IUT-F191-IO-V1-FR*

UHF read/write station with integrated IO-Link interface

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 6
1.1 Content of this Document 6
1.2 Target Group, Personnel 6
1.3 Intended Use 6
1.4 Symbols Used 7
1.5 Definitions 7

2 Certificates and approvals 10
2.1 Declaration of Conformity (RE Directive 2014/53/EU) 10
2.2 FCC Information 10
2.3 IC-Information 11
2.4 IFT Information 11
2.5 NCC-Information 11
2.6 ANATEL-Information 12
2.7 UL Information 12
2.8 Other Country-Specific Approvals 12

3 Product Description 13
3.1 RFID Frequency Bands 13
3.2 UHF general 13
3.2.1 Advantages of UHF 13
3.2.2 Applications for UHF systems 13
3.2.3 Memory Structure of a Tag in Accordance with EPC Gen 2 (ISO/IEC 18000-63) 14
3.2.4 Elektronic Product Code EPC 17
3.2.5 The Impact of Various Materials on the Detection Range 17
3.2.6 Dense Reader Mode (DRM) 18
3.2.7 Frequency Hopping Spread Spectrum 19
3.2.7.1 China 19
3.2.7.2 USA 19
3.2.8 Relevant Standards for UHF 20
3.3 Countries of Use 20
3.3.1 European Union 21
3.3.2 Australia 21
3.3.3 Brazil 21
3.3.4 China 21
3.3.5 India 22
3.3.6 Canada 22
3.3.7 Malaysia 22
3.3.8 Mexico 22
3.3.9 South Korea 22
3.3.10 Taiwan 23
3.3.11 United States of America 23

3.4 General Functions and Features 24
3.5 Indicators and Operating Elements 25
3.6 Electrical Connections 25
3.7 IO-Link Interface Properties 26
3.8 Accessories 26
3.8.1 Read/Write Tags 26
3.8.2 IO-Link cordset 26

4 Installation 27
4.1 Storage and Transportation 27
4.2 Unpacking 27
4.3 Mounting 27
4.3.1 Room Orientation 29
4.3.2 Minimum Distances 30
4.3.3 Polarization 30
4.4 Connection 31

5 Commissioning 32
5.1 Operating Modes 32

6 Operation 33
6.1 General 33
6.2 Interference Due to Multipath Propagation 33
6.3 Multiple Tags in Sensing Range 34
6.4 Read Algorithm 34

7 Easy Mode 35
7.1 Command Overview 35
7.2 Basic Structure of the Process Data 36
7.2.1 Output Process Data (PLC -> Device) 36
7.2.2 Input Process Data (Device -> PLC) 38
7.2.3 Flow Diagrams 41
7.2.4 Timing 45
7.3 Easy Mode with PACTware 46

8 ExpertMode 51
8.1 Basic Command Process 51
8.2 Legend 51
8.3 Structure of OUTPUT telegram 52

2023-07

8.4 Structure of INPUT telegram 53
8.5 Handshake Procedure 54
8.6 Command Overview 55
8.6.1 Read/Write Commands 57
8.6.2 System Commands 94
8.6.3 Filter Commands 96
8.6.4 UHF Configuration Commands 103
8.6.4.1 Basic Command Structure 104
8.6.4.2 Overview of UHF Parameters 108
8.7 Error / Status Messages 134

9 Service and Maintenance 135

10 Troubleshooting 136

11 Appendix 137
11.1 Dimensions 137
11.2 Examples for Expert Mode 137
11.2.1 Single Read EPC/UII 137
11.2.2 Enhanced Read EPC/UII 138
11.2.3 Single Write EPC/UII 139
11.2.4 Single Read 4-Byte Blocks 141
11.2.5 Enhanced Read 4-Byte Blocks 142
11.2.6 Single Write 4-Byte Blocks 144
11.2.7 Enhanced Write 4-Byte Blocks 145
11.3 ASCII table 147
11.4 Sensing Range 148

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.

Note

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

The documentation comprises the following parts:

- This document
- Datasheet

In addition, the documentation may comprise the following parts, if applicable:

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

2023-07

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

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

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

Binary

Example: 1001_{bin}

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

DECIMAL

Example: 1234

Decimal numbers are shown without additional identification.

HEX

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:

Example:

16#00 (equivalent to 0x00)

Abbreviations and Terminology

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	Protocol for simple data access of a read/write station; no function block required
Expert Mode	Protocol for high-performance data access of the RFID read/write station; the use of a function block is required
Read-only code	Unique and unchangeable number of a read/write tag
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
IUC	Pepperl+Fuchs-specific designation of a UHF read/write tag
ISO/IEC 18000-63	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
RFID read/write station	RFID read/write head with integrated IO-Link interface for contactless data transfer
SIO Mode	Standard IO mode; mode for conventional signal transmission without IO-Link data
PLC	Programmable logic controller; device for controlling a machine or plant
Vendor ID	Identification number of the device manufacturer, Pepperl+Fuchs: 0x01

2023-07

AFI	**A**pplication **F**amily **I**dentifier
CRC	**C**yclic **R**edundancy **C**heck
EIRP	**E**quivalent **I**sotopically **R**adiated **P**ower
EPC	**E**lectronic **P**roduct **C**ode
ERP	**E**ffective **R**adiated **P**ower
FCC	**F**ederal **C**ommunications **C**ommission
FHSS	**F**requency **H**opping **S**pread **S**pectrum
IC	**I**ndustry **C**anada
ISO	**I**nternational **S**tandardization **O**rganization
MDID	**M**ask **D**esigner **Id**entifier
PC	**P**rotocol **C**ontrol
RFID	**R**adio **F**requency **Id**entification
PLC	**P**rogrammable**L**ogic **C**ontroller
Tag	Read / write tag; tag
TID	**T**ag **Id**entification
TMN	**T**ag **M**odel **N**umber
UHF	**U**ltra**h**igh-**F**requency
UII	**U**nique **I**tem **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 30 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 licence-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 35 cm from all persons and must not be co-located or operated in conjunction with any other antenna or transmitter.

2.4 IFT Information

This device complies with Federal Telecommunications Institute IFT standard(s). 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.

Este dispositivo cumple con las normas del Instituto Federal de Telecomunicaciones IFT. La operación de este equipo está sujeta a las siguientes dos condiciones:

1. es posible que este equipo o dispositivo no cause interferencia perjudicial y
2. este equipo o dispositivo debe aceptar cualquier interferencia, incluyendo la que pueda causar su operación no deseada.

2.5 NCC-Information

The company, firm or user shall not

1. change the frequency,
2. increase the power or
3. change the features and functions of the original design of the certified low power RF equipment without approval.

The use of low-power RF equipment shall not affect flight safety or interfere with legal communications; if interference is found, it shall be immediately discontinued and improved until there is no interference before continued use.

The aforementioned legal communication refers to the radio communication operated in accordance with the Telecommunications Control Law. Low-power RF equipment shall tolerate interference with legal communications or radioactive electrical equipment for industrial, scientific and medical use.

取得審驗證明之低功率射頻器材, 非經核准, 公司、商號或使用者均不得擅自變更頻率、加大功率或變更原設計之特性及功能。低功率射頻器材之使用不得影響飛航安全及干擾合法通信；經發現有干擾現象時, 應立即停用, 並改善至無干擾時方得繼續使用。前述合法通信, 指依電信管理法規定作業之無線電通信。低功率射頻器材須忍受合法通信或工業、科學及醫療用電波輻射性電機設備之干擾。

2.6 ANATEL-Information

This equipment is not entitled to protection against harmful interference and may not cause interference in duly authorized systems. For more information, consult the ANATEL website www.anatel.gov.br

Este equipamento não tem direito à proteção contra interferência prejudicial e não pode causar interferência em sistemas devidamente autorizados. Para maiores informações, consulte o site da ANATEL – www.anatel.gov.br

2.7 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 -25 °C ... +70 °C for operation with non-transmission periods, or -25 °C ... +60 °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.

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.

2.8 Other Country-Specific Approvals

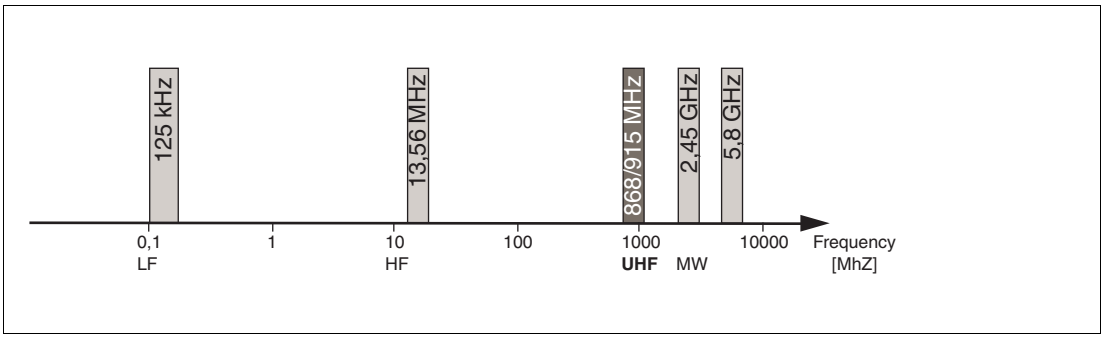
All currently valid approvals can be found in the datasheet for your device at www.pepperl-fuchs.com.

2023-07

3 Product Description

3.1 RFID Frequency Bands

The following diagram shows the different frequency bands used for RFID. The devices described in this manual operate in the frequency range from 865 MHz … 868 MHz, and from 902 MHz … 928 MHz, which is highlighted.



- 100 kHz … 135 kHz: low frequency LF
- 13.56 MHz: high frequency HF
- 865 MHz … 868 MHz (Europe), 902 MHz … 928 MHz (USA), 920 MHz … 925 MHz (China): ultra-high frequency UHF
- 2.45 GHz and 5.8 GHz: microwave MW

3.2 UHF general

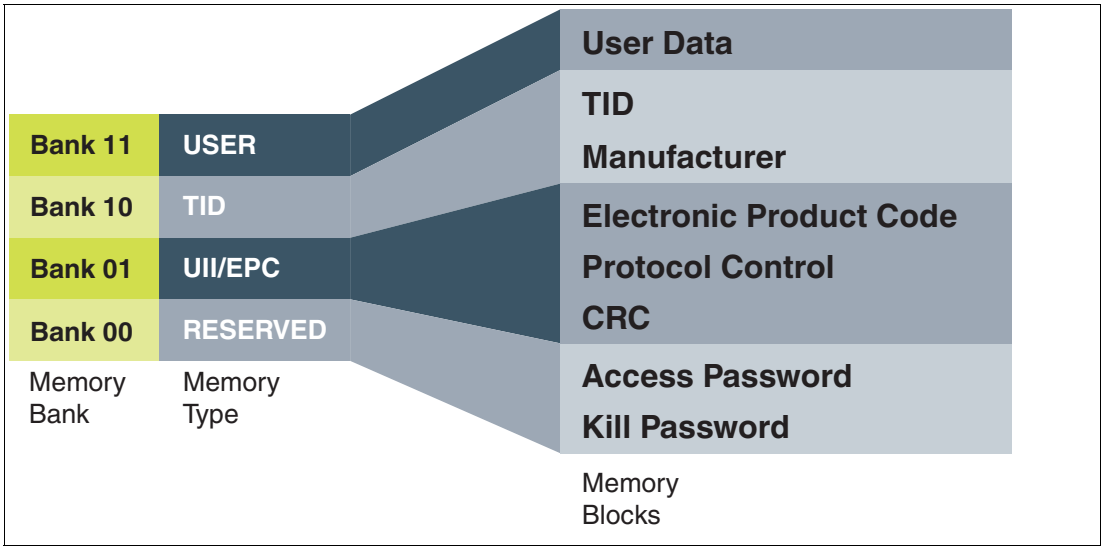
3.2.1 Advantages of UHF

- Long detection range
- UHF tags are available as cheap and space-saving adhesive labels
- High transfer rates
- Tag is available with a large working memory (user memory)
- Bulk detection

3.2.2 Applications for UHF systems

- Identification in galvanic coating or painting systems used in automotive production,
- Identification feasible over greater distances than with LF and HF systems,
- Identification of automotive superstructures in automotive production,
- Pallet identification and measurement of goods movements in the logistics sector, and
- Access control at unloading stations with HGV identification.

3.2.3 Memory Structure of a Tag in Accordance with EPC Gen 2 (ISO/IEC 18000-63)



The memory of an EPC Gen 2 (ISO/IEC 18000-63) tag is split into 4 segments (banks). The main contents of these segments are:

Segment	Function	Length
Bank 00	Password management	Depending on the tag type
Bank 01	Unique Item Identifier (UII) Electronic Product Code (EPC)	Depending on the tag type
Bank 10	Tag ID (TID)	4 bytes (MDID, TMN) + 0 bytes, 4 bytes, or 8 bytes
Bank 11	User memory	Depending on the tag type

Bank 00: Password management

The **Bank 00** segment contains password management information and is accessible only when using **ExpertMode**. It contains the access password and the kill password. To access memory bank 00, the MB (Memory Bank) parameter must first be changed to the value 16#00 (see "Memory Bank (MB)" on page 123). The following commands can be used to access memory bank 00:

- Single/Enhanced Read 4-Byte Blocks (see "Single Read 4-Byte Blocks (SR)" on page 69 and see "Enhanced Read 4-Byte Blocks (ER)" on page 71)
- Single/Enhanced Write 4-Byte Blocks (see "Single Write 4-Byte Blocks (SW)" on page 74 and see "Enhanced Write 4-Byte Blocks (EW)" on page 76)
- Single/Enhanced Read 2-Byte Words (see "Single Read 2-Byte Words (#SR)" on page 79 and see "Enhanced Read 2-Byte Words (#ER)" on page 81)
- Single/Enhanced Write 2-Byte Words (see "Single Write 2-Byte Words (#SW)" on page 84 and see "Enhanced Write 2-Byte Words (#EW)" on page 86)

Bank 01: EPC/UII

The **Bank 01** segment contains the EPC/UII. The segment is divided into several subsections:

- CRC (Cyclic Redundancy Check; checksum) 16 bits
- PC (Protocol Control; PC word) 16 bits
- EPC (Electronic Product Code) or UII (Unique Item Identifier) Variable length

Bytes 0 and 1 of the memory bank contain a checksum (CRC). This is calculated independently by the read/write tag. This subsection can only be read.

The PC word (Protocol Control) is located in bytes 2 and 3. It contains the following information:

- Length of EPC/UII
- Application Family Identifier (AFI)
- Toggle bit to indicate whether it is an EPC or ISO standard code
- Bit to indicate whether data is stored in segment 11 (User Memory)

For a detailed overview of the memory configuration see "Set Filter Mask (FI)" on page 96.

The device can identify multiple read/write tags within the sensing range when using the long-form data format. However, only tags with different EPC/UII codes can be located in the sensing range. An error message will be generated if the system detects another tag with an identical EPC/UII code.

Successful read or write access to a tag is indicated by a response telegram. The EPC/UII information consisting of the PC word and EPC/UII is always located in this response telegram. The CRC section is not transmitted. The EPC/UII information always precedes the read-in data (user memory or TID) in the event of a response to read access if the long-form data format is used.

Memory bank 01 can be accessed in both Easy Mode and Expert Mode. In both modes, information can be read and written. The following commands are used to access memory bank 01 in **ExpertMode**:

- Single/Enhanced Read EPC/UII (see "Single Read EPC/UII (SN)" on page 62 and see "Enhanced Read EPC/UII (EN)" on page 64)
- Single Write EPC/UII (see "Single Write EPC/UII (#SU)" on page 66)

Note

It is not possible to execute a permanent write command (Enhanced Write EPC/UII) because of the risk of programming several tags with identical EPC/UII information.

The MB (Memory Bank) parameter allows you to control another way of accessing Memory Bank 01. If the MB parameter is changed to 16#01, memory bank 01 can be accessed via the following commands:

- Single/Enhanced Read 4-Byte Blocks (see "Single Read 4-Byte Blocks (SR)" on page 69 and see "Enhanced Read 4-Byte Blocks (ER)" on page 71)
- Single/Enhanced Write 4-Byte Blocks (see "Single Write 4-Byte Blocks (SW)" on page 74 and see "Enhanced Write 4-Byte Blocks (EW)" on page 76)
- Single/Enhanced Read 2-Byte Words (see "Single Read 2-Byte Words (#SR)" on page 79 and see "Enhanced Read 2-Byte Words (#ER)" on page 81)
- Single/Enhanced Write 2-Byte Words (see "Single Write 2-Byte Words (#SW)" on page 84 and see "Enhanced Write 2-Byte Words (#EW)" on page 86)

This version enables memory bank 01 to be accessed either in full, including CRC, or only one part of it. Multiple consecutive bytes of the EPC/UII can be read in this way.

Bank 10: TID

The **Bank 10** segment contains the tag identifier (TID). The TID always consists of "Short Tag Identification" (short TID) and "Extended Tag Identification" (XTID) (optional). The short TID has a length of 4 bytes (32 bits) and is structured as follows:

- Class identifier (tag class; 16#E2) 8 bits
- Extended Tag Identification Indicator 1 bit
- Security Indicator and File Indicator 1 bit each
- Mask Designer Identifier (MDID) 9 bits
- Tag Model Number (TMN) 12 bits

If the bit for the extended tag identification indicator is set within the short TID, the short TID is followed by additional bytes containing extra information about the properties of the read/write tag. The XTID has a length of 8 bytes (64 bits) and the following structure:

- XTID 16 bits
- Serial number 48 bits

For a detailed overview of the memory structure of the TID see "Set Filter Mask (FI)" on page 96. Additional information about the TID can be found in ISO/IEC standard 18000-63.

In principle, memory bank 10 can only be read. The TID can be accessed in **EasyMode** and **ExpertMode**. In both cases, the corresponding EPC/UII is prefixed in the device response in addition to the read TID if the long form data format is used. This ensures that the TID is associated to the correct tag.

The following commands are used to access memory bank 10 in **ExpertMode**:

- Single/Enhanced Read read-only code (see "Single Read Fixcode (SF)" on page 57 and see "Enhanced Read Fixcode (EF)" on page 59)

More options to access memory bank 10 are allowed by modifying the Memory Bank MB parameter. If the MB parameter is changed to 16#02, memory bank 10 can be accessed via the following commands:

- Single/Enhanced Read 4-Byte Blocks (see "Single Read 4-Byte Blocks (SR)" on page 69 and see "Enhanced Read 4-Byte Blocks (ER)" on page 71)
- Single/Enhanced Read 2-Byte Words (see "Single Read 2-Byte Words (#SR)" on page 79 and see "Enhanced Read 2-Byte Words (#ER)" on page 81)

This version enables memory bank 10 to be accessed either in full or only one part of it.

Bank 11: User memory

The **Bank 11** segment contains memory to which the user has free access. The size of this memory depends on the chip type, or this memory may not be present.

The tag's user data is located in the user memory. This memory is used when application-specific data must be stored on the tag. When using chips based on EEPROM technology, the number of write accesses may be limited. An almost unlimited number of write accesses is possible when using read/write tags with FRAM memory.

The user memory can be accessed in **EasyMode** and **ExpertMode**. User data can be read and written in both versions. If read or write access to a read/write tag is successful, the response always contains the EPC/UII of the read/write tag if the long-form data format is used. This ensures that the read data or the written data is assigned to the correct tag. In the case of read access, the user memory is preceded by the EPC/UII.

The following commands can be used to access memory bank 11 in **ExpertMode**:

- Single/Enhanced Read 4-Byte Blocks (see "Single Read 4-Byte Blocks (SR)" on page 69 and see "Enhanced Read 4-Byte Blocks (ER)" on page 71)
- Single/Enhanced Write 4-Byte Blocks (see "Single Write 4-Byte Blocks (SW)" on page 74 and see "Enhanced Write 4-Byte Blocks (EW)" on page 76)
- Single/Enhanced Read 2-Byte Words (see "Single Read 2-Byte Words (#SR)" on page 79 and see "Enhanced Read 2-Byte Words (#ER)" on page 81)
- Single/Enhanced Write 2-Byte Words (see "Single Write 2-Byte Words (#SW)" on page 84 and see "Enhanced Write 2-Byte Words (#EW)" on page 86)

The MB parameter must be set to 16#03 (factory setting).

3.2.4 Elektronic Product Code EPC

The electronic product code EPC is a unique identifier in the form of a sequence of numbers. The number sequence has a set structure and a length of 64 bits, 80 bits, 96 bits, or longer, depending on the EPC used. This number sequence is saved to the RFID tag, offering worldwide unique identification of the tagged object.

The EPC was defined by GS1 for use in inventory management. Tags with memory banks for EPC codes must be programmed by the user. The memory of new tags must not contain any valid EPC codes. The EPC numbers are managed and assigned by GS1. To obtain EPC numbers, please contact the GS1 branch in your country (www.gs1.com/contact).

The EPC is defined by GS1 with at present 13 different encoding schemes. SGTIN-96 (serialized global trade item number) is given here as an example of a frequently used encoding scheme. SGTIN-96 has a defined format, and is structured as follows:

1. **Header**: The header specifies the EPC standard used, and denotes the number sequence.
2. **Filter value**: Denotes the unit of the product, for example, end product, additional packaging, pallet.
3. **Partition**: Denotes the point at which the following company prefix ends and the object data begins.
4. **Company Prefix**: Assigned sequence of numbers that identifies the producer.
5. **Object class**: Sequence of numbers that describes the object, e.g., item number.
   The company prefix and the object class are each of variable length, but together are always 44 bits long.
6. **Serial number**: Sequence of numbers that identifies the item, e.g., the sequential serial number of the item.

	Header	Filter value	Partition	Company Prefix	Objekt class	Serial number
Length [Bit]	8	3	3	20 ... 40	4 ... 24	38
Value	48_{dez}	0_{dez}	5_{dez}	4050143_{dez}	124_{dez}	203886_{dez}

Table 3.1

3.2.5 The Impact of Various Materials on the Detection Range

In the UHF frequency band, the properties of the environment, and the quality of the surface on which the tag is mounted strongly affect the system's read/write detection range. The UHF tags must be mounted on the material for which they were designed. For example, glass has a negative impact on the detection range when used as a mounting surface. When a UHF tag is mounted on damp materials, the detection range is worse compared to dry material. The mounting surface material often has a greater influence on the read range than the material between the tag and the read/write device. The graph shows the impact of different materials.

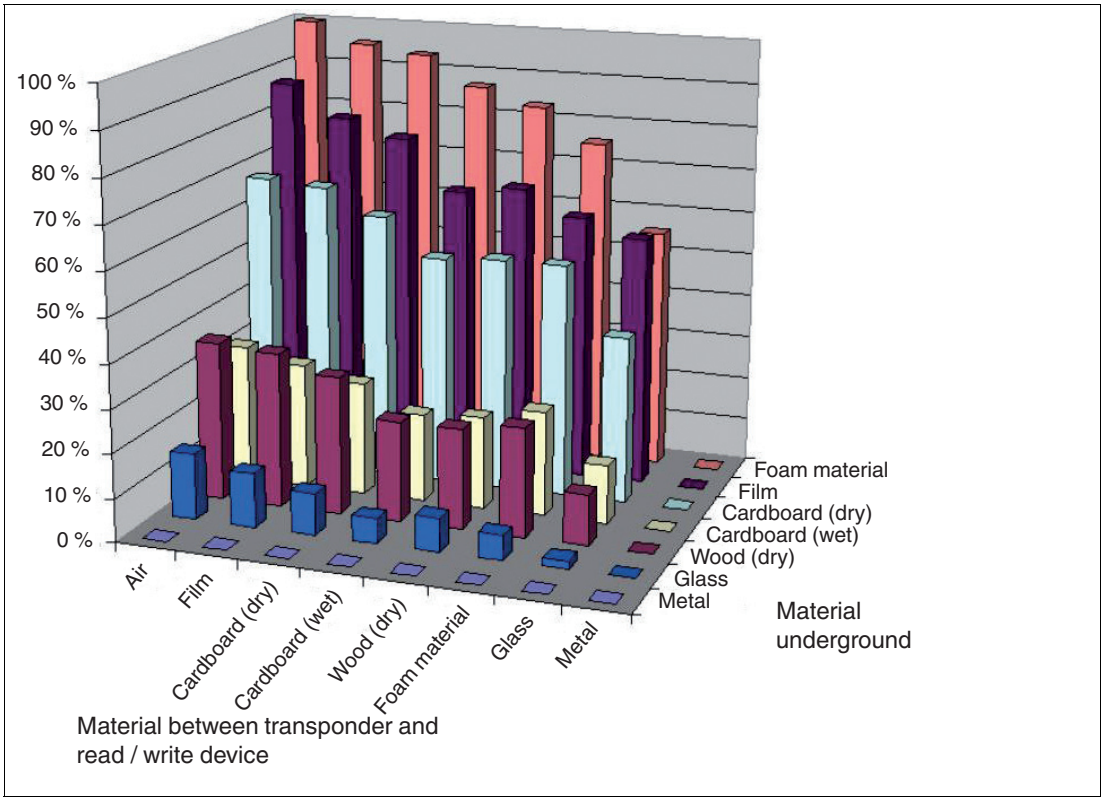


Figure 3.1

There are tags available that are optimized for mounting on metal or on an electrically conductive surface. They can be attached to metal without an additional spacer. These tags contain an "-M-" in the product name (e.g., IUC77-F151-**M**-GBL). If these tags are mounted on plastic material or non-electrically conductive materials, the achievable sensing range can be reduced compared to mounting on metal.

If the read/write tag is mounted on plastic material or non-electrically conductive materials, the abbreviation "-M-" is not included in the product name (e.g., IUC76-50-FR1). These read/write tags can be mounted on cardboard or similar material.

Generally speaking, the sensing range always depends on the mounting surface material. It is therefore impossible to make a generalization about a tag's detection range. Only a statement with restrictions is possible. The detection range achieved with the currently set parameters must be checked at each installation point of a read/write device. The transmission power can be used to affect the sensing range of the read/write device. The transmission power must be set so that the read/write tag can be read securely at the lowest possible power level. An increase in the transmission power can lead to unintentional identification of nearby read/write tags. This unintentional identification of read/write tags in the vicinity of the read/write device should also be checked.

3.2.6 Dense Reader Mode (DRM)

Europe

A special operating mode for read/write tags in accordance with the specification EPC Gen 2 (ISO/IEC 18000-63) allows several read/write devices to be operated close to each other simultaneously without interference.

In accordance with EN 302208, the read/write device uses only channels 4, 7, 10, and 13 in this mode for transmission (read/write head → read/write tag communication path). The transmission power is a maximum of 2 W_{erp} in accordance with EN 302208.

