

Einfache Kommandoschnittstelle
Modicon M340-Steuerung
Bibliotheksbeschreibung

DTE810

DTE910

Inhalt

1	Übersicht	3
2	Haftungsausschluss	3
3	Einstellen der Verbindungsparameter im RFID-Reader	3
4	Projektieren der Verbindung in Unity Pro	5
4.1	Einfügen des RFID-Readers in das Softwareprojekt	6
4.2	Einstellen der digitalen Ein- und Ausgänge des RFID-Readers als Verbindungsziel	7
4.3	Einstellen der Verbindungsparameter im Kommunikationsmodul	8
5	Arbeiten mit der Bibliothek	9
5.1	Integration der Bibliothek in Unity Pro	9
5.2	Inhalt der Bibliothek	10
5.3	Verwenden der Bausteine im Anwenderprogramm	11
6	Der Funktionsbaustein DTE810_910expl im Detail	11
6.1	Funktionsbeschreibung	11
6.2	Arbeitsweise des Funktionsbausteins DTE810_910expl	13
6.2.1	Kommunikationsaufbau zum RFID-Reader	13
6.2.2	Ausführen einer Funktion	15
6.3	Beschreibung der Funktionen	17
6.3.1	Funktions-ID 1 – EPC aller Tags lesen	18
6.3.2	Funktions-ID 2 – TID-Speicher aller Tags lesen	19
6.3.3	Funktions-ID 3 – TID-Speicher eines bestimmten Tags lesen	20
6.3.4	Funktions-ID 4 – Bereich des USER-Speichers von allen Tags lesen	21
6.3.5	Funktions-ID 5 – Bereich des USER-Speichers von einem bestimmten Tag lesen	23
6.3.6	Funktions-ID 6 – Bereich des USER-Speichers aller Tags beschreiben	24
6.3.7	Funktions-ID 7 – Bereich des USER-Speichers eines Tags beschreiben	26
6.3.8	Funktions-ID 8 – EPC auf einen unbekannten Tag schreiben	27
6.3.9	Funktions-ID 9 – EPC eines Tags ändern	29
6.3.10	Funktions-ID 15 – Antennenleistung einstellen	30
6.4	Abbruch und Aufbau der Kommunikation mit dem RFID-Reader	31
6.5	Statusanzeigen am Ausgang owStatus	32
7	Fehlercodes	33
7.1	Fehlercodes am Ausgang owStatus	33
7.2	Tagdaten-Fehlercodes	33
7.3	Tagzugriff-Fehlercodes	34
8	Beschreibung der abgeleiteten Datentypen	34
8.1	Arrays	34
8.2	Strukturen	34

1 Übersicht

Die Unity Pro Bibliothek DTE810 / DTE910 beinhaltet ein Codepaket für eine Schneider Electric Modicon M340 Steuerung, um im Anwenderprogramm der SPS den Datenaustausch mit einem RFID-Reader DTE810 oder DTE910 der Firma ifm electronic zu vereinfachen.

Folgende Funktionen werden von den Bausteinen der Bibliothek im Anwenderprogramm bereitgestellt:

- Lesen des EPC von allen Tags im Antennenfeld
- Schreiben eines EPC auf einen Tag
- Lesen des TID-Speichers von allen Tags im Antennenfeld oder von einem ausgewählten Tag
- Lesen eines Bereichs des USER-Speichers von allen Tags im Antennenfeld oder von einem ausgewählten Tag
- Beschreiben eines Bereichs des USER-Speichers aller Tags im Antennenfeld oder eines ausgewählten Tags
- Einstellen der Antennenleistung

Dieses Dokument erläutert das Einrichten der Verbindung zwischen Steuerung und RFID-Reader, und beschreibt die Einbindung und Anwendung der Bibliothek im Unity-Pro Projekt.

2 Haftungsausschluss

Das Codepaket der Bibliothek ist eine Demoversion. Es ist nur dazu gedacht, dem Benutzer als Beispiel zu dienen. Eine etwaige Anwendung dieses Codepakets zur Maschinensteuerung erfolgt auf eigene Gefahr! Die Autoren und Rechtsinhaber dieses Codepakets schließen jegliche Haftung für die Funktionstüchtigkeit oder Kompatibilität dieser Demoversion aus.

Aus urheberrechtlichen Gründen ist nur die kostenlose Weitergabe des Codepakets erlaubt!

3 Einstellen der Verbindungsparameter im RFID-Reader

Zum Einstellen der Verbindungsparameter im RFID-Reader wird die Software ReaderStart Version 2.30 oder höher benötigt.

- am PC die Software ReaderStart starten und eine Verbindung zum RFID-Reader herstellen

- auf die Schaltfläche [v] mit der nebenstehenden Bezeichnung *Kommunikationskonfiguration Intelligenzmodul* klicken, um die Port-Einstellungen zu öffnen

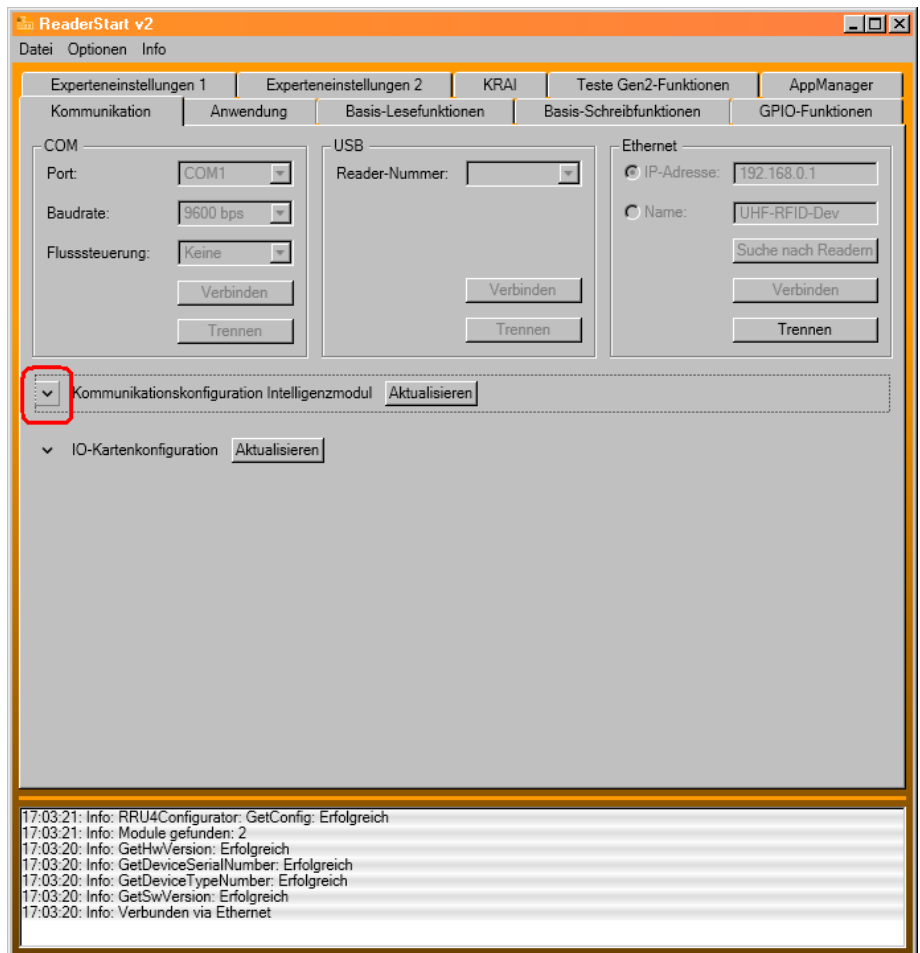


Abbildung 1

- im Bereich *Port-Einstellungen* die Parameter des Kommunikationsports einstellen
- zum Übernehmen der Einstellungen in den RFID-Reader auf [Änderungen speichern] klicken

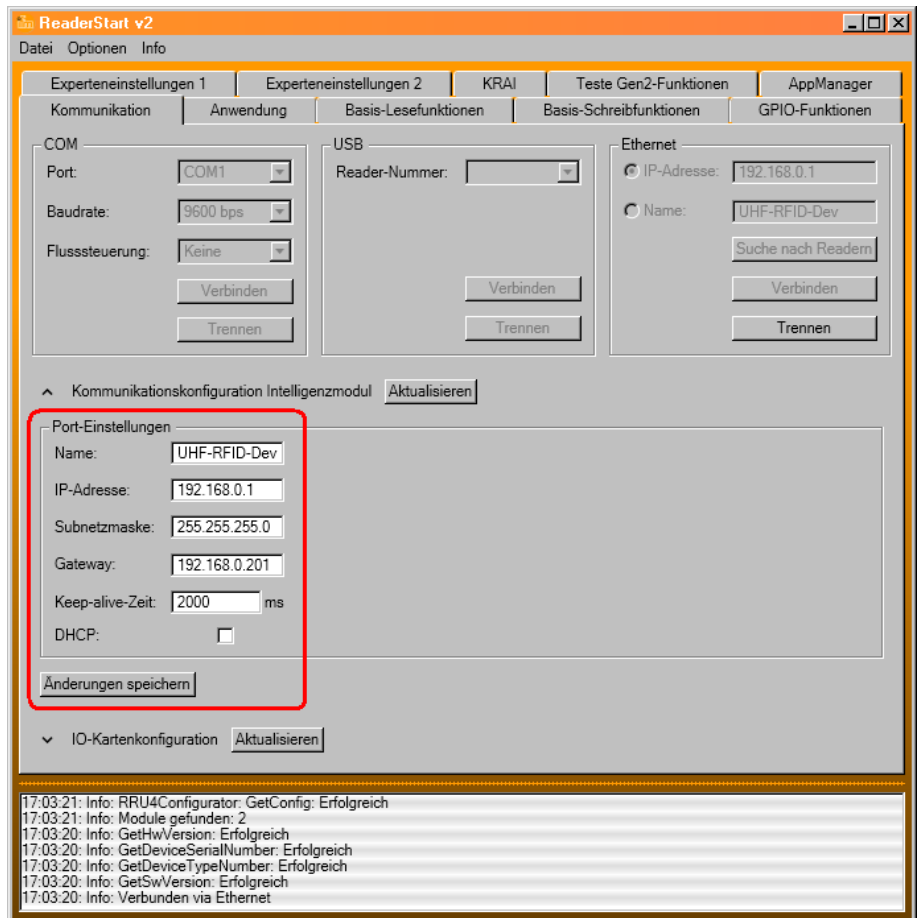


Abbildung 2

4 Projektieren der Verbindung in Unity Pro

Die Kommunikation zwischen Steuerung und RFID-Reader erfolgt über Ethernet/IP durch ein Kommunikationsmodul NOC0401 in der M340-Steuerung. Der Funktionsbaustein der Bibliothek DTE810 / DTE910 benutzt den azyklischen Dienst „Explicit Messaging“ zum Datenaustausch mit dem RFID-Reader. Dazu ist keine Projektierung einer Verbindung notwendig.

