Technische Information

SICK
RFU6xx Funktionsbaustein

Bausteinversion V2.X

SICK RFU6XX PNDP Funktionsbaustein für
Siemens Step7 Steuerungen

SICK
Sensor Intelligence.

Versionshistorie

Version	Datum	Beschreibung
V2.0	09.10.2012	Initiale Version
V2.1	26.11.2012	Fehler Free Command und Read Result Flag behoben
V2.1	12.11.2013	Dokumentation überarbeitet (Antennenauswahl, Anzahl Retries, Fehlercode W#16#001B)
V2.2	19.07.2016	Unterstützung von Multi-Instanz Offsets > 4095

Inhaltsverzeichnis

1 Zu diesem Dokument

Bitte lesen Sie dieses Kapitel sorgfältig, bevor Sie mit dieser Betriebsanleitung und den SICK RFU6xx Funktionsbaustein arbeiten.

1.1 Funktion dieses Dokuments

Diese Betriebsanleitung beschreibt den Umgang mit dem SICK RFU6XX PNDP Funktionsbaustein. Sie leitet das technische Personal des Maschinenherstellers bzw. Maschinenbetreibers zur Projektierung und Inbetriebnahme des Funktionsbausteins an.

1.2 Zielgruppe

Diese Betriebsanleitung richtet sich an fachkundiges Personal wie z.B. Techniker oder Ingenieure.

2 Allgemeines

Der Funktionsbaustein „SICK RFU6XX PNDP“ wird zur Kommunikation zwischen einer SIMATIC Steuerung und einem SICK RFU6xx RFID Sensor verwendet.

Die folgende Abbildung zeigt die Darstellung des Funktionsbausteins in der Funktionsplan-Ansicht (FUP).

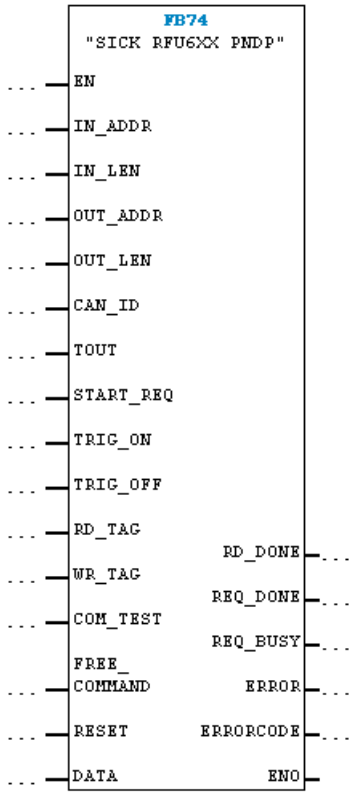

Abbildung 1: SICK RFU6XX PNDP Funktionsbaustein

Bausteinfunktionalitäten:

- Senden eines Triggerbefehls (CoLa[i] Kommando) über die SPS
- Empfang von Leseergebnissen (im SOPAS-ET[ii] Ausgabeformat definiert)
- Lesen und Schreiben von Transponder-Inhalten
- Ausführen eines Kommunikationstests
- Kommunikation über frei wählbare CoLa Kommandos (CoLa-A Protokoll)
- Ansprechen von Geräten, die untereinander via CAN-Bus kommunizieren

[i] Die Command Language (CoLa) ist ein internes SICK Protokoll zur Kommunikation mit SOPAS-Geräten
[ii] SOPAS-ET ist ein Engineering Tool zum parametrieren von SICK Sensoren

3 Hardwarekonfiguration

3.1 Unterstützte SPS-Steuerungen

Der Funktionsbaustein darf nur mit Simatic S7-Steuerungen der 300er und der 400er Familie betrieben werden. Es werden nur Steuerungen unterstützt, die die verwendete Feldbusschnittstelle integriert haben. Die Kommunikation über einen Kommunikationsprozessor wird von diesem Baustein nicht unterstützt.

3.2 Unterstützte Feldbus Gateways / Sensoren

Der SICK Sensor kommuniziert über einen Feldbus (Profibus/Profinet) mit der Steuerung. Sollte der Sensor die oben genannten Feldbusse nicht unterstützen, können Gateway-Module eingesetzt werden.

Folgenden Gateways werden vom Funktionsbaustein unterstützt:
- CDM 425 (Profinet), ab Firmware Version V3.31
- CDF 600 (Profibus), ab Firmware Version V1.15
- CDM 420 inkl. CMF400 Profibus Modul, ab Firmware Version V1.100

Notwendige RFU Firmware Version:
- RFU6xx, ab Firmware Version V1.30

3.3 Konfiguration in Step7

Bevor der Funktionsbaustein verwendet werden kann, muss in der Step7 Hardwarekonfiguration der RFU entsprechend projektiert werden. Hierfür muss die entsprechende Gerätestammdatei (GSD-Datei) in die Step7 Hardwarebibliothek importiert werden.

Der Funktionsbaustein ist speziell für den Handshake-Mode ausgelegt. Bitte nur HS-Module mit einer Länge zwischen 8…128 Bytes verwenden. Die verwendeten Adressen dürfen im Peripheriebereich oder außerhalb projektiert werden. Eine Adresszuweisung auf Peripheriebereiche, denen ein Teilprozessabbild mit OB6x-Anbindung (Taktsynchronalarme) zugeordnet ist, darf nicht verwendet werden.

Abbildung 2 zeigt eine Beispielprojektierung des RFUs.

