

IQT3-FP-IO-V1

**IO-Link RFID-Schreib-
/Lesegerät 13,56 MHz, ISO
15693**

Handbuch



 **IO-Link**

Your automation, our passion.

 **PEPPERL+FUCHS**

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

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

1	Einleitung	5
1.1	Inhalt des Dokuments	5
1.2	Zielgruppe, Personal	5
1.3	Bestimmungsgemäße Verwendung	5
1.4	Verwendete Symbole.....	6
1.5	Festlegungen.....	6
2	Zertifikate und Zulassungen.....	9
2.1	Konformitätserklärung (RE Directive 2014/53/EU).....	9
2.2	FCC-Information.....	9
2.3	IC-Information	10
2.4	UL Information	10
2.5	Weitere länderspezifische Zulassungen	10
3	Produktbeschreibung	11
3.1	Produktbeschreibung.....	11
3.2	Abmessungen	11
3.3	Anzeigen	12
3.4	Elektrischer Anschluss	12
3.5	IO-Link-Schnittstelleneigenschaften	12
3.6	Zubehör	13
3.6.1	IO-Link-Master.....	13
3.6.2	Datenträger.....	13
3.6.3	Anschlusskabel	13
4	Installation.....	14
4.1	Lagerung und Transport.....	14
4.2	Auspacken.....	14
4.3	Montage und Anschluss	14
4.4	Mindestabstände	15
5	Inbetriebnahme.....	16
5.1	Betriebsarten	16
6	Bedienung	17
6.1	Datenträger 13,56 MHz ISO15693	17
6.2	Erfassungsbereich	18

6.3	Mehrere Transponder im Erfassungsbereich.....	18
7	EasyMode	19
7.1	Befehlsübersicht	19
7.2	Grundlegende Struktur der Prozessdaten	19
7.2.1	Ausgangsprozessdaten (SPS -> Gerät)	20
7.2.2	Eingangsprozessdaten (Gerät -> SPS)	21
7.2.3	Ablaufdiagramme.....	25
7.2.4	Timing	29
7.2.5	Unterbrechung der IO-Link-Kommunikation.....	31
7.3	EasyMode mit PACTware.....	31
8	ExpertMode	39
8.1	Grundlegender Befehlsablauf	39
8.2	Legende	39
8.3	Aufbau OUTPUT-Telegramm.....	40
8.4	Aufbau INPUT-Telegramm.....	41
8.5	Handshake-Verfahren	42
8.6	Befehlsübersicht	43
8.6.1	Schreib-/Lesebefehle	44
8.6.2	Systembefehle	71
8.6.3	HF-Konfigurationbefehle	74
8.6.3.1	Grundlegende Befehlsstruktur	74
8.6.3.2	Parameterübersicht	78
8.7	Fehler-/Statusmeldungen.....	100
9	Anhang.....	101
9.1	Fehlerbehebung	101
9.2	ASCII-Tabelle.....	102

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.

Die Dokumentation besteht aus folgenden Teilen:

- vorliegendes Dokument
- Datenblatt
- IO-Link-Parameterdatenblatt

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_{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)

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 eines Schreib-/Lesegeräts; kein Funktionsbaustein erforderlich
Expert Mode	Kommunikationsprotokoll für einen erweiterten Datenzugriff des RFID-Schreib-/Lesegeräts; die Verwendung eines Funktionsbausteins ist erforderlich
Fixcode	eindeutige und nicht veränderbare Nummer eines Datenträgers, Unique Identifier UID
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
IQC	Pepperl+Fuchs-spezifische Bezeichnung eines 13,56 MHz-Datenträgers
ISO/IEC 15693	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
RSSI	Indikator für die Empfangsfeldstärke
SIO-Mode	Standard IO-Mode; Modus zur konventionellen Signalübertragung ohne IO-Link-Daten ¹
SPS	Speicherprogrammierbare Steuerung; Gerät zur Steuerung und Regelung einer Maschine bzw. Anlage
UID	siehe "Fixcode"
Vendor ID	Identifikationsnummer des Geräteherstellers; Pepperl+Fuchs: 0x01

¹. nicht unterstützt

Abkürzungen

FCC	Federal Communications Commission
HF	Hoch-Frequenz
IC	Industry Canada
ISO	International Standardisation Organisation
RFID	Radio Frequenz Identifikation
RSSI	Received Signal Strength Indicator
SPS	Speicherprogrammierbare Steuerung
Tag	Datenträger; Transponder
UID	Unique Identifier

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 heruntergeladen werden.

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



2.2 FCC-Information

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

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

Attention:

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

**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 20 cm from all persons and must not be co-located or operated in conjunction with any other antenna or transmitter.

2.3 IC-Information

This device complies with Industry Canada 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 20 cm from all persons and must not be co-located or operated in conjunction with any other antenna or transmitter.

2.4 UL Information

Technical Data and Environmental Conditions

This device is for indoor use only.

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

The ambient temperature range is from -20 °C ... +70 °C for operation with non-transmission periods, or -25 °C ... +55 °C for continuous transmission mode. The Pollution degree is 2.

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

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

The enclosure type is 1.

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

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

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

2.5 Weitere länderspezifische Zulassungen

Alle derzeit gültigen Zulassungen finden Sie auf dem Datenblatt Ihres Geräts unter www.pepperl-fuchs.com.

3.3 Anzeigen

Das RFID-Schreib-/Lesegerät besitzt LEDs zur Anzeige des Betriebszustands. Die verschiedenen Anzeigen bedeuten:

LED	Beschreibung
grün	an: Gerät ist betriebsbereit, keine IO-Link-Kommunikation blinkend: IO-Link-Kommunikation aktiv
blau	an: Schreib-/Leseversuch wird ausgeführt
gelb	an: Schreib-/Leseversuch war erfolgreich
rot	an: Störung durch externe Signale im Erfassungsbereich, Reichweite kann eingeschränkt sein. langsam blinkend (1 Hz): Störung durch Metall schnell blinkend (10 Hz): Übertemperatur

3.4 Elektrischer Anschluss

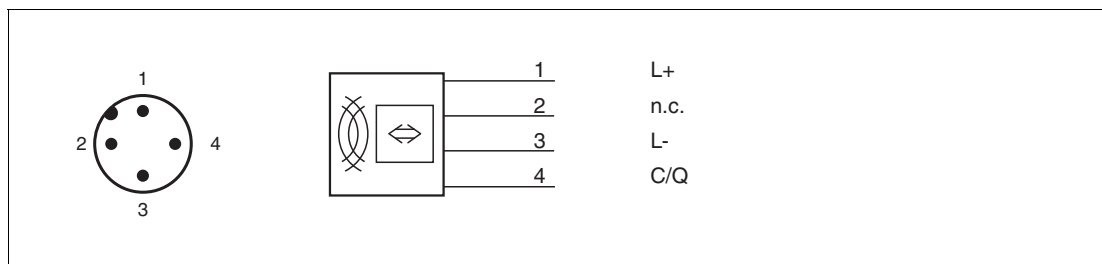


Vorsicht!

Kabelspezifikationen

Die zulässige Höchsttemperatur des Anschlusskabels muss mindestens +70 °C betragen.

Der Mindestdurchmesser des Anschlusskabels muss 22 AWG oder 0,34 mm² 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.

3.5 IO-Link-Schnittstelleneigenschaften

IO-Link-Version:	1.1
Datenübertragungsrate	COM3 (230,4 kbits/s)
Min. Zykluszeit	4 ms
Prozessdaten	Eingang 32 Byte Ausgang 32 Byte
SIO Mode-Unterstützung	nein
Kompatibler Masterport-Typ	Class A Class B
Geräte-ID	4195073 (0x400301)
Hersteller-ID	1 (0x0001)

2023-11

3.6 Zubehör

3.6.1 IO-Link-Master

Sie können das RFID-Schreib-/Lesegerät an jeden IO-Link-Master anschließen, sofern dieser den IO-Link Standard V1.1 unterstützt und über eine ausreichende Stromversorgung verfügt. Unter anderem sind folgende IO-Link-Master von Pepperl+Fuchs erhältlich:

Bestellbezeichnung	Beschreibung
IO-Link-Master02-USB	IO-Link-Master mit USB-Schnittstelle zum Anschluss an übergeordnete Geräte (z.B. PC)
ICE1-8IOL-G60L-V1D	Ethernet-IO-Link-Modul mit 8 IO-Link-Ports zum Anschluss an ein übergeordnetes System über Profinet und Ethernet/IP
ICE1-8IOL-G30L-V1D	Ethernet-IO-Link-Modul mit 8 IO-Link-Ports zum Anschluss an ein übergeordnetes System über Profinet und Ethernet/IP
ICE1-8IOL-S2-G60L-V1D	Ethernet/IP-IO-Link-Master mit PROFINET-S2-Redundanz
ICE2-8IOL-G65L-V1D	Ethernet/IP-IO-Link-Master mit 8 Ein-/Ausgängen
ICE3-8IOL-G65L-V1D	PROFINET/IO-IO-Link-Master mit 8 Ein-/Ausgängen

Tabelle 3.1



Hinweis!

Überprüfen Sie die Leistungsfähigkeit der Stromversorgung am Anschluss des Masters und den Kabelquerschnitt des Verbindungskabels, um eine stabile Kommunikation zu gewährleisten.

3.6.2 Datenträger

Das RFID-Schreib-/Lesegerät kann auf jeden Datenträger zugreifen, der den Standard ISO15693 erfüllt. Für eine Übersicht über mögliche Datenträger . Es können beispielsweise folgende Datenträger von Pepperl+Fuchs verwendet werden:

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

3.6.3 Anschlusskabel

Zum Anschluss des RFID-Schreib-/Lesegeräts 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>.



Hinweis!

Geben Sie in der Produktsuche die Bestellbezeichnung Ihres RFID-Schreib-/Lesegeräts ein. Auf der Produktdetailseite finden Sie eine Liste der zugehörigen Produkte.

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 und Anschluss



Hinweis!

Das RFID-Schreib-/Lesegerät ist für die Wandmontage und die Montage an Halterungen im Innenbereich vorgesehen.

Montieren Sie das Gerät auf einer ebenen Fläche und befestigen Sie es nur mit den am Gehäuse vorhandenen Löchern.



Warnung!

Unkontrolliert angesteuerte Prozesse gefährden die Anlage

Stellen Sie vor der Inbetriebnahme sicher, dass keine Gefahr für die Anlage entstehen kann, in die das Gerät eingebunden ist.



Vorsicht!

Heiße Oberflächen

Verbrennungsgefahr beim Hantieren mit dem Schreib-/Lesegerät! Lassen Sie das Gerät nach dem Abschalten für mindestens eine halbe Stunde abkühlen, bevor Sie es berühren.

Verwenden Sie zum Befestigen des Schreib-/Lesegeräts 4 Schrauben mit 4 mm Durchmesser, passende Unterlegscheiben und Befestigungsmaterial, dass 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 max. Anzugsdrehmoment von 2,4 Nm, um Beschädigungen am Kunststoffgehäuse zu vermeiden.



Hinweis!

Verlegen Sie das Anschlusskabel des Schreib-/Lesegeräts und des IO-Link Masters nicht in den Laufbahnbereich der Datenträger. Halten Sie mindestens einen Abstand von 10 cm ein.

Bei zu geringem Abstand kann es in seltenen Fällen zu einer fehlerhaften Erfassung von Datenträger durch induktive Kopplungen am Anschlusskabel kommen.

4.4 Mindestabstände

Beim gleichzeitigen Betrieb mehrerer Schreib-/Lesegeräte darf zu jedem Zeitpunkt jeweils nur ein Gerät 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 Geräts am Montageort.

Halten Sie bei der Montage mehrerer Schreib-/Lesegeräte einen **Mindestabstand von 750 mm** ein.

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 Datenträger 13,56 MHz ISO15693

Die Datenträger eines RFID-Systems mit 13,56 MHz bieten eine deutlich höhere Zugriffsgeschwindigkeit auf die Daten als ein vergleichbares RFID-System auf Basis einer Arbeitsfrequenz von 125 kHz. Das 13,56-MHz-System ist durch die ISO15693 standardisiert. Es wird eine Vielzahl an Datenträgern verschiedener Hersteller unterstützt, die jeweils unterschiedliche RFID-Chips verwenden.

Zur Einstellung des RFID-Schreib-/Lesegeräts auf den verwendeten Datenträger wird die Parametrierung des zugehörigen Datenträgertyps (TagType) empfohlen. Im Auslieferungszustand des RFID-Schreib-/Lesegeräts ist der Datenträgertyp 20 voreingestellt. Mit dieser Einstellung ist der Zugriff auf den Fixcode ISO15693-konformer Datenträger gewährleistet.

Die nachfolgende Tabelle zeigt die für das RFID-Schreib-/Lesegerät spezifizierten und empfohlenen Datenträgertypen.

Transpondertypen 13,56 MHz / ISO 15693

Bezeichnung Pepperl+Fuchs	Chiptyp	Hersteller	Länge Fixcode [Byte]	Größe Schreib-/Lesespeicher [Byte]	Größe Speicherblock [Byte]
IQC20	alle Datenträger gem. ISO15693	-	8	abhängig vom Datenträger	abhängig vom Datenträger
IQC21	I-Code SLI(X)	NXP	8	112	4
IQC22	Tag-it HF-I Plus	Texas Instruments	8	256	4
IQC23	my-D SRF55V02P	Infineon	8	224	4
IQC24	my-D SRF55V10P	Infineon	8	992	4
IQC27 ¹	EM4135	EM Micro-electronic	8	288	8
IQC31	Tag-it HFI standard	Texas Instruments	8	32	4
IQC32	Tag-it HFI pro	Texas Instruments	8	32	4
IQC33 ¹	FRAM MB89R118	Fujitsu	8	2000	8
IQC34	FRAM MB89R119	Fujitsu	8	232	4
IQC35	I-Code SLI-S	NXP	8	160	4
IQC36	I-Code SLI-L	NXP	8	32	4
IQC37 ²	FRAM MB89R112	Fujitsu	8	8192	32
IQC38	EM4233	EM Micro-electronic	8	208	4
IQC50	I-Code SLIX2	NXP	8	316	4

Tabelle 6.1 Transpondertypen 13,56 MHz gem. ISO 15693

1. Ausnahme: Blockgröße = 8 Byte
'Anzahl Bytes' muss Vielfaches von 8 sein

2. Ausnahme: Blockgröße = 32 Byte
'Anzahl Bytes' muss Vielfaches von 32 sein

Alle ISO15693-konformen Datenträger verfügen über einen 8 Byte langen und eindeutigen Fixcode. Der Fixcode wird vom Chip-Herstellers festgelegt. Er kann durch den Anwender nur gelesen, aber nicht geändert werden. Zusätzlich besitzen die Datenträger einen Speicherbereich für Anwenderdaten. Dieser kann mit anwendungsspezifischen Daten beschrieben sowie ausgelesen werden. In Abhängigkeit des Datenträgertyps hat der Speicher für die Anwenderdaten eine unterschiedliche Größe.

Der Speicherbereich unterteilt sich dabei in Blöcke mit einer Länge von 4 Byte. Es gibt aber auch Ausnahmen mit einer Blocklänge von 8 Byte bzw. 32 Byte. Aufgrund der Blocklänge von 32 Byte unterstützt der Easy Mode den Datenträgertyp IQC37 nicht.

Die Lese- bzw. Schreibbefehle verwenden die Parameter "Anzahl Bytes" und "Anfangsadresse". Dadurch wird definiert, ab welcher Speicheradresse auf wie viele Bytes auf den Speicherbereich der Anwenderdaten zugegriffen wird. Hat der verwendete Datenträgertyp z.B. eine Blocklänge von 4 Byte, so müssen die Werte der Parameter "Anzahl Bytes" und "Anfangsadresse" ein Vielfaches von 4 sein. Bei einer Blocklänge von 8 bzw. 32 Byte sind es Vielfache von 8 bzw. 32.

6.2 Erfassungsbereich

Das Schreib-/Lesegerät ist mit einer Erfassungsreichweite von bis zu 30 cm klassifiziert. Die Reichweite des Geräts ist abhängig von den zu identifizierenden Transpondern und kann dadurch variieren. Sie können den Erfassungsbereich durch entsprechende Änderung der Sendeleistung vergrößern oder verkleinern.

Hinweis!



Um den Grenzwert der magnetischen Feldstärke von max. 10 A/m gemäß ISO 15693-1 einzuhalten, müssen Sie bei einer Sendeleistung $PT > 1$ ein Mindestabstand von 50 mm zwischen Transponder und Schreib-/Lesegerät einhalten. Es besteht sonst die Gefahr, dass ein Transponder im Nahbereich beschädigt wird. Reduzieren Sie die Sendeleistung des Schreib-/Lesegeräts auf $PT = 1$, um den Grenzwert bei kleinen Mindestabständen einzuhalten.

6.3 Mehrere Transponder im Erfassungsbereich

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

EasyMode

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

ExpertMode

Jeder Lese- und Schreibbefehl kann auf einen, mehrere oder alle im Erfassungsbereich befindlichen Transponder zugreifen. Zur Steuerung werden Filtermasken verwendet, die mit dem Parameter **TI** verwaltet werden. Mit diesem Befehl können Sie gezielt bestimmte Transponder im Erfassungsbereich ansprechen. Siehe "Setzen eines Filters "Tag ID Filter (TI)"" auf Seite 95.

Hinweis!



Verwenden Sie bei Multi-Tag-Anwendungen gleiche Transpondertypen.

Für den Multi-Tag-Betrieb muss der IO-Link Parameter "Input Representation" auf "Long-Form" gestellt werden.

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:

Befehl	Beschreibung
Lesen des Fixcode (UID)	Auslesen des 8 Byte langen Fixcode; kein Einstellen der Parameter "Anzahl Bytes" bzw. "Anfangsadresse" erforderlich
Lesen des User Memory	Auslesen von Anwenderdaten; das Einstellen der Parameter "Anzahl Bytes" sowie "Anfangsadresse" ist erforderlich
Schreiben des User Memory	Schreiben von Anwenderdaten; das Einstellen der Parameter "Anzahl Bytes" sowie "Anfangsadresse" ist erforderlich
Autostart	Automatische Ausführung eines Leseauftrages für Fixcode oder Anwenderdaten nach Gerätestart

7.2 Grundlegende Struktur der Prozessdaten

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.

Eine detaillierte Beschreibung der Prozessdaten finden Sie in der IODD oder auf dem IO-Link-Parameterdatenblatt des Geräts. Sie finden die IODD und das Parameterdatenblatt auf der Produktdetailseite unter www.pepperl-fuchs.com.

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



Hinweis!

Ausgangsprozessdaten werden vom Gerät ignoriert, wenn das "valid"-Flag nicht gesetzt ist. Ein nicht gesetztes "valid"-Flag markiert ungültige Prozessdaten.

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 7.1

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.

Start Lesen: Sobald dieses Bit gesetzt wird (TRUE), wird ein Leseauftrag gestartet, entsprechend der Konfiguration durch den Parameter "Read Task" in der IODD-Datei. Der Leseauftrag wird dabei kontinuierlich ausgeführt. Um den Leseauftrag abzubrechen, setzen Sie das Bit zurück (FALSE).

