SIEMENS

SIMATIC Ident

RFID systems
Ident profile and Ident blocks, standard function for Ident systems

Function Manual

12/2016

C79000-G8976-C387-02

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

indicates that minor personal injury can result if proper precautions are not taken.

NOTICE

indicates that property damage can result if proper precautions are not taken.

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, 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 complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of 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
Division Process Industries and Drives
Postfach 48 48
90026 NÜRNBERG
GERMANY

C79000-G8976-C387-02
℗ 12/2016 Subject to change

Table of contents

Introduction

1

Purpose of this document

The interface to the communication services is implemented by readymade program blocks for your user program (FCs and FBs). This manual contains descriptions of the Ident blocks and the Ident profile with which you can commission and assign parameters for the various Ident systems.

It is intended for programmers and testers as well as service and maintenance technicians.

Scope of this documentation

This documentation is valid for the Ident profile or Ident blocks and describes the library version V4.0.

You will find the description of the Ident profile or Ident blocks in conjunction with the technology object "Identification Systems" in the TIA Portal help.

Documentation classification

You will find further information on the blocks and Ident devices named in this manual on the Internet (http://support.automation.siemens.com/WW/view/en/43532183/133300) in the following manuals:

- FB 45
- FB 55
- SIMATIC RF620R/RF630R
- SIMATIC RF650R/RF680R/RF685R
- Interface module ASM 456
- RF120C communications module
- RF170C communications module
- SIMATIC RF180C
- SIMATIC MV420/MV440

Specifications

The Ident blocks in the manual are based on the "Proxy Ident Function Block" protocol. You can obtain the specification of the "Proxy Ident Function Block" from the PROFIBUS User Organization.

Registered trademarks

SIMATIC®, SIMATIC RF®, MOBY®, RF MANAGER® and SIMATIC Sensors® are registered trademarks of Siemens AG.

Abbreviations and naming conventions

The following terms/abbreviations are used synonymously in this document:

Reader	Write/read device (SLG)
Transponder, tag	Mobile data storage (MDS), data carrier
Communications module (CM)	Interface module (ASM)

History

Previous edition(s) of thís function manual:

Edition	Note
10/2014	First Edition
12/2016	Revised and expanded edition

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions only form one element of such a concept.

Customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.

Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit
Link: (http://www.siemens.com/industrialsecurity)

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under
Link: (http://www.siemens.com/industrialsecurity).

Description

2

2.1 Area of application and features

The Ident library contains STEP 7 functions for identification systems. The blocks consist of Ident blocks and the Ident profile. The Ident profile can be used in the SIMATIC S7-300, S7-400, S7-1200 and S7-1500 controllers for various communications modules, RFID readers and optical reader systems. It can be configured with STEP 7 V5.5 and STEP 7 Basic / Professional V13. The Ident blocks are based on the Ident profile and can be configured with STEP 7 Basic / Professional as of V13.

Figure 2-1 Modules that can be configured using the Ident library: ASM 456, RF170C, RF180C, RF120C, RF680R/RF685R, MV440 and MV420

The Ident profile and the Ident blocks can be operated similarly in different configurations.

Table 2- 1 Configurations that can be engineered using the Ident library

Modules	Bus systems		Controllers		
	PROFIBUS	PROFINET	S7-300/-400	S7-1200/-1500	SIMOTION
ASM 456	✓	--	✓	✓	✓
RF170C	✓	✓	✓	✓	✓
RF180C	--	✓	✓	✓	✓
RF120C	--	--	--	✓ 1)	--
RF680R/RF685R	✓ 2)	✓	✓	✓	✓
MV440/MV420	✓ 2)	✓	✓	✓	✓

1) S7-1200 only

2) via communications module

These configurations can be mixed and different communications modules can also be connected.

Difference between Ident blocks and Ident profile

The Ident profile is a single complex block containing all the commands and functions for RFID systems and optical reader systems. The Ident blocks represent a simplified interface of the Ident profile. Each Ident block contains a single command of the Ident profile.

Before starting you should decide which blocks you want to use - Ident profile or Ident blocks. You can, however, only use one of the two variants. Is not possible to mix the Ident profile and Ident blocks per channel!

The following table provides an overview of the differences.

Table 2- 2 Differences between Ident blocks and Ident profile

Ident blocks	Ident profile
• Per command one block • Simple programming • System-specific blocks	• Full range of functions in one block • Complex programming
Supported range of commands: • Reader status • Inventory • Tag status • Read • Write • Set-Ant • Write-ID • Reset-Reader • ...	Supported range of commands: All commands implemented on the reader, e.g. • Inventory • Physical-Read • Get-Blacklist • Matchstring functions (only with MV) • ...
Supported range of functions: • AdvancedCMD for chained command structures (using individual commands in a chain is identical to the Ident profile)	Supported range of functions: • Repeat • Chaining

For more detailed information on the blocks, refer to the sections "Programming Ident blocks (Page 21)" and "Programming the Ident profile (Page 61)".

Recommendation for the selection of blocks

If the Ident blocks cover your functional requirements, use these. The Ident blocks are easier to program and the parameters can be assigned usually without further documentation. The program is easier to read and programming requires less effort.

The Ident profile is a complex block. We recommend that the Ident profile should only be used by trained users or when there are special requirements.

2.2 Building block structure

The program blocks act as the communication interface between an Ident device (e.g. ASM 456) and the user program. The blocks support the following functions:

- Configuration
- Editing commands
- Reading and writing of data
- Diagnostics

The Ident profile is a single complex block containing all the commands and functions for RFID systems and optical reader systems. The Ident blocks represent a simplified interface of the Ident profile. Each Ident block contains a single command of the Ident profile.

The size of the "IDENT_DATA" data buffer (with the Ident blocks), "TXREF" and "RXREF" (with the Ident profile) can be variable. The parameters are defined for S7-300 / S7-400 as "Any" pointers and for S7-1200 / S7-1500 as "Variant".

Table 2- 3 Difference

S7-300/-400 ("Any" pointer)	S7-1200/-1500 ("Variant")
"IDENT_DATA", "TXREF", "RXREF": Arrays of every type with a different length, supplied status UDTs and self-defined UDTs can be created.	"IDENT_DATA", "TXREF", "RXREF": Only arrays of the data type Byte can be created. The length is variable. Exception: With the Ident blocks "Reader-Status" and "Tag-Status", the supplied status data types can also be created for "IDENT_DATA".

2.3 Differentiation from other program blocks

Functions supported by program blocks

The following table provides an overview of the functions supported by the program blocks.

Table 2- 4 Supported functions of the program blocks

Program block	Functions supported by program block							
	Singletag	Multitag	Normal addressing	File handler	PROFIBUS	PROFINET	MV	
FB 45	✓	--	✓	--	✓	✓	✓	Recommended block for singletag applications
FB 55	✓	✓	✓	--	✓	✓	--	Recommended block for multitag applications
FC 56	✓	✓	--	✓	✓	--	--	Recommended block for file handler
Standard profile V1.19	✓	✓	✓	✓	✓	✓	✓	Recommended block for third-party controllers
Ident profile	✓	✓	✓	--	✓	✓	✓	Block based on the PNO specification (Based on the standard profile V1.19, however without file handler)
Ident blocks	✓	✓	✓	--	✓	✓	✓	Blocks based on the Ident profile
FC 44	✓	--	✓	--	✓	--	--	Only for RF160C
Application blocks RF160C	✓	--	✓	--	✓	--	--	Only for RF160C
Application blocks IO-Link	✓	--	✓	--	✓	✓	--	Only for RF200 IO-Link

Compatible program blocks

The following table shows the program blocks compatible with the interface modules/communications modules.

Table 2- 5 Compatible program blocks

Ident systems	Compatible program blocks in conjunction with ...		
	S7-300 / S7-400 and STEP 7 Classic V5.5	**S7-300 / S7-400 and STEP 7 Basic/Professional**	**S7-1200 / S7-1500 and STEP 7 Basic/Professional**
ASM 456	FB 45 FB 55 FC 56 Standard profile V1.19 Ident profile	FB 45 FB 55 FC 56 Ident profile Ident blocks	Ident profile Ident blocks PIB_1200_UID_001KB PIB_1200_UID_032KB
ASM 475	FB 45 FB 55	FB 45 FB 55	--
SIMATIC RF120C	--	--	Ident profile Ident blocks PIB_1200_UID_001KB PIB_1200_UID_032KB
SIMATIC RF160C	FC 44 Application blocks for RF160C	FC 44 Application blocks for RF160C	Application blocks for RF160C
SIMATIC RF170C	FB 45 FB 55	FB 45 FB 55 Ident profile Ident blocks	--
SIMATIC RF180C	FB 45 FB 55 Standard profile V1.19 Ident profile	FB 45 FB 55 Ident profile Ident blocks	Ident profile Ident blocks PIB_1200_UID_001KB PIB_1200_UID_032KB
SIMATIC RF680R/RF685R	Ident profile	Ident profile Ident blocks	Ident profile Ident blocks
SIMATIC MV420/MV440	FB 79 Standard profile V1.19 Ident profile	FB 79 Ident profile Ident blocks	Ident profile PIB_1200_UID_001KB PIB_1200_UID_032KB

Setting parameters for blocks

3

3.1 Overview of the Ident library

The Ident library with the Ident profile and the Ident blocks are integrated in STEP 7 as of version V13 SP1. You will find the blocks in the "Instructions" tab under "Optional packages" > "SIMATIC Ident".

The following table provides an overview of the currently available blocks.

Table 3- 1 Overview of the Ident library

Position			Symbolic name	Description
Instructions/ blocks	Ident blocks	Basic blocks	Read	Using these blocks, it is simple to program communication with the Ident systems. The basic blocks include all the blocks that are used often.
			Write	
			Reset_Reader	
			Read_MV	
			Set_MV_Program	
		Extended blocks	Config_Download	Using these blocks, it is simple to program communication with the Ident systems. The extended blocks provide functions that are required less often for operating the Ident system.
			Config_Upload	
			Inventory	
			Read_EPC_Mem	
			Read_TID	
			Read_UID	
			Set_ANT_RF300	
			Set_ANT_RF600	
			Set_Param	
			Write_EPC_ID	
			Write_EPC_Mem	
			AdvancedCmd	Advanced command set. With the "AdvancedCmd" block it is possible to access other commands from the Ident command set and to execute chained commands.
		Status blocks	Reader_Status	Using the status blocks you obtain information on the reader or transponder.
			Tag_Status	
		Reset blocks	Reset_MOBY_D	Using these blocks, it is simple to program communication with the Ident systems. The reset blocks are used for simple initialization of the Ident systems if the "Reset_Reader" block is not supported by the Ident system.
			Reset_MOBY_U	
			Reset_MV	
			Reset_RF200	
			Reset_RF300	
			Reset_RF600	
			Reset_Univ	

Position		Symbolic name	Description
	Ident profile	Ident_Profile	These blocks are available for experts to be able to include complex command structures in their own program sequence. It is also possible to use repeat commands and chaining.
PLC data types	System data types	IID_HW_CONNECT	Data type for all blocks for physical addressing of communications modules and readers and for synchronizing the function blocks used for each reader.
		IID_CMD_STRUCT	Data type for the Ident profile for setting the command parameters.
	Status data types	IID_READSTAT_ 84_MOBY_U	Data types for the result of "Reader_status" with the relevant attribute. The data types help you to interpret the data received from the reader and to process the data further directly without data type conversions.
		IID_READSTAT_ 81_RF2_3_U	
		IID_READSTAT_ 86_RF300	
		IID_READSTAT_ A0_A1_RF600	
		IID_READSTAT_ 87_RF600	
		IID_READSTAT_ 88_RF600	
		IID_READSTAT_ 89_RF68xR	
		IID_TAG_STATUS_ 83_ISO	Data types for the result of "Tag_status" with the relevant attribute. The data types help you to interpret the data received from the reader and to process the data further directly without data type conversions.
		IID_TAG_STATUS_ 80_MOBY_U	
		IID_TAG_STATUS_ 04_RF300	
		IID_TAG_STATUS_ 82_RF300	
		IID_TAG_STATUS_ 84_RF600	
		IID_TAG_STATUS_ 85_RF600	
		IID_INVENT_ 00_MOBY_U	Data types for the result of "Inventory" with the relevant attribute. The data types help you to interpret the data received from the reader and to process the data further directly without data type conversions.
		IID_INVENT_ A0_A1_RF600	
		IID_INVENT_ 82_83_RF600	
		IID_INVENT_ 85_RF600	
		IID_INVENT_ 8x_9x_RF6_D	

Note

Parallel operation using Ident blocks and Ident profile is not possible

Note that the reader cannot be operated at the same time using the Ident blocks and the Ident profile.

3.2 Setting the "IID_HW_CONNECT" data type

Before you can start parameter assignment of the blocks, you first need to create a variable of the PLC data type "IID_HW_CONNECT". The Ident system or a channel of the Ident system is addressed using the "IID_HW_CONNECT" PLC data type.

Addressing the Ident devices

When working with all the instructions/blocks, you require the "IID_HW_CONNECT" data type to address the reader. Setting the command parameters for the Ident profile is handled by the Ident blocks. The Ident profile and the "AdvancedCMD" block also require the "IID_CMD_STRUCT" data type for the parameter assignment of the individual commands. Depending on whether you work with the Ident profile or the Ident blocks, you need to link in and assign parameters for these data types as described in the following sections.

Parameter assignment of the "IID_HW_CONNECT" data type

Follow the steps below to set the parameters for the "IID_HW_CONNECT" data type for a channel:

1. In the project tree, double-click on the entry "Create new block" in the "Program block" folder.
2. Click the "Data block" button and assign a name to the block.
3. Confirm your entry with "OK".

   The data block is opened.
4. Create a new variable by entering a variable name in the "Name" column.

5. In the "Data type" column, select the "IID_HW_CONNECT" data type.

Reader_1

	Name	Data type	Start value	Retain	Accessible f...	Visible in ...	Setpoint	Comment
1	Static							
2	connect	"IID_HW_CONNECT"						
3	HW_ID	Word	16#0					only S7-1200/1500: HW identifier
4	CM_CHANNEL	Int	1					channel of communication module
5	LADDR	DWord	16#0					i/o address
6	Static	"IID_IN_SYNC"						

Figure 3-1 Creating a data block

6. Specify the address data of the Ident device or the channel.
   - HW_ID: Hardware identifier of the module (only with S7-1200 and S7-1500)
   - CM_CHANNEL: Channel of the reader on the CM or the antenna
   - LADDR: I/O address of the module

   You can read out the values of the "HW_ID" and "LADDR" parameters in the device configuration in the properties of the communications module/reader. Enter the parameter values you have read out in the "Start value" column of the corresponding parameters. Reading out parameter values is described below.

Follow the steps below to read out the parameter values "HW_ID" and "LADDR" for a channel (only with RF180C and ASM 456):

1. Open the device view.
2. Double-click on the communications module.

   The properties window of the CM opens.
3. In the "Device overview" tab, select the module "2x RS422 channels RFID_1".

   The I/O address displayed in the tab corresponds to "LADDR".

   Note that the input and output address must have the same value.

4. On the "Properties" > "General" > "Hardware identifier" tab you will find the hardware identifier that corresponds to the "HW_ID".

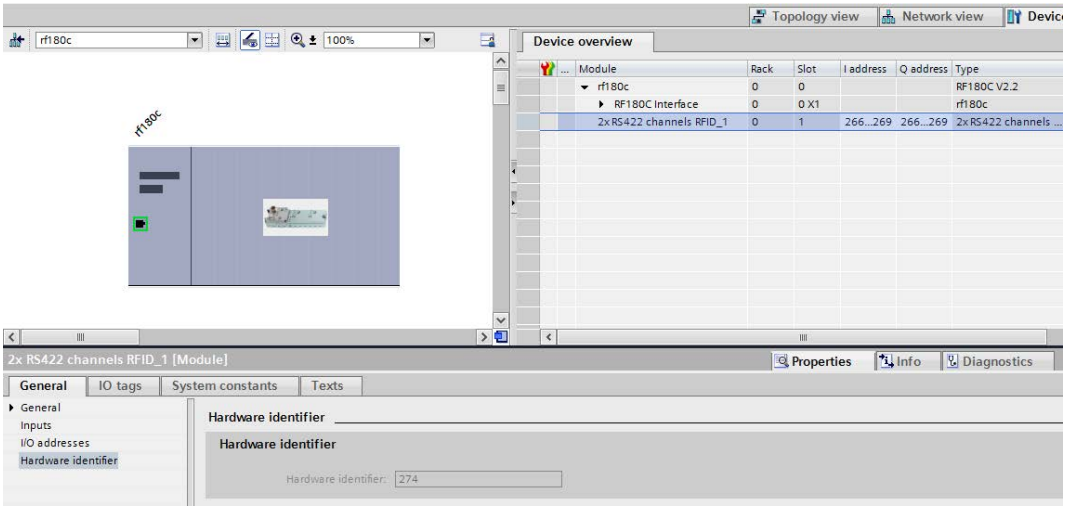


Figure 3-2 The "Hardware identifier" parameter

5. Transfer the values of "LADDR" and "HW_ID" to the PLC data type "IID_HW_CONNECT" of the reader for which you want to set parameters.

Note

Setting the user mode

Note that in the properties of the communications module, you assign the value "RFID standard profile" to the "User mode" parameter and select the suitable MOBY mode.

With all other communications modules/readers, you will find the two parameters directly in the properties of the module.

The "IID_HW_CONNECT" data type has now been created and addressed for a channel. Repeat these steps for every other reader/channel. If you want to use a different channel of the reader/CM, set this using the "CM_CHANNEL" parameter. The "HW_ID" and "LADDR" parameters remain the same for all channels/readers/antennas.

The library is now linked in and the required blocks and data types have been created in your project. The "IID_HW_CONNECT" data type has also been created and addressed. You can now start programming the blocks.

Note

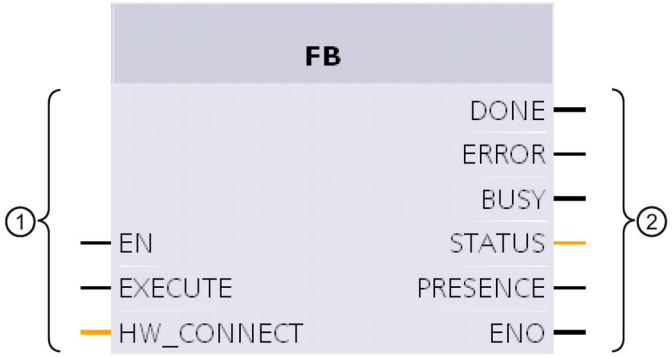
Configuring "IID_CMD_STRUCT"

If you work with the Ident profile or with the "AdvancedCmd" block, you also need to create a further element with the data type "IID_CMD_STRUCT" (Array [1...10]) in the data block you have already created.

3.3 General structure of the function blocks

Structure of the blocks based on the sample block "FB"

The following graphic shows an example of a block with input and output parameters as they exist in the same way in all blocks.



① Input parameters
② Output parameters
Data type: Bit
Data type: Byte, Word, DWORD, INT

Figure 3-3 Example of a block

Description of the parameters

Table 3- 2 Description of the input and output parameters

Parameter	Description
Input parameters	
EN	Enabling Input
EXECUTE	There must be a positive edge at this input before the block will execute the command.
HW_CONNECT	Global parameter of the type "IID_HW_CONNECT" to address the channel/reader and to synchronize the blocks. This parameter needs to be created and addressed once for each channel/reader. "HW_CONNECT" must always be transferred to the blocks to address the relevant channel/reader.

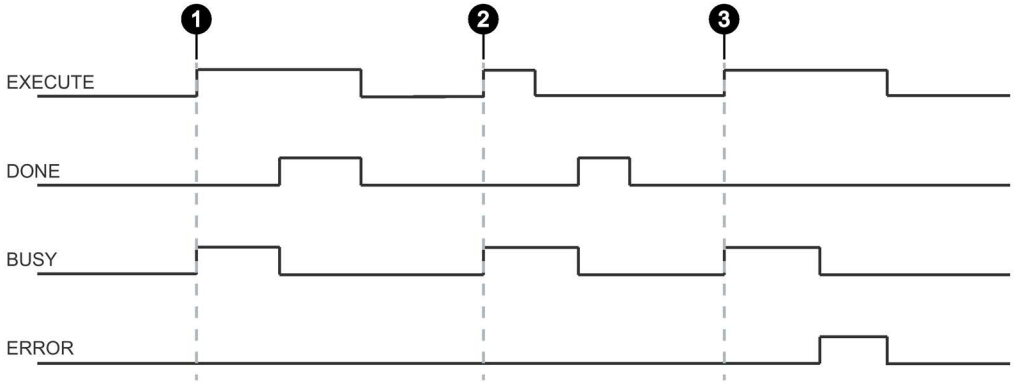
Parameter	Description
Output parameters	
DONE (BOOL)	The job was executed. If the result is positive, this parameter is set.
ERROR (BOOL)	The job was ended with an error. The error code is indicated in Status.
BUSY (BOOL)	The job is being executed.
STATUS (DWORD)	Display of the error message if the "ERROR" bit was set.
PRESENCE (BOOL)	This bit indicates the presence of a transponder. The displayed value is updated each time the block is called. This parameter does not exist in blocks specific to optical reader systems.
ENO	Enable output

General sequence when calling the blocks

Note

Different sequences with the Ident profile and standard profile V1.19

Note that the sequence of the Ident profile is not the same as that of the previous standard profile V1.19. Depending on the environmental conditions, the application conditions and the block functions used, the standard profile V1.19 cannot be used.



Case ① By setting EXECUTE (EXECUTE = 1) the function/instruction is started. If the job was completed successfully (DONE = 1), you need to reset EXECUTE. DONE is reset at the same time.

Case ② EXECUTE is set for only one cycle. As soon as BUSY is set, you can reset EXECUTE again. If the job was completed successfully, DONE is set for one cycle.

Case ③ Handling as in Case 1, however with error output. As soon as ERROR is set, the precise error code is available in the STATUS output. ERROR and STATUS retain their values as long as EXECUTE is set or for one cycle if EXECUTE was reset before the block was ended.

Figure 3-4 General sequence when calling the blocks

How the blocks work

You can only ever send one command to the reader or communications module. You can, however, call and start two or more blocks at the same time. The blocks execute in the order in which they are called.

This does not apply to the Reset blocks. If a Reset command is executed, the command active at this time is aborted.

Creating blocks

Requirement

The "IID_HW_CONNECT" data type has had parameters assigned.

Follow the steps below to link in a block and to set the call parameters:

1. Open the program block you have created by double-clicking in the "Project tree" > "Program blocks" tab.
2. Drag the required block from the block library tab to the program block.
3. Enter the variable you created earlier in the "HW_CONNECT" input parameter.

The block is called and connected to the relevant channel.

Note

Working with multiple channels

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

Note

Working with the Ident profile or with the "AdvancedCmd" block

If you work with the Ident profile or with the "AdvancedCmd" block, you also need to connect the "CMDREF" input parameter with a variable of the "IID_CMD_STRUCT" (Array [1...10]) data type.

3.4 Programming Ident blocks

3.4.1 Basic blocks

3.4.1.1 Read

The "Read" block reads the user data from the transponder and enters this in the "IDENT_DATA" buffer. The physical address and the length of the data are transferred using the "ADDR_TAG" and "LEN_DATA" parameters. With the RF68xR readers, the block reads the data from memory bank 3 (USER area). Specific access to a certain transponder is made with the "EPCID_UID" and "LEN_ID".

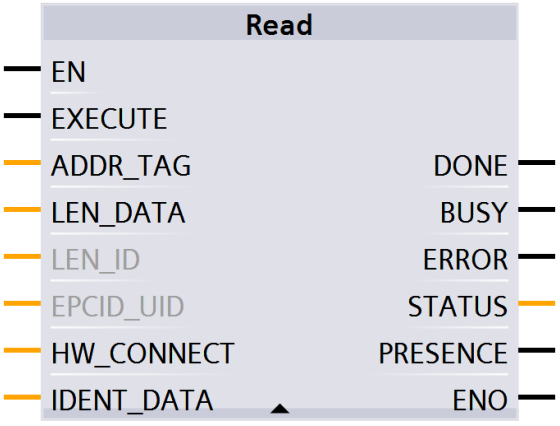


Figure 3-5 "Read" block

Table 3- 3 Explanation of the "Read" block

Parameter	Data type	Default values	Description
ADDR_TAG	DWord	DW#16#0	Physical address on the transponder where the read starts. You will find more information on addressing in the section "Transponder addressing (Page 103)". With MV: The length of the read code is located starting at address "0" (2 bytes). The read code is located starting at address "2". [1)]
LEN_DATA	Word	W#16#0	Length of the data to be read
LEN_ID	Byte	B#16#0	Length of the EPC-ID/UID Default value: 0x00 ≙ unspecified single tag access (RF680R, RF685R)

Parameter	Data type	Default values	Description
EPCID_UID	Array[1...62] of Byte	0	Buffer for up to 62 bytes EPC-ID, 8 bytes UID or 4 bytes handle ID. • 2 - 62-byte EPC-ID is entered at the start of the buffer (length is set by "LEN_ID") • 8-byte UID is entered at the start of the buffer ("LEN_ID = 8") • 4-byte handle ID must be entered in the array element [5]-[8] ("LEN_ID = 8") Default value: 0x00 ≙ unspecified single tag access (RF620R, RF630R)
IDENT_DATA	Any / Variant	0	Data buffer in which the read data is stored. Note: For Variant, currently only an "Array_of_Byte" with a variable length can be created. For Any, other data types/UDTs can also be created.

[1] You will find further information on working with optical reader systems in the operating instructions "SIMATIC MV420 / SIMATIC MV440".

3.4.1.2 Read_MV

The "Read_MV" block reads out the read result of a camera. The length of the data to be read is calculated automatically by the camera based on the length of the created receive buffer. The actual length of the read result is output in the "LEN_DATA" output parameter. The data will be saved in the "IDENT_DATA" data buffer. If the buffer is too small, the error message "0xE7FE0400" appears and the expected length is output at "LEN_DATA".

To achieve an optimum speed, we recommend that you adapt the length of the data type "IDENT_DATA" so that this is as close as possible to the maximum expected length of the read result (2 bytes code length + read code).

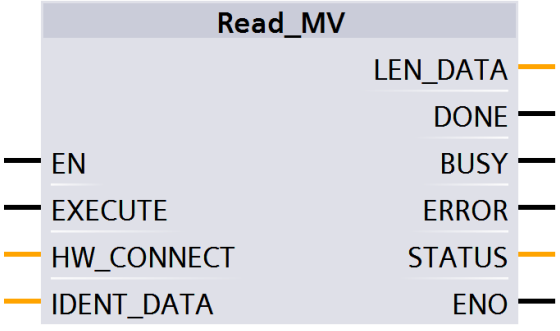


Figure 3-6 "Read_MV" block

Table 3- 4 Explanation of the "Read_MV" block

Parameter	Data type	Default values	Description
IDENT_DATA	Any / Variant	0	Read result The length of the read code is located in bytes 0 and 1.
LEN_DATA	Word	W#16#0	Length of the read result ≙ 2 bytes code length + read code Note: For Variant, an "Array_of_Byte" with a variable length can be created. For Any, other data types/UDTs can also be created.

3.4.1.3 Reset_Reader

The "Reset_Reader" block can currently only be used in conjunction with the RF680R and RF685R readers or the RF120C communications module with a reader connected.