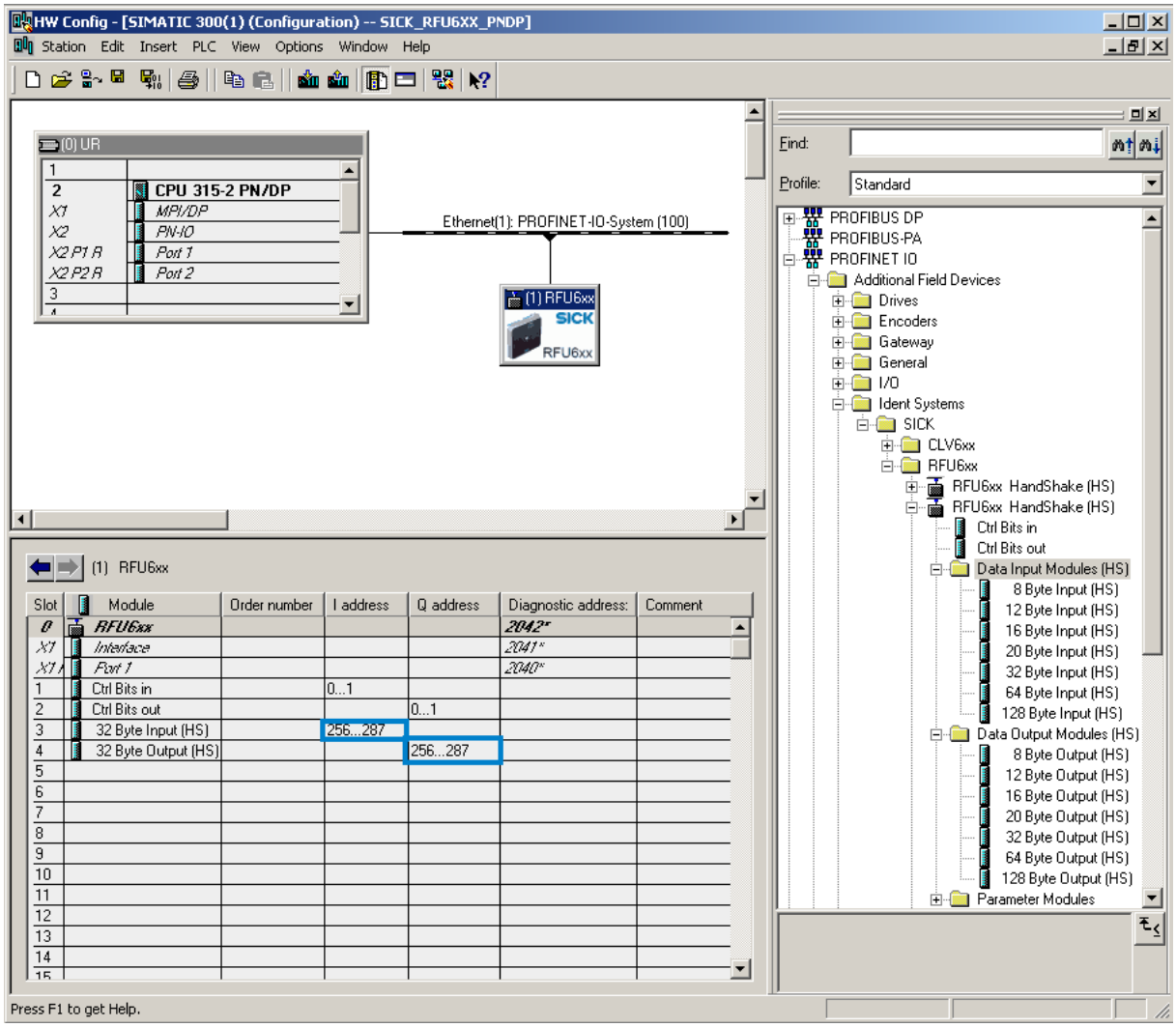

Abbildung 2: Step7 Hardwarekonfigurationsbeispiel

4 Bausteinbeschreibung

Der Funktionsbaustein ist ein asynchron arbeitender FB, d. h. die Bearbeitung erstreckt sich über mehrere FB-Aufrufe. Dies setzt voraus, dass der Baustein zyklisch im Anwenderprogramm aufgerufen wird.

Der RFU Baustein kapselt den Funktionsbaustein „SICK CCOM PNDP“ (FB10), der die Kommunikation zwischen SPS und Sensor ermöglich. Der FC10 (SICK COLA ACCESS) wird intern zur Interpretation der CoLa-Telegramme verwendet.

4.1 Bausteinspezifikationen

Bausteinnummer:	FB74
Bausteinname:	SICK RFU6XX PNDP
Version:	2.2
Aufgerufene Bausteine:	SFC 14 (DPRD_DAT) SFC 15 (DPWR_DAT) SFC20 (BLKMOV) SFB4 (TON) FB10 (SICK CCOM PNDP) FC10 (SICK COLA ACCESS)
Verwendete Datenbausteine:	DB74 (SICK RFU DATA)
Bausteinaufruf:	Zyklisch
Verwendete Merker:	keine
Verwendete Zähler:	keine
Verwendetet Register:	AR1, AR2 (für Multiinstanzaufruf)
Muliinstanzfähig:	ja
Erstellsprache:	Step7-AWL
Step7 Version:	Simatic Step7 V5.5

Die im Funktionsbaustein verwendeten Systemfunktionen (SFCs) müssen auf der jeweils verwendeten Steuerung vorhanden sein.

Beim ändern von Bausteinnummern müssen die entsprechenden Aufrufe im SICK RFU6XX PNDP Baustein aktualisiert werden.

4.2 Arbeitsweise

Um den RFU Baustein einsetzen zu können, müssen zunächst die folgenden Kommunikationsparameter angegeben werden:

IN_ADDR: Projektierte Anfangsadresse aus dem E-Bereich des verwendeten Input-Moduls. Die Eingangsadresse wird mit der Hardwareprojektierung festgelegt (siehe Kapitel 3.3). Der Wert muss im Hexadezimalformat angegeben werden (z.B. Adresse 256 = W#16#100).

IN_LEN: Länge des verwendeten Input-Moduls in der Hardwarekonfiguration. Die Länge des Eingabemoduls wird mit der Hardwareprojektierung festgelegt (siehe Kapitel 3.3).

OUT_ADDR: Projektierte Anfangsadresse aus dem A-Bereich des verwendeten Output-Moduls. Die Ausgangsadresse wird mit der Hardwareprojektierung festgelegt (siehe Kapitel siehe Kapitel 3.3). Der Wert muss im Hexadezimalformat angegeben werden (z.B. Adresse 256 = W#16#100).

OUT_LEN: Länge des verwendeten Output-Moduls in der Hardwarekonfiguration. Die Länge des Ausgabemoduls wird mit der Hardwareprojektierung festgelegt (siehe Kapitel siehe Kapitel 3.3).

DATA: Der zum Funktionsbaustein gehörende Datenbaustein (DB74) beinhaltet Ein- und Ausgabeparameter der unterstützten Bausteinaktionen. Der Datenbaustein muss dem Eingangsparameter „DATA“ des Funktionsbausteins übergeben werden.

Ausführbare Bausteinfunktionen:

- Trigger on → Öffnet das Lesetor des Gerätes per CoLa Kommando
- Trigger off → Schließt das Lesetor des Gerätes per CoLa Kommando
- Read Tag → Auslesen von Transponderdaten
- Write Tag → Schreiben von Transponderdaten
- Kommunikationstest → Prüft, ob das Gerät per „sRI0“ Kommando erreichbar ist
- Free Command → Ausführen eines frei wählbaren CoLa Kommandos

Um eine Bausteinaktion (TRIG_ON, RD_TAG, etc.) auszuführen, muss zunächst die gewünschte Aktion ausgewählt werden. Es kann immer nur eine Aktion gleichzeitig ausgeführt werden. Um die Aktion auszuführen, muss der Parameter START_REQ mit einer positiven Flanke (Signalwechsel von logisch null auf eins) angetriggert werden. Solange noch keine gültige Geräteantwort empfangen wurde, wird dies über den Parameter REQ_BUSY signalisiert.

Wenn der Baustein am Ausgangsparameter REQ_DONE = TRUE signalisiert, wurde die Aktion erfolgreich durchgeführt. Wurden bei dieser Aktion (z.B. RD_TAG) Daten vom Gerät angefordert, werden diese in den jeweiligen Datenbereich des zugehörigen Nutzdatenbausteins (DATA) kopiert.

Daten die per Triggerbefehl (TRIG_ON, TRIG_OFF) oder direkt vom Gerät gesendet werden (z.B. direkter Trigger über eine Lichtschranke), werden in den Datenbaustein (ReadingResult.arrResult) abgelegt. Der Ausgangsparameter RD_DONE zeigt für einen SPS Zyklus an, dass neue Daten empfangen wurden. Die vom Gerät gesendeten Daten können im SOPAS Ausgabeformat geändert, bzw. angepasst werden (siehe Kapitel 4.5.3).

4.3 Verhalten im Fehlerfall

Bei einem fehlerhaften Eingabewert oder einer fehlerhaften Eingangsbeschaltung des FBs, wird ein Errorbit (ERROR) gesetzt und ein Fehlercode (ERRORCODE) ausgegeben. In diesem Fall wird keine weitere Bearbeitung durchgeführt. Die Diagnoseparameter (ERROR, ERRORCODE) des FBs behalten solange ihren Wert, bis ein neuer Auftrag gestartet wird.

Über das RESET Bit ist es möglich, die Kommunikation zwischen dem Sensor und der SPS zurückzusetzen. Der Reset-Befehl wird ausgeführt, sobald das RESET Bit vorgewählt und das START_REQ Bit mit einer positiven Flanke (Signalwechsel von null auf eins) angetriggert wird. Das REQ_BUSY Bit signalisiert, dass der Befehl bearbeitet wird. Ist die Reset-Routine abgeschlossen, wird das REQ_DONE Bit gesetzt.