Dennoch ist es empfehlenswert, eine zyklische Verbindung zum RFID-Reader zu projektieren. Dadurch kann der Status der Ethernet/IP-Verbindung zum RFID-Reader einfach ermittelt werden, indem das von der Kommunikationsbaugruppe bereitgestellte Health-Bit der Verbindung ausgewertet wird.

Eine zyklische Verbindung belegt jedoch die zugeordneten Assemblies, sodass kein azyklischer Zugriff darauf mehr möglich ist. Deshalb wird empfohlen, in der SPS eine zyklische Verbindung zu den digitalen Ein- und Ausgängen des RFID-Readers zu projektieren. Auf diese Weise bleiben alle Assemblies der Antennenports für den azyklischen Dienst verfügbar.

Die digitalen Ein- und Ausgänge des RFID-Readers belegen 2 Bytes im Eingangs- und im Ausgangsbereich des Kommunikationsmoduls NOC0401. Dies ist bei der Konfiguration des Adressbereiches des Kommunikationsmoduls zu berücksichtigen.

4.1 Einfügen des RFID-Readers in das Softwareprojekt

- im DTM-Browser mit der rechten Maustaste auf das Kommunikationsmodul klicken
- aus dem Kontextmenü *Hinzufügen...* wählen



Abbildung 3

- aus der Liste der Geräte das *Generic Device* von *Schneider Electric* auswählen
- auf [DTM hinzufügen] klicken, um das Fenster *Eigenschaften des Geräts* zu öffnen

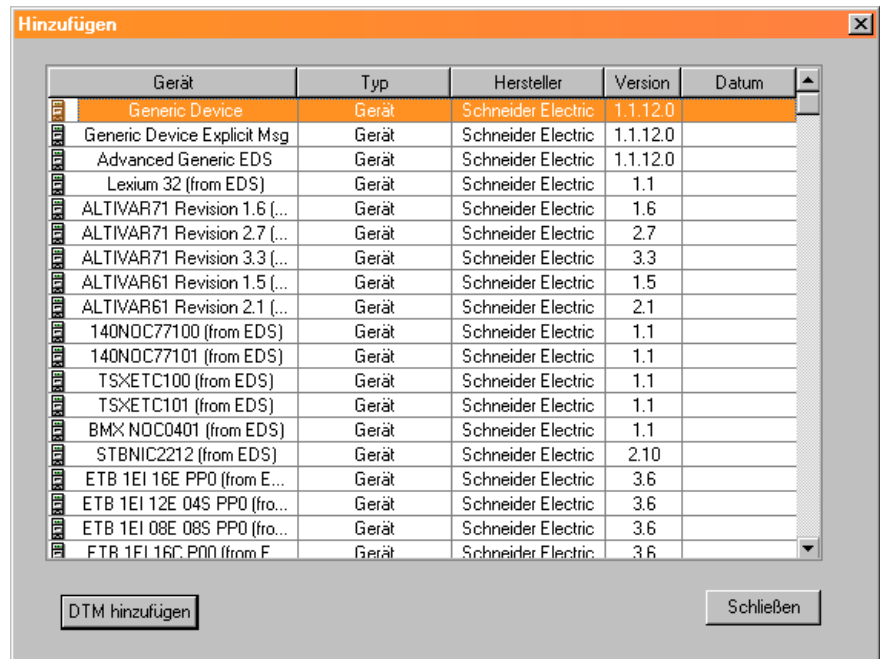


Abbildung 4

- im Tab *General* dem Gerät einen eindeutigen Alias-Namen geben

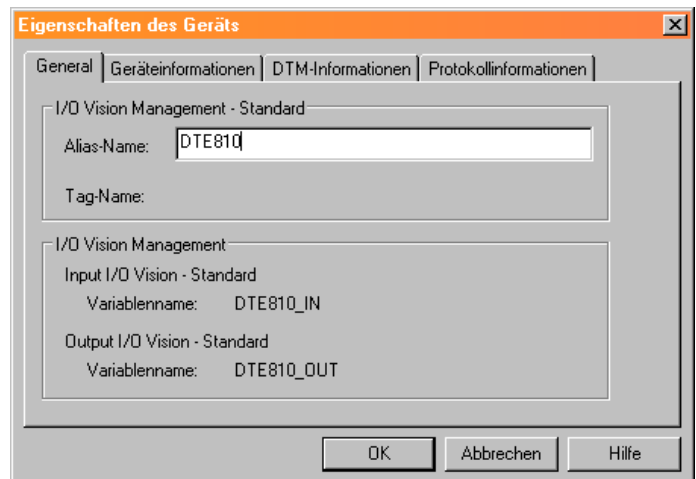


Abbildung 5

- im Tab *Protokollinformation* das erforderliche Protokoll CIP (EtherNet/IP) einstellen
- den Dialog *Eigenschaften des Geräts* mit Klick auf [OK] verlassen, um die Einstellungen zu übernehmen

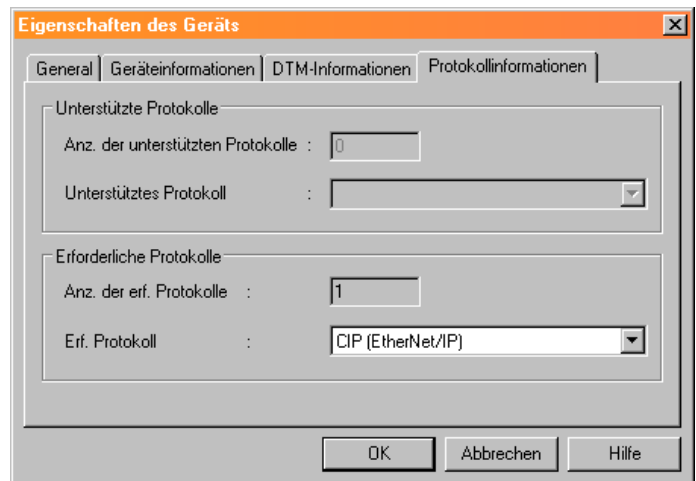


Abbildung 6

4.2 Einstellen der digitalen Ein- und Ausgänge des RFID-Readers als Verbindungsziel

- Im DTM-Browser mit der rechten Maustaste auf das hinzugefügte Gerät klicken
- aus dem Kontextmenü *Öffnen* wählen, um das Fenster *fdtConfiguration* zu öffnen

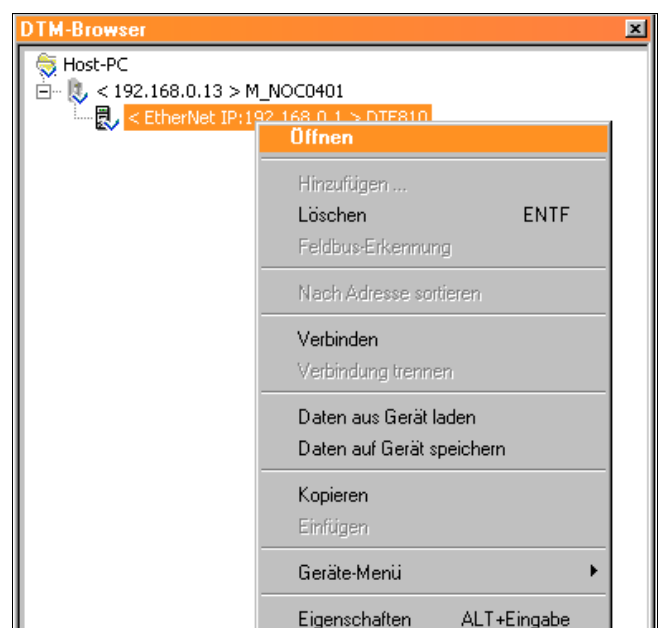


Abbildung 7

- in der Struktur auf *Exclusive Owner* klicken
- im Tab *Allgemein* die Kommunikations-Assemblies für den zyklischen Datenaustausch einstellen
- folgende Werte einstellen:
Eingangsgröße = 2
Input Instance = 101
Ausgangsgröße = 2
Output Instance = 100
- in den Tabs *Identitätsprüfung* und *Konfigurationseinstellungen* sind keine Änderungen notwendig
- mit Klick auf [OK] die Daten übernehmen

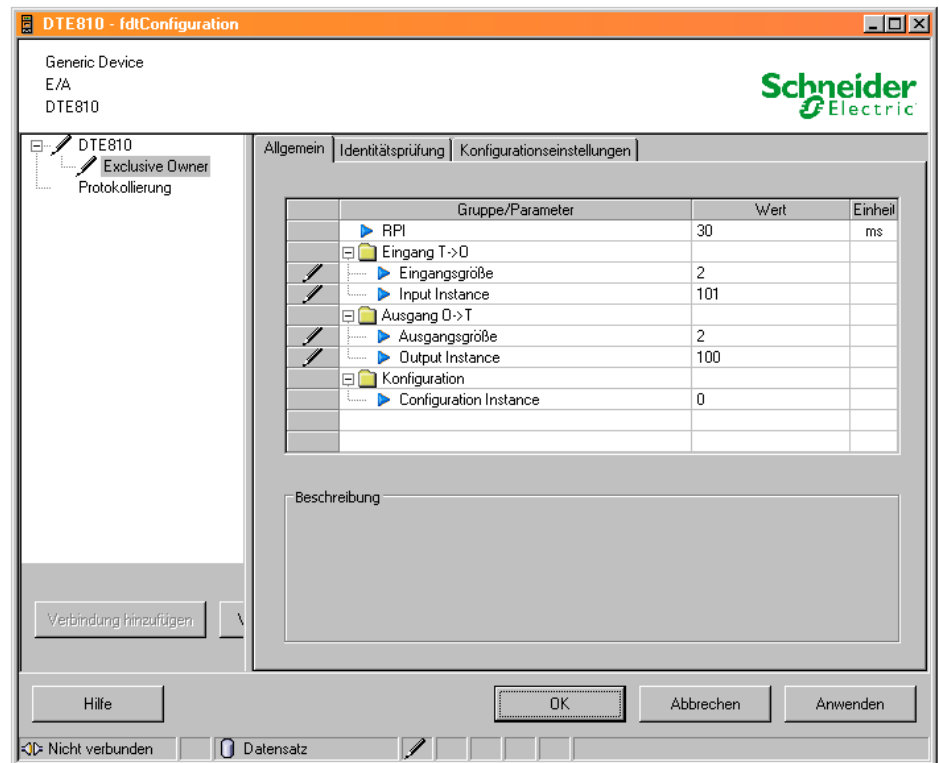


Abbildung 8

4.3 Einstellen der Verbindungsparameter im Kommunikationsmodul

- im DTM-Browser mit der rechten Maustaste auf das Kommunikationsmodul klicken
- aus dem Kontextmenü *Öffnen* wählen

