IUT-F190-B40-2V1D-*

UHF read / write station with integrated Ethernet interface

Manual



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 7
1.1 Content of this Document 7
1.2 Target Group, Personnel 8
1.3 Symbols Used 9
1.4 Glossary 10
2 Cybersecurity 11
3 Certificates and approvals 14
3.1 Declaration of Conformity (RE Directive 2014/53/EU) 14
3.2 FCC-Information 14
3.3 IC-Information 15
3.4 IFT Information 15
3.5 NCC-Information 15
3.6 ANATEL-Information 16
3.7 UL Information 16
3.8 Other Country-Specific Approvals 16
4 Product Description 17
4.1 RFID Frequency Bands 17
4.2 UHF general 17
4.2.1 Advantages of UHF 17
4.2.2 Applications for UHF systems 17
4.2.3 Memory Structure of a Tag in Accordance with EPC Gen 2 (ISO/IEC 18000-63) 18
4.2.4 Elektronic Product Code EPC 20
4.2.5 The Impact of Various Materials on the Detection Range 21
4.2.6 Dense Reader Mode (DRM) 22
4.2.7 Frequency Hopping Spread Spectrum 22
4.2.7.1 China 22
4.2.7.2 USA 23
4.2.8 Relevant Standards for UHF 23

4.3 Countries of Use 23
4.3.1 Australia 24
4.3.2 Brazil 24
4.3.3 China 24
4.3.4 European Union 25
4.3.5 India 25
4.3.6 Japan 25
4.3.7 Canada 25
4.3.8 Colombia 25
4.3.9 Malaysia 26
4.3.10 Mexico 26
4.3.11 South Korea 26
4.3.12 Taiwan 26
4.3.13 United States of America 26
4.4 General Functions and Features 27
4.5 Technical Data 29
4.6 Indicators and Operating Elements 29
4.7 Connection 31
4.8 Scope of Delivery 32
4.9 Accessories 32
4.9.1 Read/Write Tags 32
4.9.2 Ethernet Cordset 33
4.9.3 Single-Ended Female Cordset For The Power Supply 34
4.9.4 Mounting Accessory 34
5 Installation 35
5.1 Storage and Transportation 35
5.2 Unpacking 35
5.3 Mounting 35
5.3.1 Room Orientation 37
5.3.2 Minimum Distances 38
5.3.3 Polarization 38
5.4 Connection 39
5.5 EMC Concept 40
6 Commissioning 41
6.1 Protocol 41
6.2 Operating Modes 42
6.3 Definitions 43
6.3.1 Representation 43
6.3.2 Legend 43
7 Operation 45

2026-02

7.1 General 45
7.2 Interference Due to Multipath Propagation 45
7.3 Multiple Tags in Sensing Range 46
7.4 Read Algorithm 46

8 Easy Mode 47
8.1 Command Overview 47
8.2 Basic Structure of the Process Data 48
8.2.1 Output Process Data (PLC -> Device) 48
8.2.2 Input Process Data (Device -> PLC) 49
8.2.3 Flow Diagrams 52
8.2.4 Timing 56

9 ExpertMode 58
9.1 Basic Command Process 58
9.1.1 Telegram Structure 58
9.1.2 Handshake Procedure 60
9.2 Command Overview 61
9.2.1 Read / Write Commands 63
9.2.2 System Commands 95
9.2.3 Filter Commands 97
9.2.4 UHF Configuration Commands 106
9.2.4.1 Basic Command Structure 107
9.2.4.2 Overview of UHF Parameters 112
9.3 Error / Status Messages 144

10 Web Interface 145

11 REST API 146

12 Service and Maintenance 147

13 Troubleshooting 148

14 Appendix 149
14.1 Dimensions 149
14.2 Examples for Expert Mode 149
14.2.1 Single Read Fixcode 149
14.2.2 Enhanced Read Fixcode 150
14.2.3 Single Read Special Fixcode 152
14.2.4 Enhanced Read Special Fixcode 152
14.2.5 Single Program Special Fixcode 154
14.2.6 Single Read 4-Byte Blocks 155
14.2.7 Enhanced Read 4-Byte Blocks 156
14.2.8 Single Write 4-Byte Blocks 158
14.2.9 Enhanced Write 4-Byte Blocks 159

2026-02

14.3 ASCII table 160
14.4 Sensing Range 161
14.5 Antenna Diagram 162

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.

2026-02

1.3 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.

1.4 Glossary

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 Cybersecurity

General

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

Security Context

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

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

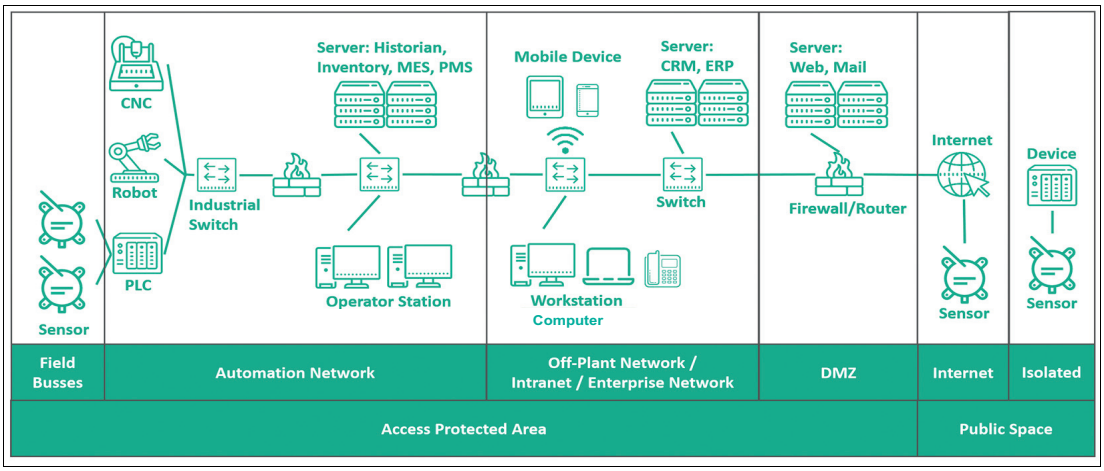


Figure 2.1 Example of the integration of devices and software into a system

Available Protocols and Services

Protocol / Service	Protocol / Port Number	Factory settings	Service status configurable	Notes
DHCP	UDP/68	Enabled	Yes	The service status depends on the setting of the rotary switch.
PROFINET	• UDP/34964 • UDP/49152	Enabled	Yes	The service status depends on the setting of the rotary switch.
Profinet DCP		Enabled	No	The service status depends on the setting of the rotary switch.
EtherNet/IP	• TCP/44818 • UDP/44818 • UDP/2222	Enabled	Yes	The service status depends on the setting of the rotary switch.
OPC UA	TCP/4840	Disabled	Yes	
HTTP (REST API)	TCP/80	Enabled	Yes	
HTTPS (REST API)	TCP/443	Disabled	Yes	**Self-generated certificates** The device can generate self-signed certificates that are valid for the device's IP address. If the IP address changes, a new certificate is generated. **Encryptions** TLSv1.2 is supported. The supported encryption suites for key pairs with ECC are: • TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 The following elliptic curves are supported: • secp256r1 • bp256r1 • Curve25519 For RSA key pairs: • TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (rsa 2048) • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (rsa 2048) **Recommendation:** If possible, use ECC key pairs with CHACHA20_POLY1305-bulk encryption to achieve the highest transfer rate. All modern clients support this. **Browser:** Google Chrome, Microsoft Edge, Apple Safari
SNMP	UDP/161	Enabled	Yes	
NTP Client	UDP/123	Disabled	Yes	

2026-02

Protocol / Service	Protocol / Port Number	Factory settings	Service status configurable	Notes
LLDP		Enabled	No	
ICMP		Enabled	No	

Hardening of the Device

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

Security Functions

- The device has protection against brute force attacks to prevent unauthorised access through password guessing attempts.
- The device implements a session timeout to automatically disconnect connections after a specified period of inactivity.
- The device is assigned a unique default password, which is printed on a sticker on the device.

User Profiles and Rights

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

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

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

Firmware

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

Decommissioning

Reset the device to factory settings.

3 Certificates and approvals

3.1 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**

3.2 FCC-Information

FCC ID: IREIUHF190V1B
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 his 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.

2026-02

3.3 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.

3.4 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.

3.5 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。

3.6 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

3.7 UL Information

Technical Data and Environmental Conditions

This device is for indoor use only.

This device may be operated in altitudes up to 2000 m.

The ambient temperature range is from -20 °C … +70 °C for operation with non-transmission periods, or -20 °C … +50 °C for continuous operation. 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 20 V … 30 V_{DC}. Supply must be PELV (Protective Extra Low Voltage). 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 external circuits intended to be connected to this unit must be galvanically isolated from the mains supply or hazardous live voltage by reinforced or double insulation and fall within the limits specified in clauses 6.3 and 9.4 of UL 61010-1.

3.8 Other Country-Specific Approvals

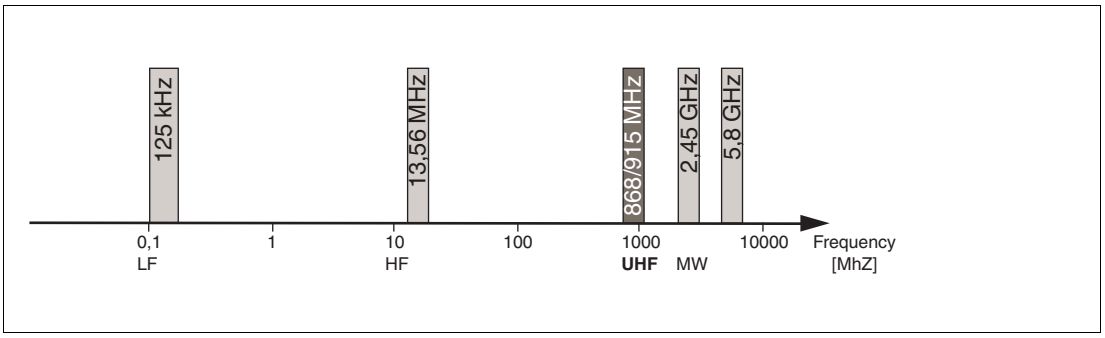
All currently valid approvals can be found in the datasheet for your device at www.pepperl-fuchs.com.

2026-02

4 Product Description

4.1 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

4.2 UHF general

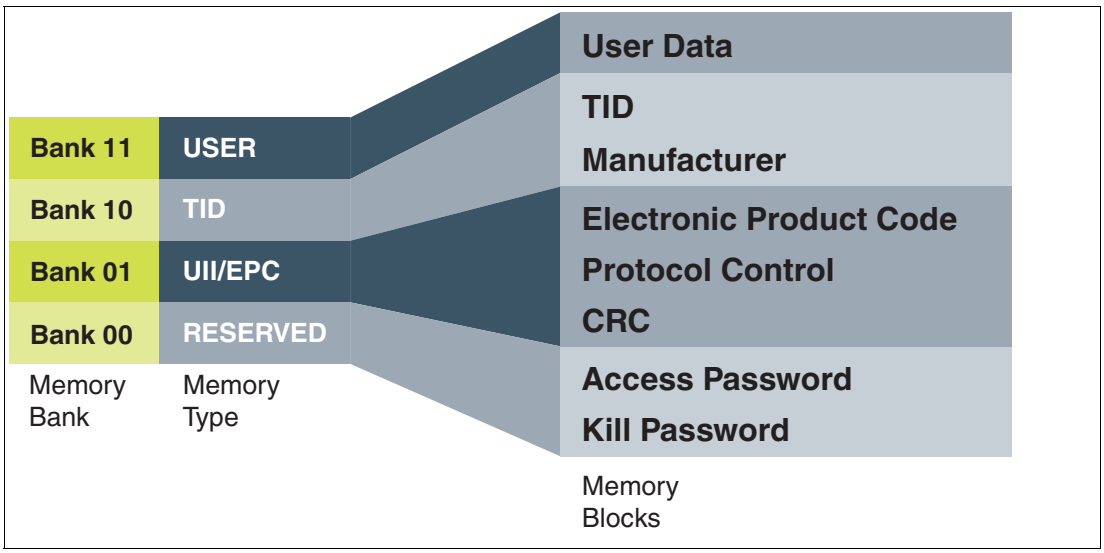
4.2.1 Advantages of UHF

- 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

4.2.2 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.

4.2.3 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 four 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

Bank 00 contains password management information and is accessible only when using expert mode. It comprises 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 (go to page 112). The following commands can be used to access memory bank 00:

- Single / Enhanced Read 4-Byte Blocks (page 57 and 60)
- Single / Enhanced Write 4-Byte Blocks (page 63 and 65)
- Single / Enhanced Read 2-Byte Words (page 67 and 70)
- Single / Enhanced Write 2-Byte Words (page 73 and 75)

Bank 01: EPC/UII

Bank 01 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 out.

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

2026-02

- Length of EPC/UII
- Application Family Identifier (AFI)
- Bit to indicate whether an EPC or ISO standard tag is being used
- Bit to indicate whether data is stored in bank 11 (User Memory)

For a detailed overview of the memory structure, go to page 83.

The device can identify multiple read / write tags in the detection zone. However, only tags with different EPC/UII codes can be located in this zone. 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 command.

When using easy mode, memory bank 01 can only be read. Write access to memory bank 01 is possible only when using expert mode.

The following commands are used to access memory bank 01 in expert mode:

- Single / Enhanced Read Special Fixcode (page 50 and 52)
- Single Program Special Fixcode (page 54)

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

More options to access memory bank 01 are allowed by modifying the MB (Memory Bank) parameter. 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 (page 57 and 60)
- Single / Enhanced Write 4-Byte Blocks (page 63 and 65)
- Single / Enhanced Read 2-Byte Words (page 67 and 70)
- Single / Enhanced Write 2-Byte Words (page 73 and 75)

With these commands, memory bank 01 can be accessed either in full—including CRC—or only one part of it. Multiple consecutive bytes of the EPC/UII can be read out in this way.

Bank 10: TID

Bank 10 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 four 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 eight bytes (64 bits) and the following structure:

- XTID — 16 bits
- Serial Number — 48 bits

For a detailed overview of the TID memory structure, go to page 84. 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 when using easy mode and expert mode. In both cases, the TID is preceded by the EPC/UII. This ensures that the TID is associated to the correct tag.

The following commands are used to access memory bank 10 in expert mode:

- Single / Enhanced Read Fixcode (page 46 and 48)

More options to access memory bank 10 are allowed by modifying the MB (Memory Bank) 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 (page 57 and 60)
- Single / Enhanced Read 2-Byte Words (page 67 and 70)

With these commands, memory bank 10 can be read out either in full or only one part of it.

Bank 11: User Memory

Bank 11 contains memory to which the user has free access. This size of the 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 is stored on the tag. When using chips based on EEPROM technology, the number of write accesses is limited to 100,000. An almost unlimited number of write accesses is possible when using read / write tags with FRAM memory.

The user memory can be accessed when using easy mode and expert mode. User data can be read and written in both versions. When a tag has been accessed successfully, by reading or writing, the response always contains the EPC/UII of the tag. In the case of read access, the user memory is preceded by the EPC/UII. This ensures that the read-in data or the written data is assigned to the correct tag.

The following commands can be used to access memory bank 11 in expert mode:

- Single / Enhanced Read 4-Byte Blocks (page 57 and 60)
- Single / Enhanced Write 4-Byte Blocks (page 63 and 65)
- Single / Enhanced Read 2-Byte Words (page 67 and 70)
- Single / Enhanced Write 2-Byte Words (page 73 and 75)

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

4.2.4 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.

2026-02

	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 4.1

4.2.5 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.

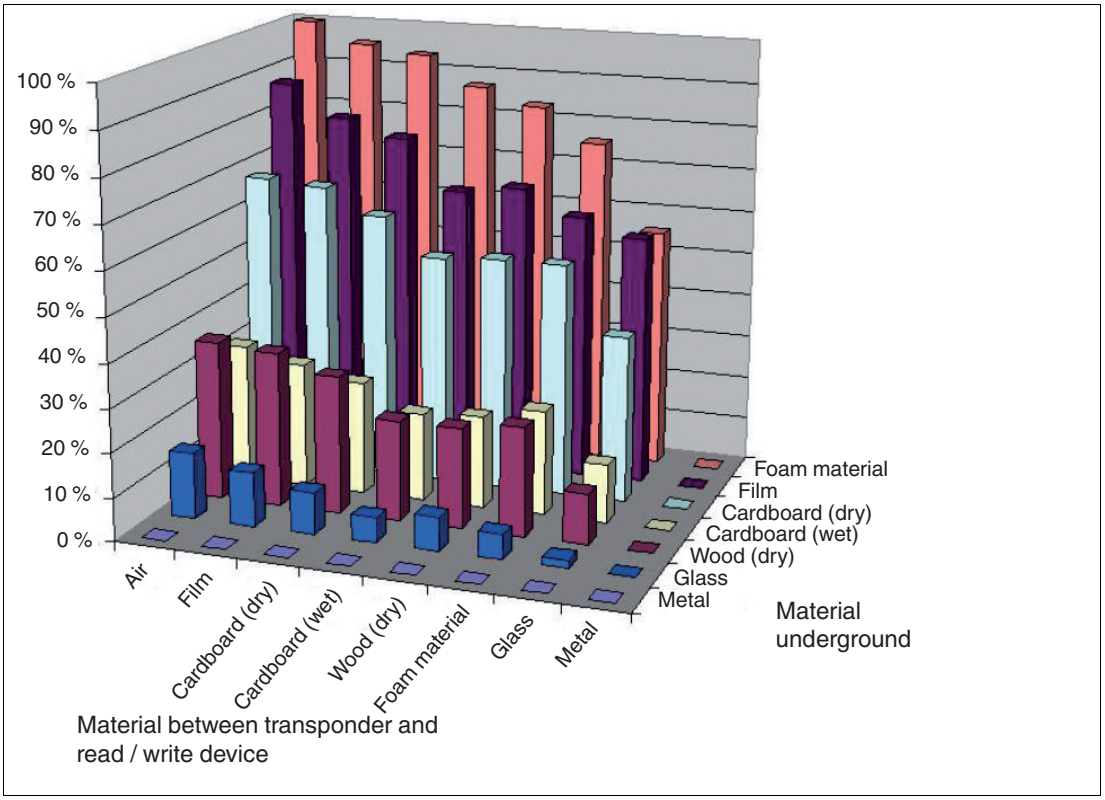


Figure 4.1

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.

4.2.6 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.

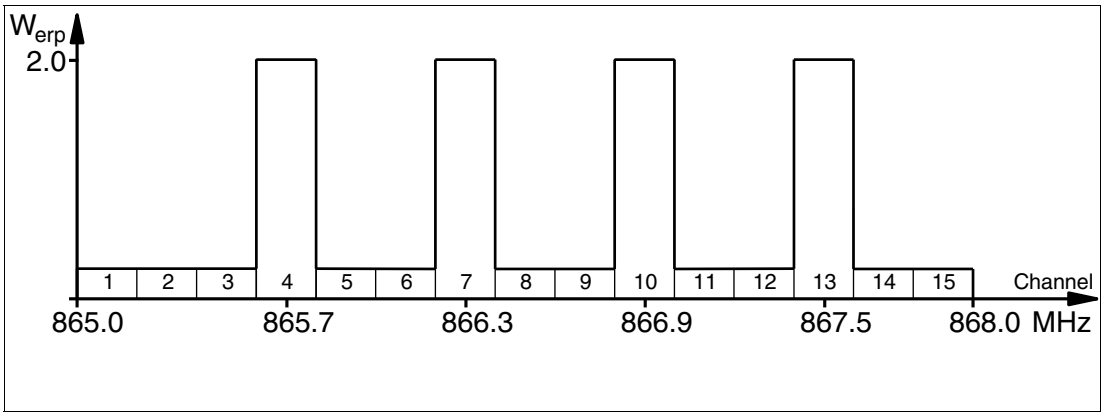


Figure 4.2

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.

4.2.7 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.

4.2.7.1 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.

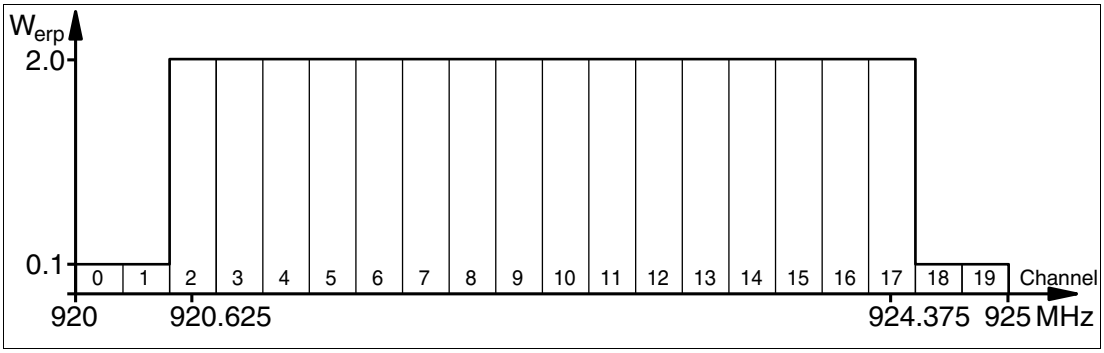


Figure 4.3

4.2.7.2 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.