Durch das Rücksetzen werden folgende Aktionen ausgeführt:
- Rücksetzen der Counter vom Confirmed Messaging Protokoll (Gerätekommunikation)
- Rücksetzen aller Fehlermeldungen

4.4 Timing

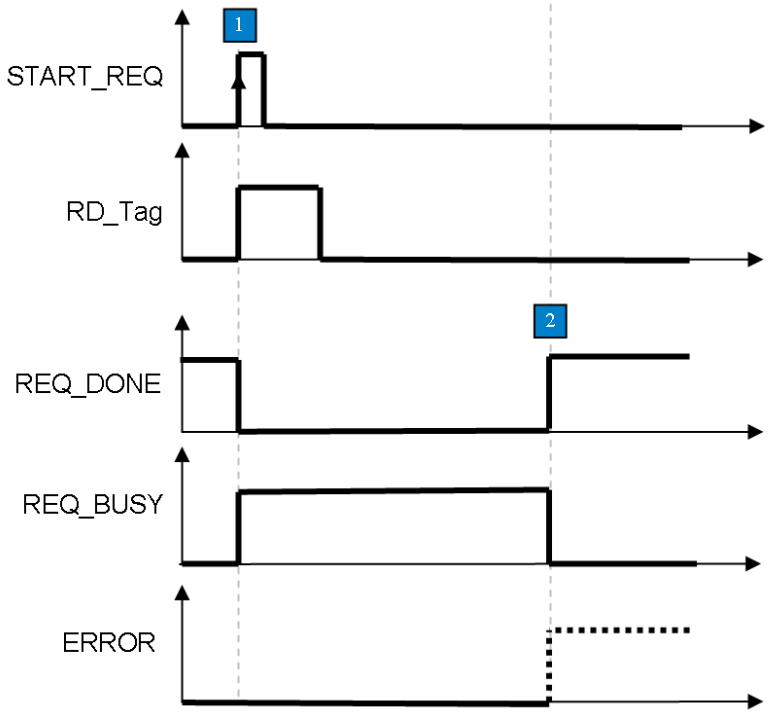

Abbildung 3: Timing Diagramm

1: Anforderung durch Pos Flanke an START_REQ
Die gewünschte Aktion (hier RD_Tag) muss vorher/zeitgleich ausgewählt werden. Es darf nur eine Aktion zeitgleich ausgewählt werden, sonst wird mit „ERROR“ abgebrochen.

2: Wenn alle Kommandos gesendet sind und alle Antworten empfangen wurden, wird die Aktion mit „REQ_Done“ beendet. Wenn die Aktion fehlerhaft verläuft, wird mit „ERROR“ beendet. Bei Abbruch mit „ERROR“ enthält „ERRORCODE“ den aufgetretenen Fehler.

4.5 Werteübergabe

Der mitgelieferte Datenbaustein „SICK RFU DATA“ (DB74) beinhaltet Ein- und Ausgabeparameter aller unterstützten Bausteinaktionen. Der Datenbaustein kann je nach Anwenderprogramm umbenannt werden. Die Datenstruktur ist fest vordefiniert und darf, bis auf den letzten Eintrag (ReadingResult.arrResult), nicht geändert werden (siehe Kapitel 4.6: Empfangen von Leseergebnissen > 200 Byte).

DB74 -- "SICK RFU DATA" -- SICK_RFU6XX_PNDP\SIMATIC 300(1)\CPU 315-2 PN/DP\...\DB74

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	Mode	STRUCT		-- MODE --
+0.0	bMode	BOOL	FALSE	1: Use a fixed UII | 0: Use the UII of the transponder in the field (IN)
+2.0	iUIILength	INT	0	Byte length of UII (IN/OUT)
+4.0	arrPC	ARRAY[1..4]		PC word (OUT)
*1.0		CHAR		
+8.0	arrUII	ARRAY[1..60]		Transponder UII (IN/OUT)
*1.0		CHAR		
+68.0	arrRSSI	ARRAY[1..4]	0	RSSI values of the 4 antennas (Internal/External)
*2.0		INT		
=76.0		END_STRUCT		
+76.0	ReadTag	STRUCT		--- READ TAG ---
+0.0	nBank	BYTE	B#16#0	Bank selection (0=RESERVED | 1=UII/EPC | 2=TID | 3=USER) (IN)
+2.0	nStartWord	WORD	W#16#0	First word to read, starting from 0 (IN)
+4.0	nWordCount	BYTE	B#16#0	Number of words to read (IN)
+5.0	nRetry	BYTE	B#16#0	Number or retries, until failure is reported (IN)
+6.0	nAntenna	BYTE	B#16#0	Antenna mask for reading (1= Internal,2,4,8)(IN)
+8.0	iDataLength	INT	0	Byte length of the reading data (OUT)
+10.0	arrData	ARRAY[1..64]		Read data (OUT)
*1.0		BYTE		
=74.0		END_STRUCT		
+150.0	WriteTag	STRUCT		--- WRITE TAG ---
+0.0	nBank	BYTE	B#16#0	Bank selection (0=RESERVED | 1=UII / EPC | 2=TID | 3=USER) (IN)
+2.0	nStartWord	WORD	W#16#0	First word to write, starting from 0 (IN)
+4.0	nWordCount	BYTE	B#16#0	Number of words to write (IN)
+5.0	nRetry	BYTE	B#16#0	Number or retries, until failure is reported (IN)
+6.0	nAntenna	BYTE	B#16#0	Antenna mask for writing (1= Internal,2,4,8)(IN)
+8.0	arrData	ARRAY[1..64]		Data to be written (IN)
*1.0		BYTE		
=72.0		END_STRUCT		
+222.0	FreeCommand	STRUCT		-- FREE COMMAND --
+0.0	iCommandLength	INT	0	Length of the free command (IN)
+2.0	arrCommand	ARRAY[1..100]		Command (SICK CoLa-A protocol without [STX]/[ETX] framing) (IN)
*1.0		CHAR		
+102.0	iResultLength	INT	0	Byte length of the free command result (OUT)
+104.0	arrResult	ARRAY[1..100]		Result (SICK CoLA-A protocol) (OUT)
*1.0		CHAR		
=204.0		END_STRUCT		
+426.0	ReadingResult	STRUCT		-- READING RESULT --
+0.0	nCounter	BYTE	B#16#0	This counter is incremented if a new reading result has arrived (OUT)
+2.0	iLength	INT	0	Byte length of the reading result (OUT)
+4.0	arrResult	ARRAY[1..200]		Reading result data (OUT)
*1.0		CHAR		
=204.0		END_STRUCT		
=630.0		END_STRUCT		

Abbildung 4: Struktur des SICK RFU DATA Nutzdaten DBs

4.5.1 Mode

Der RFU kann immer nur mit einem Transponder gleichzeitig kommunizieren. Aus diesem Grund werden Lese- und Schreibbefehle immer adressiert ausgeführt. Zum Identifizieren des Transponders verwendet der FB die UII (Unique Item Identifier).

Um zu bestimmen, mit welcher Transponder UII kommuniziert werden soll, unterstützt der Funktionsbaustein zwei Modi:

Mode 1: Es wird immer mit dem Transponder kommuniziert, der sich aktuell im Lesefeld befindet. Dieser Modus kann nur eingesetzt werden, wenn sich genau ein Tag im Feld befindet. Vorraussetzung für diesen Modus ist eine spezielle Parametrierung des RFUs in SOPAS-ET (siehe Kapitel 4.5.2 und 4.5.3).

Mode 2: Es wird eine, vom Anwender definierte Transponder-UII zur Kommunikation verwendet.

Parameter	Deklaration	Datentyp	Beschreibung
Mode.bMode	Input	BOOL	Adressierungsmodus FALSE: Mode 1 aktiv TRUE: Mode 2 aktiv
Mode.iUIILength	Input/Output	INT	Länge der verwendeten UII 0= Unadressiertes Lesen/Schreiben 1..60= Länge der im Array „Mode.arrUII“ definierten UII. *Im Mode 1 wird die UII automatisch bestimmt.*
Mode.arrPC	Output	ARRAY [1..4] OF CHAR	PC Word des verwendeten Transponders. *(Nur Mode 1)*
Mode.arrUII	Input/Output	ARRAY [1..60] OF CHAR	Transponder Identifikation (UII) *Im Mode 1 wird die UII automatisch bestimmt.*
Mode.arrRSSI	Output	ARRAY [1..4] OF INT	RSSI-Wert (Empfangsstärke) des verwendeten Transponders. *(Nur Mode 1)* [1]= RSSI-Wert von Antenne 1 (intern) [2]= RSSI-Wert von Antenne 2 (extern) [3]= RSSI-Wert von Antenne 3 (extern) [4]= RSSI-Wert von Antenne 4 (extern)

Tabelle 1: Mode Parameter

4.5.2 Mode 1: SOPAS-ET Objekttriggersteuerung

In der Objekttriggersteuerung wird festgelegt, wann das Lesetor geöffnet bzw. geschlossen wird. Nach jedem Lesetor sendet der Sensor ein Leseergebnis zur SPS-Steuerung. Der Funktionsbaustein nutzt diesen Mechanismus, um die UII, das PC-Word und die RSSI-Werte des anzusprechenden Transponders auszulesen.

