SIEMENS

SIMATIC Sensors

RFID systems
RFID standard profile;
standard function for RFID systems

Function Manual

10/2010
J31069-D0179-U001-A2-7618

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

CAUTION

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

NOTICE

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation for the specific task, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be adhered to. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

order number: J31069-D0179-U001-A1-7618
℗ 09/2010

Table of contents

Introduction 1

1.1 Preface

Purpose of this document

This Function Manual contains all the information needed to configure and commission the system. The programming is explained in STL and SCL.

It is intended both for programming and testing/debugging personnel who commission the system themselves and connect it with other units (automation systems, further programming devices), as well as for service and maintenance personnel who install expansions or carry out fault/error analyses.

Scope of this documentation

This documentation is valid for FB 101, FB 116 and FB 132. It describes the implementation of the proxy ident function block (PIB) from PROFIBUS INTERNATIONAL, Version 1.02 Order No. 3.142 for SIMATIC S7.
The specification of the proxy ident function block is available on the Internet under `http://www.profibus.com`. You will find this there in the download area of PROFIBUS INTERNATIONAL under "Profiles".

The function manual describes the supply status as of October 2010.

History

Previous edition(s) of thís function manual:

Edition	Note
03/2006	First Edition
10/2010	Revised Edition

1.2 Navigating in the Function Manual

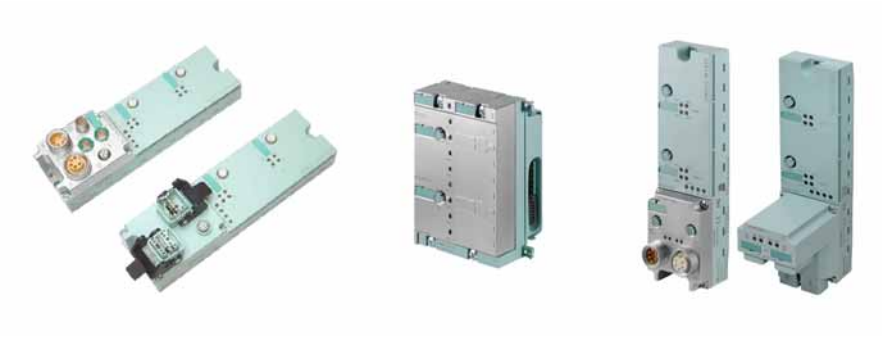
Structure of contents	Contents
Table of Contents	Organization of the documentation, including the index of pages and chapters
Introduction	Purpose, layout and description of the important topics
Description	Description of the properties and data structures of the RFID standard profile
Parameter settings	Description of the parameter settings
Commissioning	Description of the commissioning procedure
Error messages and troubleshooting	Overview of error messages and troubleshooting guide
Application examples	Describes the application of the RFID standard profile based on example applications.
Appendix: Short description of the communication modules	Description of the communication modules that can be used for the RFID standard profile
Appendix: Programming the RFID communication modules on PROFIBUS DP and PROFINET	Information for programmers of non-Siemens controllers
Appendix: Service & support	Service and support, contact partners, training centers

Description

2

2.1 Area of application and features

The RFID standard profile is a *STEP 7* function for RFID- systems. It can be used with in the SIMATIC S7-300 and S7-400 for various RFID communication modules.



ASM 456

RF170C
(available soon)

RF180C
(available soon)

Figure 2-1 Communication modules for RFID standard profile

The appendix *Brief description of the communication modules*illustrates configurations with the various communication modules. The RFID standard profile can be operated in various different configurations:

- The communication module is located in the rack of an ET 200pro. The ET 200pro is operated on an S7-300 or S7-400.
- Since the communication module is a self-contained PROFIBUS slave, it is linked to a SIMATIC S7-300 or S7-400 with integrated PROFIBUS connection.
- The communication module can be operated on PROFIBUS DP as well as on PROFINET IO.
- The communication module is operated on PROFINET IO via the IE/PB Link.
- The communication module is operated on a SIMOTION controller via SCOUT (V4.0 or higher).

These configurations can be mixed, and different interface modules can also be connected.

Demarcation of FB 45/FB 56

- RFID standard profile works with normal addressing and filehandler.
- RFID standard profile enables connection to non-Siemens controllers. This requires installation of the block for the RFID standard profile in the non-Siemens controller.

Performance features of the communication module

The performance features of the ASM 456 are described in the appendix *Short description of the communication modules > ASM 456*.

Since the performance range of individual communication modules is expanded continuously, you must always work with the latest edition of this description.

Requirement for operating the RFID standard profile

Remember that the RFID standard profile uses acyclic message frames (SFB 52/53). Older CPUs of the SIMATIC family or a small model series may not have these services. Be sure to check this during configuration.

- STEP 7 from V5.2
- SCL
- CD *RFID Systems Software & Documentation* from Edition 03/2006

Liability Disclaimer

Siemens AG accepts no functional liability for readers from other RFID vendors whose application interface is implemented in accordance with the "Proxy Ident Function Block" standard of PROFIBUS INTERNATIONAL.

2.2 Block specification

Block number:	FB 101, FB 116, FB 132
Block name:	FB 101, FB 116, FB 132
Symbolic name:	PIB1_KB, PIB16_KB, PIB32_KB
Family:	–
Work memory requirement:	11508 bytes
Local data:	34 bytes
Version:	1.06
Called blocks:	SFC 5, SFB 4, SFB 52, SFB 53
Block numbers used in LAD/FBD/STL:	FB 1, FB 2, FB 3, FB 4, FB 5, FC 1, FC 2, FC 3, FC 4, UDT 1
Data block resources:	Instance DB = 566 bytes per channel
Bit memories used:	none
Counters used:	none
Registers used:	AR1, AR2
Call:	cyclic

2.3 Building block structure

The function blocks act as the communication interface between a standard profile RFID system (e.g. ASM 456) and the user program. The function blocks (FBs) support the following functions here:

- Configuration
- Command execution
- Reading and writing of data
- Diagnostics

All function blocks support the same functions. The FB to be used is derived essentially from the length of the tag data.

Table 2- 1 Data length and the FB to be used

Length of the tag data to be processed		Function block to be used
Send data (TXBUF)	Receive data (RXBUF)	
1 KB	1 KB	FB 101 (PIB1_KB)
16 KB	16 KB	FB 116 (PIB16_KB)
32 KB	32 KB	FB 132 (PIB32_KB)

Example:

- If a data volume of 800 bytes is to be read from a 32KB tag, use of FB 101 is sufficient.
- However, if 1 KB of data have to be read, FB 116 is recommended because some parameterization data are added to the 1 KB of user data. This would exceed the maximum data length of FB 101.

Note

In this document, the description always refers to FB 101 (PIB1_KB). Function blocks FB 116 and FB 132 are configured, parameterized and programmed according to the same scheme.

Overview of the functions in FB 101

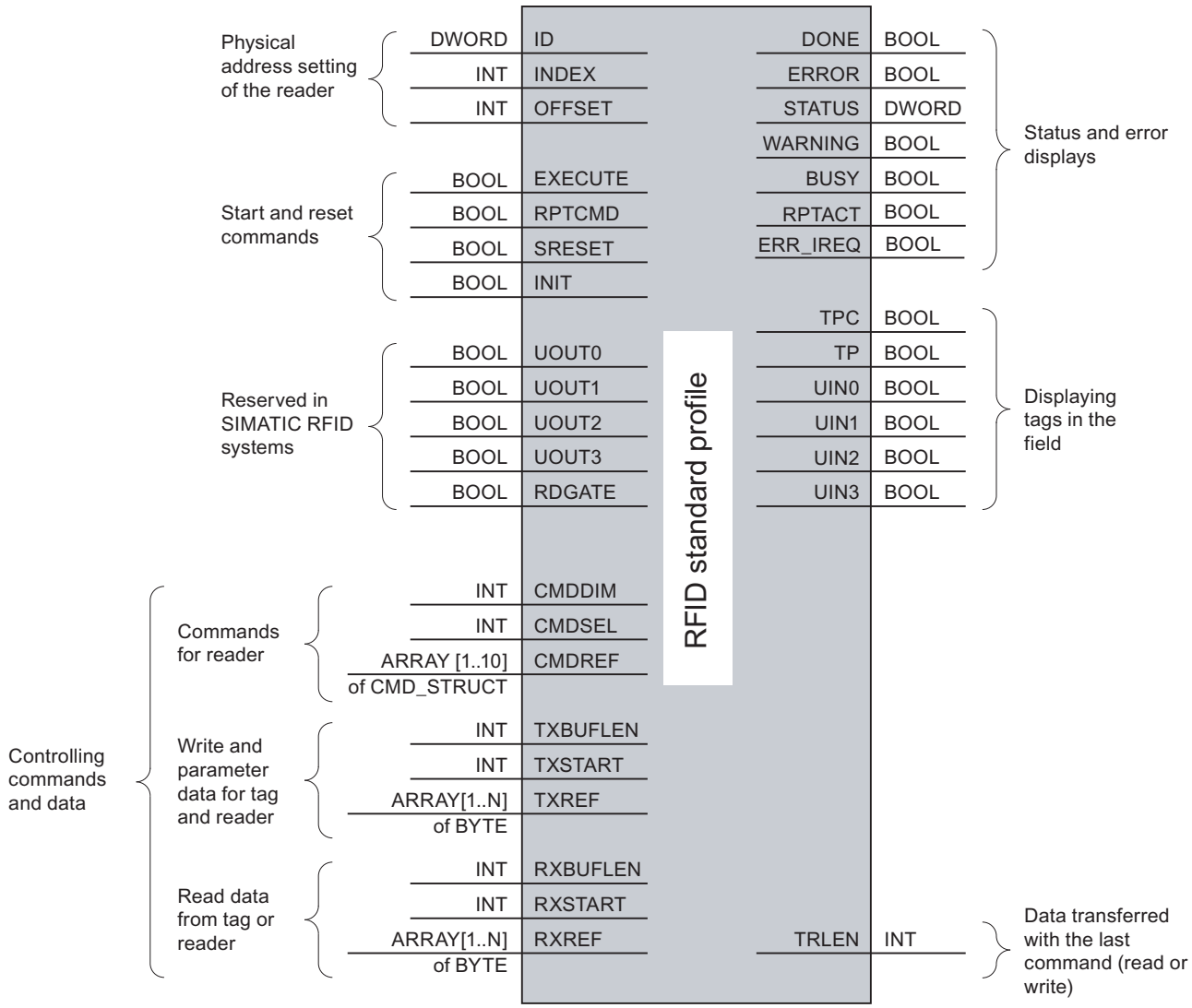
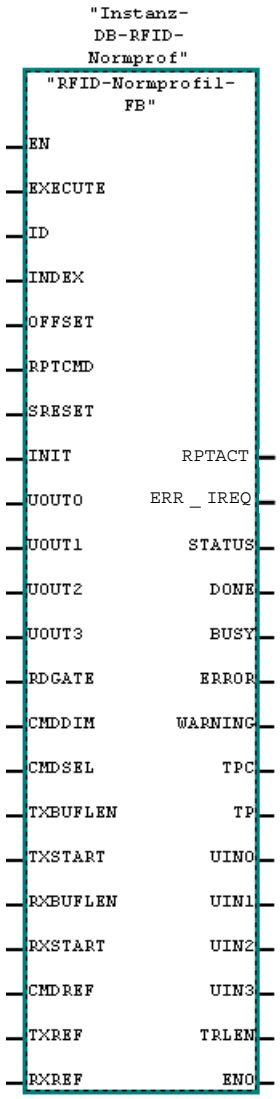


Figure 2-2 Overview of the functions in FB 101

RPTACT and ERR_IREQ under development

2.4 Configuration scheme

Table 2- 2 Configuration scheme for the function block of the RFID standard profile

Ladder logic programming box	Parameters	Data type	Default	Meaning	
	EN	BOOL	TRUE	Enabling input	
	EXECUTE	BOOL	FALSE	TRUE = Initiation of a new command Before starting, the user must set the command and the relevant parameters in the memory linked with **CMDREF**.	
	ID	DWORD	W#16#0	Basic address of the I/O device that has been set in HW Config **Important:** The basic addresses of the In- / Out-area must be identical.	
	INDEX	INT	0	Parameterization of the data record number for data exchange with the communication module	
				Channel 1	111
				Channel 2	112
	OFFSET	INT	0	Relative address to basic address of the channel-related cyclic I/O data (input parameter: ID).	
				Channel 1	0
				Channel 2	2
	RPTCMD [1]	BOOL	FALSE	TRUE = Repeating the command currently being executed or the next command to be executed by communication module.	
	SRESET	BOOL	FALSE	TRUE = Cancellation of the command currently processed in the communication module	
	INIT	BOOL	FALSE	TRUE = Communication module executes a Reset and is re-parameterized	
	UOUT0	BOOL	FALSE	With this input parameter, the user can set an output of the RFID system according to the standard profile. (not used on Siemens communication modules)	
	UOUT1	BOOL	FALSE		
	UOUT2	BOOL	FALSE		
	UOUT3	BOOL	FALSE		
	RDGATE	BOOL	FALSE	TRUE = Activation of the read gate (not used on Siemens communication modules)	
	CMDDIM	INT	0	Number of commands in the parameter (CMDREF); = 10	
	CMDSEL	INT	0	Selection of the command to be executed (CMDREF); 1 = 1st command The following must always apply: CMDSEL ≤ CMDDIM	
	TXBUFLEN	INT	0	Number of bytes used by this instance of the FB to store data to be sent.	
	TXSTART	INT	0	Relative position of the send data buffer (TXBUF) within the global memory area to which the parameter TXREF refers.	
	RXBUFLEN	INT	0	Number of bytes used by this instance of the FB to store data received.	

Ladder logic programming box	Parameters	Data type	Default	Meaning
	RXSTART	INT	0	Relative position of the receive data buffer (RXBUF) within the global memory area to which the parameter RXREF refers.
	RPTACT [1]	BOOL	FALSE	TRUE = RPTCMD is active
	ERR_IREQ [1]	BOOL	FALSE	TRUE = an error has occurred on the communication module or reader (e.g. at power-up)
	STATUS	DWORD	W#16#0	Warning and error specification If ERROR = TRUE or WARNING = TRUE, the error or warning information is contained in the STATUS parameter, see section *Error messages and troubleshooting*
	DONE	BOOL	FALSE	TRUE = Command was executed successfully
	BUSY	BOOL	FALSE	TRUE = FB processes a command, and other commands cannot be started (except INIT and SRESET)
	ERROR	BOOL	FALSE	TRUE = Error has been detected, specification of the error in the STATUS parameter The bit is reset when a new command is started.
	WARNING	BOOL	FALSE	TRUE = Warning has been detected, specification of the warning in the STATUS parameter If the ERROR parameter is not simultaneously set, the data have been correctly processed. The bit is reset when a new command is started.
	TPC	BOOL	FALSE	Target Presence Changed TRUE = new tag within range of the reader The parameter is set to FALSE following successful execution of the next INVENTORY or INIT command.
	TP	BOOL	FALSE	Target Presence TRUE = there is a tag within the range of the reader
	UIN0	BOOL	FALSE	With this output parameter, the user can read an input of the RFID system according to the standard profile. With Siemens communication modules, the number of data carriers in the field is indicated here.
	UIN1	BOOL	FALSE	
	UIN2	BOOL	FALSE	
	UIN3	BOOL	FALSE	
	TRLEN	INT	0	Number of data elements received after successful execution of the command.
	ENO	BOOL	TRUE	Enable output

Ladder logic programming box	Parameters	Data type	Default	Meaning
	CMDREF	ARRAY[1..10] of CMD_STRUCT		Field that can accommodate 10 commands The commands are complex variables of the type CMD_STRUCT (for a detailed description, see section *Commands of the RFID standard profile*).
	TXREF	ARRAY[1..n] of BYTE		Reference to global memory area for send data. The memory area can be shared with other FB instances. The following values for n apply to the individual FBs: • n = 1024 (FB _PIB_1KB) • n = 16384 (FB _PIB_16KB) • n = 32768 (FB _PIB_32KB)
	RXREF	ARRAY[1..n] of BYTE		Reference to global memory area for receive data. The memory area can be shared with other FB instances. The following values for n apply to the individual FBs: • n = 1024 (FB _PIB_1KB) • n = 16384 (FB _PIB_16KB) • n = 32768 (FB _PIB_32KB)
[1)] Available soon				

See also

Warnings (Page 95)
Error messages (Page 80)

2.5 Data structures of the function block for RFID standard profile

Every time the function block for the RFID standard profile is called, you must initialize the parameters in accordance with the description. A UDT1 (PIB_COMMAND) is available for the CMDREF parameter with which the relevant variable can be created for every call in a Parameter_DB.

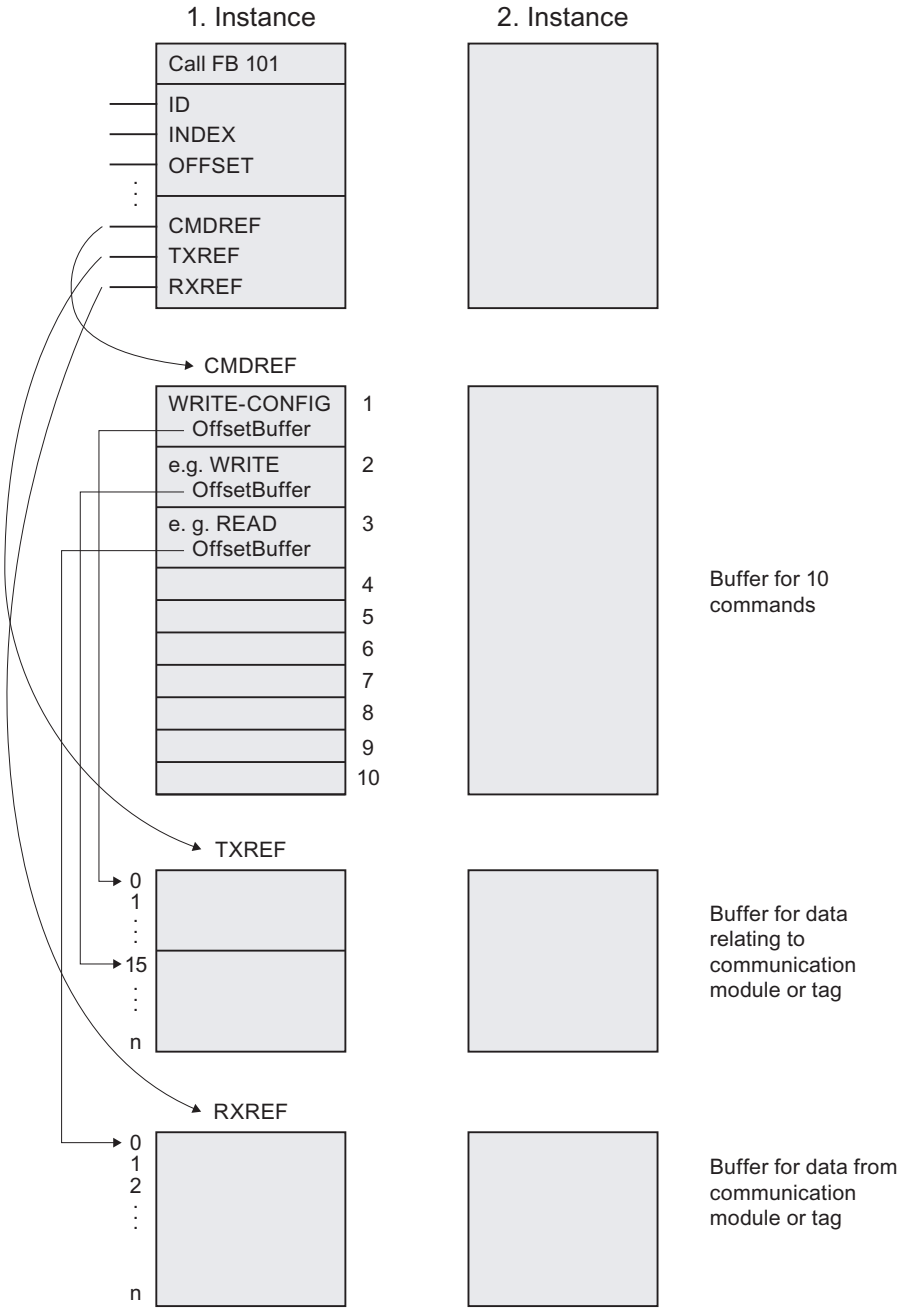


Figure 2-3 Overview of the data structure

2.6 Number of RFID channels which can be connected

Each RFID channel occupies 1 word in the input and output area of a SIMATIC S7. The maximum number of RFID modules supported by SIMATIC can be operated. The following table provides an overview.

Table 2- 3 Number of RFID channels

S7 CPU type[1]	315-2 DP		316-2 DP; 318-2 DP		416; 417; CP 443-5 Ext	
	Max. number of communication modules	Max. number of RFID channels	Max. number of communication modules	Max. number of RFID channels	Max. number of communication modules	Max. number of RFID channels
ASM 456, RF180C	64	128	123	246	123	246
RF170C (distributed via ET 200pro) [2]	64 x 9	1152	123 x 9	2214	123 x 9	2214

1) The CPU types specified here may not be complete since the range of CPUs and the associated functions are being continuously expanded.

2) Up to 9 RF170Cs can be plugged into an ET 200pro distributed I/O station.

2.7 Addressing of the RFID channels

Addressing over PROFIBUS or PROFINET

When addressing via PROFIBUS/PROFINET, any addresses for the RFID communication modules can be selected in HW Config. HW Config assigns a free address by default.

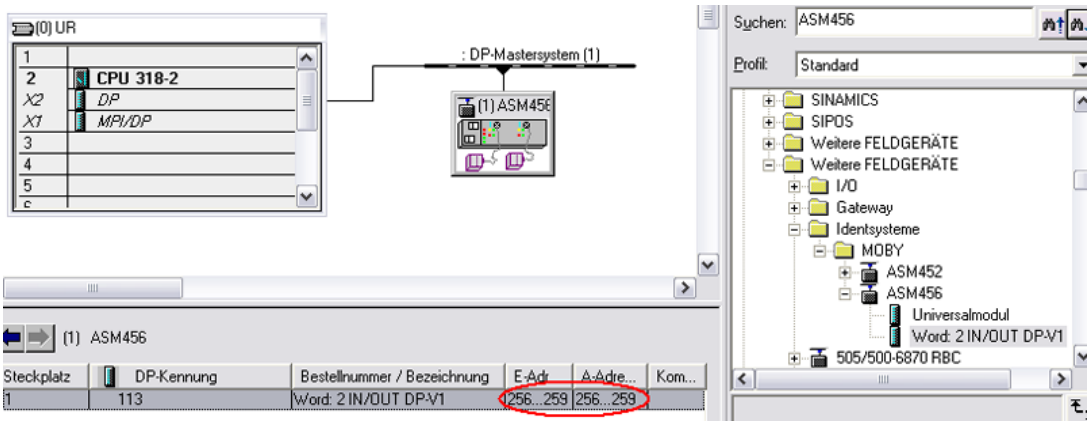


Figure 2-4 Example of automatic address generation

Assignment of the addresses

The unique I/O addresses of the communication modules from HW Config must be adopted when parameterizing the FB. It must be noted here that the address in HW Config is decimal and that it must be specified in hexadecimal when called.

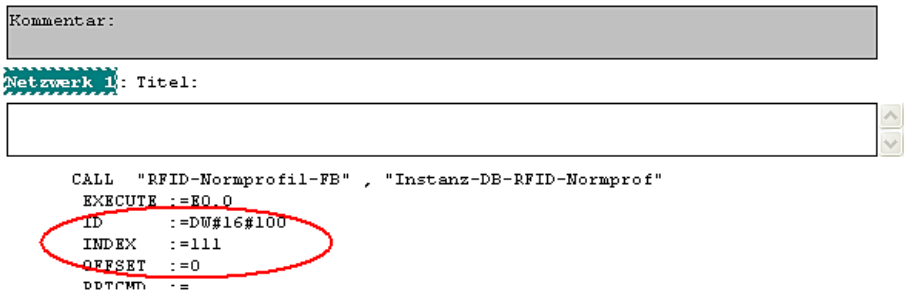


Figure 2-5 Address assignment

The RFID channel must be assigned uniquely in addition to the address (ID). For this, the INDEX (corresponding to the data record) and the OFFSET must be specified. The OFFSET is the address interval with respect to the basic address in bytes.

Note

The address of the communication module (I/O address) must be within the process image of the controller.

Parameter settings

3

3.1 Parameterizing the FB

Following call-up of the function block, you must initialize its parameters in accordance with the *configuring scheme*.
The figure below shows a simple parameterization example: Control and status values are assigned direct to inputs / outputs.

```
CALL  "PIB_1KB" , DB101
 EXECUTE :=E0.0                  //Befehlsstart
 ID      :=DW#16#100             //Basisadresse
 INDEX   :=111                   //festgelegter Datensatz für Kanal 1
 OFFSET  :=0                     //festgelegte Offset für Kanal 1
 RPTCMD  :=E0.1
 SRESET  :=E0.2
 INIT    :=E0.3
 UOUT0   :=E0.4
 UOUT1   :=E0.5
 UOUT2   :=E0.6
 UOUT3   :=E0.7
 RDGATE  :=E1.0
 CMDDIM  :=10
 CMDSEL  :=2                     //Zeiger auf den zweiten Befehl
 TXBUFLEN:=L#1024
 TXSTART :=L#1
 RXBUFLEN:=L#1024
 RXSTART :=L#1
 STATUS  :=AD0
 DONE    :=A4.0
 BUSY    :=A4.1
 ERROR   :=A4.2
 WARNING :=A4.3
 TPC     :=A4.4
 TP      :=A4.5
 UIN0    :=A4.6
 UIN1    :=A4.7
 UIN2    :=A5.0
 UIN3    :=A5.1
 TRLEN   :=AD6
 CMDREF  :="Command".Kanal1
 TXREF   :=DB2.txref
 RXREF   :=DB2.rxref
```

Figure 3-1 Calling the FB

The parameters CMDREF, TXREF and RXREF designate memory areas. The memory areas must be established in DBs before the call.

