write tag leaves the sensing range. The status 16#05 telegrams are sent quicker.

The transmission of the following telegrams is not affected by the E5 parameter setting and they are transmitted immediately:

- Status 16#00: Execution successful; data read or written

For example: Command telegram to change the E5 settings to a value of 10 (16#0A)

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#0C							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#09							
Byte 5	Command	16#BF							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#45 "E"							
Byte 8	Parameter name (low byte)	16#35 "5"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#01							
Byte 11	E5 parameter	16#0A							
Byte 12	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.52

For example: Command telegram to read the E5 settings

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#BE							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#45 "E"							
Byte 8	Parameter name (low byte)	16#35 "5"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#00							
Byte 11	Not relevant	16#00							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.53

For example: Response telegram with the E5 parameter value (16#05) set

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#08							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#05							
Byte 5	Command	16#BE							
Byte 6	Status	16#00							
Byte 7	E5 parameter	16#05							
Byte 8	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.54

Number of Tags (NT) / Stop Criterion

The NT parameter specifies the number of tags that the device searches for within the sensing range. Each command is repeated according to the Number of Attempts (TA) parameter. If the number of tags found during the repeat operations reaches or exceeds the NT value, all further runs are canceled. The command is canceled, and the data is output.

If the number of tags is set to 255 (= 16#FF), the function is deactivated. This parameter only affects single commands. It does not affect enhanced commands.

Parameter character	NT (16#4E54)
Length of NT parameter value	1 byte
Factory setting	16#FF
Value range	16#01 … 16#14; 16#FF

For example: Command telegram to change "number of tags" to the value 16#01

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#0C							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#09							
Byte 5	Command	16#BF							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#4E "N"							
Byte 8	Parameter name (low byte)	16#54 "T"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#01							
Byte 11	NT parameter	16#01							
Byte 12	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.55

For example: Command telegram to read "number of tags"

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#BE							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#4E "N"							
Byte 8	Parameter name (low byte)	16#54 "T"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#00							
Byte 11	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.56

For example: Response telegram with the set value of "number of tags" = 255 (16#FF)

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#08							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#05							
Byte 5	Command	16#BE							
Byte 6	Status	16#00							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 7	NT parameter	16#FF							
Byte 8	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.57

Operating Hours (OH)

The OH parameter outputs the information about how long the read/write device is in operation and how long a read/write command is running.

Parameter character	OH(16#4F48)
Length of OH parameter value	8 bytes • Byte 1–4: Read/write device operating time in hours[1] • Byte 5–8: Operating time read/write commands in hours[2]

1. IO-Link communication active
2. Read/write command active

For example: Command telegram to read OH

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#BF							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#4F "O"							
Byte 8	Parameter name (low byte)	16#48 "H"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#00							
Byte 11	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.58

For example: Response telegram with the operating hours

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	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#0C							
Byte 5	Command	16#BE							
Byte 6	Status	16#00							
Byte 7	Parameter OH (Byte 1)	16#00							
Byte 8	Parameter OH (Byte 2)	16#00							
Byte 9	Parameter OH (Byte 3)	16#00							
Byte 10	Parameter OH (Byte 4)	16#53 = 83_{dec} hours							
Byte 11	Parameter OH (Byte 5)	16#00							
Byte 12	Parameter OH (Byte 6)	16#00							
Byte 13	Parameter OH (Byte 7)	16#00							
Byte 14	Parameter OH (Byte 8)	16#03 = 3_{dec} hours							
Byte 15	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.59

The operating time of the read/write device is 83_{dec} hours (16#00000053). The duration of the read/write commands is 3_{dec} hours (16#00000003).

"Transmit Power" PT

The parameter PT sets the transmit power or reads out the set transmit power. With the transmit power you can influence the range and the maximum permissible ambient temperature at which the device can be used.

Parameter character	PT (16#5054)
Length of CT parameter value	1 byte
Factory setting	16#0004
Value range	16#0001 (Min) 16#0002 (Eco) 16#0003 (Normal) 16#0004 (Maximum)

By reducing the transmit power, the device can be operated permanently at higher ambient temperatures. The self-heating of the device is reduced. If the device is operated in continuous operation with "enhanced" read/write commands with maximum transmitter radiated power, the ambient temperature is a maximum of 55 °C. In Eco mode, the ambient temperature can be increased to a maximum of 70 °C. In addition to setting the transmit power, the read/write device has additional protection mechanisms that prevent the device from overheating. See "Behavior in the Event of Excess Temperature "Excess Temperature Handling (TO)"" on page 95.