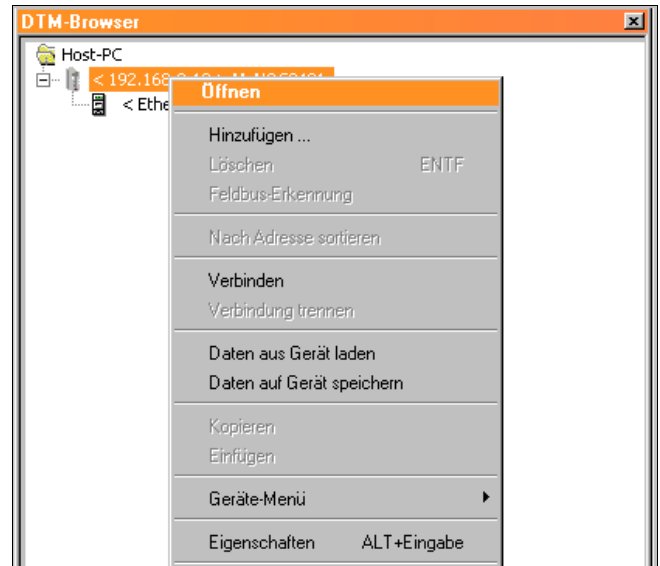


Abbildung 9

- in der Struktur den Eintrag *Geräteliste* erweitern und den RFID-Reader auswählen
- im Tab *Adresseinstellungen* die im RFID-Reader eingestellten Werte eingeben
- mit Klick auf [OK] die Einstellungen übernehmen

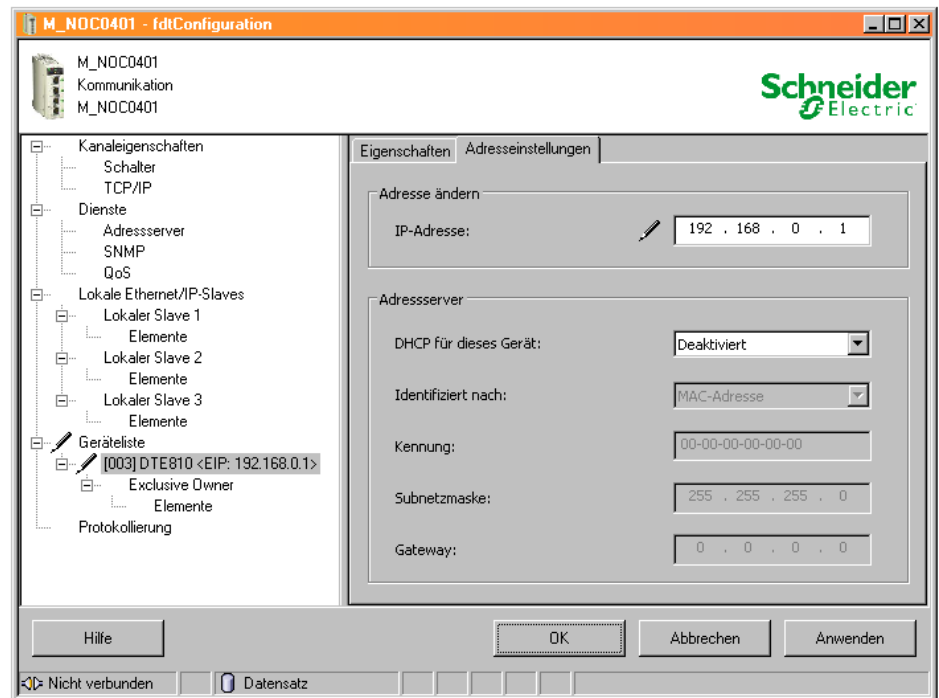


Abbildung 10

5 Arbeiten mit der Bibliothek

5.1 Integration der Bibliothek in Unity Pro

Die Bibliothek DTE810 / DTE910 wird als Archiv-Datei geliefert. Nach dem Entpacken dieses Archivs kann die Bibliothek mit dem Tool zur Typlibibliothek-Aktualisierung aus dem Unity Software Paket in die Typlibibliothek von Unity Pro integriert werden.

- das Tool zur Typlibibliothek-Aktualisierung starten
- als Quelle das Verzeichnis mit der Bibliothek einstellen
- auf [Familie installieren] klicken
- nach der Installation auf [Beenden] klicken, um das Tool zu verlassen

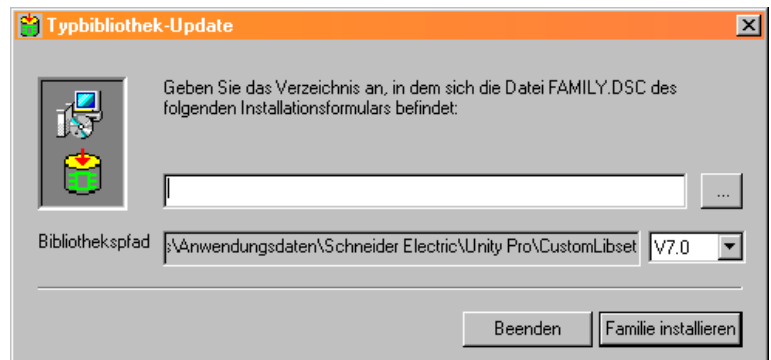


Abbildung 11

Die Bausteine werden in der Bibliothek ifm electronic in die Familie DTE810 / DTE910 eingeordnet.

<Root>	
<Libset V7.0>	
Base Lib	This is the main manufacturer library. It contains the main IEC families and some o
Communication	This library is used for exchanging data between devices connected by a bus or r
CONT_CTL	This library is for projecting process-engineering servo-loops. It contains Condition
Custom Lib	This Library is used to store Customer DFBs and DDTs.
Diagnostics	This library is used to investigate the control program for misbehaviours. It contain
I/O Management	This library contains EFBs, which are required for using I/O modules. It contains th
ifm electronic	This library contains DFBs and DDTs for devices made by ifm electronic.
DTE810 / DTE910	This library contains DFBs and DDTs for the RFID reader DTE810 and DTE910.
MotionFunctionBlock	This library contains Motion Function Blocks.
Motion	This Library contains the Axis Control, CAM control, Lexium and MMF Start Familie
Obsolete Lib	This library is used for compatibility with PL7 or Concept applications. It contains C
Safety	This library contains safety EF/EFBs for the usage in safety applications.
System	This library contains Events, SFC Management, SysClock Families.
UnitLib.L984	LL984 Libraw for Unity Pro. Contains FFBs for exclusive usage in LL984 editor onl

Abbildung 12

5.2 Inhalt der Bibliothek

Die Bibliothek DTE810 / DTE910 besteht aus dem zentralen Funktionsbaustein DTE810_910expl und einer Gruppe von speziellen Datentypen, die von dem Funktionsbaustein benötigt werden (Tabelle 1).

Name	Typ	Beschreibung
DTE810_910expl	DFB	zentraler Code-Baustein der Bibliothek
Dte810arstTags	ARRAY[1..n] OF Dte810stTagData	Array zur Aufnahme der Daten von den bearbeiteten Tags Das Array muss mit Index 1 beginnen. Der Endindex n definiert wie viele Tags maximal vom DFB bearbeitet werden können. Er kann für die Anwendung passend eingestellt werden.
Dte810arwTagEpc	ARRAY[0..n] OF WORD	Array zur Aufnahme des EPC eines Tags Das Array muss mit Index 0 beginnen. Der Endindex n definiert die maximale Länge eines EPC, die der DFB verarbeiten kann. Er kann für die Anwendung passend eingestellt werden.
Dte810arwTagMemory	ARRAY[0..n] OF WORD	Array zur Aufnahme der TID- oder USER-Speicher-Daten eines Tags Das Array muss mit Index 0 beginnen. Der Endindex n definiert die maximale Speichergröße, die der DFB verarbeiten kann. Er kann für die Anwendung passend eingestellt werden.
Dte810stAntPortPower	Strukt	Daten zur Einstellung der Antennenleistung
Dte810stDiagData	Strukt	detaillierte Fehlerinformationen
Dte810stDteSet	Strukt	Einstellungen der Verbindung zum RFID-Reader
Dte810stNocSet	Strukt	Lokalisierung der Kommunikationsbaugruppe NOC 0401 in der SPS
Dte810stReadEpc	Strukt	Parameter zum Lesen aller EPCs
Dte810stReadUserData	Strukt	Parameter zum Lesen des USER-Speichers
Dte810stSettings	Strukt	Daten zum Aufbau einer Verbindung zum RFID-Reader
Dte810stTagData	Strukt	Daten, die von einem Tag gelesen wurden
Dte810stTagEpcData	Strukt	Daten eines EPC
Dte810stTagMemoryData	Strukt	TID- bzw. USER-Daten
Dte810stWriteUserData	Strukt	Daten zum Beschreiben des USER-Speichers des Tag

Tabelle 1

Die Strukturen sind im Kapitel 8.2 näher beschrieben. Die Bedeutungen der einzelnen Elemente der Strukturen werden bei deren Verwendung in den Kapiteln 6.2 und 6.3 erläutert.

5.3 Verwenden der Bausteine im Anwenderprogramm

Wird eine Instanz des Funktionsbausteins DTE810_910expl aus der Bibliothek heraus im Anwenderprogramm erzeugt, so werden auch die verwendeten Datentypen automatisch im Anwenderprogramm angelegt.

Es ist nur eine Instanz des Funktionsbausteins pro Antennenportnummer eines RFID-Readers zulässig. Alle verwendeten Instanzen müssen im zyklischen Programm der SPS einmal pro Zyklus aufgerufen werden.

Die Größe des benötigten Speichers einer Instanz im Anwenderprogramm wird durch die Größen der Arrays *Dte810arstTags*, *Dte810arwTagEpc* und *Dte810arwTagMemory* beeinflusst. Diese können den Erfordernissen der Applikation angepasst werden. Dabei darf jeweils nur der Endindex des Arrays verändert werden, der Anfangsindex und der Datentyp müssen auf den initialen Werten bleiben.

