Training Manual for Integrated Automation Solutions

Totally Integrated Automation (TIA)

MODULE E11

Radio Frequency Identification (RFID)

with SIMATIC S7-300F-2 PN/DP and RF180C

This manual was prepared for training purposes by Siemens AG for the project *Siemens Automation Cooperates with Education* (SCE).
Siemens AG does not guarantee the contents of this document.

We wish to thank the Michael Dziallas Engineering corporation and the instructors of vocational schools as well as all those who provided support during the preparation of this manual.

PAGE

1 PREFACE 5

2 Notes regarding the usage of CPU 315F-2 PN/DP 7

3 Notes regarding the SIMATIC RFID components 8

4 RFID Fundamentals 9

5 Starting up an RFID project with CPU 315F-2 PN/DP and RF180C 10

5.1 Setting Up a New Project 11
5.2 Configuring the Hardware 13
5.3 Assigning a Device Name 20
5.4 Inserting UDT Blocks and FB45 22
5.5 Generating Data Blocks 23
5.6 Programming a Restart or Warm Restart 33
5.7 FC11 Function for a Command or Request 34
5.8 Basics of Entries at Command Block FC11 36
5.9 Command String 39
5.10 Basics of FB45 and DB45 40
5.11 FB10 Reader_Control Program 51
5.12 FB1 Control Program 55
5.13 OB1 Program Call 61
5.14 Variable Table STATUS_SLG_1 62
5.15 Variable Table STATUS_SLG_2 63
5.16 Symbol Table 64
5.17 Block Folder 65
5.18 DB49 Data View 65

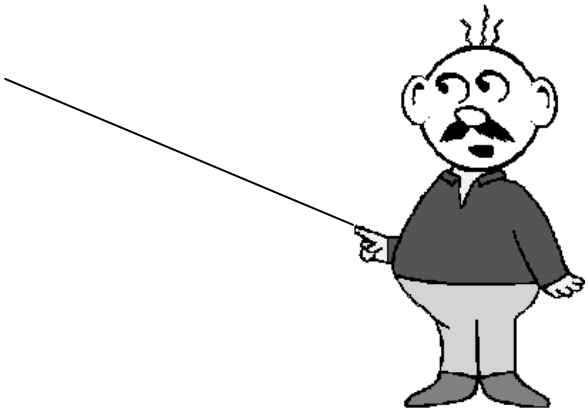
The following symbols serve as a guide through this module:

Information

Programming

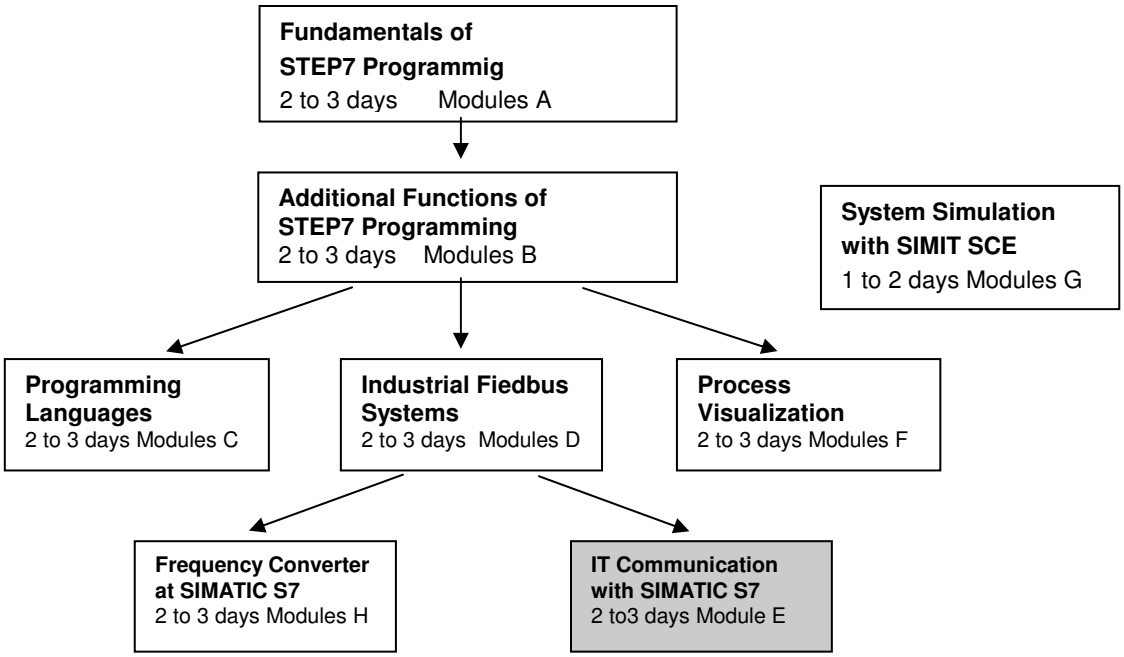
Sample Task

Notes



1 PREFACE

Regarding its content, Module E11 is part of the instruction unit **'IT Communication with SIMATIC S7'**.



Objective

In Module E11, the reader will learn how networking and data exchange between PLCs and RFID components is set up.
As PLC, the CPU 315F-2 PN/DP and as Radio Frequency Identification (RFID), a SIMATIC RFID system is used. The RFID components consist of the interface module RF180C (ASM) with write/read device RF310R (Reader or SLG (write/read device)) and different mobile data systems such as RF340T, RF350T, RF360T or ISO Moby D MDS D124 (transponder or MDS). PROFINET is used for networking the PLC and the SIMATIC RF180C.
Module E11 shows in principle the procedure for the startup, based on a brief example.

Prerequisites

To successfully work through Module E11, the following knowledge is assumed:

- How to handle Windows
- Fundamentals of PLC programming with STEP7 (for example, Module A 'Startup' PLC Programming with STEP7).

- Fundamentals of network engineering (for example, Appendix V – Basics of Network Engineering)

Hardware and Software Required

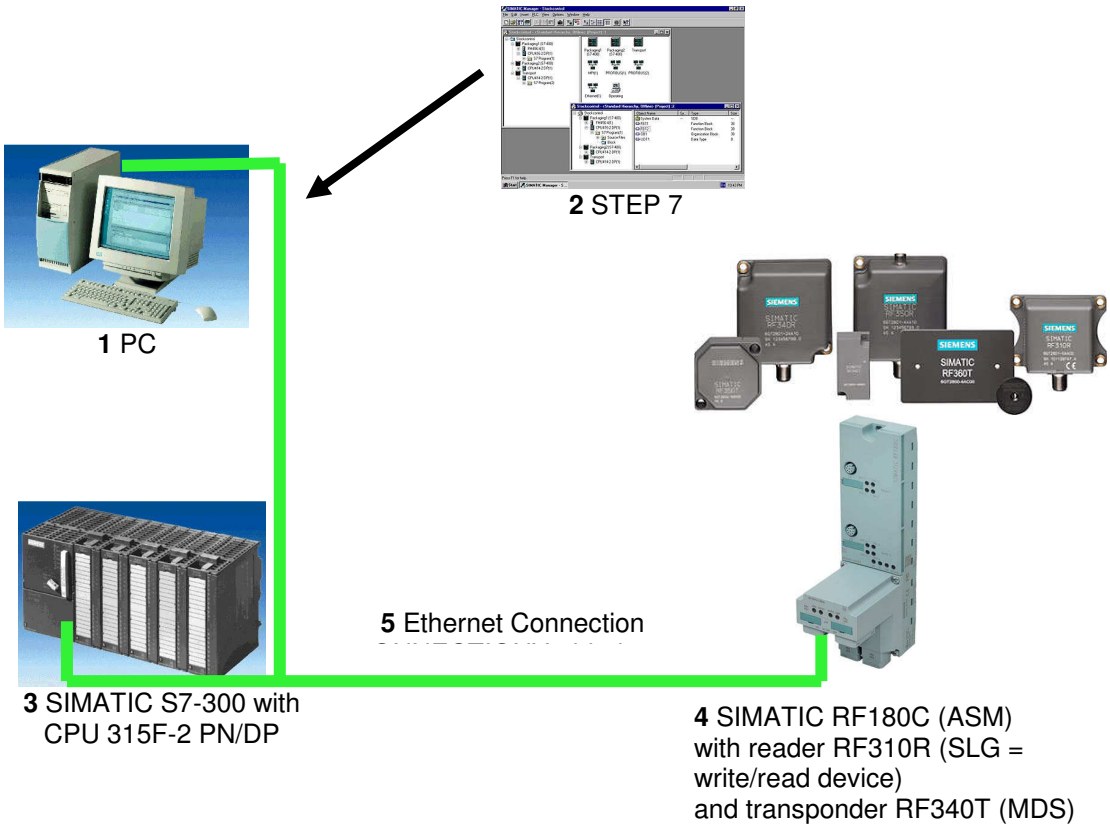
1 PC, operating system Windows XP Professional with SP2 or SP3/Vista 32 bit Ultimate and Business/Server 2003 SP2 with 600MHz (only XP)/1 GHz and 512MB (only XP)/1 GB RAM, free disk storage approx. 650 to 900 MB, MS Internet Explorer 6.0 and network card

2 Software STEP 7 V 5.4

3 PLC SIMATIC S7-300 with CPU 315F-2 PN/DP and at least one digital Input and output module
Sample configuration:
- Power supply: PS 307 2A
- CPU: CPU 315F-2 PN/DP
- Digital inputs: DI 16x24V DC
- Digital outputs: DO 16x24V DC/0.5 A

4 SIMATIC RF180C IM (interface module)
RF310R or RF340R SLG (write/read device)
Different transponder MDSs (Mobile Data System)
of the type:
RF340T (8 KB)
RF350T (32 KB)
RF360T (64 KB)
ISO Moby D MDS D124 (112 bytes)

5 Ethernet connection between PC, CPU 315F-2 PN/DP and RF180C



2 NOTES REGARDING THE USAGE OF CPU 315F-2 PN/DP

The CPU 315F-2 PN/DP is a CPU that is shipped with 2 integrated interfaces.

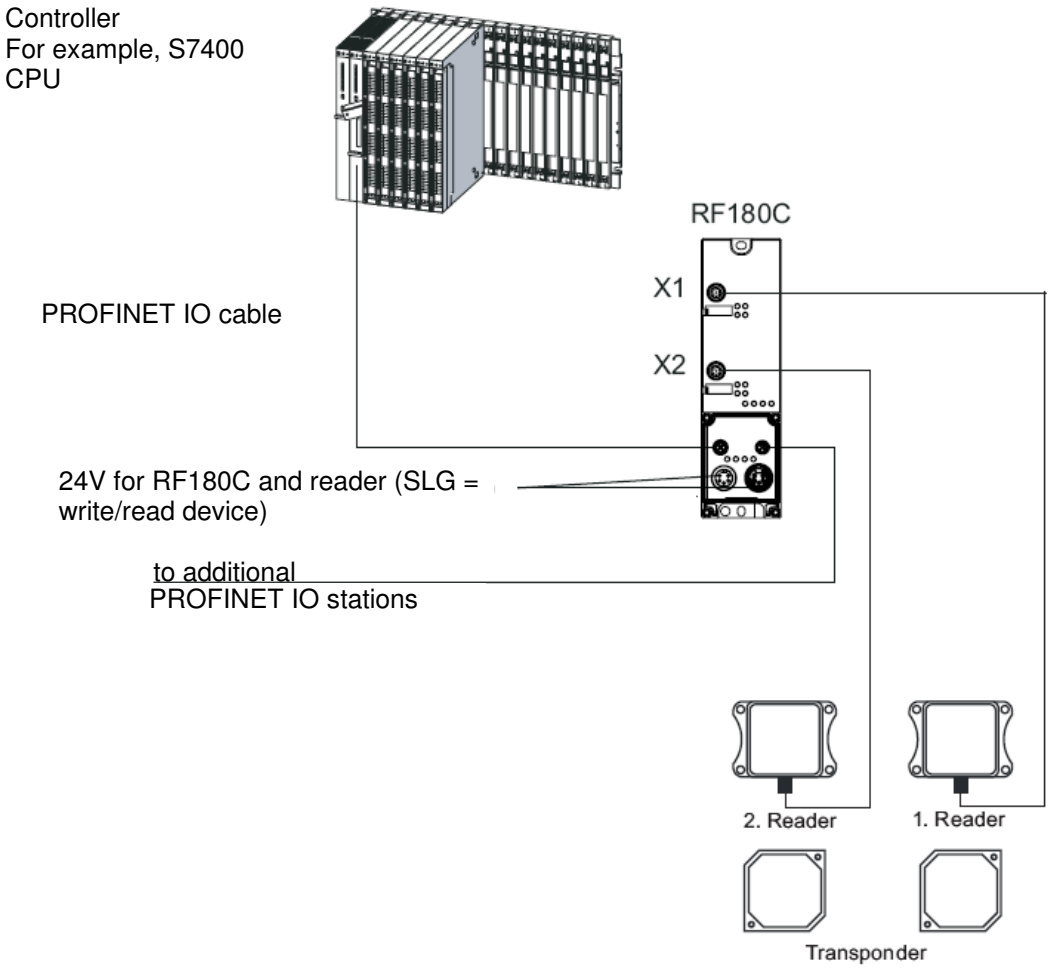
- The first interface is a combined MPI/PROFIBUS DP interface that can be used at the PROFIBUS DP as master or slave to connect distributed IO/field devices with very fast reaction time.
  In addition, it is possible to program the CPU here by means of an MPI or PROFIBUS DP
- The second interface is an integrated PROFINET interface.
  This allows for using the CPU as PROFINET IO controller to operate distributed IO on PROFINET. The CPU can be programmed by means of this interface also!
- Moreover, it is possible to use fail-safe IO devices on both interfaces.

Notes:

- In module E11, the CPU 315F-2 PN/DP is used as the controller for the data exchange of a SIMATIC RFID system on the PROFINET.
- To run this CPU, a micro-memory card is required!
- The addresses of the input and output modules can be parameterized at this CPU.

3 NOTES REGARDING THE SIMATIC RFID COMPONENTS

The communication module RF180C is a module for operating RFID components at any controller by means of PROFINET IO.
At the RF180C, up to 2 readers (SLG = write/read device) can be operated in parallel. The user can start a command in parallel on two readers (FB 45 if operated on a SIMATIC S7)
The tag data is accessed by means of addressing the tag physically.
In the SIMATIC S7, FB 45 is provided for this. FB 45 makes available to the S7user an interface that is easy to handle and is equipped with powerful commands (processing a complete tag with a single command; command chaining; S7 data structures by means of UDTs).



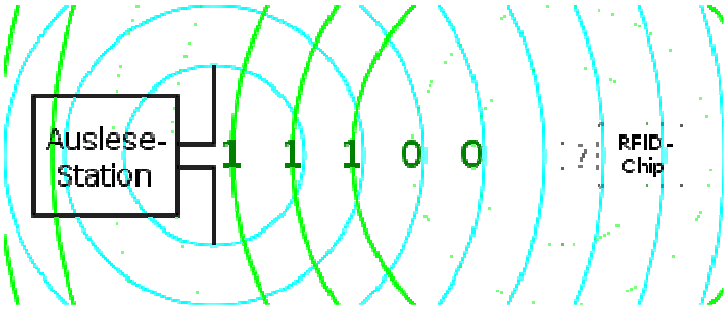
The RF180C is integrated into the hardware configuration by means of a GSDML file.
Then, the RF180C can be configured by means of HWConfig of the SIMATIC Manager.
The GSDML file is provided on the CD "RFID Systems".

Additional information is available in the operating instructions "RFID Systems Communication Module RF180C".

4 RFID FUNDAMENTALS

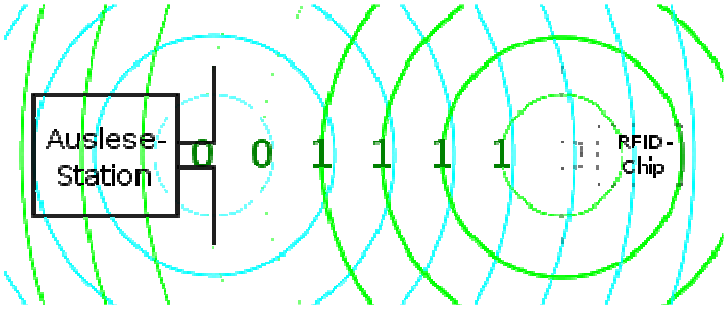
Radio Frequency Identification (**RFID**) makes it possible to automatically identify and localize objects and living beings, and thus considerably facilitates recording and storing data. The RFID system consists of the following: 1) a transponder that is located in the object or in the living being and identifies it, and 2) a reading device for reading out the transponder ID. The reading device includes a software (a micro-program) that controls the actual read process, and an RFID middleware with interfaces to other EDP systems and data bases.
As a rule, a read device generates an electro-magnetic high frequency field with a short range, preferably with induction coils. It is not only used to transmit data, but to also to supply the transponder with power. Only if larger ranges are to be obtained are active transponders used that have their own power supply. Usually, the frequency of 13.56 MHz is used (RF300, ISO). The reading device (reader) generates a high frequency electromagnetic alternating field that illuminates the aerial of the RFID transponder (RFID tag). As soon as the aerial coil enters the electro-magnetic field, an induction current is generated in it. This current is rectified, and with it, a capacitor is loaded as short time storage which, for the read process, provides for the power supply of the chip. For active tags, an installed battery takes care of the supply. The micro-chip thus activated in the RFID tag decodes the commands sent by the reader. This reader encodes and modulates the reply into the irradiated electro-magnetic field through field weakening in the contact-free short circuit, or in opposition reflection of the field that the reader transmitted. With this, the tag transmits its own unchangeable serial number, additional numbers of the marked object, or other data that the reader polled. The transponder itself does not transmit a field; it only changes the reader's electro-magnetic transmission field.

Readout Station



Transponder switch open, aerial is on "Absorption" (return value 0).

Readout Station



Transponder switch closed, aerial is on "Reflect", (return value 1).
(Source: Wikipedia)

5 STARTING UP AN RFID PROJECT WITH CPU 315F-2 PN/DP AND RF180C

Below, the startup of an RFID project is described.

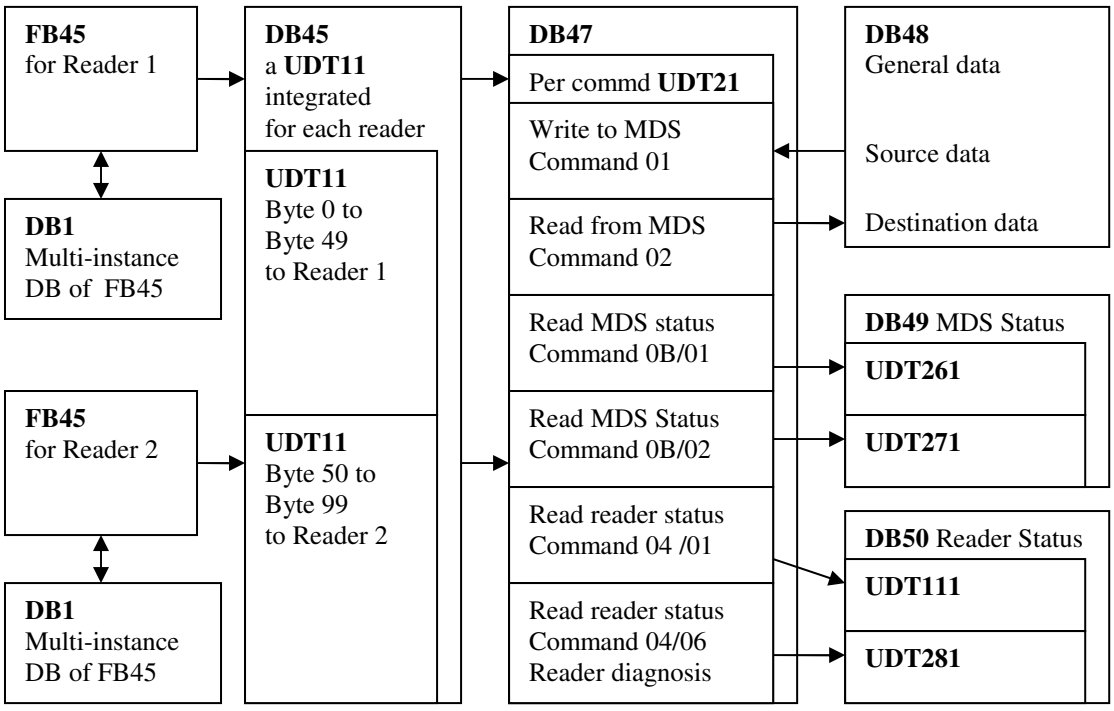
As SIMATIC S7-300 station, the CPU 315F-2 PN/DP is used.

In the CPU's control program, a data structure has to be generated -by means of a function block call (FB45)- with data blocks and embedded UDT data types.

In our example, it has to be possible to perform the following actions for each reader:

- Writing data to the transponder (DB48 of the CPU to the MDS).
- Reading the data from the transponder (MDS to DB48 of the CPU).
- Reading the transponder data information (MDS to DB49 of the CPU).
- Reading the reader status information (reader to DB50 of CPU).

The data structure of the German language UDTs is as follows:



To generate the data structure, the required UDT data types have to be imported to the Step 7 project and function block FB45.

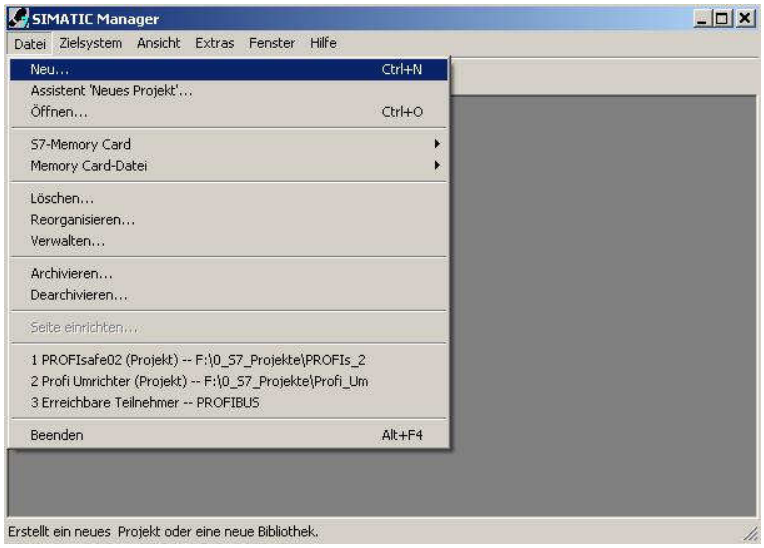
A sample program with the blocks is provided on the CD "RFID Systems".