Note

If you work with several channels, you must ensure that for each channel, the function block is called with a separate instance DB.

For the command buffer CMDREF, you must create a parameter of the type "ARRAY [1..10] OF UDT1". The UDT 1 has the symbolic name "PIB_COMMAND".

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Kanal1	ARRAY[1..10]		
*38.0		"PIB_COMMAND"		
=380.0		END_STRUCT		

Figure 3-2 Structure of the UDT 1

The UDT structure corresponds to the required structure for a command.

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	CMD	BYTE	B#16#0	Erster Befehlsbuffer reserviert für INIT
+1.0	Config	BYTE	B#16#0	RF300
+2.0	OffsetBuffer	INT	0	
+4.0	UID	ARRAY[1..8]		
*1.0		BYTE		
+12.0	FileName	ARRAY[1..8]		
*1.0		BYTE		
+20.0	Offset	DINT	L#0	
+24.0	Length	INT	5	Datenlänge
+26.0	StartAddress	DINT	L#0	
+30.0	Attributes	BYTE	B#16#0	
+31.0	NextMode	BYTE	B#16#0	
+32.0	Timeout	INT	0	
+34.0	ObjectNumber	INT	0	
+36.0	FileType	WORD	W#16#0	
=38.0		END_STRUCT		

Figure 3-3 Structure for a command

3.2 Commands of the RFID standard profile

The chapter describes the commands and associated parameters that are supported by the function blocks PIB_1KB, PIB_16KB and PIB_32KB.

NOTICE

The command INIT always uses the first command in the command buffer CMDREF regardless of the value with which CMDSEL is initialized. For this reason, the first command must be parameterized for initialization. This command area must be set once and then never changed.

The command areas CMDSEL = 2 … 10 area available for freely usable commands.

3.2.1 Structure of the commands

Before you can start a command with EXECUTE or INIT, you need to define the command. The command buffer CMDREF has been created with the help of UDT 1 for simple definition of a command. There are 10 areas to you here in which commands can be parameterized. The parameter CMDSEL defines which command [1..10] is started with EXECUTE.

The table below lists all the parameters of the command structures. Not every command uses all parameters.

Table 3- 1 Structure of CMDREF[1..10]

Parameter	Data type	Value	Meaning
CMD	Byte	B#16#0	
Config	Byte	B#16#0	• Config = 1 → reset, no configuration data • Config = 2 → no reset, configuration data to be sent • Config = 3 → reset, configuration data to be sent
OffsetBuffer	INT	0	relative offset within the receive data buffer The parameter indicates the address within the memory area where the first byte of the received data must be saved. All subsequent bytes must be stored in ascending addresses.
UID[1]	Byte	B#16#0	Identification of a tag UID = 0: any tag The tag currently captured by the reader or the tag that comes next in the field is processed. UID ≠ 0: Multitag (e.g. MOBY U) UID of the tag that is to be accessed.
UID[2]	Byte	B#16#0	
UID[3]	Byte	B#16#0	
UID[4]	Byte	B#16#0	
UID[5]	Byte	B#16#0	
UID[6]	Byte	B#16#0	
UID[7]	Byte	B#16#0	
UID[8]	Byte	B#16#0	
FileName[1]	Byte	B#16#0	Name of the files that are to be accessed (create, delete, overwrite or read out) only relevant if filehandler mode is set Range of values: 20h to 7Eh
FileName[2]	Byte	B#16#0	
FileName[3]	Byte	B#16#0	
FileName[4]	Byte	B#16#0	
FileName[5]	Byte	B#16#0	
FileName[6]	Byte	B#16#0	
FileName[7]	Byte	B#16#0	
FileName[8]	Byte	B#16#0	
Offset	DINT	L#0	relative offset within the specified file that is to be accessed
Length	INT	0	Amount of data to be read/written in bytes
StartAddress	DINT	L#0	Physical starting address within one tag
Attributes	Byte	B#16#0	Valid values: • Unlimited read/write = bit 0,1 not set • Read only = bit 0 set • Single read = bit 1 set • Defined size = bit 2 set (the file length cannot be changed with a command)

Parameter	Data type	Value	Meaning
NextMode	Byte	B#16#0	Valid values: • NextMode = 1 (for the next command only a tag with a new UID is permitted, or the next tag to come into the field)
Timeout [1]	INT	0	Maximum time for a read operation before a timeout is generated If the value of the command parameter Timeout is set to "0", there is no time monitoring. The basic setting for this parameter is 10 ms.
ObjectNumber [1]	INT	0	Linking of an object with its barcode information The barcode reader can set this number in its result message frame. If the input parameter RPTCMD of the FB is active, ObjectNumber is incremented with each read operation. Value range of the parameter: 0…1023 After the value 1023 has been exceeded, the counting begins with "0".
FileType	WORD	W#16#0	On Siemens communication modules, the value 2020h must always be set here.
[1] only relevant for barcode readers			

3.2.2 Overview of commands

Commands of the RFID standard profile

The table below contains all commands that the RFID standard profile supports with the function blocks FB 101, FB 116 and FB 132. Different commands are available, depending on the connected reader and the settings in HW Config via the GSD file (MOBY_mode).

Table 3- 2 Overview of commands

Command	Command code		Parameters used	available in MOBY_mode	
	Hex	ASCII		Normal addressing	Filehandler
CLEAR	63	'c'	UID, FileName	–	–
CREATE	68	'h'	UID, FileName, length, attributes, FileType	–	x
DELETE	64	'd'	UID, FileName	–	x
DEV-STATUS	74	't'	Attributes, OffsetBuffer	x	x
FORMAT	66	'f'	OffsetBuffer, UID, Length	x	x
GET (Get_QREAD, Get_MOVE)	62	'b'	OffsetBuffer, Length	–	x
GET-ATTRIBUTE	6B	'k'	UID, FileName	–	–
GET-DIRECTORY	6D	'm'	OffsetBuffer, UID, FileType	–	x
INVENTORY	69	'i'	Attributes, OffsetBuffer	x	x
MEM-STATUS	73	's'	UID, attributes, OffsetBuffer	x	x
NEXT	6E	'n'	UID, NextMode	–	x
PHYSICAL-READ	70	'p'	OffsetBuffer, UID, length, StartAddress	x	x
PHYSICAL-WRITE	71	'q'	OffsetBuffer, UID, length, StartAddress	x	–
PUT (Put_QWRITE, Put_LOAD)	65	'e'	OffsetBuffer, Length	–	x
READ	72	'r'	OffsetBuffer, UID, FileName, offset, length	–	x
READ-BARCODE	76	'v'	OffsetBuffer, TimeOut, ObjectNumber	–	–
READ-CONFIG	61	'a'	OffsetBuffer	x	x
SET-ATTRIBUTE	6F	'o'	UID, FileName, attributes, FileType	–	x
UPDATE	75	'u'	OffsetBuffer, UID, FileName, length	–	x
WRITE	77	'w'	OffsetBuffer, UID, FileName, offset, length	–	x
WRITE-CONFIG	78	'x'	OffsetBuffer, length, config	x	x

3.2.3 Effect of the commands

The commands used take effect as follows:

- The input parameters INIT and SRESET interrupt command execution within the communication module.
- After sending a cyclic control message (INIT, SRESET) the subsequent change of the output parameter refers to the cyclic control message and not the command interrupted by the input parameter INIT or SRESET.
- The input parameter INIT resets communication between the RFID standard profile and the communication module. Following "hard" resetting of the communication module, FB 101 automatically transfers the command WRITE-CONFIG to the communication module. That's why it is absolutely necessary that the user stores the command WRITE-CONFIG in the first element of the command buffer CMDREF.
- The command WRITE-CONFIG resets all functions within the communication block, with the exception of the communication.
- The parameter SRESET interrupts a running command.

3.2.4 Command parameterization

Note

Not every command uses all structure elements.

The parameters you must initialize for individual commands are listed.

3.2.4.1 CLEAR

Description

The CLEAR command resets the content of a file. All contents are set to "16#00".

The command is not currently supported and results with every SIMATIC RFID system in an error message from the communication module.

Call

CMD = B#16#63

Parameters

The following parameters must be initialized for the CLEAR command:

Table 3- 3 Parameters of the CLEAR command

Parameters	Description
UID	Identification of a tag UID = 0: any tag The tag currently captured by the reader or the tag that comes next in the field is processed. UID ≠ 0: Multitag (e.g. MOBY U) UID of the tag that is to be accessed.
FileName	Name of the file whose content is to be reset Value range: 20h to 7Eh

Result

The content of the designated file has been reset (deleted).

3.2.4.2 CREATE

Description

The CREATE command creates a new file on a formatted tag.

Call

CMD = B#16#68

Parameters

The following parameters must be initialized for the CREATE command:

Table 3- 4 Parameters of the CREATE command

Parameters	Description
UID	Identification of a tag UID = 0: any tag The tag currently captured by the reader or the tag that comes next in the field is processed. UID ≠ 0: Multitag (e.g. MOBY U) UID of the tag that is to be accessed.
FileName	Name of the file whose content is to be created Value range: 20h to 7Eh
Length	Number of bytes to be reserved on the tag If the length "0" is specified here, the filehandler reserves a block in the file directory. If the length of the data is not known, a length does not necessarily have to be specified here. The filehandler manages the tag-memory dynamically. For this, bit 2 of the file attribute must be specified with "0".
Attributes	Valid values: • Unlimited read/write = bit 0,1 not set • Read only = bit 0 set • Write once = bit 1 set (new data can be attached) • Defined size = bit 2 set (the file length cannot be changed with a command)
FileType	On Siemens communication modules, the value 2020h must always be set here.

Result

A new file has been created on the tag.

3.2.4.3 DELETE

Description

The DELETE command deletes a file from the tag. The filename from the folder. All the data of the file are lost.

Note

A write-protected file cannot be deleted. Before deletion, file protection must be revoked with the command SET-ATTRIBUTE.

Call

CMD = B#16#64

Parameters

The following parameters must be initialized for the DELETE command:

Table 3- 5 Parameters of the DELETE command

Parameters	Description
UID	Identification of a tag UID = 0: any tag The tag currently captured by the reader or the tag that comes next in the field is processed. UID ≠ 0: Multitag (e.g. MOBY U) UID of the tag that is to be accessed.
FileName	Name of the file whose content is to be deleted Value range: 20h to 7Eh

Result

The designated file has been deleted on the tag.

3.2.4.4 DEV-STATUS

Description

The DEV-STATUS command is usesd to read out the status of a reader. The status data are stored in the receive data buffer. Status data are vendor-specific. The output parameter TRLEN of the FB indicates the number of received bytes.

Call

CMD = B#16#74

Parameters

The following parameters must be initialized for the DEV-STATUS command:

Table 3- 6 Parameters of the DEV-STATUS command

Parameters	Description
OffsetBuffer	Relative offset within the receive data buffer (RXBUF) The parameter indicates the address within the memory area where the first byte of the received data is saved. All subsequent bytes are stored in ascending addresses.
Attributes	Specification of the information class for data from the reader Valid values: • 16#01 → warning detail (vendor-specific details) * • 16#02 → error history (vendor-specific details) * • 16#03 → command history (vendor-specific details) * • 16#04 → channel-specific identification and maintenance information (I&M data) (data recording I&M0) * • 16#05 → channel-specific I&M data (data recording I&M1) * • 16#06 → channel-specific I&M data (data recording I&M2) * • 16#07 → channel-specific I&M data (data recording I&M3) * • 16#08 → channel-specific I&M data (data recording I&M4) * • 16#81 → Status of the reader • 16#84 → UID list (identified tags)
* Not currently supported on RFID systems	

Result

Meaning of the data on RF300/MOBY I/U/D

Table 3- 7 Representation of the data in the RXBUF for attributes = 81 (normal addressing and filehandler)

Offset	Name	Type	Comment
+0.0	status_info	BYTE	Reader/SLG status mode
+1.0	hardware	CHAR	Type of hardware
+2.0	hardware_version	WORD	HW version
+4.0	loader_version	WORD	Version of loader
+6.0	firmware	CHAR	Type of firmware
+7.0	firmware_version	WORD	FW Version
+9.0	driver	CHAR	Type of driver
+10.0	driver_version	WORD	Version of driver
+12.0	interface	BYTE	Interface (RS 232/RS 422)
+13.0	baud	BYTE	Baud rate
+14.0	reserved1	BYTE	Reserved
+15.0	reserved2	BYTE	Reserved
+16.0	reserved3	BYTE	Reserved
+17.0	distance_limiting_SLG	BYTE	Range limitation
+18.0	multitag_SLG	BYTE	Multitag reader / write/read device
+19.0	field_ON_control_SLG	BYTE	BERO mode
+20.0	field_ON_time_SLG	BYTE	BERO time
+21.0	sync_SLG	BYTE	Semaphore control (synchronization with reader / write/read device)
+22.0	status_ant	BYTE	Status of antenna
+23.0	stand_by	BYTE	Standby time after command execution
+24.0	MDS_control	BYTE	Presence

Table 3- 8 Representation of the data in the RXBUF for attributes = 84 (MOBY U)

Offset	Name	Type	Comment
+0.0	status_info	BYTE	Reader/SLG status mode
+1.0	number_MDS	BYTE	Range of values: 1 … 24
+2.0	UID	ARRAY [1...24]	Identified tag number
*4.0		DWORD	

3.2.4.5 FORMAT

Description

The FORMAT command initializes the tag. After formatting, the transponder is ready for use. The user must first store the parameters for the FORMAT command in the send data buffer.

FORMAT in the case of normal addressing:
The tag is completely deleted by being overwritten with a pre-defined value. The FORMAT command is not absolutely necessary in normal addressing.

FORMAT in the case of filehandler:
In filehandler mode, it is absolutely necessary to format the tags before startup. As well as deleting the data, the FORMAT command also creates a new file folder on the tag.

Call

CMD = B#16#66

Parameters

The following parameters must be initialized for the FORMAT command:

Table 3- 9 Parameters of the FORMAT command

Parameters	Description
OffsetBuffer	Relative offset within the send data buffer (TXBUF) The parameter indicates the address within the memory area where the first byte of the data to be sent is saved. All subsequent bytes are stored in ascending addresses.
UID	Identification of a tag UID = 0: any tag The tag currently captured by the reader or the tag that comes next in the field is processed. UID ≠ 0: Multitag (e.g. MOBY U) UID of the tag that is to be accessed.
Length	Volume of parameter data to be transferred.

Result

The parameter data to be transferred to the tag are stored in the send data buffer under TXBUF+[OffsetBuffer]. The tag has been formatted (initialized).

Structure of the parameter data in the send data buffer

The length of the parameter data depends on the mode that has been set in HW Config during configuration. On SIMATIC RFID systems, it is 0Fh or 15h bytes. The parameter data have the following structure:

In the case of normal addressing:

Table 3- 10 Structure of the data attachment for the FORMAT command in the case of normal addressing

1 … 8	9	10	11	12	13	14	15
00h	06h	03h	00h	INIT value	00h	MSB	LSB

Byte 1 … 8	Reserved for security code (must be assigned 0, since SIMATIC RFID has had no code previously)
Byte 9	Length of the subsequent data, here 6
Byte 10	Permanently set to 03h
Byte 11	Permanently set to 00h
Byte12	INIT value: The data area of the tag is overwritten with this value (hex format).
Byte 13	Permanently set to 00h
Byte 14	Memory size of the tag (end address + 1; high byte, hex format)
Byte 15	Memory size of the tag (end address + 1; low byte, hex format)

Table 3- 11 Memory sizes of the tags

Tag type			Memory size	Init duration normal	INIT duration with ECC
1 KB	MOBY I:	VMDS ASM 452	**05 00**	< 0.1 s	–
2 KB	MOBY I:	RAM	**08 00**	0.4 s	5 s
8 KB	MOBY I:	FRAM	**20 00**	0.8 s	20 s
8 KB	MOBY I:	EEPROM	**20 00**	18 s	54 s
32 KB	MOBY I:	RAM / FRAM	**80 00**	3 s	75 s
752 bytes	MOBY E:	EEPROM	**02 F0**	0.8 s	–
2 KB	MOBY U:	RAM*	**08 00**	approx. 1 s	–
32 KB	MOBY U:	RAM*	**80 00**	approx. 1.5 s	–
44 bytes	MOBY D:	MI-Code 1	**00 2C**		
112 bytes	MOBY D:	ISO I-Code SLI	**00 70**		
256 bytes	MOBY D:	ISO Tag-it HF-I	**01 00**		
1000 bytes	MOBY D:	ISO my-d	**04 00**		
64 bytes	MOBY D:	ISO ST LRI512	**00 40**		
20 bytes	RF300:	EEPROM	**00 14**	approx. 0.2 s	–
8 KB	RF300:	FRAM	**20 00**	0.3 s	–
32 KB	RF300:	FRAM	**80 00**	1.2 s	–
64 KB	RF300:	FRAM	**FF 00**	2.4 s	–
*) The OTP memory is not initialized with this command.					

Example of a 32K data carrier:

00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 06h, 03h, 00h, 00h, 00h, 80h, 00h

In the case of filehandler:

Table 3- 12 Structure of the data attachment for the FORMAT command in the case filehandler

1 … 8	9	10 … 17	18	19	20	21
00h	0Ch	Volume	MDS type	option	00h	INIT value

Byte 1 … 8	Reserved for security code (must be assigned 0, since SIMATIC RFID has had no code previously)	
Byte 9	Length of the subsequent data, here 12	
Byte 10 … 17	Name of the tag, 8 ASCII characters (value range: 20h to 7Eh)	
Byte 18	Type of tag	
	2KB → 04h 8KB → 05h 32KB → 06h	2KB → 84h (with ECC; MOBY I only) 8KB → 85h (with ECC; MOBY I only) 32KB → 86h (with ECC; MOBY I only)
Byte 19	Option The option is bit-coded. Several bits can be used in accordance with the following table.	
	0000xxx0B 0000xxx1B	Always read directory + FAT Checksum mechanism switched on
	0000xx0xB 0000xx1xB	Entire tag is deleted Directory + FAT are retained, the user data area is deleted[1]
	0000x0xxB 0000x1xxB	Entire tag is deleted Directory + FAT are deleted, the user data area is retained
	00000xxxB 00001xxxB	normal The counters for write cycles or ECC correctons are retained. The counters are also retained even if there are errors in the system area (e.g. ECC error)
	1) The directory + FAT in the filehandler is always used for formatting.	
Byte 20	Permanently set to 00h	
Byte 21	INIT value: The data area of the tag is overwritten with this value (hex format).	

Example of a 32K data carrier without ECC:

00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 0Ch, 20h, 20h, 20h, 20h, 20h, 20h, 20h, 20h, 06h, 01h, 00h, 00h

3.2.4.6 GET

Description

With the GET command in SIMATIC RFID systems, additional commands are transmitted that are not defined in the "PIB standard". To this end, a corresponding data structure is defined in the send data buffer for each extended command.

Received data is also structured and saved in the receive data buffer (RXBUF). The output parameter TRLEN of the function block indicates the number of received bytes.

The following additional commands are possible via GET:

- Get_QREAD
  - Reading of several files (max. 12) or reading of an entire tag with one command.
- Get_MOVE
  - Saving the system data from the filehandler (Directory, FAT, checksum). The directory, FAT and checksum of the corresponding SLG station are saved in a data block.
  - After line voltage failure, this information can be directly transmitted to the ASM (filehandler) with the aid of the LOAD command. The checksum of the incoming tag agrees with the filehandler-internal checksum again. Work can immediately resume at full transport speed.
  - It is recommended to set up the MOVE command after a corresponding directory change (checksum change: Display by means of output parameter (STATUS bit 6 "tag with new directory"). Saving the system data with MOVE is only recommended if work is generally done with LOVE/LOAD.

Call

CMD = B#16#62

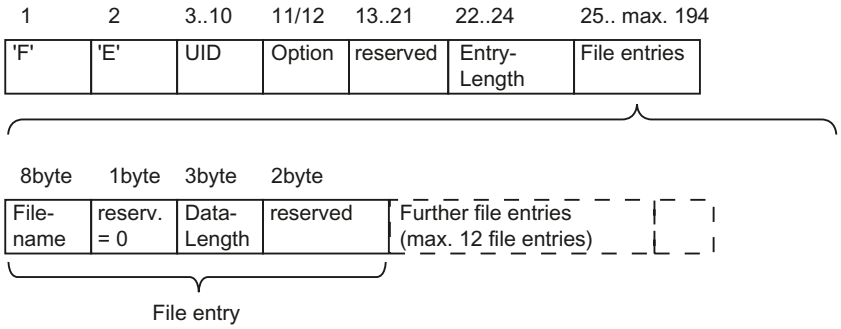
Parameters

The following parameters must be initialized for the GET command:

Table 3- 13 Parameters of the GET command

Parameters	Description
OffsetBuffer	Relative offset within the receive data buffer (RXBUF) The parameter indicates the address within the memory area where the first byte of the received data is saved. All subsequent bytes are stored in ascending addresses. No offset can be set for the send data buffer (TXBUF).
Length	The amount of the data in the send data buffer (TXBUF) to be sent to the communication module - as described below. The maximum size is 194 bytes.

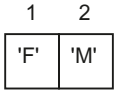
Get_QREAD



F	ASCII character F as the ID for a filehandler command	
E	ASCII character E as the command code for Get_QREAD	
UID	Unique identifier or tag number. With 0, any tags are processed. If no UID is required, 0 must be entered at this position.	
Option	0000h	No option. The data contains the files to be read.
	0002h	All files with a file length > 0 are read from the MDS. The data length must be pre-assigned with 0.
	0004h	If the actual file length is longer than the file length reserved in the data, no error message occurs.
	0008h	The user data contains the files that are not to be read; i.e. all files of the data storage except for the files specified in the command are read. The "data length" parameter in the file entry has no meaning for this option.
	0010h	The read file data is aligned word by word and are then directly appended to each other, i.e. when the file length is an odd number, a 20 hex skip byte is added. The "Skip" parameter then has the value 0001 in the acknowledgement. With this option, the data length must always be set to "FFFFFF hex" in the file entries.
	0022h	All of the files of the MDS are read. In contrast to the 0002h option, empty files are also read here (file length = 0). With this option, a data carrier can be completely read and then rewritten to a different MDS using QUEUEWRITE.
	The options are interpreted bit-by-bit. With this, several options can be set simultaneously (e.g. 0Ch, 12h, 18h, etc.).	
Reserved	Reserved. To be assigned the value 0.	
Entry length	Total length of the following entries (from byte 25 to the end of the data).	
File Entries	Area in which up to 12 file entries must be entered.	
Filename	File name (8 ASCII characters 0x20 ... 0x7E)	
reserv.	Reserved. To be assigned the value 0.	
Data-Length	Length of the data to be read for the specified file.	
	0xFFFFFF	complete file
	0x00XXXX	Reserved length in the receive buffer. If the file contains less data than was specified, it is supplemented with skip bytes. See result
reserved	Reserved. To be assigned the value 0.	

Figure 3-4 Command structure Get_QREAD

Get_MOVE



F	ASCII character F as the ID for a filehandler command
M	ASCII character M as the command code for Get_MOVE

Figure 3-5 Command structure Get_MOVE

The command Get_MOVE cannot be used with MOBY U.

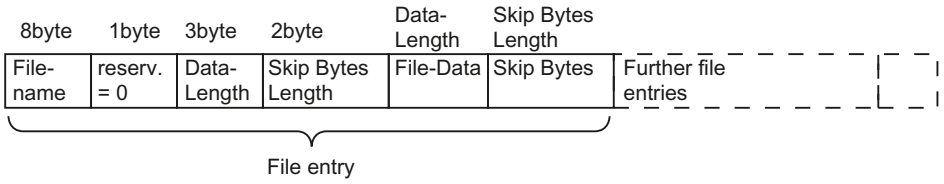
Result

The data transferred from the communication module are stored in the receive data buffer under TXBUF+[OffsetBuffer].

Data structure of the reply data of the commands Get_QREAD and Get_MOVE

The received data has the following data structure:

Get_QREAD



Filename	File name (8 ASCII characters 0x20 ... 0x7E)
reserv.	0
Data-Length	Length of the read data for the corresponding file.
Skip Bytes Length	The skip specifies the number of bytes after which the next data entry follows the end of the previous file entry. Skip is set by the filehandler. The user can control the command via the parameter "Option".
File-Data	Read file data. The length of the data is displayed via "data length".
Skip-Bytes	Area of the skip bytes, the length of which is indicated by "Skip Bytes Length". Skip bytes always have the value 0x20.

Figure 3-6 Data structure of the reply data of the Get_QREAD command

Get_MOVE

1/2	3..... Data
Length	DIR + FAT

Length Length of DIR + FAT + 2 (including the 2 bytes for the length information). The maximum data length to be expected occurs for a 32 KB tag (1335 bytes).

DIR + FAT Data to be saved and which is to be re-sent using the LOAD command. This data must not be changed.

