ifm syntron

Easy command interface
Modicon M340 controller

Library description

DTE810
DTE910

Contents

1 Overview

The Unity Pro library DTE810 / DTE910 contains a code package for a Schneider Electric Modicon M340 controller to facilitate data exchange with an RFID reader DTE810 or DTE910 RFID reader from the company ifm electronic in the user program of the PLC.
The following functions are provided in the user program by the function blocks of the library:

Read the EPC of all tags in the antenna field

Write an EPC to a tag

Read the TID memory of all tags in the antenna field or of a selected tag

Read an area of the USER memory of all tags in the antenna field or of a selected tag

Write to an area of the USER memory of all tags in the antenna field or a selected tag

Set the antenna performance

This document explains how to set up the connection between the controller and the RFID reader and describes how to integrate and use the library in the Unity-Pro project.

2 Exclusion of liability

The code package of the library is a demo version. It is only intended to serve as an example for the user. Any use of this code package for machine control is at your own risk.
The designers and holders of rights of this code package exclude any liability for malfunction or incompatibility of this demo version.
For reasons of copyright this program may only be passed on free of charge.

3 Set the connection parameters in the RFID reader

To set the connection parameters in the RFID reader the software ReaderStart version 2.30 or higher is required.

- Start the software ReaderStart on the PC and establish a connection to the RFID reader.

- Click on the [v] button with the designation *Communication configuration intelligence module* to open the port settings.

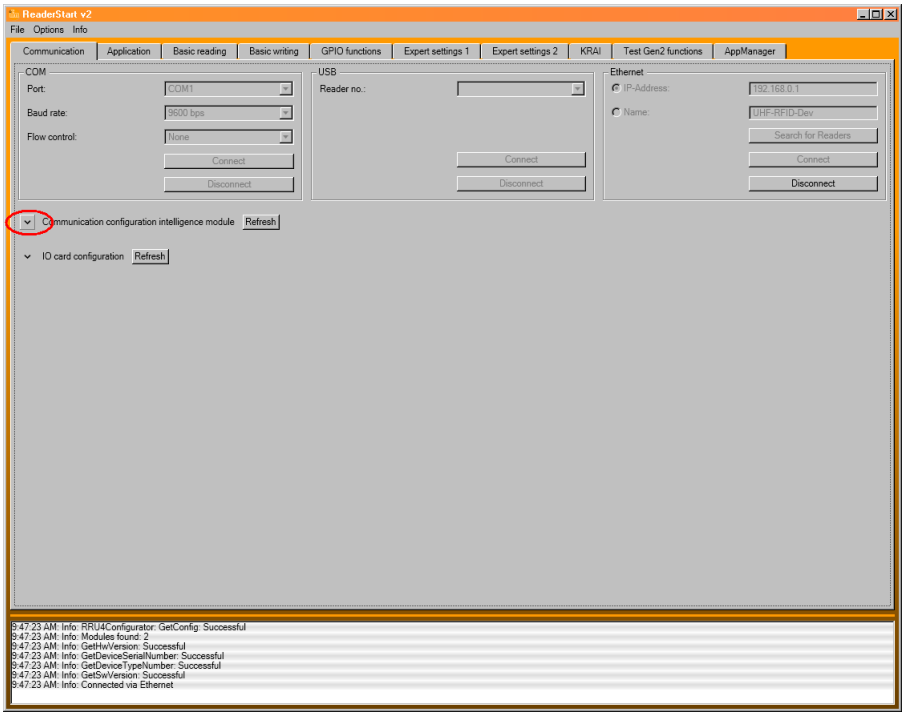


Figure 1

- Set the parameters of the communication port in *Port settings*.
- Click on [Save changes] to apply the settings to the RFID reader.

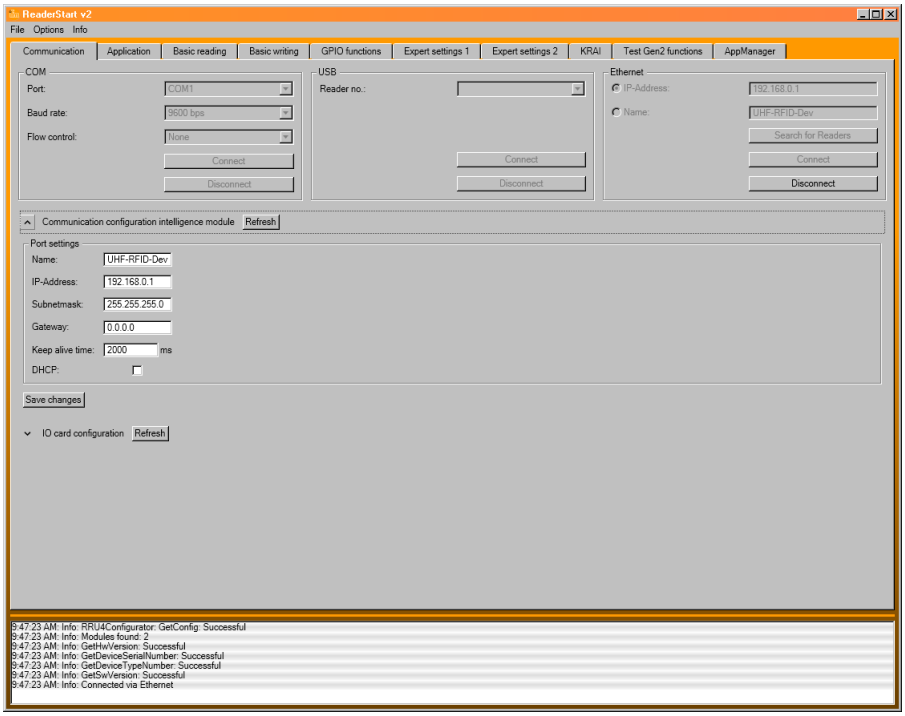


Figure 2

4 Project the connection in Unity Pro

The communication between the controller and the RFID reader is made via Ethernet/IP by means of an NOC0401 communication module in the M340 controller. The function block of the DTE810 / DTE910 library uses the acyclic service "Explicit Messaging" for the data exchange with the RFID reader. It is not necessary to configure a connection.

Nevertheless it is recommended to configure a cyclic connection to the RFID reader. Then it is easy to determine the status of the Ethernet/IP connection to the RFID reader by evaluating the health bit of the connection provided by the communication module.

A cyclic connection, however, uses the assigned assemblies so that an acyclic access to it is no longer possible. Therefore it is recommended to configure a cyclic connection to the digital inputs and outputs of the RFID reader. This means that all assemblies of the antenna ports remain available for the acyclic service.

The digital inputs and outputs of the RFID reader occupy 2 bytes in the input and ourtput area of the NOC0401 communication module. This is to be considered for the configuration of the communication module address range.

4.1 Add the RFID reader to the software project

- Right-click the communication module in the DTM browser.
- In the context menu, select *Add...*.

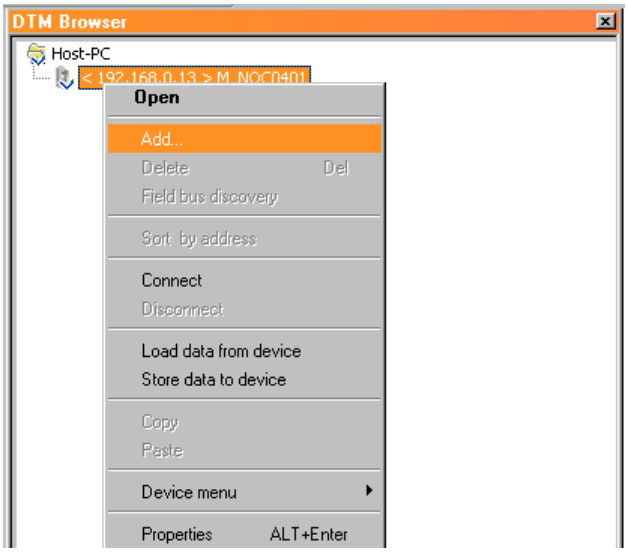


Figure 3

- From the list of the devices select the *Generic Device* from *Schneider Electric*.
- Click on [Add DTM] to open the window *Properties of device*.