Using the "Reset_Reader" block, you can reset all reader types of the Siemens RFID systems. All the readers are reset to the settings stored in the device configuration of the RF120C or that were configured in the RF68x reader using the WBM. The "Reset_Reader" block does not have any device-specific parameters and is executed using the "EXECUTE" parameter.

You will find descriptions of other Reset blocks for operation with the communications modules RF180C and ASM 456 or optical reader systems in the section "Reset blocks (Page 46)".

With the "Reset_Reader" block and the other reset blocks, you can interrupt any active Ident block at any time. These blocks are then ended with "DONE = true" and "ERROR = false".

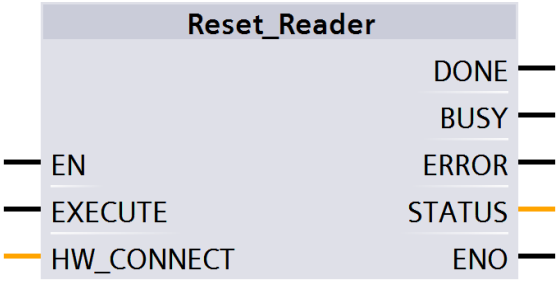


Figure 3-7 "Reset_Reader" block

3.4.1.4 Set_MV_Program

With the aid of the "Set_MV_Program" block, you can change the program in a camera. The required program number is transferred using the "PROGRAM" parameter.

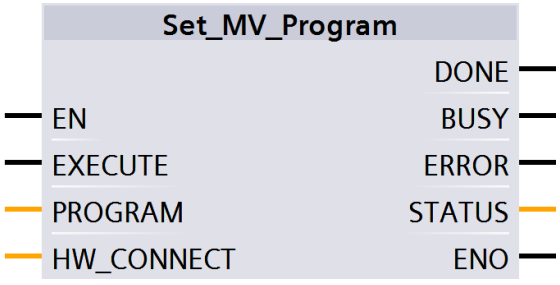


Figure 3-8 "Set_MV_Program" block

Table 3- 5 Explanation of the "Set_MV_Program" block

Parameter	Data type	Default values	Description
PROGRAM	Byte	B#16#1	Program number Range of values: 0x01 ... 0x0F

3.4.1.5 Write

The "Write" block writes the user data from the "IDENT_DATA" buffer to the transponder. The physical address and the length of the data are transferred using the "ADDR_TAG" and "LEN_DATA" parameters. With the RF68xR readers, the block writes the data to memory bank 3 (USER area). Specific access to a certain transponder is made with the "EPCID_UID" and "LEN_ID".

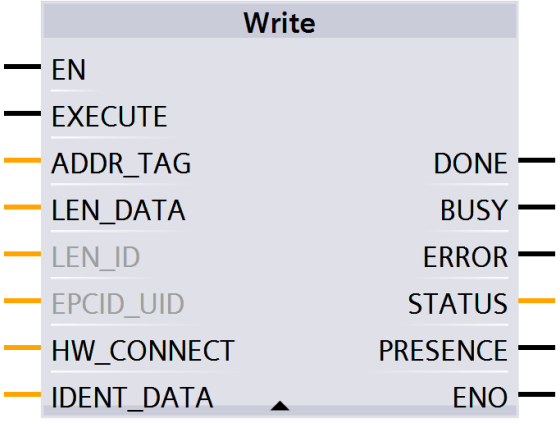


Figure 3-9 "Write" block

Table 3- 6 Explanation of the "Write" block

Parameter	Data type	Default values	Description
ADDR_TAG	DWord	DW#16#0	Physical address on the transponder where the write starts. You will find more information on addressing in the section "Transponder addressing (Page 103)". With MV: Address is always 0. [1)]
LEN_DATA	Word	W#16#0	Length of the data to be written
LEN_ID	Byte	B#16#0	Length of the EPC-ID/UID Default value: 0x00 ≙ unspecified single tag access (RF680R, RF685R)
EPCID_UID	Array[1...62] of Byte	0	Buffer for up to 62 bytes EPC-ID, 8 bytes UID or 4 bytes handle ID. • 2 - 62-byte EPC-ID is entered at the start of the buffer (length is set by "LEN_ID") • 8-byte UID is entered at the start of the buffer ("LEN_ID = 8") • 4-byte handle ID must be entered in the array element [5]-[8] (LEN_ID = 8) Default value: 0x00 ≙ unspecified single tag access (RF620R, RF630R)
IDENT_DATA	Any / Variant	0	Data buffer with the data to be written. With MV: The first byte encodes the corresponding MV command.[1)] Note: For Variant, currently only an "Array_of_Byte" with a variable length can be created. For Any, other data types/UDTs can also be created.

1) You will find further information on working with optical reader systems in the operating instructions "SIMATIC MV420 / SIMATIC MV440".

3.4.2 Extended blocks

3.4.2.1 Config_Upload/-_Download

Using the "Config_Upload" and "Config_Download" blocks, you can read out ("Config_Upload") or write ("Config_Download") the configuration of the RF680R/RF685R readers via the control program.

The configuration data is not interpretable data. Save the data on the controller so that it can be written to the reader again if a device is replaced. Bytes 6-9 (see table below) contain a configuration ID with a unique version identifier. With the configuration ID, when performing a "Config_Upload", you can check whether the configuration data read matches the configuration data stored on the controller. The configuration data has the following structure:

Table 3- 7 Structure of the configuration data

Byte	Name
0	Structure identifier (2 bytes)
2	Length information (4 bytes) Length of the version identifier and parameter block
6	Version identifier (4 bytes) Based on the identifier, you can uniquely identify the configuration. This is a time stamp in Linux format. The time stamp indicates how many seconds have passed since January 1, 1979, 00:00 (midnight). The identifier is assigned when a configuration is generated.
10 ... end "DATA"	Parameter block

"Config_Upload/Config_Download" can be executed on every channel of the RF680R/RF685R. It is always the same configuration data that is transferred.

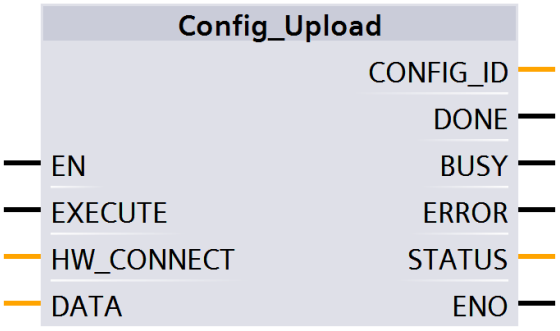


Figure 3-10 "Config_Upload" block

Table 3- 8 Explanation of the "Config_Upload" block

Parameter	Data type	Description
DATA	Any / Variant	Data buffer for configuration data. The real length of the data depends on the complexity of the configuration and the firmware version of the reader. With a standard configuration of the RF680R/RF685R reader, we recommend a memory size of 4 KB. If you use advanced reader configurations (filtering) or want to change the configuration in the future without needing to adapt the memory size of "DATA", we recommend a memory size of 8-16 KB. Note: For Variant, currently only an "Array_of_Byte" with a variable length can be created. For Any, other data types/UDTs can also be created.
CONFIG_ID	DWord	Version identifier (4 bytes) Based on the identifier, you can uniquely identify the configuration. This is a time stamp in Linux format. The time stamp indicates how many seconds have passed since January 1, 1979, 00:00 (midnight). The identifier is assigned when a configuration is generated.

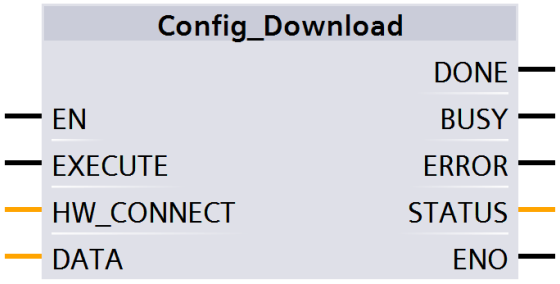


Figure 3-11 "Config_Download" block

Table 3- 9 Explanation of the "Config_Download" block

Parameter	Data type	Description
DATA	Any / Variant	Data buffer for configuration data. The real length of the data depends on the complexity of the configuration and the firmware version of the reader. With a standard configuration of the RF680R/RF685R reader, we recommend a memory size of 4 KB. If you use advanced reader configurations (filtering) or want to change the configuration in the future without needing to adapt the memory size of "DATA", we recommend a memory size of 8-16 KB. Note: For Variant, currently only an "Array_of_Byte" with a variable length can be created. For Any, other data types/UDTs can also be created.

3.4.2.2 Inventory

The "Inventory" block activates the taking of inventories. With the RF620R and RF630R readers, inventories are always taken as soon as the antenna is turned on.

Special feature of the RF680R, RF685R readers

Note that the length of the data buffer ("IDENT_DATA") must correspond to at least the length of the maximum expected data. If more transponders are identified and data read out than have space in the assigned buffer length of "IDENT_DATA", the data of these transponders is lost. This reaction is indicated by the error "E7FE0400h" (buffer overflow).

For the RF680R and RF685R readers, the parameters "DURATION" and "DUR_UNIT" are also available. Using the parameters, you can specify the duration of the inventories.

With the RF680R/RF685R readers. there are four different modes that you can select with the "ATTRIBUTE" parameter.

- At the start, a certain duration/number (period of time, number of inventories, number of "observed" events or identified transponders) is specified. A distinction is made between the following options:
  - Duration

    Take inventories for a specified period of time
  - Number of inventories

    Take a specified number of inventories
  - Number of "observed" events

    Take inventories until a specified number of transponders have been identified at the same time.

  Inventories are then taken by the reader for this time or number of inventories. When the specified time/number is reached, the block is ended and returns all identified transponders in "IDENT_DATA". In other words, other commands can only be executed when all inventories have been taken completely. The unit (time or number) is specified using "DUR_UNIT" and the value (time value or number) using "DURATION". This mode can be executed using the attributes "0x80" and "0x81". Depending on the attribute, more or less data is supplied about the identified transponders.
- With the attributes "0x86" (start "Presence_Mode") and "0x87" (end "Presence_Mode"), inventories can be taken permanently. The presence of a transponder can then always be queried using "PRESENCE" without needing to start the block with "EXECUTE". No information about the identified transponders is returned when the command executes!

  To obtain information about the identified transponders, use one of the two calls listed above (with time / number of inventories = 0).

  When this mode is active, commands relating to transponders are not executed immediately but only when a transponder is identified. This achieves shorter reaction times since the command is already pending when the transponder enters the antenna field.

  The "Presence_Mode" is practical in the context of the "Repeat command" function.

The "NUMBER_TAGS" output parameter is used to output the number of identified transponders. With the attributes "0x80" and "0x81" on completion of the read operation, the sum of all identified transponders is displayed. With the attribute "0x86" the number of currently identified transponders is shown at the "NUMBER_TAGS" output parameter (max. 15), without needing to start the module with "EXECUTE".

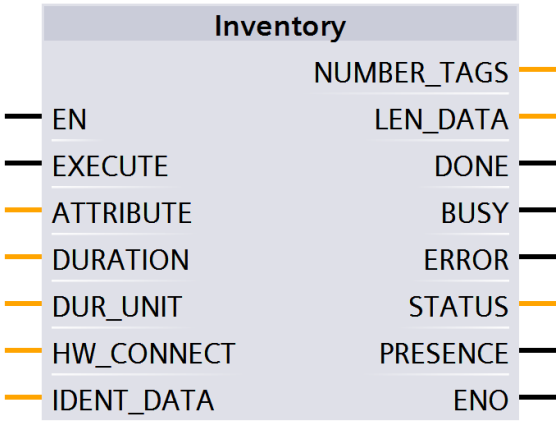


Figure 3-12 "Inventory" block

Table 3- 10 Explanation of the "Inventory" block

Parameter	Data type	Default values	Description
ATTRIBUTE	Byte	B#16#0	Selecting the status mode: • RF300, MOBY U: 0x00 • RF620R, RF630R: 0x82 (read out next data record), 0x83, 0x85, 0x90, 0x91, 0x92 • RF680R, RF685R: 0x80, 0x81, 0x86, 0x87
DURATION	Word	W#16#0	RF680R, RF685R: Duration dependent on "DUR_UNIT" Period of time or number of inventories or number of "Observed" events Example: • 0x00 ≙ no inventory • 0x01 ≙ one inventory
DUR_UNIT	Word	W#16#0	RF680R, RF685R: Unit for "DURATION" • 0x00 ≙ time [ms] • 0x01 ≙ inventories • 0x02 ≙ number of "Observed" events
IDENT_DATA	Any / Variant	0	Data buffer for inventory data Note: For Variant, an "Array_of_Byte" with a variable length can be created. For Any, other data types/UDTs can also be created.
NUMBER_TAGS	Int	0	Number of transponders in the antenna field
LEN_DATA	Word	W#16#0	Length of the valid data

Results for MOBY U

Table 3- 11 ATTRIBUTE "0x00" (data type "IID_INVENT_00_MOBY_U")

Name	Type	Comment
number MDS	WORD	Number of MDS
UID length	WORD	length of UID
UID	ARRAY[1..12] of IID_IN_I_8BYTE	
UID[1]	IID_IN_I_8BYTE	
UID	ARRAY[1..8] of BYTE	
UID[1]	BYTE	
UID[2]	BYTE	
UID[3]	BYTE	
UID[4]	BYTE	
UID[5]	BYTE	
UID[6]	BYTE	
UID[7]	BYTE	
UID[8]	BYTE	
UID[2]	"IID_IN_I_8BYTE"	
UID[3]	"IID_IN_I_8BYTE"	
UID[4]	"IID_IN_I_8BYTE"	
UID[5]	"IID_IN_I_8BYTE"	
UID[6]	"IID_IN_I_8BYTE"	
UID[7]	"IID_IN_I_8BYTE"	
UID[8]	"IID_IN_I_8BYTE"	
UID[9]	"IID_IN_I_8BYTE"	
UID[10]	"IID_IN_I_8BYTE"	
UID[11]	"IID_IN_I_8BYTE"	
UID[12]	"IID_IN_I_8BYTE"	

Results for RF620R, RF630R

Table 3- 12 ATTRIBUTE "0x83" (data type "IID_INVENT_82_83_RF600") for RF620R, RF630R with EPC-ID/UID

Name	Type	Comment
reserved0	BYTE	
number MDS	BYTE	Number of MDS
EPC	ARRAY[1..19] of "IID_IN_I_12BYTE"	
EPC[1]	"IID_IN_I_12BYTE"	
ID	ARRAY[1..12] of BYTE	
ID[1]	BYTE	
ID[2]	BYTE	
ID[3]	BYTE	
ID[4]	BYTE	

Name	Type	Comment
ID[5]	BYTE	
ID[6]	BYTE	
ID[7]	BYTE	
ID[8]	BYTE	
ID[9]	BYTE	
ID[10]	BYT	
ID[11]	BYTE	
ID[12]	BYTE	
EPC[2]	"IID_IN_I_12BYTE"	
EPC[3]	"IID_IN_I_12BYTE"	
EPC[4]	"IID_IN_I_12BYTE"	
EPC[5]	"IID_IN_I_12BYTE"	
EPC[6]	"IID_IN_I_12BYTE"	
EPC[7]	"IID_IN_I_12BYTE"	
EPC[8]	"IID_IN_I_12BYTE"	
EPC[9]	"IID_IN_I_12BYTE"	
EPC[10]	"IID_IN_I_12BYTE"	
EPC[11]	"IID_IN_I_12BYTE"	
EPC[12]	"IID_IN_I_12BYTE"	
EPC[13]	"IID_IN_I_12BYTE"	
EPC[14]	"IID_IN_I_12BYTE"	
EPC[15]	"IID_IN_I_12BYTE"	
EPC[16]	"IID_IN_I_12BYTE"	
EPC[17]	"IID_IN_I_12BYTE"	
EPC[18]	"IID_IN_I_12BYTE"	
EPC[19]	"IID_IN_I_12BYTE"	

Note

Number of EPC-IDs

"number_MDS" specifies the number of EPC-IDs (1 to 19) transferred with the "INVENTORY" block. To receive the handle IDs of all transponders located in the antenna field, it may be necessary to run the "INVENTORY" block again with ATTRIBUTE "0x82".

Table 3- 13 ATTRIBUTE "0x83", "0x90", "0x91" und "0x92" (data type "IID_INVENT_8x_9x_RF6_D") for RF620R, RF630R with handle ID

Name	Type	Comment
reserved	BYTE	
number_MDS	BYTE	Number of MDS
UID	ARRAY[1..29] of "IID_IN_I_8BYTE"	

Name	Type	Comment
UID[1]	"IID_IN_I_8BYTE"	
UID	ARRAY[1..8] of BYTE	
UID[1]	BYTE	
UID[2]	BYTE	
UID[3]	BYTE	
UID[4]	BYTE	
UID[5]	BYTE	
UID[6]	BYTE	
UID[7]	BYTE	
UID[8]	BYTE	
UID[2]	"IID_IN_I_8BYTE"	
UID[3]	"IID_IN_I_8BYTE"	
UID[4]	"IID_IN_I_8BYTE"	
UID[5]	"IID_IN_I_8BYTE"	
UID[6]	"IID_IN_I_8BYTE"	
UID[7]	"IID_IN_I_8BYTE"	
UID[8]	"IID_IN_I_8BYTE"	
UID[9]	"IID_IN_I_8BYTE"	
UID[10]	"IID_IN_I_8BYTE"	
UID[11]	"IID_IN_I_8BYTE"	
UID[12]	"IID_IN_I_8BYTE"	
UID[13]	"IID_IN_I_8BYTE"	
UID[14]	"IID_IN_I_8BYTE"	
UID[15]	"IID_IN_I_8BYTE"	
UID[16]	"IID_IN_I_8BYTE"	
UID[17]	"IID_IN_I_8BYTE"	
UID[18]	"IID_IN_I_8BYTE"	
UID[19]	"IID_IN_I_8BYTE"	
UID[20]	"IID_IN_I_8BYTE"	
UID[21]	"IID_IN_I_8BYTE"	
UID[22]	"IID_IN_I_8BYTE"	
UID[23]	"IID_IN_I_8BYTE"	
UID[24]	"IID_IN_I_8BYTE"	
UID[25]	"IID_IN_I_8BYTE"	
UID[26]	"IID_IN_I_8BYTE"	
UID[27]	"IID_IN_I_8BYTE"	
UID[28]	"IID_IN_I_8BYTE"	
UID[29]	"IID_IN_I_8BYTE"	
reserved1	DWORD	
Data	ARRAY[1..222] of BYTE	

Note

Number of handle IDs

"number_MDS" specifies the number of handle IDs (1 to 29) transferred with the "INVENTORY" block. To receive the handle IDs of all transponders located in the antenna field, it may be necessary to run the "INVENTORY" block again with ATTRIBUTE "0x82".

Table 3- 14 ATTRIBUTE "0x85" (data type "IID_INVENT_85_RF600")

Name	Type	Comment
reserved	BYTE	
number MDS	STRUCT	Number of MDS
ID	BYTE	
ID[1]	BYTE	
Handle	ARRAY[1..8] of BYTE	
Handle[1]	BYTE	
Handle[2]	BYTE	
Handle[3]	BYTE	
Handle[4]	BYTE	
Handle[5]	BYTE	
Handle[6]	BYTE	
Handle[7]	BYTE	
Handle[8]	BYTE	
EPC	ARRAY[1..12] of BYTE	
EPC[1]	BYTE	
EPC[2]	BYTE	
EPC[3]	BYTE	
EPC[4]	BYTE	
EPC[5]	BYTE	
EPC[6]	BYTE	
EPC[7]	BYTE	
EPC[8]	BYTE	
EPC[9]	BYTE	
EPC[10]	BYTE	
EPC[11]	BYTE	
EPC[12]	BYTE	
ID[2]	"IID_IN_I_20Byte"	
ID[3]	"IID_IN_I_20Byte"	
ID[4]	"IID_IN_I_20Byte"	
ID[5]	"IID_IN_I_20Byte"	
ID[6]	"IID_IN_I_20Byte"	
ID[7]	"IID_IN_I_20Byte"	

Name		Type	Comment
	ID[8]	"IID_IN_I_20Byte"	
	ID[9]	"IID_IN_I_20Byte"	
	ID[10]	"IID_IN_I_20Byte"	
	ID[11]	"IID_IN_I_20Byte"	

Note

Number of IDs transferred

"number_MDS" specifies the number of IDs (1 to 11 handle IDs and EPC-IDs) transferred with the "INVENTORY" block. To receive the IDs of all transponders located in the antenna field, it may be necessary to run the "INVENTORY" block again with ATTRIBUTE "0x82".

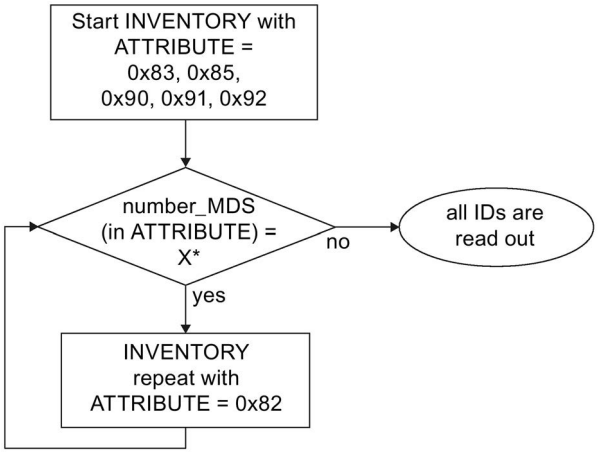
You will find more detailed information on the individual status modes in the manuals matching the modes "FB 45", "FB55" and "SIMATIC RF620R/RF630R".

The identifiers of the status modes correspond to the following identifiers in the other manuals:

0x82 ≙ 0x02
0x83 ≙ 0x03
0x85 ≙ 0x05
0x90 ≙ 0x10
0x91 ≙ 0x11
0x92 ≙ 0x12

Programming ATTRIBUTE "0x82"

If the number of transponders in the antenna field is unknown, repeat the "INVENTORY" block with the ATTRIBUTE = "0x82".



* The number of returned IDs "X" depends on the "ATTRIBUTE" used.

Figure 3-13 Program sequence of ATTRIBUTE "0x82" with unknown transponder populations

Results for RF680R, RF685R

The number of "TAG_DATA[x]" elements of the data types of the ATTRIBUTES "0x80" and "0x81" depends on the number of transponders to be expected. For this reason, you need to assemble the receive buffer yourself. Not the following structure when creating the receive buffer :"IDENT_DATA"/data type:

- The first element "NUM_MDS" is always of the type "WORD".
- The next element "TAG_DATA" is always of the type "ARRAY". The number of transponders to be expected ("n") must be entered in the "ARRAY".

The following tables show an example of the structure of the receive buffer "IDENT_DATA"/data type for the ATTRIBUTES "0x80" and "0x81".

Table 3- 15 ATTRIBUTE "0x80"

Name	Type	Comment
NUM_MDS	WORD	Number of MDS
TAG_DATA	ARRAY[1..n]of IID_IN_I_80	Length of EPC ID
TAG_DATA[1]	IID_IN_I_80	
Reserved	BYTE	
ID_Len	BYTE	Length of EPC ID
EPC_ID	ARRAY[1..62] of BYTE	EPC-ID
tagPC	WORD	
TAG_DATA[2]	IID_IN_I_80	
...	...	
TAG_DATA[n]	IID_IN_I_80	

Table 3- 16 ATTRIBUTE "0x81"

Name	Type	Comment
NUM_MDS	WORD	Number of MDS
TAG_DATA	ARRAY[1..n] of IID_IN_I_81	
TAG_DATA[1]	IID_IN_1_81	
reserved	BYTE	
ID_LEN	BYTE	EPC length
EPC_ID	ARRAY[1..62]of BYTE	EPC-ID
tagPC	WORD	
RSSI	BYTE	RSSI value
MaxRSSI	BYTE	highest RSSI value
MinRSSI	BYTE	lowest RSSI value
channel	BYTE	channel; 1..15_ESTI; 1..53:FCC
antenna	BYTE	antenna; bit coded; Bit 0=antenna 1; Bit 1=antenna 2
polarization	BYTE	polarizatuin of antenna; 0=undefined; 1=circular
time	Time_OF_Day	S7 time
power	BYTE	power in dBm

Name	Type	Comment
filterDataAvailable	BYTE	0=false; 1=true
Inventoried	WORD	1)
TAG_DATA[2]	IID_IN_1_81	
...	...	
TAG_DATA[n]	IID_IN_1_81	

1) Indicates how often the transponder was identified via the air interface before it changed to the "Observed" status.

3.4.2.3 Read_EPC_Mem

The "Read_EPC_Mem" block reads data starting at address 4 from the EPC memory of the RF600 transponder. Access is to memory cell 1 as of the start address. The length of the EPC memory to be read out is specified by the "LEN_DATA" parameter.

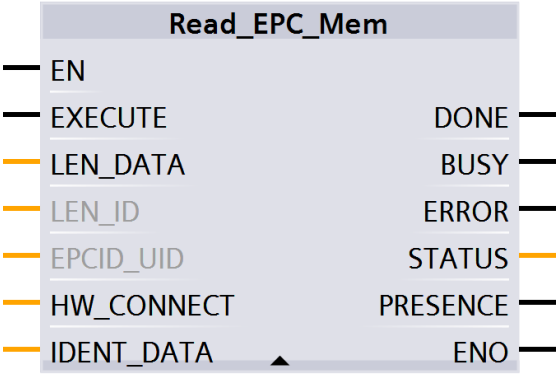


Figure 3-14 "Read_EPC_Mem" block

Table 3- 17 Explanation of the "Read_EPC_Mem" block

Parameter	Data type	Default values	Description
LEN_DATA	Word	W#16#0	Length of the EPC memory to be read out (1 ... 62 bytes)
LEN_ID	Byte	B#16#0	Length of the EPC-ID/UID Default value: 0x00 ≙ unspecified single tag access (RF680R, RF685R)

Parameter	Data type	Default values	Description
EPCID_UID	Array[1...62] of Byte	0	Buffer for up to 62 bytes EPC-ID, 8 bytes UID or 4 bytes handle ID. • 2 - 62-byte EPC-ID is entered at the start of the buffer (length is set by "LEN_ID") • 8-byte UID is entered at the start of the buffer ("LEN_ID = 8") • 4-byte handle ID must be entered in the array element [5]-[8] ("LEN_ID = 8") Default value: 0x00 ≙ unspecified single tag access (RF620R, RF630R)
IDENT_DATA	Any / Variant	0	Data buffer in which the read EPC memory data is stored. Note: For Variant, currently only an "Array_of_Byte" with a variable length can be created. For Any, other data types/UDTs can also be created.

3.4.2.4 Read_TID

The "Read_TID" block reads data from the TID memory area (Tag Identification Memory Bank) of the RF600 transponder. The length of the TID to be read is specified by the "LEN_DATA" parameter. The length of the TID varies depending on the transponder and can be found in the transponder data sheet.