Figure 3-7 Data structure of the reply data of the Get_MOVE command

3.2.4.7 GET-ATTRIBUTE

Description

The GET-ATTRIBUTE command reads out attributes belonging to a file. The attributes (attributes and file type) are stored in the directory.

The command is not currently supported and results with every SIMATIC RFID system in an error message from the communication module.

Call

CMD = B#16#6B

Parameters

The following parameters must be initialized for the GET-ATTRIBUTE command:

Table 3- 14 Parameters of the GET-ATTRIBUTE command

Parameters	Description
UID	Identification of a tag UID = 0: any tag The tag currently captured by the reader or the tag that comes next in the field is processed. UID ≠ 0: Multitag (e.g. MOBY U) UID of the tag that is to be accessed.
FileName	Name of the file whose attributes are to be read out Value range: 20h to 7Eh

Result

The reply data of this command are stored in a structure within the receive data buffer (see the figure below).

```
TYPE
    Attribute_STRUCT
        STRUCT
               Attributes : BYTE;
               FileType : WORD;
        END_STRUCT;
END_TYPE
```

Figure 3-8 Structure of the reply data of the GET-ATTRIBUTE command

The parameters of the reply data are described in the table below.

Table 3- 15 Structure of the reply data of the GET-ATTRIBUTE command

Parameters	Description
FileName	File name
Attributes	Valid values: • Unlimited read/write = bit 0,1 not set • Read only = bit 0 set • Single read = bit 1 set • Defined size = bit 2 set (the file length cannot be changed by any command)
FileType	Classification / grouping of the files according to user-specific criteria. Files with the same characteristics are assigned to the same group. These files are of the file type indicated by the value of the command parameter FileType. If the parameter is not used, all bytes must be set to "16#20".

3.2.4.8 GET-DIRECTORY

Description

The GET-DIRECTORY command reads out the directory of the tag. File names and associated attributes are passed.

Call

CMD = B#16#6D

Parameters

The following parameters must be initialized for the GET-DIRECTORY command:

Table 3- 16 Parameters of the GET-DIRECTORY command

Parameters	Description
OffsetBuffer	Relative offset within the receive data buffer (RXBUF) The parameter indicates the address within the memory area where the first byte of the received data is saved. All subsequent bytes are stored in ascending addresses.
UID	Identification of a tag UID = 0: any tag The tag currently captured by the reader or the tag that comes next in the field is processed. UID ≠ 0: Multitag (e.g. MOBY U) UID of the tag that is to be accessed.
FileType	On Siemens communication modules, the value 2020h must always be set here.

Result

The reply data of this command are stored in a structure within the receive data buffer (see the figure below).

```
TYPE
    DIRELEMENTS_STRUCT
        STRUCT
                FileName : ARRAY [1..8] OF BYTE;
                UsedLength : DINT;
                Attributes : BYTE;
                FileLength : DINT;
                FileType : WORD;
        END_STRUCT;
END_TYPE

TYPE
    DIRLIST_STRUCT
        STRUCT
                UID1 : ARRAY [1..8] OF BYTE;
                TagName : ARRAY [1..8] OF BYTE;
                FreeUserMem : DINT;
                CheckSum : WORD;
                FileCount : INT;
                FileList : ARRAY [1..FileCount] OF DIRELEMENTS_STRUCT;
        END_STRUCT;
END_TYPE
```

Figure 3-9 Structure of the reply data of the GET-DIRECTORY command

The parameters of the reply data are described in the table below.

Table 3- 17 Parameters of the reply data of the GET-DIRECTORY command

Parameters	Description
FileName	File name
UsedLength	Memory areas to be occupied by the file
Attributes	Valid values: • Unlimited read/write = bit 0,1 not set • Read only = bit 0 set • Single read = bit 1 set • Defined size = bit 2 set (the file length cannot be changed by any command)
FileLength	Maximum file length In the case of a dynamic file system (such as the Siemens filehandler) FileLength and UsedLength are identical. If FileLength = 16#00, the file has been deleted.
FileType	2020h (not used)
UID1	Identification of individual tag UID1 = 0 indicates that the tag does not contain a UID. If UID = 0 is transferred to the tag within the command, UID1 supplies the UID of the present tag.
TagName	This parameter indicates the name of the tag defined by the FORMAT command.
FreeUserMem	This parameter indicates the unused or unreserved memory on the tag.

Parameters	Description
CheckSum	Checksum of the directory on the tag If not used, all bytes are set to "16#00".
FileCount	Number of directory entries that are available on the tag and supplied by the GET-DIRECTORY command.
FileList	Structure of the directory entries This structure is stored in the result memory the same number of times as there are files on the tag. This corresponds to the contents of FileCount

3.2.4.9 INVENTORY

Description

The INVENTORY command is used to request a list of all currently accessible tags within the antenna range.
It is possible to transfer additional vendor-specific information. The effect of this is that along with UID, a limited volume of user data is simultaneously read from the tagsTags (not on SIMATIC RFID systems).

After execution of the INVENTORY command, the output parameter TPC on FB 101 is reset.

Call

CMD = B#16#69

Parameters

The following parameters must be initialized for the INVENTORY command:

Table 3- 18 Parameters of the INVENTORY command

Parameters	Description
OffsetBuffer	Relative offset within the receive data buffer (RXBUF) The parameter indicates the address within the memory area where the first byte of the received data is saved. All subsequent bytes are stored in ascending addresses.
Attributes	Specification of the information to be read Valid values: • 16#00 → all UIDs are read

Result

The reply data of this command are stored in a structure within the receive data buffer (see the figure below).

```
TYPE
        ObjectNumber : INT;
        ObjectLength : INT;
END_TYPE


TYPE
        ARRAY [1..ObjectNumber]
          UID_STRUCT STRUCT
               UID1 : ARRAY [1..8] OF BYTE;
               Data : ARRAY [1..(ObjectLength-8)] OF BYTE;
          END_STRUCT;
END_TYPE
```

Figure 3-10 Structure of the reply data of the INVENTORY command

The parameters of the reply data are described in the table below.

Table 3- 19 Parameters of the reply data of the INVENTORY command

Parameters	Description
ObjectNumber	Number of tags transferred within the acknowledgement.
ObjectLength	Number of bytes linked by an individual tag (size of the UID and additional data) In the case of Attributes = 16#00, ObjectLength = 8 is set, provided a UID is available.
UIDList	Data field with elements of type UID_Struct If Attributes = 16#00, only UIDs are stored, and no vendor-specific information is stored.

3.2.4.10 MEM-STATUS

Description

The MEM-STATUS command is used to read out the status of a tag (battery status, memory, tag type, available capacity). The status data is stored in the receive data buffer. Status data are vendor-specific. The output parameter TRLEN of the FB indicates the number of received bytes.

Call

CMD = B#16#73

Parameters

The following parameters must be initialized for the MEM-STATUS command:

Table 3- 20 Parameters of the MEM-STATUS command

Parameters	Description
OffsetBuffer	Relative offset within the receive data buffer (RXBUF) The parameter indicates the address within the memory area where the first byte of the received data is saved. All subsequent bytes are stored in ascending addresses.
UID	Identification of a tag UID = 0: any tag The tag currently captured by the reader or the tag that comes next in the field is processed. UID ≠ 0: Multitag (e.g. MOBY U) UID of the tag that is to be accessed.
Attributes	This parameter specifies the information class to be read out from the tag: • 16#04 = physical information about the tag • 16#05 = information about the file system on the tag • 16#82 = statistical information about the tag

Result

Meaning of the data in normal addressing

Table 3- 21 Representation of the data in the RXBUF for attributes = 04 (RF300)

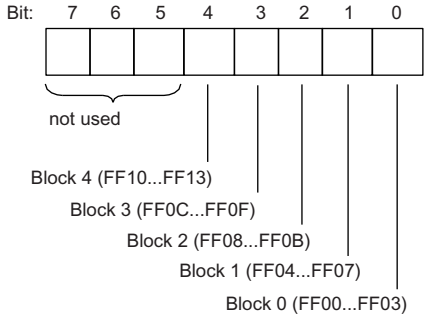
Offset	Name	Type	Comment
+0.0	status_info	BYTE	Tag status mode
+1.0	UID	ARRAY[1...8] BYTE	Tag number (unique identifier)
+9.0	MDS_type	BYTE	Tag type 01 = Tag without FRAM 02 = Tag with FRAM 8KB 03 = Tag with FRAM 32KB 04 = Tag with FRAM 64KB
+10.0	Lock_state	BYTE	EEPROM write protection status  Write protection status: 0 = block not protected (r/w) 1 = block protected (ro)
+11.0 to +16.0	res.	BYTE	Reserved

Table 3- 22 Representation of the data in the RXBUF for attributes = 82 (RF300)

Offset	Name	Type	Comment
+0.0	status_info	BYTE	Tag status mode
+1.0	UID	ARRAY[1...8] BYTE	Tag number (unique identifier)
+9.0	LFD	BYTE	Relationship between power flow density limit and actual measured value
+10.0	FZP	BYTE	Error counter, passive (errors during idle time)
+11.0	FZP	BYTE	Error counter, active (errors during communication)
+12.0	ANWZ	BYTE	Presence counter
+13.0 to +16.0	res.	BYTE	Reserved

Meaning of the data in the case of filehandler (MOBY I/U)

Table 3- 23 Representation of the data in the RXBUF for attributes = 05

Offset	Name	Type	Comment		
+0.0	UID	DINT	MDS number		
+4.0	MDS_name	ARRAY [1..8]	Name of the MDS transfered with FORMAT		
*1.0		CHAR			
+12.0	MDS_type_FH	BYTE	Logical format ID of the MDS (filehandler format)		
			without ECC	with ECC	Memory size
			04	84	2 KB
			05	85	8 KB
			06	86	32 KB
+13.0	MDS_capacity	DINT	Memory size of the MDS in bytes		
+17.0	Free_capacity	DINT	Free memory size of the MDS in bytes		
+21.0	Free_directory	INT	Free directory entries on the MDS		
+23.0	ANZ_res0	BOOL			
+23.1	ANZ_res1	BOOL			
+23.2	Battery_low	BOOL	Set: MDS or RAM battery must be replaced (MOBY I only)		
+23.3	LR_bat	BOOL	Set: Communication battery of the MDS 507 must be replaced (MOBY I only)		
+23.4	ANZ_res4	BOOL			
+23.5	ANZ_res5	BOOL			
+23.6	ANZ_res6	BOOL			
+23.7	ANZ_MDS_covered	BOOL	Permanently set to 0		
+24.0	Cycles_write	DINT	Number of SLG (read/write device) stations that have processed this MDS since the first formatting		
+28.0	ECC_corrections	BYTE	Number of automatically executed ECC corrections since the first formatting of the MDS		

3.2.4.11 NEXT

Description

The NEXT command concludes operations on a tag. The subsequent command is not executed until the next tag is detected/displayed.

Call

CMD = B#16#6E

Parameters

The following parameters must be initialized for the NEXT command:

Table 3- 24 Parameters of the NEXT command

Parameters	Description
UID	Identification of a tag UID = 0: any tag The tag currently captured by the reader or the tag that comes next in the field is processed. UID ≠ 0: Multitag (e.g. MOBY U) UID of the tag that is to be accessed.
NextMode	Valid values: • NextMode = 1 (for the next command only a tag with a new UID is permitted, or the next tag to come into the field)

Result

The addressed tag is processed in accordance with the set NextMode.

3.2.4.12 PHYSICAL-READ

Description

The PHYSICAL-READ command reads out data from a tag by using the physical start address and the length of the data to be read. The output parameter TRLEN of the FB indicates the number of received bytes.

Call

CMD = B#16#70

Parameters

The following parameters must be initialized for the PHYSICAL-READ command:

Table 3- 25 Parameters of the PHYSICAL-READ command

Parameters	Description
OffsetBuffer	Relative offset within the receive data buffer (RXBUF) The parameter indicates the address within the memory area where the first byte of the read data is saved. All subsequent bytes are stored in ascending addresses.
UID	Identification of a tag UID = 0: any tag The tag currently captured by the reader or the tag that comes next in the field is processed. UID ≠ 0: Multitag (e.g. MOBY U) UID of the tag that is to be accessed.
Length	Number of bytes to be read.
StartAddress	Physical starting address within one tag

The permissible values for start address and length depend on the RFID system used and from the memory size of the tag used. You can find the permissible value ranges in the Section *Application examples > Processing of tags*.

Result

The data from the tag are stored in the receive data buffer under RXBUF+[OffsetBuffer]. The volume of read data is contained in the parameter TRLEN.

3.2.4.13 PHYSICAL-WRITE

Description

The PHYSICAL-WRITE command writes data to a tag. It uses the physical starting address and the length of the data to be written.

Call

CMD = B#16#71

Parameters

The following parameters must be initialized for the PHYSICAL-WRITE command:

Table 3- 26 Parameters of the PHYSICAL-WRITE command

Parameters	Description
OffsetBuffer	Relative offset within the send data buffer (TXBUF) The parameter indicates the address within the memory area where the first byte of the data to be sent is saved. All subsequent bytes are stored in ascending addresses.
UID	Identification of a tag UID = 0: any tag The tag currently captured by the reader or the tag that comes next in the field is processed. UID ≠ 0: Multitag (e.g. MOBY U) UID of the tag that is to be accessed.
Length	Number of bytes to be written.
StartAddress	Physical starting address within one tag

The permissible values for start address and length depend on the RFID system used and from the memory size of the tag used. You can find the permissible value ranges in the Section *Application examples > Processing of tags*.

Result

The data to be transferred to the tag are stored in the send data buffer under TXBUF+[OffsetBuffer].

3.2.4.14 PUT

Description

With the PUT command in SIMATIC RFID systems, additional commands are transmitted that are not defined in the "PIB standard" and which do not return any receive data. To this end, a corresponding data structure is defined in the send data buffer for each extended command.

The following additional commands are possible via PUT:

- Put_QWRITE
  - Setting up a complete tag. With the aid of this command, it is possible to very quickly save individual files and the corresponding file contents and file attributes in the data memory. The individual file entries (consisting of filename, attribute, file length and data) are sequentially processed by the filehandler. This means that a file is first created in the filehandler, then data is written to it and it is provided with an attribute. Then the next file entry is processed in the same way. This function can only be executed under the prerequisite, that the addressed MDS has been formatted without errors.
- Put_LOAD
  - Transfer system data to the communication module. This command can be set by the user in order to send the data that was saved with MOVE to the filehandler. After a line voltage failure, the LOAD command must be set immediately after the RESET command (only if LOAD/MOVE are generally used in the work).

Call

CMD = B#16#65

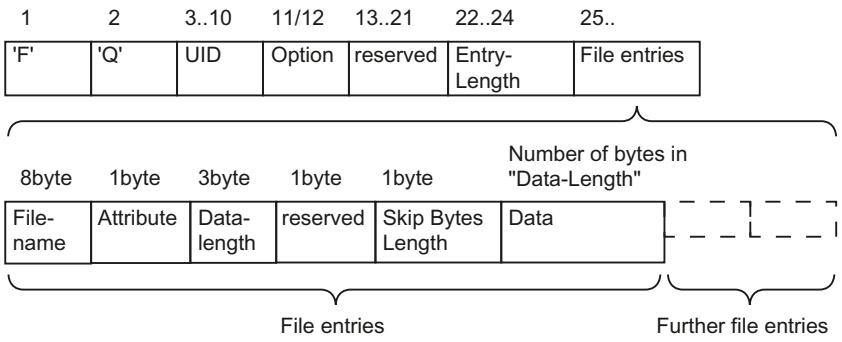
Parameters

The following parameters must be initialized for the PUT command:

Table 3- 27 Parameters of the PUT command

Parameters	Description
OffsetBuffer	Relative offset within the send data buffer (TXBUF) The parameter indicates the address within the memory area where the first byte of the data to be sent is saved. All subsequent bytes are stored in ascending addresses.
Length	The amount of the data in the send data buffer (TXBUF) to be sent to the communication module - as described below.

Put_QWRITE



F — ASCII character F as the ID for a filehandler command

Q — ASCII character Q as the command code for Put_QWRITE

UID — Unique Identifier or tag number. With 0, any tags are processed. If no UID is required, 0 must be entered at this position.

Option — If the data storage is already formatted and it contains an existing file structure, the following applies:

- 0000h — Default: The file structure is overwritten or deleted via the QUEUE-WRITE command.
- 0001h — The QUEUE-WRITE command appends the new data entries to the existing MDS file structure. This includes a check of whether the file to be created already exists.

reserved — Reserved. To be assigned the value 0

Entry length — Total length of the following entries (from byte 25 to the end of the data).

File Entries — Area in which file entries are to be put.

File entries

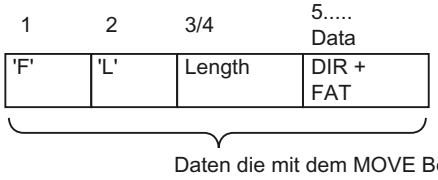
- Filename — File name (8 ASCII characters 0x20 ... 0x7E).
- Attributes — Attributes for this file. The attribute is only set after the data is written. The following values are valid (see also SET_ATTRIBUTE command):
  - Unlimited read/write = bit 0,1 not set
  - Read only = bit 0 set
  - Single read = bit 1 set
  - Defined size = bit 2 set (the file length cannot be changed with an command)
- Data-Length — Length of the data to be written for the specified file
- reserved — To be assigned the value 0
- Skip Bytes Length — The skip specifies the number of bytes after which the next data entry follows the end of the previous file entry. Valid value range: 0 to 255.

  Example Skip = 0:

  The filename of the next file entry must immediately follow the last valid data byte. For an uneven number of data, Skip = 1 can be set. With this, the next filename begins again in the data word on the left.

Figure 3-11 Command structure Put_QWRITE

Put_LOAD



F	ASCII character F as the ID for a filehandler command
L	ASCII character L as the command code for Put_LOAD
Length	Length of DIR + FAT + 2 (including the 2 bytes for the length information)
DIR+FAT	Saved data that has been received with the MOVE command. These data must not be changed.

Figure 3-12 Command structure Put_LOAD

The command Put_LOAD cannot be used with MOBY U.

Result

Put_QWRITE

The data saved in the send data buffer under TXBUF+[OffsetBuffer] has been correspondingly written on the tag.

Put_LOAD

The data saved in the send data buffer under TXBUF+[OffsetBuffer] has been transferred to the communication module.

3.2.4.15 READ

Description

The READ command reads data of a file (on RFID systems in accordance with the standard profile) from the tag. After successful execution of the command, the data are saved in the receive data buffer. The output parameter TRLEN of the FB indicates the number of received bytes.

Call

CMD = B#16#72

Parameters

The following parameters must be initialized for the READ command:

Table 3- 28 Parameters of the READ command

Parameters	Description
OffsetBuffer	Relative offset within the receive data buffer (RXBUF) The parameter indicates the address within the memory area where the first byte of the received data is saved. All subsequent bytes are stored in ascending addresses.
UID	Identification of a tag UID = 0: any tag The tag currently captured by the reader or the tag that comes next in the field is processed. UID ≠ 0: Multitag (e.g. MOBY U) UID of the tag that is to be accessed.
FileName	Name of the file whose data are to be read Value range: 20h to 7Eh
Offset	Relative offset within the specified file to be read from. 0 = starting with the first byte
Length	Number of bytes to be read –1 = read entire file

Result

The user data are stored in the receive data buffer under RXBUF+[OffsetBuffer]. The volume of read data is contained in the parameter TRLEN.

3.2.4.16 READ-BARCODE

Description

The READ-BARCODE command reads barcode data.

The command is not currently supported and results with every SIMATIC RFID system in an error message from the communication module.

Call

CMD = B#16#76

Parameters

The following parameters must be initialized for the READ-BARCODE command:

Table 3- 29 Parameters of the READ-BARCODE command

Parameters	Description
OffsetBuffer	Relative offset within the receive data buffer (RXBUF) The parameter indicates the address within the memory area where the first byte of the received data is saved. All subsequent bytes are stored in ascending addresses.
Timeout	Maximum time for a read operation before a timeout is generated If the value of the command parameter Timeout is set to "0", there is no time monitoring. The basic setting for this parameter is 10 ms.
ObjectNumber	Linking of an object with its barcode information The barcode reader can set this number in its result message frame. If the input parameter RPTCMD of the FB is active, ObjectNumber is incremented with each read operation. Value range of the parameter: 0…1023 After the value 1023 has been exceeded, the counting begins with "0".

Result

The read result is stored in the receive data buffer under RXBUF+[OffsetBuffer].

3.2.4.17 READ-CONFIG

Description

The READ-CONFIG command is used to read configuration data from the communication module. Thre receive data buffer is used as an area for configuration data. Configuration data are vendor-specific. The output parameter TRLEN of the FB indicates the number of received bytes.

Call

CMD = B#16#61

Parameters

The following parameters must be initialized for the READ-CONFIG command:

Table 3- 30 Parameters of the READ-CONFIG command

Parameters	Description
OffsetBuffer	Relative offset within the receive data buffer (RXBUF) The parameter indicates the address within the memory area where the first byte of the received data is saved. All subsequent bytes are stored in ascending addresses.

Result

The configuration data of the communication module are stored in the receive data buffer under RXBUF+[OffsetBuffer]. The volume of read data is contained in the parameter TRLEN.

3.2.4.18 SET-ATTRIBUTE

Description

The SET-ATTRIBUTE command sets/modifies the attributes or access rights belonging to a file. The attributes are stored in the file directory.

Call

CMD = B#16#6F

Parameters

The following parameters must be initialized for the SET-ATTRIBUTE command:

Table 3- 31 Parameters of the SET-ATTRIBUTE command

Parameters	Description
UID	Identification of a tag UID = 0: any tag The tag currently captured by the reader or the tag that comes next in the field is processed. UID ≠ 0: Multitag (e.g. MOBY U) UID of the tag that is to be accessed.
FileName	Name of the file whose attributes are to be modified Value range: 20h to 7Eh
Attributes	Valid values: • Unlimited read/write = bit 0.1 not set • Read only = bit 0 set • Single read = bit 1 set • Defined size = bit 2 set (the file length cannot be changed wyith an command)
FileType	On Siemens communication modules, the value 2020h must always be set here.

Result

The attributes of the designated file have been set / modified.

3.2.4.19 UPDATE

Description

The UPDATE command writes data to a file located on the tag. The file length is updated exactly to the number of written bytes. This command always refers to the entire file.

Call

CMD = B#16#75

Parameters

The following parameters must be initialized for the UPDATE command:

Table 3- 32 Parameters of the UPDATE command

Parameters	Description
OffsetBuffer	Relative offset within the send data buffer (TXBUF) The parameter indicates the address within the memory area where the first byte of the data to be written is saved. All subsequent bytes are stored in ascending addresses.
UID	Identification of a tag UID = 0: any tag The tag currently captured by the reader or the tag that comes next in the field is processed. UID ≠ 0: Multitag (e.g. MOBY U) UID of the tag that is to be accessed.
FileName	Name of the file whose content is to be updated Value range: 20h to 7Eh
Length	Number of bytes to be written.

Result

The data to be transferred to the tag are stored in the send data buffer under TXBUF+[OffsetBuffer]. The data have been written to the designated file on the tag.

3.2.4.20 WRITE

Description

The WRITE command writes data to a file located on the tag.
All data on the tag not overwritten are retained unchanged. The file length is oriented around the written data. It increases automatically when new data are written beyond the old area (only if bit 2 of the file attribute is not set).

Call

CMD = B#16#77

Parameters

The following parameters must be initialized for the WRITE command:

Table 3- 33 Parameters of the WRITE command

Parameters	Description
OffsetBuffer	Relative offset within the send data buffer (TXBUF) The parameter indicates the address within the memory area where the first byte of the data to be sent is saved. All subsequent bytes are stored in ascending addresses.
UID	Identification of a tag UID = 0: any tag The tag currently captured by the reader or the tag that comes next in the field is processed. UID ≠ 0: Multitag (e.g. MOBY U) UID of the tag that is to be accessed.
FileName	Name of the file that is to be written to value range: 20h to 7Eh
Offset	Relative offset within the specified file to be written to –1 = attach new data to the existing data
Length	Number of bytes to be written.

Result

The data to be transferred to the tag must be stored in the send data buffer under TXBUF+[OffsetBuffer] before the start of the command. No data are transferred to FB 101.

3.2.4.21 WRITE-CONFIG (also used for INIT)

Description

The WRITE-CONFIG command is used to parameterize or re-parameterize the operation of the communication module. It is possible to send new parameters (configuration data) to the relevant channel of the communication module. A running command is interrupted dependent on the Config parameter. The send data buffer is used as an area for configuration data. Configuration data are vendor-specific. Normally, the WRITE-CONFIG command is automatically executed by the FB during INIT. The WRITE-CONFIG command can be optionally started with the EXECUTE parameter. In this case, a user command cannot be interrupted.

Note

The WRITE-CONFIG command must be in the first position in the command buffer if it is to be used for the INIT command.

Call

CMD = B#16#78

Parameters

The following parameters must be initialized for the WRITE-CONFIG command:

Table 3- 34 Parameters of the WRITE-CONFIG command

Parameters	Description
Config	• Config = 1 → reset, no configuration data • Config = 2 → no reset, configuration data to be sent * • Config = 3 → reset, configuration data to be sent
Offset	Relative offset within the send data buffer (TXREF) The parameter indicates the address within the memory area where the first byte of the data to be sent is saved. All subsequent bytes are stored in ascending addresses.
Length	Volume of parameterization data in bytes
* not with RF300/MOBY D/U	

Result

The communication module is reset or parameterized in accordance with the setting ot the Config parameter.