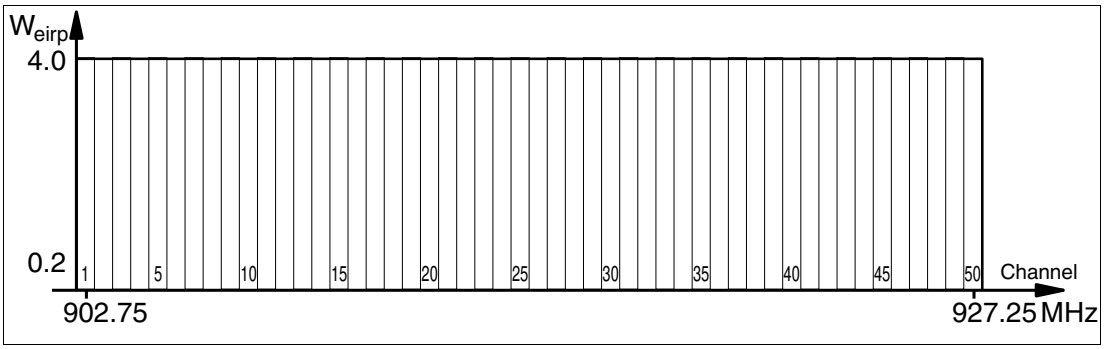


Figure 4.4

4.2.8 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)

4.3 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 4.2.7. 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 4.2.6. With this setting, you can configure one, multiple or all four channels.

4.3.1 Australia

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

- UHF frequency range: 920 MHz ... 926 MHz
- Radiated Power: 3 … 1000 mW_{EIRP}; Default = 125 mW_{EIRP}
- Channel bandwidth: 500 kHz
- Channel spacing: 500 kHz
- Frequency access method: frequency hopping spread spectrum. See chapter 4.2.7.
- 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.

4.3.2 Brazil

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

- UHF frequency range: 915 MHz … 928 MHz
- Radiated Power: 3 … 1250 mW_{EIRP}; Default = 125 mW_{EIRP}
- Channel bandwidth: 250 kHz
- Channel spacing: 250 kHz
- Frequency access method: frequency hopping spread spectrum. See chapter 4.2.7.
- 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.

4.3.3 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 frequency range: 920 … 925 MHz
- Radiated power: 3 … 800 mW_{erp}; Default = 80 mW_{erp}
- Channel bandwidth: 250 kHz
- Channel spacing: 250 kHz
- Frequency access method: Frequency Hopping (China). See chapter 4.2.7.
- 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.

2026-02

4.3.4 European Union

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

- UHF frequency range: 865 … 868 MHz
- Radiated power: 3 … 1000 mW_{erp}; Default = 50 mW_{erp}
- Channel bandwidth: 200 kHz
- Channel spacing: 600 kHz
- Frequency access method: programmable frequency list
- Number of predefined channels: 4
  Programmable 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 4.2.6.

4.3.5 India

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

- UHF frequency range: 865 MHz … 867 MHz
- Radiated Power: 3 … 800 mW_{ERP}; Default = 50 mW_{ERP}
- Channel bandwidth: 200 kHz
- Channel spacing: 200 kHz
- Frequency access method: programmable frequency list
- Number of predefined channels: 10
  Programmable channels: 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 four channels can be parameterized simultaneously and used in sequence.
  Default: dense reader mode on channel 1, 7, 4, 10. See chapter 4.2.6.

4.3.6 Japan

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

- UHF frequency range: 916,7 … 920,5 MHz
- Radiated Power: 3 … 1250 mW_{eirp}; Default = 125 mW_{eirp}
- Channel bandwidth: 200 kHz
- Channel spacing: 1200 kHz
- Frequency access method: programmable frequency list
- Number of predefined channels: 4
  Programmable channels: 5, 11, 17, 23
  Center frequencies: 916,8 MHz, 918,0 MHz, 919,2 MHz, 920,4 MHz
  Up to four channels can be parameterized and used in sequence.
  Default: Dense reader mode on channel 5, 17, 11, 23. See chapter 4.2.6.
- Listen before talk (LBT) is not supported.
- Modulation type: ASK
- License required for using 916,7 … 920,9 MHz up to 4 W_{eirp} (202-JCE144).

4.3.7 Canada

The regulations for the UHF frequency range in Canada meet the requirements for the UHF frequency range in the U.S. See chapter 4.3.13.

4.3.8 Colombia

The regulations for the UHF frequency range in Colombia meet the requirements for the UHF frequency range in Brazil. See chapter 4.3.2.

4.3.9 Malaysia

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

- UHF frequency range: 919 MHz … 923 MHz
- Radiated Power: 3 … 800 mW_{ERP}; Default = 80 mW_{ERP}
- Channel bandwidth: 500 kHz
- Channel spacing: 500 kHz
- Frequency access method: frequency hopping spread spectrum. See chapter 4.2.7.
- 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.

4.3.10 Mexico

The regulations for the UHF frequency range in Mexico meet the requirements for the UHF frequency range in the U.S.. See chapter 4.3.13.

4.3.11 South Korea

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

- UHF frequency range: 917,2 … 920,4 MHz
- Radiated Power: 3 … 1250 mW_{eirp}; Default = 125 mW_{eirp}
- Channel bandwidth: 200 kHz
- Channel spacing: 600 kHz
- Frequency access method: frequency hopping spread spectrum. See chapter 4.2.7.
- 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.

4.3.12 Taiwan

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

- UHF frequency range: 920 … 928 MHz
- Radiated Power: 3 … 500 mW_{erp}; Default = 50 mW_{erp}
- Channel bandwidth: 500 kHz
- Channel spacing: 500 kHz
- Frequency access method: Frequency Hopping. See chapter 4.2.7.
- 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.

4.3.13 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 frequency range: 902 … 928 MHz
- Radiated power: 3…1250 mW_{eirp}; Default = 125 mW_{eirp}
- Channel bandwidth: 500 kHz
- Channel spacing: 500 kHz
- Frequency access method: Frequency Hopping (USA). See chapter 4.2.7.
- 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.

2026-02

4.4 General Functions and Features



Figure 4.5

Functions

The IUT-F190-B40-2V1D-FR* devices were developed for reading and writing passive read / write tags with a UHF operating frequency. Multiple data carriers can be identified within the detection zone if all tags have unique (EPC/UII) codes.

The devices can be operated directly on an Ethernet network. The PROFINET and EtherNet/IP protocols are supported.

The device has an integrated web server. This can be used to adjust device settings and perform diagnostics. Read and write access to read / write tags can be performed via the web server.

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 two meters. The detection range of the device depends on the tags being identified and can therefore be larger or smaller than two meters.

Maximum Frequency Range

The IUT-F190-B40-2V1D-FR1* read / write stations can be used in the frequency range from 865 MHz … 868 MHz. The IUT-F190-B40-2V1D-FR2* read / write stations can be used in the frequency range from 902 MHz … 928 MHz.

Features

The read / write head has the following features:

- Three highly visible LEDs for displaying the UHF functionality on the front of the device
- Fully encapsulated industrial housing with a compact design
- Bulk tag detection
- Integrated Ethernet switch
- Connection to Ethernet networks via M12 connector (M12 x 2)
- Connection to supply voltage (24 V DC) via M12 connector (M12 x 1)
- Digital inputs and outputs (2x in, 1x out)
- Protected against electrostatic discharge
- Integrated web server
- Adjustable UHF parameterization

IntegratedÅ Antenna

The read / write heads have a linear dual polarized antenna. The read / write heads can transmit and receive waves with horizontal and vertical polarization.

Symbols



Warning marking—hot surface!
Do not touch the hot surface of the device. Let the surface of the device cool down before touching the device.

4.5 Technical Data

General data

Working frequency	Country-specific, see chapter 4.3
Radiated power	Country-specific, see chapter 4.3
Operating distance	Typically 2 m

Electrical data

Current consumption	≤ 500 mA
Power consumption	≤ 10 W
Rated operating voltage	20 V DC ... 30 V DC, PELV, limited power source (EN 60950)
Ripple	≤ 10 % at 30 V DC
Surge protection	Category 2

Ambient conditions

Ambient temperature	-20 °C ... 70 °C (-4 °F ... 158 °F) (operation with transmission pauses, adjustable) -20 °C ... 50 °C (-4 °F ... 122 °F) (continuous transmission operation)
Storage temperature	-40 °C ... 85 °C (-40 °F ... 185 °F)
Pollution degree	2
Classification	Environmental condition A (controlled environment)

Mechanical data

Degree of protection	IP67
Connection	Supply voltage: M12 connector Protective earth: M4 grounding screw Ethernet: M12 connector
Enclosure material	PA 6.6
Mounting base material	Aluminum diecast
Mass	770 g

4.6 Indicators and Operating Elements

LED

The device has three LEDs in green, blue, yellow and red 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: power on
- Blue LED: UHF transmission active
- Yellow LED: read / write operation successful
- Red LED: no communication with master (PROFINET)

The device has two LEDs next to the plug for the Ethernet connection. These LEDs indicate communication via the fieldbus interface.

Rotary Switch

There are four rotary switches on the back of the device.



Figure 4.6

- The device is a multi-protocol device. The desired protocol can be set using the rotary switches, see chapter 6.1.
- EtherNet/IP:
  Rotary switches x100, x10, and x1 are used to set only the last byte of the IP address.
- If all rotary switches are set to 9, the device will be reset to factory settings the next time it is started. All settings and password changes will be lost.

4.7 Connection

Caution!

Cable specifications

The maximum permissible temperature of the connection cable must be at least +80 °C. The connection cables must be shielded.

The cable to connect the supply voltage or the inputs and outputs must have a minimum diameter of 22 AWG or 0.34 mm^2. When selecting the connection cable, the voltage drop over the cable length must be taken into account.

The device must be connected to the Ethernet network via a CAT5 or CAT5e with 100 MHz.

IUT-F190-B40-2V1D-FR*

The read / write station has an integrated dual-port switch. When looking at the front of the device, port 1 is on the left and port 2 is on the right. The device is connected to Ethernet via a D-coded, four-pin M12 connector.

The supply voltage, inputs, and outputs are connected via the connector plug in the middle between the two Ethernet ports. An A-coded, five-pin M12 connector is used for this. To prevent unwanted connections of signal lines, unused individual wires of the connection line must be isolated individually.

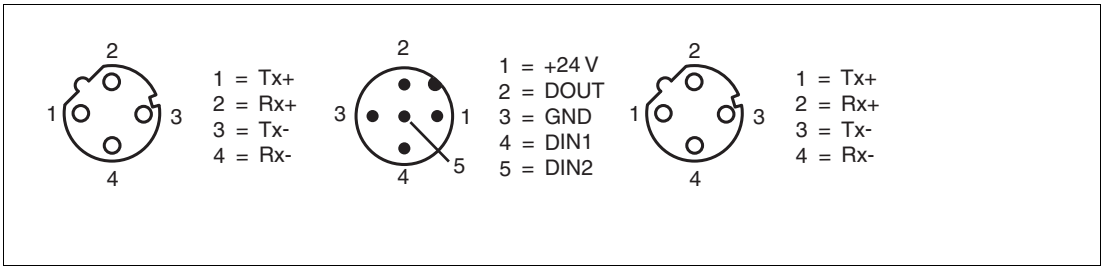


Figure 4.7

Ethernet

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

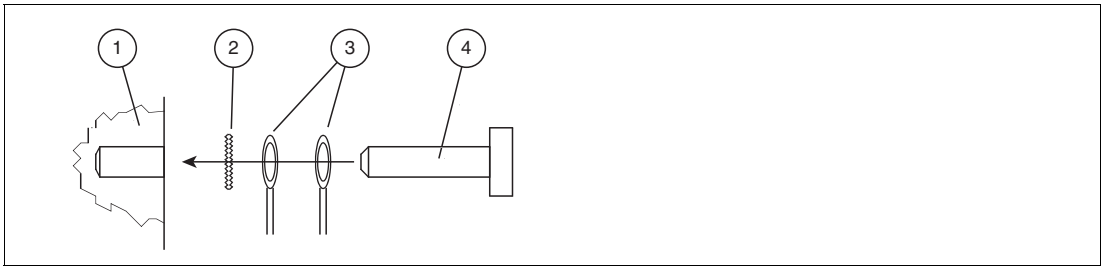
Supply/IO

1 +24 V (20 V DC … 30 V DC), I < 500 mA
2 DOUT, reserved for future applications
3 GND = ground
4 DIN1, reserved for future applications
5 DIN2, reserved for future applications

Ground Connection

The ground connection of the device is located on the right and left side, when looking from the front. The protective earth conductor is screwed to the housing with a crimp connector. To guarantee safe grounding, mount the serrated washer between the crimp connectors and the housing.

The screw connection, which is marked with the symbol ⏚ or ⏚ depending on the device type, is used to connect the protective earth or functional earth.



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

Use a protective earth conductor lead with a cross-section of at least 4 mm^2.

Tighten the retaining bolt to a tightening torque of 1.6 Nm ± 0.4 Nm.

4.8 Scope of Delivery

- Read/write head
- Quick start guide

4.9 Accessories

4.9.1 Read/Write Tags

Type	Designation
EPC Gen 2 (ISO/IEC 18000-63)	IUC76-F157-T17-M-FRx IUC76-F157-T18-M-FRx IUC76-F157-T19-M-FRx IUC76-F203-M-FRx IUC76-F204-M-FRx IUC76-F205-M-FRx IUC76-F208-M-FRx IUC76-F209-M-FRx IUC76-F290-M-FRx IUC76-83L25-GBL 50pcs IUC77-28L90-M-FRx 25pcs IUC77-34-M-FRx 10pcs IUC77-50-FRx 10pcs IUC77-F151-M-GBL 10pcs IUC77-25L100-GBL 1000pcs IUC77-25L110-GBL 1000pcs IUC82-23L50-M-FRx 500pcs

Table 4.2

2026-02

4.9.2 Ethernet Cordset

Compatible cordsets with shield are available to connect the device to an Ethernet network.

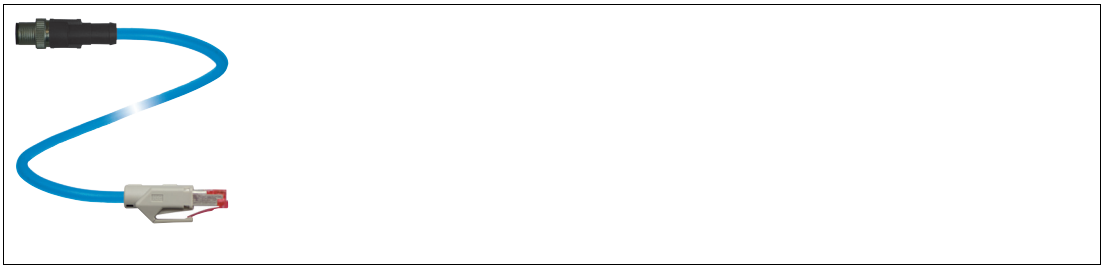


Figure 4.8

Accessories	Designation
M12 cordset (D-coded) to RJ45; straight plug; two meters; shielded	V1SD-G-2M-PUR-ABG-V45-G
M12 cordset (D-coded) to RJ45; straight plug; five meters; shielded	V1SD-G-5M-PUR-ABG-V45-G
M12 cordset (D-coded) to RJ45; straight plug; ten meters; shielded	V1SD-G-10M-PUR-ABG-V45-G
M12 cordset (D-coded) to RJ45; straight plug; 20 meters; shielded	V1SD-G-20M-PUR-ABG-V45-G
M12 cordset (D-coded) to RJ45; angled plug; two meters; shielded	V1SD-W-2M-PUR-ABG-V45-G
M12 cordset (D-coded) to RJ45; angled plug; five meters; shielded	V1SD-W-5M-PUR-ABG-V45-G
M12 cordset (D-coded) to RJ45; angled plug; ten meters; shielded	V1SD-W-10M-PUR-ABG-V45-G
M12 cordset (D-coded) to RJ45; angled plug; 20 meters; shielded	V1SD-W-20M-PUR-ABG-V45-G
M12 cordset (D-coded) to M12 (D-coded); straight plug; two meters; shielded	V1SD-G-2M-PUR-ABG-V1SD-G
M12 cordset (D-coded) to M12 (D-coded); straight plug; five meters; shielded	V1SD-G-5M-PUR-ABG-V1SD-G
M12 cordset (D-coded) to M12 (D-coded); straight plug; ten meters; shielded	V1SD-G-10M-PUR-ABG-V1SD-G
M12 cordset (D-coded) to M12 (D-coded); straight plug; 20 meters; shielded	V1SD-G-20M-PUR-ABG-V1SD-G
M12 single-ended male connector, four-pin (D-coded), field-attachable, straight	V1SD-G-ABG-PG9
M12 single-ended male connector, four-pin (D-coded), field-attachable, angled	V1SD-W-ABG-PG9
Cable ring; Ethernet cable; 100 meters; shielded	CBL-PUR-PN-GN-04x034-100M

Table 4.3

4.9.3 Single-Ended Female Cordset For The Power Supply

Suitable connection cables with flying leads are available in different lengths to connect the device.

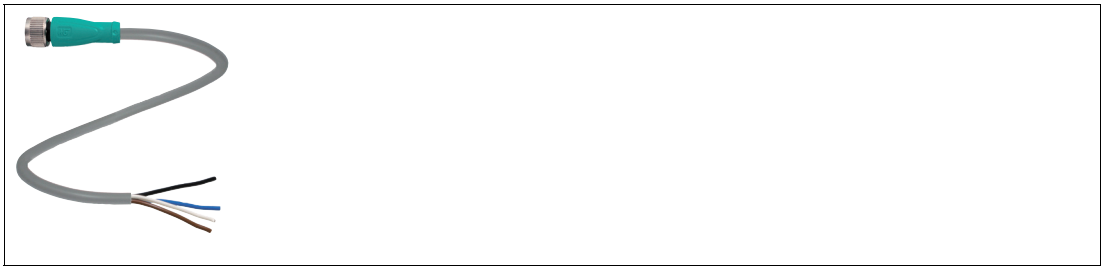


Figure 4.9

Accessories	Designation
Single-ended female cordset, M12 (A-coded); straight plug; two meters; shielded	V15-G-2M-PUR-ABG
Single-ended female cordset, M12 (A-coded); straight plug; five meters; shielded	V15-G-5M-PUR-ABG
Single-ended female cordset, M12 (A-coded); straight plug; ten meters; shielded	V15-G-10M-PUR-ABG
Single-ended female cordset, M12 (A-coded); straight plug; 20 meters; shielded	V15-G-20M-PUR-ABG

Table 4.4

4.9.4 Mounting Accessory

Various mounting brackets are available to mount the device to the wall or on a pipe.

Accessories	Designation
Mounting bracket for wall mounting	IUZ-MH13
Mounting bracket for pipe mounting	IUZ-MH15

5 Installation

5.1 Storage and Transportation

Keep the original packaging. Always store and transport the device in the original packaging.

Store the device in a clean and dry environment. The permitted ambient conditions must be considered, see datasheet.

5.2 Unpacking

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

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

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

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

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

5.3 Mounting

Warning!

Malfunctions with pacemakers

This device does **not** exceed the permissible limits for electromagnetic fields. Maintain a minimum distance of 25 cm between the device and your pacemaker.

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

Warning!

Hot surfaces

Caution: risk of burns when handling the device! Allow the device to cool for at least half an hour after it has been switched off before touching it.

The read/write device is intended for wall mounting or mounting on brackets in indoor spaces. Please mount the read/write device using only the mounting holes provided in the enclosure. 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 device, use 3 screws with a diameter of 4 mm and mounting materials that are suitable for the type of mounting surface. The tightening torque of the screws depends on the type of mounting.

Caution!