Die SOPAS-ET Einstellungen unter dem Menüpunkt *Parameter → Reading Configuration → Objekt Trigger Control* müssen so vorgenommen werden, dass das Triggerfenster über ein SOPAS-Kommando geöffnet wird und bei einem Good Read oder nach einer festgelegten Zeit (hier 1000ms) wieder geschlossen wird.

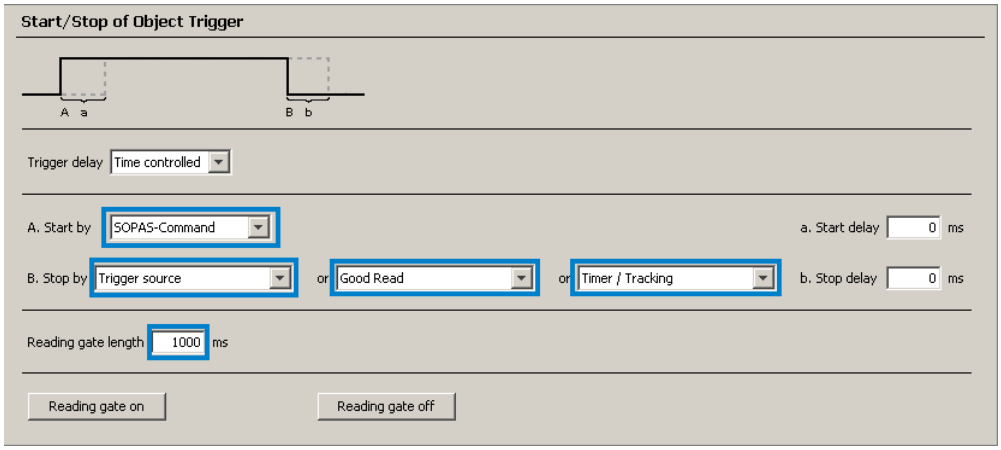

Abbildung 5: Triggereinstellung (SOPAS)

4.5.3 Mode 1: SOPAS-ET Ausgabeformat

Das Ausgabeformat definiert den Inhalt des Telegramms, das vom Gerät gesendet wird, sobald das Triggerfenster geschlossen wurde. Auf der SPS-Seite wird dieses Telegramm ausgewertet und die Informationen ausgelesen, die für das adressierte lesen/schreiben von Tag-Daten benötigt werden. Um den Funktionsbaustein im Mode 1 einsetzten zu können, muss das SOPAS Ausgabeformat wie in Abbildung 6 zu sehen aufgebaut sein.

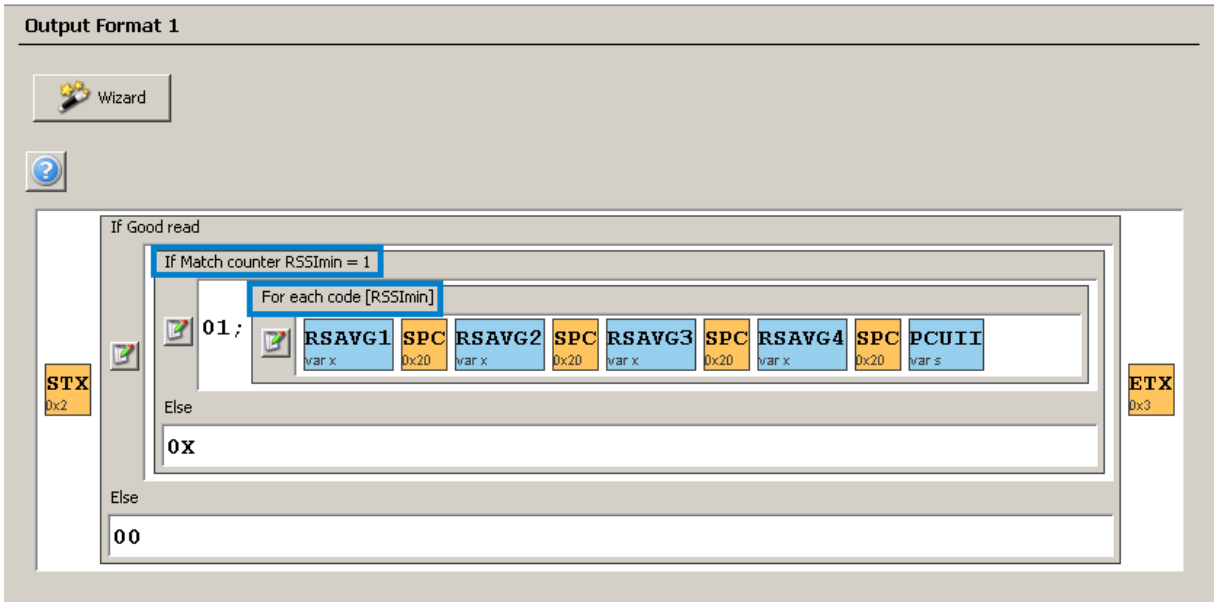

Abbildung 6: SOPAS Ausgabeformat

Es ist darauf zu achten, dass das Format der Blöcke „RSAVG1...4“ auf „Hexadecimal“ eingestellt ist (Doppelklick auf den jeweiligen Block). Der „PCUII“ Block darf nicht verändert werden.

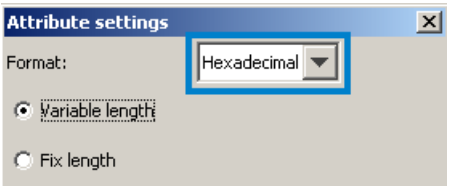

Abbildung 7: Einstellungen der RSSI-Blöcke (RSAVG1...4)

Je nach Anzahl der Tags, die sich im Empfangsbereich des RFUs befinden und der eingestellte RSSI-Schwelle, werden die folgenden Telegramme (im ASCII-Format) gesendet:

Fall 1: Ein Tag im Feld:
[STX]01;[RSSI Antenne 1] [RSSI Antenne 2] [RSSI Antenne 3] [RSSI Antenne 4] [PC+UII][ETX]

Fall 2: Mehrere Tags im Feld:
[STX]0X[ETX]

Fall 3: Kein Tag im Feld:
[STX]00[ETX]

4.5.4 Read Tag

Die Read Tag Aktion liest einen definierten Datenbereich eines Tags aus. Die Aktion ist immer nur auf einen Tag anwendbar. Mit welchem Transponder kommuniziert werden soll, ist vom gewählten Modus abhängig (siehe Kapitel 4.5.1).

Vor einem lesenden Zugriff auf einen RFID Tag müssen in der Struktur „ReadTag“ die folgenden Parameter angegeben werden.

Parameter	Deklaration	Datentyp	Beschreibung
ReadTag.nBank	Input	BYTE	Auswahl der Speicherbank, von der gelesen werden soll. 0= Reserviert 1= UII/EPC 2= TID 3= User Memory
ReadTag. nStartWord	Input	WORD	Erstes Word (16Bit) das ausgelesen werden soll.
ReadTag. nWordCount	Input	BYTE	Anzahl der Wörter (16Bit) die ausgelesen werden sollen. Gültiger Wertbereich: [1..32]
ReadTag.nRetry	Input	BYTE	Anzahl der Leseversuche die durchgeführt werden sollen. Gültiger Wertebereich: LoNibble (Retries auf einem Kanal) B#16#[0..7] HiNibble (Retries mit Kanalwechsel) B#16#[0..5] Beispiel: ReadTag.iRetry=B#16#32 führt 3 Kanalwechsel (*4 Kanäle*) mit jeweils 2 Retries pro Kanal aus (*3 Wiederholungen*), d.h. insgesamt 4x3=**12 Versuche**

Parameter	Deklaration	Datentyp	Beschreibung
ReadTag.nAntenna	Input	BYTE	Antennenauswahl für den aktuellen Leseauftrag. Pro Auftrag kann jeweils nur eine Antenne ausgewählt werden. A1= Antenne 1 (intern/extern, je nach Gerätevariation) A2= Antenne 2 (extern) A3= Antenne 3 (extern) A4= Antenne 4 (extern) Wert | A4 | A3 | A2 | A1 1 | | | | X 2 | | | X | 4 | | X | | 8 | X | | | Gültiger Wertebereich: [1, 2, 4, 8]
ReadTag. iDataLength	Output	INT	Die gültige Länge des gelesenen Inhalts in Byte. Notwendig, um zu definieren, welche Inhalte von ReadTag.arrData gültig sind.
ReadTag. arrData	Output	ARRAY [1..64] OF BYTE	Inhalt der gelesenen Daten.