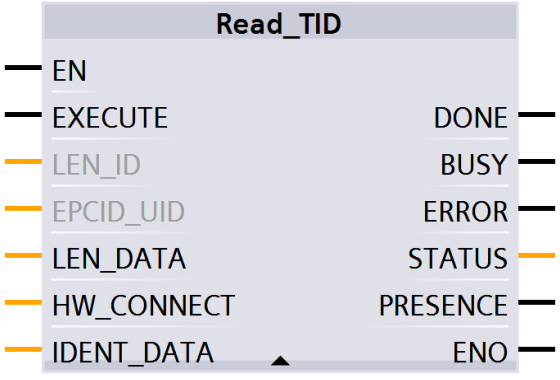


Figure 3-15 "Read_TID" block

Table 3- 18 Explanation of the "Read_TID" block

Parameter	Data type	Default values	Description
LEN_ID	Byte	B#16#0	Length of the EPC-ID/UID Default value: 0x00 ≙ unspecified single tag access (RF680R, RF685R)
EPCID_UID	Array[1...62] of Byte	0	Buffer for up to 62 bytes EPC-ID, 8 bytes UID or 4 bytes handle ID. • 2 - 62-byte EPC-ID is entered at the start of the buffer (length is set by "LEN_ID") • 8-byte UID is entered at the start of the buffer ("LEN_ID = 8") • 4-byte handle ID must be entered in the array element [5]-[8] ("LEN_ID = 8") Default value: 0x00 ≙ unspecified single tag access (RF620R, RF630R)
LEN_DATA	Word	W#16#4	Length of the EPC memory to be read out (1 ... 62 bytes)
IDENT_DATA	Any / Variant	0	Read TID Note: For Variant, currently only an "Array_of_Byte" with a variable length can be created. For Any, other data types/UDTs can also be created.

3.4.2.5 Read_UID

The "Read_UID" block reads the UID of an HF transponder. The UID always has a fixed length of 8 bytes.

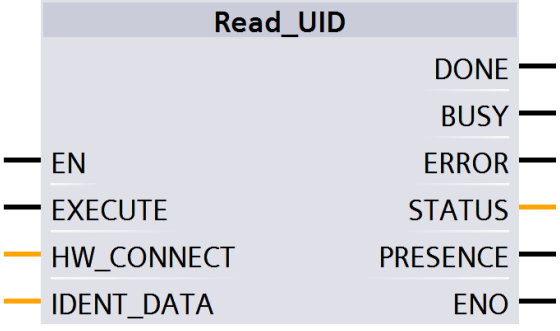


Figure 3-16 "Read_UID" block

Table 3- 19 Explanation of the "Read_UID" block

Parameter	Data type	Description
IDENT_DATA	Any / Variant	UID Note: For Variant, currently only an "Array_of_Byte" with a variable length can be created. For Any, other data types/UDTs can also be created.

3.4.2.6 Set_Ant

With the aid of the "Set_Ant" block, you can turn antennas on or off. There are different blocks for RF300 and RF600. The "Set_Ant_RF300" block can also be used for RF200, MOBY D and MOBY U. The "Set_Ant_RF600" block relates only to the RF620R and RF630R readers.

Set_Ant_RF300

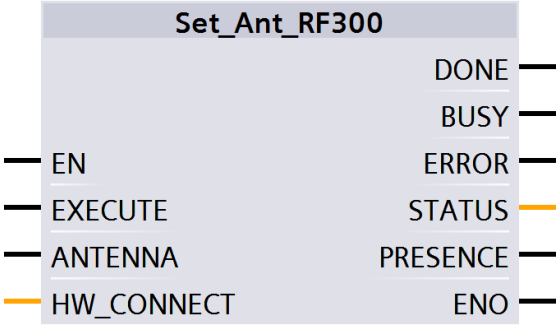


Figure 3-17 "Set_Ant_RF300" block

Table 3- 20 Explanation of the "Set_Ant_RF300" block

Parameter	Data type	Description
ANTENNA	Bool	0 = turn antenna off 1 = turn antenna on

Set_Ant_RF600

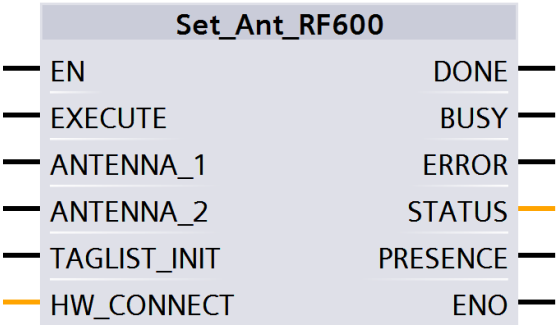


Figure 3-18 "Set_Ant_RF600" block

Table 3- 21 Explanation of the "Set_Ant_RF600" block

Parameter	Data type	Description
ANTENNA_1	Bool	0 = turn antenna 1 off 1 = turn antenna 1 on
ANTENNA_2	Bool	0 = turn antenna 2 off 1 = turn antenna 2 on
TAGLIST_INIT	Bool	0 = TagList is reset 1 = the existing TagList continues to be used

3.4.2.7 Set_Param

With the "Set_Param" block, you can change UHF parameters on an RF680R/RF685R during runtime (e.g. the antenna power).

Note

Settings saved only temporarily

Note that the parameters in the "Set_Param" block are only stored temporarily. If the power for the reader is interrupted, the stored values are lost and must be set again.

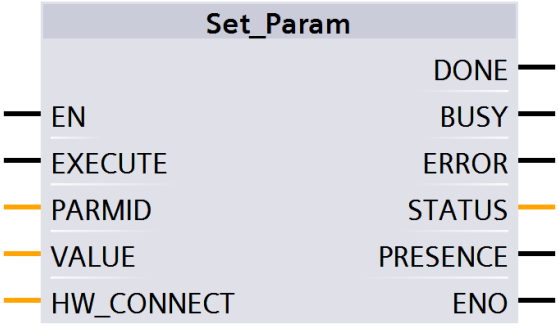


Figure 3-19 "Set_Param" block

Table 3- 22 Explanation of the "Set_Param" block

Parameter	Data type	Default values	Description
PARMID	DWORD	0x00	Parameter identifier
VALUE	DWORD	0x00	Parameter value

Table 3- 23 Parameter values

PARMID (hex)	PARMID (ASCII)	Parameter	VALUE
0x41315057	A1PW	Antenna 01: Radiated power	Range of values: 0.5 ... 33 Increment: 0.25 Radiated power of the antenna in [dBm]. Bytes 1 and 2 are not used, byte 3 represents the integer and byte 4 the decimal place. Example: A radiated power of 10.25 dBm represents a "VALUE" of "0x0A19".
0x41325057	A2PW	Antenna 02: Radiated power	
0x41335057	A3PW	Antenna 03: Radiated power	
0x41345057	A4PW	Antenna 04: Radiated power	
0x41315452	A1TR	Antenna 01: RSSI threshold	Range of values: 0 ... 255 Threshold value for RSSI. Transponders with lower values are discarded. Value without unit without a direct relationship with the radiated power.
0x41325452	A2TR	Antenna 02: RSSI threshold	
0x41335452	A3TR	Antenna 03: RSSI threshold	
0x41345452	A4TR	Antenna 04: RSSI threshold	
0x5331444C	S1DL	Read point 01: RSSI delta	Range of values: 0 ... 255 Difference for RSSI values. Transponders with lower values relative to the transponder with the highest RSSI value are discarded. Value without unit without a direct relationship with the radiated power.
0x5332444C	S2DL	Read point 02: RSSI delta	
0x5333444C	S3DL	Read point 03: RSSI delta	
0x5334444C	S4DL	Read point 04: RSSI delta	
0x4131504F	A1PO	Antenna 01: Polarization	Range of values: 0, 1, 2, 4 Polarization of the antenna (for intelligent antennas e.g. internal antenna RF685R) • 0: default, undefined • 1: circular • 2: vertical linear • 4: horizontal linear Input is bit coded. Combinations are possible (adding values).
0x4132504F	A2PO	Antenna 02: Polarization	
0x4133504F	A3PO	Antenna 03: Polarization	
0x4134504F	A4PO	Antenna 04: Polarization	
0x52364353	R6CS	Modulation scheme	Range of values: 32, 33, 34, 35, 37, 65 Modulation scheme of the read point Specification of which transponder types are identified (ISO 18000-63/-6B).

PARMID (hex)	PARMID (ASCII)	Parameter	VALUE
0x57544348	WTCH	Date and time	Range of values: 01.01.2000 00:00 a.m. ... 19.01.2038 3:14 a.m. 01.01.2000 01:00 a.m. ≙ 946684800 Date and time (UTC) Time in seconds since 01.01.1970; Setting of the internal reader clock, the date and time are set.
0x57544F44	WTOD	Time	Range of values: 0:00 – 23:59 p.m. S7 time (TOD, UTC) Milliseconds since midnight; Setting of the internal reader clock, only the time is changed but not the date.
0x57444154	WDAT	Date	Range of values: 01.01.2000 ... 18.01.2038 S7 date Days since 01.01.1970; Setting of the internal reader clock, only the date is changed but not the time.

3.4.2.8 Write_EPC_ID

The "Write_EPC_ID" block overwrites the EPC-ID of the RF600 transponder and adapts the length of the EPC-ID in the memory of the transponder. The new EPC-ID length to be written is specified with the "LEN_ID_NEW" parameter and the previous EPC-ID is specified using the "LEN_ID" and "EPCID_UID" parameters.

Make sure that when you execute the block only one transponder is located in the antenna field. This ensures that the identification when writing the ID is unique. If there is more than one transponder in the antenna field, a negative response is returned.

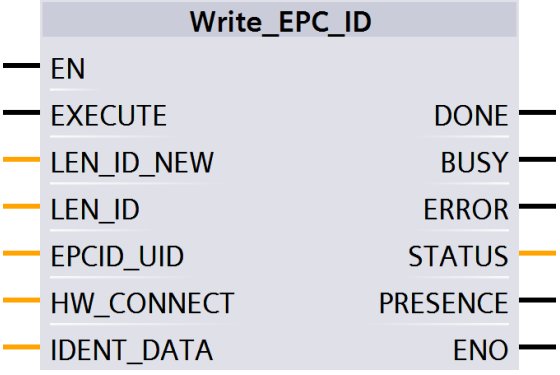


Figure 3-20 "Write_EPC_ID" block

Table 3- 24 Explanation of the "Write_EPC_ID" block

Parameter	Data type	Default values	Description
LEN_ID_NEW	Byte	W#16#0C	Length of the current EPC-ID
LEN_ID	Byte	B#16#0	Length of the previous EPC-ID
EPCID_UID	Array[1...62] of Byte	0	Previous EPC ID
IDENT_DATA	Any / Variant	0	Current EPC ID Note: For Variant, currently only an "Array_of_Byte" with a variable length can be created. For Any, other data types/UDTs can also be created.

3.4.2.9 Write_EPC_Mem

The "Write_EPC_Mem" block overwrites the EPC memory of the RF600 transponder starting at address 4. The length of the EPC memory to be overwritten is specified by the "LEN_DATA" parameter.

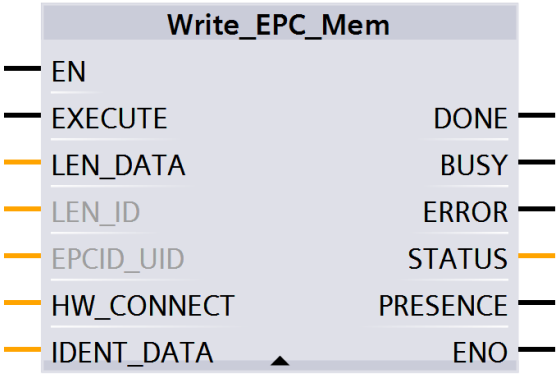


Figure 3-21 "Write_EPC_Mem" block

Table 3- 25 Explanation of the "Write_EPC_Mem" block

Parameter	Data type	Default values	Description
LEN_DATA	Word	W#16#0	Length of the EPC memory to be overwritten (1 ... 62 bytes)
LEN_ID	Byte	B#16#0	Length of the EPC-ID/UID Default value: 0x00 ≙ unspecified single tag access (RF680R, RF685R)

Parameter	Data type	Default values	Description
EPCID_UID	Array[1...62] of Byte	0	Buffer for up to 62 bytes EPC-ID, 8 bytes UID or 4 bytes handle ID. • 2 - 62-byte EPC-ID is entered at the start of the buffer (length is set by "LEN_ID") • 8-byte UID is entered at the start of the buffer ("LEN_ID = 8") • 4-byte handle ID must be entered in the array element [5]-[8] ("LEN_ID = 8") Default value: 0x00 ≙ unspecified single tag access (RF620R, RF630R)
IDENT_DATA	Any / Variant	0	Data buffer with the EPC memory data to be overwritten. Note: For Variant, currently only an "Array_of_Byte" with a variable length can be created. For Any, other data types/UDTs can also be created.

3.4.2.10 AdvancedCMD

With the "AdvancedCmd" block, every command can be executed including commands not represented by other blocks. This general structure can be used for all commands and is intended only for trained users.

This gives you the option of sending the command as a chained command. To allow this, the block provides a CMD buffer for 100 commands. All chained commands must be entered starting at the first position in the buffer. For every chained command, the "chained bit" must also be set in the CMD structure. The "chained bit" is not set in the last command in the chain. You will find further information on the "chained bit" in the section "Chaining (Page 96)".

The entire command structure must be specified in the "CMD" input parameter. You create the structure for the "CMD" parameter in a data block.

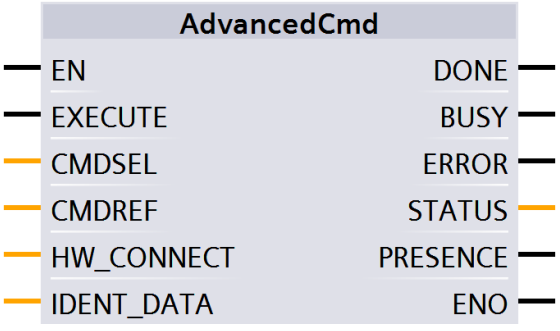


Figure 3-22 "AdvancedCmd" block

Table 3- 26 Explanation of the "AdvancedCmd" block

Parameter	Data type	Default values	Description
CMDSEL	Int	1	Selection of the command to be executed "CMDREF"; 1 ⇒ 1. Command, ... The value of the "CMDSEL" parameter can never be > 100.
CMDREF	Any / Variant	--	You will find a detailed description of the parameter in the sections: • "Commands of the Ident profile (Page 70)" • "Command structure (Page 72)"
IDENT_DATA	Any / Variant	0	Buffer for data to be written or read. Note: For Variant, currently only an "Array_of_Byte" with a variable length can be created. For Any, other data types/UDTs can also be created.

3.4.3 Reset blocks

The reset blocks described in this section are required when you want to operate the optical reader systems MV420, MV440 or the communications modules RF180C, ASM 456 with a SIMATIC S7-1200/S7-1500 controller. As an alternative you can also use these blocks for the RF120C if you have selected the appropriate setting in the device configuration.

In the system, these reset blocks have the same function as the "Reset_Reader" block described earlier. However, with the blocks described here, you need to set reader-dependent parameters.

Remember that the default value will be used automatically if you do not select a value manually.

3.4.3.1 Reset_MOBY_D

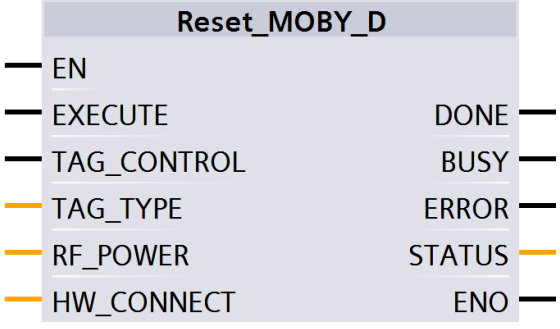


Figure 3-23 "Reset_MOBY_D" block

Table 3- 27 Explanation of the "Reset_MOBY_D" block

Parameter	Data type	Default values	Description
TAG_CONTROL	Bool	True	Presence check
TAG_TYPE	Byte	1	Transponder type: • 1 = every ISO transponder
RF_POWER	Byte	0	Output power RF power from 0.5 W to 10 W in increments of 0.25 W (range of values: 0x02 ... 0x28)

3.4.3.2 Reset_MOBY_U

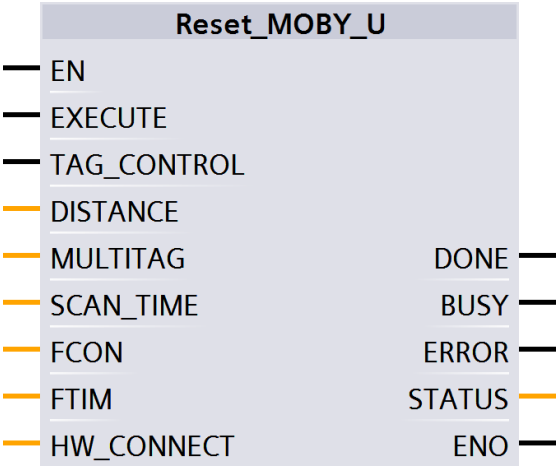


Figure 3-24 "Reset_MOBY_U" block

Table 3- 28 Explanation of the "Reset_MOBY_U" block

Parameter	Data type	Default values	Description
TAG_CONTROL	Bool	True	Presence check
DISTANCE	Byte	23h	Range limitation (range of values: 0x02 ... 0x23 or 0x82 ... 0xA3 for reduced transmit power)
MULTITAG	Byte	1	Maximum number of transponders that can be processed at the same time in the antenna field. (Range of values: 0x01 ... 0x12)
SCAN_TIME	Byte	0	Scanning time: Standby time of the transponder (range of values: 0x00 ... 0xC8)
FCON	Byte	0	field_ON_control: BERO mode (range of values: 0x00 ... 0x03)
FTIM	Byte	0	field_ON_time: Time for BERO mode (range of values: 0x00 ... 0xFF)

3.4.3.3 Reset_MV

To reset cameras of the optical reader systems, call the block and activate the "EXECUTE" parameter.

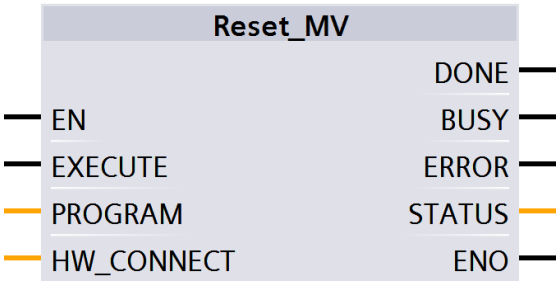


Figure 3-25 "Reset_MV" block

Table 3- 29 Explanation of the "Reset_MV" block

Parameter	Data type	Description
PROGRAM	Byte	Program selection • B#16#0: Reset without program selection or in the case of diagnostics, the error code for "IN_OP = 0" is shown at the "STATUS" output parameter. • B#16#1 ... B#16#0F: Number of the program to be started ⇒ Reset with program selection (as of firmware V5.1 of the MV4x0)

3.4.3.4 Reset_RF200

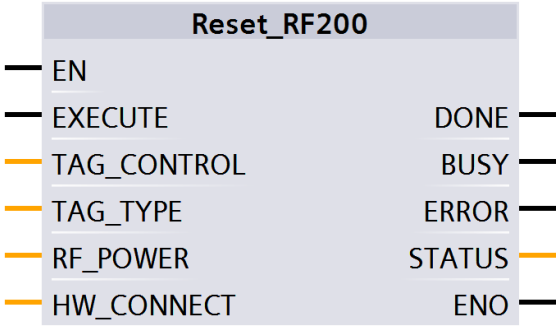


Figure 3-26 "Reset_RF200" block

Table 3- 30 Explanation of the "Reset_RF200" block

Parameter	Data type	Default values	Description
TAG_CONTROL	Byte	1	Presence check
TAG_TYPE	Byte	1	Transponder type: • 1 = every ISO transponder • 3 = MDS D3xx - optimization
RF_POWER	Byte	4	Output power; only relevant for RF290R RF power from 0.5 W to 5 W in increments of 0.25 W (range of values: 0x02 ... 0x14). Default value 0x04 ≙ 1 W.

3.4.3.5 Reset_RF300

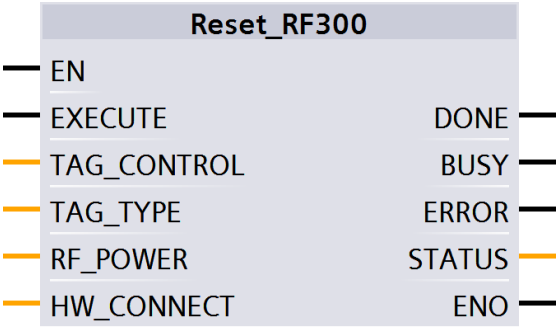


Figure 3-27 "Reset_RF300" block

Table 3- 31 Explanation of the "Reset_RF300" block

Parameter	Data type	Default values	Description
TAG_CONTROL	Byte	1	Presence check • 0 = Off • 1 = on • 4 = presence (antenna is off. The antenna is turned on only when a Read or Write command is sent.)
TAG_TYPE	Byte	0	Transponder type: • 1 = every ISO transponder • 0 = RF300 transponder
RF_POWER	Byte	0	Output power; only relevant for RF380R RF power from 0.5 W to 2 W in increments of 0.25 W (range of values: 0x02 ... 0x08). Default value 0x00 ≙ 1.25 W.

3.4.3.6 Reset_RF600

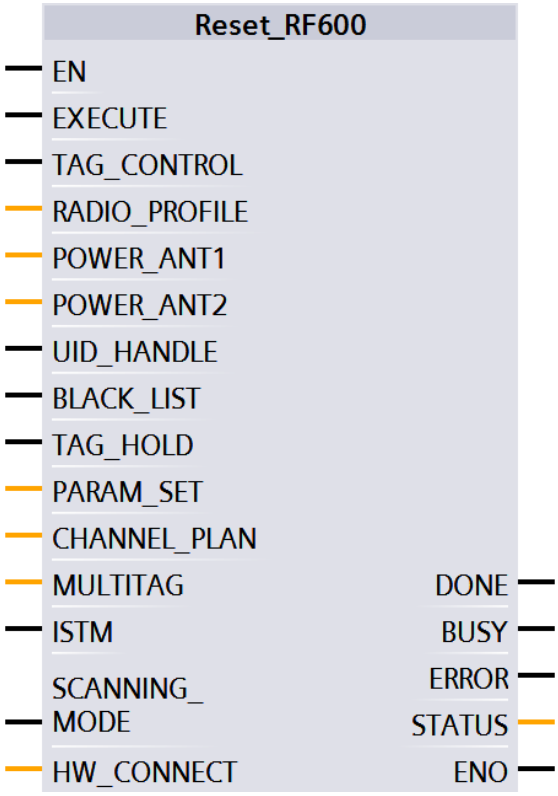


Figure 3-28 "Reset_RF600" block

Table 3- 32 Explanation of the "Reset_RF600" block

Parameter	Data type	Default values	Description
TAG_CONTROL	Bool	True	Presence check
RADIO_PROFILE	Byte	1	Scanning time: Wireless profile according to EPC Global (range of values: 0x01 ... 0x09 depending on the reader variant)
POWER_ANT1	Byte	0	Transmit power for antenna 1 or internal antenna (range of values: 0x00 ... 0x0F)
POWER_ANT2	Byte	0	Transmit power for antenna 2 or external antenna (range of values: 0x00 ... 0x0F)
UID_HANDLE	Bool	False	Meaning of the UID in the command: True = Handle ID, only the least significant 4 bytes of the UID are evaluated; False = UID/EPC-ID with a length of 8 bytes
BLACK_LIST	Bool	False	True = activate black list
TAG_HOLD	Bool	False	True = activate Tag Hold

Parameter	Data type	Default values	Description
PARAM_SET	Byte	0	Field_ON_Control (0 = fast; range of values: 0x00, 0x02)
CHANNEL_PLAN	Byte	0F	Field_ON_Time (range of values: 0x00 ... 0x0F; ETSI only)
MULTITAG	Byte	1	Maximum number of transponders that can be processed at the same time in the antenna field. (Range of values: 0x01 ... 0x50)
ISTM	Bool	False	True = activate intelligent single tag mode
SCANNING_MODE	Bool	False	True = activate scanning mode [1]

[1] Is not currently possible with the Ident blocks.

3.4.3.7 Reset_Univ

The "Reset_Univ" block is a universal reset block with which all identification systems can be reset. Use this block only after consulting Support.

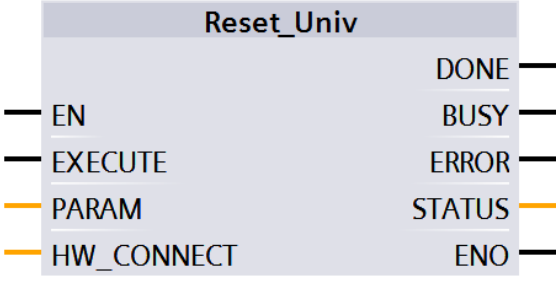


Figure 3-29 "Reset_Univ" block

Table 3- 33 Explanation of the "Reset_Univ" block

Parameter	Data type	Description
PARAM	Array [1...16] of Byte	Data for reset frame The data to be set here can be made available by Support when necessary for special settings.

Table 3- 34 Structure of the "PARAM" parameter

Byte	1	2...5	6	7...8	9	10	11	12	13...14	15	16
Value	04h	0	0Ah	0	scan-ning_ time	param	option_ 1	dis-tance_ limiting	Number of tran-sponders	field_ on_ control	field_ on_ time

3.4.4 Status blocks

3.4.4.1 Reader_Status

The "Reader_Status" block reads status information from the reader. For the various reader families, there are different status modes that you can select using the "ATTRIBUTE" parameter.

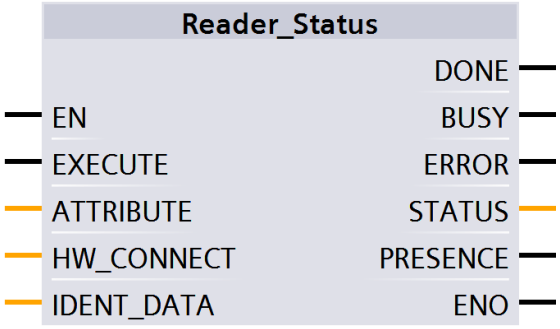


Figure 3-30 "Reader_Status" block

Table 3- 35 Explanation of the "Reader_Status" block

Parameter	Data type	Default values	Description
ATTRIBUTE	Byte	B#16#81	Identifier of the status modes / possible entries: • RF200: 0x81 • RF300: 0x81, 0x86, 0xEF • RF620R, RF630R: 0x87, 0x88, 0xA0, 0xA1 • RF680R, RF685R: 0x89 • MOBY U: 0x81, 0x84, 0x85 • MOBY D: 0x81
IDENT_DATA	Any / Variant	0	Event values depending on attributes Note: For Variant, an "Array_of_Byte" with a variable length can be created. For Any, other data types/UDTs can also be created.

Results

Apply the correct data type that is assigned to the ATTRIBUTE value at the "IDENT_DATA" input of the block so that the data can be correctly interpreted.

Note that the UDTs can only be used when the blocks "Reader_Status" or "Tag Status" used.