Mounting the read/write device

Make sure that the read/write device is firmly 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 the read/write device

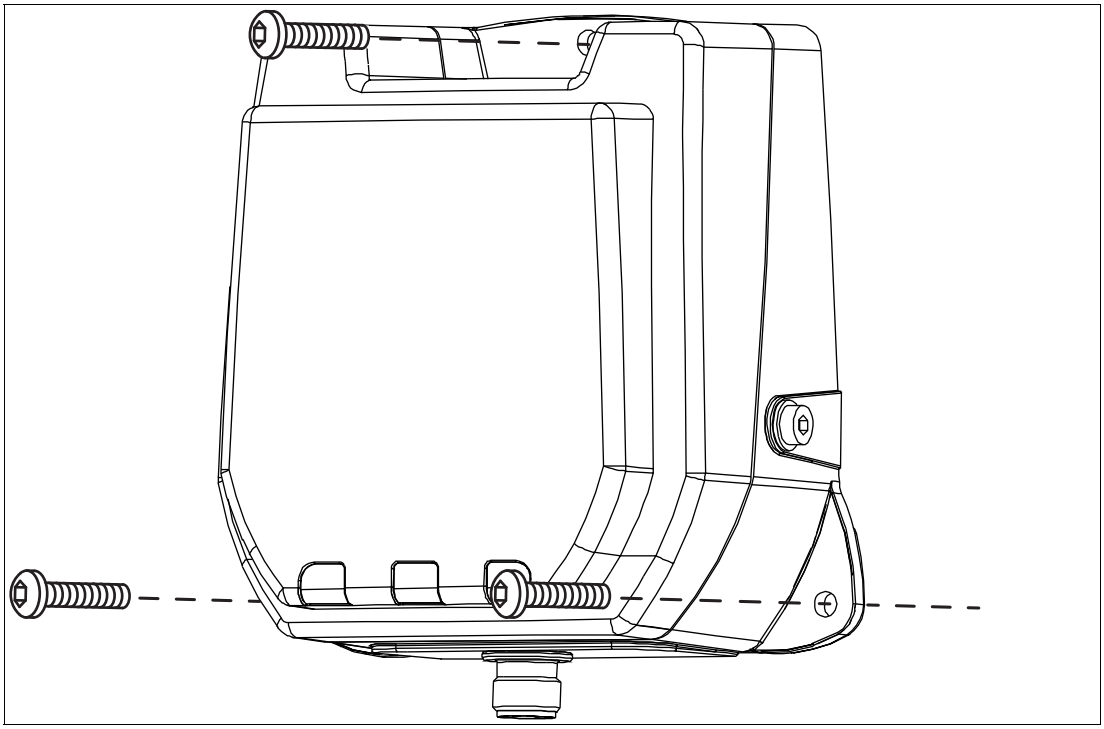
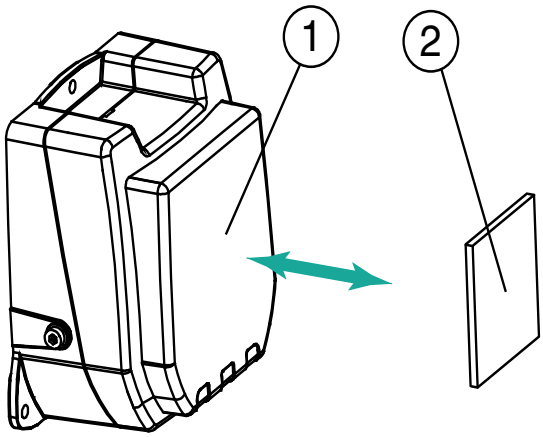


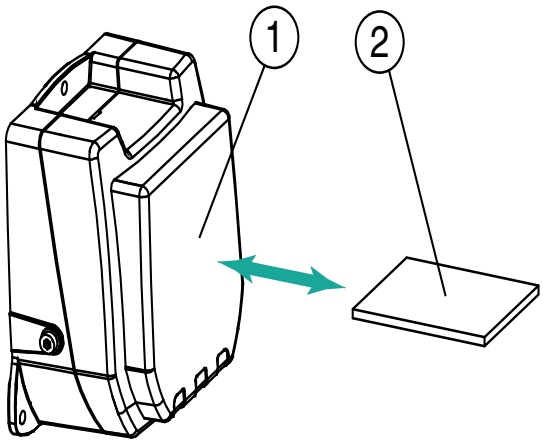
Figure 5.1

5.3.1 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

5.3.2 Minimum Distances

Observe the minimum distances when positioning the device. The lateral distance between the device and metals or liquids should be at least 50 cm. The distance between the device and the ground should be at least 50 cm.

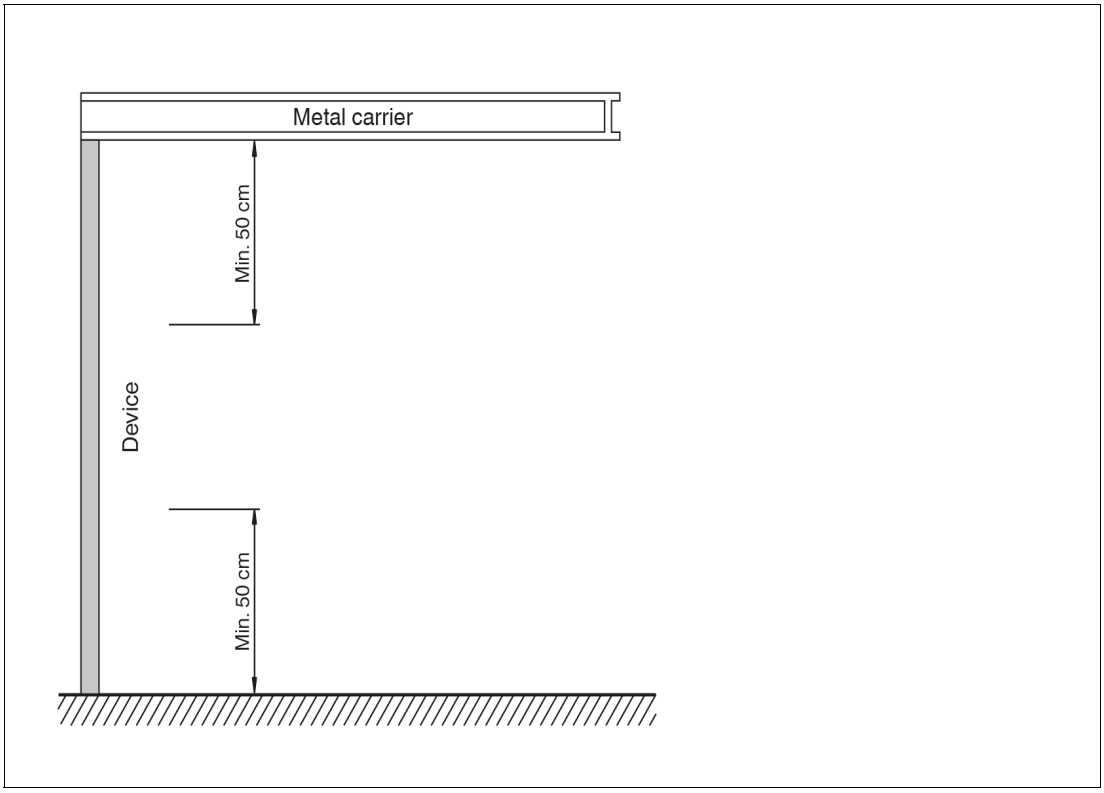


Figure 5.2

During simultaneous operation of several devices, only one device may ever communicate with a tag at any given time. Make sure that there is sufficient distance between the devices. The sensing ranges must not overlap. You can change the transmission power to enlarge or reduce the size of the sensing range. Determine the sensing range of each device at the mounting location.

Note

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

5.3.3 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 electromagnetic field component is spatially constant and therefore dependent on the position of the antenna. Linear polarization can be either vertical or horizontal.

To achieve the maximum detection range for a UHF system, the polarization of the device must match the polarization of the tag—both must have the same spatial orientation. Refer to the relevant datasheet to find the tag polarization.

The integrated device antenna has dual linear polarization. The device operates in combined mode on delivery. In combined mode, both horizontal and vertical polarization are used for each read / write access. Reading of tags with an unknown spatial location is therefore more reliable.

2026-02

If the orientation of the tags is known, you can optimize the access time by setting a fixed polarization. To do this, switch the polarization in the software to horizontally linear polarization or vertically linear polarization. The linear polarization plane refers to the preferred mounting direction with the cable connection mounted vertically downward.

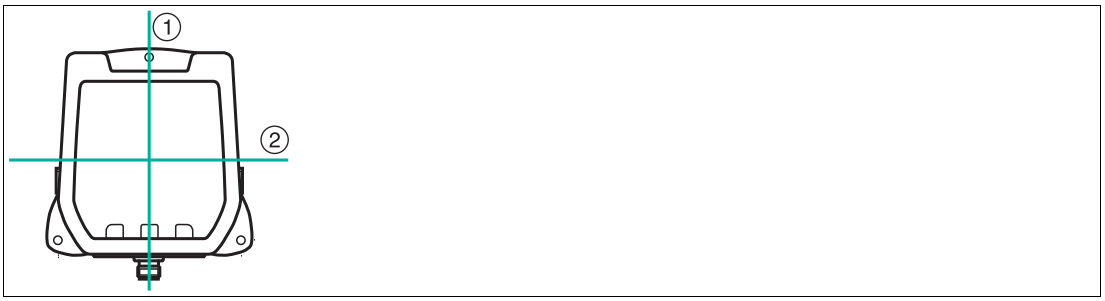


Figure 5.3 1 = Vertical polarization plane
2 = Horizontal polarization plane

5.4 Connection

Connect the device using a suitable Ethernet connection cable (see chapter 4.9.2). Use a suitable cordset (see chapter 4.9.3) for the power supply. Ensure that the shield is connected to the nut so that interference is dissipated via the ground connections to avoid communications due to EMC (see chapter 5.5).

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, the POWER LED on the device lights up green. If the LED does not light up on the device, the power supply is not connected correctly.

5.5 EMC Concept

The device owes its immunity against transmitted noise to its shielding concept. It uses the principle of the Faraday cage. Interference is stopped at the shield and safely grounded using the protective earth connections.

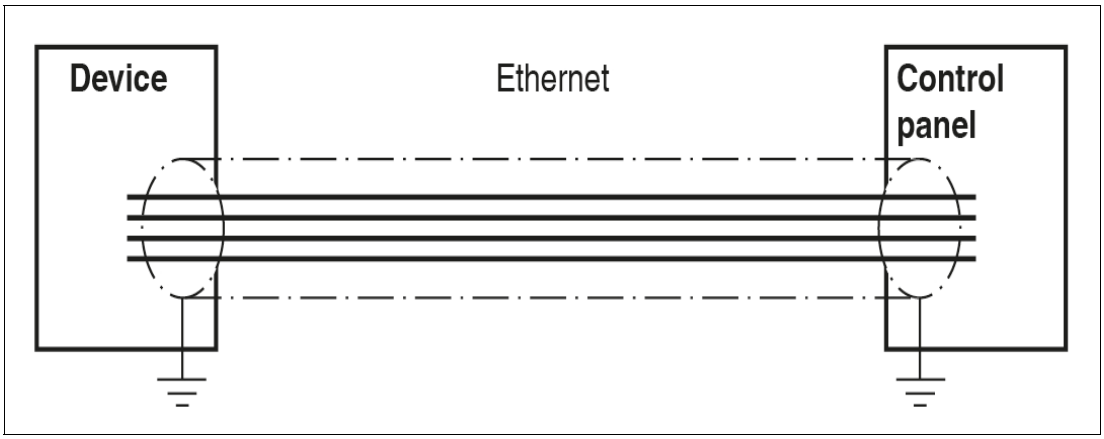


Figure 5.4

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

Note

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

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

Note

The circuit ground is electrically connected to the device housing and to the protective earth. (Connection image → see Figure 4.7 on page 31)

6 Commissioning

6.1 Protocol

Rotary Switch Settings

Mode	x100	x10	x1	Function
0	0	0	0	EtherNet/IP Network settings from non-volatile memory Factory settings: • IP: 192.168.1.250 • Subnet mask: 255.255.255.0 • Gateway: 192.168.1.1 All network settings can be changed via the web interface or by means of other software. Applied network settings are stored in non-volatile memory.
0	0	0	1	EtherNet/IP BOOTP The device always boots in BOOTP mode with this rotary switch setting. (Network settings in non-volatile memory are ignored and overwritten.) All network settings can be changed via the web interface or by means of other software. Applied network settings are stored in non-volatile memory.
0	0	0	2	EtherNet/IP DHCP The device always boots in DHCP mode with this rotary switch setting. (Network settings in non-volatile memory are ignored and overwritten.) All network settings can be changed via the web interface or by means of other software. Applied network settings are stored in non-volatile memory.
0	0 ... 2	0 ... 9	0 ... 9	EtherNet/IP Last octet by rotary switches x100, x10, x1 • IP: 192.168.1.xxx, xxx = rotary switch settings (003 … 254) • Subnet mask: 255.255.255.0 • Gateway: 192.168.1.1 The device always boots with IP: 192.168.1.xxx, xxx = rotary switch settings (0003 ... 0254). (Network settings in non-volatile memory are ignored and overwritten.) All network settings can be changed via the web interface or by means of other software. Applied network settings are stored in non-volatile memory.
1	-	-	-	PROFINET DCP
9	9	9	9	Reset to factory settings

Table 6.1

Note

The position of the rotary switches is read once during power up. Changes to the settings via the rotary switches only take effect after a restart.

Adjusting Protocols



Figure 6.1

Note

If the combination of switch positions is invalid, the invalid switch position is ignored. The last valid setting is used.

1. Set the four rotary switches to the desired position.

6.2 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.

2026-02

6.3 Definitions

6.3.1 Representation

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

ASCII

Example: "A"; "B"; "1"; "2" or alternatively A_{ASCII}; B_{ASCII}; 1_{ASCII}; 2_{ASCII}

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

HEX

Example: 0x41; 0x42; 0x31; 0x32 or alternatively 16#41; 16#42; 16#31; 16#32

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.

6.3.2 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.

Name	Length	Meaning
<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.
<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.

7 Operation

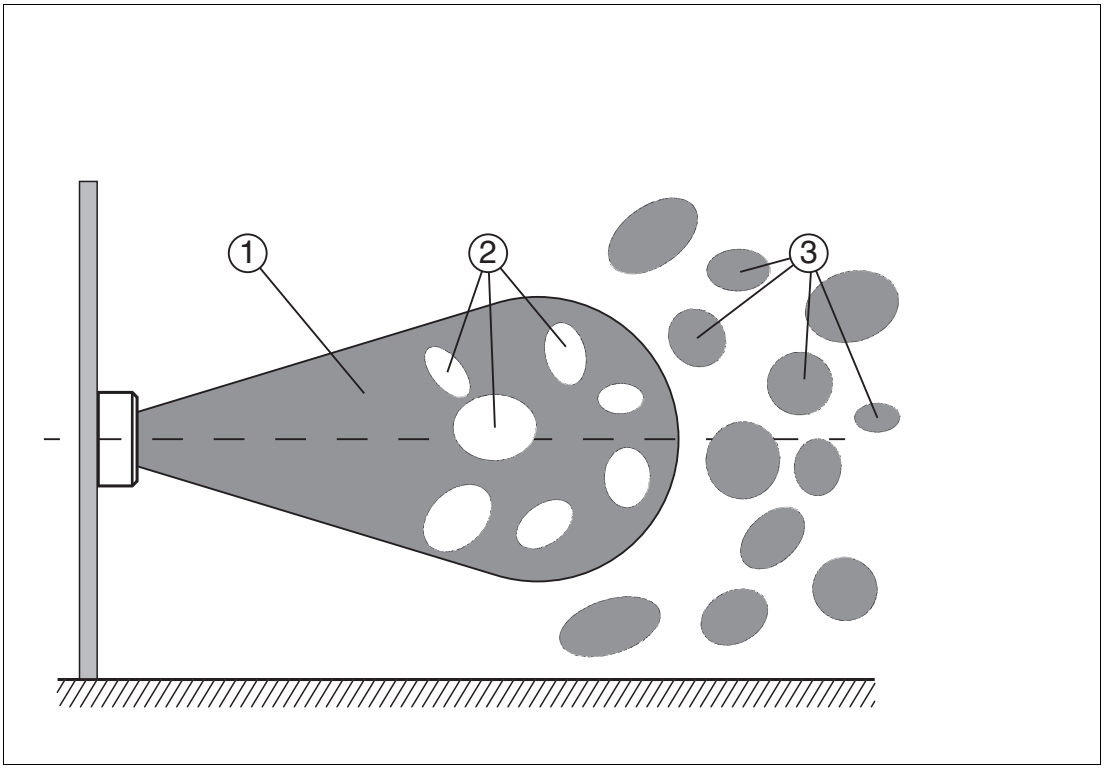
7.1 General

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 8) and Expert Mode (see chapter 9) sections.

7.2 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.

7.3 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 measurement range. The process is controlled by filter masks, which are managed via the commands Set Filter Mask (**FI**) and Activate/Deactivate Filter (**MF**). These commands allow you to detect specific tags in the measurement range. See chapter 9.2.3.

7.4 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.

8 Easy Mode

The device uses the easy mode protocol to transfer data to a higher-level system. If this protocol is used, the device can be started without an additional function block on a control panel. This makes commissioning easier.

A distinction between parameter data and process data is made in easy mode. The parameter data is set in the control panel using the GSDML file (PROFINET) or EDS file (EtherNet/IP). It is transmitted when the device is started. The file contains data for the antenna configuration, and read and write commands. The process data is transmitted in cycles. The process data is divided into input and output data. The length of the process data can be set via the GSDML file or EDS file.

Length of process data	PROFINET	EtherNet/IP
32 bytes	32 Byte In / Out Easy Mode	32 Byte In / Out Easy Mode
64 bytes	64 Byte In / Out Easy Mode	64 Byte In / Out Easy Mode
128 bytes	128 Byte In / Out Easy Mode	128 Byte In / Out Easy Mode
256 bytes	256 Byte In / Out Easy Mode	256 Byte In / Out Easy Mode
512 bytes	512 Byte In / Out Easy Mode	512 Byte In / Out Easy Mode

Table 8.1 Overview of process data variables

8.1 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 two 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 two bytes of PC. • Structure of response data: [length of EPC/UII information; two bytes][PC; two bytes][EPC/UII; variable length]
Read TID	The TID (bank 10) is read out; in the response data, the EPC/UII information is always prefixed to the TID. • Structure of the response data: [length of the EPC/UII information; two bytes][PC; two bytes][EPC/UII; variable length][length of the TID information; two bytes][TID; variable length]
Read user memory	The user memory bank (bank 11) is read out; 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 GSDML file or EDS file. • Structure of the response data: [length of the EPC/UII information; two bytes][PC; two bytes][EPC/UII; variable length][length of the user memory information; two bytes][user memory; length dependent on parameter settings]
Auto-start	Automatic execution of a read command according to the parameter settings in the GSDML file or EDS file. One of the read commands described above can be started automatically without activation via the process image. Auto-start is disabled by factory default.

Command	Description
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 GSDML file or EDS file. The data to be written begins in the output data field from byte 4. • Structure of the data: [control byte; one byte][not used; three bytes][write data; length depends on parameter settings]

Table 8.2 Overview of easy mode commands

8.2 Basic Structure of the Process Data

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

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

8.2.1 Output Process Data (PLC -> Device)

Byte	Content							
0	0	0	0	0	0	0	Write	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							
511	User data for write job or 16#00 for read job							

Table 8.3

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 according to the configuration set in the GSDML file or EDS 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 according to the configuration set in the GSDML file or EDS 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. Adjust the value 0x00 accordingly.

Byte 4 … 511 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 on the read / write tag is stored in this area.

Example: writing user memory in TIA Portal

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 data field

Byte 0: 16#02 (0000 0010bin); the "Start writing" bit is set to 1; a write job is executed.

Byte 1/2/3: 16#00; not used

Byte 4 … byte 8: 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 data field

Byte 0: 16#06 (0000 0110bin); the "Job active" bit is set to 1, which indicates that a write job is activated. The "Write successful" bit is set to 1. A read / write tag is therefore located in the detection zone and the data has been written successfully.

Byte 1: 16#10 (16dec); 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.

Bytes 2 and 3: 16#0000; not used

Bytes 4 and 5: 16#000E (14dec); length specification for the EPC/UII information. The EPC/UII information consists of the PC word with a length of two 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.

Bytes 6 and 7: 16#3400; PC word (Protocol Control)

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

8.2.2 Input Process Data (Device -> PLC)

Byte	Content							
0	0	0	0	0	Error	Job active	Write successful	Read successful
1	Length information							
2	0x00							
3	0x00							
4	Data for read / write job / error code							
5	Data for read / write job / error information							
...	Data for read / write job / error information							
511	Data for read / write job / error information							

Table 8.4

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 detection zone and the data has been read successfully. The bit remains set while the read / write tag is within the detection zone. 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 the transmission 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 detection zone and the data has been written successfully. The bit remains set while the read / write tag is within the detection zone. 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 .
Byte 1	This byte contains the number of read bytes. If a read / write tag enters the detection zone 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 2/3	These bytes are not used in easy mode. Set to 0x00.
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.
Byte 6/7	If a read / write tag is accessed successfully, the PC word (Protocol Control) is in this position.
Byte 8 … byte 511	Starting at byte 8, the EPC/UII of the read / write tag to which the information was read or written is transmitted. 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 (two bytes). This comes from the additional read-in information (TID or user memory). 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.

Example 1: reading EPC

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:	16#05 (0000 0101bin); 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 detection zone and the data has been read in.
Byte 1:	16#10 (16dec); 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.
Bytes 2 and 3:	16#0000; not used
Bytes 4 and 5:	16#000E (14dec); length specification for the EPC/UII information. The EPC/UII information consists of the PC word with a length of two 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.
Bytes 6 and 7:	16#3400; PC word (Protocol Control)
Byte 8 … byte 19:	EPC/UII of the read / write tag to which the data was written.