Additional information about the data structure is available in the function manual "RFID Systems FB45"

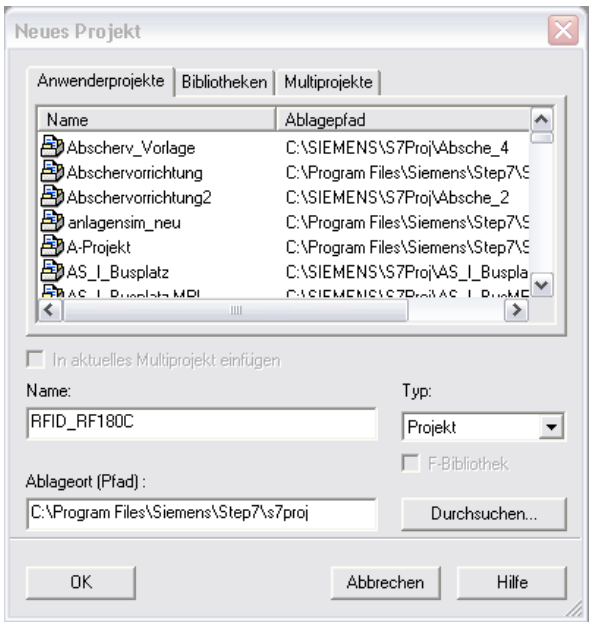
5.1 Setting Up a New Project

1. The central tool in STEP 7 is the **'SIMATIC Manager'**, which we call here with a double click. (→ SIMATIC Manager)

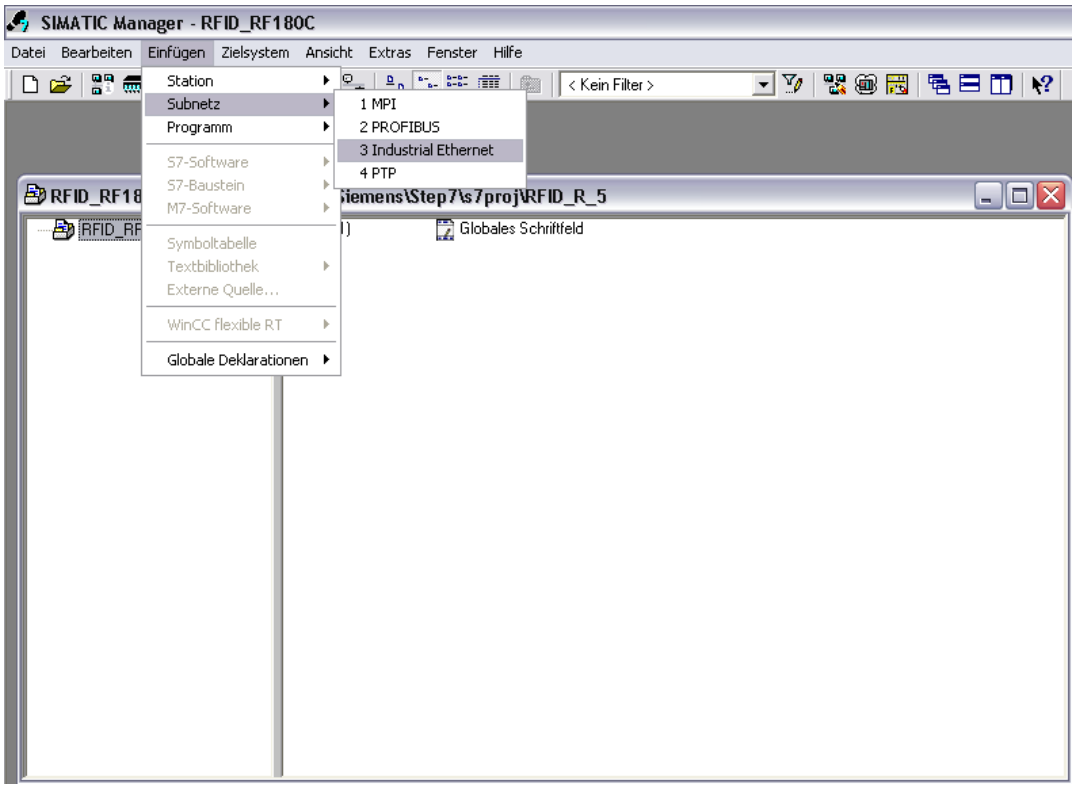
2. STEP 7 programs are managed in projects. We are now setting up such a project (→ File → New)



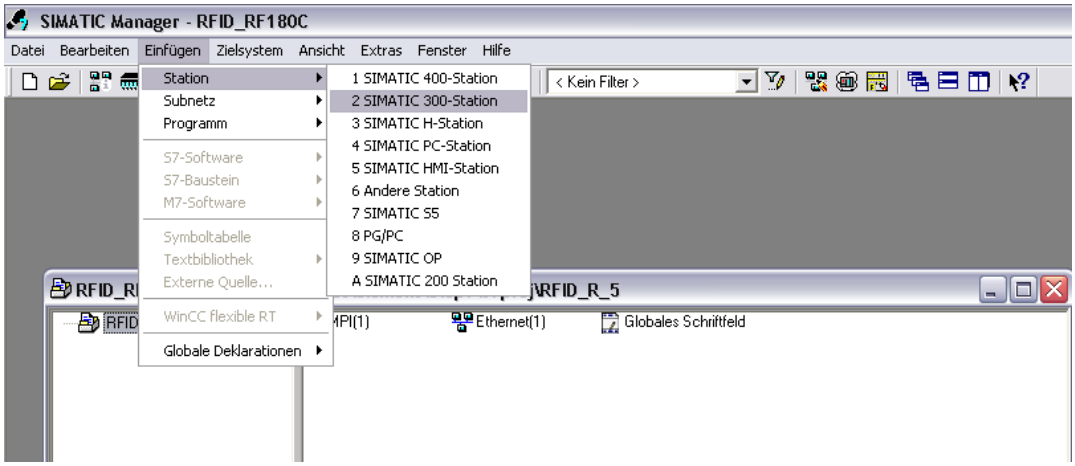
3. Next, we are assigning the **'Name' 'RFID_RF180C'** to the project (→ RFID_RF180C → OK)



4. Highlight your project and insert an '**Industrial Ethernet Subnet**' (→ RFID_RF180C → Insert → Subnet → Industrial Ethernet).

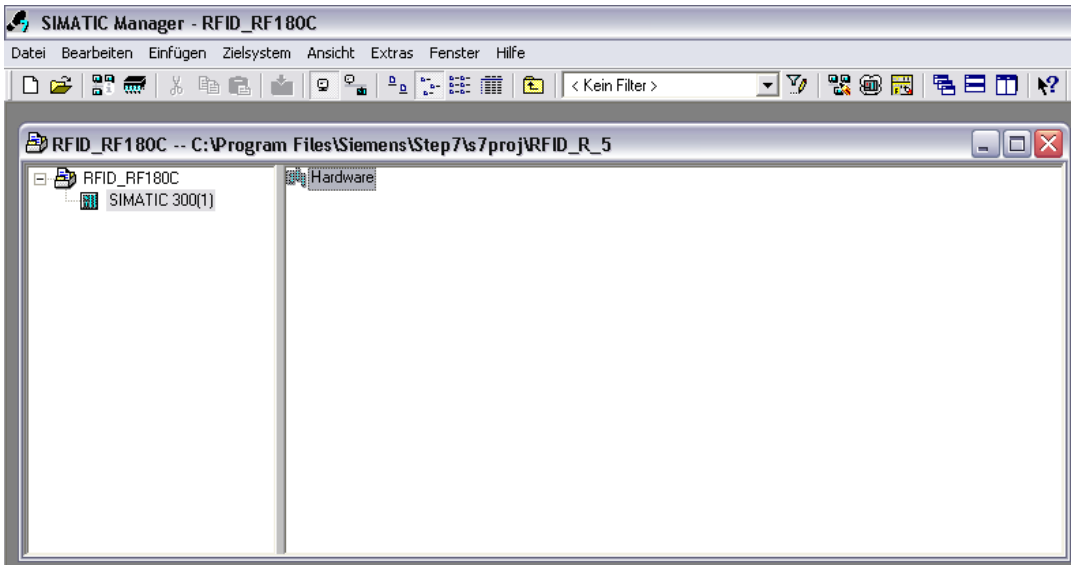


5. Then, we insert a '**SIMATIC 300 Station**'. (→ Insert → Station → SIMATIC 300 Station)

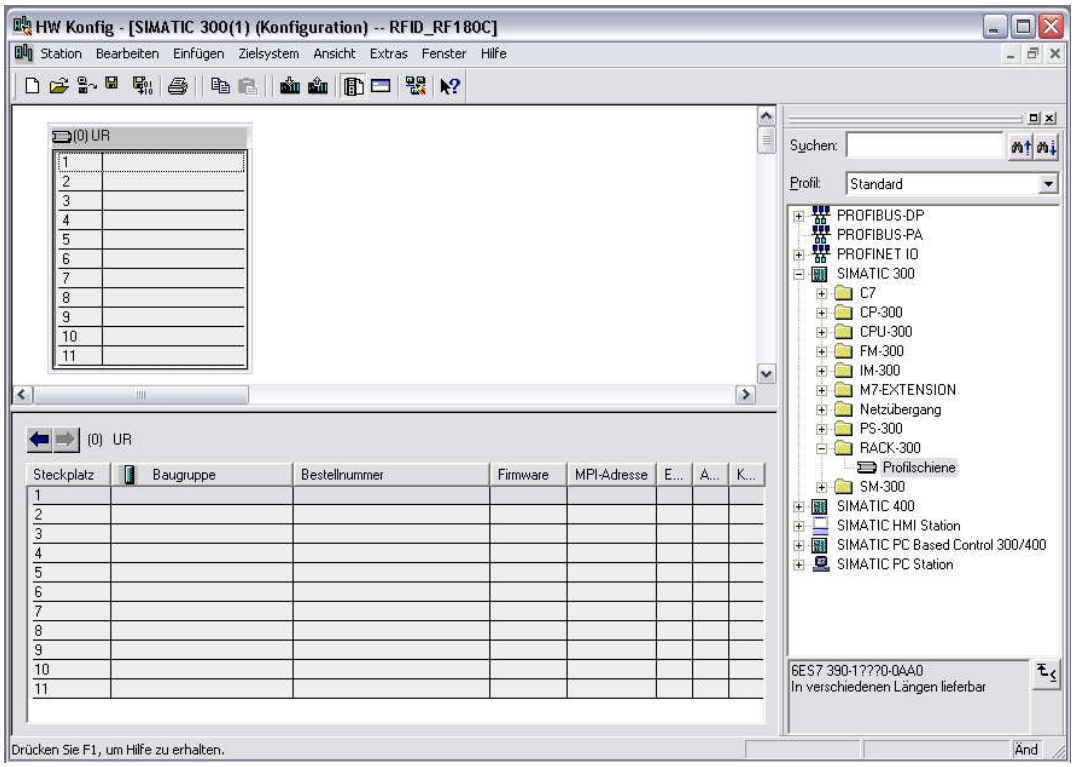


5.2 Configuring the Hardware

6. With a double click, open the configuration tool for the **'Hardware'**. (→ Hardware)

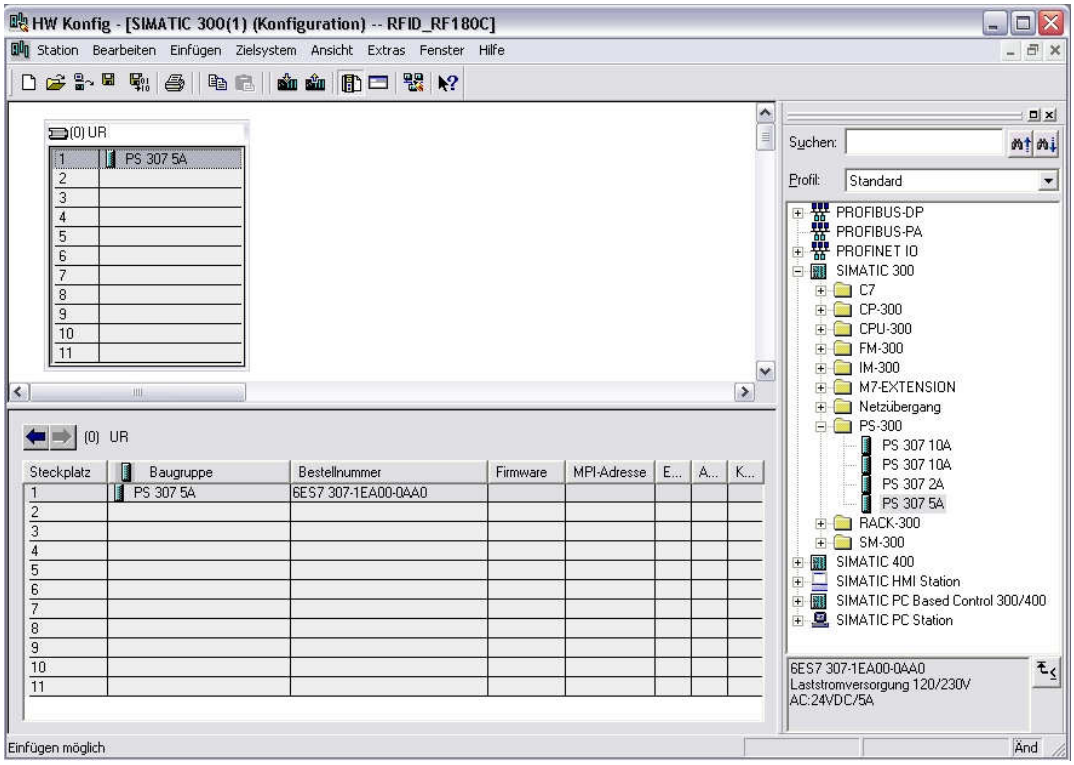


7. Open the hardware catalog by clicking on the symbol '  '. (→ )
Insert the **'Mounting channel'** with a double click (→ SIMATIC 300 → RACK 300 → Mounting channel).



Note
A configuration table for configuring Rack 0 is displayed automatically.

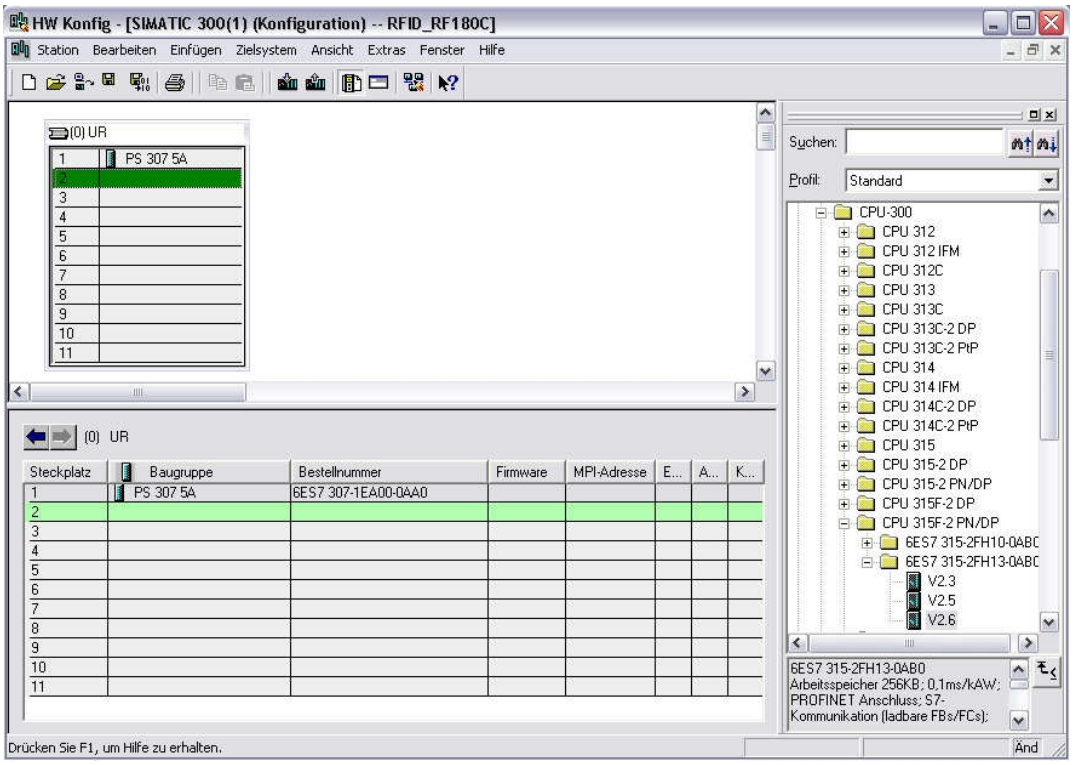
8. From the hardware catalog, we can now select all modules that are present in the actual rack and insert them in the configuration table. To this end, we click on the name of the respective module, hold the mouse key and drag it to a line in the configuration table.
   We start with the power unit **'PS 307 5A'** (→ SIMATIC 300 → PS-300 → PS 307 5A).



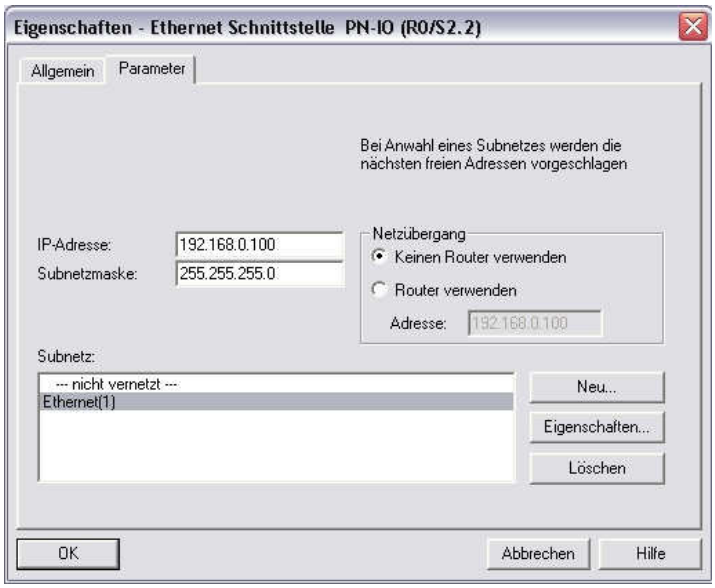
Note

If your hardware deviates from the one displayed here, simply select the corresponding modules from the catalog and insert them in your rack. The order numbers for the individual modules -that are also inscribed on the modules- are displayed in the footer of the catalog.

9. Next, we drag the **'CPU 315F-2 PN/DP'** to the second slot.
The order number and the version of the CPU are inscribed on the front of the CPU.
(→ SIMATIC 300 → CPU-300 → CPU 315F-2 PN/DP → 6ES7 315-2FH13-0AB0 → V2.6)



10. When entering the CPU, the following window appears. In this window, we do the following: assign to CPU 315F-2 PN/DP an **'IP- address'**, specify the **'Subnet screen form'** and select the **'Ethernet'** that has already been set up. Optionally, a **'Router address'** can be selected for network-overarching communication. Confirm your entries with **'OK'** (→ IP address: 192.168.0.100 → subnet screen form: 255.255.255.0 → Ethernet(1) → Don't use a router → OK)



Notes regarding networking on the Ethernet (additional information is provided in Appendix V of the training manual):

MAC address:
The MAC address consists of a permanent and a variable part. The permanent part ("Basic MAC address") identifies the manufacturer (Siemens, 3COM, ...). The variable part of the MAC address differentiates the different Ethernet stations and should be assigned globally unique. On each module, a MAC address specified by the factory is inscribed.

Value range for the IP address:
The IP address consists of 4 decimal numbers in the value range 0 to 255, separated by a period; for example: 141.80.0.16

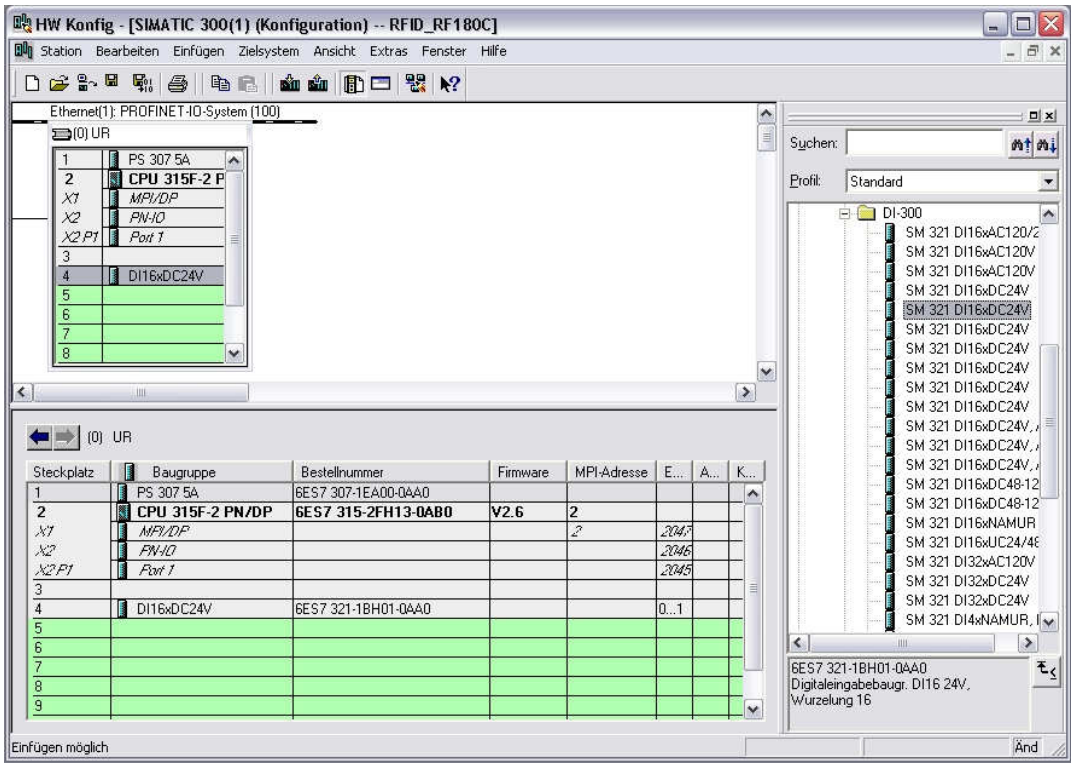
Value range for the subnet screen form:
This screen form is used to establish whether a station or its IP address belongs to the local subnet, or can be reached only by means of a router.
The subnet screen form consists of 4 decimal numbers in the value range 0 to 255, separated by a period; for example: 255.255.0.0
The 4 decimal numbers of the subnet screen form have to contain -in their binary representation- from the left a series of gapless values "1" and from the right a series of gapless values "0"..
The values "1" determine the area of the IP address for the network number. The values "0" determine the area of the IP address for the station address.
Example:

Correct values:	255.255.0.0 Decimal = 1111 1111.1111 1111.0000 0000.0000 0000 binary
	255.255.128.0 Decimal = 1111 1111.1111 1111.1000 0000.0000 0000 binary
	255.254.0.0 Decimal = 1111 1111.1111 1110.0000 0000.0000.0000 binary
Incorrect value:	255.255.1.0 Decimal = 1111 1111.1111 1111.0000 0001.0000 0000 binary

Value range for the address of the gateway (router):
The address consists of 4 decimal numbers in the value range 0 to 255 separated by a period; for example, 141.80.0.1.

Relationship of the IP addresses, router address and subnet screen form:
The IP address and the gateway address must differ only at those positions where an "0" is shown in the subnet screen form.
Example:
You entered the following: for subnet screen form 255.255.255.0; for IP address 141.30.0.5 and for router address 141.30.128.1.
The value for the IP address and the gateway address is to differ only in the 4th decimal number. However, in the example, the 3rd position already differs.
In the example, we have to alternatively change:
- the subnet screen form to: 255.255.0.0 or
- the IP address to: 141.30.128.5 or
- the gateway address to: 141.30.0.1

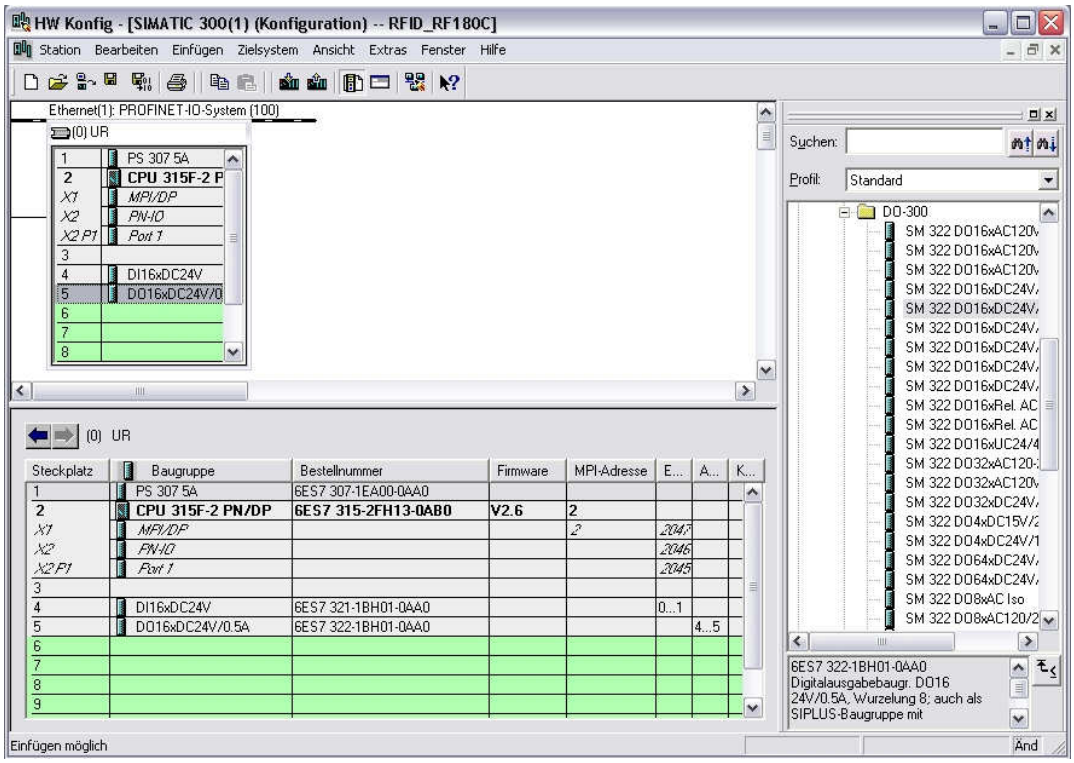
11. Next, we are dragging the input module for 16 inputs to the 4th slot. The module's order number is located on the front. (→ SIMATIC 300 → DI-300 → SM 321 DI16x24VDC).



Note

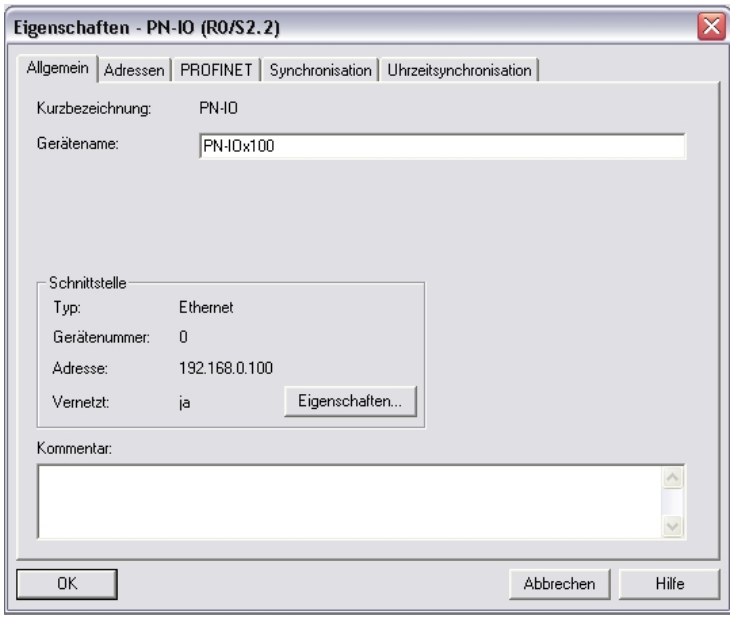
Slot 3 is reserved for interface modules and remains empty for that reason. The module's order number is indicated in the footer of the catalog.