Tabelle 2: Read Tag Parameter

4.5.5 Write Tag

Die Write Tag Funktion schreibt auf einen definierten Datenbereich eines Tags. Die Aktion ist immer nur auf einen Tag anwendbar. Mit welchem Transponder kommuniziert werden soll, ist vom gewählten Modus abhängig (siehe Kapitel 4.5.1).

Vor einem schreibenden Zugriff auf ein RFID Tag müssen in der Struktur „WriteTag“ die folgenden Parameter angegeben werden.

Parameter	Deklaration	Datentyp	Beschreibung
WriteTag.nBank	Input	BYTE	Auswahl der Speicherbank, von der gelesen werden soll. 0= Reserviert 1= UII/EPC 2= TID 3= User Memory
WriteTag. nStartWord	Input	WORD	Erstes Wort (16Bit) das beschrieben werden soll.
WriteTag. nWordCount	Input	BYTE	Anzahl der Wörter (16Bit) die beschrieben werden sollen. Gültiger Wertbereich: [1..32]

<table>
<tr><th>Parameter</th><th>Deklaration</th><th>Datentyp</th><th>Beschreibung</th></tr>
<tr><td>WriteTag.nRetry</td><td>Input</td><td>BYTE</td><td>Anzahl der Schreibversuche die durchgeführt werden sollen.

Gültiger Wertebereich:
LoNibble (Retries auf einem Kanal)
B#16#[0..7]
HiNibble (Retries mit Kanalwechsel)
B#16#[0..5]

Beispiel:
WriteTag.iRetry=B#16#32 führt 3 Kanalwechsel (4 Kanäle) mit jeweils 2 Retries pro Kanal aus (3 Wiederholungen), d.h. insgesamt 4x3=12 Versuche</td></tr>
<tr><td>WriteTag.nAntenna</td><td>Input</td><td>BYTE</td><td>Antennenauswahl für den aktuellen Schreibauftrag. Pro Auftrag kann jeweils nur eine Antenne ausgewählt werden.

A1= Antenne 1 (intern/extern, je nach Gerätevariation)
A2= Antenne 2 (extern)
A3= Antenne 3 (extern)
A4= Antenne 4 (extern)

<table><tr><th>Wert</th><th>A4</th><th>A3</th><th>A2</th><th>A1</th></tr><tr><td>1</td><td></td><td></td><td></td><td>X</td></tr><tr><td>2</td><td></td><td></td><td>X</td><td></td></tr><tr><td>4</td><td></td><td>X</td><td></td><td></td></tr><tr><td>8</td><td>X</td><td></td><td></td><td></td></tr></table>
Gültiger Wertebereich:
[1, 2, 4, 8]</td></tr>
<tr><td>WriteTag.
arrData</td><td>Output</td><td>ARRAY
[1..64] OF
BYTE</td><td>Daten, die auf den gewählten Tag-Bereich geschrieben werden sollen.</td></tr>
</table>

Tabelle 3: Write Tag Parameter

4.5.6 Free Command

Mit Hilfe des freien Kommandos hat man die Möglichkeit über ein gültiges CoLa Kommando mit dem RFU zu kommunizieren. Hierfür ist es erforderlich, das Kommando in dem Parameter „arrCommand“ der Struktur „FreeCommand“ zu hinterlegen. Die Zeichenlänge des zu übertragenden Kommandos wird in den Parameter „iCommandLength“ geschrieben. Die Kommandos können der Gerätebeschreibung oder SOPAS-ET entnommen werden.

Parameter	Deklaration	Datentyp	Beschreibung
FreeCommand. iCommandLength	Input	INT	Zeichenlänge des zu übertragenden CoLa Kommandos. Gültiger Wertebereich [1..100]
FreeCommand. arrCommand	Input	ARRAY [1..100] OF CHAR	Frei wählbares CoLa Kommando (Kommandos siehe Gerätdokumentation).
FreeCommand. iResultLength	Output	INT	Bytelänge des empfangenden CoLa Telegramms.
FreeCommand. arrResult	Output	ARRAY [1..100] OF CHAR	Empfangende Antwort des gesendeten CoLa Telegramms.

Tabelle 4: Free Command Parameter

4.5.7 Reading Result

In dem Array „ReadingResult.arrResult“ werden Daten abgelegt, die per Triggerbefehl (TRIG_ON, TRIG_OFF) oder direkt vom Gerät gesendet werden (z.B. direkter Trigger über eine Lichtschranke). Der Ausgangsparameter RD_DONE signalisiert, ob Daten empfangen wurden.

Parameter	Deklaration	Datentyp	Beschreibung
ReadingResult. nCounter	Output	BYTE	Der Empfangszähler wird um eins inkrementiert, sobald ein neues Leseergebnis empfangen wurde. Wertebereich: [0x00..0xFF]
ReadingResult. iLength	Output	INT	Bytelänge des empfangenden Leseergebnisses.
ReadingResult. arrResult	Output	ARRAY [1..200] of BYTE	Empfangende Antwort auf ein Triggersignal (Über das SOPAS Ausgabeformat definierbar). Die maximale Länge der empfangenden Daten beträgt 200 Bytes. Kapitel 4.6 beschreibt dass Vorgehen beim Empfang von längeren Datentelegrammen.

Tabelle 5: Reading Result Parameter

4.6 Empfangen von Leseergebnissen > 200 Byte

Der Funktionsbaustein ist darauf ausgelegt, Leseergebnisse bis zu einer Länge von 200 Bytes zu empfangen. Sollen längere Daten empfangen werden, muss der Funktionsbaustein an den folgenden Stellen abgeändert werden:

Änderung im SICK RFU DATA Datenbaustein:
Im mitgelieferten Nutzdatenbaustein (DB74) muss die Länge des Array „ReadingResult. arrResult“ so angepasst werden, dass das zu empfangende Leseergebnis in den Datenbereich der Variablen passt.

+426.0	ReadingResult	STRUCT		-- READING RESULT --
+0.0	nCounter	BYTE	B#16#0	This counter is incremented if a new reading result has arrived (OUT)
+2.0	iLength	INT	0	Byte length of the reading result (OUT)
+4.0	arrResult	ARRAY[1..200]		Reading result data (OUT)
*1.0		CHAR		
=204.0		END_STRUCT		

Abbildung 8: Empfangen von Leseergebnissen > 200 Bytes (Änderung im Datenbaustein)

Änderung im SICK RFU6XX PNDP Funktionsbaustein:
Im statischen Bereich der Variablenübersicht muss die Länge der Variable „arrRecord“ so angepasst werden, dass das Leseergebnis in den Datenbereich der Variablen passt. Das Array darf eine Länge von 300 Bytes nicht unterschreiten, muss aber größer/gleich der ReadingResult.arrResult Länge sein.

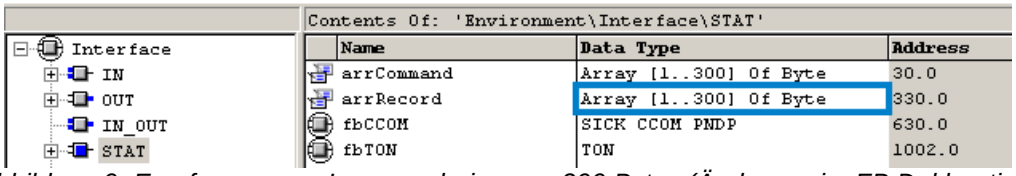

Abbildung 9: Empfangen von Leseergebnissen > 200 Bytes (Änderung im FB Deklaration)

Im Netzwerk 3 des SICK RFU6XX PNDP FBs müssen die neu definierten Arraylängen eingetragen werden.