Example 2: reading EPC + TID

Adresse	Anzei...	Beobachtungswert
%IB0	Hex	16#05
%IB1	Hex	16#1E
%IB2	Hex	16#00
%IB3	Hex	16#00
%IB4	Hex	16#00
%IB5	Hex	16#0E
%IB6	Hex	16#34
%IB7	Hex	16#00
%IB8	Hex	16#30
%IB9	Hex	16#14
%IB10	Hex	16#F7
%IB11	Hex	16#33
%IB12	Hex	16#7C
%IB13	Hex	16#00
%IB14	Hex	16#1F
%IB15	Hex	16#00
%IB16	Hex	16#00
%IB17	Hex	16#00
%IB18	Hex	16#74
%IB19	Hex	16#82
%IB20	Hex	16#00
%IB21	Hex	16#0C
%IB22	Hex	16#E2
%IB23	Hex	16#80
%IB24	Hex	16#11
%IB25	Hex	16#05
%IB26	Hex	16#20
%IB27	Hex	16#00
%IB28	Hex	16#5A
%IB29	Hex	16#9E
%IB30	Hex	16#F1
%IB31	Hex	16#A2
%IB32	Hex	16#00

Byte 0:	16#05 (0000 0101bin); 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 detection zone and the data has been read in.
Byte 1:	16#1E (30dec); the byte indicates the length of the transmitted information. This telegram is used to transmit information with a length of 30 bytes. The information starts at byte 4.
Bytes 2 and 3:	16#0000; not used
Bytes 4 and 5:	16#000E (14dec); length specification for the EPC/UII information. The EPC/UII information consists of the PC word with a length of two 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.
Bytes 6 and 7:	16#3400; PC word (Protocol Control)
Byte 8 … byte 19:	EPC/UII of the identified read / write tag
Bytes 20 and 21:	16#000C (12dec); length specification for the read-in TID
Byte 22 … byte 33:	TID of the read / write tag

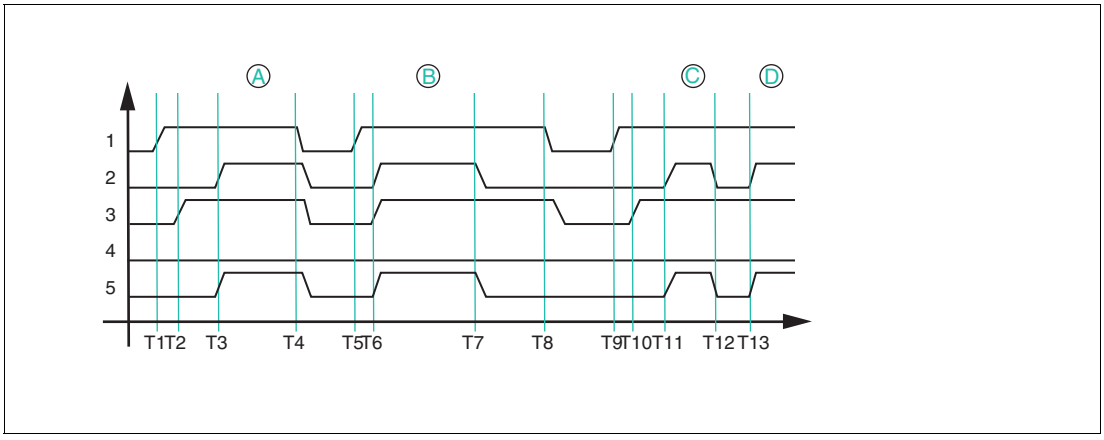
Example 3: reading EPC + user memory

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:	16#05 (0000 0101bin); 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 detection zone and the data has been read in.
Byte 1:	16#1A (26dec); 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.
Bytes 2 and 3:	16#0000; not used
Bytes 4 and 5:	16#000E (14dec); length specification for the EPC/UII information. The EPC/UII information consists of the PC word with a length of two 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.
Bytes 6 and 7:	16#3400; PC word (Protocol Control)
Byte 8 … byte 19:	EPC/UII of the identified read / write tag
Bytes 20 and 21:	16#0008 (8dec); length specification for the read-in user memory
Byte 22 … byte 33:	User memory section read in

8.2.3 Flow Diagrams

Read Job without Auto-Start Function



1 Read
2 Read successful
3 Job active
4 Error
5 Data (input)

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

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

2026-02

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

T3: Read / write tag A enters the detection zone; "Read successful" is 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 "Read" bit to FALSE while the read / write tag is located in the detection zone; the "Job active" bit and the "Read successful" bit are set to FALSE and the process data is filled with 0x00

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

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

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

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

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

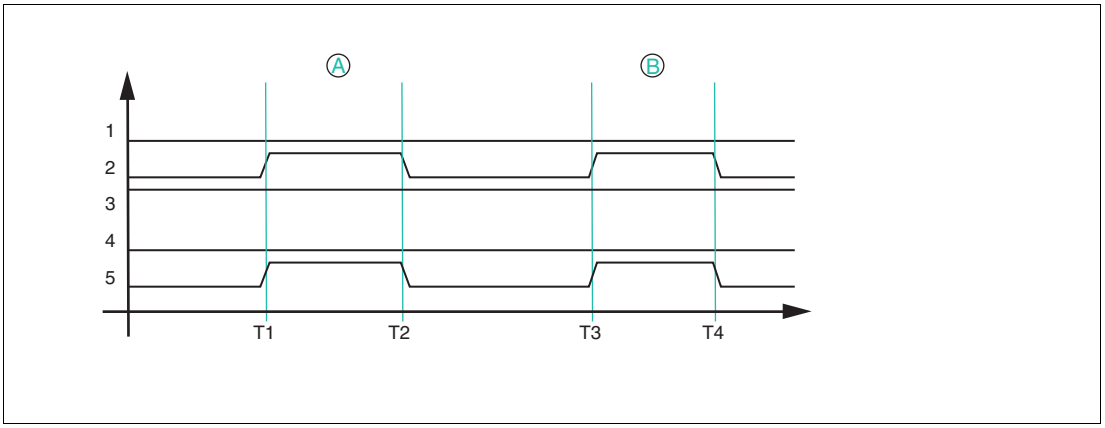
T10: Read job is executed ("Job active" = TRUE)

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

T12: Read / write tag C leaves the detection zone ("Read successful" = FALSE)

T13: Read / write tag D enters the sensing range

Read Job with Auto-Start Function



1 Read

2 Read successful

3 Job active

4 Error

5 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.

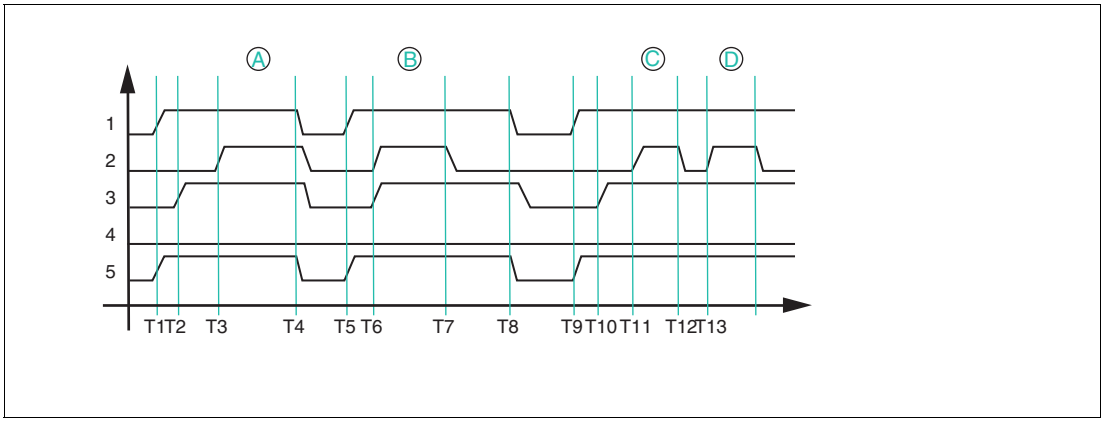
T1: Read / write tag A enters the detection zone; "Read successful" is set to TRUE and the read data is located in the input field of the process data

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

T3: Read / write tag B enters the detection zone; behavior same as T1

T4: Read / write tag B leaves the detection zone; behavior same as T2

Write Job



1 Write

2 Write successful

3 Job active

4 Error

5 Data (output)

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

T1: Start the write job by setting the "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 detection zone ("Write successful" = FALSE)

T3: Read / write tag A enters the detection zone and the data is written successfully ("Write successful" = TRUE)

T4: The write job is canceled by resetting the "Write" bit to FALSE; the "Job active" bit is 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 "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 detection zone

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

T7: Read / write tag B leaves the detection zone ("Write successful" = FALSE); write job remains active ("Job active" = TRUE)

T8: The write job is canceled by resetting the "Write" bit to FALSE; the "Job active" bit is 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 "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 detection zone ("Write successful" = FALSE)

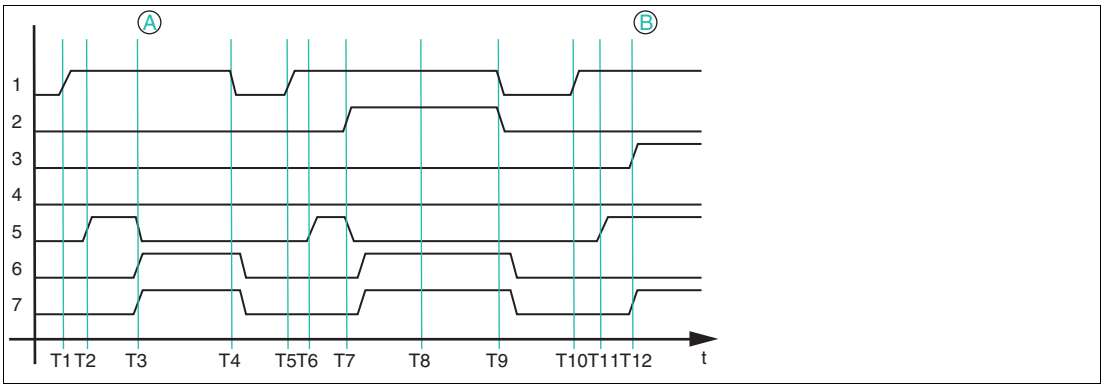
T11: Read / write tag C enters the detection zone and the data is successfully written ("Write successful" = TRUE); write job is still active ("Job active" = TRUE)

T12: Read / write tag C leaves the detection zone ("Write successful" = FALSE); write job is still active ("Job active" = TRUE)

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

2026-02

Fault State



1 Read
2 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: The read job is started at eight bytes of user memory by setting the "Read" bit to TRUE

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

T3: An IUC78 read / write tag enters the detection zone 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 field of the process data. This indicates that the read job set via the GSDML file or EDS file is not suitable for the properties of the IUC78 read / write tag. This is due to the number of bytes to be read. A maximum of four bytes can be read in to access the IUC78. In this example, the value is set to eight. To rectify the error, correct the value in the GSDML file or EDS file.

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

T5: A new read job is started by setting the "Read" bit to TRUE; at the same time, the "Error" bit is reset to FALSE and the error message in the input data is deleted

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

T7: A write job is started by setting the "Write" bit to TRUE. The "Error" bit is set and "Job active" is reset to FALSE. An error message with the error code 0x04 and the text "read AND write" 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 "Read" and "Write" bits are set

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

T10: The read job is started by setting the "Read" bit to TRUE; at the same time, the "Error" bit is reset to FALSE and the error message in the input data is deleted

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

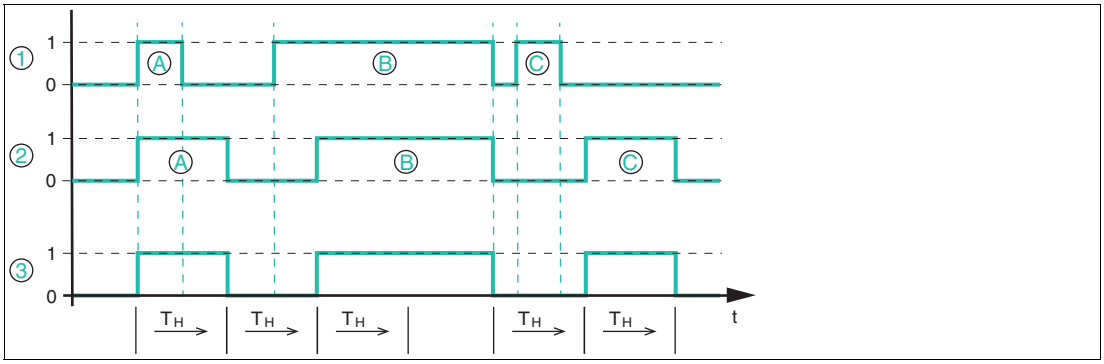
T12: A read / write tag enters the detection zone and is read successfully ("Read successful" = TRUE)

8.2.4 Timing

In easy mode, the device does not require a complex handshake procedure to transfer data. Instead, the telegrams are set in the input field for the process data (device -> PLC) and remain there for a defined hold time. Within this hold time, no changes can be made to the input data field on the device.

The minimum hold time is 50 ms.

If a new telegram is generated by the device within the hold time (e.g., a new tag is read or a tag leaves the sensing range), this telegram is sent to the input field of the process data only after the hold time of 50 ms has expired. If no new telegram occurs within the hold time, the data in the input field for the process data remains unchanged.



1 Read / write tag is in field

2 Data

3 Read successful

The image above shows the principle of the chronological sequence of 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 50 ms.

The device is activated via the auto-start function or the "Read" bit. As a result, a read job is constantly executed on the device.

At the beginning, tag A enters the sensing range of the device and the "Read successful" bit in the input field of the process data changes the signal state from 0 to 1. The read / write tag stays in the sensing range for fewer than 50 ms and leaves it shortly after entering. The input field of the process data containing information about read / write tag A is retained for the time "T_H" (= 50 ms). The input field of the process data is only updated again once this time span has expired. It now contains the information "Read successful" = FALSE (no tag) and indicates that the tag has left the sensing range. This telegram also remains in the input field of the 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 field of the process data is only updated once the hold time of 50 ms has expired. The "Read successful" bit changes the signal state from 0 to 1. The read data is simultaneously set in the input field of the process data. Read / write tag B remains within the device sensing range for more than 50 ms (> "T_H"). During this time span, the input field of the process data remains unchanged and the "Read successful" bit continues to have signal state 1.

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

2026-02

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 process image is modified accordingly. The signal state of "Read successful" changes from 0 to 1.

9 ExpertMode

9.1 Basic Command Process

The in/out length is set in the GSDML file or EDS file in the same way as in easy mode (see chapter 8). 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 512 bytes. A handshake procedure controls the data transfer.

9.1.1 Telegram Structure

Legend

D_S	Delete bit; deletes the internal memory of the device
U_M	Update master bit
U_S	Update slave bit
Frame Length	Number of valid bytes within the telegram
Fragmentation Counter	Number of fragments still to be transmitted
Telegram Length	Length of telegram including all fragments
Command	Command code
Status	Command execution status
Parameter/Data	Parameter or data
Byte N	Number of bytes within the telegram; max. 511

OUTPUT telegram

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

Table 9.1

The value of <Frame Length> depends on how many data values <Data Byte> 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.

2026-02

The <Fragmentation Counter> has a value of 0x00 because the command can be transmitted from the control panel via a fragment.

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

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

- **Read / write commands:** access to one or more read / write tags in the detection zone
- **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 are set to the value 16#00.

INPUT telegram

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
0	ControlByte / frame length	D_S	U_M	U_S	0	<FrameLength>			
1	Frame length	<Frame Length>							
2	Fragmentation counter	<Fragmentation Counter>							
3	Telegram length (high byte)	<Telegram Length (High Byte)>							
4	Telegram length (low byte)	<Telegram Length (Low Byte)>							
5	Command	<Command>							
6	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	...							
N	Parameter/data	0x00							

Table 9.2

The value of <Frame Length> depends on how many data values <Data Byte> 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 control panel via a fragment, the <Fragmentation Counter> has the value 0x00.

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 <Command>, and <Telegram Length> has the value 16#04.

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

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

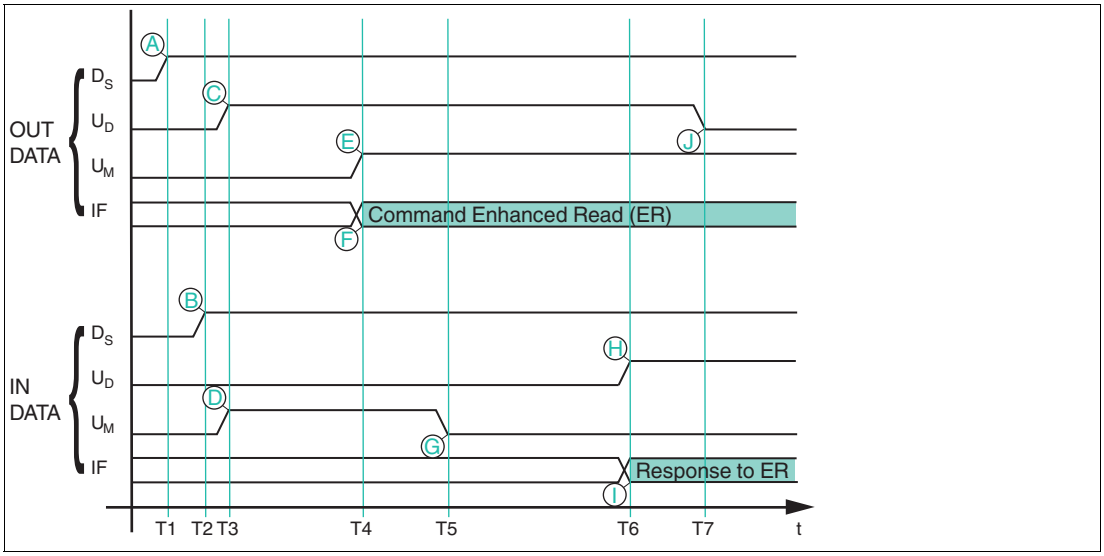
The data being transmitted from the device caused by the command is returned in the <Data Byte> bytes. This could be data read from a read / write tag or parameter values for the UHF settings.

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

9.1.2 Handshake Procedure

Data flow between a PLC and the control interface must be synchronized to ensure continuous data transfer with no losses. The input and output data is transmitted in cycles. Controlling the data flow with software is referred to as "handshaking." The necessary control bits are contained in the control byte (see table "INPUT telegram" on page 59 and see table "OUTPUT telegram" on page 58).

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.

Sample Implementation In The Control Panel

Slave delete bit D_S:

Once the device is ready for operation (PROFINET or EtherNet/IP communication = OK), this instruction must be executed once. The internal telegram memory is deleted. This memory should be deleted if an internal device fault has occurred.

```
DS_OUTPUT := NOT DS_INPUT
```

Slave update bit U_S:

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

To stop blocking transmission of the read data, the inverted state of the U_S bit must be transmitted from the input data field to the U_S bit of the output data field in each cycle.

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

Master update bit U_M:

The device (slave) 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 data fields has an inverted signal state.

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

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

```
OUTPUT[1..x] := new telegram
IF (UM_OUTPUT <> UM_INPUT) then (* Check whether the slave can receive new data *)
UM_OUTPUT := UM_INPUT (* Slave is ready to receive, accept update bit *)
End_IF
```

9.2 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 63 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 65 Permanent read access to the EPC/UII (bank 01) + TID (bank 10)
SS	16#0A	Single Read Special Fixcode see "Single Read Special Fixcode (SS)" on page 68 One-time read access to the EPC/UII (bank 01)
ES	16#71	Enhanced Read Special Fixcode see "Enhanced Read Special Fixcode (ES)" on page 70 Permanent read access to the EPC/UII (bank 01)
SP	16#0D	Single Program Special Fixcode see "Single Program Special Fixcode (SP)" on page 72 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 75 One-time read access to EPC/UII (bank 01) + four byte user memory data blocks (bank 11)
ER	16#19	Enhanced Read 4-Byte Blocks see "Enhanced Read 4-Byte Blocks (ER)" on page 77 Permanent read access to EPC/UII (bank 01) + four byte user memory data blocks (bank 11)
SW	16#40	Single Write 4-Byte Blocks see "Single Write 4-Byte Blocks (SW)" on page 80 One-time write access to four byte user memory data blocks (bank 11)
EW	16#1A	Enhanced Write 4-Byte Blocks see "Enhanced Write 4-Byte Blocks (EW)" on page 82 Permanent write access to four byte user memory data blocks (bank 11)
#SR	16#49	Single Read 2-Byte Words see "Single Read 2-Byte Words (#SR)" on page 85 One-time read access to EPC/UII (bank 01) + two byte user memory data words (bank 11)
#ER	16#4B	Enhanced Read 2-Byte Words see "Enhanced Read 2-Byte Words (#ER)" on page 87 Permanent read access to EPC/UII (bank 01) + two byte user memory data words (bank 11)
#SW	16#4A	Single Write 2-Byte Words see "Single Write 2-Byte Words (#SW)" on page 90 One-time write access to two byte user memory data words (bank 11)
#EW	16#4C	Enhanced Write 2-Byte Words see "Enhanced Write 2-Byte Words (#EW)" on page 93 Permanent write access to two byte user memory data words (bank 11)

System Commands

Abbreviation	Command code	Command description
QU	16#02	Quit see "Quit (QU)" on page 95 Cancels an active enhanced command
TM	16#9C	Set Trigger Mode Sets the trigger conditions for the inputs
OM	16#B7	Set Output Mode Sets the output behavior
VE	16#03	Version see "Version (VE)" on page 96 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 97 Sets a filter
MF	16#CB	Activate Filter Mask see "Activate Filter Mask (MF)" on page 103 Activates a filter Reads out the firmware version

Configuration Commands

Abbreviation	Command description
RP	Read Parameter see "Read Parameter (RP)" on page 107 Reads out the device parameters
WP	Write Parameter see "Write Parameter (WP)" on page 109 Sets device parameters

9.2.1 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).

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 "end" telegram has the status value 16#0F and contains the number of tags identified during execution of the command.

Command:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	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 511	Not relevant	16#00							

Table 9.3

The <FrameLength> has the value 16#06 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 telegram in bytes (<TelegramLength>) is 16#03.

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

Data telegram response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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 511	Not relevant	16#00							

Table 9.4

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

This is followed by a length specification via the EPC/UII information. It is always two 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 two 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.

Command end response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#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 511	Not relevant	16#00							

Table 9.5

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 four 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 read 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_S	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 511	Not relevant	16#00							

Table 9.6

The <FrameLength> has the value 16#06 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 telegram in bytes (<TelegramLength>) is 16#03.

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

Data telegram response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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 511	Not relevant	16#00							

Table 9.7

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 fragmentation counter <FragmentationCounter> has a starting value equal to the number of fragments to transmit minus one and is decremented each time a new fragment is transmitted. The last fragment transmitted will have a value of 16#00. 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 via the EPC/UII information. It is always two 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 two 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.