12. Now we drag the output module for 16 outputs to the 5th slot. The module's order number is located on the front (→ SIMATIC-300 → DO-300 → SM 322 DO16x24VDC/0.5A).

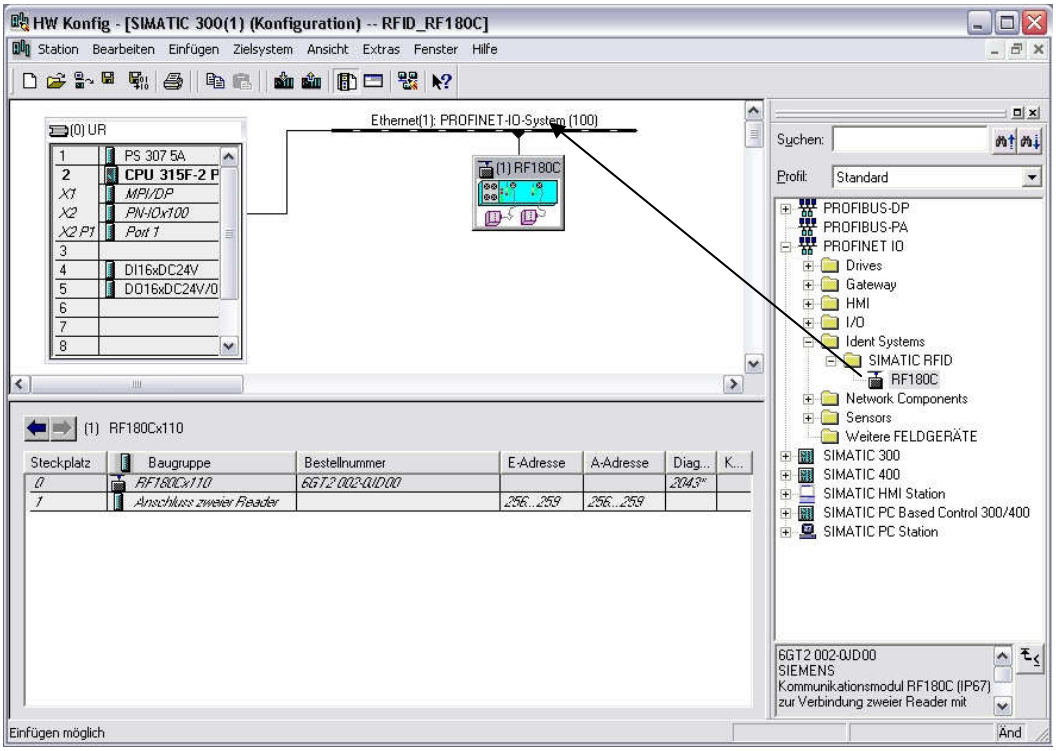


Note
The module's order number is indicated in the footer of the catalog.

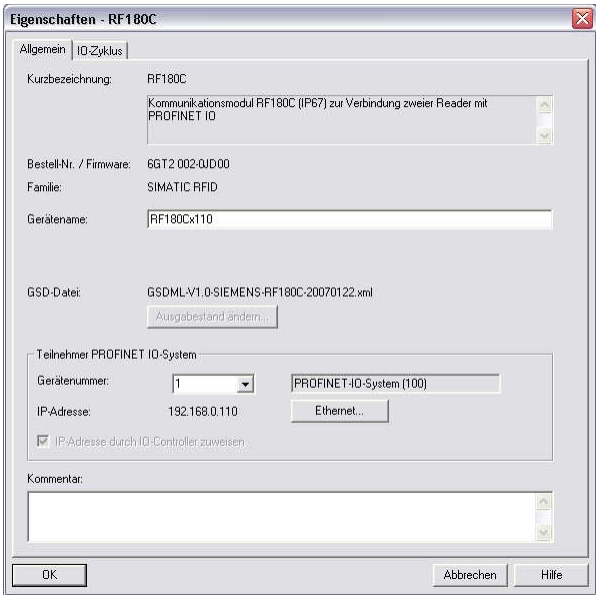
13. Now, we have to change the PROFINET device name to PN IOx100. Select '**PN-IO**' with a double click. (→ PN-IO,→ PN-IOx100,→ OK)



14. Now, drag the PROFINET IO System (100) module tier toward the right and from the folder PROFINET IO, insert the SIMATIC RFID module RF180C into the module tier by dragging it there.
    If module RF180C should not yet be selectable, it first has to be inserted by a data carrier, using the menu "Options“ Install GSD files.



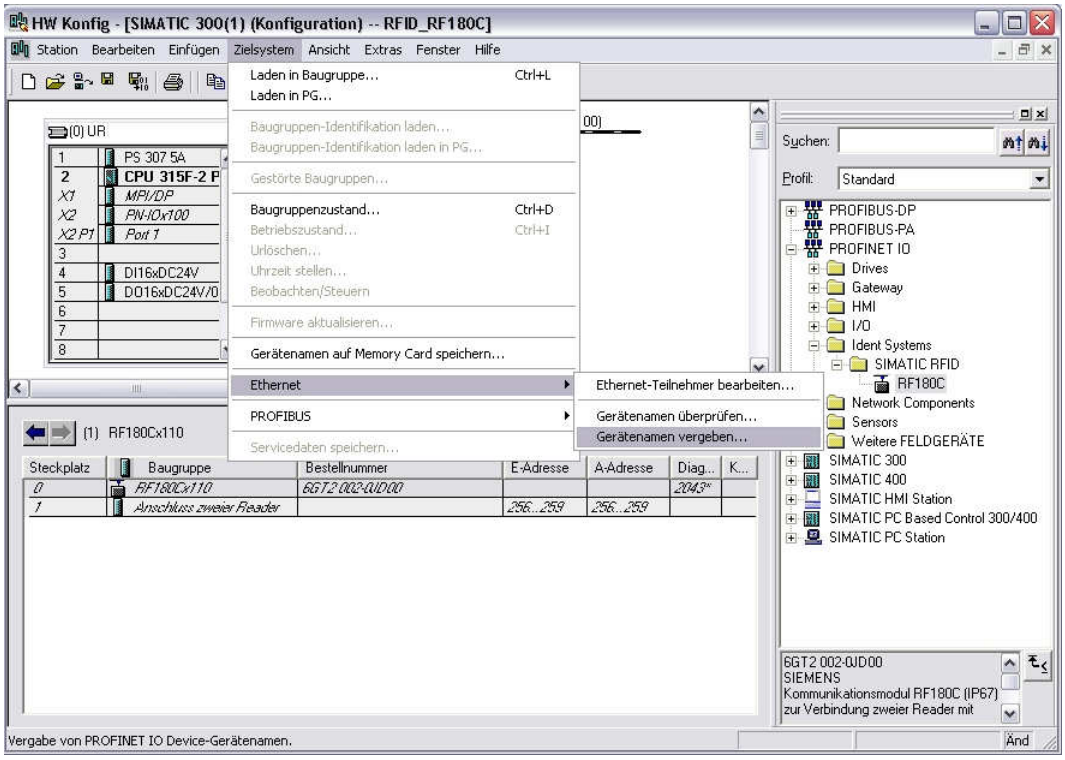
15. Then, double click on the inserted module and change the device name to RF180Cx110 and the IP address to 192.168.0.110



16. By clicking on '  ', the hardware configuration is saved and compiled.

5.3 Assigning a Device Name

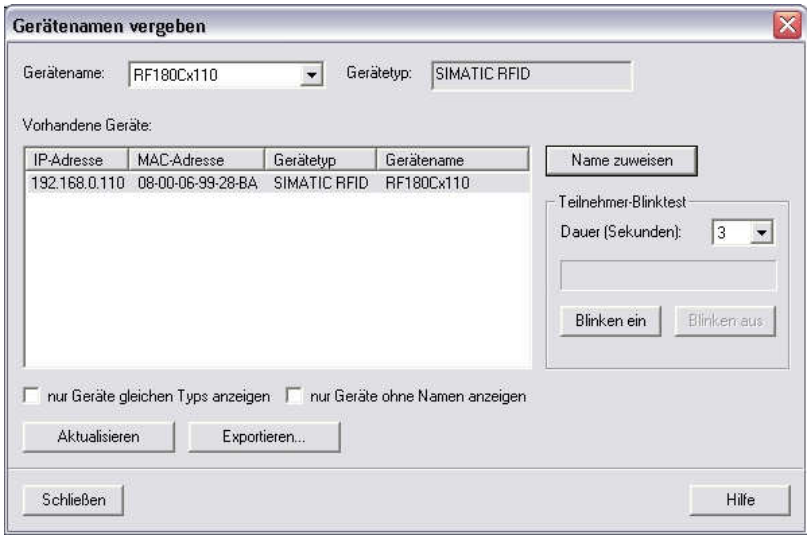
17. First, highlight the module RF180C and then select, under the menu "Destination system“ and under Ethernet **Assign device name**.



Note

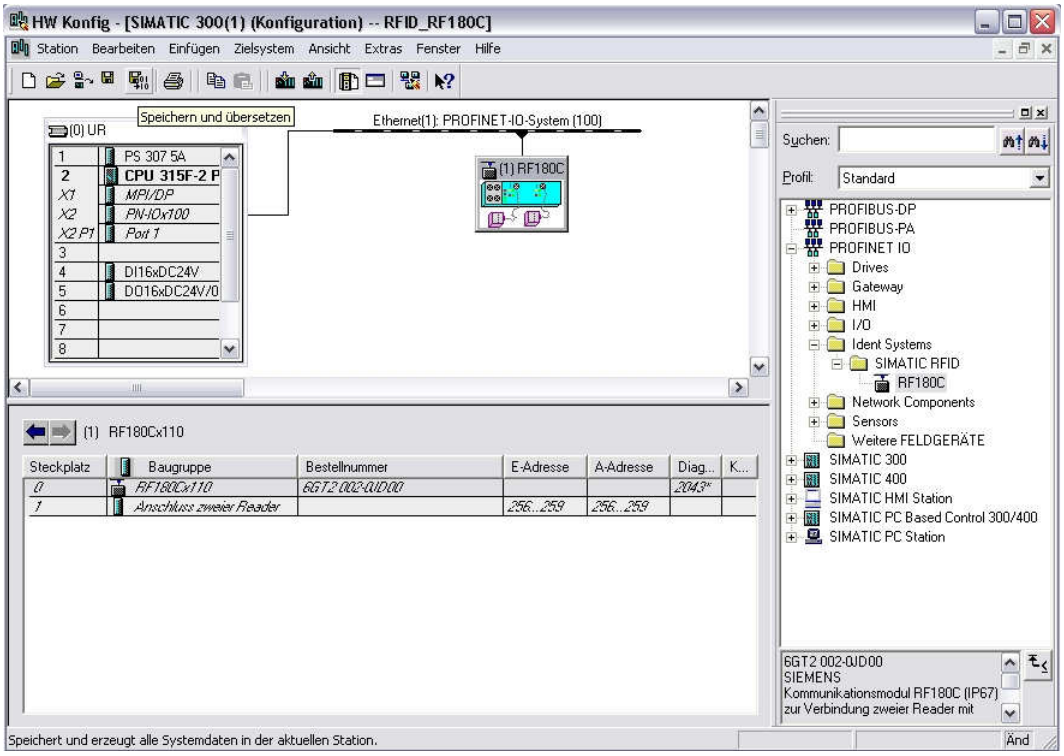
A prerequisite for this is that the die PG/PC interface is set to TCP/IP and the PC’s network card is configured correctly. For example, IP address 192.168.0.99, subnet 255.255.255.0 and router address -.-.-.- (refer to Module E02)

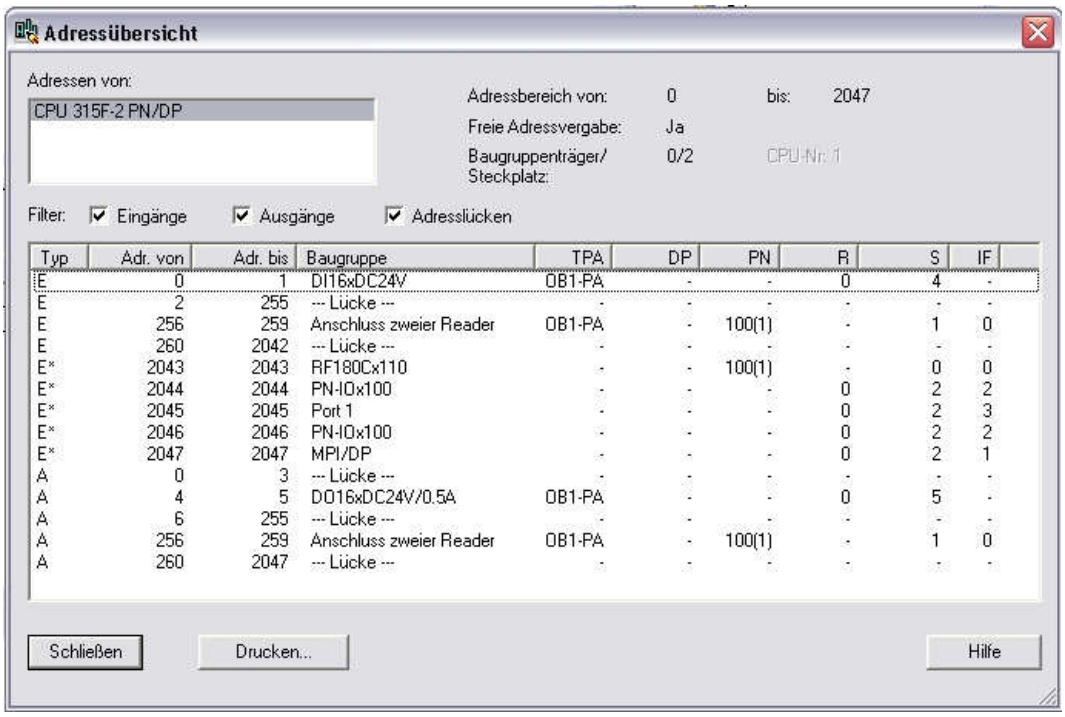
18. Highlight the SIMATIC RFID module and then click on the button “Assign name“. Then close the window.



This is what the completed hardware configuration with the associated addresses looks like

19. By clicking on '  ' we can load the hardware configuration to the PLC. The operating mode switch on the CPU should be on Stop (→ ).

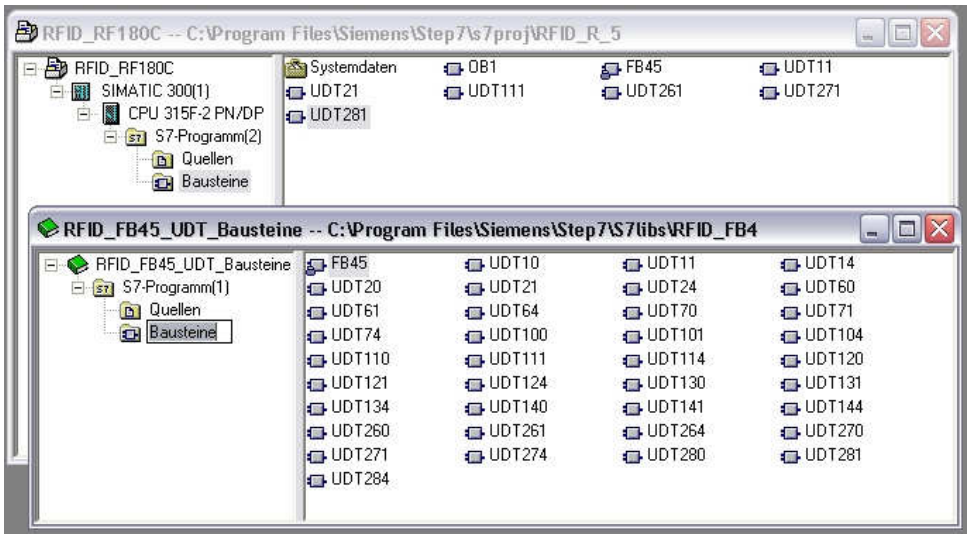




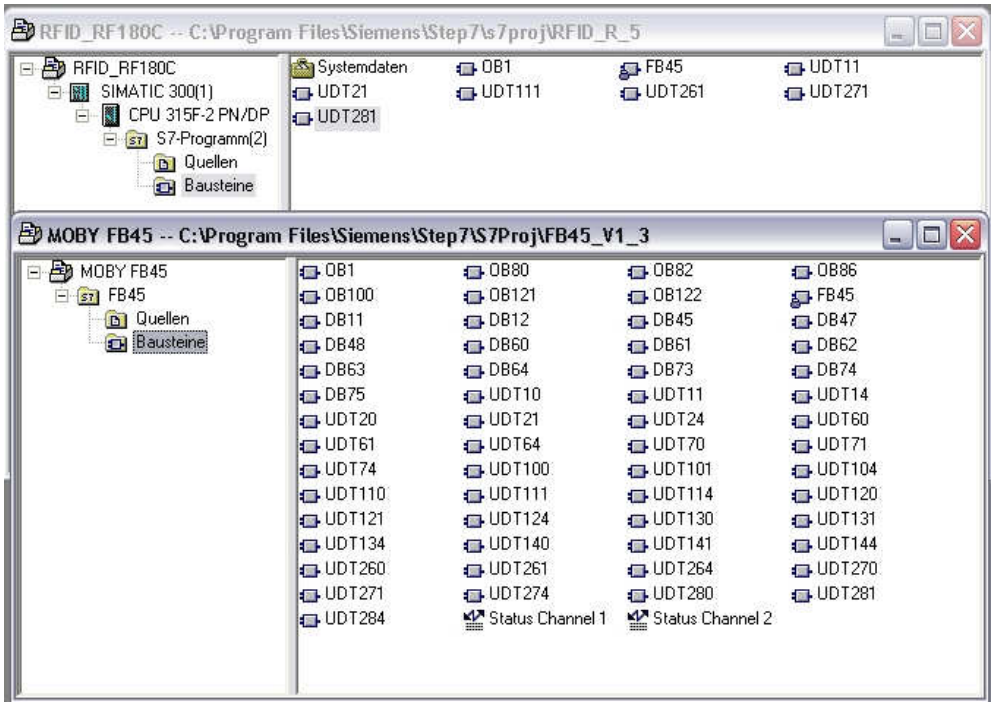
20. Close hardware configuration.

5.4 Inserting UDT Blocks and FB45

De-archive the library "**RFID_FB45_UDT_Blocks**" from the template directory and open it.
Copy **UDT11**, **UDT21, UDT 111, UDT261, UDT271, UDT281** and **FB45** and insert them in the block folder.
Close the library.



Note
Instead of the library **RFID_FB45_UDT_Blocks**, the sample program MOBY FB45 can be de-archived. The file **FB45_V1_3.zip** is available on the RFID Systems CD in the directory "Data", sub-directory "FB45".

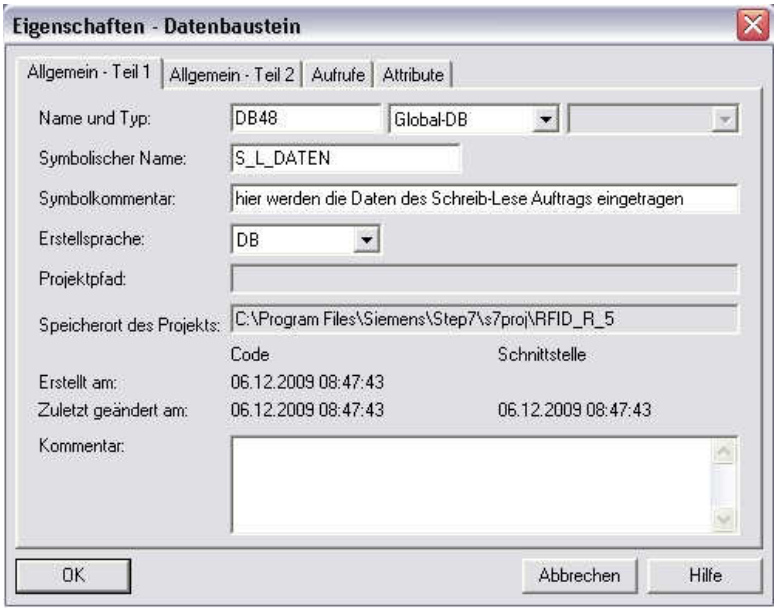


5.5 Generating Data Blocks

DB48 Write/Read Data

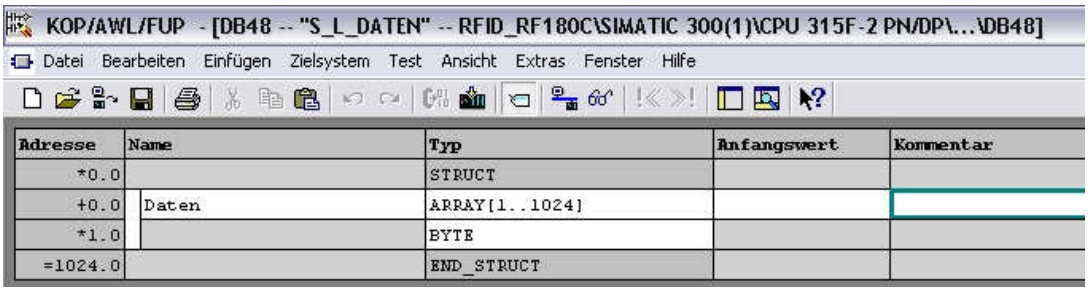
Here, the source data of a write request to the transponder is stored, or the destination data for the read request from the transponder.

Generating DB48



Open DB48, set up 1024 bytes

Under Name, enter “Data“.
At Type Combined data, select (with the right mouse key) the ARRAY data format.
Enter **1..1024** within the brackets.
In the next line, select or enter BYTE.

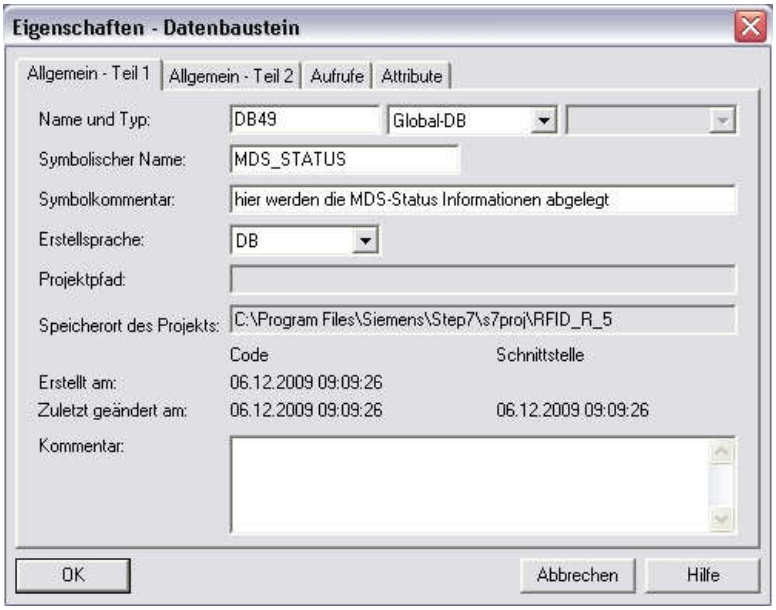


Save and close DB48

DB49 MDS Status Information

In DB49, the MDS status Information is entered.
The information is stored in a data block with a specified structure.
Read MDS status with Sub_Command 01according to UDT261 or Sub_Command 02 according to UDT271.