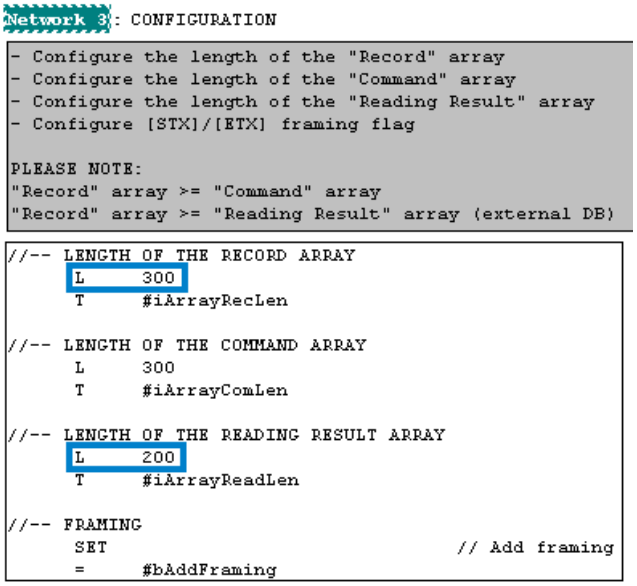

Abbildung 10: Empfangen von Leseergebnissen > 200 Bytes (Änderung im Bausteincode)

Nach der Änderung muss die Instanz des Funktionsbausteins aktualisiert werden. Anschließend muss der geänderte Nutzdatenbaustein sowie der Funktionsbaustein zusammen mit der aktualisierten Instanz neu auf die SPS übertragen werden.

5 Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E,M,D,L, Konst.	Enable Eingang (KOP und FUP)
IN_ADDR	INPUT	WORD	E,M,D,L, Konst.	Projektierte Anfangsadresse aus dem E-Bereich des gewählten Moduls.
IN_LEN	INPUT	INT	E,M,D,L, Konst.	Länge des verwendeten Input-Moduls in der Hardwarekonfiguration. Gültiger Wertebereich: [8..128]
OUT_ADDR	INPUT	WORD	E,M,D,L, Konst.	Projektierte Anfangsadresse aus dem A-Bereich des gewählten Moduls.
OUT_LEN	INPUT	INT	E,M,D,L, Konst.	Länge des verwendeten Output-Moduls in der Hardwarekonfiguration. Gültiger Wertebereich: [8..128]
CAN_ID	INPUT	INT	E,M,D,L, Konst.	CAN-ID des anzusprechenden Sensors. Wenn kein CAN-Netzwerk verwendet wird, ist die CAN-ID = B#16#00 Der Master bzw. der Multiplexer wird immer mit der CAN-ID = B#16#00 angesprochen, auch wenn dieser eine andere CAN-ID zugewiesen ist.
TOUT	INPUT	TIME	E,M,D,L, Konst.	Zeit, nachdem ein Timeout-Fehler ausgelöst wird.
START_REQ	INPUT	BOOL	E,M,D,L	Positive Flanke: Ausführen der gewählten Bausteinaktion.
TRIG_ON	INPUT	BOOL	E,M,D,L, Konst.	Bausteinaktion: Ausführen eines Geräte Triggers (Triggerfenster öffnen)
TRIG_OFF	INPUT	BOOL	E,M,D,L, Konst.	Bausteinaktion: Ausführen eines Geräte Triggers (Triggerfenster schließen) Das vom Gerät gesendet Ergebnis (SOPAS Ausgabeformat) wird in der Variablen „ReadingResult.arrResult“ des Nutzdaten DBs (DB74) abgelegt.

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
RD_TAG	INPUT	BOOL	E,M,D,L, Konst.	Bausteinaktion: Auslesen von Tag Inhalten. Die Aktion setzt voraus, dass die Parameter der Struktur „ReadTag“ des übergebenen Datenbausteins mit gültigen Werten belegt sind (siehe Kapitel 4.5.4). Welcher Transponder ausgelesen werden soll ist vom gewählten Adressierungsmodus abhängig (siehe Kapitel 4.5.1).
WR_TAG	INPUT	BOOL	E,M,D,L, Konst.	Bausteinaktion: Schreiben von Tag Inhalten. Die Aktion setzt voraus, dass die Parameter der Struktur „WriteTag“ des übergebenen Datenbausteins die Parameter mit gültigen Werten belegt sind (siehe Kapitel 4.5.5). Welcher Transponder beschrieben werden soll ist vom gewählten Adressierungsmodus abhängig (siehe Kapitel 4.5.1).
COM_TEST	INPUT	BOOL	E,M,D,L, Konst.	Bausteinaktion: Ausführen eines Kommunikationstests. REQ_DONE= TRUE: Kommunikation OK REQ_DONE= FALSE: Kommunikation nicht OK
FREE_ COMMAND	INPUT	BOOL	E,M,D,L, Konst.	Bausteinaktion: Ausführen eines freien Kommandos. Die Aktion setzt voraus, dass im Nutzdatenbaustein (DB74) in der Struktur (FreeCommand) die Parameter iCommandLength und arrCommand mit gültigen Daten belegt sind (siehe Kapitel 4.5.6). Die Kommandoantwort steht nach einer erfolgreichen Übertragung (REQ_DONE=TRUE) im RESULT Bereich des Datenbausteins zur Verfügung.

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
RESET	INPUT	BOOL	E,M,D,L, Konst.	Bausteinaktion: Rücksetzten der Kommunikation zum Gerät.
DATA	INPUT	BLOCK_DB	Konst.	Übergabe des zugehörigen Nutzdatenbausteins, der für die Parametrierung der Bausteinfunktionen sowie für das Ablegen des Leseergebnisses benötigt wird (DB74).
RD_DONE	OUTPUT	BOOL	A,M,D,L	Positive Flanke: Neues Leseergebnis empfangen
REQ_DONE	OUTPUT	BOOL	A,M,D,L	Zeigt an, ob die gewählte Bausteinaktion fehlerfrei durchgeführt wurde. TRUE: Bearbeitung abgeschlossen FALSE: Bearbeitung nicht abgeschlossen
REQ_BUSY	OUTPUT	BOOL	A,M,D,L	Auftrag ist in Bearbeitung.
ERROR	OUTPUT	BOOL	A,M,D,L	Fehler Bit: 0: Kein Fehler 1: Abbruch mit Fehler
ERROR CODE	OUTPUT	WORD	A,M,D,L	Fehlerstatus (siehe Fehlercodes)
ENO	OUTPUT	BOOL	A,M,D,L	Enable Ausgang (KOP und FUP)

Tabelle 6: Bausteinparameter

6 Fehlercodes

Der Parameter ERRORCODE enthält die folgenden Fehlerinformationen:

Fehlercode	Kurzbeschreibung	Beschreibung
W#16#0000	Kein Fehler	Kein Fehler
W#16#0001	Timeout Fehler	Auftrag konnte innerhalb der gewählten Timeoutzeit nicht ausgeführt werden Dies könnte folgende Ursachen haben: - Gerät ist nicht mit der SPS Verbunden - Kommunikationsparameter fehlerhaft - CAN-Bus Teilnehmer nicht vorhanden
W#16#0002	Interner Bausteinfehler	Interner Bausteinfehler
W#16#0003	Keine oder mehr als eine Bausteinaktion angewählt	Es kann immer nur eine Bausteinfunktion gleichzeitig ausgeführt werden
W#16#0004	Empfangendes Leseergebnis > Reading Result Array	Das empfangene Leseergebnis ist Länger als 200 Bytes. Zum Empfangen von längeren Leseergebnissen siehe Kapitel 4.6
W#16#0005	100 < FreeCommand. iCommandLength <=0	Ungültige Länge des freien Kommandos Gültiger Wertebereich: [1…100]
W#16#0006	Antwort des freien Kommandos > 100 Byte	Die Antwort auf das gesendete freie Kommando ist länger 100 Byte.
W#16#0007	63 < CAN_ID < 0	Ungültige CAN-ID Gültiger Wertebereich: [0..63]
W#16#0008	Reserviert	Reserviert
W#16#0009	Kommunikationsfehler	Kommunikation zum Gerät kann nicht hergestellt werden. Dies könnte folgende Ursachen haben: - Ungültige E/A Adressen - Ungültige Länge der E/A Adressen - Es wurde ein Telegramm > arrRecord empfangen
W#16#**XX**0A	Gerätefehler	Es ist ein Gerätefehler aufgetreten (’sFA XX’) **XX** = Gerätefehler (siehe Gerätedokumentation)