Start Schreiben: Sobald dieses Bit gesetzt wird (TRUE), wird ein Schreibauftrag gestartet. Der Schreibauftrag überträgt die Anwenderdaten, die ab Byte 4 hinterlegt sein müssen, entsprechend der Konfiguration durch den Parameter "Write Task" in der IODD-Datei. Der Schreibauftrag wird dabei kontinuierlich ausgeführt. Um den Schreibauftrag abzubrechen, 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

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

Ausgangsprozessdaten

Byte 0: 0000 0010_{bin}; Das Bit "Start Schreiben" ist auf 1 gesetzt; es wird ein Schreibauftrag ausgeführt.

Byte 1/2/3: 16#00; Nicht verwendet

Byte 4 ... 11: Schreibdaten

7.2.2 Eingangsprozessdaten (Gerät -> SPS)

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

Tabelle 7.2

Byte 0

Dieses Byte beinhaltet die Steuerbits zur Signalisierung des Zustands der Ausführung des Lese- bzw. Schreibauftrags.

Lesen erfolgreich: 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 Flankenwechsel¹ die Übertragung eines weiteren Datenträgers signalisiert.

Schreiben erfolgreich: 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.

Auftrag aktiv: Während der Lese- bzw. Schreibauftrag ausgeführt wird, ist dieses Bit gesetzt. Sobald der Auftrag beendet ist, wird dieses Bit wieder zurückgesetzt.

Fehler: 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.

Transponder vorhanden: Bit ist gesetzt, wenn sich ein oder mehrere Datenträger im Erfassungsbereich befinden. Befindet sich kein Datenträger im Erfassungsbereich, hat dieses Bit den Wert FALSE.

Byte 1

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.

- Byte 2** Der RSSI-Wert zeigt die Signalstärke der Transponder-Antwort prozentual an. Der Wertebereich liegt zwischen 0 und 100. 0 % = schwaches Signal, 100 % = starkes Signal.
- Byte 3** Reserviert.
- Byte 4/5** Wenn erfolgreich auf einen Datenträger zugegriffen wird, befindet sich an dieser Position die Länge des Fixcodes UID.²
- Byte 6 ... Byte 31** Beginnend ab Byte 6 wird der UID 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 UID enthalten. Hierdurch ist eine eindeutige Zuordnung des Datenträgers gewährleistet. Bei einem Leseauftrag des User Memory schließt sich die Längenangabe (2 Byte) des eingelesenen User Memory und der eingelesene Teilbereich des 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 die UID bzw. den 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 Fixcode (UID), Long-Form-Datenformat

Name	Address	Displ...	Monitor value
ControlByte_In	%B0	Bin	2#0001_0101
Length	%B1	DEC	10
RSSI	%B2	DEC	97
Reserved	%B3	DEC	0
Length_UID_HighByte	%B4	Hex	16#00
Length_UID_LowByte	%B5	Hex	16#08
UID_1	%B6	Hex	16#E0
UID_2	%B7	Hex	16#04
UID_3	%B8	Hex	16#01
UID_4	%B9	Hex	16#50
UID_5	%B10	Hex	16#D3
UID_6	%B11	Hex	16#23
UID_7	%B12	Hex	16#74
UID_8	%B13	Hex	16#BA

Byte 0: 0001 0101_{bin}; 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: 10; Das Byte zeigt die Länge der übertragenen Informationen an. Es wird durch dieses Telegramm eine 10 Byte lange Information übertragen. Die Information beginnt ab Byte 4.

Byte 2: 97; RSSI-Wert

Byte 3: 0; reserviert

Byte 4/5: 16#0008; Längenangabe der UID. Die Information hat eine Länge von 2 Byte. Die Länge der UID ist immer 8 Byte.

Byte 8 ...
Byte 13: UID des Datenträgers, der ausgelesen wurde

Beispiel: Lesen Fixcode (UID), Short-Form-Datenformat

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

Byte 0: 0001 0101_{bin}; 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: 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: 98; RSSI-Wert

Byte 3: 16#00; reserviert

Byte 4 ...
Byte 11: UID des Datenträgers, der ausgelesen wurde.

Beispiel: Lesen User Memory, Long-Form-Datenformat

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

Byte 0: 0001 0101_{bin}; 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: 20; Das Byte zeigt die Länge der übertragenen Informationen an. Es wird durch dieses Telegramm eine 20 Byte lange Information übertragen. Die Information beginnt ab Byte 4.

Byte 2: 97; RSSI-Wert

Byte 3: 0; reserviert

Byte 4/5: 16#0008; Längenangabe der UID. Die Information hat eine Länge von 2 Byte. Die Länge der UID ist immer 8 Byte.

Byte 6 ...
Byte 13: UID des identifizierten Datenträgers

Byte 14/15: 16#0008; Längenangabe des eingelesenen User Memory

Byte 16 ..
Byte 23: eingelesener Teilbereich des User Memory

Beispiel: Lesen User Memory, Short-Form-Datenformat

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

Byte 0: 0001 0101_{bin}; 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: 97; RSSI-Wert

Byte 3: 16#00; reserviert

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

Beispiel: Schreiben User Memory, Long-Form-Datenformat

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

Byte 0: 0001 0110_{bin}; Das Bit "Auftrag Aktiv" ist auf 1 gesetzt und signalisiert, dass einen aktivierten Schreibauftrag. Das Bit "Schreiben erfolgreich" auf 1 gesetzt. Es befindet sich 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: 10; Das Byte zeigt die Länge der übertragenen Informationen an. Es wird durch dieses Telegramm eine 10 Byte lange Information übertragen. Die Information beginnt ab Byte 4.

Byte 2: 98; RSSI-Wert

Byte 3: 0; reserviert

Byte 4/5: 16#0008; Längenangabe der UID. Die Information hat eine Länge von 2 Byte. Die Länge der UID ist immer 8 Byte.

Byte 6 ...
Byte 13: UID des Datenträgers, auf den die Daten geschrieben wurden.

Beispiel: Schreiben User Memory, Short-Form-Datenformat

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

Byte 0: 0001 0110_{bin}; Das Bit "Auftrag Aktiv" ist auf 1 gesetzt und signalisiert einen aktiven Schreibauftrag. Das Bit "Schreiben erfolgreich" ist auf 1 gesetzt. Es befindet sich ein Datenträger im Erfassungsbereich und die Daten wurden erfolgreich geschrieben. 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: 98; RSSI-Wert

Byte 3: 0; reserviert

Byte 4 ... Byte 11: UID des Datenträgers, auf den die Daten geschrieben wurden.

7.2.3 Ablaufdiagramme

Leseauftrag ohne Autostart-Funktion

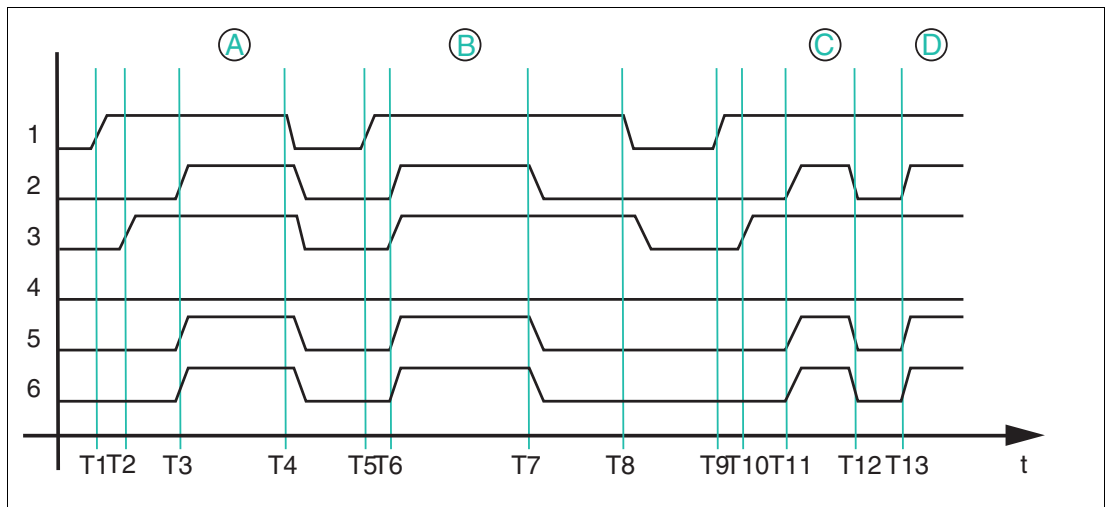


Abbildung 7.1 Zeitlicher Ablauf der Bits im Byte 0

- 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

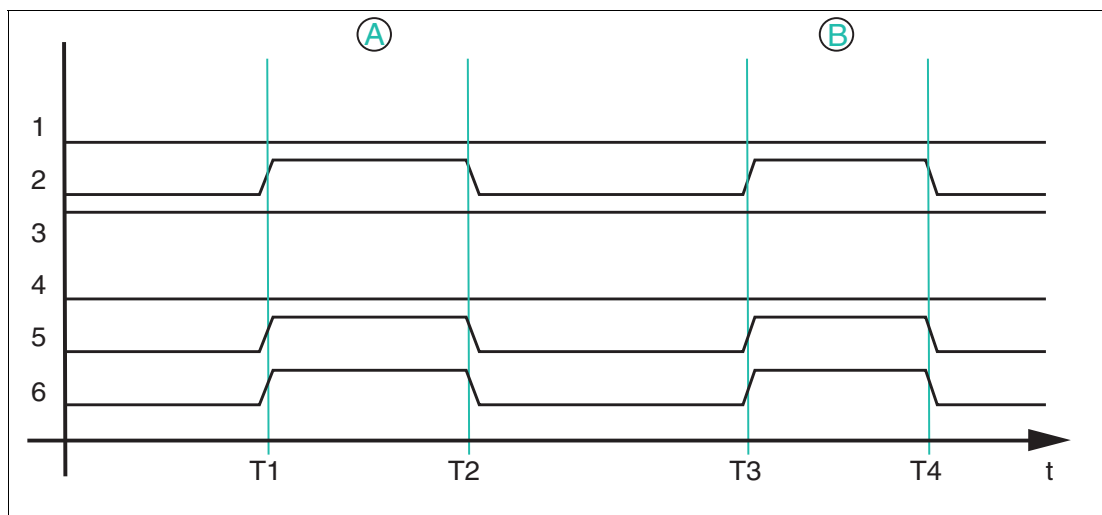


Abbildung 7.2 Zeitlicher Ablauf der Bits im Byte 0

- 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

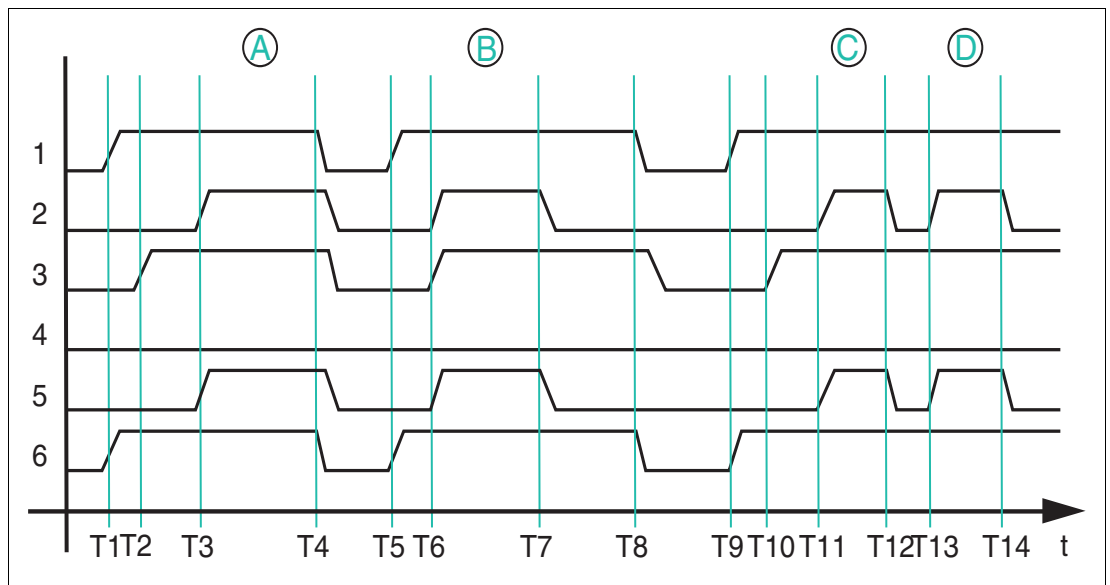


Abbildung 7.3 Zeitlicher Ablauf der Bits im Byte 0

- 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)

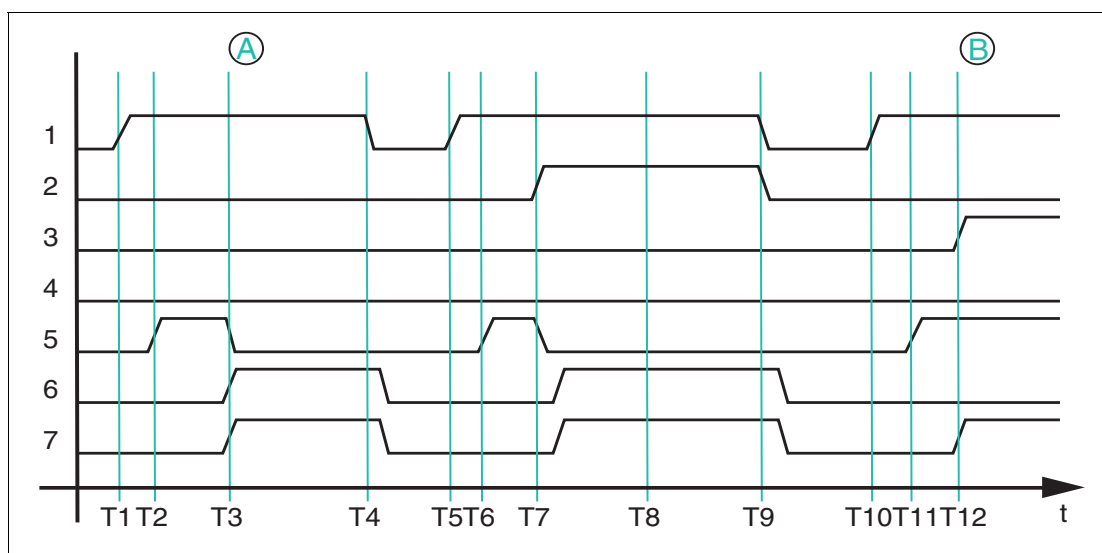
T11: Datenträger C tritt in den Erfassungsbereich ein und die Daten werden erfolgreich geschrieben ("Schreiben erfolgreich" und "Transponder vorhanden" = TRUE); Schreibauftrag weiterhin aktiv ("Auftrag aktiv" = TRUE)

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)

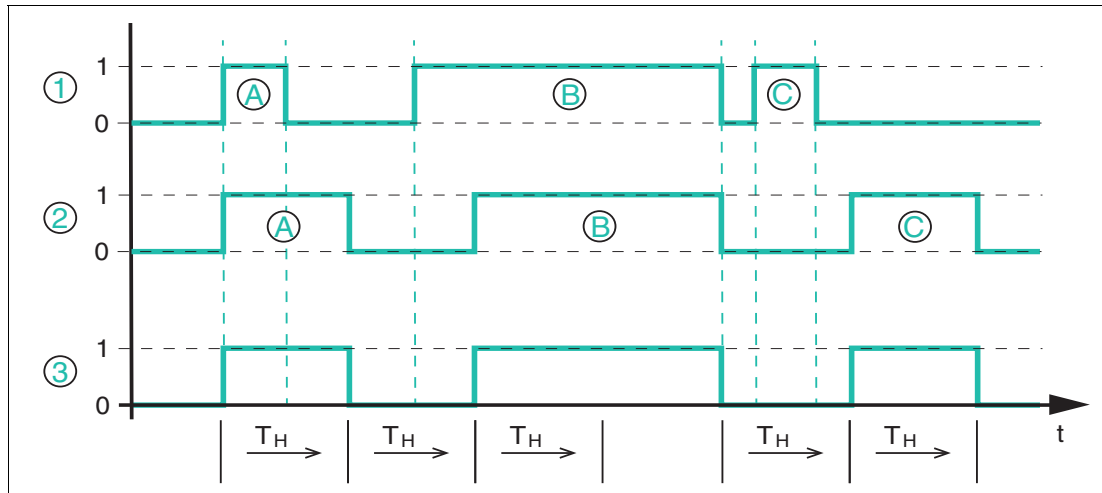
T12: Ein Datenträger B tritt in den Erfassungsbereich ein und wird erfolgreich gelesen ("Lesen erfolgreich" = TRUE)

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 bleibt 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".

7.2.5 Unterbrechung der IO-Link-Kommunikation

Wenn die IO-Link-Kommunikation unterbrocht, arbeitet das Gerät normal weiter. Ist zum Zeitpunkt des Abbruchs ein Leseauftrag aktiv, werden die empfangen Daten im Gerät zwischengespeichert. Wenn das Gerät wieder über IO-Link kommuniziert, werden diese Daten an den IO-Link-Master übertragen.



Hinweis!

Der Zwischenspeicher des Geräts ist für geringe Datenmengen ausgelegt, um kurze Unterbrechungen zu kompensieren. Bei längeren Unterbrechungen der IO-Link-Kommunikation können Daten verloren gehen.

7.3 EasyMode mit PACTware

Sie können das RFID-Schreib-/Lesegerät mit dem IO-Link-Master "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 "IODD Interpreter DTM" verwenden.

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

1. Schließen Sie das RFID-Schreib-/Lesegerät 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 das RFID-Schreib-/Lesegerät mit dem Programm **IODD DTM Configurator**.