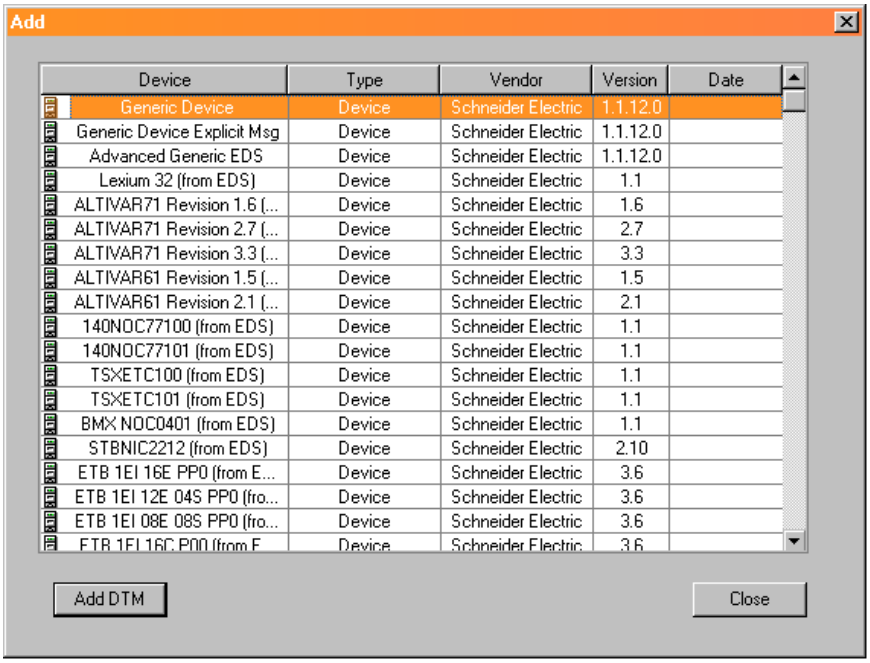


Figure 4

- In the *General* tab, assign a unique alias name to the device.

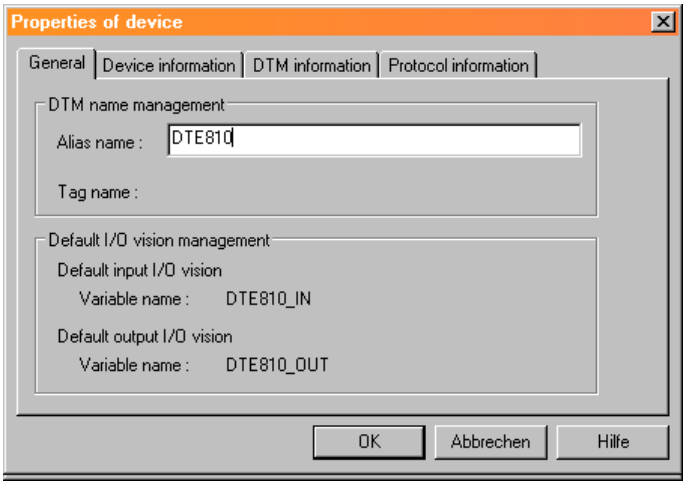


Figure 5

- In the *Protocol information* tab, set the required CIP (EtherNet/IP) protocol.
- Exit the *Properties of device* dialogue by clicking on [OK] to apply the settings.

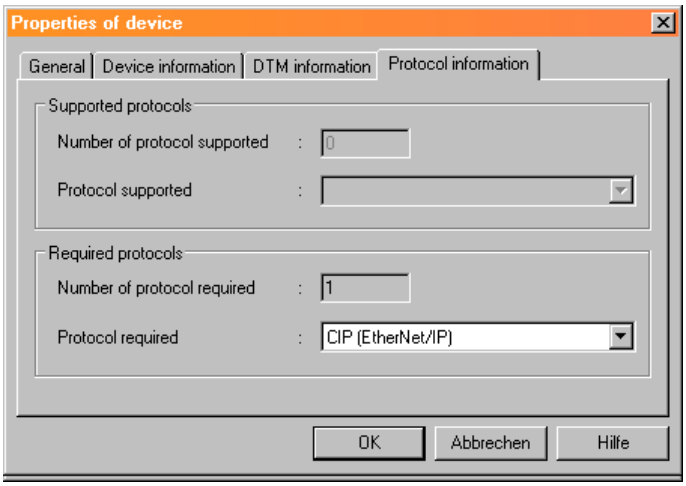


Figure 6

4.2 Set the digital inputs and outputs of the RFID reader as connection target

- Right-click the added device in the DTM browser.
- From the context menu select *Open* to open the *fdtConfiguration* window.

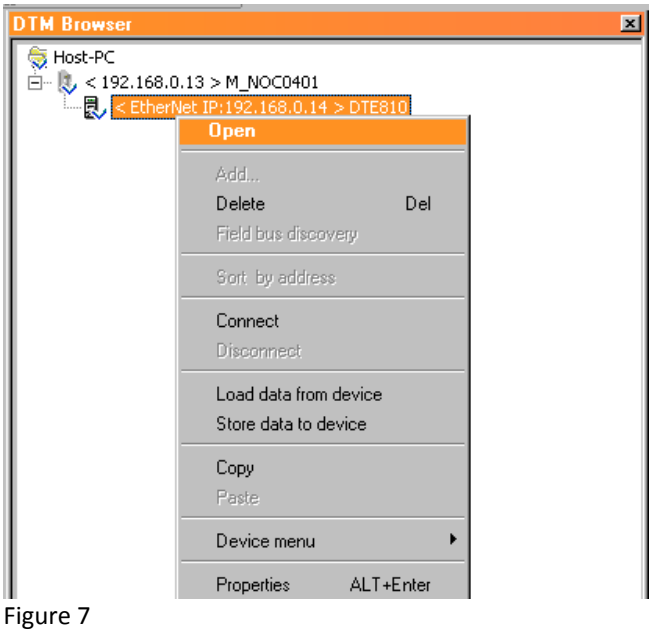


Figure 7

- In the structure, click on *Exclusive Owner*.
- In the tab *General* set the communication assemblies for the cyclic data exchange.
- Set the following values:
  Input size = 2
  Input Instance = 101
  Output size = 2
  Output Instance = 100
- No changes are necessary in the tabs *Identiy Check* and *Configuration Settings*.
- Click on [OK] to apply the data.

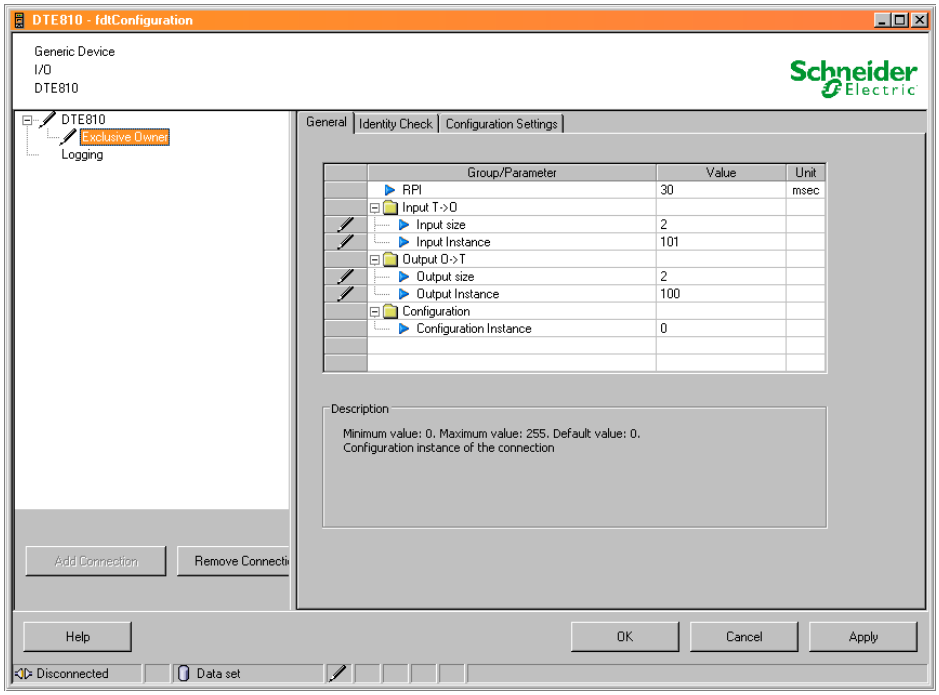


Figure 8

4.3 Set the connection parameters in the communication module.

- Right-click the communication module in the DTM browser.
- In the context menu, select *Open*.

