

**IUT-F191-IO-V1-FR\***

**UHF-Schreib-/Lesestation  
mit integrierter IO-Link-  
Schnittstelle**

**Handbuch**



---

Es gelten die Allgemeinen Lieferbedingungen für Erzeugnisse und Leistungen der Elektroindustrie, herausgegeben vom Zentralverband Elektroindustrie (ZVEI) e. V. in ihrer neuesten Fassung sowie die Ergänzungsklausel: "Erweiterter Eigentumsvorbehalt".

**Weltweit**

Pepperl+Fuchs-Gruppe

Lilienthalstr. 200

68307 Mannheim

Deutschland

Telefon: +49 621 776 - 0

E-Mail: [info@de.pepperl-fuchs.com](mailto:info@de.pepperl-fuchs.com)

<https://www.pepperl-fuchs.com>

<b>1</b>	<b>Einleitung .....</b>	<b>6</b>
1.1	Inhalt des Dokuments .....	6
1.2	Zielgruppe, Personal .....	6
1.3	Bestimmungsgemäße Verwendung .....	6
1.4	Verwendete Symbole.....	7
1.5	Festlegungen.....	7
<b>2</b>	<b>Zertifikate und Zulassungen.....</b>	<b>10</b>
2.1	Konformitätserklärung (RE Directive 2014/53/EU).....	10
2.2	FCC-Information.....	10
2.3	IC-Information .....	11
2.4	IFT-Information .....	11
2.5	NCC-Information .....	11
2.6	ANATEL-Information .....	12
2.7	UL-Information .....	12
2.8	Weitere länderspezifische Zulassungen .....	12
<b>3</b>	<b>Produktbeschreibung .....</b>	<b>13</b>
3.1	RFID-Frequenzbänder .....	13
3.2	UHF Allgemein .....	13
3.2.1	Vorteile von UHF.....	13
3.2.2	Anwendungen für UHF-Systeme .....	13
3.2.3	Speicherstruktur eines Transponders nach EPC Gen 2 (ISO/IEC 18000-63).....	14
3.2.4	Elektronischer Produkt-Code EPC.....	17
3.2.5	Einfluss verschiedener Materialien auf die Reichweite .....	17
3.2.6	Dense Reader Mode (DRM) .....	18
3.2.7	Frequenzsprungverfahren.....	19
3.2.7.1	China .....	19
3.2.7.2	USA .....	19
3.2.8	Relevante Normen zu UHF .....	20
3.3	Einsatzländer .....	20
3.3.1	Europäische Union .....	21
3.3.2	Australien.....	21
3.3.3	Brasilien.....	21
3.3.4	China .....	21
3.3.5	Indien.....	22
3.3.6	Kanada .....	22
3.3.7	Malaysia .....	22
3.3.8	Mexiko .....	22
3.3.9	Südkorea .....	22
3.3.10	Taiwan .....	23
3.3.11	Vereinigte Staaten von Amerika .....	23

3.4	<b>Allgemeine Funktionen und Merkmale .....</b>	<b>24</b>
3.5	<b>Anzeigen und Bedienelemente.....</b>	<b>25</b>
3.6	<b>Elektrischer Anschluss .....</b>	<b>25</b>
3.7	<b>IO-Link Schnittstelleneigenschaften.....</b>	<b>26</b>
3.8	<b>Zubehör.....</b>	<b>26</b>
3.8.1	Datenträger.....	26
3.8.2	Verbindungskabel IO-Link.....	26
<b>4</b>	<b>Installation .....</b>	<b>27</b>
4.1	<b>Lagerung und Transport.....</b>	<b>27</b>
4.2	<b>Auspacken .....</b>	<b>27</b>
4.3	<b>Montage .....</b>	<b>27</b>
4.3.1	Orientierung im Raum .....	29
4.3.2	Mindestabstände .....	30
4.3.3	Polarisation .....	30
4.4	<b>Anschluss .....</b>	<b>31</b>
<b>5</b>	<b>Inbetriebnahme .....</b>	<b>32</b>
5.1	<b>Betriebsarten.....</b>	<b>32</b>
<b>6</b>	<b>Bedienung .....</b>	<b>33</b>
6.1	<b>Allgemeines .....</b>	<b>33</b>
6.2	<b>Interferenz aufgrund von Mehrwegeausbreitung .....</b>	<b>33</b>
6.3	<b>Mehrere Transponder im Erfassungsbereich.....</b>	<b>34</b>
6.4	<b>Lesealgorithmus .....</b>	<b>34</b>
<b>7</b>	<b>EasyMode .....</b>	<b>35</b>
7.1	<b>Befehlsübersicht.....</b>	<b>35</b>
7.2	<b>Grundlegende Struktur der Prozessdaten .....</b>	<b>36</b>
7.2.1	Ausgangsprozessdaten (SPS -> Gerät) .....	36
7.2.2	Eingangsprozessdaten (Gerät -> SPS) .....	38
7.2.3	Ablaufdiagramme.....	42
7.2.4	Timing.....	46
7.3	<b>Easy Mode mit PACTware .....</b>	<b>48</b>
<b>8</b>	<b>ExpertMode .....</b>	<b>53</b>
8.1	<b>Grundlegender Befehlsablauf .....</b>	<b>53</b>
8.2	<b>Legende .....</b>	<b>53</b>
8.3	<b>Aufbau OUTPUT-Telegramm.....</b>	<b>54</b>

<b>8.4</b>	<b>Aufbau INPUT-Telegramm .....</b>	<b>55</b>
<b>8.5</b>	<b>Handshake-Verfahren .....</b>	<b>56</b>
<b>8.6</b>	<b>Befehlsübersicht.....</b>	<b>58</b>
8.6.1	Schreib-/Lesebefehle.....	59
8.6.2	Systembefehle.....	98
8.6.3	Filterbefehle.....	100
8.6.4	UHF-Konfigurationbefehle .....	108
8.6.4.1	Grundlegende Befehlsstruktur .....	109
8.6.4.2	Übersicht UHF-Parameter.....	113
<b>8.7</b>	<b>Fehler-/Statusmeldungen .....</b>	<b>140</b>
<b>9</b>	<b>Wartung und Pflege.....</b>	<b>141</b>
<b>10</b>	<b>Störungsbeseitigung.....</b>	<b>142</b>
<b>11</b>	<b>Anhang .....</b>	<b>143</b>
<b>11.1</b>	<b>Abmessungen.....</b>	<b>143</b>
<b>11.2</b>	<b>Beispiele Expert-Mode.....</b>	<b>143</b>
11.2.1	Single Read EPC/UII .....	143
11.2.2	Enhanced Read EPC/UII .....	144
11.2.3	Single Write EPC/UII.....	145
11.2.4	Single Read 4-Byte Blocks .....	147
11.2.5	Enhanced Read 4-Byte Blocks .....	148
11.2.6	Single Write 4-Byte Blocks.....	150
11.2.7	Enhanced Write 4-Byte Blocks .....	151
<b>11.3</b>	<b>ASCII-Tabelle .....</b>	<b>154</b>
<b>11.4</b>	<b>Erfassungsbereich .....</b>	<b>155</b>

# 1 Einleitung

## 1.1 Inhalt des Dokuments

Dieses Dokument beinhaltet Informationen, die Sie für den Einsatz Ihres Produkts in den zutreffenden Phasen des Produktlebenszyklus benötigen. Dazu können zählen:

- Produktidentifizierung
- Lieferung, Transport und Lagerung
- Montage und Installation
- Inbetriebnahme und Betrieb
- Instandhaltung und Reparatur
- Störungsbeseitigung
- Demontage
- Entsorgung



---

**Hinweis!**

Entnehmen Sie die vollständigen Informationen zum Produkt der weiteren Dokumentation im Internet unter [www.pepperl-fuchs.com](http://www.pepperl-fuchs.com).

---



---

**Hinweis!**

Sie finden spezifische Geräteinformationen wie z. B. das Baujahr, indem Sie den QR-Code auf dem Gerät scannen. Alternativ geben Sie die Seriennummer in der Seriennummernsuche unter [www.pepperl-fuchs.com](http://www.pepperl-fuchs.com) ein.

---

Die Dokumentation besteht aus folgenden Teilen:

- vorliegendes Dokument
- Datenblatt

Zusätzlich kann die Dokumentation aus folgenden Teilen bestehen, falls zutreffend:

- EU-Baumusterprüfbescheinigung
- EU-Konformitätserklärung
- Konformitätsbescheinigung
- Zertifikate
- Control Drawings
- Betriebsanleitung
- Handbuch funktionale Sicherheit
- weitere Dokumente

## 1.2 Zielgruppe, Personal

Die Verantwortung hinsichtlich Planung, Montage, Inbetriebnahme, Betrieb, Instandhaltung und Demontage liegt beim Anlagenbetreiber.

Nur Fachpersonal darf die Montage, Inbetriebnahme, Betrieb, Instandhaltung und Demontage des Produkts durchführen. Das Fachpersonal muss die Betriebsanleitung und die weitere Dokumentation gelesen und verstanden haben.

Machen Sie sich vor Verwendung mit dem Gerät vertraut. Lesen Sie das Dokument sorgfältig.

## 1.3 Bestimmungsgemäße Verwendung

Betreiben Sie das Gerät ausschließlich wie in dieser Anleitung beschrieben. Nur so ist die sichere Funktion des Geräts und der angeschlossenen Systeme gewährleistet.

Der Schutz von Betriebspersonal und Anlage ist nur gegeben, wenn das Gerät entsprechend seiner bestimmungsgemäßen Verwendung eingesetzt wird.

## 1.4 Verwendete Symbole

Dieses Dokument enthält Symbole zur Kennzeichnung von Warnhinweisen und von informativen Hinweisen.

### Warnhinweise

Sie finden Warnhinweise immer dann, wenn von Ihren Handlungen Gefahren ausgehen können. Beachten Sie unbedingt diese Warnhinweise zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden.

Je nach Risikostufe werden die Warnhinweise in absteigender Reihenfolge wie folgt dargestellt:



#### Gefahr!

Dieses Symbol warnt Sie vor einer unmittelbar drohenden Gefahr.

Falls Sie diesen Warnhinweis nicht beachten, drohen Personenschäden bis hin zum Tod.



#### Warnung!

Dieses Symbol warnt Sie vor einer möglichen Störung oder Gefahr.

Falls Sie diesen Warnhinweis nicht beachten, können Personenschäden oder schwerste Sachschäden drohen.



#### Vorsicht!

Dieses Symbol warnt Sie vor einer möglichen Störung.

Falls Sie diesen Warnhinweis nicht beachten, können das Produkt oder daran angeschlossene Systeme und Anlagen gestört werden oder vollständig ausfallen.

### Informative Hinweise



#### Hinweis!

Dieses Symbol macht auf eine wichtige Information aufmerksam.



### Handlungsanweisung

1. Dieses Symbol markiert eine Handlungsanweisung. Sie werden zu einer Handlung oder Handlungsfolge aufgefordert.

## 1.5 Festlegungen

Das Darstellungsformat von Daten variiert in der nachfolgenden Dokumentation. Es werden nachfolgende Formate verwendet:

### ASCII

Beispiel: "A"; "B"; "1"; "2"

Ein ASCII-Zeichen entspricht einem Byte. Es können  $2^8 = 256$  unterschiedliche Zeichen dargestellt werden.

### Binär

Beispiel:  $1001_{\text{bin}}$

Binärzahlen sind durch ein  $_{\text{bin}}$  gekennzeichnet.

## DEZIMAL

Beispiel: 1234

Dezimalzahlen werden ohne zusätzliche Kennzeichnung dargestellt.

## HEX

Beispiel: 0x41; 0x42; 0x31; 0x32

Die hexadezimale Darstellung eines Bytes besteht aus zwei Ziffern (z. B. 0x41). Der Teilbereich auf der linken Seite ist das höherwertige Nibble (0x4). Auf der rechten Seite befindet sich das niederwertige Nibble (0x1). Der Wertebereich erstreckt sich zwischen 0x00 ... 0xFF.

### Hinweis!

Im TIA-Portal von Siemens werden hexadezimale Zahlen wie folgt dargestellt:

#### Beispiel:

16#00 (entspricht 0x00)

### Abkürzungen und Begriffe

COM-Mode	Übertragungsgeschwindigkeit der IO-Link-Daten
Datenträger	mobiler Datenspeicher mit Anwenderdaten und eindeutiger Nummer
Device ID	Identifikationsnummer des Gerätes
Easy Mode	Kommunikationsprotokoll für einen einfachen Datenzugriff einer Schreib-/Lesestation; kein Funktionsbaustein erforderlich
Expert Mode	Kommunikationsprotokoll für einen erweiterten Datenzugriff der RFID-Schreib-/Lesestation; die Verwendung eines Funktionsbausteins ist erforderlich
Fixcode	eindeutige und nicht veränderbare Nummer eines Datenträgers
IODD	IO Device Description; Datei mit Informationen zu IO-Link-Parametern eines IO-Link-fähigen Gerätes
IO-Link	Kommunikationssystem zur Anbindung intelligenter Sensoren und Aktoren über eine Punkt-zu-Punkt-Kommunikation
IO-Link-Master	Schnittstelle zur übergeordneten Steuerung; steuert die Kommunikation zu angeschlossenen IO-Link-Geräten (IO-Link Devices)
IO-Link-Gerät	Intelligenter Sensor oder Aktor zum Anschluss an einen IO-Link-Master; besitzt gerätespezifische IO-Link-Parameter
IO-Link-Parameter	gerätespezifische Informationen zu einem IO-Link-Gerät; Parameter sind in einer IODD hinterlegt; azyklische Änderung der Parameter
IO-Link-Protokoll	Version der unterstützten IO-Link-Kommunikation; V1.0 bzw. V1.1
IUC	Pepperl+Fuchs-spezifische Bezeichnung eines UHF-Datenträgers
ISO/IEC 18000-63	Standard zur Datenübertragung für ein 13,56 MHz-RFID-System
ISDU	Indexed Service Data Unit
PACTware	Parametriersoftware für den Zugriff auf IO-Link-Parameter
Port Type	Typ des IO-Link-Ports
RFID-Schreib-/Lesestation	RFID-Schreib-/Lesekopf mit integrierter IO-Link-Schnittstelle zur berührungslosen Datenübertragung
SIO-Mode	Standard IO-Mode; Modus zur konventionellen Signalübertragung ohne IO-Link-Daten
SPS	Speicherprogrammierbare Steuerung; Gerät zur Steuerung und Regelung einer Maschine bzw. Anlage
Vendor ID	Identifikationsnummer des Geräteherstellers; Pepperl+Fuchs: 0x01

<b>AFI</b>	<b>A</b> pplication <b>F</b> amily <b>I</b> dentifier
<b>CRC</b>	<b>C</b> yclic <b>R</b> edundancy <b>C</b> heck
<b>EIRP</b>	<b>E</b> quivalent <b>I</b> sotopically <b>R</b> adiated <b>P</b> ower
<b>EPC</b>	<b>E</b> lectronic <b>P</b> roduct <b>C</b> ode
<b>ERP</b>	<b>E</b> ffective <b>R</b> adiated <b>P</b> ower
<b>FCC</b>	<b>F</b> ederal <b>C</b> ommunications <b>C</b> ommission
<b>FHSS</b>	<b>F</b> requency <b>H</b> opping <b>S</b> pread <b>S</b> pectrum
<b>IC</b>	<b>I</b> ndustry <b>C</b> anada
<b>ISO</b>	<b>I</b> nternational <b>S</b> tandardisation <b>O</b> rganisation
<b>MDID</b>	<b>M</b> ask <b>D</b> esigner <b>I</b> dentifier
<b>PC</b>	<b>P</b> rotocol <b>C</b> ontrol
<b>RFID</b>	<b>R</b> adio <b>F</b> requenz <b>I</b> dentifikation
<b>SPS</b>	<b>S</b> peicherprogrammierbare <b>S</b> teuerung
<b>Tag</b>	Datenträger; Transponder
<b>TID</b>	<b>T</b> ransponder <b>I</b> dentification
<b>TMN</b>	<b>T</b> ransponder <b>M</b> odel <b>N</b> umber
<b>UHF</b>	<b>U</b> ltrahoch- <b>F</b> requenz
<b>UII</b>	<b>U</b> nique <b>I</b> tem <b>I</b> dentifier

## 2 Zertifikate und Zulassungen

### 2.1 Konformitätserklärung (RE Directive 2014/53/EU)

Dieses Produkt wurde unter Beachtung geltender europäischer Normen und Richtlinien entwickelt und gefertigt.



#### Hinweis!

Eine Konformitätserklärung kann beim Hersteller angefordert oder im Internet unter [www.pepperl-fuchs.com](http://www.pepperl-fuchs.com) heruntergeladen werden.

Der Hersteller des Produkts, die Pepperl+Fuchs SE in 68307 Mannheim, besitzt ein zertifiziertes Qualitätssicherungssystem gemäß ISO 9001.



ISO9001

### 2.2 FCC-Information

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

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

#### Attention:

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



#### Hinweis!

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.

## 2.3 IC-Information

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

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

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes :

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

## IC Exposure Information

To comply with IC RF exposure compliance requirements, the antennas used for this transmitter must be installed to provide a separation distance of at least 35 cm from all persons and must not be co-located or operated in conjunction with any other antenna or transmitter.

## 2.4 IFT-Information

This device complies with Federal Telecommunications Institute IFT standard(s). Operation is subject to the following two conditions:

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

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

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

## 2.5 NCC-Information

The company, firm or user shall not

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

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

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

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

## 2.6 ANATEL-Information

This equipment is not entitled to protection against harmful interference and may not cause interference in duly authorized systems. For more information, consult the ANATEL website [www.anatel.gov.br](http://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](http://www.anatel.gov.br)*

## 2.7 UL-Information

### Technical Data and Environmental Conditions

This device is for indoor use only.

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

The ambient temperature range is from -25 °C ... +70 °C for operation with non-transmission periods, or -25 °C ... +60 °C for continuous transmission mode. The Pollution degree is 2.

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

Nominal power supply voltage is 24 V<sub>DC</sub>, voltage range is 18 V ... 30 V<sub>DC</sub>. Supply must be LEC (Limited Energy Circuit), LPS (Limited Power Source) or CLASS 2. The Overvoltage Category II is applied.

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

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

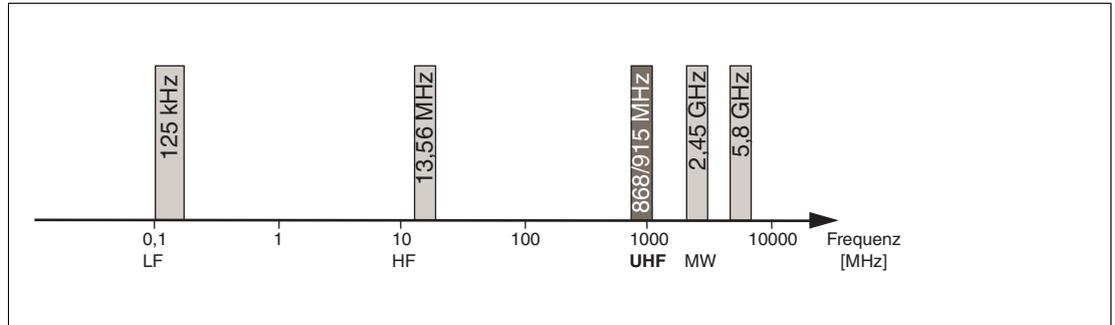
## 2.8 Weitere länderspezifische Zulassungen

Alle derzeit gültigen Zulassungen finden Sie auf dem Datenblatt Ihres Geräts unter [www.pepperl-fuchs.com](http://www.pepperl-fuchs.com).

## 3 Produktbeschreibung

### 3.1 RFID-Frequenzbänder

Die folgende Grafik zeigt Ihnen die Lage der unterschiedlichen Frequenzbänder, die für RFID verwendet werden. Die in diesem Handbuch beschriebenen Geräte arbeiten im Frequenzbereich von 865 MHz ... 868 MHz und 902 MHz ... 928 MHz, der hervorgehoben ist.



- 100 kHz ... 135 kHz: Niederfrequenz LF
- 13,56 MHz: Hochfrequenz HF
- 865 MHz ... 868 MHz (Europa), 902 MHz ... 928 MHz (USA), 920 MHz ... 925 MHz (China): Ultrahochfrequenz UHF
- 2,45 GHz und 5,8 GHz: Mikrowelle MW

### 3.2 UHF Allgemein

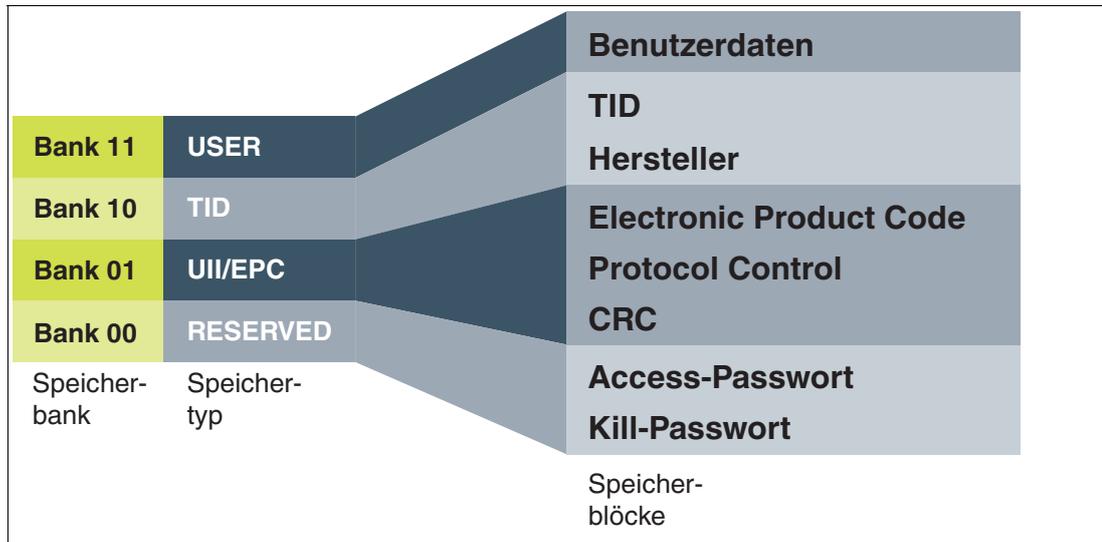
#### 3.2.1 Vorteile von UHF

- Große Reichweite
- UHF Transponder sind als günstige und platzsparende Klebeetiketten verfügbar
- Hohe Datenübertragungsraten
- Transponder mit großem Arbeitsspeicher (User Memory) verfügbar
- Pulkerfassung

#### 3.2.2 Anwendungen für UHF-Systeme

- Identifikation in galvanischen Beschichtungs- oder Lackieranlagen in der Automobilproduktion,
- Identifikation über größere Entfernungen als mit LF- und HF-Systemen realisierbar,
- Identifizierung von Fahrzeugaufbauten in der Automobilproduktion,
- Palettenidentifikation und Erfassung von Warenströmen in der Logistik und
- Zugangssteuerung bei Verladestationen durch LKW-Identifikation.

### 3.2.3 Speicherstruktur eines Transponders nach EPC Gen 2 (ISO/IEC 18000-63)



Das Speichermodul eines Transponders des Typs EPC Gen 2 (ISO/IEC 18000-63) ist in 4 Segmente unterteilt. Die wesentlichen Inhalte dieser Segmente sind:

Segment	Funktion	Länge
Bank 00	Passwortverwaltung	abhängig vom Transpondertyp
Bank 01	Unique Item Identifier (UII) Electronic Product Code (EPC)	abhängig vom Transpondertyp
Bank 10	Tag-ID (TID)	4 Byte (MDID, TMN) + 0 Byte, 4 Byte oder 8 Byte
Bank 11	User Memory	abhängig vom Transpondertyp

#### Bank 00: Passwortverwaltung

Das Segment **Bank 00** enthält die Passwortverwaltung und ist nur bei der Verwendung des **ExpertMode** erreichbar. Es beinhaltet das Access- und das Kill-Passwort. Um auf die Speicherbank 00 zugreifen zu können, ist zunächst der Parameter MB (Memory Bank) auf den Wert 16#00 abzuändern (siehe "Memory Bank MB" auf Seite 128). Für den Zugriff auf die Speicherbank 00 können folgende Befehle verwendet werden:

- Single / Enhanced Read 4-Byte Blocks (siehe "Single Read 4-Byte Blocks (SR)" auf Seite 72 und siehe "Enhanced Read 4-Byte Blocks (ER)" auf Seite 74)
- Single / Enhanced Write 4-Byte Blocks (siehe "Single Write 4-Byte Blocks (SW)" auf Seite 77 und siehe "Enhanced Write 4-Byte Blocks (EW)" auf Seite 80)
- Single / Enhanced Read 2-Byte Words (siehe "Single Read 2-Byte Words (#SR)" auf Seite 82 und siehe "Enhanced Read 2-Byte Words (#ER)" auf Seite 85)
- Single / Enhanced Write 2-Byte Words (siehe "Single Write 2-Byte Words (#SW)" auf Seite 88 und siehe "Enhanced Write 2-Byte Words (#EW)" auf Seite 90)

#### Bank 01: EPC/UII

Das Segment **Bank 01** enthält den EPC/UII. Das Segment ist in mehrere Teilbereiche gegliedert:

- CRC (Cyclic Redundancy Check; Prüfsumme) 16 Bit
- PC (Protocol Control; PC-Wort) 16 Bit
- EPC (Electronic Product Code) bzw. UII (Unique Item Identifier) Länge variabel

Die Bytes 0 und 1 der Speicherbank enthalten eine Prüfsumme (CRC). Diese wird durch den Datenträger selbstständig berechnet. Dieser Teilbereich kann nur ausgelesen werden.

In den Bytes 2 und 3 befindet sich das PC-Wort (Protocol Control). Darin sind folgende Informationen enthalten:

- Länge des EPC/UII
- Application Family Identifier (AFI)
- Togglebit zur Anzeige, ob es sich um einen EPC oder einen Code nach ISO-Standard handelt
- Bit zur Anzeige, ob innerhalb von Segment 11 (User Memory) Daten hinterlegt sind

Für eine detaillierte Übersicht über den Speicheraufbau siehe "Set Filter Mask (FI)" auf Seite 101.

Das Gerät kann mehrere Datenträger innerhalb der Erfassungszone identifizieren, wenn das Long-Form-Datenformat verwendet wird. Es muss dabei aber gewährleistet sein, dass sich nur Transponder mit unterschiedlichen EPC/UII-Codes in der Erfassungszone befinden. Wenn das System einen weiteren Transponder mit identischen EPC/UII-Code erkennt, wird eine Fehlermeldung generiert.

Ein erfolgreicher Lese- oder Schreibzugriff auf einen Transponder wird durch ein Antworttelegramm signalisiert. Innerhalb dieses Antworttelegramms befindet sich immer die EPC/UII-Information bestehend aus PC-Wort und EPC/UII. Der Bereich der CRC wird nicht mit übertragen. Die EPC/UII-Information ist bei einer Rückantwort auf einen Lesezugriff immer den eingelesenen Daten (User Memory oder TID) vorangestellt, wenn das Long-Form-Datenformat verwendet wird.

Der Zugriff auf die Speicherbank 01 ist sowohl im Easy- als auch im ExpertMode möglich. Bei beiden Modi können Informationen ausgelesen und geschrieben werden. Der Zugriff auf die Speicherbank 01 erfolgt im **ExpertMode** über folgende Befehle:

- Single / Enhanced Read EPC/UII (siehe "Single Read EPC/UII (SN)" auf Seite 64 und siehe "Enhanced Read EPC/UII (EN)" auf Seite 66)
- Single Write EPC/UII (siehe "Single Write EPC/UII (#SU)" auf Seite 68)



#### Hinweis!

Da die Gefahr besteht, mehrere Transponder mit identischen EPC/UII-Information zu programmieren, ist die Ausführung eines permanenten Schreibbefehls (Enhanced Write EPC/UII) nicht möglich.

Über den Parameter MB (Memory Bank) können Sie eine weitere Möglichkeit für den Zugriff auf die Speicherbank 01 steuern. Wenn der Parameter MB auf den Wert 16#01 verändert wird, so kann über folgende Befehle auf die Speicherbank 01 zugegriffen werden:

- Single / Enhanced Read 4-Byte Blocks (siehe "Single Read 4-Byte Blocks (SR)" auf Seite 72 und siehe "Enhanced Read 4-Byte Blocks (ER)" auf Seite 74)
- Single / Enhanced Write 4-Byte Blocks (siehe "Single Write 4-Byte Blocks (SW)" auf Seite 77 und siehe "Enhanced Write 4-Byte Blocks (EW)" auf Seite 80)
- Single / Enhanced Read 2-Byte Words (siehe "Single Read 2-Byte Words (#SR)" auf Seite 82 und siehe "Enhanced Read 2-Byte Words (#ER)" auf Seite 85)
- Single / Enhanced Write 2-Byte Words (siehe "Single Write 2-Byte Words (#SW)" auf Seite 88 und siehe "Enhanced Write 2-Byte Words (#EW)" auf Seite 90)

Durch diese Variante besteht die Möglichkeit auf die Speicherbank 01 komplett, inklusive CRC, oder nur einen Teilbereich zuzugreifen. Beispielsweise können auf diesem Wege mehrere nacheinander folgende Bytes des EPC/UII ausgelesen werden.

## Bank 10: TID

Das Segment **Bank 10** beinhaltet den Tag-Identifizierer (TID). Die TID besteht immer aus einer "Short Tag Identification" (Short-TID) sowie optional zusätzlich aus einer "Extended Tag Identification" (XTID). Die Short-TID hat eine Länge von 4 Byte (32 Bit) und ist wie folgt aufgebaut:

- Class Identifier (Klasse des Transponders; 16#E2) 8 Bit
- Extended Tag Identification Indicator 1 Bit
- Security Indicator und File Indicator je 1 Bit
- Mask Designer Identifier (MDID) 9 Bit
- Tag Model Number (TMN) 12 Bit

Wenn innerhalb der Short-TID das Bit für den Extended Tag Identification Indicator gesetzt ist, folgen der Short-TID weitere Bytes mit zusätzlichen Informationen über die Eigenschaften des Datenträgers. Die XTID hat eine Länge von 8 Byte (64 Bit) und folgende Struktur:

- XTID 16 Bit
- Serial Number 48 Bit

Für eine detaillierte Übersicht über den Speicheraufbau der TID siehe "Set Filter Mask (FI)" auf Seite 101. Zusätzliche Informationen über die TID können der Norm ISO/IEC 18000-63 entnommen werden.

Die Speicherbank 10 kann nur ausgelesen werden. Der Zugriff auf die TID ist bei der Verwendung des **EasyMode** und des **ExpertMode** möglich. In beiden Fällen ist in der Rückantwort des Geräts zusätzlich zur eingelesenen TID der zugehörige EPC/UII vorangestellt, wenn das Long-Form-Datenformat verwendet wird. Hierdurch ist die Zuordnung des Transponders gewährleistet.

Der Zugriff auf die Speicherbank 10 erfolgt im **ExpertMode** über folgende Befehle:

- Single / Enhanced Read Fixcode (siehe "Single Read Fixcode (SF)" auf Seite 59 und siehe "Enhanced Read Fixcode (EF)" auf Seite 62)

Eine weitere Möglichkeit für den Zugriff auf die Speicherbank 10 kann über den Parameter Memory Bank MB eingestellt werden. Wenn der Parameter MB auf den Wert 16#02 verändert wird, kann über folgende Befehle auf die Speicherbank 10 zugegriffen werden:

- Single / Enhanced Read 4-Byte Blocks (siehe "Single Read 4-Byte Blocks (SR)" auf Seite 72 und siehe "Enhanced Read 4-Byte Blocks (ER)" auf Seite 74)
- Single / Enhanced Read 2-Byte Words (siehe "Single Read 2-Byte Words (#SR)" auf Seite 82 und siehe "Enhanced Read 2-Byte Words (#ER)" auf Seite 85)

Durch diese Variante besteht die Möglichkeit die Speicherbank 10 komplett oder nur einen Teilbereich auszulesen.

## Bank 11: User Memory

Das Segment **Bank 11** beinhaltet einen Bereich, über den der Benutzer frei verfügen kann. Dieser Bereich hat je nach Chip-Typ verschiedene Größen oder ist nicht vorhanden.

Innerhalb des User Memory befinden sich die Nutzdaten des Transponders. Dieser Bereich kann dafür genutzt werden, anwendungsspezifische Daten auf den Transponder zu hinterlegen. Bei der Verwendung von Chips basierend auf EEPROM-Technologie ist die Anzahl der Schreibzugriffe ggf. begrenzt. Eine nahezu unbegrenzte Anzahl an Schreibzugriffen wird bei der Verwendung von Datenträgern mit FRAM-Speicher erreicht.

Der Zugriff auf den User Memory ist bei der Nutzung des **EasyMode** und des **ExpertMode** möglich. Bei beiden Varianten können die Nutzdaten gelesen und geschrieben werden. Bei einem erfolgreichen Lese- oder Schreibzugriff auf einen Datenträger enthält die Rückantwort immer den EPC/UII des Datenträgers, wenn das Long-Form-Datenformat verwendet wird. Hierdurch wird die Zuordnung der eingelesenen Daten bzw. der beschriebenen Daten zu einem Transponder gewährleistet. Bei einem Lesezugriff ist der EPC/UII in der Rückantwort immer den eingelesenen Teilbereich des User Memory vorangestellt.

Für den Zugriff auf die Speicherbank 11 können innerhalb des **ExpertMode** folgende Befehle verwendet werden:

- Single / Enhanced Read 4-Byte Blocks (siehe "Single Read 4-Byte Blocks (SR)" auf Seite 72 und siehe "Enhanced Read 4-Byte Blocks (ER)" auf Seite 74)
- Single / Enhanced Write 4-Byte Blocks (siehe "Single Write 4-Byte Blocks (SW)" auf Seite 77 und siehe "Enhanced Write 4-Byte Blocks (EW)" auf Seite 80)
- Single / Enhanced Read 2-Byte Words (siehe "Single Read 2-Byte Words (#SR)" auf Seite 82 und siehe "Enhanced Read 2-Byte Words (#ER)" auf Seite 85)
- Single / Enhanced Write 2-Byte Words (siehe "Single Write 2-Byte Words (#SW)" auf Seite 88 und siehe "Enhanced Write 2-Byte Words (#EW)" auf Seite 90)

Der Parameter MB muss hierzu auf dem Wert 16#03 (Werkseinstellung) eingestellt sein.

### 3.2.4 Elektronischer Produkt-Code EPC

Der elektronische Produkt-Code EPC ist eine eindeutige Kennzeichnung in Form einer Ziffernfolge. Diese Ziffernfolge besitzt eine festgelegte Struktur und hat eine Länge von 64 Bit, 80 Bit, 96 Bit oder länger, abhängig vom eingesetzten EPC. Diese Ziffernfolge wird auf dem RFID-Transponder gespeichert und identifiziert somit das mit dem Transponder versehene Objekt weltweit eindeutig.

Für die Anwendung in der Warenwirtschaft wurde von GS1 das System der EPC festgelegt. Transponder mit Speicher für EPCs sind vom Anwender zu programmieren. Die Speicher neuer Transponder müssen keine gültigen EPC enthalten. Die EPC werden von GS1 verwaltet und vergeben. Um einen EPC zu erhalten, wenden Sie sich bitte an die jeweilige GS1-Niederlassung in Ihrem Land ([www.gs1.com/contact](http://www.gs1.com/contact)).

Der EPC ist von GS1 in derzeit 13 verschiedenen Kodierungen definiert. Als Beispiel für eine häufig verwendete Kodierung ist hier die SGTIN-96 (serialized global trade item number) dargestellt. Die SGTIN-96 besitzt einen festgelegten Aufbau und ist folgendermaßen strukturiert:

1. **Header:** der Header gibt den verwendeten EPC-Standard vor und beschreibt die Ziffernfolge.
2. **Filterwert:** beschreibt die Einheit des Produkts, z. B. Endprodukt, Umverpackung, Palette.
3. **Partition:** beschreibt, an welcher Stelle der folgende Company Prefix endet und die Objektdaten beginnen.
4. **Company Prefix:** zugeteilte Nummernfolge, die den Produzenten identifiziert.
5. **Objektklasse:** Nummernfolge, die das Objekt beschreibt, z. B. Artikelnummer.  
Die Länge des Company Prefix und der Objektklasse ist jeweils variabel, zusammen aber immer 44 Bit lang.
6. **Seriennummer:** Nummernfolge, die den Artikel identifiziert, z. B. fortlaufende Seriennummer des Artikels.

	Header	Filterwert	Partition	Company Prefix	Objektklasse	Seriennummer
Länge [Bit]	8	3	3	20 ... 40	4 ... 24	38
Wert	48 <sub>dez</sub>	0 <sub>dez</sub>	5 <sub>dez</sub>	4050143 <sub>dez</sub>	124 <sub>dez</sub>	203886 <sub>dez</sub>

Tabelle 3.1

### 3.2.5 Einfluss verschiedener Materialien auf die Reichweite

Im UHF-Bereich hat die Beschaffenheit der Umgebung und des Untergrunds, auf dem der Transponder befestigt ist, eine gravierende Auswirkung auf die zu erzielende Reichweite des Systems. So ist eine Befestigung des UHF-Transponders auf Metall ohne Anpassungen nicht möglich. Glas hat z. B. als Untergrund einen negativen Einfluss auf die Reichweite. Wenn ein UHF-Transponder auf feuchten Materialien befestigt wird, ist die Reichweite im Vergleich zu trockenem Material jeweils schlechter. Der Montageuntergrund hat oftmals einen größeren Einfluss auf die Lesereichweite als das Material zwischen dem Transponder und dem Schreib-/Lesegerät. In der Grafik können Sie die Auswirkungen verschiedener Materialien tendenziell erkennen.

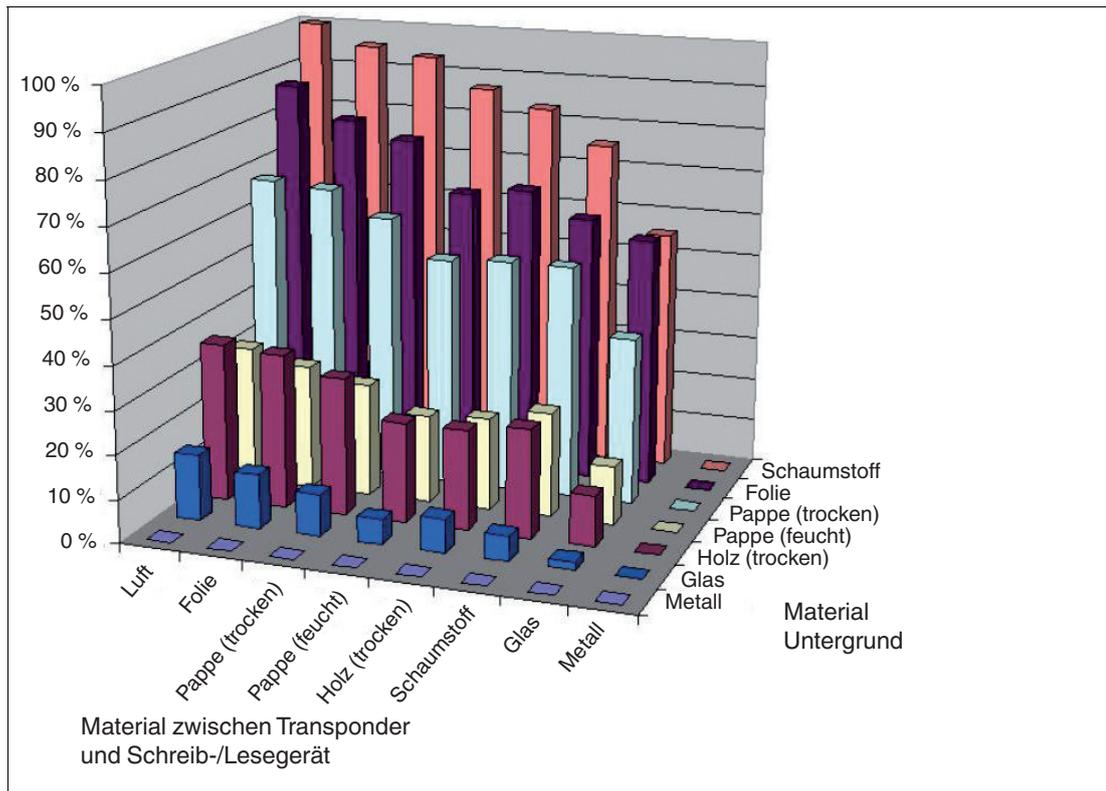


Abbildung 3.1

Für die Montage auf Metall oder einen elektrisch leitfähigen Untergrund gibt es entsprechend optimierte Transponder. Diese können ohne zusätzlichen Abstand auf Metall befestigt werden. Diese Transponder enthalten ein "-M-" in der Produktbezeichnung, z. B. IUC77-F151-M-GBL. Wenn diese Transponder auf Kunststoff bzw. elektrisch nichtleitfähige Materialien befestigt werden, kann sich die erzielbare Erfassungsreichweite, im Vergleich zu der Montage auf Metall, verringern.

Wenn der Datenträger auf Kunststoff bzw. elektrisch nichtleitfähige Materialien befestigt wird, ist in der Produktbezeichnung das Kürzel "-M-" nicht enthalten z. B. IUC77-50-FR1. Diese Datenträger können beispielsweise auch auf Pappe oder Ähnliches befestigt werden.

Prinzipiell gibt es immer eine Abhängigkeit der Erfassungsreichweite vom Montageuntergrund. Deshalb ist eine pauschale Aussage über die Reichweite eines Transponders nicht oder nur unter Einschränkungen möglich. Es ist somit erforderlich, bei jedem Installationspunkt einem Schreib-/Lesegerät die mit den aktuell eingestellten Parametern erzielte Reichweite zu überprüfen. Über die Sendeleistung kann Einfluss auf die Erfassungsreichweite dem Schreib-/Lesegerät genommen werden. Die Sendeleistung sollte so eingestellt werden, dass mit einer möglichst geringen Leistungsstufe der Datenträger sicher gelesen werden kann. Eine Erhöhung der Sendeleistung kann zu einer unbeabsichtigten Identifikation von benachbarten Datenträgern führen. Die unbeabsichtigte Identifikation von Datenträgern in der weiteren Umgebung dem Schreib-/Lesegerät sollte ebenfalls überprüft werden.

### 3.2.6 Dense Reader Mode (DRM)

#### Europa

Ein spezieller Betriebsmodus für Datenträger nach der Spezifikation EPC Gen 2 (ISO/IEC 18000-63) bewirkt, dass mehrere eng benachbarte Schreib-/Lesegeräte gleichzeitig störungsfrei betrieben werden können.