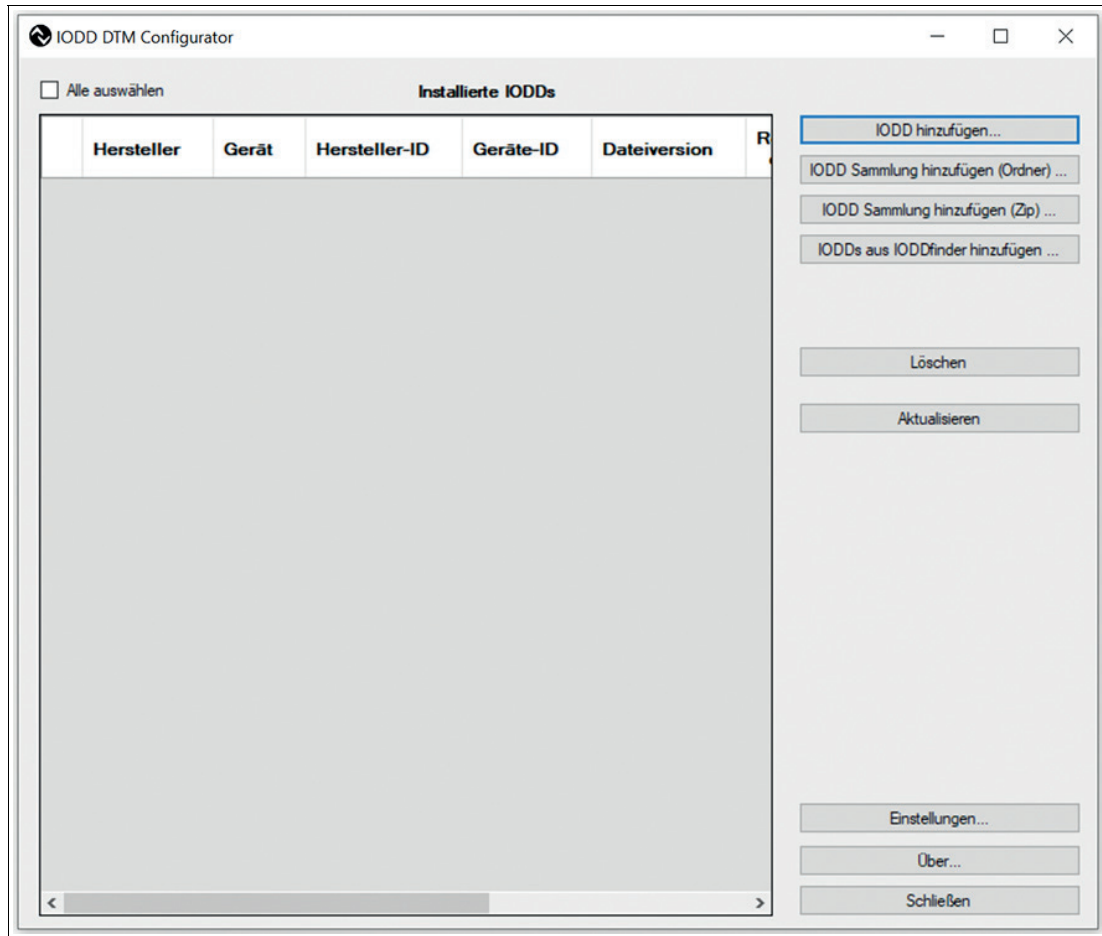


Abbildung 7.4

7. Sie haben mehrere Möglichkeiten, IODDs über den IODDfinder hinzuzufügen:
 - "IODD hinzufügen" fügt eine einzelne IODD hinzu.
 - "IODD Sammlung hinzufügen (Ordner)" fügt einen kompletten Ordner hinzu, der mehrere IODDs beinhalten kann.
 - "IODD-Sammlung hinzufügen (*.zip)" fügt einen archivierten Ordner hinzu, der mehrere IODDs beinhalten kann.¹
 - "IODD aus IODDfinder hinzufügen" erlaubt vollen Zugriff auf die IODDfinder-Datenbank ioddfinder.io-link.com.
8. Über die Filtermaske können Sie Hersteller und Gerät eingrenzen. Tragen Sie als Hersteller "Pepperl+Fuchs" und als Gerät "IQT3-FP-IO-V1" ein.
9. Wählen Sie die benötigte IODD-Dateiversion des Geräts aus (blau unterlegt)
10. Klicken Sie auf "Ausgewählte IODD hinzufügen".

¹ z.B. IODD-Download von der Pepperl+Fuchs-Homepage

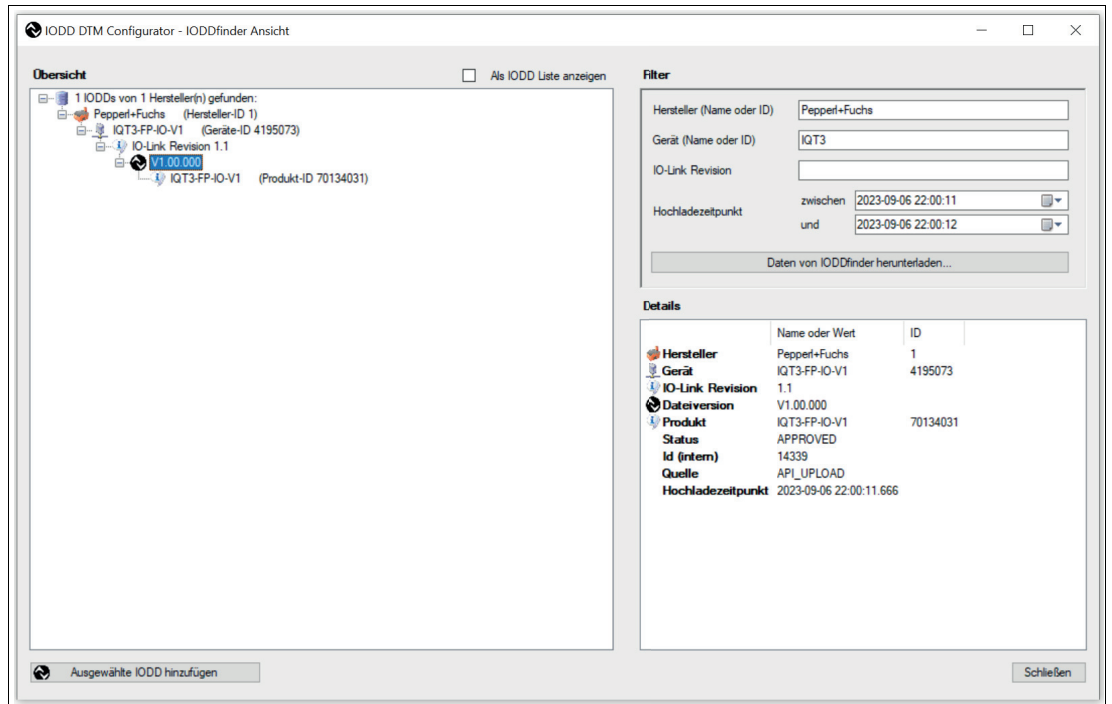


Abbildung 7.5

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

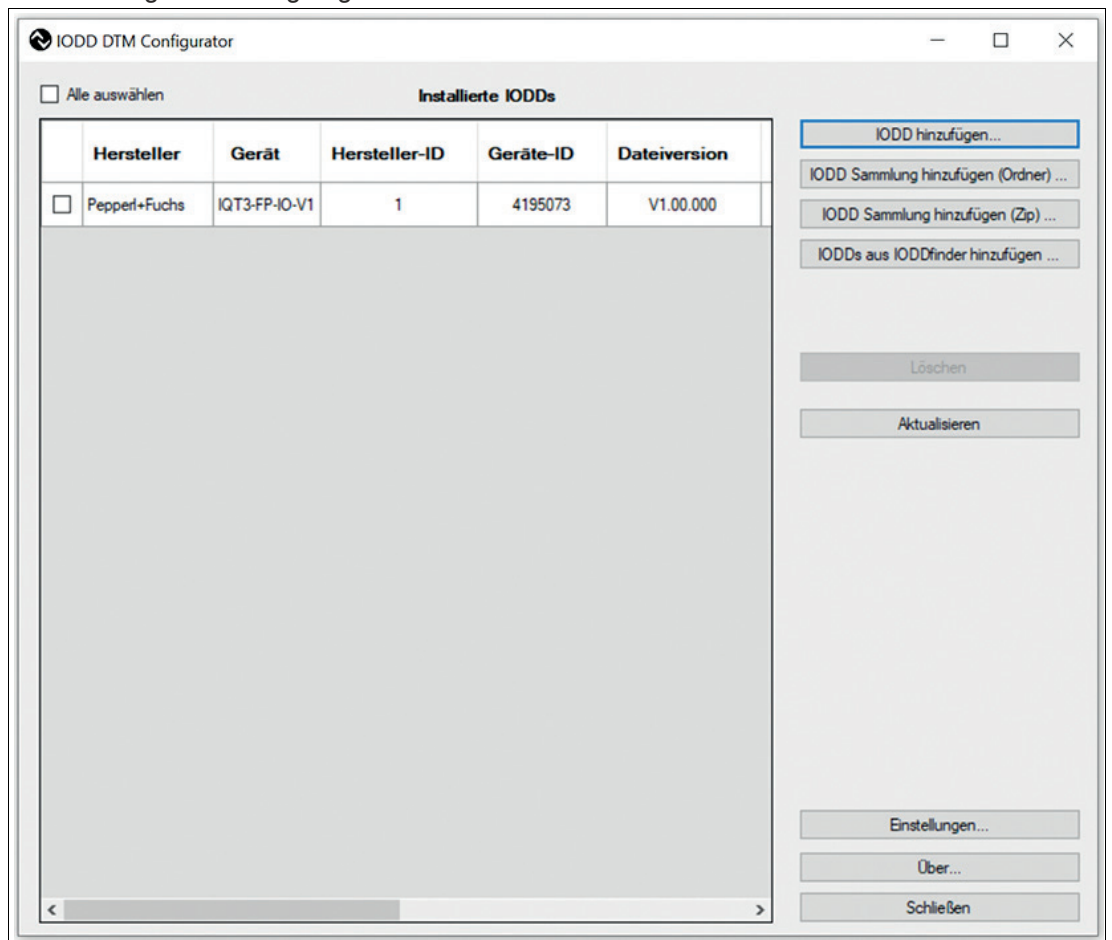


Abbildung 7.6

↳ Schließen Sie das Programm IODD DTM Configurator.

11. Starten Sie PACTware.
12. Klicken Sie mit der rechten Maustaste auf "HOST PC".
13. Wählen Sie den Menüpunkt "Add device".

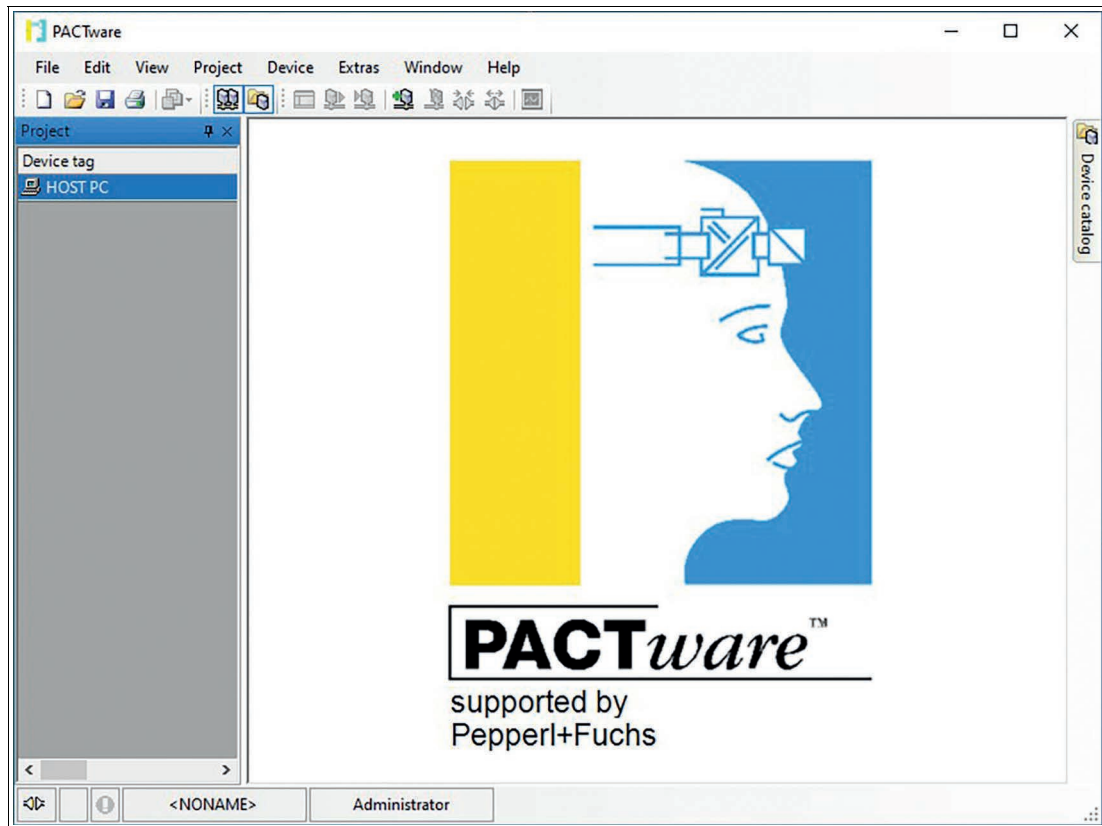


Abbildung 7.7

↳ Das Fenster "Device for" öffnet sich.

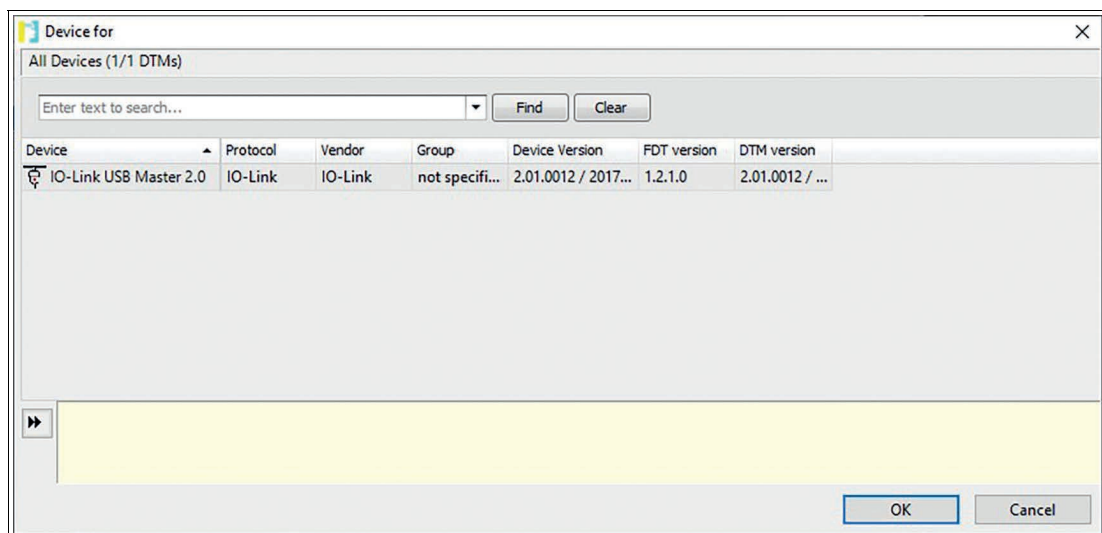


Abbildung 7.8

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



Hinweis!

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

15. Bestätigen Sie mit der Taste "OK".

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

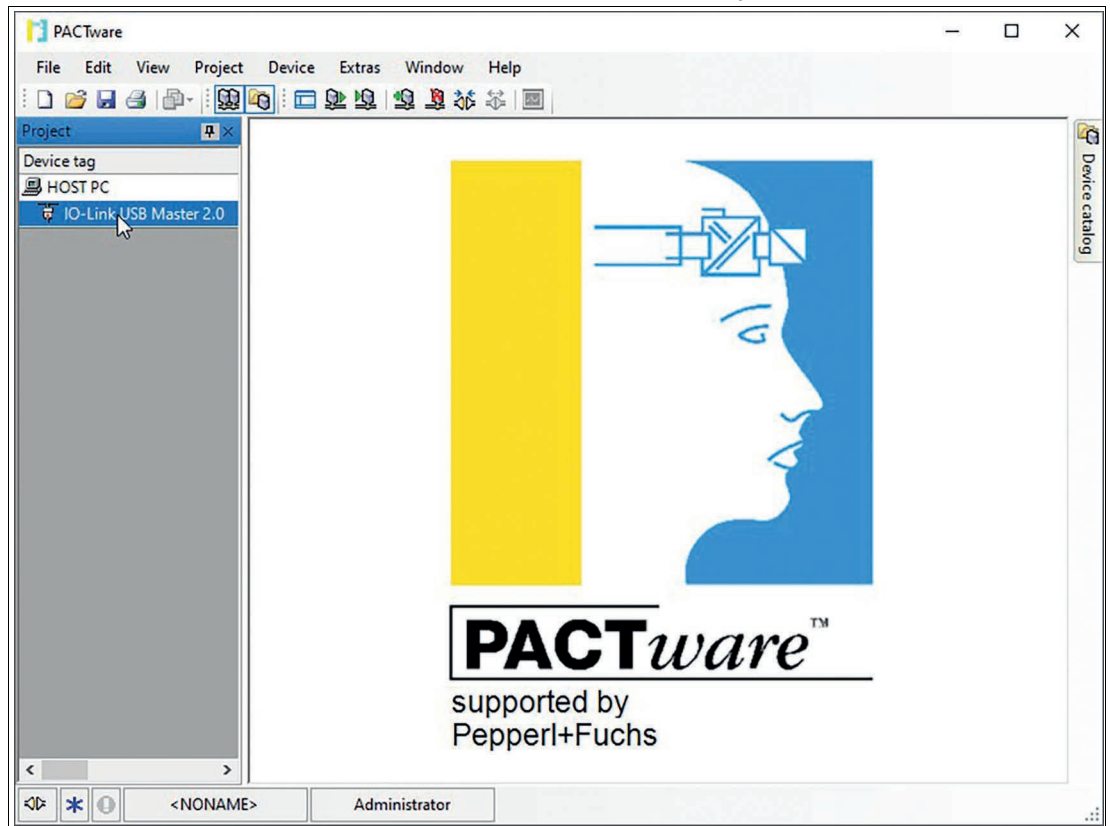


Abbildung 7.9

16. Klicken Sie mit der rechten Maustaste auf den IO-Link-Master.
17. Wählen Sie den Menüpunkt "Add device".

↳ Das Fenster "Device for" öffnet sich.

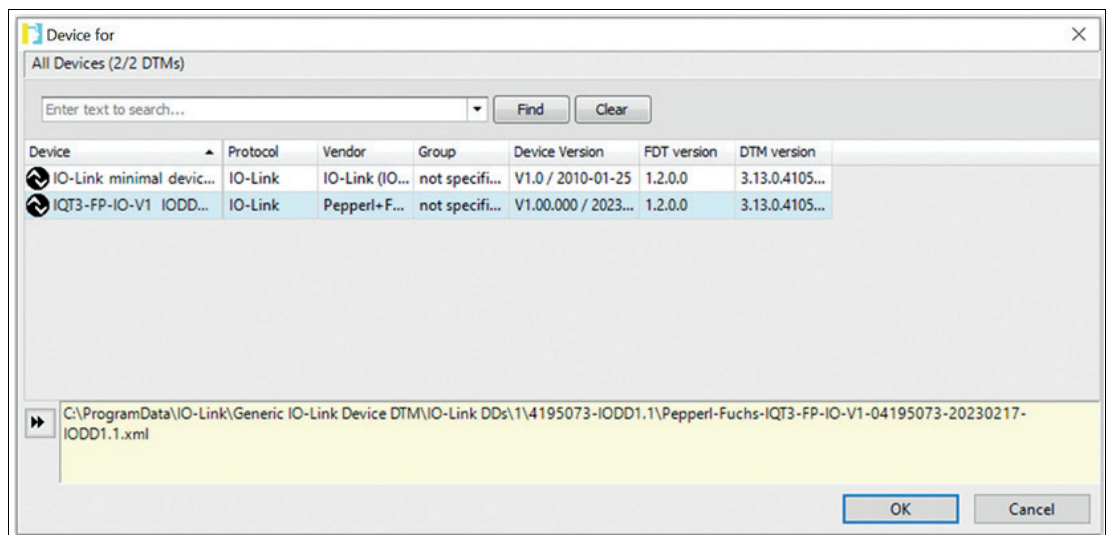


Abbildung 7.10

18. Wählen Sie das gewünschte RFID Schreib-/Lesegerät aus.
19. Bestätigen Sie mit der Taste "OK".
20. Doppelklicken Sie auf das IO-Link-Gerät.