Value	Meaning	Detection range[1]	Ambient temperature
1	Min	90 mm	- 25 … 70 °C [-13 ... 158 °F]
2	Eco	240 mm	- 25 … 70 °C [-13 ... 158 °F]
3	Normal	270 mm	- 25 … 65 °C [-13 ... 149 °F]
4	Maximum	300 mm	- 25 … 55 °C [-13 ... 131 °F]

Table 8.60

1. Reference day: IQC21-50

For example: Command telegram to change the transmit power PT to the value 3 (16#0003)

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#0D							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#0A							
Byte 5	Command	16#BF							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#50 "P"							
Byte 8	Parameter name (low byte)	16#54 "T"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#02							
Byte 11	PT parameter (high byte)	16#00							
Byte 12	PT parameter (low byte)	16#03							
Byte 13	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.61

For example: Command telegram to read the transmit power

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#BE							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#50 "P"							
Byte 8	Parameter name (low byte)	16#54 "T"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#00							
Byte 11	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.62

For example: Response telegram with the set transmit power PT (16#0004 = Maximum)

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#BE							
Byte 6	Status	16#00							
Byte 7	PT parameter (high byte)	16#00							
Byte 8	PT parameter (low byte)	16#04							
Byte 9	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.63

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 character	QW (16#5157)
Length of QW parameter value	1 byte
Factory setting	16#00
Value range	16#00 = a tag 16#01 = two tags 16#02 = multiple tags (approx. 4) 16#03 = many tags (approx. 8) 16#04 = very many tags (approx. 16)

The Q value influences the execution time of write/read commands. Reducing the Q value shortens the execution time for accessing read/write tags.

For example: Command telegram to change the QW setting to a value of 1 (16#01), i.e., 2 tags

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#0C							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#09							
Byte 5	Command	16#BF							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#51 "Q"							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 8	Parameter name (low byte)	16#57 "W"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#01							
Byte 11	QW parameter	16#01							
Byte 12	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.64

For example: Command telegram to read the QW settings

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#BE							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#51 "Q"							
Byte 8	Parameter name (low byte)	16#57 "W"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#00							
Byte 11	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.65

For example: Response telegram with the set value of the QW parameter (16#02; multiple tags, approx. 4)

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#08							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#05							
Byte 5	Command	16#BE							
Byte 6	Status	16#00							
Byte 7	QW parameter	16#02							
Byte 8	Not relevant	16#00							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.66

Resetting to Factory Setting "Reset to Default (RD)"

The RD parameter resets the device to the factory setting. The RD parameter is written in the process. Read access to this parameter is not possible.

Parameter character	RD (16#5244)

Table 8.67

For example: Command telegram for resetting to the factory settings

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	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#BE							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#52 "R"							
Byte 8	Parameter name (low byte)	16#44 "D"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#00							
Byte 11	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.68

Factory setting

Abbreviation	Parameter	Default value
CT	Tag type CT	Auto detect (16#14)
DR	Data Rate	normal (16#00)
E5	Number of unsuccessful attempts up to Status 5	16#05
NT	Search Algorithm Cancellation Criteria	Off (16#FF)
PT	Transmit Power	Maximum (16#0004)
QW	Q Value	Single tag (16#00)
TA	Number of Attempts, "Tries Allowed" TA	16#02

Abbreviation	Parameter	Default value
TO	Behavior at excess temperature "excess temperature handling"	Switch off (16#00)

Table 8.69

Status Query "Status Frontend (ST)"

The ST parameter reads the operating status of the read/write device.

Parameter character	ST (16#5354)
Length of ST parameter value	2 bytes
Value range	16#01 = Interference 16#02 = Detuning due to surrounding metal 16#04 = Excess temperature warning, internal temperature between 80 – 85 °C. 16#08 = Excess temperature error, internal temperature > 85 °C.

For example: Command telegram to read out the status

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#BF							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#53 "S"							
Byte 8	Parameter name (low byte)	16#54 "T"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#00							
Byte 11	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.70

For example: Response telegram with operating status ST

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#BE							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 6	Status	16#00							
Byte 7	ST parameter (high byte)	16#00[1]							
Byte 8	ST parameter (low byte)	16#02[2]							
Byte 9	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.71

1. Current status = no message
2. Stored status since last reading = coil detuned by metal

Number of Tries Allowed (TA) Access Attempts

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

Parameter character	TA (16#5441)
Length of TA parameter value	1 byte
Factory setting	16#02 → 2 access attempts
Value range	16#01 … 16#0A

This parameter affects the execution time of write and read commands. If the tries allowed parameter value is increased, the execution time for a command also increases, because more access attempts are made.