Gemäß EN 302208 werden beim Schreib-/Lesegerät in diesem Modus nur die Kanäle 4, 7, 10 und 13 zum Senden (Kommunikationspfad Schreib-/Lesegerät → Datenträger) verwendet. Die Sendeleistung beträgt maximal  $2 W_{erp}$  gemäß EN 302208.

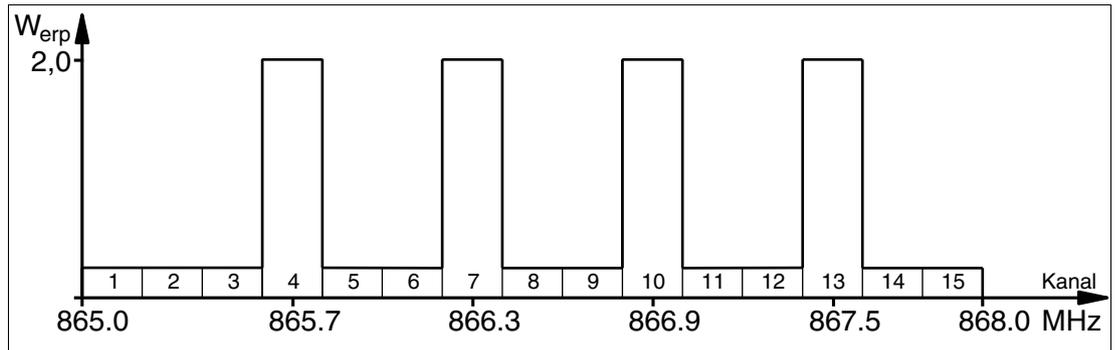


Abbildung 3.2

Die Antwort des Datenträgers erscheint durch den Frequenzversatz, der durch die in diesem Modus verwendete Modulation erzielt wird, auf den beiden Nachbarkanälen. Aufgrund des großen Pegelunterschiedes zwischen den Sendekanälen und Antwortkanälen bringt diese Technik bei der Wiederverwendung von Frequenzen große Vorteile.

### 3.2.7 Frequenzsprungverfahren

Beim Frequenzsprungverfahren FHSS (**F**requency **H**opping **S**pread **S**pectrum) wird die zu übertragende Information nacheinander auf mehrere Kanäle verteilt. Zu jedem Zeitpunkt wird immer nur ein Frequenzkanal genutzt. Dadurch ergibt sich für das Gesamtsignal eine größere Bandbreite, obwohl jeder Kanal eine kleinere Bandbreite besitzt. In diesem Abschnitt sind exemplarisch die Kanalbelegung für China und die USA grafisch dargestellt. Für beide Belegungen gelten unterschiedliche Parameter, wie z. B. Kanalanzahl und Kanalbandbreite. Darüber hinaus gibt es für weitere Länder andere Parametersätze.

#### 3.2.7.1 China

In China steht für UHF-RFID-Schreib-/Lesegeräte der Frequenzbereich 920 ... 925 MHz zur Verfügung. Der Bereich ist in Kanäle mit je 250 kHz Bandbreite eingeteilt. Auf 16 der zur Verfügung stehenden Kanäle sind maximal 2  $W_{erp}$  erlaubt. Die Sendeleistung wird in  $W_{erp}$  angegeben. Es wird FHSS mit maximal 2 Sekunden Verweilzeit eingesetzt. Das UHF-RFID-Schreib-/Lesegerät für China verwenden optional die Kanäle 2 bis 17.

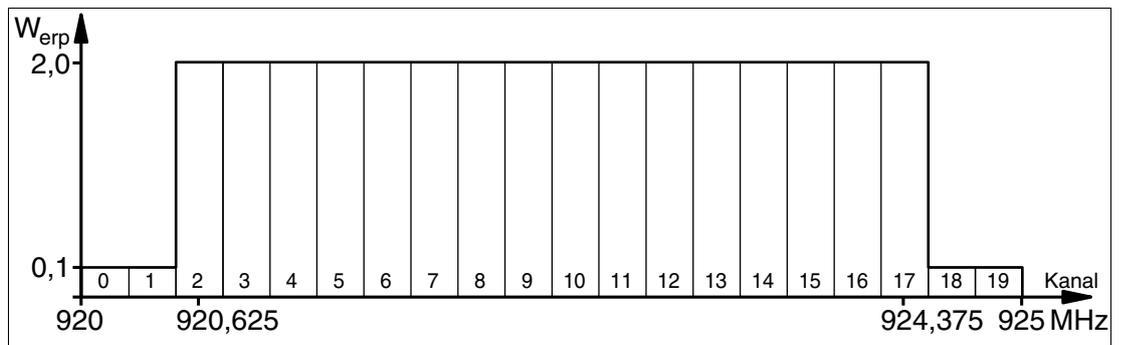


Abbildung 3.3

#### 3.2.7.2 USA

In den USA steht das ISM-Band von 902 bis 928 MHz zur Verfügung. Das Band ist in 50 Kanäle mit jeweils 500 kHz Bandbreite eingeteilt. Es wird FHSS mit einer maximalen Verweildauer von 0,4 Sekunden eingesetzt. Dabei müssen alle Kanäle genutzt werden. Eine Einschränkung der Kanäle ist nicht zulässig.

Die Sendeleistung wird im Gegensatz zu den Schreib-/Lesegeräten für Europa und China in  $W_{eirp}$  angegeben. Auf allen Kanälen sind maximal 4  $W_{eirp}$  erlaubt.

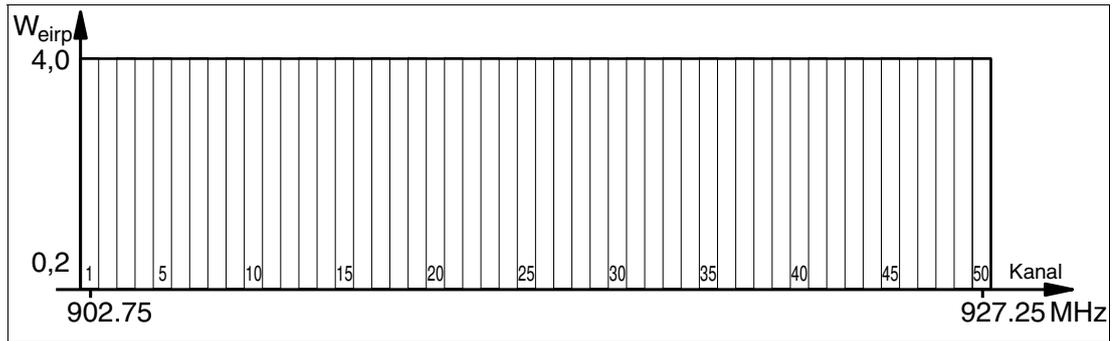


Abbildung 3.4

### 3.2.8 Relevante Normen zu UHF

Europäische Funknormen: EN 300220 und EN 302208

Einsatzempfehlungen für RFID-Label, Hinweise zur Recyclingfähigkeit, Installation von Readern und Antennen: ISO/IEC TR 24729 Teile 1-4

Installation und Inbetriebnahme von UHF-RFID-Systemen: ETSI TR 102436

Luftschnittstellenbeschreibung: EPC Gen 2 (ISO/IEC 18000-63)

## 3.3 Einsatzländer



### Hinweis!

#### Funkzulassung

Für den Betrieb dieses Geräts ist eine länderspezifische Funkzulassung erforderlich. Für die Europäische Union genügt die Konformitätserklärung des Herstellers. Alle derzeit gültigen Funkzulassungen finden Sie auf dem Datenblatt des jeweiligen Geräts unter [www.pepperl-fuchs.com](http://www.pepperl-fuchs.com).



### Hinweis!

#### Länderkennung

Alle Geräte werden innerhalb der jeweiligen gesetzlichen Vorgaben betrieben. Die länderspezifischen Einstellungen werden während der Produktion eingestellt und können danach nicht mehr verändert werden.



### Hinweis!

Falls Sie das Gerät in einem Land nutzen möchten, das nicht in diesem Kapitel aufgeführt ist, vergewissern Sie sich vorher, dass die relevanten Werte Ihres Geräts mit den örtlichen Bestimmungen übereinstimmen.

Das verwendete Frequenz-Zugriffsverfahren ist Teil der länderspezifischen Einstellungen.

#### Frequenz-Zugriffsverfahren

- In vielen Ländern, z. B. den USA und China, wird ein Frequenzsprungverfahren verwendet. Siehe Kapitel 3.2.7. Die Anzahl und Lage der Frequenzen ist fest vorgegeben und kann durch den Benutzer nicht verändert werden. Es werden alle Kanäle verwendet.
- In anderen Ländern, u. a. der Europäischen Union, Singapur oder Indien, wird eine parametrierbare Frequenzliste verwendet. Diese Frequenzliste können Sie aus vorgegebenen Kanälen zusammenstellen. In der Europäischen Union sind entsprechend des Dense Reader Modes gemäß EN 302208 vier Kanäle vorgegeben. Siehe Kapitel 3.2.6. Bei dieser Einstellung haben Sie die Möglichkeit, einen, mehrere oder alle vier Kanäle zu konfigurieren.

### 3.3.1 Europäische Union

In der Europäischen Union ist die Verwendung von RFID im UHF-Bereich durch die EN 302208 geregelt.

- UHF-Band: 865...868 MHz
- Strahlungsleistung: 3 ... 100 mW<sub>erp</sub>; Default = 100 mW<sub>erp</sub>
- Kanalbandbreite: 200 kHz
- Kanalabstand 600 kHz
- Frequenzzugriffsverfahren: parametrierbare Frequenzliste
- Vordefinierte Anzahl Kanäle: 4  
Einstellbare Kanäle: 4, 7, 10, 13  
Mittenfrequenzen: 865,7 MHz, 866,3 MHz, 866,9 MHz, 867,5 MHz  
Bis zu 4 Kanäle können parametrierbar und nacheinander verwendet werden.  
Default: Dense Reader Mode mit Kanal 4, 7, 10, 13. Siehe Kapitel 3.2.6.

### 3.3.2 Australien

In Australien ist die Verwendung von RFID im UHF-Bereich wie folgt geregelt:

- UHF-Band: 920...926 MHz
- Strahlungsleistung: 3 ... 150 mW<sub>eirp</sub>; Default = 150 mW<sub>eirp</sub>
- Kanalbandbreite: 500 kHz
- Kanalabstand: 500 kHz
- Frequenzzugriffsverfahren: Frequency Hopping. Siehe Kapitel 3.2.7.
- Anzahl Kanäle: 12  
Verwendete Kanäle: 1, 2, 3, ... 12  
Mittenfrequenzen: 919,75 MHz + (M x 0,5) MHz  
Es werden immer alle 12 Kanäle verwendet.

### 3.3.3 Brasilien

In Brasilien ist die Verwendung von RFID im UHF-Bereich wie folgt geregelt:

- UHF-Band: 915...928 MHz
- Strahlungsleistung: 3 ... 150 mW<sub>eirp</sub>; Default = 150 mW<sub>eirp</sub>
- Kanalbandbreite: 250 kHz
- Kanalabstand: 250 kHz
- Frequenzzugriffsverfahren: Frequency Hopping. Siehe Kapitel 3.2.7.
- Anzahl Kanäle: 52  
Verwendete Kanäle: 1, 2, 3, ... 52  
Mittenfrequenzen: 914,875 MHz + (M x 0,25) MHz  
Es werden immer alle 52 Kanäle verwendet.

### 3.3.4 China

In China ist die Verwendung von RFID im UHF-Bereich durch die Bestimmungen des China Ministry of Industry and Information Technology (CMIIT) geregelt.

- UHF-Band: 920...925 MHz
- Strahlungsleistung: 3 ... 100 mW<sub>erp</sub>; Default = 100 mW<sub>erp</sub>
- Kanalbandbreite: 250 kHz
- Kanalabstand: 250 kHz
- Frequenzzugriffsverfahren: Frequency Hopping (China). Siehe Kapitel 3.2.7.
- Anzahl Kanäle: 16  
Verwendete Kanäle: 2, 3, 4, ... 17  
Mittenfrequenzen: 920,125 MHz + (M x 0,25) MHz  
Es werden immer alle 16 Kanäle verwendet.

### 3.3.5 Indien

In Indien ist die Verwendung von RFID im UHF-Bereich in Anlehnung an die EN 302208 geregelt.

- UHF-Band: 865...867 MHz
- Strahlungsleistung: 3 ... 100 mW<sub>erp</sub>; Default = 100 mW<sub>erp</sub>
- Kanalbandbreite: 200 kHz
- Kanalabstand: 200 kHz
- Frequenzzugriffsverfahren: parametrierbare Frequenzliste
- Anzahl der vordefinierten Kanäle: 10  
Einstellbare Kanäle: 1, 2, 3, ...10  
Mittenfrequenzen: 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  
Bis zu 10 Kanäle können gleichzeitig parametriert und nacheinander verwendet werden.  
Default: Dense Reader Mode mit Kanal 1, 4, 7, 10. Siehe Kapitel 3.2.6.

### 3.3.6 Kanada

Die Bestimmungen für den UHF-Frequenzbereich in Kanada entsprechen den Bestimmungen für den UHF-Frequenzbereich in den USA. Siehe Kapitel 3.3.11.

### 3.3.7 Malaysia

In Malaysia ist die Verwendung von RFID im UHF-Bereich wie folgt geregelt.

- UHF-Band: 919 ... 923 MHz
- Strahlungsleistung: 3 ... 100 mW<sub>erp</sub>; Default = 100 mW<sub>erp</sub>
- Kanalbandbreite: 500 kHz
- Kanalabstand 500 kHz
- Frequenzzugriffsverfahren: Frequency Hopping.siehe Kapitel 3.2.7.
- Anzahl Kanäle: 8  
Verwendete Kanäle: 1, 2, 3, ... 8  
Mittenfrequenzen: 918,75 MHz + (M x 0,5) MHz  
Es werden immer alle 8 Kanäle verwendet.

### 3.3.8 Mexiko

Die Bestimmungen für den UHF-Frequenzbereich in Mexiko entsprechen den Bestimmungen für den UHF-Frequenzbereich in den USA. Siehe Kapitel 3.3.11.

### 3.3.9 Südkorea

In Südkorea ist die Verwendung von RFID im UHF-Bereich wie folgt geregelt:

- UHF-Band: 917,2...920,4 MHz
- Strahlungsleistung: 3 ... 150 mW<sub>eirp</sub>; Default = 150 mW<sub>eirp</sub>
- Kanalbandbreite: 200 kHz
- Kanalabstand: 600 kHz
- Frequenzzugriffsverfahren: Frequency Hopping. Siehe Kapitel 3.2.7.
- Anzahl Kanäle: 6  
Verwendete Kanäle: 1, 4, 7, 10, 13, 16  
Mittenfrequenzen: 917,1 MHz + (M x 0,20) MHz  
Es werden immer alle 6 Kanäle verwendet.

### 3.3.10 Taiwan

In Taiwan ist die Verwendung von RFID im UHF-Bereich wie folgt geregelt:

- UHF-Band: 920...928 MHz
- Strahlungsleistung: 3 ... 100 mW<sub>erp</sub>; Default = 100 mW<sub>erp</sub>
- Kanalbandbreite: 500 kHz
- Kanalabstand: 500 kHz
- Frequenzzugriffsverfahren: Frequency Hopping. Siehe Kapitel 3.2.7.
- Anzahl Kanäle: 14  
Verwendete Kanäle: 1, 2, 3, ... 14  
Mittenfrequenzen: 920,25 MHz + (M x 0,5) MHz  
Es werden immer alle 14 Kanäle verwendet.

### 3.3.11 Vereinigte Staaten von Amerika

In den USA ist die Verwendung von RFID im UHF-Bereich durch die Bestimmungen der Federal Communications Commission (FCC) geregelt.

- UHF-Band: 902...928 MHz
- Strahlungsleistung: 3...150 mW<sub>eirp</sub>; Default = 150 mW<sub>eirp</sub>
- Kanalbandbreite: 500 kHz
- Kanalabstand: 500 kHz
- Frequenzzugriffsverfahren: Frequency Hopping (USA). Siehe Kapitel 3.2.7.
- Anzahl Kanäle: 50  
Verwendete Kanäle: 1, 2, 3, ... 50  
Mittenfrequenzen: 902,25 MHz + (M x 0,5) MHz  
Es werden immer alle 50 Kanäle verwendet.

### 3.4 Allgemeine Funktionen und Merkmale



Abbildung 3.5

#### Funktionen

Die Geräte IUT-F191-IO-V1-FR\* wurden für das Schreiben und Lesen von passiven Datenträgern mit einer Betriebsfrequenz im UHF-Bereich entwickelt. Dabei können bis zu 20 Datenträger innerhalb der Erfassungszone identifiziert werden, sofern die Speicherbank 01 (EPC/UII) der Transponder unterschiedlich ist.

Die RFID-Schreib-/Lesestationen werden über einen integrierten M12-Stecker an einen IO-Link-Master angeschlossen

Es werden Transponder gemäß EPC Gen 2 (IO/IEC 18000-63) unterstützt.

#### Erfassungsbereich

Die Schreib-/Lesestation ist klassifiziert mit einer Erfassungsreichweite von einem Meter. Die Reichweite der Geräte ist abhängig von den zu identifizierenden Transpondern, der eingestellten Sendeleistung und den Umgebungsbedingungen und kann dadurch größer oder kleiner als ein Meter sein.

#### Maximaler Frequenzbereich

Die Schreib-/Lesestationen IUT-F191-IO-V1-FR1\* arbeiten im Frequenzbereich von 865 MHz ... 868 MHz. Die Schreib-/Lesestationen IUT-F191-IO-V1-FR2\* arbeiten im Frequenzbereich von 902 MHz ... 928 MHz.

## Merkmale

Die Schreib-/Lesestation verfügt über folgende Merkmale:

- 3 gut sichtbare LEDs zur Funktionsanzeige auf der Gerätevorderseite
- industrietaugliches Gehäuse in kleiner Bauform
- Pulkerfassung
- Anschluss über Steckverbinder V1 (M12 x 1) an den IO-Link-Master
- geschützt gegen elektrostatische Entladung
- einstellbare UHF-Parametrierung

## Integrierte Antenne

Die Schreib-/Lesestationen besitzen eine zirkular polarisierte Antenne. Diese Schreib-/Lesestationen können zirkular polarisierte Wellen aussenden und empfangen.

## 3.5 Anzeigen und Bedienelemente

### LED

Das Gerät besitzt 3 LEDs auf der Gerätevorderseite in den Farben grün/rot, blau und gelb. Die LEDs zeigen den Gerätezustand und die Aktivität auf der Luftschnittstelle an. Die verschiedenen Anzeigen bedeuten:

- LED grün: grün leuchtend: Betriebsbereitschaft/keine IO-Link-Kommunikation  
grün blinkend: IO-Link-Kommunikation
- LED blau: Sendebetrieb
- LED gelb: Schreib-/Leseoperation erfolgreich

## 3.6 Elektrischer Anschluss



### Vorsicht!

Kabelspezifikationen

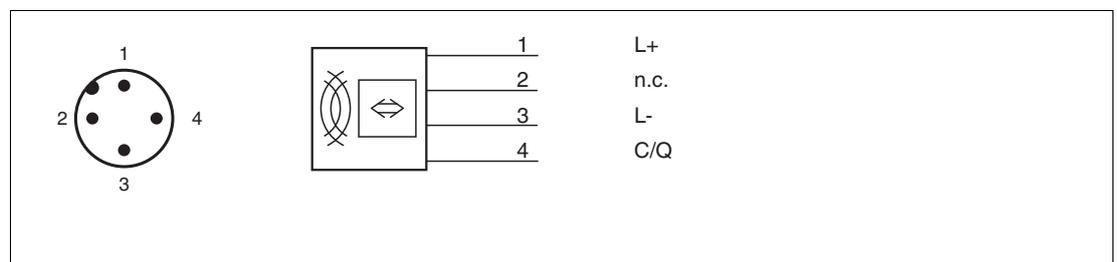
Die zulässige Höchsttemperatur des Anschlusskabels muss mindestens +90 °C betragen.

Der Mindestdurchmesser des Anschlusskabels muss 22 AWG oder 0,34 mm<sup>2</sup> betragen.

Anschlusskabel mit cULus / cURus Zulassung (PVVA/7, PVVA2/8, CYJV/7, CYJV2/8) müssen verwendet werden.

Die Bemessungsspannung des Anschlusskabels muss mindestens 60V DC betragen.

Der Bemessungsstrom des Anschlusskabels muss mindestens 1A (per pin/wire) betragen.



Pin 1	L+	+24 V
Pin 2	n.c.	nicht verbunden
Pin 3	L-	0 V/GND
Pin 4	C/Q	C/Q

Das RFID-Schreib-/Lesegerät wird über eine Punkt-zu-Punkt-Verbindung an einen IO-Link-Master angeschlossen. Gemäß der IO-Link-Installationsvorschrift darf die Länge der Anschlussleitung 20 Meter nicht überschreiten. Das RFID-Schreib-/Lesegerät wird durch den IO-Link-Master mit Energie versorgt. Technische Details siehe Produktdatenblatt.

### 3.7 IO-Link Schnittstelleneigenschaften

IO-Link Protokoll:	V1.1
COM Mode:	COM 3
Minimale Zykluszeit:	4 ms
Prozessdatenlänge:	32 Byte Eingangsdaten / 32 Byte Ausgangsdaten
SIO-Modus:	nicht unterstützt
Port-Typ:	Typ A
Device ID:	0x400201 (IUT-F191-IO-V1-FR1-01) 0x400202 (IUT-F191-IO-V1-FR2-02) 0x400203 (IUT-F191-IO-V1-FR2-03) 0x400204 (IUT-F191-IO-V1-FR1-04) 0x400207 (IUT-F191-IO-V1-FR2-07) 0x400209 (IUT-F191-IO-V1-FR2-09) 0x400210 (IUT-F191-IO-V1-FR2-10) 0x400213 (IUT-F191-IO-V1-FR2-13) 0x400217 (IUT-F191-IO-V1-FR2-17)
Vendor ID:	0x01

### 3.8 Zubehör

#### 3.8.1 Datenträger

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

Tabelle 3.2

#### 3.8.2 Verbindungskabel IO-Link

Zum Anschluss der RFID-Schreib-/Lesestation an einen IO-Link-Master können Sie ungeschirmte, drei- oder vieradrige Kabel mit einem M12-Stecker mit einer maximalen Länge von 20m verwenden. Es können beispielsweise folgende Anschlusskabel von Pepperl+Fuchs verwendet werden:

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

Weiteres passendes Zubehör finden Sie auf unserer Webseite <http://www.pepperl-fuchs.com>.

## 4 Installation

### 4.1 Lagerung und Transport

Bewahren Sie die Originalverpackung auf. Lagern oder transportieren Sie das Gerät immer in der Originalverpackung.

Lagern Sie das Gerät immer in trockener und sauberer Umgebung. Beachten Sie die zulässigen Umgebungsbedingungen, siehe Datenblatt.

### 4.2 Auspacken

Prüfen Sie die Ware beim Auspacken auf Beschädigungen. Benachrichtigen Sie im Falle eines Sachschadens die Post bzw. den Spediteur und verständigen Sie den Lieferanten.

Überprüfen Sie den Lieferumfang anhand Ihrer Bestellung und der Lieferpapiere auf:

- Liefermenge
- Gerätetyp und Ausführung laut Typenschild
- gegebenenfalls mitbestelltes Zubehör

Bewahren Sie die Originalverpackung für den Fall auf, dass Sie das Gerät zu einem späteren Zeitpunkt einlagern oder verschicken.

Bei Fragen wenden Sie sich bitte an Pepperl+Fuchs.

### 4.3 Montage



---

#### Warnung!

Funktionsstörungen bei Herzschrittmachern

Dieses Gerät überschreitet **nicht** die zulässigen Grenzwerte für elektromagnetische Felder. Halten Sie mindestens einen Abstand von 30 cm zwischen dem Gerät und Ihrem Herzschrittmacher ein.

Bei zu geringem Abstand zur Schreib-/Lesestation können Inhibitionen, Umprogrammierungen oder falsche Stimulationsimpulse auftreten.

---

Die Schreib-/Lesestation ist für die Wandmontage oder die Montage an Halterungen im Innenbereich vorgesehen. Bitte befestigen Sie das Gerät nur mit den am Gehäuse vorhandenen Löchern. Die bevorzugte Montagerichtung ist mit dem Kabelanschluss senkrecht nach unten.



---

#### Hinweis!

Verlegen Sie das Anschlusskabel nicht in den Erfassungsbereich der Antenne.

---

Verwenden Sie zum Befestigen der Schreib-/Lesestation 3 Schrauben mit 4 mm Durchmesser zusammen mit Unterlegscheiben und Befestigungsmaterial, das auf die Beschaffenheit des Untergrunds abgestimmt ist. Das Anzugsdrehmoment der Schrauben richtet sich nach der Art der Befestigung. Wir empfehlen ein Anzugsdrehmoment von 1,8 Nm. Verwenden Sie ein Anzugsdrehmoment von max. 2,4 Nm, um Beschädigungen am Kunststoffgehäuse zu vermeiden.



---

#### Vorsicht!

Montage der Schreib-/Lesestation

Stellen Sie sicher, dass die Schreib-/Lesestation fest mit dem Untergrund verbunden ist.

---

**Hinweis!**

Die in diesem Dokument gemachten Installationsempfehlungen gehen von günstigsten Rahmenbedingungen aus. Pepperl+Fuchs übernimmt keine Gewähr für die einwandfreie Funktion in systemfremden Umgebungen.

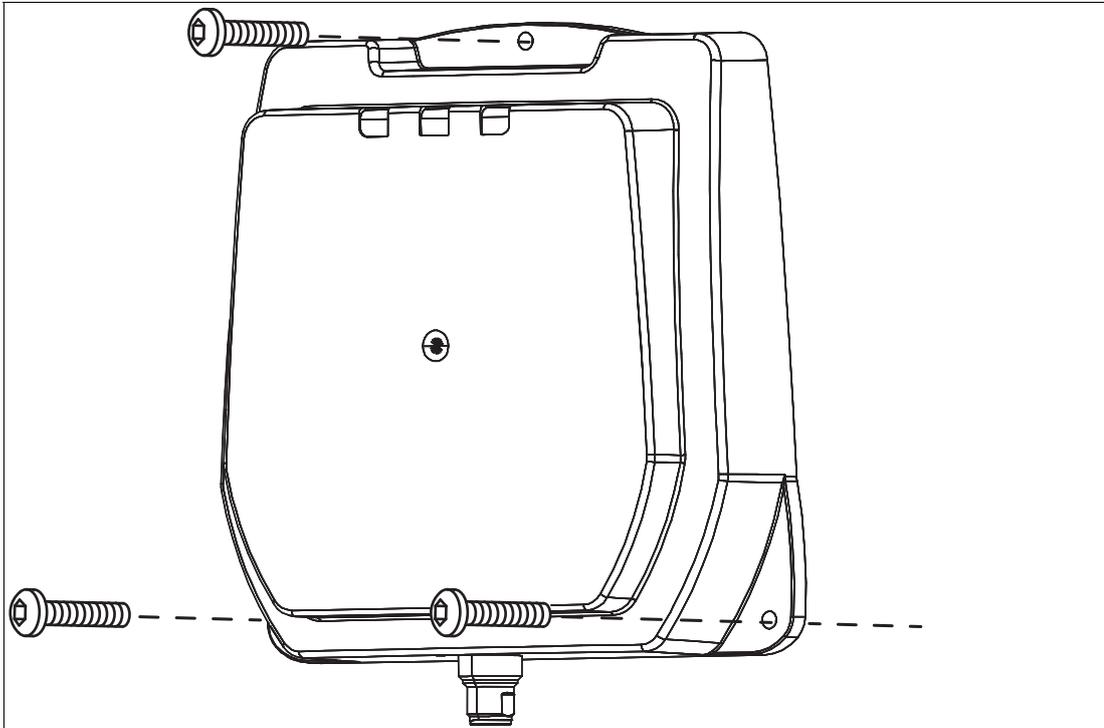
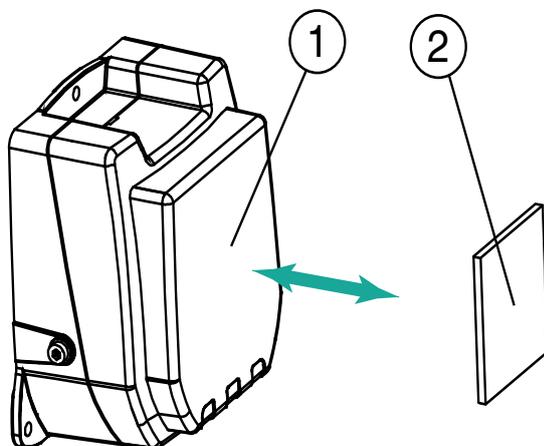
**Montage der Schreib-/Lesestation**

Abbildung 4.1

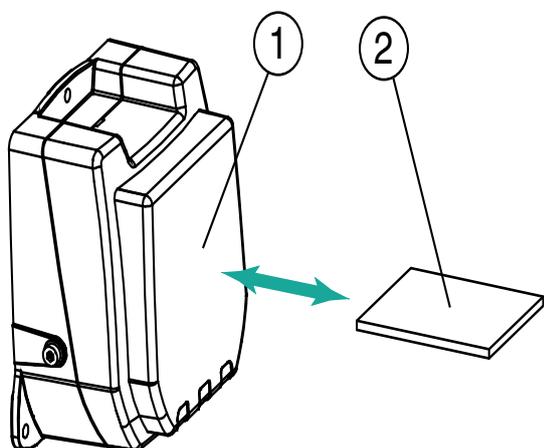
### 4.3.1 Orientierung im Raum

Die Ausrichtung der Antennen des Datenträgers in Bezug auf die Antenne des Schreib-/Lese-geräts beeinflusst die Reichweite des Systems. Achten Sie daher auf eine parallele Ausrichtung der Antennen zueinander.



Optimale Ausrichtung des Transponders

- gute Kommunikation zwischen Gerät und Transponder



Schlechte Ausrichtung des Transponders

- unzureichende Kommunikation zwischen Gerät und Transponder

- 1 Gerät
- 2 Transponder

### 4.3.2 Mindestabstände

Bei der Platzierung des Schreib-/Lesegeräts achten Sie bitte auf die Einhaltung von Mindestabständen. Der seitliche Abstand zwischen Schreib-/Lesegerät und Metallen oder Flüssigkeiten sollte mindestens 50 cm betragen. Der Abstand zwischen Schreib-/Lesegerät und dem Boden sollte ebenfalls mindestens 50 cm betragen.

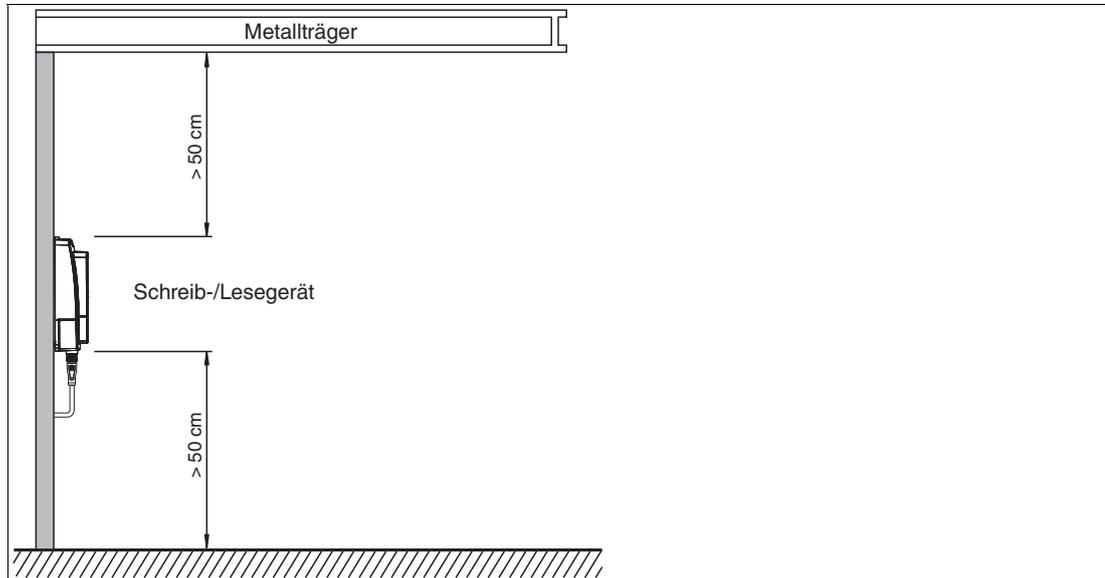


Abbildung 4.2

Beim gleichzeitigen Betrieb mehrerer Schreib-/Lesegeräte darf zu jedem Zeitpunkt jeweils nur ein Schreib-/Lesegerät mit einem Transponder kommunizieren. Wählen Sie den Abstand zwischen den Schreib-/Lesegeräten so, dass die Erfassungsbereiche nicht überlappen. Sie können den Erfassungsbereich durch entsprechende Änderung der Sendeleistung vergrößern oder verkleinern. Ermitteln Sie den Erfassungsbereich jedes Schreib-/Lesegeräts am Montageort.



#### Hinweis!

Beachten Sie bei der Montage mögliche Störungen der Schreib-/Lesegeräte untereinander. Je weiter die Sendekanäle der Schreib-/Lesegeräte voneinander entfernt sind, desto geringer ist die gegenseitige Störung.

### 4.3.3 Polarisation

Die Polarisation der von einer Antenne abgestrahlten elektromagnetischen Welle hängt von der Art der Antenne ab und ist für die elektrische Feldkomponente der elektromagnetischen Welle definiert. Man unterscheidet grundsätzlich zwischen einer linearen und einer zirkularen Polarisation. Bei einer elektromagnetischen Welle mit linearer Polarisation ist die Richtung des Vektors der elektrischen Feldkomponente im Raum konstant und daher abhängig von der Raumlage der Antenne. Lineare Polarisation liegt in vertikaler oder horizontaler Ausprägung vor.



#### Hinweis!

Die integrierte Antenne des Schreib-/Lesegeräts ist zirkular polarisiert. Die Polarisierung der Antenne kann nicht geändert werden.

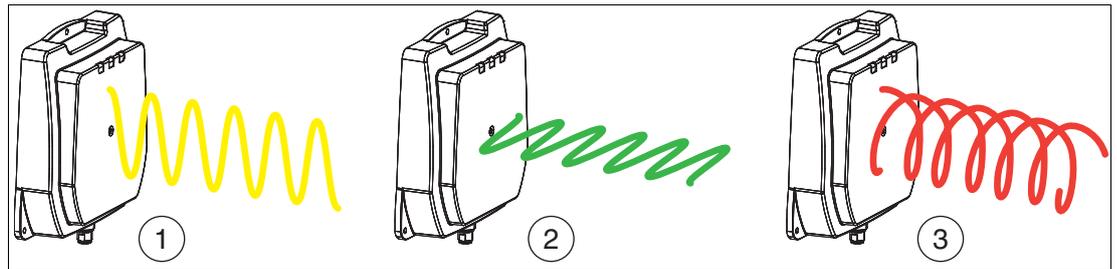


Abbildung 4.3 1 = vertikale Polarisierungsebene  
 2 = horizontale Polarisierungsebene  
 3 = zirkuläre Polarisation

## 4.4

### Anschluss

Schließen Sie die Schreib-/Lesestation mit einem Verbindungskabel an einen IO-Link-Master an.



#### Warnung!

Falscher elektrischer Anschluss

Beschädigung des Gerätes oder der Anlage durch falschen elektrischen Anschluss.

Prüfen Sie vor Inbetriebnahme des Gerätes und der Anlage alle Anschlüsse.

Nachdem Sie die Versorgungsspannung angeschlossen haben, leuchtet eine LED am Gerät grün. Wenn die LED am Gerät nicht leuchtet, ist die Spannungsversorgung falsch angeschlossen.

Eine bestehende IO-Link-Verbindung wird durch eine grün blinkende LED angezeigt.

## 5 Inbetriebnahme

### 5.1 Betriebsarten

Das Gerät unterstützt zwei Betriebsarten:

- **EasyMode**  
Der EasyMode ermöglicht eine vereinfachte Inbetriebnahme mit eingeschränktem Funktionsumfang. Bei Standardanwendungen ist diese Betriebsart zu bevorzugen.
- **ExpertMode**  
Im ExpertMode steht der komplette Befehlssatz zur Verfügung. Zur Nutzung des vollständigen Funktionsumfangs ist ein Funktionsbaustein für die Integration in die SPS notwendig.

## 6 Bedienung

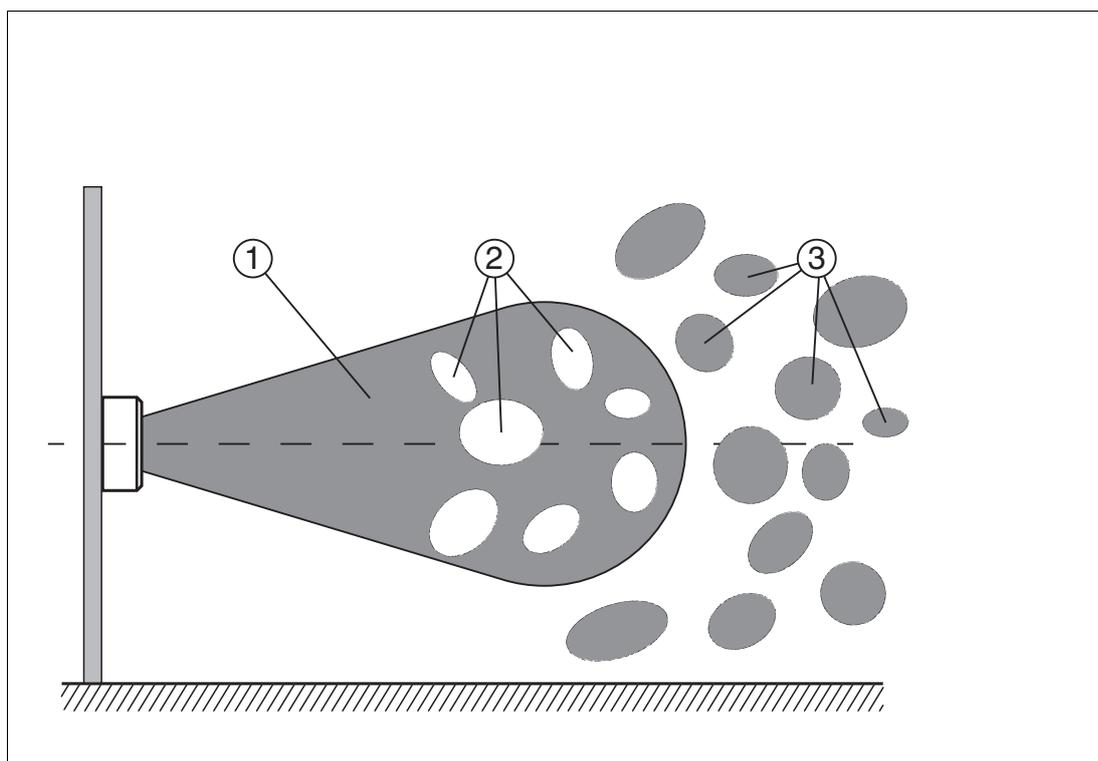
### 6.1 Allgemeines

In den folgenden Abschnitten erhalten Sie allgemeine Informationen über die Eigenschaften eines UHF-RFID-Systems. Diese sind unabhängig vom gewählten Betriebsmodus. Eine genauere Beschreibung der Betriebsmodi folgt in dem Kapitel Easy-Mode (siehe Kapitel 7) und in dem Kapitel Expert-Mode (siehe Kapitel 8).

### 6.2 Interferenz aufgrund von Mehrwegeausbreitung

Die vom Gerät abgestrahlten elektromagnetischen Wellen nehmen nicht nur den direkten Weg zum Transponder, sondern werden auch an Gegenständen der Umgebung reflektiert, so dass sich mehrere Teilwellen überlagern.

Dadurch kommt es zu Interferenzen durch Überhöhungen und Reduktionen der Empfangsfeldstärke, bis hin zur nahezu kompletten Auslöschung. Je nach Umgebung können viele Reflexionen mit unterschiedlicher Intensität und Entfernung auftreten. Dies führt zu einer nur schwer vorhersagbaren Feldstärke im Erfassungsbereich. In den Bereichen der Auslöschung ist die vorherrschende Feldstärke kleiner als die minimale Ansprechfeldstärke des Transponders, so dass der Transponder nicht zur Kommunikation aktiviert werden kann. Überhöhungen der Feldstärke können zu ungewollten Überreichweiten führen.



1. Erfassungsbereich
2. Auslöschung
3. Überreichweiten

Die Reflexionen und die daraus resultierende räumliche Inhomogenität der Feldstärke sind abhängig von der verwendeten Frequenz. Der Absolutwert der Feldstärke ist abhängig von der Sendeleistung. Da sich die Transponder im Erfassungsbereich des Geräts bewegen und sich die Umgebung ändern kann, ist es sinnvoll, die Befehle auf unterschiedlichen Sendefrequenzen und mit variierenden Leistungen zu wiederholen. Unterschiedliche Sendefrequenzen sind auch sinnvoll, da Fertigungstoleranzen und die unmittelbare Umgebung des Transponders Auswirkung auf seine Resonanzfrequenz haben.

## 6.3 Mehrere Transponder im Erfassungsbereich

Das Verhalten bei der Identifikation von Transpondern im Erfassungsbereich ist abhängig vom gewählten Betriebsmodus.

### Easy-Mode

Alle erkannten Datenträger werden übertragen. Eine Filterung auf einzelne Transponder ist nicht möglich.

### Expert-Mode

Jeder Lese- und Schreibbefehl kann auf einen, mehrere oder alle im Erfassungsbereich befindlichen Transponder zugreifen. Zur Steuerung werden Filtermasken verwendet, die mit den Befehlen Filtermaske setzen (**FI**) und Filter ein-/ausschalten (**MF**) verwaltet werden. Mit diesen Befehlen können Sie gezielt bestimmte Transponder im Erfassungsbereich ansprechen. Siehe Kapitel 8.6.3.

## 6.4 Lesealgorithmus

Um mit einer möglichst hohen Wahrscheinlichkeit mit Transpondern zu kommunizieren, ist im Gerät ein Algorithmus implementiert, der die Sendeleistung und die Frequenz variiert. Mit den Parametern Sendeleistung (**PT**), Kanalfrequenz (**CD**) bzw. Anzahl Kanäle (**NC**) beim Frequenzsprungverfahren und Anzahl der Versuche (**TA**) können Sie die entsprechenden Werte für diesen Algorithmus einstellen. Für jede Paarung aus Leistung und Frequenz wird die Anzahl von eingestellten Versuchen durchgeführt. Dieses Verfahren ist zeitaufwendig, führt aber zu einer hohen Schreib- bzw. Leserate. Der Algorithmus durchläuft alle Kombinationen, da ein Transponder möglicherweise nur durch eine bestimmte Kombination aus Leistung und Frequenz angesprochen werden kann. Dies gilt, wenn das Gerät als Frequenzzugriffsverfahren eine parametrierbare Frequenzliste verwendet. Diese Frequenzliste ist durch den Parameter **CD** definiert.