↳ Das Menü Parameter öffnet sich.

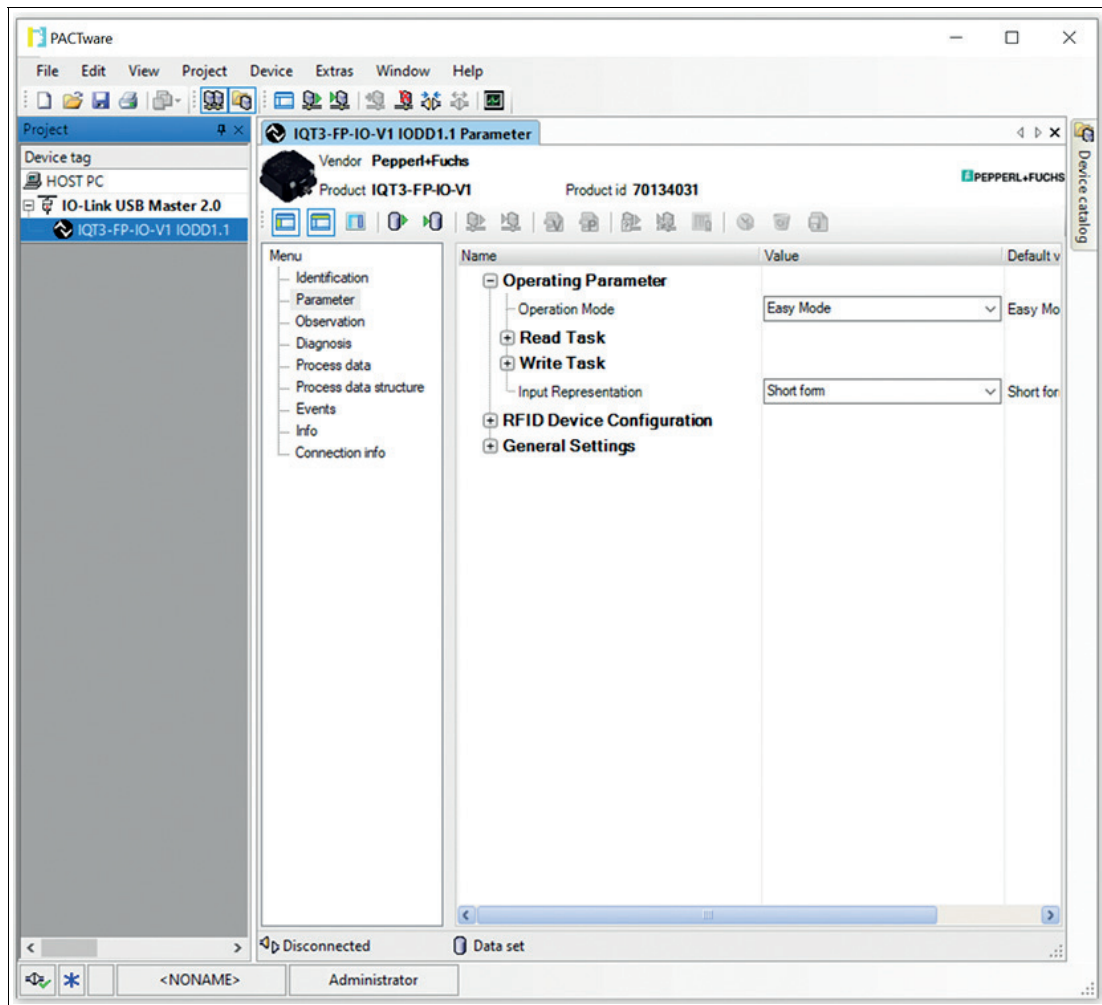


Abbildung 7.11

21. Klicken Sie mit der rechten Maustaste auf das RFID Schreib-/Lesegerät und wählen Sie den Menüpunkt "Connect".
22. Quittieren Sie den Dialog "Read from Device (Upload)?" mit "Yes".
 - ↳ Die angezeigten Parameterwerte in PACTware werden auf das Gerät übertragen.
 - ↳ Eine Verbindung zwischen dem IO-Link-Master und dem Gerät wird hergestellt.

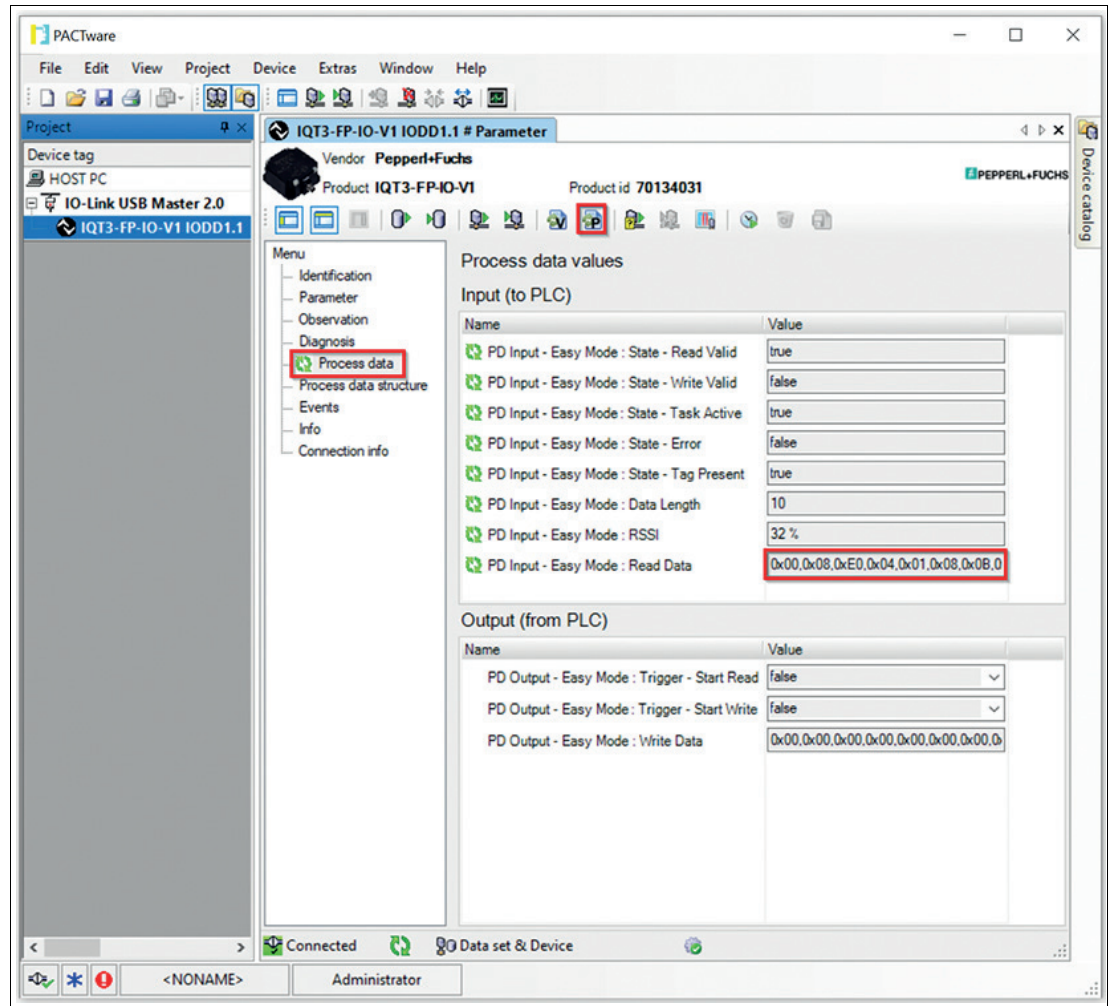


Abbildung 7.12



Tipp

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

- Sobald eine Verbindung besteht, beginnt das RFID-Schreib-/Lesegerät automatisch mit dem Lesen von im Erfassungsbereich befindlichen Transpondern. Die Daten werden in den Eingangsprozessdaten angezeigt, entsprechend der eingestellten Parameter unter "Read Task".
1. Passen Sie die Parameter des Geräts gemäß Ihrer Anwendung an.
 2. Klicken Sie im Geräte-Auswahlmenü auf den Eintrag "Parameter".
↳ Das Menü "Parameter" öffnet sich.
 3. Ändern Sie die gewünschten Parameter und bestätigen Sie die Eingabe mit der Enter-Taste. Klicken Sie anschließend auf das Icon "Write different Values to device".

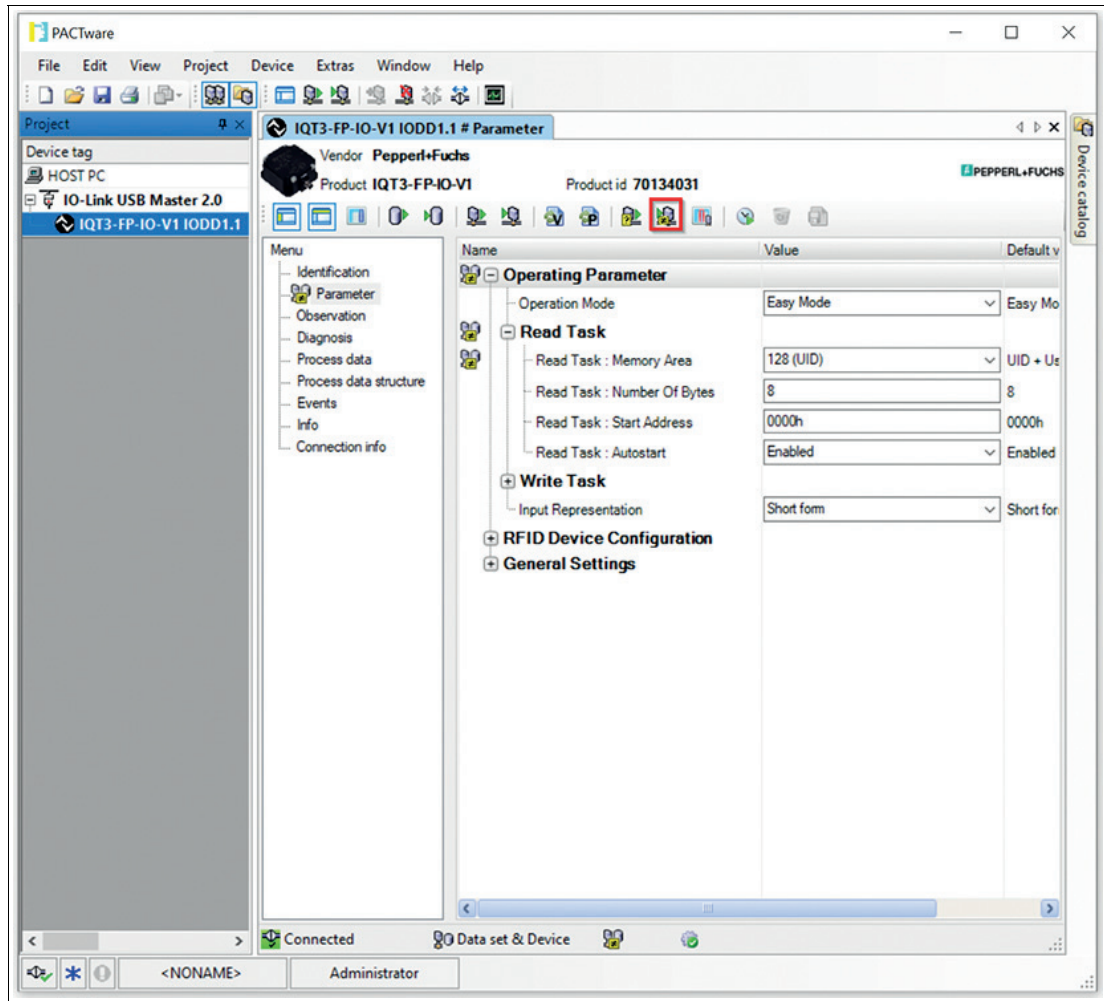


Abbildung 7.13

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. Der Wert muss ein Vielfaches von 4 sein.
<Command>	1 Byte	Befehlscode Kennung des auszuführenden Befehls
<ControlByte>	4 Bit	Steuerbits zur Realisierung der Handshake-Prozedur bzw. Löschen des Gerätespeichers
<FragmentationCounter>	1 Byte	Fragmentierungszähler Anzahl der noch zu übertragenden Fragmente
<FrameLength>	12 Bit	Anzahl der gültigen Bytes innerhalb des Telegramms
<Length User Data>	2 Byte	Länge User Data Angabe der Länge der eingelesenen Daten von der Speicherbank User Memory
<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. Der Wert muss ein Vielfaches von 4 sein.
<ParameterData>	Länge variabel	Parameterdaten Parameterdatensatz, welcher ausgelesen bzw. übertragen wurde.
<ParameterName>	2 Byte	Bezeichnung Geräteparameter Über die Bezeichnung des Geräteparameters wird festgelegt, auf welchen HF-Parameter zugegriffen werden soll.
<Status>	1 Byte	Statusbyte Statusinformation über den Ausführungszustand des Befehls.
<SystemCode>	1 Byte	Systemkennzeichen Die Systemkennung für das HF-System ist "Q" (16#51).
<TelegramLength>	2 Byte	Telegrammlänge mit allen Fragmenten
<UID>	8 Byte	Unique Identifier Eindeutige Seriennummer eines Datenträgers

Name	Länge	Bedeutung
<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 _S	U _M	U _D	0	<FrameLength>			
1	Frame Length	<Frame Length>							
2	Fragmentation Counter	<Fragmentation Counter>							
3	Telegram Length (High Byte) ¹	<Telegram Length (High Byte)>							
4	Telegram Length (Low Byte) ¹	<Telegram Length (Low Byte)>							
5	Command ¹	<Command>							
6	Parameter / Data	<Data Byte 1>							
7	Parameter / Data	<Data Byte 2>							
8	Parameter / Data	...							
...	Parameter / Data	<Data Byte X>							
...	Parameter / Data	16#00							
...	Parameter / Data	...							
31	Parameter / Data	16#00							

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
- **HF-Konfigurationsbefehle:** Einstellung der HF-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 _S	U _M	U _D	0	<FrameLength>			
1	Frame Length	<Frame Length>							
2	Fragmentation Counter	<Fragmentation Counter>							
3	Telegram Length (High Byte) ¹	<Telegram Length (High Byte)>							
4	Telegram Length (Low Byte) ¹	<Telegram Length (Low Byte)>							
5	Command ¹	<Command>							
6	Status	<Status>							
7	Parameter / Data	<Data Byte 1>							
8	Parameter / Data	<Data Byte 2>							
...	Parameter / Data	...							
...	Parameter / Data	<Data Byte X>							
...	Parameter / Data	0x00							
...	Parameter / Data	...							
31	Parameter / Data	0x00							

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. .

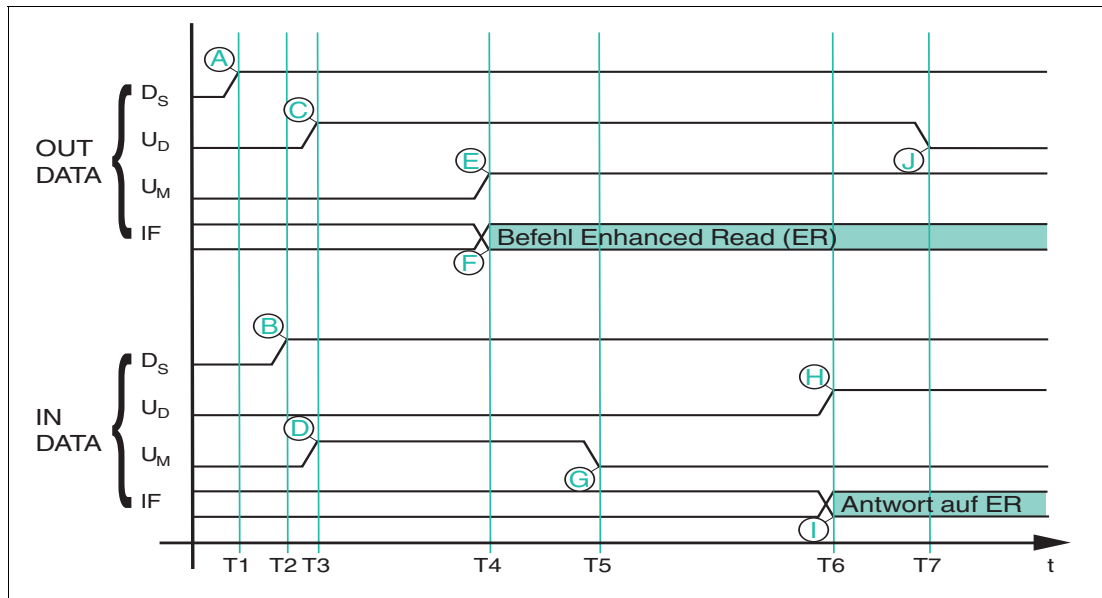
In den Bytes <Data Byte> werden die vom Gerät auf Grund der Befehlssteuerung zu übertragenden Daten zurückgesendet. Es kann sich dabei beispielsweise um von einem Datenträger eingelesene Daten handeln oder Parameterwerte der UHF-Einstellungen.