Ein Beispiel soll das verdeutlichen:

- Es sind maximal 7 Tags gleichzeitig zu erwarten.
→ Deklaration *Dte810arstTags*: ARRAY[1..7] OF *Dte810stTagData*
- Der längste EPC ist 6 Worte lang.
→ Deklaration *Dte810arwTagEpc*: ARRAY[0..5] OF WORD
- Es sollen maximal 12 Worte Daten gelesen oder geschrieben werden.
→ Deklaration *Dte810arwTagMemory*: ARRAY[0..11] OF WORD

6 Der Funktionsbaustein DTE810_910expl im Detail

6.1 Funktionsbeschreibung

Die Kommunikation muss nach einem definierten Ablauf erfolgen. **Abbildung 13** verdeutlicht dies anhand eines Ablaufplans:

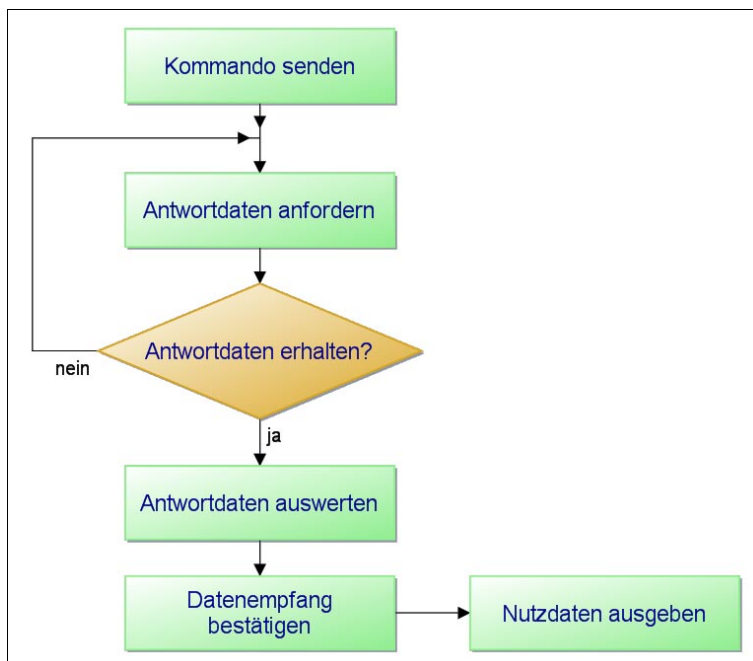


Abbildung 13

Um den Vorgang der Kommandoabwicklung für den Anwender komfortabler zu gestalten, wurde der Funktionsbaustein *DTE810_910expl* entwickelt. Dieser Baustein bearbeitet das oben aufgeführte Kommandoprotokoll selbständig.

Das folgende Bild zeigt den Funktionsbaustein DTE810_910expl:

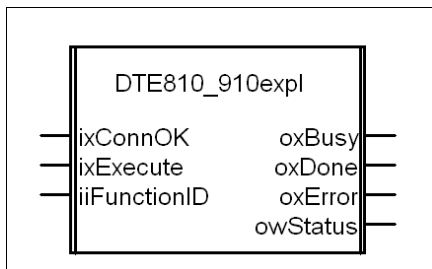


Abbildung 14

Die Parameter haben folgende Bedeutung:

Eingänge:

Name	Typ	Beschreibung
ixConnOK	BOOL	Verbindung Antenne - RFID-Reader - SPS ist hergestellt
ixExecute	BOOL	steigende Flanke aktiviert die an iiFunctionID eingestellte Funktion
iiFunctionID	INT	ID der Funktion, die ausgeführt werden soll (siehe Tabelle 6)

Tabelle 2

Ausgänge:

Name	Typ	Beschreibung
oxBusy	BOOL	Funktionsbaustein ist beschäftigt, kein neuer Befehl möglich
oxDone	BOOL	Funktion fehlerfrei beendet
oxError	BOOL	Fehler erkannt
owStatus	WORD	Status Information oder Fehlercode (siehe Tabelle 27)

Tabelle 3

Öffentliche Variablen:

Name	Typ	Beschreibung
stSettings	Dte810stSettings	Daten zur grundlegenden Einstellung der Verbindung zum RFID-Reader
stReadEpc	Dte810stReadEpc	Parameter für das Lesen der EPCs
stReadUserData	Dte810stReadUserData	Parameter für das Lesen des USER-Speichers der Tags
stSelectionEpc	Dte810stTagEpcData	EPC-Daten zur Auswahl eines bestimmten Tags
dwTagPassword	DWORD	Passwort für Tag-Zugriff
stEpcToWrite	Dte810stTagEpcData	Parameter für das Schreiben eines neuen EPC auf einen Tag
stUserDataToWrite	Dte810stWriteUserData	Parameter und Daten zum Schreiben in den USER-Speicher der Tags
stAntPortPower	Dte810stAntPortPower	Parameter für das Einstellen der Antennenleistung
stDiagData	Dte810stDiagData	Diagnosedaten
iTagCounter	INT	Anzahl der gefundenen Tags
arTag	Dte810arstTags	von den Tags gelesene Daten

Tabelle 4

6.2 Arbeitsweise des Funktionsbausteins DTE810_910expl

6.2.1 Kommunikationsaufbau zum RFID-Reader

Bevor die Kommunikation mit dem RFID-Reader gestartet werden kann, müssen alle grundlegenden Einstellungen und Parameter in die Struktur *stSettings* in den öffentlichen Variablen der Instanz eingetragen werden (Tabelle 5).

Name	Datentyp	Eingabewert
stSettings	Dte810stSettings	Daten zum Aufbau einer Verbindung zum RFID-Reader
stNOC	Dte810stNocSet	Lokalisierung der Kommunikationsbaugruppe NOC0401 in der SPS
iRackNo	INT	Rack-Nummer, in der die Kommunikationsbaugruppe montiert ist
iSlotNo	INT	Slot-Nummer, in der die Kommunikationsbaugruppe montiert ist
iChannelNo	INT	Kanal-Nummer der Kommunikationsbaugruppe
stDTE	Dte810stDteSet	Einstellungen der Verbindung zum RFID-Reader
arilPAddr	ARRAY[1..4] OF INT	IP-Adresse des RFID-Readers Beispiel: IP-Adresse = 192.168.0.1 arilPAddr[1] := 192 arilPAddr[2] := 168 arilPAddr[3] := 0 arilPAddr[4] := 1
iAntPortNo	INT	= 0 Datenaustausch mit allen aktivierten Antennenports 1 .. 4 Datenaustausch mit Antennenport 1 .. 4
iTimeOutVal	INT	= 0 interne Funktion DATA_EXCH verwendet Standard-Überwachungszeit > 0 Überwachungszeit der internen Funktion DATA_EXCH (Wert * 100ms)
iRepeatTime	INT	Verzögerungszeit zum Wiederholen des Befehls, wenn der RFID-Reader keine Daten liefert (Empfehlung = 200 ms)

Tabelle 5

Das Setzen des Eingangs *ixConnOK* = TRUE aktiviert die Kommunikation mit dem RFID-Reader.

Die Werte der Struktur *stSettings* in den öffentlichen Variablen werden übernommen.

In den öffentlichen Variablen werden die Daten im Array *arTag* und der Tagzähler *iTagCounter* mit dem Wert 0 belegt.

Der Funktionsbaustein fordert zyklisch Daten vom RFID-Reader an. Konnten Daten empfangen werden, synchronisiert der Funktionsbaustein den Datenaustausch.

Nach Abschluss wird der Ausgang *oxBusy* = FALSE ausgegeben. Der Funktionsbaustein ist nun bereit, eine Funktion auszuführen. Der Signalverlauf ist in Abbildung 15 dargestellt.

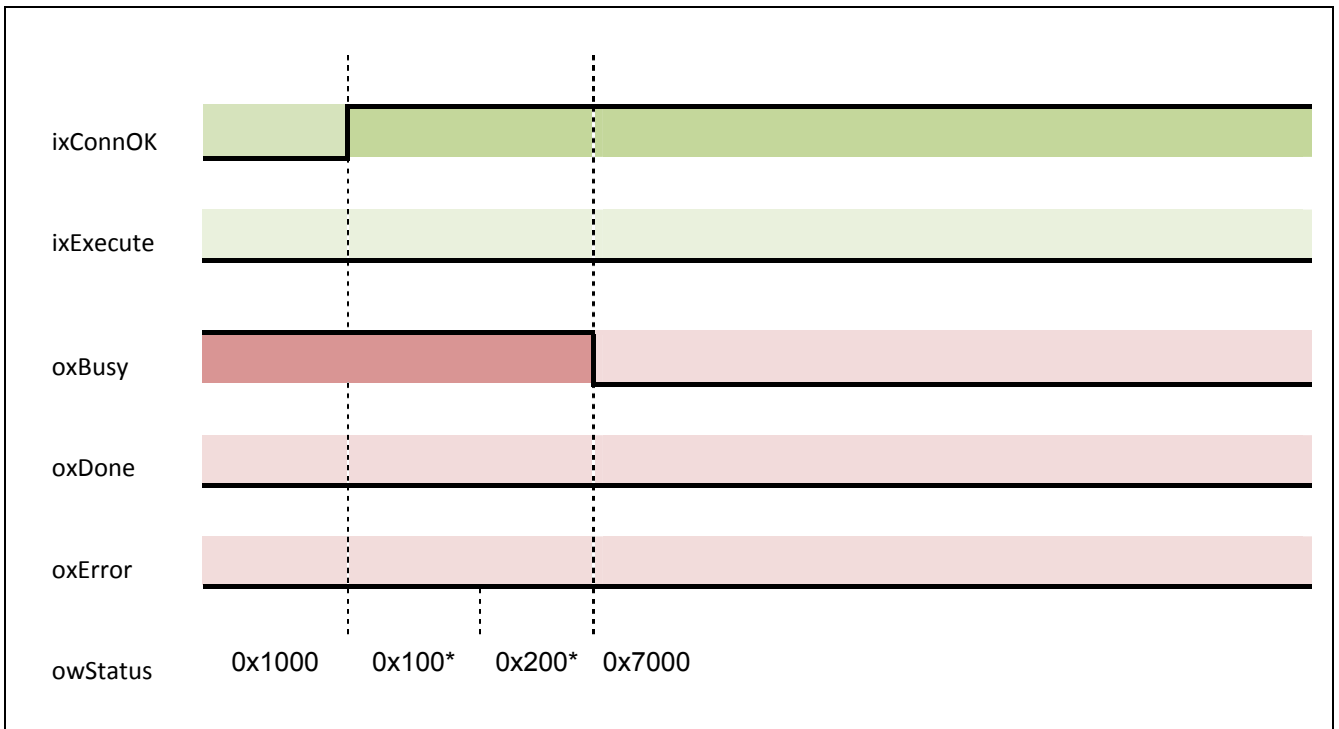


Abbildung 15

6.2.2 Ausführen einer Funktion

Der Funktionsbaustein DTE810_910expl ist ein asynchron arbeitender Funktionsbaustein, das bedeutet die Bearbeitung erstreckt sich über mehrere FB-Aufrufe.

Eine auszuführende Funktion wird durch Angabe der Funktions-ID am Parameter *iiFunctionID* ausgewählt. Die benötigten Parameter und Daten der Funktion müssen in die entsprechende Struktur in den öffentlichen Daten des Funktionsbausteins eingetragen werden. Gestartet wird die Funktion mit einer steigenden Flanke am Eingang *ixExecute*.

Über die Ausgangsparameter *oxBusy* und *owStatus* wird der Zustand des Auftrags angezeigt. Während einer Auftragsbearbeitung ist der Parameter *oxBusy* gesetzt. Der Parameter *owStatus* zeigt den Fortschritt der Auftragsbearbeitung. Mit Abschluss des Auftrags wird das Ergebnis in den Parametern *oxDone* und *oxError* mitgeteilt und bleibt erhalten, solange der Eingang *ixExecute* = TRUE ist, mindestens jedoch einen SPS-Zyklus.