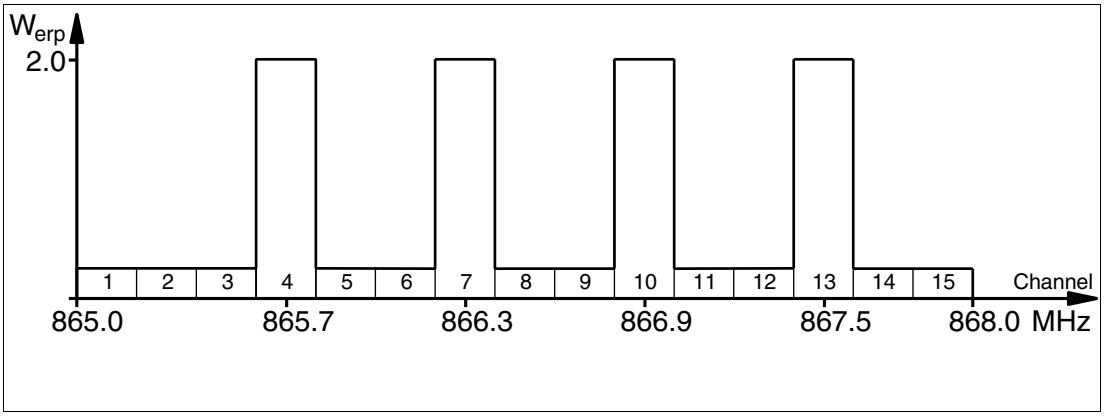


Figure 3.2

The response from the read/write tag appears via the frequency offset, which is achieved by the modulation used in this mode on the two adjacent channels. Due to the high level difference between the transmission channels and the response channels, this technology offers major benefits for reusing frequencies.

3.2.7 Frequency Hopping Spread Spectrum

With FHSS (**F**requency **H**opping **S**pread **S**pectrum), the information to be transmitted is distributed successively through multiple channels. Only one frequency channel is used at any one time. This results in a larger bandwidth for the entire signal, in spite of the fact that each channel has a smaller bandwidth. In this section the channel assignment for China and the USA is shown graphically. For both assignments, different parameters apply, such as channel number and channel bandwidth. Different parameterizations apply in other countries.

3.2.7.1 China

In China, the frequency range 920 MHz ... 925 MHz is available for UHF-RFID read/write devices. The range is split into channels, each with a bandwidth of 250 kHz. A maximum of 2 W_{erp} is permitted on 16 of the available channels. The transmission power is indicated in W_{erp}. FHSS is used with a maximum retention time of two seconds. The UHF RFID read/write device for China uses channel 2 to 17.

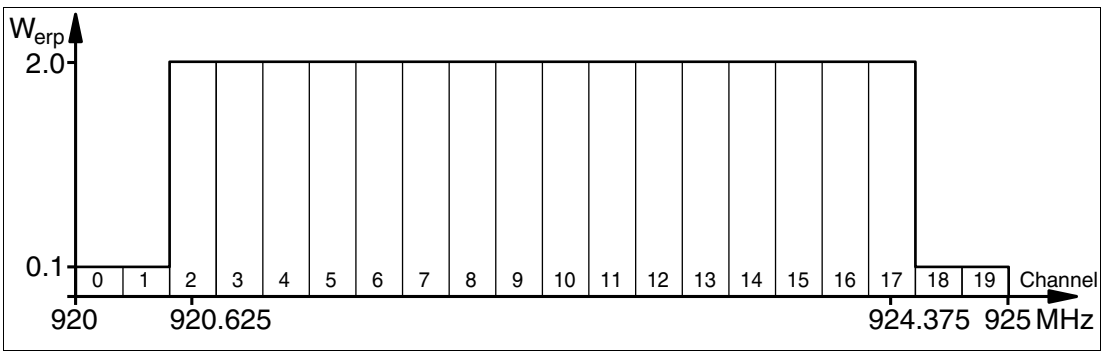


Figure 3.3

3.2.7.2 USA

The ISM band from 902 MHz ... 928 MHz is available in the USA. The band is split into 50 channels, each with a 500 kHz bandwidth. FHSS with a maximum retention time of 0.4 seconds is used. All channels must be used. Channel restriction is not permitted.

In contrast to the read/write devices for Europe and China, the transmission power is indicated in W_{eirp}. A maximum of 4 W_{eirp} is permitted on all channels.

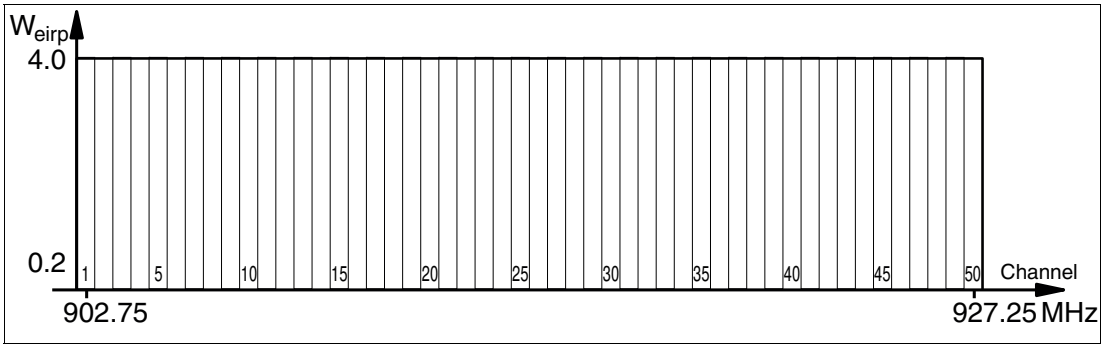


Figure 3.4

3.2.8 Relevant Standards for UHF

European radio standards: EN 300220 and EN 302208

Usage recommendations for RFID type labels, information about recycling, installation of readers and antennae: ISO/IEC TR 24729 parts 1-4

Installation and commissioning of UHF-RFID systems: ETSI TR 102436

Description of air interface: EPC Gen 2 (ISO/IEC 18000-63)

3.3 Countries of Use

Note

Transmission License

A country-specific transmission license is required to operate this device. In the European Union, the manufacturer's declaration of conformity constitutes an adequate license. All current transmission licenses can be found in the datasheet for the relevant device at www.pepperl-fuchs.com.

Note

Country Identifier

All devices are operated within the relevant legal requirements. The country-specific settings are configured during production and cannot be subsequently modified.

Note

If you wish to use the device in a country not included in this chapter, make sure the relevant values for the device are consistent with the local conditions before use.

The frequency access method used is part of the country-specific settings.

Frequency Access Method

- In many countries, including the USA and China, a frequency hopping spread spectrum is used. See chapter 3.2.7. The number and position of the frequencies is fixed and cannot be changed by the user. All channels are used.
- A parameterizable frequency list is used in other countries, including the European Union, Singapore, Vietnam and India. You can compile this frequency list from a specified set of channels. Four channels are specified in the European Union as appropriate for dense reader mode in accordance with EN 302208. See chapter 3.2.6. With this setting, you can configure one, multiple or all four channels.

2023-07

3.3.1 European Union

In the European Union, the use of RFID in the UHF range is regulated by EN 302208.

- UHF band: 865...868 MHz
- Radiated power: 3 mW … 1000 W_{erp}; Default = 100 mW_{erp}
- Channel bandwidth: 200 kHz
- Channel spacing: 600 kHz
- Frequency access method: programmable frequency list
- Number of predefined channels: 4
  Adjustable channels: 4, 7, 10, 13
  Center frequencies: 865.7 MHz, 866.3 MHz, 866.9 MHz, 867.5 MHz
  Up to four channels can be parameterized and used in sequence.
  Default: Dense Reader Mode on channel 4, 7, 10, 13. See chapter 3.2.6.

3.3.2 Australia

In Australia, the use of RFID in the UHF range is regulated as follows:

- UHF band: 920…926 MHz
- Radiated Power: 3 … 150 mW_{eirp}; Default = 150 mW_{eirp}
- Channel bandwidth: 500 kHz
- Channel spacing: 500 kHz
- Frequency access method: Frequency Hopping. See chapter 3.2.7.
- Number of channels: 12
  Channels used: 1, 2, 3, … 12
  Center frequencies: 919,75 MHz + (M x 0,5) MHz
  All 12 channels are always used.

3.3.3 Brazil

In Brazil, the use of RFID in the UHF range is regulated as follows:

- UHF band: 915…928 MHz
- Radiated Power: 3 … 150 mW_{eirp}; Default = 150 mW_{eirp}
- Channel bandwidth: 250 kHz
- Channel spacing: 250 kHz
- Frequency access method: Frequency Hopping. See chapter 3.2.7.
- Number of channels: 52
  Channels used: 1, 2, 3, … 52
  Center frequencies: 914,875 MHz + (M x 0,25) MHz
  All 52 channels are always used.

3.3.4 China

In China, the use of RFID in the UHF range is regulated by the provisions of the China Ministry of Industry and Information Technology (CMIIT).

- UHF band: 920...925 MHz
- Radiated power: 3 mW … 100 W_{erp}; Default = 100 mW_{erp}
- Channel bandwidth: 250 kHz
- Channel spacing: 250 kHz
- Frequency access method: Frequency Hopping (China). See chapter 3.2.7.
- Number of channels: 16
  Channels used: 2, 3, 4, … 17
  Center frequencies: 920.125 MHz + (M x 0.25) MHz
  All 16 channels are always used.

3.3.5 India

In India, the use of RFID in the UHF range is regulated in accordance with EN 302208.

- UHF band: 865…867 MHz
- Radiated Power: 3 … 100 mW_{erp}; Default = 100 mW_{erp}
- Channel bandwidth: 200 kHz
- Channel spacing: 200 kHz
- Frequency access method: programmable frequency list
- Number of channels: 10
  Channels used: 1, 2, 3 ...10
  Center frequencies: 865,1 MHz, 865,3 MHz, 865,5 MHz, 865,7 MHz, 865,9 MHz, 866,1 MHz, 866,3 MHz, 866,5 MHz, 866,7 MHz, 866,9 MHz
  Up to 10 channels can be parameterized simultaneously and used in sequence.
  Default: Dense Reader Mode on channel 1, 4, 7, 10. See chapter 3.2.6.

3.3.6 Canada

The regulations for the UHF frequency range in Canada meet the requirements for the UHF frequency range in the U.S.. See chapter 3.3.11.

3.3.7 Malaysia

In Malaysia, the use of RFID in the UHF range is regulated as follows:

- UHF band: 919…923 MHz
- Radiated Power: 3 … 100 mW_{erp}; Default = 100 mW_{erp}
- Channel bandwidth: 500 kHz
- Channel spacing: 500 kHz
- Frequency access method: Frequency Hopping.see chapter 3.2.7.
- Number of channels: 8
  Channels used: 1, 2, 3, … 8
  Center frequencies: 918,75 MHz + (M x 0,5) MHz
  All 8 channels are always used.

3.3.8 Mexico

The regulations for the UHF frequency range in Mexico meet the requirements for the UHF frequency range in the U.S.. See chapter 3.3.11.

3.3.9 South Korea

In South Korea, the use of RFID in the UHF range is regulated as follows:

- UHF band: 917,2…920,4 MHz
- Radiated Power: 3 … 150 mW_{eirp}; Default = 150 mW_{eirp}
- Channel bandwidth: 200 kHz
- Channel spacing: 600 kHz
- Frequency access method: Frequency Hopping. See chapter 3.2.7.
- Number of channels: 6
  Channels used: 1, 4, 7, 10, 13, 16
  Center frequencies: 917,1 MHz + (M x 0,20) MHz
  All 6 channels are always used.

2023-07

3.3.10 Taiwan

In Taiwan, the use of RFID in the UHF range is regulated as follows:

- UHF band: 920…928 MHz
- Radiated Power: 3 …100 mW_{erp}; Default = 100 mW_{erp}
- Channel bandwidth: 500 kHz
- Channel spacing: 500 kHz
- Frequency access method: Frequency Hopping. See chapter 3.2.7.
- Number of channels: 14
  Channels used: 1, 2, 3, … 14
  Center frequencies: 920,25 MHz + (M x 0,5) MHz
  All 14 channels are always used.

3.3.11 United States of America

In the USA, the use of RFID in the UHF range is regulated in accordance with the provisions set out by the Federal Communications Commission (FCC).

- UHF band: 902 MHz...928 MHz
- Radiated power: 3…150 mW_{eirp}; Default = 150 mW_{eirp}
- Channel bandwidth: 500 kHz
- Channel spacing: 500 kHz
- Frequency access method: Frequency Hopping (USA). See chapter 3.2.7.
- Number of channels: 50
  Channels used: 1, 2, 3, … 50
  Center frequencies: 902.25 MHz + (M x 0.5) MHz
  All 50 channels are always used.

3.4 General Functions and Features

Figure 3.5

Functions

The IUT-F191-IO-V1-FR* devices have been developed for reading and writing passive read/write tags with a UHF operating frequency. Up to 20 read/write tags can be identified within the sensing range if memory bank 01 (EPC/UII) of the tags is different.

The RFID read/write stations are connected to an IO-Link master using an integrated M12 plug

Tags that comply with EPC Gen 2 (IO/IEC 18000-63) are supported.

Sensing Range

The read/write station is specified with a sensing range of up to one meter. The detection range of the device depends on the tags being identified, the set transmission power and the ambient conditions and may therefore be greater or smaller than one meter.

Maximum Frequency Range

The IUT-F191-IO-V1-FR1* read/write stations operate in the frequency range from 865 MHz … 868 MHz. The IUT-F191-IO-V1-FR2* read/write stations operate in the frequency range from 902 MHz … 928 MHz.

2023-07

Features

The read/write station has the following features:

- 3 clearly visible LEDs as function indicator on the front of the device
- Industrial housing with a compact design
- Bulk tag detection
- Connection to the IO-Link master using connector V1 (M12 x 1)
- Protected against electrostatic discharge
- Adjustable UHF parameterization

Integrated antenna

The read/write stations have a circular polarized antenna. These read/write stations can transmit and receive circular polarized waves.

3.5 Indicators and Operating Elements

LED

The device has three LEDs in green/red, blue, and yellow on the front of the device. The LEDs indicate the device health and the activity on the air interface. The various indicators denote:

- Green LED: Green solid: Ready for operation/no IO-Link communication
  Green flashing: IO-Link communication
- Blue LED: Transmission mode
- Yellow LED: Read/write operation successful

3.6 Electrical Connections

Caution!

Cable Specifications

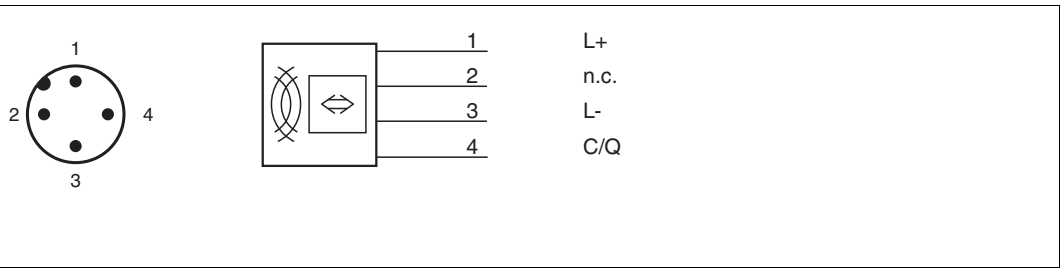
The maximum permissible temperature of the connection cable must be at least +90 °C.

The minimum diameter of the connection cable must be 22 AWG or 0,34 mm^2.

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

The rated voltage of the connection cable must be at least 60V DC.

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



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 regulation, the length of the connection cable must not exceed 20 meters. The RFID read/write device is supplied with power by the IO-Link master. For technical details, see product sheet.

3.7 IO-Link Interface Properties

IO-Link protocol:	V1.1
COM mode:	COM 3
MIN cycle time:	4 ms
Process data length:	32 byte input data / 32 byte output data
SIO mode:	Not supported
Port type:	Type A
Device ID:	0x400201 (IUT-F191-IO-V1-FR1-01) 0x400202 (IUT-F191-IO-V1-FR2-02) 0x400203 (IUT-F191-IO-V1-FR2-03) 0x400204 (IUT-F191-IO-V1-FR1-04) 0x400207 (IUT-F191-IO-V1-FR2-07) 0x400209 (IUT-F191-IO-V1-FR2-09) 0x400210 (IUT-F191-IO-V1-FR2-10) 0x400213 (IUT-F191-IO-V1-FR2-13) 0x400217 (IUT-F191-IO-V1-FR2-17)
Vendor ID:	0x01

3.8 Accessories

3.8.1 Read/Write Tags

Type	Designation
EPC Gen 2 (ISO/IEC 18000-63)	IUC76-F203-M-FRx 10pcs IUC76-F205-M-FRx 10pcs IUC76-F209-M-FRx 10pcs IUC76-28FST-M-FRx IUC76-28ST-M-FRx IUC77-F151-M-GBL 10PCS IUC77-25L100-GBL 1000PCS IUC77-25L110-GBL 1000pcs IUC77-28L90-M-FRx 25pcs IUC77-34-M-FRx 10pcs IUC77-50-FRx 10pcs IUC82-23L50-M-FRx 500pcs

Table 3.2

3.8.2 IO-Link cordset

To connect the RFID read/write station to an IO-Link master, you can use unshielded, three- or four-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 websitehttp://www.pepperl-fuchs.com.

2023-07

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

Warning!

Malfunctions with pacemakers

This device does **not** exceed the permissible limits for electromagnetic fields. Maintain a minimum distance of 30 cm between the device and your pacemaker.

Inadequate distance from the read/write station can result in inhibitions, reprogramming, or incorrect stimulation pulses.

The read/write station is intended for wall mounting or mounting on brackets in indoor areas. Please only install the device using the existing mounting holes in the housing. The preferred mounting direction is with the cable connection facing vertically downward.

Note

Do not route the connection cable in the sensing range of the antenna.

To attach the read/write station, use 3 screws with a diameter of 4 mm, together with flat washers and fastening material 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.

Caution!

Mounting of the read/write station

Make sure that the read/write station is securely attached to the mounting surface.

Note

The installation recommendations made in this document are based on favorable conditions. Pepperl+Fuchs cannot provide any guarantee that the device will function correctly in different environments.

Mounting of the read/write station

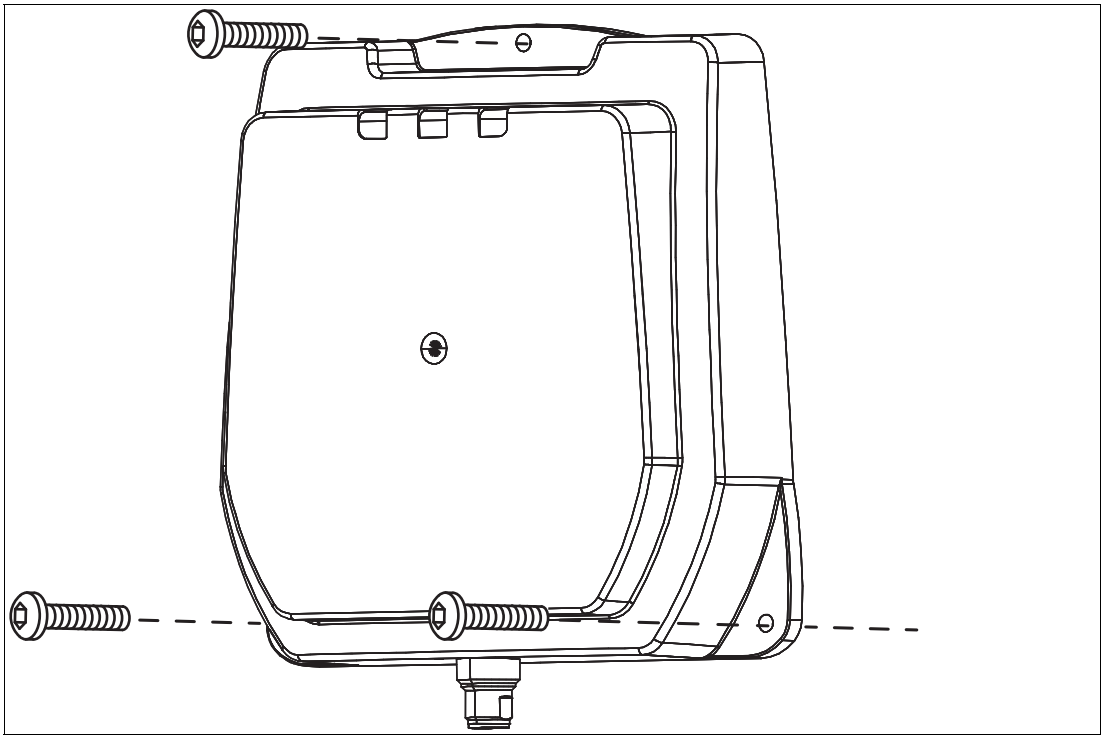
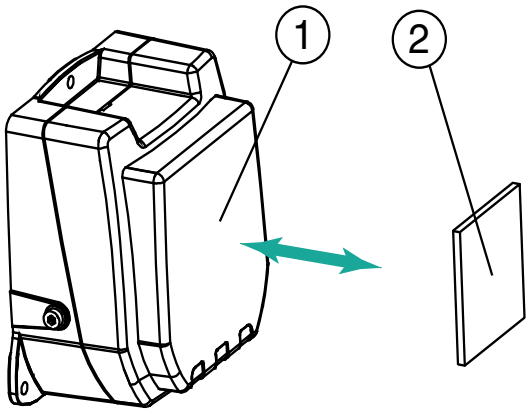


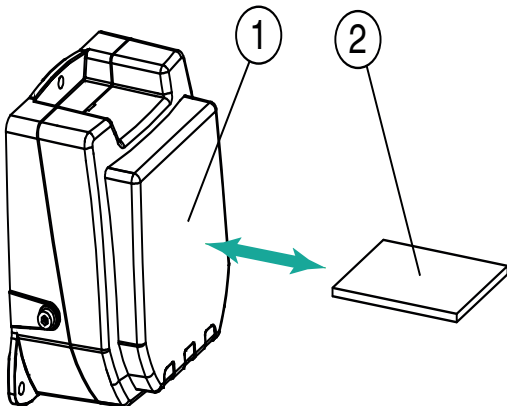
Figure 4.1

4.3.1 Room Orientation

The orientation of the read/write tag antennae in relation to the antennae of the read/write head influences the detection range of the system. Make sure the antennae are oriented parallel to each other.



Optimal tag orientation
- Good communication between the device and tag



Poor tag orientation
- Insufficient communication between the device and tag

1 Device
2 Tag

4.3.2 Minimum Distances

When positioning the read/write device, please observe the minimum distances. The lateral distance between the read/write device and metals or liquids should be at least 50 cm. The distance between the read/write device and the ground should be at least 50 cm.

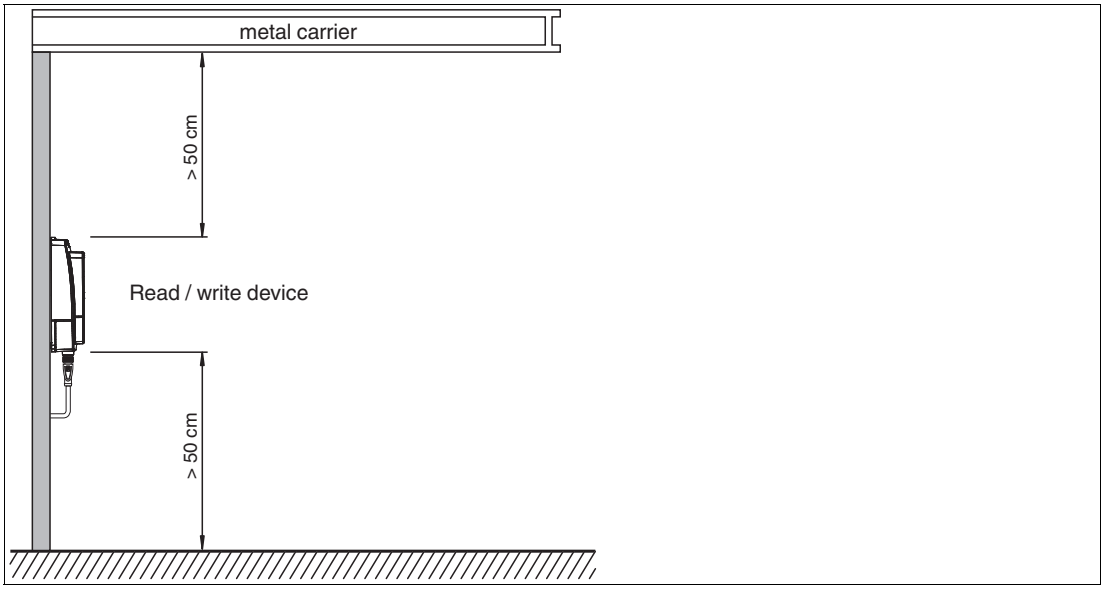


Figure 4.2

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

Note

During mounting, take into account how the read/write devices may cause interference with each other. The further the transmission channels of the read/write devices are from each other, the lower the risk of interference.

4.3.3 Polarization

The polarization of the electromagnetic wave emitted by an antenna depends on the type of antenna and is defined for the electromagnetic field component of the electromagnetic wave. Polarization can be either linear or circular. In the case of an electromagnetic wave with linear polarization, the direction of the vector of the electric field component is spatially constant and therefore dependent on the position of the antenna. Linear polarization can be either vertical or horizontal.

Note

The integrated antenna of the read/write unit has circular polarization. The polarization of the antenna cannot be changed.

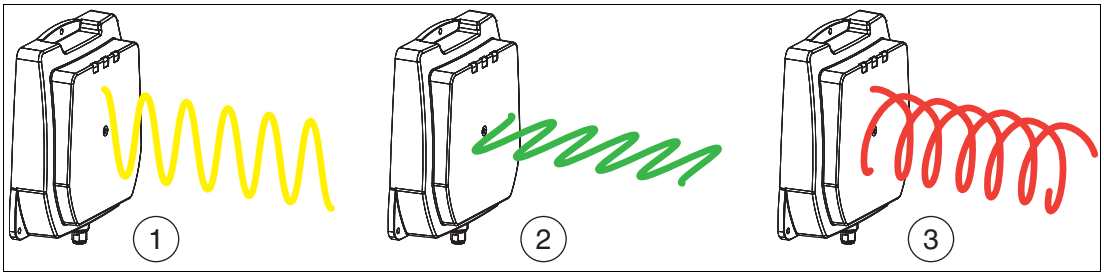


Figure 4.3 1 = Vertical polarization plane
2 = Horizontal polarization plane
3 = Circular polarization

4.4 Connection

Connect the read/write station to an IO-Link master with a cordset.

Warning!
Incorrect electrical connection
Damage to the device or plant caused by incorrect electrical connection.
Check all connections in the plant before commissioning the device.

After connecting the supply voltage, an LED on the device lights up green. If the LED does not light up on the device, the power supply is not connected correctly.