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 dem Schreib-/Lesegerät synchronisiert werden. Die Eingangs- und Ausgangsprozessdaten werden zyklisch übertragen. Die Steuerung des Datenflusses über die Ausgangsprozessdaten 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_S** Delete Bit; löscht internen Speicher des Geräts
- U_M** Update Master-Bit
- U_D** 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 (UM_OUTPUT <> UM_INPUT) then (* Prüfe, ob Device neue Daten empfangen kann *)
```

```
UM_OUTPUT := UM_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	Befehlscode	Befehlsbeschreibung
SF	16#01	Single Read Fixcode siehe "Single Read Fixcode (SF)" auf Seite 44 Einmaliger Lesezugriff auf den Fixcode (UID)
EF	16#1D	Enhanced Read Fixcode siehe "Enhanced Read Fixcode (EF)" auf Seite 47 Permanenter Lesezugriff auf den Fixcode (UID)
SR	16#10	Single Read Words siehe "Single Read Words (SR)" auf Seite 50 Einmaliger Lesezugriff auf den Nutzdatenbereich
ER	16#19	Enhanced Read Words siehe "Enhanced Read Words (ER)" auf Seite 54 Permanenter Lesezugriff auf den Nutzdatenbereich

Kürzel	Befehls-code	Befehlsbeschreibung
SW	16#40	Single Write Words siehe "Single Write Words (SW)" auf Seite 57 Einmaliger Schreibzugriff auf den Nutzdatenbereich
EW	16#1A	Enhanced Write Words siehe "Enhanced Write Words with Lock (EL)" auf Seite 68 Permanenter Schreibzugriff auf den Nutzdatenbereich
SL	16#47	Single Write Words with Locks siehe "Single Write Words with Locks (SL)" auf Seite 64 Einmaliger Schreibzugriff auf den Nutzdatenbereich mit anschließendem Schreibschutz
EL	16#48	Enhanced Write Words with Lock siehe "Enhanced Write Words (EW)" auf Seite 61 Permanenter Schreibzugriff auf den Nutzdatenbereich mit anschließendem Schreibschutz

Systembefehle

Kürzel	Befehls-code	Befehlsbeschreibung
QU	16#02	Quit siehe "Quit (QU)" auf Seite 71 Abbruch eines aktiven Enhanced Befehls
VE	16#03	Version siehe "Version (VE)" auf Seite 72 Auslesen der Firmware Version
CT	16#04	Change Tag siehe "Change Tag (CT)" auf Seite 73 Einstellen des Transpondertyps

Konfigurationsbefehle

Kürzel	Befehls-code	Befehlsbeschreibung
RP	16#BE	Read Parameter siehe "Read Parameter (RP)" auf Seite 74 Geräteparameter auslesen
WP	16#BF	Write Parameter siehe "Write Parameter (WP)" auf Seite 76 Geräteparameter setzen

8.6.1 Schreib-/Lesebefehle

Bei der Beschreibung der nachfolgenden Befehle und Antworten wird das Long-Form-Datenformat verwendet. Das Ende eines Single-Befehls wird über ein Status 16#0F-Telegramm zurückgemeldet. Bei der Verwendung des Short-Form-Datenformats entfallen der Fixcode und die zusätzlichen Längeninformationen. Das Ende eines Enhanced-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 Fixcode UID eines Transponders innerhalb der Erfassungszone durch. Die eingelesenen Informationen werden für jeden gelesenen Transponder in einem separaten Datentelegramm mit dem Statuswert 16#00 übertragen. Zusätzlich wird für jeden gelesenen Transponder ein weiteres Telegramm mit dem Statuswert 16#0B übertragen, das den RSSI-Wert zur Signalstärke der Transponder-Antwort enthält. Das Ende der Befehlsausführung wird über ein Abschlusstelegramm signalisiert. Das Abschlusstelegramm 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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#06							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#03							
Byte 5	Command	16#01							
Byte 6	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 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.



Hinweis!

Setzen Sie die nicht relevanten Bytes des Befehlsfragments auf den Wert 16#00.

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	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#01							
Byte 6	Status	16#00							
Byte 7	Parameter / Data	16#00							
Byte 8	Parameter / Data	16#08							
Byte 9	Parameter / Data	<UID Byte 1>							
Byte 10	Parameter / Data	<UID Byte 2>							
Byte 11	Parameter / Data	<UID Byte 3>							
Byte 12	Parameter / Data	<UID Byte 4>							
Byte 13	Parameter / Data	<UID Byte 5>							
Byte 14	Parameter / Data	<UID Byte 6>							
Byte 15	Parameter / Data	<UID Byte 7>							
Byte 16	Parameter / Data	<UID Byte 8>							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 17	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.4

Die <FrameLength> hat einen Wert von 16#11. Dadurch wird die Länge des Fragments bis einschließlich des Bytes <UID Byte 8> angegeben. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die <TelegramLength> hat den Wert 16#0E und spezifiziert die Länge des Antworttelegramms, beginnend ab der Telegrammlänge bis einschließlich des Bytes <UID Byte 8>.

Der Parameter <Command> hat den Wert 16#01 und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00 und signalisiert dadurch, dass der Zugriff auf den Transponder erfolgreich war.

Die Längenangabe zum Fixcode hat immer eine Größe von 2 Byte und den Wert 16#0008.

Der eingelesene Fixcode dient zur eindeutigen Identifizierung des gelesenen Transponders.

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

Antwort Signalstärke Transponderantwort, 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	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#01							
Byte 6	Status	16#0B							
Byte 7	Parameter / Data	16#01							
Byte 8	Parameter / Data	<RSSI-Wert>							
Byte 9	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.5

Die <FrameLength> ist konstant beim Wert 16#09. Das Fragment erstreckt sich dabei bis einschließlich des Bytes <RSSI-Wert>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die Telegrammlänge <TelegramLength> hat den Wert 16#06.

Das Byte <Command> wird zurückgespiegelt und hat den Wert 16#01 wie im Befehlstelegramm. Der <Status> für das Telegramm zur Signalisierung des RSSI-Werts ist 16#0B. Das erste Byte <Parameter / Data> hat den Wert 16#01. Das zweite Byte <Parameter / Data> beschreibt den <RSSI-Wert>.

Alle anderen Bytes des Antwortfragments sind nicht relevant und 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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#01							
Byte 6	Status	16#0F							
Byte 7	Parameter / Data	<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 8.6

Die <FrameLength> beträgt in der Antwort Befehlsende des "Single Read Fixcode" den Wert 16#0B. Das Fragment erstreckt sich 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.

Der Parameter <Command> hat den Wert 16#01 und wird innerhalb des Datentelegramms zurückgespiegelt. Der <Status> für das Telegramm zur Signalisierung des Befehlsendes ist 16#0F.

Die Anzahl der identifizierten Transponder wird innerhalb von 4 Byte übertragen. Dabei wird die Anzahl in ASCII dargestellt.

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

Wenn während der Befehlsausführung kein Transponder erkannt wurde, entfallen die Datentelegramme. Es wird 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 Fixcode eines Transponders innerhalb der Erfassungszone durch. Die eingelesenen Informationen werden für jeden gelesenen Transponder in einem separaten Datentelegramm mit dem Statuswert 16#00 übertragen. Zusätzlich wird für jeden gelesenen Transponder ein weiteres Telegramm mit dem Statuswert 16#0B übertragen, das den RSSI-Wert zur Signalstärke der Transponder-Antwort enthält. Wenn ein Transponder die Erfassungszone verlässt, wird dies über ein Telegramm mit dem Fixcode 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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#06							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#03							
Byte 5	Command	16#1D							
Byte 6	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.7

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 Fixcode ist 16#1D.

Hinweis!

Setzen Sie die nicht relevanten Bytes des Befehlsfragments auf den Wert 16#00.

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	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#1D							
Byte 6	Status	16#00							
Byte 7	Length UID (High Byte)	16#00							
Byte 8	Length UID (Low Byte)	16#08							
Byte 9	Parameter / Data	<UID Byte 1>							
Byte 10	Parameter / Data	<UID Byte 2>							
Byte 11	Parameter / Data	<UID Byte 3>							
Byte 12	Parameter / Data	<UID Byte 4>							
Byte 13	Parameter / Data	<UID Byte 5>							
Byte 14	Parameter / Data	<UID Byte 6>							
Byte 15	Parameter / Data	<UID Byte 7>							
Byte 16	Parameter / Data	<UID Byte 8>							

2023-11

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 17	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.8

Die <FrameLength> hat einen Wert von 16#11. Dadurch wird die Länge des Fragments bis einschließlich des Bytes <Fixcode Byte 8> angegeben. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die <TelegramLength> hat den Wert 16#0E und spezifiziert die Länge des Antworttelegramms, beginnend ab der Telegrammlänge bis einschließlich des Bytes <UID Byte 8>.

Der Parameter <Command> hat den Wert 16#1D und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00 und signalisiert, dass der Zugriff auf den Transponder erfolgreich war. Hieran schließt sich eine Längenangabe zum Fixcode. Diese hat immer eine Größe von 2 Byte und den Wert 16#0008. Danach folgt der eingelesene Fixcode zur eindeutigen Identifizierung des gelesenen Transponders.

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

Antwort Signalstärke Transponderantwort, 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	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#1D							
Byte 6	Status	16#0B							
Byte 7	Parameter / Data	16#01							
Byte 8	Parameter / Data	<RSSI-Wert>							
Byte 9	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.9

Die <FrameLength> ist konstant beim Wert 16#09. Das Fragment erstreckt sich dabei bis einschließlich des Bytes <RSSI-Wert>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die Telegrammlänge <TelegramLength> hat den Wert 16#06.

Das Byte <Command> wird zurückgespiegelt und hat den Wert 16#1D wie im Befehlstelegramm. Der <Status> für das Telegramm zur Signalisierung des RSSI-Werts ist 16#0B. Das erste Byte <Parameter / Data> hat den Wert 16#01. Das zweite Byte <Parameter / Data> beschreibt den <RSSI-Wert>.

Alle anderen Bytes des Antwortfragments sind nicht relevant und 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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#1D							
Byte 6	Status	16#05							
Byte 7	Length UID (High Byte)	16#00							
Byte 8	Length UID (Low Byte)	16#08							
Byte 9	Parameter / Data	<UID Byte 1>							
Byte 10	Parameter / Data	<UID Byte 2>							
Byte 11	Parameter / Data	<UID Byte 3>							
Byte 12	Parameter / Data	<UID Byte 4>							
Byte 13	Parameter / Data	<UID Byte 5>							
Byte 14	Parameter / Data	<UID Byte 6>							
Byte 15	Parameter / Data	<UID Byte 7>							
Byte 16	Parameter / Data	<UID Byte 8>							
Byte 17	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.10

Die <FrameLength> hat einen Wert von 16#11. Dadurch wird die Länge des Fragments bis einschließlich des Bytes <UID Byte 8> angegeben. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die <TelegramLength> hat den Wert 16#0E und spezifiziert die Länge des Antworttelegramms, beginnend ab der Telegrammlänge bis einschließlich des Bytes <UID Byte 8>.

Der Parameter <Command> hat den Wert 16#1D und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#05 und signalisiert, dass der Transponder die Erfassungszone verlassen hat. Hieran schließt sich eine Längenangabe zum Fixcode. Diese hat immer eine Größe von 2 Byte und den Wert 16#0008. Danach folgt der eingeleseene Fixcode zur eindeutigen Identifizierung des Transponders, der den Erfassungsbe-
reich verlassen hat.

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

Single Read Words (SR)

Der Befehl "Single Read Words" hat den Befehlscode 16#10 und führt einen einmaligen Lesezugriff auf den Nutzdatenbereich eines Transponders innerhalb des Erfassungsbereichs durch. Die eingelesenen Informationen werden für jeden gelesenen Transponder in einem separaten Datentelegramm mit dem Statuswert 16#00 übertragen. Zusätzlich wird für jeden gelesenen Transponder ein weiteres Telegramm mit dem Statuswert 16#0B gesendet, das den RSSI-Wert zur Signalstärke der Transponder-Antwort enthält. 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.

Die auf einem Transponder zur Verfügung stehende Nutzdatenmenge ist abhängig vom verwendeten Chiptyp und kann unterschiedlich groß sein. Für detailliertere Informationen zu Transpondertypen .

Der Parameter <ByteAddress> legt die Startadresse innerhalb des Nutzerdatenbereichs fest. Der Wert von <ByteAddress> ist bytebezogen. Es können nur Vielfache der Speicherblockgröße parametrisiert werden. Über <Number of Bytes> wird die Anzahl der zu lesenden Bytes festgelegt. Die Byteanzahl muss ein Vielfaches der Speicherblockgröße sein.

Befehl:

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#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 8.11

Die <FrameLength> hat den Wert 16#0A und erstreckt sich bis einschließlich des Bytes <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 dem Byte <Number of Bytes (Low Byte)>.

Der Befehlscode <Command> für den Befehl Enhanced Read Fixcode ist 16#10.

Der Parameter <ByteAddress> gibt die Startadresse innerhalb des Nutzerdatenbereichs an, ab der die Speicherblöcke eingelesen werden. Über den Parameter <Number of Bytes> wird die Anzahl der einzulesenden Bytes festgelegt.



Hinweis!

Setzen Sie die nicht relevanten Bytes des Befehlsfragments auf den Wert 16#00.

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	16#0			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 5	Command	16#10							
Byte 6	Status	16#00							
Byte 7	Length UID (High Byte)	16#00							
Byte 8	Length UID (Low Byte)	16#08							
Byte 9	Parameter / Data	<UID Byte 1>							
Byte 10	Parameter / Data	<UID Byte 2>							
Byte 11	Parameter / Data	<UID Byte 3>							
Byte 12	Parameter / Data	<UID Byte 4>							
Byte 13	Parameter / Data	<UID Byte 5>							
Byte 14	Parameter / Data	<UID Byte 6>							
Byte 15	Parameter / Data	<UID Byte 7>							
Byte 16	Parameter / Data	<UID Byte 8>							
Byte 17	Parameter / Data	<Length User Data (High Byte)>							
Byte 18	Parameter / Data	<Length User Data (Low Byte)>							
Byte 19	Parameter / Data	<User Data Byte 1>							
Byte 20	Parameter / Data	<User Data Byte 2>							
Byte 21	Parameter / Data	<User Data Byte 3>							
Byte 22	Parameter / Data	<User Data Byte 4>							
...	Parameter / Data	...							
...	Parameter / Data	<User Data Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.12

Die Länge des Fragments <FrameLength> ist abhängig von der Anzahl der eingelesenen Bytes des identifizierten Transponders. Die <FrameLength> beinhaltet alle Bytes bis einschließlich des Bytes <User Data 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 Anzahl der eingelesenen Bytes des Transponders. Die Größe des Telegramms erstreckt sich bis einschließlich des Bytes <User Data Byte X>.

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 zum Fixcode. Diese hat immer eine Größe von 2 Byte und den Wert 16#0008. Danach folgt der eingelesene Fixcode zur eindeutigen Identifizierung des gelesenen Transponders. Nach dem Fixcode 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 X>.

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

Antwort Signalstärke Transponderantwort, 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	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							

2023-11

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#06							
Byte 5	Command	16#10							
Byte 6	Status	16#0B							
Byte 7	Parameter / Data	16#01							
Byte 8	Parameter / Data	<RSSI-Wert>							
Byte 9	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.13

Die <FrameLength> ist konstant beim Wert 16#09. Das Fragment erstreckt sich dabei bis einschließlich des Bytes <RSSI-Wert>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die Telegrammlänge <TelegramLength> hat den Wert 16#06.

Das Byte <Command> wird zurückgespiegelt und hat den Wert 16#10 wie im Befehlstelegramm. Der <Status> für das Telegramm zur Signalisierung des RSSI-Werts ist 16#0B. Das erste Byte <Parameter / Data> hat den Wert 16#01. Das zweite Byte <Parameter / Data> beschreibt den <RSSI-Wert>.

Alle anderen Bytes des Antwortfragments sind nicht relevant und 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_S	U_M	U_D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#10							
Byte 6	Status	16#0F							
Byte 7	Parameter / Data	<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 8.14

Die <FrameLength> ist in der Rückantwort für das Befehlsende des "Single Read Words" konstant beim 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 Transponder 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 Transponder 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 Words (ER)

Der Befehl "Enhanced Read Words" hat den Befehlscode 16#19 und führt einen permanenten Lesezugriff auf den Nutzdatenbereich eines Transponders innerhalb der Erfassungszone durch. Die eingelesenen Informationen werden für jeden gelesenen Transponder in einem separaten Datentelegramm mit dem Statuswert 16#00 übertragen. Zusätzlich wird für jeden gelesenen Transponder ein weiteres Telegramm mit dem Statuswert 16#0B übertragen, welches Informationen zur Signalstärke der Transponder-Antwort (RSSI-Wert) enthält. Wenn ein Transponder die Erfassungszone verlässt, wird dies über ein Telegramm mit dem Fixcode des Transponders gemeldet. Dieses Telegramm hat den Statuswert 16#05. Die Befehlsausführung wird über ein Quit-Befehl gestoppt.

Die auf einem Transponder zur Verfügung stehende Nutzdatenmenge ist abhängig vom verwendeten Chiptyp und kann unterschiedlich groß sein. Für detailliertere Informationen zu Transpondertypen .

Der Parameter <ByteAddress> legt die Startadresse innerhalb des Nutzerdatenbereichs fest. Der Wert von <ByteAddress> ist bytebezogen. Es können nur Vielfache der Speicherblockgröße parametrisiert werden. Über <Number of Bytes> wird die Anzahl der zu lesenden Bytes festgelegt. Die Byteanzahl muss ein Vielfaches der Speicherblockgröße sein.

Befehl:

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#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 8.15

Die <FrameLength> hat den Wert 16#0A und erstreckt sich bis einschließlich des Bytes <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 dem Byte <Number of Bytes (Low Byte)>.

Der Befehlscode <Command> für den Befehl Enhanced Read Fixcode ist 16#19.

Der Parameter <ByteAddress> gibt die Startadresse innerhalb des Nutzerdatenbereichs an, ab der die Speicherblöcke eingelesen werden. Über den Parameter <Number of Bytes> wird die Anzahl der einzulesenden Bytes festgelegt.



Hinweis!

Setzen Sie die nicht relevanten Bytes des Befehlsfragments auf den Wert 16#00.

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	16#0			
Byte 1	Frame Length	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#19							
Byte 6	Status	16#00							
Byte 7	Length UID (High Byte)	16#00							
Byte 8	Length UID (Low Byte)	16#08							
Byte 9	Parameter / Data	<UID Byte 1>							
Byte 10	Parameter / Data	<UID Byte 2>							
Byte 11	Parameter / Data	<UID Byte 3>							
Byte 12	Parameter / Data	<UID Byte 4>							
Byte 13	Parameter / Data	<UID Byte 5>							
Byte 14	Parameter / Data	<UID Byte 6>							
Byte 15	Parameter / Data	<UID Byte 7>							
Byte 16	Parameter / Data	<UID Byte 8>							
Byte 17	Parameter / Data	<Length User Data (High Byte)>							
Byte 18	Parameter / Data	<Length User Data (Low Byte)>							
Byte 19	Parameter / Data	<User Data Byte 1>							
Byte 20	Parameter / Data	<User Data Byte 2>							
Byte 21	Parameter / Data	<User Data Byte 3>							
Byte 22	Parameter / Data	<User Data Byte 4>							
...	Parameter / Data	...							
...	Parameter / Data	<User Data Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.16

Die Länge des Fragments <FrameLength> ist abhängig von der Anzahl der eingelesenen Bytes des identifizierten Transponders. Die <FrameLength> beinhaltet alle Bytes bis einschließlich des Bytes <User Data 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 Anzahl der eingelesenen Bytes des Transponders. Die Größe des Telegrams erstreckt sich bis einschließlich des Bytes <User Data Byte X>.

Der Parameter <Command> hat den Wert 16#19 und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00. Hieran schließt sich eine Längenangabe zum Fixcode. Diese hat immer eine Größe von 2 Byte. Danach folgt der eingelesene Fixcode zur eindeutigen Identifizierung des gelesenen Transponders. Nach dem Fixcode folgt eine Längenangabe für die eingelesenen User Data. Diese hat immer eine Größe von 2 Byte und den Wert 16#0008. Abschließend folgen die eingelesenen User Data des identifizierten Transponders bis einschließlich <User Data Byte X>.