Tag has left the detection zone:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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 511	Not relevant	16#00							

Table 9.8

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 via the EPC/UII information. It is always two 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 Special Fixcode (SS)

The "Single Read Special Fixcode" 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 "end" 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_S	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#0A							
Byte 6	Fix length	16#00							
Byte 7	Not relevant	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.9

The <FrameLength> has the value 16#07 and is the length of the fragment in bytes. The fragment ends after <FixLength>. 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#04. The command code <Command> for the "Single Read Special Fixcode" command is 16#0A.

Data telegram response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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#0A							
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>							

2026-02

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.10

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> 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 of the tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#0A and is reflected within the response telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification via the EPC/UII information. It is always two 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.

Command end response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#0A							
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 511	Not relevant	16#00							

Table 9.11

The <FrameLength> constantly has a value of 16#0F in the response for the end of the "Single Read Special 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#0A 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 four 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 Special Fixcode (ES)

The "Enhanced Read Special Fixcode" command has the command code 16#71 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_S	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#71							
Byte 6	Fix length	16#00							
Byte 7	Not relevant	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.12

The <FrameLength> has the value 16#07 and is the length of the fragment in bytes. The fragment ends after <FixLength>. 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#04. The command code <Command> for the "Enhanced Read Special Fixcode" command is 16#71.

Data telegram response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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#71							
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)>							

2026-02

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 511	Not relevant	16#00							

Table 9.13

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> but will not be larger than the maximum allowed length of the fragment. The fragmentation counter <FragmentationCounter> has a starting value equal to the number of fragments to transmit minus one and is decremented each time a new fragment is transmitted. The last fragment transmitted will have a value of 16#00. 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#71 and is reflected within the data telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification via the EPC/UII information. It is always two 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.

Tag has left the detection zone:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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#71							
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 511	Not relevant	16#00							

Table 9.14

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> 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 of the tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#71 and is reflected within the telegram. The <Status> parameter has the value 16#05.

This is followed by a length specification via the EPC/UII information. It is always two 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 Program Special Fixcode (SP)

The "Single Program Special Fixcode" command has the command code 16#0D and performs a one-time write operation to the EPC/UII (bank 01) of a tag within the detection zone. 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. 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 Program Special Fixcode" command can be used to write an EPC/UII with a maximum length of 12 bytes to a read / write tag. 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

Table 9.15

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

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

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

Table 9.16

2026-02

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.

Command:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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#0D							
Byte 6	Fix length	<FixLength>							
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 511	Not relevant	16#00							

Table 9.17

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 Program Special Fixcode" command is 16#0D. Parameterization of <FixLength> 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 comes after. This command can be used to program a code with a maximum length of 12 bytes.

Data telegram response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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#0D							
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	Content	Bit number							
		7	6	5	4	3	2	1	0
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 511	Not relevant	16#00							

Table 9.18

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> 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 of the tag. The size of the telegram extends up to and including <EPC/UII Byte X>.

The <Command> parameter has the value 16#0D and is reflected within the response telegram. The <Status> parameter has the value 16#00.

This is followed by a length specification via the EPC/UII information. It is always two 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 read tag.

Command end response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#0D							
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 511	Not relevant	16#00							

Table 9.19

The <FrameLength> constantly has a value of 16#0F in the response for the end of the "Single Program Special Fixcode" command. The fragment extends up to and including <Number of Read/Write Tags Byte 0>. 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#0D command telegram. The <Status> for the telegram that indicates the command end is 16#0F.

2026-02

The number of identified read / write tags is transmitted within four 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.

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 four 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.

By factory default, this command accesses four 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, data from the EPC/UII (bank 01) and TID (bank 10) can also be accessed.

The parameter <ByteAddress> specifies the start location of data to be read. The value of <ByteAddress> is based on bytes. This means that only multiples of four can be accessed. <Number of Bytes> defines the number of bytes to be read. The number of bytes must be a multiple of four.

Command:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#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 511	Not relevant	16#00							

Table 9.20

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 "Single Read 4-Byte Blocks" command is 16#10.

The parameter <ByteAddress> specifies the start location of data to be 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. They must all be set to the value 16#00.

Data telegram response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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#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 511	Not relevant	16#00							

Table 9.21

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> but will not be larger than the maximum allowed length of the fragment. The fragmentation counter <FragmentationCounter> has a starting value equal to the number of fragments to transmit minus one and is decremented each time a new fragment is transmitted. The last fragment transmitted will have a value of 16#00. 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 via the EPC/UII information. The EPC/UII information is always two 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 two 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.

Command end response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#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 511	Not relevant	16#00							

Table 9.22

The <FrameLength> has a value of 16#0F 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 four bytes. The number is displayed in ASCII coding.

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

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

Enhanced Read 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 four byte data blocks of the user memory (bank 11) for one or more tags within the detection zone. The read 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.

By factory default, this command accesses four 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, data from the EPC/UII (bank 01) and TID (bank 10) can also be accessed.

The parameter <ByteAddress> specifies the start location of data to be read. The value of <ByteAddress> is based on bytes. This means that only multiples of four can be accessed. <Number of Bytes> defines the number of bytes to be read. The number of bytes must be a multiple of four.

Command:

Byte	Contents	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0A							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#07							
Byte 5	Command	16#19							
Byte 6	Parameter/data	<ByteAddress (High Byte)>							
Byte 7	Parameter/data	<ByteAddress (Low Byte)>							
Byte 8	Parameter/data	<Number of Bytes (High Byte)>							
Byte 9	Parameter/data	<Number of Bytes (Low Byte)>							
Byte 10	Not relevant	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.23

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 parameter <ByteAddress> specifies the start location of data to be 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.

Data telegram response:

Byte	Contents	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	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)>							

2026-02

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

Table 9.24

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> but will not be larger than the maximum allowed length of the fragment. The fragmentation counter <FragmentationCounter> has a starting value equal to the number of fragments to transmit minus one and is decremented each time a new fragment is transmitted. The last fragment transmitted will have a value of 16#00. 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 via the EPC/UII information. It is always two 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 two 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.

Tag has left the detection zone:

Byte	Contents	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	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	Contents	Bit number							
		7	6	5	4	3	2	1	0
Byte 511	Not relevant	16#00							

Table 9.25

The length of the fragment <FrameLength> depends on the size of the EPC/UII leaving the detection zone. 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 via the EPC/UII information. It is always two 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 four 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 final telegram has the status value 16#0F and contains the number of tags that were written during execution of the command.

By factory default, this command accesses two 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, data for the EPC/UII (bank 01) can also be accessed.

The parameter <ByteAddress> specifies the start location of data to be written. The value of <ByteAddress> is based on bytes. This means that only multiples of four can be accessed. <Number of Bytes> defines the number of bytes to be written. The number of bytes must be a multiple of four.

Command:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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	...							

2026-02

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 511	Not relevant	16#00							

Table 9.26

The value of <FrameLength> depends on the number of bytes to be written but will not be larger than the maximum allowed length of the fragment. The fragment ends with the <Write Data Byte Y> byte. The fragmentation counter <FragmentationCounter> has a starting value equal to the number of fragments to transmit minus one and is decremented each time a new fragment is transmitted. The last fragment transmitted will have a value of 16#00. 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 parameter <ByteAddress> specifies the start location of data to be 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.

Data telegram response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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 511	Not relevant	16#00							

Table 9.27

The length of the fragment <FrameLength> depends on the size of the EPC/UII of the written 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 via the EPC/UII information. The EPC/UII information is always two 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.

Command end response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#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 511	Not relevant	16#00							

Table 9.28

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 tags written is transmitted within the four 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 four 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 der Werkseinstellung des Geräts greift der Befehl auf 4-Byte große Datenblöcke des User Memorys (Bank 11) zu. Über den Parameter "Memory Bank" (MB) kann die Speicherbank umgestellt werden. Hierdurch kann dieser Befehl ebenso Datenbereiche aus den Speicherbänken EPC/UII (Bank 01) und TID (Bank 10) beschreiben.

2026-02

The parameter <ByteAddress> specifies the start location of data to be written. The value of <ByteAddress> is based on bytes. This means that only multiples of four can be accessed. <Number of Bytes> defines the number of bytes to be written. The number of bytes must be a multiple of four.

Command:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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 511	Not relevant	16#00							

Table 9.29

The value of <FrameLength> depends on the number of bytes to be written but must not be larger than the maximum fragment size allowed. The fragment ends with the <Write Data Byte Y> byte. The fragmentation counter <FragmentationCounter> has a starting value equal to the number of fragments to transmit minus one and is decremented each time a new fragment is transmitted. The last fragment transmitted will have a value of 16#00. The length of the command telegram (<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 parameter <ByteAddress> specifies the start location of data to be 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 fragment are not relevant to the command and must all be set to 16#00.

Data telegram response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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	Content	Bit number							
		7	6	5	4	3	2	1	0
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 511	Not relevant	16#00							

Table 9.30

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

This is followed by a length specification via the EPC/UII information. The EPC/UII information is always two 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.

Tag has left the detection zone:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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							

2026-02

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 511	Not relevant	16#00							

Table 9.31

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 via the EPC/UII information. The EPC/UII information is always two 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 two 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.

By factory default, 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, data from the EPC/UII (bank 01) and TID (bank 10) can also be accessed.

The parameter <ByteAddress> specifies the start location of data to be read. The value of <ByteAddress> is based on bytes. This means that only multiples of two can be accessed. <Number of Bytes> defines the number of bytes to be read. The number of bytes must be a multiple of two.

Command:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#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 511	Not relevant	16#00							

Table 9.32

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 parameter <ByteAddress> specifies the start location of data to be 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. They must all be set to the value 16#00.

Data telegram response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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 511	Not relevant	16#00							

Table 9.33

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> but will not be larger than the maximum allowed length of the fragment. The fragmentation counter <FragmentationCounter> has a starting value equal to the number of fragments to transmit minus one and is decremented each time a new fragment is transmitted. The last fragment transmitted will have a value of 16#00. 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 via the EPC/UII information. The EPC/UII information is always two 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 two 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.

Command end response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#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 511	Not relevant	16#00							

Table 9.34

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 four bytes. The number is displayed in ASCII coding.

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

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

Enhanced Read 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 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.

By factory default, this command accesses four 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, data from the EPC/UII (bank 01) and TID (bank 10) can also be accessed.

The parameter <ByteAddress> specifies the start location of data to be read. The value of <ByteAddress> is based on bytes. This means that only multiples of two can be accessed. <Number of Bytes> defines the number of bytes to be read. The number of bytes must be a multiple of two.

Command:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame Length	16#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 511	Not relevant	16#00							

Table 9.35

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 parameter <ByteAddress> specifies the start location of data to be 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.

Data telegram response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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>							

2026-02

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

Table 9.36

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> but will not be larger than the maximum allowed length of the fragment. The fragmentation counter <FragmentationCounter> has a starting value equal to the number of fragments to transmit minus one and is decremented each time a new fragment is transmitted. The last fragment transmitted will have a value of 16#00. 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 via the EPC/UII information. It is always two 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 two 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.

Tag has left the detection zone:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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>							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
...	Parameter/data	...							
...	Parameter/data	<EPC / UII Byte X>							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.37

The length of the fragment <FrameLength> depends on the size of the EPC/UII leaving the detection zone. 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 via the EPC/UII information. It is always two 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 two 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 final telegram has the status value 16#0F and contains the number of tags that were written during execution of the command.

By factory default, 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, data for the EPC/UII (bank 01) can also be accessed.

The parameter <ByteAddress> specifies the start location of data to be written. The value of <ByteAddress> is based on bytes. This means that only multiples of two can be accessed. <Number of Bytes> defines the number of bytes to be written. The number of bytes must be a multiple of two.

Command:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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)>							

2026-02

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 10	Parameter/data	<Write Data Byte 1>							
Byte 11	Parameter/data	<Write Data Byte 2>							
...	Parameter/data	...							
Byte x	Parameter/data	<Write Data Byte Y>							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.38

The value of <FrameLength> depends on the number of bytes to be written but will not be larger than the maximum allowed length of the fragment. The fragment ends with the <Write Data Byte Y> byte. The fragmentation counter <FragmentationCounter> has a starting value equal to the number of fragments to transmit minus one and is decremented each time a new fragment is transmitted. The last fragment transmitted will have a value of 16#00. 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 parameter <ByteAddress> specifies the start location of data to be 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.

Data telegram response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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 511	Not relevant	16#00							

Table 9.39

The length of the fragment <FrameLength> 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 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 via the EPC/UII information. The EPC/UII information is always two 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.

Command end response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#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 511	Not relevant	16#00							

Table 9.40

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 tags written is transmitted within the four 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 two 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 der Werkseinstellung des Geräts greift der Befehl auf 2-Byte große Datenwörter des User Memorys (Bank 11) zu. Über den Parameter "Memory Bank" (MB) kann die Speicherbank umgestellt werden. Hierdurch kann dieser Befehl ebenso Datenbereiche aus den Speicherbänken EPC/UII (Bank 01) und TID (Bank 10) beschreiben.

The parameter <ByteAddress> specifies the start location of data to be written. The value of <ByteAddress> is based on bytes. This means that only multiples of two can be accessed. <Number of Bytes> defines the number of bytes to be written. The number of bytes must be a multiple of two.

Command:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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 511	Not relevant	16#00							

Table 9.41

The value of <FrameLength> depends on the number of bytes to be written but must not be larger than the maximum fragment size allowed. The fragment ends with the <Write Data Byte Y> byte. The fragmentation counter <FragmentationCounter> has a starting value equal to the number of fragments to transmit minus one and is decremented each time a new fragment is transmitted. The last fragment transmitted will have a value of 16#00. The length of the command telegram (<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 parameter <ByteAddress> specifies the start location of data to be 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 fragment are not relevant to the command and must all be set to 16#00.

Data telegram response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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 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 511	Not relevant	16#00							

Table 9.42

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 via the EPC/UII information. The EPC/UII information is always two 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.

Tag has left the detection zone:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	<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)>							

2026-02

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

Table 9.43

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 via the EPC/UII information. The EPC/UII information is always two 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.

9.2.2 System Commands

The system commands can be used to change the device configuration. Tags are not accessed and the UHF properties are not affected by this.

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_S	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							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.44

The <FrameLength> has the value 16#06 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 telegram in bytes (<TelegramLength>) is 16#03. The <Command> command code for the Quit command is 16#02.

Command end response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#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							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.45

The length of the <FrameLength> fragment constantly has a value of 17#07 because no further parameters are transmitted within the response. The <FrameLength> contains all bytes up to and including <Status>. The <FragmentationCounter> 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 reflected 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_S	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							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.46

Command end response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	<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							
...	...	16#XX							
Byte X	Version byte X	16#XX							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.47

9.2.3 Filter Commands

In the factory setting, each write/read command accesses all detected read / write tags in the detection zone. 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 out using the "Read Parameter" (RP) command via the FL parameter.

Set Filter Mask (FI)

The "Set Filter Mask" (FI) command can be used to asset 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_S	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	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 11	Parameter/data	<Length Mask (Low Byte)>							
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 511	Parameter/data	16#00							

Table 9.48

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_S	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 511	Not relevant	16#00							

Table 9.49

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.

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 9.50

Bit 0 <Truncation>

0 2 All filter data is transmitted; the default value is always 0 2

1 2 Only the part of the EPC or the UII that follows the mask is transmitted. The CRC is not recalculated; only the saved 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 2 OR link or value for setting only one filter

1 2 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 2 Not negated; transfer of all read / write tags that match the filter condition

1 2 Negated; transfer of all read / write tags that do not match the filter condition

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

00 2 Bank 00; not allowed

01 2 Bank 01; EPC/UII

10 2 Bank 10; TID (fixcode)

11 2 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 given in relation to the bit.

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):

Bank 01	Address no.															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0x0000	0x0001	0x0002	0x0003	0x0004	0x0005	0x0006	0x0007	0x0008	0x0009	0x000A	0x000B	0x000C	0x000D	0x000E	0x000F
Area	CRC															
	CRC 15	CRC 14	CRC 13	CRC 12	CRC 11	CRC 10	CRC 9	CRC 8	CRC 7	CRC 6	CRC 5	CRC 4	CRC 3	CRC 2	CRC 1	CRC 0

Bank 01	Address no.															
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	0x00 10	0x00 11	0x00 12	0x00 13	0x00 14	0x00 15	0x00 16	0x00 17	0x00 18	0x00 19	0x00 1A	0x00 1B	0x00 1C	0x00 1D	0x00 1E	0x00 1F
Area	PC word (high byte)								PC word (low byte)							
	Length4	Length3	Length2	Length1	Length0	UMI	XPC	Tog-gle	AFI/ RFU 7	AFI/ RFU 6	AFI/ RFU 5	AFI/ RFU 4	AFI/ RFU 3	AFI/ RFU 72	AFI/ RFU 1	AFI/ RFU 0
Bank 01	Address no.															
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
	0x00 20	0x00 21	0x00 22	0x00 23	0x00 24	0x00 25	0x00 26	0x00 27	0x00 28	0x00 29	0x00 2A	0x00 2B	0x00 2C	0x00 2D	0x00 2E	0x00 2F
Area	EPC/UII byte 1								EPC/UII byte 2							
	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC
Bank 01	Address no.															
	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
	0x00 30	0x00 31	0x00 32	0x00 33	0x00 34	0x00 35	0x00 36	0x00 37	0x00 38	0x00 39	0x00 3A	0x00 3B	0x00 3C	0x00 3D	0x00 3E	0x00 3F
Area	EPC/UII byte 3								EPC/UII byte 4							
	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC
Bank 01	Address no.															
	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
	0x00 40	0x00 41	0x00 42	0x00 43	0x00 44	0x00 45	0x00 46	0x00 47	0x00 48	0x00 49	0x00 4A	0x00 4B	0x00 4C	0x00 4D	0x00 4E	0x00 4F
Area	EPC/UII byte 5								EPC/UII byte 6							
	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC	EPC

Table 9.51

- 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 16#0020; EPC/UII length in Length0 … Length4 coded

Bank 10 Memory Structure (TID):

Bank 10	Address no.															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0x00 00	0x00 01	0x00 02	0x00 03	0x00 04	0x00 05	0x00 06	0x00 07	0x00 08	0x00 09	0x00 0A	0x00 0B	0x00 0C	0x00 0D	0x00 0E	0x00 0F
Area	Class Identifier Value 16#E0 or 16#E2											Tag Designer (Manufacturer)				
	1	1	1	0	0	0	1	0	XTID	S	F	MDID	MDID	MDID	MDID	MDID
Bank 10	Address no.															
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	0x00 10	0x00 11	0x00 12	0x00 13	0x00 14	0x00 15	0x00 16	0x00 17	0x00 18	0x00 19	0x00 1A	0x00 1B	0x00 1C	0x00 1D	0x00 1E	0x00 1F

2026-02

Area	Tag Designer (Manufacturer)				Tag Model											
	MDID	MDID	MDID	MDID	TMN	TMN	TMN	TMN	TMN	TMN	TMN	TMN	TMN	TMN	TMN	TMN
Bank 10	Address no.															
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
	0x0020	0x0021	0x0022	0x0023	0x0024	0x0025	0x0026	0x0027	0x0028	0x0029	0x002A	0x002B	0x002C	0x002D	0x002E	0x002F
Area	XTID Header Segment															
	TMDI	TMDI	TMDI	TMDI	TM	TM	TM	TM	TM	TM	TM	TM	TM	TM	TM	TM
Bank 10	Address no.															
	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
	0x0030	0x0031	0x0032	0x0033	0x0034	0x0035	0x0036	0x0037	0x0038	0x0039	0x003A	0x003B	0x003C	0x003D	0x003E	0x003F
Area	Serial Number Segment [47:32]															
	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN
Bank 10	Address no.															
	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
	0x0040	0x0041	0x0042	0x0043	0x0044	0x0045	0x0046	0x0047	0x0048	0x0049	0x004A	0x004B	0x004C	0x004D	0x004E	0x004F
Area	Serial Number Segment [31:16]															
	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN
Bank 10	Address no.															
	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
	0x0050	0x0051	0x0052	0x0053	0x0054	0x0055	0x0056	0x0057	0x0058	0x0059	0x005A	0x005B	0x005C	0x005D	0x005E	0x005F
Area	Serial Number Segment [15:0]															
	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN	SN

Table 9.52

- 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

Manufacturer	MDID	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			
Tyco International	000001110			

2026-02

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

Table 9.53

Bank 11 memory structure (user memory):

Bank 11	Address no.														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0x0000	0x0001	0x0002	0x0003	0x0004	0x0005	0x0006	0x0007	0x0008	0x0009	0x000A	0x000B	0x000C	0x000D	0x000E	0x000F
Area	User memory byte 1 (16#31)								User memory byte 2 (16#32)							
	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	0
Bank 11	Address no.															
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	0x0010	0x0011	0x0012	0x0013	0x0014	0x0015	0x0016	0x0017	0x0018	0x0019	0x001A	0x001B	0x001C	0x001D	0x001E	0x001F
Area	User memory byte 3 (16#33)								User memory byte 4 (16#34)							
	0	0	1	1	0	0	1	1	0	0	1	1	0	1	0	0
Bank 11	Address no.															
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
	0x0020	0x0021	0x0022	0x0023	0x0024	0x0025	0x0026	0x0027	0x0028	0x0029	0x002A	0x002B	0x002C	0x002D	0x002E	0x002F
Area	User memory byte 5 (16#41)								User memory byte 6 (16#42)							
	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0
Bank 11	Address no.															
	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
	0x0030	0x0031	0x0032	0x0033	0x0034	0x0035	0x0036	0x0037	0x0038	0x0039	0x003A	0x003B	0x003C	0x003D	0x003E	0x003F
Area	User memory byte 7 (16#43)								User memory byte 8 (16#44)							
	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0

Table 9.54

Activate Filter Mask (MF)

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

Command:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#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	Parameter/data	<FilterMode>							
Byte 7	Not relevant	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.55

Response filter activated:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#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	Status	16#00							
Byte 7	Not relevant	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.56

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.