Ist kein Fehler aufgetreten, wird *oxDone* gesetzt und im Parameter *owStatus* der Wert 0x0000 eingetragen. Bei Aufträgen mit Antwortdaten von den Tags werden diese Daten im Array *arTag* in den öffentlichen Variablen zur Verfügung gestellt. Die öffentliche Variable *iTagCounter* zeigt an, wieviele Tags von der Funktion erkannt oder beeinflusst wurden.

Tritt während der Auftragsbearbeitung ein Fehler auf, wird *oxError* gesetzt. Zur näheren Beschreibung des aufgetretenen Fehlers wird im Parameter *owStatus* ein Fehlercode angezeigt. Die möglichen Fehlercodes sind in Tabelle 28 aufgelistet.

Der Signalverlauf ist in Abbildung 16 dargestellt.

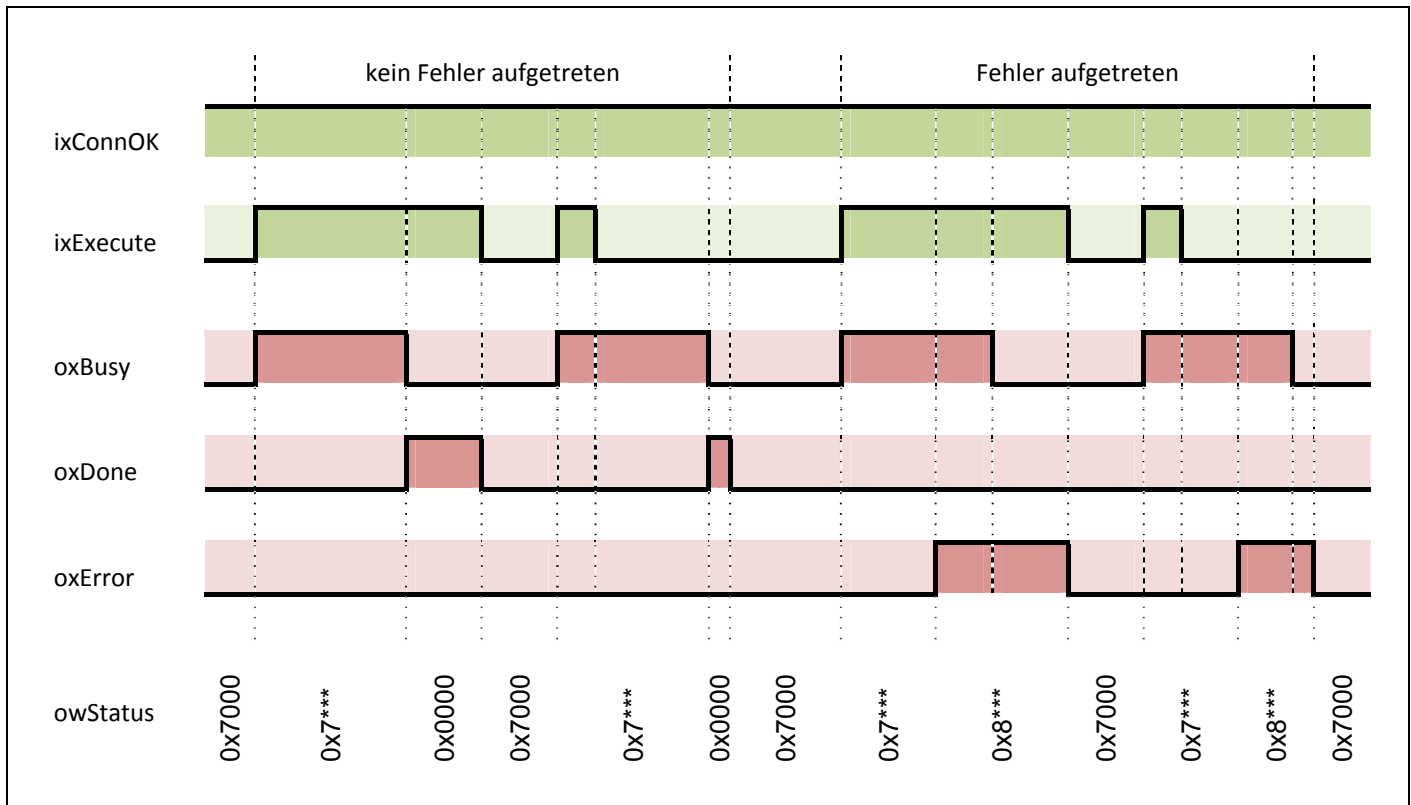


Abbildung 16

6.3 Beschreibung der Funktionen

Funktion	ID (dezimal)	Erforderliche Parameter	Rückgaben	Beschreibung
EPC aller Tags lesen	1	stReadEpc	iTagCounter, arTag	siehe Kapitel 6.3.1
TID-Speicher aller Tags lesen	2	dwTagPassword	iTagCounter, arTag	siehe Kapitel 6.3.2
TID-Speicher eines Tags lesen	3	dwTagPassword, stSelectionEpc	iTagCounter, arTag	siehe Kapitel 6.3.3
Bereich des USER-Speichers von allen Tags lesen	4	dwTagPassword, stReadUserData	iTagCounter, arTag	siehe Kapitel 6.3.4
Bereich des USER-Speichers von einem Tag lesen	5	dwTagPassword, stSelectionEpc, stReadUserData	iTagCounter, arTag	siehe Kapitel 6.3.5
Bereich des USER-Speichers aller Tags beschreiben	6	dwTagPassword, stUserDataToWrite	iTagCounter, arTag	siehe Kapitel 6.3.6
Bereich des USER-Speichers eines Tags beschreiben	7	dwTagPassword, stSelectionEpc, stUserDataToWrite	iTagCounter, arTag	siehe Kapitel 6.3.7
EPC auf einen unbekannten Tag schreiben	8	dwTagPassword, stEpcToWrite	iTagCounter, arTag	siehe Kapitel 6.3.8
EPC eines Tags ändern	9	dwTagPassword, stSelectionEpc, stEpcToWrite	iTagCounter, arTag	siehe Kapitel 6.3.9
Antennenleistung einstellen	15	stAntPortPower		siehe Kapitel 6.3.10

Tabelle 6

6.3.1 Funktions-ID 1 – EPC aller Tags lesen

Diese Funktion erstellt eine Liste mit den EPCs aller Tags, die während der eingestellten Lesezeit erfasst wurden, unabhängig ob sich die Tags bei Beendigung des Lesens noch im Antennenfeld befinden oder nicht.

Erforderliche Parameter in den öffentlichen Variablen:

Name	Datentyp	Eingabewert
stReadEpc	Dte810stReadEpc	Parameter zum Lesen aller EPCs
iReadingTime	INT	= 0 einmaliges Lesen der EPCs aller Tags > 0 Aktivierungszeit (in ms) für das Lesen der EPCs aller Tags

Tabelle 7

Rückgaben in den öffentlichen Variablen:

Name	Datentyp	Rückgabewert
iTagCounter	INT	Anzahl der bearbeiteten Tags
arTag[1]	DTE810stTagData	von Tag 1 gelesene Daten
bDataErrorCode	BYTE	Tagdaten-Fehlercode (siehe Tabelle 29)
bRssi	BYTE	RSSI-Wert (Stärke der Kopplung Tag – Antenne)
stEPC	DTE810stTagEPCData	gelesene EPC-Daten
iLength	INT	Anzahl der EPC-Datenworte
arwEpcWord[0]	WORD	gelesenes EPC-Datenwort 0
arwEpcWord[...]	WORD	gelesenes EPC-Datenwort ...
arwEpcWord[n]	WORD	gelesenes EPC-Datenwort n
bTagErrorCode	BYTE	0x00
stMemoryData	Dte810stTagMemoryData	0x0000
arTag[...]	DTE810stTagData	von Tag ... gelesene Daten
arTag[n]	DTE810stTagData	von Tag n gelesene Daten

Tabelle 8

6.3.2 Funktions-ID 2 – TID-Speicher aller Tags lesen

Diese Funktion liest die TID-Speicher aller Tags im Antennenfeld, deren Zugriffs-Passwort dem angegebenen Passwort entspricht. Standardmäßig ist auf den Tags als Passwort 0x0000 0000 eingestellt.

Erforderliche Parameter in den öffentlichen Variablen:

Name	Datentyp	Eingabewert
dwPassword	DWORD	Tagzugriff-Passwort (auf Tag gespeichert)

Tabelle 9

Rückgaben in den öffentlichen Variablen:

Name	Datentyp	Rückgabewert
iTagCounter	INT	Anzahl der bearbeiteten Tags
arTag[1]	DTE810stTagData	von Tag 1 gelesene Daten
bDataErrorCode	BYTE	Tagdaten-Fehlercode (siehe Tabelle 29)
bRssi	BYTE	RSSI-Wert (Stärke der Kopplung Tag – Antenne)
stEPC	DTE810stTagEPCData	gelesene EPC-Daten
iLength	INT	Anzahl der EPC-Datenworte
arwEpcWord[0]	WORD	gelesenes EPC-Datenwort 0
arwEpcWord[...]	WORD	gelesenes EPC-Datenwort ...
arwEpcWord[n]	WORD	gelesenes EPC-Datenwort n
bTagErrorCode	BYTE	Tagzugriff-Fehlercode (siehe Tabelle 30)
stMemoryData	Dte810stTagMemoryData	gelesene TID-Daten
iLength	INT	Anzahl der TID-Datenworte
arwDataWord[0]	WORD	gelesenes TID-Datenwort 0
arwDataWord[...]	WORD	gelesenes TID-Datenwort ...
arwDataWord[n]	WORD	gelesenes TID-Datenwort n
arTag[...]	DTE810stTagData	von Tag ... gelesene Daten
arTag[n]	DTE810stTagData	von Tag n gelesene Daten

Tabelle 10

6.3.3 Funktions-ID 3 – TID-Speicher eines bestimmten Tags lesen

Diese Funktion liest den TID-Speicher von einem bestimmten Tag. Der zu lesende Tag wird über den EPC ausgewählt. Ebenso ist das Passwort des Tags anzugeben. Standardmäßig ist auf dem Tag als Passwort 0x0000 0000 eingestellt.