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

Antwort Signalstärke Transponderantwort, 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	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#19							
Byte 6	Status	16#0B							
Byte 7	Parameter / Data	16#01							
Byte 8	Parameter / Data	<RSSI-Wert>							
Byte 9	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.17

Die <FrameLength> ist konstant beim Wert 16#09. Das Fragment erstreckt sich dabei bis einschließlich des Bytes <RSSI-Wert>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die Telegrammlänge <TelegramLength> hat den Wert 16#06.

Das Byte <Command> wird zurückgespiegelt und hat den Wert 16#19 wie im Befehlstelegramm. Der <Status> für das Telegramm zur Signalisierung des RSSI-Werts ist 16#0B. Das erste Byte <Parameter / Data> hat den Wert 16#01. Das zweite Byte <Parameter / Data> beschreibt den <RSSI-Wert>.

Alle anderen Bytes des Antwortfragments sind nicht relevant und 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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							

2023-11

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#19							
Byte 6	Status	16#05							
Byte 7	Length UID (High Byte)	16#00							
Byte 8	Length UID (Low Byte)	16#08							
Byte 9	Parameter / Data	<UID Byte 1>							
Byte 10	Parameter / Data	<UID Byte 2>							
Byte 11	Parameter / Data	<UID Byte 3>							
Byte 12	Parameter / Data	<UID Byte 4>							
Byte 13	Parameter / Data	<UID Byte 5>							
Byte 14	Parameter / Data	<UID Byte 6>							
Byte 15	Parameter / Data	<UID Byte 7>							
Byte 16	Parameter / Data	<UID Byte 8>							
Byte 17	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.18

Die <FrameLength> hat einen Wert von 16#11. Dadurch wird die Länge des Fragments bis einschließlich des Bytes <UID Byte 8> angegeben. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die <TelegramLength> hat den Wert 16#0E und spezifiziert die Länge des Antworttelegramms, beginnend ab der Telegrammlänge bis einschließlich des Bytes <UID Byte 8>.

Der Parameter <Command> hat den Wert 16#19 und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#05 und signalisiert, dass der Transponder die Erfassungszone verlassen hat. Hieran schließt sich eine Längenangabe zum Fixcode. Diese hat immer eine Größe von 2 Byte und den Wert 16#0008. Danach folgt der eingelesene Fixcode zur eindeutigen Identifizierung des Transponders, der den Erfassungsbe-
reich verlassen hat.

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

Single Write Words (SW)

Der Befehl "Single Write Words" hat den Befehlscode 16#40 und führt einen einmaligen Schreibvorgang auf den Nutzdatenbereich eines Transponders innerhalb des Erfassungsbe-
reichs durch. Der erfolgreiche Schreibvorgang wird für jeden beschriebene Transponder in einem separaten Datentelegramm mit dem Statuswert 16#00 signalisiert. Das Datentelegramm enthält den Fixcode des Transponders, auf dem der Nutzdatenbereich beschrieben wurde. Zusätzlich wird für jeden beschriebenen Transponder ein weiteres Telegramm mit dem Statuswert 16#0B übertragen, welches Informationen zur Signalstärke der Transponder-Antwort (RSSI-Wert) enthält. 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.

Die auf einen Datenträger zur Verfügung stehende Nutzdatenmenge ist abhängig vom verwendeten Chiptyp und kann unterschiedlich groß sein. Sie finden detailliertere Informationen in der Übersicht der Transpondertypen.

Der Parameter <ByteAddress> legt die Startadresse innerhalb des Nutzdatenbereichs fest. Der Wert von <ByteAddress> ist bytebezogen. Es können nur Vielfache der Speicherblockgröße parametrisiert werden. Über <Number of Bytes> wird die Anzahl der zu schreibenden Bytes festgelegt. Die Byteanzahl muss ein Vielfaches der Speicherblockgröße sein.

Befehl:

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	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#40							
Byte 6	Parameter / Data	<Byte Address (High Byte)>							
Byte 7	Parameter / Data	<Byte Address (Low Byte)>							
Byte 8	Parameter / Data	<Number of Bytes (High Byte)>							
Byte 9	Parameter / Data	<Number of Bytes ((Low Byte)>							
Byte 10	Parameter / Data	<Write Data Byte 1>							
Byte 11	Parameter / Data	<Write Data Byte 2>							
Byte 12	Parameter / Data	<Write Data Byte 3>							
Byte 13	Parameter / Data	<Write Data Byte 4>							
...	Parameter / Data	...							
...	Parameter / Data	<Write Data Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.19

Der Wert von <FrameLength> ist abhängig von der Anzahl der zuschreibenden Bytes. Das Fragment endet mit dem Byte <Write Data 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 Anzahl der zuschreibenden Bytes. Das Telegramm endet mit dem Byte <Write Data Byte X>.

Der Befehlscode <Command> für den Befehl "Single Write Words" ist 16#40.

Der Parameter <ByteAddress> gibt die Startadresse innerhalb des Nutzdatenbereichs an, ab der die Daten geschrieben werden. Über den Parameter <Number of Bytes> wird die Anzahl der zu schreibenden Bytes festgelegt. Anschließend folgen die auf den Transponder zu schreibenden Informationen <Write Data Byte>.



Hinweis!

Setzen Sie die nicht relevanten Bytes des Befehlsfragments auf den Wert 16#00.

Antwort Datentelegramm, Daten erfolgreich geschrieben, 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	16#0			
Byte 1	Frame Length	16#11							

2023-11

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#40							
Byte 6	Status	16#00							
Byte 7	Length UID (High Byte)	16#00							
Byte 8	Length UID (Low Byte)	16#08							
Byte 9	Parameter / Data	<UID Byte 1>							
Byte 10	Parameter / Data	<UID Byte 2>							
Byte 11	Parameter / Data	<UID Byte 3>							
Byte 12	Parameter / Data	<UID Byte 4>							
Byte 13	Parameter / Data	<UID Byte 5>							
Byte 14	Parameter / Data	<UID Byte 6>							
Byte 15	Parameter / Data	<UID Byte 7>							
Byte 16	Parameter / Data	<UID Byte 8>							
Byte 17	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.20

Die <FrameLength> hat einen Wert von 16#11. Dadurch wird die Länge des Fragments bis einschließlich des Bytes <UID Byte 8> angegeben. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die <TelegramLength> hat den Wert 16#0E und spezifiziert die Länge des Antworttelegramms, beginnend ab der Telegrammlänge bis einschließlich des Bytes <UID Byte 8>.

Der Parameter <Command> hat den Wert 16#40 und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00 und signalisiert, dass der Zugriff auf den Transponder erfolgreich war. Hieran schließt sich eine Längenangabe zum Fixcode. Diese hat immer eine Größe von 2 Byte und den Wert 16#0008. Danach folgt der eingelesene Fixcode zur eindeutigen Identifizierung des gelesenen Transponders.

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

Antwort Signalstärke Transponderantwort, 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	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#40							
Byte 6	Status	16#0B							
Byte 7	Parameter / Data	16#01							
Byte 8	Parameter / Data	<RSSI-Wert>							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 9	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.21

Die <FrameLength> ist konstant beim Wert 16#09. Das Fragment erstreckt sich dabei bis einschließlich des Bytes <RSSI-Wert>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die Telegrammlänge <TelegramLength> hat den Wert 16#06.

Das Byte <Command> wird zurückgespiegelt und hat den Wert 16#40 wie im Befehlstelegramm. Der <Status> für das Telegramm zur Signalisierung des RSSI-Werts ist 16#0B. Das erste Byte <Parameter / Data> hat den Wert 16#01. Das zweite Byte <Parameter / Data> beschreibt den <RSSI-Wert>.

Alle anderen Bytes des Antwortfragments sind nicht relevant und 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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#40							
Byte 6	Status	16#0F							
Byte 7	Parameter / Data	<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 8.22

Die <FrameLength> ist in der Rückantwort für das Befehlsende des "Single Read Words" konstant beim 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 Befehlsendes ist 16#0F.

Die Anzahl der identifizierten Transponder 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 Transponder 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 Write Words (EW)

Der Befehl "Enhanced Write Words" hat den Befehlscode 16#1A und führt einen permanenten Schreibvorgang auf den Nutzdatenbereich eines Transponders innerhalb des Erfassungsbereichs durch. Der erfolgreiche Schreibvorgang wird für jeden beschriebene Transponder in einem separaten Datentelegramm mit dem Statuswert 16#00 signalisiert. Das Datentelegramm enthält den Fixcode des Transponders auf dem der Nutzdatenbereich beschrieben wurde. Zusätzlich wird für jeden beschriebenen Transponder ein weiteres Telegramm mit dem Statuswert 16#0B übertragen, welches Informationen zur Signalstärke der Transponder-Antwort (RSSI-Wert) enthält. Wenn ein Transponder die Erfassungszone verlässt, wird dies über ein Telegramm mit dem Fixcode des Transponders gemeldet. Dieses Telegramm hat den Statuswert 16#05. Die Befehlsausführung wird über ein Quit-Befehl gestoppt.

Die auf einem Transponder zur Verfügung stehende Nutzdatenmenge ist abhängig vom verwendeten Chiptyp und kann unterschiedlich groß sein. Hierzu detailliertere Informationen in der Übersicht der Transpondertypen.

Der Parameter <ByteAddress> legt die Startadresse innerhalb des Nutzdatenbereichs fest. Der Wert von <ByteAddress> ist bytebezogen. Es können nur Vielfache der Speicherblockgröße parametrisiert werden. Über <Number of Bytes> wird die Anzahl der zu schreibenden Bytes festgelegt. Die Byteanzahl muss ebenfalls ein Vielfaches der Speicherblockgröße sein.

Befehl:

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	<FrameLength>							
Byte 2	Fragmentation Counter	<Fragmentation Counter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#1A							
Byte 6	Parameter / Data	<Byte Address (High Byte)>							
Byte 7	Parameter / Data	<Byte Address (Low Byte)>							
Byte 8	Parameter / Data	<Number of Bytes (High Byte)>							
Byte 9	Parameter / Data	<Number of Bytes ((Low Byte)>							
Byte 10	Parameter / Data	<Write Data Byte 1>							
Byte 11	Parameter / Data	<Write Data Byte 2>							
Byte 12	Parameter / Data	<Write Data Byte 3>							
Byte 13	Parameter / Data	<Write Data Byte 4>							
...	Parameter / Data	...							
...	Parameter / Data	<Write Data Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.23

Der Wert von <FrameLength> ist abhängig von der Anzahl der zu schreibenden Bytes. Das Fragment endet mit dem Byte <Write Data 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 Anzahl der zu schreibenden Bytes. Das Telegramm endet mit dem Byte <Write Data Byte X>.

Der Befehlscode <Command> für den Befehl "Enhanced Write Words" ist 16#1A.

Der Parameter <ByteAddress> gibt die Startadresse innerhalb des Nutzdatenbereichs an, ab der die Daten geschrieben werden. Über den Parameter <Number of Bytes> wird die Anzahl der zu schreibenden Bytes festgelegt. Anschließend folgen die auf den Transponder zu schreibenden Informationen <Write Data Byte>.

Hinweis!

Setzen Sie die nicht relevanten Bytes des Befehlsfragments auf den Wert 16#00.

Antwort Datentelegramm Daten erfolgreich geschrieben, 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	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#1A							
Byte 6	Status	16#00							
Byte 7	Parameter / Data	16#00							
Byte 8	Parameter / Data	16#08							
Byte 9	Parameter / Data	<UID Byte 1>							
Byte 10	Parameter / Data	<UID Byte 2>							
Byte 11	Parameter / Data	<UID Byte 3>							
Byte 12	Parameter / Data	<UID Byte 4>							
Byte 13	Parameter / Data	<UID Byte 5>							
Byte 14	Parameter / Data	<UID Byte 6>							
Byte 15	Parameter / Data	<UID Byte 7>							
Byte 16	Parameter / Data	<UID Byte 8>							
Byte 17	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.24

Die <FrameLength> hat einen Wert von 16#11. Dadurch wird die Länge des Fragments bis einschließlich des Bytes <UID Byte 8> angegeben. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die <TelegramLength> hat den Wert 16#0E und spezifiziert die Länge des Antworttelegramms, beginnend ab der Telegrammlänge bis einschließlich des Bytes <UID Byte 8>.

Der Parameter <Command> hat den Wert 16#1A und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00 und signalisiert, dass der Zugriff auf den Transponder erfolgreich war. Hieran schließt sich eine Längenangabe zum Fixcode. Diese hat immer eine Größe von 2 Byte und den Wert 16#0008. Danach folgt der eingelesene Fixcode zur eindeutigen Identifizierung des gelesenen Transponders.

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

Antwort Signalstärke Transponderantwort, 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	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#1A							
Byte 6	Status	16#0B							
Byte 7	Parameter / Data	16#01							
Byte 8	Parameter / Data	<RSSI-Wert>							
Byte 9	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.25

Die <FrameLength> ist konstant beim Wert 16#09. Das Fragment erstreckt sich dabei bis einschließlich des Bytes <RSSI-Wert>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die Telegrammlänge <TelegramLength> hat den Wert 16#06.

Das Byte <Command> wird zurückgespiegelt und hat den Wert 16#1A wie im Befehlstelegramm. Der <Status> für das Telegramm zur Signalisierung des RSSI-Werts ist 16#0B. Das erste Byte <Parameter / Data> hat den Wert 16#01. Das zweite Byte <Parameter / Data> beschreibt den <RSSI-Wert>.

Alle anderen Bytes des Antwortfragments sind nicht relevant und 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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#1A							
Byte 6	Status	16#05							
Byte 7	Length UID (High Byte)	16#00							
Byte 8	Length UID (Low Byte)	16#08							
Byte 9	Parameter / Data	<UID Byte 1>							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 10	Parameter / Data	<UID Byte 2>							
Byte 11	Parameter / Data	<UID Byte 3>							
Byte 12	Parameter / Data	<UID Byte 4>							
Byte 13	Parameter / Data	<UID Byte 5>							
Byte 14	Parameter / Data	<UID Byte 6>							
Byte 15	Parameter / Data	<UID Byte 7>							
Byte 16	Parameter / Data	<UID Byte 8>							
Byte 17	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.26

Die <FrameLength> hat einen Wert von 16#11. Dadurch wird die Länge des Fragments bis einschließlich des Bytes <UID Byte 8> angegeben. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die <TelegramLength> hat den Wert 16#0E und spezifiziert die Länge des Antworttelegramms, beginnend ab der Telegrammlänge bis einschließlich des Bytes <UID Byte 8>.

Der Parameter <Command> hat den Wert 16#1A und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#05 und signalisiert, dass der Transponder die Erfassungszone verlassen hat. Hieran schließt sich eine Längenangabe zum Fixcode. Diese hat immer eine Größe von 2 Byte und den Wert 16#0008.. Danach folgt der eingelesene Fixcode zur eindeutigen Identifizierung des Transponders, der den Erfassungsbe- reich verlassen hat.

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

Single Write Words with Locks (SL)

Der Befehl "Single Write Words with Lock" hat den Befehlscode 16#47 und führt einen einmaligen Schreibvorgang auf den Nutzdatenbereich eines Transponders innerhalb der Erfassungszone durch. Nach dem Schreiben werden die Daten vor einem Überschreiben geschützt, wenn der Transponder diese Funktion anbieten. Der Schreibschutz ist permanent und kann nicht rückgängig gemacht werden. Der Schreibschutz wird nur für die Speicherblöcke eingeschaltet, die jeweils beschrieben wurden. Alle anderen Speicherblöcke können weiterhin beschrieben werden.

Der erfolgreiche Schreibvorgang wird für jeden beschriebene Transponder in einem separaten Datentelegramm mit dem Statuswert 16#00 signalisiert. Das Datentelegramm enthält den Fixcode des Transponders, auf dem der Nutzdatenbereich beschrieben wurde. Zusätzlich wird für jeden beschriebenen Transponder ein weiteres Telegramm mit dem Statuswert 16#0B übertragen, welches Informationen zur Signalstärke der Transponder-Antwort (RSSI-Wert) enthält. 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 Parameter <ByteAddress> legt die Startadresse innerhalb des Nutzdatenbereichs fest. Der Wert von <ByteAddress> ist bytebezogen. Es können nur Vielfache der Speicherblockgröße parametrisiert werden. Über <Number of Bytes> wird die Anzahl der zu schreibenden Bytes festgelegt. Die Byteanzahl muss ebenfalls ein Vielfaches der Speicherblockgröße sein.

Befehl:

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	<FrameLength>							
Byte 2	Fragmentation Counter	<Fragmentation Counter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#47							
Byte 6	Parameter / Data	<Byte Address (High Byte)>							
Byte 7	Parameter / Data	<Byte Address (Low Byte)>							
Byte 8	Parameter / Data	<Number of Bytes (High Byte)>							
Byte 9	Parameter / Data	<Number of Bytes ((Low Byte)>							
Byte 10	Parameter / Data	<Write Data Byte 1>							
Byte 11	Parameter / Data	<Write Data Byte 2>							
Byte 12	Parameter / Data	<Write Data Byte 3>							
Byte 13	Parameter / Data	<Write Data Byte 4>							
...	Parameter / Data	...							
...	Parameter / Data	<Write Data Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.27

Der Wert von <FrameLength> ist abhängig von der Anzahl der zuschreibenden Bytes. Das Fragment endet mit dem Byte <Write Data 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 Anzahl der zuschreibenden Bytes. Das Telegramm endet mit dem Byte <Write Data Byte X>.

Der Befehlscode <Command> für den Befehl "Single Write Words" ist 16#47.

Der Parameter <ByteAddress> gibt die Startadresse innerhalb des Nutzdatenbereichs an, ab der die Daten geschrieben werden. Über den Parameter <Number of Bytes> wird die Anzahl der zu schreibenden Bytes festgelegt. Anschließend folgen die auf den Transponder zu schreibenden Informationen <Write Data Byte>.

**Hinweis!**

Setzen Sie die nicht relevanten Bytes des Befehlsfragments auf den Wert 16#00.

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	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 5	Command	16#47							
Byte 6	Status	16#00							
Byte 7	Parameter / Data	16#00							
Byte 8	Parameter / Data	16#08							
Byte 9	Parameter / Data	<UID Byte 1>							
Byte 10	Parameter / Data	<UID Byte 2>							
Byte 11	Parameter / Data	<UID Byte 3>							
Byte 12	Parameter / Data	<UID Byte 4>							
Byte 13	Parameter / Data	<UID Byte 5>							
Byte 14	Parameter / Data	<UID Byte 6>							
Byte 15	Parameter / Data	<UID Byte 7>							
Byte 16	Parameter / Data	<UID Byte 8>							
Byte 17	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.28

Die <FrameLength> hat einen Wert von 16#11. Dadurch wird die Länge des Fragments bis einschließlich des Bytes <UID Byte 8> angegeben. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die <TelegramLength> hat den Wert 16#0E und spezifiziert die Länge des Antworttelegramms, beginnend ab der Telegrammlänge bis einschließlich des Bytes <UID Byte 8>.