An existing IO-Link connection is indicated by a green flashing LED.

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

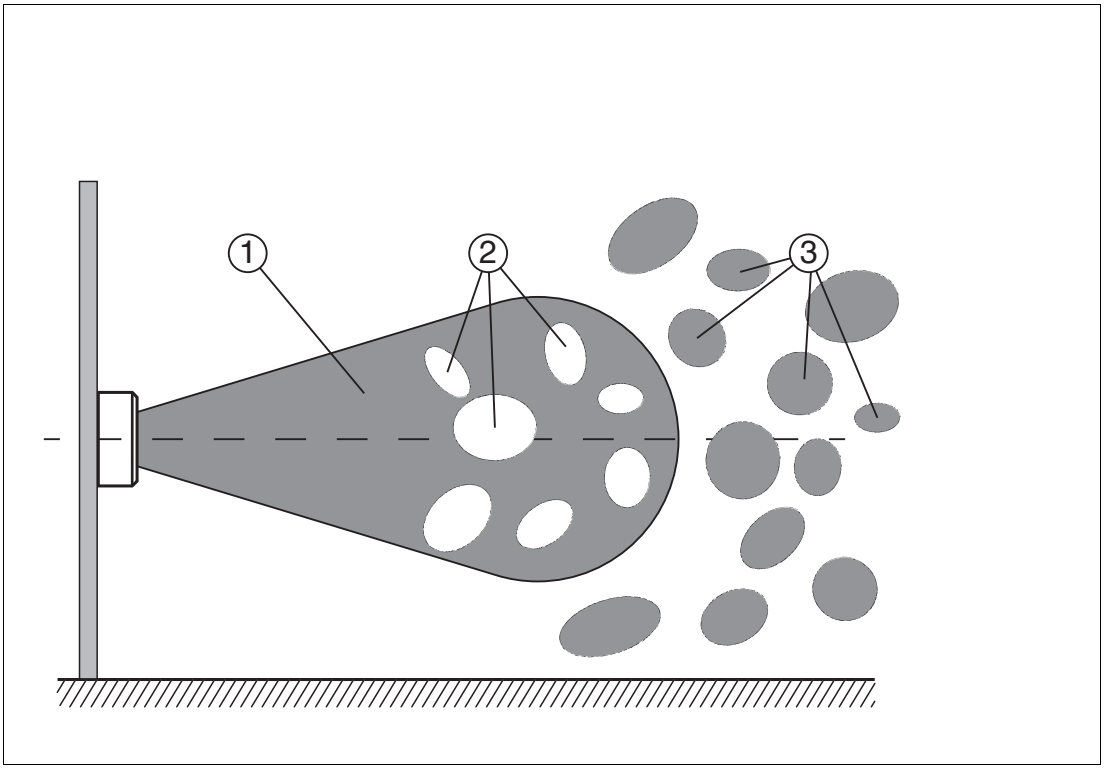
6.1 General

The following sections provide general information about the properties of a UHF RFID system. These properties are independent of the selected operating mode. A more detailed description of the operating modes can be found in the Easy Mode (see chapter 7) and Expert Mode (see chapter 8) sections.

6.2 Interference Due to Multipath Propagation

The electromagnetic waves radiated by the device do not just follow the direct route to the tag, but are reflected off objects in the vicinity. This means that multiple partial waves overlap.

This overlap causes interference in the form of excessive peaks and reductions of the received field strength, leading to complete system degradation. Depending on the environment, several reflections may occur with differing intensity and distance. These different reflections lead to a field strength in the sensing range that is difficult to predict. In the areas of degradation, the prevailing field strength is weaker than the minimum detection field strength of the tag. As a result, the tag cannot be activated for communication. Excessive peaks of the field strength may lead to unwanted excessive detection ranges.



1. Sensing range
2. Degradation
3. Excessive detection ranges

The reflections and the resulting spatial inhomogeneity of the field strength may depend on the frequency used. The absolute value of the field strength depends on the transmission power. Since the tags move within the device sensing range, and the environment can change, it is advisable to repeat the commands at different transmission frequencies and at varying power. Different transmission frequencies are advisable, since the manufacturing tolerances and the immediate environment of the tag have an effect on the tag's resonance frequency.

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 "Set Filter Mask" (**FI**) and the "Activate Filter" (**MF**) commands are used for control. These commands allow you to detect specific tags in the sensing range.see chapter 8.6.3.

6.4 Read Algorithm

To communicate with tags with the maximum possible probability, the device uses an algorithm that varies the transmitting power and frequency. You can set the corresponding values for this algorithm via the parameters power transmit (**PT**), channel frequency (**CD**), or number of channels (**NC**) for the frequency hopping spread spectrum, and number of attempts (**TA**). The number of set attempts is executed for each pairing of power and frequency. This procedure takes a long time, but leads to a high read/write rate. The algorithm runs through all combinations, since it is possible that a tag may be detected only by one specific combination of power and frequency. This applies when the device uses a parameterizable frequency list as the frequency access method. The **CD** parameter defines this frequency list.

If the country identifier specifies a frequency hopping spread spectrum, the channels defined there are used. You can adjust the number of channels used for each attempt via the number of channels parameter (**NC)**.

With the parameter search algorithm cancellation criteria (**NT**), you can specify the number of tags to be processed. If you know the number of tags in the sensing range, you can use this total number as the value for parameter **NT**. If the number of tags found corresponds to or exceeds the value defined in parameter **NT**, the algorithm cancels any further runs to save time.

Tip

If analysis of a specific application shows that a particular frequency and transmission power are sufficient to execute commands successfully, the parameters can be set accordingly, subject to national legislation. This measure reduces the processing time.

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

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

Command	Description
Read EPC/UII	Read the EPC memory bank (bank 01); the 2-byte protocol control (PC) and the EPC/UII are read. • Example: Read/write tags of type IUC76-* usually have an EPC/UII length of 12 bytes plus 2 bytes of PC. • Structure of response data: long-form data format: [length of EPC/UII information; 2 bytes][PC; 2 bytes][EPC/UII; variable length] • Structure of the response data, short-form data format: [PC; 2 bytes][EPC/UII; variable length]
Read TID	The TID (bank 10) is read; in the response data, the EPC/UII information is always prefixed to the TID. • Structure of the response data, long-form data format: [length of the EPC/UII information; 2 bytes][PC; 2 bytes][EPC/UII; variable length][length of the TID information; 2 bytes][TID; variable length] • Response data structure, short-form data format: [TID; variable length]
Read user memory	The user memory bank (bank 11) is read; in the response data, the EPC/UII information of the user memory data is always prefixed. The start address and the amount of data to be read must be configured using the parameters in the IODD file. • Structure of the response data (long-form data format: [length of the EPC/UII information; 2 bytes][PC; 2 bytes][EPC/UII; variable length][length of the user memory information; 2 bytes][user memory; length dependent on parameter settings] • Response data structure, short-form data format: [User Memory; length depends on parameter settings]
Auto-start	Automatic execution of a read command using the parameter settings in the IODD file. One of the read commands described above can be started automatically without activation via the process image. Auto-start is disabled in the factory settings. 8 bytes of user memory are read in from start address 0.
Input representation	Defines the content of the read data. If the long-form data format is used, the UII/EPC and additional length information is always output with a read access to the user memory or the TID.

Command	Description
Write EPC/UII	Writing the EPC/UII bank (bank 01); The data volume and contents to be written must be configured using the parameters of the IODD file. The data to be written begins in the output data field from byte 4. • Structure of data to be written: 1. PC+EPC/UII [PC; 2 bytes][EPC/UII; variable length up to 26 bytes] 2. EPC only, PC is calculated automatically [EPC; variable length up to 28 bytes]
Write user memory	The user memory bank (bank 11) is written; The start address and the amount of data to be written must be configured using the parameters in the IODD file. The data to be written begins in the output data field from byte 4. • Structure of the data: [control byte; 1 byte][not used; 3 bytes][write data; length depends on parameter settings]

Table 7.1 Easy Mode command overview

7.2 Basic Structure of the Process Data

Caution!

The buffer is only designed for a small amount of data to compensate for short interruptions.

In the event of a prolonged interruption of communication or large amounts of data during an interruption, data may be lost.

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

Output process data for which the "valid" flag is not set (i.e. process data marked as invalid) is ignored by the device. If the IO-Link communication is interrupted, the device continues to operate normally. If, for example, a read job is currently active at this time, the data received from the tag is temporarily stored in the device and transferred to the IO-Link master when communication is resumed.

A detailed description of the process data can be found in the IODD of the device. You can find the IODD 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.

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							
...	User data for write job or 16#00 for read job							
31	User data for write job or 16#00 for read job							

Table 7.2

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.
Read: As soon as this bit is set (TRUE), a read job is started using the configuration set by the parameters in the IODD file. The read job runs continuously. To cancel the read job, reset the bit (FALSE).
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 parameters 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.

Example: Write user memory in the TIA portal, long-form data format

Name	Address	Displ...	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_Byte_1"	%QB4	Hex	16#01
"WriteData_Byte_2"	%QB5	Hex	16#02
"WriteData_Byte_3"	%QB6	Hex	16#03
"WriteData_Byte_4"	%QB7	Hex	16#04
"WriteData_Byte_5"	%QB8	Hex	16#05
"WriteData_Byte_6"	%QB9	Hex	16#06
"WriteData_Byte_7"	%QB10	Hex	16#07
"WriteData_Byte_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

Name	Address	Displa...	Monitor value
"ControlByte"	%IB0	Bin	2#0001_0110
"Length"	%IB1	DEC	16
"RSSI"	%IB2	DEC	100
"PowerTransmit_dBm"	%IB3	DEC	17
"Length_EPC/UII_HighByte"	%IB4	Hex	16#00
"Length_EPC/UII_LowByte"	%IB5	Hex	16#0E
"PC_Word_HighByte"	%IB6	Hex	16#34
"PC_Word_LowByte"	%IB7	Hex	16#00
"EPC/UII_Byte_1"	%IB8	Hex	16#30
"EPC/UII_Byte_2"	%IB9	Hex	16#14
"EPC/UII_Byte_3"	%IB10	Hex	16#F7
"EPC/UII_Byte_4"	%IB11	Hex	16#33
"EPC/UII_Byte_5"	%IB12	Hex	16#7C
"EPC/UII_Byte_6"	%IB13	Hex	16#00
"EPC/UII_Byte_7"	%IB14	Hex	16#1F
"EPC/UII_Byte_8"	%IB15	Hex	16#00
"EPC/UII_Byte_9"	%IB16	Hex	16#00
"EPC/UII_Byte_10"	%IB17	Hex	16#00
"EPC/UII_Byte_11"	%IB18	Hex	16#D1
"EPC/UII_Byte_12"	%IB19	Hex	16#A7

Input process data

Byte 0: 0001 0110_{bin}; The "Job active" bit is set to 1. This signals that a write job has been activated. The "Write successful" bit is set to 1. A read/write tag is therefore 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: 16; the byte indicates the length of the transmitted information. This telegram is used to transmit information with a length of 16 bytes. The information starts at byte 4.

Byte 2: 100, RSSI value

Byte 3: 17; transmission power in dBm. 17 dBm = 50 mW

Byte 4/5: 16#000E (14dec); length specification for the EPC/UII information. The EPC/UII information consists of the PC word with a length of 2 bytes plus the EPC/UII. The length of the EPC/UII is variable.
In this example, the length of the EPC/UII is 12 bytes.

Byte 6/7: 16#3400; PC word (Protocol Control)

Byte 8 … Byte 19: EPC/UII of the read/write tag to which the data was written.

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	Transmission power in dBm							
4	Data for read/write job / error code							
5	Data for read/write job / error information							
...	Data for read/write job / error information							
31	Data for read/write job / error information							

Table 7.3

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 and an error description is located in the process data. For a detailed description of the error see chapter 8.7.
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 in percent. The value range is between 0 and 100. 0 = weak signal, 100 = strong signal.

Byte 3 Transmit power of the read/write station in dBm at which the tag was identified.

Byte 4/5 If a read/write tag is accessed successfully, the length of the EPC/UII information (PC word + EPC/UII) is located at this position.[2]

2023-07

Byte 6/7 If a read/write tag is accessed successfully, the PC word (Protocol Control) is in this position.

Byte 8 … Byte 31 Starting at byte 8, the EPC/UII of the read/write tag from which the information was read or to which the information was written. The EPC/UII is always included in the response to a read or write job. This information ensures unique assignment of the read/write tag. When a read job is executed, it is followed by a length specification (2 bytes). This comes from the additional read-in information (TID or user memory).

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 PC word (Protocol Control) and the EPC/UII or the user memory of the tag are transmitted 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 EPC, long-form data format

Name	Address	Displa...	Monitor value
"ControlByte"	%IB0	Bin	2#0001_0101
"Length"	%IB1	DEC	16
"RSSI"	%IB2	DEC	26
"PowerTransmit_dBm"	%IB3	DEC	17
"Length_EPC/UII_HighByte"	%IB4	Hex	16#00
"Length_EPC/UII_LowByte"	%IB5	Hex	16#0E
"PC_Word_HighByte"	%IB6	Hex	16#34
"PC_Word_LowByte"	%IB7	Hex	16#00
"EPC/UII_Byte_1"	%IB8	Hex	16#30
"EPC/UII_Byte_2"	%IB9	Hex	16#14
"EPC/UII_Byte_3"	%IB10	Hex	16#F7
"EPC/UII_Byte_4"	%IB11	Hex	16#33
"EPC/UII_Byte_5"	%IB12	Hex	16#7C
"EPC/UII_Byte_6"	%IB13	Hex	16#00
"EPC/UII_Byte_7"	%IB14	Hex	16#1F
"EPC/UII_Byte_8"	%IB15	Hex	16#00
"EPC/UII_Byte_9"	%IB16	Hex	16#00
"EPC/UII_Byte_10"	%IB17	Hex	16#00
"EPC/UII_Byte_11"	%IB18	Hex	16#D1
"EPC/UII_Byte_12"	%IB19	Hex	16#A7

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 range.

Byte 1: 16; the byte indicates the length of the transmitted information. This telegram is used to transmit information with a length of 16 bytes. The information starts at byte 4.

Byte 2: 26; RSSI value

Byte 3: 17; transmit power in dBm, 17 dBm = 50 mW

Byte 4/5: 16#000E; length specification for the EPC/UII information. The EPC/UII information consists of the PC word with a length of 2 bytes plus the EPC/UII. The length of the EPC/UII is variable.
In this example, the length of the EPC/UII is 12 bytes.

Byte 6/7: 16#3400; PC word (Protocol Control)

Byte 8 … Byte 19: EPC/UII of the read/write tag that was read

For example: Read EPC, short-form data format

Name	Address	Displa...	Monitor value
"ControlByte"	%IB0	Bin	2#0001_0101
"Length"	%IB1	DEC	14
"RSSI"	%IB2	DEC	26
"PowerTransmit_dBm"	%IB3	DEC	17
"PC_Word_HighByte"	%IB4	Hex	16#34
"PC_Word_LowByte"	%IB5	Hex	16#00
"EPC/UII_Byte_1"	%IB6	Hex	16#30
"EPC/UII_Byte_2"	%IB7	Hex	16#14
"EPC/UII_Byte_3"	%IB8	Hex	16#F7
"EPC/UII_Byte_4"	%IB9	Hex	16#33
"EPC/UII_Byte_5"	%IB10	Hex	16#7C
"EPC/UII_Byte_6"	%IB11	Hex	16#00
"EPC/UII_Byte_7"	%IB12	Hex	16#1F
"EPC/UII_Byte_8"	%IB13	Hex	16#00
"EPC/UII_Byte_9"	%IB14	Hex	16#00
"EPC/UII_Byte_10"	%IB15	Hex	16#00
"EPC/UII_Byte_11"	%IB16	Hex	16#D1
"EPC/UII_Byte_12"	%IB17	Hex	16#A7

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 range.
Byte 1:	14; the byte indicates the length of the transmitted information. This telegram is used to transmit information with a length of 14 bytes. The information starts at byte 4.
Byte 2:	26; RSSI value
Byte 3:	17; transmit power in dBm, 17 dBm = 50 mW
Byte 4/5:	16#3400; PC word (Protocol Control)
Byte 6 … Byte 17:	EPC/UII of the read/write tag that was read

For example: Read user memory, long form data format

Name	Address	Displa...	Monitor value
"ControlByte"	%IB0	Bin	2#0001_0101
"Length"	%IB1	DEC	26
"RSSI"	%IB2	DEC	20
"PowerTransmit_dBm"	%IB3	DEC	17
"Length_EPC/UII_HighByte"	%IB4	Hex	16#00
"Length_EPC/UII_LowByte"	%IB5	Hex	16#0E
"PC_Word_HighByte"	%IB6	Hex	16#34
"PC_Word_LowByte"	%IB7	Hex	16#00
"EPC/UII_Byte_1"	%IB8	Hex	16#30
"EPC/UII_Byte_2"	%IB9	Hex	16#14
"EPC/UII_Byte_3"	%IB10	Hex	16#F7
"EPC/UII_Byte_4"	%IB11	Hex	16#33
"EPC/UII_Byte_5"	%IB12	Hex	16#7C
"EPC/UII_Byte_6"	%IB13	Hex	16#00
"EPC/UII_Byte_7"	%IB14	Hex	16#1F
"EPC/UII_Byte_8"	%IB15	Hex	16#00
"EPC/UII_Byte_9"	%IB16	Hex	16#00
"EPC/UII_Byte_10"	%IB17	Hex	16#00
"EPC/UII_Byte_11"	%IB18	Hex	16#D1
"EPC/UII_Byte_12"	%IB19	Hex	16#A7
"Length_Data_HighByte"	%IB20	Hex	16#00
"Length_Data_LowByte"	%IB21	Hex	16#08
"Data_Byte_1"	%IB22	Hex	16#01
"Data_Byte_2"	%IB23	Hex	16#02
"Data_Byte_3"	%IB24	Hex	16#03
"Data_Byte_4"	%IB25	Hex	16#04
"Data_Byte_5"	%IB26	Hex	16#05
"Data_Byte_6"	%IB27	Hex	16#06
"Data_Byte_7"	%IB28	Hex	16#07
"Data_Byte_8"	%IB29	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 range.
Byte 1:	26; the byte indicates the length of the transmitted information. This telegram is used to transmit information with a length of 26 bytes. The information starts at byte 4.
Byte 2:	20; RSSI value
Byte 3:	17; transmit power in dBm, 17 dBm = 50 mW
Byte 4/5:	16#000E; length specification for the EPC/UII information. The EPC/UII information consists of the PC word with a length of 2 bytes plus the EPC/UII. The length of the EPC/UII is variable. In this example, the length of the EPC/UII is 12 bytes.
Byte 6/7:	16#3400; PC word (Protocol Control)
Byte 8 … Byte 19:	EPC/UII of the identified read/write tag
Byte 20/21:	16#0008; length specification for the read-in user memory
Byte 22 … Byte 29:	User memory section read in

2023-07

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

Name	Address	Displa...	Monitor value
"ControlByte"	%IB0	Bin	2#0001_0101
"Length"	%IB1	DEC	8
"RSSI"	%IB2	DEC	33
"PowerTransmit_dBm"	%IB3	DEC	17
"Data_Byte_1"	%IB4	Hex	16#01
"Data_Byte_2"	%IB5	Hex	16#02
"Data_Byte_3"	%IB6	Hex	16#03
"Data_Byte_4"	%IB7	Hex	16#04
"Data_Byte_5"	%IB8	Hex	16#05
"Data_Byte_6"	%IB9	Hex	16#06
"Data_Byte_7"	%IB10	Hex	16#07
"Data_Byte_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 range.
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:	33, RSSI value
Byte 3:	17; transmit power in dBm, 17 dBm = 50 mW
Byte 4 … Byte 11:	User memory section read in

7.2.3 Flow Diagrams

Read Job without Auto-Start Function

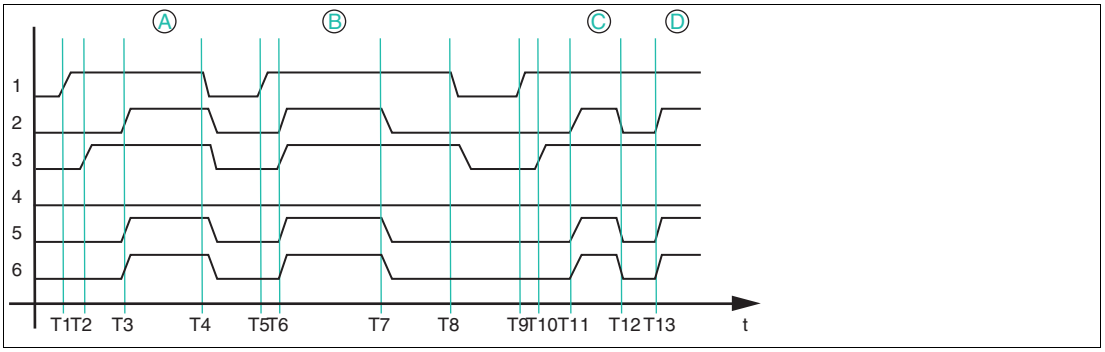


Figure 7.1 Timing sequence of bits in byte 0

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

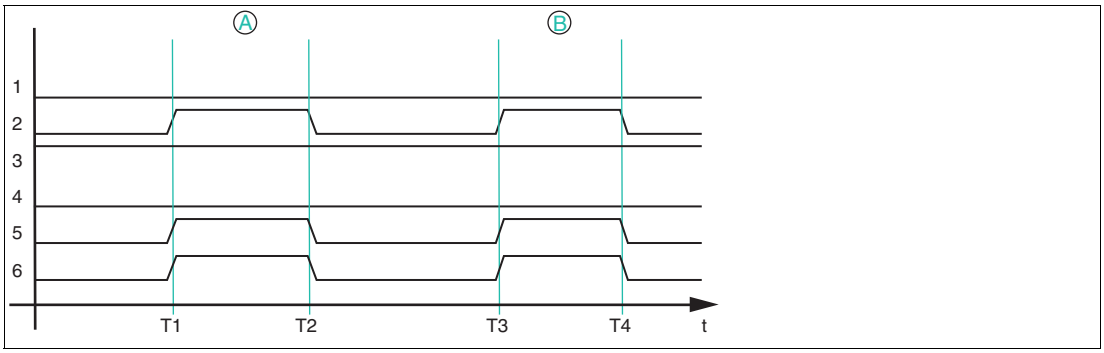


Figure 7.2 Timing sequence of bits in byte 0

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

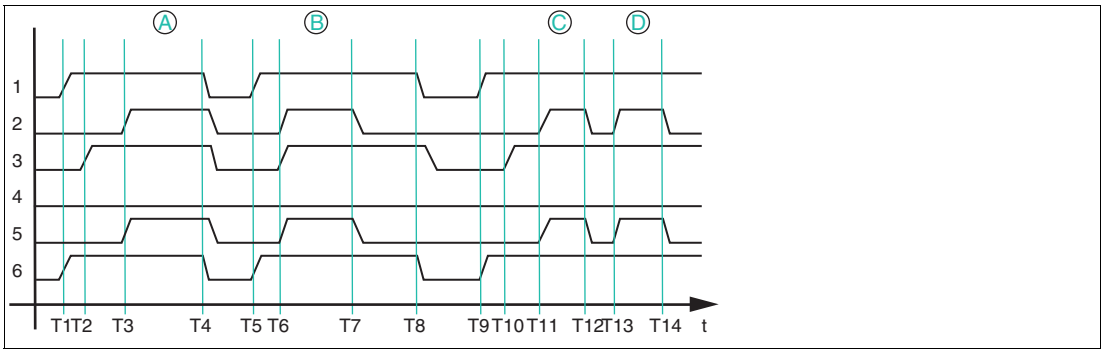


Figure 7.3 Timing sequence of bits in byte 0

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)

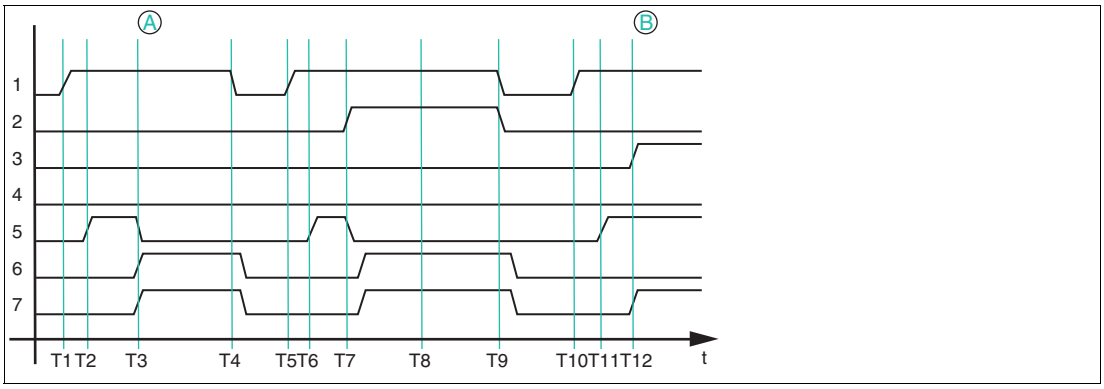
T11: Read/write tag C enters the sensing range and the data is successfully written ("Write successful" and "Tag present" = TRUE); write job is still active ("Job active" = TRUE)

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)

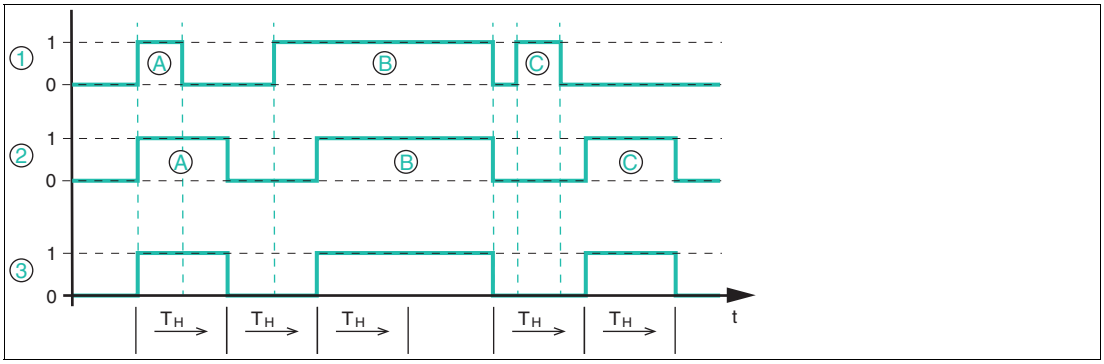
2023-07

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."

7.3 Easy Mode with PACTware

You can commission the RFID read/write station 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 tool 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 station 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 station with the **IODD DTM Configurator** program.