Wenn durch die Länderkennung ein Frequenzsprungverfahren vorgegeben ist, werden die darin definierten Kanäle verwendet. Sie können die Anzahl der verwendeten Kanäle für jeden Versuch über den Parameter Anzahl Kanäle (**NC**) steuern.

Mit dem Parameter Abbruchkriterium Suchalgorithmus (**NT**) können Sie die Anzahl der zu verarbeitenden Transponder vorgeben. Wenn Ihnen die Anzahl der Transponder im Erfassungsbereich bekannt ist, können Sie diese Gesamtanzahl als Wert für den Parameter **NT** nutzen. Wenn die Anzahl der gefundenen Transponder dem im Parameter **NT** hinterlegten Wert entspricht oder überschreitet, bricht der Algorithmus weitere Durchläufe ab, um Zeit zu sparen.



### Tipp

Wenn Untersuchungen einer speziellen Anwendung zeigen, dass eine Frequenz und eine Sendeleistung zum erfolgreichen Ausführen der Befehle ausreicht, können die Parameter im Rahmen der nationalen Bestimmungen entsprechend eingestellt werden. Dadurch wird die Bearbeitungszeit verkürzt.

---

## 7 EasyMode

Das RFID-Schreib-/Lesegerät verwendet zur Datenübertragung an ein übergeordnetes System das Kommunikationsprotokoll "EasyMode" auf Basis von IO-Link. Bei Verwendung dieses Protokolls kann das RFID-Schreib-/Lesegerät ohne zusätzlichen Funktionsbaustein an einer Steuerung in Betrieb genommen werden. Dadurch vereinfacht sich der Aufwand zur Inbetriebnahme des Schreib-/Lesegeräts.

Bei der Nutzung des "EasyMode" wird zwischen Parameter- und Prozessdaten unterschieden. Die Parameterdaten sind IO-Link-Parameter, die azyklisch übertragen werden. Dabei handelt es sich um Daten für die Konfiguration der Schreib-/Leseaufträge, zur Parametrierung der Geräteeigenschaften, z.B. Sendeleistung und Servicedaten, z.B. Betriebsstundenzähler. Die Prozessdaten werden zyklisch übertragen. Die Prozessdaten unterteilen sich in Eingangs- und Ausgangsdaten. Sie haben eine Länge von je 32 Byte und beinhalten die Steuerwerte zur Ausführung der Lese- und Schreibbefehle und die zugehörigen Werte.

Die IO-Link-Parameter zur Einstellung des RFID-Schreib-/Lesegeräts sind durch eine gerätespezifische IODD-Datei definiert. Die Einstellung der IO-Link-Parameter erfolgt mit einer geeigneten Konfigurations-Software. Dabei werden die IO-Link-Parameter nichtflüchtig in dem RFID-Schreib-/Lesegerät gespeichert.

### 7.1 Befehlsübersicht

Der Easy-Mode unterstützt die folgenden Lese- und Schreibbefehle über das Prozessabbild:

Befehl	Beschreibung
Lesen EPC/UII	<p>Auslesen der EPC Memory Bank (Bank 01); Es wird das 2 Byte Protocol Control (PC) und die EPC/UII gelesen.</p> <ul style="list-style-type: none"> <li>Beispiel: Datenträger vom Typ IUC76-* haben in der Regel eine EPC/UII Länge von 12 Byte zzgl. 2 Byte PC.</li> <li>Struktur der Antwort-Daten, Long-Form-Datenformat: [Länge der EPC/UII Information; 2 Byte][PC; 2 Byte][EPC/UII; variable Länge]</li> <li>Struktur der Antwort-Daten, Short-Form-Datenformat: [PC; 2 Byte][EPC/UII; variable Länge]</li> </ul>
Lesen TID	<p>Auslesen der TID (Bank 10); In den Antwort-Daten wird immer die EPC/UII Information der TID vorangestellt.</p> <ul style="list-style-type: none"> <li>Struktur der Antwort-Daten, Long-Form-Datenformat: [Länge der EPC/UII Information; 2 Byte][PC; 2 Byte][EPC/UII; variable Länge][Länge der TID-Information; 2 Byte][TID; variable Länge]</li> <li>Struktur der Antwort-Daten, Short-Form-Datenformat: [TID; variable Länge]</li> </ul>
Lesen User Memory	<p>Auslesen der User Memory Bank (Bank 11); In den Antwort-Daten wird immer die EPC/UII Information der User Memory Daten vorangestellt. Die Startadresse und die zu lesende Datenmenge sind über die Parameter in der IODD-Datei zu konfigurieren.</p> <ul style="list-style-type: none"> <li>Struktur der Antwort-Daten, Long-Form-Datenformat: [Länge der EPC/UII Information; 2 Byte][PC; 2 Byte][EPC/UII; variable Länge][Länge der User Memory Information; 2 Byte][User Memory; Länge abhängig von Parametereinstellungen]</li> <li>Struktur der Antwort-Daten, Short-Form-Datenformat: [User Memory; Länge abhängig von Parametereinstellungen]</li> </ul>
Autostart	<p>Automatische Ausführung eines Lesebefehls gemäß der Parametereinstellungen in der IODD-Datei. Es kann jeweils einer der oben beschriebenen Lesebefehle automatisch ohne Ansteuerung über das Prozessabbild gestartet werden. In der werksseitigen Einstellung ist der Autostart aktiviert. Es werden 8 Byte User Memory ab der Startadresse 0 eingelesen.</p>
Input Darstellung	<p>Definiert den Inhalt der gelesenen Daten. Wenn das Langform-Datenformat verwendet wird, wird bei einem Lesezugriff auf den User-Memory oder die TID, immer das UII/EPC und zusätzliche Längenangaben mit ausgegeben.</p>

Befehl	Beschreibung
Schreiben EPC/UII	<p>Schreiben der EPC/UII Bank (Bank 01); Die zu schreibende Datenmenge und Inhalte sind über die Parameter der IODD-Datei zu konfigurieren. Die zu schreibenden Daten beginnen im Ausgangsdatenfeld ab Byte 4.</p> <ul style="list-style-type: none"> <li>Struktur der zu schreibenden Daten: <ol style="list-style-type: none"> <li>PC+EPC/UII [PC; 2 Byte][EPC/UII; variable Länge bis zu 26 Byte]</li> <li>nur EPC, PC wird automatisch berechnet [EPC; variable Länge bis zu 28 Byte]</li> </ol> </li> </ul>
Schreiben User Memory	<p>Schreiben der User Memory Bank (Bank 11); Die Startadresse und die zu schreibende Datenmenge sind über die Parameter in der IODD-Datei zu konfigurieren. Die zu schreibenden Daten beginnen im Ausgangsdatenfeld ab Byte 4.</p> <ul style="list-style-type: none"> <li>Struktur der Daten: <p>[Control Byte; 1 Byte][not used; 3 Byte][Schreibdaten; Länge abhängig von Parametereinstellungen]</p> </li> </ul>

Tabelle 7.1 EasyMode-Befehlsübersicht

## 7.2 Grundlegende Struktur der Prozessdaten



### Vorsicht!

Der Zwischenspeicher ist nur für eine geringe Menge an Daten ausgelegt, um kurze Unterbrechungen zu kompensieren.

Bei längerer Unterbrechung der Kommunikation oder größeren Datenmengen während einer Unterbrechung, können Daten verloren gehen.

Die Prozessdaten des Geräts werden zyklisch zwischen einem übergeordneten System (z. B. einer SPS) und dem Gerät ausgetauscht. Dabei wird zwischen den Ausgangsprozessdaten und den Eingangsprozessdaten unterschieden. Die Prozessdaten der Ausgänge werden von der Steuerung in Richtung Gerät übertragen. Die Prozessdaten der Eingänge werden vom Gerät in Richtung Steuerung übertragen.

Ausgangsprozessdaten, bei dem das "valid"-Flag nicht gesetzt ist (sprich Prozessdaten, die als ungültig markiert sind), werden vom Gerät ignoriert. Wird die IO-Link-Kommunikation unterbrochen, arbeitet das Gerät normal weiter. Ist zum Beispiel zu diesem Zeitpunkt gerade ein Leseauftrag aktiv, werden die vom Transponder empfangenen Daten im Gerät zwischengespeichert und bei erneuter Kommunikation an den IO-Link-Master übertragen.

Eine detaillierte Beschreibung der Prozessdaten finden Sie in der IODD des Geräts. Sie finden die IODD auf der Produktdetailseite unter [www.pepperl-fuchs.com](http://www.pepperl-fuchs.com).

Die Ein- und Ausgangsdaten der Prozessdaten haben eine feste Länge von 32 Byte.

### 7.2.1 Ausgangsprozessdaten (SPS -> Gerät)

Byte	Inhalt
0	0 0 0 0 0 0 0 Start Schreiben Start Lesen
1	0x00
2	0x00
3	0x00
4	Anwenderdaten für Schreibauftrag bzw. 16#00 bei Leseauftrag
...	Anwenderdaten für Schreibauftrag bzw. 16#00 bei Leseauftrag
31	Anwenderdaten für Schreibauftrag bzw. 16#00 bei Leseauftrag

Tabelle 7.2

- Byte 0**                      Dieses Byte beinhaltet die Steuerbits zum Start eines Leseauftrags bzw. eines Schreibauftrags. Wenn die Autostart-Funktion aktiviert ist, haben die Steuerbits keine Auswirkung.  
**Lesen:** Sobald dieses Bit gesetzt (TRUE) wird, wird ein Leseauftrag gestartet entsprechend der Konfiguration durch die Parameter in der IODD-Datei. Der Leseauftrag wird dabei kontinuierlich ausgeführt. Um den Leseauftrag abbrechen, setzen Sie das Bit zurück (FALSE).  
**Schreiben:** Sobald dieses Bit gesetzt (TRUE) wird, wird ein Schreibauftrag gestartet. Der Schreibauftrag überträgt die Anwenderdaten, die ab Byte 4 hinterlegt sein müssen, entsprechend der Konfiguration durch die Parameter in der IODD-Datei. Der Schreibauftrag wird dabei kontinuierlich ausgeführt. Um den Schreibauftrag abbrechen, setzen Sie das Bit zurück (FALSE). Beachten Sie, dass beide Bits nicht gleichzeitig gesetzt werden. Die restlichen Bits haben keine Bedeutung.
- Byte 1/2/3**                      Diese Bytes werden beim Easy-Modes nicht verwendet. Stellen Sie den Wert 0x00 ein.
- Byte 4 ... 31**                      Bei der Ausführung eines Leseauftrags haben diese Bytes keine Bedeutung und sind mit dem Wert 0x00 belegt.  
 Bei der Ausführung eines Schreibauftrags sind in diesem Bereich die Anwenderdaten, die auf den Datenträger geschrieben werden sollen, hinterlegt.

**Beispiel: Schreiben User Memory im TIA-Portal, Long-Form-Datenformat**

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

**Ausgangsprozessdaten**

- Byte 0:                      0000 0010<sub>bin</sub>; Das Bit "Start Schreiben" ist auf 1 gesetzt; es wird ein Schreibauftrag ausgeführt.
- Byte 1/2/3:                      16#00; Nicht verwendet
- Byte 4 ...                      Schreibdaten
- Byte 11:

## Eingangsprozessdaten

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

Byte 0: 0001 0110<sub>bin</sub>; Das Bit "Auftrag Aktiv" ist auf 1 gesetzt. Es wird hierdurch signalisiert, dass ein Schreibauftrag aktiviert ist. Ebenso ist das Bit "Schreiben erfolgreich" auf 1 gesetzt. Es befindet sich somit ein Datenträger im Erfassungsbereich und die Daten wurden erfolgreich geschrieben. Zusätzlich ist das Bit "Transponder vorhanden" gesetzt, da sich mindestens ein Transponder im Erfassungsbereich befindet

Byte 1: 16; Das Byte zeigt die Länge der übertragenen Informationen an. Es wird durch dieses Telegramm eine 16-Byte-lange Information übertragen. Die Information beginnt ab Byte 4.

Byte 2: 100, RSSI-Wert

Byte 3: 17; Sendeleistung in dBm. 17 dBm = 50 mW

Byte 4/5: 16#000E (14dez); Längenangabe der EPC/UII-Information. Die EPC/UII-Information besteht aus dem PC-Word mit der Länge von 2 Byte zuzüglich des EPC/UII. Die Länge des EPC/UII ist variabel.  
In diesem Beispiel ist die Länge des EPC/UII 12 Byte.

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

Byte 8 ...  
Byte 19: EPC/UII des Datenträgers, auf den die Daten geschrieben wurden.

## 7.2.2 Eingangsprozessdaten (Gerät -&gt; SPS)

Byte	Inhalt
0	0      0      0      Transponder vorhanden      Fehler      Auftrag aktiv      Schreiben erfolgreich      Lesen erfolgreich
1	Längenangabe
2	RSSI-Wert
3	Sendeleistung in dBm
4	Daten bei Lese-/Schreibauftrag / Fehlercode
5	Daten bei Lese-/Schreibauftrag / Fehlerinformation
...	Daten bei Lese-/Schreibauftrag / Fehlerinformation
31	Daten bei Lese-/Schreibauftrag / Fehlerinformation

Tabelle 7.3

<b>Byte 0</b>	<p>Dieses Byte beinhaltet die Steuerbits zur Signalisierung des Zustands der Ausführung des Lese- bzw. Schreibauftrags.</p> <p><b>Lesen erfolgreich:</b> Dieses Bit signalisiert ein erfolgreiches Einlesen von Daten eines Datenträgers. Wenn ein Datenträger in den Erfassungsbereich eintritt und die Daten erfolgreich eingelesen wurden, wird dieses Bit gesetzt. Während der Datenträger sich innerhalb des Erfassungsbereichs aufhält, bleibt das Bit gesetzt. Sobald der Datenträger diesen Bereich verlassen hat, wird das Bit wieder zurückgesetzt. Bei mehreren Datenträgern, die sich gleichzeitig im Erfassungsbereich befinden, wird durch einen positiven Flankenwechsel<sup>1</sup> die Übertragung eines weiteren Datenträgers signalisiert.</p> <p><b>Schreiben erfolgreich:</b> Dieses Bit signalisiert ein erfolgreiches Schreiben von Daten auf einen Datenträger. Wenn ein Datenträger in den Erfassungsbereich eintritt und die Daten erfolgreich geschrieben wurden, wird dieses Bit gesetzt. Während der Datenträger sich innerhalb des Erfassungsbereichs aufhält, bleibt das Bit gesetzt. Sobald der Datenträger diesen Bereich verlassen hat, wird das Bit wieder zurückgesetzt. Bei mehreren Datenträgern, die sich gleichzeitig im Erfassungsbereich befinden, wird durch einen positiven Flankenwechsel das erfolgreiche Schreiben eines weiteren Datenträgers signalisiert.</p> <p><b>Auftrag aktiv:</b> Während der Lese- bzw. Schreibauftrag ausgeführt wird, ist dieses Bit gesetzt. Sobald der Auftrag beendet ist, wird dieses Bit wieder zurückgesetzt.</p> <p><b>Fehler:</b> Falls bei der Durchführung eines Lese- bzw. Schreibauftrags ein Fehler auftritt oder die Einstellung eines Parameters nicht korrekt ist, wird dies durch dieses Bit signalisiert. Gleichzeitig befindet sich in den Prozessdaten zusätzliche Fehlerinformation in Form eines Fehlercodes sowie einer Fehlerbeschreibung. Für eine detaillierte Fehlerbeschreibung siehe Kapitel 8.7.</p> <p><b>Transponder vorhanden:</b> Bit ist gesetzt, wenn sich ein oder mehrere Datenträger im Erfassungsbereich befinden. Befindet sich kein Datenträger im Erfassungsbereich, hat dieses Bit den Wert FALSE.</p>
<b>Byte 1</b>	Dieses Byte enthält die Anzahl der übertragenen Bytes. Wenn ein Datenträger in den Erfassungsbereich eintritt und die Daten erfolgreich gelesen wurden (Lesen erfolgreich = TRUE), zeigt dieses Byte die Länge der eingelesenen Daten an. Wenn bei der Ausführung eines Auftrags ein Fehler auftritt (Fehler = TRUE), enthält das Byte eine Längenangabe zur Fehlerinformation.
<b>Byte 2</b>	Der RSSI-Wert zeigt die Signalstärke der Transponder-Antwort in Prozent an. Der Wertebereich liegt zwischen 0 und 100. 0 = schwaches Signal, 100 = starkes Signal.
<b>Byte 3</b>	Sendeleistung der Schreib-/Lesestation in dBm, bei der der Transponder identifiziert wurde.
<b>Byte 4/5</b>	Wenn erfolgreich auf einen Datenträger zugegriffen wird, befindet sich an dieser Position die Länge der EPC/UII Information (PC-Wort + EPC/UII). <sup>2</sup>
<b>Byte 6/7</b>	Wenn erfolgreich auf einen Datenträger zugegriffen wird, befindet sich das PC-Wort (Protocol Control) an dieser Position.
<b>Byte 8 ... Byte 31</b>	Beginnend ab Byte 8 wird der EPC/UII des Datenträgers übertragen, von dem die Informationen eingelesen bzw. auf den die Informationen geschrieben wurden. In der Rückantwort auf einen Lese- oder Schreibauftrag ist immer die EPC/UII enthalten. Hierdurch ist eine eindeutige Zuordnung des Datenträgers gewährleistet. Bei der Ausführung eines Leseauftrags schließt sich daran eine Längenangabe (2 Byte) über die zusätzlich eingelesenen Informationen (TID oder User Memory) an.

1. Positiver Flankenwechsel: Wechsel von 0 auf 1
2. Gilt nur für das Long-Form-Datenformat. Im Short-Form-Datenformat wird ab Byte 4 direkt das PC-Wort (Protocol Control) und der EPC/UII bzw. der Nutzspeicher des Transponders übertragen.

Wenn bei der Ausführung eines Lese- oder Schreibauftrags ein Fehler auftritt (Bit "Fehler" = TRUE), enthält Byte 4 einen Fehlercode. Im Fehlerzustand wird beginnend ab Byte 5 eine Fehlermeldung in Klarschrift (ASCII) übertragen. Hierdurch wird auf eine möglichen Fehlerursache hingewiesen.

### Beispiel: Lesen EPC, Long-Form-Datenformat

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_EPCUII_HighByte*	%IB4	Hex	16#00
*Length_EPCUII_LowByte*	%IB5	Hex	16#0E
*PC_Word_HighByte*	%IB6	Hex	16#34
*PC_Word_LowByte*	%IB7	Hex	16#00
*EPCUII_Byte_1*	%IB8	Hex	16#30
*EPCUII_Byte_2*	%IB9	Hex	16#14
*EPCUII_Byte_3*	%IB10	Hex	16#F7
*EPCUII_Byte_4*	%IB11	Hex	16#33
*EPCUII_Byte_5*	%IB12	Hex	16#7C
*EPCUII_Byte_6*	%IB13	Hex	16#00
*EPCUII_Byte_7*	%IB14	Hex	16#1F
*EPCUII_Byte_8*	%IB15	Hex	16#00
*EPCUII_Byte_9*	%IB16	Hex	16#00
*EPCUII_Byte_10*	%IB17	Hex	16#00
*EPCUII_Byte_11*	%IB18	Hex	16#D1
*EPCUII_Byte_12*	%IB19	Hex	16#A7

Byte 0:

0001 0101<sub>bin</sub>; Das Bit "Auftrag Aktiv" ist auf 1 gesetzt und es wird hierdurch signalisiert, dass eine Leseauftrag aktiviert ist. Ebenso ist das Bit "Lesen erfolgreich" auf 1 gesetzt. Es befindet sich somit ein Datenträger im Erfassungsbereich und die Daten wurden eingelesen. Zusätzlich ist das Bit "Transponder vorhanden" gesetzt, da sich mindestens ein Transponder in der Erfassungszone befindet.

Byte 1:

16; Das Byte zeigt die Länge der übertragenen Informationen an. Es wird durch dieses Telegramm eine 16 Byte lange Information übertragen. Die Information beginnt ab Byte 4.

Byte 2:

26; RSSI-Wert

Byte 3:

17; Sendeleistung in dBm, 17 dBm = 50 mW

Byte 4/5:

16#000E; Längenangabe der EPC/UII-Information. Die EPC/UII-Information besteht aus dem PC-Wort mit der Länge von 2 Byte zuzüglich des EPC/UII. Die Länge des EPC/UII ist variabel. In diesem Beispiel ist die Länge des EPC/UII 12 Byte.

Byte 6/7:

16#3400; PC-Wort (Protocol Control)

Byte 8 ...

EPC/UII des Datenträgers, der ausgelesen wurde

Byte 19:

**Beispiel: Lesen EPC, Short-Form-Datenformat**

Name	Address	Displa...	Monitor value
*ControlByte*	%B0	Bin	2#0001_0101
*Length*	%B1	DEC	14
*RSSI*	%B2	DEC	26
*PowerTransmit_dBm*	%B3	DEC	17
*PC_Word_HighByte*	%B4	Hex	16#34
*PC_Word_LowByte*	%B5	Hex	16#00
*EPC/UII_Byte_1*	%B6	Hex	16#30
*EPC/UII_Byte_2*	%B7	Hex	16#14
*EPC/UII_Byte_3*	%B8	Hex	16#F7
*EPC/UII_Byte_4*	%B9	Hex	16#33
*EPC/UII_Byte_5*	%B10	Hex	16#7C
*EPC/UII_Byte_6*	%B11	Hex	16#00
*EPC/UII_Byte_7*	%B12	Hex	16#1F
*EPC/UII_Byte_8*	%B13	Hex	16#00
*EPC/UII_Byte_9*	%B14	Hex	16#00
*EPC/UII_Byte_10*	%B15	Hex	16#00
*EPC/UII_Byte_11*	%B16	Hex	16#D1
*EPC/UII_Byte_12*	%B17	Hex	16#A7

Byte 0:

0001 0101<sub>bin</sub>; Das Bit "Auftrag Aktiv" ist auf 1 gesetzt und es wird hierdurch signalisiert, dass eine Leseauftrag aktiviert ist. Ebenso ist das Bit "Lesen erfolgreich" auf 1 gesetzt. Es befindet sich somit ein Datenträger im Erfassungsbereich und die Daten wurden eingelesen. Zusätzlich ist das Bit "Transponder vorhanden" gesetzt, da sich mindestens ein Transponder in der Erfassungszone befindet.

Byte 1:

14; Das Byte zeigt die Länge der übertragenen Informationen an. Es wird durch dieses Telegramm eine 14 Byte lange Information übertragen. Die Information beginnt ab Byte 4.

Byte 2:

26; RSSI-Wert

Byte 3:

17; Sendeleistung in dBm, 17 dBm = 50 mW

Byte 4/5:

16#3400; PC-Word (Protocol Control)

Byte 6 ...

EPC/UII des Datenträgers, der ausgelesen wurde

Byte 17:

**Beispiel: Lesen User Memory, Long-Form-Datenformat**

Name	Address	Displa...	Monitor value
*ControlByte*	%B0	Bin	2#0001_0101
*Length*	%B1	DEC	26
*RSSI*	%B2	DEC	20
*PowerTransmit_dBm*	%B3	DEC	17
*Length_EPC/UII_HighByte*	%B4	Hex	16#00
*Length_EPC/UII_LowByte*	%B5	Hex	16#0E
*PC_Word_HighByte*	%B6	Hex	16#34
*PC_Word_LowByte*	%B7	Hex	16#00
*EPC/UII_Byte_1*	%B8	Hex	16#30
*EPC/UII_Byte_2*	%B9	Hex	16#14
*EPC/UII_Byte_3*	%B10	Hex	16#F7
*EPC/UII_Byte_4*	%B11	Hex	16#33
*EPC/UII_Byte_5*	%B12	Hex	16#7C
*EPC/UII_Byte_6*	%B13	Hex	16#00
*EPC/UII_Byte_7*	%B14	Hex	16#1F
*EPC/UII_Byte_8*	%B15	Hex	16#00
*EPC/UII_Byte_9*	%B16	Hex	16#00
*EPC/UII_Byte_10*	%B17	Hex	16#00
*EPC/UII_Byte_11*	%B18	Hex	16#D1
*EPC/UII_Byte_12*	%B19	Hex	16#A7
*Length_Data_HighByte*	%B20	Hex	16#00
*Length_Data_LowByte*	%B21	Hex	16#08
*Data_Byte_1*	%B22	Hex	16#01
*Data_Byte_2*	%B23	Hex	16#02
*Data_Byte_3*	%B24	Hex	16#03
*Data_Byte_4*	%B25	Hex	16#04
*Data_Byte_5*	%B26	Hex	16#05
*Data_Byte_6*	%B27	Hex	16#06
*Data_Byte_7*	%B28	Hex	16#07
*Data_Byte_8*	%B29	Hex	16#08

Byte 0:

0001 0101<sub>bin</sub>; Das Bit "Auftrag Aktiv" ist auf 1 gesetzt und signalisiert einen aktiven Leseauftrag. Das Bit "Lesen erfolgreich" ist auf 1 gesetzt. Es befindet sich ein Datenträger im Erfassungsbereich und die Daten wurden eingelesen. Zusätzlich ist das Bit "Transponder vorhanden" gesetzt, da sich mindestens ein Transponder in der Erfassungszone befindet.

Byte 1:

26; Das Byte zeigt die Länge der übertragenen Informationen an. Es wird durch dieses Telegramm eine 26 Byte lange Information übertragen. Die Information beginnt ab Byte 4.

Byte 2:

20; RSSI-Wert

Byte 3:

17; Sendeleistung in dBm, 17 dBm = 50 mW

Byte 4/5:

16#000E; Längenangabe der EPC/UII-Information. Die EPC/UII-Information besteht aus dem PC-Word mit der Länge von 2 Byte zuzüglich des EPC/UII. Die Länge des EPC/UII ist variabel. In diesem Beispiel ist die Länge des EPC/UII 12 Byte.

Byte 6/7:

16#3400; PC-Word (Protocol Control)

Byte 8 ...

EPC/UII des identifizierten Datenträgers

Byte 19:

Byte 20/21:

16#0008; Längenangabe des eingelesenen User Memory

Byte 22 ...

eingelesener Teilbereich des User

Byte 29:

Memory

**Beispiel: Lesen User Memory, Short-Form-Datenformat**

Name	Address	Displa...	Monitor value
*ControlByte*	%IB0	Bin	2#0001_0101
*Length*	%IB1	DEC	8
*RSSI*	%IB2	DEC	33
*PowerTransmit_dBm*	%IB3	DEC	17
*Data_Byte_1*	%IB4	Hex	16#01
*Data_Byte_2*	%IB5	Hex	16#02
*Data_Byte_3*	%IB6	Hex	16#03
*Data_Byte_4*	%IB7	Hex	16#04
*Data_Byte_5*	%IB8	Hex	16#05
*Data_Byte_6*	%IB9	Hex	16#06
*Data_Byte_7*	%IB10	Hex	16#07
*Data_Byte_8*	%IB11	Hex	16#08

Byte 0: 0001 0101<sub>bin</sub>; Das Bit "Auftrag Aktiv" ist auf 1 gesetzt und signalisiert einen aktiven Leseauftrag. Das Bit "Lesen erfolgreich" ist auf 1 gesetzt. Es befindet sich ein Datenträger im Erfassungsbereich und die Daten wurden eingelesen. Zusätzlich ist das Bit "Transponder vorhanden" gesetzt, da sich mindestens ein Transponder in der Erfassungszone befindet.

Byte 1: 8; Das Byte zeigt die Länge der übertragenen Informationen an. Es wird durch dieses Telegramm eine 8 Byte lange Information übertragen. Die Information beginnt ab Byte 4.

Byte 2: 33; RSSI-Wert

Byte 3: 17; Sendeleistung in dBm, 17 dBm = 50 mW

Byte 4 ...  
Byte 11: eingelesener Teilbereich des User Memory

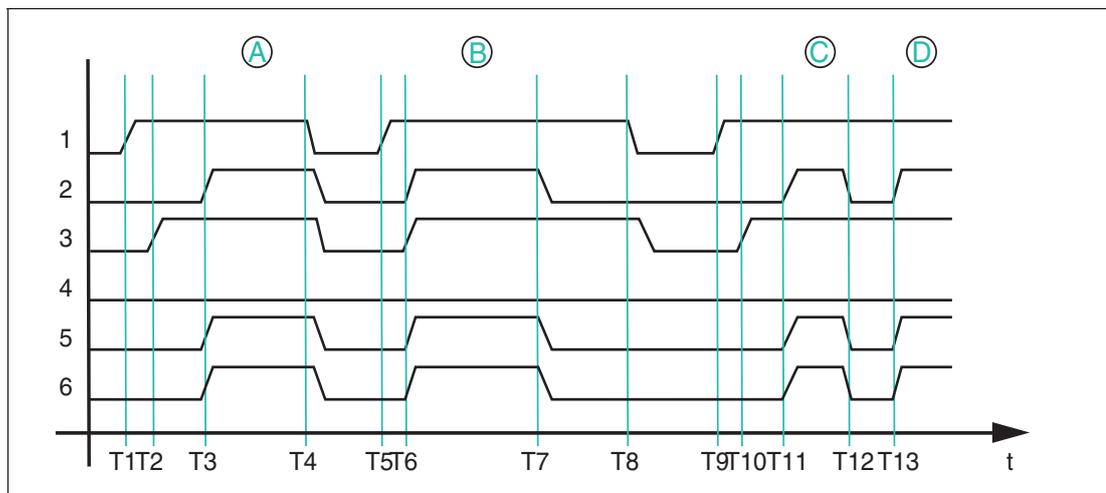
**7.2.3****Ablaufdiagramme****Leseauftrag ohne Autostart-Funktion**

Abbildung 7.1 Zeitlicher Ablauf der Bits im Byte 0

- 1 Start Lesen
- 2 Lesen erfolgreich
- 3 Auftrag aktiv
- 4 Fehler
- 5 Transponder vorhanden
- 6 Daten (Eingang)

Wenn die Autostart-Funktion nicht verwendet wird, wird der Leseauftrag durch das Bit "Start Lesen" gestartet. Der Leseauftrag wird so lange ausgeführt, bis das Bit "Start Lesen" wieder auf FALSE zurückgesetzt wird.

T1: Starten des Leseauftrags durch Setzen des Bits "Start Lesen" auf TRUE

T2: Leseauftrag wird ausgeführt und durch das Bit "Auftrag aktiv" signalisiert ("Auftrag aktiv" = TRUE)

T3: Datenträger A tritt in den Erfassungsbereich ein; "Lesen erfolgreich" und "Transponder vorhanden" werden auf TRUE gesetzt und die eingelesenen Daten befinden sich im Eingangsfeld der Prozessdaten

T4: Abbruch des Leseauftrags durch Rücksetzen des Bits "Start Lesen" auf FALSE während sich der Datenträger im Erfassungsbereich befindet; das Bit "Auftrag aktiv", das Bit "Lesen erfolgreich" und das Bit "Transponder vorhanden" werden auf FALSE gesetzt und die Prozessdaten mit 0x00 aufgefüllt

T5: Starten des Leseauftrags durch Setzen des Bits "Start Lesen" auf TRUE; zum Zeitpunkt des Starts befindet sich bereits ein Datenträger B im Erfassungsbereich

T6: Leseauftrag wird ausgeführt ("Auftrag aktiv" = TRUE) und die Daten werden erfolgreich eingelesen ("Lesen erfolgreich" und "Transponder vorhanden" = TRUE); die eingelesenen Daten befinden sich innerhalb im Eingangsfeld der Prozessdaten

T7: Datenträger verlässt den Erfassungsbereich ("Lesen erfolgreich" und "Transponder vorhanden" = FALSE); der Bereich des Eingangsfeldes mit den eingelesenen Prozessdaten wird auf den Wert 0x00 gesetzt

T8: Abbruch des Leseauftrags ("Start Lesen" = FALSE); das Bit "Auftrag aktiv" wird zurückgesetzt

T9: Starten des Leseauftrags durch Setzen des Bits "Start Lesen" auf TRUE; zum Zeitpunkt des Starts befindet sich kein Datenträger im Erfassungsbereich; Leseauftrag bleibt dauerhaft aktiv

T10: Der Leseauftrag wird ausgeführt ("Auftrag aktiv" = TRUE)

T11: Datenträger C tritt in den Erfassungsbereich ein und die Daten werden gelesen ("Lesen erfolgreich" und "Transponder vorhanden" = TRUE); eingelesene Daten befinden sich im Eingangsfeld der Prozessdaten

T12: Der Datenträger C verlässt den Erfassungsbereich ("Lesen erfolgreich" und "Transponder vorhanden" = FALSE)

T13: Datenträger D tritt in den Erfassungsbereich ein

### Leseauftrag mit Autostart-Funktion

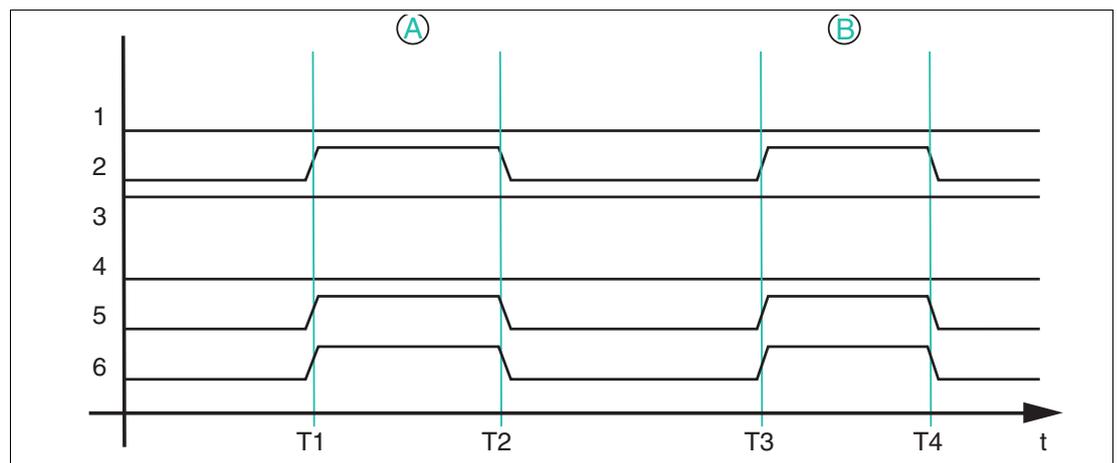


Abbildung 7.2 Zeitlicher Ablauf der Bits im Byte 0

- 1 Start Lesen
- 2 Lesen erfolgreich
- 3 Auftrag aktiv
- 4 Fehler
- 5 Transponder vorhanden
- 6 Daten (Eingang)

Bei der Autostart-Funktion wird ein Lesezugriff permanent ausgeführt, das Bit "Auftrag aktiv" der Eingangsprozessdaten ist permanent gesetzt.

T1: Datenträger A tritt in den Erfassungsbereich ein; "Lesen erfolgreich" und "Transponder vorhanden" werden auf TRUE gesetzt und die eingelesenen Daten befinden sich im Eingangsfeld der Prozessdaten

T2: Datenträger A verlässt den Erfassungsbereich; "Lesen erfolgreich" und "Transponder vorhanden" werden auf FALSE zurückgesetzt; der Bereich mit den zuvor eingelesenen Daten wird mit 0x00 aufgefüllt

T3: Datenträger B tritt in den Erfassungsbereich ein; identisches Verhalten zu T1

T4: Datenträger B verlässt den Erfassungsbereich; identisches Verhalten zu T2

## Schreibauftrag

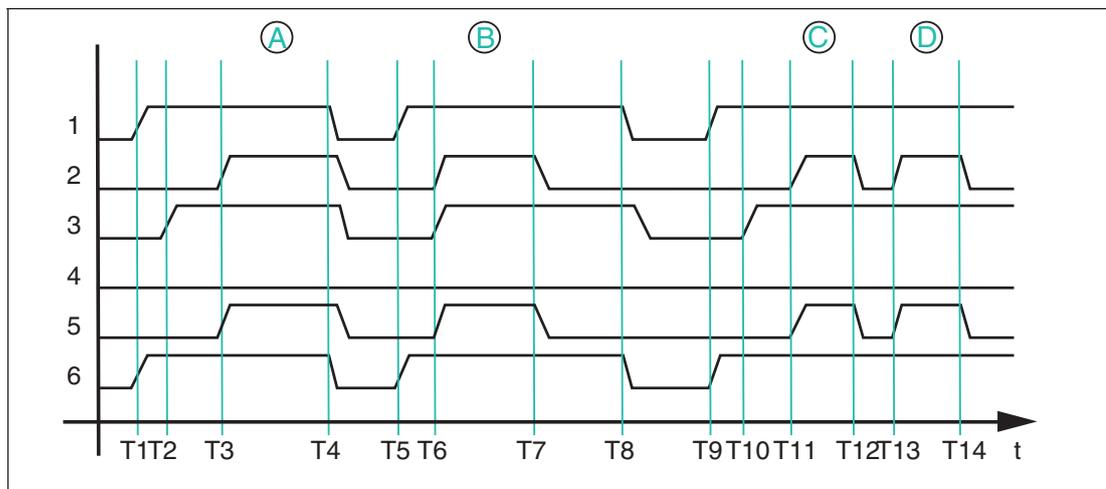


Abbildung 7.3 Zeitlicher Ablauf der Bits im Byte 0

- 1 Start Schreiben
- 2 Schreiben erfolgreich
- 3 Auftrag aktiv
- 4 Fehler
- 5 Transponder vorhanden
- 6 Daten (Eingang)

Die Ausführung eines Schreibauftrages kann nicht über die Autostart-Funktion erfolgen. Um einen Schreibauftrag zu starten, setzen Sie das Bit "Start Schreiben" auf TRUE.

T1: Starten des Schreibauftrags durch Setzen des Bits "Start Schreiben" auf TRUE; gleichzeitig sind die auf den Datenträger zu schreibenden Nutzdaten in das Ausgangsfeld der Prozessdaten zu übergeben

T2: Schreibauftrag ist aktiv ("Auftrag aktiv" = TRUE) und es befindet sich kein Datenträger im Erfassungsbereich ("Schreiben erfolgreich" = FALSE)

T3: Datenträger A tritt in den Erfassungsbereich ein und die Daten werden erfolgreich geschrieben ("Schreiben erfolgreich" und "Transponder vorhanden" = TRUE)

T4: Abbruch des Schreibauftrags durch Rücksetzen des Bits "Start Schreiben" auf FALSE; das Bit "Auftrag aktiv", Schreiben erfolgreich" und "Transponder vorhanden" werden auf FALSE zurückgesetzt und die Nutzdaten werden vom Anwender auf den Wert 0x00 zurückgesetzt

T5: Starten des Schreibauftrags durch Setzen des Bits "Start Schreiben" auf TRUE bei gleichzeitiger Übergabe der zu schreibenden Daten in das Ausgangsfeld der Prozessdaten; zum Zeitpunkt des Starts befindet sich Datenträger B im Erfassungsbereich

T6: Der Schreibauftrag ist aktiv ("Auftrag aktiv" = TRUE) und der Datenträger B wird erfolgreich beschrieben ("Schreiben erfolgreich" und "Transponder vorhanden" = TRUE)

T7: Datenträger B verlässt den Erfassungsbereich ("Schreiben erfolgreich" und "Transponder vorhanden" = FALSE); Schreibauftrag bleibt weiterhin aktiv ("Auftrag aktiv" = TRUE)

T8: Abbruch des Schreibauftrags durch Rücksetzen des Bits "Start Schreiben" auf FALSE; das Bit "Auftrag aktiv", "Schreiben erfolgreich" und "Transponder vorhanden" werden auf FALSE zurückgesetzt und die Nutzdaten werden vom Anwender auf den Wert 0x00 zurückgesetzt

T9: Starten des Schreibauftrags durch Setzen des Bits "Start Schreiben" auf TRUE; gleichzeitig werden die auf den Datenträger zu schreibenden Nutzdaten in das Ausgangsfeld der Prozessdaten übergeben

T10: Schreibauftrag ist aktiv ("Auftrag aktiv" = TRUE) und es befindet sich kein Datenträger im Erfassungsbereich ("Schreiben erfolgreich" = FALSE)

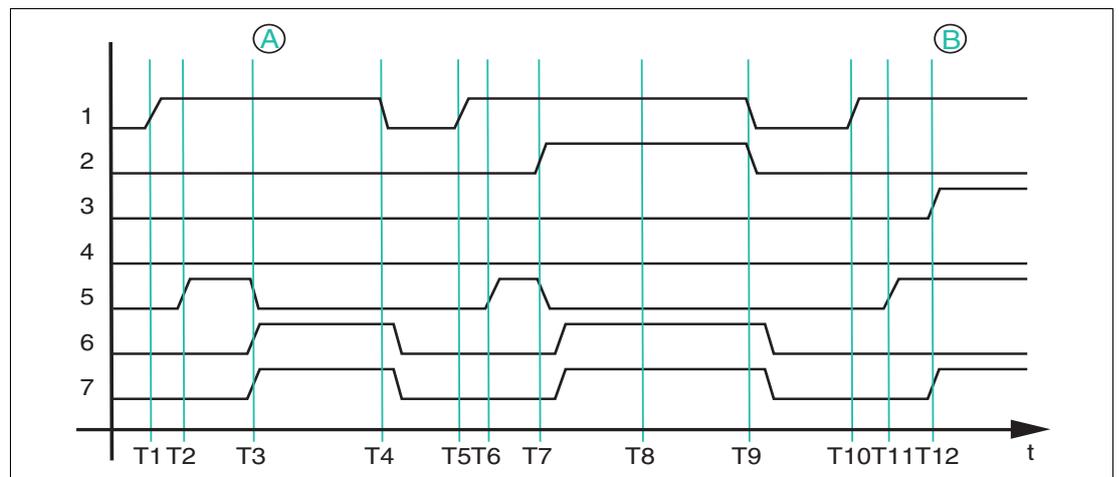
T11: Datenträger C tritt in den Erfassungsbereich ein und die Daten werden erfolgreich geschrieben ("Schreiben erfolgreich" und "Transponder vorhanden" = TRUE); Schreibauftrag weiterhin aktiv ("Auftrag aktiv" = TRUE)

T12: Datenträger C verlässt den Erfassungsbereich ("Schreiben erfolgreich" und "Transponder vorhanden" = FALSE); Schreibauftrag weiterhin aktiv ("Auftrag aktiv" = TRUE)

T13: Datenträger D tritt in den Erfassungsbereich ein und die Daten werden erfolgreich geschrieben ("Schreiben erfolgreich" und "Transponder vorhanden" = TRUE); Schreibauftrag weiterhin aktiv ("Auftrag aktiv" = TRUE)

T14: Datenträger D verlässt die Erfassungszone ("Lesen erfolgreich" und "Transponder vorhanden" = FALSE)

### Fehlerzustand



- 1 Start Lesen
- 2 Start Schreiben
- 3 Lesen erfolgreich
- 4 Schreiben erfolgreich
- 5 Auftrag aktiv
- 6 Fehler
- 7 Daten (Eingang)

Wenn bei der Ausführung eines Lese- oder Schreibauftrags ein Fehler auftritt, wird dieser Zustand über das Bit "Fehler" signalisiert. Im Bereich der Eingangsprozessdaten wird gleichzeitig eine Fehlermeldung übertragen.