Initialization

- Following a power-up (voltage ON), a WRITE-CONFIG with data attachment is always required first (Config = 3).
- The data attachment described below is not necessary if the CONFIG-byte = 1.

Structure of the data attachment with normal addressing

MOBY I/E:

In the header of the RFID standard profile, the data attachment of the WRITE-CONFIG command must be specified with a length of 0Bh.

Table 3- 35 Structure of the data attachment for the WRITE-CONFIG command in the case of normal addressing MOBY I/E

1	2…5	6	7	8	9	10	11
04h	res.	05h	0	0	t_{SCAN}	Param	OPT1

Byte 1	Length of the subsequent sub-parameter in hex, here 4							
Byte 2 … 5	Reserved, permanently set to 0							
Byte 6	Length of the subsequent sub-parameter in hex, here 5							
Byte 7	Permanently set to 0							
Byte 8	Permanently set to 0							
Byte 9	Scanning_time is the scanning time for the MDS 507 of MOBY I. A value of 00 can be used here for all other types of MDS. The scan time setting is shown below (see also configuration manual for SLG (read/write device) 44/MDS 507). Example: A scanning time of 1 second results in the parameter scanning_time = 81h.							
	Bit	7	6	5	4	3	2	1	0
		Time base: 00 = 0.01 s 01 = 0.1 s 10 = 1 s 11 = 10 s		Time value: 00 … 3F (multiplied by time base)					
Byte 10	Setting of the MOBY operating mode							
	Bit	7	6	5	4	3	2	1	0
		Presence check and MDS control: 0 = no presence check 1 = no MDS control; presence check via firmware (default) 2 = MDS control and presence check via firmware			1 = ECC driver activated	MOBY mode: 0 = default 1 = MOBY I / E 4 = MOBY I with MDS 507 5 … 7 = reserved 8 = MOBY I dialog			
Byte 11	Bit	7	6	5	4	3	2	1	0
		unassigned				TST_ON (only with MOBY I) *	Timeout **	Reset ERR-LED ***	unassigned
* 1 = ASM replies with error if there are field errors on the write/read device ** 1 = ASM responds with error if no MDS present. *** 1 = delete flashing									

Example for MOBY I with presence check:

04h, 00h, 00h, 00h, 00h, 05h, 00h, 00h, 00h, 21h, 00h

RF300/MOBY D/U:

In the header of the RFID standard profile, the data attachment of the WRITE-CONFIG command must be specified with a length of 10h.

Table 3- 36 Structure of the data attachment for the WRITE-CONFIG command in the case of normal addressing RF300/MOBY D/U

1	2…5	6	7	8	9	10	11	12	13…14	15	16
04h	res.	0Ah	0	0	Standby	Param	OPT1	dili	Number of MDSs	fcon	ftim

Byte 1	Length of the subsequent sub-parameter in hex, here 4								
Byte 2 … 5	Reserved, permanently set to 0								
Byte 6	Length of the subsequent sub-parameter in hex, here 10d								
Byte 7	Permanently set to 0								
Byte 8	Permanently set to 0								
Byte 9	Standby time for the MDS								
	MOBY U:	00h = no standby mode 01h … C8h = 7 ms … 1400 ms standby time							
	RF300/MOBY D:	not used (00h)							
Byte 10	Setting of the MOBY operating mode								
	Bit	7	6	5	4	3	2	1	0
		Presence check and MDS control: 0 = no presence check 1 = no MDS control; presence check via firmware (default)			Reserved	MOBY mode: 5 = MOBY D / U or RF300 (without multitag)			
Byte 11	Bit	7	6	5	4	3	2	1	0
		unassigned						Reset ERR-LED on the SLG	unassigned
Byte 12	**MOBY U:** Range limitation 05h; 0Ah; 0Fh; 14h; 19h; 1Eh; 23h = 0.5; 1.0; 1.5; 2.0; 2.5; 3.0; 3.5 m 85h; 8Ah; 8Fh; 94h; 99h; 9Eh; A3h = ditto with reduced transmission power								
	MOBY D: HF-transmission power of 0.5 W to 10 W in 0.25 W steps 02h = 0.5 W : 10h = 4 W (default) : 28h = 10 W								
	RF300: not used (00h)								
Byte 13 … 14	Number of acquirable data carriers (max. 12 with MOBY U)								

Byte 15	**MOBY U:** BERO mode; automatic activation/deactivation of antenna field. The "Antenna ON/OFF" command is superimposed by the BERO mode.	
	00h =	without BEROs; no SLG synchronization
	01h =	one or two BEROs The BEROs are combined according to logic OR. The field is ON during actuation of a BERO
	02h =	One or two BEROs The 1st BERO switches the field on and the 2nd BERO switches the field off. If two BEROs are present and a field_ON_time is parameterized, the field is automatically switched off if the 2nd BERO does not switch within this BERO time. If no field_ON_time has been parameterized, the field remains switched on until actuation of the 2nd BERO.
	03h =	Activate write/read device synchronization via cable connection (see manual for configuring, mounting and service for MOBY U)
	RF300/MOBY D: not used (00h)	
Byte 16	**MOBY U:** Time for BERO mode (fcon = 02)	
	00h =	Timeout monitoring is deactivated. The 2nd BERO is needed in order to switch the field off.
	01h … FFh =	1 ... 255s turn on time for the SLG field.
	MOBY D: MDS type	
	00h =	I-code 1 (e.g. MDS D139)
	01h =	ISO transponder
	RF300: not used (00h)	

Example for MOBY U with presence check:

04h, 00h, 00h, 00h, 00h, 0Ah, 00h, 00h, 00h, 25h, 00h, 0Fh, 00h, 01h, 00h, 00h

Structure of the data attachment with filehandler

MOBY I:

In the header of the RFID standard profile, the data attachment of the WRITE-CONFIG command must be specified with a length of 0Dh.

Table 3- 37 Structure of the data attachment for the WRITE-CONFIG command in the case of filehandler MOBY I

1	2…5	6	7	8	9	10…11	12	13
04h	res.	07h	0	MDS_IO_CONTROL	ECC	SLG No.	Prio	res.

Byte 1	Length of the subsequent sub-parameter in hex, here 4		
Byte 2 … 5	Reserved, permanently set to 0		
Byte 6	Length of the subsequent sub-parameter in hex, here 7		
Byte 7	Permanently set to 0		
Byte 8	The different incoming/outgoing checks of the MDS can be controlled with MDS_IO_CONTROL.		
	MDS_IO_ CONTROL	Value	MDS control
	0	30h	Next mode: The "NEXT" command must be programmed by the user. Gap-free MDS control.
	1	31h	With timeout: There must be an MDS in the transmission window before the start of the command. "NEXT" command is not mandatory. No gap-free MDS control.
	4	34h	No incoming/outgoing check by RFID standard profile and ASM. "NEXT" command not mandatory. No gap-free MDS control.
	5	35h	No incoming/outgoing check by RFID standard profile and ASM. "NEXT" command not permissible. No gap-free MDS control.
	Please note: When working with multitag (MOBY_mode = 6), MDS_IO_CONTROL must be set to 1.		
Byte 9	ECC on → 31h ECC off → 30h		
Byte 10 … 11	Number of the reader This number is written to the tag / MDS before every execution procedure. This tells the filehandler whether or not a "new tag/MDS" has entered the field. **The user must ensure the parameter "SLG No." has a different value on every reader in every application.**		
	0001h ... FFFEh		Normal mode
	FFFFh		Test function: the SLG No. is not evaluated by the filehandler. In this way, the incoming/outgoing check of the filehandler (or the user program) can be tested with only one reader and one tag / MDS. The same tag/MDS can be repeatedly brought into the transmission window of a reader and processed there.
Byte 12	This byte defines which commands are permitted for this channel.		
	"R" (read)	Only commands of type "R" are permitted. It is not possible to overwrite the MDS.	
	"W" (write)	Commands of type "R" and "W" are permitted. User data can be read and written	
	"D" (delete)	All commands are executed. The MDS can be formatted.	
Byte 13	Reserved, permanently set to 0		

Example for MOBY I without ECC, MDS_IO_CONTROL = 4 without sampling interval:

04h, 00h, 00h, 00h, 00h, 07h, 00h, 34h, 30h, FFh, FFh, 'D', 00h

MOBY U:

In the header of the RFID standard profile, the data attachment of the WRITE-CONFIG command must be specified with a length of 13h.

Table 3- 38 Structure of the data attachment for the WRITE-CONFIG command in the case of filehandler MOBY U

1	2…5	6	7	8	9	10…11	12	13	14	15	16…17	18	19
04h	res.	0Dh	0	MDS_IO_CONTROL	ECC	SLG No.	Prio	res.	scanning_time	dili	Number of MDSs	fcon	ftim

Byte 1	Length of the subsequent sub-parameter in hex, here 4		
Byte 2 … 5	Reserved, permanently set to 0		
Byte 6	Length of the subsequent sub-parameter in hex, here 13d		
Byte 7	Permanently set to 0		
Byte 8	The different incoming/outgoing checks of the MDS can be controlled with MDS_IO_CONTROL.		
	MDS_IO_CONTROL	Value	MDS control
	0	30h	Next mode: The "NEXT" command must be programmed by the user. Gap-free MDS control.
	1	31h	With timeout: There must be an MDS in the transmission window before the start of the command. "NEXT" command is not mandatory. No gap-free MDS control.
	4	34h	No incoming/outgoing check by RFID standard profile and ASM. "NEXT" command not mandatory. No gap-free MDS control.
	5	35h	No incoming/outgoing check by RFID standard profile and ASM. "NEXT" command not permissible. No gap-free MDS control.
	Please note: When working with multitag (MOBY_mode = 6), MDS_IO_CONTROL must be set to 1.		
Byte 9	Permanently set to 30h		
Byte 10 … 11	Number of the reader This number is written to the tag / MDS before every execution procedure. This tells the filehandler whether or not a "new tag/MDS" has entered the field. **The user must ensure the parameter "SLG No." has a different value on every reader in every application.**		
	0001h ... FFFEh		Normal mode
	FFFFh		Test function: the SLG No. is not evaluated by the filehandler. In this way, the incoming / outgoing check of the filehandler (or the user program) can be tested with only one reader and one tag / MDS. The same tag/MDS can be repeatedly brought into the transmission window of a reader and processed there.

Byte 12	This byte defines which commands are permitted for this channel.	
	"R" (read)	Only commands of type "R" are permitted. It is not possible to overwrite the MDS.
	"W" (write)	Commands of type "R" and "W" are permitted. User data can be read and written
	"D" (delete)	All commands are executed. The MDS can be formatted.
Byte 13	Reserved, permanently set to 0	
Byte 14	Scanning_time describes the standby time for the MDS. If the MDS receives another command before expiry of the scanning_time, this can be executed immediately. If the MDS receives a command after expiry of the scanning_time, execution of the command is delayed by the sleep_time. Scanning_time should only be set when if • the MDS is processed with several commands, and • Execution procedure must be concluded within a minimum time. 00 hex = no standby time (default) 01 hex = 7 ms standby time 02 hex = 14 ms standby time … … C8 hex = 1400 ms standby time Remember: Scanning_time affects the life of the battery. The longer scanning_time, the shorter the life of the battery. For precise calculations, see the MOBY U manual for configuration, mounting and service.	
Byte 15	Range limitation 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	
Byte 16 … 17	Maximum number of MDSs being processed in parallel in the field. Permissible values = 0001h...000Ch	
Byte 18	BERO-mode; automatic switching on/off of the antenna field The "Antenna ON/OFF" command is superimposed by the BERO mode.	
	00h =	without BEROs
	01h =	one or two BEROs. The BEROs are combined according to logic OR. The field is ON during actuation of a BERO
	02h =	One or two BEROs. The 1st BERO switches the field on and the 2nd BERO switches the field off. If two BEROs are available and a field_ON_time has been parameterized, the field is automatically switched off if the 2nd BERO does not switch within this BERO time. If no field_ON_time has been parameterized, the field remains switched on until actuation of the 2nd BERO.
Byte 19	Time for BERO mode (fcon = 02)	
	00h =	Timeout monitoring is deactivated. The 2nd BERO is needed in order to switch the field off.
	01h...FFh =	1 ... 255s turn on time for the SLG field.

3.2.5 Command execution procedure

Command execution involves the following steps:

1. Write a command (1…10). Command 1 is reserved for initialization. It is executed if the input INIT of FB 101 is set.
2. Transer the data to be written to the send data buffer TXBUF.
3. Select the previously written command (1...10) with the parameter CMDSEL.
4. Process the command with EXECUTE (EXECUTE must be set to "1").

   The outputs BUSY and DONE must assume the values BUSY = TRUE and DONE = FALSE after the next FB 101 call
5. Wait in your program until the bits BUSY = FALSE and DONE = TRUE. The command has now been executed without errors.
   However, if ERROR = TRUE, continute with Point 6. Otherwise, continue with Step 7.
6. Evaluate the errors that have occurred.
7. Reset the EXECUTE bit.

The diagram below shows the general timing of FB 101. A command start always takes place with the positive edge of EXECUTE, INIT or SRESET.

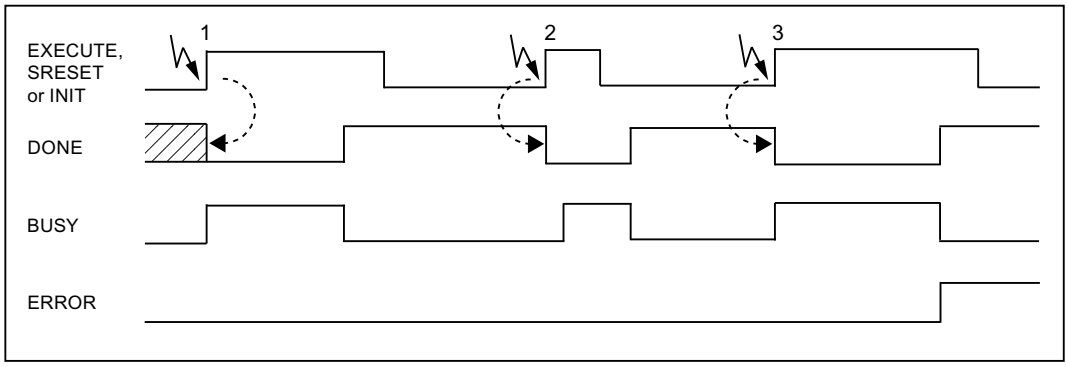


Figure 3-13 General timing of the FB 101

Case 1: The input EXECUTE/SRESET/INIT remains set until FB 101 indicates termination of the command. The input must be reset by the user after DONE = 1 of FB 101 has been set.

Case 2: The user outputs a pulse to EXECUTE / SRESET / INIT for at least one cycle.

Case 3: However, as in case 1, the command is terminated with errors.

Commissioning

4

You can program and startup the function block FB 101

- with STL/LAD/FBD programming

  You will find information on this in the following chapters:

  – *Configuring the hardware*

  – *Programming in STL (statement list)*

- via Structured Control Language (SCL)

  You will find information on this in the following chapters:

  – *Configuring the hardware*

  – *Programming in SCL (Structured Control Language)*

4.1 Configuring the hardware

Communication between the controller and the communication module is via PROFIBUS DP or PROFINET. I/O addresses from 256 are recommended for the RFID system in accordance with the standard profile (e.g. ASM 456).

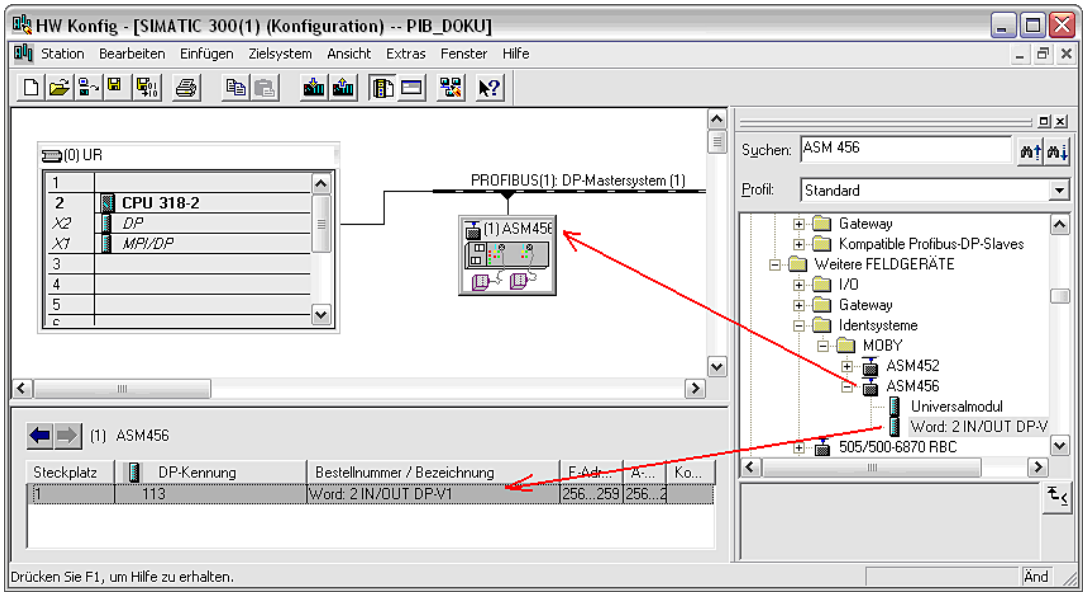


Figure 4-1 HW configuration for ASM 456 (example for addressing)

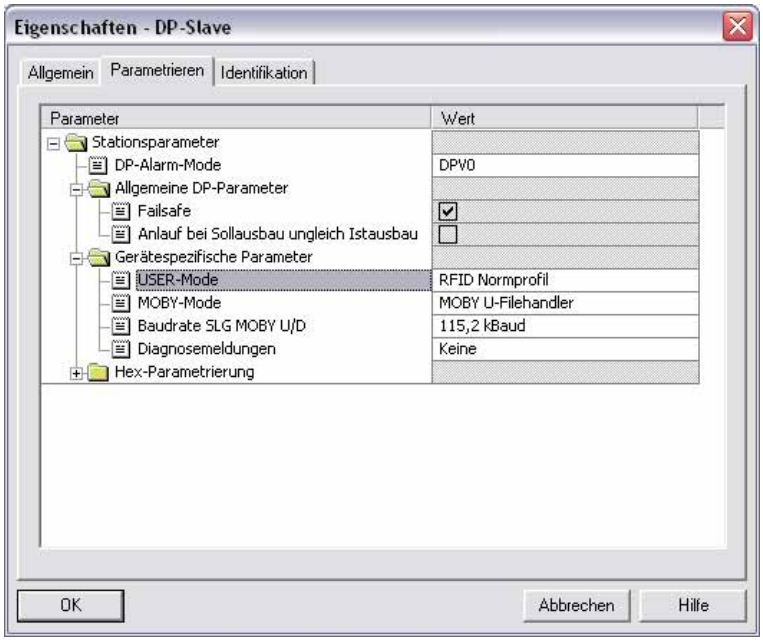


Figure 4-2 Parameterizing of the communication module (example)

4.2 Programming in STL (statement list)

4.2.1 Copy required blocks

Requirements

An already available project must not contain the following blocks:

FB 1, FB 2, FB 3, FB 4, FB 5, FC 1, FC 2, FC 3, FC 4, UDT 1

Insert first the required function block (FB 101, FB 116 or FB 132) and all other blocks from the sample project into the new project.

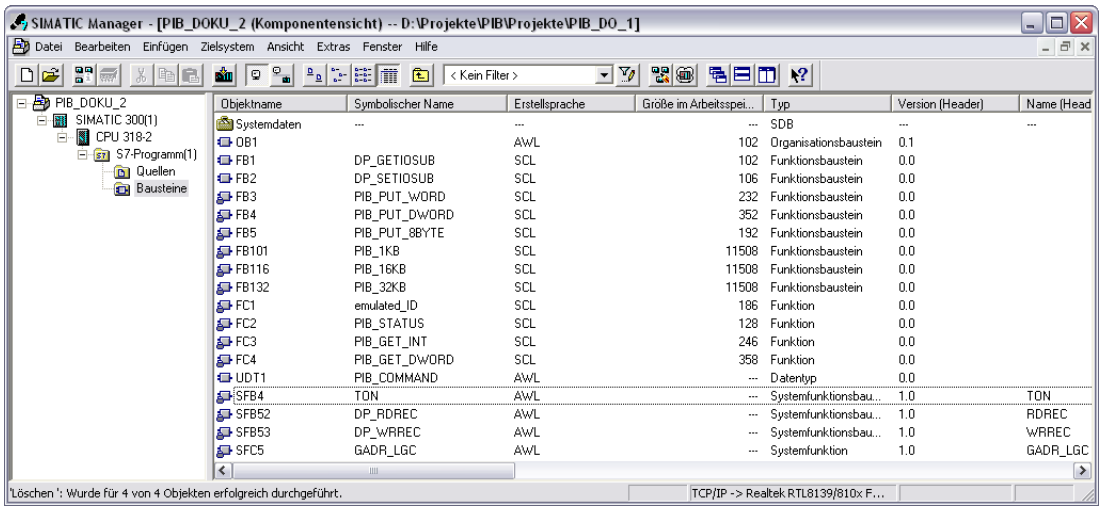


Figure 4-3 Blocks of the sample project

Note

Renaming the blocks FB 1 … FC 4

The necessary renaming of these blocks is only possible through programming in SCL (Structured Control Language).
You can find information on this in the section *Programming in SCL (Structured Control Language)*.

4.2.2 Create data areas for FB 101

UDT 1 PIB_COMMAND is used for the command area.

1. Create a new DB via the menu **Insert > S7 Block > Data Block**

2. Open this and configure it as shown in figure below.

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Com_Kanal1	ARRAY[1..10]		
*38.0		"PIB_COMMAND"		
+380.0	TXBUF	ARRAY[1..1024]		
*1.0		BYTE		
+1404.0	RXBUF	ARRAY[1..1024]		
*1.0		BYTE		
=2428.0		END_STRUCT		

Figure 4-4 Configuring UDT 1

NOTICE

The sizes of the receive data buffer and the send data buffer must correspond to the sizes of the variables RXREF and TXREF. These differ in the case of PIB_1KB, PIB_16KB and PIB_32KB.

When working with FB PIB_32KB, you cannot combine the data areas in one data block without exceeding the maximum permissible size for DBs. In such a case, a separate data block must be created for the receive data buffer.

4.2.3 Create commands in the current values of the structure PIB_COMMAND

Procedure

1. Store the command WRITE-CONFIG in the first structure of PIB_COMMAND.

Adresse	Name	Typ	Anfangswert	Aktualwe	Komm
0.0	command[1].CMD	BYTE	B#16#0	B#16#78	
1.0	command[1].Config	BYTE	B#16#0	B#16#3	
2.0	command[1].OffsetBuffer	INT	0	0	
4.0	command[1].UID[1]	BYTE	B#16#0	B#16#0	
5.0	command[1].UID[2]	BYTE	B#16#0	B#16#0	
6.0	command[1].UID[3]	BYTE	B#16#0	B#16#0	
7.0	command[1].UID[4]	BYTE	B#16#0	B#16#0	
8.0	command[1].UID[5]	BYTE	B#16#0	B#16#0	
9.0	command[1].UID[6]	BYTE	B#16#0	B#16#0	
10.0	command[1].UID[7]	BYTE	B#16#0	B#16#0	
11.0	command[1].UID[8]	BYTE	B#16#0	B#16#0	
12.0	command[1].FileName[1]	BYTE	B#16#0	B#16#0	
13.0	command[1].FileName[2]	BYTE	B#16#0	B#16#0	
14.0	command[1].FileName[3]	BYTE	B#16#0	B#16#0	
15.0	command[1].FileName[4]	BYTE	B#16#0	B#16#0	
16.0	command[1].FileName[5]	BYTE	B#16#0	B#16#0	
17.0	command[1].FileName[6]	BYTE	B#16#0	B#16#0	
18.0	command[1].FileName[7]	BYTE	B#16#0	B#16#0	
19.0	command[1].FileName[8]	BYTE	B#16#0	B#16#0	
20.0	command[1].Offset	DINT	L#0	L#0	
24.0	command[1].Length	INT	0	13	
26.0	command[1].StartAddress	DINT	L#0	L#0	
30.0	command[1].Attributes	BYTE	B#16#0	B#16#0	
31.0	command[1].NextMode	BYTE	B#16#0	B#16#0	
32.0	command[1].Timeout	INT	0	0	
34.0	command[1].ObjectNumber	INT	0	0	
36.0	command[1].FileType	WORD	W#16#0	W#16#0	

2. Write the parameterization data for WRITE-CONFIG into the first location of the TXBUF memory.

380.0	RXBUF[1]	BYTE	B#16#0	B#16#4
381.0	RXBUF[2]	BYTE	B#16#0	B#16#0
382.0	RXBUF[3]	BYTE	B#16#0	B#16#0
383.0	RXBUF[4]	BYTE	B#16#0	B#16#0
384.0	RXBUF[5]	BYTE	B#16#0	B#16#0
385.0	RXBUF[6]	BYTE	B#16#0	B#16#7
386.0	RXBUF[7]	BYTE	B#16#0	B#16#0
387.0	RXBUF[8]	BYTE	B#16#0	B#16#34
388.0	RXBUF[9]	BYTE	B#16#0	B#16#30
389.0	RXBUF[10]	BYTE	B#16#0	B#16#FF
390.0	RXBUF[11]	BYTE	B#16#0	B#16#FF
391.0	RXBUF[12]	BYTE	B#16#0	B#16#0
392.0	RXBUF[13]	BYTE	B#16#0	B#16#F0

3. Store a command for the RFID reader (e.g. PHYSICAL-READ) in at least the second structure of PIB_COMMAND.

38.0	command[2].CMD	BYTE	B#16#0	B#16#70
39.0	command[2].Config	BYTE	B#16#0	B#16#0
40.0	command[2].OffsetBuffer	INT	0	0
42.0	command[2].UID[1]	BYTE	B#16#0	B#16#0
43.0	command[2].UID[2]	BYTE	B#16#0	B#16#0
44.0	command[2].UID[3]	BYTE	B#16#0	B#16#0
45.0	command[2].UID[4]	BYTE	B#16#0	B#16#0
46.0	command[2].UID[5]	BYTE	B#16#0	B#16#0
47.0	command[2].UID[6]	BYTE	B#16#0	B#16#0
48.0	command[2].UID[7]	BYTE	B#16#0	B#16#0
49.0	command[2].UID[8]	BYTE	B#16#0	B#16#0
50.0	command[2].FileName[1]	BYTE	B#16#0	B#16#0
51.0	command[2].FileName[2]	BYTE	B#16#0	B#16#0
52.0	command[2].FileName[3]	BYTE	B#16#0	B#16#0
53.0	command[2].FileName[4]	BYTE	B#16#0	B#16#0
54.0	command[2].FileName[5]	BYTE	B#16#0	B#16#0
55.0	command[2].FileName[6]	BYTE	B#16#0	B#16#0
56.0	command[2].FileName[7]	BYTE	B#16#0	B#16#0
57.0	command[2].FileName[8]	BYTE	B#16#0	B#16#0
58.0	command[2].Offset	DINT	L#0	L#0
62.0	command[2].Length	INT	0	1000
64.0	command[2].StartAddress	DINT	L#0	L#0
68.0	command[2].Attributes	BYTE	B#16#0	B#16#0
69.0	command[2].NextMode	BYTE	B#16#0	B#16#0
70.0	command[2].Timeout	INT	0	0
72.0	command[2].ObjectNumber	INT	0	0
74.0	command[2].FileType	WORD	W#16#0	W#16#0

4.2.4 Call the function block and interconnect the variables

Open the block for this in which the function block for the RFID standard profile is to be called, and enter the following command line.