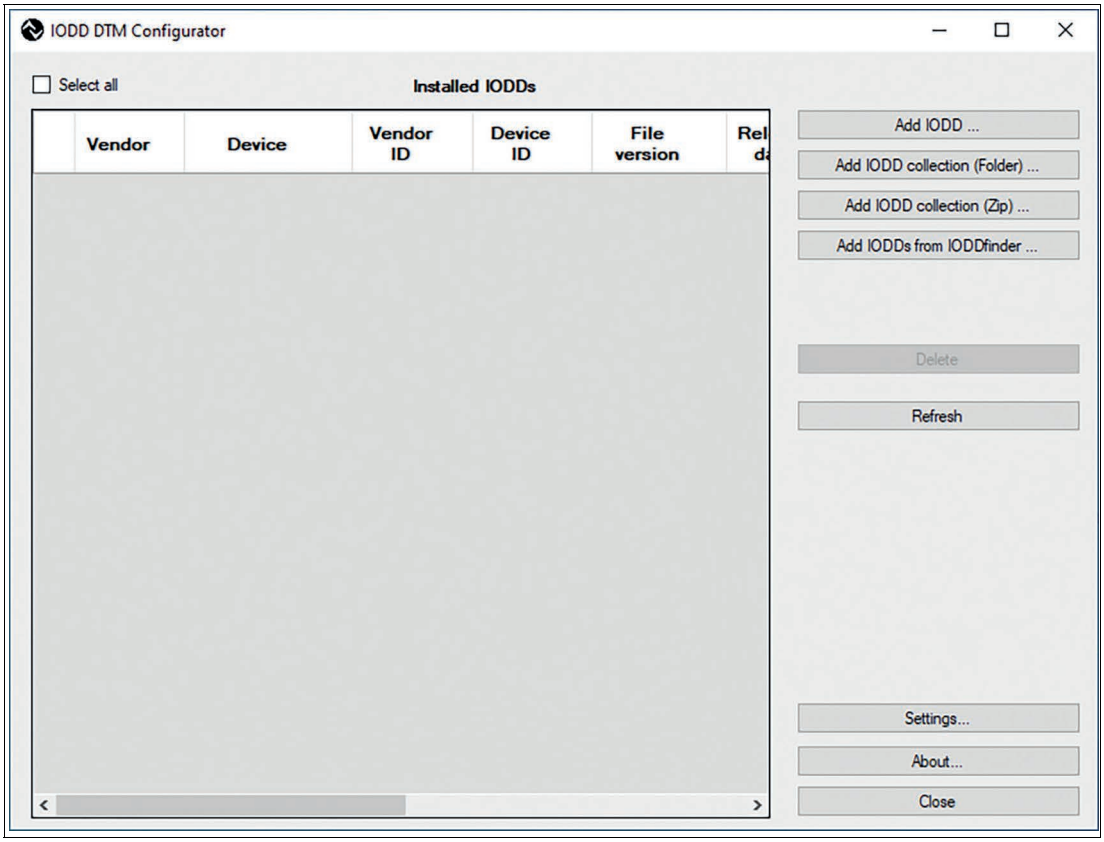


Figure 7.4

↳ Successfully added IODDs appear in "Installed IODDs".

2023-07

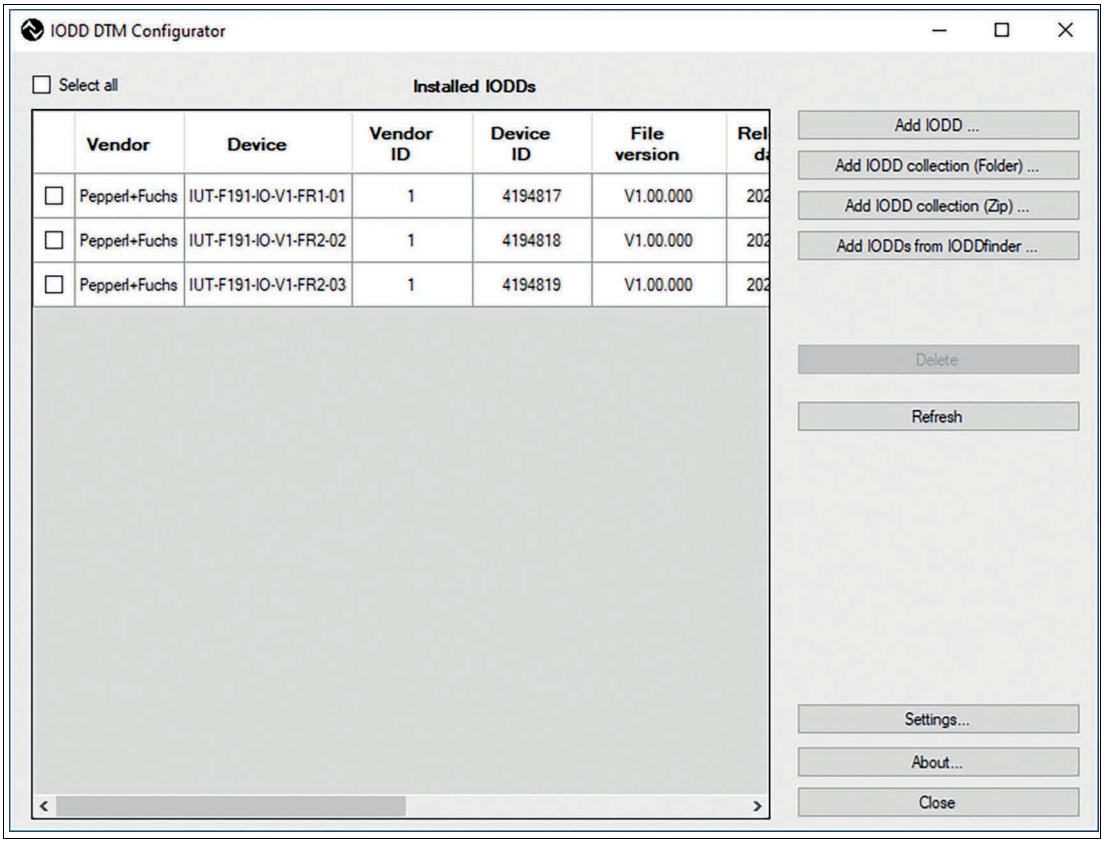


Figure 7.5

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

7. Start PACTware.
8. Right-click on the "HOST PC".

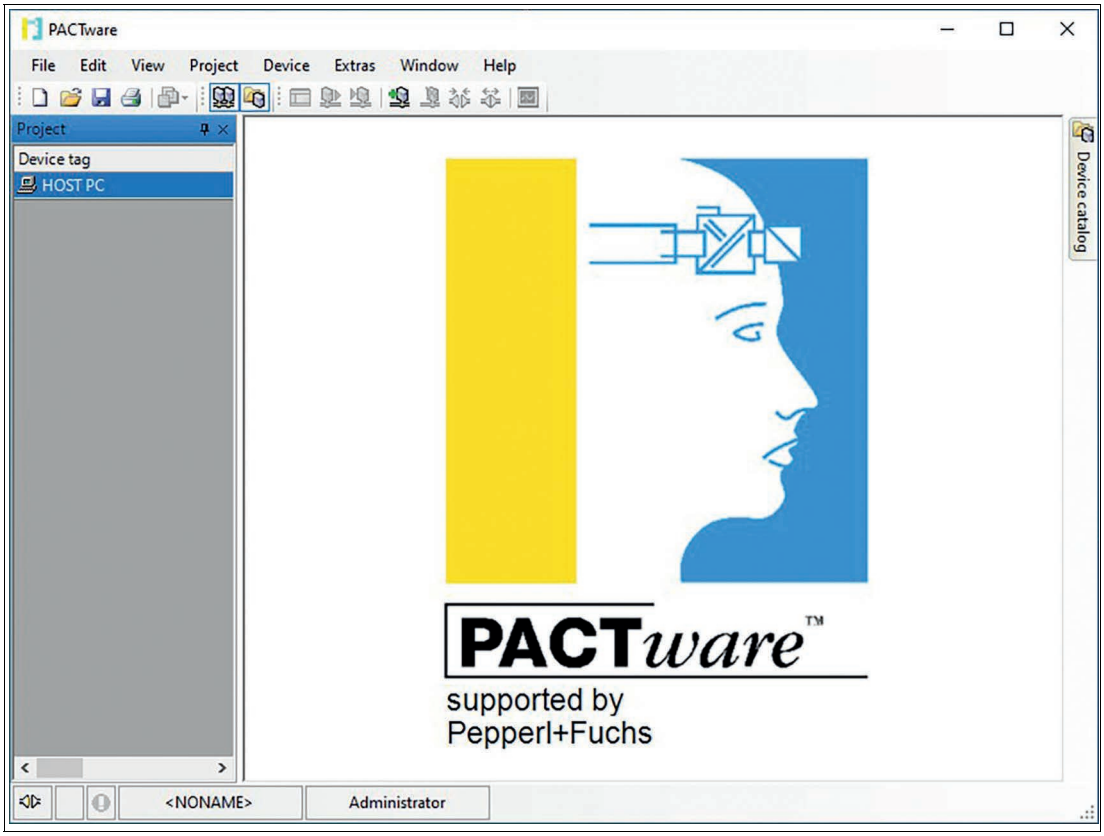


Figure 7.6

↳ The "Device for" window opens.

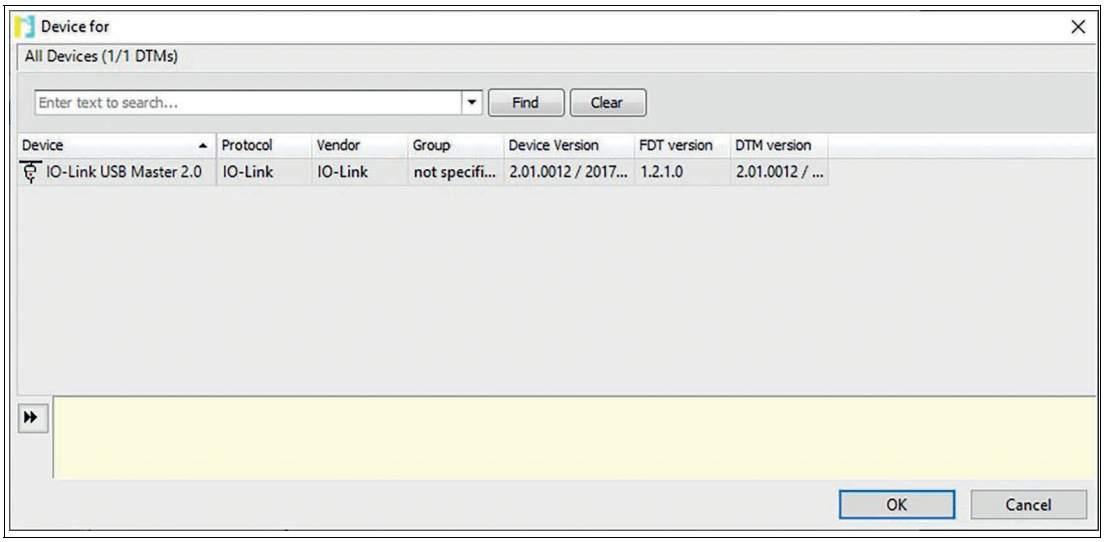


Figure 7.7

9. Select IO-Link USB Master 2.0.

Note

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

10. Confirm with "OK".

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

2023-07

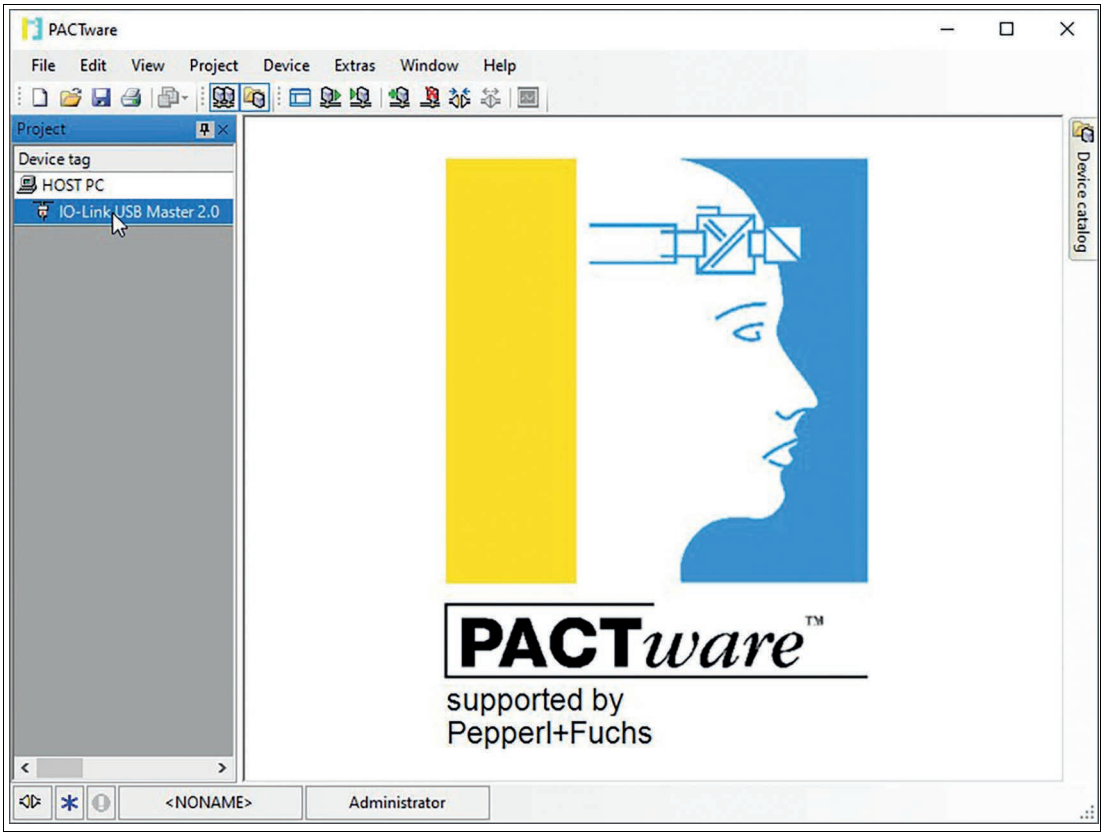


Figure 7.8

11. Right-click on the IO-Link master.

↳ The "Device for" window opens.

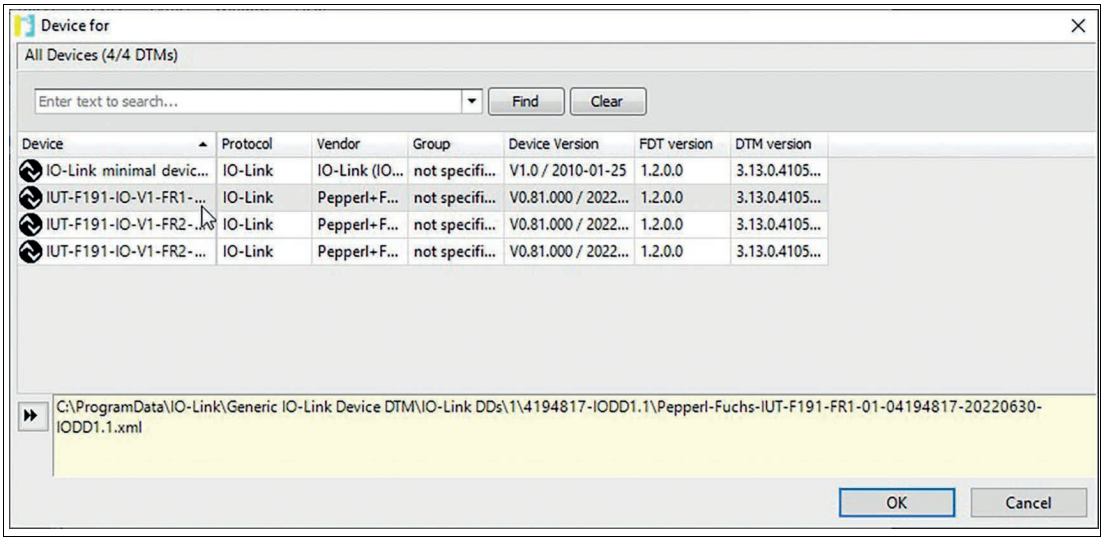


Figure 7.9

12. Select the required RFID read/write station.
13. Confirm with "OK".
14. Double-click the IO-Link device.

↳ The Parameters menu opens.

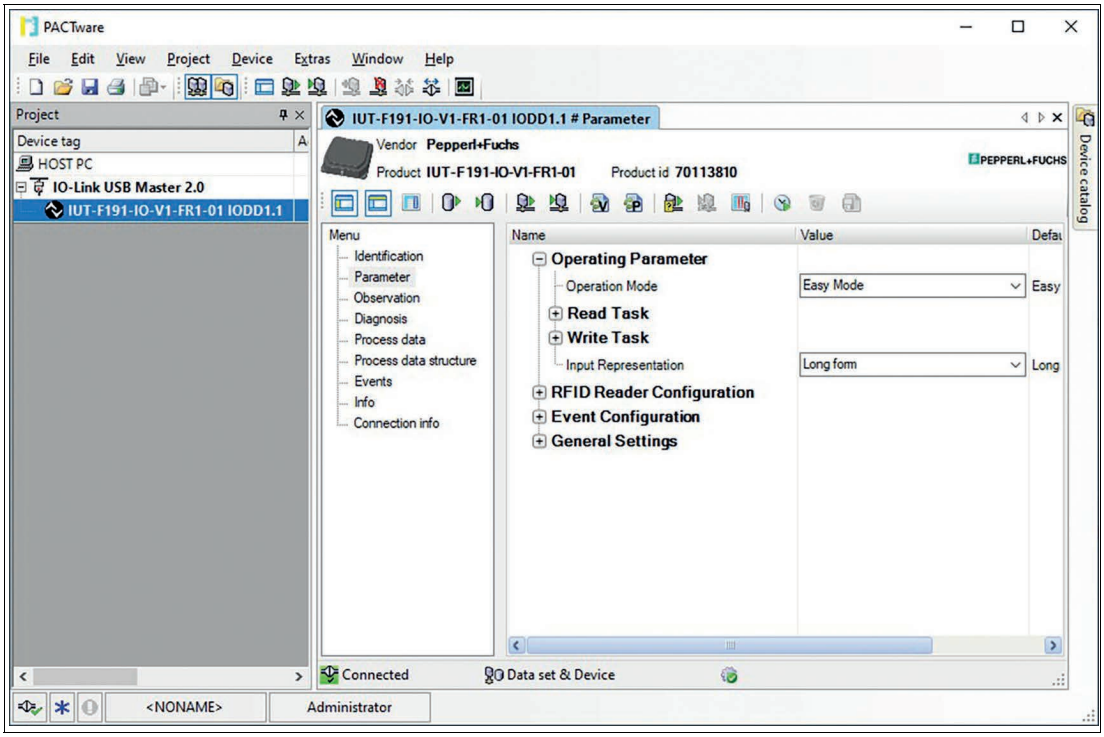


Figure 7.10

15. Adjust the parameters of the device according to your application. See chapter 8.6.4.

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

As soon as a connection is established, the RFID read/write station 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".

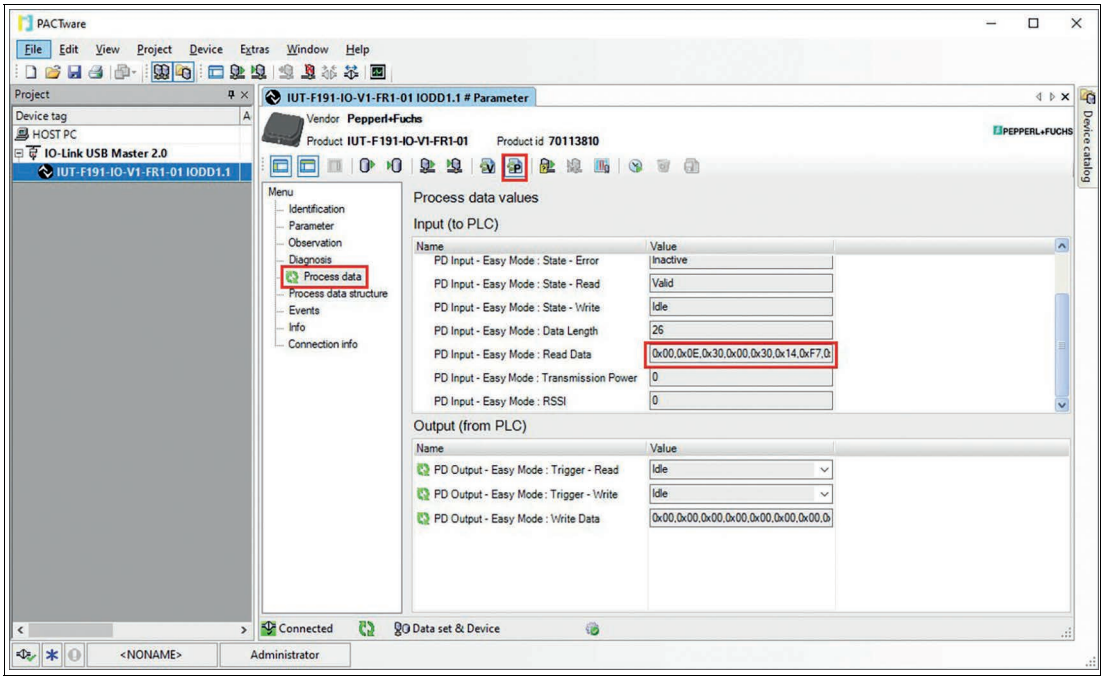


Figure 7.11

Tip

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

2023-07

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 When executing 4-byte block commands, the value must be a multiple of 4. When executing 2-byte word commands, the value must be a multiple of 2.
<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
<EPC/UII>	Variable length	EPC or UII EPC/UII code of the identified read/write tag; length depends on programming; all EPC/UII within the sensing range must be different
<FragmentationCounter>	1 byte	Fragmentation counter Number of fragments still to be transmitted
<FrameLength>	12 bits	Number of valid bytes within the telegram
<Length EPC/UII>	2 bytes	Length of EPC / UII information Length specification includes <PC-Word> and <EPC/UII>
<Length User Data>	2 bytes	Length of user data Specifies the length of the data read in from the user memory bank
<Length TID>	2 bytes	TID length Length specification via the TID of the read-in read/write tag
<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. When executing 4-byte block commands, the value must be a multiple of 4. When executing 2-byte word commands, the value must be a multiple of 2.
<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 UHF parameter to be accessed.

Name	Length	Meaning
<PC-Word>	2 bytes	Protocol control Contains additional information on the EPC/UII coding.
<Status>	1 byte	Status byte Status information about the execution state of the command.
<SystemCode>	1 byte	System identifier The system identifier for the UHF system is "U" (16#55).
<TelegramLength>	2 bytes	Length of telegram including all fragments
<TID>	Variable length	Tag identification 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.

2023-07

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 range
- **System Commands:** Execution of device settings; no access to read/write tags
- **Filter Commands:** Setting of filters to access read/write tags
- **UHF Configuration Commands:** Setting of the device UHF 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>							
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. See chapter 8.7.

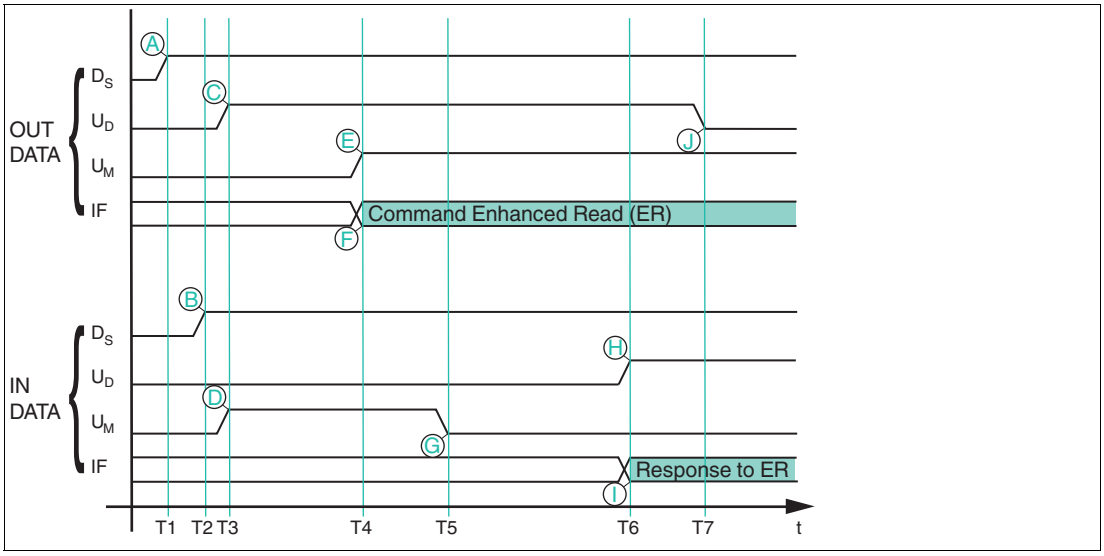
The data being transmitted from the device as a result of the command is returned in the <Data Byte> bytes. This could be data read from a read/write tag or parameter values for the UHF settings.

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 station 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-OUTPUT (**J**). Only now can the device send another telegram.