T1: Starten des Leseauftrags durch Setzen des "Start Lesen" Bits auf TRUE

T2: Leseauftrag ist aktiv und wird ausgeführt ("Auftrag aktiv" = TRUE)

T3: Der Datenträger A mit 4 Byte Memory tritt in den Erfassungsbereich ein und das Bit "Fehler" wird auf TRUE gesetzt. Ebenso wird das Bit "Auftrag aktiv" auf FALSE zurückgesetzt und gleichzeitig in die Eingangsprozessdaten der Fehlercode 0x04 und der Text "invalid command" eingetragen. Dadurch wird angezeigt, dass der durch die IODD-Datei eingestellte Leseauftrag nicht zu den Eigenschaften des Datenträgers passt. Ursache dabei ist die Anzahl der einzulesenden Bytes. Für den Zugriff auf den Datenträger können maximal 4 Byte eingelesen werden. In diesem Beispiel ist der Wert 8 eingestellt. Um den Fehler zu beheben, korrigieren Sie den Wert innerhalb der IODD-Datei.

T4: Abbruch des Leseauftrages durch Rücksetzen des Bits "Start Lesen" auf FALSE; gleichzeitig wird das Bit "Fehler" auf FALSE zurückgesetzt und die Fehlermeldung in den Eingangsdaten gelöscht

T5: Start eines neuen Leseauftrages durch Setzen von Bit "Start Lesen" auf TRUE

T6: Leseauftrag ist aktiv und wird ausgeführt ("Auftrag aktiv" = TRUE)

T7: Es wird zusätzlich noch ein Schreibauftrag durch Setzen des Bits "Start Schreiben" auf TRUE gestartet. Das Bit "Fehler" wird gesetzt und "Auftrag aktiv" wieder zurückgesetzt auf FALSE. In das Eingangsfeld der Prozessdaten wird eine Fehlermeldung mit dem Fehlercode 0x04 und dem Text "read AND write set" übertragen. Dadurch wird angezeigt, dass ein Lese- und ein Schreibauftrag gleichzeitig angesteuert wurden. Dies ist für das Gerät nicht zulässig.

T8: Der Fehlerzustand ist weiterhin aktiv ("Fehler" = TRUE), da die Bits "Start Lesen" und "Start Schreiben" gesetzt sind

T9: Die Bits "Start Lesen" und "Start Schreiben" werden auf FALSE zurückgesetzt; gleichzeitig wird das Bit "Fehler" auf FALSE zurückgesetzt und die Fehlermeldung in den Eingangsdaten gelöscht

T10: Starten des Leseauftrages durch Setzen des Bits "Start Lesen" auf TRUE

T11: Leseauftrag ist aktiv und wird ausgeführt ("Auftrag aktiv" = TRUE)

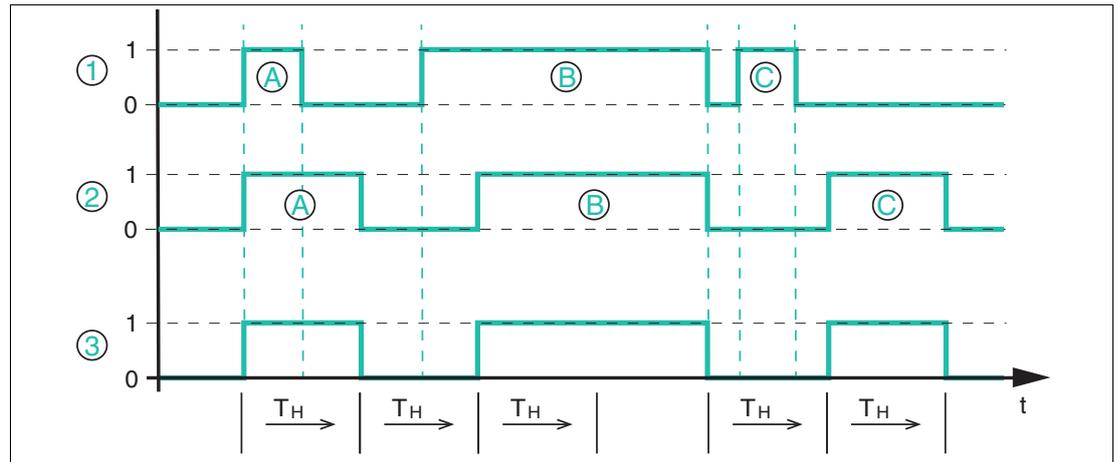
T12: Ein Datenträger B tritt in den Erfassungsbereich ein und wird erfolgreich gelesen ("Lesen erfolgreich" = TRUE)

## 7.2.4 Timing

Für die Datenübertragung im Easy-Mode verzichtet das Gerät auf das komplexe Handshake-Verfahren. Die Telegramme werden in die Eingangsprozessdaten eingestellt und bleiben dort für eine definierte Haltezeit bestehen. Innerhalb dieser Haltezeit kann das Gerät die Eingangsprozessdaten nicht ändern.

Die Haltezeit beträgt das Zehnfache der eingestellten Zykluszeit. Die Haltezeit ist mindestens 40 ms lang, da die kürzestmögliche Zykluszeit 4 ms beträgt.

Das Gerät kann innerhalb der Haltezeit ein neues Telegramm generieren, wenn ein neuer Datenträger gelesen wird oder ein Datenträger den Erfassungsbereich verlässt. Dieses Telegramm wird erst nach Ablauf der 40 ms in die Eingangsprozessdaten eingestellt. Wenn innerhalb der Haltezeit kein neues Telegramm auftritt, bleiben die Eingangsprozessdaten unverändert.



- 1 Datenträger im Feld
- 2 Daten
- 3 Lesen erfolgreich

Das Diagramm zeigt den prinzipiellen zeitlichen Ablauf der Datenübertragung in Abhängigkeit der Anwesenheit eines Datenträgers im Erfassungsbereich des Geräts.

" $T_H$ " entspricht der Haltezeit des Geräts von mindestens 40 ms.

Das Gerät wird dabei über die Autostart-Funktion oder durch das Bit "Start Lesen" angesteuert. Das Gerät führt dauerhaft einen Leseauftrag aus.

Zu Anfang tritt Datenträger A in den Erfassungsbereich des Geräts ein und das Bit "Lesen erfolgreich" in den Eingangsprozessdaten wechselt den Signalzustand auf "TRUE" (1). Der Datenträger hat eine Verweildauer von weniger als 40 ms im Erfassungsbereich und verlässt diesen kurz nach Eintritt wieder. Die Eingangsprozessdaten mit den Informationen zu Datenträger A bleiben für die Zeit " $T_H$ " (= 40 ms) erhalten. Erst nach Ablauf dieser Zeitspanne werden die Eingangsprozessdaten wieder aktualisiert und enthält nun die Information "Lesen erfolgreich" = FALSE (kein Datenträger) und signalisiert, dass der Datenträger den Erfassungsbereich verlassen hat. Auch dieses Telegramm bleibt für die Haltezeit von " $T_H$ " in den Eingangsprozessdaten bestehen.

Datenträger B tritt noch vor Ablauf der Haltezeit des vorhergehenden Telegramms in den Erfassungsbereich. Die Eingangsprozessdaten werden erst nach Ablauf der Haltezeit von 40 ms aktualisiert und das Bit "Lesen erfolgreich" wechselt auf "TRUE". Gleichzeitig werden die eingelesenen Daten in die Eingangsprozessdaten eingestellt. Datenträger B hat eine Verweildauer von mehr als 40 ms ( $> T_H$ ) im Erfassungsbereich des Geräts. Für diese Zeitspanne bleiben die Eingangsprozessdaten unverändert und das Bit "Lesen erfolgreich" hat weiterhin den Signalzustand "TRUE".

Datenträger B verlässt den Erfassungsbereich und in den Eingangsprozessdaten ändert sich der Signalzustand des Bits "Lesen erfolgreich" von 1 auf 0. Vor Ablauf der Haltezeit " $T_H$ " tritt Datenträger C in den Erfassungsbereich ein. Die Eingangsprozessdaten bleiben für die Zeitspanne der Haltezeit unverändert und wechseln nach Ablauf von " $T_H$ " den Signalzustand von "Lesen erfolgreich" auf "TRUE". Dadurch wird die Anwesenheit des Datenträgers C signalisiert und die eingelesenen Daten dieses Datenträgers werden übertragen.

Datenträger C verlässt den Erfassungsbereich vor Ablauf der Haltezeit " $T_H$ ". Nach Ablauf der Haltezeit der vorhergehenden Information (Datenträger B hat Erfassungsbereich verlassen) werden die Eingangsprozessdaten entsprechend geändert. Der Signalzustand von "Lesen erfolgreich" wechselt auf "TRUE".

## 7.3 Easy Mode mit PACTware

Sie können die RFID-Schreib-/Lesestation mit Hilfe des IO-Link-Masters "IO-Link-Master02-USB" in Betrieb nehmen.



### Inbetriebnahme mit PACTware



#### Hinweis!

Zur Bedienung des Systems verwenden Sie die Software PACTware Version 5.0.

Zur Einbindung der IODD-Dateien in die PACTware auf dem PC können Sie die Software-Tool IODD Interpreter DTM verwenden.

Sie finden die Software, die IODD-Datei und den Treiber auf der Homepage von Pepperl+Fuchs.

1. Schließen Sie die RFID-Schreib-/Lesestation an den IO-Link-Master an.
2. Schließen Sie den IO-Link-Master an eine Stromversorgung an.
3. Verbinden Sie den IO-Link-Master mit einem USB-Kabel mit einem PC.
4. Installieren Sie die beiden Software-Pakete auf ihrem PC.
5. Installieren Sie den Treiber **IO-Link USB Master DTM 2.0**.
6. Importieren Sie die IODD-Datei für die RFID-Schreib-/Lesestation mit dem Programm **IODD DTM Configurator**.

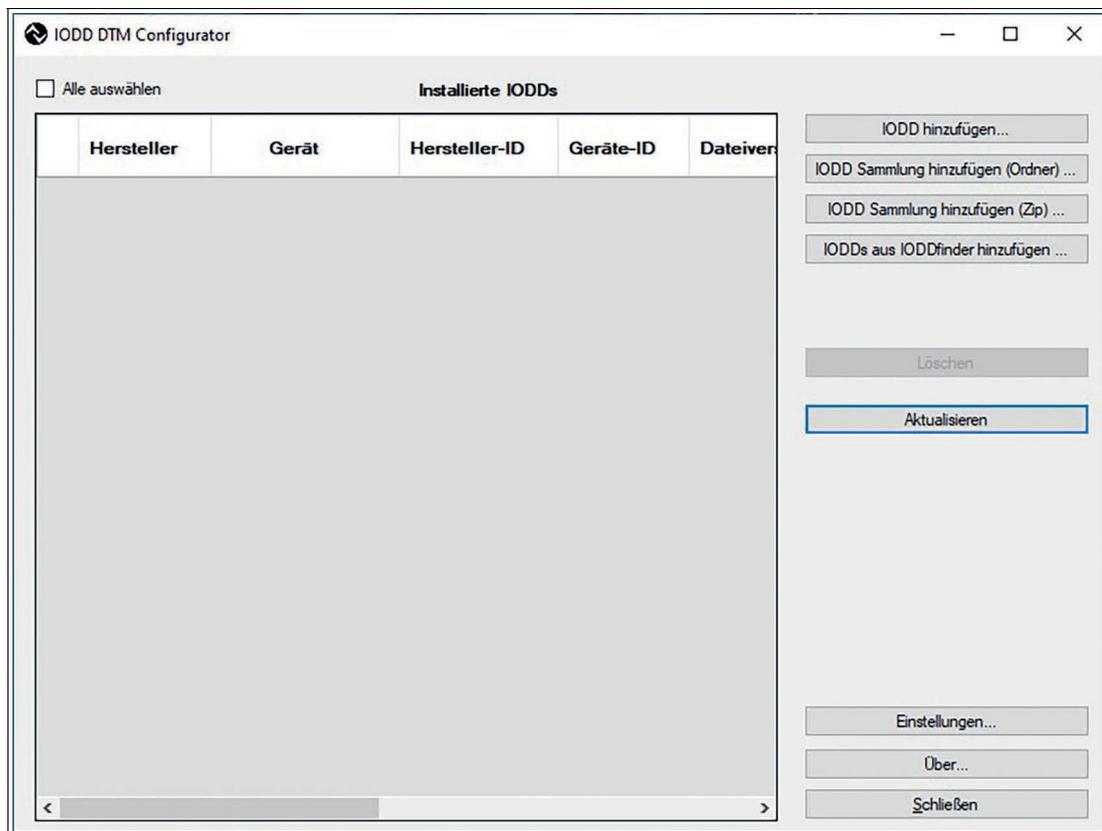


Abbildung 7.4

↳ Die erfolgreich hinzugefügten IODDs erscheinen in "Installierte IODDs".

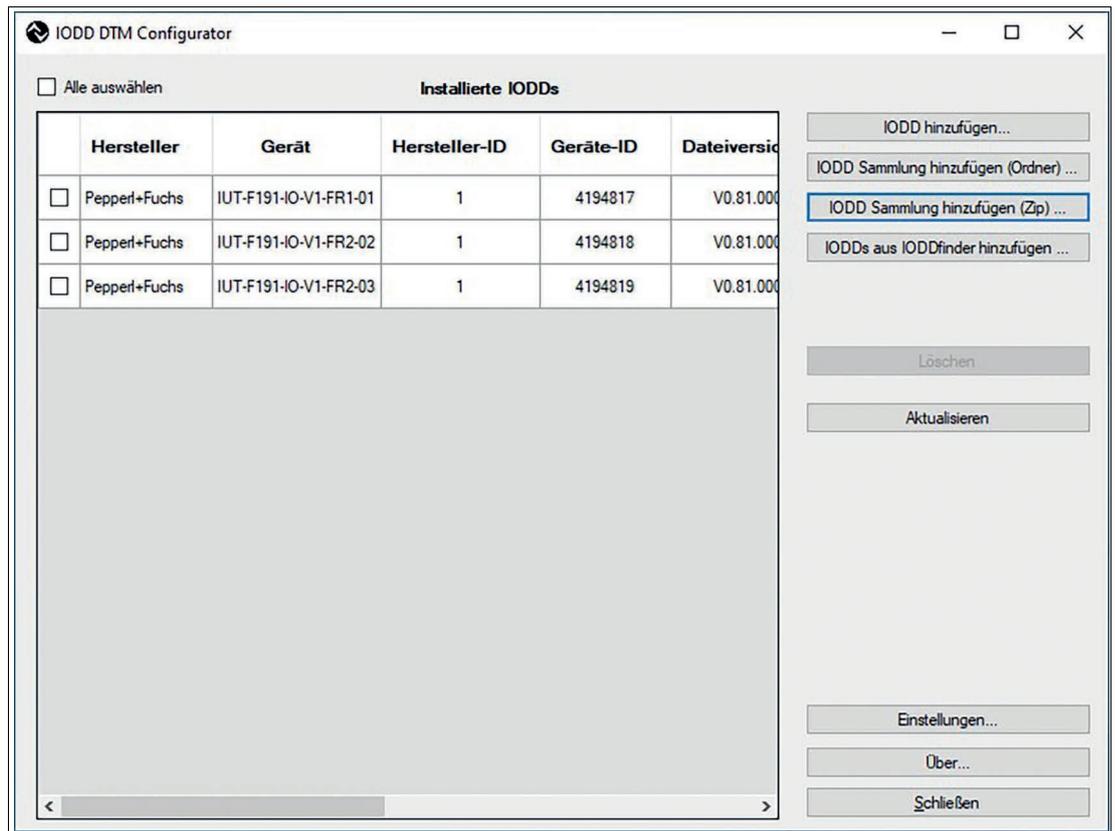


Abbildung 7.5

↳ Schließen Sie das Programm **IODD DTM Configurator**.

7. Starten Sie PACTware.
8. Klicken Sie mit der rechten Maustaste auf "HOST PC".

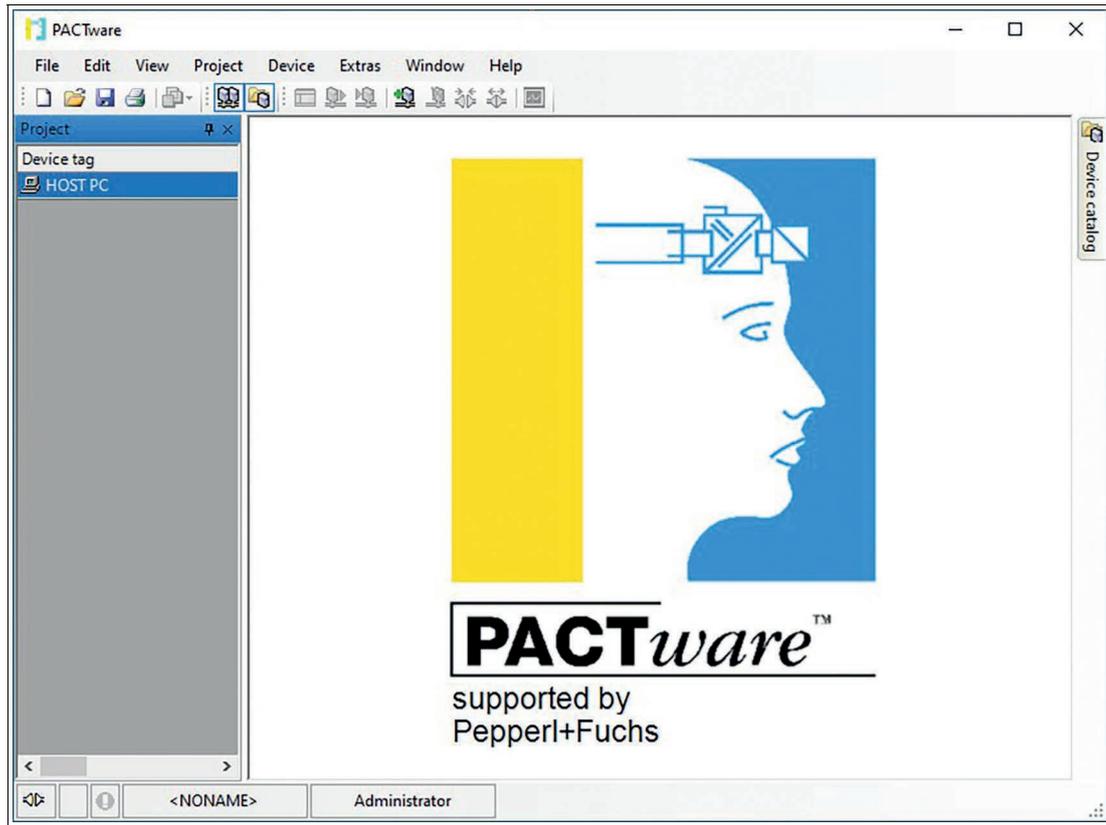


Abbildung 7.6

↳ Das Fenster "Device for" öffnet sich.

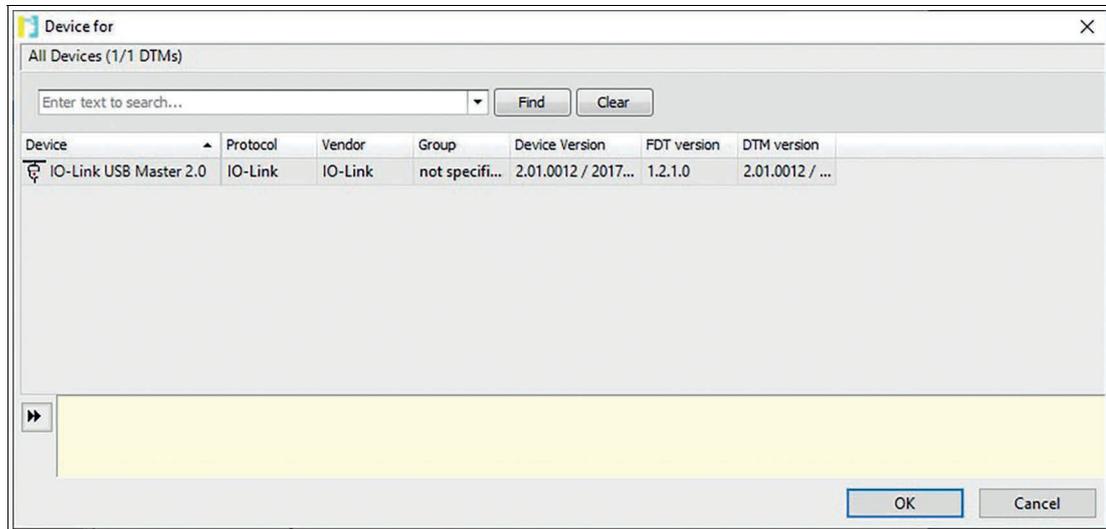


Abbildung 7.7

9. Wählen Sie den IO-Link USB Master 2.0 aus.



#### Hinweis!

Falls Sie einen anderen IO-Link Master verwenden, wählen Sie diesen aus.

10. Bestätigen Sie mit der Taste "OK".

↳ Der IO-Link-Master erscheint im Menü links unter Ihrem Projekt.

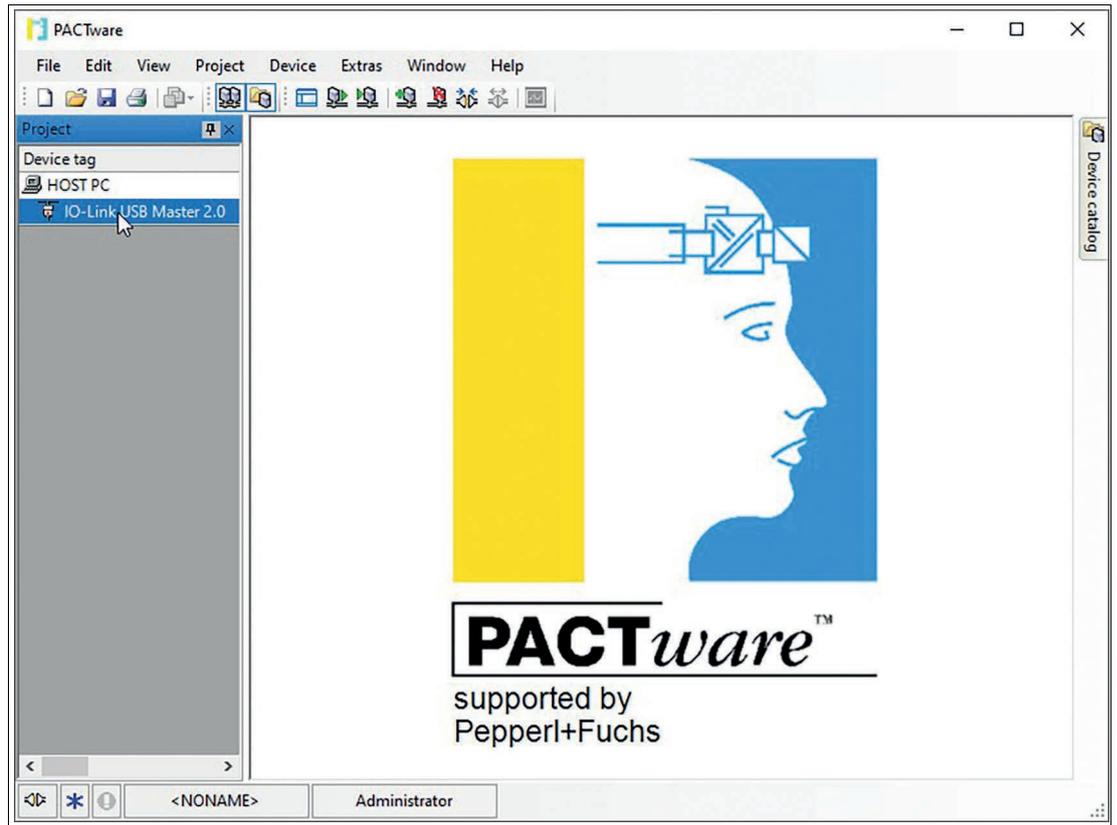


Abbildung 7.8

11. Klicken Sie mit der rechten Maustaste auf den IO-Link-Master.

↳ Das Fenster "Device for" öffnet sich.

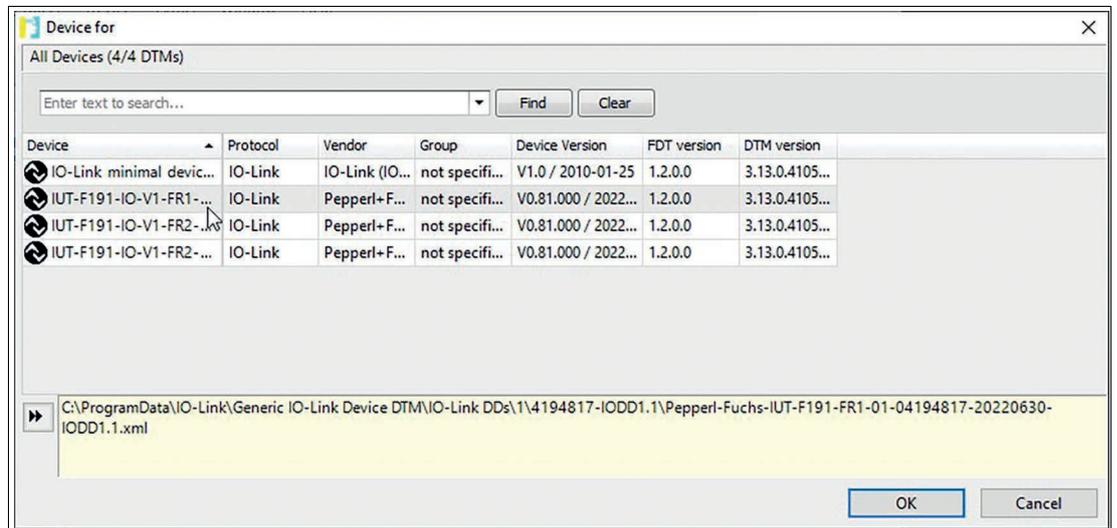


Abbildung 7.9

12. Wählen Sie die gewünschte RFID Schreib-/Lesestation aus.
13. Bestätigen Sie mit der Taste "OK".
14. Doppelklicken Sie auf das IO-Link-Gerät.

↳ Das Parameter-Menü öffnet sich.

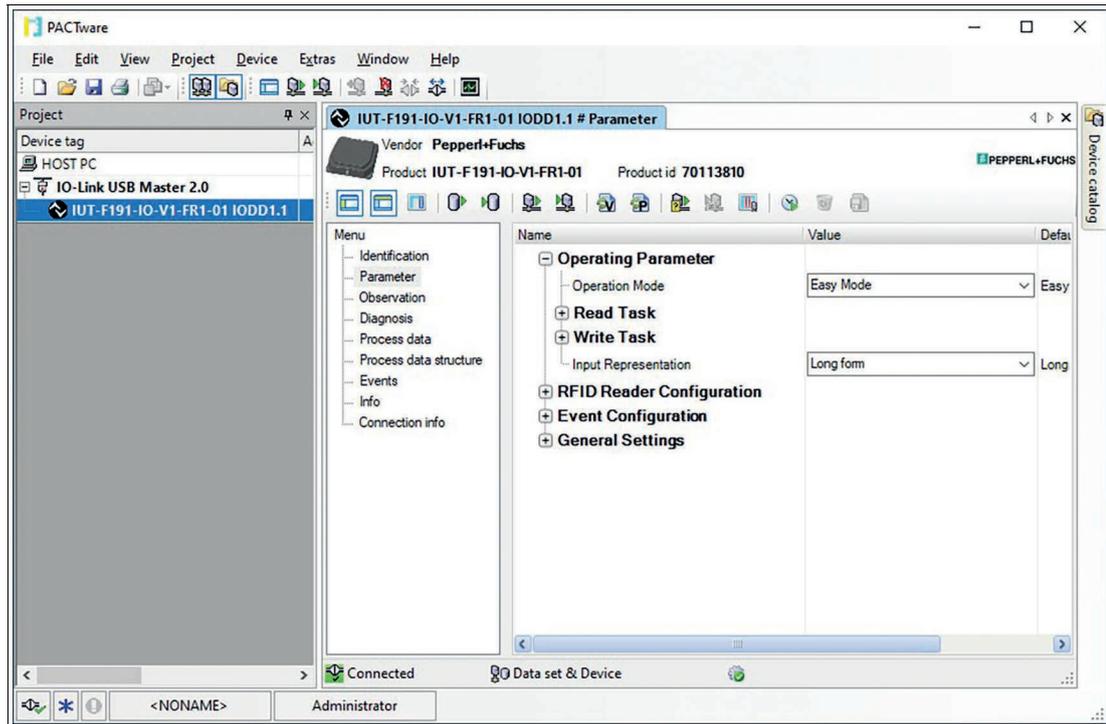


Abbildung 7.10

15. Passen Sie die Parameter des Geräts gemäß Ihrer Anwendung an. Siehe Kapitel 8.6.4.

↳ Eine Verbindung zwischen dem IO-Link-Master und dem Gerät wird hergestellt.

Sobald eine Verbindung besteht, beginnt die RFID-Schreib-/Lesestation automatisch mit dem Lesen von im Erfassungsbereich befindlichen Transpondern. Die Daten werden in den Eingangsprozessdaten angezeigt, entsprechend der eingestellten Parameter unter "Read Task".

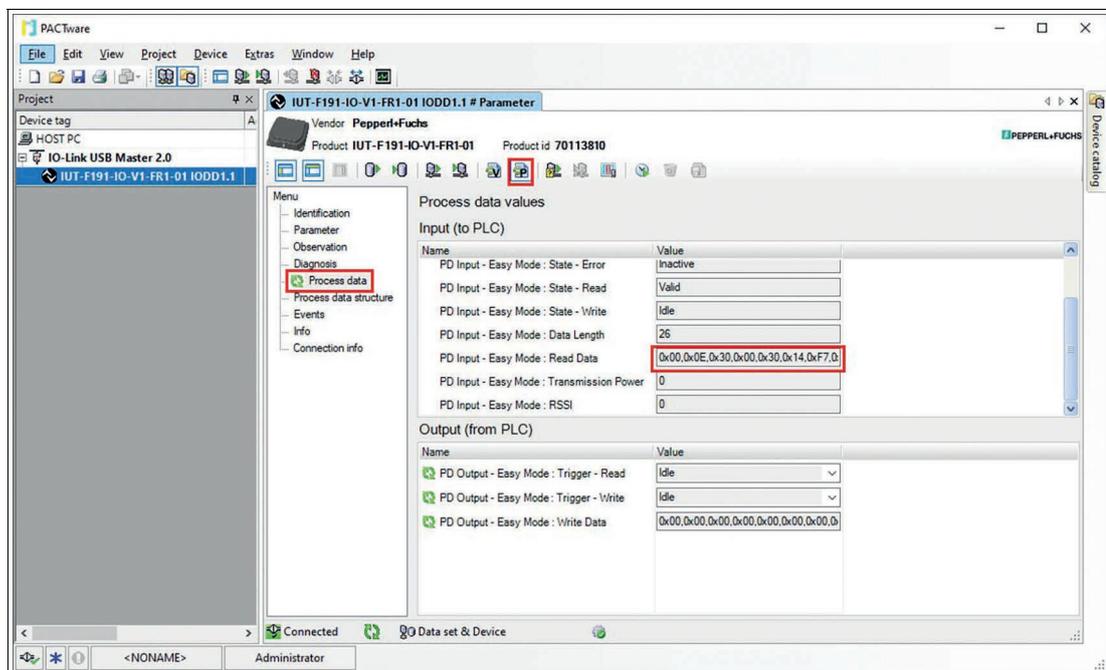


Abbildung 7.11



### Tipp

Aktivieren Sie die zyklische Aktualisierung der Prozessdaten, um die Daten in PACTware anzuzeigen.

## 8 ExpertMode

### 8.1 Grundlegender Befehlsablauf

Die Länge der Eingangs- und Ausgangsprozessdaten beträgt wie im Easy Mode 32 Byte, siehe Kapitel 7. Die Daten werden zunächst zu einem Telegramm zusammengefügt. Dieses Telegramm kann eine deutlich größere als die eingestellte Länge haben. Die Übertragung des Telegramms wird schrittweise über einzelne Fragmente durchgeführt. Die Größe eines Fragments beträgt 32 Byte. Die Steuerung der Datenübertragung wird über ein Handshake-Verfahren realisiert.

### 8.2 Legende

Name	Länge	Bedeutung
<Anzahl Datenträger>	4 Byte	Anzahl der während der Ausführung eines Single-Read- bzw. Write-Befehls identifizierten Datenträger. Die Kodierung der Anzahl der Datenträger erfolgt in ASCII Format
<ByteAddress>	2 Byte	Startadresse für den Lese- bzw. Schreibzugriff auf den User Memory des Datenträgers Bei der Ausführung von 4-Byte-Block-Befehlen muss der Wert ein Vielfaches von 4 sein. Bei der Ausführung der 2-Byte-Word-Befehle muss der Wert ein Vielfaches von 2 sein.
<Command>	1 Byte	Befehlscode Kennung des auszuführenden Befehls
<ControlByte>	4 Bit	Steuerbits zur Realisierung der Handshake-Prozedur bzw. Löschen des Gerätespeichers
<EPC/UII>	Länge variabel	EPC bzw. UII EPC/UII-Code des identifizierten Datenträgers; Länge ist abhängig von der Programmierung; alle EPC/UII innerhalb der Erfassungszone müssen unterschiedlich sein
<FragmentationCounter>	1 Byte	Fragmentierungszähler Anzahl der noch zu übertragenden Fragmente
<FrameLength>	12 Bit	Anzahl der gültigen Bytes innerhalb des Telegramms
<Length EPC/UII>	2 Byte	Länge EPC/UII-Information Längenangabe umfasst <PC-Word> und <EPC/UII>
<Length User Data>	2 Byte	Länge User Data Angabe der Länge der eingelesenen Daten von der Speicherbank User Memory
<Length TID>	2 Byte	Länge TID Längenangabe über die TID des eingelesenen Datenträgers
<LengthParameter>	2 Byte	Länge Parameter Angabe der Länge des Parameterdatensatzes, welches an das Gerät übertragen werden soll.
<Number of Bytes>	2 Byte	Anzahl der Bytes auf die während der Ausführung eines Lese- bzw. Schreibbefehls zugegriffen werden soll. Bei der Ausführung von 4-Byte-Block-Befehlen muss der Wert ein Vielfaches von 4 sein. Bei der Ausführung der 2-Byte-Word-Befehle muss der Wert ein Vielfaches von 2 sein.
<ParameterData>	Länge variabel	Parameterdaten Parameterdatensatz, welcher ausgelesen bzw. übertragen wurde.

Name	Länge	Bedeutung
<ParameterName>	2 Byte	Bezeichnung Geräteparameter Über die Bezeichnung des Geräteparameters wird festgelegt, auf welchen UHF-Parameter zugegriffen werden soll.
<PC-Word>	2 Byte	Protocol Control Beinhaltet zusätzliche Informationen zu der Kodierung des EPC/UII.
<Status>	1 Byte	Statusbyte Statusinformation über den Ausführungszustand des Befehls.
<SystemCode>	1 Byte	Systemkennzeichen Die Systemkennung für das UHF-System ist "U" (16#55).
<TelegramLength>	2 Byte	Telegrammlänge mit allen Fragmenten
<TID>	Länge variable	Tag Identification Eindeutige Seriennummer eines Datenträgers
<User Data>	Länge variabel	Eingelesene Daten Eingelesene Daten bei Lesezugriff auf die Speicherbank User Memory
<Write Data>	Länge variabel	Schreibdaten Datensatz, der auf den Datenträger übertragen werden soll.

### 8.3 Aufbau OUTPUT-Telegramm

#### OUTPUT-Telegramm

Byte	Inhalt	Bit Nummer							
		7	6	5	4	3	2	1	0
0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
1	Frame Length	<Frame Length>							
2	Fragmentation Counter	<Fragmentation Counter>							
3	Telegram Length (High Byte) <sup>1</sup>	<Telegram Length (High Byte)>							
4	Telegram Length (Low Byte) <sup>1</sup>	<Telegram Length (Low Byte)>							
5	Command <sup>1</sup>	<Command>							
6	Parameter / Data	<Data Byte 1>							
7	Parameter / Data	<Data Byte 2>							
8	Parameter / Data	...							
...	Parameter / Data	<Data Byte X>							
...	Parameter / Data	16#00							
...	Parameter / Data	...							
31	Parameter / Data	16#00							

Tabelle 8.1

1. Ab zweitem Fragment werden ab Byte 3 Parameter / Daten übertragen.

Der Wert von <Frame Length> ist abhängig davon, wie viele Datenwerte <Data Byte> für die Ausführung eines Befehls mit übertragen werden müssen. Es wird hierdurch die Länge des Fragments bis einschließlich <Data Byte X> festgelegt. Wenn für die Ausführung des Befehls keine zusätzlichen Befehlsparameter erforderlich sind, erstreckt sich die Länge des Fragments bis <Command> und hat den Wert 16#06.

Der <Fragmentation Counter> hat den Wert 16#00, da der Befehl über ein Fragment von der Steuerung übertragen werden kann.

Die <Telegram Length> spezifiziert die Länge des Telegramms beginnend ab der Telegrammlänge selbst und einschließlich des <Data Byte X> Bytes. Wenn keine weiteren Befehlsparameter mit übertragen werden, endet das Telegramm bei <Command> und <Telegram Length> hat den Wert 16#03.

Das Byte <Command> legt den auszuführenden Befehl fest. In Abhängigkeit des Werts in <Command> werden unterschiedliche Befehle ausgeführt. Es gibt folgende Klassen von Befehlen:

- **Schreib-/Lesebefehle:** Zugriff auf einen oder mehrere Datenträger in der Erfassungszone
- **Systembefehle:** Durchführung von Geräteeinstellungen; kein Zugriff auf Datenträger
- **Filterbefehle:** Einstellung von Filtern für den Zugriff auf Datenträger
- **UHF-Konfigurationsbefehle:** Einstellung der UHF-Eigenschaften des Geräts

Über <Data Byte> werden für die Ausführung eines Befehls erforderliche Daten mit übertragen. Dabei kann es sich um zusätzliche Befehlsparameter (z. B. Startadresse) handeln oder Anwenderdaten die auf einen Datenträger geschrieben werden sollen.

Die innerhalb des Telegrammrahmens ungenutzten Bereiche werden auf den Wert 16#00 gesetzt.

## 8.4 Aufbau INPUT-Telegramm

### INPUT-Telegramm

Byte	Inhalt	Bit Nummer							
		7	6	5	4	3	2	1	0
0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
1	Frame Length	<Frame Length>							
2	Fragmentation Counter	<Fragmentation Counter>							
3	Telegram Length (High Byte) <sup>1</sup>	<Telegram Length (High Byte)>							
4	Telegram Length (Low Byte) <sup>1</sup>	<Telegram Length (Low Byte)>							
5	Command <sup>1</sup>	<Command>							
6	Status	<Status>							
7	Parameter / Data	<Data Byte 1>							
8	Parameter / Data	<Data Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<Data Byte X>							
...	Parameter / Data	0x00							
...	Parameter / Data	...							
31	Parameter / Data	0x00							

Tabelle 8.2

1. Ab zweitem Fragment werden ab Byte 3 Parameter / Daten übertragen.

Der Wert von <Frame Length> ist abhängig davon, wie viele Datenwerte <Data Byte> von dem Gerät innerhalb der Befehlsantwort mit zurückgesendet werden. Es wird hierdurch die Länge des Fragments bis einschließlich <Data Byte X> angegeben. Wenn innerhalb der Befehlsantwort keine zusätzlichen Datenwerte enthalten sind, erstreckt sich die Länge des Fragments bis <Status> und hat den Wert 16#07.

Da die Befehlsantwort über ein Fragment von der Steuerung übertragen werden kann, hat der <Fragmentation Counter> den Wert 16#00.

Die <Telegram Length> spezifiziert die Länge des Telegramms beginnend ab der Telegrammlänge selbst und einschließlich des <Data Byte X> Bytes. Wenn keine weiteren Antwortparameter mit übertragen werden, endet das Telegramm bei <Status> und <Telegram Length> hat den Wert 16#04.

Das Byte <Command> ist die Spiegelung des Befehlscodes aus dem Befehl innerhalb der Rückantwort.

Über den Wert innerhalb von <Status> lässt sich der Zustand der Befehlsausführung erkennen. Es werden Fehlerzustände durch entsprechende Statuswerte signalisiert. Siehe Kapitel 8.7.

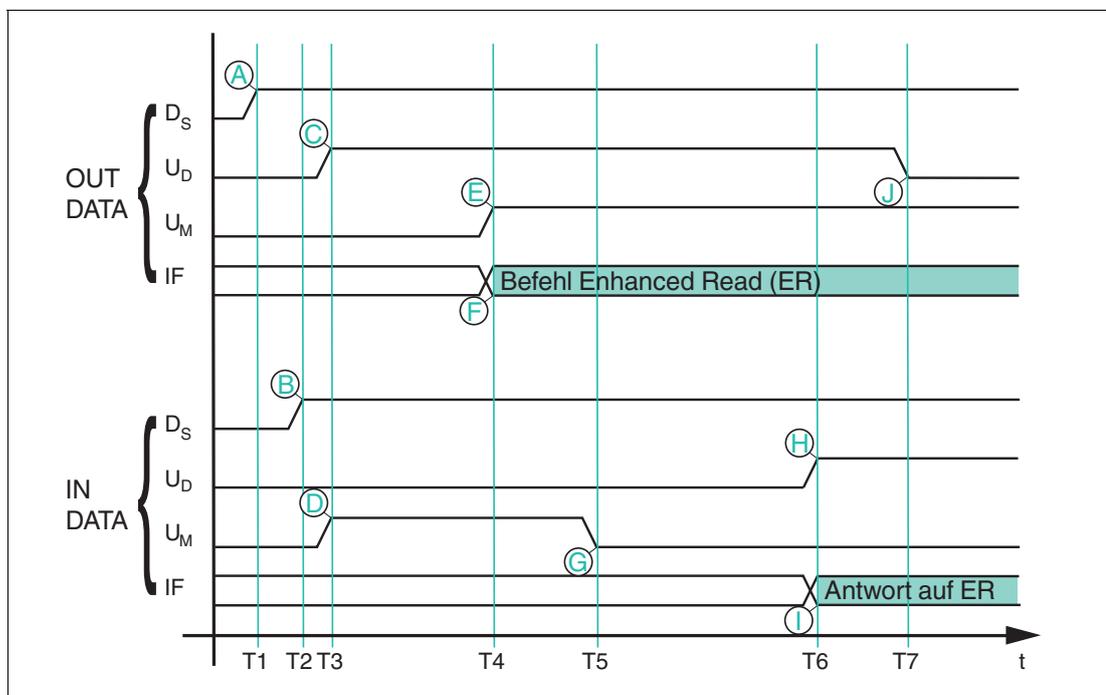
In den Bytes <Data Byte> werden die vom Gerät auf Grund der Befehlsansteuerung zu übertragenden Daten zurückgesendet. Es kann sich dabei beispielsweise um von einem Datenträger eingelesene Daten handeln oder Parameterwerte der UHF-Einstellungen.