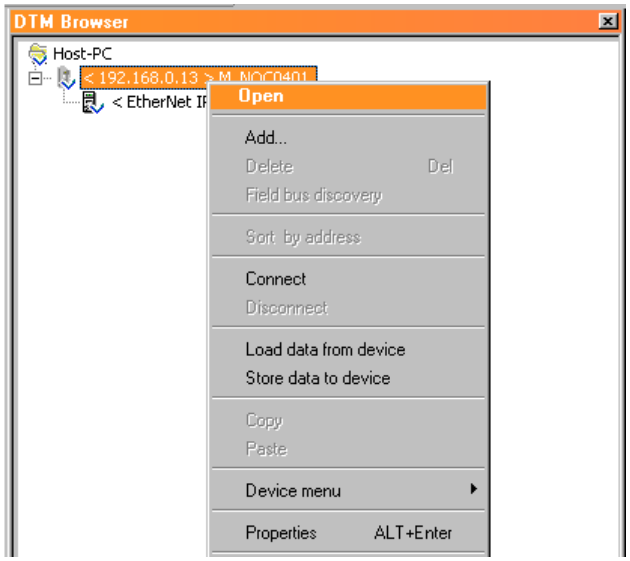


Figure 9

- In the structure, expand *Device List* and select the RFID reader.
- In the tab *Address Setting* enter the values set in the RFID reader.
- Click on [OK] to apply the settings.

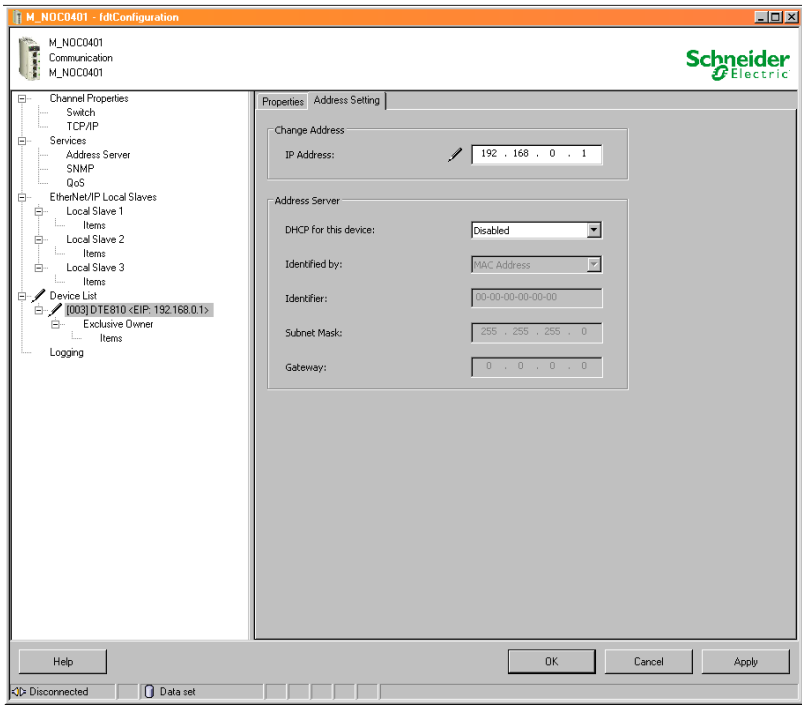


Figure 10

5 Working with the library

5.1 Integration into the library in Unity Pro

The DTE810 / DTE910 library is supplied as archive file. After this archive has been unzipped, the library can be integrated into the types library of Unity Pro by means of the tool for the types library update from the Unity software package.

- Start the tool for the types library update.
- Set the directory with the library as source.
- Click on [Install family].
- After the installation, click on [Exit] to exit the tool.

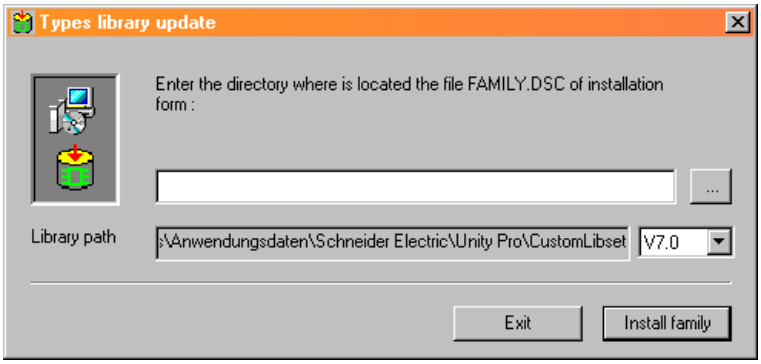


Figure 11

The function blocks are added to the ifm electronic library in the DTE810 / DTE910 family.

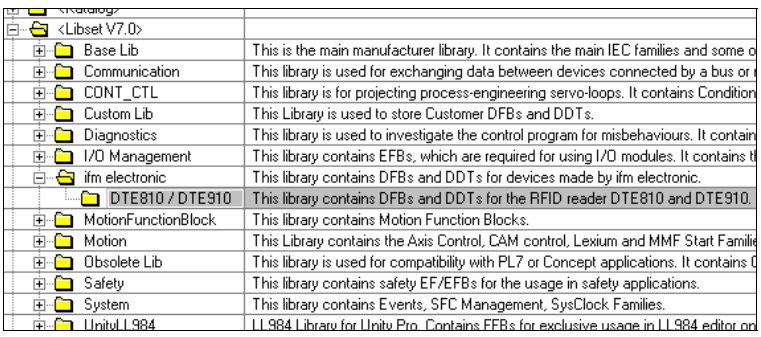


Figure 12

5.2 Library contents

The library DTE810 / DTE910 consists of the central function block DTE810_910expl and a group of special data types required by the function block (Table 1).

Name	Type	Description
DTE810_910expl	DFB	Central code function block of the library
Dte810arstTags	ARRAY[1..n] OF Dte810stTagData	Array for the data of the processed tags The array must start with index 1. The ending index n defines how many tags can be processed by the DFB as maximum. It can be adjusted to the application.
Dte810arwTagEpc	ARRAY[0..n] OF WORD	Array for the EPC of a tag The array must start with index 0. The ending index n defines the max. length of an EPC that can be processed by the DFB. It can be adjusted to the application.
Dte810arwTagMemory	ARRAY[0..n] OF WORD	Array for the TID or USER memory data of a tag The array must start with index 0. The ending index n defines the max. memory size that can be processed by the DFB. It can be adjusted to the application.
Dte810stAntPortPower	Strukt	Data to set the antenna performance
Dte810stDiagData	Strukt	Detailed error information
Dte810stDteSet	Strukt	Settings of the connection to the RFID reader
Dte810stNocSet	Strukt	Localisation of the communication module NOC 0401 in the PLC
Dte810stReadEpc	Strukt	Parameter to read all EPCs
Dte810stReadUserData	Strukt	Parameter to read the USER memory
Dte810stSettings	Strukt	Data to establish a connection to the RFID reader
Dte810stTagData	Strukt	Data read by a tag
Dte810stTagEpcData	Strukt	Data of an EPC
Dte810stTagMemoryData	Strukt	TID or USER data
Dte810stWriteUserData	Strukt	Data to write to the USER memory of the tag

Table 1

The structures are described in more detail in chapter 8.2. The meaning of the individual elements of the structures and their use are explained in chapters 6.2 and 6.3.

5.3 Use of the function blocks in the user program

If an instance of the function block DTE810_910expl is generated in the user program from the library, the data types used are also automatically created in the user program.

Only one instance of the function block is permissible for each antenna port number of an RFID reader. All instances used must be called once per cycle in the cyclic program of the PLC.

The size of the required memory of an instance in the user program is influenced by the sizes of the arrays *Dte810arstTags*, *Dte810arwTagEpc* and *Dte810arwTagMemory*. They can be adapted to the requirements of the application. Only the ending index of the array may be changed, the beginning index and the data type must remain on their initial values.