Der Parameter <Command> hat den Wert 16#47 und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00 und signalisiert, dass der Zugriff auf den Transponder erfolgreich war. Hieran schließt sich eine Längenangabe zum Fixcode. Diese hat immer eine Größe von 2 Byte und den Wert 16#0008. Danach folgt der eingelesene Fixcode zur eindeutigen Identifizierung des gelesenen Transponders.

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

Antwort Signalstärke Transponderantwort, 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	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#47							
Byte 6	Status	16#0B							
Byte 7	Parameter / Data	16#01							
Byte 8	Parameter / Data	<RSSI-Wert>							
Byte 9	Nicht relevant	16#00							
...	Nicht relevant	16#00							

2023-11

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

Tabelle 8.29

Die <FrameLength> ist konstant beim Wert 16#09. Das Fragment erstreckt sich dabei bis einschließlich des Bytes <RSSI-Wert>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die Telegrammlänge <TelegramLength> hat den Wert 16#06.

Das Byte <Command> wird zurückgespiegelt und hat den Wert 16#47 wie im Befehlstelegramm. Der <Status> für das Telegramm zur Signalisierung des RSSI-Werts ist 16#0B. Das erste Byte <Parameter / Data> hat den Wert 16#01. Das zweite Byte <Parameter / Data> beschreibt den <RSSI-Wert>.

Alle anderen Bytes des Antwortfragments sind nicht relevant und 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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#47							
Byte 6	Status	16#0F							
Byte 7	Parameter / Data	<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 8.30

Die <FrameLength> ist in der Rückantwort für das Befehlsende des "Single Read Words" konstant beim 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#47. Der <Status> für das Telegramm zur Signalisierung des Befehlsendes ist 16#0F.

Die Anzahl der identifizierten Transponder 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 Transponder 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 Write Words with Lock (EL)

Der Befehl "Enhanced Write Words with Lock" hat den Befehlscode 16#48 und führt einen permanenten Schreibvorgang auf den Nutzdatenbereich eines Transponders innerhalb der Erfassungszone durch. Nach dem Schreiben werden die Daten vor einem Überschreiben geschützt, wenn die Transponder diese Funktion anbieten. Der Schreibschutz ist permanent und kann nicht rückgängig gemacht werden. Der Schreibschutz wird nur für die Speicherblöcke eingeschaltet, die jeweils beschrieben wurden. Alle anderen Speicherblöcke können weiterhin beschrieben werden.

Der erfolgreiche Schreibvorgang wird für jeden beschriebene Transponder in einem separaten Datentelegramm mit dem Statuswert 16#00 signalisiert. Das Datentelegramm enthält den Fixcode des Transponders auf dem der Nutzdatenbereich beschrieben wurde. Zusätzlich wird für jeden beschriebenen Transponder ein weiteres Telegramm mit dem Statuswert 16#0B übertragen, welches Informationen zur Signalstärke der Transponder-Antwort (RSSI-Wert) enthält. Wenn ein Transponder die Erfassungszone verlässt, wird dies über ein Telegramm mit dem Fixcode des Transponders gemeldet. Dieses Telegramm hat den Statuswert 16#05. Die Befehlsausführung wird über ein Quit-Befehl gestoppt.

Die auf einem Transponder zur Verfügung stehende Nutzdatenmenge ist abhängig vom verwendeten Chiptyp und kann unterschiedlich groß sein. Hierzu detailliertere Informationen in der Übersicht der Transpondertypen.

Der Parameter <ByteAddress> legt die Startadresse innerhalb des Nutzdatenbereichs fest. Der Wert von <ByteAddress> ist bytebezogen. Es können nur Vielfache der Speicherblockgröße parametrisiert werden. Über <Number of Bytes> wird die Anzahl der zu schreibenden Bytes festgelegt. Die Byteanzahl muss ebenfalls ein Vielfaches der Speicherblockgröße sein.

Befehl:

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	<FrameLength>							
Byte 2	Fragmentation Counter	<FragmentationCounter>							
Byte 3	Telegram Length (High Byte)	<TelegramLength (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<TelegramLength (Low Byte)>							
Byte 5	Command	16#48							
Byte 6	Parameter / Data	<Byte Address (High Byte)>							
Byte 7	Parameter / Data	<Byte Address (Low Byte)>							
Byte 8	Parameter / Data	<Number of Bytes (High Byte)>							
Byte 9	Parameter / Data	<Number of Bytes ((Low Byte)>							
Byte 10	Parameter / Data	<Write Data Byte 1>							
Byte 11	Parameter / Data	<Write Data Byte 2>							
Byte 12	Parameter / Data	<Write Data Byte 3>							
Byte 13	Parameter / Data	<Write Data Byte 4>							
...	Parameter / Data	...							
...	Parameter / Data	<Write Data Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.31

Der Wert von <FrameLength> ist abhängig von der Anzahl der zu schreibenden Bytes. Das Fragment endet mit dem Byte <Write Data 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 Anzahl der zu schreibenden Bytes. Das Telegramm endet mit dem Byte <Write Data Byte X>.

Der Befehlscode <Command> für den Befehl "Enhanced Write Words" ist 16#48.

Der Parameter <ByteAddress> gibt die Startadresse innerhalb des Nutzdatenbereichs an, ab der die Daten geschrieben werden. Über den Parameter <Number of Bytes> wird die Anzahl der zu schreibenden Bytes festgelegt. Anschließend folgen die auf den Transponder zu schreibenden Informationen <Write Data Byte>.



Hinweis!

Setzen Sie die nicht relevanten Bytes des Befehlsfragments auf den Wert 16#00.

Antwort Datentelegramm, Daten erfolgreich geschrieben, 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	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#48							
Byte 6	Status	16#00							
Byte 7	Length UID (High Byte)	16#00							
Byte 8	Length UID (Low Byte)	16#08							
Byte 9	Parameter / Data	<UID Byte 1>							
Byte 10	Parameter / Data	<UID Byte 2>							
Byte 11	Parameter / Data	<UID Byte 3>							
Byte 12	Parameter / Data	<UID Byte 4>							
Byte 13	Parameter / Data	<UID Byte 5>							
Byte 14	Parameter / Data	<UID Byte 6>							
Byte 15	Parameter / Data	<UID Byte 7>							
Byte 16	Parameter / Data	<UID Byte 8>							
Byte 17	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.32

Die <FrameLength> hat einen Wert von 16#11. Dadurch wird die Länge des Fragments bis einschließlich des Bytes <UID Byte 8> angegeben. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die <TelegramLength> hat den Wert 16#0E und spezifiziert die Länge des Antworttelegramms, beginnend ab der Telegrammlänge bis einschließlich des Bytes <UID Byte 8>.

Der Parameter <Command> hat den Wert 16#48 und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#00 und signalisiert, dass der Zugriff auf den Transponder erfolgreich war. Hieran schließt sich eine Längenangabe zum Fixcode. Diese hat immer eine Größe von 2 Byte und den Wert 16#0008. Danach folgt der eingelesene Fixcode zur eindeutigen Identifizierung des gelesenen Transponders.

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

Antwort Signalstärke Transponderantwort, 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	16#0			
Byte 1	Frame Length	16#09							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#06							
Byte 5	Command	16#48							
Byte 6	Status	16#0B							
Byte 7	Parameter / Data	16#01							
Byte 8	Parameter / Data	<RSSI-Wert>							
Byte 9	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.33

Die <FrameLength> ist konstant beim Wert 16#09. Das Fragment erstreckt sich dabei bis einschließlich des Bytes <RSSI-Wert>. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die Telegrammlänge <TelegramLength> hat den Wert 16#06.

Das Byte <Command> wird zurückgespiegelt und hat den Wert 16#48 wie im Befehlstelegramm. Der <Status> für das Telegramm zur Signalisierung des RSSI-Werts ist 16#0B. Das erste Byte <Parameter / Data> hat den Wert 16#01. Das zweite Byte <Parameter / Data> beschreibt den <RSSI-Wert>.

Alle anderen Bytes des Antwortfragments sind nicht relevant und 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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#11							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0E							
Byte 5	Command	16#48							
Byte 6	Status	16#05							
Byte 7	<Length UID (High Byte)>	16#00							
Byte 8	<Length UID (Low Byte)>	16#08							
Byte 9	Parameter / Data	<UID Byte 1>							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 10	Parameter / Data	<UID Byte 2>							
Byte 11	Parameter / Data	<UID Byte 3>							
Byte 12	Parameter / Data	<UID Byte 4>							
Byte 13	Parameter / Data	<UID Byte 5>							
Byte 14	Parameter / Data	<UID Byte 6>							
Byte 15	Parameter / Data	<UID Byte 7>							
Byte 16	Parameter / Data	<UID Byte 8>							
Byte 17	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.34

Die <FrameLength> hat einen Wert von 16#11. Dadurch wird die Länge des Fragments bis einschließlich des Bytes <UID Byte 8> angegeben. Der <FragmentationCounter> hat den Wert 16#00, da alle Antwortdaten innerhalb eines Fragments übertragen werden können. Die <TelegramLength> hat den Wert 16#0E und spezifiziert die Länge des Antworttelegramms, beginnend ab der Telegrammlänge bis einschließlich des Bytes <UID Byte 8>.

Der Parameter <Command> hat den Wert 16#48 und wird innerhalb des Datentelegramms zurückgespiegelt. Der Parameter <Status> hat den Wert 16#05 und signalisiert, dass der Transponder die Erfassungszone verlassen hat. Hieran schließt sich eine Längenangabe zum Fixcode. Diese hat immer eine Größe von 2 Byte und den Wert 16#0008. Danach folgt der eingelesene Fixcode zur eindeutigen Identifizierung des Transponders, der den Erfassungsbe- reich verlassen hat.

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

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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#06							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#03							
Byte 5	Command	16#02							
Byte 6	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.35

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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#07							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#04							
Byte 5	Command	16#02							
Byte 6	Status	16#00							
Byte 7	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.36

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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#06							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#03							
Byte 5	Command	16#03							
Byte 6	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.37

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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	<Frame Length>							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	<Telegram Length (High Byte)>							
Byte 4	Telegram Length (Low Byte)	<Telegram Length (Low Byte)>							
Byte 5	Command	16#03							
Byte 6	Status	16#00							
Byte 7	Version Byte 1	16#XX							
...	...	16#XX							
Byte X	Version Byte X	16#XX							
Byte 31	Nicht relevant	16#00							

Tabelle 8.38

Change Tag (CT)

Der Befehl "CT" hat den Befehlscode 16#04 und stellt den Transpondertyp ein, mit dem das Schreib-/Lesegerät die Leseinheit kommuniziert. Der Auslieferungsstatus ist Typ "20".

Befehl:

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#07							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#04							
Byte 5	Command	16#04							
Byte 6	TagType	16#14							
Byte 7	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.39

Antwort Befehlsende:

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#07							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#04							
Byte 5	Command	16#04							
Byte 6	Status	16#00							
Byte 7	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.40

Der Transpondertyp kann auch über einen Parameter eingestellt werden, siehe "Transpondertyp "Change Tag (CT)"" auf Seite 79.

8.6.3 HF-Konfigurationbefehle

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

Alle Parameterwerte außer dem Parameter "TI" werden nicht-flüchtig gespeichert und bleiben nach einer Spannungsunterbrechung unverändert bestehen. Die Antwort auf einen Konfigurationsbefehl ist eine Statusmeldung vom Schreib-/Lesegerät. Beim Lesen kommt als Antwort eine Statusmeldung und die entsprechenden Daten.

Für den Zugriff auf die HF-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 "Q" (16#51).

8.6.3.1 Grundlegende Befehlsstruktur

Read Parameter (RP)

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

Befehl:

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	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#51 "Q"							
Byte 7	Parameter Name (High Byte)	<ParameterName (High Byte)>							
Byte 8	Parameter Name (Low Byte)	<ParameterName (Low Byte)>							

2023-11

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
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 8.41

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#51 "Q".

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.



Hinweis!

Setzen Sie die nicht relevanten Bytes des Befehlsfragments auf den Wert 16#00.

Antwort:

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#BE							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
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 8.42

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.

Write Parameter (WP)

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

Befehl:

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	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#51 "Q"							
Byte 7	Parameter Name (High Byte)	<ParameterName (High Byte)>							
Byte 8	Parameter Name (Low Byte)	<ParameterName (Low Byte)>							
Byte 9	Length Parameter (High Byte)	<LengthParameter (High Byte)>							
Byte 10	Length Parameter (Low Byte)	<LengthParameter (Low Byte)>							

2023-11

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 11	Parameter Data Byte 1	<Parameter Data Byte 1>							
Byte 12	Parameter Data Byte 2	<Parameter Data Byte 2>							
...							
...	Parameter Data Byte X	<Parameter Data Byte X>							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.43

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#51 ("Q").

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.



Hinweis!

Setzen Sie die nicht relevanten Bytes des Befehlsfragments auf den Wert 16#00.

Antwort:

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#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 8.44

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.

8.6.3.2 Parameterübersicht

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

Parameterkürzel	Parametername/Funktion	Parameter lesbar / schreibbar
CT 16#4354	Siehe "Transpondertyp "Change Tag (CT)"" auf Seite 79 Einstellung des Transpondertyps	lesbar / schreibbar
DR 16#4452	Siehe "Übertragungsrate "Data Rate (DR)"" auf Seite 80 Einstellung der Übertragungsrate	lesbar / schreibbar
E5 16#4535	Siehe "Anzahl erfolgloser Versuche bis Status 5 "Enhanced Status 5 (E5)"" auf Seite 82 Einstellung der Anzahl erfolgloser Versuche bis Status 5	lesbar / schreibbar
NT 16#4E54	Siehe "Number of Tags NT (Abbruchkriterium)" auf Seite 84 Einstellung des Abbruchkriteriums der Schreib-/Lesebefehle	lesbar / schreibbar
OH 16#4F48	Siehe "Betriebsstunden "Operating Hours (OH)"" auf Seite 86 Auslesen der Betriebsstunden	lesbar
PT 16#5054	Siehe "Sendeleistung "Power Transmit (PT)"" auf Seite 87 Einstellung der Sendeleistung	lesbar / schreibbar
QW 16#5157	Siehe "Q-Wert QW" auf Seite 89 Einstellung der Zeitschlitz für den Zugriff auf Datenträger	lesbar / schreibbar
RD 16#5244	Siehe "Zurücksetzen auf Werkseinstellung "Reset to Default (RD)"" auf Seite 91 Wiederherstellung Default-Zustand	schreibbar
ST 16#5354	Siehe "Statusabfrage "Status Frontend (ST)"" auf Seite 92 Ausgabe des Status des Frontends	lesbar
TA 16#5441	Siehe "Anzahl der Zugriffsversuche "Tries Allowed (TA)"" auf Seite 93 Einstellung Anzahl Versuche	lesbar / schreibbar
TE 16#5445	Siehe "Statusabfrage "Temperatursausgabe (TE)"" auf Seite 94 Ausgabe der internen Temperatur	lesbar
TI 16#5449	Siehe "Setzen eines Filters "Tag ID Filter (TI)"" auf Seite 95 Setzen eines Filters auf die Tag-ID	lesbar / schreibbar
TO 16#544F	Siehe "Verhalten bei Übertemperatur "Over Temperature Handling (TO)"" auf Seite 97 Einstellung des Verhaltens bei Übertemperatur	lesbar / schreibbar

Tabelle 8.45

Transpondertyp "Change Tag (CT)"

Der Parameter CT stellt den Transpondertyp ein, mit dem das Schreib-/Lesegerät kommuniziert.

Parameterzeichen	CT (16#4354)
Länge Parameterwert CT	1 Byte
Werkseinstellung	16#14
Wertebereich	16#14 ... 16#26; 16#32

Für unterstützte Datenträger siehe Kapitel 6.1.

Beispiel: Befehlstelegramm für die Änderung des Transpondertyps auf den Wert 16#15 (IQC21)

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#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#51 "Q"							
Byte 7	ParameterName (High Byte)	16#43 "C"							
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 CT	16#15							
Byte 12	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.46

Beispiel: Befehlstelegramm zum Auslesen des Transpondertyps

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#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#51 "Q"							
Byte 7	ParameterName (High Byte)	16#43 "C"							
Byte 8	ParameterName (Low Byte)	16#54 "T"							
Byte 9	LengthParameter (High Byte)	16#00							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 10	LengthParameter (Low Byte)	16#00							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.47

Beispiel: Antworttelegramm mit dem eingestellten Transpondertyp = 20 (16#14)

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#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#05							
Byte 5	Command	16#BE							
Byte 6	Status	16#00							
Byte 7	Parameter CT	16#14							
Byte 8	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.48



Hinweis!

Der Transpondertyp kann auch über den Systembefehl CT umgestellt werden, siehe "Change Tag (CT)" auf Seite 73.

Übertragungsrate "Data Rate (DR)"

Der Parameter DR stellt die Übertragungsrate zwischen Schreib-/Lesegerät und Datenträger ein.

Parameterzeichen	DR (16#4452)
Länge Parameterwert DR	1 Byte
Werkseinstellung	16#00
Wertebereich	0 (16#00) = normal [26 kBit/s] 1 (16#01) = fast read mode gem. ISO/IEC 15693-2 X2 [53 kBit/s]

Die Übertragungsrate "fast read mode" wird unterstützt von

- Pepperl+Fuchs-Transpondertypen IQC21,-33,-37 und -50, siehe Kapitel 6.1
- Transponder mit Zugriff auf mindestens 2 Speicherblöcke

Beispiel: Befehlstelegramm für die Änderung der Übertragungsrate auf den Wert 16#01 (fast read mode)

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#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#51 "Q"							
Byte 7	ParameterName (High Byte)	16#44 "D"							
Byte 8	ParameterName (Low Byte)	16#52 "R"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#01							
Byte 11	Parameter DR	16#01							
Byte 12	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.49

Beispiel: Befehlstelegramm zum Auslesen der Übertragungsrate

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#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#51 "Q"							
Byte 7	ParameterName (High Byte)	16#44 "D"							
Byte 8	ParameterName (Low Byte)	16#52 "R"							
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 8.50

Beispiel: Antworttelegramm mit der eingestellten Übertragungsrate 16#00 (normal)

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#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 DR	16#00							
Byte 8	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.51

Anzahl erfolgloser Versuche bis Status 5 "Enhanced Status 5 (E5)"

Der Parameter E5 stellt bei der Ausführung eines Enhanced-Befehls die Anzahl der erfolgreichen Schreib-/Leseversuche ein, bevor das Gerät das Telegramm mit dem Statuswert 16#05 ausgibt. Über das Telegramm mit dem Status 16#05 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 Schreib-/Leseversuche
Wertebereich	16#00 ... 16#0A = 0 ... 10 erfolglose 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 Telegramme mit Status 16#05. Bei dynamischen Anwendungen lassen sich über den Parameter Lücken im Erfassungsbereich ausblenden, ohne dass bei kleinen Unterbrechungen der Datenträgerkommunikation eine Status-16#05-Meldung empfangen wird. Die Erfassungszone wirkt homogener.

Wenn in einer dynamischen Anwendung viele Datenträger gleichzeitig erfasst werden, kann durch eine Vergrößerung des Parameterwerts von E5 der Empfang der Telegramme mit Status 16#05 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-Telegramme werden zeitlich versetzt übermittelt.