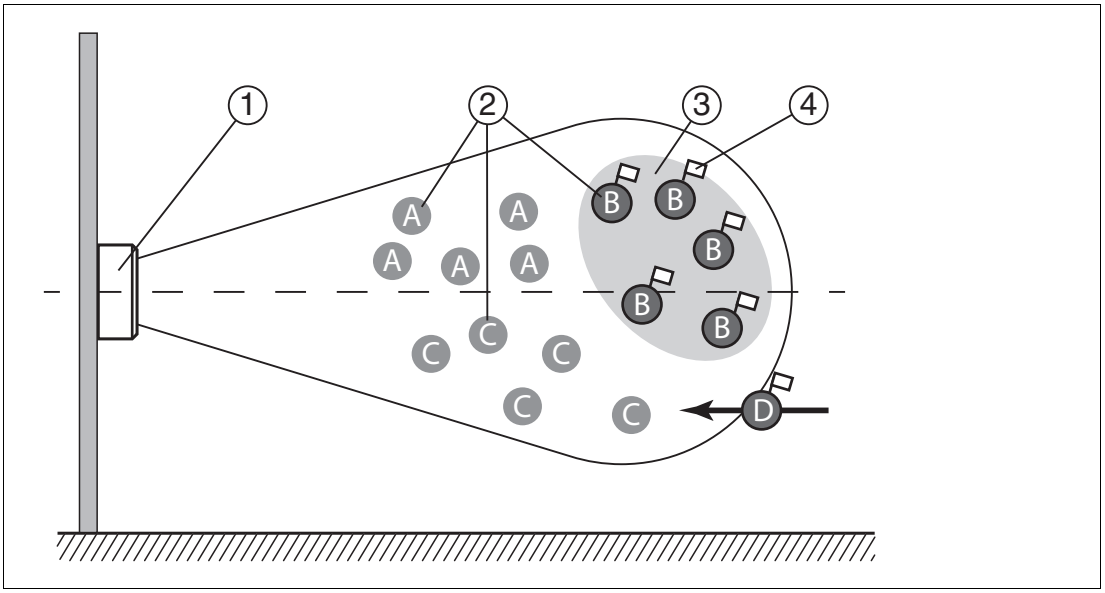


Figure 9.1

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

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.

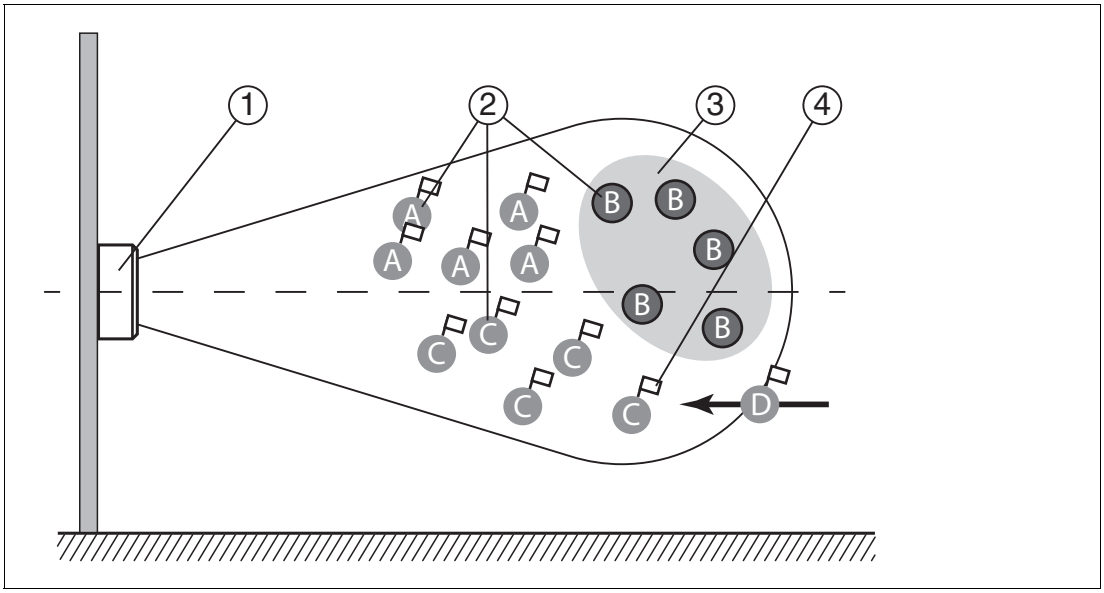


Figure 9.2

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.

9.2.4 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).

9.2.4.1 Basic Command Structure

Read Parameter (RP)

The "Read Parameter" (RP) command has the command code 16#BE and is used to read out 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_S	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)	<LengthParameter (High Byte)>							
Byte 10	Length parameter (low byte)	<LengthParameter (Low Byte)>							
Byte 11	Parameter data byte 1	<Parameter Data Byte 1>							
Byte 12	Parameter data byte 2	<Parameter Data Byte 2>							
...	...	...							
...	Parameter data byte X	<Parameter Data Byte X>							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.57

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 <LengthParameter (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 <LengthParameter (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").

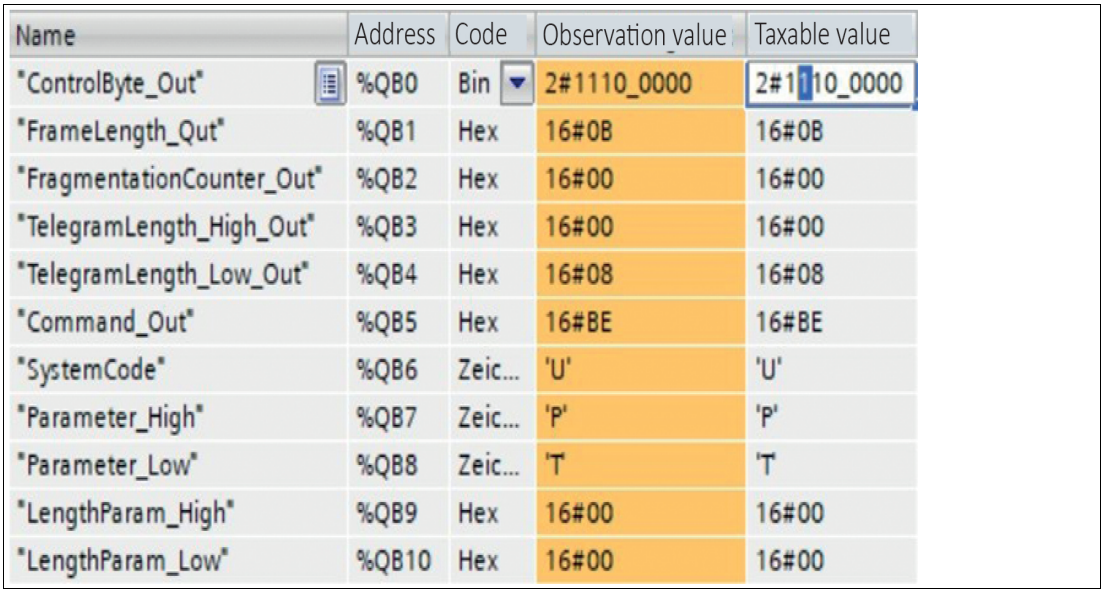
The <ParameterName> parameter specifies the parameter to be read. The value of <ParameterName> corresponds to the two characters of the parameter's short name.

<LengthParameter> 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.

Example: Command telegram to read out the Transmission Power PT (16#5054)



Name	Address	Code	Observation value	Taxable value
"ControlByte_Out"	%QB0	Bin	2#1110_0000	2#1110_0000
"FrameLength_Out"	%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 9.3

Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	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 511	Not relevant	16#00							

Table 9.58

The value of <FrameLength> depends on the size of the read-in 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-in 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-in parameter value. If <Status> has a different value, an error has occurred.

226722

The read-in <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.

Example: Response telegram to read out the Transmission 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 9.4

Write Parameter (WP)

The "Write Parameter" (WP) command has the command code 16#BF and is used to change a parameter in 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_S	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)>							
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	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 511	Not relevant	16#00							

Table 9.59

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 in. The value of <ParameterName> corresponds to the two 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 are not relevant and must all be set to 16#00. They must all be set to 16#00.

Example: Command telegram to change the Transmission Power PT (16#5054) to 50 mW (16#0032)

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 9.5

Response:

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#07							
Byte 2	Fragmentation counter	16#00							

2026-02

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#04							
Byte 5	Command	16#BF							
Byte 6	Status	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.60

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.

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 9.6

9.2.4.2 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	Transmission Power (PT) see "Transmission Power (PT)" on page 113 Sets the transmission power of the device	Read + write
NT 16#4E54	Number of Tags (NT) see "Number of Tags (NT) / Cancelation Criterion" on page 117 cancelation criterion for write/read command	Read + write
AP 16#4150	Antenna Polarization (AP) see "Antenna Polarization (AP)" on page 118 Sets the polarization plane of the device	Read + write
TA 16#5441	Tries Allowed (TA) see "Tries Allowed (TA) / Number of Access Attempts" on page 120 Number of access attempts on the read / write tag	Read + write
IF 16#4946	Additional Information (IF) see "Additional Information (IF) / Output of Additional Information" on page 122 Outputs additional information about accessing the read / write tag	Read + write
CD 16#4344	Transmission Channels (CD) see "Transmission Channels (CD)" on page 124 Sets the transmission channels used	• Device versions FR1* and FR2-08: read + write • Device version FR2*: read
E5 16#4535	Enhanced Status 5 (E5) see "Enhanced Status 5 (E5)" on page 128 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 130 Sets the memory bank for the "Read/Write Words" commands	Read + write
MF 16#4D46	Measure Reflection (MF) on page XX Measures the reflected transmission power	Read
NC 16#4E43	Number of Channels (NC) see "Number of Channels (NC)" on page 133 Sets the transmission channels used	• Device versions FR1* and FR2-08: read • Device version FR2*: read + write
QW 16#5157	Q Value QW see "Q Value QW" on page 136 Sets the number of time slots for accessing a tag	Read + write
RC 16#5243	Region Code (RC) see "Region Code (RC)" on page 138 Country identifier	Read
RD 16#5244	Reset to Default (RD) Reset to factory settings	Write
SM 16#534D	Sending Mode (SM) see "Sensing Mode (SM)" on page 140 Sets transmission pauses	Read + write
FL 16#464C	Filter List FL see "Filter List (FL) / Read Out Filter Mask" on page 142	Read

Table 9.61

The parameters are saved in a non-volatile memory.

2026-02

Transmission Power (PT)

The "Transmission Power" (PT) parameter can be used to change the power of the device. The transmission 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 transmission power level of PT1 is preset in the factory setting. It is possible to use up to ten power levels (PT1 … PT10) in the device. This enables a ramp to be generated for a continuous increase in the transmission power. When used in conjunction with a "Single Read" or "Single Write" command and the definition of the cancelation 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 transmission 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 cancelation 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 → 50mW$_{erp}$ (16#0032) • FR2-02 → 125mW$_{eirp}$ (16#007D) • FR2-03 → 80mW$_{erp}$ (16#0050) • FR1-04 → 50mW$_{erp}$ (16#0032) • FR2-07 →125mW$_{eirp}$ (16#007D) • FR2-08 →125mW$_{eirp}$ (16#007D) • FR2-09 → 125mW$_{eirp}$(16#007D) • FR2-10 →125mW$_{eirp}$ (16#007D) • FR2-13 → 80mW$_{erp}$ (16#0050) • FR2-17 → 80mW$_{erp}$ (16#0050)
Value range FR1-01, FR1-04	3mW (16#0003); 4mW (16#0004); 5mW (16#0005); 6mW (16#0006); 8mW (16#0008); 10mW (16#000A); 13mW (16#000D); 15mW (16#000F); 20mW (16#0014); 25mW (16#0019); 30mW (16#001E); 40mW (16#0028); 50mW (16#0032); 60mW (16#003C); 80mW (16#0050); 100mW (16#0064); 125mW (16#007D); 150mW (16#0096); 200mW (16#00C8); 250mW (16#00FA); 300mW (16#012C); 400mW (16#0190); 500mW (16#01F4); 600mW(16#0258); 800mW (16#0320); 1000mW (16#03E8)
Value range FR2-03, FR2-13	3mW (16#0003); 4mW (16#0004); 5mW (16#0005); 6mW (16#0006); 8mW (16#0008); 10mW (16#000A); 13mW (16#000D); 15mW (16#000F); 20mW (16#0014); 25mW (16#0019); 30mW (16#001E); 40mW (16#0028); 50mW (16#0032); 60mW (16#003C); 80mW (16#0050); 100mW (16#0064); 125mW (16#007D); 150mW (16#0096); 200mW (16#00C8); 250mW (16#00FA); 300mW (16#012C); 400mW (16#0190); 500mW (16#01F4); 600mW(16#0258); 800mW (16#0320)

Value range FR2-02, FR2-07,FR2-08, FR2-09 and FR2-10	3mW (16#0003); 4mW (16#0004); 5mW (16#0005); 6mW (16#0006); 8mW (16#0008); 10mW (16#000A); 13mW (16#000D); 15mW (16#000F); 20mW (16#0014); 25mW (16#0019); 30mW (16#001E); 40mW (16#0028); 50mW (16#0032); 60mW (16#003C); 80mW (16#0050); 100mW (16#0064); 125mW (16#007D); 150mW (16#0096); 200mW (16#00C8); 250mW (16#00FA); 300mW (16#012C); 400mW (16#0190); 500mW (16#01F4); 600mW(16#0258); 800mW (16#0320); 1000mW (16#03E8); 1250mW (16#04E2)
Value range FR2-17	3mW (16#0003); 4mW (16#0004); 5mW (16#0005); 6mW (16#0006); 8mW (16#0008); 10mW (16#000A); 13mW (16#000D); 15mW (16#000F); 20mW (16#0014); 25mW (16#0019); 30mW (16#001E); 40mW (16#0028); 50mW (16#0032); 60mW (16#003C); 80mW (16#0050); 100mW (16#0064); 125mW (16#007D); 150mW (16#0096); 200mW (16#00C8); 250mW (16#00FA); 300mW (16#012C); 400mW (16#0190); 500mW (16#01F4)

Table 9.62

⊕ Higher detection range if you increase the transmission power.

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

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

Tip

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

Note

You can operate the device only with the internally specified transmission power levels. You can use the Write PT command in the software to enter one or more transmission 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 transmission power levels:

The Write PT command can be used to set power levels PT1, PT2, and PT3 to the values 50 mW, 100 mW, and 500 mW.

- NT cancelation criterion at 16#FF (factory setting):
  Each read and write command is executed in succession for all three transmission power values. If one or more tags are found and successfully read/written for the first transmission power, the command is still executed with all other transmission powers to reach any other tags that may be available.
- NT cancelation criterion at 16#01 (canceled after one read / written tag):
  If no tags were detected during the inventory runs, the transmission 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 ten transmission power values. If several additional transmission channels are selected (go to "Transmission Channels (CD)" on page 50), all set powers are executed on each transmission channel for each read or write command.

Example: Command telegram to change the transmission power to PT1 = 500 mW (16#01F4)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#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#01							
Byte 12	PT1 parameter (low byte)	16#F4							
Byte 13	Not relevant	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.63

Example: Command telegram to change the transmission power to PT1 = 50 mW, PT2 = 100 mW, and PT3 = 500 mW

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#11							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#0E							
Byte 5	Command	16#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#32							
Byte 13	PT2 parameter (high byte)	16#00							
Byte 14	PT2 parameter (low byte)	16#64							
Byte 15	PT3 parameter (high byte)	16#01							
Byte 16	PT3 parameter (low byte)	16#F4							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 17	Not relevant	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.64

Example: Command telegram to read out the transmission power

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#BE							
Byte 6	System code	16#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 511	Not relevant	16#00							

Table 9.65

Example: Response telegram with the transmission 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_S	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 511	Not relevant	16#00							

Table 9.66

2026-02

Number of Tags (NT) / Cancelation 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 Transmission Power (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 character	NT (16#4E54)
Length of NT parameter value	1 byte
Factory setting	NT = 16#FF
Value range	16#01 … 16#FF

Table 9.67

Example: Command telegram to change the "Number of Tags" to 16#01

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	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 10	Length parameter (low byte)	16#01							
Byte 11	NT parameter	16#01							
Byte 12	Not relevant	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.68

Example: Command telegram to read out "Number of Tags"

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#BE							

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

Table 9.69

Example: Response telegram with the set value of "Number of Tags" = 255

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#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 511	Not relevant	16#00							

Table 9.70

Antenna Polarization (AP)

The Antenna Polarization (AP) parameter can be used to set the polarization mode of the device. The integrated antenna supports horizontal and vertical polarization planes. Combined mode is activated by factory default. The system can therefore dynamically switch between horizontal and vertical polarization planes. Each read / write attempt is carried out on both polarization planes.

Parameter character	AP (16#4150)
Length of NT parameter value	1 byte
Factory setting	"C" (combined mode) = 16#43
Value range	16#43 → Combined mode 16#48 → Horizontal polarization plane 16#56 → Vertical polarization plane

Table 9.71

The antenna for a UHF read / write tag has a polarization direction. This direction can be found in the tag datasheet. To achieve an optimal detection range, the orientation of the polarization planes in the read / write tag and the device must be the same.

If the orientation of the read / write tag or its polarization is not known, use the combined mode setting.

When using combined mode, a read / write attempt is made on both polarization planes. It will take longer to execute a read / write attempt if both polarization planes are used instead of one. To reduce the execution time, you can switch to the horizontal or vertical polarization.

Example: Command telegram to change the AP polarization to a horizontal plane (H = 16#48)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#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#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#01							
Byte 11	AP parameter	16#48							
Byte 12	Not relevant	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.72

Example: Command telegram to read out the antenna polarization

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#BE							
Byte 6	System code	16#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 511	Not relevant	16#00							

Table 9.73

Example: Response telegram with the set value of antenna polarization "C" (combined mode)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#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	AP parameter	16#43							
Byte 8	Not relevant	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.74

Tries Allowed (TA) / Number of Access Attempts

The Tries Allowed (TA) parameter is used to set the number of access attempts on a read / write tag.

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

Table 9.75

This parameter affects the execution of a "Single Read" or "Single Write" command. When the Tries Allowed parameter value is increased, the execution time for a single command 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.

If no read / write tag was detected while a read / write operation was executed, this is indicated by a response from the station. This is a status 16#0F telegram that contains the number of identified read / write tags. In this case, the number (<Number of Tags> parameter) is 16#30303030 ("0000"). The Tries Allowed value should be increased.

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

The Tries Allowed setting does not affect execution of the enhanced read or write commands, since these commands are continuously running.

Example: Command telegram to change TA to 5

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0C							

2026-02

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#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 511	Not relevant	16#00							

Table 9.76

Example: Command telegram to read out TA

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#BE							
Byte 6	System code	16#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 511	Not relevant	16#00							

Table 9.77

Example: Response telegram with the TA value set to 16#02

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#08							
Byte 2	Fragmentation counter	16#00							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#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 511	Not relevant	16#00							

Table 9.78

Additional Information (IF) / Output of Additional Information

The Additional Information (IF) parameter can be used to enable the transmission of additional information about read / write tag access. The additional information is transmitted if the read / write access to a read / write tag was successful.

The additional information is transmitted in a separate telegram with the status value (<Status>) 16#0B. This telegram directly follows the status 16#00 telegram with the read / write tag information.

The telegram with the additional information contains the following parameters:

<Info Type>	One byte type of the information telegram
<RSSI>	One byte of signal strength of the read / write tag response
<Channel>	One byte transmission channel
<Transmission Power>	Two byte transmission power

Table 9.79

The <Info Type> parameter specifies which data set is transmitted via the status 16#0B telegram. The <Info Type> value is 16#01. This data set contains the RSSI value of the read / write tag response.

The <RSSI> parameter returns the RSSI value of the read / write tag response. An RSSI value is a signal strength of the read / write tag response. The range of values for <RSSI> is between 16#00 … 16#64 (0 dec … 100 dec). The RSSI value is an indication of the signal strength of the communication between the device and tag. Based on two different RSSI values, and all other factors being equal, a lower RSSI value means a lower signal strength and a greater distance between the device and the tag. A larger RSSI value means a smaller distance between the tag and the device. The RSSI value is not a measurement of the distance of a read / write tag. Rather, it is a guideline for whether a tag is further away (lower value) from the device, or whether the tag is closer to the device (higher value).

When communicating with a read / write tag, frequency channels are used on the air interface. The <Channel> parameter indicates the number of the frequency channel on which the read / write tag was detected. The frequency channels used depend on the device version.

The <Transmission Power> parameter is used to specify the transmission power level at which the tag was identified. The device can be used to parameterize several transmission power levels (PT1 … PT10). These power levels are automatically applied when a device is running. <Transmission Power> indicates on which of the power levels the read / write tag was detected. It can be assumed that a tag that is further away is identified at a higher power level.