An example for illustration:

- Max. 7 tags are expected simultaneously.
  → Declaration Dte810arstTags: ARRAY[1..7] OF Dte810stTagData
- The longest EPC is 6 words long.
  → Declaration Dte810arwTagEpc: ARRAY[0..5] OF WORD
- Max. 12-word data should be read or written.
  → Declaration Dte810arwTagMemory: ARRAY[0..11] OF WORD

6 The function block DTE810_910expl in detail

6.1 Function description

The communication has to follow a defined process. **Figure 13** illustrates this on the basis of a flow chart:

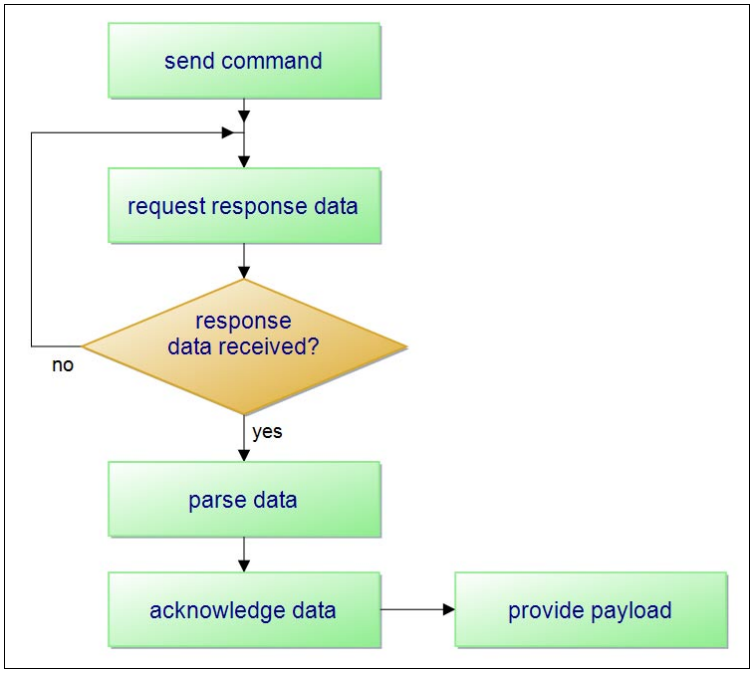


Figure 13

To make command processing more user-friendly, the *DTE810_910expl* function block was developed. This function block processes the above-mentioned command protocol automatically.

The folling figure shows the function block DTE810_910expl:

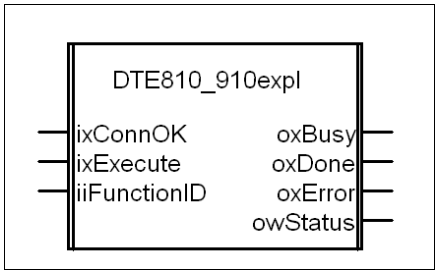


Figure 14

The parameters have the following meaning:

Inputs:

Name	Type	Description
ixConnOK	BOOL	Connection antenna - RFID reader - PLC has been established
ixExecute	BOOL	Rising edge activates the function set at iiFunctionID
iiFunctionID	INT	ID of the function to be executed (see Table 6)

Table 2

Outputs:

Name	Type	Description
oxBusy	BOOL	Function block is busy, no new command possible
oxDone	BOOL	Function completed without error
oxError	BOOL	Error detected
owStatus	WORD	Status information or error code (see Table 27)

Table 3

Public variables:

Name	Type	Description
stSettings	Dte810stSettings	Data of the basic settings of the connection to the RFID reader
stReadEpc	Dte810stReadEpc	Parameter for reading the EPCs
stReadUserData	Dte810stReadUserData	Parameter for reading the USER memory of the tags
stSelectionEpc	Dte810stTagEpcData	EPC data to select a certain tag
dwTagPassword	DWORD	Password for tag access
stEpcToWrite	Dte810stTagEpcData	Parameter for writing a new EPC to a tag
stUserDataToWrite	Dte810stWriteUserData	Parameter and data for writing to the USER memory of the tag
stAntPortPower	Dte810stAntPortPower	Parameter for setting the antenna performance
stDiagData	Dte810stDiagData	Diagnostic data
iTagCounter	INT	Number of found tags
arTag	Dte810arstTags	Data read from the tags

Table 4

6.2 Operation of the function block DTE810_910expl

6.2.1 Communication set-up to the RFID reader

Before the communication with the RFID reader can be started, all basic settings and parameters in the structure *stSettings* have to be entered in the public variables of the instance (Table 5).

Name	Data type	Input value
stSettings	Dte810stSettings	Data to establish a connection to the RFID reader
stNOC	Dte810stNocSet	Localisation of the communication module NOC0401 in the PLC
iRackNo	INT	Rack number in which the communication module is installed
iSlotNo	INT	Slot number in which the communication module is installed
iChannelNo	INT	Channel number of the communication module
stDTE	Dte810stDteSet	Settings of the connection to the RFID reader
ariIPAddr	ARRAY[1..4] OF INT	IP address of the RFID reader Example: IP address = 192.168.0.1 ariIPAddr[1] := 192 ariIPAddr[2] := 168 ariIPAddr[3] := 0 ariIPAddr[4] := 1
iAntPortNo	INT	= 0 data exhange with all activated antenna ports 1 .. 4 data exchange with antenna ports 1 .. 4
iTimeOutVal	INT	= 0 internal function DATA_EXCH uses the standard monitoring time > 0 monitoring time of the internal function DATA_EXCH (value * 100ms)
iRepeatTime	INT	Delay time to repeat the command if the RFID reader does not supply any data (recommended = 200 ms)

Table 5

Setting the input *ixConnOK* = TRUE activates the communication with the RFID reader.

The values of the structure *stSettings* in the public variables are applied.

In the public variables the data in the array *arTag* and the tag counter *iTagCounter* are assigned the value 0.

The function block cyclically requests data from the RFID reader. When data could be received, the function block synchronises the data exchange.

After completion the output *oxBusy* = FALSE is provided. The function block is now ready to perform a function. The signal sequence is shown in Figure 15.

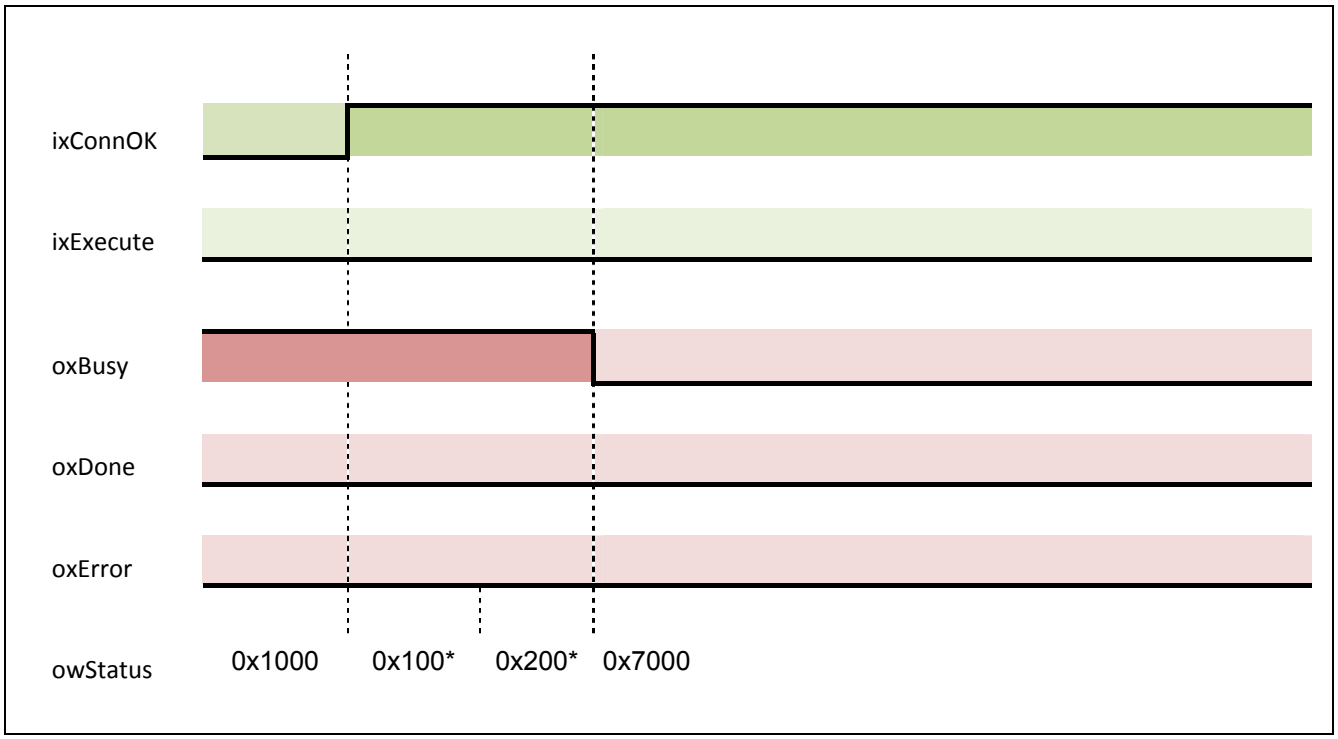


Figure 15

6.2.2 Perform a function

The DTE810_910expl function block is a function block working in an asynchronous way, i.e. processing stretches across several function block calls.