Die innerhalb des Antworttelegramms ungenutzten Bereiche werden auf den Wert 16#00 gesetzt.

## 8.5 Handshake-Verfahren

Um eine kontinuierliche Datenübermittlung ohne Verluste zu gewährleisten, muss der Datenfluss zwischen einer SPS und der Schreib-/Lesestation synchronisiert werden. Die Eingangs- und Ausgangsprozessdaten werden zyklisch übertragen. Die Steuerung des Datenflusses über die Software wird auch als "Handshaking" bezeichnet. Die hierfür notwendigen Steuerbits sind im Steuerbyte enthalten (siehe Kapitel 8.4 und siehe Kapitel 8.3).

Um Telegramme sicher und schnell zwischen SPS und Gerät zu übermitteln, wird folgendes Handshake-Verfahren eingesetzt:



**D<sub>S</sub>** Delete Bit; löscht internen Speicher des Geräts

**U<sub>M</sub>** Update Master-Bit

**U<sub>D</sub>** Update Device-Bit

- T1** Die SPS wechselt das Deletebit  $D_S$  in den Ausgangsprozessdaten auf High (**A**) und veranlasst damit das Löschen des FIFO-Speichers im Gerät
- T2** Das Schreib-/Lesegerät wechselt als Antwort das Deletebit in den Eingangsprozessdaten (**B**) und löscht den gesamten Inhalt des FIFO-Speichers.
- T3** Die SPS spiegelt den invertierten Zustand von  $U_D$ -INPUT aus den Eingangs- in die Ausgangsprozessdaten (**C**). In gleicher Weise spiegelt das Gerät den invertierten Zustand von  $U_M$ -OUTPUT in die Eingangsprozessdaten (**D**). Damit signalisieren beide Kommunikationspartner die Bereitschaft, ein Telegramm zu empfangen.
- T4** Die SPS trägt einen Enhanced Read Befehl (ER) in IF-OUT (Ident Frame) ein (F). Gleichzeitig übernimmt die SPS  $U_M$ -INPUT in  $U_M$ -OUTPUT (**E**) und signalisiert damit die Gültigkeit eines neuen Telegramms.
- T5** Das Gerät spiegelt den invertierten Zustand aus  $U_M$ -OUTPUT in  $U_M$ -INPUT (**G**). Damit wird die Steuerung über den Empfang des Telegramms informiert.
- T6** Das Gerät hat den Befehl ER bearbeitet und trägt die Antwort auf den Befehl in die Eingangsprozessdaten ein (I). Im selben Telegramm wird  $U_D$ -OUTPUT in  $U_D$ -INPUT gespiegelt (**H**).
- T7** Die SPS hat das geänderte  $U_D$ -INPUT empfangen und spiegelt den invertierten Zustand in  $U_D$ -OUTPUT (**J**). Erst jetzt kann das Gerät ein weiteres Telegramm senden.

## Beispielimplementierung in der Steuerung

### Deletebit Device $D_S$ :

Nachdem das Gerät betriebsbereit ist (IO-Link-Kommunikation = OK), ist diese Anweisung einmalig durchzuführen. Es wird hierdurch der interne Telegrammspeicher gelöscht. Wenn ein interner Gerätefehler aufgetreten ist, sollte der interne Telegrammspeicher ebenfalls gelöscht werden.

```
DS_OUTPUT := NOT DS_INPUT
```

### Updatebit Device $U_D$ :

Wenn  $U_D$ -Bits in den Eingangs- und Ausgangsprozessdaten den gleichen Wert haben, liegen in den Eingangsprozessdaten neue gültige Daten. Erst wenn die Steuerung die Eingangsprozessdaten gelesen hat, d. h. das  $U_D$ -Bit in den Eingangs- und Ausgangsprozessdaten einen invertierten Signalzustand vorweist, schreibt das Gerät neue Lesedaten in die Eingangsprozessdaten.

Um die Übertragung der Lesedaten nicht zu blockieren, muss in jedem Zyklus der invertierte Zustand des  $U_D$ -Bit von den Eingangsprozessdaten an das  $U_D$ -Bit der Ausgangsprozessdaten übertragen werden.

```
UD_OUTPUT := NOT UD_INPUT (* kopiere das invertierte Updatebit aus dem INPUT-Telegramm in das OUTPUT-Telegramm *)
```

### Updatebit Master $U_M$ :

Bevor ein Befehl gesendet wird, muss geprüft werden, ob das Gerät (Device) für neue Telegramme empfangsbereit ist. Dies ist der Fall, wenn das Bit  $U_M$  in den Eingangs- und im Ausgangsprozessdaten einen invertierten Signalzustand vorweist.

Anschließend sind die Befehlsparameter an die zugehörigen Positionen innerhalb der Ausgangsprozessdaten zu übertragen.

Sobald das Bit  $U_M$  in den Ausgangsprozessdaten auf den gleichen Signalzustand wie das Bit  $U_M$  in den Eingangsprozessdaten gesetzt wird, überträgt die Steuerung den neuen Befehl an das Gerät.

```
OUTPUT[1..x] := Neues Telegramm
```

```
IF (U_M_OUTPUT <> U_M_INPUT) then (* Prüfe, ob Device neue Daten empfangen kann *)
```

```
U_M_OUTPUT := U_M_INPUT (* Device ist empfangsbereit, übernehme Updatebit *)
```

```
End_IF
```

## 8.6 Befehlsübersicht

Die in der Liste aufgeführten Befehle sind auf den folgenden Seiten ausführlich beschrieben.

### Schreib-/Lesebefehle

Kürzel	Befehls-code	Befehlsbeschreibung
SF	16#01	Single Read Fixcode siehe "Single Read Fixcode (SF)" auf Seite 59 Einmaliger Lesezugriff auf den EPC/UII (Bank 01) + TID (Bank 10)
EF	16#1D	Enhanced Read Fixcode siehe "Enhanced Read Fixcode (EF)" auf Seite 62 Permanenter Lesezugriff auf den EPC/UII (Bank 01) + TID (Bank 10)
SN	16#D2	Single Read EPC/UII siehe "Single Read EPC/UII (SN)" auf Seite 64 Einmaliger Lesezugriff auf den EPC/UII (Bank 01)
EN	16#D3	Enhanced Read EPC/UII siehe "Enhanced Read EPC/UII (EN)" auf Seite 66 Permanenter Lesezugriff auf den EPC/UII (Bank 01)
#SU	16#CE	Single Write EPC/UII siehe "Single Write EPC/UII (#SU)" auf Seite 68 Einmaliger Schreibzugriff auf den EPC/UII (Bank 01)
SR	16#10	Single Read 4-Byte Blocks siehe "Single Read 4-Byte Blocks (SR)" auf Seite 72 Einmaliger Lesezugriff auf EPC/UII (Bank 01) + 4-Byte Datenblöcke User Memory (Bank 11)
ER	16#19	Enhanced Read 4-Byte Blockssiehe "Enhanced Read 4-Byte Blocks (ER)" auf Seite 74 Permanenter Lesezugriff auf EPC/UII (Bank 01) + 4-Byte Datenblöcke User Memory (Bank 11)
SW	16#40	Single Write 4-Byte Blocks siehe "Single Write 4-Byte Blocks (SW)" auf Seite 77 Einmaliger Schreibzugriff auf 4-Byte Datenblöcke User Memory (Bank 11)
EW	16#1A	Enhanced Write 4-Byte Blocks siehe "Enhanced Write 4-Byte Blocks (EW)" auf Seite 80 Permanenter Schreibzugriff auf 4-Byte Datenblöcke User Memory (Bank 11)
#SR	16#49	Single Read 2-Byte Words siehe "Single Read 2-Byte Words (#SR)" auf Seite 82 Einmaliger Lesezugriff auf EPC/UII (Bank 01) + 2-Byte Datenwörter User Memory (Bank 11)
#ER	16#4B	Enhanced Read 2-Byte Words siehe "Enhanced Read 2-Byte Words (#ER)" auf Seite 85 Permanenter Lesezugriff auf EPC/UII (Bank 01) + 2-Byte Datenwörter User Memory (Bank 11)
#SW	16#4A	Single Write 2-Byte Words siehe "Single Write 2-Byte Words (#SW)" auf Seite 88 Einmaliger Schreibzugriff auf 2-Byte Datenwörter User Memory (Bank 11)
#EW	16#4C	Enhanced Write 2-Byte Words siehe "Enhanced Write 2-Byte Words (#EW)" auf Seite 90 Permanenter Schreibzugriff auf 2-Byte Datenwörter User Memory (Bank 11)

2024-08

Kürzel	Befehls-code	Befehlsbeschreibung
LO	16#D5	Lock siehe "Lock (LO)" auf Seite 93 Einmaliger Schreibzugriff auf einen oder mehrere Transponder
KI	16#B9	Kill siehe "Kill (KI)" auf Seite 96 Setzt Transponder in einen Zustand, in dem kein Zugriff mehr möglich ist.

### Systembefehle

Kürzel	Befehls-code	Befehlsbeschreibung
QU	16#02	Quit siehe "Quit (QU)" auf Seite 98 Abbruch eines aktiven Enhanced Befehls
VE	16#03	Version siehe "Version (VE)" auf Seite 99 Auslesen der Firmware Version

### Filterbefehle

Kürzel	Befehls-code	Befehlsbeschreibung
FI	16#CA	Set Filter Mask siehe "Set Filter Mask (FI)" auf Seite 101 Einstellung eines Filters
MF	16#CB	Activate Filter Mask siehe "Activate Filter Mask (MF)" auf Seite 106 Filter aktivieren

### Konfigurationsbefehle

Kürzel	Befehls-code	Befehlsbeschreibung
RP	16#BE	Read Parameter siehe "Read Parameter (RP)" auf Seite 109 Geräteparameter auslesen
WP	16#BF	Write Parameter siehe "Write Parameter (WP)" auf Seite 111 Geräteparameter setzen

#### 8.6.1

### Schreib-/Lesebefehle

Den folgenden Schreib-/Lesebefehlen liegt die Speicherstruktur des Transponders nach EPC Gen 2 (ISO/IEC 18000-63) zu Grunde.

Bei der Beschreibung der nachfolgenden Befehle und Antworten wird das Long-Form-Datenformat verwendet. Dadurch befindet sich in der Antwort immer der EPC/UII sowie zusätzliche Längeninformationen. Das Ende eines Single-Befehls wird über ein Status-16#0F-Telegramm zurückgemeldet. Bei der Verwendung des Short-Form-Datenformat entfällt der EPC/UII und die zusätzlichen Längeninformationen. Das Ende eines Single-Befehls wird nicht zurückgemeldet.

#### Single Read Fixcode (SF)

Der Befehl "Single Read Fixcode" hat den Befehlscode 16#01 und führt einen einmaligen Lesevorgang auf den EPC/UII (Bank 01) und Fixcode (TID; Bank 10) eines oder mehrerer Transponder innerhalb der Erfassungszone durch. Die eingelesenen Informationen werden für jeden gelesenen Transponder in einem separaten Datentelegramm mit den Statuswert 16#00 übertragen. Das Ende der Befehlsausführung wird über ein Abschlusstelegramm signalisiert. Dieses Telegramm hat den Statuswert 16#0F und enthält die Anzahl der während der Befehlsausführung identifizierten Transponder.

**Befehl:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.3

Die <FrameLength> hat den Wert 16#06, da keine weiteren Befehlsparameter übertragen werden müssen und das Fragment nach <Command> endet. Der <FragmentationCounter> hat den Wert 16#00, da keine zusätzlichen Fragmente für die Übertragung des Befehlstelegramms erforderlich sind. Das Befehlstelegramm hat eine Länge (<TelegramLength>) von 16#03, da der Befehl mit dem Byte <Command> endet.

Der Befehlscode <Command> für den Befehl Single Read Fixcode ist 16#01.

**Antwort Datentelegramm, Long-Form-Datenformat:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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>							

2024-08

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.4

Die Länge des Fragmentes <FrameLength> ist abhängig von der Größe des EPC/UII und der TID des eingelesenen Datenträgers. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <TID Byte Y>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII und der TID des eingelesenen Datenträgers. Die Größe des Telegrams erstreckt sich bis einschließlich <TID Byte Y>.

Der Parameter <Command> hat den Wert 16#01 und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Diese hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII zur eindeutigen Identifizierung des gelesenen Transponders.

Nach dem EPC/UII folgt eine Längenangabe für die TID. Diese hat immer eine Größe von 2 Byte. Abschließend folgt die TID des identifizierten Transponders bis einschließlich <TID Byte Y>. Alle nachfolgenden Bytes innerhalb des Datentelegramms haben den Wert 16#00.

#### Antwort Befehlsende, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	<Anzahl Datenträger Byte 1>							
Byte 8	Parameter / Data	<Anzahl Datenträger Byte 2>							
Byte 9	Parameter / Data	<Anzahl Datenträger Byte 3>							
Byte 10	Parameter / Data	<Anzahl Datenträger Byte 4>							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.5

Die <FrameLength> ist in der Rückantwort für das Befehlsende des Single Read Fixcode konstant bei den Wert 16#0B. Das Fragment erstreckt sich dabei bis einschließlich <Anzahl Datenträger Byte 4>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die Telegrammlänge <TelegramLength> hat den Wert 16#08. Das Byte <Command> wird zurückgespiegelt und hat denselben Wert wie im Befehlstelegramm 16#01.

Der <Status> für das Telegramm zur Signalisierung des Befehlsendes ist 16#0F.

Die Anzahl der identifizierten Datenträger wird innerhalb von 4 Byte übertragen. Dabei wird die Anzahl in ASCII kodierter Darstellung wiedergegeben.

Bei der Identifikation von einem Transponder hat die <Anzahl Datenträger> den Wert "0001" (ASCII) bzw. 16#30303031.

Wenn während der Befehlsausführung kein Datenträger erkannt wurde, entfallen die Datentelegramme und es wird nur das Telegramm zur Signalisierung des Befehlsendes gesendet. Die <Anzahl Datenträger> hat dabei den Wert "0000" (ASCII) bzw. 16#30303030.

### Enhanced Read Fixcode (EF)

Der Befehl "Enhanced Read Fixcode" hat den Befehlscode 16#1D und führt einen permanenten Lesevorgang auf den EPC/UII (Bank 01) und den Fixcode (TID; Bank 10) eines oder mehrerer Transponder innerhalb der Erfassungszone durch. Die eingelesenen Informationen werden für jeden gelesenen Transponder in einem separaten Datentelegramm mit den Statuswert 16#00 übertragen. Wenn ein Transponder die Erfassungszone verlässt, wird dies über ein Telegramm mit dem EPC/UII des Transponders gemeldet. Dieses Telegramm hat den Statuswert 16#05. Die Befehlsausführung wird über ein Quit-Befehl gestoppt.

#### Befehl:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#06							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#03							
Byte 5	Command	16#1D							
Byte 6	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.6

Die <FrameLength> hat den Wert 16#06, da keine weiteren Befehlsparameter übertragen werden müssen und das Fragment nach <Command> endet. Der <FragmentationCounter> hat den Wert 16#00, da keine zusätzlichen Fragmente für die Übertragung des Befehlsstelegramms erforderlich sind. Das Befehlsstelegramm hat eine Länge (<TelegramLength>) von 16#03, da der Befehl mit dem Byte <Command> endet.

Der Befehlscode <Command> für den Befehl Enhanced Read Fixcode ist 16#1D.

#### Antwort Datentelegramm, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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)>							

2024-08

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 8	Parameter / Data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter / Data	<PC-Word (High Byte)>							
Byte 10	Parameter / Data	<PC-Word (Low Byte)>							
Byte 11	Parameter / Data	<EPC / UII Byte 1>							
Byte 12	Parameter / Data	<EPC / UII Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<EPC / UII Byte X>							
...	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>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.7

Die Länge des Fragments <FrameLength> ist abhängig von der Größe des EPC/UII und der TID des eingelesenen Datenträgers. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <TID Byte Y>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII und der TID des eingelesenen Datenträgers. Die Größe des Telegrams erstreckt sich bis einschließlich <TID Byte Y>. Der Parameter <Command> hat den Wert 16#1D und wird innerhalb des Datentelegramms zurückgespiegelt. Der <Status> Parameter hat den Wert 16#00.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Diese hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII zur eindeutigen Identifizierung des gelesenen Transponders.

Nach dem EPC/UII folgt eine Längenangabe für die TID. Diese hat immer eine Größe von 2 Byte. Abschließend folgt die TID des identifizierten Transponders bis einschließlich <TID Byte Y>. Alle nachfolgenden Bytes innerhalb des Datentelegramms haben den Wert 16#00.

#### Antwort Transponder hat Erfassungszone verlassen, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 8	Parameter / Data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter / Data	<PC-Word (High Byte)>							
Byte 10	Parameter / Data	<PC-Word (Low Byte)>							
Byte 11	Parameter / Data	<EPC / UII Byte 1>							
Byte 12	Parameter / Data	<EPC / UII Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<EPC / UII Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.8

Die Länge des Fragments <FrameLength> ist abhängig von der Größe des EPC/UII des Datenträgers, der den Erfassungsbereich verlässt. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <EPC/UII Byte X>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII des Datenträgers. Die Größe des Telegramms erstreckt sich bis einschließlich <EPC/UII Byte X>.

Der Parameter <Command> hat den Wert 16#1D und wird innerhalb des Telegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#05.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Diese hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII zur eindeutigen Identifizierung des Transponders, der den Erfassungsbereich verlassen hat. Alle nachfolgenden Bytes innerhalb des Telegramms haben den Wert 16#00.

### Single Read EPC/UII (SN)

Der Befehl "Single Read EPC/UII" hat den Befehlscode 16#D2 und führt einen einmaligen Lesevorgang auf den EPC/UII (Bank 01) eines oder mehrerer Transponder innerhalb der Erfassungszone durch. Die eingelesenen Informationen werden für jeden gelesenen Transponder in einem separaten Datentelegramm mit den Statuswert 16#00 übertragen. Das Ende der Befehlsausführung wird über ein Abschlusstelegramm signalisiert. Dieses Telegramm hat den Statuswert 16#0F und enthält die Anzahl der Transponder, die während der Befehlsausführung identifiziert wurden.

#### Befehl:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#06							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#03							
Byte 5	Command	16#D2							
Byte 6	Nicht relevant	16#00							
Byte 7	Nicht relevant	16#00							
...	Nicht relevant	16#00							

2024-08

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 31	Nicht relevant	16#00							

Tabelle 8.9

Die <FrameLength> hat den Wert 16#06, da das Fragment nach <Command> endet. Der <FragmentationCounter> hat den Wert 16#00, da keine zusätzlichen Fragmente für die Übertragung des Befehlstelegramms erforderlich sind. Das Befehlstelegramm hat eine Länge (<TelegramLength>) von 16#03, da der Befehl mit dem Byte <Command> endet. Der Befehlscode <Command> für den Befehl Single Read EPC/UII ist 16#D2.

#### Antwort Datentelegramm, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	$D_S$	$U_M$	$U_D$	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#D2							
Byte 6	Status	16#00							
Byte 7	Parameter / Data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter / Data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter / Data	<PC-Word (High Byte)>							
Byte 10	Parameter / Data	<PC-Word (Low Byte)>							
Byte 11	Parameter / Data	<EPC / UII Byte 1>							
Byte 12	Parameter / Data	<EPC / UII Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<EPC / UII Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.10

Die Länge des Fragmentes <FrameLength> ist abhängig von der Größe des EPC/UII des eingelesenen Datenträgers. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <EPC/UII Byte X>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII des eingelesenen Datenträgers. Die Größe des Telegramms erstreckt sich bis einschließlich <EPC/UII Byte X>.

Der Parameter <Command> hat den Wert 16#D2 und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Diese hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII zur eindeutigen Identifizierung des gelesenen Transponders. Alle nachfolgenden Bytes innerhalb des Datentelegramms haben den Wert 16#00.

**Antwort Befehlsende, Long-Form-Datenformat:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#D2							
Byte 6	Status	16#0F							
Byte 7	Parameter / Data	<Anzahl Datenträger Byte 1>							
Byte 8	Parameter / Data	<Anzahl Datenträger Byte 2>							
Byte 9	Parameter / Data	<Anzahl Datenträger Byte 3>							
Byte 10	Parameter / Data	<Anzahl Datenträger Byte 4>							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.11

Die <FrameLength> ist in der Rückantwort für das Befehlsende des Single Read EPC/U1I konstant bei den Wert 16#0B. Das Fragment erstreckt sich dabei bis einschließlich <Anzahl Datenträger Byte 4>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die Telegrammlänge <TelegramLength> hat den Wert 16#08. Das Byte <Command> wird zurückgespiegelt und hat denselben Wert wie im Befehlstelegramm 16#D2. Der <Status> für das Telegramm zur Signalisierung des Befehlsendes ist 16#0F.

Die Anzahl der identifizierten Datenträger wird innerhalb von 4 Byte übertragen. Dabei wird die Anzahl in ASCII kodierter Darstellung wiedergegeben. Bei der Identifikation von einem Transponder hat die <Anzahl Datenträger> den Wert "0001" (ASCII) bzw. 16#30303031. Wenn während der Befehlsausführung kein Datenträger erkannt wurde, entfallen die Datentelegramme und es wird nur das Telegramm zur Signalisierung des Befehlsendes gesendet. Die <Anzahl Datenträger> hat dabei den Wert "0000" (ASCII) bzw. 16#30303030.

**Enhanced Read EPC/U1I (EN)**

Der Befehl "Enhanced Read EPC/U1I" hat den Befehlscode 16#D3 und führt einen permanenten Lesevorgang auf den EPC/U1I (Bank 01) eines oder mehrerer Transponder innerhalb der Erfassungszone durch. Die eingelesenen Informationen werden für jeden gelesenen Transponder in einem separaten Datentelegramm übertragen. Wenn ein Transponder die Erfassungszone verlässt, wird dies über ein Telegramm mit dem EPC/U1I des Transponders gemeldet. Die Befehlsausführung wird über ein Quit-Befehl gestoppt.

**Befehl:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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							

2024-08

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 5	Command	16#D3							
Byte 6	Nicht relevant	16#00							
Byte 7	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.12

Die <FrameLength> hat den Wert 16#06, da keine weiteren Befehlsparameter übertragen werden müssen und das Fragment nach <Command> endet. Der <FragmentationCounter> hat den Wert 16#00, da keine zusätzlichen Fragmente für die Übertragung des Befehlstelegramms erforderlich sind. Das Befehlstelegramm hat eine Länge (<TelegramLength>) von 16#03, da der Befehl mit dem Byte <Command> endet. Der Befehlscode <Command> für den Befehl Enhanced Read EPC/UII ist 16#D3.

#### Antwort Datentelegramm, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#D3							
Byte 6	Status	16#00							
Byte 7	Parameter / Data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter / Data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter / Data	<PC-Word (High Byte)>							
Byte 10	Parameter / Data	<PC-Word (Low Byte)>							
Byte 11	Parameter / Data	<EPC / UII Byte 1>							
Byte 12	Parameter / Data	<EPC / UII Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<EPC / UII Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.13

Die Länge des Fragments <FrameLength> ist abhängig von der Größe des EPC/UII des eingelesenen Datenträgers. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <EPC/UII Byte X>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII des eingelesenen Datenträgers. Die Größe des Telegramms erstreckt sich bis einschließlich <EPC/UII Byte X>.

Der Parameter <Command> hat den Wert 16#D3 und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Diese hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII zur eindeutigen Identifizierung des gelesenen Transponders. Alle nachfolgenden Bytes innerhalb des Datentelegramms haben den Wert 16#00.

#### Antwort Transponder hat Erfassungszone verlassen, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#D3							
Byte 6	Status	16#05							
Byte 7	Parameter / Data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter / Data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter / Data	<PC-Word (High Byte)>							
Byte 10	Parameter / Data	<PC-Word (Low Byte)>							
Byte 11	Parameter / Data	<EPC / UII Byte 1>							
Byte 12	Parameter / Data	<EPC / UII Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<EPC / UII Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.14

Die Länge des Fragments <FrameLength> ist abhängig von der Größe des EPC/UII des Datenträgers, der den Erfassungsbereich verlässt. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <EPC/UII Byte X>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII des Datenträgers. Die Größe des Telegramms erstreckt sich bis einschließlich <EPC/UII Byte X>.

Der Parameter <Command> hat den Wert 16#D3 und wird innerhalb des Telegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#05.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Diese hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII zur eindeutigen Identifizierung des Transponders, der den Erfassungsbereich verlassen hat. Alle nachfolgenden Bytes innerhalb des Telegramms haben den Wert 16#00.

#### Single Write EPC/UII (#SU)

Der Befehl "Single Write EPC/UII" hat den Befehlscode 16#CE und führt einen einmaligen Schreibvorgang auf den EPC/UII (Bank 01) eines Transponder innerhalb der Erfassungszone durch. Es darf sich während der Befehlsausführung immer nur ein Transponder innerhalb des Erfassungsbereichs befinden. Die erfolgreiche Programmierung des EPC/UII wird über ein Datentelegramm signalisiert. Sobald ein Datenträger erkannt und mit dem neuen EPC/UII

beschrieben wurde, bricht die Befehlsausführung ab. Es werden keine zusätzlichen Inventory-Durchläufe ausgeführt. Das Datentelegramm enthält den neu programmierten EPC/UII des Transponders. Das Ende der Befehlsausführung wird über ein Abschlusstelegramm signalisiert. Das Abschlusstelegramm enthält die Anzahl der Transponder, die während der Befehlsausführung identifiziert wurden.

Durch den Befehl "Single Write EPC/UII" lässt sich ein EPC/UII mit einer beliebigen Länge auf einen Datenträger schreiben. Die Länge des EPC/UII muss dabei ein Vielfaches von 2 Bytes sein. Das PC-Wort wird über diesen Befehl ebenfalls übertragen und wird den eigentlichen EPC/UII in dem Datenfeld vorangestellt. Der korrekte Wert des PC-Worts muss zuvor durch den Anwender selbst berechnet werden.

Zur Unterscheidung zwischen einem EPC-Code und einem UII-Code wird innerhalb des PC-Worts das Togglebit verwendet. Bei der Programmierung eines EPC-Codes muss dieses Bit auf 0 gesetzt sein. Bei der Programmierung eines UII-Codes ist dieses Bit hingegen auf 1 gesetzt.

Für das PC-Wort gelten folgende Werte in Abhängigkeit der Länge des EPC/UII:

Länge EPC/UII	EPC PC-Wort	UII PC-Wort
2 Byte	16#0800	16#0900
4 Byte	16#1000	16#1100
6 Byte	16#1800	16#1900
8 Byte	16#2000	16#2100
10 Byte	16#2800	16#2900
12 Byte	16#3000	16#3100
14 Byte	16#3800	16#3900
22 Byte	16#5800	16#5900

Der Parameter <Length EPC/UII> legt die Länge der EPC/UII-Information fest. Die EPC/UII-Information besteht aus dem PC-Wort mit der Länge von 2 Byte zuzüglich des EPC/UII. Die Länge des EPC/UII ist variabel.

Für <Length EPC/UII> gelten folgende Werte in Abhängigkeit der Länge des EPC/UII:

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

Tabelle 8.15

**Befehl:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#CE							
Byte 6	Length EPC/UII	<Length EPC/UII>							
Byte 7	Parameter / Data	<PC-Word (High Byte)>							
Byte 8	Parameter / Data	<PC-Word (Low Byte)>							
Byte 9	Parameter / Data	<EPC/UII Byte 1>							
Byte 10	Parameter / Data	<EPC/UII Byte 2>							
...	Parameter / Data	...							
Byte x	Parameter / Data	<EPC/UII Byte x>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Der Wert von <FrameLength> ist abhängig von der Länge des zu programmierenden EPC/UII-Codes. Das Fragment endet mit den Byte <EPC/UII Byte X>. Der <FragmentationCounter> hat den Wert 16#00, da keine zusätzlichen Fragmente für die Übertragung des Befehlstelegramms erforderlich sind. Die Länge des Befehlstelegramms (<TelegramLength>) ist abhängig von der Länge des zu programmierenden EPC/UII-Codes. Das Telegramm endet mit den Byte <EPC/UII Byte X>. Der Befehlscode <Command> für den Befehl Single Write EPC/UII ist 16#CE. Danach kommt die Parametrierung von <Length EPC/UII>. Der einzustellende Wert ist abhängig von der Länge des EPC/UII. Hieran schließt das <PC-Word> an. Der Wert des PC-Worts ist abhängig vom Typ des Codes (EPC oder UII) und der Codelänge. Anschließend folgt der auf den Datenträger zu programmierende EPC/UII-Code.

#### Antwort Datentelegramm, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#CE							
Byte 6	Status	16#00							
Byte 7	Parameter / Data	<Length EPC/UII (High Byte)>							
Byte 8	Parameter / Data	<Length EPC/UII (Low Byte)>							
Byte 9	Parameter / Data	<PC-Word (High Byte)>							
Byte 10	Parameter / Data	<PC-Word (Low Byte)>							
Byte 11	Parameter / Data	<EPC/UII Byte 1>							
Byte 12	Parameter / Data	<EPC/UII Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<EPC/UII Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#0							

Die Länge des Fragments <FrameLength> ist abhängig von der Größe des EPC/UII des neu programmierten Datenträgers. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <EPC/UII Byte X>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII des Datenträgers. Die Größe des Telegramms erstreckt sich bis einschließlich <EPC/UII Byte X>.

Der Parameter <Command> hat den Wert 16#CE und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Diese hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der neu in den Datenträger programmierte EPC/UII zur eindeutigen Identifizierung des beschriebenen Transponders.

#### Antwort Befehlsende, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#CE							
Byte 6	Status	16#0F							
Byte 7	Parameter / Data	<Anzahl Datenträger Byte 1>							
Byte 8	Parameter / Data	<Anzahl Datenträger Byte 2>							
Byte 9	Parameter / Data	<Anzahl Datenträger Byte 3>							
Byte 10	Parameter / Data	<Anzahl Datenträger Byte 4>							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Die <FrameLength> ist in der Rückantwort für das Befehlsende des Single Write EPC/UII konstant bei dem Wert 16#0B. Das Fragment erstreckt sich dabei bis einschließlich <Anzahl Datenträger Byte 4>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die Telegrammlänge <TelegramLength> hat den Wert 16#08. Das Byte <Command> wird zurückgespiegelt und hat denselben Wert wie im Befehlstelegramm 16#CE. Der <Status> für das Telegramm zur Signalisierung des Befehlsendes ist 16#0F.

Die Anzahl der beschriebenen Datenträger wird innerhalb von 4 Byte übertragen. Dabei wird die Anzahl in ASCII kodierter Darstellung wiedergegeben.

Bei der erfolgreichen Programmierung eines Transponders hat die <Anzahl Datenträger> den Wert "0001" (ASCII) bzw. 16#30303031.

Wenn während der Befehlsausführung kein Datenträger erkannt wurde, entfallen die Datentelegramme und es wird nur das Telegramm zur Signalisierung des Befehlsendes gesendet. Die <Anzahl Datenträger> hat dabei den Wert "0000" (ASCII) bzw. 16#30303030.

## Single Read 4-Byte Blocks (SR)

Der Befehl "Single Read 4-Byte Blocks" hat den Befehlscode 16#10 und führt einen einmaligen Lesevorgang auf den EPC/UII (Bank 01) und 4-Byte große Datenblöcke des User Memory (Bank 11) eines oder mehrerer Transponder innerhalb der Erfassungszone durch. Die eingelesenen Informationen werden für jeden gelesenen Transponder in einem separaten Datentelegramm mit den Statuswert 16#00 übertragen. Das Ende der Befehlsausführung wird über ein Abschlusstelegramm signalisiert. Das Abschlusstelegramm hat den Statuswert 16#0F und enthält die Anzahl der Transponder, die während der Befehlsausführung identifiziert wurden.

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 Speicherbanken Reserved (Bank 00), EPC/UII (Bank 01) und TID (Bank 10) auslesen.

Der Parameter <ByteAddress> legt die Startadresse innerhalb der Speicherbank fest, ab der die 4-Byte Blöcke eingelesen werden. Der Wert von <ByteAddress> ist bytebezogen. Somit können nur Vielfache von 4 parametrisiert werden. Über <Number of Bytes> wird die Anzahl der einzulesenden Bytes festgelegt. Die Byteanzahl muss ebenfalls ein Vielfaches von 4 sein.

### Befehl:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.16

Die <FrameLength> hat den Wert 16#0A und erstreckt sich bis einschließlich <Number of Bytes (Low Byte)>. Der <FragmentationCounter> hat den Wert 16#00, da keine zusätzlichen Fragmente für die Übertragung des Befehlsstelegramms erforderlich sind. Das Befehlsstelegramm hat eine Länge (<TelegramLength>) von 16#07 und endet mit <Number of Bytes (Low Byte)>. Der Befehlscode <Command> für den Befehl Single Read 4-Byte Blocks ist 16#10.

Der Parameter <ByteAddress> gibt die Startadresse innerhalb der Speicherbank an, ab der die 4-Byte Blöcke eingelesen werden. Über den Parameter <Number of Bytes> wird die Anzahl der einzulesenden Bytes festgelegt. Alle anderen Bytes des Befehlsfragments sind nicht für den Befehl relevant. Sie sind alle auf den Wert 16#00 zu setzen.

### Antwort Datentelegramm, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							

2024-08

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#10							
Byte 6	Status	16#00							
Byte 7	Parameter / Data	<Length EPC / Ull (High Byte)>							
Byte 8	Parameter / Data	<Length EPC / Ull (Low Byte)>							
Byte 9	Parameter / Data	<PC-Word (High Byte)>							
Byte 10	Parameter / Data	<PC-Word (Low Byte)>							
Byte 11	Parameter / Data	<EPC / Ull Byte 1>							
Byte 12	Parameter / Data	<EPC / Ull Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<EPC / Ull 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>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.17

Die Länge des Fragments <FrameLength> ist abhängig von der Größe des EPC/Ull sowie der Anzahl der eingelesenen Bytes des identifizierten Datenträgers. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <User Data Byte Y>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/Ull und der Anzahl der eingelesenen Bytes des Datenträgers. Die Größe des Telegrams erstreckt sich bis einschließlich <User Data Byte Y>.

Der Parameter <Command> hat den Wert 16#10 und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00.

Hieran schließt sich eine Längenangabe der EPC/Ull-Information an. Die Längenangabe hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/Ull. Danach folgt der EPC/Ull zur eindeutigen Identifizierung des gelesenen Transponders.

Nach dem EPC/Ull folgt eine Längenangabe für die eingelesenen User Data. Diese hat immer eine Größe von 2 Byte. Abschließend folgen die eingelesenen User Data des identifizierten Transponders bis einschließlich <User Data Byte Y>. Alle nachfolgenden Bytes innerhalb des Datentelegramms haben den Wert 16#00.

**Antwort Befehlsende, Long-Form-Datenformat:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	<Anzahl Datenträger Byte 1>							
Byte 8	Parameter / Data	<Anzahl Datenträger Byte 2>							
Byte 9	Parameter / Data	<Anzahl Datenträger Byte 3>							
Byte 10	Parameter / Data	<Anzahl Datenträger Byte 4>							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.18

Die <FrameLength> ist in der Rückantwort für das Befehlsende des Single Read 4-Byte Blocks konstant bei dem Wert 16#0B. Das Fragment erstreckt sich dabei bis einschließlich <Anzahl Datenträger Byte 4>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die Telegrammlänge <TelegramLength> hat den Wert 16#08. Das Byte <Command> wird zurückgespiegelt und hat denselben Wert wie im Befehlstelegramm 16#10. Der <Status> für das Telegramm zur Signalisierung des Befehlsendes ist 16#0F.

Die Anzahl der identifizierten Datenträger wird innerhalb von 4 Byte übertragen. Dabei wird die Anzahl in ASCII kodierter Darstellung wiedergegeben.

Bei der Identifikation von einem Transponder hat die <Anzahl Datenträger> den Wert "0001" (ASCII) bzw. 16#30303031.

Wenn während der Befehlsausführung kein Datenträger erkannt wurde, entfallen die Datentelegramme und es wird nur das Telegramm zur Signalisierung des Befehlsendes gesendet. Die <Anzahl Datenträger> hat dabei den Wert "0000" (ASCII) bzw. 16#30303030.

**Enhanced Read 4-Byte Blocks (ER)**

Der Befehl "Enhanced Read 4-Byte Blocks" hat den Befehlscode 16#19 und führt einen permanenten Lesevorgang auf den EPC/Ull (Bank 01) und 4-Byte große Datenblöcke des User Memory (Bank 11) eines oder mehrerer Transponder innerhalb der Erfassungszone durch. Die eingelesenen Informationen werden für jeden gelesenen Transponder in einem separaten Datentelegramm mit den Statuswert 16#00 übertragen. Wenn ein Transponder die Erfassungszone verlässt, wird dies über ein Telegramm mit dem EPC/Ull des Transponders gemeldet. Dieses Telegramm hat den Statuswert 16#05. Die Befehlsausführung wird über ein Quit-Befehl gestoppt.

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 Speicherbanken Reserved (Bank 00), EPC/Ull (Bank 01) und TID (Bank 10) auslesen.

Der Parameter <ByteAddress> legt die Startadresse innerhalb der Speicherbank fest, ab der die 4-Byte Blöcke eingelesen werden. Der Wert von <ByteAddress> ist bytebezogen. Somit können nur Vielfache von 4 parametrisiert werden. Über <Number of Bytes> wird die Anzahl der einzulesenden Bytes festgelegt. Die Byteanzahl muss ebenfalls ein Vielfaches von 4 sein.

**Befehl:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.19

Die <FrameLength> hat den Wert 16#0A und erstreckt sich bis einschließlich <Number of Bytes (Low Byte)>. Der <FragmentationCounter> hat den Wert 16#00, da keine zusätzlichen Fragmente für die Übertragung des Befehlstelegramms erforderlich sind. Das Befehlsstelegramm hat eine Länge (<TelegramLength>) von 16#07 und endet mit <Number of Bytes (Low Byte)>. Der Befehlscode <Command> für den Befehl Enhanced Read 4-Byte Blocks ist 16#19.

Der Parameter <ByteAddress> gibt die Startadresse innerhalb der Speicherbank an, ab der die 4-Byte Blöcke eingelesen werden. Über den Parameter <Number of Bytes> wird die Anzahl der einzulesenden Bytes festgelegt. Alle anderen Bytes des Befehlsfragments sind nicht für den Befehl relevant und sind alle auf den Wert 16#00 zusetzen.

**Antwort Datentelegramm, Long-Form-Datenformat:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Inhalt	Bit-Nummer							
		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>							
...	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>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.20

Die Länge des Fragmentes <FrameLength> ist abhängig von der Größe des EPC/UII sowie der Anzahl der eingelesenen Bytes des identifizierten Datenträgers. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <User Data Byte Y>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII und der Anzahl der eingelesenen Bytes des Datenträgers. Die Größe des Telegramms erstreckt sich bis einschließlich <User Data Byte Y>.

Der Parameter <Command> hat den Wert 16#19 und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> Parameter hat den Wert 16#00.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Diese hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII zur eindeutigen Identifizierung des gelesenen Transponders.

Nach dem EPC/UII folgt eine Längenangabe für die eingelesenen User Data. Diese hat immer eine Größe von 2 Byte. Abschließend folgen die eingelesenen User Data des identifizierten Transponders bis einschließlich <User Data Byte Y>. Alle nachfolgenden Bytes innerhalb des Datentelegramms haben den Wert 16#00.

#### Antwort Transponder hat Erfassungszone verlassen, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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)>							

2024-08

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 10	Parameter / Data	<PC-Word (Low Byte)>							
Byte 11	Parameter / Data	<EPC / Ull Byte 1>							
Byte 12	Parameter / Data	<EPC / Ull Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<EPC / Ull Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.21

Die Länge des Fragmentes <FrameLength> ist abhängig von der Größe des EPC/Ull des Datenträgers, der den Erfassungsbereich verlässt. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <EPC/Ull Byte X>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/Ull des Datenträgers. Die Größe des Telegramms erstreckt sich bis einschließlich <EPC/Ull Byte X>.

Der Parameter <Command> hat den Wert 16#19 und wird innerhalb des Telegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#05.

Hieran schließt sich eine Längenangabe der EPC/Ull-Information an. Diese hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/Ull. Danach folgt der EPC/Ull zur eindeutigen Identifizierung des Transponders, der den Erfassungsbereich verlassen hat. Alle nachfolgenden Bytes innerhalb des Telegramms haben den Wert 16#00.

### Single Write 4-Byte Blocks (SW)

Der Befehl "Single Write 4-Byte Blocks" hat den Befehlscode 16#40 und führt einen einmaligen Schreibvorgang auf 4-Byte große Datenblöcke des User Memory (Bank 11) eines oder mehrerer Transponder innerhalb der Erfassungszone durch. Der erfolgreiche Schreibvorgang auf den User Memory wird für jeden beschriebenen Transponder über ein Datentelegramm mit den Statuswert 16#00 signalisiert. Das Datentelegramm enthält den EPC/Ull des Transponders, auf dem der User Memory beschrieben wurde. Das Ende der Befehlsausführung wird über ein Abschlusstelegramm signalisiert. Das Abschlussdiagramm hat den Statuswert 16#0F und enthält die Anzahl der Transponder, die während der Befehlsausführung beschrieben wurden.

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 Speicherbanken Reserved (Bank 00) und EPC/Ull (Bank 01) beschreiben.

Der Parameter <ByteAddress> legt die Startadresse innerhalb der Speicherbank fest, ab der die 4-Byte Blöcke geschrieben werden. Der Wert von <ByteAddress> ist bytebezogen. Somit können nur Vielfache von 4 parametrisiert werden. Über <Number of Bytes> wird die Anzahl der zu schreibenden Bytes festgelegt. Die Byteanzahl muss ebenfalls ein Vielfaches von 4 sein.