Generating DB49



Open DB49 and insert UDT261 and UDT271

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	MDS_Status0	"MOBY P MDS-Status 0_d"		hier ist der UDT261 abgelegt
+18.0	MDS_Status1	"MOBY P MDS-Status 1_d"		hier ist der UDT271 abgelegt
=36.0		END_STRUCT		

DB49 data view

Adresse	Name	Typ	Anfangswert	Aktualwert	Kommentar
0.0	MDS_Status0.reserved0	BYTE	B#16#0	B#16#0	
1.0	MDS_Status0.status_info	BYTE	B#16#0	B#16#0	Modus MDS-Status
2.0	MDS_Status0.UID[1].Byte_1_4	DWORD	DW#16#0	DW#16#0	MDS-Nummer (unique identifier)
6.0	MDS_Status0.UID[1].Byte_5_8	DWORD	DW#16#0	DW#16#0	
10.0	MDS_Status0.MDS_type	BYTE	B#16#0	B#16#0	MDS-Typ
11.0	MDS_Status0.Lock_state	BYTE	B#16#0	B#16#0	Schreibschutzstatus EEPROM
12.0	MDS_Status0.reserved1[1]	BYTE	B#16#0	B#16#0	
13.0	MDS_Status0.reserved1[2]	BYTE	B#16#0	B#16#0	
14.0	MDS_Status0.reserved1[3]	BYTE	B#16#0	B#16#0	
15.0	MDS_Status0.reserved1[4]	BYTE	B#16#0	B#16#0	
16.0	MDS_Status0.reserved1[5]	BYTE	B#16#0	B#16#0	
17.0	MDS_Status0.reserved1[6]	BYTE	B#16#0	B#16#0	
18.0	MDS_Status1.reserved0	BYTE	B#16#0	B#16#0	
19.0	MDS_Status1.status_info	BYTE	B#16#0	B#16#0	Modus MDS-Status
20.0	MDS_Status1.UID[1].Byte_1_4	DWORD	DW#16#0	DW#16#0	MDS-Nummer (unique identifier)
24.0	MDS_Status1.UID[1].Byte_5_8	DWORD	DW#16#0	DW#16#0	
28.0	MDS_Status1.LFD	BYTE	B#16#0	B#16#0	Leistungsflussdichte: Beziehung zwischen Grenzwert und gemessenem Wert
29.0	MDS_Status1.FZP	BYTE	B#16#0	B#16#0	Fehlerzähler passiv (Ruhefehlerzähler)
30.0	MDS_Status1.FZA	BYTE	B#16#0	B#16#0	Fehlerzähler aktiv (Fehler während Kommunikation)
31.0	MDS_Status1.ANWZ	BYTE	B#16#0	B#16#0	Anwesenheitszähler
32.0	MDS_Status1.reserved1[1]	BYTE	B#16#0	B#16#0	
33.0	MDS_Status1.reserved1[2]	BYTE	B#16#0	B#16#0	
34.0	MDS_Status1.reserved1[3]	BYTE	B#16#0	B#16#0	

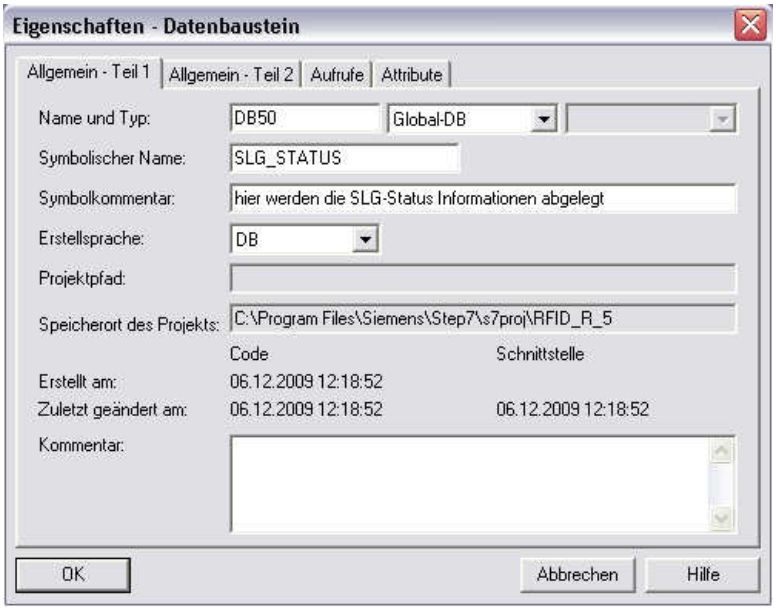
The start address in DB49 is "**0**" for Sub_Command 01 and "**18**" for Sub_Command 02.
The data length is 18 bytes in both cases

Save and close DB49

DB50 Reader Status Information

The reader status information is entered in DB50.
Depending on the request, the information has to be stored in a data block with a specified data structure.
Read reader status with Sub_Command 01 according to UDT111 or Sub_Command 06 according to UDT281.

Generating DB50



Open DB50 and insert UDT111 and UDT281

Open DB50 and insert UDT111 for Reader_Status.
For Reader_diagnosis, insert UDT281.

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	SLG_Status	"MOBY SLG-Status_d"		hier ist der UDT111 abgelegt
+28.0	SLG_Diagnose	"MOBY P SLG-Status_d"		hier ist der UDT281 abgelegt
=56.0		END_STRUCT		

Data view of the Reader status in DB50 (UDT111) Byte 0 to 27

Adresse	Name	Typ	Anfangswert	Aktualwert	Kommentar
0.0	SLG_Status.status_info	BYTE	B#16#0	B#16#0	Modus SLG-Status
1.0	SLG_Status.hardware	CHAR	' '	' '	HW-Variante
2.0	SLG_Status.hardware_version	WORD	W#16#0	W#16#0	HW-Version
4.0	SLG_Status.loader_version	WORD	W#16#0	W#16#0	Urlader-Version
6.0	SLG_Status.firmware	CHAR	' '	' '	FW-Variante
8.0	SLG_Status.firmware_version	WORD	W#16#0	W#16#0	FW-Version
10.0	SLG_Status.driver	CHAR	' '	' '	Treiber-Variante
12.0	SLG_Status.driver_version	WORD	W#16#0	W#16#0	Treiber-Version
14.0	SLG_Status.interface	BYTE	B#16#0	B#16#0	Schnittstelle (RS 232 / RS 422)
15.0	SLG_Status.baud	BYTE	B#16#0	B#16#0	Baudrate
16.0	SLG_Status.reserved1	BYTE	B#16#0	B#16#0	Reserviert
17.0	SLG_Status.reserved2	BYTE	B#16#0	B#16#0	Reserviert
18.0	SLG_Status.reserved3	BYTE	B#16#0	B#16#0	Reserviert
19.0	SLG_Status.distance_limiting_SLG	BYTE	B#16#0	B#16#0	Reichweitenbegrenzung
20.0	SLG_Status.multitag_SLG	BYTE	B#16#0	B#16#0	Multitag SLG
21.0	SLG_Status.field_ON_control_SLG	BYTE	B#16#0	B#16#0	Berobetriebsart
22.0	SLG_Status.field_ON_time_SLG	BYTE	B#16#0	B#16#0	Berozeit
23.0	SLG_Status.sync_SLG	BYTE	B#16#0	B#16#0	Semaphorensteuerung (Synchronisation mit SLG)
24.0	SLG_Status.status_ant	BYTE	B#16#0	B#16#0	Status Antenne
25.0	SLG_Status.stand_by	BYTE	B#16#0	B#16#0	Standby Zeit nachdem ein Befehl ausgeführt wurde
26.0	SLG_Status.MDS_control	BYTE	B#16#0	B#16#0	Anwesenheit

Data view of the Reader diagnosis in DB50 (UDT281) Byte 28 to 55

28.0	SLG_Diagnose.status_info	BYTE	B#16#0	B#16#0	Modus SLG-Status
29.0	SLG_Diagnose.FZP	BYTE	B#16#0	B#16#0	Fehlerzähler passiv (Ruhefehlerzähler)
30.0	SLG_Diagnose.ABZ	BYTE	B#16#0	B#16#0	Abbruchzähler
31.0	SLG_Diagnose.CFZ	BYTE	B#16#0	B#16#0	Codefehlerzähler
32.0	SLG_Diagnose.SFZ	BYTE	B#16#0	B#16#0	Signaturfehlerzähler
33.0	SLG_Diagnose.CRCFZ	BYTE	B#16#0	B#16#0	CRC-Fehlerzähler
34.0	SLG_Diagnose.BSTAT	BYTE	B#16#0	B#16#0	Befehlsstatus
35.0	SLG_Diagnose.ASMFZ	BYTE	B#16#0	B#16#0	Fehlerzähler Hostinterface (ASM)
36.0	SLG_Diagnose.reserved0[1]	BYTE	B#16#0	B#16#0	
37.0	SLG_Diagnose.reserved0[2]	BYTE	B#16#0	B#16#0	
38.0	SLG_Diagnose.reserved0[3]	BYTE	B#16#0	B#16#0	
39.0	SLG_Diagnose.reserved0[4]	BYTE	B#16#0	B#16#0	
40.0	SLG_Diagnose.reserved0[5]	BYTE	B#16#0	B#16#0	
41.0	SLG_Diagnose.reserved0[6]	BYTE	B#16#0	B#16#0	
42.0	SLG_Diagnose.reserved0[7]	BYTE	B#16#0	B#16#0	
43.0	SLG_Diagnose.reserved0[8]	BYTE	B#16#0	B#16#0	
44.0	SLG_Diagnose.reserved0[9]	BYTE	B#16#0	B#16#0	
45.0	SLG_Diagnose.reserved0[10]	BYTE	B#16#0	B#16#0	
46.0	SLG_Diagnose.reserved0[11]	BYTE	B#16#0	B#16#0	
47.0	SLG_Diagnose.reserved0[12]	BYTE	B#16#0	B#16#0	
48.0	SLG_Diagnose.reserved0[13]	BYTE	B#16#0	B#16#0	
49.0	SLG_Diagnose.reserved0[14]	BYTE	B#16#0	B#16#0	
50.0	SLG_Diagnose.reserved0[15]	BYTE	B#16#0	B#16#0	
51.0	SLG_Diagnose.reserved0[16]	BYTE	B#16#0	B#16#0	
52.0	SLG_Diagnose.reserved0[17]	BYTE	B#16#0	B#16#0	
53.0	SLG_Diagnose.reserved0[18]	BYTE	B#16#0	B#16#0	
54.0	SLG_Diagnose.reserved0[19]	BYTE	B#16#0	B#16#0	
55.0	SLG_Diagnose.reserved0[20]	BYTE	B#16#0	B#16#0	

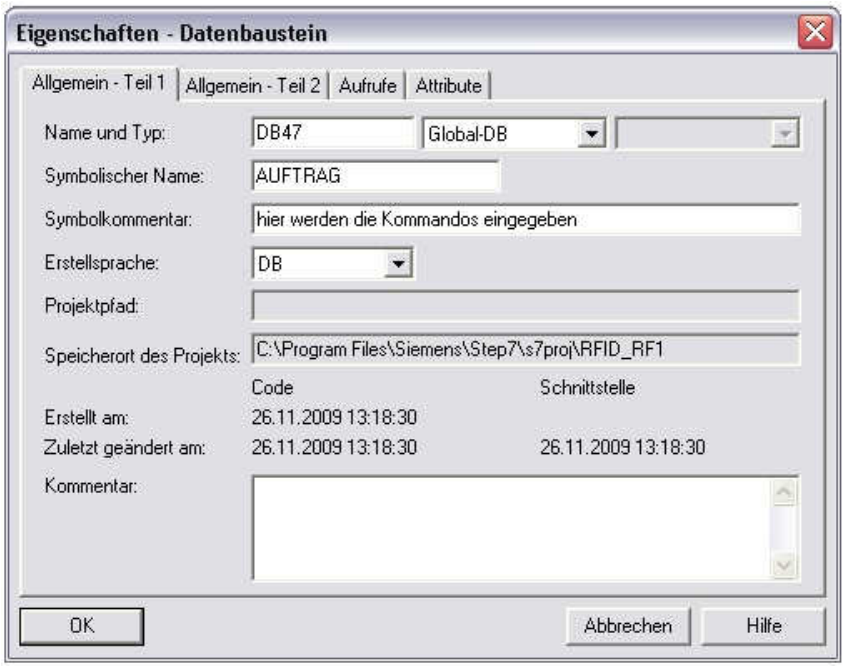
For Sub_Command 01, the start address in DB50 is "**0**“ Reader status information and for Sub_Command 06 "**28**“ Reader diagnosis information.
The data length in both cases is 28 bytes.

Save and close DB50

DB47 Request Data Block

The DB47 is set by means of the UDT11 in DB45.
In DB47, a UDT21 is embedded for each reader.
The request commands are then entered in the respective UDT21.
The data of the read/write request is then stored in DB48.
The reference to DB48 is assigned in UDT21 of DB47.

Generate DB47



Open DB47 and add UDT21 for each request

Open DB47 and for Reader1, insert the UDT21 five times by means of ARRAY format.
For Reader2, also insert UDT21 five times by means of ARRAY format.

Adresse	Name	Typ	Anfangswert	Kommentar
*0.0		STRUCT		
+0.0	Kanal_1_Befehl	ARRAY[1..5]		SLG1 Aufträge 1 bis 5
*10.0		"MOBY CMD_d"		
+50.0	Kanal_2_Befehl	ARRAY[1..5]		SLG2 Aufträge 1 bis 5
*10.0		"MOBY CMD_d"		
=100.0		END_STRUCT		

Note
In our example, five request commands are possible for each reader or channel.
Thus, the UDT21 is embedded in DB47 10 times.
The commands or requests of the second reader start with address 50.

Data View of DB47

Adresse	Name	Typ	Anfangswert	Aktualwert	Kommentar
0.0	Kanal_1_Befehl[1].command	BYTE	B#16#2	B#16#2	MDS Befehl:1=schreiben,2=lesen,3=init,4=slg-status,8=end,A=set-ant,B=mds-status
1.0	Kanal_1_Befehl[1].sub_command	BYTE	B#16#0	B#16#0	Bitmuster für INIT-Befehl; Modus für END,SET-ANT,MDS-STATUS,SLG-STATUS
2.0	Kanal_1_Befehl[1].length	INT	1	1	Anzahl der zu schreibenden/lesenden Daten in Bytes
4.0	Kanal_1_Befehl[1].address_MDS	WORD	W#16#0	W#16#0	Anfangsadresse auf MDS; Endadresse bei INIT;KW/Jahr für MDS-STATUS
6.0	Kanal_1_Befehl[1].DAT_DB_number	INT	48	48	Nummer des DAT DB; Daten DB für MDS Daten
8.0	Kanal_1_Befehl[1].DAT_DB_address	INT	0	0	Zeiger auf das Anfangswort im DAT DB
10.0	Kanal_1_Befehl[2].command	BYTE	B#16#2	B#16#2	MDS Befehl:1=schreiben,2=lesen,3=init,4=slg-status,8=end,A=set-ant,B=mds-status
11.0	Kanal_1_Befehl[2].sub_command	BYTE	B#16#0	B#16#0	Bitmuster für INIT-Befehl; Modus für END,SET-ANT,MDS-STATUS,SLG-STATUS
12.0	Kanal_1_Befehl[2].length	INT	1	1	Anzahl der zu schreibenden/lesenden Daten in Bytes
14.0	Kanal_1_Befehl[2].address_MDS	WORD	W#16#0	W#16#0	Anfangsadresse auf MDS; Endadresse bei INIT;KW/Jahr für MDS-STATUS
16.0	Kanal_1_Befehl[2].DAT_DB_number	INT	48	48	Nummer des DAT DB; Daten DB für MDS Daten
18.0	Kanal_1_Befehl[2].DAT_DB_address	INT	0	0	Zeiger auf das Anfangswort im DAT DB
20.0	Kanal_1_Befehl[3].command	BYTE	B#16#2	B#16#2	MDS Befehl:1=schreiben,2=lesen,3=init,4=slg-status,8=end,A=set-ant,B=mds-status
21.0	Kanal_1_Befehl[3].sub_command	BYTE	B#16#0	B#16#0	Bitmuster für INIT-Befehl; Modus für END,SET-ANT,MDS-STATUS,SLG-STATUS
22.0	Kanal_1_Befehl[3].length	INT	1	1	Anzahl der zu schreibenden/lesenden Daten in Bytes
24.0	Kanal_1_Befehl[3].address_MDS	WORD	W#16#0	W#16#0	Anfangsadresse auf MDS; Endadresse bei INIT;KW/Jahr für MDS-STATUS
26.0	Kanal_1_Befehl[3].DAT_DB_number	INT	48	48	Nummer des DAT DB; Daten DB für MDS Daten
28.0	Kanal_1_Befehl[3].DAT_DB_address	INT	0	0	Zeiger auf das Anfangswort im DAT DB
30.0	Kanal_1_Befehl[4].command	BYTE	B#16#2	B#16#2	MDS Befehl:1=schreiben,2=lesen,3=init,4=slg-status,8=end,A=set-ant,B=mds-status
31.0	Kanal_1_Befehl[4].sub_command	BYTE	B#16#0	B#16#0	Bitmuster für INIT-Befehl; Modus für END,SET-ANT,MDS-STATUS,SLG-STATUS
32.0	Kanal_1_Befehl[4].length	INT	1	1	Anzahl der zu schreibenden/lesenden Daten in Bytes
34.0	Kanal_1_Befehl[4].address_MDS	WORD	W#16#0	W#16#0	Anfangsadresse auf MDS; Endadresse bei INIT;KW/Jahr für MDS-STATUS
36.0	Kanal_1_Befehl[4].DAT_DB_number	INT	48	48	Nummer des DAT DB; Daten DB für MDS Daten
38.0	Kanal_1_Befehl[4].DAT_DB_address	INT	0	0	Zeiger auf das Anfangswort im DAT DB
40.0	Kanal_1_Befehl[5].command	BYTE	B#16#2	B#16#2	MDS Befehl:1=schreiben,2=lesen,3=init,4=slg-status,8=end,A=set-ant,B=mds-status
41.0	Kanal_1_Befehl[5].sub_command	BYTE	B#16#0	B#16#0	Bitmuster für INIT-Befehl; Modus für END,SET-ANT,MDS-STATUS,SLG-STATUS
42.0	Kanal_1_Befehl[5].length	INT	1	1	Anzahl der zu schreibenden/lesenden Daten in Bytes
44.0	Kanal_1_Befehl[5].address_MDS	WORD	W#16#0	W#16#0	Anfangsadresse auf MDS; Endadresse bei INIT;KW/Jahr für MDS-STATUS
46.0	Kanal_1_Befehl[5].DAT_DB_number	INT	48	48	Nummer des DAT DB; Daten DB für MDS Daten
48.0	Kanal_1_Befehl[5].DAT_DB_address	INT	0	0	Zeiger auf das Anfangswort im DAT DB
50.0	Kanal_2_Befehl[1].command	BYTE	B#16#2	B#16#2	MDS Befehl:1=schreiben,2=lesen,3=init,4=slg-status,8=end,A=set-ant,B=mds-status
51.0	Kanal_2_Befehl[1].sub_command	BYTE	B#16#0	B#16#0	Bitmuster für INIT-Befehl; Modus für END,SET-ANT,MDS-STATUS,SLG-STATUS
52.0	Kanal_2_Befehl[1].length	INT	1	1	Anzahl der zu schreibenden/lesenden Daten in Bytes
54.0	Kanal_2_Befehl[1].address_MDS	WORD	W#16#0	W#16#0	Anfangsadresse auf MDS; Endadresse bei INIT;KW/Jahr für MDS-STATUS
56.0	Kanal_2_Befehl[1].DAT_DB_number	INT	48	48	Nummer des DAT DB; Daten DB für MDS Daten
58.0	Kanal_2_Befehl[1].DAT_DB_address	INT	0	0	Zeiger auf das Anfangswort im DAT DB
60.0	Kanal_2_Befehl[2].command	BYTE	B#16#2	B#16#2	MDS Befehl:1=schreiben,2=lesen,3=init,4=slg-status,8=end,A=set-ant,B=mds-status
61.0	Kanal_2_Befehl[2].sub_command	BYTE	B#16#0	B#16#0	Bitmuster für INIT-Befehl; Modus für END,SET-ANT,MDS-STATUS,SLG-STATUS

(Starting with byte 50, the commands for Reader2 or channel 2 start)

Save and close DB47

DB45 Reader Parameter Block

Each MOBY channel (reader) needs its own parameters. These are predefined in a data structure as UDT 10 (with English comments), UDT 11 (with German comments) and UDT 14 (with Spanish comments). This UDT has to be called for each MOBY channel in a data block. In UDT 11, different variables are defined:

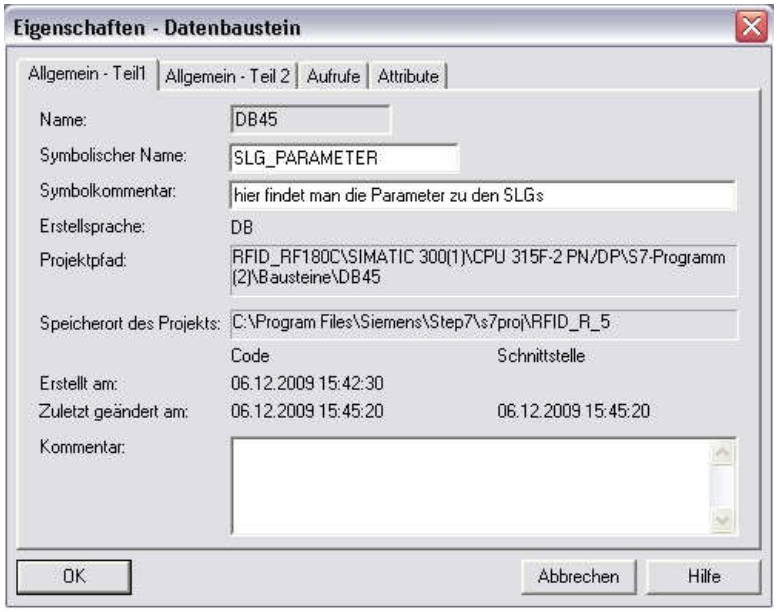
• INPUT parameters: The user has to enter these variables once during configuration (exception: command_DB_number/command_DB_address). It is not necessary to change or poll these parameters during the entire execution time.
Please note that when an INPUT parameter is changed, an init_run has to be performed before the new setting takes effect (refer to chapter "Programming Restarts and Warm Restarts").

• Control bits: With these Boolean variables, the user starts his commands.

• Displays: The displays show the user the progress of his commands. Errors are easy to analyze.

• Internal FB variables: These variables are of no significance to the user. They must not be changed by the application. Otherwise, malfunction and data corruptions would be the result.

Generating DB45



Open DB45 and insert UDT11 for each reader

In DB45, UDT11 is called for each reader

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	SLG_1	"MOBY Param_d"		Parameter des ersten SLG
+50.0	SLG_2	"MOBY Param_d"		Parameter des zweiten SLG
=100.0		END_STRUCT		

Entries in DB45

The base address for the RF180C is 256 (refer to hardware), here to be entered at address 0.0 and 50.0
The selection of Reader1 has to be entered under address 2.0.
The selection of Reader2 has to be entered under address 52.0.

DB47 (request data block) is referred to in DB45.

The requests of the first reader have to be entered in DB47 starting with DBB0, here at address 4.0 and 6.0
The requests of the second reader have to be entered in DB47 starting with DBB50, here at address 54.0 and 56.0.

No other values are changed.

For the entries in DB45, take note that the DBs continue to be chained correctly.

Data view of DB45 to byte 33:

Adresse	Name	Typ	Anfangswert	Aktualwert	Kommentar
0.0	SLG_1.ASM_address	INT	256	256	Input: Basisadresse des ASM (zyklisches Wort)
2.0	SLG_1.ASM_channel	INT	1	1	Input: Nummer des Kanals (1 ... 4)
4.0	SLG_1.command_DB_number	INT	47	47	Input: Befehlsdatenbaustein-Nummer
6.0	SLG_1.command_DB_address	INT	0	0	Input: Anfangsadresse der Daten im BEDB
8.0	SLG_1.MDS_control	BYTE	B#16#1	B#16#1	Input: Anwesenheitskontrolle und MDS-Steuerung (0, 1, 2)
9.0	SLG_1.ECC_mode	BOOL	FALSE	FALSE	Input: Betriebsart mit ECC
9.1	SLG_1.RESET_long	BOOL	TRUE	TRUE	Input: true: langes RESET-Telegramm, nur für MOBY-Mode 5, 6
10.0	SLG_1.MOBY_mode	BYTE	B#16#5	B#16#5	Input: MOBY-Betriebsart
11.0	SLG_1.scanning_time	BYTE	B#16#0	B#16#0	Input: Abtastzeit für MOBY I/U
12.0	SLG_1.option_1	BYTE	B#16#0	B#16#0	Input: RESET-Befehl Option 1
13.0	SLG_1.distance_limiting	BYTE	B#16#F	B#16#F	Input: Reichweitenbegrenzung
14.0	SLG_1.multitag	BYTE	B#16#1	B#16#1	Input: max. Anzahl MDS im Feld
15.0	SLG_1.field_ON_control	BYTE	B#16#0	B#16#0	Input: BERO-Betriebsart
16.0	SLG_1.field_ON_time	BYTE	B#16#0	B#16#0	Input: BERO-Zeit
17.0	SLG_1.reserved	BYTE	B#16#0	B#16#0	
18.0	SLG_1.ANZ_MDS_present	BOOL	FALSE	FALSE	Anwesenheit eines MDS
18.1	SLG_1.ANZ_cancel	BOOL	FALSE	FALSE	Cancel-Bit im PEW ist gesetzt
18.2	SLG_1.ANZ_ECC	BOOL	FALSE	FALSE	Fehlerkorrektur wurde durchgeführt
18.3	SLG_1.reserved0	BOOL	FALSE	FALSE	
18.4	SLG_1.LR_bat	BOOL	FALSE	FALSE	Batterie des MDS 507
18.5	SLG_1.battery_low	BOOL	FALSE	FALSE	Batterieüberwachung
18.6	SLG_1.error	BOOL	FALSE	FALSE	Fehler während der Befehlsbearbeitung
18.7	SLG_1.ready	BOOL	FALSE	FALSE	Der Befehl bzw. Befehlskette ist ausgeführt
19.0	SLG_1.cancel	BOOL	FALSE	FALSE	Set: Befehl bzw. Befehlskette abbrechen
19.1	SLG_1.command_start	BOOL	FALSE	FALSE	Set: Startsignal für Befehl bzw. Befehlskette
19.2	SLG_1.repeat_command	BOOL	FALSE	FALSE	Set: Letzten Befehl wiederholen
19.3	SLG_1.init_run	BOOL	TRUE	TRUE	Set: ASM rücksetzen und neu parametrieren
19.4	SLG_1.ASM_failure	BOOL	FALSE	FALSE	OB122 Set: ASM ausgefallen
19.5	SLG_1.FB45_active	BOOL	FALSE	FALSE	FC-aktiv
19.6	SLG_1.ANZ_next	BOOL	FALSE	FALSE	Befehl NEXT war letzter Befehl
19.7	SLG_1.ANZ_reset	BOOL	FALSE	FALSE	Befehl RESET war letzter Befehl
20.0	SLG_1.ASM_busy	BOOL	FALSE	FALSE	ASM bearbeitet einen Befehl
20.1	SLG_1.command_rep_active	BOOL	FALSE	FALSE	ASM führt Befehlswiederholung durch
21.0	SLG_1.number_MDS	BYTE	B#16#0	B#16#0	Anzahl der im Feld befindlichen MDS
22.0	SLG_1.error_MOBY	BYTE	B#16#0	B#16#0	Fehleranzeige vom Anschaltmodul
23.0	SLG_1.error_FB	BYTE	B#16#0	B#16#0	Fehleranzeige von FC
24.0	SLG_1.error_BUS	WORD	W#16#0	W#16#0	Fehleranzeige von PROFIBUS
26.0	SLG_1.version_MOBY	WORD	W#16#0	W#16#0	Firmwareversion MOBY
28.0	SLG_1.counter_customer	BYTE	B#16#0	B#16#0	Interner Schleifenzähler
29.0	SLG_1.counter_notused	BYTE	B#16#0	B#16#0	FB intern
30.0	SLG_1.reserved1[1]	DWORD	DW#16#0	DW#16#0	FB-interne Variablen. Diese dürfen vom Anwender nicht verändert werden.