Call FB101, DB101

Note

Data block 101 is the DB that is to be used as the instance for this FB call. The number can be freely selected.

4.2.5 Interconnect the parameters after the FB call

After calling the FB, you must interconnect the individual parameters accordingly.

```
CALL  "PIB_1KB" , DB101
 EXECUTE :=E0.0                    //Befehlsstart
 ID      :=DW#16#100               //Basisadresse
 INDEX   :=111                     //festgelegter Datensatz für Kanal 1
 OFFSET  :=0                       //festgelegte Offset für Kanal 1
 RPTCMD  :=E0.1
 SRESET  :=E0.2
 INIT    :=E0.3
 UOUT0   :=E0.4
 UOUT1   :=E0.5
 UOUT2   :=E0.6
 UOUT3   :=E0.7
 RDGATE  :=E1.0
 CMDDIM  :=10
 CMDSEL  :=2                       //Zeiger auf den zweiten Befehl
 TXBUFLEN:=L#1024
 TXSTART :=L#1
 RXBUFLEN:=L#1024
 RXSTART :=L#1
 STATUS  :=AD0
 DONE    :=A4.0
 BUSY    :=A4.1
 ERROR   :=A4.2
 WARNING :=A4.3
 TPC     :=A4.4
 TP      :=A4.5
 UIN0    :=A4.6
 UIN1    :=A4.7
 UIN2    :=A5.0
 UIN3    :=A5.1
 TRLEN   :=AD6
 CMDREF  :="Command".Kanal1
 TXREF   :=DB2.txref
 RXREF   :=DB2.rxref
```

Figure 4-5 Interconnecting the parameters of the FB

4.2.6 Load the program and test the devices

The program is tested in the following steps:

1. Load the project onto the SIMATIC CPU.
2. Connect the reader to one channel of the ASM 456.
3. After restarting the SIMATIC CPU (STOP → RUN), the CPU must not go to STOP.
   If the CPU goes to STOP, you must carry out a fault analysis. To do this, analyze the diagnostics messages of the CPU (via the menu pointPLC > Module State).
   Fault causes can include:
   - The I/O address of the module in HW Config and the parameterized ID do not agree, or the parameterized ID is not available in the I/O.
   - OFFSET and INDEX are incorrect. Check the values according to the specifications in the section *Description > Configuration scheme*.
4. Start parameterization of the communication module with the INIT switch (I0.3). The command must be completed with DONE = TRUE and ERROR = FALSE. After this step, communication between FB 101 and the communication will function.
5. Start the command to the reader with the EXECUTE switch (I0.1). You receive the read result as soon as you move a tag into the transmission window. The presence of a tag can be indicated at output TP (A4.5).

   After this command has been completed with DONE = TRUE and ERROR = FALSE, startup of the communication module and the connected reader is complete.

4.3 Programming in SCL (Structured Control Language)

Requirements

The option package *S7-SCL* must be installed on your PC / PD.

4.3.1 Copy required SCL sources into the project

Requirements

The function blocks FB 101, FB 116 and FB 132 require five FBs, four FCs and one UDT for processing. You must assign the numbers for these blocks in the symbol table.

Table 4- 1 Required blocks for SCL

Number	Block name	Block address
FBs		
1 2 3 4 5 6 7 8	DP_ GETIOSUB DP_SETIOSUB PIB_PUT_WORD PIB_PUT_DWORD PIB_PUT_8BYTE PIB_1KB PIB_16KB PIB_32KB	FB 1 FB 2 FB 3 FB 4 FB 5 FB 101 FB 116 FB 132
FCs		
9 10 11 12	emulated_ID PIB_STATUS PIB_GET_INT PIB_GET_DWORD	FC 1 FC 2 FC 3 FC 4
UDT		
17	PIB_COMMAND	UDT 1

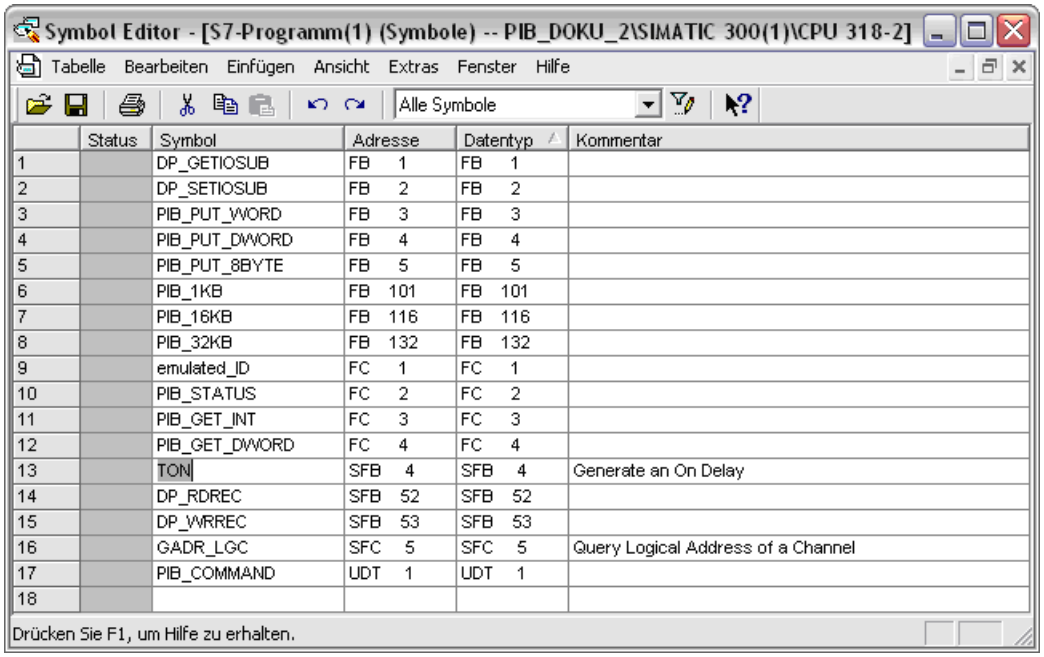
Information for handling the blocks:

- The block names must not be changed.
- The block addresses are freely selectable.

Assign the following symbolic names:

- SFB 52 is called DP_RDREC
- SFB 53 is called DP_WRREC

The FB and FC numbers assigned in the symbol table below are not mandatory. Only the SFBs and SFCs must be entered in the table in exactly this way.



	Status	Symbol	Adresse	Datentyp	Kommentar
1		DP_GETIOSUB	FB 1	FB 1	
2		DP_SETIOSUB	FB 2	FB 2	
3		PIB_PUT_WORD	FB 3	FB 3	
4		PIB_PUT_DWORD	FB 4	FB 4	
5		PIB_PUT_8BYTE	FB 5	FB 5	
6		PIB_1KB	FB 101	FB 101	
7		PIB_16KB	FB 116	FB 116	
8		PIB_32KB	FB 132	FB 132	
9		emulated_ID	FC 1	FC 1	
10		PIB_STATUS	FC 2	FC 2	
11		PIB_GET_INT	FC 3	FC 3	
12		PIB_GET_DWORD	FC 4	FC 4	
13		TON	SFB 4	SFB 4	Generate an On Delay
14		DP_RDREC	SFB 52	SFB 52	
15		DP_WRREC	SFB 53	SFB 53	
16		GADR_LGC	SFC 5	SFC 5	Query Logical Address of a Channel
17		PIB_COMMAND	UDT 1	UDT 1	
18					

Figure 4-6 Symbol table for SCL (example)

Copy SCL sources into the project

How to proceed:

1. Copy the required SCL sources into the project.
2. Compile all sources for later processing.

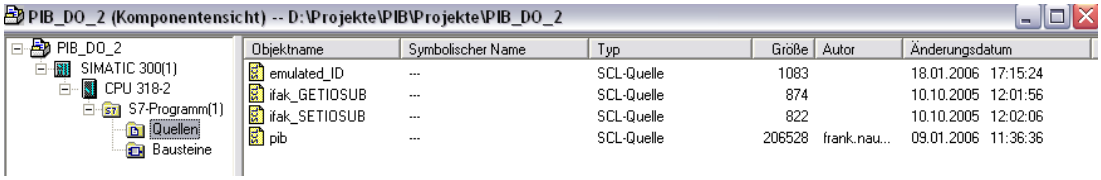


Figure 4-7 Copying the SCL sources

4.3.2 Create required data areas

UDT PIB_COMMAND is used for the command area. The data areas can be created in a data block at 1 KB or 16 KB

How to proceed:

1. Insert a new global data block into a new SCL source (**Insert > Block Template > DB**)
2. Assign a number or a symbolic name to the DB (this must already have been entered in the symbol table).
3. Enter the necessary variables (see the figure below).

```
DATA_BLOCK DB2
//
// Baustein-Kommentar ...
//
  STRUCT
Com_Kanal1 : ARRAY [1..10] OF UDT1;
TXBUF : ARRAY [1..1024] OF BYTE;
RXBUF : ARRAY [1..1024] OF BYTE;

  END_STRUCT
BEGIN

END_DATA_BLOCK
```

Figure 4-8 Variables for the global data block

NOTICE
The sizes of the receive data buffer and the send data buffer must correspond to the sizes of the variables RXREF and TXREF. These differ in the case of PIB_1KB, PIB_16KB and PIB_32KB. When working with FB PIB_32KB, you cannot insert the data areas in one data block without exceeding the maximum permissible size for DBs. In such a case, a separate data block must be created for the receive data buffer.

4.3.3 Call the function block and interconnect the variables

1. Insert a block (OB) via the menu point **Inert > Block Template > OB**. Assign this OB the number 1.

```
ORGANIZATION_BLOCK OB1

VAR_TEMP
  // reserviert
  info : ARRAY[0..19] OF BYTE;
  // temporäre Variablen

END_VAR

  // Anweisungen

  ;
END_ORGANIZATION_BLOCK
```

Figure 4-9 Insert OB 1

Note

If you use an FC or FB at this point, you must remember that this will be called later by OB 1.

2. Start the code arey of the OB after teh statement END_VAR with the line START. Click on the relevant block (PIB_1KB) in the project. from the menu point **Insert > Block Call…**

Note

If you cannot find the block in the current project, the inserted SCL sources have not been compiled.

```
ORGANIZATION_BLOCK OB1

VAR_TEMP
  // reserviert
  info : ARRAY[0..19] OF BYTE;
  // temporäre Variablen

END_VAR

  // Anweisungen
BEGIN
PIB_1KB.DB101(EXECUTE :=  // IN: BOOL
              ,ID :=  // IN: DWORD
              ,INDEX :=  // IN: INT
              ,OFFSET :=  // IN: INT
              ,RPTCMD :=  // IN: BOOL
              ,SRESET :=  // IN: BOOL
              ,INIT :=  // IN: BOOL
              ,UOUT0 :=  // IN: BOOL
              ,UOUT1 :=  // IN: BOOL
              ,UOUT2 :=  // IN: BOOL
              ,UOUT3 :=  // IN: BOOL
              ,RDGATE :=  // IN: BOOL
              ,CMDDIM :=  // IN: INT
              ,CMDSEL :=  // IN: INT
              ,TXBUFLEN :=  // IN: DINT
              ,TXSTART :=  // IN: DINT
              ,RXBUFLEN :=  // IN: DINT
              ,RXSTART :=  // IN: DINT
              ,CMDREF :=  // INOUT: ARRAY
              ,TXREF :=  // INOUT: ARRAY
              ,RXREF :=  // INOUT: ARRAY
              );
   := DBxxx.STATUS; // OUT: DWORD
   := DBxxx.DONE; // OUT: BOOL
   := DBxxx.BUSY; // OUT: BOOL
   := DBxxx.ERROR; // OUT: BOOL
   := DBxxx.WARNING; // OUT: BOOL
   := DBxxx.TPC; // OUT: BOOL
   := DBxxx.TP; // OUT: BOOL
   := DBxxx.UIN0; // OUT: BOOL
   := DBxxx.UIN1; // OUT: BOOL
   := DBxxx.UIN2; // OUT: BOOL
   := DBxxx.UIN3; // OUT: BOOL
   := DBxxx.TRLEN; // OUT: DINT
 ;
END_ORGANIZATION_BLOCK
```

Figure 4-10 Inserting PIB_1KB into OB 1

3. The instance DB is offered with the number xxx. Assign another number for the DB here.

Note

Data block 101 is the DB that is to be used as the instance for this FB call. The number can be freely selected.

4.3.4 Interconnect the parameters after the FB call

After calling the FB, you must interconnect the individual parameters accordingly.

```
BEGIN
PIB_1KB.DB101(EXECUTE := E0.0 // IN: BOOL
              ,ID := DW#16#100 // IN: DWORD
              ,INDEX := 111 // IN: INT
              ,OFFSET := 0 // IN: INT
              ,RPTCMD := E0.1 // IN: BOOL
              ,SRESET := E0.2 // IN: BOOL
              ,INIT := E0.3 // IN: BOOL
              ,UOUT0 := E0.4 // IN: BOOL
              ,UOUT1 := E0.5 // IN: BOOL
              ,UOUT2 := E0.6 // IN: BOOL
              ,UOUT3 := E0.7 // IN: BOOL
              ,RDGATE := E1.0 // IN: BOOL
              ,CMDDIM := 10 // IN: INT
              ,CMDSEL := 2 // IN: INT
              ,TXBUFLEN := 1024 // IN: DINT
              ,TXSTART := 1 // IN: DINT
              ,RXBUFLEN := 1024 // IN: DINT
              ,RXSTART := 1 // IN: DINT
              ,CMDREF := DB2.COM_Kanal1 // INOUT: ARRAY
              ,TXREF := DB2.TXBUF // INOUT: ARRAY
              ,RXREF := DB2.RXBUF // INOUT: ARRAY
              );
  AD0 := DB101.STATUS; // OUT: DWORD
  A4.0 := DB101.DONE; // OUT: BOOL
  A4.1 := DB101.BUSY; // OUT: BOOL
  A4.2 := DB101.ERROR; // OUT: BOOL
  A4.3 := DB101.WARNING; // OUT: BOOL
  A4.4 := DB101.TPC; // OUT: BOOL
  A4.5 := DB101.TP; // OUT: BOOL
  A4.6 := DB101.UIN0; // OUT: BOOL
  A4.7 := DB101.UIN1; // OUT: BOOL
  A5.0 := DB101.UIN2; // OUT: BOOL
  A5.1 := DB101.UIN3; // OUT: BOOL
  AD6 := DB101.TRLEN; // OUT: DINT AD6 muss als  DINT in der Symboltabelle eingetragen sein,
                      //           sonst wird die SCL-Quelle nicht fehlerfrei übersetzt!

  ;
END_ORGANIZATION_BLOCK
```

Figure 4-11 Interconnecting the FB parameters

4.3.5 Compile, load and test program.

The program is tested in the following steps:

1. Compile the entire program.
2. Carry out the remaining startup as described in the section *Programming in STL (statement list) > Load the program and test the devices*.

Error messages and troubleshooting 5

There is always an error state in the RFID standard profile function if the output parameter ERROR = TRUE. The error can be analyzed (decoded) using the STATUS output parameter.

5.1 Structure of the status output parameter

The status output parameter comprises 4 bytes:

- Byte 0 (function numbers)

  Bits 0.3 to 0.0 indicate which components of the RFID system report the error (status).

- Byte 1 (error numbers)

  This byte defines the meaning of the error code and the warnings. The error numbers have the followinig meaning:

  - 0x00 Not a warning, not an error
  - 0x80 Error message from PROFIBUS DP-V1 or PROFINET (in accordance with IEC 61158-6)
  - 0x81 … 0x8F The controllers reports an error in accordance with the parameter "x" (0x8x).
  - 0xFE Error of RFID standard profile or communication module/reader

- Byte 2 (error code)
- Byte 3 (warnings)

  In this byte, each bit has a separate meaning.

5.2 Error messages

Errors from the communication module / reader

The causes of these errors can include:

- Communication between the communication module/ASM and the reader/ write/read device or between the reader/ write/read device and the tag contains errors.
- The communication module/ASM is unable to process the command.

Byte 3 of the STATUS is not relevant for the error messages.

With certain error messages of the communication module/ASM, the ERR-LED also flashes (see both tables below).

Table 5- 1 Evaluation of the ERR-LED in normal addressing

Flashing of ERR LED	Errors that have occurred
1x	No error Communication module/ASM has executed a startup and is waiting for an init_run.
2x	Presence error/processing error
3x	Error in the connection to the reader / write/read device, write/read device does not answer.
4x	Error in tag memory
5x	Unknown command
6x	Field disturbance on reader/SLG
7x	Too many transmit errors
8x	CRC sending error
9x	Only during initialization: CRC error during acknowledgment receipt from tag
10x	Only during initialization: Tag is unable to perform the initialization command.
11x	When formatting, the tag must be in the transmission window of the reader/SLG, otherwise a timeout error will occur.
12x	The tag memory cannot be write-accessed.
13x	Address error
14x	ECC error (only possible when ECC_mode = TRUE)
16x	NEXT-command not possible or not permitted
18x	Internal communication error of the communication module/ASM
19x	Communication module/ASM/SLG U does not have enough buffer to store the command intermediately.
20x	Internal monitoring error of the communication module/ASM
21x	Incorrect parameterization of the communication module/ASM
22x	The command cannot be executed with the parameterization of the communication module/ASM on PROFIBUS or PROFINET.
23x	Communication error between RFID standard profile and communication module / ASM. Handshake error.
25x	An error has occurrerd that makes a WRITE-CONFIG with Config = 3 necessary.

Flashing of ERR LED	Errors that have occurred
28x	The antenna of the reader/SLG is turned off. A tag command to the communication module/ASM was started in this state.
30x	Error when processing the command

Table 5- 2 Evaluation of the ERR-LED with filehandler

Flashing of ERR LED	Errors that have occurred
1x	The filehandler will now only accept a WRITE-CONFIG command mit Config = 3.
2x	During certain important operations (e.g. writing to the system area of the tag, format tag), the tag must not exit the transmisison window of the reader/SLG, since otherwise this error would cancel the command.
3x	Error in the connection to the reader / write/read device
4x	The tag signals a memory error.
5x	The commands FORMAT or PHYSICAL-READ have been issued with incorrect parameters. The physically accessed address is not available on the tag (tag memory is smaller than specified in the command).
6x	Field disturbance on reader/SLG
7x	Too many transmit errors
8x	CRC sending error
9x	Field disturbance on reader/SLG
10x	Tag is unable to perform the FORMAT command. The tag is defective.
11x	When formatting, the tag must be in the transmission window of the reader/SLG, otherwise a timeout error will occur.
12x	The tag memory cannot be write-accessed.
13x	The commands FORMAT or PHYSICAL-READ have been issued with incorrect parameters. The physically accessed address is not available on the tag (tag memory is smaller than specified in the command).
14x	ECC error (only possible when ECC_mode = TRUE)
15x	The filehandler is not working correctly.
18x	Internal communication error of the communication module/ASM • Hardware is defective • Carry out a restart
20x	Internal overflow of the communication module / ASM; stack overflow; SPC memory overflow; diagnostics not functioning Send WRITE-CONFIG or complete restart Switch communication module off and on again. Check bus parameterization
21x	Incorrect parameterization of the communication module/ASM • Check parameters in HW Config
30x	Corrupt message frame from reader / write/read device

Table 5- 3 Error messages from communication module / reader via the STATUS output parameter

Byte			Description
0	1	2	
E1h	FEh	01h	**In the case of normal addressing:** The tag memory cannot be write-accessed. • Tag memory is defective. • EEPROM tag has been overwritten too frequently and has reached the end of its service life
			In the case of filehandler: The tag memory cannot be write-accessed. • Tag has a smaller memory than specified in the FORMAT command, that is, parameterize the tag type correctly • Tag memory is defective. • EEPROM tag has been overwritten too frequently and has reached the end of its service life
E1h	FEh	02h	**In the case of normal addressing:** Presence error: The tag has moved out of the transmission window of the reader / write/read device. The command was executed only partially. Read command: No data are transmitted to the RFID standard profile. Write command: The tag which just left the field contains an incomplete data record. • Operating distance from reader / write/read device to tag is not being maintained. • Configuration error: The data record to be processed is too large (in dynamic mode) The next command is automatically executed on the next tag. A read, write or NEXT command is possible. • With timeout: No tag in field
			In the case of filehandler: During certain important operations (e.g. writing to the system area of the tag, format tag), the tag must not exit the transmisison window of the reader/SLG, since otherwise this error would cancel the command. • Start command again • Tag is in the boundary area of the write/read device transmission window
E1h	FEh	03h	**In the case of normal addressing:** Address error The address area of the tag is exceeded. • Starting address of the command start has been incorrectly issued • The tag is not the right type. • RF300: Attempted write access to write-protected areas (Address area FF00 … FF90)
			In the case of filehandler: • The commands FORMAT or PHYSICAL-READ have been issued with incorrect parameters. The physically accessed address is not available on the tag (tag memory is smaller than specified in the command). • With READ/WRITE/UPDATE: Branching to the FAT is defective; it is shown on a block that is not available on the tag. • Address error. The address area of the tag is exceeded (the tag is not of the right type) • The logically accessed address is outside the file. There is an error in the FAT. The tag must be reformatted. • A FAT block follow-on error has been detected in the READ or WRITE command. The block branching table (FAT) is defective. The tag must be reformatted.