Erforderliche Parameter in den öffentlichen Variablen:

Name	Datentyp	Eingabewert
dwPassword	DWORD	Tagzugriff-Passwort (auf Tag gespeichert)
stSelectionEPC	DTE810stTagEPCData	EPC-Daten zur Auswahl eines Tags
iLength	INT	Anzahl der EPC-Datenworte des gewählten Tags
arwEpcWord[0]	WORD	EPC-Datenwort 0 des gewählten Tags
arwEpcWord[...]	WORD	EPC-Datenwort ... des gewählten Tags
arwEpcWord[n]	WORD	EPC-Datenwort n des gewählten Tags

Tabelle 11

Rückgaben in den öffentlichen Variablen:

Name	Datentyp	Rückgabewert
iTagCounter	INT	0 .. 1 Anzahl der bearbeiteten Tags
arTag[1]	DTE810stTagData	von Tag 1 gelesene Daten
bDataErrorCode	BYTE	Tagdaten-Fehlercode (siehe Tabelle 29)
bRssi	BYTE	RSSI-Wert (Stärke der Kopplung Tag – Antenne)
stEPC	DTE810stTagEPData	gelesene EPC-Daten
iLength	INT	Anzahl der EPC-Datenworte
arwEpcWord[0]	WORD	gelesenes EPC-Datenwort 0
arwEpcWord[...]	WORD	gelesenes EPC-Datenwort ...
arwEpcWord[n]	WORD	gelesenes EPC-Datenwort n
bTagErrorCode	BYTE	Tagzugriff-Fehlercode (siehe Tabelle 30)
stMemoryData	Dte810stTagMemoryData	gelesene TID-Daten
iLength	INT	Anzahl der TID-Datenworte
arwDataWord[0]	WORD	gelesenes TID-Datenwort 0
arwDataWord[...]	WORD	gelesenes TID-Datenwort ...
arwDataWord[n]	WORD	gelesenes TID-Datenwort n
arTag[...]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Tabelle 12

6.3.4 Funktions-ID 4 – Bereich des USER-Speichers von allen Tags lesen

Diese Funktion liest einen anzugebenden Bereich des USER-Speichers von allen Tags im Antennenfeld, deren Zugriffs-Passwort dem angegebenen Passwort entspricht. Standardmäßig ist auf den Tags als Passwort 0x0000 0000 eingestellt.

Erforderliche Parameter in den öffentlichen Variablen:

Name	Datentyp	Eingabewert
dwPassword	DWORD	Tagzugriff-Passwort (auf Tag gespeichert)
stReadUserData	DTE810stReadUserData	Parameter zum Lesen des USER-Speichers
diStartAddress	DINT	Speicheradresse, ab der gelesen werden soll
iLength	INT	Anzahl der zu lesenden Datenworte

Tabelle 13

Rückgaben in den öffentlichen Variablen:

Name	Datentyp	Rückgabewert
iTagCounter	INT	Anzahl der bearbeiteten Tags
arTag[1]	DTE810stTagData	von Tag 1 gelesene Daten
bDataErrorCode	BYTE	Tagdaten-Fehlercode (siehe Tabelle 29)
bRssi	BYTE	RSSI-Wert (Stärke der Kopplung Tag – Antenne)
stEPC	DTE810stTagEPCData	gelesene EPC-Daten
iLength	INT	Anzahl der EPC-Datenworte
arwEpcWord[0]	WORD	gelesenes EPC-Datenwort 0
arwEpcWord[...]	WORD	gelesenes EPC-Datenwort ...
arwEpcWord[n]	WORD	gelesenes EPC-Datenwort n
bTagErrorCode	BYTE	Tagzugriff-Fehlercode (siehe Tabelle 30)
stMemoryData	Dte810stTagMemoryData	gelesene USER-Speicher-Daten
iLength	INT	Anzahl der USER-Speicher-Datenworte
arwDataWord[0]	WORD	gelesenes USER-Speicher-Datenwort 0
arwDataWord[...]	WORD	gelesenes USER-Speicher-Datenwort ...
arwDataWord[n]	WORD	gelesenes USER-Speicher-Datenwort n
arTag[...]	DTE810stTagData	von Tag ... gelesene Daten
arTag[n]	DTE810stTagData	von Tag n gelesene Daten

Tabelle 14

6.3.5 Funktions-ID 5 – Bereich des USER-Speichers von einem bestimmten Tag lesen

Diese Funktion liest einen anzugebenden Bereich des USER-Speichers von einem bestimmten Tag. Der zu lesende Tag wird über den EPC ausgewählt. Ebenso ist das Passwort des Tags anzugeben. Standardmäßig ist auf dem Tag als Passwort 0x0000 0000 eingestellt.

Erforderliche Parameter in den öffentlichen Variablen:

Name	Datentyp	Eingabewert
dwPassword	DWORD	Tagzugriff-Passwort (auf Tag gespeichert)
stSelectionEPC	DTE810stTagEPCData	EPC-Daten zur Auswahl eines Tags
iLength	INT	Anzahl der EPC-Datenworte des gewählten Tags
arwEpcWord[0]	WORD	EPC-Datenwort 0 des gewählten Tags
arwEpcWord[...]	WORD	EPC-Datenwort ... des gewählten Tags
arwEpcWord[n]	WORD	EPC-Datenwort n des gewählten Tags
stReadUserData	DTE810stReadUserData	Parameter zum Lesen des USER-Speichers
diStartAddress	DINT	Speicheradresse, ab der gelesen werden soll
iLength	INT	Anzahl der zu lesenden Datenworte

Tabelle 15

Rückgaben in den öffentlichen Variablen:

Name	Datentyp	Rückgabewert
iTagCounter	INT	0 .. 1 Anzahl der bearbeiteten Tags
arTag[1]	DTE810stTagData	von Tag 1 gelesene Daten
bDataErrorCode	BYTE	Tagdaten-Fehlercode (siehe Tabelle 29)
bRssi	BYTE	RSSI-Wert (Stärke der Kopplung Tag – Antenne)
stEPC	DTE810stTagEPCData	gelesene EPC-Daten
iLength	INT	Anzahl der EPC-Datenworte
arwEpcWord[0]	WORD	gelesenes EPC-Datenwort 0
arwEpcWord[...]	WORD	gelesenes EPC-Datenwort ...
arwEpcWord[n]	WORD	gelesenes EPC-Datenwort n
bTagErrorCode	BYTE	Tagzugriff-Fehlercode (siehe Tabelle 30)
stMemoryData	Dte810stTagMemoryData	gelesene USER-Speicher-Daten
iLength	INT	Anzahl der USER-Speicher-Datenworte
arwDataWord[0]	WORD	gelesenes USER-Speicher-Datenwort 0
arwDataWord[...]	WORD	gelesenes USER-Speicher-Datenwort ...
arwDataWord[n]	WORD	gelesenes USER-Speicher-Datenwort n
arTag[...]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Tabelle 16

6.3.6 Funktions-ID 6 – Bereich des USER-Speichers aller Tags beschreiben

Diese Funktion schreibt Daten in einen anzugebenden Bereich des USER-Speichers aller Tags im Antennenfeld, deren Zugriffs-Passwort dem angegebenen Passwort entspricht. Standardmäßig ist auf den Tags als Passwort 0x0000 0000 eingestellt.

Sind zu viele Tags im Antennenfeld, so werden keine Daten geschrieben und ein Fehler ausgegeben.

Erforderliche Parameter in den öffentlichen Variablen:

Name	Datentyp	Eingabewert
dwPassword	DWORD	Tagzugriff-Passwort (auf Tag gespeichert)
stUserDataToWrite	Dte810WriteUserData	Parameter zum Schreiben von USER-Speicher-Daten
diStartAddress	DINT	Speicheradresse, ab der geschrieben werden soll
iLength	INT	Anzahl der zu schreibenden Datenworte
arwDataWord[0]	WORD	zu schreibendes Datenwort 0
arwDataWord[...]	WORD	zu schreibendes Datenwort ...
arwDataWord[n]	WORD	zu schreibendes Datenwort n

Tabelle 17

Rückgaben in den öffentlichen Variablen:

Name	Datentyp	Rückgabewert
iTagCounter	INT	Anzahl der bearbeiteten Tags
arTag[1]	DTE810stTagData	von Tag 1 gelesene Daten
bDataErrorCode	BYTE	Tagdaten-Fehlercode (siehe Tabelle 29)
bRssi	BYTE	RSSI-Wert (Stärke der Kopplung Tag – Antenne)
stEPC	DTE810stTagEPCData	gelesene EPC-Daten
iLength	INT	Anzahl der EPC-Datenworte
arwEpcWord[0]	WORD	gelesenes EPC-Datenwort 0
arwEpcWord[...]	WORD	gelesenes EPC-Datenwort ...
arwEpcWord[n]	WORD	gelesenes EPC-Datenwort n
bTagErrorCode	BYTE	Tagzugriff-Fehlercode (siehe Tabelle 30)
stMemoryData	Dte810stTagMemoryData	0x0000
arTag[...]	DTE810stTagData	von Tag ... gelesene Daten
arTag[n]	DTE810stTagData	von Tag n gelesene Daten

Tabelle 18

6.3.7 Funktions-ID 7 – Bereich des USER-Speichers eines Tags beschreiben

Diese Funktion schreibt Daten in einen anzugebenden Bereich des USER-Speichers auf einem bestimmten Tag. Der zu beschreibende Tag wird über den EPC ausgewählt. Ebenso ist das Passwort des Tags anzugeben. Standardmäßig ist auf dem Tag als Passwort 0x0000 0000 eingestellt.

Erforderliche Parameter in den öffentlichen Variablen:

Name	Datentyp	Eingabewert
dwPassword	DWORD	Tagzugriff-Passwort (auf Tag gespeichert)
stSelectionEPC	DTE810stTagEPCData	EPC-Daten zur Auswahl eines Tags
iLength	INT	Anzahl der EPC-Datenworte des gewählten Tags
arwEpcWord[0]	WORD	EPC-Datenwort 0 des gewählten Tags
arwEpcWord[...]	WORD	EPC-Datenwort ... des gewählten Tags
arwEpcWord[n]	WORD	EPC-Datenwort n des gewählten Tags
stUserDataToWrite	Dte810WriteUserData	Parameter zum Schreiben von USER-Speicher-Daten
diStartAddress	DINT	Speicheradresse, ab der geschrieben werden soll
iLength	INT	Anzahl der zu schreibenden Datenworte
arwDataWord[0]	WORD	zu schreibendes Datenwort 0
arwDataWord[...]	WORD	zu schreibendes Datenwort ...
arwDataWord[n]	WORD	zu schreibendes Datenwort n

Tabelle 19

Rückgaben in den öffentlichen Variablen:

Name	Datentyp	Rückgabewert
iTagCounter	INT	0 .. 1 Anzahl der bearbeiteten Tags
arTag[1]	DTE810stTagData	von Tag 1 gelesene Daten
bDataErrorCode	BYTE	Tagdaten-Fehlercode (siehe Tabelle 29)
bRssi	BYTE	RSSI-Wert (Stärke der Kopplung Tag – Antenne)
stEPC	DTE810stTagEPCData	gelesene EPC-Daten
iLength	INT	Anzahl der EPC-Datenworte
arwEpcWord[0]	WORD	gelesenes EPC-Datenwort 0
arwEpcWord[...]	WORD	gelesenes EPC-Datenwort ...
arwEpcWord[n]	WORD	gelesenes EPC-Datenwort n
bTagErrorCode	BYTE	Tagzugriff-Fehlercode (siehe Tabelle 30)
stMemoryData	Dte810stTagMemoryData	0x0000
arTag[...]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Tabelle 20

6.3.8 Funktions-ID 8 – EPC auf einen unbekannten Tag schreiben

Diese Funktion schreibt den angegebenen EPC auf einen Tag. Dazu darf nicht mehr als ein Tag im Antennenfeld sein. Ebenso ist das Passwort des Tags anzugeben. Standardmäßig ist auf dem Tag als Passwort 0x0000 0000 eingestellt.