By increasing the parameter value, the reliability for writing and reading tag data can be increased if communication between the device and read/write tag is unstable.

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

For example: Command telegram to change TA to a value of 5

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#0C							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#09							
Byte 5	Command	16#BF							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#54 "T"							
Byte 8	Parameter name (low byte)	16#41 "A"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#01							
Byte 11	TA parameter	16#05							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 12	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.72

For example: Command telegram to read TA

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#BE							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#54 "T"							
Byte 8	Parameter name (low byte)	16#41 "A"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#00							
Byte 11	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.73

For example: Response telegram with the set value 16#02 of TA

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#08							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#05							
Byte 5	Command	16#BE							
Byte 6	Status	16#00							
Byte 7	TA parameter	16#02							
Byte 8	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.74

Status Query "Temperature Output (TE)"

The TE parameter indicates the internal temperature of the read/write device.

Parameter character	TE (16#5445)
Length of TE parameter value	2 bytes • Byte 1: Temperature on the power amplifier • Byte 2: Temperature on the microcontroller

For example: Command telegram to read the temperature

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#BF							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#54 "T"							
Byte 8	Parameter name (low byte)	16#45 "E"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#00							
Byte 11	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.75

For example: Response telegram with the temperature TE

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#BE							
Byte 6	Status	16#00							
Byte 7	TE parameter	16#20[1]							
Byte 8	TE parameter	16#1E[2]							
Byte 9	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.76

1. 32_{dec} °C.
2. 30_{dec} °C.

Setting a Tag ID Filter (TI) Filter

The TI parameter sets a filter to the UID of a read/write tag. The read/write commands are therefore only executed for read/write tags with the UID that corresponds to the set filter. The filter can be set to the entire UID (8 bytes) or the leading bytes.

Parameter character	TI (16#5449)
Value range	0 ... 8-byte UID

Note

The TI parameter is stored in the volatile memory.

For example: Command telegram to change the filter (TI) on UIDs that begin with the value 16#E0040108

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#0F							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#0C							
Byte 5	Command	16#BF							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#54 "T"							
Byte 8	Parameter name (low byte)	16#49 "I"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#04							
Byte 11	TI parameter	16#E0							
Byte 12	TI parameter	16#04							
Byte 13	TI parameter	16#01							
Byte 14	TI parameter	16#08							
Byte 15	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.77

For example: Command telegram to read out the current filter TI

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#BF							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#54 "T"							
Byte 8	Parameter name (low byte)	16#49 "I"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#00							
Byte 11	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.78

For example: Response telegram with the set filter 16#E0080148 = UID starting with the 4 bytes E0.08.01.48

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#BE							
Byte 6	Status	16#00							
Byte 7	TI parameter	16#E0							
Byte 8	TI parameter	16#08							
Byte 9	TI parameter	16#01							
Byte 10	TI parameter	16#48							
Byte 11	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.79

Behavior in the Event of Excess Temperature "Excess Temperature Handling (TO)"

The TO parameter defines the behavior of the read/write device in the event of excess temperature. Setting the parameter can protect the appliance from overheating.

Parameter character	TO (16#544F)
Length of TO parameter value	1 byte
Factory setting	16#00
Value range	16#00 = Switch off 16#01 = Reduce power 16#02 = Reduce duty cycle

Value	Mode	Function
0	Switch off	The device switches off automatically if an internal temperature > 85 °C is measured. If the internal temperature falls below 80 °C, the device switches on automatically.
1	Reduce power	The device automatically reduces the transmitpower when an internal temperature is measured between 80 – 85 °C. The device switches off automatically if an internal temperature > 85 °C is measured.
2	Reduce duty cycle	The device pauses for 0.5 ms – 1s after a transmission cycle when the internal temperature is measured between 80 – 85 °C. The length of the pause time depends on the internal temperature and the duration of the last read/write command. The device switches off automatically if an internal temperature > 85 °C is measured.

Note

In Reduce duty cycle mode, note that the read/write device does not identify any read/write tags during the pause time.

Consider this pause for applications with read/write tags that move dynamically through the sensing range. If necessary, use a different protection mechanism.