Fehlercode	Kurzbeschreibung	Beschreibung
W#16#000B	Ungültige Kommando-antwort	Die gewählte Aktion wurde nicht ausgeführt. Dies kann je nach Aktion die folgenden Ursachen haben: - Triggereinstellung in der SOPAS Gerätekonfiguration fehlerhaft - Gerät befindet sich nicht im „Run-Mode“ - Sende/Empfangsleistung zu gering - Tag nicht lang genug im Feld - Zugriff auf einen nicht existierenden Tag-Bereich (nStartWord und nWordCount Parameter prüfen) - Gewählte Antenne kann nicht auf den gewählten Tag zugreifen (nAntenna Parameter prüfen). - Ungültige UII (Mode.arrUII und Mode.iUiiLength prüfen)
W#16#000C – W#16#000F	Reserviert	Reserviert
W#16#0010	60 < Mode.iUIILength < 0	Ungültige UII Länge Gültiger Wertebereich: [0..60]
W#16#0011	3 < ReadTag.nBank < 0	Ungültige Bank (Read Tag) Gültiger Wertebereich: [0..3]
W#16#0012	32 < ReadTag. nWordCount <= 0	Der Funktionsbaustein kann maximal 32 Wörter (64 Bytes) vom Tag lesen. Gültiger Wertebereich [1..32]
W#16#0013	Ungültiger Retry (ReadTag.nRetry)	Ungültiger Retry Parameter (Read Tag) Gültiger Wertebereich: Low Nibble = [0..7] High Nibbel = [0..5]
W#16#0014	15 < ReadTag.nAntenna < 1	Ungültige Antennenanwahl (Read Tag) Gültiger Wertebereich: [1..15]
W#16#0015	3 < WriteTag.nBank < 0	Ungültige Bank (Write Tag) Gültiger Wertebereich: [0..3]
W#16#0016	32 < WriteTag. nWordCount <= 0	Der Funktionsbaustein kann maximal 32 Wörter (64 Bytes) auf den Tag schreiben Gültiger Wertebereich [1..32]

Fehlercode	Kurzbeschreibung	Beschreibung
W#16#0017	Ungültiger Retry (WriteTagTag.nRetry)	Ungültiger Retry Parameter (Write Tag) Gültiger Wertebereich: Low Nibble = [0..7] High Nibbel = [0..5]
W#16#0018	15 < WriteTag. nAntenna < 1	Ungültige Antennenanwahl (Write Tag) Gültiger Wertebereich: [1..15]
W#16#0019	Ungültiges Outputformat	Ungültiges SOPAS-ET Ausgabeformat. Prüfen Sie das Ausgabeformat (siehe Kapitel 4.5.3) Dieser Fehler kann nur im Mode 1 auftreten.
W#16#001A	Kein Tag im Feld	Es befindet sich kein Tag im Empfangsbereich des RFUs. Dieser Fehler kann nur im Mode 1 auftreten.
W#16#001B	Mehr als ein Tag im Feld oder Auswertebedingung RSSImin nicht erfüllt	Dieser Fehler kann mehrer Bedeutungen haben: - Es befindet sich mehr als ein Tag im Empfangsbereich des RFUs. - Die Auswertebedingung RSSImin ist nicht erfüllt (siehe SOPAS-ET). Dieser Fehler kann nur im Mode 1 auftreten.

Tabelle 7: Fehlercodes

7 Beispiele

Abbildung 11 zeigt eine Beispielbeschaltung des RFU6XX FBs. Die logische Eingangs- und Ausgangsadresse fängt bei Byte 256 (W#16#100) an. Die Länge der in der Hardwarekonfiguration projektierten Module beträgt 32 Bytes. Da der RFU nicht in einem CAN-Netzwerk betrieben wird, wird als CAN-ID fest eine null eingetragen.

Programmaufruf:

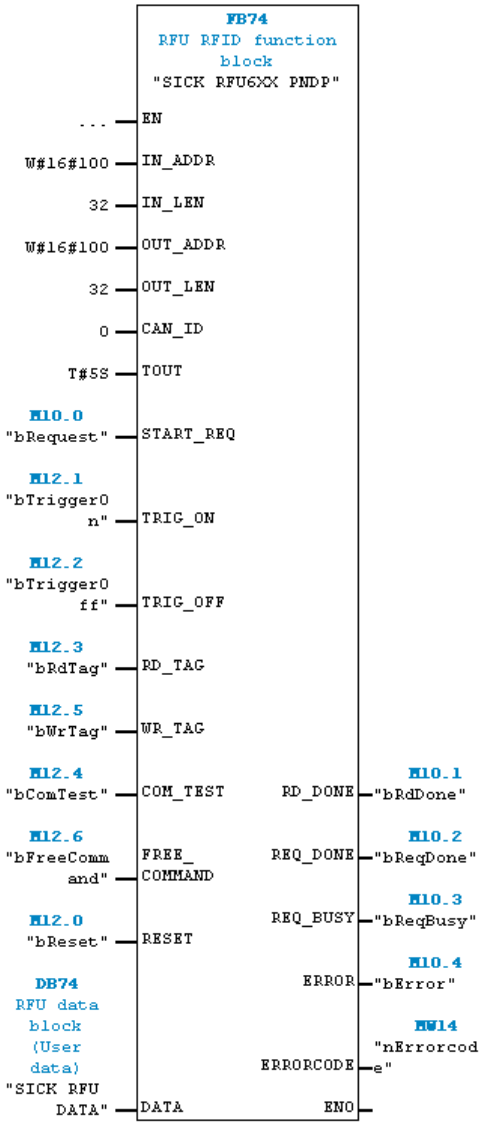

Abbildung 11: Beispielbeschaltung des SICK RFU6XX PNDP FBs

Slot	Module	Order number	I address	Q address	Diagnostic address:
0	RFU6xx				2042*
X1	Interface				2041*
X1 P1	Port 1				2040*
1	Ctrl Bits in		0...1		
2	Ctrl Bits out			0...1	
3	32 Byte Input (HS)		256...287		
4	32 Byte Output (HS)			256...287	

Abbildung 12: Step7 Hardwareprojektierung

7.1 Auslesen von Tag-Inhalten

Zunächst muss bestimmt werden, mit welchem Transponder kommuniziert werden soll. Ist das Bit „Mode.bMode = FALSE“ wird mit dem Transponder kommuniziert, welcher sich aktuell im Lesebereich des RFID Sensors befindet. Vorraussetzung für diesen Modus ist eine spezielle Parametrierung des RFUs in SOPAS-ET siehe Kapitel 4.5.2 und 4.5.3).

// ========== Mode ==========				
DB74.DBX	0.0	"SICK RFU DATA".Mode.bMode	BOOL	false

Abbildung 13: Auswahl des Kommunikationsmodus

Anschließend muss definiert werden, welche Inhalte aus dem Transponder ausgelesen werden sollen.

Bank: 3 (User Memory)
Tag-Bereich: 0 … 6 Wörter (12 Byte)
Anzahl der Retries: 0x57 (5 Kanalwechsel mit jeweils 7 Retries pro Kanal)
Antennenauswahl: 1 (Antenne 1)

// ========== Read Tag ==========			
DB74.DBB 76	"SICK RFU DATA".ReadTag.nBank	DEC	3
DB74.DBW 78	"SICK RFU DATA".ReadTag.nStartWord	DEC	0
DB74.DBB 80	"SICK RFU DATA".ReadTag.nWordCount	DEC	6
DB74.DBB 81	"SICK RFU DATA".ReadTag.nRetry	HEX	B#16#57
DB74.DBB 82	"SICK RFU DATA".ReadTag.nAntenna	DEC	1

Abbildung 14: Definition der Leseparameter

Die Leseaktion (bRdTag) wird ausgeführt, sobald das Bit „bRequest“ mit einer positiven Flanke angetriggert wird.