2023-07

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.

```
DS_OUTPUT := NOT DS_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.

`UD_OUTPUT := NOT UD_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 (UM_OUTPUT <> UM_INPUT) then` (* check whether the device can receive new data *)

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

`End_IF`

8.6 Command Overview

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 57 One-time read access to the EPC/UII (bank 01) + TID (bank 10)
EF	16#1D	Enhanced Read Fixcode see "Enhanced Read Fixcode (EF)" on page 59 Permanent read access to the EPC/UII (bank 01) + TID (bank 10)
SN	16#D2	Single Read EPC/UII see "Single Read EPC/UII (SN)" on page 62 One-time read access to the EPC/UII (bank 01)
EN	16#D3	Enhanced Read EPC/UII see "Enhanced Read EPC/UII (EN)" on page 64 Permanent read access to the EPC/UII (bank 01)
#SU	16#CE	Single Write EPC/UII see "Single Write EPC/UII (#SU)" on page 66 One-time write access to the EPC/UII (bank 01)

Abbreviation	Command code	Command description
SR	16#10	Single Read 4-Byte Blocks see "Single Read 4-Byte Blocks (SR)" on page 69 One-time read access to EPC/UII (bank 01) + 4-byte user memory data blocks (bank 11)
ER	16#19	Enhanced Read 4-Byte Blocksssee "Enhanced Read 4-Byte Blocks (ER)" on page 71 Permanent read access to EPC/UII (bank 01) + 4-byte user memory data blocks (bank 11)
SW	16#40	Single Write 4-Byte Blocks see "Single Write 4-Byte Blocks (SW)" on page 74 One-time write access to 4-byte user memory data blocks (bank 11)
EW	16#1A	Enhanced Write 4-Byte Blocks see "Enhanced Write 4-Byte Blocks (EW)" on page 76 Permanent write access to 4-byte user memory data blocks (bank 11)
#SR	16#49	Single Read 2-Byte Words see "Single Read 2-Byte Words (#SR)" on page 79 One-time read access to EPC/UII (bank 01) + 2-byte user memory data words (bank 11)
#ER	16#4B	Enhanced Read 2-Byte Words see "Enhanced Read 2-Byte Words (#ER)" on page 81 Permanent read access to EPC/UII (bank 01) + 2-byte user memory data words (bank 11)
#SW	16#4A	Single Write 2-Byte Words see "Single Write 2-Byte Words (#SW)" on page 84 One-time write access to 2-byte user memory data words (bank 11)
#EW	16#4C	Enhanced Write 2-Byte Words see "Enhanced Write 2-Byte Words (#EW)" on page 86 Permanent write access to 2-byte user memory data words (bank 11)
LO	16#D5	Lock see "Lock (LO)" on page 89 One-time write access to one or more tags
KI	16#B9	Kill see "Kill (KI)" on page 92 Sets a tag to a state where no further access is possible.

System Commands

Abbreviation	Command code	Command description
QU	16#02	Quit see "Quit (QU)" on page 94 Cancels an active enhanced command
VE	16#03	Version see "Version (VE)" on page 95 Reads out the firmware version

Filter Commands

Abbreviation	Command code	Command description
FI	16#CA	Set filter mask see "Set Filter Mask (FI)" on page 96 Sets a filter
MF	16#CB	Activate filter mask see "Activate Filter Mask (MF)" on page 101 Activates a filter

2023-07

Configuration Commands

Abbreviation	Command code	Command description
RP	16#BE	Read parameter see "Read Parameter (RP)" on page 104 Reads out the device parameters
WP	16#BF	Write parameter see "Write Parameter (WP)" on page 106 Sets device parameters

8.6.1 Read/Write Commands

The tag's memory structure is based on the following read/write commands in accordance with EPC Gen 2 (ISO/IEC 18000-63).

The following commands and responses are described using the long-form data format. This means that the answer always contains the EPC/UII and additional length information. The end of a single command is signaled back via a STATUS 16#0F telegram. When using the short-form data format, the EPC/UII and the additional length information are omitted. The end of a single 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 EPC/UII (bank 01) and the fixcode (TID; bank 10) of one or more tags within the detection zone. The information is transmitted for each tag in a separate data telegram with the status value 16#00. The end of the command execution is indicated by an end telegram. This 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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.3

The <FrameLength> has the value 16#06 as 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 "Single Read Fixcode" command is 16#01.

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	<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#01							
Byte 6	Status	16#00							
Byte 7	Parameter/data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							
...	Parameter/data	<Length TID (High Byte)>							
...	Parameter/data	<Length TID (Low Byte)>							
...	Parameter/data	<TID Byte 1>							
...	Parameter/data	<TID Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<TID Byte Y>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.4

The length of the <FrameLength> fragment depends on the size of the EPC/UII and TID of the read-in read/write tag. The <FrameLength> contains all bytes up to and including <TID Byte Y>. The <FragmentationCounter> has the value 16#00, because all response data can be transmitted within one fragment. The value of <TelegramLength> depends on the length of the EPC/UII and TID of the tag. The size of the telegram extends up to and including <TID Byte Y>.

The <Command> parameter has the value 16#01 and is reflected within the data telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification of the EPC/UII information. It is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag that has been read.

The EPC/UII is followed by a length specification for the TID. It is always 2 bytes in size. The final piece of information is the TID of the identified tag up to and including <TID Byte Y>. All subsequent bytes within the data telegram have the value 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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.5

The <FrameLength> constantly has a value of 16#0B in the response for the end of the "Single Read Fixcode" command. 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#01 command telegram.

The <Status> for the telegram that indicates the command end is 16#0F.

The number of identified read/write 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 read/write 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 Fixcode (EF)

The "Enhanced Read Fixcode" command has the command code 16#1D and performs a permanent read operation on the EPC/UII (bank 01) and the fixcode (TID; bank 10) of one or more tags within the detection zone. The information is transmitted for each tag in a separate data telegram with the status value 16#00. A tag leaving the detection zone is indicated by a response telegram containing the EPC/UII of the tag as well. This telegram however 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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.6

The <FrameLength> has the value 16#06 as 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.

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	<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#1D							
Byte 6	Status	16#00							
Byte 7	Parameter/data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							
...	Parameter/data	<Length TID (High Byte)>							
...	Parameter/data	<Length TID (Low Byte)>							
...	Parameter/data	<TID Byte 1>							
...	Parameter/data	<TID Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<TID Byte Y>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.7

The length of the fragment <FrameLength> depends on the size of the EPC/UII and TID of the tag. The <FrameLength> contains all bytes up to and including <TID Byte Y> but will not be larger than the maximum allowed length of the fragment. The <FragmentationCounter> has the value 16#00 because all response data can be transmitted within one fragment. The value of <TelegramLength> depends on the length of the EPC/UII and TID of the tag. The size of the telegram extends up to and including <TID Byte Y>. The <Command> parameter has the value 16#1D and is reflected within the response telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification of the EPC/UII information. It is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag that has been read.

The EPC/UII is followed by a length specification for the TID. It is always 2 bytes in size. The final piece of information is the TID of the identified tag up to and including <TID Byte Y>. All subsequent bytes within the data telegram have the value 16#00.

Response tag has left sensing range, 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	<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#1D							
Byte 6	Status	16#05							
Byte 7	Parameter/data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.8

The length of the fragment <FrameLength> depends on the size of the EPC/UII of the tag leaving the sensing range. The <FrameLength> contains all bytes up to and including <EPC/UII 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 length of the EPC/UII of the tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#1D and is reflected within the telegram. The <Status> parameter has the value 16#05.

This is followed by a length specification of the EPC/UII information. It is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag leaving the sensing range. All subsequent bytes within the telegram have the value 16#00.

Single Read EPC/UII (SN)

The "Single Read Special EPC/UII" command has the command code 16#0A and performs a one-time read operation on the EPC/UII (bank 01) of one or more tags within the detection zone. The information is transmitted for each tag in a separate data telegram with the status value 16#00. The end of the command execution is indicated by an end telegram. This telegram has the status value 16#0F and contains the number of tags that were 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#D2							
Byte 6	Not relevant	16#00							
Byte 7	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.9

The <FrameLength> has the value 16#06 because 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> command code for the "Single Read EPC/UII" command is 16#D2.

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	<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#D2							
Byte 6	Status	16#00							
Byte 7	Parameter/data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							

2023-07

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

Table 8.10

The length of the <FrameLength> fragment depends on the size of the EPC/UII of the read-in read/write tag. The <FrameLength> contains all bytes up to and including <EPC/UII 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 length of the EPC/UII of the tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#D2 and is reflected within the data telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification of the EPC/UII information. It is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag that has been read. All subsequent bytes within the data telegram have the value 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#D2							
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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.11

The <FrameLength> has a constant value of 16#0B in the response for the end of the "Single Read EPC/UII command. 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 reflected and has the same value as the 16#D2 command telegram. The <Status> for the telegram that indicates the command end is 16#0F.

The number of identified read/write 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 read/write 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 EPC/UII (EN)

The "Enhanced Read Special EPC/UII" command has the command code 16#D3 and performs a permanent read operation on the EPC/UII (bank 01) of one or more tags within the detection zone. The read information is transmitted for each read tag in a separate data telegram. A tag leaving the detection zone is indicated by a response telegram containing the EPC/UII of the tag as well. 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 4	Telegram length (low byte)	16#03							
Byte 5	Command	16#D3							
Byte 6	Not relevant	16#00							
Byte 7	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.12

The <FrameLength> has the value 16#06, because no further command parameters have to be transmitted 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 command is 16#D3.

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	<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#D3							
Byte 6	Status	16#00							
Byte 7	Parameter/data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							

2023-07

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

Table 8.13

The length of the fragment <FrameLength> depends on the size of the EPC/UII and the number of bytes read in from the tag. The <FrameLength> contains all bytes up to and including <EPC/UII 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 length of the EPC/UII of the tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#D3 and is reflected within the response telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification of the EPC/UII information. It is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag that has been read. All subsequent bytes within the data telegram have the value 16#00.

Response tag has left sensing range, 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	<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#D3							
Byte 6	Status	16#05							
Byte 7	Parameter/data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.14

The length of the fragment <FrameLength> depends on the size of the EPC/UII of the tag leaving the sensing range. The <FrameLength> contains all bytes up to and including <EPC/UII 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 length of the EPC/UII of the tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#D3 and is reflected within the telegram. The <Status> parameter has the value 16#05.

This is followed by a length specification of the EPC/UII information. It is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag leaving the sensing range. All subsequent bytes within the telegram have the value 16#00.

Single Write EPC/UII (#SU)

The "Single Write EPC/UII" command has the command code 16#CE and performs a one-time write operation to the EPC/UII (bank 01) of a tag within the sensing range. Only one tag may be within the sensing range at a time while the command is executed. A data telegram indicates successful programming of the EPC/UII. As soon as a read/write tag has been detected and written with the new EPC/UII, the command execution is canceled. No additional inventory runs are performed. This data telegram contains the newly programmed EPC/UII of the tag. The end of the command execution is indicated by an end telegram. This telegram contains the number of tags that were identified during execution of the command.

The "Single Write EPC/III" command can be used to write an EPC/UII of any length to a read/write tag. The length of the EPC/UII must be a multiple of 2 bytes. The PC word is transmitted via this command and is preceded by the actual EPC/UII in the data field. The correct value of the PC word must first be calculated by the user.

The <Toggle> bit is used within the PC word to distinguish between an EPC code and a UII code. This bit must be set to 0 when programming an EPC code. However, this bit is set to 1 when programming a UII code.

Depending on the length of the EPC/UII, the PC word will have the following values:

Length of EPC/UII	EPC PC word	UII PC word
2 bytes	16#0800	16#0900
4 bytes	16#1000	16#1100
6 bytes	16#1800	16#1900
8 bytes	16#2000	16#2100
10 bytes	16#2800	16#2900
12 bytes	16#3000	16#3100
14 bytes	16#3800	16#3900
22 bytes	16#5800	16#5900

The <Length EPC/UII> parameter defines the length of the EPC/UII information. The EPC/UII information consists of the PC word with a length of 2 bytes plus the EPC/UII. The length of the EPC/UII is variable.

Depending on the length of the EPC/UII, <Length EPC/UII> will have the following values:

Length of EPC/UII	<Length EPC/UII>
2 bytes	16#04
4 bytes	16#06
6 bytes	16#08
8 bytes	16#0A
10 bytes	16#0C
12 bytes	16#0E
14 bytes	16#10
22 bytes	16#18

Table 8.15

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	<FragmentationCounter>							
Byte 3	Telegram length (high byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram length (low byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#CE							
Byte 6	Length EPC/UII	<Length EPC/UII>							
Byte 7	Parameter/data	<PC word (High Byte)>							
Byte 8	Parameter/data	<PC word (Low Byte)>							
Byte 9	Parameter/data	<EPC/UII Byte 1>							
Byte 10	Parameter/data	<EPC/UII Byte 2>							
...	Parameter/data	...							
Byte x	Parameter/data	<EPC/UII Byte x>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

The value of <FrameLength> depends on the length of the EPC/UII code to be programmed. The fragment ends with the <EPC/UII 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 telegram (<TelegramLength>) depends on the length of the EPC/UII code to be programmed. The telegram ends with the <EPC/UII Byte X> byte. The command code <Command> for the Single Write EPC/UII command is 16#CE. Parameterization of <Length EPC/UII> follows this. The value to be set depends on the length of the EPC/UII. The <PC word> follows this. The value of the PC word depends on the type of code (EPC or UII) and the code length. The EPC/UII code to be programmed to the read/write tag follows.

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	<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#CE							
Byte 6	Status	16#00							
Byte 7	Parameter/data	<Length EPC/UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC/UII (Low Byte)>							
Byte 9	Parameter/data	<PC word (High Byte)>							
Byte 10	Parameter/data	<PC word (Low Byte)>							
Byte 11	Parameter/data	<EPC/UII Byte 1>							
Byte 12	Parameter/data	<EPC/UII Byte 2>							
...	Parameter/data	...							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
...	Parameter/data	<EPC/UII Byte X>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#0							

The length of the <FrameLength> fragment depends on the size of the EPC/UII of the newly programmed read/write tag. The <FrameLength> contains all bytes up to and including <EPC/UII 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 length of the EPC/UII of the tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#CE and is reflected within the response telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification of the EPC/UII information. It is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. The newly programmed EPC/UII in the read/write tag is used to uniquely identify the written tag.

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#CE							
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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

The <FrameLength> has a constant value of 16#0B in the response for the end of the "Single Write EPC/UII command. 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#CE command telegram. The <Status> for the telegram that indicates the command end is 16#0F.

The number of read/write tags written is transmitted within the 4 bytes. The number is displayed in ASCII coding.

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

If no read/write 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.

2023-07

Single Read 4-Byte Blocks (SR)

The "Single Read 4-Byte Blocks" command has the command code 16#10 and performs a one-time read operation on the EPC/UII (Bank 01) and 4 byte data blocks of the user memory (bank 11) for one or more tags within the detection zone. The information is transmitted for each tag in a separate data telegram with the status value 16#00. 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.

In the factory setting, this command accesses 4 byte data blocks of the user memory (bank 11). The memory bank can be changed using the "Memory Bank" (MB) parameter. By changing the memory bank, this command can also read data from the Reserved (bank 00), EPC/UII (bank 01) and TID (bank 10).

The <ByteAddress> parameter specifies the start address within the memory bank from which the 4 byte blocks are read. The value of <ByteAddress> is based on bytes. This means that only multiples of 4 can be accessed. <Number of Bytes> defines the number of bytes to be read. The number of bytes must be a multiple of 4.

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

Table 8.16

The <FrameLength> has a value of 16#0A and is the length of the fragment in bytes. 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>) is 16#07 and ends with <Number of Bytes (Low Byte)>. The command code <Command> for the "Single Read 4-Byte Blocks" command is 16#10.

The <ByteAddress> parameter specifies the start address within the memory bank from which the 4 byte blocks are read. The <Number of Bytes> parameter is used to specify the number of bytes to be read. All other bytes of the command fragment are not relevant for this command. They must all be set 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	<FrameLength>			
Byte 1	Frame length	<FrameLength>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
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	Parameter/data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							
...	Parameter/data	<Length User Data (High Byte)>							
...	Parameter/data	<Length User Data (Low Byte)>							
...	Parameter/data	<User Data Byte 1>							
...	Parameter/data	<User Data Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<User Data Byte Y>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.17

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

The <Command> parameter has the value 16#10 and is reflected within the data telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification of the EPC/UII information. The length specification is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag that has been read.

The EPC/UII is followed by a length specification of the 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 Y>. All subsequent bytes within the data telegram have the value 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							

2023-07

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

Table 8.18

The <FrameLength> has a value of 16#0B in the response for the end of the Single Read 4-Byte Blocks command. 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 reflected 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 read/write 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 read/write 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 4-Byte Blocks (ER)

The "Enhanced Read 4-Byte Blocks" command has the command code 16#19 and performs a permanent read operation on the EPC/UII (Bank 01) and 4 byte data blocks of the user memory (bank 11) for one or more tags within the detection zone. The information is transmitted for each tag in a separate data telegram with the status value 16#00. A tag leaving the detection zone is indicated by a response telegram containing the EPC/UII of the tag as well. This telegram however has the status value 16#05. A Quit command stops the command execution.

In the factory setting, this command accesses 4 byte data blocks of the user memory (bank 11). The memory bank can be changed using the "Memory Bank" (MB) parameter. By changing the memory bank, this command can also read data from the Reserved (bank 00), EPC/UII (bank 01) and TID (bank 10).

The <ByteAddress> parameter specifies the start address within the memory bank from which the 4 byte blocks are read. The value of <ByteAddress> is based on bytes. This means that only multiples of 4 can be accessed. <Number of Bytes> defines the number of bytes to be read. The number of bytes must also be a multiple of 4.

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

Table 8.19

The <FrameLength> has a value of 16#0A and is the length of the fragment in bytes. 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>) is 16#07. The command code <Command> for the "Enhanced Read 4-Byte Blocks" is 16#19.

The <ByteAddress> parameter specifies the start address within the memory bank from which the 4 byte blocks are read. The <Number of Bytes> parameter is used to specify the number of bytes to be read. All other bytes of the command are not relevant and must all be set to 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	<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#19							
Byte 6	Status	16#00							
Byte 7	Parameter/data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							
...	Parameter/data	<Length User Data (High Byte)>							
...	Parameter/data	<Length User Data (Low Byte)>							
...	Parameter/data	<User Data Byte 1>							
...	Parameter/data	<User Data Byte 2>							

2023-07

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
...	Parameter/data	...							
...	Parameter/data	<User Data Byte Y>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.20

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

The <Command> parameter has the value 16#19 and is reflected within the data telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification of the EPC/UII information. It is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag that has been read.

The EPC/UII is followed by a length specification of the 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 Y>. All subsequent bytes within the data telegram have the value 16#00.

Response tag has left sensing range, 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	<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#19							
Byte 6	Status	16#05							
Byte 7	Parameter/data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.21

The length of the fragment <FrameLength> depends on the size of the EPC/UII leaving the sensing range. The <FrameLength> contains all bytes up to and including <EPC/UII 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 length of the EPC/UII of the tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#19 and is reflected within the telegram. The <Status> parameter has the value 16#05.

This is followed by a length specification of the EPC/UII information. It is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag leaving the sensing range. All subsequent bytes within the telegram have the value 16#00.

Single Write 4-Byte Blocks (SW)

The "Single Write 4-Byte Blocks" command has the command code 16#40 and executes a one-time write operation on 4 byte data blocks of the user memory (bank 11) for one or more tags within the detection zone. A response telegram with the status value 16#00 indicates a successful write operation for each tag. The data telegram contains the EPC/UII of the tag to which the user memory was written. The end of the command execution is indicated by an end telegram. The end telegram has the status value 16#0F and contains the number of tags that were written during execution of the command.

In the factory setting, this command accesses 4 byte data blocks of the user memory (bank 11). The memory bank can be changed using the "Memory Bank" (MB) parameter. By changing the memory bank, this command can also read data from the Reserved (bank 00) and EPC/UII (bank 01).

The <ByteAddress> parameter specifies the start address within the memory bank from which the 4 byte blocks are written. The value of <ByteAddress> is based on bytes. This means that only multiples of 4 can be accessed. <Number of Bytes> defines the number of bytes to be written. The number of bytes must be a multiple of 4.

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	<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	<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	Parameter/data	<Write Data Byte 1>							
Byte 11	Parameter/data	<Write Data Byte 2>							
...	Parameter/data	...							
Byte x	Parameter/data	<Write Data Byte Y>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.22

2023-07

The value of <FrameLength> depends on the number of bytes to be written. The fragment ends with the <Write Data Byte Y> 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 Y> byte. The command code <Command> for the "Single Write 4-Byte Blocks" command is 16#40.

The <ByteAddress> parameter specifies the start address within the memory bank from which the 4 byte blocks are written. The <Number of Bytes> parameter is used to specify the number of bytes to be written. This is followed by the <Write Data Byte> with the information to be written to the tag. All other bytes of the command are not relevant and must all be set to 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	<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#40							
Byte 6	Status	16#00							
Byte 7	Parameter/data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.23

The length of the <FrameLength> fragment depends on the size of the EPC/UII of the tag. The <FrameLength> contains all bytes up to and including <EPC/UII 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 length of the EPC/UII of the tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#40 and is reflected within the data telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification of the EPC/UII information. The length specification is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. The EPC/UII is used to uniquely identify the tag that was written.

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	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.24

The <FrameLength> has a value of 16#0F in the response for the end of the "Single Write 4-Byte Blocks" command. 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 reflected and has the same value as the 16#40 command telegram. The <Status> for the telegram that indicates the command end is 16#0F.

The number of read/write tags written is transmitted within the 4 bytes. The number is displayed in ASCII coding.

If the write operation to a tag is successful, the <Number of Read/Write Tags> has the value "0001" (ASCII) or 16#30303031.

If no read/write tag was written 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 4-Byte Blocks (EW)

The "Enhanced Write 4-Byte Blocks" command has the command code 16#1A and executes a permanent write operation on 4 byte data blocks of the user memory (bank 11) for one or more tags within the detection zone. A response telegram with the status value 16#00 indicates a successful write operation for each tag. The response telegram contains the EPC/UII of the tags that were written. A tag leaving the detection zone is indicated by a response telegram containing the EPC/UII of the tag as well. This telegram however has the status value 16#05. A Quit command stops the command execution.