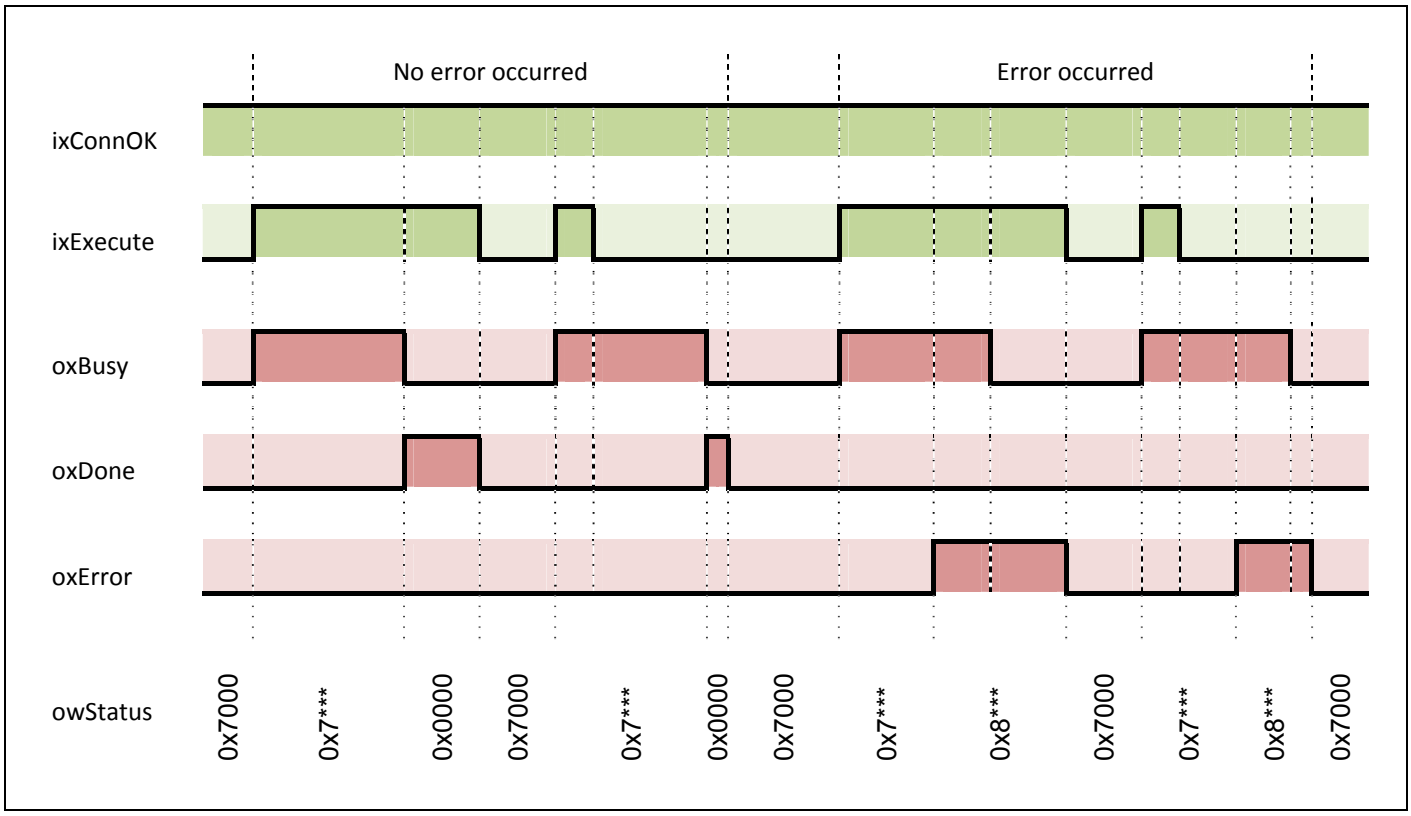
A function to be performed is selected by indicating the function ID at the parameter iiFunctionID. The required parameters and data of the function have to be entered in the respective structure in the public data of the function block. The function is started with a rising edge on input *iiExecute*.

The status of the command is displayed via the output parameters *oxBusy* and *owStatus*.
During command processing the parameter *oxBusy* is set. The parameter *owStatus* indicates the progress of command processing. With completion of the command the result is indicated in the parameters *oxDone* and *oxError* and remains until the input *ixExecute* = TRUE, min. however for one PLC cycle.

If no error has occurred, *oxDone* is set and the value 0x0000 is entered in the parameter *owStatus*. For commands with reply data from the tags this data is provided in array *arTag* in the public variables. The public variable *iTagCounter* indicates how many tags were detected or influenced by the function.

If an error occurs during command processing, *oxError* is set. In the parameter *owStatus* an error code is indicated for detailed description of an error occurred. The possible error codes are listed in Table 28.

The signal sequence is shown in Figure 16.


No error occurred
Error occurred
ixConnOK
ixExecute
oxBusy
oxDone
oxError
owStatus
0x7000
0x7***
0x0000
0x7000
0x7***
0x0000
0x7000
0x7***
0x8***
0x7000
0x7***
0x8***
0x7000
Figure 16

6.3 Description of the functions

Function	ID (decimal)	Required parameters	Response data	Description
Read EPC of all tags	1	stReadEpc	iTagCounter, arTag	See chapter 6.3.1
Read TID memory of all tags	2	dwTagPassword	iTagCounter, arTag	See chapter 6.3.2
Read TID memory of one tag	3	dwTagPassword, stSelectionEpc	iTagCounter, arTag	See chapter 6.3.3
Read an area of the USER memory of all tags	4	dwTagPassword, stReadUserData	iTagCounter, arTag	See chapter 6.3.4
Read an area of the USER memory of one tag	5	dwTagPassword, StSelectionEpc, stReadUserData	iTagCounter, arTag	See chapter 6.3.5
Write data to an area of the USER memory of all tags	6	dwTagPassword, stUserDataToWrite	iTagCounter, arTag	See chapter 6.3.6
Write data to an area of the USER memory of one tag	7	dwTagPassword, stSelectionEpc, stUserDataToWrite	iTagCounter, arTag	See chapter 6.3.7
Write EPC to an unknown tag	8	dwTagPassword, stEpcToWrite	iTagCounter, arTag	See chapter 6.3.8
Change EPC of a tag	9	dwTagPassword, stSelectionEpc, stEpcToWrite	iTagCounter, arTag	See chapter 6.3.9
Set antenna performance	15	stAntPortPower		See chapter 6.3.10

Table 6

6.3.1 Function ID 1 - Read EPC of all tags

This function creates a list with the EPCs of all tags that were found during the set read time regardless if the tags are still in the antenna field or not when reading is finished.

Required parameters in public variables:

Name	Data type	Input value
stReadEpc	Dte810stReadEpc	Parameter to read all EPCs
iReadingTime	INT	= 0 read the EPCs of all tags once > 0 activation time (in ms) for reading the EPCs of all tags

Table 7

Response data in the public variables:

Name	Data type	Return value
iTagCounter	INT	Number of tags processed
arTag[1]	DTE810stTagData	Data read from tag 1
DataErrorCode	BYTE	Tag data error code (see Table 29)
bRssi	BYTE	RSSI value (signal strength between tag and antenna)
stEPC	DTE810stTagEPCData	EPC data read
iLength	INT	Number of the EPC data words
arwEpcWord[0]	WORD	Read EPC data word 0
arwEpcWord[…]	WORD	Read EPC data word ...
arwEpcWord[n]	WORD	Read EPC data word n
bTagErrorCode	BYTE	0x00
stMemoryData	Dte810stTagMemoryData	0x0000
arTag[…]	DTE810stTagData	Data read from tag ...
arTag[n]	DTE810stTagData	Data read from tag n

Table 8

6.3.2 Function ID 2 - read TID memory of all tags

This function reads the TID memory of all tags in the antenna field whose access password is identical with the entered password. As a standard, the password 0x0000 0000 is set on the tags.

Required parameters in public variables:

Name	Data type	Input value
dwPassword	DWORD	Tag access password (stored on tag)

Table 9

Response data in the public variables:

Name	Data type	Return value
iTagCounter	INT	Number of tags processed
arTag[1]	DTE810stTagData	Data read from tag 1
DataErrorCode	BYTE	Tag data error code (see Table 29)
bRssi	BYTE	RSSI value (signal strength between tag and antenna)
stEPC	DTE810stTagEPCData	EPC data read
iLength	INT	Number of the EPC data words
arwEpcWord[0]	WORD	Read EPC data word 0
arwEpcWord[...]	WORD	Read EPC data word ...
arwEpcWord[n]	WORD	Read EPC data word n
bTagErrorCode	BYTE	Tag access error code (see Table 30)
stMemoryData	Dte810stTagMemoryData	TID data read
iLength	INT	Number of TID data words
arwDataWord[0]	WORD	TID data word 0 read
arwDataWord[...]	WORD	TID data word ... read
arwDataWord[n]	WORD	TID data word n read
arTag[...]	DTE810stTagData	Data read from tag ...
arTag[n]	DTE810stTagData	Data read from tag n

Table 10

6.3.3 Function ID 3 - read TID memory of a certain tag

This function reads the TID memory of a certain tag. The tag to be read is selected via the EPC. The password of the tag also has to be entered. As a standard, the password 0x0000 0000 is set on the tag.