Data view of DB45 starting with byte 34 (SLG = Reader)

34.0	SLG_1.reserved1[2]	DWORD	DW#16#0	DW#16#0	
38.0	SLG_1.reserved1[3]	DWORD	DW#16#0	DW#16#0	
42.0	SLG_1.reserved2	WORD	W#16#0	W#16#0	FB intern
44.0	SLG_1.initRUN_timeout	INT	1000	1000	FB intern
46.0	SLG_1.PEW_timout_ASM_fail	BYTE	B#16#5	B#16#5	FB intern
47.0	SLG_1.PEW_timeout	BYTE	B#16#32	B#16#32	FB intern
48.0	SLG_1.reserved3	BYTE	B#16#0	B#16#0	FB intern
49.0	SLG_1.Testbyte	BYTE	B#16#0	B#16#0	FB intern
50.0	SLG_2.ASM_address	INT	256	256	Input: Basisadresse des ASM (zyklisches Wort)
52.0	SLG_2.ASM_channel	INT	1	2	Input: Nummer des Kanals (1 ... 4)
54.0	SLG_2.command_DB_number	INT	47	47	Input: Befehlsdatenbaustein-Nummer
56.0	SLG_2.command_DB_address	INT	0	50	Input: Anfangsadresse der Daten im BEDB
58.0	SLG_2.MDS_control	BYTE	B#16#1	B#16#1	Input: Anwesenheitskontrolle und MDS-Steuerung (0, 1, 2)
59.0	SLG_2.ECC_mode	BOOL	FALSE	FALSE	Input: Betriebsart mit ECC
59.1	SLG_2.RESET_long	BOOL	TRUE	TRUE	Input: true: langes RESET-Telegramm, nur für MOBY-Mode 5, 6
60.0	SLG_2.MOBY_mode	BYTE	B#16#5	B#16#5	Input: MOBY-Betriebsart
61.0	SLG_2.scanning_time	BYTE	B#16#0	B#16#0	Input: Abtastzeit für MOBY I/U
62.0	SLG_2.option_1	BYTE	B#16#0	B#16#0	Input: RESET-Befehl Option 1
63.0	SLG_2.distance_limiting	BYTE	B#16#F	B#16#F	Input: Reichweitenbegrenzung
64.0	SLG_2.multitag	BYTE	B#16#1	B#16#1	Input: max. Anzahl MDS im Feld
65.0	SLG_2.field_ON_control	BYTE	B#16#0	B#16#0	Input: BERO-Betriebsart
66.0	SLG_2.field_ON_time	BYTE	B#16#0	B#16#0	Input: BERO-Zeit
67.0	SLG_2.reserved	BYTE	B#16#0	B#16#0	
68.0	SLG_2.ANZ_MDS_present	BOOL	FALSE	FALSE	Anwesenheit eines MDS
68.1	SLG_2.ANZ_cancel	BOOL	FALSE	FALSE	Cancel-Bit im PEW ist gesetzt
68.2	SLG_2.ANZ_ECC	BOOL	FALSE	FALSE	Fehlerkorrektur wurde durchgeführt
68.3	SLG_2.reserved0	BOOL	FALSE	FALSE	
68.4	SLG_2.LR_bat	BOOL	FALSE	FALSE	Batterie des MDS 507
68.5	SLG_2.battery_low	BOOL	FALSE	FALSE	Batterieüberwachung
68.6	SLG_2.error	BOOL	FALSE	FALSE	Fehler während der Befehlsbearbeitung
68.7	SLG_2.ready	BOOL	FALSE	FALSE	Der Befehl bzw. Befehlskette ist ausgeführt
69.0	SLG_2.cancel	BOOL	FALSE	FALSE	Set: Befehl bzw. Befehlskette abbrechen
69.1	SLG_2.command_start	BOOL	FALSE	FALSE	Set: Startsignal für Befehl bzw. Befehlskette
69.2	SLG_2.repeat_command	BOOL	FALSE	FALSE	Set: Letzten Befehl wiederholen
69.3	SLG_2.init_run	BOOL	TRUE	TRUE	Set: ASM rücksetzen und neu parametrieren
69.4	SLG_2.ASM_failure	BOOL	FALSE	FALSE	OB122 Set: ASM ausgefallen
69.5	SLG_2.FB45_active	BOOL	FALSE	FALSE	FC-aktiv
69.6	SLG_2.ANZ_next	BOOL	FALSE	FALSE	Befehl NEXT war letzter Befehl
69.7	SLG_2.ANZ_reset	BOOL	FALSE	FALSE	Befehl RESET war letzter Befehl
70.0	SLG_2.ASM_busy	BOOL	FALSE	FALSE	ASM bearbeitet einen Befehl
70.1	SLG_2.command_rep_active	BOOL	FALSE	FALSE	ASM führt Befehlswiederholung durch
71.0	SLG_2.number_MDS	BYTE	B#16#0	B#16#0	Anzahl der im Feld befindlichen MDS
72.0	SLG_2.error_MOBY	BYTE	B#16#0	B#16#0	Fehleranzeige vom Anschaltmodul
73.0	SLG_2.error_FB	BYTE	B#16#0	B#16#0	Fehleranzeige von FC
74.0	SLG_2.error_BUS	WORD	W#16#0	W#16#0	Fehleranzeige von PROFIBUS
76.0	SLG_2.version_MOBY	WORD	W#16#0	W#16#0	Firmwareversion MOBY
78.0	SLG_2.counter_customer	BYTE	B#16#0	B#16#0	Interner Schleifenzähler
79.0	SLG_2.counter_notused	BYTE	B#16#0	B#16#0	FB intern
80.0	SLG_2.reserved1[1]	DWORD	DW#16#0	DW#16#0	FB-interne Variablen. Diese dürfen vom Anwender nicht verändert werden.
84.0	SLG_2.reserved1[2]	DWORD	DW#16#0	DW#16#0	
88.0	SLG_2.reserved1[3]	DWORD	DW#16#0	DW#16#0	
92.0	SLG_2.reserved2	WORD	W#16#0	W#16#0	FB intern
94.0	SLG_2.initRUN_timeout	INT	1000	1000	FB intern
96.0	SLG_2.PEW_timout_ASM_fail	BYTE	B#16#5	B#16#5	FB intern
97.0	SLG_2.PEW_timeout	BYTE	B#16#32	B#16#32	FB intern
98.0	SLG_2.reserved3	BYTE	B#16#0	B#16#0	FB intern
99.0	SLG_2.Testbyte	BYTE	B#16#0	B#16#0	FB intern

Save and close DB45

Note
The inputs in the data block are entered as actual value.
Take note that when the data block is initialized or a general reset is performed on the CPU, the actual values are overwritten with the start values of the data blocks.

Additional information about setting up the data structure with data blocks and about the individual UDTs is provided in the function manual "RFID Systems FB45“.

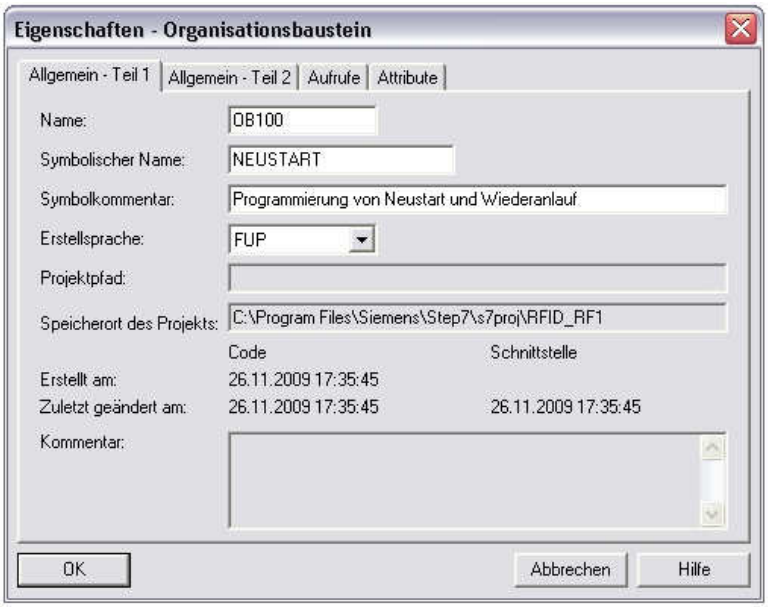
5.6 Programming a Restart or Warm Restart

A restart of the MOBYIM (interface module) is carried out by setting the variable "**init_run**". With "**init_run**", the IM and FB45 are re-parameterized and synchronized.

"**init_run**" is necessary after the following:

- Switching on the SIMATIC (OB 100)
- Switching on the power supply for the IM
- PROFINET communication is interrupted
- An error indication by the variable "error_BUS"
- A transponder type change; for example from RF300 to ISO or vice versa
- Changing an INPUT parameter in DB45 (Reader_Parameter)

Generating OB100



Open OB100 and enter the program

```
OB100 : Neustart und Wiederanlauf
Kommentar:

Netzwerk 1: Setzen der Variablen "init_run"
Kommentar:

      SET
      S     "SLG_PARAMETER".SLG_1.init_run    DB45.DBX19.3       -- Set: ASM rücksetzen und neu parametrieren
      S     "SLG_PARAMETER".SLG_2.init_run    DB45.DBX69.3       -- Set: ASM rücksetzen und neu parametrieren
```

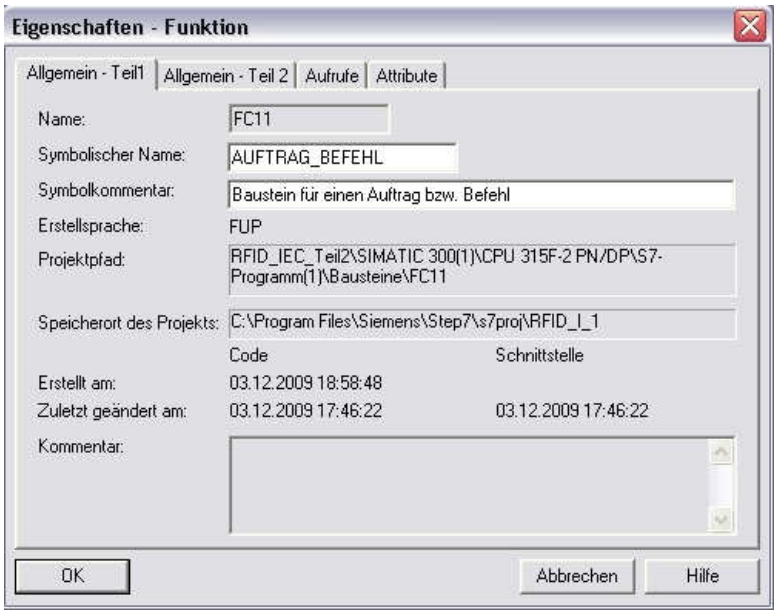
Save and close OB100

5.7 FC11 Function for a Command or Request

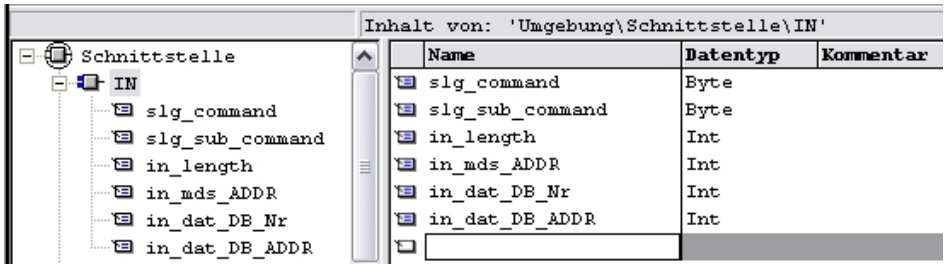
Before starting a MOBY command with **"command_start"**, we have to define it. For a simple definition of a command, UDT 21 (German comments) is provided. UDT21 is embedded in DB47 multiple times.
So that not all inputs have to be carried out in DB47, here a block for one command or request is generated.
It is then possible to call this block in the control program multiple times; for example for a command string.

Generating Function FC11

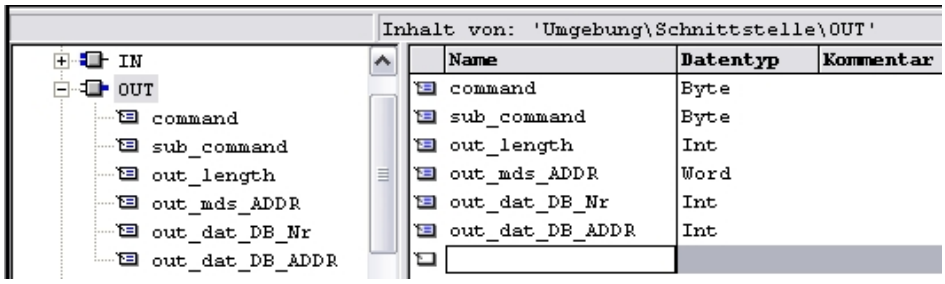
Set up a new FC11 function.



Set up IN variables



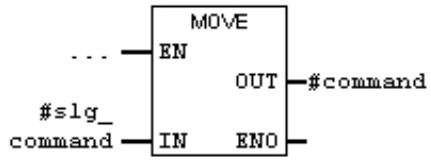
Set up OUT variables



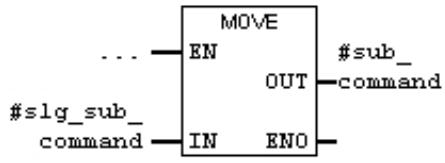
Enter Networks 1 to 6

```
FC11 : Request or command
```

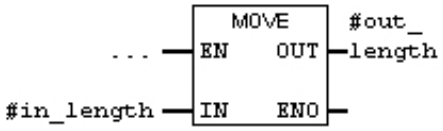
Network 1 Command



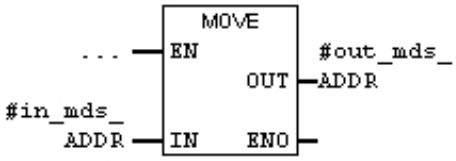
Network 2 Sub_Command



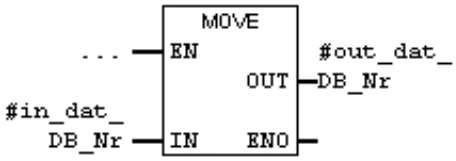
Network 3 Data Length



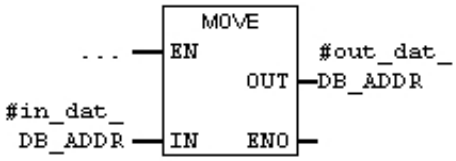
Network 4 MDS start address



Network 5 DB number for data storage destination or source



Network 6 Destination or target address of the DB of data storage



Save and close FC11

5.8 Basics of Entries at Command Block FC11

The values for DB47 are specified by means of the input variables of FC11.
One UDT21 is used for each request or command.

Data View of the UDT21

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	command	BYTE	B#16#2	MDS Befehl:1=schreiben,2=lesen,3=init,4=slg-status,8=end,A=set-ant,B=mds-status
+1.0	sub_command	BYTE	B#16#0	Bitmuster für INIT-Befehl; Modus für END,SET-ANT,MDS-STATUS,SLG-STATUS
+2.0	length	INT	1	Anzahl der zu schreibenden/lesenden Daten in Bytes
+4.0	address_MDS	WORD	W#16#0	Anfangsadresse auf MDS; Endadresse bei INIT;KW/Jahr für MDS-STATUS
+6.0	DAT_DB_number	INT	48	Nummer des DAT DB; Daten DB für MDS Daten
+8.0	DAT_DB_address	INT	0	Zeiger auf das Anfangswort im DAT DB
=10.0		END_ST		

Command Overview

Command (hex)		**Command**
Normal	Chained*	
01	41	Write data to MDS
02	42	Read data from MDS
03	43	Initialize MDS
04	44	Reader status
06	--	NEXT
08	48	END; end communication with the MDS
0A	4A	Aerial on/off
0B	4B	MDS status
*) Not all readers or IM modules support chained commands. Note the information in the MOBY manuals for configuration, installation and service.		

Writing data to the transponder

Command (hex)	Sub_command	length [dec]	address_MDS [hex]	DAT_DB [dec]	Comment
01	-	1 to 32767 * Length of the MSD data to be written	0000 to FFFF Starting with this start address, the data is written to the MDS	Pointer to the user data that is to be written to the MDS	

Reading data from the transponder

Command (hex)	Sub_command	length [dec]	address_MDS [hex]	DAT_DB [dec]	Comment
02	-	1 to 32767 * Length of the MSD data to be read	0000 to FFFF Starting with this start address, the data is read from the MDS	Pointer to the user data. Here, FB45 stores the MDS data that	

				was read.	

Initialize Transponder

Command (hex)	Sub_command	length [dec]	address_MDS [hex]	DAT_DB [dec]	Comment
03	00 to FF hex value that is written to the MDS	-	Memory size of the MDS to be initialized	-	

Read out Reader status

Command (hex)	Sub_command	length [dec]	address_MDS [hex]	DAT_DB [dec]	Comment
04	01 = status according to UDT 110[1] 02 = status according to UDT 120[1] (last commands) 03 = status according to UDT 130[1] (error indications) 04 = status according to UDT 140[1] (MDS in the field) 05 = status according to UDT 150 (communication quality) 06 = status according to UDT 280 (diagnosis data)			Pointer to the result. The result is represented with the corresponding UDT (refer to sub_command)	MOBY U/D or RF300 MOBY U MOBY U MOBY U RF300

(In our example, we are using the German language UDT111 and UDT281 for RF300)

Command NEXT

Command (hex)	Sub_command	length [dec]	address_MDS [hex]	DAT_DB [dec]	Comment
06	-	-	-	-	NEXT: processing this MDS is completed

Command END

Command (hex)	Sub_command	length [dec]	address_MDS [hex]	DAT_DB [dec]	Comment
08	00 = Processing with the MDS is ended 01 = Processing	-	-	-	ANZ_MDS_present is reset ANZ_MDS_present

	pause with the MDS[1]				remains set

Switching the reader aerial on/off

Command (hex)	Sub_command	length [dec]	address_MDS [hex]	DAT_DB [dec]	Comment
0A	01 = switch on aerial 02 = standby: Switch aerial off 09 = adjust aerial to the environment (FFT)	-	-	-	The command Aerial on/off can not be started with command repetition (refer to chapter "Command Repetition) Only Reader 80 (MOBY F)

Transponder Status

Command (hex)	Sub_command	length [dec]	address_MDS [hex]	DAT_DB [dec]	Comment
0B	00 = status and diagnosis	-	Today' s date (week/year) to calculate the life of the battery (for example, 1401 hex = 20th week of year 2001)	Pointer to result. The result is represented with UDT 100	only MOBY U (refer to chapter "The UDTs of FB45")
	01 = type and write protection status	-	-	Pointer to result. The result is represented with UDT 260	RF 300 (refer to chapter "The UDTs of FB45")
	02 = diagnosis data	-	-	Pointer to result. The result is represented with UDT 270	RF 300 (refer to chapter "The UDTs of FB45")

(In our example, we are using the German language UDT261 and UDT271 for RF300)

Notes

_____

_____

_____

_____

5.9 Command String

The user configures the command string by storing a corresponding number of UDT 21 one after the other in a DB. All commands that are strung together have to be of the "Command" type "4x". The last command of a string has to be of the type 0x. With it, the FB 45 recognizes the end of a command string.

Example of a command string in DB47

Four data records are to be processed by an MDS.
The command structure is stored in the request DB, as shown below.
The destination and source data of the MDS are stored consecutively in DB48.

Read MDS address 0000 hex length 600
Read MDS address 1000 hex length 100
Read MDS address 1200 hex length 1
Write MDS address 1200 hex length 1

Name	Initial Value	Comment
Teilbefehl [1].command	42	Read command, another command follows
Teilbefehl [1].pattern	00	
Teilbefehl [1].length	600	
Teilbefehl [1].adress_MDS	0000	
Teilbefehl [1].DAT_DB_number	48	
Teilbefehl [1].DAT_DB_address	0	
Teilbefehl [2].command	42	Read command, another command follows
Teilbefehl [2].pattern	00	
Teilbefehl [2].length	100	
Teilbefehl [2].adress_MDS	1000	
Teilbefehl [2].DAT_DB_number	48	
Teilbefehl [2].DAT_DB_address	600	
Teilbefehl [3].command	42	Read command, another command follows
Teilbefehl [3].pattern	00	
Teilbefehl [3].length	1	
Teilbefehl [3].adress_MDS	1200	
Teilbefehl [3].DAT_DB_number	48	
Teilbefehl [3].DAT_DB_address	700	
Teilbefehl [4].command	01	Write command, last command in string
Teilbefehl [4].pattern	00	
Teilbefehl [4].length	1	
Teilbefehl [4].adress_MDS	1200	
Teilbefehl [4].DAT_DB_number	48	
Teilbefehl [4].DAT_DB_address	701	

Notes

_____

_____

_____

_____

5.10 Basics of FB45 and DB45

FB45 accesses DB45. In DB45, a UDT11 is embedded for each reader.