In the factory setting, this command accesses 4 byte data blocks of the user memory (bank 11). The memory bank can be changed using the "Memory Bank" (MB) parameter. By changing the memory bank, this command can also read data from the Reserved (bank 00) and EPC/UII (bank 01).

The <ByteAddress> parameter specifies the start address within the memory bank from which the 4 byte blocks are written. The value of <ByteAddress> is based on bytes. This means that only multiples of 4 can be accessed. <Number of Bytes> defines the number of bytes to be written. The number of bytes must be a multiple of 4.

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	<FragmentationCounter>							
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	<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	Parameter/data	<Write Data Byte 1>							
Byte 11	Parameter/data	<Write Data Byte 2>							
...	Parameter/data	...							
Byte x	Parameter/data	<Write Data Byte Y>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.25

The value of <FrameLength> depends on the number of bytes to be written. The fragment ends with the <Write Data Byte Y> 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 Y> byte. The <Command> command code for the "Enhanced Write 4-Byte Blocks" command is 16#1A.

The <ByteAddress> parameter specifies the start address within the memory bank from which the 4 byte blocks are written. The <Number of Bytes> parameter is used to specify the number of bytes to be written. This is followed by the <Write Data Byte> with the information to be written to the tag. All other bytes of the command are not relevant and must all be set to 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	<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#1A							
Byte 6	Status	16#00							
Byte 7	Parameter/data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.26

The length of the <FrameLength> fragment depends on the size of the EPC/UII of the tag. The <FrameLength> contains all bytes up to and including <EPC/UII Byte X>. The <Fragmentation-Counter> has the value 16#00 because all response data can be transmitted within one fragment. The value of <TelegramLength> depends on the length of the EPC/UII of the written tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#1A and is reflected within the response telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification of the EPC/UII information. The length specification is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag that has been written. All subsequent bytes within the data telegram have the value 16#00.

Response tag has left sensing range, 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	<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#1A							
Byte 6	Status	16#05							
Byte 7	Parameter/data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.27

The length of the fragment <FrameLength> depends on the size of the EPC/UII of the tag leaving the sensing range. The <FrameLength> contains all bytes up to and including <EPC/UII 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 length of the EPC/UII of the tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#1A and is reflected within the telegram. The <Status> parameter has the value 16#05.

This is followed by a length specification of the EPC/UII information. The length specification is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag leaving the sensing range. All subsequent bytes within the telegram have the value 16#00.

Single Read 2-Byte Words (#SR)

The "Single Read 2-Byte Words" command has the command code 16#49 and performs a one-time read operation on the EPC/UII (bank 01) and 2 byte data words of the user memory (bank 11) for one or more tags within the detection zone. The information is transmitted for each tag in a separate data telegram with the status value 16#00. 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.

In the factory setting, this command accesses 2 byte data words of the user memory (bank 11). The memory bank can be changed using the "Memory Bank" (MB) parameter. By changing the memory bank, this command can also read data from the Reserved (bank 00), EPC/UII (bank 01) and TID (bank 10).

The parameter <ByteAddress> specifies the start location in the memory bank from which the 2 byte words are to be read. The value of <ByteAddress> is based on bytes. This means that only multiples of 2 can be accessed. <Number of Bytes> defines the number of bytes to be read. The number of bytes must be a multiple of 2.

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#49							
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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.28

The <FrameLength> has a value of 16#0A and is the length of the fragment in bytes. 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>) is 16#07. The <Command> code for the "Single Read 2-Byte Words" is 16#49.

The <ByteAddress> parameter specifies the start address within the memory bank from which the 2 byte words are read. The <Number of Bytes> parameter is used to specify the number of bytes to be read. All other bytes of the command fragment are not relevant for this command. They must all be set 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	<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#49							
Byte 6	Status	16#00							
Byte 7	Parameter/data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							
...	Parameter/data	<Length User Data (High Byte)>							
...	Parameter/data	<Length User Data (Low Byte)>							
...	Parameter/data	<User Data Byte 1>							
...	Parameter/data	<User Data Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<User Data Byte Y>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.29

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

The <Command> parameter has the value 16#49 and is reflected within the data telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification of the EPC/UII information. The length specification is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag that has been read.

The EPC/UII is followed by a length specification of the 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 Y>. All subsequent bytes within the data telegram have the value 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#49							
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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.30

The <FrameLength> has a value of 16#0B in the response for the end of the "Single Read 2-Byte Words" command. 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 reflected and has the same value as the 16#49 command telegram. The <Status> for the telegram that indicates the command end is 16#0F.

The number of identified read/write 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 read/write 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 2-Byte Words (#ER)

The "Enhanced Read 2-Byte Words" command has the command code 16#4B and performs a permanent read operation on the EPC/UII (bank 01) and two byte data words of the user memory (bank 11) for one or more tags within the sensing range. The information is transmitted for each tag in a separate data telegram with the status value 16#00. A tag leaving the sensing range is indicated by a response telegram containing the EPC/UII of the tag as well. This telegram however has the status value 16#05. A Quit command stops the command execution.

In the factory setting, this command accesses two byte data words of the user memory (bank 11). The memory bank can be changed using the "Memory Bank" (MB) parameter. By changing the memory bank, this command can also read data from the Reserved (bank 00), EPC/UII (bank 01) and TID (bank 10).

The parameter <ByteAddress> specifies the start location in the memory bank from which the 2 byte words are to be read. The value of <ByteAddress> is based on bytes. This means that only multiples of 2 can be accessed. <Number of Bytes> defines the number of bytes to be read. The number of bytes must be a multiple of 2.

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#4B							
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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.31

The <FrameLength> has a value of 16#0A and is the length of the fragment in bytes. 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>) is 16#07. The command code <Command> for the "Enhanced Read 2-Byte Words" command is 16#4B.

The <ByteAddress> parameter specifies the start address within the memory bank from which the two byte words are read in. The <Number of Bytes> parameter is used to specify the number of bytes to be read. All other bytes of the command are not relevant and must all be set to 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	<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#4B							
Byte 6	Status	16#00							
Byte 7	Parameter/data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
...	Parameter/data	<Length User Data (High Byte)>							
...	Parameter/data	<Length User Data (Low Byte)>							
...	Parameter/data	<User Data Byte 1>							
...	Parameter/data	<User Data Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<User Data Byte Y>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.32

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

The <Command> parameter has the value 16#4B and is reflected within the data telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification of the EPC/UII information. It is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag that has been read.

The EPC/UII is followed by a length specification of the 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 Y>. All subsequent bytes within the data telegram have the value 16#00.

Response tag has left sensing range, 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	<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#4B							
Byte 6	Status	16#05							
Byte 7	Parameter/data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							
...	Not relevant	16#00							

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

Table 8.33

The length of the fragment <FrameLength> depends on the size of the EPC/UII of the tag leaving the sensing range. The <FrameLength> contains all bytes up to and including <EPC/UII 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 length of the EPC/UII of the tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#4B and is reflected within the telegram. The <Status> parameter has the value 16#05.

This is followed by a length specification of the EPC/UII information. It is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag leaving the sensing range. All subsequent bytes within the telegram have the value 16#00.

Single Write 2-Byte Words (#SW)

The "Single Write 2-Byte Words" command has the command code 16#4A and executes a one-time write operation on 2 byte data words of the user memory (bank 11) for one or more tags within the detection zone. A response telegram with the status value 16#00 indicates a successful write operation for each tag. The data telegram contains the EPC/UII of the tag to which the user memory was written. The end of the command execution is indicated by an end telegram. The end telegram has the status value 16#0F and contains the number of tags that were written during execution of the command.

In the factory setting, this command accesses 2 byte data words of the user memory (bank 11). The memory bank can be changed using the "Memory Bank" (MB) parameter. By changing the memory bank, this command can also read data from the Reserved (bank 00) and EPC/UII (bank 01).

The <ByteAddress> parameter specifies the start address within the memory bank from which the 2 byte blocks are written. The value of <ByteAddress> is based on bytes. This means that only multiples of 2 can be accessed. <Number of Bytes> defines the number of bytes to be written. The number of bytes must be a multiple of 2.

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	<FragmentationCounter>							
Byte 3	Telegram length (high byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram length (low byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#4A							
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	Parameter/data	<Write Data Byte 1>							
Byte 11	Parameter/data	<Write Data Byte 2>							
...	Parameter/data	...							

2023-07

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

Table 8.34

The value of <FrameLength> depends on the number of bytes to be written. The fragment ends with the <Write Data Byte Y> 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 Y> byte. The command code <Command> for the "Single Write 2-Byte Words" command is 16#4A.

The <ByteAddress> parameter specifies the start address within the memory bank from which the 2 byte blocks are written. The <Number of Bytes> parameter is used to specify the number of bytes to be written. This is followed by the <Write Data Byte> with the information to be written to the tag. All other bytes of the command are not relevant and must all be set to 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	<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#4A							
Byte 6	Status	16#00							
Byte 7	Parameter/data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.35

The length of the <FrameLength> fragment depends on the size of the EPC/UII of the tag. The <FrameLength> contains all bytes up to and including <EPC/UII 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 length of the EPC/UII of the tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#4A and is reflected within the response telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification of the EPC/UII information. It is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. The EPC/UII is used to uniquely identify the tag that was written.

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#4A							
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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.36

The <FrameLength> has a value of 16#0F in the response for the end of the "Single Write 2-Byte Words" command. 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#4A command telegram. The <Status> for the telegram that indicates the command end is 16#0F.

The number of read/write tags written is transmitted within the 4 bytes. The number is displayed in ASCII coding.

If the write operation to a tag is successful, the <Number of Read/Write Tags> has the value "0001" (ASCII) or 16#30303031.

If no read/write tag was written 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 2-Byte Words (#EW)

The "Enhanced Write 2-Byte Words" command has the command code 16#4C and executes a permanent write operation on 2 byte data blocks of the user memory (bank 11) for one or more tags within the detection zone. A response telegram with the status value 16#00 indicates a successful write operation for each tag. The response telegram contains the EPC/UII of the tags that were written. A tag leaving the detection zone is indicated by a response telegram containing the EPC/UII of the tag as well. This telegram however has the status value 16#05. A Quit command stops the command execution.

In the factory setting, this command accesses 2 byte data words of the user memory (bank 11). The memory bank can be changed using the "Memory Bank" (MB) parameter. By changing the memory bank, this command can also read data from the memory banks (bank 00) and EPC/UII (bank 01).

The <ByteAddress> parameter specifies the start address within the memory bank from which the 2 byte blocks are written. The value of <ByteAddress> is based on bytes. This means that only multiples of 2 can be accessed. <Number of Bytes> defines the number of bytes to be written. The number of bytes must be a multiple of 2.

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	<FragmentationCounter>							
Byte 3	Telegram length (high byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram length (low byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#4C							
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	Parameter/data	<Write Data Byte 1>							
Byte 11	Parameter/data	<Write Data Byte 2>							
...	Parameter/data	...							
Byte x	Parameter/data	<Write Data Byte Y>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.37

The value of <FrameLength> depends on the number of bytes to be written. The fragment ends with the <Write Data Byte Y> 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 Y> byte. The <Command> command code for the "Enhanced Write 2-Byte Words" command is 16#4C.

The <ByteAddress> parameter specifies the start address within the memory bank from which the 2 byte blocks are written. The <Number of Bytes> parameter is used to specify the number of bytes to be written. This is followed by the <Write Data Byte> with the information to be written to the tag. All other bytes of the command are not relevant and must all be set to 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	<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#4C							
Byte 6	Status	16#00							
Byte 7	Parameter/data	<Length EPC / UII (High Byte)>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.38

The length of the <FrameLength> fragment depends on the size of the EPC/UII of the tag. The <FrameLength> contains all bytes up to and including <EPC/UII Byte X>. The <Fragmentation-Counter> has the value 16#00, because all response data can be transmitted within one fragment. The value of <TelegramLength> depends on the length of the EPC/UII of the written tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#4C and is reflected within the response telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification of the EPC/UII information. It is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag that has been written. All subsequent bytes within the data telegram have the value 16#00.

Response tag has left sensing range, 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	<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#4C							
Byte 6	Status	16#05							
Byte 7	Parameter/data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC / UII Byte 1>							
Byte 12	Parameter/data	<EPC / UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.39

2023-07

The length of the fragment <FrameLength> depends on the size of the EPC/UII of the tag leaving the sensing range. The <FrameLength> contains all bytes up to and including <EPC/UII 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 length of the EPC/UII of the tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#4C and is reflected within the telegram. The <Status> parameter has the value 16#05.

This is followed by a length specification of the EPC/UII information. The length specification is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag leaving the sensing range. All subsequent bytes within the telegram have the value 16#00.

Lock (LO)

The "Lock" command has the command code 16#D5 and executes a one-time write access to one or more tags within the sensing range. A data telegram with the status value 16#00 indicates a successful write operation for each tag. The data telegram contains the EPC/UII of the tag to which the configuration was written. The end of the command execution is indicated by a final 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 value of <FrameLength> depends on the length of the EPC/UII mask. The fragment ends with the <Payload (Byte 2)> byte. The <FragmentationCounter> has the value 16#00 because the command telegram can be transmitted within one fragment. The length of the command telegram <TelegramLength> depends on the length of the EPC/UII mask. The telegram ends with the <Payload (Byte 2)> byte.

<Length EPC/III Mask> can be used to set the length of the EPC/III mask (bank 01) of one or more tags on which the command is to act. The length of the mask must not be 0. The tags on which the command is to act are selected using the <EPC/III Mask> parameter. The previously saved access password must be entered in Access Password. The password must not be 0. The various lock functions are configured using the <Payload> parameter.

Payload

Payload	Bit	Function	Action
User memory	0	User Action	perma lock
	1		pwd write
TID Memory	2	TID Action	perma lock
	3		pwd write
UII Memory	4	UII Action	perma lock
	5		pwd write
Access Password	6	Access Action	perma lock
	7		pwd write
Kill Password	8	Kill Action	perma lock
	9		pwd read/write
User memory	10	User Mask	skip / write
	11		skip / write
TID Memory	12	TID Mask	skip / write
	13		skip / write
UII Memory	14	UII Mask	skip / write
	15		skip / write
Access Password	16	Access Mask	skip / write
	17		skip / write
Kill Password	18	Kill Mask	skip / write
	19		skip / write

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	<FragmentationCounter>							
Byte 3	Telegram length (high byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram length (low byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#D5							
Byte 6	Length EPC/UII Mask (High Byte)	<Length EPC/UII Mask (High Byte)>							
Byte 7	Length EPC/UII Mask (Low Byte)	<Length EPC/UII Mask (Low Byte)>							
Byte 8	EPC/UII mask (byte 0)	<EPC/UII Mask (Byte 0)>							
Byte 9	EPC/UII mask (byte 1)	<EPC/UII Mask (Byte 1)>							
...	...	...							
Byte x	EPC/UII mask (byte Y)	<EPC/UII Mask (Byte Y)>							
Byte x+1	Access Password (Byte 0)	<Access Password (Byte 0)>							
Byte x+2	Access Password (Byte 1)	<Access Password (Byte 1)>							
Byte x+3	Access Password (Byte 2)	<Access Password (Byte 2)>							
Byte x+4	Access Password (Byte 3)	<Access Password (Byte 3)>							
Byte x+5	Payload (Byte 0)	<Payload (Byte 0)>							
Byte x+6	Payload (Byte 1)	<Payload (Byte 1)>							
Byte x+7	Payload (Byte 2)	<Payload (Byte 2)>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.40

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	<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#D5							
Byte 6	Status	16#00							
Byte 7	Parameter/data	<Length EPC/UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC/UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC/UII Byte 1>							
Byte 12	Parameter/data	<EPC/UII Byte 2>							
...	Parameter/data	...							

2023-07

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
...	Parameter/data	<EPC/UII Byte X>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

The length of the <FrameLength> fragment depends on the size of the EPC/UII of the tag. The <FrameLength> contains all bytes up to and including <EPC/UII 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 length of the EPC/UII of the written tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#D5 and is reflected within the response telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification of the EPC/UII information. The length specification is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag that has been written. All subsequent bytes within the data telegram have the value 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#D5							
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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

The <FrameLength> has a constant value of 16#0B in the response for the end of the lock command. 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#D5 command telegram. The <Status> for the telegram that indicates the command end is 16#0F.

The number of tags written is transmitted within the 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 read/write 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.

Kill (KI)

The "Kill" command has the command code 16#D5 and executes a one-time write access to one or more tags within the sensing range. This command sets a tag to a state where no further access is possible. A response telegram with the status value 16#00 indicates a successful write operation for each tag. The data telegram contains the EPC/UII of the tag to which the configuration was written. The end of the command execution is indicated by an end telegram. The end telegram has the status value 16#0F and contains the number of tags that were written during execution of the command.

The value of <FrameLength> depends on the length of the EPC/UII mask. The fragment ends with the <Pecom Bit> byte. The <FragmentationCounter> has the value 16#00 because the command telegram can be transmitted within one fragment. The length of the command telegram <TelegramLength> depends on the length of the EPC/UII mask. The telegram ends with the byte <Pecom Bits>, which always has the value 16#00.

The length of the EPC/UII mask (bank 01) of one or more tags on which the command is to act can be set via the <Length EPC/UII Mask> parameter. The length of the mask must not be 0. The tags on which the command is to act are selected using the <EPC/III Mask> parameter. The previously saved kill password must be entered in <Kill Password>, which must not be 0.

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	<FragmentationCounter>							
Byte 3	Telegram length (high byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram length (low byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#B9							
Byte 6	Length EPC/UII Mask (High Byte)	<Length EPC/UII Mask (High Byte)>							
Byte 7	Length EPC/UII Mask (Low Byte)	<Length EPC/UII Mask (Low Byte)>							
Byte 8	EPC/UII mask (byte 0)	<EPC/UII Mask (Byte 0)>							
Byte 9	EPC/UII Mask (Byte 1)	<EPC/UII Mask (Byte 1))>							
...	...	...							
Byte x	EPC/UII Mask (Byte Y)	<EPC/UII Mask (Byte Y)>							
Byte x+0	Not relevant	16#00							
Byte x+1	Kill Password (Byte 0)	<Kill Password (Byte 0)>							
Byte x+2	Kill Password (Byte 1)	<Kill Password (Byte 1)>							
Byte x+3	Kill Password (Byte 2)	<Kill Password (Byte 2)>							
Byte x+4	Kill Password (Byte 3)	<Kill Password (Byte 3)>							
Byte x+5	Payload (Byte 0)	<Payload (Byte 0)>							
Byte x+6	Payload (Byte 1)	<Payload (Byte 1)>							
Byte x+7	Payload (Byte 2)	<Payload (Byte 2)>							
Byte x+8	Pecom Bit	<Pecom Bit>; 16#00							
...	...	...							
Byte 31	Not relevant	16#00							

Table 8.41

2023-07

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	<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#B9							
Byte 6	Status	16#00							
Byte 7	Parameter/data	<Length EPC/UII (High Byte)>							
Byte 8	Parameter/data	<Length EPC/UII (Low Byte)>							
Byte 9	Parameter/data	<PC-Word (High Byte)>							
Byte 10	Parameter/data	<PC-Word (Low Byte)>							
Byte 11	Parameter/data	<EPC/UII Byte 1>							
Byte 12	Parameter/data	<EPC/UII Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<EPC/UII Byte X>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

The length of the <FrameLength> fragment depends on the size of the EPC/UII of the tag. The <FrameLength> contains all bytes up to and including <EPC/UII Byte X>. The <Fragmentation-Counter> has the value 16#00 because all response data can be transmitted within one fragment. The value of <TelegramLength> depends on the length of the EPC/UII of the written tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#B9 and is reflected within the data telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification of the EPC/UII information. The length specification is always 2 bytes in size. The length refers to the size, in bytes, of the PC word and the EPC/UII. This is followed by the PC and EPC/UII to uniquely identify the tag that has been written. All subsequent bytes within the data telegram have the value 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#B9							
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	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 10	Parameter/data	<Number of Read/Write Tags Byte 4>							
Byte 11	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

The <FrameLength> has a constant value of 16#0B in the response for the end of Kill command. 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 reflected and has the same value as the 16#B9 command telegram. The <Status> for the telegram that indicates the command end is 16#0F.

The number of tags written is transmitted within the 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 read/write 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.

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 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 8.42

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.

2023-07

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 8.43

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 8.44

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 8.45

8.6.3 Filter Commands

In the factory setting, each read/write command accesses all detected read/write tags in the sensing range. The filters can be used to make a selection. This allows you to specify if read/write commands access only one or multiple read/write tags.

The "Set Filter Mask" (FI) and "Activate Filter" (MF) commands must be used to parameterize access to certain tags. The "Set Filter Mask" (FI) command sets filter masks or filter conditions in the device. The "Activate Filter" (MF) command is used to activate filtering.

The set filter masks can be read using the "Read Parameter" (RP) command via the FL parameter.

Set Filter Mask (FI)

The "Set Filter Mask" (FI) command can be used to parameterize a filter mask in the device. The command code is 16#CA. A maximum of three filter masks can be set.

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	<FragmentationCounter>							
Byte 3	Telegram length (high byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram length (low byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#CA							
Byte 6	Parameter/data	<FilterNumber>							
Byte 7	Parameter/data	<FilterCondition>							
Byte 8	Parameter/data	<FilterAddress (High Byte)>							
Byte 9	Parameter/data	<FilterAddress (Low Byte)>							
Byte 10	Parameter/data	<LengthMask (High Byte)>							
Byte 11	Parameter/data	<Length Mask (Low Byte)>							

2023-07

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 12	Parameter/data	<Mask Data Byte 1>							
Byte 13	Parameter/data	<Mask Data Byte 2>							
...	Parameter/data	...							
...	Parameter/data	<Mask Data Byte X>							
...	Parameter/data	16#00							
Byte 31	Parameter/data	16#00							

Table 8.46

The value of <FrameLength> depends on the size of the filter mask to be set. This determines the length of the fragment up to and including <Mask 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 <Mask Data Byte X> byte. The telegram length therefore depends on the length of the filter mask.

The <Command> byte specifies the command to be executed. The <Command> byte has the value 16#CA to execute the "Set Filter Mask" command.

Filter mask response 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#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#CA							
Byte 6	Status	16#00							