#### Befehl:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 5	Command	16#40							
Byte 6	Parameter / Data	<ByteAddress (High Byte)>							
Byte 7	Parameter / Data	<ByteAddress (Low Byte)>							
Byte 8	Parameter / Data	<Number of Bytes (High Byte)>							
Byte 9	Parameter / Data	<Number of Bytes (Low Byte)>							
Byte 10	Parameter / Data	<Write Data Byte 1>							
Byte 11	Parameter / Data	<Write Data Byte 2>							
...	Parameter / Data	...							
Byte x	Parameter / Data	<Write Data Byte Y>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.22

Der Wert von <FrameLength> ist abhängig von der Anzahl der zuschreibenden Bytes. Das Fragment endet mit den Byte <Write Data Byte Y>. Der <FragmentationCounter> hat den Wert 16#00, da keine zusätzlichen Fragmente für die Übertragung des Befehlstelegramms erforderlich sind. Die Länge des Befehlstelegramms (<TelegramLength>) ist abhängig von der Anzahl der zuschreibenden Bytes. Das Telegramm endet mit den Byte <Write Data Byte Y>. Der Befehlscode <Command> für den Befehl Single Write 4-Byte Blocks ist 16#40.

Der Parameter <ByteAddress> gibt die Startadresse innerhalb der Speicherbank an, ab der die 4-Byte Blöcke geschrieben werden. Über den Parameter <Number of Bytes> wird die Anzahl der zuschreibenden Bytes festgelegt. Anschließend folgen die auf den Datenträger zuschreibenden Informationen <Write Data Byte>. Alle anderen Bytes des Befehlsfragments sind nicht für den Befehl relevant und sind alle auf den Wert 16#00 zusetzen.

#### Antwort Datentelegramm, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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>							
...	Nicht relevant	16#00							

2024-08

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 31	Nicht relevant	16#00							

Tabelle 8.23

Die Länge des Fragments <FrameLength> ist abhängig von der Größe des EPC/UII des beschriebenen Datenträgers. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <EPC/UII Byte X>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII des Datenträgers. Die Größe des Telegramms erstreckt sich bis einschließlich <EPC/UII Byte X>.

Der Parameter <Command> hat den Wert 16#40 und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Die Längenangabe hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII des Datenträgers zur eindeutigen Identifizierung des beschriebenen Transponders.

#### Antwort Befehlsende, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	<Anzahl Datenträger Byte 1>							
Byte 8	Parameter / Data	<Anzahl Datenträger Byte 2>							
Byte 9	Parameter / Data	<Anzahl Datenträger Byte 3>							
Byte 10	Parameter / Data	<Anzahl Datenträger Byte 4>							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.24

Die <FrameLength> ist in der Rückantwort für das Befehlsende des Single Write 4-Byte Blocks konstant bei dem Wert 16#0B. Das Fragment erstreckt sich dabei bis einschließlich <Anzahl Datenträger Byte 4>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die Telegrammlänge <TelegramLength> hat den Wert 16#08. Das Byte <Command> wird zurückgespiegelt und hat denselben Wert wie im Befehlstelegramm 16#40. Der <Status> für das Telegramm zur Signalisierung des Befehlendes ist 16#0F.

Die Anzahl der beschriebenen Datenträger wird innerhalb von 4 Byte übertragen. Dabei wird die Anzahl in ASCII kodierter Darstellung wiedergegeben.

Bei einem erfolgreichen Schreibvorgang auf einen Transponder hat die <Anzahl Datenträger> den Wert "0001" (ASCII) bzw. 16#30303031.

Wenn während der Befehlsausführung kein Datenträger erkannt bzw. beschrieben wurde, entfallen die Datentelegramme und es wird nur das Telegramm zur Signalisierung des Befehlsendes gesendet. Die <Anzahl Datenträger> hat dabei den Wert "0000" (ASCII) bzw. 16#30303030.

### Enhanced Write 4-Byte Blocks (EW)

Der Befehl "Enhanced Write 4-Byte Blocks" hat den Befehlscode 16#1A und führt einen permanenten Schreibvorgang auf 4-Byte große Datenblöcke des User Memory (Bank 11) eines oder mehrerer Transponder innerhalb der Erfassungszone durch. Der erfolgreiche Schreibvorgang auf den User Memory wird für jeden beschriebenen Transponder über ein Datentelegramm mit den Statuswert 16#00 signalisiert. Das Datentelegramm enthält den EPC/UII des Transponders auf dem der User Memory beschrieben wurde. Wenn ein Transponder die Erfassungszone verlässt, wird dies über ein Telegramm mit dem EPC/UII des Transponders gemeldet. Dieses Telegramm hat den Statuswert 16#05. Die Befehlsausführung wird über ein Quit-Befehl gestoppt.

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 Speicherbanken Reserved (Bank 00) und EPC/UII (Bank 01) beschreiben.

Der Parameter <ByteAddress> legt die Startadresse innerhalb der Speicherbank fest, ab der die 4-Byte Blöcke geschrieben werden. Der Wert von <ByteAddress> ist bytebezogen. Somit können nur Vielfache von 4 parametrisiert werden. Über <Number of Bytes> wird die Anzahl der zuschreibenden Bytes festgelegt. Die Byteanzahl muss ebenfalls ein Vielfaches von 4 sein.

#### Befehl:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.25

Der Wert von <FrameLength> ist abhängig von der Anzahl der zuschreibenden Bytes. Das Fragment endet mit den Byte <Write Data Byte Y>. Der <FragmentationCounter> hat den Wert 16#00, da keine zusätzlichen Fragmente für die Übertragung des Befehlstelegramms erforderlich sind. Die Länge des Befehlstelegramms (<TelegramLength>) ist abhängig von der Anzahl der zuschreibenden Bytes. Das Telegramm endet mit den Byte <Write Data Byte Y>. Der Befehlscode <Command> für den Befehl Enhanced Write 4-Byte Blocks ist 16#1A.

Der Parameter <ByteAddress> gibt die Startadresse innerhalb der Speicherbank an, ab der die 4-Byte Blöcke geschrieben werden. Über den Parameter <Number of Bytes> wird die Anzahl der zuschreibenden Bytes festgelegt. Anschließend folgen die auf den Datenträger zuschreibenden Informationen <Write Data Byte>. Alle anderen Bytes des Befehlsfragments sind nicht für den Befehl relevant und sind alle auf den Wert 16#00 zusetzen.

#### Antwort Datentelegramm, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#1A							
Byte 6	Status	16#00							
Byte 7	Parameter / Data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter / Data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter / Data	<PC-Word (High Byte)>							
Byte 10	Parameter / Data	<PC-Word (Low Byte)>							
Byte 11	Parameter / Data	<EPC / UII Byte 1>							
Byte 12	Parameter / Data	<EPC / UII Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<EPC / UII Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.26

Die Länge des Fragments <FrameLength> ist abhängig von der Größe des EPC/UII des beschriebenen Datenträgers. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <EPC/UII Byte X>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII des beschriebenen Datenträgers. Die Größe des Telegramms erstreckt sich bis einschließlich <EPC/UII Byte X>.

Der Parameter <Command> hat den Wert 16#1A und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Die Längenangabe hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII zur eindeutigen Identifizierung des beschriebenen Transponders. Alle nachfolgenden Bytes innerhalb des Datentelegramms haben den Wert 16#00.

#### Antwort Transponder hat Erfassungszone verlassen, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
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>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.27

Die Länge des Fragments <FrameLength> ist abhängig von der Größe des EPC/UII des Datenträgers, der den Erfassungsbereich verlässt. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <EPC/UII Byte X>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII des Datenträgers. Die Größe des Telegramms erstreckt sich bis einschließlich <EPC/UII Byte X>.

Der Parameter <Command> hat den Wert 16#1A und wird innerhalb des Telegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#05.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Die Längenangabe hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII zur eindeutigen Identifizierung des Transponders, der den Erfassungsbereich verlassen hat. Alle nachfolgenden Bytes innerhalb des Telegramms haben den Wert 16#00.

### Single Read 2-Byte Words (#SR)

Der Befehl "Single Read 2-Byte Words" hat den Befehlscode 16#49 und führt einen einmaligen Lesevorgang auf den EPC/UII (Bank 01) und 2-Byte große Datenwörter des User Memory (Bank 11) eines oder mehrerer Transponder innerhalb der Erfassungszone durch. Die eingelesenen Informationen werden für jeden gelesenen Transponder in einem separaten Datentelegramm mit den Statuswert 16#00 übertragen. Das Ende der Befehlsausführung wird über ein Abschlusstelegramm signalisiert. Das Abschlusstelegramm hat den Statuswert 16#0F und enthält die Anzahl der Transponder, die während der Befehlsausführung identifiziert wurden.

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 Speicherbanken Reserved (Bank 00), EPC/UII (Bank 01) und TID (Bank 10) auslesen.

Der Parameter <ByteAddress> legt die Startadresse innerhalb der Speicherbank fest, ab der die 2-Byte Wörter eingelesen werden. Der Wert von <ByteAddress> ist bytebezogen. Somit können nur Vielfache von 2 parametrisiert werden. Über <Number of Bytes> wird die Anzahl der einzulesenden Bytes festgelegt. Die Byteanzahl muss ebenfalls ein Vielfaches von 2 sein.

**Befehl:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.28

Die <FrameLength> hat den Wert 16#0A und erstreckt sich bis einschließlich <Number of Bytes (Low Byte)>. Der <FragmentationCounter> hat den Wert 16#00, da keine zusätzlichen Fragmente für die Übertragung des Befehlsstelegramms erforderlich sind. Das Befehlsstelegramm hat eine Länge (<TelegramLength>) von 16#07 und endet mit <Number of Bytes (Low Byte)>. Der Befehlscode <Command> für den Befehl Single Read 2-Byte Words ist 16#49.

Der Parameter <ByteAddress> gibt die Startadresse innerhalb der Speicherbank an, ab der die 2-Byte Wörter eingelesen werden. Über den Parameter <Number of Bytes> wird die Anzahl der einzulesenden Bytes festgelegt. Alle anderen Bytes des Befehlsfragments sind nicht für den Befehl relevant. Sie sind alle auf den Wert 16#00 zu setzen.

**Antwort Datentelegramm, Long-Form-Datenformat:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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>							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
...	Parameter / Data	<Length User Data (High Byte)>							
...	Parameter / Data	<Length User Data (Low Byte)>							
...	Parameter / Data	<User Data Byte 1>							
...	Parameter / Data	<User Data Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<User Data Byte Y>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.29

Die Länge des Fragments <FrameLength> ist abhängig von der Größe des EPC/UII sowie der Anzahl der eingelesenen Bytes des identifizierten Datenträgers. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <User Data Byte Y>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII und der Anzahl der eingelesenen Bytes des Datenträgers. Die Größe des Telegrams erstreckt sich bis einschließlich <User Data Byte Y>.

Der Parameter <Command> hat den Wert 16#49 und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Die Längenangabe hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII zur eindeutigen Identifizierung des gelesenen Transponders.

Nach dem EPC/UII folgt eine Längenangabe für die eingelesenen User Data. Diese hat immer eine Größe von 2 Byte. Abschließend folgen die eingelesenen User Data des identifizierten Transponders bis einschließlich <User Data Byte Y>. Alle nachfolgenden Bytes innerhalb des Datentelegramms haben den Wert 16#00.

#### Antwort Befehlsende, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	<Anzahl Datenträger Byte 1>							
Byte 8	Parameter / Data	<Anzahl Datenträger Byte 2>							
Byte 9	Parameter / Data	<Anzahl Datenträger Byte 3>							
Byte 10	Parameter / Data	<Anzahl Datenträger Byte 4>							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							

2024-08

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 31	Nicht relevant	16#00							

Tabelle 8.30

Die <FrameLength> ist in der Rückantwort für das Befehlsende des Single Read 2-Byte Words konstant bei dem Wert 16#0B. Das Fragment erstreckt sich dabei bis einschließlich <Anzahl Datenträger Byte 4>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die Telegrammlänge <TelegramLength> hat den Wert 16#08. Das Byte <Command> wird zurückgespiegelt und hat denselben Wert wie im Befehlstelegramm 16#49. Der <Status> für das Telegramm zur Signalisierung des Befehlsendes ist 16#0F.

Die Anzahl der identifizierten Datenträger wird innerhalb von 4 Byte übertragen. Dabei wird die Anzahl in ASCII kodierter Darstellung wiedergegeben.

Bei der Identifikation von einem Transponder hat die <Anzahl Datenträger> den Wert "0001" (ASCII) bzw. 16#30303031.

Wenn während der Befehlsausführung kein Datenträger erkannt wurde, entfallen die Datentelegramme und es wird nur das Telegramm zur Signalisierung des Befehlsendes gesendet. Die <Anzahl Datenträger> hat dabei den Wert "0000" (ASCII) bzw. 16#30303030.

### Enhanced Read 2-Byte Words (#ER)

Der Befehl "Enhanced Read 2-Byte Words" hat den Befehlscode 16#4B und führt einen permanenten Lesevorgang auf den EPC/UII (Bank 01) und 2-Byte große Datenwörter des User Memory (Bank 11) eines oder mehrerer Transponder innerhalb der Erfassungszone durch. Die eingelesenen Informationen werden für jeden gelesenen Transponder in einem separaten Datentelegramm mit den Statuswert 16#00 übertragen. Wenn ein Transponder die Erfassungszone verlässt, wird dies über ein Telegramm mit dem EPC/UII des Transponders gemeldet. Dieses Telegramm hat den Statuswert 16#05. Die Befehlsausführung wird über ein Quit-Befehl gestoppt.

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 Speicherbanken Reserved (Bank 00), EPC/UII (Bank 01) und TID (Bank 10) auslesen.

Der Parameter <ByteAddress> legt die Startadresse innerhalb der Speicherbank fest, ab der die 2-Byte Wörter eingelesen werden. Der Wert von <ByteAddress> ist bytebezogen. Somit können nur Vielfache von 2 parametrisiert werden. Über <Number of Bytes> wird die Anzahl der einzulesenden Bytes festgelegt. Die Byteanzahl muss ebenfalls ein Vielfaches von 2 sein.

#### Befehl:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 10	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.31

Die <FrameLength> hat den Wert 16#0A und erstreckt sich bis einschließlich <Number of Bytes (Low Byte)>. Der <FragmentationCounter> hat den Wert 16#00, da keine zusätzlichen Fragmente für die Übertragung des Befehlstelegramms erforderlich sind. Das Befehlstelegramm hat eine Länge (<TelegramLength>) von 16#07 und endet mit <Number of Bytes (Low Byte)>. Der Befehlscode <Command> für den Befehl Enhanced Read 2-Byte Words ist 16#4B.

Der Parameter <ByteAddress> gibt die Startadresse innerhalb der Speicherbank an, ab der die 2-Byte Wörter eingelesen werden. Über den Parameter <Number of Bytes> wird die Anzahl der einzulesenden Bytes festgelegt. Alle anderen Bytes des Befehlsfragments sind nicht für den Befehl relevant und sind alle auf den Wert 16#00 zusetzen.

#### Antwort Datentelegramm, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#4B							
Byte 6	Status	16#00							
Byte 7	Parameter / Data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter / Data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter / Data	<PC-Word (High Byte)>							
Byte 10	Parameter / Data	<PC-Word (Low Byte)>							
Byte 11	Parameter / Data	<EPC / UII Byte 1>							
Byte 12	Parameter / Data	<EPC / UII Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<EPC / UII Byte X>							
...	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>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.32

Die Länge des Fragmentes <FrameLength> ist abhängig von der Größe des EPC/UII sowie der Anzahl der eingelesenen Bytes des identifizierten Datenträgers. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <User Data Byte Y>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII und der Anzahl der eingelesenen Bytes des Datenträgers. Die Größe des Telegramms erstreckt sich bis einschließlich <User Data Byte Y>.

Der Parameter <Command> hat den Wert 16#4B und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> Parameter hat den Wert 16#00.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Diese hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII zur eindeutigen Identifizierung des gelesenen Transponders.

Nach dem EPC/UII folgt eine Längenangabe für die eingelesenen User Data. Diese hat immer eine Größe von 2 Byte. Abschließend folgen die eingelesenen User Data des identifizierten Transponders bis einschließlich <User Data Byte Y>. Alle nachfolgenden Bytes innerhalb des Datentelegramms haben den Wert 16#00.

#### Antwort Transponder hat Erfassungszone verlassen, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#4B							
Byte 6	Status	16#05							
Byte 7	Parameter / Data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter / Data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter / Data	<PC-Word (High Byte)>							
Byte 10	Parameter / Data	<PC-Word (Low Byte)>							
Byte 11	Parameter / Data	<EPC / UII Byte 1>							
Byte 12	Parameter / Data	<EPC / UII Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<EPC / UII Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.33

Die Länge des Fragmentes <FrameLength> ist abhängig von der Größe des EPC/UII des Datenträgers, der den Erfassungsbereich verlässt. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <EPC/UII Byte X>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII des Datenträgers. Die Größe des Telegramms erstreckt sich bis einschließlich <EPC/UII Byte X>.

Der Parameter <Command> hat den Wert 16#4B und wird innerhalb des Telegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#05.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Diese hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII zur eindeutigen Identifizierung des Transponders, der den Erfassungsbereich verlassen hat. Alle nachfolgenden Bytes innerhalb des Telegramms haben den Wert 16#00.

### Single Write 2-Byte Words (#SW)

Der Befehl "Single Write 2-Byte Words" hat den Befehlscode 16#4A und führt einen einmaligen Schreibvorgang auf 2-Byte große Datenwörter des User Memory (Bank 11) eines oder mehrerer Transponder innerhalb der Erfassungszone durch. Der erfolgreiche Schreibvorgang auf den User Memory wird für jeden beschriebenen Transponder über ein Datentelegramm mit den Statuswert 16#00 signalisiert. Das Datentelegramm enthält den EPC/UII des Transponders, auf dem der User Memory beschrieben wurde. Das Ende der Befehlsausführung wird über ein Abschlusstelegramm signalisiert. Das Abschlusssdiagramm hat den Statuswert 16#0F und enthält die Anzahl der Transponder, die während der Befehlsausführung beschrieben wurden.

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 Speicherbanken Reserved (Bank 00) und EPC/UII (Bank 01) beschreiben.

Der Parameter <ByteAddress> legt die Startadresse innerhalb der Speicherbank fest, ab der die 2-Byte Wörter geschrieben werden. Der Wert von <ByteAddress> ist bytebezogen. Somit können nur Vielfache von 2 parametrisiert werden. Über <Number of Bytes> wird die Anzahl der zu schreibenden Bytes festgelegt. Die Byteanzahl muss ebenfalls ein Vielfaches von 2 sein.

#### Befehl:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#4A							
Byte 6	Parameter / Data	<ByteAddress (High Byte)>							
Byte 7	Parameter / Data	<ByteAddress (Low Byte)>							
Byte 8	Parameter / Data	<Number of Bytes (High Byte)>							
Byte 9	Parameter / Data	<Number of Bytes (Low Byte)>							
Byte 10	Parameter / Data	<Write Data Byte 1>							
Byte 11	Parameter / Data	<Write Data Byte 2>							
...	Parameter / Data	...							
Byte x	Parameter / Data	<Write Data Byte Y>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.34

Der Wert von <FrameLength> ist abhängig von der Anzahl der zuschreibenden Bytes. Das Fragment endet mit den Byte <Write Data Byte Y>. Der <FragmentationCounter> hat den Wert 16#00, da keine zusätzlichen Fragmente für die Übertragung des Befehlstelegramms erforderlich sind. Die Länge des Befehlstelegramms (<TelegramLength>) ist abhängig von der Anzahl der zuschreibenden Bytes. Das Telegramm endet mit den Byte <Write Data Byte Y>. Der Befehlscode <Command> für den Befehl Single Write 2-Byte Words ist 16#4A.

Der Parameter <ByteAddress> gibt die Startadresse innerhalb der Speicherbank an, ab der die 2-Byte Wörter geschrieben werden. Über den Parameter <Number of Bytes> wird die Anzahl der zuschreibenden Bytes festgelegt. Anschließend folgen die auf den Datenträger zuschreibenden Informationen <Write Data Byte>. Alle anderen Bytes des Befehlsfragments sind nicht für den Befehl relevant und sind alle auf den Wert 16#00 zusetzen.

#### Antwort Datentelegramm, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.35

Die Länge des Fragments <FrameLength> ist abhängig von der Größe des EPC/UII des beschriebenen Datenträgers. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <EPC/UII Byte X>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII des Datenträgers. Die Größe des Telegramms erstreckt sich bis einschließlich <EPC/UII Byte X>.

Der Parameter <Command> hat den Wert 16#4A und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Diese hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII des Datenträgers zur eindeutigen Identifizierung des beschriebenen Transponders.

#### Antwort Befehlsende, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#4A							
Byte 6	Status	16#0F							
Byte 7	Parameter / Data	<Anzahl Datenträger Byte 1>							
Byte 8	Parameter / Data	<Anzahl Datenträger Byte 2>							
Byte 9	Parameter / Data	<Anzahl Datenträger Byte 3>							
Byte 10	Parameter / Data	<Anzahl Datenträger Byte 4>							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.36

Die <FrameLength> ist in der Rückantwort für das Befehlsende des Single Write 2-Byte Words konstant bei dem Wert 16#0B. Das Fragment erstreckt sich dabei bis einschließlich <Anzahl Datenträger Byte 4>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die Telegrammlänge <TelegramLength> hat den Wert 16#08. Das Byte <Command> wird zurückgespiegelt und hat denselben Wert wie im Befehlstelegramm 16#4A. Der <Status> für das Telegramm zur Signalisierung des Befehlsendes ist 16#0F.

Die Anzahl der beschriebenen Datenträger wird innerhalb von 4 Byte übertragen. Dabei wird die Anzahl in ASCII kodierter Darstellung wiedergegeben.

Bei einem erfolgreichen Schreibvorgang auf einen Transponder hat die <Anzahl Datenträger> den Wert "0001" (ASCII) bzw. 16#30303031.

Wenn während der Befehlsausführung kein Datenträger erkannt bzw. beschrieben wurde, entfallen die Datentelegramme und es wird nur das Telegramm zur Signalisierung des Befehlsendes gesendet. Die <Anzahl Datenträger> hat dabei den Wert "0000" (ASCII) bzw. 16#30303030.

### Enhanced Write 2-Byte Words (#EW)

Der Befehl "Enhanced Write 2-Byte Words" hat den Befehlscode 16#4C und führt einen permanenten Schreibvorgang auf 2-Byte große Datenblöcke des User Memory (Bank 11) eines oder mehrerer Transponder innerhalb der Erfassungszone durch. Der erfolgreiche Schreibvorgang auf den User Memory wird für jeden beschriebenen Transponder über ein Datentelegramm mit dem Statuswert 16#00 signalisiert. Das Datentelegramm enthält den EPC/UII des Transponders auf dem der User Memory beschrieben wurde. Wenn ein Transponder die Erfassungszone verlässt, wird dies über ein Telegramm mit dem EPC/UII des Transponders gemeldet. Dieses Telegramm hat den Statuswert 16#05. Die Befehlsausführung wird über ein Quit-Befehl gestoppt.

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 Speicherbanken (Bank 00) und EPC/UII (Bank 01) beschreiben.

Der Parameter <ByteAddress> legt die Startadresse innerhalb der Speicherbank fest, ab der die 2-Byte Wörter geschrieben werden. Der Wert von <ByteAddress> ist bytebezogen. Somit können nur Vielfache von 2 parametrisiert werden. Über <Number of Bytes> wird die Anzahl der zuschreibenden Bytes festgelegt. Die Byteanzahl muss ebenfalls ein Vielfaches von 2 sein.

**Befehl:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.37

Der Wert von <FrameLength> ist abhängig von der Anzahl der zuschreibenden Bytes. Das Fragment endet mit den Byte <Write Data Byte Y>. Der <FragmentationCounter> hat den Wert 16#00, da keine zusätzlichen Fragmente für die Übertragung des Befehlstelegramms erforderlich sind. Die Länge des Befehlstelegramms (<TelegramLength>) ist abhängig von der Anzahl der zuschreibenden Bytes. Das Telegramm endet mit den Byte <Write Data Byte Y>. Der Befehlscode <Command> für den Befehl Enhanced Write 2-Byte Words ist 16#4C.

Der Parameter <ByteAddress> gibt die Startadresse innerhalb der Speicherbank an, ab der die 2-Byte Wörter geschrieben werden. Über den Parameter <Number of Bytes> wird die Anzahl der zuschreibenden Bytes festgelegt. Anschließend folgen die auf den Datenträger zuschreibenden Informationen <Write Data Byte>. Alle anderen Bytes des Befehlsfragments sind nicht für den Befehl relevant und sind alle auf den Wert 16#00 zusetzen.

**Antwort Datentelegramm, Long-Form-Datenformat:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Inhalt	Bit-Nummer							
		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>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.38

Die Länge des Fragments <FrameLength> ist abhängig von der Größe des EPC/UII des beschriebenen Datenträgers. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <EPC/UII Byte X>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII des beschriebenen Datenträgers. Die Größe des Telegramms erstreckt sich bis einschließlich <EPC/UII Byte X>.

Der Parameter <Command> hat den Wert 16#4C und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Diese hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII zur eindeutigen Identifizierung des beschriebenen Transponders. Alle nachfolgenden Bytes innerhalb des Datentelegramms haben den Wert 16#00.

#### Antwort Transponder hat Erfassungszone verlassen, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#4C							
Byte 6	Status	16#05							
Byte 7	Parameter / Data	<Length EPC / UII (High Byte)>							
Byte 8	Parameter / Data	<Length EPC / UII (Low Byte)>							
Byte 9	Parameter / Data	<PC-Word (High Byte)>							
Byte 10	Parameter / Data	<PC-Word (Low Byte)>							
Byte 11	Parameter / Data	<EPC / UII Byte 1>							
Byte 12	Parameter / Data	<EPC / UII Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<EPC / UII Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.39

Die Länge des Fragments <FrameLength> ist abhängig von der Größe des EPC/UII des Datenträgers, der den Erfassungsbereich verlässt. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <EPC/UII Byte X>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII des Datenträgers. Die Größe des Telegramms erstreckt sich bis einschließlich <EPC/UII Byte X>.

Der Parameter <Command> hat den Wert 16#4C und wird innerhalb des Telegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#05.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Die Längenangabe hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII zur eindeutigen Identifizierung des Transponders, der den Erfassungsbereich verlassen hat. Alle nachfolgenden Bytes innerhalb des Telegramms haben den Wert 16#00.

## Lock (LO)

Der Befehl "Lock" hat den Befehlscode 16#D5 und führt einen einmaligen Schreibzugriff auf einen oder mehrere Transponder innerhalb der Erfassungszone aus. Der erfolgreiche Schreibvorgang wird für jeden beschriebenen Transponder über ein Datentelegramm mit den Statuswert 16#00 signalisiert. Das Datentelegramm enthält den EPC/UII des Transponders, auf den die Konfiguration geschrieben wurde. Das Ende der Befehlsausführung wird über ein Abschlusstelegramm signalisiert. Das Abschlusstelegramm hat den Statuswert 16#0F und enthält die Anzahl der Transponder, die während der Befehlsausführung beschrieben wurden.

Der Wert von <FrameLength> ist abhängig von der Länge der EPC/UII-Mask. Das Fragment endet mit den Byte <Payload (Byte 2)>. Der <FragmentationCounter> hat den Wert 16#00, da keine zusätzlichen Fragmente für die Übertragung des Befehlsstelegramms erforderlich sind. Die Länge des Befehlsstelegramms <TelegramLength> ist abhängig von der Länge der EPC/UII-Mask. Das Telegramm endet mit den Byte <Payload (Byte 2)>.

Mit <Length EPC/UII Mask> kann die Länge der EPC/UII-Maske (Bank 01) eines oder mehrere Transponder eingestellt werden, auf die der Befehl wirken soll. Die Länge der Maske darf nicht 0 betragen. Über den Parameter <EPC/UII Mask> werden die Transponder ausgewählt, auf die der Befehl wirken soll. Im Access Password muss das zuvor gespeicherte Zugangspasswort eingetragen werden. Das Passwort darf nicht 0 sein. Über den Parameter <Payload> werden die verschiedenen Lock-Funktionen konfiguriert.

## Payload

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

Payload	Bit	Funktion	Aktion
Access Password	16	Access Mask	skip / write
	17		skip / write
Kill Password	18	Kill Mask	skip / write
	19		skip / write

**Befehl:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#D5							
Byte 6	Length EPC/UII Mask (High Byte)	<Length EPC/UII Mask (High Byte)>							
Byte 7	Length EPC/UII Mask (Low Byte)	<Length EPC/UII Mask (Low Byte)>							
Byte 8	EPC/UII Mask (Byte 0)	<EPC/UII Mask (Byte 0)>							
Byte 9	EPC/UII Mask (Byte 1)	<EPC/UII Mask (Byte 1)>							
...	...	...							
Byte x	EPC/UII Mask (Byte Y)	<EPC/UII Mask (Byte Y)>							
Byte x+1	Access Password (Byte 0)	<Access Password (Byte 0)>							
Byte x+2	Access Password (Byte 1)	<Access Password (Byte 1)>							
Byte x+3	Access Password (Byte 2)	<Access Password (Byte 2)>							
Byte x+4	Access Password (Byte 3)	<Access Password (Byte 3)>							
Byte x+5	Payload (Byte 0)	<Payload (Byte 0)>							
Byte x+6	Payload (Byte 1)	<Payload (Byte 1)>							
Byte x+7	Payload (Byte 2)	<Payload (Byte 2)>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.40

**Antwort Datentelegramm, Long-Form-Datenformat:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#D5							
Byte 6	Status	16#00							
Byte 7	Parameter / Data	<Length EPC/UII (High Byte)>							

2024-08

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 8	Parameter / Data	<Length EPC/UII (Low Byte)>							
Byte 9	Parameter / Data	<PC-Word (High Byte)>							
Byte 10	Parameter / Data	<PC-Word (Low Byte)>							
Byte 11	Parameter / Data	<EPC/UII Byte 1>							
Byte 12	Parameter / Data	<EPC/UII Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<EPC/UII Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Die Länge des Fragments <FrameLength> ist abhängig von der Größe des EPC/UII des beschriebenen Datenträgers. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <EPC/UII Byte X>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII des beschriebenen Datenträgers. Die Größe des Telegramms erstreckt sich bis einschließlich <EPC/UII Byte X>.

Der Parameter <Command> hat den Wert 16#D5 und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Die Längenangabe hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII zur eindeutigen Identifizierung des beschriebenen Transponders. Alle nachfolgenden Bytes innerhalb des Datentelegramms haben den Wert 16#00.

#### Antwort Befehlsende, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#D5							
Byte 6	Status	16#0F							
Byte 7	Parameter / Data	<Anzahl Datenträger Byte 1>							
Byte 8	Parameter / Data	<Anzahl Datenträger Byte 2>							
Byte 9	Parameter / Data	<Anzahl Datenträger Byte 3>							
Byte 10	Parameter / Data	<Anzahl Datenträger Byte 4>							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Die <FrameLength> ist in der Rückantwort für das Befehlsende des Lock-Befehls konstant bei dem Wert 16#0B. Das Fragment erstreckt sich dabei bis einschließlich <Anzahl Datenträger Byte 4>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die Telegrammlänge <TelegramLength> hat den Wert 16#08. Das Byte <Command> wird zurückgespiegelt und hat denselben Wert wie im Befehlstelegramm 16#D5. Der <Status> für das Telegramm zur Signalisierung des Befehlsendes ist 16#0F.

Die Anzahl der programmierten Datenträger wird innerhalb von 4 Byte übertragen. Dabei wird die Anzahl in ASCII kodierter Darstellung wiedergegeben.

Bei der Identifikation von einem Transponder hat die <Anzahl Datenträger> den Wert "0001" (ASCII) bzw. 16#30303031.

Wenn während der Befehlsausführung kein Datenträger erkannt wurde, entfallen die Datentelegramme und es wird nur das Telegramm zur Signalisierung des Befehlsendes gesendet. Die <Anzahl Datenträger> hat dabei den Wert "0000" (ASCII) bzw. 16#30303030.

## Kill (KI)

Der Befehl "Kill" hat den Befehlscode 16#B9 und führt einen einmaligen Schreibzugriff auf einen oder mehrere Transponder innerhalb der Erfassungszone aus. Der Befehl versetzt Transponder in einen Zustand, in dem kein Zugriff mehr möglich ist. Der erfolgreiche Schreibvorgang wird für jeden beschriebenen Transponder über ein Datentelegramm mit den Statuswert 16#00 signalisiert. Das Datentelegramm enthält den EPC/UII des Transponders auf den die Konfiguration geschrieben wurde. Das Ende der Befehlsausführung wird über ein Abschlusstelegramm signalisiert. Das Abschlussdiagramm hat den Statuswert 16#0F und enthält die Anzahl der Transponder, die während der Befehlsausführung beschrieben wurden.

Der Wert von <FrameLength> ist abhängig von der Länge der EPC/UII-Maske. Das Fragment endet mit dem Byte <Pecom Bit>. Der <FragmentationCounter> hat den Wert 16#00, da keine zusätzlichen Fragmente für die Übertragung des Befehlsstelegramms erforderlich sind. Die Länge des Befehlsstelegramms <TelegramLength> ist abhängig von der Länge der EPC/UII-Maske. Das Telegramm endet mit dem Byte <Pecom Bits>, welches immer den Wert 16#00 hat.

Über den Parameter <Length EPC/UII Mask> kann die Länge der EPC/UII-Maske (Bank 01) eines oder mehrere Transponder eingestellt werden, auf die der Befehl wirken soll. Die Länge der Maske darf nicht 0 betragen. Über den Parameter <EPC/UII Mask> werden die Transponder ausgewählt, auf die der Befehl wirken soll. In <Kill Password> muss das zuvor gespeicherte Killpasswort eingetragen werden, welches nicht 0 sein darf.

### Befehl:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#B9							
Byte 6	Length EPC/UII Mask (High Byte)	<Length EPC/UII Mask (High Byte)>							
Byte 7	Length EPC/UII Mask (Low Byte)	<Length EPC/UII Mask (Low Byte)>							
Byte 8	EPC/UII Mask (Byte 0)	<EPC/UII Mask (Byte 0)>							
Byte 9	EPC/UII Mask (Byte 1)	<EPC/UII Mask (Byte 1)>							
...	...	...							
Byte x	EPC/UII Mask (Byte Y)	<EPC/UII Mask (Byte Y)>							
Byte x+0	Nicht relevant	16#00							

2024-08

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte x+1	Kill Password (Byte 0)	<Kill Password (Byte 0)>							
Byte x+2	Kill Password (Byte 1)	<Kill Password (Byte 1)>							
Byte x+3	Kill Password (Byte 2)	<Kill Password (Byte 2)>							
Byte x+4	Kill Password (Byte 3)	<Kill Password (Byte 3)>							
Byte x+5	Payload (Byte 0)	<Payload (Byte 0)>							
Byte x+6	Payload (Byte 1)	<Payload (Byte 1)>							
Byte x+7	Payload (Byte 2)	<Payload (Byte 2)>							
Byte x+8	Pecom Bit	<Pecom Bit>; 16#00							
...	...	...							
Byte 31	Nicht relevant	16#00							

Tabelle 8.41

**Antwort Datentelegramm, Long-Form-Datenformat:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#B9							
Byte 6	Status	16#00							
Byte 7	Parameter / Data	<Length EPC/UII (High Byte)>							
Byte 8	Parameter / Data	<Length EPC/UII (Low Byte)>							
Byte 9	Parameter / Data	<PC-Word (High Byte)>							
Byte 10	Parameter / Data	<PC-Word (Low Byte)>							
Byte 11	Parameter / Data	<EPC/UII Byte 1>							
Byte 12	Parameter / Data	<EPC/UII Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<EPC/UII Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Die Länge des Fragments <FrameLength> ist abhängig von der Größe des EPC/UII des beschriebenen Datenträgers. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <EPC/UII Byte X>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist abhängig von der Länge des EPC/UII des beschriebenen Datenträgers. Die Größe des Telegramms erstreckt sich bis einschließlich <EPC/UII Byte X>.

Der Parameter <Command> hat den Wert 16#B9 und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00.

Hieran schließt sich eine Längenangabe der EPC/UII-Information an. Die Längenangabe hat immer eine Größe von 2 Byte. Die Längenangabe bezieht sich auf die Größe des PC-Worts und des EPC/UII. Danach folgt der EPC/UII zur eindeutigen Identifizierung des beschriebenen Transponders. Alle nachfolgenden Bytes innerhalb des Datentelegramms haben den Wert 16#00.

#### Antwort Befehlsende, Long-Form-Datenformat:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#B9							
Byte 6	Status	16#0F							
Byte 7	Parameter / Data	<Anzahl Datenträger Byte 1>							
Byte 8	Parameter / Data	<Anzahl Datenträger Byte 2>							
Byte 9	Parameter / Data	<Anzahl Datenträger Byte 3>							
Byte 10	Parameter / Data	<Anzahl Datenträger Byte 4>							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Die <FrameLength> ist in der Rückantwort für das Befehlsende des Kill-Befehls konstant bei dem Wert 16#0B. Das Fragment erstreckt sich dabei bis einschließlich <Anzahl Datenträger Byte 4>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die Telegrammlänge <TelegramLength> hat den Wert 16#08. Das Byte <Command> wird zurückgespiegelt und hat denselben Wert wie im Befehlstelegramm 16#B9. Der <Status> für das Telegramm zur Signalisierung des Befehlsendes ist 16#0F.

Die Anzahl der programmierten Datenträger wird innerhalb von 4 Byte übertragen. Dabei wird die Anzahl in ASCII kodierter Darstellung wiedergegeben.

Bei der Identifikation von einem Transponder hat die <Anzahl Datenträger> den Wert "0001" (ASCII) bzw. 16#30303031.

Wenn während der Befehlsausführung kein Datenträger erkannt wurde, entfallen die Datentelegramme und es wird nur das Telegramm zur Signalisierung des Befehlsendes gesendet. Die <Anzahl Datenträger> hat dabei den Wert "0000" (ASCII) bzw. 16#30303030.

## 8.6.2 Systembefehle

### Quit (QU)

Der Befehl "Quit" hat den Befehlscode 16#02 und stoppt die Ausführung eines aktiven Befehls am Gerät. Dadurch werden die Enhanced Read bzw. Enhanced Write-Befehle beendet. Die erfolgreiche Befehlsausführung wird über ein Telegramm mit den Statuswert 16#00 signalisiert.

**Befehl:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.42

Die <FrameLength> hat den Wert 16#06, da keine weiteren Befehlsparameter übertragen werden müssen und das Fragment nach <Command> endet. Der <Fragmentation Counter> hat den Wert 16#00, da keine zusätzlichen Fragmente für die Übertragung des Befehlstelegramms erforderlich sind. Das Befehlstelegramm hat eine Länge (<TelegramLength>) von 16#03, da der Befehl mit dem Byte <Command> endet. Der Befehlscode <Command> für den Befehl Quit ist 16#02.

**Antwort Befehlsende:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.43

Die Länge des Fragments <FrameLength> ist konstant beim Wert 16#07, da keine weiteren Parameter innerhalb der Antwort übertragen werden. Die <FrameLength> beinhaltet alle Bytes bis einschließlich <Status>. Der <Fragmentation Counter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Der Wert von <TelegramLength> ist 16#04 und die Größe des Telegramms erstreckt sich bis einschließlich <Status>.

Der Parameter <Command> hat den Wert 16#02 und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00.

Alle nachfolgenden Bytes innerhalb des Datentelegramms haben den Wert 16#00.

**Version (VE)**

Der Befehl "VE" hat den Befehlscode 16#03 und liest die Firmwareversion des Geräts aus.

**Befehl:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.44

Die <FrameLength> hat den Wert 16#06, da keine weiteren Befehlsparameter übertragen werden müssen und das Fragment nach <Command> endet. Der <Fragmentation Counter> hat den Wert 16#00, da keine zusätzlichen Fragmente für die Übertragung des Befehlstelegramms erforderlich sind. Das Befehlstelegramm hat eine Länge (<Telegram Length>) von 16#03, da der Befehl mit dem Byte <Command> endet. Der Befehlscode <Command> für den Befehl Version ist 16#03.

**Antwort Befehlsende:**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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							
Byte 31	Nicht relevant	16#00							

Tabelle 8.45

**8.6.3 Filterbefehle**

In der Werkseinstellung greift jeder Schreib-/Lesebefehl auf alle erkannten Datenträger in der Erfassungszone zu. Mit Hilfe der Filter kann eine Selektierung durchgeführt werden. Hierdurch kann festgelegt werden, dass Schreib-/Lesebefehle nur noch auf einen einzelnen oder auf mehrere Datenträger zugreift.

Für die Parametrierung des Zugriffs auf bestimmte Transponder ist die Nutzung der Befehle "Set Filter Mask" (FI) und "Activate Filter" (MF) erforderlich. Der Befehl "Set Filter Mask" (FI) stellt Filtermasken bzw. Filterbedingung im Gerät ein. Die Filterung wird durch den Befehl "Activate Filter" (MF) aktiviert.

Die eingestellten Filtermasken können über den Befehl "Read Parameter" (RP) über den Parameter FL ausgelesen werden.

### Set Filter Mask (FI)

Mit Hilfe des Befehls "Set Filter Mask" (FI) lässt sich eine Filtermaske im Gerät parametrieren. Der Befehlscode lautet 16#CA. Es lassen sich bis zu 3 unterschiedliche Filtermasken einstellen.

#### Befehl:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#CA							
Byte 6	Parameter / Data	<FilterNumber>							
Byte 7	Parameter / Data	<FilterCondition>							
Byte 8	Parameter / Data	<FilterAddress (High Byte)>							
Byte 9	Parameter / Data	<FilterAddress (Low Byte)>							
Byte 10	Parameter / Data	<LengthMask (High Byte)>							
Byte 11	Parameter / Data	<Length Mask (Low Byte)>							
Byte 12	Parameter / Data	<Mask Data Byte 1>							
Byte 13	Parameter / Data	<Mask Data Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<Mask Data Byte X>							
...	Parameter / Data	16#00							
Byte 31	Parameter / Data	16#00							

Tabelle 8.46

Der Wert von <FrameLength> ist abhängig davon, wie groß die einzustellende Filtermaske ist. Es wird hierdurch die Länge des Fragments bis einschließlich <Mask Data Byte X> festgelegt.

Der <FragmentationCounter> hat den Wert 16#00, da der Befehl über ein Fragment von der Steuerung übertragen werden kann.

Die <TelegramLength> spezifiziert die Länge des Telegramms beginnend ab der Telegrammlänge selbst und einschließlich des Bytes <Mask Data Byte X>. Die Telegrammlänge ist somit ebenfalls abhängig von der Länge der Filtermaske.

Das Byte <Command> legt den auszuführenden Befehl fest. Für die Ausführung des Befehls "Set Filter Mask" hat das Byte <Command> den Wert 16#CA.

#### Antwort Filtermaske eingestellt:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#07							
Byte 2	Fragmentation Counter	16#00							

Byte	Inhalt	Bit-Nummer							
		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#CA							
Byte 6	Status	16#00							
Byte 7	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.47

Über den Parameter <FilterNumber> wird die Nummer des einzustellenden Filters festgelegt. Das Gerät kann bis zu 3 Filter abspeichern.

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

Der Parameter <FilterCondition> definiert verschiedene Bedingungen, die bei der Filterung der Daten berücksichtigt werden. Das Byte <FilterCondition> ist dabei in Bereiche unterteilt, die die jeweiligen Filterbedingungen repräsentieren.