Erforderliche Parameter in den öffentlichen Variablen:

Name	Datentyp	Eingabewert
dwPassword	DWORD	Tagzugriff-Passwort (auf Tag gespeichert)
stEpcToWrite	Dte810TagEpcData	zu schreibende EPC-Daten
iLength	INT	Anzahl der zu schreibenden EPC-Datenworte
arwEpcWord[0]	WORD	zu schreibendes EPC-Datenwort 0
arwEpcWord[...]	WORD	zu schreibendes EPC-Datenwort ...
arwEpcWord[n]	WORD	zu schreibendes EPC-Datenwort n

Tabelle 21

Rückgaben in den öffentlichen Variablen:

Name	Datentyp	Rückgabewert
iTagCounter	INT	0 .. 1 Anzahl der bearbeiteten Tags
arTag[1]	DTE810stTagData	von Tag 1 gelesene Daten
bDataErrorCode	BYTE	Tagdaten-Fehlercode (siehe Tabelle 29)
bRssi	BYTE	RSSI-Wert (Stärke der Kopplung Tag – Antenne)
stEPC	DTE810stTagEPCData	vor der Änderung gelesene EPC-Daten
iLength	INT	Anzahl der EPC-Datenworte
arwEpcWord[0]	WORD	gelesenes EPC-Datenwort 0
arwEpcWord[...]	WORD	gelesenes EPC-Datenwort ...
arwEpcWord[n]	WORD	gelesenes EPC-Datenwort n
bTagErrorCode	BYTE	Tagzugriff-Fehlercode (siehe Tabelle 30)
stMemoryData	Dte810stTagMemoryData	0x0000
arTag[...]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Tabelle 22

6.3.9 Funktions-ID 9 – EPC eines Tags ändern

Diese Funktion schreibt den angegebenen EPC auf einem bestimmten Tag. Der zu beschreibende Tag wird über den EPC ausgewählt. Ebenso ist das Passwort des Tags anzugeben. Standardmäßig ist auf dem Tag als Passwort 0x0000 0000 eingestellt.

Erforderliche Parameter in den öffentlichen Variablen:

Name	Datentyp	Eingabewert
dwPassword	DWORD	Tagzugriff-Passwort (auf Tag gespeichert)
stSelectionEPC	DTE810stTagEPCData	EPC-Daten zur Auswahl eines Tags
iLength	INT	Anzahl der EPC-Datenworte des gewählten Tags
arwEpcWord[0]	WORD	EPC-Datenwort 0 des gewählten Tags
arwEpcWord[...]	WORD	EPC-Datenwort ... des gewählten Tags
arwEpcWord[n]	WORD	EPC-Datenwort n des gewählten Tags
stEpcToWrite	Dte810TagEpcData	zu schreibende EPC-Daten
iLength	INT	Anzahl der zu schreibenden EPC-Datenworte
arwEpcWord[0]	WORD	zu schreibendes EPC-Datenwort 0
arwEpcWord[...]	WORD	zu schreibendes EPC-Datenwort ...
arwEpcWord[n]	WORD	zu schreibendes EPC-Datenwort n

Tabelle 23

Rückgaben in den öffentlichen Variablen:

Name	Datentyp	Rückgabewert
iTagCounter	INT	0 .. 1 Anzahl der bearbeiteten Tags
arTag[1]	DTE810stTagData	von Tag 1 gelesene Daten
bDataErrorCode	BYTE	Tagdaten-Fehlercode (siehe Tabelle 29)
bRssi	BYTE	RSSI-Wert (Stärke der Kopplung Tag – Antenne)
stEPC	DTE810stTagEPCData	vor der Änderung gelesene EPC-Daten
iLength	INT	Anzahl der EPC-Datenworte
arwEpcWord[0]	WORD	gelesenes EPC-Datenwort 0
arwEpcWord[...]	WORD	gelesenes EPC-Datenwort ...
arwEpcWord[n]	WORD	gelesenes EPC-Datenwort n
bTagErrorCode	BYTE	Tagzugriff-Fehlercode (siehe Tabelle 30)
stMemoryData	Dte810stTagMemoryData	0x0000
arTag[...]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Tabelle 24

6.3.10 Funktions-ID 15 – Antennenleistung einstellen

Mit dieser Funktion kann die Antennenleistung eingestellt werden. Es wird nur der verbundene Antennenport beeinflusst. Besteht die Verbindung zu allen Antennen, so ist die zu beeinflussende Antennenport-Nummer anzugeben.

Erforderliche Parameter in den öffentlichen Variablen:

Name	Datentyp	Eingabewert
stAntPortPower	Dte810stAntPortPower	Parameter zum Einstellen der Antennenleistung
iPortNo	INT	nur relevant, wenn der Baustein mit allen Antennen arbeitet (stSettings.stDTE.iAntPortNo = 0) 1 .. 4 Portnummer, die beeinflusst werden soll
iPortPower	INT	0 Antenne aus 68 .. 132 Antennenleistung in ¼ dBm

Tabelle 25

Rückgaben in den öffentlichen Variablen:

Name	Datentyp	Rückgabewert
iTagCounter	INT	0
arTag[1]	DTE810stTagData	0x0000
arTag[...]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Tabelle 26

6.4 Abbruch und Aufbau der Kommunikation mit dem RFID-Reader

Der Funktionsbaustein erkennt einen Verbindungsabbruch zum RFID-Reader auf 2 Wegen:

Eingang *ixConnOK* = FALSE

Ethernet/IP-Status meldet Fehler während des Datenaustausches

Der Ausgang *oxBusy* wird TRUE gesetzt. Ein Verbindungsabbruch während der Ausführung einer Funktion bricht die Funktion ab und setzt auch den Ausgang *oxError* = TRUE. Im Parameter *owStatus* wird ein Fehlercode ausgegeben.

Ist der Eingang *ixConnOK* = TRUE, startet der Kommunikationsaufbau wie in Kapitel 6.2.1 beschrieben.

Nach Abschluss wird der Ausgang *oxBusy* = FALSE ausgegeben.

Der Signalverlauf ist in Abbildung 17 dargestellt.

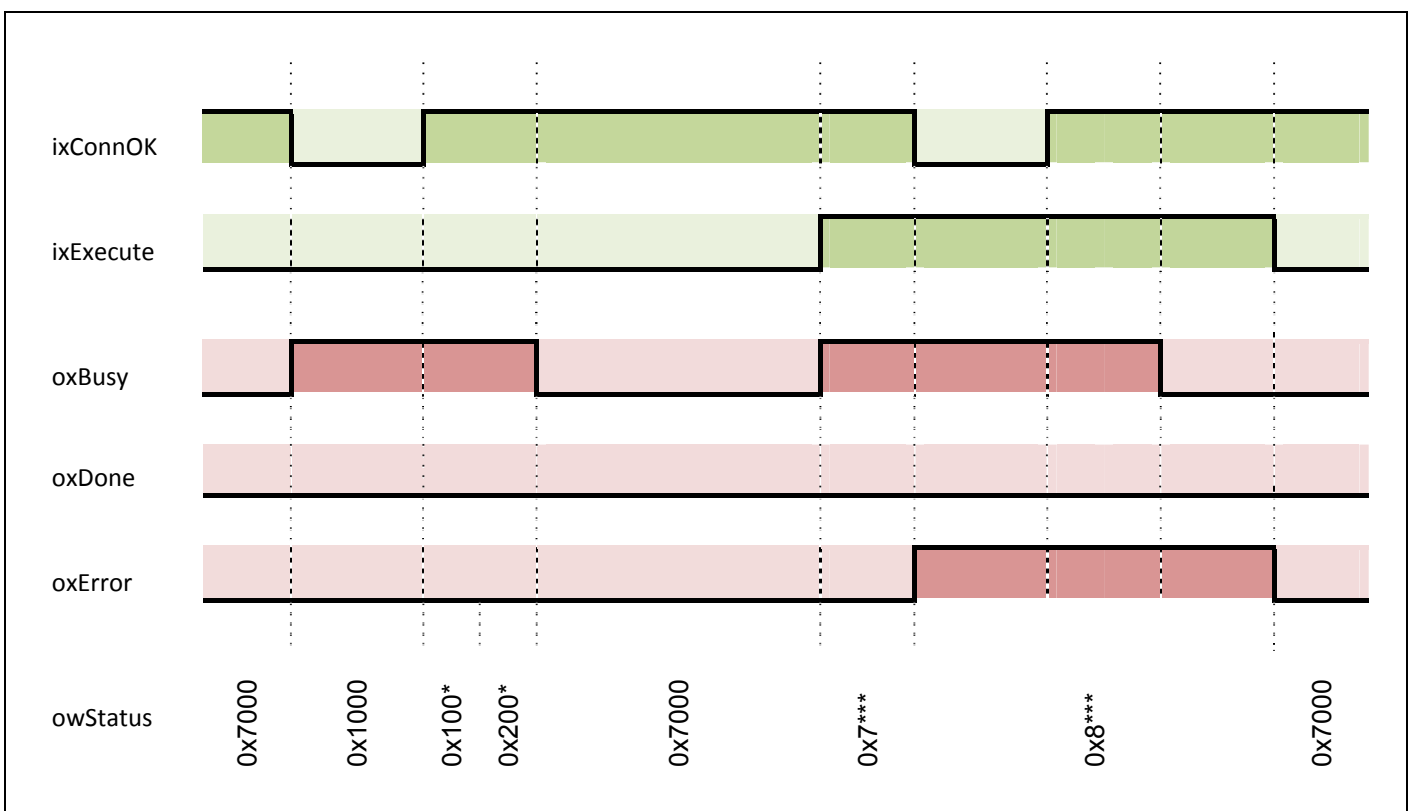


Abbildung 17

Wichtig: Der Funktionsbaustein erkennt nur die Verbindung zum RFID-Reader. Der korrekte Anschluss der Antenne wird nicht überwacht.