Data view of the UDT11

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	ASM_address	INT	0	Input: Basisadresse des ASM (zyklisches Wort)
+2.0	ASM channel	INT	1	Input: Nummer des Kanals (1 ... 4)
+4.0	command_DB_number	INT	47	Input: Befehlsdatenbaustein-Nummer
+6.0	command_DB_address	INT *	0	Input: Anfangsadresse der Daten im BEDB
+8.0	MDS_control	BYTE	B#16#1	Input: Anwesenheitskontrolle und MDS(0, 1, 2)
+9.0	ECC mode	BOOL	FALSE	Input: Betriebsart mit ECC
+9.1	RESET_long	BOOL	FALSE	Input: true: langes RESET-Telegramm, nur für MOBY-Mode 5
+10.0	MOBY mode	BYTE	B#16#1	Input: MOBY-Betriebsart
+11.0	scanning_time	BYTE	B#16#0	Input: Abtastzeit für MOBY I/U
+12.0	option_1	BYTE	B#16#0	Input: RESET-Befehl Option 1
+13.0	distance_limiting	BYTE	B#16#F	Input: Reichweiten-/Leistungseinstellung
+14.0	multitag	BYTE	B#16#1	Input: max. Anzahl MDS im Feld
+15.0	field ON control	BYTE	B#16#0	Input: BERO
+16.0	field_ON_time	BYTE	B#16#0	Input: MOBY U: BERO-Zeit MOBY D: MDS-Typ
+17.0	reserved0	BYTE	B#16#0	
+18.0	ANZ MDS present	BOOL	FALSE	Anwesenheit eines MDSs
+18.1	ANZ cancel	BOOL	FALSE	Cancel-Bit im PEW ist gesetzt
+18.2	ANZ_ECC	BOOL	FALSE	Fehlerkorrektur wurde durchgeführt
+18.3	reserved	BOOL	FALSE	
+18.4	LR bat	BOOL	FALSE	Batterie des MDS 507
+18.5	battery_low	BOOL	FALSE	Batterieüberwachung
+18.6	error	BOOL	FALSE	Fehler während der Befehlsbearbeitung
+18.7	ready	BOOL	FALSE	Der Befehl bzw. Befehlskette ist ausgeführt
+19.0	cancel	BOOL	FALSE	Set: Befehl bzw. Befehlskette abbrechen
+19.1	command_start	BOOL	FALSE	Set: Startsignal für Befehl bzw. Befehlskette
+19.2	repeat_command	BOOL	FALSE	Set: letzten Befehl wiederholen
+19.3	init_run	BOOL	TRUE	Set: ASM rücksetzen und neu parametrieren
+19.4	ASM failure	BOOL	FALSE	OB122 Set: ASM ausgefallen
+19.5	FB45 active	BOOL	FALSE	FB-aktiv
+19.6	ANZ next	BOOL	FALSE	Befehl NEXT war letzter Befehl
+19.7	ANZ reset	BOOL	FALSE	Befehl RESET war letzter Befehl
+20.0	ASM busy	BOOL	FALSE	ASM bearbeitet einen Befehl
+20.1	command_rep_active	BOOL	FALSE	ASM führt Befehlswiederholung durch
+21.0	number_MDS	BYTE	B#16#0	Anzahl der im Feld befindlichen MDS
+22.0	error MOBY	BYTE	B#16#0	Fehleranzeige vom Anschaltmodul
+23.0	error FB	BYTE	B#16#0	Fehleranzeige von FB
+24.0	error_BUS	WORD	W#16#0	Fehleranzeige von PROFIBUS
+26.0	version MOBY	WORD	W#16#0	Firmwareversion MOBY
+28.0	reserved2	ARRAY[1…4]	DW#16#0	FB-interne Variablen. Diese dürfen vom Anwender nicht verändert werden.
*4.0		DWORD		
+44.0	initRUN_timeout	INT	1000	FB-interne Variablen. Diese dürfen vom Anwender nicht verändert werden.
+46.0	PEW_timeout_ASM_fail	BYTE	B#16#5	FB-interne Variablen. Diese dürfen vom Anwender nicht verändert werden.
+47.0	PEW_timeout	BYTE	B#16#32	FB-interne Variablen. Diese dürfen vom Anwender nicht verändert werden.
+48.0	reserved3	BYTE	B#16#0	FB-interne Variablen. Diese dürfen vom Anwender nicht verändert werden.
+49.0	Testbyte	BYTE	B#16#0	FB-interne Variablen. Diese dürfen vom Anwender nicht verändert werden.
=50.0		END_STRUCT		

Input Parameters of the UDT11

The user has to enter these variables during configuration (exception: command_DB_number/ command_DB_address).
It is not necessary to change or poll these parameters during the entire execution time.

Please note that before the new setting takes effect, "init_run" has to be performed after an INPUT parameter is changed (refer to chapter "Programming Restarts and Warm Restarts").

<table>
<tr><th>Variable</th><th colspan="3">Description</th></tr>
<tr><td>IM Address</td><td colspan="3">IM's logic base address. This address has to match the IM "start address" in HWConfig of the SIMATIC Manager. Please note that this address has nothing to do with the PROFIBUS address that is set at the IM or the ET200M.</td></tr>
<tr><td rowspan="4">IM Channel</td><td colspan="3">Number of the MOBY channel that is to be used for processing:</td></tr>
<tr><td colspan="2">ASM Type</td><td>Value Range</td></tr>
<tr><td colspan="2">IM475, 452, 456,RF170C
IM 454, 754, 854
ASM 473, 850</td><td>1,2,
1,2,3,4
1</td></tr>
<tr><td rowspan="2">command_DB_ number</td><td colspan="2" rowspan="2">Number of data block where the MDS command is specified</td><td rowspan="3">These INPUT parameters can be changed if ready = 1. After a change of these parameters, init_run does not have to be performed</td></tr>
<tr><td rowspan="2">command_DB_ address</td></tr>
<tr><td colspan="2">Address within the "command_DB". The next MDS command starts on this address. "command_DB_number" and "command_DB_address generate a data pointer to the next command (refer to chapter "Configuration Scheme").</td></tr>
<tr><td></td><td colspan="3">Please note:
The input parameters command_DB_number and command_DB_address are to be changed only if ready = 1. After changing these parameters, init_run does not have to be performed.</td></tr>
<tr><td rowspan="5">MDS Control</td><td colspan="3">MDS_control switches the attendance check or the MDS control on or off on the IM (refer to chapter "Attendance Check and MDS Control").</td></tr>
<tr><td>Value</td><td>MDS Control</td><td>IM Type</td></tr>
<tr><td>0</td><td>Attendance check is switched off. The variable ANZ_MSD_present does not indicate a valid value</td><td>All</td></tr>
<tr><td>1</td><td>Attendance check is switched on. The MDS control is switched off. The variable ANZ_MDS_present indicates an MDS in the transfer window of a reader</td><td>All</td></tr>
<tr><td>2</td><td>Attendance check is switched on. The MDS control is switched on and happens by means of the attendance check of the MDS. The NEXT command has to be sent to the IM after each MDS processing.</td><td>454</td></tr>
<tr><td>ECC_mode</td><td colspan="3">Switches on the ECC mode on (true) or off (false). Take note that the ECC mode is permitted only for MOBYI.</td></tr>
<tr><td>RESET_long</td><td colspan="3">With the command init_run, all INPUT parameters are transmitted to the IM. For MOBY U/D or RF300 operation, this bit has to be set to true (MOBU_mode = 5)</td></tr>
</table>

Notes

Parameter "**ASM_address**" value = 256 (refer to hardware configuration) _____

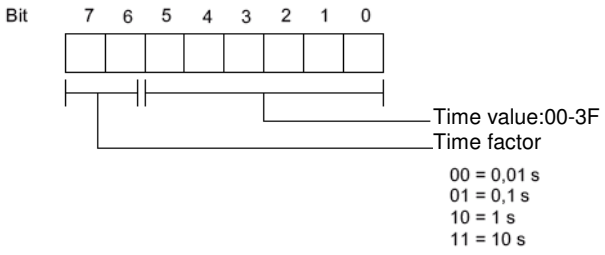
Parameter "**ASM_channel**" value = 1 for Reader1 _____

Parameter "**ASM_channel**" value = 2 for Reader2 _____

Parameter "**command_DB_number**" value = 47 for both readers _____

Parameter "**command_DB_address**" value = 0 for Reader1 _____

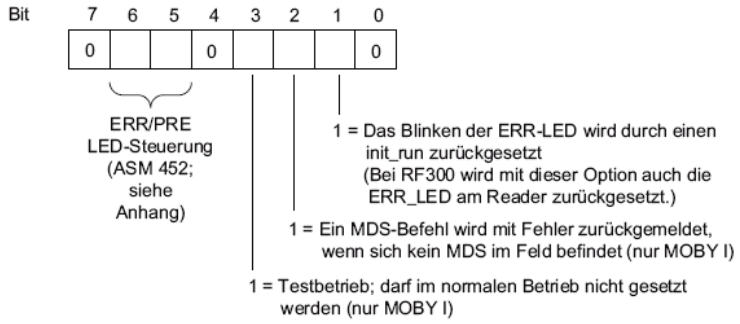
Parameter “**command_DB_address**“ Value = 50 for Reader2 _____

<table>
<tr><th>Variable</th><th colspan="3">Description</th></tr>
<tr><td>MOBY_mode</td><td colspan="3">Setting the MOBY Operating</td></tr>
<tr><td></td><td>Value</td><td>Operating Mode</td><td>IM Type</td></tr>
<tr><td></td><td>0</td><td>Default</td><td>-; reserved for setting with switch or GSD parameterization: different interfaces without switch under- stand under MOBY mode = 0 the MOBY I mode.</td></tr>
<tr><td></td><td>1</td><td>MOBY I or MOBY E (without MDS 507)</td><td>All</td></tr>
<tr><td></td><td>4</td><td>MOBY I with MDS 507</td><td>452; 454; 456</td></tr>
<tr><td></td><td>5</td><td>MOBY U/D or RF300 – without multi tag handling</td><td>475; 473; 452; 456; RF170C</td></tr>
<tr><td></td><td>6</td><td>Res. for MOBY U – with multi tag handling</td><td>–</td></tr>
<tr><td></td><td>7</td><td>Reserved for MOBY D or RF300 - with multi tag handling (FB55)</td><td>–</td></tr>
<tr><td></td><td>8</td><td>MOBY I-Dialog</td><td>454; 452; 456</td></tr>
<tr><td></td><td>9</td><td>MOBY V</td><td>452</td></tr>
<tr><td></td><td>A</td><td>MOBY F with MDS F1xx</td><td>475; 473: 452; 854; 850</td></tr>
<tr><td></td><td>B</td><td>MOBY F with MDS F4xx</td><td>475; 473: 452; 854; 850</td></tr>
<tr><td></td><td>C</td><td>MOBY F (reserved for MDS</td><td>–</td></tr>
<tr><td></td><td colspan="3">To be noted:
MOBY mode must only be changed after an IM is switched on</td></tr>
<tr><td>scanning_time</td><td colspan="3">MOBY I/V:
Scannung_time is the scanning time for the MDS 507 of MOBY I and MOBY V. For all other MDS types, the value 00 can be used.here. The figure below shows the setting for the scanning time (refer also to configguration manual for Reader 44/MDS 507):
 

Example: The result of a scanning time of 1 second = 81 hex for the parameter ABTA (scanning time)</td></tr>
</table>

Notes

<table>
<tr><th>Variable</th><th>Beschreibung</th></tr>
<tr><td>scanning_time</td><td>MOBY U:
Scanning_time beschreibt die Standby-Zeit für den MDS. Erhält der MDS vor Ablauf der scanning_time einen weiteren Befehl, so kann dieser sofort bearbeitet werden. Erhält der MDS einen Befehl nach Ablauf der scanning_time, so wird die Befehlsbearbeitung um die sleep_time des MDS verzögert.
Eine scanning_time sollte nur eingestellt werden, wenn
• der MDS mit mehreren Befehlen bearbeitet wird und
• der Bearbeitungsvorgang in einer minimalen Zeit abgeschlossen sein muss.

00 hex = keine Standby-Zeit (Default)
01 hex = 7 ms Standby-Zeit
02 hex = 14 ms Standby-Zeit
:
C8 hex = 1400 ms Standby-Zeit
Beachte:
Die scanning_time beeinflusst die Batterielebensdauer. Je größer die eingestellte scanning_time, desto kürzer die Batterielebensdauer. Genauere Berechnungen finden Sie im MOBY U-Handbuch für Projektierung, Montage und Service.
MOBY D bzw. RF300: 00 hex (reserviert)</td></tr>
<tr><td>option_1</td><td>Dieses Byte ist bitweise kodiert. Es hat standardmäßig den Wert B#16#0. Es können damit besondere Steuerungen im ASM vorgenommen werden.
 
</td></tr>
<tr><td>distance_limiting</td><td>MOBY U: Reichweitenbegrenzung
<table>
<tr><th>normale Sendeleistung</th><th colspan="2">reduzierte Sendeleistung</th></tr>
<tr><td>05 hex = 0,5 m
0A hex = 1,0 m
0F hex = 1,5 m
14 hex = 2,0 m
19 hex = 2,5 m
1E hex = 3,0 m
23 hex = 3,5 m</td><td>85 hex
8A hex
8F hex
91 hex
99 hex
9E hex
A3 hex</td><td>Die reduzierte Sendeleistung ist einzustellen, wenn mehrere SLG nahe beieinander positioniert sind oder wenn Datenspeicher, die sich in der Nähe eines SLG aufhalten, später oder nicht mehr erkannt werden sollen.
Nachteil: Die Feldkeule wird kleiner und damit steht für die Kommunikation weniger Zeit zur Verfügung bzw. es muss genauer positioniert werden.</td></tr>
</table>
MOBY D: HF-Leistung von 0,5 W bis 10 W in 0,25 W-Schritten
02 hex = 0,5 W
:
10 hex = 4 W (default)
:
28 hex = 10 W
RF300: 00 hex (reserviert)</td></tr>
</table>

Notes

Parameter "**option_1**“ value = 2 to reset the red flashing of the error LED at the reader with “Init_Run” _____

_____

_____

Variable	Description		
multitag	MOBY U/D or RF300; maximum number of MDS processed in parallel in the field. Permissible values: 1		
field_ON_control	MOBY U: BERO operating mode; aerial field is switched on/off automatically. The command "Aerial ON/OFF" is overlaid by the BERO operating mode.		
	00 hex	=	Without BEROs; no reader synchronization
	01 hex	=	One or two BEROs; The BEROs are logically ored. While a BERO is operated, the field is switched on
	02 hex	=	One or two BEROs. The 1st BERO switches the field on and the 2nd BERO switches the field off. If there are 2 BEROs *and* one field_ON_time is parameterized, the field is switched off automatically if the 2nd BERO does not switch within the BERO time. If no field_ON_time is parameterized, the field remains switched on until the 2nd BERO is operated.
	03 hex	=	Activate reader synchronization by means of cable connection (refer to manual for Configuration, Installation and Service for MOBY U)
	MOBY D or RF300: 00 hex (reserved)		
field_ON_time	**MOBY U**: time for BERO operating mode (field_ON_control = 02)		
	00 hex	=	Time monitoring is switched off. For field switch-off, the 2nd BERO is needed.
	01 hex .. FF hex	=	1...255s switch-on time for the Reader field
	MOBY D: MDS type		
	00 hex	=	I-code 1 (for example, MDS D139)
	01 hex	=	ISO MDS
	RF300: 00 hex (reserved)		
reserved	Reserved		

Notes

Parameter "**field_ON_time**" value = 0 for MDS type RF300 _____

Parameter "**field_ON_time**" value = 1 for MDS type ISO_____

_____

_____

_____

_____

_____

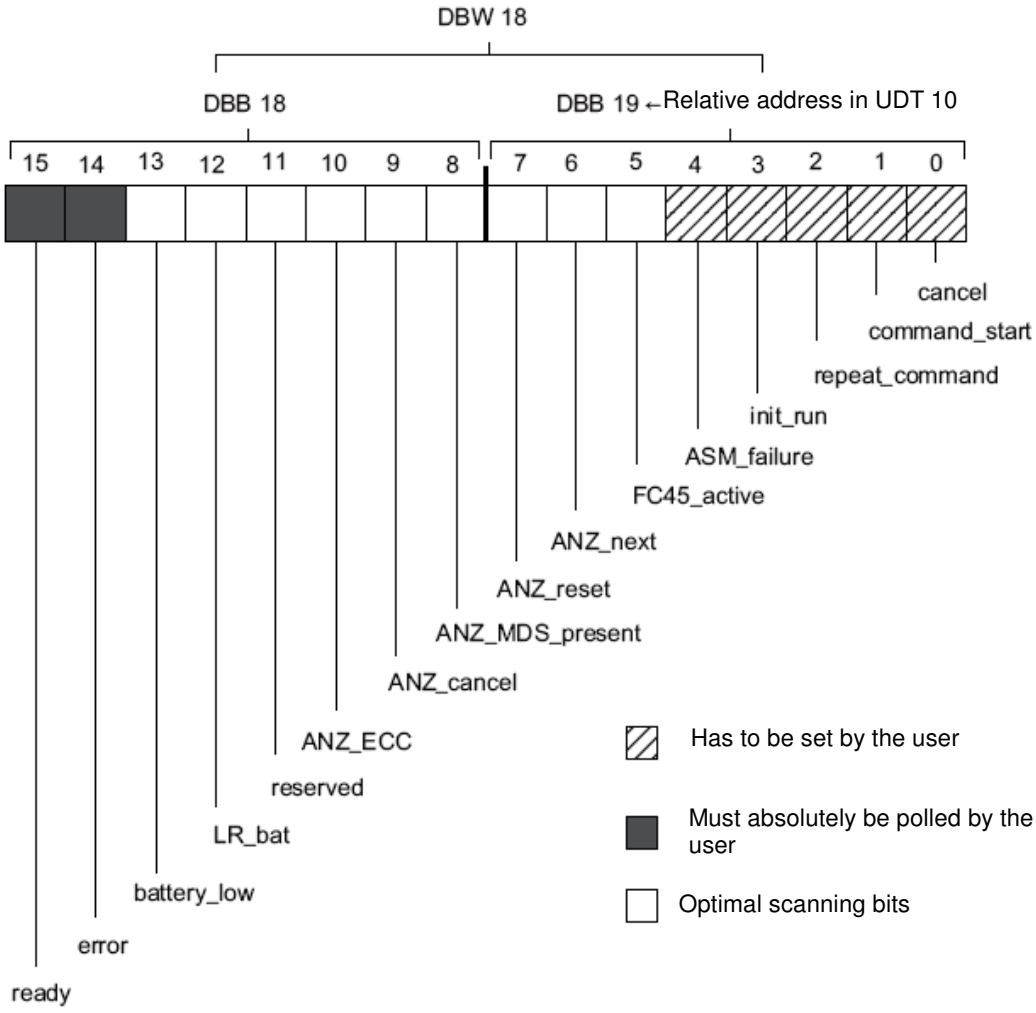
_____

_____

Command and Status Word "BEST"

The control bits of FB45 are defined in the command and status word.
The command and status word with the variables is generated with UDT 11.
The variables and the associated relative addresses in UDT 11 are shown in the figure below.

Control bits:
The user starts his commands with these Boolean variables.



Control bits from Bit0 to Bit7

Variable	Description
cancel	*True* =interruption of a current command or a command string. FB45 then sets the variable ready. **MOBY U/D or RF300**: the variable 'cancel' is not available. A command has to be canceled with the variable init_run.
command_start	*True* = start of a command or a command string
repeat_ command	*True* = command repetition: The command or command chain stored last in the IM is reprocessed with the next MDS. However, command processing for the MDS is started only after the MDS that was processed has exited the transmission window (ANZ_MDS_present= 0) and a new MDS has entered the transmission window of the reader (ANZ_MDS_present: 0 → 1). *False* = no command repetition, or command repetition is stopped after the command that was started with the repeat command is processed. Please note that **the user** has to reset this bit in order to stop command repetition. The result of the command repetition is fetched by the use setting command_start. Repeat_command is not reset automatically by FB45 after the command is processed. The commands init_run and cancel reset the variable repeat_command. This also interrupts a command repetition in the IM. repeat_command can be set again by the application with the next command_start. Handling of command repetition is described in the chapter "Command Repetition".
Init_run	*True* = IM restart. In this case, FB45 is also reset and the IM re-parameterized. All data and commands in the IM are lost. This bit has to be set in the restart OB (OB100) for each MOBY channel or IM. After a MOBY-IM fault, the error error_MOBY=0F is indicated to the user. The user then has to perform an init_run. Please note: • When loading a parameter data block from the programming device to the SIMATIC, bit init_run is pre-assigned TRUE. The result is the automatic execution of an IM restart. • The time to execute init_run is normally in the millisecond range. If there is an error, this time may extend to 15s.
IM_failure	*True* = the IM failed. The user sets this bit in OB122 (refer to chapter "Programming Module Failure"). FB45 then signals an error to the user (error_FB = 09) and interrupts the current command. If the user does not program OB122, the PLC enters the STOP mode if the IM fails.
FB45_active	FB45 is just processing a command. This variable is set when the command is started (command_start=True) and remains active until • FB45 has received the last acknowledgement from the IM • The init_run bit was set • The cancel bit was set • The IM signaled an error
ANZ_next	This bit is set if the command executed last was a NEXT command
ANZ_reset	This bit indicates that the command executed last was a RESET. The user started the RESET command with "init_run".

Notes

_____

_____

_____

_____

Control bits from Bit8 to Bit15

Variable	Description
ANZ_MDS_present	Indicates the presence of an MDS in the transmission window of the Read/Write MDS. ANZ_MDS_present is indicated only if the user set the INPUT parameter MDS_control. Please note that when init_run is executed, the ANZ_MDS_present indication briefly disappears even if an MDS stays permanently in the transmission window.
ANZ_cancel	The command executed last was a command interruption (cancel).The bit is set if the IM displays a cancel acknowledgement by means of the cyclical word (refer to chapter "Cyclical Control Word between Master and MOBY IM". Reset is automatic by starting a new command.
ANZ_ECC	**Only MOBY I**: If the ECC driver is switched on (INPUT parameter "ECC mode" = TRUE), the bit indicates that the data read by the MDS was corrected. ANZ_ECC is *not* an error indication since the data is OK. ANZ_ECC indicates that in the near future the MDS memory that was just processed may completely fail.
reserved	Presently not assigned
LR_bat	This bit is of significance only if at **MOBY I** processing takes place with the MDS507. It indicates an empty MDS507 dialog battery. For all other MDSs, this bit can take on any state.
battery_low	**Only for MOBY I/V with RAM MDS** The back-up battery of the RAM MDS is below the threshold. Although it is still possible to process several months at room temperature with the remaining capacity, it is recommended to immediately change the battery or to replace the MDS if the battery can't be changed.
error	FB45 sets this bit if a command is concluded faulty. The error bit is the sum error bit for all occurring errors. The exact cause for the error is located in the variable error_MOBY, error_FB or error_BUS (refer also to chapter "Additional Indications" or chapter "Error Indications and Error Search"). Restarting the command resets the error bit.
ready	Ready indication: after ready = TRUE was indicated, the error bit = FALSE has to be polled. This ensures that the command was processed without error. Please note: The ready bit does not have to be set for starting init_run or cancel.

Notes

Additional indications

The displays indicate to the user the progress of a command. Error analyses are easy to perform.

Variable	Description
IM_busy	This bit is set when the IM processes a command. Normally, "IM_busy" is inverted to "ready". IM_busy is indicated by the IM by means of the cyclical word (refer to chapter "Cyclical Control Word between Master and MOBY IM" under "IM_busy"). If processing takes place with the automatic command start repeat_command, this bit indicates the processing of a new MDS with the command that is to be executed.
command_rep_active	The IM is just executing a command repetition. The bit is set as a response to the control variable repeat_command. After init_run, FB45 first resets the command_rep_active; it is set again delayed since FB45 first transmits the MOBY commands to the IM.
number_MDS	**MOBY U/D or RF300**: The number of MDSs is indicated that are presently located in the transmission window. If more than 15 MDSs are in the field, the display number_MDS stops at 0F hex.
error_MOBY	The IM signaled this error. As a rule, this error is indicated also on the ERR LED on the IM channel display (refer to chapter "Error Indications and Error Search").
error_FB	Error indication from FB45 (refer to chapter "Error Indications and Error Search").
error_BUS	The transmission path between FB45 and the IM signals an error. As a rule, this is a PROFIBUS error (refer to chapter "Error Indications and Error Search"). This error is indicated by the system functions SFC 58/59.
version_MOBY	Display of the firmware version of the MOBY IM. The value entered here is updated after every IM power-up . It is ASCII encoded. Example: DBB26: 31 hex ("1"); DBB27: 30 hex ("0") → Version 1.0

Notes

_____

_____

_____

_____

_____

_____

_____

_____

All other UDT 11 variables are for internal FB use only.
The user must not change them at all.

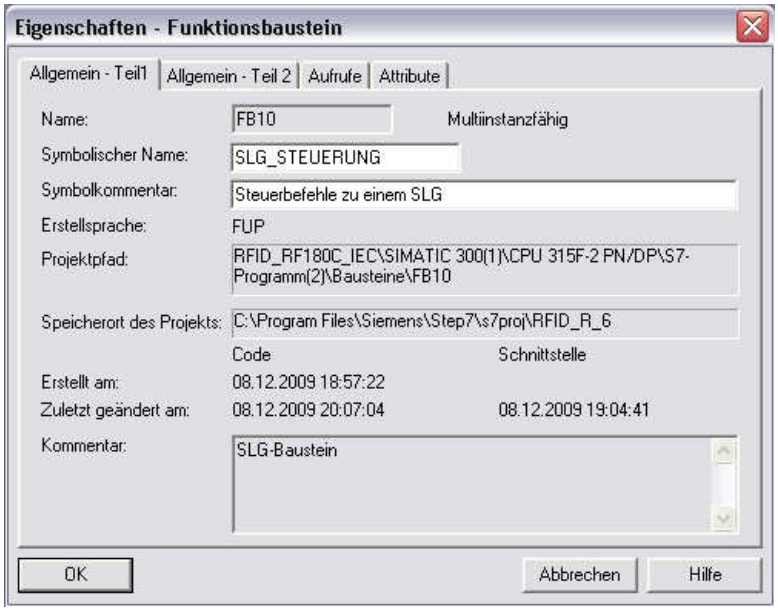
5.11 FB10 Reader_Control Program

Now, we are going to program the following in FB10: the control program for the control commands of a reader, and the call of FB45.