Durch eine Verringerung des Parameterwertes 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

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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#0C							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#09							
Byte 5	Command	16#BF							
Byte 6	SystemCode	16#51 "Q"							
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 8.52

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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#BE							
Byte 6	SystemCode	16#51 "Q"							
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 8.53

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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#08							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#05							
Byte 5	Command	16#BE							
Byte 6	Status	16#00							
Byte 7	Parameter E5	16#05							
Byte 8	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.54

Number of Tags NT (Abbruchkriterium)

Der Parameter NT gibt die Anzahl der Transponder im Erfassungsbereich an, die das Gerät sucht. Jeder Befehl wird entsprechend des Parameters 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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#0C							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#09							
Byte 5	Command	16#BF							
Byte 6	SystemCode	16#51 "Q"							
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#01							

2023-11

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 11	Parameter NT	16#01							
Byte 12	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.55

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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#BE							
Byte 6	SystemCode	16#51 "Q"							
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 8.56

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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#08							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#05							
Byte 5	Command	16#BE							
Byte 6	Status	16#00							
Byte 7	Parameter NT	16#FF							
Byte 8	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.57

2023-11

Betriebsstunden "Operating Hours (OH)"

Der Parameter OH gibt die Information aus, wie lange das Schreib-/Lesegerät im Betrieb ist und wie lange ein Schreib-/Lesebefehl läuft.

Parameterzeichen	OH(16#4F48)
Länge Parameterwert OH	8 Byte
	<ul style="list-style-type: none"> • Byte 1-4: Betriebszeit Schreib-/Lesegerät in Stunden¹ • Byte 5-8: Betriebszeit Schreib-/Lesebefehle in Stunden²

1. IO-Link-Kommunikation aktiv

2. Schreib-/Lesebefehl aktiv

Beispiel: Befehlstelegramm zum Auslesen von OH

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#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#BF							
Byte 6	SystemCode	16#51 "Q"							
Byte 7	ParameterName (High Byte)	16#4F "O"							
Byte 8	ParameterName (Low Byte)	16#48 "H"							
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 8.58

Beispiel: Antworttelegramm mit den Betriebsstunden

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#0E							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0C							
Byte 5	Command	16#BE							
Byte 6	Status	16#00							
Byte 7	Parameter OH (Byte 1)	16#00							
Byte 8	Parameter OH (Byte 2)	16#00							
Byte 9	Parameter OH (Byte 3)	16#00							
Byte 10	Parameter OH (Byte 4)	16#53 = 83 _{dez} Stunden							

2023-11

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 11	Parameter OH (Byte 5)	16#00							
Byte 12	Parameter OH (Byte 6)	16#00							
Byte 13	Parameter OH (Byte 7)	16#00							
Byte 14	Parameter OH (Byte 8)	16#03 = 3 _{dez} Stunden							
Byte 15	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.59

Die Betriebszeit des Schreib-/Lesegerät beträgt 83_{dez} Stunden (16#00000053). Die Dauer der Schreib-/Lesebefehle beträgt 3_{dez} Stunden (16#00000003).

Sendeleistung "Power Transmit (PT)"

Der Parameter PT setzt die Sendeleistung oder liest die gesetzte Sendeleistung aus. Mit der Sendeleistung können Sie die Reichweite und die maximale zulässige Umgebungstemperatur beeinflussen, bei welcher das Gerät eingesetzt werden kann.

Parameterzeichen	PT (16#5054)
Länge Parameterwert CT	1 Byte
Werkseinstellung	16#0004
Wertebereich	16#0001 (Min) 16#0002 (Eco) 16#0003 (Normal) 16#0004 (Maximum)

Durch eine Verminderung der Sendeleistung lässt sich das Gerät dauerhaft bei höheren Umgebungstemperaturen betreiben. Die Eigenerwärmung des Geräts wird verringert. Wenn das Gerät im Dauerbetrieb bei "enhanced"-Schreib-/Lesebefehle mit maximaler Sendeleistung betrieben wird, beträgt die Umgebungstemperatur maximal 55 °C. Im Eco-Modus lässt sich die Umgebungstemperatur auf maximal 70 °C steigern. Neben der Einstellung der Sendeleistung verfügt das Schreib-/Lesegerät über weitere Schutzmechanismen, die ein Überhitzen des Geräts verhindern. Siehe "Verhalten bei Übertemperatur "Over Temperature Handling (TO)"" auf Seite 97.

Wert	Bedeutung	Reichweite ¹	Umgebungstemperatur
1	Min	90 mm	- 25 ... 70 °C [-13 ... 158 °F]
2	Eco	240 mm	- 25 ... 70 °C [-13 ... 158 °F]
3	Normal	270 mm	- 25 ... 65 °C [-13 ... 149 °F]
4	Maximum	300 mm	- 25 ... 55 °C [-13 ... 131 °F]

Tabelle 8.60

1. Referenztag: IQC21-50

Beispiel: Befehlstelegramm für die Änderung der Sendeleistung PT auf den Wert 3 (16#0003)

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#BF							
Byte 6	SystemCode	16#51 "Q"							
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 PT (High Byte)	16#00							
Byte 12	Parameter PT (Low Byte)	16#03							
Byte 13	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#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#51 "Q"							
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 PT (16#0004 = Maximum)

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#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 PT (High Byte)	16#00							
Byte 8	Parameter PT (Low Byte)	16#04							
Byte 9	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.63

Q-Wert QW

Mit dem Parameter QW wird die Anzahl der Transponder definiert, die im Erfassungsbereich erwartet werden. Diese Einstellung wirkt sich auf den Zeitschlitz bei der Kommunikation mit einem Transponder aus, um Kollisionen durch die gleichzeitige Kommunikation mehrerer Transponder zu vermeiden.

Parameterzeichen	QW (16#5157)
Länge Parameterwert QW	1 Byte
Werkseinstellung	16#00
Wertebereich	16#00 = ein Transponder 16#01 = zwei Transponder 16#02 = mehrere Transponder (ca. 4) 16#03 = viele Transponder (ca. 8) 16#04 = sehr viele Transponder (ca. 16)

Der Q-Wert hat Einfluss auf die Ausführungszeit von Schreib-/Lesebefehlen. Durch eine Verringerung des Q-Werts 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 Transponder

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#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#51 "Q"							
Byte 7	ParameterName (High Byte)	16#51 "Q"							

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
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 8.64

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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#BE							
Byte 6	SystemCode	16#51 "Q"							
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 8.65

Beispiel: Antworttelegramm mit dem eingestellten Wert des Parameters QW (16#02; mehrere Transponder, ca. 4)

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#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							

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

Zurücksetzen auf Werkseinstellung "Reset to Default (RD)"

Der Parameter RD setzt das Gerät auf die Werkseinstellung zurück. Der Parameter RD wird dabei geschrieben. Ein Lesezugriff auf diesen Parameter ist nicht möglich.

Parameterzeichen RD (16#5244)

Tabelle 8.67

Beispiel: Befehlstelegramm zum Zurücksetzen auf die Werkseinstellung

Byte	Inhalt	Bit-Nummer							
		7	6	5	4	3	2	1	0
Byte 0	ControlByte / Frame Length	D _S	U _M	U _S	0	16#0			
Byte 1	Frame Length	16#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#BE							
Byte 6	SystemCode	16#51 "Q"							
Byte 7	ParameterName (High Byte)	16#52 "R"							
Byte 8	ParameterName (Low Byte)	16#44 "D"							
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 8.68

Werkseinstellung

Kürzel	Parameter	Voreingestellter Wert
CT	Tagtype CT	Auto detect (16#14)
DR	Data Rate	normal (16#00)
E5	Anzahl erfolgloser Versuche bis Status 5	16#05
NT	Abbruchkriterium Suchalgorithmus	aus (16#FF)
PT	Sendeleistung "power transmit"	Maximum (16#0004)
QW	Q-Wert	Single tag (16#00)
TA	Anzahl Versuche "tries allowed"	16#02
TO	Verhalten bei Übertemperatur "over temperature handling"	Switch off (16#00)

Tabelle 8.69

Statusabfrage "Status Frontend (ST)"

Der Parameter ST liest den Betriebsstatus des Schreib-/Lesegeräts aus.

Parameterzeichen	ST (16#5354)
Länge Parameterwert ST	2 Byte
Wertebereich	16#01 = Störung 16#02 = Verstimmung durch umgebendes Metall 16#04 = Übertemperatur Warnung, interne Temperatur zwischen 80 - 85 °C 16#08 = Übertemperatur Fehler, interne Temperatur > 85 °C

Beispiel: Befehlstelegramm zum Auslesen des Status

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#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#BF							
Byte 6	SystemCode	16#51 "Q"							
Byte 7	ParameterName (High Byte)	16#53 "S"							
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 8.70

Beispiel: Antworttelegramm mit Betriebsstatus ST

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#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 ST (High Byte)	16#00 ¹							
Byte 8	Parameter ST (Low Byte)	16#02 ²							
Byte 9	Nicht relevant	16#00							
...	Nicht relevant	16#00							

2023-11

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

Tabelle 8.71

1. Aktueller Status = keine Meldung

2. Gespeicherter Status seit dem letzten Auslesen = Verstimmung der Spule durch Metall

Anzahl der Zugriffsversuche "Tries Allowed (TA)"

Der Parameter Tries Allowed (TA) stellt die Anzahl der Zugriffsversuche während der Ausführung einer Schreib-/Leseoperation auf einen Datenträger ein.

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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#0C							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#09							
Byte 5	Command	16#BF							
Byte 6	SystemCode	16#51 "Q"							
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 31	Nicht relevant	16#00							

Tabelle 8.72

Beispiel: Befehlstelegramm zum Auslesen von TA

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#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#51 "Q"							
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 8.73

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 _S	U _M	U _D	0	16#0			
Byte 1	Frame Length	16#08							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#05							
Byte 5	Command	16#BE							
Byte 6	Status	16#00							
Byte 7	Parameter TA	16#02							
Byte 8	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.74

Statusabfrage "Temperaturlausgabe (TE)"

Der Parameter TE gibt die innere Temperatur des Schreib-/Lesegeräts aus.

Parameterzeichen TE (16#5445)

Länge Parameterwert TE 2 Byte

- Byte 1: Temperatur am Leistungsverstärker
- Byte 2: Temperatur am Mikrocontroller

Beispiel: Befehlstelegramm zum Auslesen der Temperatur

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#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#BF							
Byte 6	SystemCode	16#51 "Q"							
Byte 7	ParameterName (High Byte)	16#54 "T"							
Byte 8	ParameterName (Low Byte)	16#45 "E"							
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 8.75

Beispiel: Antworttelegramm mit der Temperatur TE

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#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 TE	16#20 ¹							
Byte 8	Parameter TE	16#1E ²							
Byte 9	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.76

1.32_{dez} °C2.30_{dez} °C**Setzen eines Filters "Tag ID Filter (TI)"**

Der Parameter TI setzt einen Filter auf die UID eines Datenträger. Die Schreib-/Lesebefehle werden so nur für Datenträger mit der UID ausgeführt, die dem gesetzten Filter entspricht. Der Filter kann auf die gesamte UID (8 Byte) oder die führenden Bytes gesetzt werden.

Parameterzeichen	TI (16#5449)
Wertebereich	0 ... 8 Byte UID

**Hinweis!**

Der Parameter TI wird flüchtig gespeichert.

Beispiel: Befehlstelegramm für die Änderung des Filter (TI) auf UIDs, die mit dem Wert 16#E0040108 beginnen

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#0F							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#0C							
Byte 5	Command	16#BF							
Byte 6	SystemCode	16#51 "Q"							
Byte 7	ParameterName (High Byte)	16#54 "T"							
Byte 8	ParameterName (Low Byte)	16#49 "I"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#04							
Byte 11	Parameter TI	16#E0							
Byte 12	Parameter TI	16#04							
Byte 13	Parameter TI	16#01							
Byte 14	Parameter TI	16#08							
Byte 15	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.77

Beispiel: Befehlstelegramm zum Auslesen des aktuellen Filters TI

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#0B							
Byte 2	Fragmentation Counter	16#00							
Byte 3	Telegram Length (High Byte)	16#00							
Byte 4	Telegram Length (Low Byte)	16#08							
Byte 5	Command	16#BF							
Byte 6	SystemCode	16#51 "Q"							
Byte 7	ParameterName (High Byte)	16#54 "T"							
Byte 8	ParameterName (Low Byte)	16#49 "I"							

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 8.78

Beispiel: Antworttelegramm mit dem gesetzten Filter 16#E0080148 = UID beginnend mit den 4 Bytes E0.08.01.48

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#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 T1	16#E0							
Byte 8	Parameter T1	16#08							
Byte 9	Parameter T1	16#01							
Byte 10	Parameter T1	16#48							
Byte 11	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.79

Verhalten bei Übertemperatur "Over Temperature Handling (TO)"

Der Parameter TO definiert das Verhalten des Schreib-/Lesegeräts bei Übertemperatur. Mit der Einstellung des Parameters kann das Gerät vor Überhitzung geschützt werden.

Parameterzeichen	TO (16#544F)
Länge Parameterwert TO	1 Byte
Werkseinstellung	16#00
Wertebereich	16#00 = Switch off 16#01 = Reduce power 16#02 = Reduce duty cycle

Wert	Modus	Funktion
0	Switch off	Das Gerät schaltet sich automatisch ab, wenn eine interne Temperatur > 85 °C gemessen wird. Wenn die interne Temperatur unter 80 °C fällt, schaltet sich das Gerät automatisch ein.
1	Reduce power	Das Gerät reduziert automatisch die Sendeleistung, wenn eine interne Temperatur zwischen 80 – 85 °C gemessen wird. Das Gerät schaltet sich automatisch ab, wenn eine interne Temperatur > 85 °C gemessen wird.
2	Reduce duty cycle	Das Gerät pausiert nach einem Übertragungszyklus für 0,5 ms – 1s, wenn die interne Temperatur zwischen 80 – 85 °C gemessen wird. Die Länge der Pausenzeit ist abhängig von der interne Temperatur und der Dauer des letzten Schreib-/Lesebefehls. Das Gerät schaltet sich automatisch ab, wenn eine interne Temperatur > 85 °C gemessen wird.

Hinweis!



Beachten Sie im Modus "Reduce duty cycle", dass das Schreib-/Lesegerät während der Pausenzeit keine Datenträger identifiziert.

Berücksichtigen Sie diese Pause bei Anwendungen mit Datenträgern, die sich dynamisch durch den Erfassungsbereich bewegen. Verwenden Sie ggf. einen anderen Schutzmechanismus.

Beispiel: Befehlstelegramm für die Änderung von TO auf den Wert 16#02 (Reduce duty cycle)

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#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#51 "Q"							
Byte 7	ParameterName (High Byte)	16#54 "T"							
Byte 8	ParameterName (Low Byte)	16#4F "O"							
Byte 9	LengthParameter (High Byte)	16#00							
Byte 10	LengthParameter (Low Byte)	16#01							
Byte 11	Parameter TO	16#02							
Byte 12	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.80

Beispiel: Befehlstelegramm zum Auslesen von TO

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#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#51 "Q"							
Byte 7	ParameterName (High Byte)	16#54 "T"							
Byte 8	ParameterName (Low Byte)	16#4F "O"							
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 8.81

Beispiel: Antworttelegramm mit dem eingestellten Wert 16#00 von TO

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#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 TO	16#00							
Byte 8	Nicht relevant	16#00							
...	Nicht relevant	16#00							
Byte 31	Nicht relevant	16#00							

Tabelle 8.82

8.7 Fehler-/Statusmeldungen

Status	Bedeutung
16#00	Der Befehl wurde fehlerfrei ausgeführt.
16#01	Übertemperatur
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.
16#05	Datenträger hat die Erfassungszone verlassen.
16#06	Hardwarefehler, z. B. Fehler bei Selbsttest oder Gerät defekt.
16#07	interner Gerätefehler
16#09	Der parametrisierte Datenträgertyp passt nicht zum angeschlossenen Schreib-/Lesegerät.
16#0A	Short-Form-Datenformat: mehrere Transponder im Erfassungsbe- reich. Long-Form-Datenformat: mehrere Transponder im Erfassungsbereich mit gleichem UID.
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	Kennzeichnet das Ende einer Ausgabe im Multiframe-Protokoll.

9 Anhang

9.1 Fehlerbehebung

Index	Beschreibung	Lösung
1	Keine blaue LED an; es blinkt nur die grüne LED	Ist die Autostart-Funktion aktiv?
2	Keine orangefarbene LED, wenn Datenträger innerhalb des Erfassungsbereichs	Ist der passender Datenträgertyp eingestellt? Lesen Sie den IO-Link Parameter "Tag Type" (Index 106) aus und vergleichen Sie das Ergebnis mit der Datenträgerliste. Passt die Anzahl der Bytes zu der Blocklänge des Datenträgers?
3	Keine grün blinkende LED	Ist die IO-Link-Konfiguration des Masters korrekt? Wird die IO-Link-Version V1.1 unterstützt? Ist das Anschlusskabel korrekt verbunden?
4	Innerhalb Byte 0 des Eingangsdatenfeldes steht 0x40	Dieser Wert signalisiert, dass der Expert-Mode aktiviert ist. In diesem Fall ist der Wert des IO-Link Parameters "Operation Mode" 0x00 (Expert Mode). Ändern Sie den IO-Link Parameter "Operation Mode" auf den Wert 0x80 (Easy Mode). Das höherwertige Nibble hat im Easy-Mode immer den Wert 0x00.
5	Der Datenträger IQC33 kann nicht gelesen bzw. beschrieben werden	Ist die Anzahl der Bytes innerhalb des IO-Link-Parameters "Leseauftrag" bzw. "Schreibauftrag" ein Vielfaches von 8?
6	Der Datenträger IQC37 kann nicht gelesen bzw. beschrieben werden	Der IQC37 wird nur unter Verwendung des Expert-Mode unterstützt.
7	Keine Änderung der eingelesenen Daten trotz Änderung der Anfangsadresse	Die Adresse wird byteweise gezählt. Beachten Sie die richtige Blocklänge des Datenträgertyps: 4 bzw 8 Byte. Die Daten werden geändert, wenn die Adresse um die Blocklänge erhöht wird.
8	Nach der Einstellung des Datenträgertyps IQC33 wird "invalid command" angezeigt	Stellen Sie die Anzahl der Bytes auf 8 oder ein Vielfaches davon.
9	Ein unbekannter Datenträger kann nicht gelesen werden	Setzen Sie das RFID-Schreib-/Lesegerät auf Werkseinstellung zurück. Stellen Sie die Ausführung des Leseauftrages auf den Fixcode um: Ändern Sie den IO-Link-Parameters "Leseauftrag". Kann der Fixcode gelesen werden? Ist der Fixcode lesbar, so kann auf den Datenträger zugegriffen werden. Eventuell müssen Sie den Datenträgertyp ändern; ist der Zugriff nicht möglich, so ist der Datenträger nicht kompatibel zur ISO15693.

9.2 ASCII-Tabelle

hex	dez	ASCII	hex	dez	ASCII	hex	dez	ASCII	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

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