Table 3- 36 ATTRIBUTE "0x81" (data type "IID_READSTAT_81_RF2_3_U")

Name	Type	Comment
status info	BYTE	SLG status mode
hardware	CHAR	Type of hardware
hardware version	WORD	Version of hardware
loader version	WORD	Version of loader
firmware	CHAR	Type of firmware
firmware version HB	BYTE	Version of firmware
firmware_version_LB	BYTE	
driver	CHAR	Type of driver
driver version	WORD	Version of driver
interface	BYTE	Type of interface (RS 232/RS 422)
baud	BYTE	Baudrate
reserved1	BYTE	Reserved
reserved2	BYTE	Reserved
reserved3	BYTE	Reserved
distance limiting SLG	BYTE	Distance limiting of SLG
multitag SLG	BYTE	Multitag SLG
field ON control SLG	BYTE	Field ON control
field ON time SLG	BYTE	Field On time
sync SLG	BYTE	Synchronization with SLG
status ant	BYTE	Status of antenne
stand by	BYTE	Time of standby after command
MDS control	BYTE	Presence mode

Table 3- 37 ATTRIBUTE "0x84" (data type "IID_READSTAT_84_MOBY_U")

Name	Type	Comment
status info	BYTE	SLG status mode
number MDS	BYTE	Range 1..24
UID	ARRAY [1..24] of DWord	

Table 3- 38 ATTRIBUTE "0x86" (data type "IID_READSTAT_86_RF300")

Name	Type	Comment
status info	BYTE	SLG status mode
FZP	BYTE	Error counter passive: distortion without communication
ABZ	BYTE	Dropout counter
CFZ	BYTE	Code error counter
SFZ	BYTE	Signature error counter
CRCFZ	BYTE	CRC-error counter
BSTAT	BYTE	Status of last command
ASMFZ	BYTE	Error counter for host interface (ASM)
reserved0	ARRAY [1..17]	

Table 3- 39 ATTRIBUTE "0x87" (data type "IID_READSTAT_87_RF600")

Name	Type	Comment
status_info	BYTE	SLG status mode
hardware	CHAR	Type of hardware
hardware_version	WORD	Version of hardware
reserved0	WORD	
firmware	CHAR	Type of firmware
firmware_version_HB	BYTE	Version of firmware highbyte
firmware_version_LB	BYTE	Version of firmware lowbyte
driver	CHAR	Type of driver
current_time_hour	BYTE	Hours [1]
current_time_min	BYTE	Minutes
current_time_sec	BYTE	Seconds
reserved1	BYTE	
SLG_version	BYTE	SLG version
baud	BYTE	Baudrate
reserved2	BYTE	
distance_limiting_SLG	BYTE	Selected transmit power
multitag_SLG	BYTE	Multitag SLG
field_ON_control_SLG	BYTE	Selected comunication typ
field_ON_time_SLG	BYTE	Selected channel
expert_mode	BYTE	Expert mode
status_ant	BYTE	Status of antenna [2]
scanning_time_SLG	BYTE	Radio communication profile (country specific radio standart)
MDS_control	BYTE	Presence mode

1) The internal time stamp of the reader that relates to this event is output. The internal reader time stamp is not synchronized with UTC.

2) The antenna status relates to the "ATTRIBUTE" (bits 0 and 1) of the last executed "SET-ANT" or to the default value set by "init-run". In "init_run" of the RF620R, the default value is "1" (int. antenna on), with the RF630R, it is "3" (antennas 1 and 2 on).

Table 3- 40 ATTRIBUTE "0x88" (data type "IID_READSTAT_88_RF600")

Name	Type	Comment
status_info	BYTE	SLG status mode(Subcommand)
hardware	CHAR	Type of hardware
hardware_version	WORD	Version of hardware
reserved_word1	WORD	Reserved
firmware	CHAR	Type of firmware
firmware_version_HB	BYTE	Version of firmware (High-Byte)
firmware_version_LB	BYTE	Version of firmware (Low-Byte)
driver	CHAR	Type of driver
current_time_hour	BYTE	Hours [1]
current_time_minute	BYTE	Minutes [1]
current_time_sec	BYTE	Seconds [1]
current_time_reservByte	BYTE	
SLG_version	BYTE	SLG-Version
baud	BYTE	Baudrate
reserved_byte1	BYTE	Reserved
distance_limiting_SLG	BYTE	Selected transmit power
multitag_SLG	BYTE	Multitag SLG
field_ON_control_SLG	BYTE	Selected communication type
field_ON_time_SLG	BYTE	Selected channel

Name	Type	Comment
expert mode	BYTE	Expert mode
status_ant	BYTE	Status of antenna 2)
scanning_time_SLG	BYTE	Radio communication profile (country specific radio standart)
MDS control	BYTE	Presence mode
blink pattern	BYTE	Blink Pattern
act algor Single Tag	Bool	Single Tag [1]
act algor ITF Phase2	Bool	ITF Phase2 [2]
act algor ITF Phase1	Bool	ITF Phase1 [3]
act algor Smoothing	Bool	Smoothing [4]
act algor Blacklist	Bool	Blacklist [5]
act algor RSSI Threshold	Bool	RSSI Threshold [6]
act algor Power Ramp	Bool	Power Ramp [7]
act algor Power Gap	Bool	Power Gap [8]
Reserved1	Bool	Reserved1 [1]
Reserved2	Bool	Reserved2 [2]
Reserved3	Bool	Reserved3 [3]
Reserved4	Bool	Reserved4 [4]
act algor EPC MemBankFilter	Bool	EPC MemBankFilteres [5]
act algor Tag Holg	Bool	Tag Hold [6]
act algor Multi Tag	Bool	Multi Tag [7]
act algor ISTM	Bool	ISTM [8]
reserved word2	WORD	Reserved
reserved word3	WORD	Reserved
reserved word4	WORD	Reserved
filtered max rssi	BYTE	Maximum RSSI value of a tag, of all filtered tags
reserved byte2	BYTE	Reserved
filtered tags rssi	BYTE	Number of tags, filtered out by the RSSI threshold
reserved byte3	BYTE	Reserved
filtered tags black list	WORD	Number of tags, filtered out via Black-List
filtered tags epc data	WORD	Number of tags, filtered out via EPC Data Filter
filtered tags smoothing	WORD	Number of tags in Tag List of status Not-Observed
itf ph1 max detect	WORD	Number of reads of a Tag, filtered out via ITF-phase 1
itf ph1 tags detect	WORD	Number of tags, filtered out via ITF-phase 1
itf ph2 max detect	WORD	Number of reads of a Tag, filtered out via ITF-phase 2
itf ph2 tags detect	WORD	Number of tags, filtered out via ITF-phase 2
filtered_istm_min_dist	WORD	Minimum distance of tags according to sorting criterion of ISTM
filtered istm tags	WORD	Number of tags, filtered out via ISTM algorithm
last error	BYTE	error code of the last occuring error (last command)
reserved byte4	BYTE	Reserved
error command1	WORD	Last command (has lead to error code) "last error"
error command2	WORD	Last command (has lead to error code) "last error"
error command3	WORD	Last command (has lead to error code) "last error"
reserved word5	WORD	Reserved
reserved_array_byte	ARRAY[1..30] of Byte	

Table 3- 41 ATTRIBUTE "0x89" (data type "IID_READSTAT_89_RF68xR")

Name	Type	Comment
status info	BYTE	SLG status mode(Subcommand)
hardware version	BYTE	Version of hardware
firmware_version	ARRAY[1..4] of CHAR	Version of firmware
config ID	DWORD	Unix timestamp
inventory_status	WORD	0=inventory not active; 1=inventory active; 2=presence mode active
sum of filtered tags	WORD	All filtered Tags
filtered smoothing	WORD	Filtered Tags trough Smoothing
filtered blacklist	WORD	Filtered Tags trough Blacklist

Name	Type	Comment
filtered data-filter	WORD	Filtered Tags trough Data-Filter
filtered RSSI threshold	WORD	Filtered Tags trough RSSI Threshold
filtered RSSI delta	WORD	Filtered Tags trough RSSI Delta

Table 3- 42 ATTRIBUTE "0xA0" and "0xA1" (data type "IID_READSTAT_A0_A1_RF600")

Name	Type	Comment
reserved	BYTE	
Status info	BYTE	Status-Info, SLG-Status SubCommand 20/21
number tags frame	BYTE	Number of Tags in this frame
number tags next frames	BYTE	Number of Tags in the next frames
reserved byte1	BYTE	Reserved
reserved byte2	BYTE	Reserved
reserved byte3	BYTE	Reserved
reserved byte4	BYTE	Reserved
reserved byte5	BYTE	Reserved
reserved byte6	BYTE	Reserved
Black_List_ID	ARRAY[1..13] of "IID_IN_Blacklist"	EPC-ID Length
Black_List_ID[1]	"IID_IN_Blacklist"	
EPC_Length	BYTE	EPC-ID Length
Antenna	BYTE	Antenna = Default 3
Filtered_Tag	WORD	Number of times - EPC-ID filtered out via BlackList
EPC	ARRAY[1..12] of Byte	EPC-ID
Black_List_ID[2]	"IID_IN_Blacklist"	
Black_List_ID[3]	"IID_IN_Blacklist"	
Black_List_ID[4]	"IID_IN_Blacklist"	
Black_List_ID[5]	"IID_IN_Blacklist"	
Black_List_ID[6]	"IID_IN_Blacklist"	
Black_List_ID[7]	"IID_IN_Blacklist"	
Black_List_ID[8]	"IID_IN_Blacklist"	
Black_List_ID[9]	"IID_IN_Blacklist"	
Black_List_ID[10]	"IID_IN_Blacklist"	
Black_List_ID[11]	"IID_IN_Blacklist"	
Black_List_ID[12]	"IID_IN_Blacklist"	
Black_List_ID[13]	"IID_IN_Blacklist"	

You will find more detailed information on the individual status modes in the manuals matching the modes "FB 45", "FB55" and "SIMATIC RF620R/RF630R".

The identifiers of the status modes correspond to the following identifiers in the other manuals:

0x81	≙	0x01
0x82	≙	0x02
0x83	≙	0x03
0x85	≙	0x05
0x87	≙	0x07
0x88	≙	0x08
0x90	≙	0x10
0x91	≙	0x11
0x92	≙	0x12
0xA0	≙	0x20
0xA1	≙	0x21

3.4.4.2 Tag_Status

The "Tag_Status" block reads the status information of the transponder. For the various transponder types and reader families, there are different status modes that you can select using the "ATTRIBUTE" parameter.

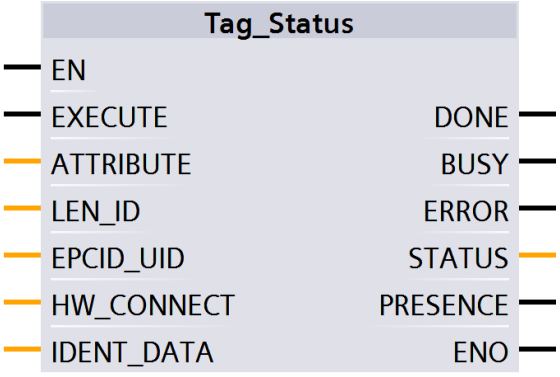


Figure 3-31 "Tag_Status" block

Table 3- 43 Explanation of the "Tag_Status" block

Parameter	Data type	Default values	Description
ATTRIBUTE	Byte	B#16#0	Identifier of the status modes / possible entries: • RF200: 0x83 • RF300: 0x04, 0x82, 0x83 (only ISO transponders) • RF620R, RF630R: 0x84, 0x85 • MOBY D: 0x83 [1)] • MOBY U: 0x80
LEN_ID	Byte	B#16#0	Length of the EPC-ID/UID Default value: 0x00 ≙ unspecified single tag access (RF680R, RF685R)
EPCID_UID	Array[1...62] of Byte	0	Buffer for up to 62 bytes EPC-ID, 8 bytes UID or 4 bytes handle ID. • 2 - 62-byte EPC-ID is entered at the start of the buffer (length is set by "LEN_ID") • 8-byte UID is entered at the start of the buffer ("LEN_ID = 8") • 4-byte handle ID must be entered in the array element [5]-[8] ("LEN_ID = 8") Default value: 0x00 ≙ unspecified single tag access (RF620R, RF630R)
IDENT_DATA	Any / Variant	0	Event values depending on attributes Note: For Variant, an "Array_of_Byte" with a variable length can be created. For Any, other data types/UDTs can also be created.

1) SLG D10S only

Results

Note that the UDTs can only be displayed when the blocks "Reader_Status" or "Tag_Status" are used.

Table 3- 44 ATTRIBUTE "0x04" ("IID_TAG_STATUS_04_RF300" data type)

Name	Type	Comment
reserved	BYTE	
status info	BYTE	MDS status mode
UID	ARRAY [1..8] of BYTE	
MDS type	BYTE	Type of MDS
Lock state	BYTE	Write Protection Status EEPROM
Reserved1	ARRAY[1..6] of BYTE	

Table 3- 45 ATTRIBUTE "0x80" ("IID_TAG_STATUS_80_MOBY_U" data type)

Name	Type	Comment
UID	ARRAY [1..4] of BYTE	Unique indentifier (MDS-Number)
MDS type	BYTE	Type of MDS
sum subframe access 1	BYTE	Sum of subframe access Byte 1
sum subframe access 2	BYTE	Sum of subframe access Byte 2
sum subframe access 3	BYTE	Sum of subframe access Byte 3
sum subframe access 4	BYTE	Sum of subframe access Byte 4
sum searchmode access 1	BYTE	Sum of search mode access Byte 1
sum searchmode access 2	BYTE	Sum of search mode access Byte 2
ST date Week	BYTE	Date of last sleep-time change (week of year)
ST date Year	BYTE	Date of last sleep-time change (year)
battery left 1	BYTE	Battery power left (percent) Byte 1
battery left 2	BYTE	Battery power left (percent) Byte 2
ST	BYTE	Actual sleep-time on MDS

Table 3- 46 ATTRIBUTE "0x82" ("IID_TAG_STATUS_82_RF300" data type)

Name	Type	Comment
reserved	BYTE	
status info	BYTE	MDS status mode
UID	ARRAY [1..8] of BYTE	
LFD	BYTE	Magnetic flux density: correlation between limit-value
FZP	BYTE	Error counter passive: distortion without communication
FZA	BYTE	Error counter active: distortion during communication
ANWZ	BYTE	Presence counter: measure value for presence time
reserved1	ARRAY [1..3] of BYTE	

Table 3- 47 ATTRIBUTE "0x83" ("IID_TAG_STATUS_83_ISO" data type)

Name	Type	Comment
reserved	BYTE	
status info	BYTE	MDS status mode
UID	ARRAY [1..8] of BYTE	
MDS Type	BYTE	Type of MDS
IC version	BYTE	Chip version
size HB	BYTE	Size of Memory (high Byte)
size LB	BYTE	Size of memory (low Byte)
lock state	BYTE	Write protection status EEPROM
block size	BYTE	Size of a block in addressable memory
number of block	BYTE	Number of blocks in addressable memory

Table 3- 48 ATTRIBUTE "0x84" ("IID_TAG_STATUS_84_RF600" data type)

Name	Type	Comment
reserved	BYTE	
status info	BYTE	MDS status mode
UID	ARRAY [1..8] of BYTE	
antenna	BYTE	Antenna which has observed the MDS
RSSI	BYTE	RSSI value
last observed hour	BYTE	Last observed time hour
last observed min	BYTE	Last observed time minute
last observed sec	BYTE	Last observed time seconds
last observed channel	BYTE	Last observed time channel
EPC length	BYTE	EPC-Length
reserved1	BYTE	

1) The internal time stamp of the internal reader clock that relates to this event is output. The internal reader clock is not synchronized with UTC.

Table 3- 49 ATTRIBUTE "0x85" ("IID_TAG_STATUS_85_RF600" data type)

Name	Type	Comment
status info	BYTE	MDS status mode
antenna	BYTE	Antenna which has observed the MDS
channel	BYTE	Channel
UID	ARRAY [1..8] of BYTE	
DT_glimpsed_1	BYTE	Time elasped between acknowledgement and first read in [ms]1 Highbyte
DT_glimpsed_2	BYTE	Time elasped between acknowledgement and first read in [ms]2
DT_glimpsed_3	BYTE	Time elasped between acknowledgement and first read in [ms]3
DT_glimpsed_4	BYTE	Time elasped between acknowledgement and first read in [ms]4 Low-Byte
reserved1	BYTE	
reserved2	BYTE	
reserved3	BYTE	
reserved4	BYTE	
last observed hour	BYTE	Last observed time hour
last_observed_min	BYTE	Last observed time minutes 1)
last_observed_sec	BYTE	Last observed time seconds 1)
last observed EPC length	BYTE	Last observed time EPC length
EPC_ID_Byte	ARRAY [1..62] of BYTE	EPC-ID
reads HB	BYTE	Number of Reads of MDS in Inventory (1 - 65535)
reads LB	BYTE	Number of Reads of MDS in Inventory (1 - 65535)
RSSI	BYTE	Current RSSI value of MDS 2)
mean RSSI	BYTE	Mean RSSI value of MDS
max RSSI	BYTE	Max RSSI value of MDS
min RSSI	BYTE	Min RSSI value of MDS
min POWER	BYTE	Min Power value of MDS

Name	Type	Comment
current_POWER	BYTE	Current Power value of MDS [3]
reserved5	ARRAY[1..137] of BYTE	

[1] The internal time stamp of the reader that relates to this event is output. The internal reader time stamp is not synchronized with UTC.

[2] The value "Reads" indicates the total transponder recognitions (inventories) regardless of the set smoothing parameters. In this way, in extreme situations, the "Reads" counter can reach extremely high values without the transponder ever reaching the "Observed" status.

[3] The "current_Power" value is specified as transmit power in 0.25 dBm steps (ERP/EIRP). A "current_Power" value of "72" (0x48) therefore corresponds to 18 dBm (ERP/EIRP).

You will find more detailed information on the individual status modes in the manuals matching the modes "FB 45", "FB55" and "SIMATIC RF620R/RF630R".

The identifiers of the status modes correspond to the following identifiers in the other manuals:

0x04 ≙ 0x01
0x82 ≙ 0x02
0x83 ≙ 0x03
0x84 ≙ 0x04
0x85 ≙ 0x05

3.5 Programming the Ident profile

3.5.1 Changing to Ident blocks / profile

The Ident blocks or the Ident profile replace "PIB_1200_UID_001KB" and its blocks. Apart from the name changes, functional changes were also made to the block. Note the following points if you want to upgrade an existing project with PIB blocks/"PIB_1200_UID_001KB" from the library version V1.04 to the Ident blocks or the Ident profile from the library V2.0:

- Delete all previous blocks from the program.
- Adapt each point of use to the call for the new instruction.
- Change the data type of the following variables:
  - "HW_CONNECT_VAR" → "IID_HW_CONNECT"
  - "CMD_STRUCT" → "IID_CMD_STRUCT"

Example: Changing without multi-instance

To change from a block without multi-instance to Ident blocks/profile, follow the steps below:

1. Delete all previous blocks ("PIB_1200_UID_001KB", "Read", "Write", etc.) and their instance DBs from the "Program blocks" folder of the project tree.
2. Delete the previous data types "HW_CONNECT_VAR" and "CMD_STRUCT" from the "PLC data types" folder of the project tree.
3. Drag the required Ident block from the library tab to the open block.

   Make sure that you use the name of the old block call again in the new block call (e.g. "Reset_RF300_DB").

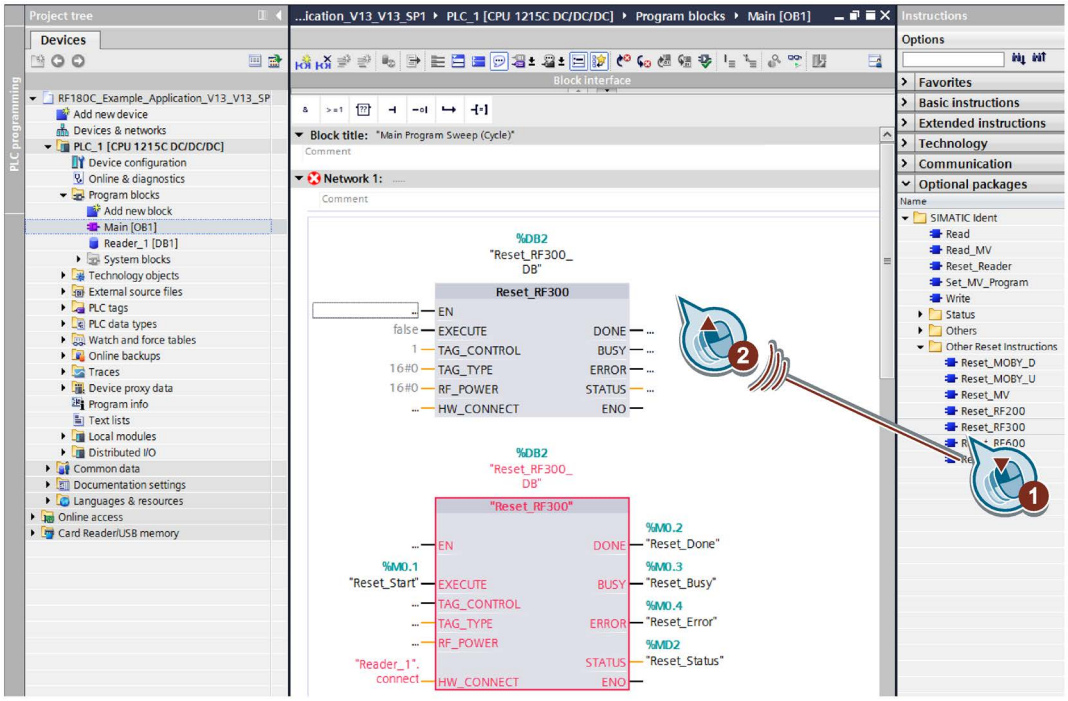


Figure 3-32 Inserting Ident blocks

4. Copy the variables of the old block call into the new block all (e.g. using drag and drop).
5. Delete the old block call.
6. Open the data block in which the "HW_CONNECT_VAR" variable was created.
7. Note down the address parameters of the variables.
8. Change the data type from "HW_CONNECT_VAR" to "IID_HW_CONNECT" and enter the address parameters.

   If you have created a variable of the type "CMD_STRUCT", change the data type to "IID_CMD_STRUCT".
9. Repeat steps 3 to 5 for each created block.
10. Repeat steps 6 to 8 for each channel/reader.

Example: Changing with multi-instance

To change from a block with multi-instance to Ident blocks/profile, follow the steps below:

1. Delete all previous blocks ("PIB_1200_UID_001KB", "Read", "Write", etc.) and their instance DBs from the "Program blocks" folder of the project tree.
2. Delete the previous data types "HW_CONNECT_VAR" and "CMD_STRUCT" from the "PLC data types" folder of the project tree.
3. Open the data block in which you use a block as a multi-instance.
4. Change the data type of the multi-instance:
   - With a PIB block (e.g. "Read")

     Delete the quotes of the data type.

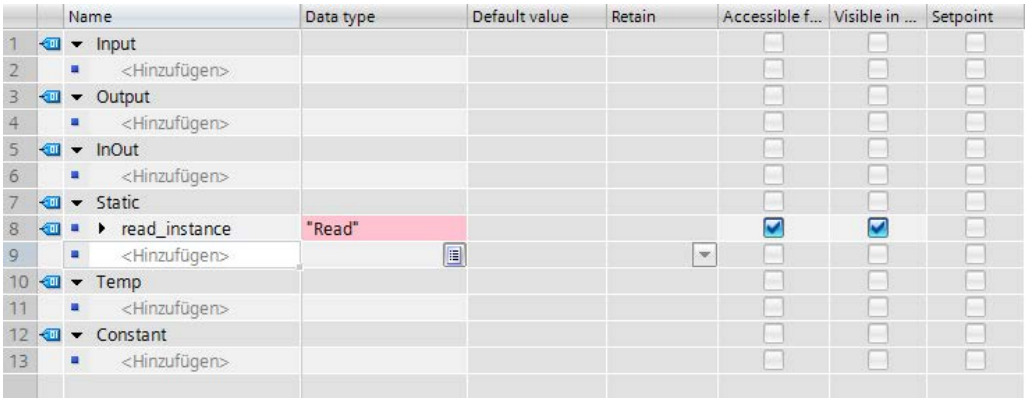


Figure 3-33 Changing a data type

   - With "PIB_1200_UID_001KB"

     Change the data type of "PIB_1200_UID_001KB" to "Ident_Profile".
5. Open the data block in which the "HW_CONNECT_VAR" variable was created.
6. Note down the address parameters of the variables.
7. Change the data type from "HW_CONNECT_VAR" to "IID_HW_CONNECT" and enter the address parameters.

   If you have created a variable of the type "CMD_STRUCT", change the data type to "IID_CMD_STRUCT".
8. Repeat steps 3 to 5 for each created block.
9. Repeat steps 6 to 8 for each channel/reader.

With both variants, the variable table "PIB_CONSTANTS" is omitted. This no longer exists in the library and must also be deleted from the user program.

Note

Check the content of the "PIB_CONSTANTS" variable table before deleting it

Before you delete the "PIB_CONSTANTS" variable table, make sure that you have not defined any of your own variables in this variable table. If you have created your own variables, check whether these need to be moved to a different variable table.

3.5.2 Structure of the Ident profile

Note

Parallel operation using Ident blocks and Ident profile is not possible

Note that the CM or reader cannot be operated at the same time using the Ident blocks and the Ident profile.

The blocks described in the section"Programming Ident blocks (Page 21)" represent a simplified interface of the Ident profile. If the functionality available with the blocks is not adequate for your application, you can use the Ident profile as an alternative. Using the Ident profile, you can set complex command structures and work with command repetition. The following graphic shows the Ident profile including the commands that can be implemented with it.

Note

Ident profile for trained users

The Ident profile is a complex block containing all the functionality of the Ident blocks. The Ident profile was developed specially for trained block users who want to configure complex functions with their own blocks. For untrained users, we recommend using the Ident blocks.