Required parameters in public variables:

Name	Data type	Input value
dwPassword	DWORD	Tag access password (stored on tag)
stSelectionEPC	DTE810stTagEPCData	EPC data to select a tag
iLength	INT	Number of the EPC data words of the selected tag
arwEpcWord[0]	WORD	EPC data word 0 of the selected tag
arwEpcWord[…]	WORD	EPC data word ... of the selected tag
arwEpcWord[n]	WORD	EPC data word n of the selected tag

Table 11

Response data in the public variables:

Name	Data type	Return value
iTagCounter	INT	0 .. 1 Number of tags processed
arTag[1]	DTE810stTagData	Data read from tag 1
DataErrorCode	BYTE	Tag data error code (see Table 29)
bRssi	BYTE	RSSI value (signal strength between tag and antenna)
stEPC	DTE810stTagEPCData	EPC data read
iLength	INT	Number of the EPC data words
arwEpcWord[0]	WORD	Read EPC data word 0
arwEpcWord[…]	WORD	Read EPC data word ...
arwEpcWord[n]	WORD	Read EPC data word n
bTagErrorCode	BYTE	Tag access error code (see Table 30)
stMemoryData	Dte810stTagMemoryData	TID data read
iLength	INT	Number of TID data words
arwDataWord[0]	WORD	TID data word 0 read
arwDataWord[…]	WORD	TID data word ... read
arwDataWord[n]	WORD	TID data word n read
arTag[…]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Table 12

6.3.4 Function ID 4 - read area of the USER memory of all tags

This function reads an area of the USER memory to be specified of all tags in the antenna field whose access password is identical with the entered password. As a standard, the password 0x0000 0000 is set on the tags.

Required parameters in public variables:

Name	Data type	Input value
dwPassword	DWORD	Tag access password (stored on tag)
stReadUserData	DTE810stReadUserData	Parameter to read the USER memory
diStartAddress	DINT	Memory address from which reading is to be started
iLength	INT	Number of data words to be read

Table 13

Response data in the public variables:

Name	Data type	Return value
iTagCounter	INT	Number of tags processed
arTag[1]	DTE810stTagData	Data read from tag 1
DataErrorCode	BYTE	Tag data error code (see Table 29)
bRssi	BYTE	RSSI value (signal strength between tag and antenna)
stEPC	DTE810stTagEPCData	EPC data read
iLength	INT	Number of the EPC data words
arwEpcWord[0]	WORD	Read EPC data word 0
arwEpcWord[…]	WORD	Read EPC data word ...
arwEpcWord[n]	WORD	Read EPC data word n
bTagErrorCode	BYTE	Tag access error code (see Table 30)
stMemoryData	Dte810stTagMemoryData	USER memory data read
iLength	INT	Number of USER memory data words
arwDataWord[0]	WORD	USER memory data word 0 read
arwDataWord[…]	WORD	USER memory data word ... read
arwDataWord[n]	WORD	USER memory data word n read
arTag[…]	DTE810stTagData	Data read from tag ...
arTag[n]	DTE810stTagData	Data read from tag n

Table 14

6.3.5 Function ID 5 - read area of the USER memory of a certain tag

This function reads an area of the USER memory to be specified of a certain tag. The tag to be read is selected via the EPC. The password of the tag also has to be entered. As a standard, the password 0x0000 0000 is set on the tag.

Required parameters in public variables:

Name	Data type	Input value
dwPassword	DWORD	Tag access password (stored on tag)
stSelectionEPC	DTE810stTagEPCData	EPC data to select a tag
iLength	INT	Number of the EPC data words of the selected tag
arwEpcWord[0]	WORD	EPC data word 0 of the selected tag
arwEpcWord[…]	WORD	EPC data word ... of the selected tag
arwEpcWord[n]	WORD	EPC data word n of the selected tag
stReadUserData	DTE810stReadUserData	Parameter to read the USER memory
diStartAddress	DINT	Memory address from which reading is to be started
iLength	INT	Number of data words to be read

Table 15

Response data in the public variables:

Name	Data type	Return value
iTagCounter	INT	0 .. 1 Number of tags processed
arTag[1]	DTE810stTagData	Data read from tag 1
DataErrorCode	BYTE	Tag data error code (see Table 29)
bRssi	BYTE	RSSI value (signal strength between tag and antenna)
stEPC	DTE810stTagEPCData	EPC data read
iLength	INT	Number of the EPC data words
arwEpcWord[0]	WORD	Read EPC data word 0
arwEpcWord[…]	WORD	Read EPC data word ...
arwEpcWord[n]	WORD	Read EPC data word n
bTagErrorCode	BYTE	Tag access error code (see Table 30)
stMemoryData	Dte810stTagMemoryData	USER memory data read
iLength	INT	Number of USER memory data words
arwDataWord[0]	WORD	USER memory data word 0 read
arwDataWord[…]	WORD	USER memory data word ... read
arwDataWord[n]	WORD	USER memory data word n read
arTag[…]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Table 16

6.3.6 Function ID 6 - write to area of the USER memory of all tags

This function writes data to an area of the USER memory to be specified of all tags in the antenna field whose access passwords is identical with the entered password. As a standard, the password 0x0000 0000 is set on the tags.

If there are too many tags in the antenna field, no data is written and an error is returned.

Required parameters in public variables:

Name	Data type	Input value
dwPassword	DWORD	Tag access password (stored on tag)
stUserDataToWrite	Dte810WriteUserData	Parameter to write USER memory data
diStartAddress	DINT	Memory address from which writing is to be started
iLength	INT	Number of data words to be written
arwDataWord[0]	WORD	Data word 0 to be written
arwDataWord[…]	WORD	Data word ... to be written
arwDataWord[n]	WORD	Data word n to be written

Table 17

Response data in the public variables:

Name	Data type	Return value
iTagCounter	INT	Number of tags processed
arTag[1]	DTE810stTagData	Data read from tag 1
DataErrorCode	BYTE	Tag data error code (see Table 29)
bRssi	BYTE	RSSI value (signal strength between tag and antenna)
stEPC	DTE810stTagEPCData	EPC data read
iLength	INT	Number of the EPC data words
arwEpcWord[0]	WORD	Read EPC data word 0
arwEpcWord[…]	WORD	Read EPC data word ...
arwEpcWord[n]	WORD	Read EPC data word n
bTagErrorCode	BYTE	Tag access error code (see Table 30)
stMemoryData	Dte810stTagMemoryData	0x0000
arTag[…]	DTE810stTagData	Data read from tag ...
arTag[n]	DTE810stTagData	Data read from tag n

Table 18

6.3.7 Function ID 7 - write to area of the USER memory of one tag

This function writes data to an area of the USER memory to be specified of a certain tag. The tag to be written to is selected via the EPC. The password of the tag also has to be entered. As a standard, the password 0x0000 0000 is set on the tag.

Required parameters in public variables:

Name	Data type	Input value
dwPassword	DWORD	Tag access password (stored on tag)
stSelectionEPC	DTE810stTagEPCData	EPC data to select a tag
iLength	INT	Number of the EPC data words of the selected tag
arwEpcWord[0]	WORD	EPC data word 0 of the selected tag
arwEpcWord[…]	WORD	EPC data word ... of the selected tag
arwEpcWord[n]	WORD	EPC data word n of the selected tag
stUserDataToWrite	Dte810WriteUserData	Parameter to write USER memory data
diStartAddress	DINT	Memory address from which writing is to be started
iLength	INT	Number of data words to be written
arwDataWord[0]	WORD	Data word 0 to be written
arwDataWord[…]	WORD	Data word ... to be written
arwDataWord[n]	WORD	Data word n to be written

Table 19

Response data in the public variables:

Name	Data type	Return value
iTagCounter	INT	0 .. 1 Number of tags processed
arTag[1]	DTE810stTagData	Data read from tag 1
DataErrorCode	BYTE	Tag data error code (see Table 29)
bRssi	BYTE	RSSI value (signal strength between tag and antenna)
stEPC	DTE810stTagEPCData	EPC data read
iLength	INT	Number of the EPC data words
arwEpcWord[0]	WORD	Read EPC data word 0
arwEpcWord[…]	WORD	Read EPC data word ...
arwEpcWord[n]	WORD	Read EPC data word n
bTagErrorCode	BYTE	Tag access error code (see Table 30)
stMemoryData	Dte810stTagMemoryData	0x0000
arTag[…]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Table 20

6.3.8 Function ID 8 – write EPC to an unknown tag

This function writes the specified EPC to a tag. Only one tag may be in the antenna field. The password of the tag also has to be entered. As a standard, the password 0x0000 0000 is set on the tag.