Byte			Description
0	1	2	
E1h	FEh	04h	**In the case of normal addressing:** Only during initialization: Tag is unable to perform the initialization command. • Tag is defective
			In the case of filehandler: Tag is unable to perform the FORMAT command. The tag is defective.
E1h	FEh	05h	**In the case of filehandler:** WRITE command: Insufficient memory available on the tag The data are not fully written to the tag. CREATE command: No data block can be reserved for a file when it is created. No further blocks are free
E1h	FEh	06h	**In the case of normal addressing:** Error in the tag memory The MDS has never been write-accessed or has lost the contents of its memory due to battery failure. • Replace tag (if battery bit is set). • Initialize tag with the STG • Re-initialize tag
			In the case of filehandler: • Tag reports a memory error The tag has never been write-accessed or has lost the contents of its memory due to battery failure (not with EEPROM tag). – Replace tag or battery (if battery_low is set). – Test tag by attempting to initialize it with the STG – Format tag with FORMAT • The tag could not be identified by the filehandler. The tag must be reformatted.
E1h	FEh	07h	**With normal addressing and filehandler:** ECC error (only possible if ECC_mode = TRUE) The data cannot be read by the tag. • Data of the tag have been lost (tag defective). • Tag has not been formatted with ECC driver → Reformat tag • Tag with EEPROM has reached the end of its service life; the data have been lost. → Replace tag • When overwriting, the tag was moved out of the field → The tag is not correctly positioned • Command to communication module / ASM was issued incorrectly by user.
E1h	FEh	08h	**In the case of normal addressing:** The tag in the field does not have the expected UID or has no UID.
			In the case of filehandler: The wrong UID has been entered in the command or the tag with the UID entered in teh command is not (or no longer) in the filed. • Application error; check UID in the command

Byte			Description
0	1	2	
E2h	FEh	01h	**With normal addressing and filehandler:** • Field interference on the reader/SLG The reader / write/read device is receiving interference from its environment. – External interference field; the field of interference can be verified with the "inductive field indicator" of the STG. – The distance between two readers / write/read device is too small and does not correspond to the configuration guidelines – The connecting cable to the reader / write/read device is defective or too long or does not comply with the specification – MOBY U: MDS has left the field during communication. – MOBY U: Communication between the write/read device and the MDS was terminated by interference (e.g. person/foreign body moving between reader / write/read device and MDS). • Too many transmission errors The tag was not able to correctly receive the command or the write data from the communication module / ASM even after several attempts. – The tag is positioned exactly on the boundary of the transmission window. – Data transmission to the tag is being affected by external interference. • CRC sending error – The receiver monitor has detected at least one fault during transmission. (Cause as for field interference on the reader / write/read device) – The tag reports CRC error frequently (tag is positioned on the boundry of the reader / write/read device; tag and/or reader / write/read device has/have a hardware defect) • Only during initialization: CRC error on receipt of acknowledgement from tag (cause as for field interference on the reader / write/read device) • When formatting, the tag must be in the transmission window of the reader / write/read device, otherwise a timeout error will occur, that is: – The tag is positioned exactly on the boundary of the transmission window – The tag needs too much current (defect) – EEPROM tag for FORMAT incorrectly parameterized • on MOBY I with MDS 507: – Check MOBY_mode and scanning_time parameters – Dialog battery is discharged (check Bit LR_bat; measuring voltage on battery)
E2h	FEh	02h	**In the case of normal addressing:** There are more tags in the transmission window than the reader / write/read device can process simultaneously (only 1 tag can be processed at any time with the RFID standard profile).
			In the case of filehandler: The number of tags in the field is higher than the parameterized number of tags of "multitag". • Remote the excess tags in the field • The configuring of distance_limiting has been incorrectly set • Search the vicinity of the reader / write/read device for a tag that happens to be in the field • With MOBY I, generally only one tag can be processed
E3h	FEh	02h	**In the case of filehandler:** • The file accessed by a command (e.g. WRITE) is not available in the directory. The file has to be created with CREATE. • Check filename (possibly not in ASCII format)

Byte			Description
0	1	2	
E3h	FEh	03h	**In the case of filehandler:** • The type of tag present before the reader / write/read device does not match the ECC mode set. The MDS must be reformatted in accordance with the desired ECC mode. • The tag is not a filehandler tag; format tag
E3h	FEh	04h	**In the case of filehandler:** There is no longer a directory entry free. The file specified in the CREATE command can no longer be created.
E3h	FEh	05h	**In the case of filehandler:** The file specified in the CREATE command already exists in the directory (double names not permissible).
E3h	FEh	06h	**In the case of filehandler:** • The tag has been locked with the help of the COVER command. A write command (e.g. UPDATE, CREATE) must not disturb the tag layout and is thus rejected. • Write access (WRITE, UPDATE or DELETE) to a file that must not be modified (protected with an appropriate attribute)
E4h	FEh	01h	**In the case of normal addressing:** Short circuit or overload of the 24 V outputs (DQ, error code, presence) • The affected output is turned off. • All outputs are turned off when total overload occurs. • A reset can only be performed by turning the 24 V voltage off and on again. • Then start init_run
			In the case of filehandler: Fault in voltage suppply
E4h	FEh	03h	**In the case of normal addressing:** • Error in the connection to the reader / wrie/read device; reader / write/read device does not answer. – Cable between communication module/ASM and reader/SLG is wired incorrectly or cable break. – The 24 V supply voltage is not connected or is not on or has failed briefly. – Automatic fuse on the communication module / ASM has blown. – Hardware defect – Another reader / write/read device is in the vicinity and is active. – Interference on DI/DO, write/read device or PROFIBUS/PROFINET line – Execute init_run after error correction • The antenna of the reader/SLG is turned off. A tag command to the communication module/ASM was started in this state. – Turn on the antenna with the command "antenna on/off." – The antenna is turned on (off) and has received an additional turn-on (turn-off) command.

Byte			Description
0	1	2	
			In the case of filehandler: • Error in the connection to the reader / write/read device: This error is not indicated when starting system commands (WRITE-CONFIG, NEXT, DEV-STATUS, MEM-STATUS). – Cable between communication module/ASM and reader/SLG is wired incorrectly or cable break. – The 24 V supply voltage is not connected or it has been switched off. – Automatic fuse on the communication module / ASM has blown. – Hardware defect • Antenna not switched on, or SET-ANT = ON with already switched-on antenna – User error; note order of commands
E4h	FEh	04h	**In the case of normal addressing:** Communication module/ASM/SLG U does not have enough buffer to store the command intermediately.
			In the case of filehandler: Buffer overflow in the driver of the communication module / reader; system-internal error • Execute init_run of the communication module/ASM
E4h	FEh	06h	**In the case of normal addressing:** NEXT-command not possible or not permitted • Communication module / ASM works without tag control (MDS_IO_CONTROL = 0.1). • Communication module / ASM has already received a NEXT command. • Communication module / ASM or reader / write/read device doesn't recognize NEXT command.
			In the case of filehandler: The started command is not permissible (not defined). • Correct the command parameter in the call
E4h	FEh	07h	**In the case of normal addressing:** Startup message from communication module / ASM. The communication module / ASM was off and has not yet received a WRITE-CONFIG command. • Execute an init_run • The same physical ASM channel is used in two (or more) UDT structures. Check ASM_address and ASM_channel in all UDT structures.
			In the case of filehandler: Unmotivated startup message of the tag driver in the communication module / ASM • Execute init_run of the communication module/ASM

Byte			Description
0	1	2	
E4h	FEh	8Ch	**In the case of normal addressing:** • Communication error between RFID standard profile and communication module / ASM. Handshake error. – UDT of this communication module / ASM is overwritten by other program sections – Check parameters of the communication module / ASM in the UDT – Check the RFID standard profile command which caused this error. – Start init_run command after error correction. • PROFIBUS DP/PROFINET error occurred This error is only indicated when access monitoring has been enabled in the PROFIBUS configuration. – PROFIBUS DP/PROFINET bus connection was interrupted (wire break on the bus; bus connector on the communication module / ASM was briefly unplugged) – PROFIBUS DP/PROFINET master no longer addresses the communication module / ASM – Execute an init_run – The communication module / ASM has detected a message frame interruption on the bus. The PROFIBUS or PROFINET may have been reconfigured (e.g. with HW Config)
			In the case of filehandler: Checksum error on the communication module / ASM
E4h	FEh	8Dh	**In the case of normal addressing:** • Internal communication error of the communication module/ASM – Connector contact problem on the communication module / ASM – Hardware of the communication module / ASM has a defect; → Send in communication module / ASM for repair – Start init_run command after error correction. • Internal monitoring error of the communication module/ASM – Program execution error on the communication module / ASM – Turn power of the communication module / ASM off and on again. – Start init_run command after error correction. – MOBY U: Watchdog errror on the reader.
			In the case of filehandler: • Inernal driver error of the communication module / ASM; communication fault between filehandler and tag driver (AB byte) – Execute init_run of the communication module/ASM • The filehandler is not working correctly. – Check command structure or command sequence – The hardware of the communication modlue / ASM (firmware) has a defect • Operating system error (AMOS mailbox) – Execute init_run of the communication module/ASM • Operating system error; watchdog error in the communication module / ASM or reader – Execute init_run of the communication module/ASM
E4h	FEh	8Eh	**In the case of normal addressing:** Running command canceled by WRITE-CONFIG (init_run or cancel) or bus connector removed • Communication with the tag was terminated by init_run. • This error can only be reported on init_run or cancel

Byte			Description
0	1	2	
			In the case of filehandler: Communication error between filehandler and tag drivere; the tag driver reports cancellation of WRITE-CONFIG although the filehandler does not process WRITE-CONFIG commands • Execute init_run of the communication module/ASM
E5h	FEh	01h	**With normal addressing and filehandler:** Incorrect sequence number order (SN) in the communication module / ASM
E5h	FEh	04h	**In the case of normal addressing:** Invalid data block number (DBN) in the communication module / ASM
			In the case of filehandler: The message frame control parameters (DBN or command code) do not come in the correct order. Two or more message frames are written to the same communication module / ASM. • Check the ASM_address and ASM_channel parameter settings. • Do not execute the command start with the function "Modify variable"
E5h	FEh	06h	**In the case of normal addressing:** Invalid data block length (DBL) in the communication module / ASM
			In the case of filehandler: The data block from the reader/SLG is too long and cannot be transferred over PROFIBUS or PROFINET. • The "block length" parameter in the WRITE-CONFIG command is too long (RFID standard profile error or user error) • Program execution error in the reader / write/read device • Execute complete restart of the communication module / ASM and restart the command
E5h	FEh	08h	**In the case of normal addressing:** Previous command is active or buffer overflow The user sent a new command to the ASM communication module although the last command was still active. • Active command can only be terminated with an init_run. • Before a new command can be started, the DONE bit must be 1; exception: init_run. • Two RFID standard profile calls were parameterized with the same "ASM_address" and "ASM_channel" parameters. • Two RFID standard profile calls are using the same pointer. • Start init_run command after error correction. • When command repetition (e.g., fixed code tag) is used, no data are fetched from the tag. The data buffer on the communication module / ASM has overflowed. Tag data have been lost.
			In the case of filehandler: • The filehandler is currently executing a command. A WRITE-CONFIG must be executed urgently. • Communication error; driver is active while a new command is sent – Check command sequences in the application – Execute complete restart of the communication module / ASM

Byte			Description
0	1	2	
E6h	FEh	01h	**In the case of normal addressing:** Unknown command RFID standard profile transfers an uninterpretable command to the communication module / ASM. • "PIB_COMMAND" contains invalid command parameters • "PIB_COMMAND" has been overwritten by the user • The tag has reported an address error.
			In the case of filehandler: Communication error between filehandler and tag driver; the driver on the communication module / ASM does not recognize the command from the filehandler. • Execute init_run of the communication module/ASM
E6h	FEh	02h	**In the case of normal addressing:** Invalid command index CI
			In the case of filehandler: Invalid command index CI
E6h	FEh	03h	**In the case of normal addressing:** • Incorrect parameterization of the communication module / ASM – Check INPUT parameters in the FB 101/116/132 – Check parameters in HW Config – WRITE-CONFIG command has been incorrectly parameterized – After a startup, the communication module / ASM has still not received an init_run. • The command cannot be executed with the parameterization of the communication module/ASM on PROFIBUS/PROFINET. – Length of the input/output areas too small for the cyclic I/O word. Did you use the right GSD file? – Command (e.g. READ) issued with too much user data • Error when processing the command – The data in "PIB_COMMAND" contain errors (e.g. WRITE command with length = 0); check "PIB_COMMAND" and execute init_run – Communication module / ASM hardware defective: Communication module / ASM receives incorrect data during init_run. – AB byte does not comply with the user data length

Byte			Description
0	1	2	
			In the case of filehandler: • Impermissible receive ID – System error; cannot occur with RFID standard profile • The commands FORMAT, CREATE, WRITE, ATTRIB, UPDATE, COVER, QUEUE-READ or QUEUE-WRITE have been transmitted with impermissible parameters. – FORMAT with impermissible tag name or tag type – CREATE with impermissible filename – WRITE/UPDATE with length 0 (DLNG=0) – Attribute impermissible – QUEUE-WRITE or QUEUE-READ with impermissible option or number of files – COVER with impermisisble user (only 0 or 1 permissible) • The system data transferred by the LOAD command are incorrect. – DLNG in LOAD incorrectly parameterized – Incorrect data block specified or wrongly parameterized – MOVE command incorrectly executed; DIR + FAT on the tag does not match checksum • The MOVE command cannot be executed. The checksum does not match DIR + FAT. The data memory has apparently exited the transmission window during execution of system operations (e.g. write DIR + FAT). • MOBY U: The LOAD and MOVE commands are not supported • A WRITE-CONFIG has been executed by the RFID standard profile with impermissible parameters. The cause of the error is located in the user program. – Check INPUT parameters of the UDT call • COVER command: The tag name specified in the command does not agree with the actual tag name. • QUEUE-READ command: Specified file length less than actual file length • QUEUE-READ command: The skip calculated by the filehandler is greater than 0FFFh • QUEUE-WRITE command has been incorrectly parameterized (DAT_DB_number/DAT_DB_address or length) Option 0000h: The file entry parameterized in DAT_DB_number with the number xxx or xxx + 1 contains error(s). Counting of the file entries in DAT_DB_number begins with 1. Option 0001h: The file entry parameterized in DAT_DB_number with the number xxx or xxx + 1 contains a filename that already exists on teh tag. Counting of the file entries in DAT_DB_number begins with 1. **Note:** The file entries are incremented in decimal.

Byte			Description
0	1	2	
E6h	FEh	04h	**In the case of normal addressing:** • Presence error: An tag has passed by a reader / write/read device without being processed by a command. • Processing error: Command processing of a tag (read and/or write) has not been concluded with NEXT. This error message is not reported immediately. Instead, the communication modle / ASM waits for the next command (read, write, NEXT). This command is immediately replied to with this error. This means that a read or write command is not processed. The next command is executed normally by the communication module / ASM again. • An init_run from the RFID standard profile also resets this error state. • Bit 2 is set in parameter OPT1 and no tag is in the transmission window.
			In the case of filehandler: • MDS_IO_CONTROL = 1: – A command has been started but there is no tag in the transmission window of the reader / write/read device. – The dialog battery on the MDS 507 has discharged (the LR_bat bit did not have to be set; check battery voltage) • MDS_IO_CONTROL = 0: – The old/current tag has exitted the transmission window and the next/new tag has entered the transmission window. A command has been started (not NEXT). This command refers to the new tag, but the old/current tag has not yet been terminated with NEXT. – A new tag enters the transmission window of the reader / write/read device and then exits again without a command having been executed with this tag. ("Slip-through of the MDS")
E6h	FEh	05h	**In the case of normal addressing:** An error has occurrerd that makes a WRITE-CONFIG with Config = 3 necessary. • Error(s) in the WRITE-CONFIG command • Start init_run command after error correction. • Check parameters ASM_address, ASM_channel, and MOBY_mode.
			In the case of filehandler: The filehandler will now only accept a WRITE-CONFIG command mit Config = 3. • The filehandler has not yet been initialized by an init_run • The state can only be resolved by an init_run
Fxh	FEh	xxh	An FxFExxh error is identical with the corresponding ExFExxh error (see ExFExxh). Byte 3 additionally contains warning information.

Errors of the RFID standard profile

The causes of these errors can include:

- Incorrect parameterization of the RFID standard profile function.

Byte 3 of the STATUS is not relevant for the error messages.

Table 5- 4 Error messages from the RFID standard profile via the STATUS output parameter

Byte			Description
0	1	2	
E5h	FEh	02h	Incorrect sequence number order (SN) in the RFID standard profile
E5h	FEh	05h	Invalid data block number (DBN) in the RFID standard profile
E5h	FEh	07h	Invalid data block number (DBL) in the RFID standard profile
E5h	FEh	09h	The communication module / ASM executes a hardware reset (INIT_ACTIVE set to "1"). INIT is expected from the RFID standard profile (bit 15 in the cyclic control word).
E5h	FEh	0Ah	The "CMD" command code and the relevant acknowledgement do not match. This can be a software error or synchronization error that cannot occur in normal operation.
E5h	FEh	0Bh	Incorrect sequence of acknowledgement message frames (TDB / DBN)
E5h	FEh	0Ch	Synchronization error (incorrect increment of AC_H / AC_L and CC_H / CC_L in the cyclic control word). INIT had to be executed
E6h	FEh	06h	INIT timer has expired
E7h	FEh	01h	In this state, only the INIT command is permitted.
E7h	FEh	02h	The CMD command code is not permissible.
E7h	FEh	03h	The length parameter of the command is too long. It does not match the global data reserved within the send data buffer (TXBUF).
E7h	FEh	04h	Overeflow of the receive data buffer (RXBUF) More data were received than memory available in the RXBUF.
E7h	FEh	05h	This error tells the user that only an INIT command is permissible as the next command. All other commands are rejected.
E7h	FEh	06h	Wrong index (outside range of 101 to 108)
E7h	FEh	07h	Communication module / ASM does not respond to INIT (INIT_ACTIVE is expected in cyclic status message).
E7h	FEh	08h	Time violation during INIT (60 seconds in accordance with TC3WG9)
E7h	FEh	09h	Command repetition is not supported.
Fxh	FEh	xxh	An FxFExxh error is identical with the corresponding ExFExxh error (see ExFExxh). Byte 3 additionally contains warning information.

PROFIBUS DP-V1/PROFINET error

The transport layer of PROFIBUS or PROFINET is signaling an error. A PROFIBUS tracer and a PROFIBUS tester (BT 200; Order No. 6ES7181-0AA00-0AA0) are invaluable tools for accurate troubleshooting. For PROFINET, the open source software "Wireshark" can be used. The PROFIBUS or PROFINET system diagnostics can provide further information about the cause of the error.

Table 5- 5 Error messages from PROFIBUS DP-V1/PROFINET via the STATUS output parameter

Byte			Description
0	1	2	
Cxh	80h	0Ah	Communication module / ASM is not ready (temporary message). • This message is given to a user who is not using the RFID standard profile and is polling the communication modules / ASMs acyclically, one after the other, very quickly.
Cxh	8xh	7Fh	Internal error on parameter x. Cannot be remedied by the user.
Cxh	8xh	22h	Area length error on reading a parameter. This error code indicates that parameter x is partially or completely outside the operand area or the length of a bit array for an ANY parameter is not divisible by 8.
Cxh	8xh	23h	Area length error on writing a parameter. This error code indicates that parameter x is partially or completely outside the operand area or the length of a bit array for an ANY parameter is not divisible by 8.
Cxh	8xh	24h	Area error on reading a parameter. This error code indicates that parameter x is within an area not allowed for the system function.
Cxh	8xh	25h	Area error on writing a parameter. This error code indicates that parameter x is within an area not allowed for the system function.
Cxh	8xh	26h	Parameter contains a time cell number which is too high.
Cxh	8xh	27h	Parameter contains a counter cell number which is too high.
Cxh	8xh	28h	Alignment error on reading a parameter. The reference to parameter x is an operand whose bit address is not equal to 0.
Cxh	8xh	29h	Alignment error on writing a parameter. The reference to parameter x is an operand whose bit address is not equal to 0.
Cxh	8xh	30h	Parameter is in write-protected global DB.
Cxh	8xh	31h	Parameter is in write-protected instance DB.
Cxh	8xh	32h	Parameter contains DB number which is too high.
Cxh	8xh	34h	Parameter contains FC number which is too high.
Cxh	8xh	35h	Parameter contains FB number which is too high.
Cxh	8xh	3Ah	Parameter contains the number of a DB that is not loaded.
Cxh	8xh	3Ch	Parameter contains the number of an FC that is not loaded.
Cxh	8xh	3Eh	Parameter contains the number of an FB that is not loaded.
Cxh	8xh	42h	An access error has occurred while the system wanted to read out a parameter from the I/O area of the inputs.
Cxh	8xh	43h	An access error has occurred while the system wanted to write a parameter to the I/O area of the outputs.
Cxh	8xh	44h	Error on n-th (n > 1) read access after occurrence of an error.
Cxh	8xh	45h	Error on n-th (n > 1) write access after occurrence of an error.
Cxh	80h	90h	Specified logical base address is invalid: No assignment in SDB1/SDB2x exists, or it is not a base address.

Byte			Description
0	1	2	
Cxh	80h	92h	A type other than BYTE has been specified in an ANY reference.
Cxh	80h	93h	The area identifier contained in the configuration (SDB1, SDB2x) of the logical address is not permitted for these SFCs. Permitted: • 0 = S7-400 • 1 = S7-300 • 2, 7 = DP modules
Cxh	80h	A0h	Negative acknowledgment while reading from module; RFID standard profile fetches acknowledgment although no acknowledgment is ready A user who is not using the RFID standard profile would like to fetch DS 101 (or DS 102 to104) although no acknowledgment is available. • Perform an init_run for new synchronization between the communication module / ASM and the application.
Cxh	80h	A1h	Negative acknowledgment while writing to the module; RFID standard profile sends command although communication module / ASM is unable to receive a command.
Cxh	80h	A2h	DP protocol error in layer 2 could be a hardware defect.
Cxh	80h	A3h	DP protocol error in Direct-Data-Link-Mapper or User-Interface/User could be a hardware defect.
Cxh	80h	B0h	• SFC not possible for module type • Data record unknown to module • Data record number ≥ 241 is not allowed. • Data records 0 and 1 are not permitted for SFC 58 "WR_REC."
Cxh	80h	B1h	The length specified in the RECORD parameter is wrong.
Cxh	80h	B2h	The configured slot is not occupied.
Cxh	80h	B3h	Actual module type is not the module type specified in SDB1.
Cxh	80h	C0h	• RDREC: The module has record, but it doesn't have any read data. • WRREC: Communication module / ASM is not ready to receive new data – Wait until the cyclic counter has been incremented
Cxh	80h	C1h	The data of the preceding write job on the module for the same data record have not yet been processed by the module.
Cxh	80h	C2h	The module is currently processing the maximum possible number of jobs for a CPU.
Cxh	80h	C3h	Required resources (memory, etc.) are currently in use. This error is not reported by the RFID standard profile. If this error occurs, the RFID standard profile waits until the system is able to provide resources again.
Cxh	80h	C4h	Communication Errors • Parity error • SW ready not set • Error in block length management • Checksum error on CPU side Checksum error on module side
Cxh	80h	C5h	Distributed I/O not available
Dxh	8xh	xxh	A Dx8xxxh error is identical with the corresponding Cx8xxxh error (see Cx8xxxh). Byte 3 additionally contains warning information.

5.3 Warnings

Byte 3 of the STATUS output parameter indicates warnings if byte 0 of the STATUS (function numbers) has the value Fxh or Dxh.
The table below shows the possible warnings when working with the RFID standard profile.

Table 5- 6 Possible warnings when working with the RFID standard profile

Byte 0 … 2	Byte 3	Meaning
FxFExxh	Bit 0	Bit is always "0"
	Bit 1	ECC correction has been carried out
	Bit 2	Low Battery if an RFID reader is used
	Bit 3	Auxiliary battery is low
	Bit 4	Tag was out of the field
	Bit 5	READ command has length "0" or file is smaller than read length
	Bit 6	Tag with new directory has come into the field
	Bit 7	Reserved

Application examples 6

6.1 Processing of tags

Tag types

Tags (mobile data memories or transponders) with different storage capacities are available. The following table specifies the memory capacities currently available.

Table 6- 1 Available memory capacities

Memory capacity	Memory type	RFID system	Tag type
2 (1.7) KB	RAM	MOBY I	e.g. MDS 302
8 (7) KB	EEPROM / FRAM	MOBY I	e.g. MDS 413E
32 (28) KB	FRAM	MOBY I	e.g. MDS 514
752 bytes	EEPROM	MOBY E	e.g. MDS E600
2KB	RAM, 16 bytes OTP	MOBY U	e.g. MDS U313
32 KB	RAM, 16 bytes OTP	MOBY U	e.g. MDS U524
44 bytes	EEPROM	MOBY D	e.g. MDS D139 / I-Code 1
112 bytes	EEPROM	MOBY D	I-Code SLI
256 bytes	EEPROM	MOBY D	Tag-it HF-I
1000 bytes	EEPROM	MOBY D	my-d
20 bytes	EEPROM	RF300	RF320T
8 KB	FRAM	RF300	e.g. RF340T
32 KB	FRAM	RF300	e.g. RF350T
64 KB	FRAM	RF300	e.g. RF350T (64K)

Addressing

The tags are addressed linearly from address 0000 (or the specified starting address) to the end address. The communication module or reader automatically recognizes the size of the memory on the tag. If the end address on the tag is exceeded, the user receives an error message.

The next table shows the address space of the individual tag versions. The StartAddress and Length parameters in the command structure must be set in accordance with this address space.