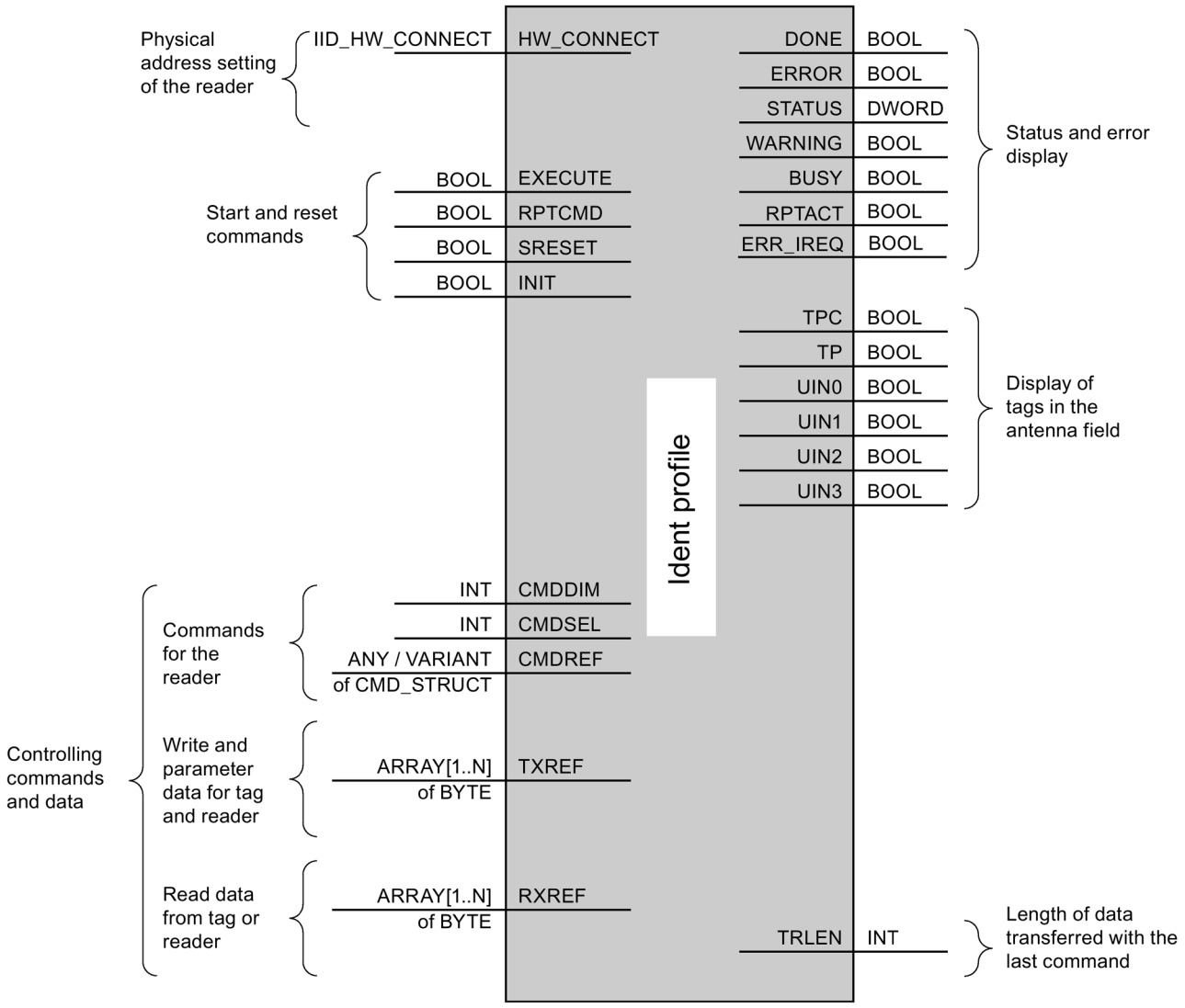


Figure 3-34 The input parameters of the Ident profile

Note

Working with multiple channels

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

Interface description

Table 3- 50 Input parameter

Input parameter	Data type	Default value	Description
HW_CONNECT	HW_CONNECT	--	Own data type for physical addressing of communications modules and readers and for synchronizing the blocks used for each reader. The addressing is as described in the section "Setting the "IID_HW_CONNECT" data type (Page 15)".
EXECUTE	BOOL	FALSE	TRUE = triggers a new command Before starting you need to set the command and the corresponding parameters in the memory linked to "CMDREF".
RPTCMD	BOOL	FALSE	TRUE = Repeating the command currently being executed or the next command to be executed by communications module
SRESET	BOOL	FALSE	TRUE = Cancellation of the command currently processed on the communications module
INIT	BOOL	FALSE	TRUE = Communications module executes a Reset and is re-assigned parameters
CMDDIM	INT	10	Number of commands in the parameter "CMDREF"
CMDSEL	INT	0	Selection of the command to be executed "CMDREF"; 1 ⇒ 1. command, ... The value of the "CMDSEL" parameter can never be higher than the value of the "CMDDIM" parameter.
CMDREF	ARRAY[1...10] of CMD_STRUCT	--	Command field The field can hold up to 10 commands. The commands are complex variables of the type "CMD_STRUCT". You will find more information on "CMDREF" in the section "Commands of the Ident profile (Page 70)".
TXREF	ARRAY[1...n] of BYTE	--	Reference to global memory area for send data. The memory area can be shared with other block instances. The value "n" of the individual blocks is variable and can be up to 32 KB in size.
RXREF	ARRAY[1...n] of BYTE	--	Reference to global memory area for receive data. The memory area can be shared with other block instances. The value "n" of the individual blocks is variable and can be up to 32 KB in size.

Table 3- 51 Output parameter

Output parameter	Data type	Default value	Description
DONE	BOOL	FALSE	TRUE = Command was executed free of errors.
ERROR	BOOL	FALSE	TRUE = Error was detected. The error is output in the "STATUS" parameter. The bit is reset automatically when a new command is started.
STATUS	DWORD	FALSE	Warning and error If "ERROR = TRUE" or "WARNING = TRUE", the error or warning information is contained in the "STATUS" parameter. For more information, refer to the section "Error messages (Page 111)".
WARNING	BOOL	FALSE	TRUE = Warning was detected. The warning is output in the "STATUS" parameter. If the "ERROR" parameter is not set at the same time, the data was correctly processed. The bit is reset automatically when a new command is started.
BUSY	BOOL	FALSE	TRUE = the block is executing a command. Other commands except for "INIT" and "SRESET" cannot be started.
RPTACT	BOOL	FALSE	TRUE = "RPTCMD" is active The acknowledgement bit shows that the "Repeat mode" of the CM/reader is active.
ERR_IREQ	BOOL	FALSE	TRUE = An error has occurred on the communications module or reader (e.g. at power-up or connection termination)
TPC	BOOL	FALSE	Transponder Presence Changed (only when the Presence_Mode is active) TRUE = A new transponder enters the antenna field of the reader or a transponder has left the antenna field. The parameter is set to "FALSE" after the successful execution of the next "INVENTORY" (0x80, 0x81, 0x87) or "INIT" command.
TP	BOOL	FALSE	Transponder Presence TRUE = There is a transponder in the antenna field of the reader.
UIN0	BOOL	FALSE	With RFID readers, the number of transponders in the antenna field is indicated. With optical reader devices, the various statuses of the reader device are displayed. UIN0: Corresponds to IN_OP bit of the reader device UIN1: Corresponds to RDY bit of the reader UIN2 + UIN3: These two bits are interpreted as an unsigned value (bit 2 is the less significant bit) that represents the number of available decoded codes. If the value is = 3, three or more decoded codes are available.
UIN1	BOOL	FALSE	
UIN2	BOOL	FALSE	
UIN3	BOOL	FALSE	
TRLEN	INT	0	Number of data elements received after successful execution of the command.

3.5.3 Data structure of the Ident profile

Each time the Ident profile is called, you need to supply the parameters ("HW_CONNECT", "CMDREF", "TXREF" and "RXREF") with values as described in section "Structure of the Ident profile (Page 64)".

The call for the Ident profile is always via the input parameter "HW_CONNECT" and the "IN/OUT" parameters "CMDREF", "TXREF" and "RXREF". All three parameters need to be created in a data block. The relationship between the three "IN/OUT" parameters is described in greater detail below:

- CMDREF (command buffer):

  Array[1...10] of CMD_STRUCT

- TXREF (send buffer):

  Array[1...n] of Byte

- RXREF (receive buffer):

  Array[1...n] of Byte

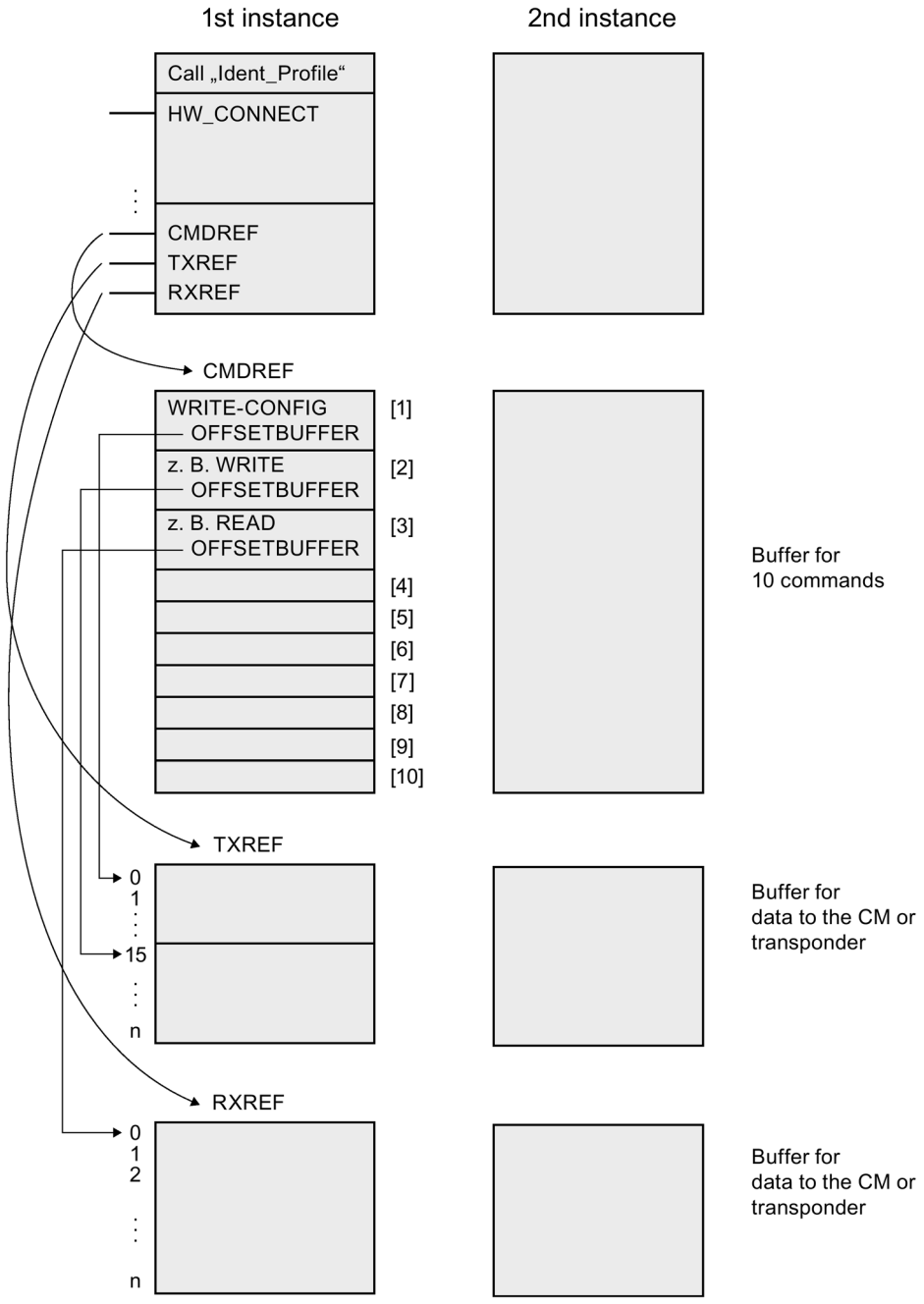


Figure 3-35 Data structure example of the Ident profile

Explanation of the data structure example

- CMDREF[1]:

  Command "WRITE-CONFIG", OFFSETBUFFER = 0

  At the CMDREF[1] point you need to set the "WRITE-CONFIG" command so that the "INIT/Reset" is correctly executed.

- CMDREF[2]:

  Command "WRITE", OFFSETBUFFER = 15

- CMDREF[3]:

  Command "READ", OFFSETBUFFER = 0

If the "CMDREF[2]" command is selected, a write command is started and the data to be written is fetched starting at byte 15 of the "TXREF" parameter. If the "CMDREF[3]" command is selected, the read data is stored starting at byte 0 in the "RXREF" parameter.

3.5.4 Commands of the Ident profile

The following table contains all the commands supported by the Ident profile and the "AdvancedCMD" block.

Table 3- 52 Commands of the Ident profile

Command	Command code		Parameters used	Description
	HEX	ASCII		
PHYSICAL-READ	70	'p'	OFFSETBUFFER, EPCID_UID, LEN_ID, LEN_DATA, ADDR_TAG, MEM_BANK, PSWD	Reads data from a transponder / optical reader system by specifying the physical start address, the length and the password.
PHYSICAL-WRITE	71	'q'	OFFSETBUFFER, EPCID_UID, LEN_ID, LEN_DATA, ADDR_TAG, MEM_BANK, PSWD	Writes data to a transponder / optical reader system by specifying the physical start address, the length and the password.
READER-STATUS	74	't'	OFFSETBUFFER, ATTRIBUTES	Reads out the status of the reader.
TAG-STATUS	73	's'	OFFSETBUFFER, EPCID_UID, LEN_ID, ATTRIBUTES	Reads out the status of a transponder.
INVENTORY	69	'i'	OFFSETBUFFER, ATTRIBUTES, DURATION, DUR_UNIT	Requests a list of all currently accessible transponders within the antenna range.
FORMAT	66	'f'	OFFSETBUFFER, EPCID_UID, LEN_ID, LEN_DATA	Initializes the transponder.
PUT	65	'e'	OFFSETBUFFER, EPCID_UID, LEN_ID, LEN_DATA	Transfers further commands not specified in the standard profile. To this end, a corresponding data structure is defined in the send data buffer for each command.

Command	Command code		Parameters used	Description
	HEX	ASCII		
WRITE-ID	67	‘g‘	OFFSETBUFFER, EPCID_UID, LEN_ID, NEW-LEN_ID, PSWD	RF680R/RF685R: Writes a new EPC-ID to the transponder.
KILL-TAG	6A	‘j‘	EPCID_UID, LEN_ID, PSWD	RF680R/RF685R: The transponder is permanently deactivated.
LOCK-TAG-BANK	79	‘y‘	EPCID_UID, LEN_ID, PSWD, ACTION, MASK	RF680R/RF685R: The corresponding memory area of the transponder is blocked as specified.
EDIT-BLACKLIST	7A	‘z‘	EPCID_UID, LEN_ID, MODE	RF680R/RF685R: The black list is processed. The current transponder can be added, all identified transponders added, individual transponders deleted or all transponders deleted.
GET-BLACKLIST	6C	‘l‘	OFFSETBUFFER	RF680R/RF685R: The entire TagIDs are read out from the black list.
READ-CONFIG	61	'a'	--	Reads out the parameters from the communications module/reader.
WRITE-CONFIG	78	'x'	LEN_DATA, CONFIG	Sends new parameters to the communications module/reader.

3.5.4.1 Command structure

Before you can start a command with "EXECUTE" or "INIT", you need to define the command. To allow simple definition of a command, the command buffer "CMDREF" was created using the "IID_CMD_STRUCT" data type. In the command buffer, you have 10 areas available in which commands can be set. The parameter "CMDSEL" specifies which command [1...10] is started with "EXECUTE".

Remember that the first element in the buffer is always reserved for "INIT". In other words if "INIT" is set, "CMDSEL" must be set to "1" and element "1" in the CMD buffer must be filled with the relevant settings. The following table contains the command structure of the parameters. Not every command uses all parameters.

Table 3- 53 Command structure of the parameters

Parameter	Data type	Default value	Description
CMD	BYTE	B#16#0	Command code (compare the table in the section "Commands of the Ident profile (Page 70)".)
OFFSETBUFFER	INT	0	Relative offset within the received data buffer. The parameter specifies the address within the memory area at which the first byte of the received data must be stored or the first byte of the data to be sent is expected. All subsequent bytes must be stored in ascending addresses.
EPCID_UID	ARRAY[1...62] OF BYTE	B#16#0	Buffer for up to 62 bytes EPC-ID, 8 bytes UID or 4 bytes handle ID. • 2 - 62-byte EPC-ID is entered at the start of the buffer (length is set by "LEN_ID") • 8-byte UID is entered at the start of the buffer ("LEN_ID = 8") • 4-byte handle ID must be entered in the array element [5]-[8] ("LEN_ID = 8") Default value: 0x00 ≙ unspecified single tag access
LEN_DATA	WORD	W#16#0	Amount of data to be read/written in bytes
ADDR_TAG	DWORD	DW#16#0	Physical start address on the transponder
ATTRIBUTES	BYTE	B#16#0	Sub command name for several commands (e.g. "READER-STATUS", "INVENTORY", etc,)
CHAINED	BOOL	FALSE	• 0x00 = not chained • 0x01 = chained All chained commands must have this bit set except the last command. The commands are worked through in the order in which they are located in the CMD structure.
CONFIG	BYTE	B#16#0	• 0x01 = reset, no configuration data • 0x02 = no reset, configuration data to be sent • 0x03 = reset, configuration data to be sent • 0x80 = no reset, only individual parameters

Parameter		Data type	Default value	Description
EXT_UHF		STRUCT	--	Structure for additional parameters (RF680R/RF685R only)
	LEN_ID	BYTE	B#16#0	Length of the valid data in the "EPCID_UID" field.
	MEM_BANK	BYTE	B#16#3	Memory bank on the transponder • 0x00 = RESERVED • 0x01 = EPC • 0x02 = TID • 0x03 = USER
	PSWD	DWORD	DW#16#0	Password for transponder access 0x00 ≙ no password
	EDIT_BLACKLIST_MODE	BYTE	B#16#0	Mode • 0x00 = add TagID • 0x01 = add all "Observed" transponders • 0x02 = delete TagID • 0x03 = delete all
	INVENTORY_DURATION	WORD	W#16#0	Duration Period of time or number of inventories or number of "Observed" events Example: • 0x00 ≙ no inventory • 0x01 ≙ one inventory
	INVENTORY_DUR_UNIT	WORD	W#16#0	Unit for "DURATION" • 0x00 = time [ms] • 0x01 = inventories • 0x02 = number of "Observed" events
	LOCK-TAG-BANK_ACTION	WORD	W#16#0	Lock-Action (see "EPC Specification")
	LOCK-TAG-BANK_MASK	WORD	W#16#0	Lock-Mask (see "EPC Specification")

3.5.4.2 Commands

Table 3- 54 PHYSICAL-READ

CMD	OFFSET BUFFER	LEN_ DATA	ADDR_ TAG	CHAINED	EPCID_ UID	LEN_ID	MEM_ BANK	PSWD	RXREF
0x70	Offset in the "RXREF" receive buffer	Length of re-ceived data	Address on the tran-sponder	True = chained False = not chained	Buffer for up to 62 bytes EPC-ID, 8 bytes UID or 4 bytes handle ID. • 2 - 62-byte EPC-ID is entered at the start of the buffer (length is set by "LEN_ID") • 8-byte UID is entered at the start of the buffer ("LEN_ID = 8") • 4-byte handle ID must be entered in the array element [5]-[8] ("LEN_ID = 8") Default value: 0x00 ≙ unspecified single tag access	Length of the EPC-ID (2-62 bytes) 0x00 ≙ unspeci-fied single tag ac-cess	Memory bank • 0x00 ≙ reserved • 0x01 ≙ EPC • 0x02 ≙ TID • 0x03 ≙ USER	Pass-word 0x00 ≙ no pass-word	Read data

Table 3- 55 PHYSICAL-WRITE

CMD	OFFSET BUFFER	LEN_ DATA	ADDR_ TAG	CHAINED	EPCID_ UID	LEN_ID	MEM_ BANK	PSWD	TXREF
0x71	Offset in the "TXREF" send buffer	Length of the data to be written	Address on the tran-sponder	True = chained False = not chained	Buffer for up to 62 bytes EPC-ID, 8 bytes UID or 4 bytes handle ID. • 2 - 62-byte EPC-ID is entered at the start of the buffer (length is set by "LEN_ID") • 8-byte UID is entered at the start of the buffer ("LEN_ID = 8") • 4-byte handle ID must be entered in the ar-ray element [5]-[8] ("LEN_ID = 8") Default value: 0x00 ≙ unspecified single tag access	Length of the EPC-ID (2-62 bytes) 0x00 ≙ unspeci-fied single tag ac-cess	Memory bank • 0x00 ≙ reserved • 0x01 ≙ EPC • 0x02 ≙ TID • 0x03 ≙ USER	Pass-word 0x00 ≙ no pass-word	Data to be written

Table 3- 56 READER-STATUS

CMD	OFFSETBUFFER	ATTRIBUTES	RXREF
0x74	Offset in the "RXREF" receive buffer	Identifier of the status modes / possible entries: • RF200: 0x81 • RF300: 0x81, 0x86 • RF620R, RF630R: 0x87, 0x88, 0xA0, 0xA1 • RF680R, RF685R: 0x89 • MOBY U: 0x81, 0x84, 0x85 • MOBY D: 0x81	Received status data You will find the data structure of the status modes in the section "Reader_Status (Page 52)".

Table 3- 57 TAG-STATUS

CMD	OFFSETBUFFER	ATTRIBUTES	EPCID_ UID	LEN_ID	RXREF
0x73	Offset in the "RXREF" receive buffer	Identifier of the status modes / possible entries: • RF200: 0x83 • RF300: 0x04, 0x82, 0x83 (only ISO transponders) • RF600, R680R, RF685R: 0x84, 0x85 • MOBY D: 0x83 [1)] • MOBY U: 80	Buffer for up to 62 bytes EPC-ID, 8 bytes UID or 4 bytes handle ID. • 2 - 62-byte EPC-ID is entered at the start of the buffer (length is set by "LEN_ID") • 8-byte UID is entered at the start of the buffer ("LEN_ID = 8") • 4-byte handle ID must be entered in the array element [5]-[8] ("LEN_ID = 8") Default value: 0x00 ≙ unspecified single tag access	Length of the EPC-ID/UID	Received status data You will find the data structure of the status modes in the section "Tag_Status (Page 57)".

[1)] SLG D10S only

Table 3- 58 INVENTORY

CMD	OFFSET BUFFER	ATTRIBUTES	INVENTORY_ DURATION	INVENTORY_ DUR_UNIT	RXREF
0x69	Offset in the "RXREF" receive buffer	Identifier of the status modes / possible entries: RF680R/RF685R: • 0x80 ≙ inventory with brief transponder information • 0x81 ≙ inventory with a lot of transponder information • 0x86 ≙ Presence mode on • 0x87 ≙ Presence mode off RF620R/RF630R: • 0x82 ≙ read out the next data record • 0x83 ≙ read handle ID when MOBY_mode ≙ 6 and EPC-ID when MOBY_mode ≙ 7 • 0x85 ≙ read out handle IDs and EPC-IDs sorted in descending order according to the mean RSSI value • 0x91 ≙ read out handle IDs sorted in descending order according to the maximum RSSI value • 0x92 ≙ read out handle IDs sorted in descending order according to read frequency • 0xA0 ≙ read out first entries from Black List • 0xA1 ≙ read out further entries from Black List RF300/MOBY U: • 0x00 ≙ list of all tags with UID	Only for 0x80 and 0x81: Duration Period of time or number of inventories or number of "Observed" events Example: • 0x00 ≙ no inventory • 0x01 ≙ one inventory	Only for 0x80 and 0x81: Unit for "DURATION" • 0x00 ≙ time [ms] • 0x01 ≙ inventories • 0x02 ≙ number of "Observed" events	With RF680R/RF685R only when 0x80 and 0x81: Data received With RF620R/RF630R/RF300/MOBY U: Data received You will find the data structure of the status modes in the section "Inventory (Page 28)".

Table 3- 59 FORMAT (not with RF680R/RF685R)

CMD	OFFSETBUFFER	LEN_DATA	EPCID_UID	LEN_ID	TXREF
0x66	Offset in the "TXREF" send buffer	Length of the parameter data to be sent	Buffer for up to 62 bytes EPC-ID, 8 bytes UID or 4 bytes handle ID. • 2 - 62-byte EPC-ID is entered at the start of the buffer (length is set by "LEN_ID") • 8-byte UID is entered at the start of the buffer ("LEN_ID = 8") • 4-byte handle ID must be entered in the array element [5]-[8] ("LEN_ID = 8") Default value: 0x00 ≙ unspecified single tag access	Length of the EPC-ID/UID	Parameter data to be written

Table 3- 60 Structure of the data attachment for the "FORMAT" command with normal addressing

Byte	1...8	9	10	11	12	13	14	15
Value	00h	06h	03h	00h	INIT-Wert	00h	MSB	LSB

Table 3- 61 Explanation of the structure of the data attachment for the "FORMAT" command

Byte	Description
Bytes 1...8	Reserved for security code (must be assigned "0", since SIMATIC RFID has had no code previously)
Byte 9	Length of the following data, here 6
Byte 10	Permanently set to "0x03"
Byte 11	Permanently set to "0x00"
Byte 12	"INIT" value: The data area of the transponder is written with this value (hex format).
Byte 13	Permanently set to "00h"
Byte 14	Memory size of the transponder (end address + 1; high byte, hex format)
Byte 15	Memory size of the transponder (end address + 1; low byte, hex format)

Table 3- 62 Memory sizes of the transponders

Transponder type			Memory size	INIT duration
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	I-Code 1	00 2C	approx. 0.4 s
112 bytes	MOBY D	ISO I-Code SLI	00 70	approx. 0.5 s
256 bytes	MOBY D	ISO Tag-it HF-I	01 00	approx. 1 s
992 bytes	MOBY D	ISO my-d	03 E0	approx. 3 s
2000 bytes	MOBY D	FRAM	07 D0	approx. 3 s
20 bytes	RF300	EEPROM	00 14	approx. 0.2 s
8 KB	RF300	FRAM *)	20 00	0.9 s
32 KB	RF300	FRAM *)	80 00	3.6 s
64 KB	RF300	FRAM *)	FF 00	7.2 s

*) The OTP memory is not initialized with this command.

Table 3- 63 PUT (not with RF680R/RF685R)

CMD	OFFSETBUFFER	LEN_DATA	TXREF
0x65	Offset in the "TXREF" send buffer	Length of the parameter data to be sent	Parameter data to be written

Table 3- 64 Data structure of the PUT command

Put_SET_ANT		Switches the antenna of the reader off and on. 1: 'N' | 2: 'A' | 3: Mode
	Mode	RF200/RF300, MOBY U/D: • 0x01 ≙ antenna on • 0x02 ≙ antenna off RF600: • Bit 0 ≙ ANT 1 / internal antenna (1 = on) • Bit 1 ≙ ANT 2 / external antenna (1 = on) • Bit 4 ≙ TagList (0 = initialize, 1 = continue working with the existing list)
	Length	3
Put_END		Terminates communication with a transponder (MOBY U only). 1: 'N' | 2: 'K' | 3 ... 10: UID | 11: Mode
	UID	UID of the transponder
	Mode	• 0x00 ≙ end processing of the transponder • 0x01 ≙ processing pause of the transponder
	Length	11

Table 3- 65 WRITE-ID (RF620R/RF630R, RF680R/RF685R only)

CMD	OFFSET BUFFER	EPCID_ UID	LEN_ID	LEN_DATA	PSWD	TXREF
0x67	Offset in the "TXREF" send buffer	Previous EPC ID 0x00 ≙ unspecified single tag access	Length of the previous EPC-ID (2-62 bytes) 0x00 ≙ unspecified single tag access	Length of the new EPC-ID	Password 0x00 ≙ no password	New EPC-ID

Table 3- 66 KILL-TAG (RF680R/RF685R only)

CMD	EPCID_ UID	LEN_ID	PSWD
0x6A	EPC ID 0x00 ≙ unspecified single tag access	Length of the EPC-ID (2-62 bytes) 0x00 ≙ unspecified single tag access	Password must be ≠ 0x00

Table 3- 67 LOCK-TAG-BANK (RF680R/RF685R only)

CMD	EPCID_ UID	LEN_ID	PSWD	LOCK_TAG_ BANK_ACTION	LOCK_TAG_ BANK_MASK
0x79	EPC ID 0x00 ≙ unspecified single tag access	Length of the EPC-ID (2-62 bytes) 0x00 ≙ unspecified single tag access	Password 0x00 ≙ no password	See EPC standard	See EPC standard

Table 3- 68 EDIT-BLACKLIST (RF680R/RF685R only)

CMD	EDIT_ BLACKLIST_MODE	EPCID_ UID	LEN_ID
0x7A	• 0x00 ≙ add EPC-ID • 0x01 ≙ add all "OBSERVED" transponders • 0x02 ≙ delete EPC-ID • 0x03 ≙ delete all	EPC ID 0x00 ≙ unspecified single tag access [1]	Length of the EPC-ID (2-62 bytes) 0x00 ≙ unspecified single tag access

[1] If "EDIT_BLACKLIST_MODE" = 0x00 or 0x02 was selected, the EPC-ID including the ID length must be specified.