6.5 Statusanzeigen am Ausgang owStatus

owStatus	Beschreibung	
0x0000	Funktion ohne Fehler abgeschlossen	
0x1000	Eingang ixConnOK = FALSE oder keine Verbindung zum RFID-Reader	
0x1001	Synchronisierungs-Kommando senden	
0x1002	Synchronisierungs-Kommando aktivieren	
0x1003	Antwortdaten anfordern	
0x1004	auf Daten warten, Empfangsbestätigung senden	
0x1005	Empfangsbestätigung aktivieren	
0x2000	Initialisierungs-Kommando erstellen	
0x2001	Initialisierungs-Kommando senden	
0x2002	Initialisierungs-Kommando aktivieren	
0x2003	Antwortdaten anfordern	
0x2004	auf Daten warten, Empfangsbestätigung senden	
0x2005	Empfangsbestätigung aktivieren	
0x7000	auf Start einer Funktion warten	a = aktive Funktions-ID b = 0 warte auf Daten zur Auswertung b = 1 überprüfe Zugehörigkeit der Antwortdaten b = 2 zähle die erfassten Tags b = 3 übertragen der Tagdaten in das Array arTag in den öffentlichen Variablen
0x7001	Kommando senden	
0x7002	Kommando aktivieren	
0x7ab3	Antwortdaten anfordern	
0x7ab4	auf Daten warten, Empfangsbestätigung senden	
0x7ab5	Empfangsbestätigung aktivieren	
0x8***	Fehlercode (siehe Tabelle 28)	

Tabelle 27

7 Fehlercodes

7.1 Fehlercodes am Ausgang owStatus

owStatus	Beschreibung
0x8001	RFID-Reader meldet NODATA (Direct Mode: Keine Antwort vom Tag bei Gen2 Funktionen)
0x8002	RFID-Reader meldet CRCERROR (Direct Mode: Gen2-Funktionen - Prüfsummenvergleich Tag falsch)
0x8003	RFID-Reader meldet NOLICENSE (Ausführen einer Funktion die nicht über den Lizenzschlüssel abgedeckt ist)
0x8004	
0x8005	RFID-Reader meldet OUTOFRANGE (falscher Wertebereich)
0x8006	RFID-Reader meldet NOSTANDARD (Direct Mode: es ist kein Kommunikationsstandard gewählt)
0x8007	RFID-Reader meldet NOANTENNA (Direct Mode: keine Antenne gesetzt)
0x8008	RFID-Reader meldet NOFREQUENCY (Direct Mode: keine Frequenz ausgewählt)
0x8009	RFID-Reader meldet NOCARRIER (Direct Mode: Träger nicht angeschaltet)
0x800A	RFID-Reader meldet ANTENNAERROR (Direct Mode: Antennenfehler)
0x800B	RFID-Reader meldet NOTAG (kein Tag im Feld oder kein Tag mit angegebenem EPC im Feld)
0x800C	RFID-Reader meldet MORETHANONETAGINFIELD (mehr als 1 Tag im Feld z.B. bei SyncWriteEpcToSingleTag)
0x800D	
0x800E	RFID-Reader meldet WRONGLICENSEKEY (Falscher Lizenz-Key)
0x800F	RFID-Reader meldet FWREJECTED (Firmware abgewiesen beim Versuch eine nicht konforme FW zu laden)
0x8010	RFID-Reader meldet WRONGCFM falscher Modus (Direct Mode-Befehl im Normal Mode und umgekehrt)
0x8080	RFID-Reader meldet NOHANDLE (wenn das Handle (16-bit Autentisierung) des Tags nicht antwortet)
	RFID-Reader meldet NOPROFILE (Direct Mode: Kein Kommunikationsprofil gesetzt)
	RFID-Reader meldet UNSPECIFIED (unbestimmter Fehler)
0x8101	ungültiger Antennenport (stSettings.stDTE.iAntPortNo)
0x8102	ungültige Assembly-Größe im RFID-Reader eingestellt
0x8201	Eingang ixConnOK = FALSE während eine Funktion aktiv ist
0x8202	Fehler der internen Kommunikationsfunktion DATA_EXCH
0x8203	Ethernet/IP Systemfehler
0x8204	Ethernet/IP Protokollfehler
0x8301	Start einer Funktion während Funktion aktiv (ixExecute)
0x8302	Funktions-ID ungültig (iiFunctionID)
0x8303	Länge des Auswahl-EPC ungültig (stSelectionEpc.iLength)
0x8304	Länge des zu schreibenden EPC ungültig (stEpcToWrite.iLength)
0x8305	ungültige Datenlänge der zu schreibenden USER-Speicherdaten (stUserDataToWrite.iLength)
0x8401	Empfangene Antwortdaten Anfang fehlt
0x8402	Antwortdaten gehören nicht zum gesendeten Kommando
0x8403	EPC eines Tags im Antennenfeld zu lang
0x8404	zu viele Tags im Antennenfeld
0x8405	Fehler in den Daten eines Tags → Tagdaten-Fehlercodes auswerten (siehe Tabelle 29)

Tabelle 28

7.2 Tagdaten-Fehlercodes

arTags[x].bDataErrorCode	Beschreibung
0x00	kein Fehler in den Tagdaten
0x80	Fehler bei Tagzugriff, Tagzugriffs-Fehlercode auswerten (siehe Tabelle 30)
0x81	gelesene EPC-Daten zu lang
0x82	TID- oder USER-Speicher-Daten zu lang

Tabelle 29

7.3 Tagzugriff-Fehlercodes

arTags[x].bTagErrorCode	Beschreibung
0x00	kein Tag-Zugriffsfehler
0x01	keine Antwort vom Tag
0x02	Zugriff auf Tag verweigert (Passwort)
0x03	Vergleich nach dem Schreiben fehlgeschlagen
0x04	unbestimmter Fehler
0x80	alle Fehler die nicht von anderen Fehlercodes beschrieben werden
0x83	Tag Speicher kleiner als adressiert oder adressierter Bereich existiert nicht
0x84	Speicher des Tags gesperrt (Passwort) - keine Lesen/Schreiben möglich
0x8B	Ungenügende Leistung (zum Beschreiben des Tags)
0x8F	Tag unterstützt die spezifizierten Fehlercodes nicht

Tabelle 30

8 Beschreibung der abgeleiteten Datentypen

8.1 Arrays

Name	Typ	Beschreibung
Dte810arstTags	ARRAY[1..n] OF Dte810stTagData	Array zur Aufnahme der Daten von den bearbeiteten Tags Das Array muss mit Index 1 beginnen. Der Endindex n definiert wieviele Tags maximal vom DFB bearbeitet werden können. Er kann für die Anwendung passend eingestellt werden.
Dte810arwTagEpc	ARRAY[0..n] OF WORD	Array zur Aufnahme des EPC eines Tags Das Array muss mit Index 0 beginnen. Der Endindex n definiert die maximale Länge eines EPC, die der DFB verarbeiten kann. Er kann für die Anwendung passend eingestellt werden.
Dte810arwTagMemory	ARRAY[0..n] OF WORD	Array zur Aufnahme der TID- oder USER-Speicher-Daten eines Tags Das Array muss mit Index 0 beginnen. Der Endindex n definiert die maximale Speichergröße, die der DFB verarbeiten kann. Er kann für die Anwendung passend eingestellt werden.

Tabelle 31

8.2 Strukturen

Dte810stAntPortPower		Daten zur Einstellung der Antennenleistung
iPortNo	INT	Portnummer, die beeinflusst werden soll
iPortPower	INT	Antenneleistung in ¼ dBm

Tabelle 32

Dte810stDiagData		Detaillierte Fehlerinformationen
iLastFailedFctId	INT	Funktions-ID, bei der zuletzt ein Fehler aufgetreten ist
wLastFailedCmdId	WORD	DTE Kommando-ID, bei der zuletzt ein Fehler aufgetreten ist
wLastErrorCode	WORD	zuletzt vom Funktionsbaustein angezeigter Fehlercode
wEthipEventCode	WORD	zuletzt von Ethernet/IP gemeldeter Ereigniscode
bEthipCipErrorCode	BYTE	zuletzt von Ethernet/IP gemeldeter CIP-Fehlercode
wDataExchErrCode	WORD	zuletzt von der internen Funktion DATA_EXCH gemeldeter Fehlercode

Tabelle 33

Dte810stDteSet		Einstellungen der Verbindung zum RFID-Reader
ariIPAddr	ARRAY[1..4] OF INT	IP-Adresse des RFID-Readers
iAntPortNo	INT	Antennenport
iTimeOutVal	INT	Überwachungszeit der internen Funktion DATA_EXCH
iRepeatTime	INT	Verzögerungszeit zum Wiederholen des Befehls, wenn der RFID-Reader keine Daten liefert

Tabelle 34

Dte810stNocSet		Lokalisierung der Kommunikationsbaugruppe NOC 0401 in der SPS
iRackNo	INT	Rack-Nummer, in der die Kommunikationsbaugruppe montiert ist
iSlotNo	INT	Slot-Nummer, in der die Kommunikationsbaugruppe montiert ist
iChannelNo	INT	Kanal-Nummer der Kommunikationsbaugruppe, die zum Datenaustausch projektiert wurde

Tabelle 35

Dte810stReadEpc		Parameter zum Lesen aller EPCs
iReadingTime	INT	Aktivierungszeit (in ms) für das Lesen der EPCs aller Tags

Tabelle 36

Dte810stReadUserData		Parameter zum Lesen des USER-Speichers
diStartAddress	DINT	Speicheradresse, ab der gelesen werden soll
iLength	INT	Anzahl der zu lesenden Datenworte (1 Wort = 16 Bit)

Tabelle 37

Dte810stSettings		Daten zum Aufbau einer Verbindung zum RFID-Reader
stNOC	Dte810stNocSet	Daten zur Lokalisierung der Kommunikationsbaugruppe (siehe Tabelle 35)
stDTE	Dte810stDteSet	Parameter zum Datenaustausch mit dem RFID-Reader (siehe Tabelle 34)

Tabelle 38

Dte810stTagData		Daten, die von einem Tag gelesen wurden
bDataErrorCode	BYTE	Tagdaten-Fehlercode
bRssi	BYTE	RSSI-Wert (Stärke der Kopplung Tag – Antenne)
stEpc	Dte810stTagEpcData	EPC-Daten (siehe Tabelle 40)
bTagErrorCode	BYTE	Tagzugriff Fehlercode
stMemoryData	Dte810stTagMemoryData	gelesene Tagdaten (siehe Tabelle 41)

Tabelle 39

Dte810stTagEpcData		Daten eines EPC
iLength	INT	Anzahl der EPC-Datenworte (1 Wort = 16 Bit)
arwEpcWord	Dte810arwTagEpc	Array mit EPC-Datenworten (siehe Tabelle 31)

Tabelle 40

Dte810stTagMemoryData		TID- bzw. USER-Daten
iLength	INT	Anzahl der Datenworte (1 Wort = 16 Bit)
arwDataWord	Dte810arwTagMemory	Array mit Datenworten (siehe Tabelle 31)

Tabelle 41

Dte810stWriteUserData		Daten zum Beschreiben des USER-Speichers des Tag
diStartAddress	DINT	Speicheradresse, ab der geschrieben werden soll
iLength	INT	Anzahl der zu schreibenden Datenworte (1 Wort = 16 Bit)
arwDataWord	Dte810arwTagMemory	Array mit den zu schreibenden Daten (siehe Tabelle 31)

Tabelle 42