For example: Command telegram to change TO to the value 16#02 (Reduce duty cycle)

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#0C							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#09							
Byte 5	Command	16#BF							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#54 "T"							
Byte 8	Parameter name (low byte)	16#4F "O"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#01							
Byte 11	TO parameter	16#02							
Byte 12	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.80

For example: Command telegram to read TO

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#BE							
Byte 6	System code	16#51 "Q"							
Byte 7	Parameter name (high byte)	16#54 "T"							
Byte 8	Parameter name (low byte)	16#4F "O"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#00							
Byte 11	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.81

For example: Response telegram with the set value 16#00 of TO

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#08							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#05							
Byte 5	Command	16#BE							
Byte 6	Status	16#00							
Byte 7	TO parameter	16#00							
Byte 8	Not relevant	16#00							
etc.	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.82

8.7 Error / Status Messages

Status	Meaning
16#00	The command was executed correctly.
16#01	Excess temperature
16#04	Parameter error If this status message is received immediately after the command is sent, a parameter within the command is outside the value range or the telegram structure is incorrect.
16#05	The read/write tag has left the sensing zone.
16#06	Hardware error, e.g., error on self test or device defect.
16#07	Internal device error
16#09	Programmed tag type is not compatible with the connected read/write device.
16#0A	Short-form data format: several tags in the sensing range. Long-form data format: several tags in the sensing range with the same UID.
16#0B	Telegram with additional information This additional information (e.g., RSSI value) is transmitted in a telegram with this status.
16#0E	Buffer overflow The size of the internal telegram memory has been exceeded. The device generated telegrams faster than could be transmitted to the controller. The telegram memory is deleted by inverting the delete bit. In addition, the functionality of the handshake procedure must be checked.
16#0F	Indicates the end of an output in the multiframe protocol.

9 Appendix

9.1 Fault Repair

Index	Description	Solution
1	No blue LED on; only the green LED flashes	Is the auto-start function active?
2	No orange LED if read/write tag is within the sensing range	Is the matching tag type set? Read out the IO-Link parameter Tag Type (Index 106) and compare the result with the read/write tag list. Does the number of bytes fit with the block length of the tag?
3	No flashing green LED	Is the IO-Link configuration of the master correct? Is the IO-Link version V1.1 supported? Is the connection cable connected correctly?
4	Byte 0 of the input data field contains 0x40	This value indicates that the Expert Mode is enabled. In this case, the value of the "Operation Mode" IO-Link parameter is 0x00 (Expert Mode). Change the "Operation Mode" IO-Link parameter to the value 0x80 (Easy Mode). The higher nibble always has the value 0x00 in Easy Mode.
5	The read/write tag IQC33 cannot be read or written	Is the number of bytes within the "Read job" or "Write job" IO-Link parameter a multiple of 8?
6	The read/write tag IQC37 cannot be read or written	The IQC37 is only supported when using the Expert Mode.
7	No change of the read data despite change of the start address	The address is counted by bytes. Please note the correct block length of the read/write tag type: 4 or 8 bytes. The data will be changed when the address is increased by the block length.
8	After setting the tag type IQC33, "invalid command" is displayed	Set the number of bytes to 8 or a multiple thereof.
9	An unknown read/write tag cannot be read	Reset the RFID read/write device to the factory setting. Switch the execution of the read job to the read-only code: Change the IO-Link parameter "Read job." Can the read-only code be read? If the read-only code is readable, the read/write tag can be accessed. You may need to change the tag type; if access is not possible, the read/write tag is incompatible with ISO15693.

9.2 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**	**l**
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.

Explosion Protection

- Intrinsic Safety Barriers
- Signal Conditioners
- FieldConnex® Fieldbus
- Remote I/O Systems
- Electrical Ex Equipment
- Purge and Pressurization
- Industrial HMI
- Mobile Computing and Communications
- HART Interface Solutions
- Surge Protection
- Wireless Solutions
- Level Measurement

Industrial Sensors

- Proximity Sensors
- Photoelectric Sensors
- Industrial Vision
- Ultrasonic Sensors
- Rotary Encoders
- Positioning Systems
- Inclination and Acceleration Sensors
- Fieldbus Modules
- AS-Interface
- Identification Systems
- Displays and Signal Processing
- Connectivity

Pepperl+Fuchs Quality
Download our latest policy here:

www.pepperl-fuchs.com/quality

PEPPERL+FUCHS

www.pepperl-fuchs.com

/ DOCT-8162A