Parameter character	IF (16#4946)
Length of IF parameter value	1 byte
Factory setting	16#02 → Two access attempts 16#00 → No transmission of additional information
Value range	16#00 → No transmission of additional information 16#01 → Additional information with RSSI value is transmitted

Table 9.80

Example: Command telegram to activate the transmission of additional information

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#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#49 "I"							
Byte 8	Parameter name (low byte)	16#46 "F"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#01							
Byte 11	Parameter IF	16#01							
Byte 12	Not relevant	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.81

Example: Command telegram to read out the IF settings

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#BE							
Byte 6	System code	16#55 "U"							
Byte 7	Parameter name (high byte)	16#49 "I"							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 8	Parameter name (low byte)	16#46 "F"							
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 511	Not relevant	16#00							

Table 9.82

Example: Response telegram when additional information is being transmitted (16#01)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	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	Parameter IF	16#01							
Byte 8	Not relevant	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.83

Transmission Channels (CD)

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

- Device versions FR1* and FR2-08:
  These devices use the selected frequency list. The CD parameter allows the user to define the number of 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 number and sequence. In this case, the setting can only be read out.

Parameter character	CD (16#4344)
Length of CD parameter value	Variable depending on the device version • FR1-01 → Max. four bytes; max. four channels, adjustable • FR2-02 → 50 bytes; 50 channels, cannot be changed • FR2-03 → 16 bytes; 16 channels, cannot be changed • FR1-04 → Max. four bytes; max. four channels, adjustable • FR2-07 → 52 bytes; 52 channels, cannot be changed • FR2-08 → Max. four bytes; max. 4 channels, adjustable • 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
Factory setting	Depending on the device version • FR1-01 → 4, 10, 7, 13 (all four channels; length of four bytes) • FR2-02 → 31, 20, 49, 4, 33, ... (length 50 bytes) • FR2-03 → 3, 16, 13, 10, 7, ... (length 16 bytes) • FR1-04 → 1, 7 , 4, 10 (all four channels; length of four bytes) • FR2-07 → 38, 20, 42, 9, 27, ... (length 52 bytes) • FR2-08 → 5, 17, 11, 23 (length 4 bytes) • FR2-09 → 1, 7, 13, 4, 10, 16 (lenght 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 factory setting • FR1-04 → Up to four channels can be set; the number and sequence of channels is freely adjustable 16#01 → Channel 1 16#04 → Channel 4 16#07 → Channel 7 16#0A → Channel 10 • FR2-07 → The parameter values are fixed; no changes can be made; the value range is the factory setting • FR2-08 → Up to four channels can be set; the number and sequence of channels is freely adjustable 16#05 → Channel 5 16#0B → Channel 11 16#11 → Channel 17 16#17 → Channel 23 • 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

Table 9.84

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.see "Transmission Power (PT)" on page 113

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 reduced 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 two transmission channels, the two channels outside the frequency range must be set, i.e., channel 4 and channel 13.

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_S	0	16#0			
Byte 1	Frame length	16#0D							
Byte 2	Fragmentation counter	16#00							

2026-02

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#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 511	Not relevant	16#00							

Table 9.85

Example: Command telegram to read out CD

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#BE							
Byte 6	System code	16#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 511	Not relevant	16#00							

Table 9.86

Example: Response telegram with a 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_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#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 511	Not relevant	16#00							

Table 9.87

Enhanced Status 5 (E5)

The Enhanced Status 5 (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 → Five unsuccessful write attempts until status 16#05 telegram is output
Value range	16#00 … 16#FC → 0 to a maximum of 252 read / write attempts

Table 9.88

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 detection zone 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. This increase reduces the communication load between the device and the control system. 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

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_S	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 511	Not relevant	16#00							

Table 9.89

Example: Command telegram to read out 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_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#BE							
Byte 6	System code	16#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 511	Not relevant	16#00							

Table 9.90

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_S	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 511	Not relevant	16#00							

Table 9.91

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 four byte blocks and 16 bit 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 16-Bit Words
- Single / Enhanced Write 16-Bit Words

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

- Single / Enhanced Read Fixcode (access to the TID)
- Single / Enhanced Read Special Fixcode (access to EPC/UII)
- Single Program Special Fixcode (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 16#01 → Access to the EPC/UII memory bank 16#02 → Access to the TID memory bank 16#03 → Access to the user memory bank

Table 9.92

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_S	0	16#0			
Byte 1	Frame length	16#0C							
Byte 2	Fragmentation counter	16#00							

2026-02

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#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 511	Not relevant	16#00							

Table 9.93

Example: Command telegram to read out 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_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#BE							
Byte 6	System code	16#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 511	Not relevant	16#00							

Table 9.94

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_S	0	16#0			
Byte 1	Frame length	16#08							
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#05							
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 511	Not relevant	16#00							

Table 9.95

Measure Reflection MF (Measurement of reflected transmission power)

The parameter MF is used to measure the transmission power reflected at the antenna and environment. The output is one byte per transmission channel. The transmission channels depend on the country-specific settings,see chapter 4.3.The parameter is not output for the antenna polarization "combined". The reflected power in dBm is obtained by subtracting the value 100 from the return value in decimal notation.

Parameter character	MF
Value range	-18 .. +21 dBm[1]

Table 9.96

1. for values outside the value range, 0 or 255 is returned.

Example: Command telegram for measuring the reflected transmission power

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#BE							
Byte 6	System code	16#55 "U"							
Byte 7	Parameter name (high byte)	16#4D "M"							
Byte 8	Parameter name (low byte)	16#46 "F"							
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 511	Not relevant	16#00							

Table 9.97

Example: Response telegram with the reflected transmission power MF for the Region Code RC = 1 (4 channels)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram Length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#BE							
Byte 6	Status	16#00							
Byte 7	Parameter MF	16#53							
Byte 8	Parameter MF	16#54							
Byte 9	Parameter MF	16#54							
Byte 10	Parameter MF	16#54							
Byte 11	Not relevant	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.98

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 versions FR1* and FR2-08:
  These devices use a specific frequency list. The CD parameter is used to set the frequency list for this device. With the FR1* and FR2-08 device versions, the NC parameter can only be read out. It is not possible to use the NC parameter to change the frequency list. Changes must be made using the CD parameter.
- Device version FR2*:
  These device versions use 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 out 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, 10, 7, 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 4, 10, 7, and 13) • FR2-07 → 16#34 (52 channels; channel 1, 2, 3, ... 52) • FR2-08 → 16#04 (4 channels; channel 5, 17, 11 and 23) • FR2-09 → 16#06 (6 channels; channel 1, 4, 7, 10, 13 and 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 range of values 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 … 50 (16#01 … 16#32). • FR2-03 → 16#01 … 16#10 The number of channels used can be between 1 … 16 (16#01 … 16#10). • FR1-04 → 16#01 ... 16#04 The range of values 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. • FR2-07 → 16#01 ... 16#34 The number of channels used can be between 1 ... 52 (16#01 ... 16#34). • FR2-08 → 16#01 ... 16#04 The range of values 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. • FR2-09 → 16#01 ... 16#06 The number of channels used can be between 1 ... 6 (16#01 ... 16#06). • FR2-10 → 16#01 ... 16#0C The number of channels used can be between 1 ... 12 (16#01 ... 16#0C). • FR2-13 → 16#01 ... 16#08 The number of channels used can be between 1 ... 8 (16#01 ... 16#08). • FR2-17 → 16#01 ... 16#0E The number of channels used can be between 1 ... 14 (16#01 ... 16#0E).

Table 9.99

If you have parameterized several transmission channels with the parameter NC, 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. See "Transmission Power (PT)" on page 113

The number of transmission channels used affects the execution time of read / write commands. With 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., six channels, the following frequency channels are used when executing single commands:

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

2026-02

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_S	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 511	Not relevant	16#00							

Table 9.100

Example: Command telegram to read out 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_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#BE							
Byte 6	System code	16#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 511	Not relevant	16#00							

Table 9.101

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_S	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 511	Not relevant	16#00							

Table 9.102

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 character	QW (16#5157)
Length of QW parameter value	1 byte
Factory setting	16#02 → 2^2 = four time slots are used
Value range	16#00 … 16#07 → Between 2^0 … 2^7 (1 … 128) time slots are set.

Table 9.103

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 settings to a value of 1 (16#01), i.e., two time slots

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#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"							

2026-02

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 511	Not relevant	16#00							

Table 9.104

Example: Command telegram to read out the QW settings

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#BE							
Byte 6	System code	16#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 511	Not relevant	16#00							

Table 9.105

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

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#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 511	Not relevant	16#00							

Table 9.106

Region Code (RC)

The Region Code (RC) parameter can be used to read out the country identifier of the device. The country identifier can be used to identify the country-specific device settings.

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

Parameter character	RC (16#5243)
Length of RC parameter value	2 bytes
Factory setting	Depending on the device version • FR1-01 → 16#8001 (Europe and other countries that follow EN 302208) • FR2-02 → 16#8002 (USA, Canada, Mexico, Argentina) • FR2-03 → 16#8003 (China) • FR1-04 → 16#8004 (India) • FR2-07 → 16#8007 (Brazil, Colombia) • FR2-08 → 16#8008 (Japan) • FR2-09 → 16#8009 (South Korea) • FR2-10 → 16#800A (Australia) • FR2-13 → 16#800D (Malaysia) • FR2-17 → 16#8011 (Taiwan)
Value range	Parameter value cannot be changed

Table 9.107

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, Colombia
16#08	916,7 MHz ... 920,5 MHz Parameterizable frequency list	Japan

2026-02

Country Identifier	Occupied Frequency Bandwidth Frequency Access Method	Country or Region
16#09	917,2 MHz ... 920,4 MHz Frequency hopping spread spectrum	South Korea
16#0A	920 MHz ... 928 MHz Frequency hopping spread spectrum	Australia
16#0D	919 MHz … 923 MHz Frequency hopping spread spectrum	Malaysia
16#11	920 MHz ... 928 MHz Frequency hopping spread spectrum	Taiwan

Table 9.108

Example: Command telegram to read out the RC

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#BE							
Byte 6	SystemCode	16#55 "U"							
Byte 7	ParameterName (high byte)	16#52 "R"							
Byte 8	ParameterName (low byte)	16#43 "C"							
Byte 9	LengthParameter (high byte)	16#00							
Byte 10	LengthParameter (low byte)	16#00							
Byte 11	Not relevant	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.109

Example: Response telegram with the set RC value (16#8001; Europe)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#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	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 7	RC parameter (high byte)	16#80							
Byte 8	RC parameter (low byte)	16#01							
Byte 9	Not relevant	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.110

Sensing Mode (SM)

The Sensing Mode (SM) parameter defines a pause time in milliseconds. The pause time begins once all inventory rounds have been executed within an enhanced command. The device does not transmit during the parameterized time period. The device will start to transmit again once this pause time has elapsed.

This time allows the device to be operated continuously in higher ambient conditions; this is because interrupting UHF field transmission reduces heating of the device. If no transmission pauses are performed (factory setting), the maximum ambient temperature is 50 °C. Using transmission pauses can increase the maximum ambient temperature to 70 °C.

Parameter character	SM (16#534D)
Length of SM parameter value	2 bytes
Factory setting	Variable depending on the device version • FR1-01 → 16#0064 (100 ms) • FR2-02 → 16#0000 (0 ms, no pause) • FR2-03 → 16#0000 (0 ms, no pause) • FR1-04 → 16#0064 (100 ms) • FR2-07 → 16#0064 (100 ms) • FR2-08 → 16#0000 (0 ms, no pause) • FR2-10 → 16#0000 (0 ms, no pause) • FR2-13 → 16#0000 (0 ms, no pause) • FR2-17 → 16#0000 (0 ms, no pause)
Value range	Variable depending on the device version • FR1-01 → 16#0064 (100 ms) … 16#FFFF (65,535 ms) • FR2-02 → 16#0000 (0 ms, no pause) … 16#FFFF (65,535 ms) • FR2-03 → 16#0000 (0 ms, no pause) … 16#FFFF (65,535 ms) • FR1-04 → 16#0064 (100 ms) … 16#FFFF (65,535 ms) • FR2-07 → 16#0064 (100 ms) … 16#FFFF (65,535 ms) • FR2-08 → 16#0000 (0 ms, no pause) … 16#FFFF (65,535 ms) • FR2-10 → 16#0000 (0 ms, no pause) … 16#FFFF (65,535 ms) • FR2-13 → 16#0000 (0 ms, no pause) … 16#FFFF (65,535 ms) • FR2-17 → 16#0000 (0 ms, no pause) … 16#FFFF (65,535 ms)

Table 9.111

When using pause times, make note that no read / write tags can be identified during this time interval. This lack of identification needs to be considered for applications in which read / write tags move dynamically through the detection zone. In this case, the pause time should be shorter than the duration of the read / write tag's stay in the detection zone.

Example: Command telegram to change the pause time to 100 ms (SM = 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_S	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#53 "S"							
Byte 8	Parameter name (low byte)	16#4D "M"							
Byte 9	Length parameter (high byte)	16#00							
Byte 10	Length parameter (low byte)	16#02							
Byte 11	SM parameter (high byte)	16#00							
Byte 12	SM parameter (low byte)	16#64							
Byte 13	Not relevant	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.112

Example: Command telegram to read out the SM pause time

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#0B							
Byte 2	Fragmentation counter	16#00							
Byte 3	Telegram length (high byte)	16#00							
Byte 4	Telegram length (low byte)	16#08							
Byte 5	Command	16#BE							
Byte 6	System code	16#55 "U"							
Byte 7	Parameter name (high byte)	16#53 "S"							
Byte 8	Parameter name (low byte)	16#4D "M"							
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 511	Not relevant	16#00							

Table 9.113

Example: Response telegram with the set SM pause time (SM = 16#0000; no pause)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#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	SM parameter (high byte)	16#00							
Byte 8	SM parameter (low byte)	16#00							
Byte 9	Not relevant	16#00							
...	Not relevant	16#00							
Byte 511	Not relevant	16#00							

Table 9.114

Filter List (FL) / Read Out 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 character	FL (16#464C)
Value range	0 ... 2

Table 9.115

Example: Command telegram to read out the current configuration of filter mask 2 (16#02)

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 0	Control byte / frame length	D_S	U_M	U_S	0	16#0			
Byte 1	Frame length	16#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#00							
...	Not relevant	16#00							

2026-02

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
Byte 511	Not relevant	16#00							

Table 9.116

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	Control byte / frame length	D_S	U_M	U_S	0	<Frame length>			
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#BE							
Byte 6	Status	16#00							
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 511	Not relevant	16#00							

Table 9.117

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.

9.3 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.

10 Web Interface

The device can be conveniently configured and operated via an integrated web interface. The communication protocol settings are not important here.

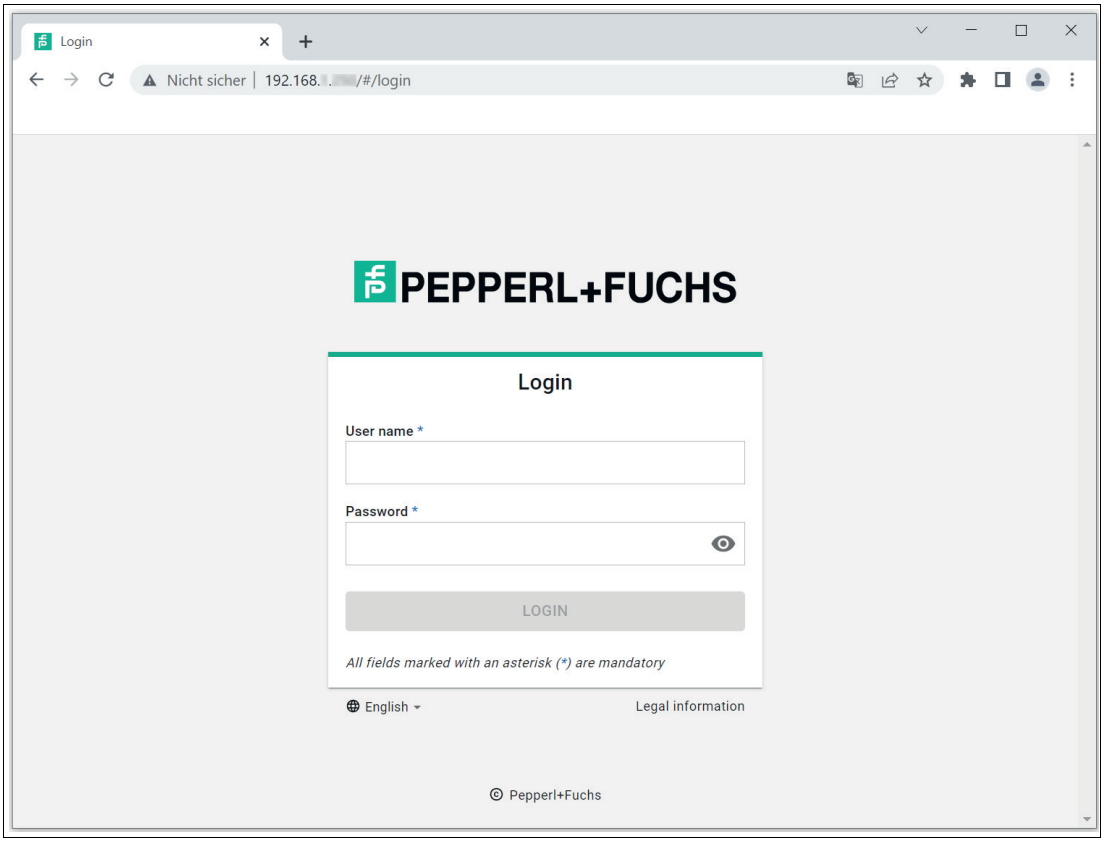


Figure 10.1

Logging in via the web interface

1. Make sure that your computer and the device are on the same network.
2. Enter the device's IP address in the address bar of your browser.

Note

For security reasons, each device is supplied with a unique password, which can be found in the enclosed documents or on the sticker attached to the device.

- Username: admin
- Password: See "Factory Password" sticker on the device

3. Enter the login information.

   The web interface is continuously expanded and can be operated intuitively using the text fields.

11 REST API

The device has a REST API (Representational State Transfer) for programming. An interface description is available on request.

12 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.

13 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 13.1

14 Appendix

14.1 Dimensions

IUT-F190-B40-2V1D-*

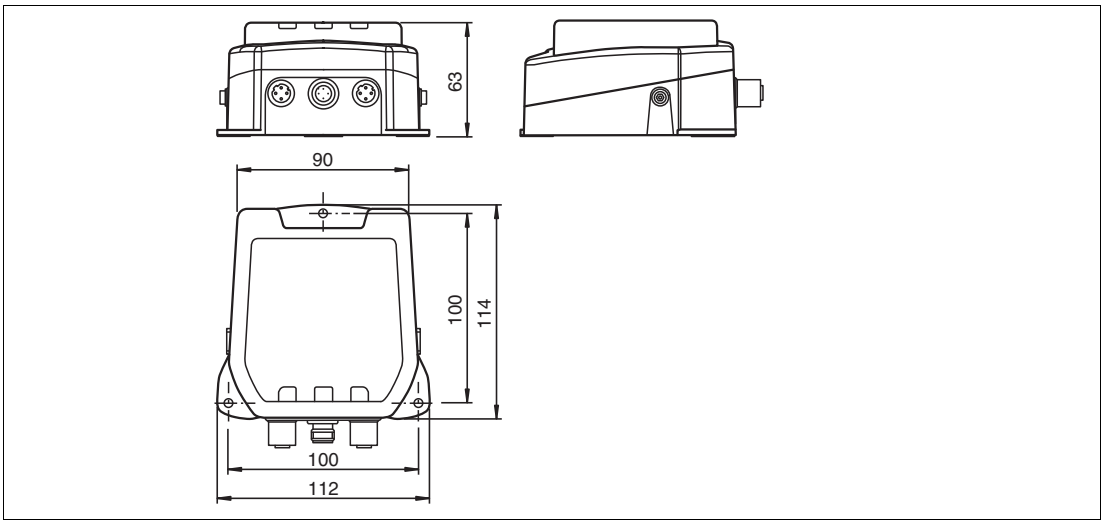


Figure 14.1

14.2 Examples for Expert Mode

14.2.1 Single Read Fixcode

"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) and 12 byte TID (16#E2 80 11 05 20 00 5A)

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	<Frame Length>		<Fragmentation Counter>	<Telegram Length>		<Command>	<Status>	<Length EPC/UII>		<PC-Word>		<EPC Byte 1>	<EPC Byte 2>	<EPC Byte 3>	<EPC Byte 4>	<EPC Byte 5>
HEX	16#X025		16#00	16#0022		16#01	16#00	16#000E		16#3400		16#30	16#14	16#F7	16#33	16#7C

Byte	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	<EPC Byte 6>	<EPC Byte 7>	<EPC Byte 8>	<EPC Byte 9>	<EPC Byte 10>	<EPC Byte 11>	<EPC Byte 12>	<Length TID>		<TID Byte 1>	<TID Byte 2>	<TID Byte 3>	<TID Byte 4>	<TID Byte 5>	<TID Byte 6>	<TID Byte 7>
HEX	16#00	16#1F	16#00	16#00	16#00	16#74	16#83	16#000C		16#E2	16#80	16#11	16#05	16#20	16#00	16#5A

Byte	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
	<TID Byte 8>	<TID Byte 9>	<TID Byte 10>	<TID Byte 11>	<TID Byte 12>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>
HEX	16#xx	16#xx	16#xx	16#xx	16#xx	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Table 14.1