Required parameters in public variables:

Name	Data type	Input value
dwPassword	DWORD	Tag access password (stored on tag)
stEpcToWrite	Dte810TagEpcData	EPC data to be written
iLength	INT	Number of EPC data words to be written
arwEpcWord[0]	WORD	EPC word 0 to be written
arwEpcWord[…]	WORD	EPC data word ... to be written
arwEpcWord[n]	WORD	EPC data word n to be written

Table 21

Response data in the public variables:

Name	Data type	Return value
iTagCounter	INT	0 .. 1 Number of tags processed
arTag[1]	DTE810stTagData	Data read from tag 1
DataErrorCode	BYTE	Tag data error code (see Table 29)
bRssi	BYTE	RSSI value (signal strength between tag and antenna)
stEPC	DTE810stTagEPCData	EPC data read before the change
iLength	INT	Number of the EPC data words
arwEpcWord[0]	WORD	Read EPC data word 0
arwEpcWord[…]	WORD	Read EPC data word ...
arwEpcWord[n]	WORD	Read EPC data word n
bTagErrorCode	BYTE	Tag access error code (see Table 30)
stMemoryData	Dte810stTagMemoryData	0x0000
arTag[…]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Table 22

6.3.9 Function ID 9 - change the EPC of a tag

This function writes the specified EPC to a certain tag. The tag to be written to is selected via the EPC. The password of the tag also has to be entered. As a standard, the password 0x0000 0000 is set on the tag.

Required parameters in public variables:

Name	Data type	Input value
dwPassword	DWORD	Tag access password (stored on tag)
stSelectionEPC	DTE810stTagEPCData	EPC data to select a tag
iLength	INT	Number of the EPC data words of the selected tag
arwEpcWord[0]	WORD	EPC data word 0 of the selected tag
arwEpcWord[…]	WORD	EPC data word ... of the selected tag
arwEpcWord[n]	WORD	EPC data word n of the selected tag
stEpcToWrite	Dte810TagEpcData	EPC data to be written
iLength	INT	Number of EPC data words to be written
arwEpcWord[0]	WORD	EPC word 0 to be written
arwEpcWord[…]	WORD	EPC data word ... to be written
arwEpcWord[n]	WORD	EPC data word n to be written

Table 23

Response data in the public variables:

Name	Data type	Return value
iTagCounter	INT	0 .. 1 Number of tags processed
arTag[1]	DTE810stTagData	Data read from tag 1
DataErrorCode	BYTE	Tag data error code (see Table 29)
bRssi	BYTE	RSSI value (signal strength between tag and antenna)
stEPC	DTE810stTagEPCData	EPC data read before the change
iLength	INT	Number of the EPC data words
arwEpcWord[0]	WORD	Read EPC data word 0
arwEpcWord[…]	WORD	Read EPC data word ...
arwEpcWord[n]	WORD	Read EPC data word n
bTagErrorCode	BYTE	Tag access error code (see Table 30)
stMemoryData	Dte810stTagMemoryData	0x0000
arTag[…]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Table 24

6.3.10 Function ID 15 - set antenna performance

By means of this function the antenna performance can be set. Only the connected antenna port is influenced. If there is a connection to all antennas, the port number of the antenna to be influenced is to be specified.

Required parameters in public variables:

Name	Data type	Input value
stAntPortPower	Dte810stAntPortPower	Parameters for setting the antenna performance
iPortNo	INT	Only relevant if the module works with all antennas (stSettings.stDTE.iAntPortNo = 0) 1 .. 4 Port number to be influenced
iPortPower	INT	0 Antenna off 68 .. 132 Antenna performance in ¼ dBm

Table 25

Response data in the public variables:

Name	Data type	Return value
iTagCounter	INT	0
f	f	0x0000
arTag[…]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Table 26

6.4 Disconnection and reconnection of the communication with the RFID reader

The function block detects a disconnection of the RFID reader in 2 ways:

Input *ixConnOK* = FALSE

The Ethernet/IP status signals an error during data exchange

The output *oxBusy* is set to TRUE. A disconnection during the execution of a function stops the function and sets the output *oxError* = TRUE. An error code is issued in the parameter *owStatus*.
If the input *ixConnOK* = TRUE, the communication build-up starts as described in chapter 6.2.1

After completion the output *oxBusy* = FALSE is provided.

The signal sequence is shown in Figure 17

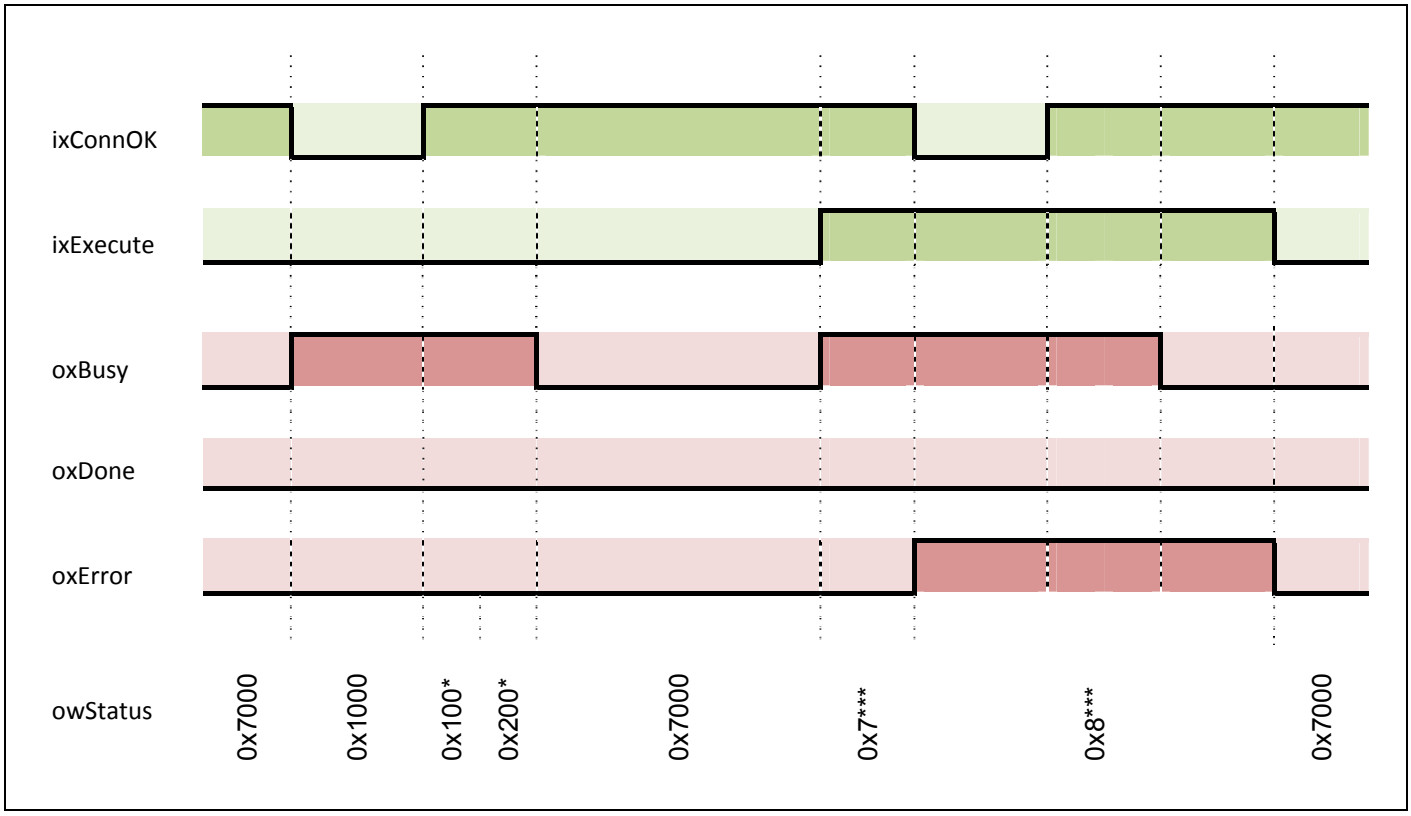


Figure 17

Important: The function block only detects the connection to the RFID reader. The correct connection of the antenna is not monitored.