// SICK RFU6XX PNDP Function Block Example			
MW 16	"iCanID"	DEC	0
M 10.0	"bRequest"	BOOL	true
M 10.2	"bReqDone"	BOOL	true
M 10.3	"bReqBusy"	BOOL	false
M 10.4	"bError"	BOOL	false
MW 14	"nErrorcode"	HEX	W#16#0000
// Selection of the FB action to be executed			
M 12.1	"bTriggerOn"	BOOL	false
M 12.2	"bTriggerOff"	BOOL	false
M 12.3	"bRdTag"	BOOL	true
M 12.5	"bWrTag"	BOOL	false
M 12.4	"bComTest"	BOOL	false
M 12.6	"bFreeCommand"	BOOL	false
M 12.0	"bReset"	BOOL	false

Abbildung 15: Starten der Bausteinfunktion

Die Leseaktion ist abgeschlossen sobald das Bit „bReqDone = TRUE“ signalisiert. Der gelesenen Tag-Inhalte stehen im Array „ReadTag.arrData“ des Nutzdatenbausteins zur Verfügung. Die Variable „ReadTag.iDataLength“ gibt an, wie viele Bytes empfangen wurden bzw. gültig sind.

// ========== Read Tag ==========			
DB74.DBB 76	"SICK RFU DATA".ReadTag.nBank	DEC	3
DB74.DBW 78	"SICK RFU DATA".ReadTag.nStartWord	DEC	0
DB74.DBB 80	"SICK RFU DATA".ReadTag.nWordCount	DEC	6
DB74.DBB 81	"SICK RFU DATA".ReadTag.nRetry	HEX	B#16#57
DB74.DBB 82	"SICK RFU DATA".ReadTag.nAntenna	DEC	1
DB74.DBW 84	"SICK RFU DATA".ReadTag.iDataLength	DEC	12
DB74.DBB 86	"SICK RFU DATA".ReadTag.arrData[1]	CHARACTER	'H'
DB74.DBB 87	"SICK RFU DATA".ReadTag.arrData[2]	CHARACTER	'e'
DB74.DBB 88	"SICK RFU DATA".ReadTag.arrData[3]	CHARACTER	'l'
DB74.DBB 89	"SICK RFU DATA".ReadTag.arrData[4]	CHARACTER	'l'
DB74.DBB 90	"SICK RFU DATA".ReadTag.arrData[5]	CHARACTER	'o'
DB74.DBB 91	"SICK RFU DATA".ReadTag.arrData[6]	CHARACTER	' '
DB74.DBB 92	"SICK RFU DATA".ReadTag.arrData[7]	CHARACTER	'W'
DB74.DBB 93	"SICK RFU DATA".ReadTag.arrData[8]	CHARACTER	'o'
DB74.DBB 94	"SICK RFU DATA".ReadTag.arrData[9]	CHARACTER	'r'
DB74.DBB 95	"SICK RFU DATA".ReadTag.arrData[10]	CHARACTER	'l'
DB74.DBB 96	"SICK RFU DATA".ReadTag.arrData[11]	CHARACTER	'd'
DB74.DBB 97	"SICK RFU DATA".ReadTag.arrData[12]	CHARACTER	' '
DB74.DBB 98	"SICK RFU DATA".ReadTag.arrData[13]	CHARACTER	B#16#00
DB74.DBB 99	"SICK RFU DATA".ReadTag.arrData[14]	CHARACTER	B#16#00
DB74.DBB 100	"SICK RFU DATA".ReadTag.arrData[15]	CHARACTER	B#16#00

Abbildung 16: Gelesene Tag-Inhalte

7.2 Schreiben von Tag-Inhalten

Zunächst muss bestimmt werden, mit welchem Transponder kommuniziert werden soll. Ist das Bit „Mode.bMode = TRUE“ wird mit einem vorgegebenen Transponder kommuniziert, dessen UII im Vorfeld bekannt ist (hier: 12345678).

// ========== Mode ==========			
DB74.DBX 0.0	"SICK RFU DATA".Mode.bMode	BOOL	true
DB74.DBW 2	"SICK RFU DATA".Mode.iUIILength	DEC	8
DB74.DBB 8	"SICK RFU DATA".Mode.arrUII[1]	CHARACTER	'1'
DB74.DBB 9	"SICK RFU DATA".Mode.arrUII[2]	CHARACTER	'2'
DB74.DBB 10	"SICK RFU DATA".Mode.arrUII[3]	CHARACTER	'3'
DB74.DBB 11	"SICK RFU DATA".Mode.arrUII[4]	CHARACTER	'4'
DB74.DBB 12	"SICK RFU DATA".Mode.arrUII[5]	CHARACTER	'5'
DB74.DBB 13	"SICK RFU DATA".Mode.arrUII[6]	CHARACTER	'6'
DB74.DBB 14	"SICK RFU DATA".Mode.arrUII[7]	CHARACTER	'7'
DB74.DBB 15	"SICK RFU DATA".Mode.arrUII[8]	CHARACTER	'8'
DB74.DBB 16	"SICK RFU DATA".Mode.arrUII[9]	CHARACTER	B#16#00
DB74.DBB 17	"SICK RFU DATA".Mode.arrUII[10]	CHARACTER	B#16#00

Abbildung 17: Vorgabe der Transponder UII

Anschließend muss definiert werden, welche Inhalte auf den Tag geschrieben werden sollen und wo diese abgelegt werden sollen.

Bank:	3 (User Memory)
Tag-Bereich:	0 … 3 Wörter (6 Byte/Zeichen)
Anzahl der Retries:	0x57 (5 Kanalwechsel mit jeweils 7 Retries pro Kanal)
Antennenauswahl:	1 (Antenne 1)
Zu schreibender Inhalt:	„Tag 01“ (6 ASCII Zeichen)

// ========== Write Tag ==========			
DB74.DBB 150	"SICK RFU DATA".WriteTag.nBank	DEC	3
DB74.DBW 152	"SICK RFU DATA".WriteTag.nStartWord	DEC	0
DB74.DBB 154	"SICK RFU DATA".WriteTag.nWordCount	DEC	3
DB74.DBB 155	"SICK RFU DATA".WriteTag.nRetry	HEX	B#16#57
DB74.DBB 156	"SICK RFU DATA".WriteTag.nAntenna	HEX	B#16#01
DB74.DBB 158	"SICK RFU DATA".WriteTag.arrData[1]	CHARACTER	'T'
DB74.DBB 159	"SICK RFU DATA".WriteTag.arrData[2]	CHARACTER	'a'
DB74.DBB 160	"SICK RFU DATA".WriteTag.arrData[3]	CHARACTER	'g'
DB74.DBB 161	"SICK RFU DATA".WriteTag.arrData[4]	CHARACTER	' '
DB74.DBB 162	"SICK RFU DATA".WriteTag.arrData[5]	CHARACTER	'0'
DB74.DBB 163	"SICK RFU DATA".WriteTag.arrData[6]	CHARACTER	'1'

Abbildung 18: Definition der Leseparameter

Die Schreibaktion (bWrTag) wird ausgeführt, sobald das Bit „bRequest“ mit einer positiven Flanke angetriggert wird.

// SICK RFU6XX PNDP Function Block Example			
MW 16	"iCanID"	DEC	0
M 10.0	"bRequest"	BOOL	true
M 10.2	"bReqDone"	BOOL	true
M 10.3	"bReqBusy"	BOOL	false
M 10.4	"bError"	BOOL	false
MW 14	"nErrorcode"	HEX	W#16#0000
// Selection of the FB action to be executed			
M 12.1	"bTriggerOn"	BOOL	false
M 12.2	"bTriggerOff"	BOOL	false
M 12.3	"bRdTag"	BOOL	false
M 12.5	"bWrTag"	BOOL	true
M 12.4	"bComTest"	BOOL	false
M 12.6	"bFreeCommand"	BOOL	false
M 12.0	"bReset"	BOOL	false

Abbildung 19: Starten der Bausteinfunktion

Die Schreibaktion ist abgeschlossen sobald das Bit „bReqDone = TRUE“ signalisiert.