Table 3- 69 GET-BLACKLIST (RF680R/RF685R only)

CMD	OFFSETBUFFER	RXREF
0x6C	Offset in the "RXREF" receive buffer	Read black list IDs

Table 3- 70 READ-CONFIG

CMD	OFFSETBUFFER	RXREF
0x61	Offset in the "RXREF" receive buffer	Read reset parameters

Table 3- 71 WRITE-CONFIG

CMD	OFFSET BUFFER	LEN_DATA	CONFIG	TXREF
0x78	Offset in the "TXREF" send buffer	Length of the parameter data	• 0x01 ≙ communication reset, no configuration data • 0x02 ≙ no communication reset, configuration data to be sent • 0x03 ≙ communication reset, configuration data to be sent • 0x80 ≙ no communication reset, individual parameters	Configuration data to be sent

Structure of the configuration data attachment of WRITE-CONFIG

For RF680R/RF685R

- When CONFIG = 0x01:
  Reset_Reader; LEN_DATA = 0x00
- When CONFIG = 0x03:

  When replacing a module, it is possible to read all the configuration data from the reader and to store it on the controller. When the module is replaced, this data can then be loaded on the reader from the controller. The command "WRITE-CONFIG" (0x03) is used for the download to the reader and "READ-CONFIG" for the upload from the reader.

Byte	Name
0	Structure identifier (2 bytes)
2	Length information (4 bytes) Length of the version identifier and parameter block
6	Version identifier (4 bytes) Based on the identifier, you can uniquely identify the configuration. This is a time stamp in Linux format. The time stamp indicates how many seconds have passed since January 1, 1979, 00:00 (midnight). The identifier is assigned when a configuration is generated.
10 ... end "DATA"	Parameter block

  LEN_DATA = size of the configuration data + 6 bytes
- When CONFIG = 0x80:
  The structure of the send buffer corresponds to that of the block "Set_Param (Page 41)".
  LEN_DATA = 0x08

For RF200, RF300, RF620R, RF630R, MOBY D/U

Table 3- 72 For RF200, RF300, RF620R, RF630R, MOBY D/U when CONFIG = 0x03

Byte	1	2...5 [1]	6 [2]	7...8	9	10	11	12	13...14	15	16
Value	0x04	0x00	0x0A 0x05	0x00	scanning_ time	param	op- tion_1	distance_ limiting	Number of tran- sponders	field_on_ control	field_on_ time

[1] In the communications module RF180C as of V2.2 in conjunction with MOBY U byte 4 is preset with the calendar week and 5 with the year.

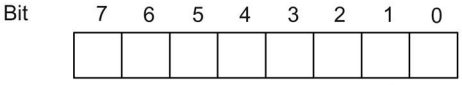
[2] With the readers named in the title of the table the value "0x0A" (LEN_DATA = 0x10) is used in byte 6. In the MOBY I migration in RF300 readers of the second generation the value "0x05" (LEN_DATA = 0x0B) is used.

Table 3- 73 Bytes of the "PARAM" parameter

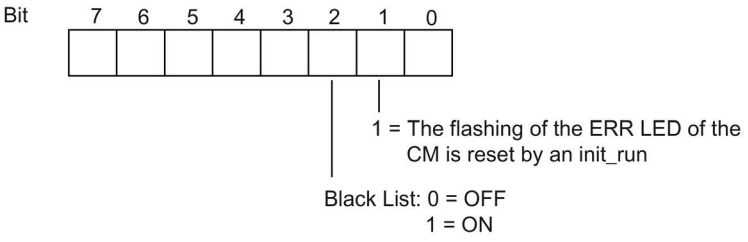
<table>
<tr><th>Byte</th><th>Value</th><th>RFID system</th><th colspan="5">Description</th></tr>
<tr><td rowspan="5">Byte 9</td><td rowspan="5">scanning_
time</td><td>MOBY U</td><td colspan="5">"scanning_time" describes the standby time for the transponder. If the transponder receives a further command before "scanning_time" has expired, this command can be executed immediately. If the transponder receives a command after "scanning_time" has expired, command execution is delayed by the "sleep_time" of the transponder.
"scanning_time" should only be set when
• the transponder is processed with several commands and
• the execution must be completed within a minimum time.

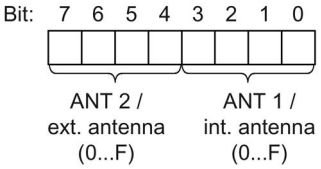
0x00 = no standby time (default)
0x01 = 7 ms standby time
0x02 = 14 ms standby time
:
C8 hex = 1400 ms standby time
Note that the "scanning_time" affects the working life of the battery. The longer "scanning_time" is, the shorter the life of the battery. For precise calculations, see the MOBY U manual for configuration, mounting and service.</td></tr>
<tr><td>RF200,
RF300,
MOBY D</td><td colspan="5">0x00 (reserved)</td></tr>
<tr><td rowspan="8">RF600</td><td colspan="5">"scanning_time" describes the radio profile according to EPC Global. Set the correct standard according to the country in which you want to operate the reader. Please check which standard and which reader type is applicable to your country before you select a wireless profile.</td></tr>
<tr><td colspan="2"></td><td colspan="3">RF600 reader variant</td></tr>
<tr><td>Value</td><td>Description</td><td>ETSI</td><td>FCC</td><td>CMIIT</td></tr>
<tr><td>0</td><td>No standard selected; the error "0x15" is output</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>1</td><td>Reader works with the default wireless profile. Value of the default wireless profile:</td><td>ETSI new</td><td>FCC</td><td>China</td></tr>
<tr><td>2</td><td>ETSI new: EU, EFTA, Turkey; 4-channel plan</td><td>X</td><td>--</td><td>--</td></tr>
<tr><td>3</td><td>ETSI old: EU, EFTA, Turkey; readers commissioned after December 31, 2009, must not be operated with this setting.</td><td>X</td><td>--</td><td>--</td></tr>
</table>

<table>
<tr><th>Byte</th><th>Value</th><th>RFID system</th><th colspan="5">Description</th></tr>
<tr><td></td><td></td><td></td><td>4</td><td>FCC: e.g. USA, Canada, Mexico</td><td>--</td><td>X</td><td>--</td></tr>
<tr><td></td><td></td><td></td><td>5</td><td>reserved</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td></td><td></td><td></td><td>6</td><td>China</td><td>--</td><td>--</td><td></td></tr>
<tr><td></td><td></td><td></td><td>7</td><td>Thailand</td><td>--</td><td>X</td><td>--</td></tr>
<tr><td></td><td></td><td></td><td>8</td><td>Brazil</td><td>--</td><td>X</td><td>--</td></tr>
<tr><td></td><td></td><td></td><td>9</td><td>South Korea</td><td>--</td><td>X</td><td>--</td></tr>
<tr><td></td><td></td><td></td><td>C0</td><td>India</td><td>X</td><td>--</td><td>--</td></tr>
<tr><td></td><td></td><td></td><td colspan="5">Note: If you select country profiles other than those defined for the particular reader variant, the error message "09" is acknowledged.</td></tr>
<tr><td>Byte 10</td><td>param</td><td>RF200, RF300, MOBY D, MOBY U</td><td colspan="5">Setting for the RFID mode and presence check</td></tr>
<tr><td></td><td></td><td></td><td>Bit</td><td colspan="2">7 ... 5</td><td>4</td><td>3 ... 0</td></tr>
<tr><td></td><td></td><td></td><td></td><td colspan="2">Presence check and transponder control:
• 0 = Presence check is off. The parameter "ANZ_MDS_present" does not indicate a valid value.
• 1 = Presence check is on. The transponder control is off. The parameter "ANZ_MDS_present" indicates a transponder in the transmission window of a reader.
• 4 = Presence check is off. The transponder control is on as long as a command is active.</td><td>ECC_mode
To be assigned the value "0".</td><td>RFID mode:</td></tr>
<tr><td></td><td></td><td></td><td>Value of bit 3 ... 0</td><td>Operating mode</td><td colspan="3">Note</td></tr>
<tr><td></td><td></td><td></td><td>0</td><td>reserved</td><td colspan="3">reserved for the setting with the switch or GSD parameter assignment</td></tr>
<tr><td></td><td></td><td></td><td>1</td><td>MOBY I</td><td colspan="3">If RF300 readers of generation 2 are to be operated in MOBY I mode, the value = 1 must be set.
Short "INIT" (only the "param" and "option_1" parameters are transferred to the reader).</td></tr>
<tr><td></td><td></td><td></td><td>5</td><td>MOBY U/D, RF200, RF300 - without multitag handling</td><td colspan="3">ASM 475, ASM 456, RF170C, RF180C</td></tr>
</table>

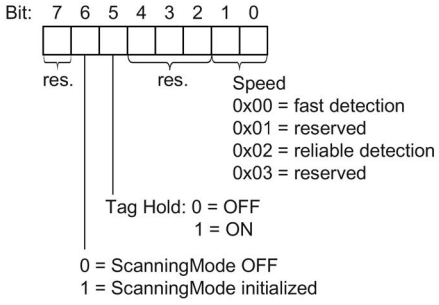
<table>
<tr><th>Byte</th><th>Value</th><th>RFID system</th><th colspan="3">Description</th></tr>
<tr><td rowspan="11"></td><td rowspan="11"></td><td rowspan="3"></td><td>6</td><td>MOBY U - with multitag handling (FB 55)</td><td>• Parameter setting with Multitag > 1 and more than one transponder in the antenna field: the UID parameter must be supplied with the transponder ID.
• Parameter setting with Multitag = 1 and only one transponder in the antenna field: the UID parameter can be supplied with the correct transponder ID or zero.
ASM 475, ASM 456, RF170C, RF180C</td></tr>
<tr><td>7</td><td>MOBY D, RF300 - with multitag handling (FB 55)</td><td>ASM 475, ASM 456, RF170C, RF180C</td></tr>
<tr><td colspan="3">Note: Note that after a parameter change the CM must be restarted.</td></tr>
<tr><td rowspan="6">RF600</td><td colspan="3">RFID mode setting</td></tr>
<tr><td>Value</td><td colspan="2">Operating mode</td></tr>
<tr><td>4</td><td colspan="2">ISTM mode</td></tr>
<tr><td>5</td><td colspan="2">Single tag mode</td></tr>
<tr><td>6</td><td colspan="2">• with single tag handling (UID = 0x00), 4 bytes UID of the 8 byte handle ID
• with multitag handling, 4 bytes UID as handle ID for access to transponders with an EPC-ID of any length</td></tr>
<tr><td>7</td><td colspan="2">• with single tag handling (UID = 0x00), 8 bytes UID
• with multitag handling, 8 bytes UID of bytes 5-12 of the 12-byte long EPC-ID</td></tr>
<tr><td colspan="3">Note: Note that after a parameter change the CM must be restarted.</td></tr>
<tr><td>Byte 11</td><td>option_1</td><td>RF200, RF300, MOBY D, MOBY U</td><td colspan="3">This byte is bit-coded. As default, it has the value "B#16#0". With this byte, special controllers can be implemented on the CM/reader.
 

1 = The flashing of the ERR LED is reset by an init_run.
With RF200/RF300 this resets the flashing of the ERR-LED on the communications module and on the reader.</td></tr>
</table>

<table>
<tr><th>Byte</th><th>Value</th><th>RFID system</th><th colspan="3">Description</th></tr>
<tr><td></td><td></td><td>RF600</td><td colspan="3">This byte is bit-coded. As default, it has the value "B#16#0".
 
</td></tr>
<tr><td rowspan="6">Byte 12</td><td rowspan="6">distance_ limiting</td><td rowspan="3">MOBY U</td><td colspan="3">Range limitation</td></tr>
<tr><td>Normal output power[1)]</td><td colspan="2">Reduced output power</td></tr>
<tr><td>0x05 = 0.5 m
0x0A = 1.0 m
0x0F = 1.5 m
0x14 = 2.0 m
0x19 = 2.5 m
0x1E = 3.0 m
0x23 = 3.5 m</td><td>0x85
0x8A
0x8F
0x91
0x99
0x9E
0xA3</td><td>Set reduced transmit power when several readers are positioned close together or when transponders which are located in the vicinity of a reader are detected later or no longer.
Disadvantage: The field lobe becomes smaller and there is less time for communication or positioning must be more precise.</td></tr>
<tr><td colspan="3">[1)] Intermediate values in steps of 0.1 m are possible (0x02, 0x03, ..., 0x23)</td></tr>
<tr><td>MOBY D</td><td colspan="3">Transmit power from 0.5 W to 10 W in increments of 0.25 W
Only effective with SLG D10S; a power of 1 W (04 hex) is set for SLG D11S / D12S and cannot be changed.
0x02 = 0.5 W
:
0x10 = 4 W (default)
:
0x28 = 10 W</td></tr>
<tr><td>RF200</td><td colspan="3">0x00 (reserved)</td></tr>
<tr><td rowspan="4">RF300 (only RF380R)</td><td colspan="3">With this parameter you can change the transmit power of the RF380R reader. When doing this, remember that the change to the transmit power will affect both the upper and lower limit range, as well as the minimum distance that is to be maintained between adjacent RF380Rs. You will find more information on this in the "System manual RF300".
The following settings are possible:</td></tr>
<tr><td>Bit</td><td colspan="2">Transmit power</td></tr>
<tr><td>02</td><td colspan="2">0.5 W</td></tr>
<tr><td>03</td><td colspan="2">0.75 W</td></tr>
</table>

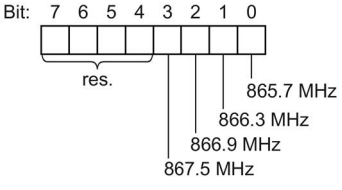
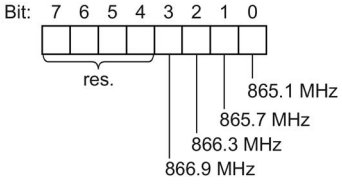
<table>
<tr><th>Byte</th><th>Value</th><th>RFID system</th><th colspan="6">Description</th></tr>
<tr><td></td><td></td><td></td><td colspan="2">04</td><td colspan="4">1.0 W</td></tr>
<tr><td></td><td></td><td></td><td colspan="2">05</td><td colspan="4">1.25 W (default)</td></tr>
<tr><td></td><td></td><td></td><td colspan="2">06</td><td colspan="4">1.5 W</td></tr>
<tr><td></td><td></td><td></td><td colspan="2">07</td><td colspan="4">1.75 W</td></tr>
<tr><td></td><td></td><td></td><td colspan="2">08</td><td colspan="4">2.0 W</td></tr>
<tr><td></td><td></td><td></td><td colspan="6">If settings are made outside the specified values, the default value = 1.25 W is set automatically. For reasons of compatibility, no error message is output.
With the RF380R reader of generation 2 (MLFB 6GT2801-3BAx0) this setting is no longer needed. For reasons of compatibility these values can nevertheless remain set.</td></tr>
<tr><td></td><td></td><td>RF600</td><td colspan="6">The transmit power of the reader is set with "distance_limiting".
 

By default, ANT 1 is used with the preset transmit power.</td></tr>
<tr><td></td><td></td><td></td><td>Hex value</td><td>RF630R transmit power</td><td colspan="3">RF620R radiated power (internal antenna)</td><td>RF620R transmit power</td></tr>
<tr><td></td><td></td><td></td><td></td><td>dBm / (mW)</td><td>ETSI
dBm / (mW) ERP</td><td>FCC
dBm / (mW) EIRP</td><td>CMIIT
dBm / (mW) ERP</td><td>dBm / (mW)</td></tr>
<tr><td></td><td></td><td></td><td>0</td><td>18 / (65)</td><td>18 / (65)</td><td>20 / (105)</td><td>18 / (65)</td><td>18 / (65)</td></tr>
<tr><td></td><td></td><td></td><td>1</td><td>19 / (80)</td><td>19 / (80)</td><td>21 / (130)</td><td>19 / (80)</td><td>19 / (80)</td></tr>
<tr><td></td><td></td><td></td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr>
<tr><td></td><td></td><td></td><td>9</td><td>27 / (500)</td><td>27 / (500)</td><td>29 / (795)</td><td>27 / (500)</td><td>27 / (500)</td></tr>
<tr><td></td><td></td><td></td><td>O</td><td>27 / (500)</td><td>28 / (630)</td><td>30 / (1000)</td><td>28 / (630)</td><td>27 / (500)</td></tr>
<tr><td></td><td></td><td></td><td>B (...F)</td><td>27 / (500)</td><td>29 / (800)</td><td>31 / (1260)</td><td>29 / (800)</td><td>27 / (500)</td></tr>
<tr><td>Bytes 13...14</td><td>Anzahl der Tran-sponder</td><td>RF300</td><td colspan="6">The reader is basically intended for multitag operation, which is, however, not currently released.
Maximum number of transponders that can be processed at the same time in the antenna field. Currently permitted values "1".</td></tr>
<tr><td></td><td></td><td>RF600</td><td colspan="6">Number of transponders expected in the antenna field.
Permitted values:
• 0x01 ... 0x28 for RF620R
• 0x01 ... 0x50 for RF630R with 2 antennas (SET-ANT = 0x03)
• 0x01 ... 0x28 for RF630R with 1 antenna (SET-ANT = 0x01 or SET-ANT = 0x02).
The value specified here defines the maximum expected number of transponders to be read (EPC-ID) in the inventory.
The value does not restrict the number of transponders to be processed in the antenna field. To allow an efficient inventory of transponders in the antenna field, the values given here should not deviate from the maximum number of transponders expected in the antenna field by more than approx. 10%.</td></tr>
</table>

<table>
<tr><th>Byte</th><th>Value</th><th>RFID system</th><th colspan="2">Description</th></tr>
<tr><td rowspan="7">Byte 15</td><td rowspan="7">field_on_
control</td><td rowspan="4">MOBY U</td><td colspan="2">BERO mode; automatic activation/deactivation of the antenna field. The "Antenna ON/OFF" command is superimposed by the BERO mode.</td></tr>
<tr><td>0x00</td><td>without BEROs; no reader synchronization</td></tr>
<tr><td>0x01</td><td>One or two BEROs
The BEROs are ORed. The antenna field is turned on during the actuation of a BERO.</td></tr>
<tr><td>0x02</td><td>One or two BEROs
The 1st BERO switches the antenna field on and the 2nd BERO switches the antenna field off. If there are two BEROs and a " field_ON_time" is set, the antenna field is automatically turned off if the 2nd BERO does not switch within this BERO time.
If no "field_ON_time" is set, the antenna field remains turned on until the 2nd BERO is turned on.</td></tr>
<tr><td>0x03</td><td>Activating reader synchronization over cable connection
You will find further information in the MOBY U manual for configuration, mounting and service.</td></tr>
<tr><td>RF200,
RF300,
MOBY D</td><td colspan="2">0x00 (reserved)</td></tr>
<tr><td rowspan="2">RF600</td><td colspan="2">"field_ON_control" sets the communications speed (fast/slow) and Tag Hold (ON/OFF).
 

Reader parameter assignments that have been optimized depending on the application are available with Speed:
• 0x00 = fast detection
• 0x02 = slower, more reliable detection</td></tr>
<tr><td colspan="2">ScanningMode (relevant for multitag mode):
• Bit 6 = 0: Normal multitag mode (including "repeat_command")
• Bit 6 = 1: Unspecified read commands (UID = 0x00) are also accepted by the CM/reader if there is more than one transponder in the antenna field.
By setting bit 6 to 1, the reader in multitag mode is prepared for the use of "ScanningMode".</td></tr>
</table>

Byte	Value	RFID system	Description		
Byte 16	field_on_ time	MOBY U	Time for BERO mode (field_ON_control = 02)		
			0x00	Timeout monitoring is deactivated. To switch the field off, the 2nd BERO is required.	
			0x01 ... 0xFF	1 ... 255 s turn on time for the reader antenna field	
		MOBY D	Transponder type		
			0 ... 255	Transponder type	
			0x00	I-Code 1 (e.g. MDS D139)	
			0x01	ISO transponder	
			0x02	I-Code 1 and ISO transponder	
			0x03	ISO-my-D (with SLG D10S only; the value "0x01" is set for ISO-my-D with SLG D11S / D12S)	
			0x04	ISO-FRAM (with SLG D11S / D12S only; the value "0x01" is set for ISOFRAM with SLG D10S)	
		RF200	Transponder type		
			0x01		Any ISO transponder
		RF300	This parameter decides the mode the reader will be operated in relating to the transponders. This parameter decides the selection or combination of transponder types for the reader. The table below shows the possible settings. With the setting "0x01" (ISO general) the readers of generation 2 always use the commands with which the highest performance can be achieved for the given transponder. With readers of generation 1, the value "01" activates the general ISO mode with rudimentary ISO commands (see note No. 4). With this setting, the performance is generally limited, but the operation is basically guaranteed with each ISO-compatible transponder. The transponder chip types of the transponders specified in the system manual "SIMATIC RF300" in the section "RF300 transponders" support these commands. The following values can be set:		
			0x00	RF300	For all transponders of the type "RF3xxT"
			0x01	Any ISO transponder	Activation of the general ISO mode with rudimentary ISO commands. With this setting, operation is basically guaranteed with every ISO-compatible transponder.
			0x03	ISO my-d (Infineon SRF 55V10P)	e.g. MDS D324, D339
			0x04	ISO (Fujitsu MB89R118)	e.g. MDS D421, D422, D423, D424, D425, D426, D428, D460
			0x05	ISO I-Code SLI (NXP SL2 ICS20)	e.g. MDS D100, D124, D126, D139, D150, D165
			0x06	ISO Tag-it HFI (Texas Instruments)	e.g. MDS D200 (order no. GT2600-1AA00-0AX0), D261

<table>
<tr><th>Byte</th><th>Value</th><th>RFID system</th><th colspan="3">Description</th></tr>
<tr><td rowspan="11"></td><td rowspan="11"></td><td rowspan="10"></td><td>0x07</td><td>ISO (ST LRI2K)</td><td>e.g. MDS D200 (order no. 6GT2600-1AA01-0AX0), D261</td></tr>
<tr><td>0x08 1)</td><td>ISO (Fujitsu MB89R112)</td><td>e.g. MDS D521, D522, D524, D525, D528</td></tr>
<tr><td>0x10 1)</td><td>RF300</td><td>For all transponders of the type "RF3xxT"</td></tr>
<tr><td>0x20 1)</td><td>MOBY E</td><td>e.g. MDS E600, E611, E623, E624</td></tr>
<tr><td>0x31 1)</td><td>ISO, RF300, MOBY E</td><td>Activation of the so-called "General mode" for processing all possible transponder types currently RF3x0T, ISO 15693 and MOBY E. With this setting, operation is basically guaranteed with every compatible transponder. If ISO mode and RF300 mode are activated a transponder can be processed up to maximum address "8192" even if the RF300 transponder has a memory of 64 kB.</td></tr>
<tr><td colspan="3">Note:
• The following ISO special functions are not supported:
– AFI (Application Family Identifier)
– DSFID (Data Storage Format Identifier)
– Chip-specific added functions such as EAS, Kill commands, etc.
• If a previously unknown transponder cannot be identified based on the parameters above, an error message is generated ("error_MOBY 0D" [hex]).
• Invalid parameters are rejected with an error message ("error_MOBY 15" [hex]).
• By selecting the values "03 ... 07", chip-specific commands are used if they exist. The commands have a positive effect on the communication time between the reader and transponder and can therefore also allow faster data transfer depending on the transponder.
• With "TAG_TYPE(ftim) = 01", RF300 readers of generation 1 support the ISO commands "Inventory", "ReadSingleBlock", "WriteSingleBlock", "LockBlock," "GetSystemInformation" and for multitags also "Select" and "ResetToReady".</td></tr>
<tr><td colspan="3">1) Applies only to readers of generation 2.</td></tr>
<tr><td>RF600</td><td colspan="3">ETSI/India variant: 0x00 ... 0x0F</td></tr>
<tr><td colspan="2">Changing the channel assignment in the ETSI wireless profile ("scanning_time = 0x02"):
 
</td><td>Changing the channel assignment in the India wireless profile ("scanning_time = 0xC0"):
 
</td></tr>
</table>

Byte	Value	RFID system	Description
			0x00: Default; the channels of the reader are used in four channel mode. Note: The setting "0x0F" is identical to "0x00". With bits 0 to 3 of the "field_ON_time" byte, a channel (frequency) plan can be created for the situation in which several readers are operated in close proximity. Readers that use different channels will interfere with each other to a lesser extent. If only one channel is used per reader, the reader must pause for 100 ms at intervals of 4 seconds (as of ETSI EN 302 208 V1.2.1). With time-critical applications, a smaller loss in performance can therefore be assumed in contrast to 2 to 4-channel mode of a reader. If 2 to 4 channels per reader are used, the reader switches to another channel after 0.1 seconds in two-antenna mode and after 4 seconds in single-antenna mode. If only one of the 4 channels is selected, a pause of 100 ms is forced after 4 seconds according to the standard. FCC and CMIIT variant: Normal: 0x00

3.5.4.3 Expanded commands for optical reader systems (MV400)

The "WRITE-CONFIG" command

During initialization ("INIT"), the Ident profile automatically executes the "WRITE-CONFIG" command. The parameter values of the "WRITE-CONFIG" command depend on whether the Ident profile is used with or without a communications module.

Table 3- 74 WRITE-CONFIG

CMD	OFFSET BUFFER	LEN_DATA	CONFIG	TXREF
0x78	Offset in the "TXREF" send buffer	Length of the parameter data	• 0x01 ≙ communication reset, no configuration data (LEN_DATA = 0) • 0x03 ≙ communication reset, configuration data to be sent	Configuration data to be sent

Structure of the configuration data attachment of WRITE-CONFIG

Table 3- 75 MV400 when CONFIG = 0x03; LEN_DATA = 0x10

Byte	1	2...5	6	7...8	9	10	11	12 ... 13	14	15	16
Value	0x04	0x00	0x0A	0x00	0x00	0x25	0x02	0x00	0x01	0x00	0x00 ... 0x0F [1]

[1] 0x00: "INIT" without program selection

0x01 ... 0x0F: Number of the program to be started ("INIT" with program selection)

The "PHYSICAL-WRITE" command

The optical reader systems MV400 have further commands that can be transferred with the "PHYSICAL-WRITE" command.