Address space of MDS versions for MOBY I, E, U and D

System	Addressing	16-Bit Hexadecimal Number		Integer number	
MOBY I	**2 KB data memory with RAM**				
	Start address	0000	0000 (with ECC)	+0	+0 (with ECC)
	End address	07FC	06F1 (with ECC)	+2044	+1777 (with ECC)
	8 KB data memory with EEPROM/RAM/FRAM				
	Start address	0000	0000 (with ECC)	+0	+0 (with ECC)
	End address	1FFC	1BF1 (with ECC)	+8188	+7153 (with ECC)
	32 KB data memory with RAM/FRAM				
	Start address	0000	0000 (with ECC)	+0	+0 (with ECC)
	End address	7FFC	6FF1 (with ECC)	+32764	+28657 (with ECC)
MOBY E	**752 byte data memory with EEPROM**				
	Start address	0000		+0	
	End address	02EF		+751	
	ID no.: (fixed-coded; can only be read as a whole)				
	Start address	1FF0		+8176	
	Length	0004		+4	
MOBY U	**2 KB data memory**				
	Start address	0000		+0	
	End address	07FF		+2047	
	Read OTP memory (write access only possible once) The OTP memory of MOBY U can only be processed completely, i.e. the start address must always be specified with value FFF0h and the length with value 10h.				
	Start address	FFF0		-16	
	Length	10		+16	
	ID no.: (four fixed-coded bytes; can only be read with the MEM-STATUS command)				
	32 KB data memory				
	Start address	0000		+0	
	End address	7FFF		+32767	
	Read OTP memory (write access only possible once) The OTP memory of MOBY U can only be processed completely, i.e. the start address must always be specified with value FFF0h and the length with value 10h.				
	Start address	FFF0		-16	
	Length	10		+16	
	ID no.: (four fixed-coded bytes; can only be read with the MEM-STATUS command)				
MOBY D	**MDS D139 (I-Code 1; 44 bytes)**				
	Start address	0000		+0	
	End address	002B		+43	
	ID no.: (fixed-coded; can only be read as a whole)				
	Start address	FFF0		-16	
	Length	0008		+8	

System	Addressing	16-Bit Hexadecimal Number	Integer number
	ISO-MDS (I-Code SLI; 112 bytes)		
	Start address	0000	+0
	End address	006F	+111
	ID no.: (fixed-coded; can only be read as a whole)		
	Start address	FFF0	-16
	Length	0008	+8
	ISO MDS (Tag-it HF-I; 256 bytes)		
	Start address	0000	+0
	End address	00FF	+255
	ID no.: (fixed-coded; can only be read as a whole)		
	Start address	FFF0	-16
	Length	0008	+8
	ISO MDS (my-d SRF55V10P; 1000 bytes)		
	Start address	0018	+24
	End address	03FF	+1023
	ID no.: (fixed-coded; can only be read as a whole)		
	Start address	FFF0	-16
	Length	0008	+8

Address space of the transponder versions for RF300

System	Addressing	16-Bit Hexadecimal Number	Integer number
RF300	**20 bytes of data memory (EEPROM)**		
	R/W or OTP memory (EEPROM) (The EEPROM user memory for RF300 can be used either as R/W memory or as an OTP memory (see RF 300 system manual))		
	Start address End address	FF00 FF13	-256 -237
	ID no.: (fixed-coded; can only be output as a whole)		
	Start address Length	FFF0 0008	-16 +8
	8 KB data memory (FRAM/EEPROM)		
	R/W or OTP memory (EEPROM) (The EEPROM user memory for RF300 can be used either as R/W memory or as an OTP memory (see RF 300 system manual))		
	Start address End address	FF00 FF13	-256 -237
	R / W memory (FRAM)		
	Start address End address	0000 1FFC	+0 +8188
	ID no.: (fixed-coded, can only be read as a whole)		
	Start address Length	FFF0 0008	-16 +8
	32 KB data memory (FRAM/EEPROM)		
	R/W or OTP memory (EEPROM) (The EEPROM user memory for RF300 can be used either as R/W memory or as an OTP memory (see RF 300 system manual))		
	Start address End address	FF00 FF13	-256 -237
	R/W memory (FRAM)		
	Start address End address	0000 7FFC	+0 +32764
	ID no.: (fixed-coded; can only be output as a whole)		
	Start address Length	FFF0 0008	-16 +8
RF300	**64 KB data memory (FRAM/EEPROM)**		
	R/W or OTP memory (EEPROM) (The EEPROM user memory for RF300 can be used either as R / W memory or as an OTP memory (see RF300 system manual))		
	Start address End address	FF00 FF13	-256 -237
	R/W memory (FRAM)		
	Start address End address	0000 FEFC	+0 –

System	Addressing	16-Bit Hexadecimal Number	Integer number
	ID no.: (fixed-coded; can only be output as a whole)		
	Start address	FFF0	-16
	Length	0008	+8

RF300: Address mapping on the transponder

R / W EEPROM memory and OTP memory is only available once on the transponder.

The following table shows the mapping of addresses on the transponder.

Data can be read via the R / W address or the OTP address.

R / W EEPROM		Write OTP once	
Address (hex)	Length	Address (hex)	Length
FF00	1 … 20	FF80	4, 8, 12, 16, 20
FF01	1 … 19		
FF02	1 … 18		
FF03	1 … 17		
FF04	1 … 16	FF84	4, 8, 12, 16
FF05	1 … 15		
FF06	1 … 14		
FF07	1 … 13		
FF08	1 … 12	FF88	4, 8, 12
FF09	1 … 11		
FF0A	1 … 10		
FF0B	1 … 9		
FF0C	1 … 8	FF8C	4, 8
FF0D	1 … 7		
FF0E	1 … 6		
FF0F	1 … 5		
FF10	1 … 4	FF90	4
FF11	1 … 3		
FF12	1 … 2		
FF13	1		

NOTICE

Write access to addresses starting at FF80h to FF93h activates the write protection (OTP function) on the EEPROM user memory. This operation is not reversible. Write protection must always be activated in ascending order without gaps and starting at address FF80h.

6.2 Calling FB 101 in STL

Requirements

We assume that an ASM 456 logical base address 256 has been configured in HW Config. A reader is connected to channel 2 of the ASM 456.
Channel 2 will therefore be accessed with address 256 in the example below.

Create data block (DB "Channel2")

After you have copied the necessary blocks into the project, generate a data block (DB "Channel2") in which the command area (with the help of UDT 1) and the data buffer are created.

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	CMD	STRUCT		Befehlsbereich
+0.0	Command	ARRAY[1..10]		
*38.0		"PIB_COMMAND"		
=380.0		END_STRUCT		
+380.0	RXBUF	ARRAY[1..1024]		
*1.0		BYTE		
+1404.0	TXBUF	ARRAY[1..1024]		
*1.0		BYTE		
=2428.0		END_STRUCT		

Figure 6-1 Creating the command area and the data buffer in the DB "Channel2"

Call FB 101

Call FB 101 in the code area of OB 1 using the Call command and initialize the variables.

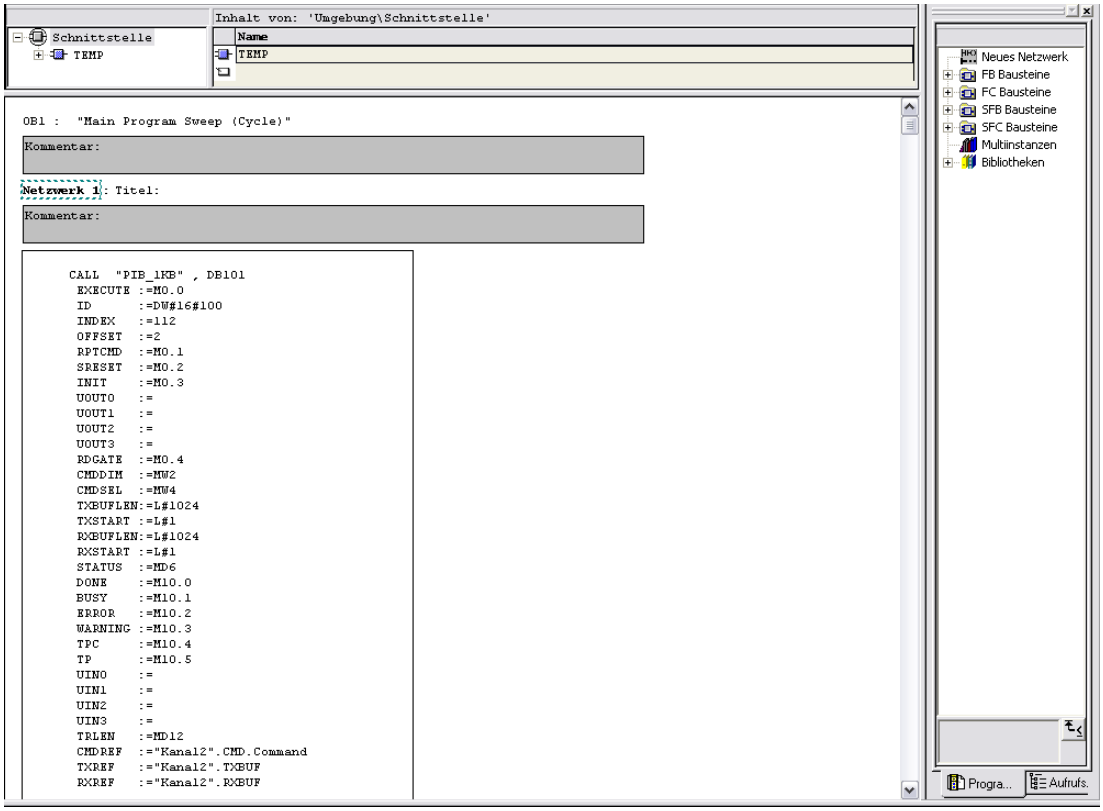


Figure 6-2 Initialize the variables of FB 101 (STL)

Assigning symbolic names to bit memories

Assign symbolic names to the bit memories (for easier handling).

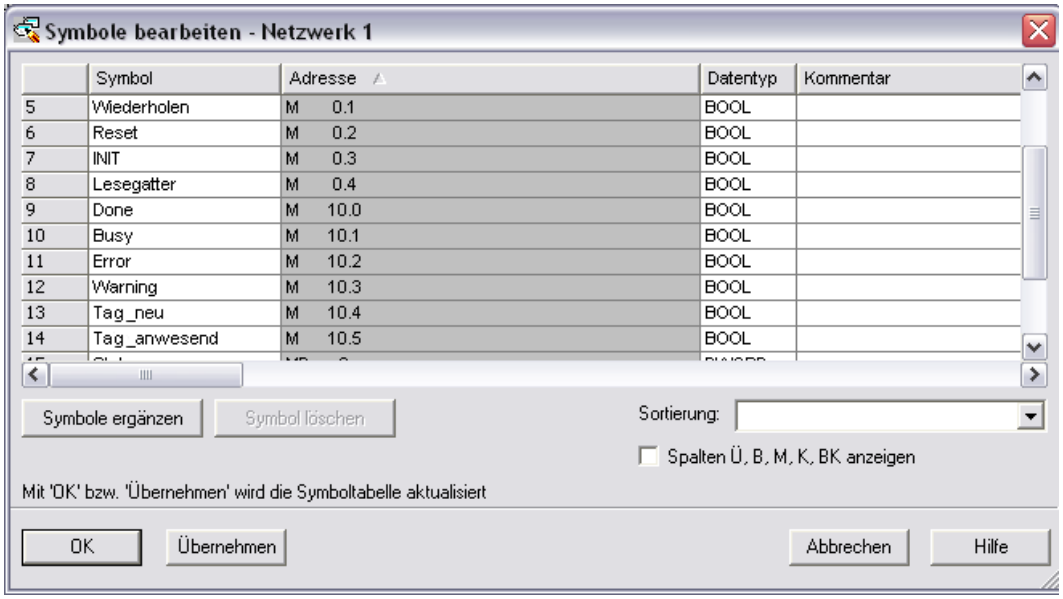


Figure 6-3 Assignment of symbolic names to bit memories (STL)

Note

After assigning symbolic names, you must adapt the data types. If, for example, an integer variable is required, *STEP 7* will not accept a variable of type WORD.

Saving and transferring a project

Save the entire project and transfer it to the CPU.

Control and test the program

You can control and test the program with the help of a variable table.

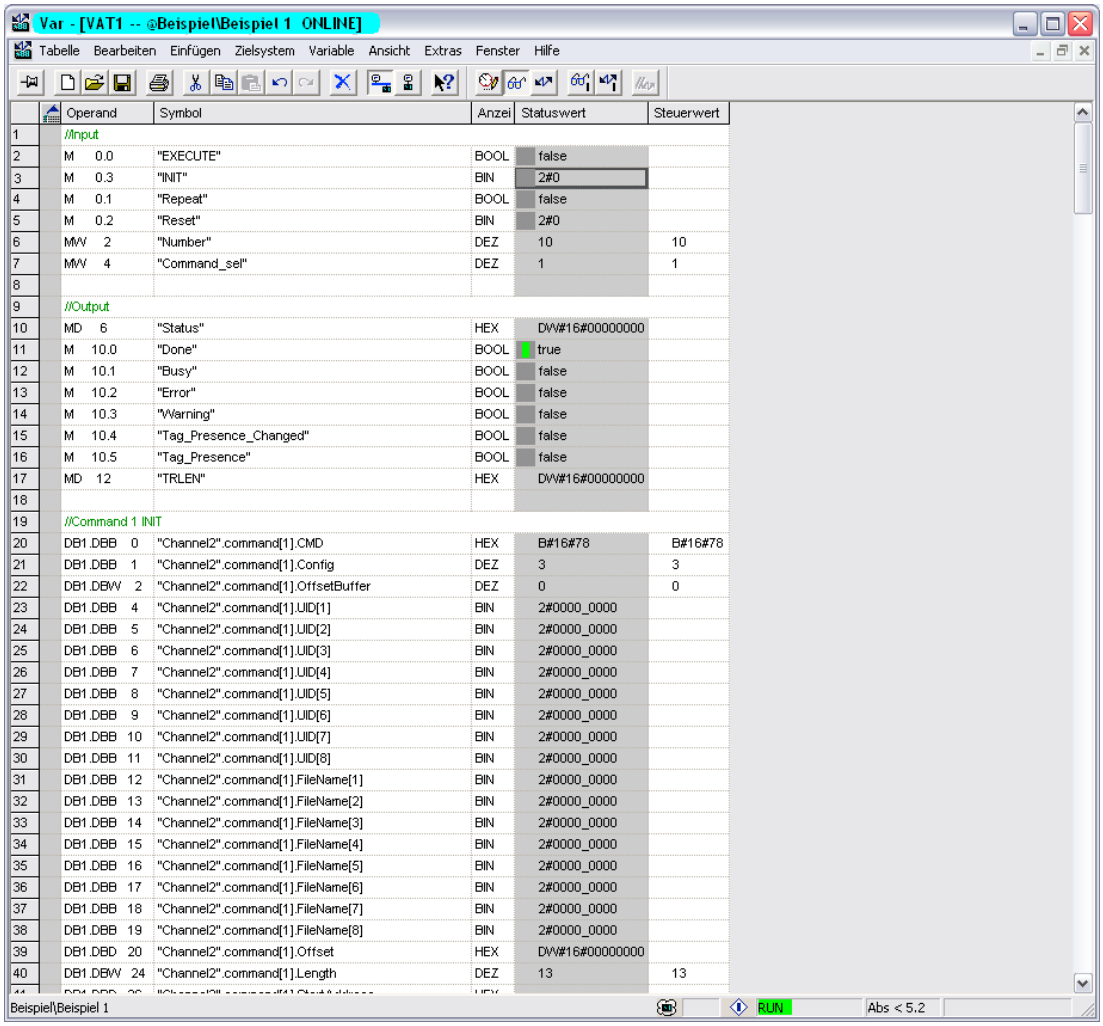


Figure 6-4 Variable table for program control (STL)

Note

First write command 1 for initialization (see WRITE-CONFIG command) and the receive data buffer. Then execute the initialization by setting and then resetting INIT.

See also

WRITE-CONFIG (also used for INIT) (Page 56)

6.3 Calling FB 101 in SCL

Requirements

We assume that an ASM 456 logical base address 260 has been configured in HW Config. A reader is connected to channel 1 of the ASM 456.
Channel 1 will therefore be accessed with address 260 in the example below.

Fill in symbol table

After you have copied the necessary SCL blocks into the project **Sources Folder** fill in the symbol table as shown in the figure below.

Beispiel 2 (Symbole) -- Beispiel

	Status	Symbol	Adresse	Datentyp	Kommentar
1		DP_GETIOSUB	FB 1	FB 1	
2		DP_SETIOSUB	FB 2	FB 2	
3		PIB_PUT_WORD	FB 3	FB 3	
4		PIB_PUT_DWORD	FB 4	FB 4	
5		PIB_PUT_8BYTE	FB 5	FB 5	
6		PIB_1KB	FB 101	FB 101	
7		PIB_16KB	FB 116	FB 116	
8		PIB_32KB	FB 132	FB 132	
9		emulated_ID	FC 1	FC 1	
10		PIB_STATUS	FC 2	FC 2	
11		PIB_GET_INT	FC 3	FC 3	
12		PIB_GET_DWORD	FC 4	FC 4	
13		TON	SFB 4	SFB 4	Generate an On Delay
14		DP_RDREC	SFB 52	SFB 52	
15		DP_WRREC	SFB 53	SFB 53	
16		GADR_LGC	SFC 5	SFC 5	Query Logical Address of a Channel
17		PIB_COMMAND	UDT 1	UDT 1	
18					

Figure 6-5 Symbol table for SCL blocks

Compile sources

Compile all sources.

Create new SCL source

1. Create a new SCL source in the sources folder and open it.

2. Create the data block DB 1 and create the command area (with the help of UDT 1) and the data buffer there:
   – Create the block via the menu point **Insert > Block Template > DB**.
   – Assign this data block the number 1.
   – Update the command lines as shown in the figure below.

```
DATA_BLOCK DB1
//
// Baustein-Kommentar ...
//
  STRUCT
  command : ARRAY[1..10] OF UDT1;
  RXBUF : ARRAY[1..1024] OF BYTE;
  TXBUF : ARRAY[1..1024] OF BYTE;
  END_STRUCT
BEGIN

END_DATA_BLOCK
```

Figure 6-6 Command lines for structuring DB 1

Create OB1

1. Create the block via the menu point **Insert > Block Template > OB**.
2. Assign this organization block the number 1.
3. Start the code section after the lilne "//Statements" with the BEGIN command (see figure below).

```
ORGANIZATION_BLOCK OB1

VAR_TEMP
  // reserviert
  info : ARRAY[0..19] OF BYTE;
  // temporäre Variablen

END_VAR

  // Anweisungen
BEGIN


  ;
END_ORGANIZATION_BLOCK
```

Figure 6-7 Structure of OB1

Insert block call

Insert FB 101 into OB 1 via the menu point **Insert > Block Call**.
Select block FB 101 from the project and confirm with OK.

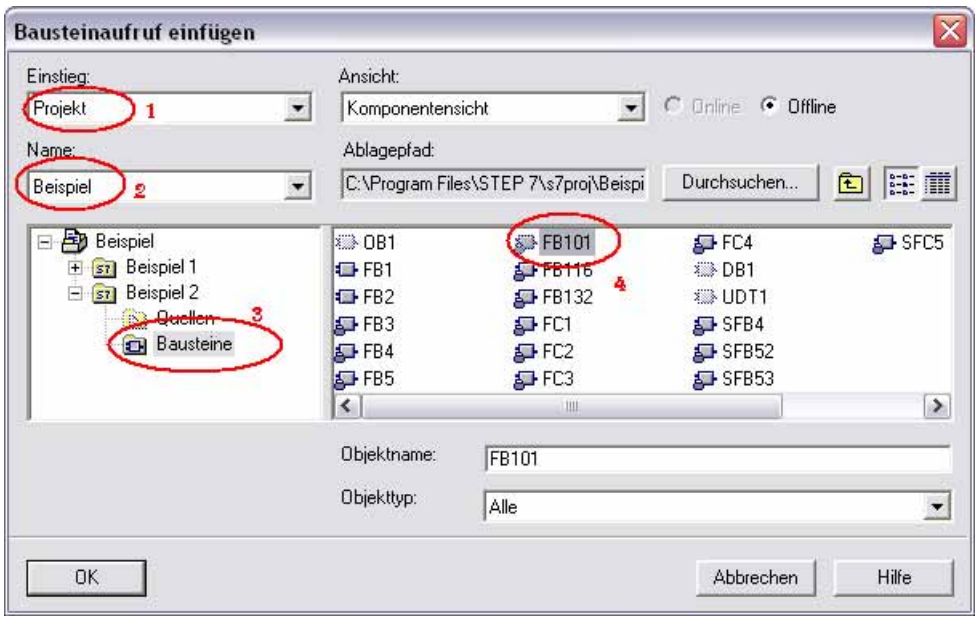


Figure 6-8 Insert FB 101 into OB 1

Assign block number and initialize variables

1. Give the instance data block a number (red marking in the figure below).

2. Initialize the variables as shown in the figure below.
   The variables you do not initialize must be deleted, since otherwise, the source cannot be compiled (blue marking in the figure below)

```
BEGIN

PIB_1KB.DB101(EXECUTE :=E0.0  // IN: BOOL
              ,ID := 104 // IN: DWORD // h104 entspricht 260 dezimal
              ,INDEX := 111 // IN: INT
              ,OFFSET := 0 // IN: INT
              ,RPTCMD := E0.1 // IN: BOOL
              ,SRESET := E0.2 // IN: BOOL
              ,INIT := E0.3 // IN: BOOL
              ,UOUT0 :=  // IN: BOOL
              ,UOUT1 :=  // IN: BOOL
              ,UOUT2 :=  // IN: BOOL
              ,UOUT3 :=  // IN: BOOL
              ,RDGATE := E0.4 // IN: BOOL
              ,CMDDIM := MW2 // IN: INT
              ,CMDSEL := MW4 // IN: INT
              ,TXBUFLEN := 1024 // IN: DINT
              ,TXSTART := 1 // IN: DINT
              ,RXBUFLEN := 1024 // IN: DINT
              ,RXSTART := 1 // IN: DINT
              ,CMDREF := DB1.command // INOUT: ARRAY
              ,TXREF := DB1.RXBUF // INOUT: ARRAY
              ,RXREF := DB1.TXBUF // INOUT: ARRAY
              );
  MD6 := DB101.STATUS; // OUT: DWORD
  M10.0 := DB101.DONE; // OUT: BOOL
  M10.1 := DB101.BUSY; // OUT: BOOL
  M10.2 := DB101.ERROR; // OUT: BOOL
  M10.3 := DB101.WARNING; // OUT: BOOL
  M10.4 := DB101.TPC; // OUT: BOOL
  M10.5 := DB101.TP; // OUT: BOOL
   := DB101.UIN0; // OUT: BOOL
   := DB101.UIN1; // OUT: BOOL
   := DB101.UIN2; // OUT: BOOL
   := DB101.UIN3; // OUT: BOOL
   := DB101.TRLEN; // OUT: DINT

  ;
END_ORGANIZATION_BLOCK
```

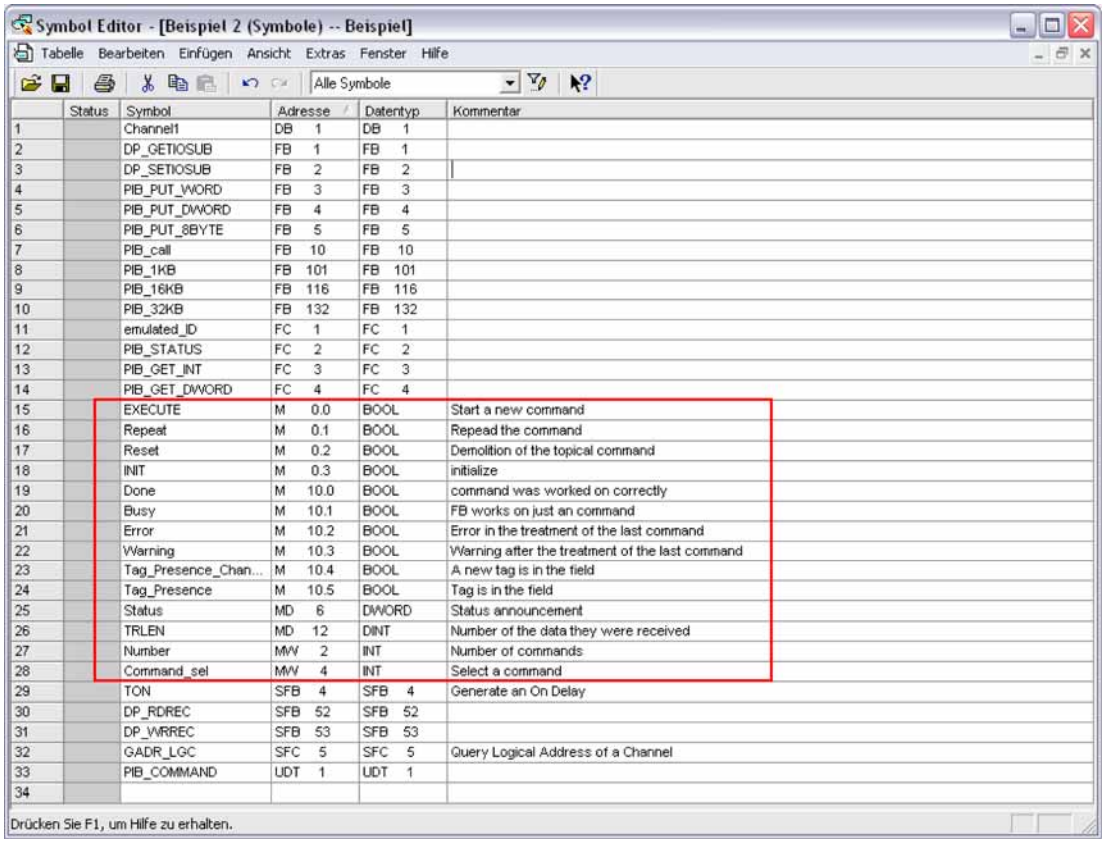
Figure 6-9 Initializing the variables of FB 101 (SCL)

Note

You must define bit memory words 2 and 4 as integers, otherwise problems will arise during compiling.

Assigning symbolic names to bit memories

Assign symbolic names to the bit memories used (for easier handling).