"Command end" telegram with a tag read in while executing the command

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	<Frame Length>		<Fragmentation Counter>	<Telegram Length>		<Command>	<Status>	<No. of Tags Byte 1>	<No. of Tags Byte 2>	<No. of Tags Byte 3>	<No. of Tags Byte 4>	<unused>	<unused>	<unused>	<unused>	<unused>
HEX	16#X00B		16#00	16#0008		16#01	16#0F	16#30	16#30	16#30	16#31	16#00	16#00	16#00	16#00	16#00

Table 14.2

14.2.2 Enhanced Read Fixcode

"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) and 12 byte TID (16#E2 80 11 05 20 00 5A)

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	<Frame Length>		<Fragmentation Counter>	<Telegram Length>		<Command>	<Status>	<Length EPC/UII>		<PC-Word>		<EPC Byte 1>	<EPC Byte 2>	<EPC Byte 3>	<EPC Byte 4>	<EPC Byte 5>
Hex	16#X025		16#00	16#0022		16#1D	16#00	16#000E		16#3400		16#30	16#14	16#F7	16#33	16#7C

2026-02

Byte	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	<EPC Byte 6>	<EPC Byte 7>	<EPC Byte 8>	<EPC Byte 9>	<EPC Byte 10>	<EPC Byte 11>	<EPC Byte 12>	<Length TID>		<TID Byte 1>	<TID Byte 2>	<TID Byte 3>	<TID Byte 4>	<TID Byte 5>	<TID Byte 6>	<TID Byte 7>
HEX	16#00	16#1F	16#00	16#00	16#00	16#74	16#83	16#000C		16#E2	16#80	16#11	16#05	16#20	16#00	16#5A

Byte	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
	<TID Byte 8>	<TID Byte 9>	<TID Byte 10>	<TID Byte 11>	<TID Byte 12>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>
HEX	16#xx	16#xx	16#xx	16#xx	16#xx	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Table 14.3

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

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	<Frame Length>		<Fragmentation Counter>	<Telegram Length>		<Command>	<Status>	<Length EPC/UII>		<PC-Word>		<EPC Byte 1>	<EPC Byte 2>	<EPC Byte 3>	<EPC Byte 4>	<EPC Byte 5>
Hex	16#X017		16#00	16#0014		16#1D	16#05	16#000E		16#3400		16#30	16#14	16#F7	16#33	16#7C

Byte	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	<EPC Byte 6>	<EPC Byte 7>	<EPC Byte 8>	<EPC Byte 9>	<EPC Byte 10>	<EPC Byte 11>	<EPC Byte 12>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>
HEX	16#00	16#1F	16#00	16#00	16#00	16#74	16#83	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Table 14.4

14.2.3 Single Read Special Fixcode

"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	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	<Frame Length>		<Fragmentation Counter>	<Telegram Length>		<Command>	<Status>	<Length EPC/UII>		<PC-Word>		<EPC Byte 1>	<EPC Byte 2>	<EPC Byte 3>	<EPC Byte 4>	<EPC Byte 5>
Hex	16#X017		16#00	16#0014		16#0A	16#00	16#000E		16#3400		16#30	16#14	16#F7	16#33	16#7C

Byte	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	<EPC Byte 6>	<EPC Byte 7>	<EPC Byte 8>	<EPC Byte 9>	<EPC Byte 10>	<EPC Byte 11>	<EPC Byte 12>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>
HEX	16#00	16#1F	16#00	16#00	16#00	16#74	16#83	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Table 14.5

14.2.4 Enhanced Read Special Fixcode

"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	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	<Frame Length>		<Fragmentation Counter>	<Telegram Length>		<Command>	<Status>	<Length EPC/UII>		<PC-Word>		<EPC Byte 1>	<EPC Byte 2>	<EPC Byte 3>	<EPC Byte 4>	<EPC Byte 5>
Hex	16#X017		16#00	16#0014		16#71	16#00	16#000E		16#3400		16#30	16#14	16#F7	16#33	16#7C

Byte	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	<EPC Byte 6>	<EPC Byte 7>	<EPC Byte 8>	<EPC Byte 9>	<EPC Byte 10>	<EPC Byte 11>	<EPC Byte 12>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>
HEX	16#00	16#1F	16#00	16#00	16#00	16#74	16#83	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Table 14.6

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

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	<Frame Length>		<Fragmentation Counter>	<Telegram Length>		<Command>	<Status>	<Length EPC/UII>		<PC-Word>		<EPC Byte 1>	<EPC Byte 2>	<EPC Byte 3>	<EPC Byte 4>	<EPC Byte 5>
Hex	16#X017		16#00	16#0014		16#71	16#05	16#000E		16#3400		16#30	16#14	16#F7	16#33	16#7C

Byte	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	<EPC Byte 6>	<EPC Byte 7>	<EPC Byte 8>	<EPC Byte 9>	<EPC Byte 10>	<EPC Byte 11>	<EPC Byte 12>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>
HEX	16#00	16#1F	16#00	16#00	16#00	16#74	16#83	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Table 14.7

14.2.5 Single Program Special Fixcode

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

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	<Frame Length>		<Fragmentation Counter>	<Telegram Length>		<Command>	<Fix Length>	<PC-Word>		<Write EPC/UII Byte 1>	<Write EPC/UII Byte 2>	<Write EPC/UII Byte 3>	<Write EPC/UII Byte 4>	<Write EPC/UII Byte 5>	<Write EPC/UII Byte 6>	<Write EPC/UII Byte 7>
Hex	16#X015		16#00	16#0012		16#0D	16#0E	16#3000		16#11	16#22	16#33	16#44	16#55	16#66	16#77

Byte	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	<Write EPC/UII Byte 8>	<Write EPC/UII Byte 9>	<Write EPC/UII Byte 10>	<Write EPC/UII Byte 11>	<Write EPC/UII Byte 12	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>
HEX	16#88	16#99	16#00	16#11	16#22	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Table 14.8

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

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	<Frame Length>		<Fragmentation Counter>	<Telegram Length>		<Command>	<Fix Length>	<PC-Word>		<Write EPC/UII Byte 1>	<Write EPC/UII Byte 2>	<Write EPC/UII Byte 3>	<Write EPC/UII Byte 4>	<unused>	<unused>	<unused>
Hex	16#X00D		16#00	16#000A		16#0D	16#06	16#1100		16#aa	16#bb	16#cc	16#dd	16#00	16#00	16#00

Table 14.9

2026-02

14.2.6 Single Read 4-Byte Blocks

Command telegram to read out eight bytes starting from byte address 4

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	<Frame Length>		<Fragmentation Counter>	<Telegram Length>		<Command>	<ByteAddress>		<Number of Bytes>		<unused>	<unused>	<unused>	<unused>	<unused>	<unused>
Hex	16#X00A		16#00	16#0007		16#10	16#0004		16#0008		16#00	16#00	16#00	16#00	16#00	16#00

Table 14.10

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 data read in (16#3132333435363738 / ASCII "12345678") from the user memory

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	<Frame Length>		<Fragmentation Counter>	<Telegram Length>		<Command>	<Status>	<Length EPC/UII>		<PC-Word>		<EPC Byte 1>	<EPC Byte 2>	<EPC Byte 3>	<EPC Byte 4>	<EPC Byte 5>
Hex	16#X021		16#00	16#001E		16#10	16#00	16#000E		16#3400		16#30	16#14	16#F7	16#33	16#7C

Byte	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	<EPC Byte 6>	<EPC Byte 7>	<EPC Byte 8>	<EPC Byte 9>	<EPC Byte 10>	<EPC Byte 11>	<EPC Byte 12>	<Length Data>		<User Data Byte 1>	<User Data Byte 2>	<User Data Byte 3>	<User Data Byte 4>	<User Data Byte 5>	<User Data Byte 6>	<User Data Byte 7>
HEX	16#00	16#1F	16#00	16#00	16#00	16#74	16#83	16#0008		16#31	16#32	16#33	16#34	16#35	16#36	16#37

Byte	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
	<User Data Byte 8>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>
HEX	16#38	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Table 14.11

14.2.7 Enhanced Read 4-Byte Blocks

Command telegram to read out 12 bytes starting from byte address 0

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	<Frame Length>		<Fragmentation Counter>	<Telegram Length>		<Command>	<ByteAddress>		<Number of Bytes>		<unused>	<unused>	<unused>	<unused>	<unused>	<unused>
Hex	16#X00A		16#00	16#0007		16#19	16#0000		16#000C		16#00	16#00	16#00	16#00	16#00	16#00

Table 14.12

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 data read in (16#313233343536373839303132 / ASCII "123456789012") from the user memory

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	<Frame Length>		<Fragmentation Counter>	<Telegram Length>		<Command>	<Status>	<Length EPC/UII>		<PC-Word>		<EPC Byte 1>	<EPC Byte 2>	<EPC Byte 3>	<EPC Byte 4>	<EPC Byte 5>
Hex	16#X021		16#00	16#001E		16#19	16#00	16#000E		16#3400		16#30	16#14	16#F7	16#33	16#7C

Byte	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	<EPC Byte 6>	<EPC Byte 7>	<EPC Byte 8>	<EPC Byte 9>	<EPC Byte 10>	<EPC Byte 11>	<EPC Byte 12>	<Length Data>		<User Data Byte 1>	<User Data Byte 2>	<User Data Byte 3>	<User Data Byte 4>	<User Data Byte 5>	<User Data Byte 6>	<User Data Byte 7>
HEX	16#00	16#1F	16#00	16#00	16#00	16#74	16#83	16#000C		16#31	16#32	16#33	16#34	16#35	16#36	16#37

Byte	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
	<User Data Byte 8>	<User Data Byte 9>	<User Data Byte 10>	<User Data Byte 11>	<User Data Byte 12>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>
HEX	16#38	16#39	16#30	16#31	16#32	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Table 14.13

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

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	<Frame Length>		<Fragmentation Counter>	<Telegram Length>		<Command>	<Status>	<Length EPC/UII>		<PC-Word>		<EPC Byte 1>	<EPC Byte 2>	<EPC Byte 3>	<EPC Byte 4>	<EPC Byte 5>
Hex	16#X017		16#00	16#0014		16#19	16#05	16#000E		16#3400		16#30	16#14	16#F7	16#33	16#7C

Byte	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	<EPC Byte 6>	<EPC Byte 7>	<EPC Byte 8>	<EPC Byte 9>	<EPC Byte 10>	<EPC Byte 11>	<EPC Byte 12>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>
HEX	16#00	16#1F	16#00	16#00	16#00	16#74	16#83	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Table 14.14

14.2.8 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							

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							

2026-02

Byte	Content	Bit number							
		7	6	5	4	3	2	1	0
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							

14.2.9 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	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	<Frame Length>		<Fragmentation Counter>	<Telegram Length>		<Command>	<ByteAddress>		<Number of Bytes> 16#0010		<Write Data Byte 1>	<Write Data Byte 2>	<Write Data Byte 3>	<Write Data Byte 4>	<Write Data Byte 5>	<Write Data Byte 6>
Hex	16#X01A		16#00	16#0017		16#1A	16#0000		16#3400		16#00	16#11	16#22	16#33	16#44	16#55

Byte	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	<Write Data Byte 7>	<Write Data Byte 8>	<Write Data Byte 9>	<Write Data Byte 10>	<Write Data Byte 11>	<Write Data Byte 12>	<Write Data Byte 13>	<Write Data Byte 14>	<Write Data Byte 15>	<Write Data Byte 16>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>
HEX	16#66	16#77	16#88	16#99	16#aa	16#bb	16#cc	16#dd	16#ee	16#ff	16#00	16#00	16#00	16#00	16#00	16#00

Table 14.15

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	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	<Frame Length>		<Fragmentation Counter>	<Telegram Length>		<Command>	<Status>	<Length EPC/UII>		<PC-Word>		<EPC Byte 1>	<EPC Byte 2>	<EPC Byte 3>	<EPC Byte 4>	<EPC Byte 5>
Hex	16#X017		16#00	16#0014		16#1A	16#05	16#000E		16#3400		16#30	16#14	16#F7	16#33	16#7C

Byte	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	<EPC Byte 6>	<EPC Byte 7>	<EPC Byte 8>	<EPC Byte 9>	<EPC Byte 10>	<EPC Byte 11>	<EPC Byte 12>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>	<unused>
HEX	16#00	16#1F	16#00	16#00	16#00	16#74	16#83	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00

Table 14.16

14.3 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**

2026-02

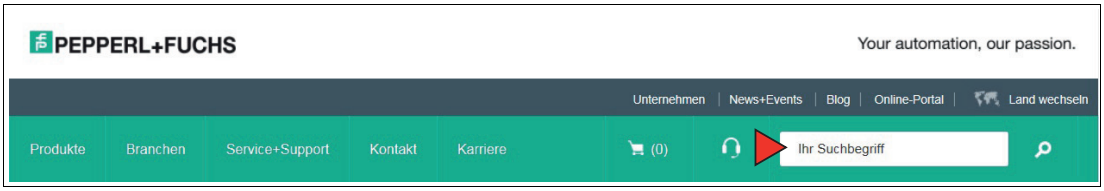
PEPPERL+FUCHS

hex	dec	ASCII	hex	dec	ASCII	hex	dec	ASCII	hex	dec	ASCII
15	21	NAK	**35**	**53**	**5**	55	85	U	**75**	**117**	**u**
16	22	SYN	**36**	**54**	**6**	56	86	V	**76**	**118**	**v**
17	23	ETB	**37**	**55**	**7**	57	87	W	**77**	**119**	**w**
18	24	CAN	**38**	**56**	**8**	58	88	X	**78**	**120**	**x**
19	25	EM	**39**	**57**	**9**	59	89	Y	**79**	**121**	**y**
1A	26	SUB	**3A**	**58**	**:**	5A	90	Z	**7A**	**122**	**z**
1B	27	ESC	**3B**	**59**	**;**	5B	91	[	**7B**	**123**	**{**
1C	28	FS	**3C**	**60**	**<**	5C	92	\	**7C**	**124**	**|**
1D	29	GS	**3D**	**61**	**=**	5D	93	]	**7D**	**125**	**}**
1E	30	RS	**3E**	**62**	**>**	5E	94	^	**7E**	**126**	**~**
1F	31	US	**3F**	**63**	**?**	5F	95	_	**7F**	**127**	**DEL**

14.4 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**.



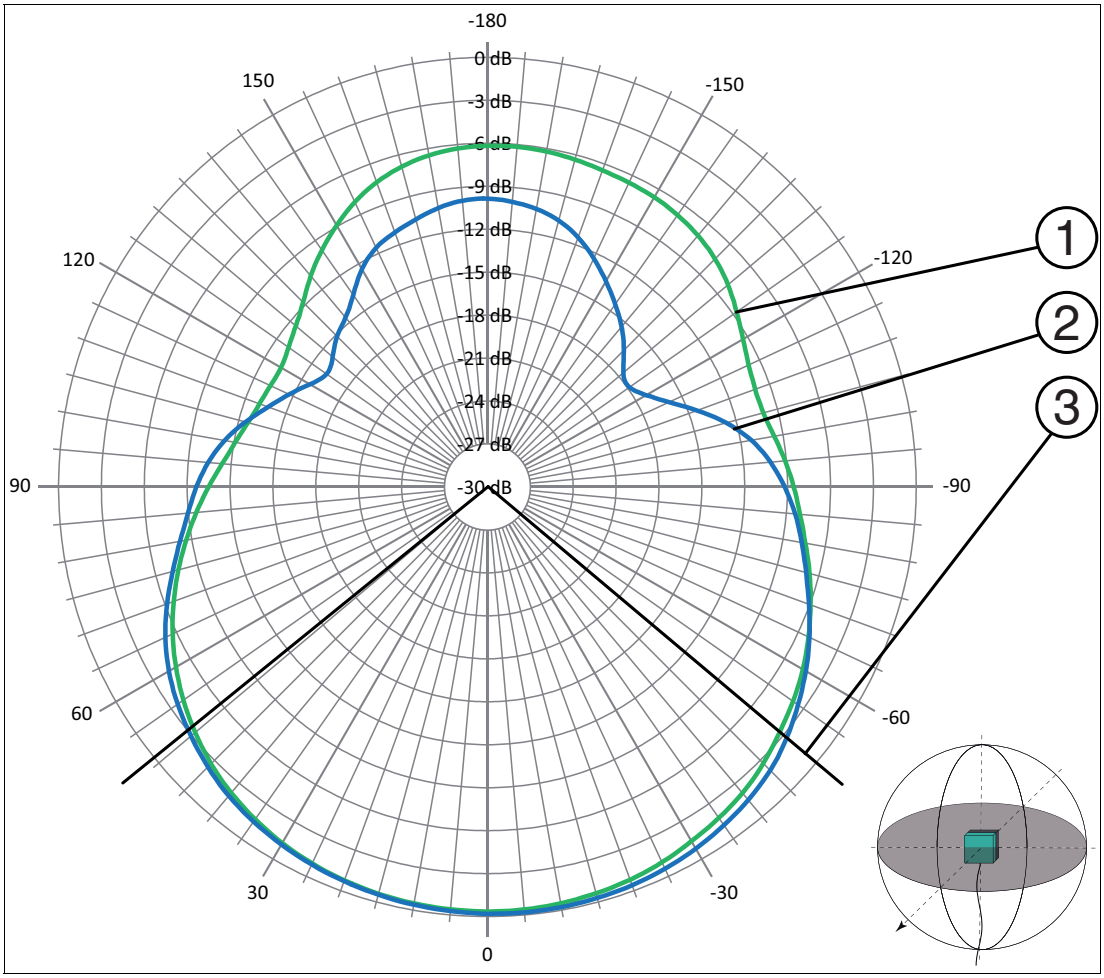
A list of all available documents is displayed.

14.5 Antenna Diagram

The antenna diagrams show the electric field strength in the far field depending on the orientation. The front of the device points toward 0°.

IUT-F190-B40-2V1D-FR2*

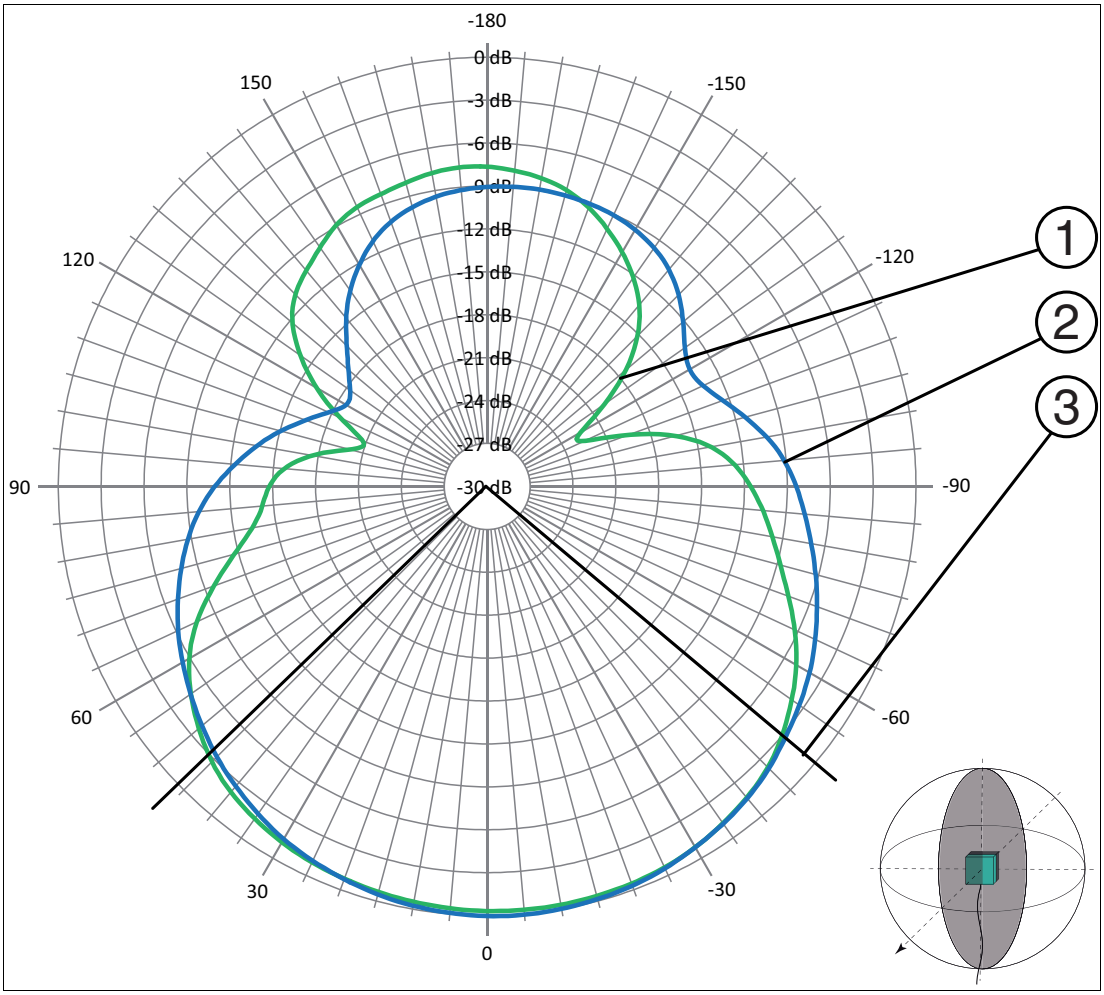
Horizontal cut[1]



1. Device in 0° direction

Figure 14.2 1 - Horizontal polarization
2 - Vertical polarization
3 - Full width at half maximum

Vertical cut[1]



1. Device in 0° direction

Figure 14.3 1 - Horizontal polarization
2 - Vertical polarization
3 - Full width at half maximum

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

© Pepperl+Fuchs · Subject to modifications

/ DOCT-6943J