Table 3- 76 PHYSICAL-WRITE

CMD	OFFSET BUFFER	ADDR_TAG	LEN_DATA	TXREF
0x71	Offset in the "TXREF" send buffer	0x0000	Length of data to be sent to the reader:	Sub command with data to be sent to the reader. The first SINT contains the command identifier:
			• 02	• 01 = program change
			• 01	• 02 = activate read program number
			• Match string length + 3	• 03 = write match string
			• 01	• 04 = activate read match string
			• 01	• 05 = set Disa bit
			• 01	• 06 = reset Disa bit
			• Total length of the XMATCH user data + 4	• 07= write trigger-synchronized match string (XMATCH)
			• 07	• 08 = set Digital Out

Table 3- 77 Command data area "TXREF" command identifier 03 (write match string)

Address	Value	Description
0x0000	0x03	Command identifier "Write match string"
0x0001	0x00-0xFF	Match string length high byte
0x0002	0x00-0xFF	Match string length low byte
0x0003	--	1st character of the match string
...	...	...
n + 2	--	(n-1)th character of the match string
n + 3	--	nth character of the match string

Table 3- 78 Command data area "TXREF" command identifier 07 (XMATCH)

Address	Value	Description
0x0000	0x07	Command identifier "XMATCH"
0x0001	0x00	Reserved
0x0002	You will find detailed information in the manual "SIMATIC MV420 / SIMATIC MV440".	XMATCH user data
…		
0xN		

Table 3- 79 Command data area "TXREF" command identifier 08 (set Digital Out)

Address	Value	Description
0x0000	0x08	Command identifier "Set digital out".
0x0001	0x1-0x4	Number of the logical external signal. Corresponds to "EXT_1", "EXT_2", "EXT_3" and "EXT_4".
0x0002	0x0-0x2	Level of the signal • 0x0: Set level statically to "low". • 0x1: Set level statically to "high". • 0x2: Set level for configured pulse time to "high".
0x0003	0x1-0x7	Type of logic operation • 0x1: Logical "OR" • 0x2: Logical "AND" • 0x3: Logical "Exclusive OR" • 0x4: no logic operation • 0x5: Logical "OR not" • 0x6: Logical "AND not" • 0x7: Logical "Exclusive OR not"
0x0004	0x0-0x5	Logical signal linked to. If the logic operation type is 0x4, the parameter has no significance. • 0x0: Logical signal "IN_OP" • 0x1: Logical signal "TRD" • 0x2: Logical signal "RDY" • 0x3: Logical signal "READ" • 0x4: Logical signal "MATCH" • 0x5: Logical signal "NOK"
0x0005	0x0	Reserved, must be 0x0 to retain upwards compatibility.
0x0006	0x0	Reserved, must be 0x0 to retain upwards compatibility.

The "PHYSICAL-READ" command

The "PHYSICAL-READ" command is used for the following functions:

- Reading codes
- Follow-on command after "activate read program number" for reading out the program number
- Follow-on command after "activate read match string" for reading out the match string

Table 3- 80 PHYSICAL-READ

CMD	OFFSET BUFFER	ADDR_TAG	LEN_DATA	RXREF
0x70	Offset in the "TXREF" send buffer	0x0000	Length of the data to be fetched from the reader:	Data fetched from the reader:
			• ≥ code length +2	• Code data
			• = 01	• Program number
			• ≥ Match string length +2	• Match string

3.5.4.4 Effect of the commands

The commands used take effect as follows:

- The input parameters "INIT" and "RESET" interrupt command execution within the communications module.
- The completed message that follows the "INIT" or "SRESET" ("DONE" or "ERROR") always relates to the input parameter "INIT" or "SRESET" and not to the interrupted command.
- The input parameter "INIT" resets communication between the Ident profile and the communications module. Following "hard" resetting of the communications module, the Ident profile automatically transfers the "WRITE-CONFIG" command to the communications module. This is why it is absolutely necessary that you store the "WRITE-CONFIG" command in the first element of the command buffer "CMDREF".
- The "WRITE-CONFIG" command resets all functions within the communications module, with the exception of the communication.
- The parameter "SRESET" interrupts a running command.

3.5.4.5 Editing commands

Follow the steps below to edit the commands:

1. Write the "CMDREF" (Array [1…10]) parameter with the required commands.

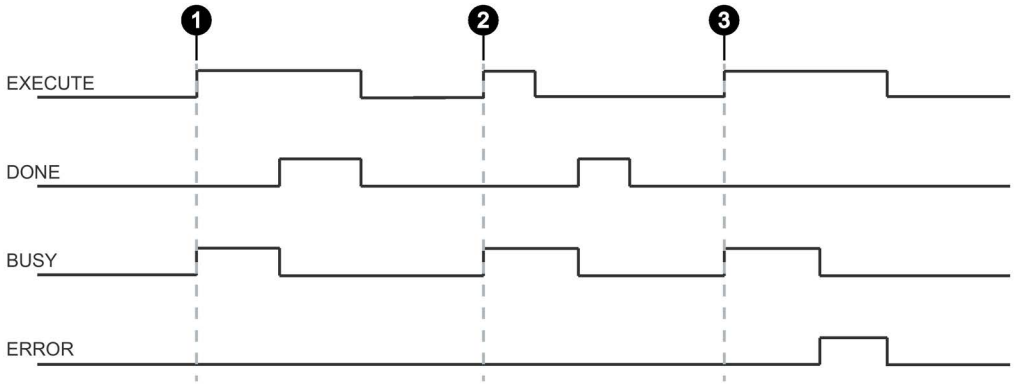
   The content of "CMDREF" = [1] is reserved for initialization. It is executed when the "INIT" input of the Ident profile is set and "CMDSEL" is = [1].
2. Transfer the data to be written to the send data buffer "TXBUF".
3. Select the previously written command (Array [1…10]) with the parameter "CMDSEL".
4. Execute the command using the "EXECUTE" parameter ("EXECUTE" = 1").

   Wait until the bits "BUSY = FALSE" and "DONE = TRUE" are set.

   The command is now executed free of errors.

   If "ERROR = TRUE" is set, continue at point 5. Otherwise, continue with Step 6.
5. Evaluate the errors that have occurred.
6. Reset the "EXECUTE" bit.

The following diagram illustrates the running of the Ident profile over time. A command is always started on the positive edge of "EXECUTE", "INIT" or "SRESET".



Case ① By setting EXECUTE (EXECUTE = 1) the function/instruction is started. If the job was completed successfully (DONE = 1), you need to reset EXECUTE. DONE is reset at the same time.

Case ② EXECUTE is set for only one cycle. As soon as BUSY is set (and DONE is reset), you can reset EXECUTE again. If the job was completed successfully, DONE is set for one cycle.

Case ③ Handling as in Case 1, however with error output. As soon as ERROR is set, the precise error code is available in the STATUS output. ERROR and STATUS retain their value as long as EXECUTE is set.

Figure 3-36 General sequence of the Ident profile

3.5.4.6 Parameter assignment for starting up and restarting

The communications module and the reader are restarted by setting the "INIT" parameter. With the parameter, the CM or the reader and the Ident profile are reassigned parameters and synchronized.

An "INIT" is necessary after

- switching on or restarting the SIMATIC controller (OB 100 / Startup)
- turning on the power supply of the CM/reader
- plugging the reader onto the CM
- interruption in PROFIBUS/PROFINET communication
- An error message by the "STATUS" parameter

3.5.4.7 Chaining

With the Ident profile and the Advanced block, it is possible to send chained commands. Chained commands are sent in their entirety to the reader without waiting for the results of the first command. This function allows you to execute various transponder commands with one command start.

With both blocks, you have a command buffer of 10 commands available (Array [1...10] of the "IID_CMD_STRUCT"). In each command structure there is a "chained" bit. This bit must be set for each chained command. In the last chained command, this bit must not be set so that the block recognizes that the chain has ended.

Note

Chaining function is device-specific

Please check whether or not the Ident device you are using supports chaining.

Chaining is currently supported only by the RF680R/RF685R readers (status October 2014)

Overview of the commands

Table 3- 81 Overview of the commands with which chaining is possible

Command	Command code		Description
	HEX	ASCII	
PHYSICAL-READ	70	'p'	Reads data from a transponder / optical reader system by specifying the physical start address, the length and the password.
PHYSICAL-WRITE	71	'q'	Writes data to a transponder / optical reader system by specifying the physical start address, the length and the password.
READER-STATUS	74	't'	Reads out the status of a communications module. This command must not be the last command within the chain.
TAG-STATUS	73	's'	Reads out the status of a transponder.
INVENTORY	69	'i'	Requests a list of all currently accessible transponders within the antenna range.
WRITE-ID	67	‘g‘	RF680R/RF685R: Writes a new EPC-ID to the transponder.
KILL-TAG	6A	‘j‘	RF680R/RF685R: The transponder is permanently deactivated.
LOCK-TAG-BANK	79	‘y‘	RF680R/RF685R: Defines a password for transponder access.

Example of command structure

Table 3- 82 Example of a command structure with 3 commands (without EPC-ID)

Command	Parameter	Value	Description
Command 1	IID_CMD_STRUCT[2].CMD	0x69	Execute an inventory with a duration of 2 inventories.
	IID_CMD_STRUCT[2].ATTRIBUTES	0x80	
	IID_CMD_STRUCT[2].EXT_UHF. INVENTORY.DURATION	2	
	IID_CMD_STRUCT[2].EXT_UHF. INVENTORY.DUR_UNIT	1	
	IID_CMD_STRUCT[2].OPTIONS.CHAINED	true	
Command 2	IID_CMD_STRUCT[3].CMD	0x70	Read 10 bytes from the user bank starting at address 0.
	IID_CMD_STRUCT[3].EXT_UHF. MEM_BANK	3	
	IID_CMD_STRUCT[3].LEN_DATA	10	
	IID_CMD_STRUCT[3].ADDR_TAG	0	
	IID_CMD_STRUCT[3].OPTIONS.CHAINED	true	
Command 3	IID_CMD_STRUCT[4].CMD	0x71	Write 10 bytes to the user bank starting at address 20.
	IID_CMD_STRUCT[4].EXT_UHF. MEM_BANK	3	
	IID_CMD_STRUCT[4].LEN_DATA	10	
	IID_CMD_STRUCT[4].ADDR_TAG	20	
	IID_CMD_STRUCT[4].OPTIONS.CHAINED	false	

In the chaining, the entire "IID_CMD_STRUCT" buffer ("IID_CMD_STRUCT[1...10]") can be used. The start of the chain is set with the "CMDSEL" parameter.

If several commands are executed in the chain for which data is returned, the position of the data in the receive buffer "RXREF" can be set for each individual command using the "IID_CMD_STRUCT[x].OFFSETBUFFER" parameter.

Note

"IID_CMD_STRUCT[1]" reserved for "INIT"

In the Ident profile, the "IID_CMD_STRUCT[1]" parameter is normally reserved for "INIT". If you want to use "IID_CMD_STRUCT[1]" for another command, make sure that the reset parameters are written into this parameter when there is an "INIT".

3.5.4.8 Command repetition

The Ident profile supports command repetition (Repeat command).

The command repetition is currently supported by the readers RF680R/RF685R as of V3.0 and ASM 456 as of 5.0 in conjunction with the readers RF680R/RF685R (November 2016).

Note

Command repetition function is device-specific

With all devices please check whether or not the Ident device you are using supports command repetition.

How it works

After restart (or "INIT") of the reader, the Ident profile transfers the command or command chain once to the reader. Transmission of the command is automatic with the first "EXECUTE". This command (or the last command or the command chain) always remains buffered on the reader. If command repetition is started, the temporarily stored command on the reader is executed again, and the result(s) transferred to the Ident profile.

Make sure that the "EPC-ID/UID" of the commands to be repeated have the value 0. If the EPC-ID as a different value, an error message is generated.

Effects of command repetition

- The data transfer on PROFIBUS/PROFINET is minimized. This reduction has a positive effect particularly with extensive bus configurations and slow transmission speeds.
- The reader processes each transponder regardless of the Ident profile. This has a particularly advantageous effect on gate applications since all transponders are always identified with the full reader scan speed.
- Total data throughput is increased considerably particularly with controllers that have few system resources for acyclic frames.

Overview of the commands

Table 3- 83 Overview of the commands with which command repetition is possible

Command	Command code		Description
	HEX	ASCII	
PHYSICAL-READ	70	'p'	Reads data from a transponder / optical reader system by specifying the physical start address, the length and the password.
PHYSICAL-WRITE	71	'q'	Writes data to a transponder / optical reader system by specifying the physical start address, the length and the password.
INVENTORY	69	'i'	Requests a list of all currently accessible transponders within the antenna range.

Command	Command code		Description
	HEX	ASCII	
KILL-TAG	6A	‘j‘	RF680R/RF685R: The transponder is permanently deactivated.
LOCK-TAG-BANK	79	‘y‘	RF680R/RF685R: Defines a password for transponder access.

Starting command repetition

You have the option of using command repetition with or without transfer of the command. The various procedures are described below.

Sequence of the repeat command with simultaneous command transfer:

1. If applicable, enable the presence mode (with RF68xR).
2. Start the command using the input parameter "EXECUTE" while "RPTCMD" is set at the same time. ①

   The command is processed and the result transferred to the Ident profile.

   The Repeat command is activated on the reader.
3. The reader confirms activation with the output parameter "RPTACT" of the Ident profile. The confirmation is made only after the first command has been executed. ②

   The reader executes the command automatically as soon as a transponder is identified in the antenna field.

   If the reader does not support the Repeat command, "RPTACT" remains inactive. If "EXECUTE" is nevertheless set, the error "E7FE0900h" is output after a timeout of 10 seconds.
4. You can read out the individual results by repeatedly setting the "EXECUTE" input parameter. ③

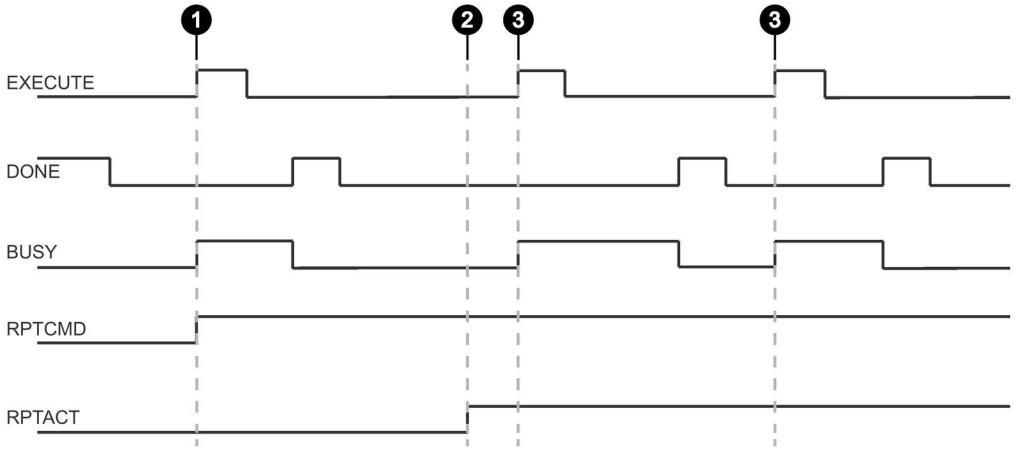


Figure 3-37 Sequence of the repeat command with simultaneous command transfer

Sequence of the repeat command without command transfer:

This sequence is only possible if the command involved has already been transferred.

1. If applicable, enable the presence mode (with RF68xR).
2. Set the "RPTCMD" input parameter. ①

   The Repeat command is activated on the reader.
3. The reader confirms activation with the output parameter "RPTACT" of the Ident profile. The confirmation is made only after the first command has been executed. ②

   If the reader does not support the Repeat command, "RPTACT" remains inactive. If "EXECUTE" is nevertheless set, the error "E7FE0900h" is output after a timeout of 10 seconds.
4. You can read out the individual results by repeatedly setting the "EXECUTE" input parameter. ③

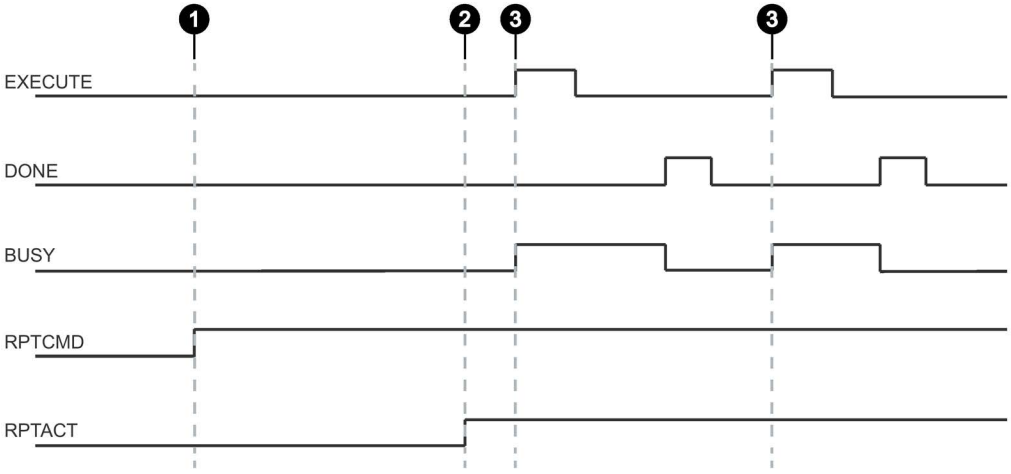


Figure 3-38 Sequence of the repeat command without command transfer

Ending command repetition

You have the option of ending command repetition by resetting "RPTCMD" or using the "INIT" or "SRESET" commands. The various procedures are described below

End the Repeat command and reset "RPTCMD":

1. Reset the "RPTCMD" input parameter. ①
2. Fetch any existing acknowledgments using the "EXECUTE" input parameter. ②

   The "RPTACT" output parameter remains set by the reader as long as there are acknowledgements present.
3. When there are no more acknowledgments, "RPTACT" is reset by the reader. ③

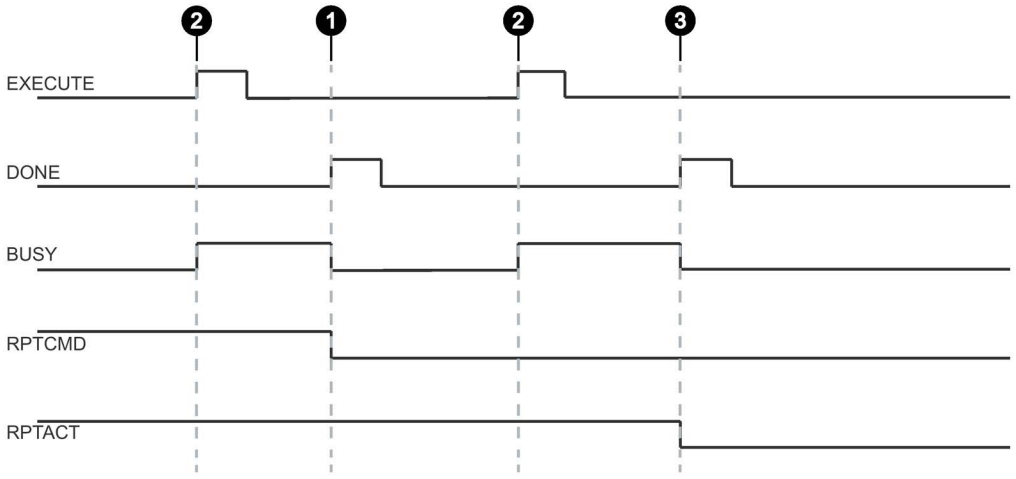


Figure 3-39 End the Repeat command by resetting "RPTCMD" (ended normally)

The "RPTACT" output parameter is reset by the reader. Under certain circumstances, it is possible that resetting "RPTACT" will be delayed. In other words not at the same time as the "DONE" of the last acknowledgement. If the block is now restarted with "EXECUTE" and "RPTACT" is still set although there are no longer any results in the buffer, the block is not ended (BUSY = 1). In this case, you can wait until the next transponders are read out. As an alternative, the block can be ended with "INIT" or "SRESET".

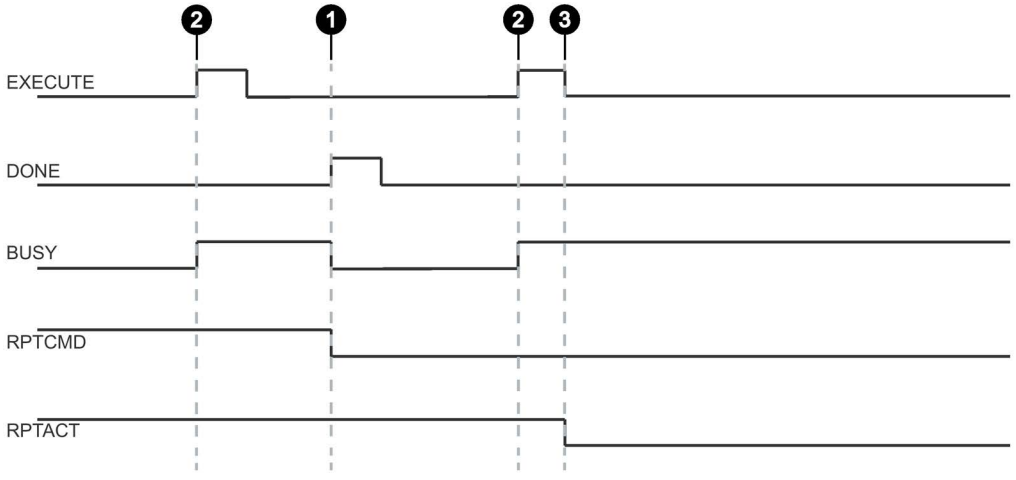


Figure 3-40 End the Repeat command by resetting "RPTCMD" (the last command remains pending)

Note

End the Repeat command with "INIT" or "SRESET"

End the Repeat command using the input parameters "INIT" or "SRESET" if it is not known how many transponders were still processed after resetting the "RPTCMD" input parameter.

Normally, an "SRESET" is performed significantly faster because no reset routine is run through.

Ending the Repeat command with "INIT":

1. Reset the "RPTCMD" input parameter and set the "INIT" input parameter. ①

   If "RPTCMD" is not reset, the Repeat command is activated again on the reader. This response triggers an error message because there is no command.
2. The reader resets the "RPTACT" output parameter due to the "INIT" input parameter. ②

Ending the Repeat command with "SRESET":

1. Reset the "RPTCMD" input parameter and set the "SRESET" input parameter. ①
2. The "DONE" output parameter is set and the reader resets the "RPTACT" output parameter. ②

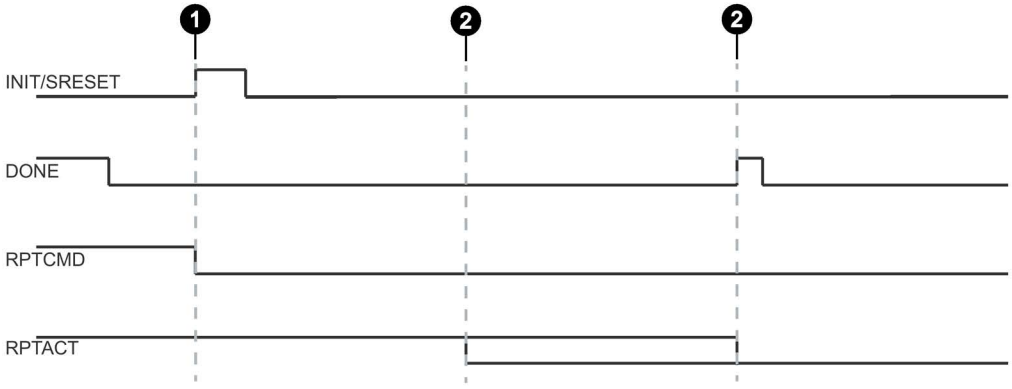


Figure 3-41 Ending the Repeat command with "INIT"/"SRESET"

Data buffer

Permanent command repetition can lead to the data being transferred more slowly to the Ident profile than new transponders are processed. In this case, the reader buffers the results. The reader has a number of buffers for this. If the buffers are full, no new data is fetched by the Ident profile; in other words newly arriving transponders are no longer processed.

Table 3- 84 Readers and communications modules that support command repetition

Device type	Number of buffers (number of commands)	Max. user data that can be processed with command repetition
RF300 reader	246	233 bytes × 246 = 57 318 bytes
RF620R/RF630R	150	233 bytes × 150 = 34 950 bytes
RF680R/RF685R	250	1034 bytes × 250 = 258 500 bytes
RF180C	150	233 bytes × 150 = 34 950 bytes
ASM 456	150	233 bytes × 150 = 34 950 bytes

Note

Restriction of command repetition

In the case of RFID systems with unique tag IDs (UID or EPC-ID) (e.g. RF300, RF600, MOBY U), the stored command is only repeated when different transponders enter the antenna field. If the same transponder (identical UID / EPC-ID) enters the antenna field again and again, the transponder will not be processed again.

3.6 Transponder addressing

Addressing

Addressing of the transponders is linear from address "0000" (or the specified start address) to the end address. The CM or reader automatically recognizes the size of the memory on the transponder. If the end address on the transponder is exceeded, you receive an error message.

The next table shows the address space of the individual transponder parameters. The "ADDR_TAG" and "LEN_DATA" parameters must be assigned according to this address space.

Address space of the transponder/MDS variants according to ISO 15693 for RF200, RF300 and MOBY D

System	Addressing	16-bit hexadecimal number
RF200, RF300, MOBY D	**MDS D139 (I-Code 1; 44 bytes)**	
	Start address	0000
	End address	002B
	ID-Nr.: (fixed-coded, can only be read as a whole)	
	Start address	FFF0
	Length	0008
	ISO-MDS (I-Code SLI; 112 bytes)	
	Start address	0000
	End address	006F
	ID-Nr.: (fixed-coded, can only be read as a whole)	
	Start address	FFF0
	Length	0008
	ISO MDS (Tag-it HF-I; 256 bytes)	
	Start address	0000
	End address	00FF
	ID-Nr.: (fixed-coded, can only be read as a whole)	
	Start address	FFF0
	Length	0008

System	Addressing	16-bit hexadecimal number
	ISO MDS (my-d SRF55V10P; 992 bytes)	
	Start address	0000
	End address	03DF
	ID-Nr.: (fixed-coded, can only be read as a whole)	
	Start address	FFF0
	Length	0008
	ISO-MDS (MB 89R118B, 2000 bytes)	
	Start address	0000
	End address	07CF
	ID-Nr.: (fixed-coded, can only be read as a whole)	
	Start address	FFF0
	Length	0008

Address space of the transponder versions for RF300

System	Addressing	16-bit hexadecimal 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 RF300 system manual))	
	Start address	FF00
	End address	FF13
	ID-Nr.: (fixed-coded; can only be output as a whole)	
	Start address	FFF0
	Length	0008
	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 RF300 system manual))	
	Start address	FF00
	End address	FF13
	R/W memory (FRAM)	
	Start address	0000
	End address	1FFC
	Id-Nr.: (fixed-coded, can only be read out as a whole)	
	Start address	FFF0
	Length	0008
	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 RF300 system manual))	
	Start address	FF00
	End address	FF13

System	Addressing	16-bit hexadecimal number
	R/W memory (FRAM)	
	Start address	0000
	End address	7FFC
	ID-Nr.: (fixed-coded; can only be output as a whole)	
	Start address	FFF0
	Length	0008
	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	FF00
	End address	FF13
	R/W memory (FRAM)	
	Start address	0000
	End address	FEFC
	ID-Nr.: (fixed-coded; can only be output as a whole)	
	Start address	FFF0
	Length	0008

RF300: General notes on the meaning of the OTP memory

RF300 transponders and ISO transponders have a memory area that can be protected against overwriting. This memory area is called OTP. The following 5 block addresses are available for activating the OTP function:

- FF80
- FF84
- FF88
- FF8C
- FF90

A write command to this block address with a valid length (4, 8, 12, 16, 20 depending on the block address) protects the written data from subsequent overwriting.

Note

Using the OTP area only in static mode

Only use the OTP area in static mode.