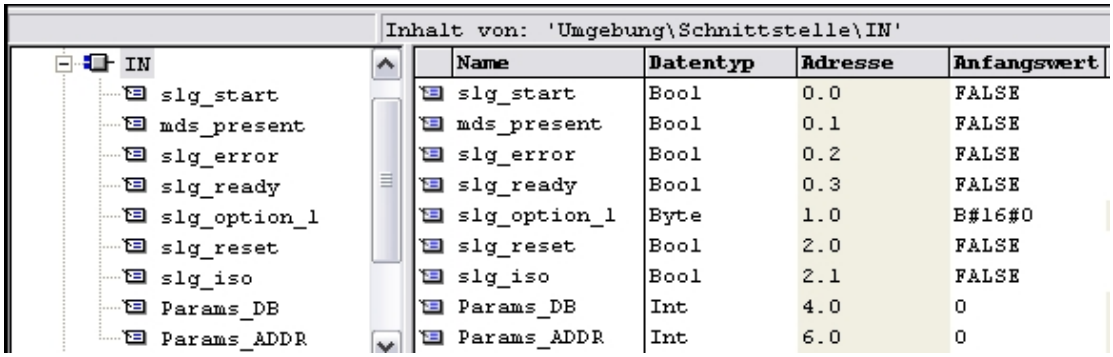
In addition, the attendance time of the transponder is to be recorded.

Generating FB10

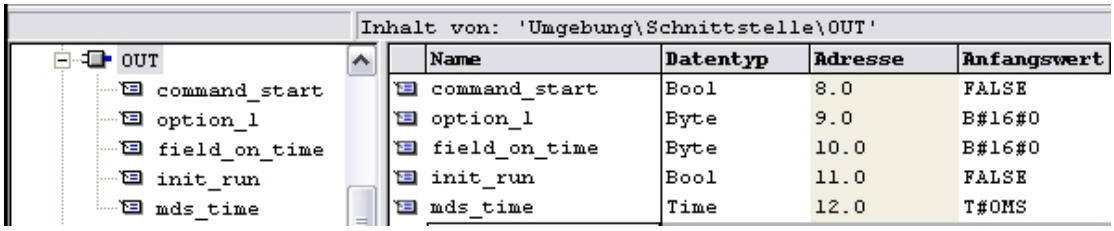
First, set up a new function block FB10.



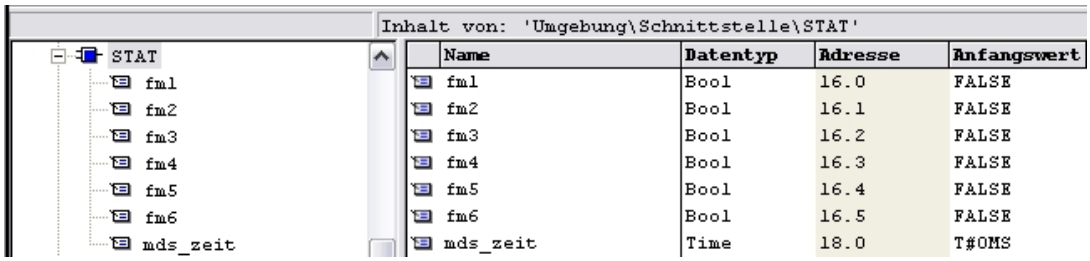
Then, set up IN variables



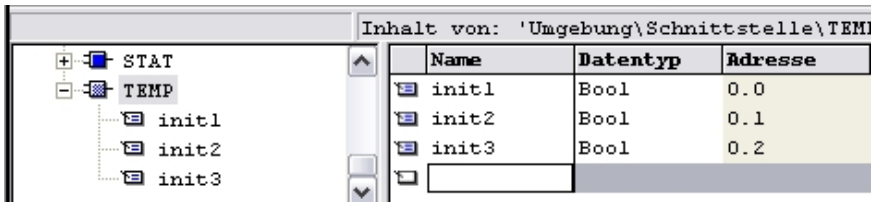
Next, set up OUT variables



Then, set up STAT variables



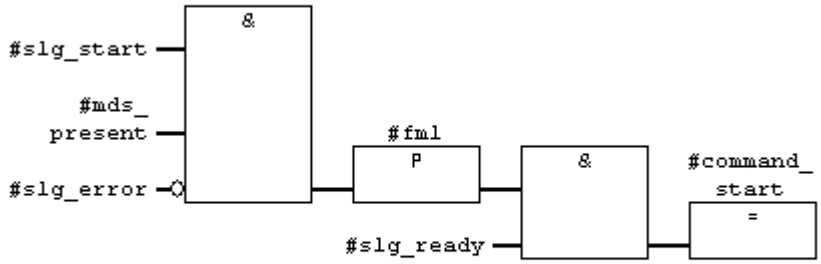
Next, set up TEMP variables



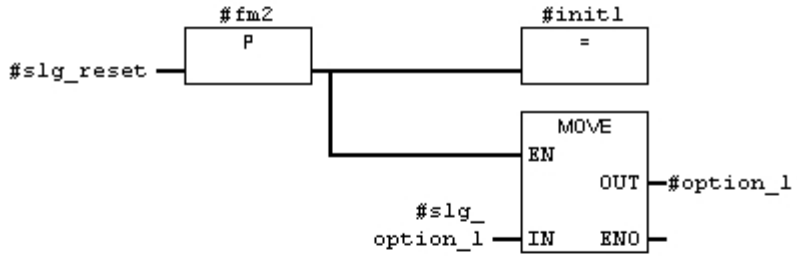
Networks 1 to 3

FB10 : Reader block

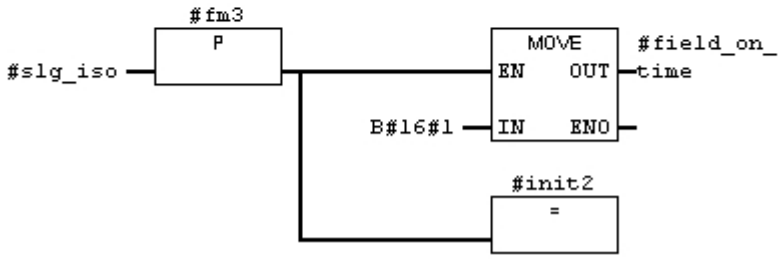
Network 1 :Command start



Network 2 : Reset Reader

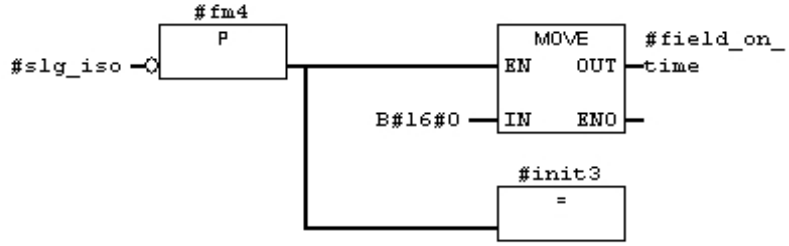


Network 3: ISO mode

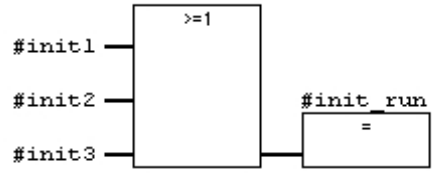


Networks 4 to 8

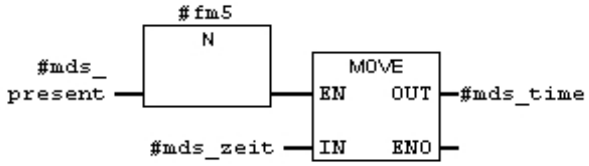
Network 4 RF300 mode



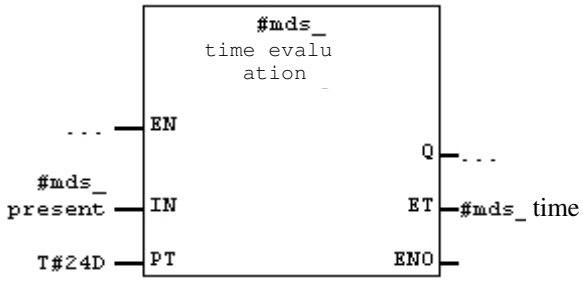
Network 5 : init_run



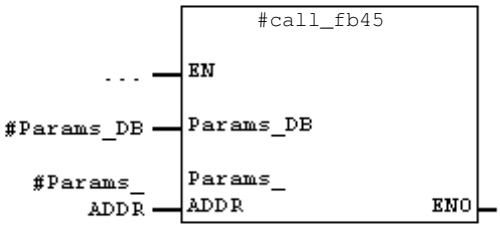
Network 6 : Attendance time MDS at reader



Network 7 : Reader1 MDS time evaluaton



Network 8 : FB45 call



Note regarding networks 7and 8:

The switch-on delay **TON (SFB4)** in Network 7 and the **FB45** call in Network 8 are added as multi-instance. After insertion in the program, click on the block with the right mouse key and select "**Change to multi-instance call**".
Then enter the name of the multi-instance in the window (refer to NW7 or NW8 **without #**).

Save and close FB10

5.12 FB1 Control Program

It is now possible to generate the control program with the completed blocks.

Task:
In our sample program, two command strings with two requests each will be processed.

With the **START_SLG1 (I0.0),** the command chain of the first reader is executed.
First, the MDS status information will be read, and then a write command is carried out on Reader1.
With Reader1, we write 8 bytes that are stored in DB48 starting with DBB0 to the MDS. With the **RESET_SLG1** (I0.1), the error is reset if there is an error (LED at Reader1 flashes red) and a "init_run command" to reset the first reader is executed.
With the **RF300_ISO** (I02), switching between MDS types (for example RF360T) or ISO transponder (for example, Moby D ISO) is to be carried out on the first reader (RF310R).

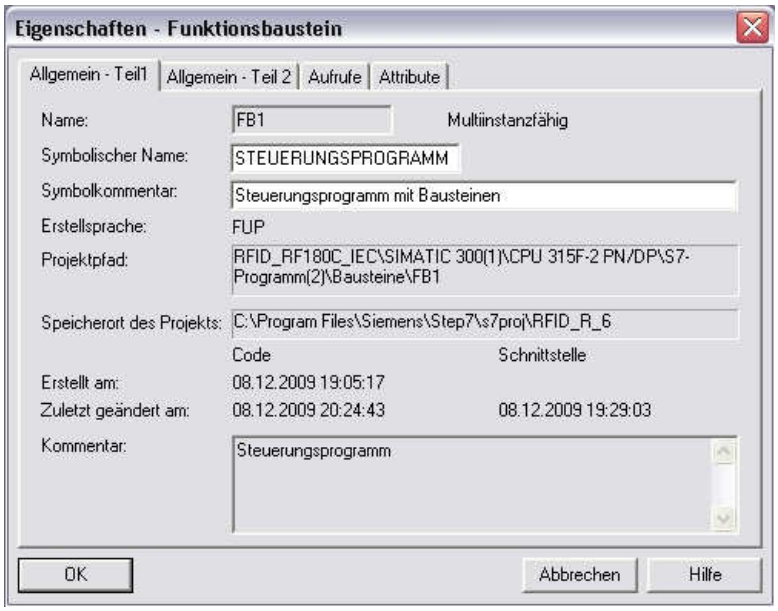
With the **START_SLG2** (I1.0), the command string of the second reader is executed.
First, the MDS diagnosis information will be read and then a read command will be executed at Reader2. With Reader2, we read 8 bytes from the transponder and then write them to DB48 starting with DBB50.
With **RESET_SLG2** (I1.1) the error is reset if there is an error (LED at SLG2 flashes red) and a "init_run command" is executed to reset the second reader.
In addition, the attendance time of the transponders is to be recorded on the readers.

Supplementing the symbol table

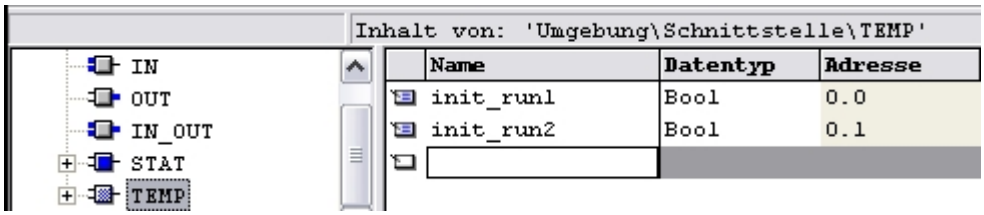
Symbol	Address		Data Type	Comment
START_SLG1	I	0.0	BOOL	Command start of Reader1
RESET_SLG1	I	0.1	BOOL	Reset Reader1 error
RF300_ISO	I	0.2	BOOL	Value 0 = RF300, Value 1 = ISO
START_SLG2	I	1.0	BOOL	Command start of SLG2
RESET_SLG2	I	1.1	BOOL	Reset Reader2 error
AWZ_SLG1	MD	40	TIME	Attendance time of transponder at SLG1
AWZ_SLG2	MD	50	TIME	Attendance time of transponder at SLG2
STATUS_SLG_1	VAT	1		SLG1 variable table
STATUS_SLG_2	VAT	2		SLG2 variable table

Generating FB1

Set up a new FB1.


Eigenschaften - Funktionsbaustein
Allgemein - Teil1
Allgemein - Teil 2
Aufrufe
Attribute
Name:
FB1
Multiinstanzfähig
Symbolischer Name:
STEUERUNGSPROGRAMM
Symbolkommentar:
Steuerungsprogramm mit Bausteinen
Erstellsprache:
FUP
Projektpfad:
RFID_RF180C_IEC\SIMATIC 300(1)\CPU 315F-2 PN/DP\S7-Programm(2)\Bausteine\FB1
Speicherort des Projekts:
C:\Program Files\Siemens\Step7\s7proj\RFID_R_6
Code
Schnittstelle
Erstellt am:
08.12.2009 19:05:17
Zuletzt geändert am:
08.12.2009 20:24:43
08.12.2009 19:29:03
Kommentar:
Steuerungsprogramm
OK
Abbrechen
Hilfe

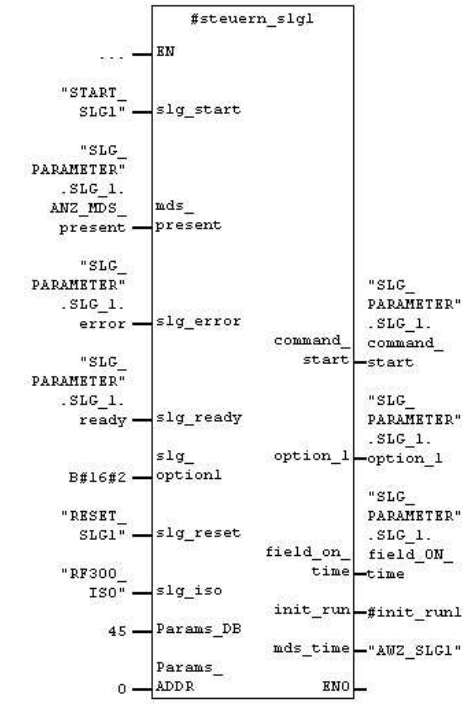
Setting up TEMP Variables



Network 1 to 2

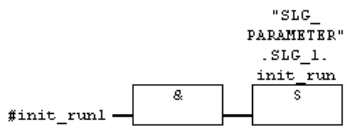
FB10 is inserted as a multi-instance block.

```
FB1 : Steuerungsprogramm
Netzwerk 1: Steuerbefehle zu SLG1 aufrufen
```



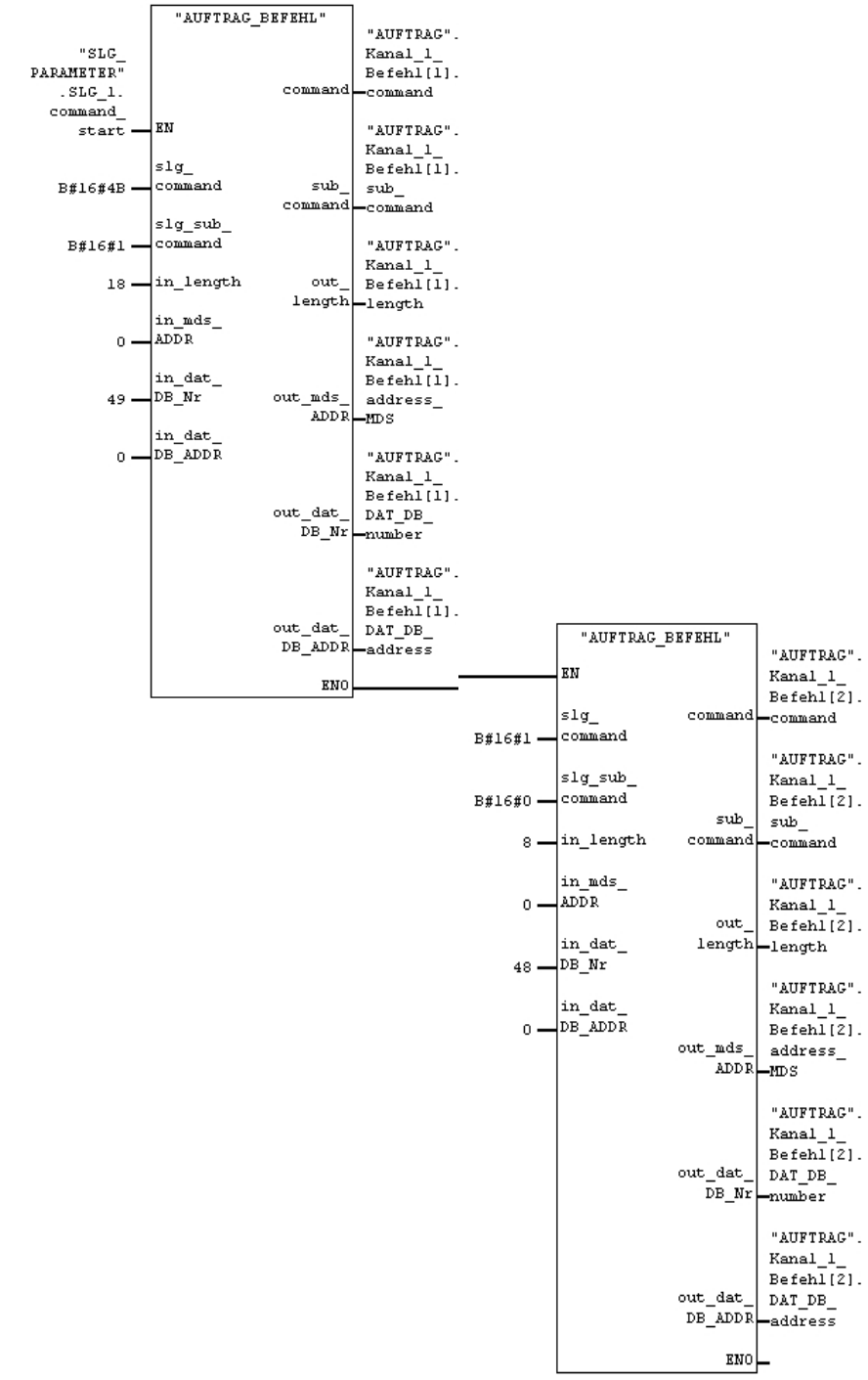

```
Symbolinformation:
START_SLG1                                  E0.0          -- Befehlsstart des SLG1
"SLG_PARAMETER".SLG_1.ANZ_MDS_present       DB45.DBX18.0  -- Anwesenheit eines MDS
"SLG_PARAMETER".SLG_1.error                 DB45.DBX18.6  -- Fehler während der Befehlsbearbeitung
"SLG_PARAMETER".SLG_1.ready                 DB45.DBX18.7  -- Der Befehl bzw. Befehlskette ist ausgeführt
RESET_SLG1                                  E0.1          -- SLG1 Fehler rücksetzen
RF300_ISO                                   E0.2          -- Wert 0 = RF300, Wert 1 = ISO
"SLG_PARAMETER".SLG_1.command_start         DB45.DBX19.1  -- Set: Startsignal für Befehl bzw. Befehlskette
"SLG_PARAMETER".SLG_1.option_1              DB45.DBB12    -- Input: RESET-Befehl Option 1
"SLG_PARAMETER".SLG_1.field_ON_time         DB45.DBB16    -- Input: BERO-Zeit
AWZ_SLG1                                    MD40          -- Anwesenheitszeit des Transpnders am SLG1
```

```
Netzwerk 2: Set:SLG1 ASM rücksetzen und neu parametrieren
```



Network 3

Netzwerk 3: Auftrag oder Befehl zu SLG1 aufrufen

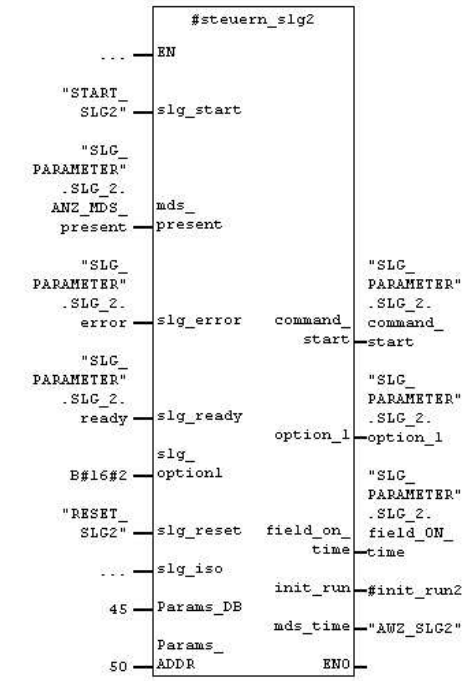


Symbolinformation:

AUFTRAG_BEFEHL	FC11	-- Baustein für einen Auftrag bzw. Befehl
"SLG_PARAMETER".SLG_1.command_start	DB45.DBX19.1	-- Set: Startsignal für Befehl bzw. Befehlskette
"AUFTRAG".Kanal_1_Befehl[1].command	DB47.DBB0	-- MDS Befehl:1=schreiben,2=lesen,3=init,4=slg-status,8=end,A=set-ant,B=mds-status
"AUFTRAG".Kanal_1_Befehl[1].sub_command	DB47.DBB1	-- Bitmuster für INIT-Befehl; Modus für END,SET-ANT,MDS-STATUS,SLG-STATUS
"AUFTRAG".Kanal_1_Befehl[1].length	DB47.DBW2	-- Anzahl der zu schreibenden/lesenden Daten in Bytes
"AUFTRAG".Kanal_1_Befehl[1].address_MDS	DB47.DBW4	-- Anfangsadresse auf MDS; Endadresse bei INIT;KW/Jahr für MDS-STATUS
"AUFTRAG".Kanal_1_Befehl[1].DAT_DB_number	DB47.DBW6	-- Nummer des DAT DB; Daten DB für MDS Daten
"AUFTRAG".Kanal_1_Befehl[1].DAT_DB_address	DB47.DBW8	-- Zeiger auf das Anfangswort im DAT DB
"AUFTRAG".Kanal_1_Befehl[2].command	DB47.DBB10	-- MDS Befehl:1=schreiben,2=lesen,3=init,4=slg-status,8=end,A=set-ant,B=mds-status
"AUFTRAG".Kanal_1_Befehl[2].sub_command	DB47.DBB11	-- Bitmuster für INIT-Befehl; Modus für END,SET-ANT,MDS-STATUS,SLG-STATUS
"AUFTRAG".Kanal_1_Befehl[2].length	DB47.DBW12	-- Anzahl der zu schreibenden/lesenden Daten in Bytes
"AUFTRAG".Kanal_1_Befehl[2].address_MDS	DB47.DBW14	-- Anfangsadresse auf MDS; Endadresse bei INIT;KW/Jahr für MDS-STATUS
"AUFTRAG".Kanal_1_Befehl[2].DAT_DB_number	DB47.DBW16	-- Nummer des DAT DB; Daten DB für MDS Daten
"AUFTRAG".Kanal_1_Befehl[2].DAT_DB_address	DB47.DBW18	-- Zeiger auf das Anfangswort im DAT DB

Network 4 to 5

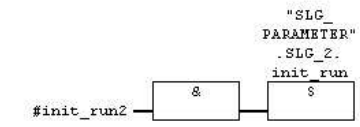
Netzwerk 4 : Steuerbefehle zu SLG2 aufrufen



Symbolinformation:

START_SLG2	E1.0	-- Befehlsstart des SLG2
"SLG_PARAMETER".SLG_2.ANZ_MDS_present	DB45.DBX68.0	-- Anwesenheit eines MDS
"SLG_PARAMETER".SLG_2.error	DB45.DBX68.6	-- Fehler während der Befehlsbearbeitung
"SLG_PARAMETER".SLG_2.ready	DB45.DBX68.7	-- Der Befehl bzw. Befehlskette ist ausgeführt
RESET_SLG2	E1.1	-- SLG2 Fehler rücksetzen
"SLG_PARAMETER".SLG_2.command_start	DB45.DBX69.1	-- Set: Startsignal für Befehl bzw. Befehlskette
"SLG_PARAMETER".SLG_2.option_1	DB45.DBB62	-- Input: RESET-Befehl Option 1
"SLG_PARAMETER".SLG_2.field_ON_time	DB45.DBB66	-- Input: BERO-Zeit
AWZ_SLG2	MD50	-- Anwesenheitszeit des Transpnders am SLG2