Byte 7	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.47

The <FilterNumber> parameter is used to specify the number of the filter to be set. The device can store up to three filters.

- Filter 1: <FilterNumber> := 16#00
- Filter 2: <FilterNumber> := 16#01
- Filter 3: <FilterNumber> := 16#02

The <FilterCondition> parameter defines various conditions that are considered when filtering the data. The <FilterCondition> byte is split into sections that represent the respective filter conditions.

Content	Bit number							
	7	6	5	4	3	2	1	0
<FilterCondition>	0	0	0	<MemBank>		<Negation>	<LogicalOperation>	<Truncation>

Table 8.48

Bit 0 <Truncation>

0_{bin} All filter data is transmitted; the default value is always 0bin

1_{bin} Only the part of the EPC or the UII that follows the mask is transmitted. The CRC is not recalculated. The stored CRC is sent.

Bit 1 <LogicalOperation>
If more than one filter condition has been set, the <LogicalOperation> bit can be used to logically link multiple filter conditions. This bit is not of significance if only one filter condition has been parameterized.

0_{bin} OR link or value for setting only one filter

1_{bin} AND link

Bit 2 <Negation>
The <Negation> parameter can be used to set a negation of the filter condition. All read/write tags that do not match the filter condition are transmitted.

0_{bin} Not negated; transfer of all read/write tags that match the filter condition

1_{bin} Negated; transfer of all read/write tags that do not match the filter condition

Bit 3/ Bit 4 <MemBank>
The <MemBank> parameter defines the memory bank of the read/write tag to which the filter condition is applied.

00_{bin} Bank 00; not allowed

01_{bin} Bank 01; EPC/UII

10_{bin} Bank 10; TID (read-only code)

11_{bin} Bank 11; user memory

The <FilterAddress> parameter specifies the start address of the memory area to which the filter condition is applied. The address is related to individual bits in the memory. The following tables show the structure of the different memory banks with the associated values for <FilterAddress>. The tables do not include the complete structure of the memory banks.

The <LengthMask> parameter specifies length of the filter mask. The length is expressed in bits.

The filter data can be found in the <Mask Data Byte> parameters. The length is variable and is specified by <LengthMask>.

Bank 01 Memory Structure (EPC/UII):

Function	Address No. (hex) box	Bit number															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPC/UII	0#40 - 0#4F	EPC/UII byte 6								EPC/UII byte 5							
	0#30 - 0#3F	EPC/UII byte 4								EPC/UII byte 3							
	0#20 - 0#2F	EPC/UII byte 2								EPC/UII byte 1							

2023-07

Function	Address No. (hex) box	Bit number															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Protocol Control Bits **PC**	0#10 - 0#1F	AFI/RFU0	AFI/RFU1	AFI/RFU2	AFI/RFU3	AFI/RFU4	AFI/RFU5	AFI/RFU6	AFI/RFU7	Toggle	XPC	UMI	Length0	Length1	Length2	Length3	Length4
Cyclic Redundancy Check **CRC**	0#00 - 0#0F	CRC0	CRC1	CRC2	CRC3	CRC4	CRC5	CRC6	CRC7	CRC8	CRC9	CRC10	CRC11	CRC12	CRC13	CRC14	CRC15

Table 8.49

- CRC0 … CRC15: 16 bit checksum for memory bank 01; automatic calculation of the checksum by the device
- Length0 … Length4: EPC length specification in relation to two byte words
- UMI: User Memory Indicator
- XPC: XPC Indicator
- Toggle: 0 (False) → Bank 01 contains an EPC; 1 (True) → Bank 01 contains a UII
- AFI/RFU0 … AFI/RFU7: Attribute Bits
- EPC/UII: Memory area for the EPC or UII; start address for the EPC/UII is 0#20; EPC/UII length coded in Length0 … Length4

Bank 10 Memory Structure (TID):

Function	Address No. (hex) box	Bit number															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Serial Number Segment [15:0]	0#50- 0#5F	SN32	SN33	SN34	SN35	SN36	SN37	SN38	SN39	SN40	SN41	SN42	SN43	SN44	SN45	SN46	SN47
Serial Number Segment [31:16]	0#40 - 0#4F	SN16	SN17	SN18	SN19	SN20	SN21	SN22	SN23	SN24	SN25	SN26	SN27	SN28	SN29	SN30	SN31
Serial Number Segment [47:32]	0#30 - 0#3F	SN0	SN1	SN2	SN3	SN4	SN5	SN6	SN7	SN8	SN9	SN10	SN11	SN12	SN13	SN14	SN15
XTID Header Segment	0#20 - 0#2F	TMDI0	TMDI1	TMDI2	TMDI3	TM0	TM1	TM2	TM3	TM4	TM5	TM6	TM7	TM8	TM9	TM10	TM11
Manufacturer ID **MDID** / /Model Number **TMN**	0#10 - 0#1F	MDID5	MDID6	MDID7	MDID8	TMN0	TMN1	TMN2	TMN3	TMN4	TMN5	TMN6	TMN7	TMN8	TMN9	TMN10	TMN11
Manufacturer ID **MDID**	0#00 - 0#0F	1	1	1	0	0	0	1	0	XTID	S	F	MDID0	MDID1	MDID2	MDID3	MDID4

Table 8.50

- Class Identifier: Contains the value 16#E0 or 16#E2 based on ISO/IEC 15963
- XTID: Extended Tag Identification; 0 → no extended tag identification; 1 → read/write tag has extended tag identification
- S: Security indicator; read/write tag supports additional security-related commands
- XPC: XPC Indicator
- F: File indicator; read/write tag supports the "FileOpen" command
- MDID: Mask Designer Identifier; chip manufacturer
- TMN: Tag Model Number; chip type
- SN: Serial number of read/write tag

Allocation of read/write tag

Manufacturer	MDID (bin)	TMN (bin)	TMN (hex)	Chip
Impinj	000000001	000101110000	16#170	Monza R6-P
		000100001100	16#10C	Monza 4E
		000100000101	16#105	Monza 4QT
		000101100000	16#160	Monza R6
		000100110000	16#130	Monza R5
		000110010001	16#191	M730
		000110010000	16#190	M750
		000101110001	16#171	Monza R6-A
		000101110001	16#171	Monza R6-B
		000100010100	16#114	Monza 4i
		000100000000	16#100	Monza 4D
		000101010000	16#150	Monza X-8K
		000101000000	16#140	Monza X-2K
Texas Instruments	000000010			
Alien Technology	000000011	010000010001	16#411	Higgs 2
		010000010010	16#412	Higgs 3
		010000010100	16#414	Higgs 4
		100000010001	16#811	Higgs EC
Intelleflex	000000100			
Atmel	000000101			
NXP Semiconductors	000000110	000000000001	16#001	G2
		000000000011	16#003	G2 XM
		000000000100	16#004	G2 XL
		100000000110	16#806	G2 iL
		100000000111	16#807	G2 iL+
		100000001010	16#80A	G2 iM
		100000001011	16#80B	G2 iM+
		100000010000	16#810	UCODE 7
		100000010001	16#811	UCODE 7m
		100000010010	16#812	UCODE 7xm
		110100010010	16#D12	UCODE 7xm
		110110010010	16#D92	UCODE 7xm+
		110000010010	16#C12	UCODE DNA TRACK
		111110010010	16#F92	UCODE DNA
		100010010100	16#894	UCODE 8
		100110010100	16#994	UCODE 8m
ST Microelectronics	000000111	001001000000	16#240	XRAG2
EM Microelectronics	000001011			
Renesas Technology Group	000001100			
Mstar	000001101			

2023-07

Manufacturer	MDID (bin)	TMN (bin)	TMN (hex)	Chip
Tyco International	000001110			
Quanray Electronics	000001111			
Fujitsu	000010000			

Table 8.51

Bank 11 memory structure (user memory):

Function	Address No. (hex) box	Bit number															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
User memory	0#30 - 0#3F	User memory byte 8 (16#44)								User memory byte 7 (16#43)							
	0#20 - 0#2F	User memory byte 6 (16#42)								User memory byte 5 (16#41)							
	0#10 - 0#1F	User memory byte 4 (16#34)								User memory byte 3 (16#33)							
	0#00 - 0#0F	User memory byte 2 (16#32)								User memory byte 1 (16#31)							

Table 8.52

Activate Filter Mask (MF)

The "Activate Filter Mask" (MF) command activates the filter function of the device. The command code is 16#CB. 2 different filter modes can be activated.

Command:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / 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#CB							
Byte 6	Parameter/data	<FilterMode>							
Byte 7	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.53

Response for when filter is activated:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / 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#CB							

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

Table 8.54

The <FilterMode> parameter sets the mode for filtering. The following settings are possible:

- 16#00: Filter switched off
- 16#01: Filter mode 1
- 16#02: Filter mode 2

Filter Mode 1

In total, there are 15 tags in the device sensing range, divided into 5 groups of tags marked as A, B, or C. The filter is now set to "B" by the command FI.

If you execute command MF11 (Activate filter—mode 1), this command affects all subsequent read/write commands.

If a write command is subsequently executed, all "B" tags in the sensing range are selected and are assigned a "Selected" flag. The write command is executed only for tags with a "Selected" flag.

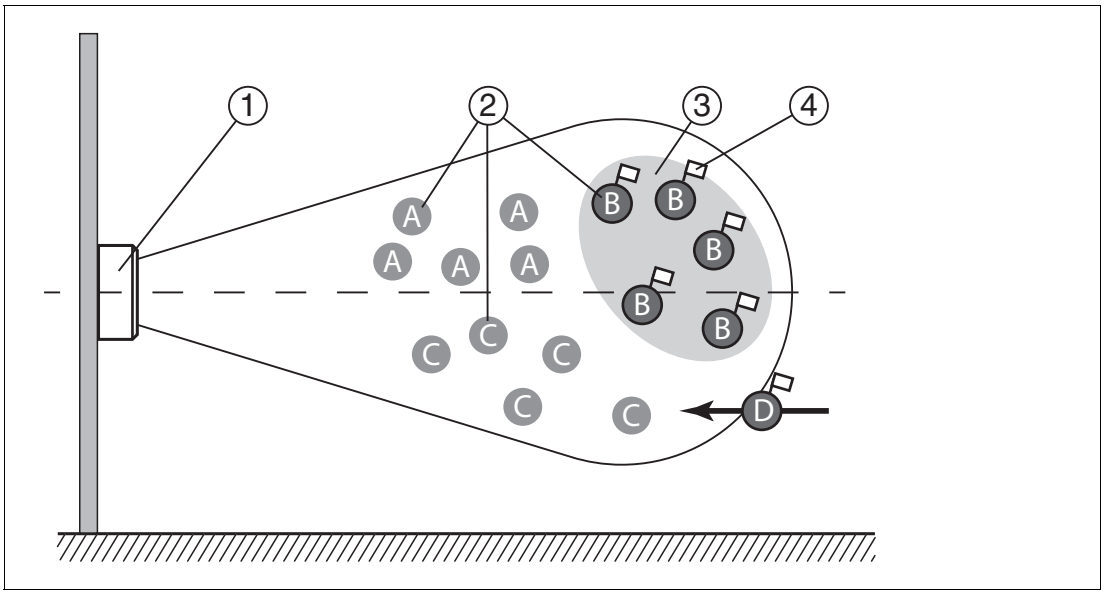


Figure 8.1

When the filter is set to "B", command MF11 (Filter activated—mode 1) selects all "B" tags; the subsequent commands address the selected "B" tags.

1 Device

2 Tags "A," "B," "C"

3 Filter mask

4 Selected flag

2023-07

Filter Mode 2

In total, there are 15 tags in the device sensing range, divided into 5 groups of tags marked as A, B, or C. The filter is set to "B" by the FI command.

If you execute command MF12 (Activate filter—mode 2), this command affects all subsequent commands.

If a write command is subsequently executed, all tags in the sensing range that are not "B" tags are selected. These tags are assigned a "Selected" flag. The write command is executed only for tags with no "Selected" flag.

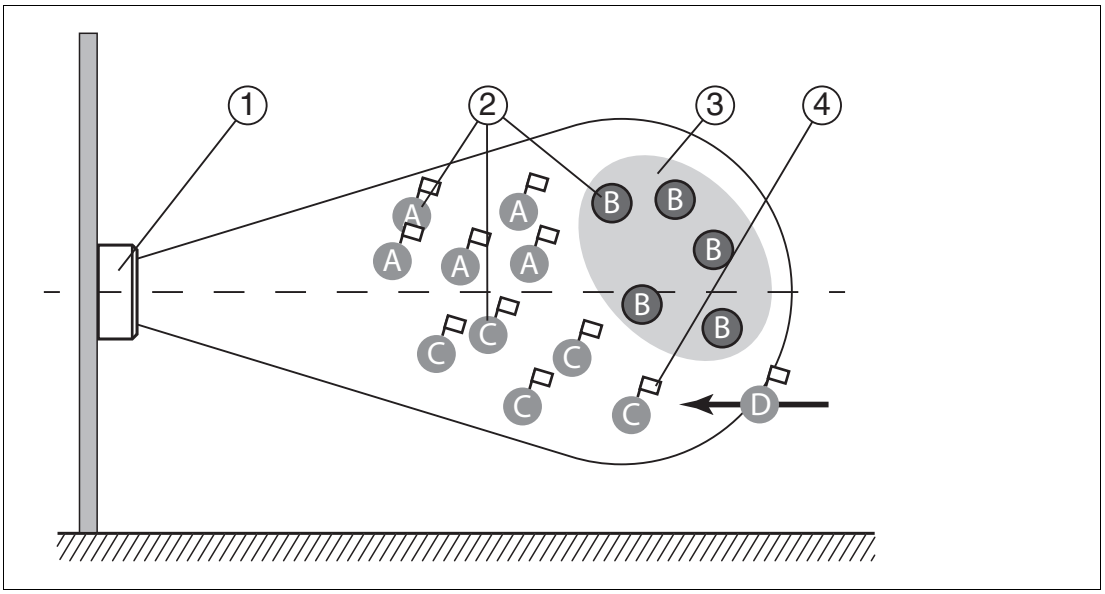


Figure 8.2

When the filter is set to "B," command MF12 (Filter activated—mode 2) selects all "A" and "C" tags, and the subsequent commands address the **unselected** "B" tags.

1 Device

2 Tags "A," "B," "C"

3 Filter mask inverted

4 Selected flag

Differences Between Filter Mode 1 and Filter Mode 2

In both modes, the following commands are only applied to B tags within the sensing range.

In filter mode 1, B tags are assigned a "Selected" flag. In filter mode 2, B tags are not assigned a "Selected" flag.

If a D tag with a "Selected" flag is moved from the area of a neighboring device to the detection zone again, this D tag will execute any subsequent commands in mode 1. However, this D tag will not execute subsequent commands in mode 2.

8.6.4 UHF Configuration Commands

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

On delivery, the parameter values for the UHF interface are preset. The factory setting varies depending on the device version.

All parameter values are stored in a non-volatile memory and remain unchanged after a power interruption.

A system code is required to access the UHF parameters on the device. This distinguishes between other systems in which parameters can be changed. This device uses the system code "U" (16#55).

8.6.4.1 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 UHF 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#55 "U"							
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>							
...	...	...							
...	Parameter data byte X	<Parameter Data Byte X>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.55

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#55 ("U").

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.

All other bytes of the command fragment are not relevant to the command. They must all be set to 16#00.

For example: Command telegram to read the Transmit Power PT (16#5054)

Name	Address	Code	Observation value	Taxable value
"ControlByte_Out"	%QB0	Bin	2#1110_0000	2#1110_0000
"FrameLength_Qut"	%QB1	Hex	16#0B	16#0B
"FragmentationCounter_Out"	%QB2	Hex	16#00	16#00
"TelegramLength_High_Out"	%QB3	Hex	16#00	16#00
"TelegramLength_Low_Out"	%QB4	Hex	16#08	16#08
"Command_Out"	%QB5	Hex	16#BE	16#BE
"SystemCode"	%QB6	Zeic...	'U'	'U'
"Parameter_High"	%QB7	Zeic...	'P'	'P'
"Parameter_Low"	%QB8	Zeic...	'T'	'T'
"LengthParam_High"	%QB9	Hex	16#00	16#00
"LengthParam_Low"	%QB10	Hex	16#00	16#00

Figure 8.3

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>							
...	...	...							
...	Parameter data byte X	<Parameter Data Byte X>							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.56

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.

For example: Response telegram to read Transmit Power PT (16#5054)

Name	Address	Code	Observation value
"ControlByte_In"	%IB0	Bin	2#1010_0000
"FrameLength_In"	%IB1	Hex	16#09
"FragmentationCounter_In"	%IB2	Hex	16#00
"TelegramLength_High_In"	%IB3	Hex	16#00
"TelegramLength_Low_In"	%IB4	Hex	16#06
"Command_In"	%IB5	Hex	16#BE
"Status_In"	%IB6	Hex	16#00
	%IB7	Hex	16#00
	%IB8	Hex	16#32

Figure 8.4

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 UHF 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#55 "U"							
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)>							

2023-07

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

Table 8.57

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#55 ("U").

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.

All other bytes of the command fragment are not relevant for this command. They must all be set to 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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.58

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

For example: Response telegram to change the Transmission Power PT (16#5054)

Name	Address	Code	Observation value
"ControlByte_In"	%IB0	Bin	2#1010_0000
"FrameLength_In"	%IB1	Hex	16#07
"FragmentationCounter_In"	%IB2	Hex	16#00
"TelegramLength_High_In"	%IB3	Hex	16#00
"TelegramLength_Low_In"	%IB4	Hex	16#04
"Command_In"	%IB5	Hex	16#BF
"Status_In"	%IB6	Hex	16#00

Figure 8.5

8.6.4.2 Overview of UHF Parameters

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

Parameter abbreviation	Parameter name	Read/write
PT 16#5054	Power transmit (PT) see "Transmit Power (PT)" on page 109 Sets the transmission power of the device	Read + write
NT 16#4E54	Number of tags (NT) see "Number of Tags (NT) / Cancellation Criterion" on page 113 Cancellation criterion for read/write command	Read + write
AP 16#4150	Antenna polarization (AP) see "Antenna Polarization (AP)" on page 115 Sets the polarization plane of the device	Read
TA 16#5441	Tries allowed (TA) see "Tries Allowed (TA) / Number of Access Attempts" on page 116 Number of access attempts on the read/write tag	Read + write
CD 16#4344	Transmission channels (CD) see "Transmission Channels (CD)" on page 118 Sets the transmission channels used	• Device version FR1*: read + write • Device version FR2*: read
NC 16#4E43	Number of channels (NC) see "Number of Channels (NC)" on page 125 Sets the transmission channels used	• Device version FR1*: read • Device version FR2*: read + write

2023-07

Parameter abbreviation	Parameter name	Read/write
E5 16#4535	Tag Lost Smoothing 5 (E5) see "Tag Lost Smoothing (E5)" on page 121 Sets the number of unsuccessful access attempts on a read/write tag before a status 5 message	Read + write
MB 16#4D42	Memory bank (MB) see "Memory Bank (MB)" on page 123 Sets the memory bank for various read/write commands	Read + write
QW 16#5157	Q Value QW see "Q Value QW" on page 128 Sets the number of time slots for accessing a read/write tag	Read + write
RC 16#5243	Region code (RC) see "Region Code (RC)" on page 130 Country identifier	Read
FL 16#464C	Reading the filter mask	Read

Table 8.59

The parameters are saved in a non-volatile memory.

Transmit Power (PT)

The "Transmit Power" (PT) parameter can be used to change the power of the device. The transmit power is given in mW. and affects the detection range of a tag's detection zone. As the transmission power increases, the detection range for read and write access to a tag is increased to a certain extent.

A transmit power level of PT1 is preset in the factory setting. It is possible to use up to 5 power levels (PT1 … PT5) in the device. This enables a ramp to be generated for a continuous increase in the transmit power. When used in conjunction with a "Single Read" or "Single Write" command and the definition of the cancellation criterion (NT parameter, Number of Tags), the read or write operation is canceled as soon as the set number of read/write tags is detected.

Setting multiple transmit power levels increases the execution time of a read or write operation. This is due to the inventory runs that are executed for each power level. The NT cancellation criterion can be set to reduce the execution time.

Parameter character	PT (16#5054)
Length of PTx parameter value	2 bytes
PT1 factory setting	Variable depending on the device version • FR1-01 → 100 mW_{erp} (16#0064) • FR2-02 → 150 mW_{eirp} (16#0096) • FR2-03 → 100 mW_{erp} (16#0064) • FR1-04 → 100 mW_{erp} (16#0064) • FR2-07 → 150 mW_{eirp} (16#0096) • FR2-09 → 150 mW_{eirp} (16#0096) • FR2-10 → 150 mW_{eirp} (16#0096) • FR2-13 → 100 mW_{erp} (16#0064) • FR2-17 → 100 mW_{erp} (16#0064)

Value range FR1-01, FR2-03, FR1-04, FR2-13 and FR2-17	3 mW (16#0003) -> 5 dBm 4 mW (16#0004) -> 6 dBm 5 mW (16#0005) -> 7 dBm 6 mW (16#0006) -> 8 dBm 8 mW (16#0008) -> 9 dBm 10 mW (16#000A) -> 10 dBm 13 mW (16#000D) -> 11 dBm 15 mW (16#000F) -> 12 dBm 20 mW (16#0014) -> 13 dBm 25 mW (16#0019) -> 14 dBm 30 mW (16#001E) -> 15 dBm 40 mW (16#0028) -> 16 dBm 50 mW (16#0032) -> 17 dBm 60 mW (16#003C) -> 18 dBm 80 mW (16#0050) -> 19 dBm 100 mW (16#0064) -> 20 dBm
Value range FR2-02, FR2-07, FR2-09 and FR2-10	3 mW (16#0003) -> 5 dBm 4 mW (16#0004) -> 6 dBm 5 mW (16#0005) -> 7 dBm 6 mW (16#0006) -> 8 dBm 8 mW (16#0008) -> 9 dBm 10 mW (16#000A) -> 10 dBm 13 mW (16#000D) -> 11 dBm 15 mW (16#000F) -> 12 dBm 20 mW (16#0014) -> 13 dBm 25 mW (16#0019) -> 14 dBm 30 mW (16#001E) -> 15 dBm 40 mW (16#0028) -> 16 dBm 50 mW (16#0032) -> 17 dBm 60 mW (16#003C) -> 18 dBm 80 mW (16#0050) -> 19 dBm 100 mW (16#0064) -> 20 dBm 125 mW (16#007D) -> 21 dBm 150 mW (16#0096) -> 22 dBm

⊕ Higher detection range if you increase the transmit power.

⊖ Potential excessive detection ranges if you increase the transmission power.

⊖ Adjacent devices may be affected if the detection range increases.

Tip

The highest transmit power does not necessarily lead to the largest read range. Vary the transmit power to achieve optimal read results.

Note

You can operate the device only with the internally specified transmit power levels. You can use the Write PT command in the software to enter one or more transmit power levels within the specified value range. The device automatically sets the transmission power to the next lowest value available.

Any entries outside the specified value range are returned as errors with status value 16#04.

Tip

The device can be parameterized with multiple transmit power levels:

The Write PT command can be used to set power levels PT1, PT2, and PT3 to the values 20 mW, 50 mW, and 100 mW.

- NT cancellation criterion at 16#FF (factory setting):
  Each read and write command is executed in succession for all 3 transmit power values. If one or more tags are found and successfully read/written for the first transmit power, the command is still executed with all other transmit powers to reach any other tags that may be available.
- NT cancellation criterion at 16#01 (canceled after one read/write tag):
  If no read/write tags were detected during the inventory runs, the transmit power is set to the set next higher value. As soon as at least one read/write tag has been detected in an inventory round, the write/read command stops.
  You can specify a maximum of 5 transmit power values. If several additional transmitting channels are selected (see "Transmission Channels (CD)" on page 118), all set powers are executed on each transmitting channel for each read or write command.

For example: Command telegram to change the transmit power to PT1 = 50 mW (16#0032)

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#55 "U"							
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	PT1 parameter (high byte)	16#00							
Byte 12	PT1 parameter (low byte)	16#32							
Byte 13	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.60

For example: Command telegram to change the transmit power to PT1 = 20 mW (16#0014), PT2 = 50 mW (16#0032) and PT3 = 100 mW (16#0064)

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	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 4	Telegram length (low byte)	16#0E							
Byte 5	Command	16#BF							
Byte 6	System code	16#55 "U"							
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#06							
Byte 11	PT1 parameter (high byte)	16#00							
Byte 12	PT1 parameter (low byte)	16#14							
Byte 13	PT2 parameter (high byte)	16#00							
Byte 14	PT2 parameter (low byte)	16#32							
Byte 15	PT3 parameter (high byte)	16#00							
Byte 16	PT3 parameter (low byte)	16#64							
Byte 17	Not relevant	16#00							
...	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#0A							
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#55 "U"							
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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.62

For example: Response telegram with the transmit power set to PT1 = 50 mW (16#0032)

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	PT1 parameter (high byte)	16#00							
Byte 8	PT1 parameter (low byte)	16#32							
Byte 9	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.63

Number of Tags (NT) / Cancellation Criterion

The NT parameter specifies the number of tags that the device searches for within the sensing range. Each command is repeated in accordance with the power transmit (PT), transmission channel (CD) or number of channels (NC), and number of attempts (TA) parameters. 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 characters	NT (16#4E54)
Length of NT parameter value	1 byte
Factory setting	16#FF
Value range	16#01 … 16#14; 16#FF

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	ControlByte / 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#55 "U"							
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	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 10	Length parameter (low byte)	16#01							
Byte 11	NT parameter	16#01							
Byte 12	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.64

Example: Command telegram to read "number of tags"

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / 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#55 "U"							
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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.65

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	ControlByte / 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	NT parameter	16#FF							
Byte 8	Not relevant	16#00							
...	Not relevant	16#00							

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

Table 8.66

Antenna Polarization (AP)

Read the device's polarization mode using the antenna polarization (AP) parameter. The integrated antenna supports circular polarization.

Parameter characters	AP (16#4150)
Length of NT parameter value	1 byte
Factory setting	"L" (left-hand circular polarization) = 16#4C

The circular polarization ensures that the tags can be identified independently of their polarization and orientation.

Example: Command telegram to read the antenna polarization

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / 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#55 "U"							
Byte 7	Parameter name (high byte)	16#41 "A"							
Byte 8	Parameter name (low byte)	16#50 "P"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#00							
Byte 11	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.67

Example: Response telegram with the set value of the antenna polarization "L" (left-hand circular)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / 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	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 6	Status	16#00							
Byte 7	AP parameter	16#4C "L"							
Byte 8	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.68

Tries Allowed (TA) / Number of Access Attempts

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

Parameter characters	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 cancellation criterion. This will stop the command execution as soon as the set number of read/write tags has been identified.

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	ControlByte / 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#55 "U"							
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 12	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.69

2023-07

Example: Command telegram to read TA

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / 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#55 "U"							
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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.70

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	ControlByte / 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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.71

Transmission Channels (CD)

The Transmission Channels (CD) parameter is used to access the transmission channels that the device uses.

- Device version FR1*:
  These devices use a specific frequency list. The CD parameter allows the user to define the transmission channels and their sequence.
- Device version FR2*:
  These device versions use frequency hopping spread spectrum. The CD parameter cannot be used to change the transmission channels and their sequence. In this case, the setting can only be read.

Parameter character	CD (16#4344)
Length of CD parameter value	Variable depending on the device version • FR1-01 → Max. 4 bytes; max. 4 channels, adjustable • FR2-02 → 50 bytes; 50 channels, cannot be changed • FR2-03 → 16 bytes; 16 channels, cannot be changed • FR1-04 → Max. 10 bytes; max. 10 channels, adjustable • FR2-07 → 52 bytes; 52 channels, cannot be changed • FR2-09 → 6 bytes; 6 channels, cannot be changed • FR2-10 → 12 bytes; 12 channels, cannot be changed • FR2-13 → 8 bytes; 8 channels, cannot be changed • FR2-17 → 14 bytes; 14 channels, cannot be changed

2023-07

Factory setting	Depending on the device version • FR1-01 → 4, 7, 10, 13 (all 4 channels; length 4 bytes) • FR2-02 → 31, 20, 49, 4, 33, ... (length 50 bytes) • FR2-03 → 3, 16, 13, 10, 7, .. (length 16 bytes) • FR1-04 → 1, 4, 7, 10 (4 channels; length 4 bytes) • FR2-07 → 38, 20, 42, 9, 27, ... (length 52 bytes) • FR2-09 → 1, 7, 13, 4, 10, 16 (length 6 bytes) • FR2-10 → 5, 10, 3, 8, 1, ... (length 12 bytes) • FR2-13 → 1, 6, 3, 8, 5, ... (length 8 bytes) • FR2-17 → 3, 8, 14, 4, 7, ... (length 14 bytes)
Value range	Depending on the device version • FR1-01 → Up to four channels can be set; the number and sequence of channels is freely adjustable 16#04 → Channel 4 16#07 → Channel 7 16#0A → Channel 10 16#0D → Channel 13 • FR2-02 → The parameter values are fixed; no changes can be made; the value range is the factory setting • FR2-03 → The parameter values are fixed; no changes can be made; the value range is the facory setting • FR1-04 → Up to 10 channels can be set; the number and sequence of channels is freely adjustable 16#01 → Channel 1 16#02 → Channel 2 16#03 → Channel 3 16#04 → Channel 4 16#05 → Channel 5 16#06 → Channel 6 16#07 → Channel 7 16#08 → Channel 8 16#09 → Channel 9 16#0A → Channel 10 • FR2-07 → The parameter values are fixed; no changes can be made; the value range is the factory setting • FR2-09 → The parameter values are fixed; no changes can be made; the value range is the factory setting • FR2-10 → The parameter values are fixed; no changes can be made; the value range is the factory setting • FR2-13 → The parameter values are fixed; no changes can be made; the value range is the factory setting • FR2-17 → The parameter values are fixed; no changes can be made; the value range is the factory setting

If you have parameterized several transmission channels with the CD parameter, the device executes each write or read command on all defined transmission channels sequentially. If several additional transmission power values are parameterized, all set transmission powers are executed on each transmission channel for each read or write command.

The number of parameterized transmission channels affects the execution time of read/write commands. In the FR1-01 device version, the execution time can be decreased by reducing the number of transmission channels. If only one transmission channel should be used, we recommend that you set one of the two channels in the middle of the frequency range, i.e., channel 7 or channel 10. When parameterizing 2 transmission channels, the two channels outside the frequency range must be set, i.e., channel 4 and channel 13.

For example: Command telegram to change the CD parameter on channel 4 and channel 13

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#55 "U"							
Byte 7	Parameter name (high byte)	16#43 "C"							
Byte 8	Parameter name (low byte)	16#44 "D"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#02							
Byte 11	CD parameter	16#04							
Byte 12	CD parameter	16#0D							
Byte 13	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.72

For example: Command telegram to read CD

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#55 "U"							
Byte 7	Parameter name (high byte)	16#43 "C"							
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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.73

For example: Response telegram with the set sequence for channels 4, 10, 7, and 13

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	CD parameter	16#04							
Byte 8	CD parameter	16#0A							
Byte 9	CD parameter	16#07							
Byte 10	CD parameter	16#0D							
Byte 11	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.74

Tag Lost Smoothing (E5)

The Tag Lost Smoothing (E5) parameter can be used to set the number of unsuccessful write/read attempts when executing an Enhanced command before the device outputs the telegram with the status value 16#05. The device uses a status 16#05 telegram to indicate 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 write attempts until status 16#05 telegram is output
Value range	16#00 … 16#FC → 0 to a maximum of 10 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 status 16#05 telegrams received. In dynamic applications, gaps in the sensing range can therefore be bridged without receiving a status 16#05 message if there are minor interruptions in read/write tag communication. This makes the detection zone more homogeneous.

If a large number of read/write tags are detected at the same time during a dynamic application, the receipt of the status 16#05 telegrams can be delayed by increasing the parameter value of E5. The status 16#00 telegrams containing the information read from the tags are transmitted first. The status 16#05 messages indicating that the read/write tag has left the sensing range are transmitted second.

Reducing the parameter value of 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
- Status 16#0A: Multiple read/write tags with the same EPC detected

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#55 "U"							
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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.75

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#55 "U"							
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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.76

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

Table 8.77

Memory Bank (MB)