	Status	Symbol	Adresse	Datentyp	Kommentar
1		Channel1	DB 1	DB 1	
2		DP_GETIOSUB	FB 1	FB 1	
3		DP_SETIOSUB	FB 2	FB 2	
4		PIB_PUT_WORD	FB 3	FB 3	
5		PIB_PUT_DWORD	FB 4	FB 4	
6		PIB_PUT_8BYTE	FB 5	FB 5	
7		PIB_call	FB 10	FB 10	
8		PIB_1KB	FB 101	FB 101	
9		PIB_16KB	FB 116	FB 116	
10		PIB_32KB	FB 132	FB 132	
11		emulated_ID	FC 1	FC 1	
12		PIB_STATUS	FC 2	FC 2	
13		PIB_GET_INT	FC 3	FC 3	
14		PIB_GET_DWORD	FC 4	FC 4	
15		EXECUTE	M 0.0	BOOL	Start a new command
16		Repeat	M 0.1	BOOL	Repead the command
17		Reset	M 0.2	BOOL	Demolition of the topical command
18		INIT	M 0.3	BOOL	initialize
19		Done	M 10.0	BOOL	command was worked on correctly
20		Busy	M 10.1	BOOL	FB works on just an command
21		Error	M 10.2	BOOL	Error in the treatment of the last command
22		Warning	M 10.3	BOOL	Warning after the treatment of the last command
23		Tag_Presence_Chan...	M 10.4	BOOL	A new tag is in the field
24		Tag_Presence	M 10.5	BOOL	Tag is in the field
25		Status	MD 6	DWORD	Status announcement
26		TRLEN	MD 12	DINT	Number of the data they were received
27		Number	MW 2	INT	Number of commands
28		Command_sel	MW 4	INT	Select a command
29		TON	SFB 4	SFB 4	Generate an On Delay
30		DP_RDREC	SFB 52	SFB 52	
31		DP_WRREC	SFB 53	SFB 53	
32		GADR_LGC	SFC 5	SFC 5	Query Logical Address of a Channel
33		PIB_COMMAND	UDT 1	UDT 1	
34					

Figure 6-10 Assignment of symbolic names to bit memories (SCL)

Compile SCL source and transfer project

1. Compile the SCL source with DB 1 and OB 1.
2. Transfer the project to the CPU.

Control and test the program

You can control and test the program with the help of a variable table.

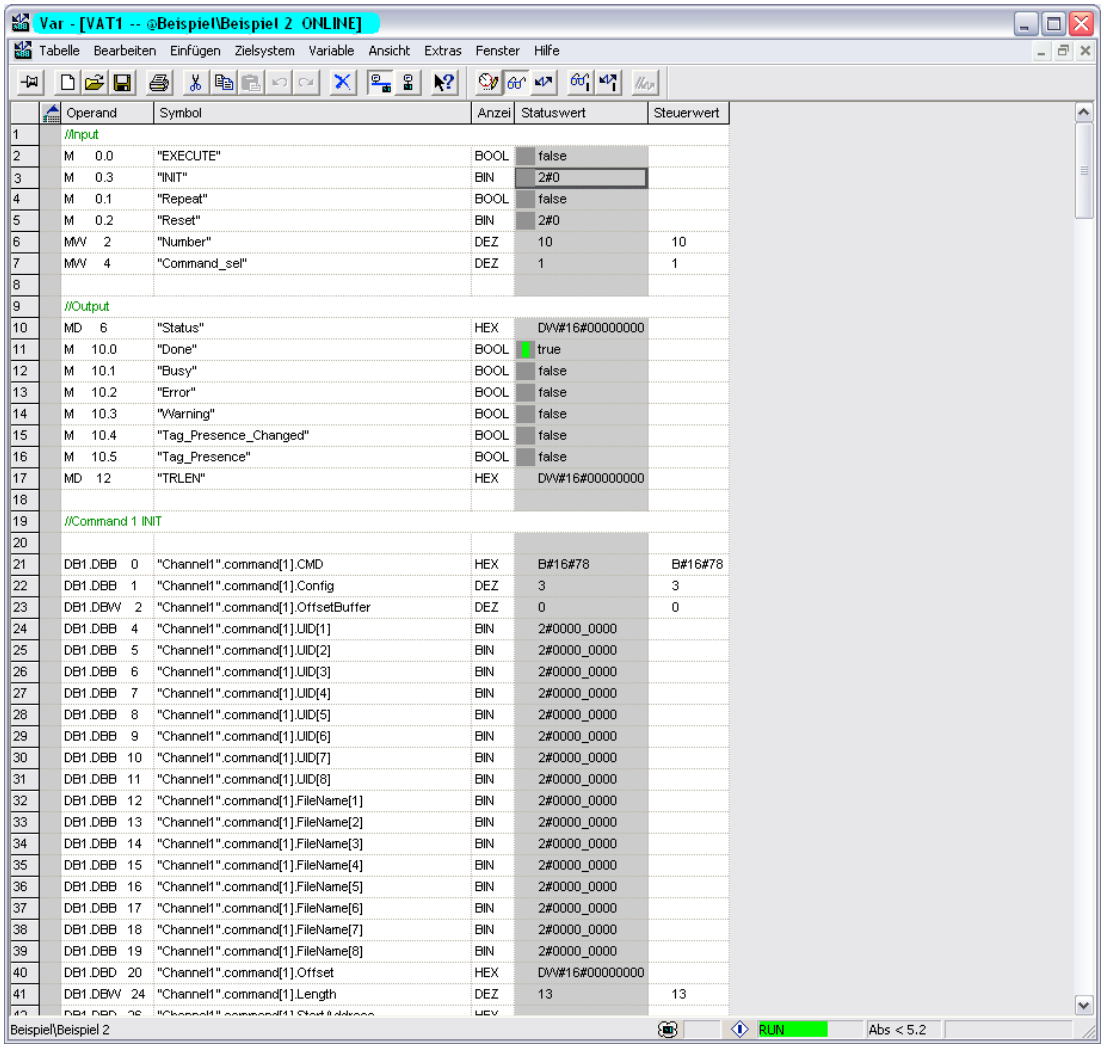


Figure 6-11 Variable table for program control (SCL)

Note

First write command 1 for initialization (see WRITE-CONFIG command) and the receive data buffer. Then execute the initialization by setting and then resetting INIT.

See also

WRITE-CONFIG (also used for INIT) (Page 56)

Short description of the communication modules

A

A.1 ASM 456

Applications

The ASM 456 communication modules are slave modules for operating RFID components via the PROFIBUS DP/DP-V1 on any control systems.

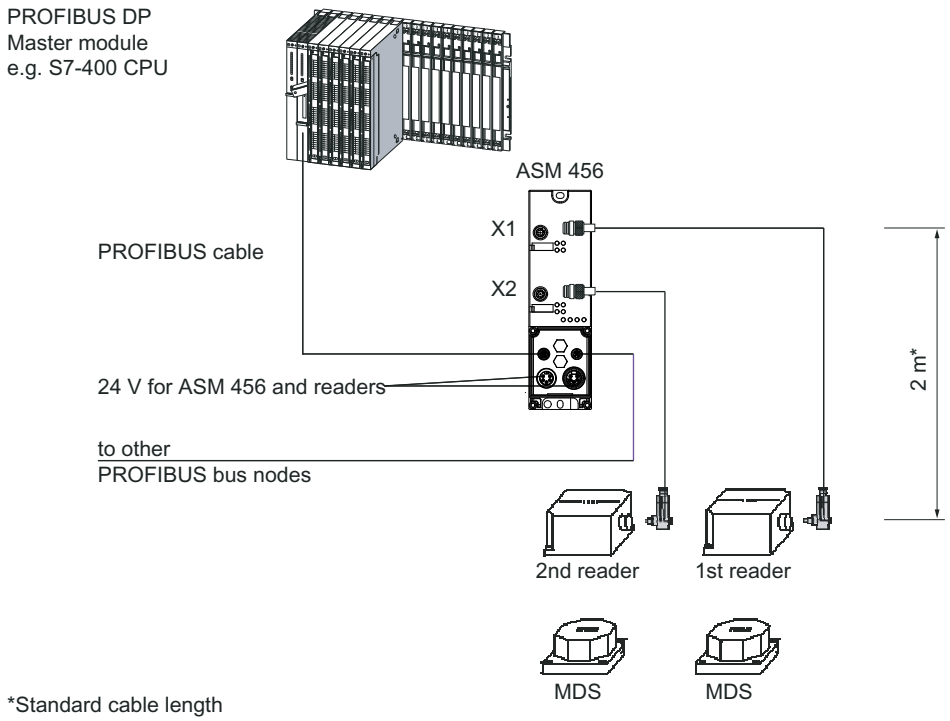


Figure A-1 ASM 456 configurator

Layout

The ASM 456 has the same housing as the ET 200eco distributed I/O system.

For connecting to PROFIBUS DP, the ASM has a connection block that is available optionally in the ECOFAST version or M12, 7/8".

The figure below shows the basic design of the ASM 456.

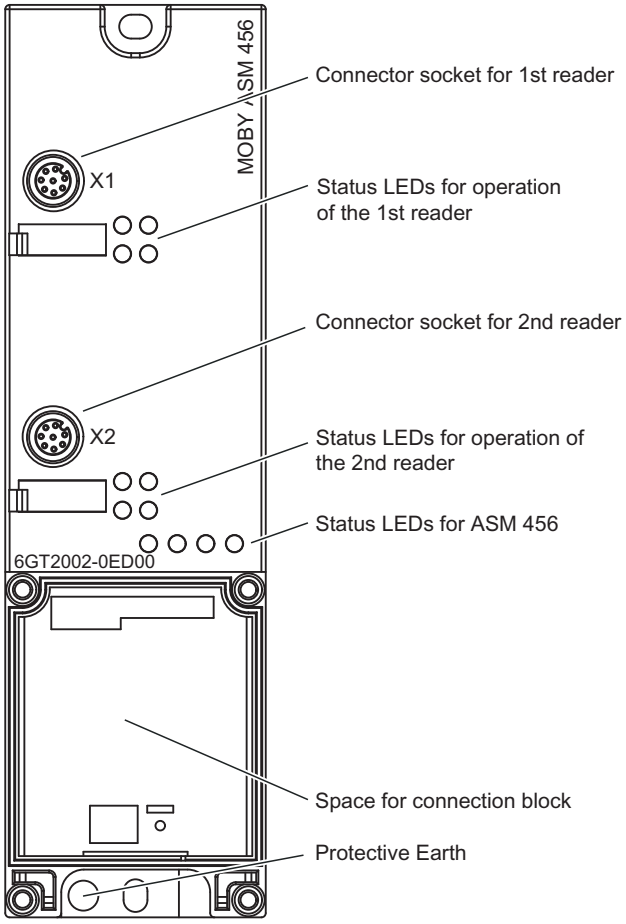


Figure A-2 Basic design of the ASM 456

Hardware configuration

The ASM 456 is integrated into the hardware configuration of the SIMATIC Manager or into another PROFIBUS Master by means of the GSD file SIEM8114.GSD. The file is incorporated into HW-Config of the SIMATIC Manager using the function "Tools - Install new GSD ...". This file is located on the daten\profi_gsd\ASM456 directory of the *RFID Systems Software & Documentation* CD.

For full functionality (diagnostics texts, firmware update), support for GSD revision 5 or higher is required.

Parameter setting by means of GSD file

In addition to the PROFIBUS-relevant control parameters, several RFID-relevant control parameters are also defined for the ASM 456 in the GSD file. The RFID-relevant parameters are set using the "Object properties" of the slave in the hardware configuration. The following table shows the possible settings:

Table A- 1 Setting of RFID-relevant parameters

Parameter name	Value	Note
USER_Mode	FB 45 / FC 45	Default
	FC 55	
	FB 56 / FC 56	
	RFID standard profile	Use FB 101/116/132
MOBY_Mode	MOBY I, E normal addressing	Default
	MOBY I filehandler	only with FB 56/FC 56
	RF300 / MOBY U/D normal addressing	
	MOBY U filehandler	only with FB 56/FC 56 (multitag)
Baud rate for RF300 read/write device/ MOBY U/D	19.2 kbaud	
	57.6 kbaud	[1]
	115.2 kbaud	Default [1]
Diagnostics with diagnostic messages (see Section *Error messages and troubleshooting*)	none	Standard diagnostics only
	Hard errors	Hardware-related messages only
	Hard / soft errors low priority	All messages
	Hard / soft errors high priority	All messages high-priority
[1] not permitted with MOBY D with SLG D11S/D12S		

Input parameters for addressing the ASM 456

Table A- 2 Input parameters for ASM 456

Address	Name	Permissible values	Comment
+2.0	ID	W#16#100, W#16#104, …	Each ASM 456 occupies four bytes of I/O in the I/O area of the controller.
+6.0	INDEX	• Channel 1: 101 • Channel 2: 102	Identification of the data of a channel
+8.0	OFFSET	• Channel 1: 0 • Channel 2: 2	Address offset from the base address (input parameter ID) of the channel-related cyclic I/O data.

Command table of the RFID standard profile with ASM 456

Allocation is undertaken in UDT 1 using the "CMD" variable.

Table A- 3 Commands of the RFID standard profile with ASM 456

Command	Command code		Description	available in MOBY_mode	
	HEX	ASCII		Normal addressing	Filehandler
CREATE	68	'h'	Creates a new file	–	x
DELETE	64	'd'	Deletes a file from the tag	–	x
DEV-STATUS	74	't'	Reads out the status of a reader	x	x
FORMAT	66	'f'	Initializes the tag	x	x
GET Get_QREAD, Get_MOVE	62	'b'	 Reading of several files or reading of an entire tag with one command. Save system data from the filehandler. (Directory, FAT, checksum).	-	x
GET-DIRECTORY	6D	'm'	Reads the directory from the tag	–	x
INVENTORY	69	'i'	Requests a list of all currently accessible tags within the range of the antenna	x	x
MEM-STATUS	73	's'	Reads out the status of a tag	x	x
NEXT	6E	'n'	Does not permit any more operations on a tag	–	x
PHYSICAL-READ	70	'p'	Reads the data from a tag by specifying the physical starting address and the length	x	x
PHYSICAL-WRITE	71	'q'	Writes data to a tag using the physical starting address and the length	x	–
PUT Put_QWRTITE Put_LOAD	65	'e'	 Setting up a complete tag Transfer system data to the communication module	-	x
READ	72	'r'	Reads the data of a file	–	x
READ-CONFIG	61	'a'	Reads the configuration data from the communication module	x	x
SET-ATTRIBUTE	6F	'o'	Sets / modifies the attributes belonging to a file	–	x
UPDATE	75	'u'	Writes data to a file	–	x
WRITE	77	'w'	Writes data to a file	–	x
WRITE-CONFIG	78	'x'	Sends new parameters to the communication module	x	x

Diagnosis using LEDs

The following figure shows details of the LEDs of the ASM 456.

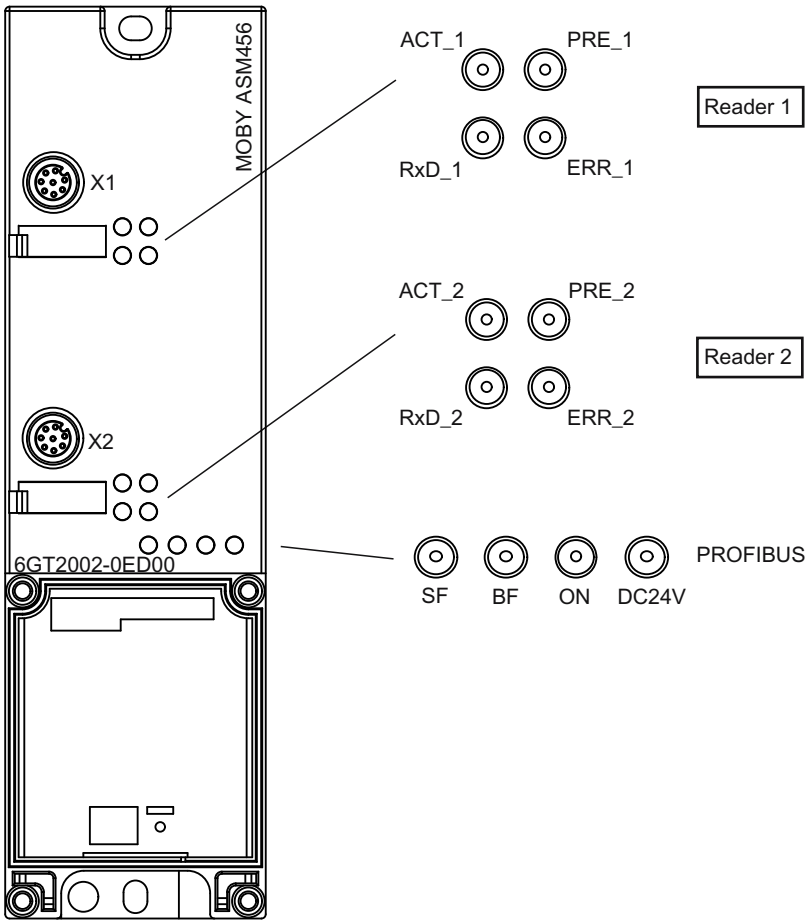


Figure A-3 LEDs of the ASM 456

Table A- 4 Status LEDs for ASM 456

LEDs	Meaning*
ON	Lights up when there is logic voltage at the communication module (is generated by the 24 V supply voltage.)
24V DC	Lights up when the 24 V supply voltage is connected to the communication module.
ACT_1, ACT_2	The corresponding reader is active in processing a user command.
ERR_1, ERR_2 *	A flashing pattern indicates the last error to occur.
PRE_1, PRE_2 **	Indicates the presence of a tag..
RxD_1, RxD_2	Indicates live communication with the reader. May also indicate malfunctions on the reader.
*) The meaning of the individual flash patterns and the associated error descriptions can be found in the relevant FB and FC documentation. **) In multitag mode, this LED uses a flash interval to indicate the number of data media currently within the range of the reader.	

Table A- 5 LED display for PROFIBUS diagnosis

BF	SF	Cause of error	Error correction
On	–	• Communication module is in start-up mode.	–
		• Connection to DP Master failed. • Communication module not detecting a baud rate	• Check the PROFIBUS DP connection. • Check the DP Master
		• Bus interruption • DP Master not functioning	• Check all cables on your PROFIBUS DP network. • Check whether the connector plugs for the PROFIBUS DP are securely plugged into the communication module.
Flashing	On	• The project data sent to the communication by the DP Master do not match the configuration of the communication module.	• Check the configuring of the communication module (input/output, PROFIBUS address). • Correct GSD file being used?
Flashing	–	• The communication module has detected the baud rate, but is not accessed by the DP Master. • Communication module has not been configured.	• Check the PROFIBUS address set in the communication module and/or in the configuring software. • Check the configuring of the communication module (station type).
On	Flashing	• There is a hardware defect in the communication module.	• Replace the communication module.
Off	On	• Diagnosis available	• Evaluate the diagnostic information.
On	Off	• The set PROFIBUS address is incorrect or greater than 99.	• Set the address in the range 1...99 and carry out new ramp-up.
– = Status not relevant			

Other communication module operating modes are indicated by the PRE, ERR, SF, ACT and ON LEDs:

ON	SF	PRE_1	ERR_1	ACT_1	PRE_2	ERR_2	ACT_2	Description
On	Off	Off	Off	On	Off	Off	Off	Ramp-up active
Off	On	Off	On	Off	Off	Off	Off	Checksum error at ramp-up
Off	On	Off	Off	Off	Off	On	Off	Firmware invalid
On	On	On	On	On	On	On	On	LED test for approximately 4 seconds; otherwise firmware fault
Off	On	Off	On	On	Off	On	On	Checksum error at ramp-up
Off	On	On	On	On	Off	On	On	Checksum error of the firmware
Off	On	On	On	On	On	On	On	External RAM defective
Off	On	On	Off	On	On	On	On	DPC-RAM defective
Off	On	Off	On	On	On	On	On	ID error firmware
On	–	Off	1 x flash every 3 s	Off	Off	1 x flash every 3 s	Off	Communication module successfully ramped up, waiting for reset command
–	–	–	n x flash every 3 s	–	–	m x flash every 3 s	–	The last reported error code of the relevant channel can be seen from the number of flashes (n, m).
On	–	–	Flashing	Rapid flashing	–	Flashing	Rapid flashing	Firmware update; alternate flashing of the error LEDs at approximately 1 Hz
– = not relevant								

Terminal assignment

Table A- 6 Connection assignment for ECOFAST connector plugs

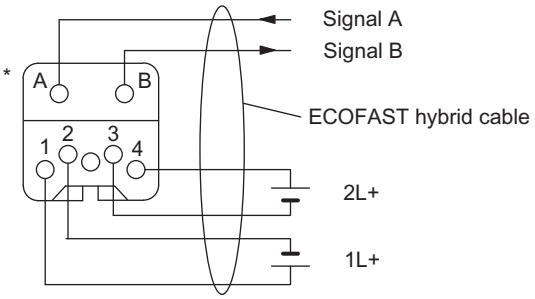
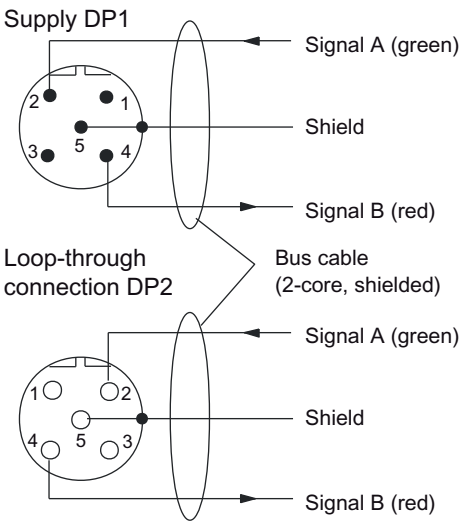
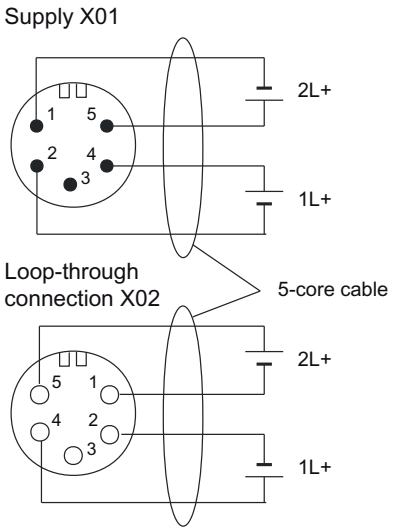
Pin	Assignment	View of ECOFAST connector plug (wiring end for supply and loop-through connection)
A	PROFIBUS DP signal A	
B	PROFIBUS DP signal B	
1	Electronics / encoder supply (1L+) (voltage supply for ASM 456 and reader)	
2	Ground for electronic / encoder supply (1M)	
3	Load voltage ground (2M)	
4	Load voltage supply (2L+) (unused on ASM 456)	
*) You will find the assembly instructions in the packaging of the Han Brid Cu cable connector and/or Han Brid Cu cable socket.		

Table A- 7 Connection assignment for M12 connector (PROFIBUS DP)

Pin	Assignment	View of M12 connector (wiring side)
1	Supply positive (P5V2) *	
2	Data line A (RxD / TxD-N)	
3	Data reference potential (M5V2) *	
4	Data line B (RxD / TxD-P)	
5	Shield	
Thread	Shield	

*) Can only be used for the M12 terminating resistor. Looping the voltage through to the next connector via a 5-core cable is not permitted.

Table A- 8 Connection assignment for 7/8" connector (supply voltages)

Pin	Assignment	View of 7/8" connector (wiring side)
1	Load voltage ground (2M)	
2	Ground for electronic/encoder supply (1M)	
3	PE	
4	Electronics / encoder supply (1L+) (voltage supply for ASM 456 and reader)	
5	Load voltage supply (2L+) (unused on ASM 456)	

See also

Warnings (Page 95)
Error messages (Page 80)

Programming the RFID communication modules on PROFIBUS DP and PROFINET

B

If you are not operating your RFID system in a SIMATIC environment, you must create a separate function block for it.
The programmer gets the necessary information for programming the RFID communication module from the specification of the Proxy Ident Function Block (PIB) of PROFIBUS INTERNATIONAL, Version 1.02 Order No. 3.142.

The specification of the Proxy Ident Function Block is available on the Internet under `http://www.profibus.com` . You will find this there in the download area of PROFIBUS INTERNATIONAL under "Profiles".

Service & Support

C

Contact partner

If you have any further questions on the use of our products, please contact one of our representatives at your local Siemens office.

The addresses are found on the following pages:

- On the Internet (www.siemens.com/automation/partner)
- In Catalog CA 01
- In Catalog FS 10 specially for factory automation sensors

Technical Support

You can access technical support for all IA/DT projects via the following:

- Phone: + 49 (0) 911 895 7222
- Fax: + 49 (0) 911 895 7223
- E-mail (mailto:support.automation@siemens.com)
- Internet: Online support request form: (www.siemens.com/automation/support-request)

Service & support for industrial automation and drive technologies

You can find various services on the Support homepage (www.siemens.com/automation/service&support) of IA/DT on the Internet.

There you will find the following information, for example:

- Our newsletter containing up-to-date information on your products.
- Relevant documentation for your application, which you can access via the search function in "Product Support".
- A forum for global information exchange by users and specialists.
- Your local contact for IA/DT on site.
- Information about on-site service, repairs, and spare parts. Much more can be found under "Our service offer".

RFID homepage

For general information about our identification systems, visit RFID homepage (www.siemens.com/simatic-sensors/rf).

Technical documentation on the Internet

A guide to the technical documentation for the various products and systems is available on the Internet:

SIMATIC Guide manuals (www.siemens.com/simatic-tech-doku-portal)

Online catalog and ordering system

The online catalog and the online ordering system can also be found on the Industry Mall Homepage (http://www.siemens.com/industrymall).

Training center

We offer appropriate courses to get you started. Please contact your local training center or the central training center in

D-90327 Nuremberg.

Phone: +49 (0) 180 523 56 11
(€ 0.14 /min. from the German landline network, deviating mobile communications prices are possible)

For information about courses, see the SITRAIN homepage (www.sitrain.com).

Index

C

T