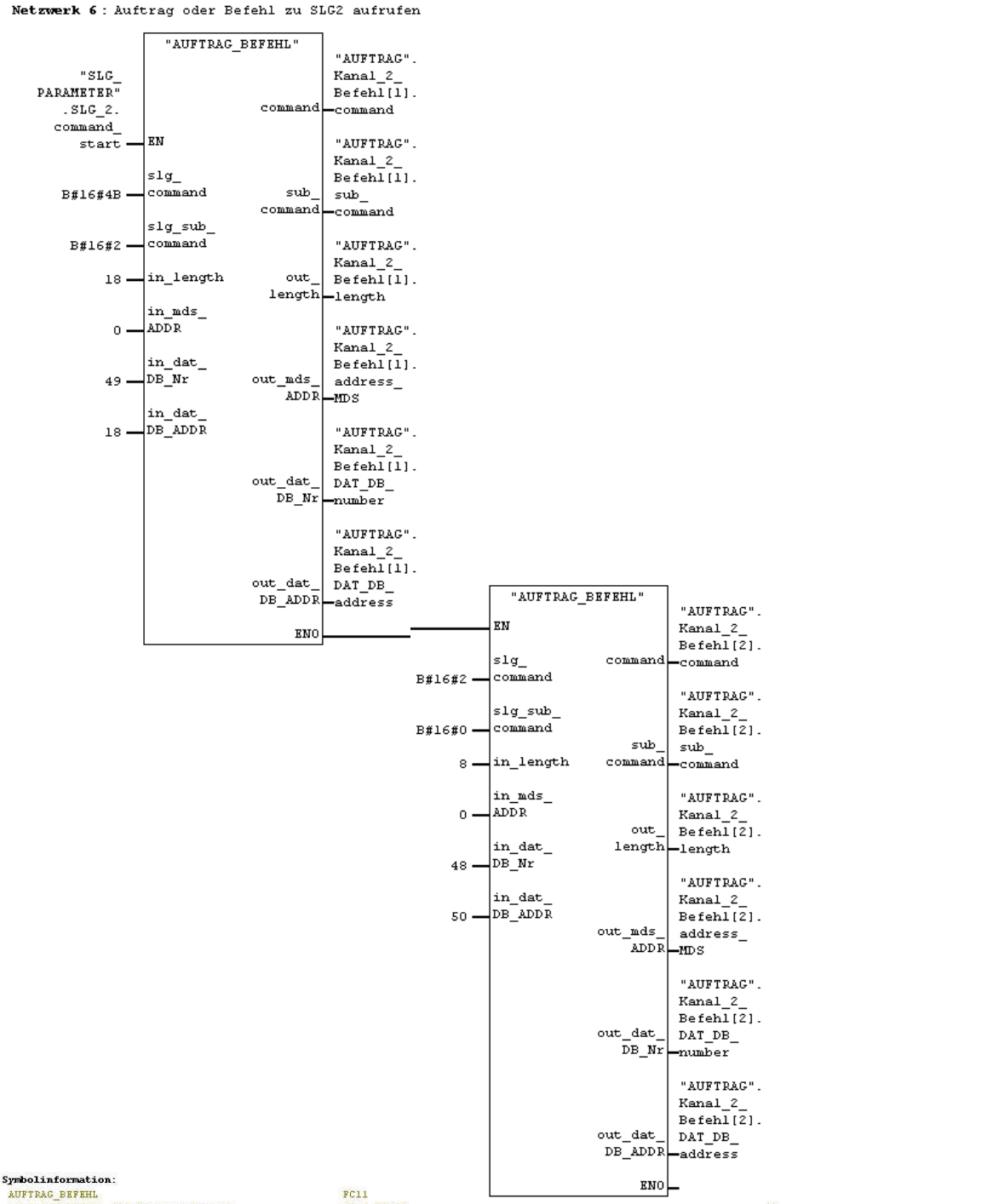
Netzwerk 5 : Set: SLG2 ASM rücksetzen und neu parametrieren



Symbolinformation:

"SLG_PARAMETER".SLG_2.init_run	DB45.DBX69.3	-- Set: ASM rücksetzen und neu parametrieren

Network 6

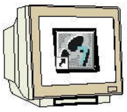


Symbolinformation:

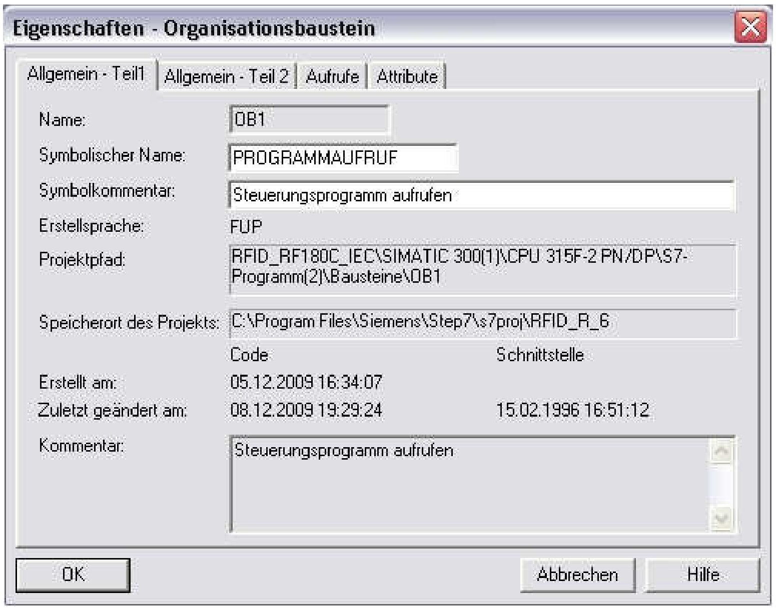
AUFTRAG_BEFEHL	FC11	
"SLG_PARAMETER".SLG_2.command_start	DB45.DBX69.	te
"AUFTRAG".Kanal_2_Befehl[1].command	DB47.DBB50	-- MDS Befehl:1=schreiben,2=lesen,3=init,4=slg-status,8=end,A=set-ant,B=mds-status
"AUFTRAG".Kanal_2_Befehl[1].sub_command	DB47.DBB51	-- Bitmuster für INIT-Befehl; Modus für END,SET-ANT,MDS-STATUS,SLG-STATUS
"AUFTRAG".Kanal_2_Befehl[1].length	DB47.DBW52	-- Anzahl der zu schreibenden/lesenden Daten in Bytes
"AUFTRAG".Kanal_2_Befehl[1].address_MDS	DB47.DBW54	-- Anfangsadresse auf MDS; Endadresse bei INIT;KW/Jahr für MDS-STATUS
"AUFTRAG".Kanal_2_Befehl[1].DAT_DB_number	DB47.DBW56	-- Nummer des DAT DB; Daten DB für MDS Daten
"AUFTRAG".Kanal_2_Befehl[1].DAT_DB_address	DB47.DBW58	-- Zeiger auf das Anfangswort im DAT DB
"AUFTRAG".Kanal_2_Befehl[2].command	DB47.DBB60	-- MDS Befehl:1=schreiben,2=lesen,3=init,4=slg-status,8=end,A=set-ant,B=mds-status
"AUFTRAG".Kanal_2_Befehl[2].sub_command	DB47.DBB61	-- Bitmuster für INIT-Befehl; Modus für END,SET-ANT,MDS-STATUS,SLG-STATUS
"AUFTRAG".Kanal_2_Befehl[2].length	DB47.DBW62	-- Anzahl der zu schreibenden/lesenden Daten in Bytes
"AUFTRAG".Kanal_2_Befehl[2].address_MDS	DB47.DBW64	-- Anfangsadresse auf MDS; Endadresse bei INIT;KW/Jahr für MDS-STATUS
"AUFTRAG".Kanal_2_Befehl[2].DAT_DB_number	DB47.DBW66	-- Nummer des DAT DB; Daten DB für MDS Daten
"AUFTRAG".Kanal_2_Befehl[2].DAT_DB_address	DB47.DBW68	-- Zeiger auf das Anfangswort im DAT DB

Save and close FB1

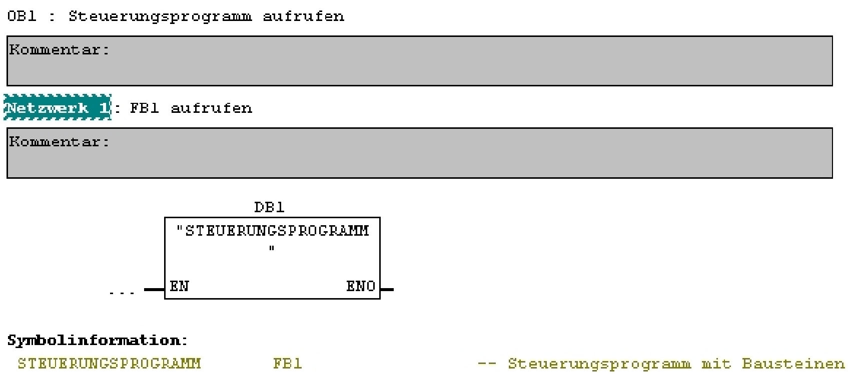
5.13 OB1 Program Call



Double click on OB1 in the project window, or open the object properties of OB1 and enter the symbolic name and the symbol comment.



Open OB1 and call FB1 with DB1



Confirm the window with the query for generating DB1 by clicking on **"Yes"**.

Save and close OB1

We can now load the program into the controller and test it.

5.14 Variable Table STATUS_SLG_1

Operand	Symbol	Anzeigeformat	Statuswert	Steuerwert
// Cancel				
DB45.DBX 19.0	"SLG_PARAMETER".SLG_1.cancel	BOOL	false	
// Command Start				
DB45.DBX 19.1	"SLG_PARAMETER".SLG_1.command_start	BOOL	false	
// System Start Up				
DB45.DBX 19.3	"SLG_PARAMETER".SLG_1.init_run	BOOL	false	
// Ready				
DB45.DBX 18.7	"SLG_PARAMETER".SLG_1.ready	BOOL	true	
// Presence of a MDS				
DB45.DBX 18.0	"SLG_PARAMETER".SLG_1.ANZ_MDS_present	BOOL	false	
MD 40	"AWZ_SLG1"	ZEIT	T#5s21ms	
// Error				
DB45.DBX 18.6	"SLG_PARAMETER".SLG_1.error	BOOL	false	
// Errors				
DB45.DBB 22	"SLG_PARAMETER".SLG_1.error_MOBY	HEX	B#16#00	
DB45.DBB 23	"SLG_PARAMETER".SLG_1.error_FB	HEX	B#16#00	
// MOBY Command				
DB47.DBB 0	"AUFTRAG".Kanal_1_Befehl[1].command	HEX	B#16#4B	
DB47.DBB 1	"AUFTRAG".Kanal_1_Befehl[1].sub_command	HEX	B#16#01	
DB47.DBW 2	"AUFTRAG".Kanal_1_Befehl[1].length	DEZ	18	
DB47.DBW 4	"AUFTRAG".Kanal_1_Befehl[1].address_MDS	HEX	W#16#0000	
DB47.DBW 6	"AUFTRAG".Kanal_1_Befehl[1].DAT_DB_number	DEZ	49	
DB47.DBW 8	"AUFTRAG".Kanal_1_Befehl[1].DAT_DB_address	DEZ	0	
DB48.DBB 0	"S_L_DATEN".Daten[1]	ZEICHEN	'R'	'R'
DB48.DBB 1	"S_L_DATEN".Daten[2]	ZEICHEN	'F'	'F'
DB48.DBB 2	"S_L_DATEN".Daten[3]	ZEICHEN	'I'	'I'
DB48.DBB 3	"S_L_DATEN".Daten[4]	ZEICHEN	'D'	'D'
DB48.DBB 4	"S_L_DATEN".Daten[5]	ZEICHEN	'T'	'T'
DB48.DBB 5	"S_L_DATEN".Daten[6]	ZEICHEN	'E'	'E'
DB48.DBB 6	"S_L_DATEN".Daten[7]	ZEICHEN	'S'	'S'
DB48.DBB 7	"S_L_DATEN".Daten[8]	ZEICHEN	'T'	'T'
DB48.DBB 50	"S_L_DATEN".Daten[51]	ZEICHEN	B#16#00	
DB48.DBB 51	"S_L_DATEN".Daten[52]	ZEICHEN	B#16#00	
DB48.DBB 52	"S_L_DATEN".Daten[53]	ZEICHEN	B#16#00	
DB48.DBB 53	"S_L_DATEN".Daten[54]	ZEICHEN	B#16#00	
DB48.DBB 54	"S_L_DATEN".Daten[55]	ZEICHEN	B#16#00	
DB48.DBB 55	"S_L_DATEN".Daten[56]	ZEICHEN	B#16#00	
DB48.DBB 56	"S_L_DATEN".Daten[57]	ZEICHEN	B#16#00	
DB48.DBB 57	"S_L_DATEN".Daten[58]	ZEICHEN	B#16#00	
DB45.DBB 16	"SLG_PARAMETER".SLG_1.field_ON_time	HEX	B#16#00	
DB47.DBB 10	"AUFTRAG".Kanal_1_Befehl[2].command	HEX	B#16#01	
DB47.DBB 11	"AUFTRAG".Kanal_1_Befehl[2].sub_command	HEX	B#16#00	
DB47.DBW 12	"AUFTRAG".Kanal_1_Befehl[2].length	DEZ	8	
DB47.DBW 14	"AUFTRAG".Kanal_1_Befehl[2].address_MDS	HEX	W#16#0000	
DB47.DBW 16	"AUFTRAG".Kanal_1_Befehl[2].DAT_DB_number	DEZ	48	
DB47.DBW 18	"AUFTRAG".Kanal_1_Befehl[2].DAT_DB_address	DEZ	0	

5.15 Variable Table STATUS_SLG_2

Operand	Symbol	Anzeigeformat	Statuswert	Steuerwert
// Cancel				
DB45.DBX 69.0	"SLG_PARAMETER".SLG_2.cancel	BOOL	false	
// Command Start				
DB45.DBX 69.1	"SLG_PARAMETER".SLG_2.command_start	BOOL	false	
// System Start Up				
DB45.DBX 69.3	"SLG_PARAMETER".SLG_2.init_run	BOOL	false	
// Ready				
DB45.DBX 68.7	"SLG_PARAMETER".SLG_2.ready	BOOL	true	
// Presence of a MDS				
DB45.DBX 68.0	"SLG_PARAMETER".SLG_2.ANZ_MDS_present	BOOL	false	
MD 50	"AWZ_SLG2"	ZEIT	T#3s994ms	
// Error				
DB45.DBX 68.6	"SLG_PARAMETER".SLG_2.error	BOOL	false	
// Errors				
DB45.DBB 72	"SLG_PARAMETER".SLG_2.error_MOBY	HEX	B#16#00	
DB45.DBB 73	"SLG_PARAMETER".SLG_2.error_FB	HEX	B#16#00	
// MOBY Command				
DB47.DBB 50	"AUFTRAG".Kanal_2_Befehl[1].command	HEX	B#16#4B	
DB47.DBB 51	"AUFTRAG".Kanal_2_Befehl[1].sub_command	HEX	B#16#02	
DB47.DBW 52	"AUFTRAG".Kanal_2_Befehl[1].length	DEZ	18	
DB47.DBW 54	"AUFTRAG".Kanal_2_Befehl[1].address_MDS	HEX	W#16#0000	
DB47.DBW 56	"AUFTRAG".Kanal_2_Befehl[1].DAT_DB_number	DEZ	49	
DB47.DBW 58	"AUFTRAG".Kanal_2_Befehl[1].DAT_DB_address	DEZ	18	
DB48.DBB 0	"S_L_DATEN".Daten[1]	ZEICHEN	'R'	
DB48.DBB 1	"S_L_DATEN".Daten[2]	ZEICHEN	'F'	
DB48.DBB 2	"S_L_DATEN".Daten[3]	ZEICHEN	'I'	
DB48.DBB 3	"S_L_DATEN".Daten[4]	ZEICHEN	'D'	
DB48.DBB 4	"S_L_DATEN".Daten[5]	ZEICHEN	'T'	
DB48.DBB 5	"S_L_DATEN".Daten[6]	ZEICHEN	'E'	
DB48.DBB 6	"S_L_DATEN".Daten[7]	ZEICHEN	'S'	
DB48.DBB 7	"S_L_DATEN".Daten[8]	ZEICHEN	'T'	
DB48.DBB 50	"S_L_DATEN".Daten[51]	ZEICHEN	'R'	
DB48.DBB 51	"S_L_DATEN".Daten[52]	ZEICHEN	'F'	
DB48.DBB 52	"S_L_DATEN".Daten[53]	ZEICHEN	'I'	
DB48.DBB 53	"S_L_DATEN".Daten[54]	ZEICHEN	'D'	
DB48.DBB 54	"S_L_DATEN".Daten[55]	ZEICHEN	'T'	
DB48.DBB 55	"S_L_DATEN".Daten[56]	ZEICHEN	'E'	
DB48.DBB 56	"S_L_DATEN".Daten[57]	ZEICHEN	'S'	
DB48.DBB 57	"S_L_DATEN".Daten[58]	ZEICHEN	'T'	
DB47.DBB 60	"AUFTRAG".Kanal_2_Befehl[2].command	HEX	B#16#02	
DB47.DBB 61	"AUFTRAG".Kanal_2_Befehl[2].sub_command	HEX	B#16#00	
DB47.DBW 62	"AUFTRAG".Kanal_2_Befehl[2].length	DEZ	8	
DB47.DBW 64	"AUFTRAG".Kanal_2_Befehl[2].address_MDS	HEX	W#16#0000	
DB47.DBW 66	"AUFTRAG".Kanal_2_Befehl[2].DAT_DB_number	DEZ	48	
DB47.DBW 68	"AUFTRAG".Kanal_2_Befehl[2].DAT_DB_address	DEZ	50	

5.16 Symbol Table

Symbol	Adresse	Datentyp	Kommentar
SLG_PARAMETER	DB 45	DB 45	hier findet man die Parameter zu den SLGs
AUFTRAG	DB 47	DB 47	hier werden die Kommandos eingegeben
S_L_DATEN	DB 48	DB 48	hier werden die Daten des Schreib-Lese Auftrags eingetragen
MDS_STATUS	DB 49	DB 49	hier werden die MDS-Status Informationen abgelegt
SLG_STATUS	DB 50	DB 50	hier werden die SLG-Status Informationen abgelegt
START_SLG1	E 0.0	BOOL	Befehlsstart des SLG1
RESET_SLG1	E 0.1	BOOL	SLG1 Fehler rücksetzen
RF300_ISO	E 0.2	BOOL	Wert 0 = RF300, Wert 1 = ISO
START_SLG2	E 1.0	BOOL	Befehlsstart des SLG2
RESET_SLG2	E 1.1	BOOL	SLG2 Fehler rücksetzen
STEUERUNGSPROGR...	FB 1	FB 1	Steuerungsprogramm mit Bausteinen
SLG_STEUERUNG	FB 10	FB 10	Steuerbefehle zu einem SLG
MOBY FB	FB 45	FB 45	
AUFTRAG_BEFEHL	FC 11	FC 11	Baustein für einen Auftrag bzw. Befehl
AWZ_SLG1	MD 40	TIME	Anwesenheitszeit des Transpnders am SLG1
AWZ_SLG2	MD 50	TIME	Anwesenheitszeit des Transpnders am SLG2
PROGRAMMAUFRUF	OB 1	OB 1	Steuerungsprogramm aufrufen
NEUSTART	OB 100	OB 100	Programmierung von Neustart und Wiederanlauf
TON	SFB 4	SFB 4	Generate an On Delay
RDREC	SFB 52	SFB 52	Read a Process Data Record
WRREC	SFB 53	SFB 53	Write a Process Data Record
MOBY Param_d	UDT 11	UDT 11	Parametrier DB
MOBY CMD_d	UDT 21	UDT 21	Command DB Normaladressierung
MOBY SLG-Status_d	UDT 111	UDT 111	MOBY: SLG Status Daten
MOBY P MDS-Status ...	UDT 261	UDT 261	MOBY P: MDS Status Daten
MOBY P MDS-Status ...	UDT 271	UDT 271	MOBY P: MDS Status Daten
MOBY P SLG-Status_d	UDT 281	UDT 281	MOBY P: SLG Status Daten
STATUS_SLG_1	VAT 1		Variablentabelle des SLG1
STATUS_SLG_2	VAT 2		Variablentabelle des SLG2

5.17 Block Folder

Objektname	Symbolischer Name	Erstellsprache	Größe im Arbeitsspei...	Typ	Versic
Systemdaten	---	---	---	SDB	---
OB1	PROGRAMMAUFRUF	FUP	72	Organisationsbaustein	0.1
OB100	NEUSTART	AWL	52	Organisationsbaustein	0.1
FB1	STEUERUNGSPROGRA...	FUP	1242	Funktionsbaustein	0.1
FB10	SLG_STEUERUNG	FUP	346	Funktionsbaustein	0.1
FB45	MOBY FB	AWL	7846	Funktionsbaustein	1.3
FC11	AUFTRAG_BEFEHL	FUP	98	Funktion	0.1
DB1		DB	972	Instanzdatenbaustei...	0.0
DB45	SLG_PARAMETER	DB	136	Datenbaustein	0.1
DB47	AUFTRAG	DB	136	Datenbaustein	0.1
DB48	S_L_DATEN	DB	1060	Datenbaustein	0.1
DB49	MDS_STATUS	DB	72	Datenbaustein	0.1
DB50	SLG_STATUS	DB	92	Datenbaustein	0.1
UDT11	MOBY Param_d	AWL	---	Datentyp	1.3
UDT21	MOBY CMD_d	AWL	---	Datentyp	1.3
UDT111	MOBY SLG-Status_d	AWL	---	Datentyp	1.3
UDT261	MOBY P MDS-Status 0_d	AWL	---	Datentyp	1.3
UDT271	MOBY P MDS-Status 1_d	AWL	---	Datentyp	1.3
UDT281	MOBY P SLG-Status_d	AWL	---	Datentyp	1.3
STATUS_SLG_1	STATUS_SLG_1		---	Variablentabelle	1.3
STATUS_SLG_2	STATUS_SLG_2		---	Variablentabelle	1.3
SFB4	TON	AWL	---	Systemfunktionsbau...	1.0
SFB52	RDREC	AWL	---	Systemfunktionsbau...	1.0
SFB53	WRREC	AWL	---	Systemfunktionsbau...	1.0

5.18 DB49 Data View

@DB49 -- "MDS_STATUS" -- RFID_RF180C_IEC\SIMATIC 300(1)\CPU 315F-2 PN/DP\...\DB49 ONLINE

Adresse	Name	Typ	Anfangswert	Aktualwert	Kommentar
0.0	MDS_Status0.reserved0	BYTE	B#16#0	B#16#01	
1.0	MDS_Status0.status_info	BYTE	B#16#0	B#16#00	Modus MDS-Status
2.0	MDS_Status0.UID[1].Byte_1_4	DWORD	DW#16#0	DW#16#00000000	MDS-Nummer (unique identifier)
6.0	MDS_Status0.UID[1].Byte_5_8	DWORD	DW#16#0	DW#16#557AFEFA	
10.0	MDS_Status0.MDS_type	BYTE	B#16#0	B#16#02	MDS-Typ
11.0	MDS_Status0.Lock_state	BYTE	B#16#0	B#16#00	Schreibschutzstatus EEPROM
12.0	MDS_Status0.reserved1[1]	BYTE	B#16#0	B#16#00	
13.0	MDS_Status0.reserved1[2]	BYTE	B#16#0	B#16#00	
14.0	MDS_Status0.reserved1[3]	BYTE	B#16#0	B#16#00	
15.0	MDS_Status0.reserved1[4]	BYTE	B#16#0	B#16#00	
16.0	MDS_Status0.reserved1[5]	BYTE	B#16#0	B#16#00	
17.0	MDS_Status0.reserved1[6]	BYTE	B#16#0	B#16#00	
18.0	MDS_Status1.reserved0	BYTE	B#16#0	B#16#02	
19.0	MDS_Status1.status_info	BYTE	B#16#0	B#16#00	Modus MDS-Status
20.0	MDS_Status1.UID[1].Byte_1_4	DWORD	DW#16#0	DW#16#00000000	MDS-Nummer (unique identifier)
24.0	MDS_Status1.UID[1].Byte_5_8	DWORD	DW#16#0	DW#16#557BA57F	
28.0	MDS_Status1.LFD	BYTE	B#16#0	B#16#0D	Leistungsflussdichte: Beziehung zwischen Grenzwert ur
29.0	MDS_Status1.FZP	BYTE	B#16#0	B#16#00	Fehlerzähler passiv (Ruhefehlerzähler)
30.0	MDS_Status1.FZA	BYTE	B#16#0	B#16#03	Fehlerzähler aktiv (Fehler während Kommunikation)
31.0	MDS_Status1.ANWZ	BYTE	B#16#0	B#16#FF	Anwesenheitszähler
32.0	MDS_Status1.reserved1[1]	BYTE	B#16#0	B#16#00	
33.0	MDS_Status1.reserved1[2]	BYTE	B#16#0	B#16#00	
34.0	MDS_Status1.reserved1[3]	BYTE	B#16#0	B#16#00	