The Memory Bank (MB) parameter can be used to select a tag memory bank, which is accessed by the read/write commands for 4 byte blocks and 2 byte words.

The memory bank setting has an impact on the data access of the following commands:

- Single / Enhanced Read 4-Byte Blocks
- Single / Enhanced Write 4-Byte Blocks
- Single / Enhanced Read 2-Byte Words
- Single / Enhanced Write 2-Byte Words

The MB parameter setting does not have any effect on the execution of the following read/write commands.

- Single / Enhanced Read-Only Code (access to the TID)
- Single / Enhanced Read-Only Code (access to EPC/UII)
- Single Write EPC/UII (access to EPC/UII)

Parameter character	MB (16#4D42)
Length of MB parameter value	1 byte
Factory setting	16#03 → Access to the user memory bank
Value range	16#00 → Access to the memory bank with the password section (bank 00) 16#01 → Access to the EPC/UII memory bank (bank 01) 16#02 → Access to the TID memory bank (bank 10) 16#03 → Access to the user memory bank (bank 11)

For example: Command telegram to change the MB settings for accessing the EPC/UII memory bank

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	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 4	Telegram length (low byte)	16#09							
Byte 5	Command	16#BF							
Byte 6	System code	16#55 "U"							
Byte 7	Parameter name (high byte)	16#4D "M"							
Byte 8	Parameter name (low byte)	16#42 "B"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#01							
Byte 11	MB parameter	16#01							
Byte 12	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.78

For example: Command telegram to read the MB parameter 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#55 "U"							
Byte 7	Parameter name (high byte)	16#4D "M"							
Byte 8	Parameter name (low byte)	16#42 "B"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#00							
Byte 11	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.79

For example: Response telegram with the set value (16#03 -> User Memory) of the MB parameter

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							

2023-07

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 5	Command	16#BE							
Byte 6	Status	16#00							
Byte 7	MB parameter	16#03							
Byte 8	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.80

Number of Channels (NC)

The NC parameter indicates the number of frequency channels on which a one-time write or read attempt is performed. Access to the NC parameter depends on the device version used.

- Device version FR1*:
  These devices use a specific frequency list. The CD parameter is used to set the frequency list for this device. With the FR1-01 device version, the NC parameter can only be read. It is not possible to use the NC parameter to change the frequency list. Changes must be made using the CD parameter.
- Device versions FR2*:
  These device versions use a frequency hopping spread spectrum. With these devices, the NC parameter can be used to set the number of frequency channels. The NC parameter can be read and changed. All available frequency channels are always used, but only as many per single command as specified by the NC parameter.

Parameter character	NC (16#4E43)
Length of NC parameter value	1 byte

Factory setting	Depending on the device version • FR1-01 → 16#04 (four channels; channel 4, 7, 10 and 13) • FR2-02 → 16#32 (50 channels; channel 1, 2, 3, ... 50) • FR2-03 → 16#10 (16 channels; channel 2, 3, 4, ... 17) • FR1-04 → 16#04 (four channels; channel 1, 4, 7, 10) • FR2-07 → 16#34 (52 channels; channel 1, 2, 3, ... 52) • FR2-09 → 16#06 (6 channels; channel 1, 4, 7, 10, 13, 16) • FR2-10 → 16#0C (12 channels; channel 1, 2, 3, ...12) • FR2-13 → 16#08 (8 channels; channel 1, 2, 3, ...8) • FR2-17 → 16#0E (14 channels; channel 1, 2, 3, ... 14)
Value range	Depending on the device version • FR1-01 → 16#01 … 16#04 The value depends on the CD parameter settings. When using a frequency channel, the value of the NC parameter is 16#01. If all permitted frequency channels are used, the value is 16#04. • FR2-02 → 16#01 … 16#32 The number of channels used can be between 1 and 50. • FR2-03 → 16#01 … 16#10 The number of channels used can be between 1 and 16. • FR1-04 → 16#01 ... 16#0A The value depends on the CD parameter settings. When using a frequency channel, the value of the NC parameter is 16#01. If all permitted frequency channels are used, the value is 16#0A. • FR2-07 → 16#01 ... 16#34 The number of channels used can be between 1 and 52. • FR2-09 → 16#01 ... 16#06 The number of channels used can be between 1 and 6. • FR2-10 → 16#01 ... 16#0C The number of channels used can be between 1 and 12. • FR2-13 → 16#01 ... 16#08 The number of channels used can be between 1 and 8. • FR2-17 → 16#01 ... 16#0E The number of channels used can be between 1 and 14.

If you have parameterized several transmission channels with the NC parameter, the device executes each write or read command on all defined transmission channels sequentially. If several additional transmission power values are parameterized, all set transmission powers are executed on each transmission channel for each read or write command.

The number of transmission channels used affects the execution time of read/write commands. With the device version FR2*, the execution time of read/write commands can be shortened by reducing the number of frequency channels.

Example

The FR2-03 (China) device version uses a total of 16 channels for communication. Channels 2 … 17 are used for this. When the NC parameter is set to 16#06, i.e., 6 channels, the following frequency channels are used when executing single commands:

- 1st single command: Channel 3, 16, 13, 10, 7, and 4
- 2nd single command: Channel 17, 14, 11, 8, 5, and 2
- 3rd single command: Channel 15, 12, 9, 6, 3, and 16
- etc.

For example: Command telegram to change the NC 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#55 "U"							
Byte 7	Parameter name (high byte)	16#4E "N"							
Byte 8	Parameter name (low byte)	16#43 "C"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#01							
Byte 11	NC parameter	16#0A							
Byte 12	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.81

For example: Command telegram to read the NC 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#55 "U"							
Byte 7	Parameter name (high byte)	16#4E "N"							
Byte 8	Parameter name (low byte)	16#43 "C"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#00							
Byte 11	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.82

For example: Response telegram with the NC parameter value set (16#32; 50 channels)

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	NC parameter	16#32							
Byte 8	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.83

Q Value QW

The Slotted ALOHA principle is used to transfer data between one or more read/write tags via the air interface. The communication time period between the device and read/write tag is divided into time slots. To prevent collisions caused by the simultaneous communication of multiple tags, the number of time slots should correspond to the number of expected tags. The QW parameter defines the number of time slots as 2^Q.

Parameter characters	QW (16#5157)
Length of QW parameter value	1 byte
Factory setting	16#02 → 2^2 = 4 time slots are used
Value range	16#00 … 16#04 → between 2^0 ... 2^4 (1 ... 16) time slots can be set.

The number of time slots used affects the execution time of read/write commands. Reducing the time slots shortens the execution time for accessing read/write tags.

Example: Command telegram to change the QW setting to a value of 1 (16#01), i.e., 2 time slots

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / 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#55 "U"							
Byte 7	Parameter name (high byte)	16#51 "Q"							
Byte 8	Parameter name (low byte)	16#57 "W"							

2023-07

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

Table 8.84

Example: Command telegram to read the QW settings

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / 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#55 "U"							
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							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.85

Example: Response telegram with the set value of the QW parameter (16#02; 4 time slots)

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

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

Table 8.86

Region Code (RC)

The Region Code (RC) parameter can be used to read the country code of the device. The country code specifies the country-specific setting of the device.

The country-specific settings are preset during production; the RC parameter cannot be changed. The parameter value of RC can only be read.

Parameter characters	RC (16#5243)
Length of RC parameter value	2 bytes
Factory setting	Depending on the device version • FR1-01 → 16#0001 (Europe and other countries that follow EN 302208) • FR2-02 → 16#0002 (USA, Canada, Mexico, Argentina) • FR2-03 → 16#0003 (China) • FR1-04 → 16#0004 (India) • FR2-07 → 16#0007 (Brazil) • FR2-09 → 16#0009 (South Korea) • FR2-10 → 16#000A (Australia) • FR2-13 → 16#000D (Malaysia) • FR2-17 → 16#0011 (Taiwan)
Value range	Parameter value cannot be changed

Country Identifiers

Country Identifier	Occupied Frequency Bandwidth Frequency Access Method	Country or Region
16#01	865 MHz ... 868 MHz Parameterizable frequency list	EU and other countries subject to EN 302208
16#02	902 MHz ... 928 MHz Frequency hopping spread spectrum	USA, Canada, Mexico, Argentina
16#03	920 MHz ... 925 MHz Frequency hopping spread spectrum	China
16#04	865,0 MHz ... 867,0 MHz Parameterizable frequency list	India
16#07	915 MHz ... 928 MHz Frequency hopping spread spectrum	Brazil
16#09	917,2 MHz ... 920,4 MHz Frequency hopping spread spectrum	South Korea
16#0A	920 MHz ... 926 MHz Frequency hopping spread spectrum	Australia

2023-07

Country Identifier	Occupied Frequency Band-width Frequency Access Method	Country or Region
16#0D	919 MHz … 923 MHz Frequency hopping spread spectrum	Malaysia
16#11	920 MHz ... 928 MHz Frequency hopping spread spectrum	Taiwan

Table 8.87

Example: Command telegram for reading the regional code

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / 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#55 "U"							
Byte 7	Parameter name (high byte)	16#52 "R"							
Byte 8	Parameter name (low byte)	16#43 "C"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#00							
Byte 11	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.88

Example: Response telegram with the set value of the region code 16#0001 (Europe)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / 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	RC parameter (high byte)	16#00							
Byte 8	RC parameter (low byte)	16#01							
Byte 9	Not relevant	16#00							
...	Not relevant	16#00							

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

Table 8.89

Filter List (FL) / Read Filter Mask

The FL parameter contains the current configuration of the filter masks as set during execution of the FI command. The output format corresponds to the data input format of the FI command without the filter number. The parameter can be read, but not set.

Parameter characters	FL
Value range	0 ... 2

Example: Command telegram for reading filter mask 2 (FL = 16#464C)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / 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#BE							
Byte 6	System code	16#55 "U"							
Byte 7	Parameter name (high byte)	16#46 "F"							
Byte 8	Parameter name (low byte)	16#4C "L"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#01							
Byte 11	Filter number	16#02							
Byte 12	Not relevant	16#07							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.90

Example: Response telegram after reading out filter mask 2 (Memory Bank = EPC/UII; Negation = 0; Logical Operation = 0 "OR"; Truncation = 0 "Send all")

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame length	D_S	U_M	U_D	0	<FrameLength>			
Byte 1	Frame length	<FrameLength>							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	<Telegram Length (High Byte)>							
Byte 4	Telegram length (low byte)	<Telegram Length (Low Byte)>							
Byte 5	Command	16#BE							
Byte 6	Status	16#00							

2023-07

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 7	Memory bank / Negation / Logical operation / Truncation	$2\#00001000_{bin}$							
Byte 8	Bit address (high byte)	16#00							
Byte 9	Bit address (low byte)	16#20 "Start at EPC/UII"							
Byte 10	Length mask	16#08 "8 Bit"							
Byte 11	Reserved	16#00							
Byte 12	Mask data	16#E2							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Table 8.91

The filter setting of filter mask 2 has a length of 1 byte, a content of 16#E2 and starts from memory bank 01 (EPC/UII) from address 16#20. Only read/write tags for which the EPC/UII begins with 16#E2 are transferred by this mask.

Content	Bit number							
	7	6	5	4	3	2	1	0
<Memory bank / negation / Logical operation / Truncation>	0	0	0	<MemBank>		<Negation>	<LogicalOperation>	<Truncation>

<MemBank>

00 not used
01 EPC
10 TID (read-only code)
11 User

<Negation>

0 not negated
1 negated

If you negate the filter, all tags that do not fit into the filter are transferred.

<LogicalOperation>

0 OR
1 AND

The <Logical Operation> bit logically links different filters together. If only one filter is used, this bit is not relevant.

<Truncation>

0 send all
1 Send only part of the EPC or UII following the filter mask

Bit address Start address of the bit at which the filter mask begins. Defined as a hexadecimal value.

For the address of memory bank 01 (EPC/UII) see chapter 3.2.3.

8.7 Error / Status Messages

Status	Meaning
16#00	The command was executed correctly. With a read command, this telegram contains the EPC code of the read-in read/write tag and any additional information. With a write command, this telegram contains the EPC code of the newly written tag. When this telegram is received, the data is securely written to the tag.
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. This status message is generated if a physically non-existent memory section needs to be accessed. Example: The volume of data to be read in when executing the "Read 4-Byte Blocks" command is greater than the memory available.
16#05	The read/write tag has left the sensing range. When an enhanced read or write command is executed, this status message indicates that a read/write tag has left the sensing range. This telegram contains the EPC code of the read/write tag. If there is no read/write tag in the sensing range immediately after an enhanced command is started, this status message is sent without an EPC code.
16#0A	Multiple read/write tags with the same EPC code within the sensing range During execution of a read or write command, only tags with different EPC codes may be present in the sensing range. This status message indicates that an identical EPC code of more than one tag has been detected. This telegram contains the EPC code of the additional read/write tag.
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	End message when executing single commands This message indicates that single read or single write commands have been completed. This telegram contains the number of read/write tags read or written during command execution.

9 Service and Maintenance

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

10 Troubleshooting

Problem	Solution
Interference from several devices in the direct vicinity	• Check the distance from other devices. • Change the setting of the transmission channels. • Reduce the transmission power.
Status A message	• Check whether there are multiple tags in the sensing range: Remove the tag from the sensing range by placing the tag e.g., in a sealed metal container. Repeat the read or write operation. • Use filter commands. • Determine whether multiple tags have the same UII/EPC.

Table 10.1

11 Appendix

11.1 Dimensions

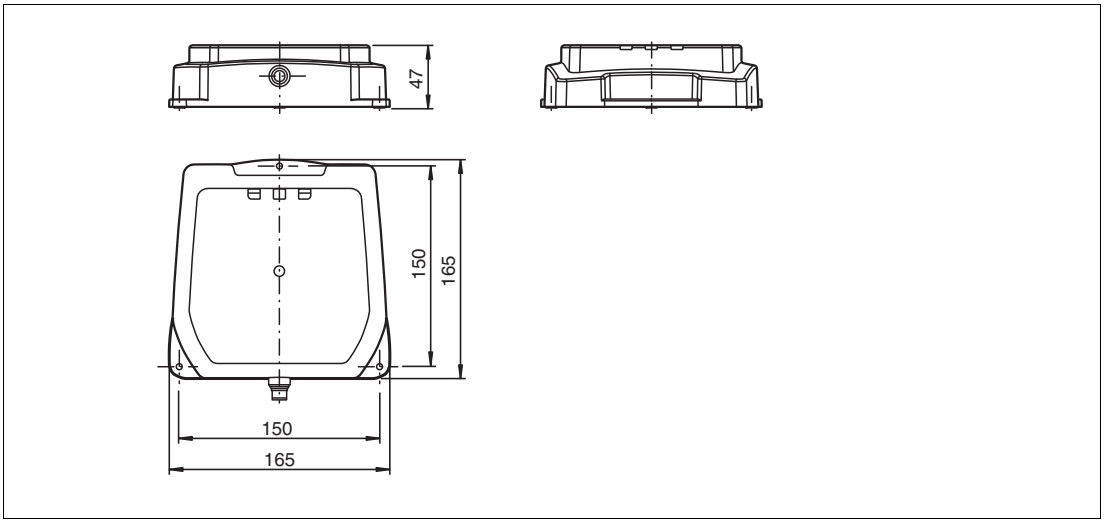


Figure 11.1

11.2 Examples for Expert Mode

11.2.1 Single Read EPC/UII

"Single Read Fixcode" data telegram for a tag with a 12 byte EPC
(16#30 14 F7 33 7C 00 1F 00 00 00 74 83)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame length	16#17							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#14							
Byte 5	Command	16#D2							
Byte 6	Status	16#00							
Byte 7	Length EPC/UII	16#00							
Byte 8	Length EPC/UII	16#0E							
Byte 9	PC word	16#34							
Byte 10	PC word	16#00							
Byte 11	EPC byte 1	16#30							
Byte 12	EPC byte 2	16#14							
Byte 13	EPC byte 3	16#F7							
Byte 14	EPC byte 4	16#33							
Byte 15	EPC byte 5	16#7C							
Byte 16	EPC byte 6	16#00							
Byte 17	EPC byte 7	16#1F							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 18	EPC byte 8	16#00							
Byte 19	EPC byte 9	16#00							
Byte 20	EPC byte 10	16#00							
Byte 21	EPC byte 11	16#74							
Byte 22	EPC byte 12	16#83							
Byte 23	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

11.2.2 Enhanced Read EPC/UII

"Enhanced Read Fixcode" data telegram for a tag with a 12 byte EPC (16#30 14 F7 33 7C 00 1F 00 00 00 74 83)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame length	16#17							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#14							
Byte 5	Command	16#D3							
Byte 6	Status	16#00							
Byte 7	Length EPC/UII	16#00							
Byte 8	Length EPC/UII	16#0E							
Byte 9	PC word	16#34							
Byte 10	PC word	16#00							
Byte 11	EPC byte 1	16#30							
Byte 12	EPC byte 2	16#14							
Byte 13	EPC byte 3	16#F7							
Byte 14	EPC byte 4	16#33							
Byte 15	EPC byte 5	16#7C							
Byte 16	EPC byte 6	16#00							
Byte 17	EPC byte 7	16#1F							
Byte 18	EPC byte 8	16#00							
Byte 19	EPC byte 9	16#00							
Byte 20	EPC byte 10	16#00							
Byte 21	EPC byte 11	16#74							
Byte 22	EPC byte 12	16#83							
Byte 23	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

2023-07

Tag with 12-byte EPC (16#30 14 F7 33 7C 00 1F 00 00 00 74 83) has left the sensing range

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame length	16#17							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#14							
Byte 5	Command	16#D3							
Byte 6	Status	16#05							
Byte 7	Length EPC/UII	16#00							
Byte 8	Length EPC/UII	16#0E							
Byte 9	PC word	16#34							
Byte 10	PC word	16#00							
Byte 11	EPC byte 1	16#30							
Byte 12	EPC byte 2	16#14							
Byte 13	EPC byte 3	16#F7							
Byte 14	EPC byte 4	16#33							
Byte 15	EPC byte 5	16#7C							
Byte 16	EPC byte 6	16#00							
Byte 17	EPC byte 7	16#1F							
Byte 18	EPC byte 8	16#00							
Byte 19	EPC byte 9	16#00							
Byte 20	EPC byte 10	16#00							
Byte 21	EPC byte 11	16#74							
Byte 22	EPC byte 12	16#83							
Byte 23	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

11.2.3 Single Write EPC/UII

Command telegram to program a 12-byte EPC (16#11 22 33 44 55 66 77 88 99 00 11 22)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame length	16#15							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length	16#00							
Byte 4	Telegram length	16#12							
Byte 5	Command	16#CE							
Byte 6	Fix length	16#0E							
Byte 7	PC word	16#30							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 8	PC word	16#00							
Byte 9	Write EPC byte 1	16#11							
Byte 10	Write EPC byte 2	16#22							
Byte 11	Write EPC byte 3	16#33							
Byte 12	Write EPC byte 4	16#44							
Byte 13	Write EPC byte 5	16#55							
Byte 14	Write EPC byte 6	16#66							
Byte 15	Write EPC byte 7	16#77							
Byte 16	Write EPC byte 8	16#88							
Byte 17	Write EPC byte 9	16#99							
Byte 18	Write EPC byte 10	16#00							
Byte 19	Write EPC byte 11	16#11							
Byte 20	Write EPC byte 12	16#22							
Byte 21	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Command telegram to program a 4-byte UII (16#AA BB CC DD)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / 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#CE							
Byte 6	Fix length	16#06							
Byte 7	PC word	16#11							
Byte 8	PC word	16#00							
Byte 9	Write EPC byte 1	16#AA							
Byte 10	Write EPC byte 2	16#BB							
Byte 11	Write EPC byte 3	16#CC							
Byte 12	Write EPC byte 4	16#DD							
Byte 13	Not relevant	16#00							
Byte 14	Not relevant	16#00							
Byte 15	Not relevant	16#00							

2023-07

11.2.4 Single Read 4-Byte Blocks

Command telegram to read 4 bytes starting from byte address 4

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / 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	16#00							
Byte 4	Telegram length	16#07							
Byte 5	Command	16#10							
Byte 6	ByteAddress	16#00							
Byte 7	ByteAddress	16#04							
Byte 8	Number of bytes	16#00							
Byte 9	Number of bytes	16#04							
Byte 10	Not relevant	16#00							
...	Not relevant	16#00							
Byte 15	Not relevant	16#00							

Data telegram from a tag with a 12-byte EPC (16#30 14 F7 33 7C 00 1F 00 00 00 74 83) and the read-in data (16#31323334 / ASCII "1234") from the user memory

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame length	16#1D							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length	16#00							
Byte 4	Telegram length	16#1A							
Byte 5	Command	16#10							
Byte 6	Status	16#00							
Byte 7	Length EPC/UII	16#00							
Byte 8	Length EPC/UII	16#0E							
Byte 9	PC word	16#34							
Byte 10	PC word	16#00							
Byte 11	EPC byte 1	16#30							
Byte 12	EPC byte 2	16#14							
Byte 13	EPC byte 3	16#F7							
Byte 14	EPC byte 4	16#33							
Byte 15	EPC byte 5	16#7C							
Byte 16	EPC byte 6	16#00							
Byte 17	EPC byte 7	16#1F							
Byte 18	EPC byte 8	16#00							
Byte 19	EPC byte 9	16#00							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 20	EPC byte 10	16#00							
Byte 21	EPC byte 11	16#74							
Byte 22	EPC byte 12	16#83							
Byte 23	Length data	16#00							
Byte 24	Length data	16#04							
Byte 25	User data byte 1	16#31							
Byte 26	User data byte 2	16#32							
Byte 27	User data byte 3	16#33							
Byte 28	User data byte 4	16#34							
Byte 29	Not relevant	16#00							
Byte 30	Not relevant	16#00							
Byte 31	Not relevant	16#00							

11.2.5 Enhanced Read 4-Byte Blocks

Command telegram to read 4 bytes starting from byte address 0

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	16#00							
Byte 4	Telegram length	16#07							
Byte 5	Command	16#19							
Byte 6	ByteAddress	16#00							
Byte 7	ByteAddress	16#00							
Byte 8	Number of bytes	16#00							
Byte 9	Number of bytes	16#04							
Byte 10	Not relevant	16#00							
...	Not relevant	16#00							
Byte 15	Not relevant	16#00							

Data telegram from a tag with a 12 byte EPC (16#30 14 F7 33 7C 00 1F 00 00 00 74 83) and the read user memory data (16#31323334 / ASCII "1234")

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#1D							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length	16#00							
Byte 4	Telegram length	16#1A							

2023-07

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 5	Command	16#19							
Byte 6	Status	16#00							
Byte 7	Length EPC/UII	16#00							
Byte 8	Length EPC/UII	16#0E							
Byte 9	PC Word	16#34							
Byte 10	PC Word	16#00							
Byte 11	EPC Byte 1	16#30							
Byte 12	EPC Byte 2	16#14							
Byte 13	EPC Byte 3	16#F7							
Byte 14	EPC Byte 4	16#33							
Byte 15	EPC Byte 5	16#7C							
Byte 16	EPC Byte 6	16#00							
Byte 17	EPC Byte 7	16#1F							
Byte 18	EPC Byte 8	16#00							
Byte 19	EPC Byte 9	16#00							
Byte 20	EPC Byte 10	16#00							
Byte 21	EPC Byte 11	16#74							
Byte 22	EPC Byte 12	16#83							
Byte 23	Length Data	16#00							
Byte 24	Length Data	16#04							
Byte 25	User Data Byte 1	16#31							
Byte 26	User Data Byte 2	16#32							
Byte 27	User Data Byte 3	16#33							
Byte 28	User Data Byte 4	16#34							
Byte 29	Not relevant	16#00							
Byte 30	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Tag with 12 byte EPC (16#30 14 F7 33 7C 00 1F 00 00 00 74 83) has left the sensing range

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#1D							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length	16#00							
Byte 4	Telegram length	16#14							
Byte 5	Command	16#19							
Byte 6	Status	16#05							
Byte 7	Length EPC/UII	16#00							
Byte 8	Length EPC/UII	16#0E							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 9	PC Word	16#34							
Byte 10	PC Word	16#00							
Byte 11	EPC Byte 1	16#30							
Byte 12	EPC Byte 2	16#14							
Byte 13	EPC Byte 3	16#F7							
Byte 14	EPC Byte 4	16#33							
Byte 15	EPC Byte 5	16#7C							
Byte 16	EPC Byte 6	16#00							
Byte 17	EPC Byte 7	16#1F							
Byte 18	EPC Byte 8	16#00							
Byte 19	EPC Byte 9	16#00							
Byte 20	EPC Byte 10	16#00							
Byte 21	EPC Byte 11	16#74							
Byte 22	EPC Byte 12	16#83							
Byte 23	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

11.2.6 Single Write 4-Byte Blocks

"Single Write 4-Byte Blocks" command telegram for writing 4 bytes of user data (16#01020304) from byte address 8

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame length	16#0E							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length	16#00							
Byte 4	Telegram length	16#0B							
Byte 5	Command	16#40							
Byte 6	ByteAddress	16#00							
Byte 7	ByteAddress	16#08							
Byte 8	Number of bytes	16#00							
Byte 9	Number of bytes	16#04							
Byte 10	Write data byte 1	16#01							
Byte 11	Write data byte 2	16#02							
Byte 12	Write data byte 3	16#03							
Byte 13	Write data byte 4	16#04							
Byte 14	Not relevant	16#00							
Byte 15	Not relevant	16#00							

2023-07

Data telegram from a tag with a 12-byte EPC (16#30 14 F7 33 7C 00 1F 00 00 00 74 83) with confirmation of a successful write operation

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame length	16#1D							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length	16#00							
Byte 4	Telegram length	16#14							
Byte 5	Command	16#40							
Byte 6	Status	16#00							
Byte 7	Length EPC/UII	16#00							
Byte 8	Length EPC/UII	16#0E							
Byte 9	PC word	16#34							
Byte 10	PC word	16#00							
Byte 11	EPC byte 1	16#30							
Byte 12	EPC byte 2	16#14							
Byte 13	EPC byte 3	16#F7							
Byte 14	EPC byte 4	16#33							
Byte 15	EPC byte 5	16#7C							
Byte 16	EPC byte 6	16#00							
Byte 17	EPC byte 7	16#1F							
Byte 18	EPC byte 8	16#00							
Byte 19	EPC byte 9	16#00							
Byte 20	EPC byte 10	16#00							
Byte 21	EPC byte 11	16#74							
Byte 22	EPC byte 12	16#83							
Byte 23	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

11.2.7 Enhanced Write 4-Byte Blocks

"Enhanced Write 4-Byte Blocks" command telegram to write 16 bytes of user data (16#00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF) starting from byte address 0.

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame length	16#1A							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length	16#00							
Byte 4	Telegram length	16#17							
Byte 5	Command	16#1A							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 6	ByteAddress	16#00							
Byte 7	ByteAddress	16#00							
Byte 8	Number of bytes	16#00							
Byte 9	Number of bytes	16#10							
Byte 10	Write data byte 1	16#00							
Byte 11	Write data byte 2	16#11							
Byte 12	Write data byte 3	16#22							
Byte 13	Write data byte 4	16#33							
Byte 14	Write data byte 5	16#44							
Byte 15	Write data byte 6	16#55							
Byte 16	Write data byte 7	16#66							
Byte 17	Write data byte 8	16#77							
Byte 18	Write data byte 9	16#88							
Byte 19	Write data byte 10	16#99							
Byte 20	Write data byte 11	16#AA							
Byte 21	Write data byte 12	16#BB							
Byte 22	Write data byte 13	16#CC							
Byte 23	Write data byte 14	16#DD							
Byte 24	Write data byte 15	16#EE							
Byte 25	Write data byte 16	16#FF							
Byte 26	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

Data telegram from a tag with a 12-byte EPC (16#30 14 F7 33 7C 00 1F 00 00 00 74 83) with confirmation of a successful write operation

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame length	D_S	U_M	U_D	0	16#0			
Byte 1	Frame length	16#17							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length	16#00							
Byte 4	Telegram length	16#14							
Byte 5	Command	16#1A							
Byte 6	Status	16#05							
Byte 7	Length EPC/UII	16#00							
Byte 8	Length EPC/UII	16#0E							
Byte 9	PC word	16#34							
Byte 10	PC word	16#00							
Byte 11	EPC byte 1	16#30							
Byte 12	EPC byte 2	16#14							

2023-07

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 13	EPC byte 3	16#F7							
Byte 14	EPC byte 4	16#33							
Byte 15	EPC byte 5	16#7C							
Byte 16	EPC byte 6	16#00							
Byte 17	EPC byte 7	16#1F							
Byte 18	EPC byte 8	16#00							
Byte 19	EPC byte 9	16#00							
Byte 20	EPC byte 10	16#00							
Byte 21	EPC byte 11	16#74							
Byte 22	EPC byte 12	16#83							
Byte 23	Not relevant	16#00							
...	Not relevant	16#00							
Byte 31	Not relevant	16#00							

11.3 ASCII table

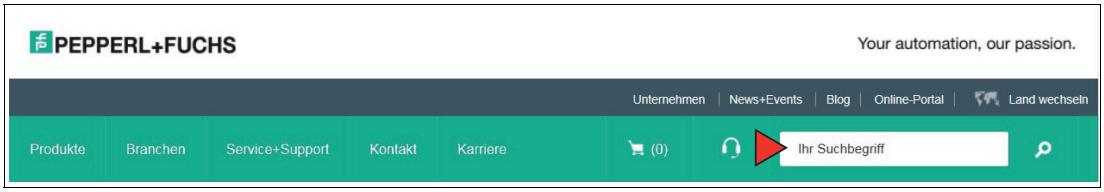
hex	dec	ASCII	hex	dec	ASCII	hex	dec	ASCII	hex	dec	ASCII
00	0	NUL	**20**	**32**	**Space**	40	64	@	**60**	**96**	'
01	1	SOH	**21**	**33**	**!**	41	65	A	**61**	**97**	**a**
02	2	STX	**22**	**34**	**"**	42	66	B	**62**	**98**	**b**
03	3	ETX	**23**	**35**	**#**	43	67	C	**63**	**99**	**c**
04	4	EOT	**24**	**36**	**$**	44	68	D	**64**	**100**	**d**
05	5	ENQ	**25**	**37**	**%**	45	69	E	**65**	**101**	**e**
06	6	ACK	**26**	**38**	**&**	46	70	F	**66**	**102**	**f**
07	7	BEL	**27**	**39**	**'**	47	71	G	**67**	**103**	**g**
08	8	BS	**28**	**40**	**(**	48	72	H	**68**	**104**	**h**
09	9	HT	**29**	**41**	**)**	49	73	I	**69**	**105**	**I**
0A	10	LF	**2A**	**42**	*****	4A	74	J	**6A**	**106**	**j**
0B	11	VT	**2B**	**43**	**+**	4B	75	K	**6B**	**107**	**k**
0C	12	FF	**2C**	**44**	**,**	4C	76	L	**6C**	**108**	**I**
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**

hex	dec	ASCII	hex	dec	ASCII	hex	dec	ASCII	hex	dec	ASCII
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**

11.4 Sensing Range

The device has a typical sensing range of approx. one meter; this range is determined by the tag used and can be adjusted by changing the transmission power. Other influencing factors include the assembly and installation in the specific application, interference from any materials present (in particular metal), and the ambient conditions. The read and write distances for the relevant tag, which are detailed separately, have been established in a test laboratory under ideal conditions. For this reason, the combination of device and tag must be tested for the intended application under real conditions.

Please note the distance tables. The distance tables and additional information regarding your product can be found at http://www.pepperl-fuchs.com. Simply enter the product name or item number in the **Search** box and click the **Search** key.



Select your product from the list of search results. Click on the information you require in the product information list, e.g., **Technical documents**.

A list of all available documents is displayed.

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-8362B