Inhalt	Bit-Nummer							
	7	6	5	4	3	2	1	0
<FilterCondition>	0	0	0	<MemBank>		<Negation>	<LogicalOperation>	<Truncation>

Tabelle 8.48

**Bit 0** <Truncation>

**0<sub>bin</sub>** Es werden alle Daten des Filters übertragen; Standardwert ist immer 0bin

**1<sub>bin</sub>** Es wird nur der Teil des EPC oder der Ull übertragen, der auf die Maske folgt. Die CRC wird nicht neu berechnet. Es wird das gespeicherte CRC gesendet.

**Bit 1** <LogicalOperation>

Wenn mehr als eine Filterbedingung eingestellt wurde, kann durch das Bit <LogicalOperation> eine logische Verknüpfung von mehreren Filterbedingungen durchgeführt werden. Wenn nur eine Filterbedingung parametrisiert worden ist, hat dieses Bit keine Bedeutung.

**0<sub>bin</sub>** Oder-Verknüpfung bzw. Wert für die Einstellung von nur einen Filter

**1<sub>bin</sub>** Und-Verknüpfung

**Bit 2** <Negation>

Über den Parameter <Negation> kann eine Negation der Filterbedingung eingestellt werden. Es werden dann alle Datenträger übertragen, die nicht zu der Filterbedingung passen.

**0<sub>bin</sub>** nicht negiert; Übertragung aller Datenträger, die zu der Filterbedingung passen

**1<sub>bin</sub>** negiert; Übertragung aller Datenträger, die nicht zu der Filterbedingung passen

**Bit 3/** <MemBank>

**Bit 4** Der Parameter <MemBank> definiert die Speicherbank des Datenträgers auf den die Filterbedingung angewendet werden soll.

**00<sub>bin</sub>** Bank 00; nicht zulässig

**01<sub>bin</sub>** Bank 01; EPC/UII

**10<sub>bin</sub>** Bank 10; TID (Fixcode)

**11<sub>bin</sub>** Bank 11; User Memory

Die Startadresse des Speicherbereichs auf den die Filterbedingung angewendet werden soll, wird durch den Parameter <FilterAddress> festgelegt. Die Adresse ist dabei bezogen auf einzelne Bits des Speichers. In den folgenden Tabellen ist der Aufbau der verschiedenen Speicherbänke mit den zugehörigen Werten für <FilterAddress> ersichtlich. Die Tabellen umfassen nicht den kompletten Aufbau der Speicherbänke.

Die Länge der Filtermaske wird über den Parameter <LengthMask> angegeben. Die Länge wird in Bit angegeben.

Die Filterdaten befinden sich in den Parametern <Mask Data Byte>. Die Länge ist variabel und wird durch <LengthMask> vorgegeben.

**Speicheraufbau Bank 01 (EPC/UII):**

Funktion	Address No. (hex)	Bit-Nummer															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPC/UII	0#40 - 0#4F	EPC/UII Byte 6										EPC/UII Byte 5					
	0#30 - 0#3F	EPC/UII Byte 4										EPC/UII Byte 3					
	0#20 - 0#2F	EPC/UII Byte 2										EPC/UII Byte 1					
Protocol Control Bits <b>PC</b>	0#10 - 0#1F	AFI/RFU0	AFI/RFU1	AFI/RFU2	AFI/RFU3	AFI/RFU4	AFI/RFU5	AFI/RFU6	AFI/RFU7	Toggle	XPC	UMI	Length0	Length1	Length2	Length3	Length4
Cyclic Redundancy Check <b>CRC</b>	0#00 - 0#0F	CRC0	CRC1	CRC2	CRC3	CRC4	CRC5	CRC6	CRC7	CRC8	CRC9	CRC10	CRC11	CRC12	CRC13	CRC14	CRC15

Tabelle 8.49

- CRC0 ... CRC15: 16-Bit Checksumme für die Speicherbank 01; automatische Berechnung der Checksumme durch das Gerät
- Length0 ... Length4: Längenangabe des EPC bezogen auf 2-Byte Wörter
- UMI: User Memory Indicator
- XPC: XPC Indicator
- Toggle: 0 (False) → Bank 01 enthält einen EPC; 1 (True) → Bank 01 enthält einen UII
- AFI/RFU0 ... AFI/RFU7: Attribut Bits
- EPC/UII: Speicherbereich für den EPC bzw. UII; Startadresse für den EPC/UII ist 0#20; Länge EPC/UII in Length0 ... Length4 kodiert

**Speicheraufbau Bank 10 (TID):**

Funktion	Address No. (hex)	Bit-Nummer															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Serial Number Segment [15:0]	0#50 - 0#5F	SN32	SN33	SN34	SN35	SN36	SN37	SN38	SN39	SN40	SN41	SN42	SN43	SN44	SN45	SN46	SN47
Serial Number Segment [31:16]	0#40 - 0#4F	SN16	SN17	SN18	SN19	SN20	SN21	SN22	SN23	SN24	SN25	SN26	SN27	SN28	SN29	SN30	SN31
Serial Number Segment [47:32]	0#30 - 0#3F	SN0	SN1	SN2	SN3	SN4	SN5	SN6	SN7	SN8	SN9	SN10	SN11	SN12	SN13	SN14	SN15
XTID Header Segment	0#20 - 0#2F	TDI0	TDI1	TDI2	TDI3	TDI0	TDI1	TDI2	TDI3	TDI4	TDI5	TDI6	TDI7	TDI8	TDI9	TDI10	TDI11

2024-08

Funktion	Address No. (hex)	Bit-Nummer															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manufacturer ID <b>MDID</b> / /Model Number <b>TMN</b>	0#10 - 0#1F	MD ID5	MD ID6	MD ID7	MD ID8	TM N0	TM N1	TM N2	TM N3	TM N4	TM N5	TM N6	TM N7	TM N8	TM N9	TM N10	TM N11
Manufacturer ID <b>MDID</b>	0#00 - 0#0F	1	1	1	0	0	0	1	0	XTID	S	F	MD ID0	MD ID1	MD ID2	MD ID3	MD ID4

Tabelle 8.50

- Class Identifier: enthält den Wert 16#E0 oder 16#E2 auf Basis der ISO/IEC 15963
- XTID: Extended Tag Identification; 0 → kein Extended Tag Identification; 1 → Datenträger hat Extended Tag Identification
- S: Security Indikator; Datenträger unterstützt zusätzliche sicherheitsrelevante Befehle
- XPC: XPC Indikator
- F: File Indikator; Datenträger unterstützt den Befehl "FileOpen"
- MDID: Mask-Designer Identifier; Chiphersteller
- TMN: Tag Model Number; Chiptyp
- SN: Seriennummer Datenträger

### Zuordnung Datenträger

Hersteller	MDID (bin)	TMN (bin)	TMN (hex)	Chip
Impinj	00000001	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	00000010			
Alien Technology	00000011	010000010001	16#411	Higgs 2
		010000010010	16#412	Higgs 3
		010000010100	16#414	Higgs 4
		100000010001	16#811	Higgs EC
Intelleflex	000000100			
Atmel	000000101			

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

Tabelle 8.51

**Speicheraufbau Bank 11 (User Memory):**

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

Tabelle 8.52

## Activate Filter Mask (MF)

Der Befehl "Activate Filter Mask" (MF) aktiviert die Filterfunktion des Geräts. Der Befehlscode lautet 16#CB. Dabei können 2 unterschiedliche Filtermoden aktiviert werden.

### Befehl:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#07							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#04							
Byte 5	Command	16#CB							
Byte 6	Parameter / Data	<FilterMode>							
Byte 7	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.53

### Antwort Filter aktiviert:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.54

Durch den Parameter <FilterMode> wird der Modus der Filterung eingestellt. Folgende Einstellungen sind möglich:

- 16#00: Filter ausgeschaltet
- 16#01: Filtermodus 1
- 16#02: Filtermodus 2

### Filtermodus 1

Im Erfassungsbereich des Geräts befinden sich insgesamt 15 Transponder, je 5 davon sind durch A, B und C charakterisiert. Der Filter wird nun durch den Befehl FI auf "B" gesetzt.

Wenn Sie den Befehl MF11 (Filter aktivieren - Modus 1) ausführen, hat dies Auswirkungen auf alle folgenden Schreib- und Lesebefehle.

Wenn als nächstes ein Schreibbefehl ausgeführt wird, werden alle Transponder "B" im Erfassungsbereich selektiert und erhalten ein Selektiert-Flag. Der Schreibbefehl wird nur für Transponder ausgeführt, die ein Selektiert-Flag gesetzt haben.

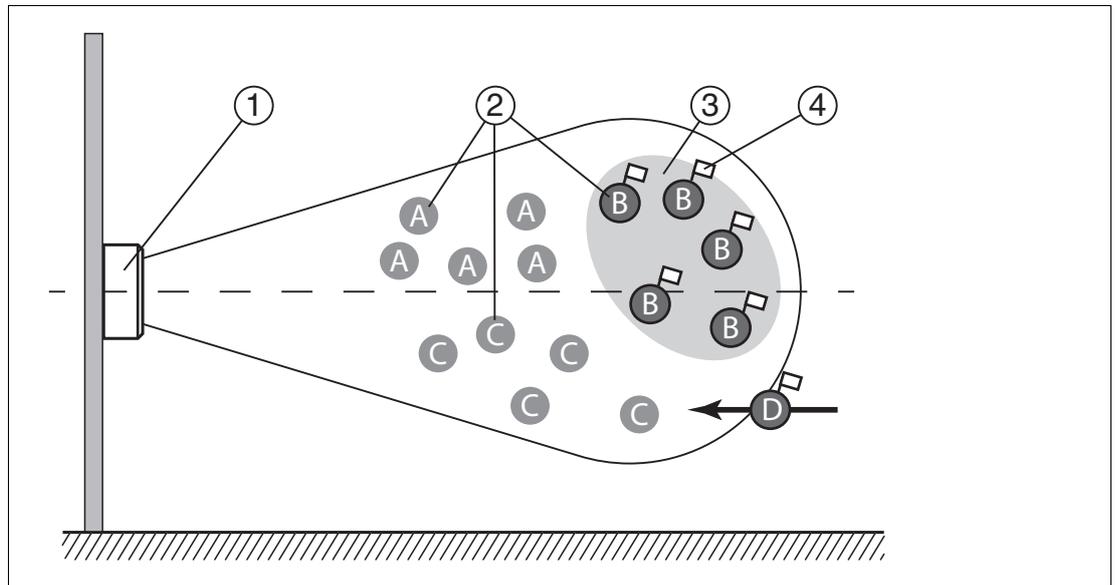


Abbildung 8.1

Filter auf "B" gesetzt, Befehl MF11 (Filter aktiviert - Modus 1) selektiert alle Transponder "B", nachfolgende Befehle sprechen die selektierten Transponder "B" an.

- 1 Gerät
- 2 Transponder "A", "B", "C"
- 3 Filtermaske
- 4 Selektiert-Flag

## Filtermodus 2

Im Erfassungsbereich des Geräts befinden sich insgesamt 15 Transponder, je 5 davon sind durch A, B und C charakterisiert. Der Filter wird durch den Befehl FI auf "B" gesetzt.

Wenn Sie den Befehl MF12 (Filter aktivieren - Modus 2) ausführen, hat die Auswirkungen auf alle folgenden Schreib- und Lesebefehle.

Wenn als nächstes ein Schreibbefehl ausgeführt wird, werden alle Transponder im Erfassungsbereich selektiert, die nicht "B" sind. Diese Transponder erhalten ein Selektiert-Flag. Der Schreibbefehl wird nur für die Transponder ausgeführt, die kein Selektiert-Flag gesetzt haben.

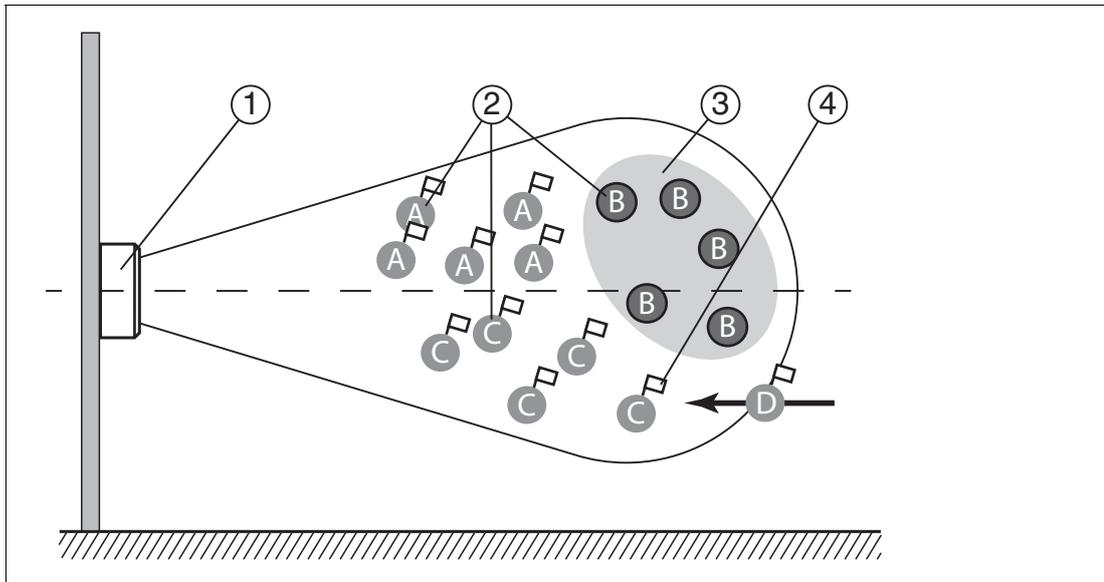


Abbildung 8.2

Filter auf "B" gesetzt, Befehl MF12 (Filter aktiviert - Modus 2) selektiert alle Transponder "A" und "C", nachfolgende Befehle sprechen die **nicht** selektierten Transponder "B" an.

- 1 Gerät
- 2 Transponder "A", "B", "C"
- 3 Filtermaske invertiert
- 4 Selektiert-Flag

## Unterschied zwischen Filtermodus 1 und Filtermodus 2

In beiden Fällen werden die nachfolgenden Befehle nur auf Transponder "B" angewendet, die sich im Erfassungsbereich befinden.

Bei Filtermodus 1 erhalten die Transponder "B" ein Selektiert-Flag. Bei Filtermodus 2 erhalten die Transponder "B" kein Selektiert-Flag.

Wenn ein Transponder "D" mit einem gesetzten Selektiert-Flag aus dem Bereich eines benachbarten Geräts neu in die Erfassungszone gelangt, wird dieser Transponder "D" bei Modus 1 die nachfolgenden Befehle ausführen. Bei Modus 2 hingegen wird dieser Transponder "D" die nachfolgenden Befehle nicht ausführen.

### 8.6.4 UHF-Konfigurationbefehle

Über die Konfigurationbefehle "Read Parameter" und "Write Parameter" lassen sich die Parameter für die UHF-Schnittstelle auslesen bzw. verändern. Hierdurch kann das Verhalten des Geräts auf Seiten der Luftschnittstelle angepasst werden.

Im Auslieferungszustand sind für die UHF-Schnittstelle Parameterwerte voreingestellt. Dabei variiert die Werkseinstellung in Abhängigkeit der Geräteversion.

Alle Parameterwerte werden nicht-flüchtig gespeichert und bleiben nach einer Spannungsunterbrechung unverändert bestehen.

Für den Zugriff auf die UHF-Parameter am Gerät ist die Angabe eines Systemcodes erforderlich. Hierdurch wird zwischen anderen Systemen unterschieden, bei denen ebenfalls Parametersätze verändert werden können. Bei diesem Gerät lautet der Systemcode "U" (16#55).

### 8.6.4.1 Grundlegende Befehlsstruktur

#### Read Parameter (RP)

Der Befehl "Read Parameter" (RP) hat den Befehlscode 16#BE und es wird dadurch ein Parameter der UHF-Einstellung ausgelesen.

##### Befehl:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	SystemCode	16#55 "U"							
Byte 7	Parameter Name (High Byte)	<ParameterName (High Byte)>							
Byte 8	Parameter Name (Low Byte)	<ParameterName (Low Byte)>							
Byte 9	Parameter Length (High Byte)	<ParameterLength (High Byte)>							
Byte 10	Parameter Length (Low Byte)	<ParameterLength (Low Byte)>							
Byte 11	Parameter Data Byte 1	<Parameter Data Byte 1>							
Byte 12	Parameter Data Byte 2	<Parameter Data Byte 2>							
...	...	...							
...	Parameter Data Byte X	<Parameter Data Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.55

Der Wert von <FrameLength> ist abhängig davon, ob bei dem Befehl "Read Parameter" noch Parameterwerte mit übertragen werden sollen. Es wird hierdurch die Länge des Fragments bis einschließlich <Parameter Data Byte X> festgelegt. Bei der überwiegenden Mehrzahl der Parameter sind bei der Ausführung des Lesezugriffs keine zusätzlichen Parameterwerte erforderlich. In diesen Fällen endet das Fragment bei <ParameterLength (Low Byte)> und der Wert ist 16#0B.

Der <FragmentationCounter> hat den Wert 16#00, da der Befehl über ein Fragment von der Steuerung übertragen werden kann.

Die <TelegramLength> spezifiziert die Länge des Telegramms beginnend ab der Telegrammlänge selbst und einschließlich des Bytes <Parameter Data Byte X>. Wenn der Befehl keine zusätzlichen Parameterwerte besitzt, endet das Telegramm bei <ParameterLength (Low Byte)> und die <TelegramLength> beträgt 16#08.

Das Byte <Command> legt den auszuführenden Befehl fest. Für die Ausführung des Befehls "Read Parameter" hat das Byte <Command> den Wert 16#BE.

Der <SystemCode> für das Gerät ist 16#55 ("U").

Der Parameter <ParameterName> legt den Parameter fest, der eingelesen werden soll. Der Wert von <ParameterName> entspricht den 2 Zeichen der Parameterkürzel. Achten Sie auf Groß- und Kleinschreibung.

Über <ParameterLength> wird eine Längenangabe für einen Parametersatz innerhalb des Befehls "Read Parameter" vorgenommen.

Bei einigen Parametern beinhaltet der Befehl "Read Parameter" noch einen Parametersatz <ParameterData>. Die Länge ist abhängig vom zugehörigen Parameter.

Alle anderen Bytes des Befehlsfragments sind nicht für den Befehl relevant. Sie sind alle auf den Wert 16#00 zusetzen.

Beispiel: Befehlsstelegramm zum Auslesen der Sendeleistung PT (16#5054)

Name	Adresse	Anze...	Beobachtungswert	Steuerwert
"ControlByte_Out"	%QB0	Bin	2#1110_0000	2#1110_0000
"FrameLength_Out"	%QB1	Hex	16#08	16#08
"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

Abbildung 8.3

Antwort:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.56

Der Wert der <FrameLength> ist abhängig von der Größe des eingelesenen Parameters. Durch die <FrameLength> wird die Länge des Fragments bis einschließlich <Parameter Data Byte X> angegeben.

Der <FragmentationCounter> hat den Wert 16#00, da die Befehlsantwort über ein Fragment vom Gerät übertragen werden kann.

Die <TelegramLength> spezifiziert die Länge des Antworttelegramms beginnend ab der Telegrammlänge selbst und einschließlich des Bytes <Parameter Data Byte X>. Der Wert der Telegrammlänge für diese Befehlsantwort ist abhängig von der Länge des eingelesenen Parameters.

Das Byte <Command> enthält den gespiegelten Befehlscode. Bei der Ausführung eines "Read Parameter"-Befehls hat das Byte <Command> den Wert 16#BE in der Rückantwort.

Das Byte <Status> hat den Wert 16#00 und signalisiert dadurch, dass der Befehl korrekt ausgeführt wurde und dieses Telegramm den eingelesenen Parameterwert enthält. Wenn sich bei <Status> ein anderer Wert befindet, ist ein Fehler aufgetreten.

Hieran schließen sich die eingelesenen Parameterwerte <ParameterByte> an. Die Anzahl der Parameterwerte ist variabel.

Alle anderen Bytes des Antwortfragments sind nicht relevant und haben den Wert 16#00.

Beispiel: Antworttelegramm zum Auslesen der Sendeleistung PT (16#5054)

Name	Adresse	Anze...	Beobachtungswert
*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

Abbildung 8.4

### Write Parameter (WP)

Der Befehl "Write Parameter" (WP) hat den Befehlscode 16#BF. Der Befehl kann dazu verwendet werden, Parameter der UHF-Einstellung ändern.

#### Befehl:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	SystemCode	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>							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.57

Der Wert von <FrameLength> ist abhängig davon, wie viele Parameterwerte <Parameter Data Byte> bei dem Befehl "Write Parameter" übertragen werden sollen. Es wird hierdurch die Länge des Fragments bis einschließlich <Parameter Data Byte X> festgelegt.

Der <FragmentationCounter> hat den Wert 16#00, da der Befehl über ein Fragment von der Steuerung übertragen werden kann.

Die <TelegramLength> spezifiziert die Länge des Telegramms beginnend ab der Telegrammlänge selbst und einschließlich des Bytes <Parameter Data Byte X>.

Das Byte <Command> legt den auszuführenden Befehl fest. Für die Ausführung des Befehls "Write Parameter" hat das Byte <Command> den Wert 16#BF.

Der <SystemCode> für das Gerät ist 16#55 ("U").

Der Parameter <ParameterName> legt den Parameter fest der eingelesen werden soll. Der Wert von <ParameterName> entspricht den 2 Zeichen der Parameterkürzel.

Über <LengthParameter> wird eine Längenangabe für einen Parametersatz innerhalb des Befehls "Write Parameter" vorgenommen.

Die Länge des Parametersatzes <Parameter Data Byte> ist variabel und parameterabhängig.

Alle anderen Bytes des Befehlsfragments sind nicht für den Befehl relevant. Sie sind alle auf den Wert 16#00 zusetzen.

#### Antwort:

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#07							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#04							
Byte 5	Command	16#BF							
Byte 6	Status	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.58

Der Wert der <FrameLength> beträgt 16#07. Durch die <FrameLength> wird die Länge des Fragments bis einschließlich des Bytes <Status> angegeben.

Der <FragmentationCounter> hat den Wert 16#00, da die Befehlsantwort über ein Fragment vom Gerät übertragen werden kann.

Die <TelegramLength> spezifiziert die Länge des Antworttelegramms beginnend ab der Telegrammlänge selbst und einschließlich des Bytes <Status>. Der Wert der Telegrammlänge für diese Befehlsantwort ist 16#04.

Das Byte <Command> enthält den gespiegelten Befehlscode. Bei der Ausführung eines "Write Parameter"-Befehls hat das Byte <Command> den Wert 16#BF in der Rückantwort.

Wenn der <Status> den Wert 16#00 hat, wurde der Befehl "Write Parameter" erfolgreich ausgeführt. Ein abweichender Wert bei <Status> signalisiert einen Fehler.

Alle anderen Bytes des Antwortfragments sind nicht relevant und haben den Wert 16#00.

Beispiel: Antworttelegramm zum Ändern der Sendeleistung PT (16#5054)

Name	Adresse	Anze...	Beobachtungswert
*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

Abbildung 8.5

### 8.6.4.2 Übersicht UHF-Parameter

Mit den Konfigurationsbefehlen Read Parameter (RP) und Write Parameter (WP) können Sie die folgenden Parameter lesen bzw. schreiben:

Parameterkürzel	Parametername	Lesen / Schreiben
PT 16#5054	Power Transmit PT siehe "Power Transmit PT (Sendeleistung)" auf Seite 114 Einstellung der Sendeleistung des Geräts	lesen + schreiben
NT 16#4E54	Number of Tags NT siehe "Number of Tags NT (Abbruchkriterium)" auf Seite 118 Abbruchkriterium Schreib-/Lesebefehl	lesen + schreiben
AP 16#4150	Antennenpolarisation AP siehe "Antennenpolarisation AP" auf Seite 120 Einstellung der Polarisationsebene des Geräts	lesen
TA 16#5441	Tries Allowed TA siehe "Tries Allowed TA (Anzahl der Zugriffsversuche)" auf Seite 121 Anzahl Zugriffsversuche auf Datenträger	lesen + schreiben
CD 16#4344	Sendekanäle CD siehe "Sendekanäle CD" auf Seite 123 Einstellung der verwendeten Sendekanäle	<ul style="list-style-type: none"> <li>• Geräteversion FR1*: lesen + schreiben</li> <li>• Geräteversion FR2*: lesen</li> </ul>
NC 16#4E43	Number of Channels NC siehe "Number of Channels NC" auf Seite 130 Einstellung der verwendeten Sendekanäle	<ul style="list-style-type: none"> <li>• Geräteversion FR1*: lesen</li> <li>• Geräteversion FR2*: lesen + schreiben</li> </ul>
E5 16#4535	Tag Lost Smoothing (E5) siehe "Tag Lost Smoothing (E5)" auf Seite 126 Einstellung der Anzahl erfolgloser Zugriffe auf einen Datenträger vor Status 5 Meldung	lesen + schreiben
MB 16#4D42	Memory Bank MB siehe "Memory Bank MB" auf Seite 128 Einstellung der Speicherbank für versch. Schreib-/Lesebefehle	lesen + schreiben
QW 16#5157	Q-Wert QW siehe "Q-Wert QW" auf Seite 133 Einstellung der Zeitschlitz für den Zugriff auf Datenträger	lesen + schreiben
RC 16#5243	Region Code RC siehe "Region Code RC" auf Seite 135 Länderkennung	lesen
FL 16#464C	Auslesen der Filtermaske	lesen

Tabelle 8.59

Die Parameter werden nicht flüchtig im Gerät gespeichert.

### Power Transmit PT (Sendeleistung)

Durch den Parameter PT (Power Transmit) kann die Sendeleistung des Geräts verändert werden. Die Angabe der Sendeleistung erfolgt dabei in mW. Die Sendeleistung hat Einfluss auf die Reichweite der Erfassungszone eines Transponders. Mit einer Vergrößerung der Sendeleistung wird bis zu einem gewissen Grad die Reichweite für den Lese- und Schreibzugriff auf einen Transponder vergrößert.

In der werksseitigen Einstellung ist eine Sendeleistungsstufe PT1 voreingestellt. Es ist möglich, bis zu 5 Leistungsstufen (PT1 ... PT5) im Gerät einzustellen. Hierdurch lässt sich eine Rampe für eine kontinuierliche Steigerung der Sendeleistung generieren. In Verbindung mit der Nutzung eines "Single Read"- bzw. "Single Write"-Befehls und der Definition des Abbruchkriteriums (Parameter NT, Number of Tags) wird der Lese- bzw. Schreibvorgang abgebrochen, sobald die eingestellte Anzahl an Datenträger erkannt wurde.

Bei der Einstellung von mehreren Sendeleistungsstufen verlängert sich die Ausführungszeit eines Lese- bzw. Schreibvorgangs. Ursache hierfür sind die Inventory-Durchläufe, die für jede Leistungsstufe ausgeführt werden. Über die Einstellung des Abbruchkriteriums NT lässt sich die Ausführungszeit verringern.

Parameterzeichen	PT (16#5054)
Länge Parameterwert PTx	2 Byte
Werkseinstellung PT1	variabel in Abhängigkeit der Geräteversion <ul style="list-style-type: none"> <li>• FR1-01 → 100 mW<sub>erp</sub> (16#0064)</li> <li>• FR2-02 → 150 mW<sub>eirp</sub> (16#0096)</li> <li>• FR2-03 → 100 mW<sub>erp</sub> (16#0064)</li> <li>• FR1-04 → 100 mW<sub>erp</sub> (16#0064)</li> <li>• FR2-07 → 150 mW<sub>eirp</sub> (16#0096)</li> <li>• FR2-09 → 150 mW<sub>eirp</sub> (16#0096)</li> <li>• FR2-10 → 150 mW<sub>eirp</sub> (16#0096)</li> <li>• FR2-13 → 100 mW<sub>erp</sub> (16#0064)</li> <li>• FR2-17 → 100 mW<sub>erp</sub> (16#0064)</li> </ul>

Wertebereich FR1-01, FR2-03, FR1-04, FR2-13 und FR2-17

3 mW (16#0003)	-> 5 dBm
4 mW (16#0004)	-> 6 dBm
5 mW (16#0005)	-> 7 dBm
6 mW (16#0006)	-> 8 dBm
8 mW (16#0008)	-> 9 dBm
10 mW (16#000A)	-> 10 dBm
13 mW (16#000D)	-> 11 dBm
15 mW (16#000F)	-> 12 dBm
20 mW (16#0014)	-> 13 dBm
25 mW (16#0019)	-> 14 dBm
30 mW (16#001E)	-> 15 dBm
40 mW (16#0028)	-> 16 dBm
50 mW (16#0032)	-> 17 dBm
60 mW (16#003C)	-> 18 dBm
80 mW (16#0050)	-> 19 dBm
100 mW (16#0064)	-> 20 dBm

Wertebereich FR2-02, FR2-07, FR2-09 und FR2-10

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

- ⊕ Höhere Reichweite, wenn Sie die Sendeleistung erhöhen.
- ⊖ Evtl. Überreichweiten, wenn Sie die Sendeleistung erhöhen.
- ⊖ Durch erhöhte Reichweite evtl. Beeinflussung benachbarter Geräte.



#### Tipp

Die höchste Sendeleistung muss nicht zur größten Lesereichweite führen. Variieren Sie die Sendeleistung, um das optimale Leseergebnis zu erzielen.



#### Hinweis!

Sie können das Gerät nur mit intern vorgegebenen Sendeleistungsstufen betreiben. Softwareseitig können Sie mit dem Befehl WP eine oder mehrere beliebige Sendeleistungsstufen innerhalb des vorgegebenen Wertebereichs eingeben. Das Gerät stellt die Sendeleistung automatisch auf den nächsttieferen zur Verfügung stehenden Wert ein.

Eingaben außerhalb des vorgegebenen Wertebereiches werden als Fehler mit den Statuswert 16#04 zurückgemeldet.



### Tip

Sie haben die Möglichkeit, das Gerät mit mehreren Sendeleistungsstufen zu parametrieren: Über den Befehl WP können die Leistungsstufen PT1, PT2 und PT3 auf die Werte 20 mW, 50 mW und 100 mW eingestellt werden.

- Abbruchkriterium NT auf 16#FF (Werkseinstellung):  
Jeder Schreib- und Lesebefehl wird für alle 3 Sendeleistungswerte nacheinander ausgeführt. Wenn bei der ersten Sendeleistung ein oder mehrere Transponder gefunden und erfolgreich gelesen/beschrieben werden, wird der Befehl trotzdem mit allen weiteren Sendeleistungen durchgeführt, um evtl. weitere Transponder zu erreichen.
- Abbruchkriterium NT auf 16#01 (Abbruch nach einen Datenträger):  
Wenn innerhalb der Inventory-Durchläufe kein Datenträger erkannt wurde, wird die Sendeleistung auf den nächst größeren Wert gestellt. Sobald mindestens ein Datenträger in einer Inventory-Runde erkannt wurde, bricht der Schreib-/Lesebefehl ab.  
Sie haben die Möglichkeit, maximal 5 Sendeleistungswerte anzugeben. Wenn zusätzlich mehrere Sendekanäle ausgewählt sind (siehe "Sendekanäle CD" auf Seite 123), werden für jeden Schreib- oder Lesebefehl alle eingestellten Leistungen bei jedem Sendekanal durchgeführt.

### Beispiel: Befehlstelegramm für die Änderung der Sendeleistung auf PT1 = 50 mW (16#0032)

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	SystemCode	16#55 "U"							
Byte 7	ParameterName (High Byte)	16#50 "P"							
Byte 8	ParameterName (Low Byte)	16#54 "T"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#02							
Byte 11	Parameter PT1 (High Byte)	16#00							
Byte 12	Parameter PT1 (Low Byte)	16#32							
Byte 13	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.60

### Beispiel: Befehlstelegramm für die Änderung der Sendeleistung auf PT1 = 20 mW (16#0014), PT2 = 50 mW (16#0032) und PT3 = 100 mW (16#0064)

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							

2024-08

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#BF							
Byte 6	SystemCode	16#55 "U"							
Byte 7	ParameterName (High Byte)	16#50 "P"							
Byte 8	ParameterName (Low Byte)	16#54 "T"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#06							
Byte 11	Parameter PT1 (High Byte)	16#00							
Byte 12	Parameter PT1 (Low Byte)	16#14							
Byte 13	Parameter PT2 (High Byte)	16#00							
Byte 14	Parameter PT2 (Low Byte)	16#32							
Byte 15	Parameter PT3 (High Byte)	16#00							
Byte 16	Parameter PT3 (Low Byte)	16#64							
Byte 17	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.61

**Beispiel: Befehlstelegramm zum Auslesen der Sendeleistung**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	$D_S$	$U_M$	$U_D$	0	16#0			
Byte 1	Frame Length	16#0A							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (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#50 "P"							
Byte 8	ParameterName (Low Byte)	16#54 "T"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#00							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.62

**Beispiel: Antworttelegramm mit der eingestellten Sendeleistung von PT1 = 50 mW (16#0032)**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Parameter PT1 (High Byte)	16#00							
Byte 8	Parameter PT1 (Low Byte)	16#32							
Byte 9	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.63

### Number of Tags NT (Abbruchkriterium)

Der Parameter NT gibt die Anzahl der Transponder im Erfassungsbereich an, die das Gerät sucht. Jeder Befehl wird entsprechend der Parameter Sendeleistung (PT), Sendekanal (CD) bzw. Anzahl Kanäle (NC) und Anzahl Versuche (TA) wiederholt. Falls die Anzahl der gefundenen Transponder während der Wiederholungen den Wert NT erreicht oder überschreitet, werden alle weiteren Durchläufe abgebrochen. Der Befehl wird beendet und die Daten werden ausgegeben.

Wenn die Anzahl der Transponder auf 255 (=16#FF) gesetzt wird, ist die Funktion ausgeschaltet. Dieser Parameter hat keine Auswirkung auf "Enhanced"-Befehle, sondern nur auf "Single"-Befehle.

Parameterzeichen	NT (16#4E54)
Länge Parameterwert NT	1 Byte
Werkseinstellung	16#FF
Wertebereich	16#01 ... 16#14; 16#FF

**Beispiel: Befehlstelegramm für die Änderung von "Number of Tags" auf den Wert 16#01**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	SystemCode	16#55 "U"							
Byte 7	ParameterName (High Byte)	16#4E "N"							

2024-08

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 8	ParameterName (Low Byte)	16#54 "T"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#01							
Byte 11	Parameter NT	16#01							
Byte 12	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.64

**Beispiel: Befehlstelegramm zum Auslesen von "Number of Tags"**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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#4E "N"							
Byte 8	ParameterName (Low Byte)	16#54 "T"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#00							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.65

**Beispiel: Antworttelegramm mit dem eingestellten Wert von "Number of Tags" = 255 (16#FF)**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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 NT	16#FF							
Byte 8	Nicht relevant	16#00							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.66

## Antennenpolarisation AP

Über den Parameter Antennenpolarisation (AP) lässt sich der Polarisationsmodus des Geräts auslesen. Die integrierte Antenne unterstützt eine zirkulare Polarisation.

Parameterzeichen	AP (16#4150)
Länge Parameterwert NT	1 Byte
Werkseinstellung	"L" (Left Hand Circular Polarisation) = 16#4C

Durch die zirkulare Polarisation ist gewährleistet, dass die Transponder unabhängig von ihrer Polarisation und Ausrichtung identifiziert werden können.

### Beispiel: Befehlstelegramm zum Auslesen der Antennenpolarisation

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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#41 "A"							
Byte 8	ParameterName (Low Byte)	16#50 "P"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#00							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.67

### Beispiel: Antworttelegramm mit dem eingestellten Wert der Antennenpolarisation "L" (Left Hand Circular)

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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							

2024-08

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 5	Command	16#BE							
Byte 6	Status	16#00							
Byte 7	Parameter AP	16#4C "L"							
Byte 8	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.68

### Tries Allowed TA (Anzahl der Zugriffsversuche)

Durch den Parameter Tries Allowed (TA) lässt sich die Anzahl der Zugriffsversuche während der Ausführung einer Schreib-/Leseoperation auf einen Datenträger einstellen.

Parameterzeichen	TA (16#5441)
Länge Parameterwert TA	1 Byte
Werkseinstellung	16#02 → 2 Zugriffsversuche
Wertebereich	16#01 ... 16#0A

Dieser Parameter hat Einfluss auf die Ausführungsdauer von Schreib- und Lesebefehlen. Wenn der Parameterwert von Tries Allowed vergrößert wird, steigt die Ausführungsdauer eines Befehls, da mehr Zugriffsversuche durchgeführt werden.

Durch eine Vergrößerung des Parameterwerts kann bei einer instabilen Kommunikation zwischen Gerät und Datenträger die Zuverlässigkeit für das Schreiben und Lesen der Transponderdaten erhöht werden.

Um die Steigerung der Ausführungszeit durch die Erhöhung von Tries Allowed zu begrenzen, empfiehlt sich die Parametrierung des Abbruchkriteriums NT. Dadurch stoppt die Befehlsausführung, sobald die eingestellte Anzahl an Datenträger identifiziert wurde.

#### Beispiel: Befehlstelegramm für die Änderung von TA auf einen Wert 5

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	SystemCode	16#55 "U"							
Byte 7	ParameterName (High Byte)	16#54 "T"							
Byte 8	ParameterName (Low Byte)	16#41 "A"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#01							
Byte 11	Parameter TA	16#05							
Byte 12	Nicht relevant	16#00							
...	Nicht relevant	16#00							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 31	Nicht relevant	16#00							

Tabelle 8.69

**Beispiel: Befehlstelegramm zum Auslesen von TA**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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#54 "T"							
Byte 8	ParameterName (Low Byte)	16#41 "A"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#00							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.70

**Beispiel: Antworttelegramm mit dem eingestellten Wert 16#02 von TA**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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 TA	16#02							
Byte 8	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.71

## Sendekanäle CD

Über den Parameter Sendekanäle CD wird auf die Sendekanäle zugegriffen, die das Gerät verwendet.

- Geräteversion FR1\*:  
Diese Geräte nutzen eine parametrierbare Frequenzliste. Hierdurch kann der Anwender durch den Parameter CD die Sendekanäle und deren Reihenfolge selbst festlegen.
- Geräteversion FR2\*:  
Diese Gerätevarianten nutzen ein Frequenzsprungverfahren. Eine Änderung der Sendekanäle und der Reihenfolge ist durch den Parameter CD nicht möglich. In diesem Fall kann die Einstellung nur ausgelesen werden.

Parameterzeichen	CD (16#4344)
Länge Parameterwert CD	variabel in Abhängigkeit der Geräteversion <ul style="list-style-type: none"> <li>• FR1-01 → max. 4 Bytes; max. 4 Kanäle einstellbar</li> <li>• FR2-02 → 50 Bytes; 50 Kanäle unveränderbar</li> <li>• FR2-03 → 16 Bytes; 16 Kanäle unveränderbar</li> <li>• FR1-04 → max. 10 Bytes; max. 10 Kanäle einstellbar</li> <li>• FR2-07 → 52 Bytes; 52 Kanäle unveränderbar</li> <li>• FR2-09 → 6 Bytes; 6 Kanäle unveränderbar</li> <li>• FR2-10 → 12 Bytes; 12 Kanäle unveränderbar</li> <li>• FR2-13 → 8 Bytes; 8 Kanäle unveränderbar</li> <li>• FR2-17 → 14 Bytes; 14 Kanäle unveränderbar</li> </ul>

## Werkseinstellung

## in Abhängigkeit der Geräteversion

- FR1-01 → 4, 7, 10, 13 (alle 4 Kanäle; Länge 4 Bytes)
- FR2-02 → 31, 20, 49, 4, 33, ... (Länge 50 Bytes)
- FR2-03 → 3, 16, 13, 10, 7, ... (Länge 16 Bytes)
- FR1-04 → 1, 4, 7, 10 (4 Kanäle; Länge 4 Bytes)
- FR2-07 → 38, 20, 42, 9, 27, ... (Länge 52 Bytes)
- FR2-09 → 1, 7, 13, 4, 10, 16 (Länge 6 Bytes)
- FR2-10 → 5, 10, 3, 8, 1, ... (Länge 12 Bytes)
- FR2-13 → 1, 6, 3, 8, 5, ... (Länge 8 Bytes)
- FR2-17 → 3, 8, 14, 4, 7, ... (Länge 14 Bytes)

## Wertebereich

## in Abhängigkeit der Geräteversion

- FR1-01 → es können bis zu 4 Kanäle eingestellt werden; die Anzahl und die Reihenfolge der Kanäle ist frei einstellbar  
16#04 → Kanal 4  
16#07 → Kanal 7  
16#0A → Kanal 10  
16#0D → Kanal 13
- FR2-02 → die Parameterwerte sind fest eingestellt; eine Änderung ist nicht möglich; der Wertebereich entspricht der Werkseinstellung
- FR2-03 → die Parameterwerte sind fest eingestellt; eine Änderung ist nicht möglich; der Wertebereich entspricht der Werkseinstellung
- FR1-04 → es können bis zu 10 Kanäle eingestellt werden; die Anzahl und die Reihenfolge der Kanäle ist frei einstellbar  
16#01 → Kanal 1  
16#02 → Kanal 2  
16#03 → Kanal 3  
16#04 → Kanal 4  
16#05 → Kanal 5  
16#06 → Kanal 6  
16#07 → Kanal 7  
16#08 → Kanal 8  
16#09 → Kanal 9  
16#0A → Kanal 10
- FR2-07 → die Parameterwerte sind fest eingestellt; eine Änderung ist nicht möglich; der Wertebereich entspricht der Werkseinstellung
- FR2-09 → die Parameterwerte sind fest eingestellt; eine Änderung ist nicht möglich; der Wertebereich entspricht der Werkseinstellung
- FR2-10 → die Parameterwerte sind fest eingestellt; eine Änderung ist nicht möglich; der Wertebereich entspricht der Werkseinstellung
- FR2-13 → die Parameterwerte sind fest eingestellt; eine Änderung ist nicht möglich; der Wertebereich entspricht der Werkseinstellung
- FR2-17 → die Parameterwerte sind fest eingestellt; eine Änderung ist nicht möglich; der Wertebereich entspricht der Werkseinstellung

Wenn Sie mit dem Parameter CD mehrere Sendekanäle parametrieren, führt das Gerät jeden Schreib- oder Lesebefehl sequentiell entsprechend der festgelegten Reihenfolge auf allen festgelegten Sendekanälen durch. Wenn zusätzlich mehrere Sendeleistungswerte parametrieren sind, werden für jeden Schreib- oder Lesebefehl alle eingestellten Sendeleistungen bei jedem Senderkanal durchgeführt.