6.5 Status displays on the output owStatus

owStatus	Description	
0x0000	Function completed without error	
0x1000 0x1001 0x1002 0x1003 0x1004 0x1005	Input ixConnOK = FALSE or no connection to the RFID reader Send synchronisation command Activate synchronisation command Request response data Wait for data, send acknowledgement of receipt Activate acknowledgement of receipt	
0x2000 0x2001 0x2002 0x2003 0x2004 0x2005	Create initialisation command Send initialisation command Activate initialisation command Request response data Wait for data, send acknowledgement of receipt Activate acknowledgement of receipt	
0x7000 0x7001 0x7002 0x7ab3 0x7ab4 0x7ab5	Wait for a function to start Send command Activate command Request response data Wait for data, send acknowledgement of receipt Activate acknowledgement of receipt	a = active function ID b = 0 wait for data for evaluation b = 1 check affiliation of the response data b = 2 count the detected tags b = 3 transfer the tag data to the array arTag in the public variables
0x8***	Error code (see Table 28)	

Table 27

7 Error codes

7.1 Error code on the output owStatus

owStatus	Description
0x8001	RFID reader returns NODATA (direct mode: no response from the tag for Gen2 functions)
0x8002	RFID reader returns CRCERROR (direct mode: Gen2 functions - checksum comparison tag wrong)
0x8003	RFID reader returns NOLICENSE (execution of a function which is not covered by the license key)
0x8004	RFID reader returns OUTOFRANGE (wrong value range)
0x8005	RFID reader returns NOSTANDARD (direct mode: no communication standard selected)
0x8006	RFID reader returns NOANTENNA (direct mode: no antenna selected)
0x8007	RFID reader returns NOFREQUENCY (direct mode: no frequency selected)
0x8008	RFID reader returns NOCARRIER (direct mode: carrier not switched on)
0x8009	RFID reader returns ANTENNAERROR (direct mode: antenna error)
0x800A	RFID reader returns NOTAG (no tag in the field or no tag with indicated EPC in the field)
0x800B	RFID reader returns MORETHANONETAGINFIELD (more than 1 tag in the field e.g. with
0x800C	SyncWriteEpcToSingleTag)
0x800D	RFID reader returns WRONGLICENSEKEY (wrong licence key)
0x800E	RFID reader returns FWREJECTED (firmware rejected during attempt to load a non-compliant firmware)
0x800F	RFID reader returns WRONGCFM wrong mode (direct mode command in normal mode and vice versa)
0x8010	RFID reader returns NOHANDLE (when the handle (16-bit authentication) of the tag does not reply)
0x8080	RFID reader returns NOPROFILE (direct mode: no communication profile set)
	RFID reader returns NONSPECIFIED (unspecified error)
0x8101	Invalid antenna port (stSettings.stDTE.iAntPortNo)
0x8102	Invalid assembly size set in the RFID reader
0x8201	Input ixConnOK = FALSE while a function is active
0x8202	Error of the internal communication function DATA_EXCH
0x8203	Ethernet/IP system error
0x8204	Ethernet/IP protocol error
0x8301	Start of a function while function is active (ixExecute)
0x8302	Function ID invalid (iiFunctionID)
0x8303	Length of the selection EPC invalid (stSelectionEpc.iLength)
0x8304	Length of the EPC to be written invalid (stEpcToWrite.iLength)
0x8305	Invalid data length of the USER storage data to be written (stUserDataToWrite.iLength)
0x8401	Received response data beginning missing
0x8402	Response data do not belong to the sent command
0x8403	EPC of a tag in the antenna field too long
0x8404	Too many antennas in the antenna field
0x8405	Error in the data of a tag → evaluate tag data error codes (see Table 29)

Table 28

7.2 Tag data error codes

arTags[x].bDataErrorCode	Description
0x00	No error in the tag data
0x80	Error with tag access, evaluate tag access error code (see Table 30)
0x81	Read EPC data too long
0x82	TID or USER memory data too long

Table 29

7.3 Tag access error codes

arTags[x].bTagErrorCode	Description
0x00	No tag access error
0x01	No answer from the tag
0x02	Access to tag denied (password)
0x03	Comparison failed after writing
0x04	Unspecified error
0x80	All errors that are not described by other error codes
0x83	Tag memory smaller than addressed or addressed area does not exist
0x84	Memory of the tag locked (password) - no reading/writing possible
0x8B	Insufficient performance (to write the tag)
0x8F	Tag does not support the specified error code

Table 30

8 Description of the derived data types

8.1 Arrays

Name	Type	Description
Dte810arstTags	ARRAY[1..n] OF Dte810stTagData	Array for the data of the processed tags The array must start with index 1. The ending index n defines how many tags can be processed by the DFB as maximum. It can be adjusted to the application.
Dte810arwTagEpc	ARRAY[0..n] OF WORD	Array for the EPC of a tag The array must start with index 0. The ending index n defines the max. length of an EPC that can be processed by the DFB. It can be adjusted to the application.
Dte810arwTagMemory	ARRAY[0..n] OF WORD	Array for the TID or USER memory data of a tag The array must start with index 0. The ending index n defines the max. memory size that can be processed by the DFB. It can be adjusted to the application.

Table 31

8.2 Structures

Dte810stAntPortPower			Data to set the antenna performance
	iPortNo	INT	Port number to be influenced
	iPortPower	INT	Antenna performance in ¼ dBm

Table 32

Dte810stDiagData		Detailed error information
iLastFailedFctId	INT	Function ID at which an error occurred last
wLastFailedCmdId	WORD	DTE command ID at which an error occurred last
wLastErrorCode	WORD	Error code last returned by the function block
wEthipEventCode	WORD	Event code last returned by Ethernet/IP
bEthipCipErrorCode	BYTE	CIP error code last returned by Ethernet/IP
wDataExchErrCode	WORD	Error code last returned by the internal function DATA_EXCH

Table 33

Dte810stDteSet		Settings of the connection to the RFID reader
ariIPAddr	ARRAY[1..4] OF INT	IP address of the RFID reader
iAntPortNo	INT	Antenna port
iTimeOutVal	INT	Monitoring time of the internal function DATA_EXCH
iRepeatTime	INT	Delay time to repeat the command if the RFID reader does not supply any data

Table 34

Dte810stNocSet		Localisation of the communication module NOC 0401 in the PLC
iRackNo	INT	Rack number in which the communication module is installed
iSlotNo	INT	Slot number in which the communication module is installed
iChannelNo	INT	Channel number of the communication module that was configured for data exchange

Table 35

Dte810stReadEpc		Parameter to read all EPCs
iReadingTime	INT	Activation time (in ms) for reading the EPCs of all tags

Table 36

Dte810stReadUserData		Parameter to read the USER memory
diStartAddress	DINT	Memory address from which reading is to be started
iLength	INT	Number of the data words to be read (1 word = 16 bits)

Table 37

Dte810stSettings		Data to establish a connection to the RFID reader
stNOC	Dte810stNocSet	Data for localisation of the communication module (see Table 35)
stDTE	Dte810stDteSet	Parameter for data exchange with the RFID reader (see Table 34)

Table 38

Dte810stTagData		Data read by a tag
DataErrorCode	BYTE	Tag data error code
bRssi	BYTE	RSSI value (signal strength between tag and antenna)
stEpc	Dte810stTagEpcData	EPC data (see Table 40)
bTagErrorCode	BYTE	Tag access error code
stMemoryData	Dte810stTagMemoryData	Tag data read (see Table 41)

Table 39

Dte810stTagEpcData		Data of an EPC
iLength	INT	Number of the EPC data words (1 word = 16 bits)
arwEpcWord	Dte810arwTagEpc	Array with EPC data words (see Table 31)

Table 40

Dte810stTagMemoryData		TID or USER data
iLength	INT	Number of the data words (1 word = 16 bits)
arwDataWord	Dte810arwTagMemory	Array with data words (see Table 31)

Table 41

Dte810stWriteUserData		Data to write to the USER memory of the tag
diStartAddress	DINT	Memory address from which writing is to be started
iLength	INT	Number of the data words to be written (1 word = 16 bits)
arwDataWord	Dte810arwTagMemory	Array with the data to be written (see Table 31)

Table 42