Die Anzahl der parametrisierten Sendekanäle hat einen Einfluss auf die Ausführungszeit von Schreib-/Lesebefehlen. Bei der Geräteversion FR1-01 kann die Ausführungszeit durch eine Reduktion der Sendekanäle verringert werden. Wenn nur ein Sendekanal verwendet werden soll, empfehlen wir, einer der beiden im Frequenzbereich mittig liegenden Kanäle einzustellen, Kanal 7 oder Kanal 10. Bei der Parametrierung von 2 Sendekanälen sind die beiden im Frequenzbereich außen liegenden Kanäle einzustellen, Kanal 4 und Kanal 13.

#### Beispiel: Befehlstelegramm für die Änderung des Parameters CD auf Kanal 4 und Kanal 13

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	SystemCode	16#55 "U"							
Byte 7	ParameterName (High Byte)	16#43 "C"							
Byte 8	ParameterName (Low Byte)	16#44 "D"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#02							
Byte 11	Parameter CD	16#04							
Byte 12	Parameter CD	16#0D							
Byte 13	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.72

#### Beispiel: Befehlstelegramm zum Auslesen von CD

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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#43 "C"							
Byte 8	ParameterName (Low Byte)	16#44 "D"							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#00							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.73

### Beispiel: Antworttelegramm mit der eingestellten Reihenfolge der Kanäle 4, 10, 7 und 13

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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 CD	16#04							
Byte 8	Parameter CD	16#0A							
Byte 9	Parameter CD	16#07							
Byte 10	Parameter CD	16#0D							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.74

## Tag Lost Smoothing (E5)

Durch den Parameter Tag Lost Smoothing (E5) kann bei der Ausführung eines Enhanced-Befehls die Anzahl der erfolglosen Schreib-/Leseversuche eingestellt werden, bevor das Gerät das Telegramm mit dem Statuswert 16#05 ausgibt. Über das Status 16#05 Telegramm signalisiert das Gerät, dass ein Transponder den Erfassungsbereich verlassen hat bzw. nicht mehr identifiziert werden konnte.

Dieser Parameter wird nur bei der Ausführung von Enhanced-Befehlen berücksichtigt. Bei der Ausführung von Single-Befehlen hat der Parameter keine Bedeutung.

Parameterzeichen	E5 (16#4535)
Länge Parameterwert E5	1 Byte
Werkseinstellung	16#05 → 5 erfolglose Schreibleseversuche bis zur Ausgabe des Status 16#05 Telegramms
Wertebereich	16#00 ... 16#0A → 0 bis maximal 10 Schreib-/Leseversuche

Der Wert für den Parameter E5 kann bei einer instabilen Kommunikation zwischen Transponder und Gerät vergrößert werden. Dadurch verringert sich die Anzahl der empfangenen Status 16#05 Telegramme. Bei dynamischen Anwendungen lassen sich somit Lücken im Erfassungsbereich ausblenden, ohne dass bei kleinen Unterbrechungen der Datenträgerkommunikation eine Status-16#05-Meldung empfangen wird. Die Erfassungszone wirkt dadurch homogener.

Wenn in einer dynamischen Anwendung viele Datenträger gleichzeitig erfasst werden, kann durch eine Vergrößerung des Parameterwerts von E5 der Empfang der Status 16#05 Telegramme zeitlich nach hinten verschoben werden. Dadurch können zunächst die Status 16#00 Telegramme mit den eingelesenen Informationen der Transponder übertragen werden. Die Status 16#05 Meldungen, dass der Datenträger den Erfassungsbereich verlassen hat, werden zeitlich versetzt übermittelt.

Durch eine Verringerung des Parameterwertes von E5 verkürzt sich die Reaktionszeit des Systems, wenn ein Datenträger den Erfassungsbereich verlässt. Die Status 16#05 Telegramme werden schneller gesendet.

Die Übertragung folgender Telegramme wird durch die Einstellung des Parameters E5 nicht beeinflusst und werden sofort übertragen:

- Status 16#00: Ausführung erfolgreich; Daten gelesen bzw. geschrieben
- Status 16#0A: mehrere Datenträger mit gleichen EPC erkannt

#### Beispiel: Befehlstelegramm für die Änderung der Einstellung von E5 auf den Wert 10 (16#0A)

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	SystemCode	16#55 "U"							
Byte 7	ParameterName (High Byte)	16#45 "E"							
Byte 8	ParameterName (Low Byte)	16#35 "5"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#01							
Byte 11	Parameter E5	16#0A							
Byte 12	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.75

#### Beispiel: Befehlstelegramm zum Auslesen der Einstellung von E5

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 5	Command	16#BE							
Byte 6	SystemCode	16#55 "U"							
Byte 7	ParameterName (High Byte)	16#45 "E"							
Byte 8	ParameterName (Low Byte)	16#35 "5"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#00							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.76

### Beispiel: Antworttelegramm mit dem eingestellten Wert (16#05) des Parameters E5

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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 E5	16#05							
Byte 8	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.77

## Memory Bank MB

Über den Parameter Memory Bank MB ist es möglich eine Speicherbank des Transponders auszuwählen, auf die die Schreib-/Lesebefehle für 4-Byte-Blöcke und 2-Byte-Worte zugreifen.

Die Einstellung der Speicherbank beeinflusst den Datenzugriff von folgenden Befehlen:

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

Die Ausführung folgender Lese-/Schreibbefehle wird durch die Parametereinstellung vom MB nicht beeinflusst.

- Single / Enhanced Read Fixcode (Zugriff auf die TID)
- Single / Enhanced Read EPC/UII (Zugriff auf EPC/UII)
- Single Write EPC/UII (Zugriff auf EPC/UII)

Parameterzeichen                      MB (16#4D42)

Länge Parameterwert MB              1 Byte

Werkseinstellung	16#03 → Zugriff auf die Speicherbank "User Memory"
Wertebereich	16#00 → Zugriff auf die Speicherbank mit den Passwortbereich (Bank 00) 16#01 → Zugriff auf die Speicherbank EPC/UII (Bank 01) 16#02 → Zugriff auf die Speicherbank TID (Bank 10) 16#03 → Zugriff auf die Speicherbank User Memory (Bank 11)

**Beispiel: Befehlstelegramm für die Änderung der Einstellung von MB für den Zugriff auf die Speicherbank EPC/UII**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	SystemCode	16#55 "U"							
Byte 7	ParameterName (High Byte)	16#4D "M"							
Byte 8	ParameterName (Low Byte)	16#42 "B"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#01							
Byte 11	Parameter MB	16#01							
Byte 12	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.78

**Beispiel: Befehlstelegramm zum Auslesen der Einstellung des Parameters MB**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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#4D "M"							
Byte 8	ParameterName (Low Byte)	16#42 "B"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#00							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 31	Nicht relevant	16#00							

Tabelle 8.79

### Beispiel: Antworttelegramm mit dem eingestellten Wert 16#03 (-> User Memory) des Parameters MB

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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 MB	16#03							
Byte 8	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.80

## Number of Channels NC

Der Parameter NC gibt an, auf wie vielen Frequenzkanälen ein einmaliger Schreib- oder Leseversuch durchgeführt wird. Der Zugriff auf den Parameter NC ist abhängig von der verwendeten Gerätevariante.

- Geräteversion FR1\*:  
Diese Geräte nutzen eine parametrierbare Frequenzliste. Die Frequenzliste wird bei diesem Gerät über den Parameter CD eingestellt. Bei der Gerätevariante FR1-01 kann der Parameter NC nur ausgelesen werden. Eine Änderung der Frequenzliste durch den Parameter NC ist nicht möglich. Änderungen sind über den Parameter CD vorzunehmen.
- Geräteversion FR2\*:  
Diese Gerätevarianten nutzen ein Frequenzsprungverfahren. Bei diesen Geräten kann durch den Parameter NC die Anzahl der Frequenzkanäle eingestellt werden. Der Parameter NC kann ausgelesen und verändert werden. Es werden immer alle zur Verfügung stehenden Frequenzkanäle verwendet, aber pro Single-Befehl nur so viele, wie durch den Parameter NC vorgegeben.

Parameterzeichen                      NC (16#4E43)

Länge Parameterwert NC              1 Byte

## Werkseinstellung

in Abhängigkeit der Geräteversion

- FR1-01 → 16#04 (4 Kanäle; Kanal 4, 7, 10 und 13)
- FR2-02 → 16#32 (50 Kanäle; Kanal 1, 2, 3, ... 50)
- FR2-03 → 16#10 (16 Kanäle; Kanal 2, 3, 4, ... 17)
- FR1-04 → 16#04 (4 Kanäle; Kanal 1, 4, 7, 10)
- FR2-07 → 16#34 (52 Kanäle; Kanal 1, 2, 3, ... 52)
- FR2-09 → 16#06 (6 Kanäle; Kanal 1, 4, 7, 10, 13, 16)
- FR2-10 → 16#0C (12 Kanäle; Kanal 1, 2, 3, ... 12)
- FR2-13 → 16#08 (8 Kanäle; Kanal 1, 2, 3, ... 8)
- FR2-17 → 16#0E (14 Kanäle; Kanal 1, 2, 3, ... 14)

## Wertebereich

in Abhängigkeit der Geräteversion

- FR1-01 → 16#01 ... 16#04  
Der Wert ist abhängig von der Einstellung des Parameters CD. Bei der Nutzung von einem Frequenzkanal ist der Wert des Parameters NC = 16#01. Wenn alle zulässigen Frequenzkanäle verwendet werden, ist der Wert 16#04.
- FR2-02 → 16#01 ... 16#32  
Die Anzahl der verwendeten Kanäle kann zwischen 1 ... 50 eingestellt werden.
- FR2-03 → 16#01 ... 16#10  
Die Anzahl der verwendeten Kanäle kann zwischen 1 ... 16 eingestellt werden.
- FR1-04 → 16#01 ... 16#0A  
Der Wert ist abhängig von der Einstellung des Parameters CD. Bei der Nutzung von einem Frequenzkanal ist der Wert des Parameters NC = 16#01. Wenn alle zulässigen Frequenzkanäle verwendet werden, ist der Wert 16#0A.
- FR2-07 → 16#01 ... 16#34  
Die Anzahl der verwendeten Kanäle kann zwischen 1 ... 52 eingestellt werden.
- FR2-09 → 16#01 ... 16#06  
Die Anzahl der verwendeten Kanäle kann zwischen 1 ... 6 eingestellt werden.
- FR2-10 → 16#01 ... 16#0C  
Die Anzahl der verwendeten Kanäle kann zwischen 1 ... 12 eingestellt werden.
- FR2-13 → 16#01 ... 16#08  
Die Anzahl der verwendeten Kanäle kann zwischen 1 ... 8 eingestellt werden.
- FR2-17 → 16#01 ... 16#0E  
Die Anzahl der verwendeten Kanäle kann zwischen 1 ... 14 eingestellt werden.

Wenn Sie mit dem Parameter NC mehrere Sendekanäle parametrisiert haben, führt das Gerät jeden Schreib- oder Lesebefehl sequentiell entsprechend einer festgelegten Reihenfolge auf allen festgelegten Sendekanälen durch. Wenn zusätzlich mehrere Sendeleistungswerte parametrisiert sind, werden für jeden Schreib- oder Lesebefehl alle eingestellten Sendeleistungen bei jedem Senderkanal durchgeführt.

Die Anzahl der Sendekanäle hat einen Einfluss auf die Ausführungszeit von Schreib-/Lesebefehlen. Bei der Geräteversion FR2\* kann durch eine Reduzierung der Anzahl der Frequenzkanäle die Ausführungszeit von Schreib-/Lesebefehlen verkürzt werden.



### Beispiel

Die Geräteversion FR2-03 (China) nutzt für die Kommunikation insgesamt 16 Kanäle. Dabei werden die Kanäle 2 ... 17 verwendet. Wenn der Parameter NC auf den Wert 16#06 eingestellt wird, d. h. 6 Kanäle, werden bei der Ausführung von Single-Befehlen folgende Frequenzkanäle verwendet:

- 1. Single-Befehl: Kanal 3, 16, 13, 10, 7 und 4
- 2. Single Befehl: Kanal 17, 14, 11, 8, 5 und 2
- 3. Single Befehl: Kanal 15, 12, 9, 6, 3 und 16
- usw.

### Beispiel: Befehlstelegramm für die Änderung der Einstellung von NC auf den Wert 10 (16#0A)

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	SystemCode	16#55 "U"							
Byte 7	ParameterName (High Byte)	16#4E "N"							
Byte 8	ParameterName (Low Byte)	16#43 "C"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#01							
Byte 11	Parameter NC	16#0A							
Byte 12	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.81

### Beispiel: Befehlstelegramm zum Auslesen der Einstellung von NC

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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#4E "N"							
Byte 8	ParameterName (Low Byte)	16#43 "C"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#00							

2024-08

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.82

**Beispiel: Antworttelegramm mit dem eingestellten Wert des Parameters NC (16#32; 50 Kanäle)**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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 NC	16#32							
Byte 8	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.83

### Q-Wert QW

Die Datenübertragung zwischen einem bzw. mehreren Datenträgern über die Luftschnittstelle wird durch das Slotted-ALOHA-Prinzip realisiert. Die Zeitdauer der Kommunikation zwischen Gerät und Datenträger wird dabei in Zeitschlitze unterteilt. Um Kollisionen durch die gleichzeitige Kommunikation von mehreren Transpondern zu vermeiden, sollte die Anzahl der Zeitschlitze der Anzahl der erwarteten Transponder entsprechen. Durch den Parameter QW wird die Anzahl der Zeitschlitze über die Formel  $2^Q$  festgelegt.

Parameterzeichen	QW (16#5157)
Länge Parameterwert QW	1 Byte
Werkseinstellung	16#02 → es werden $2^2 = 4$ Zeitschlitze verwendet
Wertebereich	16#00 ... 16#04 → es können zwischen $2^0 ... 2^4$ (1 ... 16) Zeitschlitze eingestellt werden.

Die Anzahl der verwendeten Zeitschlitze hat einen Einfluss auf die Ausführungszeit von Schreib-/Lesebefehlen. Durch eine Verringerung der Zeitschlitze verkürzt sich die Ausführungszeit für den Datenträgerzugriff.

**Beispiel: Befehlstelegramm für die Änderung der Einstellung von QW auf den Wert 1 (16#01), d. h. 2 Zeitslitze**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	SystemCode	16#55 "U"							
Byte 7	ParameterName (High Byte)	16#51 "Q"							
Byte 8	ParameterName (Low Byte)	16#57 "W"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#01							
Byte 11	Parameter QW	16#01							
Byte 12	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.84

**Beispiel: Befehlstelegramm zum Auslesen der Einstellung von QW**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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#51 "Q"							
Byte 8	ParameterName (Low Byte)	16#57 "W"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#00							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.85

**Beispiel: Antworttelegramm mit dem eingestellten Wert des Parameters QW (16#02; 4 Zeitschlitze)**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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 QW	16#02							
Byte 8	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.86

**Region Code RC**

Über den Parameter Region Code RC kann die Länderkennung des Geräts ausgelesen werden. Die Länderkennung gibt die länderspezifische Einstellung des Geräts vor.

Die länderspezifischen Einstellungen werden bei der Produktion vorgegeben, der Parameter RC ist nicht veränderbar. Der Parameterwert von RC kann nur ausgelesen werden.

Parameterzeichen	RC (16#5243)
Länge Parameterwert RC	2 Byte
Werkseinstellung	in Abhängigkeit der Geräteversion <ul style="list-style-type: none"> <li>• FR1-01 → 16#0001 (Region Europa und andere Länder, die sich nach EN 302208 richten)</li> <li>• FR2-02 → 16#0002 (USA, Kanada, Mexico, Argentinien)</li> <li>• FR2-03 → 16#0003 (China)</li> <li>• FR1-04 → 16#0004 (Indien)</li> <li>• FR2-07 → 16#0007 (Brasilien)</li> <li>• FR2-09 → 16#0009 (Südkorea)</li> <li>• FR2-10 → 16#000A (Australien)</li> <li>• FR2-13 → 16#000D (Malaysia)</li> <li>• FR2-17 → 16#0011 (Taiwan)</li> </ul>
Wertebereich	Parameterwert ist nicht veränderbar

**Länderkennungen**

Länderkennung	Belegte Frequenzbandbreite Frequenzzugriffsverfahren	Land oder Region
16#01	865 MHz ... 868 MHz parametrierbare Frequenzliste	EU und andere Länder, die sich nach EN 302208 richten
16#02	902 MHz ... 928 MHz Frequenzsprungverfahren	USA, Kanada, Mexico, Argentinien

Länderkennung	Belegte Frequenzbandbreite Frequenzzugriffsverfahren	Land oder Region
16#03	920 MHz ... 925 MHz Frequenzsprungverfahren	China
16#04	865,0 MHz ... 867,0 MHz parametrierbare Frequenzliste	Indien
16#07	915 MHz ... 928 MHz Frequenzsprungverfahren	Brasilien
16#09	917,2 MHz ... 920,4 MHz Frequenzsprungverfahren	Südkorea
16#0A	920 MHz ... 926 MHz Frequenzsprungverfahren	Australien
16#0D	919 MHz ... 923 MHz Frequenzsprungverfahren	Malaysia
16#11	920 MHz ... 928 MHz Frequenzsprungverfahren	Taiwan

Tabelle 8.87

**Beispiel: Befehlstelegramm zum Auslesen des Regionalkodes**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.88

**Beispiel: Antworttelegramm mit dem eingestellten Wert des Regionalkodes 16#0001 (Europa)**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							

2024-08

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#BE							
Byte 6	Status	16#00							
Byte 7	Parameter RC (High Byte)	16#00							
Byte 8	Parameter RC (Low Byte)	16#01							
Byte 9	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.89

### Filter List FL (Filtermaske auslesen)

Der Parameter FL beinhaltet die aktuelle Konfiguration der Filtermasken, wie sie durch die Ausführung des Befehls FI gesetzt wurden. Das Ausgabeformat entspricht dem Eingabeformat des Befehls FI ohne die Filternummer. Der Parameter kann nur ausgelesen, aber nicht gesetzt werden.

Parameterzeichen	FL
Wertebereich	0 ... 2

#### Beispiel: Befehlstelegramm zum Auslesen der Filtermaske 2 (FL = 16#464C)

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Nicht relevant	16#07							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.90

**Beispiel: Antworttelegram nach Auslesen der Filtermaske 2 (Memory Bank = EPC/UII; Negation = 0; Logical Operation = 0 "OR"; Truncation = 0 "Send all")**

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	<FrameLength>			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	<Telegram Length (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<Telegram Length (Low Byte)>							
Byte 5	Command	16#BE							
Byte 6	Status	16#00							
Byte 7	Memory Bank / Negation / Logical Operation / Truncation	2#00001000 <sub>bin</sub>							
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	Reserviert	16#00							
Byte 12	Mask Data	16#E2							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.91

Die Filtereinstellung der Filtermaske 2 hat eine Länge von 1 Byte, einen Inhalt von 16#E2 und beginnt ab der Memory Bank 01 (EPC/UII) ab der Adresse 16#20. Durch diese Maske werden nur Datenträger übertragen, bei dem der EPC/UII mit 16#E2 beginnt.

Inhalt	Bit-Nummer							
	7	6	5	4	3	2	1	0
<Memory Bank / Negation / Logical Operation / Truncation>	0	0	0	<MemBank>		<Negation>	<Logical Operation>	<Truncation>

#### <MemBank>

- 00 not used
- 01 EPC
- 10 TID (Fixcode)
- 11 User

#### <Negation>

- 0 not negated
- 1 negated

Wenn Sie den Filter negieren, werden alle Transponder übertragen, die nicht in den Filter passen.

**<LocalOperation>**

- 0 OR
- 1 AND

Das Bit <Logical Operation> verknüpft verschiedene Filter logisch miteinander. Wenn nur ein Filter verwendet wird, ist dieses Bit nicht relevant.

**<Truncation>**

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

**Bit Ad- dress** Startadresse des Bits, bei dem die Filtermaske beginnt. Definiert als hexadezimaler Wert.

Für die Adresse der Memory Bank 01 (EPC/UII) siehe Kapitel 3.2.3.

## 8.7 Fehler-/Statusmeldungen

Status	Bedeutung
16#00	Der Befehl wurde fehlerfrei ausgeführt. Bei einem Lesebefehl enthält dieses Telegramm den EPC-Code des eingelesenen Datenträgers und eventuell zusätzliche Informationen. Bei einem Schreibbefehl enthält dieses Telegramm den EPC-Code des neu beschriebenen Datenträgers. Mit Empfang dieses Telegramms sind die Daten sicher auf den Datenträger geschrieben.
16#04	Parameterfehler Wenn diese Statusmeldung direkt nach dem Senden des Befehls empfangen wird, ist ein Parameter innerhalb des Befehls außerhalb des Wertebereichs oder der Telegrammaufbau ist nicht korrekt. Diese Statusmeldung wird ebenfalls erzeugt, wenn auf einen physikalisch nicht vorhandenen Speicherabschnitt zugegriffen werden soll. Beispiel: Die einzulesende Datenmenge bei der Ausführung des Befehls Read 4-Byte Blocks ist größer als der vorhandene Speicher.
16#05	Datenträger hat die Erfassungszone verlassen. Bei der Ausführung eines Enhanced Read oder Write Befehls signalisiert diese Statusmeldung, dass ein Datenträger die Erfassungszone verlassen hat. Dieses Telegramm enthält den EPC-Code des Datenträgers. Wenn sich direkt nach dem Start eines Enhanced Befehls kein Datenträger in der Erfassungszone befindet, wird diese Statusmeldung ohne einen EPC-Code gesendet.
16#0A	Mehrere Datenträger mit gleichen EPC-Code innerhalb der Erfassungszone Während der Ausführung eines Schreib- bzw. Lesebefehls dürfen sich nur Transponder mit unterschiedlichen EPC-Code in der Erfassungszone aufhalten. Diese Statusmeldung signalisiert, dass ein identischer EPC-Code von mehr als einem Datenträger erkannt wurde. Dieses Telegramm enthält den EPC-Code des zusätzlichen Datenträgers.
16#0B	Telegramm mit Zusatzinformationen Diese zusätzlichen Informationen (z. B. RSSI-Wert) werden in einem Telegramm mit diesem Status übermittelt.
16#0E	Pufferüberlauf Die Größe des internen Telegrammspeichers wurde überschritten. Das Gerät hat Telegramme schneller erzeugt als an die Steuerung übertragen werden konnte. Der Telegrammspeicher wird durch die Invertierung des Delete-Bits gelöscht. Zusätzlich ist die Funktion des Handshake-Verfahrens zu überprüfen.
16#0F	Endemeldung bei der Ausführung von Single Befehlen Das Ende der Ausführung von Single Read oder Single Write Befehlen wird über diese Statusmeldung signalisiert. Dieses Telegramm enthält die Anzahl der während des Befehls gelesenen bzw. beschriebenen Datenträger.

## 9 **Wartung und Pflege**

Das Gerät ist darauf ausgelegt und konstruiert, seine Funktion über lange Zeiträume stabil zu halten. Daher sind keine regelmäßigen Wartungs- und Reinigungsarbeiten notwendig.

## 10 Störungsbeseitigung

Problem	Lösung
Störungen durch mehrere Geräte in näherer Umgebung	<ul style="list-style-type: none"> <li>• Prüfen Sie den Abstand zu anderen Geräten.</li> <li>• Ändern Sie die Einstellung der Sendekanäle.</li> <li>• Reduzieren Sie die Sendeleistung.</li> </ul>
Status A-Meldung	<ul style="list-style-type: none"> <li>• Prüfen Sie, ob sich mehrere Transponder im Erfassungsbereich befinden: Nehmen Sie den Transponder aus dem Erfassungsbereich, indem Sie den Transponder z. B. in ein geschlossenes Metallgefäß legen. Wiederholen Sie den Schreib- oder Lesevorgang.</li> <li>• Arbeiten Sie mit Filterbefehlen.</li> <li>• Prüfen Sie, ob mehrere Transponder die gleiche UID/EPC haben.</li> </ul>

Tabelle 10.1

# 11 Anhang

## 11.1 Abmessungen

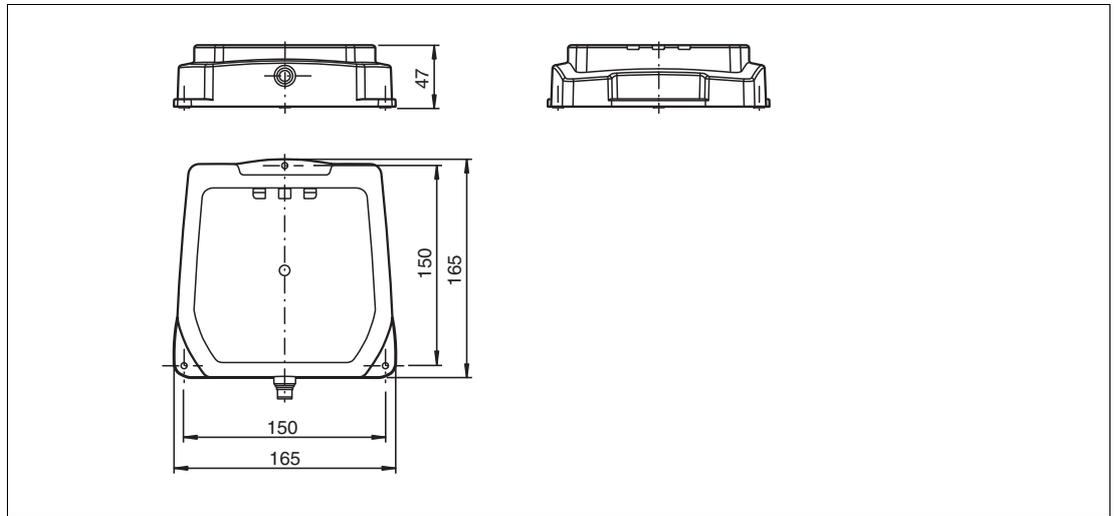


Abbildung 11.1

## 11.2 Beispiele Expert-Mode

### 11.2.1 Single Read EPC/UII

Datentelegramm Single Read Fixcode für einen Transponder mit einem 12 Byte langen EPC (16#30 14 F7 33 7C 00 1F 00 00 00 74 83)

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#17							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#14							
Byte 5	Command	16#D2							
Byte 6	Status	16#00							
Byte 7	Length EPC/UII	16#00							
Byte 8	Length EPC/UII	16#0E							
Byte 9	PC-Word	16#34							
Byte 10	PC-Word	16#00							
Byte 11	EPC Byte 1	16#30							
Byte 12	EPC Byte 2	16#14							
Byte 13	EPC Byte 3	16#F7							
Byte 14	EPC Byte 4	16#33							
Byte 15	EPC Byte 5	16#7C							
Byte 16	EPC Byte 6	16#00							
Byte 17	EPC Byte 7	16#1F							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 18	EPC Byte 8	16#00							
Byte 19	EPC Byte 9	16#00							
Byte 20	EPC Byte 10	16#00							
Byte 21	EPC Byte 11	16#74							
Byte 22	EPC Byte 12	16#83							
Byte 23	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

### 11.2.2 Enhanced Read EPC/UII

Datentelegramm Enhanced Read Fixcode für einen Transponder mit einem 12 Byte langen EPC (16#30 14 F7 33 7C 00 1F 00 00 00 74 83)

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#17							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#14							
Byte 5	Command	16#D3							
Byte 6	Status	16#00							
Byte 7	Length EPC/UII	16#00							
Byte 8	Length EPC/UII	16#0E							
Byte 9	PC-Word	16#34							
Byte 10	PC-Word	16#00							
Byte 11	EPC Byte 1	16#30							
Byte 12	EPC Byte 2	16#14							
Byte 13	EPC Byte 3	16#F7							
Byte 14	EPC Byte 4	16#33							
Byte 15	EPC Byte 5	16#7C							
Byte 16	EPC Byte 6	16#00							
Byte 17	EPC Byte 7	16#1F							
Byte 18	EPC Byte 8	16#00							
Byte 19	EPC Byte 9	16#00							
Byte 20	EPC Byte 10	16#00							
Byte 21	EPC Byte 11	16#74							
Byte 22	EPC Byte 12	16#83							
Byte 23	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

2024-08

Transponder mit 12 Byte langen EPC (16#30 14 F7 33 7C 00 1F 00 00 00 74 83) hat den Erfassungsbereich verlassen

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#17							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#14							
Byte 5	Command	16#D3							
Byte 6	Status	16#05							
Byte 7	Length EPC/UII	16#00							
Byte 8	Length EPC/UII	16#0E							
Byte 9	PC-Word	16#34							
Byte 10	PC-Word	16#00							
Byte 11	EPC Byte 1	16#30							
Byte 12	EPC Byte 2	16#14							
Byte 13	EPC Byte 3	16#F7							
Byte 14	EPC Byte 4	16#33							
Byte 15	EPC Byte 5	16#7C							
Byte 16	EPC Byte 6	16#00							
Byte 17	EPC Byte 7	16#1F							
Byte 18	EPC Byte 8	16#00							
Byte 19	EPC Byte 9	16#00							
Byte 20	EPC Byte 10	16#00							
Byte 21	EPC Byte 11	16#74							
Byte 22	EPC Byte 12	16#83							
Byte 23	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

### 11.2.3 Single Write EPC/UII

Befehlstelegramm für die Programmierung eines 12 Byte langen EPC (16#11 22 33 44 55 66 77 88 99 00 11 22)

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#15							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length	16#00							
Byte 4	Telegram Length	16#12							
Byte 5	Command	16#CE							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 6	Fix Length	16#0E							
Byte 7	PC-Word	16#30							
Byte 8	PC-Word	16#00							
Byte 9	Write EPC Byte 1	16#11							
Byte 10	Write EPC Byte 2	16#22							
Byte 11	Write EPC Byte 3	16#33							
Byte 12	Write EPC Byte 4	16#44							
Byte 13	Write EPC Byte 5	16#55							
Byte 14	Write EPC Byte 6	16#66							
Byte 15	Write EPC Byte 7	16#77							
Byte 16	Write EPC Byte 8	16#88							
Byte 17	Write EPC Byte 9	16#99							
Byte 18	Write EPC Byte 10	16#00							
Byte 19	Write EPC Byte 11	16#11							
Byte 20	Write EPC Byte 12	16#22							
Byte 21	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Befehlstelegramm für die Programmierung eines 4 Byte langen Ull (16#AA BB CC DD)

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	$D_S$	$U_M$	$U_D$	0	16#0			
Byte 1	Frame Length	16#0D							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0A							
Byte 5	Command	16#CE							
Byte 6	Fix Length	16#06							
Byte 7	PC-Word	16#11							
Byte 8	PC-Word	16#00							
Byte 9	Write EPC Byte 1	16#AA							
Byte 10	Write EPC Byte 2	16#BB							
Byte 11	Write EPC Byte 3	16#CC							
Byte 12	Write EPC Byte 4	16#DD							
Byte 13	Nicht relevant	16#00							
Byte 14	Nicht relevant	16#00							
Byte 15	Nicht relevant	16#00							

## 11.2.4 Single Read 4-Byte Blocks

Befehlstelegramm zum Auslesen von 4 Byte ab der Byte-Adresse 4

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#0A							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length	16#00							
Byte 4	Telegram Length	16#07							
Byte 5	Command	16#10							
Byte 6	ByteAddress	16#00							
Byte 7	ByteAddress	16#04							
Byte 8	Number of Bytes	16#00							
Byte 9	Number of Bytes	16#04							
Byte 10	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 15	Nicht relevant	16#00							

Datentelegramm von einem Transponder mit einem 12 Byte langen EPC (16#30 14 F7 33 7C 00 1F 00 00 00 74 83) und den eingelesenen Daten (16#31323334 / ASCII "1234") des User Memory

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#1D							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length	16#00							
Byte 4	Telegram Length	16#1A							
Byte 5	Command	16#10							
Byte 6	Status	16#00							
Byte 7	Length EPC/UII	16#00							
Byte 8	Length EPC/UII	16#0E							
Byte 9	PC-Word	16#34							
Byte 10	PC-Word	16#00							
Byte 11	EPC Byte 1	16#30							
Byte 12	EPC Byte 2	16#14							
Byte 13	EPC Byte 3	16#F7							
Byte 14	EPC Byte 4	16#33							
Byte 15	EPC Byte 5	16#7C							
Byte 16	EPC Byte 6	16#00							
Byte 17	EPC Byte 7	16#1F							
Byte 18	EPC Byte 8	16#00							
Byte 19	EPC Byte 9	16#00							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 20	EPC Byte 10	16#00							
Byte 21	EPC Byte 11	16#74							
Byte 22	EPC Byte 12	16#83							
Byte 23	Length Data	16#00							
Byte 24	Length Data	16#04							
Byte 25	User Data Byte 1	16#31							
Byte 26	User Data Byte 2	16#32							
Byte 27	User Data Byte 3	16#33							
Byte 28	User Data Byte 4	16#34							
Byte 29	Nicht relevant	16#00							
Byte 30	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

### 11.2.5 Enhanced Read 4-Byte Blocks

Befehlstelegramm zum Auslesen von 4 Byte ab der Byte-Adresse 0

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#0A							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length	16#00							
Byte 4	Telegram Length	16#07							
Byte 5	Command	16#19							
Byte 6	ByteAddress	16#00							
Byte 7	ByteAddress	16#00							
Byte 8	Number of Bytes	16#00							
Byte 9	Number of Bytes	16#04							
Byte 10	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 15	Nicht relevant	16#00							

Datentelegramm von einem Transponder mit einem 12 Byte langen EPC (16#30 14 F7 33 7C 00 1F 00 00 00 74 83) und den eingelesenen Daten (16#31323334 / ASCII "1234") des User Memory

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#1D							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length	16#00							

2024-08

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 4	Telegram Length	16#1A							
Byte 5	Command	16#19							
Byte 6	Status	16#00							
Byte 7	Length EPC/UII	16#00							
Byte 8	Length EPC/UII	16#0E							
Byte 9	PC-Word	16#34							
Byte 10	PC-Word	16#00							
Byte 11	EPC Byte 1	16#30							
Byte 12	EPC Byte 2	16#14							
Byte 13	EPC Byte 3	16#F7							
Byte 14	EPC Byte 4	16#33							
Byte 15	EPC Byte 5	16#7C							
Byte 16	EPC Byte 6	16#00							
Byte 17	EPC Byte 7	16#1F							
Byte 18	EPC Byte 8	16#00							
Byte 19	EPC Byte 9	16#00							
Byte 20	EPC Byte 10	16#00							
Byte 21	EPC Byte 11	16#74							
Byte 22	EPC Byte 12	16#83							
Byte 23	Length Data	16#00							
Byte 24	Length Data	16#04							
Byte 25	User Data Byte 1	16#31							
Byte 26	User Data Byte 2	16#32							
Byte 27	User Data Byte 3	16#33							
Byte 28	User Data Byte 4	16#34							
Byte 29	Nicht relevant	16#00							
Byte 30	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Transponder mit 12 Byte langen EPC (16#30 14 F7 33 7C 00 1F 00 00 00 74 83) hat den Erfassungsbereich verlassen

Byte	Inhalt	Bit-Nummer							
		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#19							
Byte 6	Status	16#05							
Byte 7	Length EPC/UII	16#00							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 8	Length EPC/UII	16#0E							
Byte 9	PC-Word	16#34							
Byte 10	PC-Word	16#00							
Byte 11	EPC Byte 1	16#30							
Byte 12	EPC Byte 2	16#14							
Byte 13	EPC Byte 3	16#F7							
Byte 14	EPC Byte 4	16#33							
Byte 15	EPC Byte 5	16#7C							
Byte 16	EPC Byte 6	16#00							
Byte 17	EPC Byte 7	16#1F							
Byte 18	EPC Byte 8	16#00							
Byte 19	EPC Byte 9	16#00							
Byte 20	EPC Byte 10	16#00							
Byte 21	EPC Byte 11	16#74							
Byte 22	EPC Byte 12	16#83							
Byte 23	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

### 11.2.6 Single Write 4-Byte Blocks

Befehlstelegramm Single Write 4-Byte Blocks zum Schreiben von 4 Byte User Data (16#01020304) ab der Byte-Adresse 8

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	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	Nicht relevant	16#00							
Byte 15	Nicht relevant	16#00							

2024-08

Datentelegramm von einem Transponder mit einem 12 Byte langen EPC (16#30 14 F7 33 7C 00 1F 00 00 00 74 83) mit der Bestätigung eines erfolgreichen Schreibvorgangs

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#1D							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length	16#00							
Byte 4	Telegram Length	16#14							
Byte 5	Command	16#40							
Byte 6	Status	16#00							
Byte 7	Length EPC/UII	16#00							
Byte 8	Length EPC/UII	16#0E							
Byte 9	PC-Word	16#34							
Byte 10	PC-Word	16#00							
Byte 11	EPC Byte 1	16#30							
Byte 12	EPC Byte 2	16#14							
Byte 13	EPC Byte 3	16#F7							
Byte 14	EPC Byte 4	16#33							
Byte 15	EPC Byte 5	16#7C							
Byte 16	EPC Byte 6	16#00							
Byte 17	EPC Byte 7	16#1F							
Byte 18	EPC Byte 8	16#00							
Byte 19	EPC Byte 9	16#00							
Byte 20	EPC Byte 10	16#00							
Byte 21	EPC Byte 11	16#74							
Byte 22	EPC Byte 12	16#83							
Byte 23	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

### 11.2.7 Enhanced Write 4-Byte Blocks

Befehlstelegramm Enhanced Write 4-Byte Blocks zum Schreiben von 16 Byte User Data (16#00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF) ab der Byteadresse 0.

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#1A							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length	16#00							
Byte 4	Telegram Length	16#17							
Byte 5	Command	16#1A							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 6	ByteAddress	16#00							
Byte 7	ByteAddress	16#00							
Byte 8	Number of Bytes	16#00							
Byte 9	Number of Bytes	16#10							
Byte 10	Write Data Byte 1	16#00							
Byte 11	Write Data Byte 2	16#11							
Byte 12	Write Data Byte 3	16#22							
Byte 13	Write Data Byte 4	16#33							
Byte 14	Write Data Byte 5	16#44							
Byte 15	Write Data Byte 6	16#55							
Byte 16	Write Data Byte 7	16#66							
Byte 17	Write Data Byte 8	16#77							
Byte 18	Write Data Byte 9	16#88							
Byte 19	Write Data Byte 10	16#99							
Byte 20	Write Data Byte 11	16#AA							
Byte 21	Write Data Byte 12	16#BB							
Byte 22	Write Data Byte 13	16#CC							
Byte 23	Write Data Byte 14	16#DD							
Byte 24	Write Data Byte 15	16#EE							
Byte 25	Write Data Byte 16	16#FF							
Byte 26	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Datentelegramm von einem Transponder mit einem 12 Byte langen EPC (16#30 14 F7 33 7C 00 1F 00 00 00 74 83) mit der Bestätigung eines erfolgreichen Schreibvorgangs

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D <sub>S</sub>	U <sub>M</sub>	U <sub>D</sub>	0	16#0			
Byte 1	Frame Length	16#17							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length	16#00							
Byte 4	Telegram Length	16#14							
Byte 5	Command	16#1A							
Byte 6	Status	16#05							
Byte 7	Length EPC/UII	16#00							
Byte 8	Length EPC/UII	16#0E							
Byte 9	PC-Word	16#34							
Byte 10	PC-Word	16#00							
Byte 11	EPC Byte 1	16#30							

2024-08

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 12	EPC Byte 2	16#14							
Byte 13	EPC Byte 3	16#F7							
Byte 14	EPC Byte 4	16#33							
Byte 15	EPC Byte 5	16#7C							
Byte 16	EPC Byte 6	16#00							
Byte 17	EPC Byte 7	16#1F							
Byte 18	EPC Byte 8	16#00							
Byte 19	EPC Byte 9	16#00							
Byte 20	EPC Byte 10	16#00							
Byte 21	EPC Byte 11	16#74							
Byte 22	EPC Byte 12	16#83							
Byte 23	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

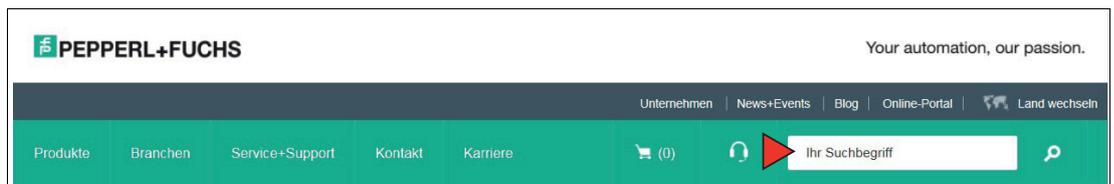
## 11.3 ASCII-Tabelle

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

## 11.4 Erfassungsbereich

Das Gerät hat einen typischen Erfassungsbereich von etwa 1 m, der vom verwendeten Transponder mitbestimmt wird und durch die Wahl der Sendeleistung angepasst werden kann. Weitere Einflussfaktoren sind der Aufbau bzw. Einbau für die spezifische Anwendung, die Beeinflussung durch vorhandene Materialien (insbesondere Metall) sowie die Umgebungsbedingungen. Die separat angegebenen Schreib- und Leseabstände für die jeweiligen Transponder sind in einem Testlabor unter idealen Bedingungen ermittelt worden. Testen Sie die Kombination aus Gerät und Transponder für die gewünschte Anwendung daher unter Realbedingungen.

Beachten Sie dazu bitte die Abstandstabellen. Die Abstandstabellen und weitere Informationen zu Ihrem Produkt finden Sie auf <http://www.pepperl-fuchs.com>. Geben Sie dazu die Produktbezeichnung oder Artikelnummer in das **Suche**-Feld ein und klicken Sie auf die Taste **Suche**.



Wählen Sie aus der Liste der Suchergebnisse Ihr Produkt aus. Klicken Sie in der Liste der Produktinformationen auf Ihre benötigte Information, z. B. **Technische Dokumente**.



Hier finden Sie in einer Listendarstellung alle verfügbaren Dokumente.

# Your automation, our passion.

## Explosionsschutz

- Eigensichere Barrieren
- Signaltrenner
- Feldbusinfrastruktur FieldConnex®
- Remote-I/O-Systeme
- Elektrisches Ex-Equipment
- Überdruckkapselungssysteme
- Bedien- und Beobachtungssysteme
- Mobile Computing und Kommunikation
- HART Interface Solutions
- Überspannungsschutz
- Wireless Solutions
- Füllstandsmesstechnik

## Industrielle Sensoren

- Näherungsschalter
- Optoelektronische Sensoren
- Bildverarbeitung
- Ultraschallsensoren
- Drehgeber
- Positioniersysteme
- Neigungs- und Beschleunigungssensoren
- Feldbusmodule
- AS-Interface
- Identifikationssysteme
- Anzeigen und Signalverarbeitung
- Connectivity

### Pepperl+Fuchs Qualität

Informieren Sie sich über unsere Qualitätspolitik:

[www.pepperl-fuchs.com/qualitaet](http://www.pepperl-fuchs.com/qualitaet)