Note

Use of the OTP area is not reversible

If you use the OPT area, you cannot undo this assignment, because the OPT area can only be written to once.

RF300: Address mapping of OTP memory on the RF300 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		RF300, write OTP once	
Address	Length	Address	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		

Note

Enabling write protection

Write access to addresses starting at FF80 to FF93 activates the write protection (OTP function) on the EEPROM user memory. This operation is not reversible. Switching on write protection must always take place in ascending order without gaps, starting at address FF80.

Address space of the transponder versions for RF600

Table 3- 85 Address spaces of the transponder variants for RF620R/RF630R

Tags	Chip type	User [1] [hex]	EPC		TID (read only)	RESERVED (passwords)	Special	
		Area / length	Area / length (max. and default)	Access	Area / length	Area / length	KILL-PW	Lock function
RF630L (-2AB00, -2AB01)	Impinj Monza 2	-	FF00-FF0B / 96 bits FF00-FF0B / 96 bits	read/ write	FFC0-FFC3 4 bytes	FF80-FF87 8 bytes	yes	yes
RF630L (-2AB02)	Impinj Monza 4QT [2]	00 - 3F 64 bytes	FF00-FF0F / 128 bits FF00-FF0B / 96 bits	read/ write	FFC0-FFCB 12 bytes	FF80-FF87 8 bytes	yes	yes
RF630L (-2AB03)	NXP G2XM	00 - 3F 64 bytes	FF00-FF1D / 240 bits FF00-FF0B / 96 bits	read/ write	FFC0-FFC7 8 bytes	FF80-FF87 8 bytes	yes	yes
RF640L	Alien Higgs 3	00 - 0F/3F [3] 16/64 bytes	FF00-FF3C / 480 bits FF00-FF0B / 96 bits	read/ write	FFC0-FFD8 24 bytes	FF80-FF87 8 bytes	yes	yes
RF680L	NXP G2XM	00 - 3F 64 bytes	FF00-FF1D / 240 bits FF00-FF0B / 96 bits	read/ write	FFC0-FFC7 8 bytes	FF80-FF87 8 bytes	yes	yes
RF690L	Alien Higgs 3	00 - 0F/3F [3] 16/64 bytes	FF00-FF3C / 480 bits FF00-FF0B / 96 bits	read/ write	FFC0-FFD8 24 bytes	FF80-FF87 8 bytes	yes	yes
RF610T	NXP G2XM	00 - 3F 64 bytes	FF00-FF1D / 240 bits FF00-FF0B / 96 bits	read/ write	FFC0-FFC7 8 bytes	FF80-FF87 8 bytes	LOCKED	yes
RF620T	Impinj Monza 4QT [2]	00 - 3F 64 bytes	FF00-FF0F / 128 bits FF00-FF0B / 96 bits	read/ write	FFC0-FFCB 12 bytes	FF80-FF87 8 bytes	LOCKED	yes
RF625T	Impinj Monza 4QT [2]	00 - 3F 64 bytes	FF00-FF0F / 128 bits FF00-FF0B / 96 bits	read/ write	FFC0-FFCB 12 bytes	FF80-FF87 8 bytes	LOCKED	yes
RF630T	NXP G2XM	00 - 3F 64 bytes	FF00-FF1D / 240 bits FF00-FF0B / 96 bits	read/ write	FFC0-FFC7 8 bytes	FF80-FF87 8 bytes	LOCKED	yes
RF640T	NXP G2XM	00 - 3F 64 bytes	FF00-FF1D / 240 bits FF00-FF0B / 96 bits	read/ write	FFC0-FFC7 8 bytes	FF80-FF87 8 bytes	LOCKED	yes
RF680T	NXP G2XM	00 - 3F 64 bytes	FF00-FF1D / 240 bits FF00-FF0B / 96 bits	read/ write	FFC0-FFC7 8 bytes	FF80-FF87 8 bytes	LOCKED	yes
RF630T	NXP G2XM	00 - 3F 64 bytes	FF00-FF1D / 240 bits FF00-FF0B / 96 bits	read/ write	FFC0-FFC7 8 bytes	FF80-FF87 8 bytes	LOCKED	yes

Tags	Chip type	User [1] [hex]	EPC		TID (read only)	RESERVED (passwords)	Special	
		Area / length	Area / length (max. and default)	Access	Area / length	Area / length	KILL-PW	Lock function
RF640T	NXP G2XM	00 - 3F 64 bytes	FF00-FF1D / 240 bits FF00-FF0B / 96 bits	read/ write	FFC0-FFC7 8 bytes	FF80-FF87 8 bytes	LOCKED	yes
RF680T	NXP G2XM	00 - 3F 64 bytes	FF00-FF1D / 240 bits FF00-FF0B / 96 bits	read/ write	FFC0-FFC7 8 bytes	FF80-FF87 8 bytes	LOCKED	yes

1) The user area also applies to the new readers RF650R/RF680R/RF685R in memory bank 3.

2) Uses User Memory Indicator (UMI).

3) The EPC memory area of the Alien Higgs chips can be increased at the cost of the user memory. You will find further information in the relevant transponder sections.

Address spaces of the transponder variants for RF650R/RF680R/RF685R

With the new readers RF650R/RF680R/RF685R, the user data, TID, EPC and passwords are read out via the relevant memory banks. To read out the required data, the relevant memory bank must be selected.

The table above shows the area and length of the user data ("USER" column). You can read out the EPC-ID using an inventory command. As an alternative, you can also read out the EPC-ID using a Read command to memory bank 1, start address 0x04.

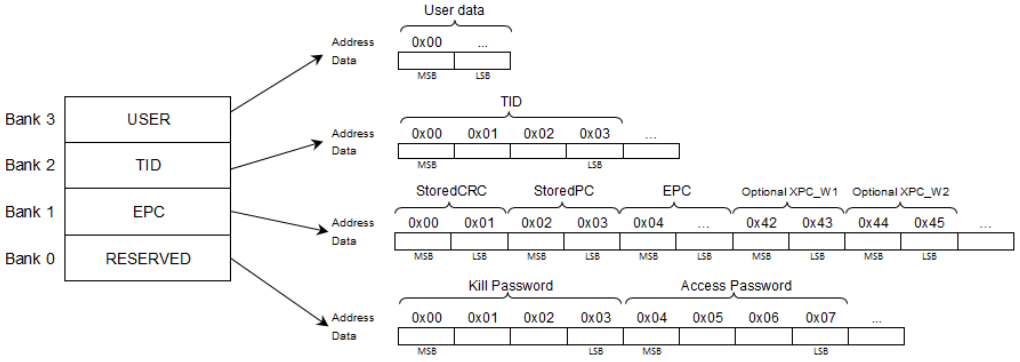


Figure 3-42 Memory configuration

Address space of the transponder/MDS variants for MOBY U

System	Addressing	16-bit hexadecimal number
MOBY U	**2 KB data memory**	
	Start address	0000
	End address	07FF
	Read OPT memory (write access only possible once. The OTP memory of MOBY U can only be processed completely. In other words, the start address must always be specified with value FFF0 hex and the length with value 10 hex.)	
	Start address	FFF0
	Length	10
	ID-Nr. (4 fixed-coded bytes; can only be read out with the MDS-Status command)	
	32 KB data memory	
	Start address	0000
	End address	7FFF
	Read OTP memory (write access only possible once)*	
	Start address	FFF0
	Length	10
	ID-Nr.: (4 fixed-coded bytes; can only be read out with the MDS-Status command)	

Error messages

4

4.1 Structure of the "STATUS" output parameter

There is always an error status in the Ident profile function if the output parameter "ERROR = TRUE" is set. The error can be analyzed (decoded) using the "STATUS" output parameter.

The "STATUS" output parameter is made up of the following 4 bytes:

Table 4- 1 Bytes of the "STATUS" output parameter

Byte	Meaning
Byte 0	Function numbers • 80/C0/DE/DF - error in bus communication (backplane bus, PROFINET, PROFIBUS) • E1 - transponder-related error • E2 - error on the air interface • E4 - reader hardware fault • E5 - error in the communication between reader and FB • E6 - error in the user command • E7 - error message generated by the FB
Byte 1	Error numbers This byte defines the meaning of the error code and the warnings. The error numbers have the following meaning: • 0x00 - no error, no warning • 0x80 - error message from the backplane bus or from PROFIBUS DP-V1 or PROFINET (in accordance with IEC 61158-6) • 0x81...0x8F - The controller reports an error according to the parameter "x" (0x8x). • 0xFE - error from the Ident profile or communications module/reader
Byte 2	Error code
Byte 3	Warnings In this byte, each bit has a separate meaning.

4.2 STEP 7 - error messages

If you have inserted the blocks and data types in your project and you encounter problems during compilation, please check the following points:

- The block name, the block number and the data type name must not be changed.
- The content of the data types "IID_CMD_STRUCT" and "IID_HW_CONNECT" must not be changed. The content may only be adapted when these have been linked into a data block as variables.
- If you use the Ident blocks, the data types "IID_HW_CONNECT" and "IID_CMD_STRUCT" and the block "Ident_Profile_1KB" must always exist in your project.

4.3 Errors from the communications module/reader

The causes of these errors can, for example, be as follows:

- Errors have occurred in communication between the CM and the reader or between the reader and the transponder.
- The communications module is unable to process the command.

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

Table 4- 2 Error messages from communications module/reader or from the Ident profile via the "STATUS" output parameter

Error message (hex)	Description
0xE1FE0100	Memory of the transponder cannot be written to • Transponder memory is defective • Transponder EEPROM was written too frequently and has reached the end of its service life • Transponder is write protected (Memory Lock)
0xE1FE0200	Presence error: The transponder has moved out of the transmission window of the reader. The command was executed only partially. Read command: "IDENT_DATA" has no valid data. Write command: The transponder that has just left the antenna field contains an incomplete data record. • Operating distance from reader to transponder is not being maintained • Configuration error: The data record to be processed is too large (in dynamic mode) • With timeout: No transponder in the antenna field
0xE1FE0300	Address error The address area of the transponder has been exceeded. • Start address of the command start has been incorrectly set • Transponder is not the correct type • Attempted write access to write-protected areas

Error message (hex)	Description
0xE1FE0400	Only during initialization: Transponder is unable to execute the initialization command • Transponder is defective
0xE1FE0600	Error in transponder memory The transponder has never been written to or has lost the contents of its memory due to battery failure. • Replace transponder (if battery bit is set) • Re-initialize transponder
0xE1FE0700	Password error RF620R/RF630R: Incorrect password
0xE1FE0800	The transponder in the antenna field does not have the expected UID or has no UID.
0xE1FE0A00	The transponder is read/write-protected.
0xE1FE8100	The transponder is not responding.
0xE1FE8200	The transponder password is incorrect. Access is denied.
0xE1FE8300	The verification of the written transponder data has failed.
0xE1FE8400	General transponder error
0xE1FE8500	The transponder has too little power to execute the command.

Error message (hex)	Description
0xE2FE0100	• Field disturbance on reader Reader is receiving interference pulses from the environment. – External interference field. The interference field can be detected with the "inductive field indicator" of the STG. – The distance between two readers is too short and does not correspond to the configuration guidelines – The connecting cable to the reader is disrupted, too long or does not comply with the specification – MOBY U: Transponder has left the antenna field during communication. – MOBY U: Communication between reader and transponder was aborted due to a disruption (e.g. person/foreign body moving between reader and transponder). • Too many transmit errors The transponder was unable to receive the command or the write data from the communications module correctly even after several attempts. – The transponder is positioned exactly in the limit area of the transmission window – Data transmission to the transponder is being affected by external interference • CRC sending error – The transponder reports CRC error frequently (transponder is positioned in the limit area of the reader; transponder and/or reader has a hardware defect) • Only during initialization: CRC error on receipt of acknowledgement from transponder (cause as for field interference on the reader) • When formatting, the transponder must be in the transmission window of the reader, otherwise a timeout error will occur, in other words: – The transponder is located exactly in the limit area of the transmission window – The transponder is consuming too much power (defective) – Bad FORMAT parameter setting for transponder EEPROM • RF600: – No ETSI channel free – Wrong communications standard selected in the "INIT" command – Bad expert parameter – Power check of the ETSI wireless profile is incorrect
0xE2FE0200	• More transponders are located in the transmission window than can be processed at the same time by the reader. • There is more than one transponder in the transmission window. • RF620R/RF630R: Transponder power supply close to limit. Increase the antenna power or reduce the distance to the transponder.
0xE2FE8100	There is no transponder with the required EPC-ID in the transmission window or there is no transponder at all in the antenna field.
0xE2FE8200	The requested data is not available.
0xE2FE8300	The transponder signals a CRC error.
0xE2FE8400	The selected antenna is not enabled.
0xE2FE8500	The selected frequency is not enabled.

Error message (hex)	Description
0xE2FE8600	The carrier signal is not activated.
0xE2FE8700	There is more than one transponder in the transmission window.
0xE2FE8800	General radio protocol error
0xE4FE0100	Short circuit or overload of the 24 V outputs • The reader is using too much current. • The reader cable is causing a short-circuit. Possible consequences: • 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 • and then starting "Reset_Reader"
0xE4FE0300	• Error in the connection to the reader; the reader is not answering. – The cable between the communications module and reader is wired incorrectly or there is a cable break – The 24 V supply voltage is not connected or is not on or has failed briefly – Automatic cutout on the communications module has responded – Hardware defect – Another reader is in the vicinity and is active – Execute "init_run" after correcting the error • The antenna of the reader is turned off. A tag command to the communications module was started in this status. – 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 • The mode in the "SET_ANT" command is unknown • The antenna on the reader is turned off or the antenna cable is defective
0xE4FE0400	The buffer on the communications module or reader is not adequate to store the command temporarily.
0xE4FE0500	The buffer on the communications module or reader is not adequate to store the data temporarily.
0xE4FE0600	The command is not permitted in this status or is not supported. RF680R/RF685R: Command repetition is only possible in Presence mode.
0xE4FE0700	Startup message from reader/communications module. The reader or communications module was turned off and has not yet received a "Reset_Reader" ("WRITE_CONFIG") command. • Execute "INIT" • The same physical address in the "IID_HW_CONNECT" parameter is being used more than once. Check your "IID_HW_CONNECT" parameter settings. • Check connection to the reader • The transmission speed was switched over but power has not yet been cycled
0xE4FE8100	Reserved
0xE4FE8A00	General error
0xE4FE8B00	No or bad configuration data was transferred.

Error message (hex)	Description
0xE4FE8C00	• Communication error between Ident profile and communications module. Handshake error. – UDT of this communications module is overwritten by other program sections – Check parameter settings of communications modules in the UDT – Check the Ident profile command that caused this error – Start "INIT" after correcting the error • Backplane bus / PROFIBUS DP / PROFINET error occurred This error is only indicated when access monitoring has been enabled in the PROFIBUS configuration. – Backplane bus / PROFIBUS DP / PROFINET bus connection was interrupted (wire break on the bus; bus connector on the communications module was briefly unplugged) – Backplane bus / PROFIBUS DP / PROFINET master no longer addressing communications module – Execute "INIT" – The communications module has detected a frame interruption on the bus. The backplane bus, PROFIBUS or PROFINET may have been reconfigured (e.g. with HW Config or TIA Portal)
0xE4FE8D00	• Internal communications error of the communications module/reader – Connector contact problem on the communications module / reader – Hardware of the communications module / reader has a defect; → Send in communications module / reader for repair – Start "INIT" after correcting the error • Internal monitoring error of the communications module/reader – Program execution error on the communications module / reader – Turn the power supply of the communications module/reader off and on again – Start "INIT" after correcting the error – MOBY U: Watchdog error on the reader
0xE4FE8E00	Active command canceled by "WRITE-CONFIG ("INIT" or "SRESET") or bus connector unplugged • Communication with the transponder was aborted by "INIT" • This error can only be reported if there is an "INIT" or "SRESET"
0xE5FE0100	Incorrect sequence number order (SN) on the reader/communications module
0xE5FE0200	Incorrect sequence number order (SN) in the Ident profile Possible cause: User mode "RFID standard profile" is not set in the device configuration.
0xE5FE0400	Invalid data block number (DBN) on the reader/communications module
0xE5FE0500	Invalid data block number (DBN) in the Ident profile
0xE5FE0600	Invalid data block length (DBL) on the reader/communications module
0xE5FE0700	Invalid data block length (DBL) in the Ident profile

Error message (hex)	Description
0xE5FE0800	Previous command is active or buffer overflow A new command was sent to the reader or communications module although the last command was still active. • Active command can only be terminated with an "INIT" • Before a new command can be started, "DONE bit = 1" must be set; exception: "INIT" • Two Ident profile calls had the same "HW_ID", "CM_CHANNEL" and "LADDR" parameter settings • Two Ident profile calls are using the same pointer • Start "INIT" after correcting the error • When working with command repetition (e.g., fixed code transponder), no data is being fetched from the transponder. The data buffer on the reader/communications module has overflowed. Transponder data has been lost.
0xE5FE0900	The reader or communications module executes a hardware reset ("INIT_ACTIVE" set to "1"). "INIT" is expected from the Ident profile (bit 15 in the cyclic control word).
0xE5FE0A00	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.
0xE5FE0B00	Incorrect sequence of acknowledgement frames (TDB / DBN)
0xE5FE0C00	Synchronization error (incorrect increment of AC_H / AC_L and CC_H / CC_L in the cyclic control word). "INIT" had to be executed
0xE5FE8100	Communications error between reader and communications module Access denied
0xE5FE8200	Communications error between reader and communications module Resource is occupied
0xE5FE8300	Communications error between reader and communications module Functional error of the serial interface
0xE5FE8400	Communications error between reader and communications module Other faults/errors
0xE6FE0100	Unknown command Ident profile is sending an uninterpretable command to the communications module. • The "AdvancedCmd" block was supplied with an incorrect "CMD". • The "CMD" input of the "AdvancedCmd" block was overwritten by the user. • The transponder has signaled an address error.
0xE6FE0200	Invalid command index CI

Error message (hex)	Description
0xE6FE0300	• Bad parameter assignment of the communications module or reader – Check "INPUT" parameter in the Ident profile. – Check parameter settings in HW Config / TIA Portal. – "WRITE-CONFIG" command has incorrect parameter settings. – After a startup, the reader or communications module has still not received an INIT". • The parameter assignment of the reader or communications module on PROFIBUS/PROFINET was incorrect and the command cannot be executed. – Length of the input/output areas too small for the cyclic I/O word. Correct GSD file being used? – User data length set with command (e.g. "READ") too high. • Error when processing the command – Bad data in the "AdvancedCmd" or "IID_CMD_STRUCT" (e.g. "WRITE" command with length = 0); check "AdvancedCmd" or "IID_CMD_STRUCT" and execute "INIT". – Reader/communications module hardware defective: The reader or communications module receives bad data with "INIT". – AB byte does not comply with the useful data length. • Wrong reset block was selected – Regardless of the selected reader system, use the "Reset_Reader" function block.
0xE6FE0400	Presence error: A transponder has passed by a reader without being processed by a command. • This error message is not reported immediately. Instead, the reader or communications module is waiting for the next command (read, write). 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 again by the reader/communications module. • An "INIT" from the Ident profile also resets this error status. • Bit 2 is set in the OPT1 parameter and there is no transponder in the transmission window.
0xE6FE0500	An error has occurred that makes a Reset_Reader ("WRITE-CONFIG" with "Config = 3") necessary. • The "WRITE-CONFIG" command is incorrect. • Start "INIT" after correcting the error • Check the "IID_HW_CONNECT" parameter.
0xE6FE0600	The reset timer has expired.
0xE6FE8100	A parameter is missing.
0xE6FE8200	The parameter has an invalid format.
0xE6FE8300	The parameter type is invalid.
0xE6FE8400	Unknown parameter
0xE6FE8500	The command or the frame has an invalid format.
0xE6FE8600	The inventory command failed.
0xE6FE8700	Read access to the transponder has failed.
0xE6FE8800	Write access to the transponder has failed.
0xE6FE8900	Writing the EPC-ID on the transponder has failed.

Error message (hex)	Description
0xE6FE8A00	Enabling write protection on the transponder has failed.
0xE6FE8B00	The "Kill" command failed.
0xE7FE0100	In this status, only the "Reset_Reader" command ("WRITE-CONFIG") is permitted.
0xE7FE0200	The "CMD" command code or the value in "CMD SEL" is not permitted.
0xE7FE0300	The value of the "LEN_DATA" parameter of the command is too long. It does not match the global data reserved in the send data buffer (TXBUF). Possible cause / action to be taken: • The "CMDSEL" or "CMDDIM" parameter was set incorrectly. Check the parameters. The value of the parameter "CMDSEL" must be between 1 and 100 and may never be higher than the value of the "CMDDIM" parameter (max. 100).
0xE7FE0400	The receive data buffer (RXBUF) or the send data buffer (TXBUF) is too small, the buffer created at TXBUF/RXBUF does not have the correct data types or the parameter "LEN_DATA" as a negative value. Possible cause / action to be taken: • Check whether the buffers TXBUF/RXBUF are at least as large as specified in LEN_DATA. • With S7-1200/1500: – In the Ident profile, only an "Array of Byte" may be created for TXBUF and RXBUF. – In the "Tag_Status" and "Reader_Status" block, only an "Array of Byte" or the corresponding data types ("IID_TAG_STATUS_XX_XXX" or "IID_READER_STATUS_XX_XXX") may be created With all other blocks only an "Array of Byte" may be created.
0xE7FE0500	This error tells you that only an "INIT" command is permitted as the next command. All other commands are rejected.
0xE7FE0600	Wrong index (outside range of "101 ... 108" and "-20401 ... -20418")
0xE7FE0700	The reader or communications module does not respond to "INIT" ("INIT_ACTIVE" is expected in the cyclic status message). The next steps: • Check the address parameter "LADDR".
0xE7FE0800	Timeout during "INIT" (60 seconds according to "TC3WG9")
0xE7FE0900	Command repetition is not supported.
0xE7FE0A00	Error during the transfer of the PDU (Protocol Data Unit).
0xE7FE0B00	The "CMDREF" parameter was set incorrectly. Check the parameter. The parameter "CMDREF" must be created as "ARRAY of IID_CMD_STRUCT" and may contain a max. of 100 elements.
0xFxFExx00	An "FxFExx" error is identical to the corresponding "ExFExx" error (see "ExFExx" error). Byte 3 contains additional warning information.

4.4 Errors from the optical reader

With error messages, the "IN_OP" signal (in operation) is reset and the "STATE/SF" LED is lit red permanently. In addition, the "Ready" or "Done" bit is reset with a connection via the Ident profile.

Table 4- 3 Error messages from the optical reader or from the Ident profile via the STATUS output parameter

Error message (hex)	Description
0xE1FE0200	The connection from the internal interface to the image sensor is disrupted. If the error continues to occur after turning on the device again, contact technical support.
0xE1FE0400	Transmission error The send buffer is full because the data cannot be queried in an adequately short time. Reduce the trigger frequency or process the results faster. If necessary, change the update time in the PROFINET configuration. If necessary, the transmission speed of the CM connection can also be increased. To cover short peaks, the image buffer size of the program can be increased on the "Image acquisition", "Control" tab.
0xE1FE0600	Program cannot be started because there is not enough memory or the program is damaged. Reduce the memory requirements and repeat "Save program".
0xE1FE0700	Comparison error The program could not be created due to bad match settings. Adapt the match settings or use a suitable test object.
0xE4FE0400	Internal file error An error occurred while saving to read-only memory. Please contact technical support if this error occurs frequently.
0xE4FE0600	Lamp overload The connected lamp was overloaded. The configured or default "Maximum duty cycle" in "Options", "Lighting" tab, was exceeded. Decrease the trigger frequency, reduce the exposure time or use a more powerful lamp.
0xE4FE8400	Error in last command sequence. May occur if triggering is too fast. The Ident profile can only process one command before a new command can be executed.
0xE6FE0400	The program could not be created or saved. • While the program is being saved, the DISA signal is changed at an invalid point or the time sequence of applied signals is not adhered to. Check the sequence of applied signals. Start the program saving process again. • The program could not be created. Adapt the parameter assignment, the placement of test objects in the image or the image quality. • An attempt is being made to save a program under an invalid number via the interface set in "Controller". Select a program number between 1 and 15. For SIMATIC MV420 SR-B, 1 to 5 and 15 are valid program numbers.

With the following error messages, the "IN_OP" signal (in operation) is not reset and the "STATE/SF" LED is not lit red.

Table 4- 4 Messages from the optical reader or from the Ident profile via the STATUS output parameter

Message (hex)	Description
0xE1FE0300	Bad parameter in MV command The command was incorrectly structured. Possible causes: • The specified address for a WRITE command is ≠ "0x0000". • MV command program change – Length of the data to be written > "0x1". – Program number transferred is > "0xF". The transferred program number is not saved.
0xE6FE0100	Command not permitted or the command was aborted. The precise error message can be obtained with "INIT" without program selection. Possible causes: • The send buffer is full. • The program is damaged. • The Ethernet interface is in operation and there is a problem. • The connected lamp is overloaded.
0xE6FE0300	Initialization with program selection ("init_run"/"INIT") is not possible. Possible causes: • Program number transferred is not stored. • Reader is still in self-test.

4.5 Errors from the bus/backplane bus

The transport layer of the bus system being used (backplane bus, PROFIBUS, PROFINET) is signaling an error. For precise troubleshooting and analysis, a PROFIBUS tracer can be useful. 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.

Error messages that start with "80/C0/DE/DF" relate to errors from the bus/backplane bus. You will find details of the error message in the STEP 7 help on the blocks "WRREC" or "RDREC" (SFB52/SFB53).

4.6 Warnings

Byte 3 of the "STATUS" output parameter indicates warnings if byte 0 of the "STATUS" (function numbers) has the value "Fx" or "Dx".

Table 4- 5 Possible warnings when working with the Ident profile

Bytes 0...2	Byte 3	Meaning
FxFExx	Bit 0	The bit is always set to "0"
	Bit 1	Depends on the manufacturer
	Bit 2	Battery low
	Bit 3	Depends on the manufacturer
	Bit 4	Depends on the manufacturer
	Bit 5	Depends on the manufacturer
	Bit 6	Depends on the manufacturer
	Bit 7	Depends on the manufacturer

Appendix A

A.1 Internal status parameter

Status variables

Every Ident block has status outputs to allow a suitable reaction in the user program if an error occurs and to simplify error diagnostics on the device. In addition to this, every Ident block has a time stamp and an error memory to be able to better understand previous problems.

These variables are stored in the relevant instance DB of the block.

Table A- 1 Status variables in the instance data block

Name	Data type	Description
statLastErrorStatus	DWORD	This variable contains the last instruction status if an error occurs. This value is always overwritten if a new error occurs with the block.
statLastErrorTimestamp	DTL (S7-1200/-1500) DATE_AND_TIME S7-300/-400)	This variable stores the time stamp of the last error to occur (Last_error_status) with the instruction.

You will find the status variables on the following path: "Instance data block/Ident_Instance/Static/*Name*".

Further status variables exist in the "IID_HW_CONNECT" variable.

Table A- 2 Status variables in "IID_HW_CONNECT"

Name	Data type	Description
STATUS_IN_WORK	BOOL	Command is currently being executed • True = a block or the Ident profile is accessing this channel/reader. • False = the channel/reader is not currently being used.
STATUS_INITIALISATION	BOOL	Reset display • True = a reset is active on this reader/channel. • False = no reset is active on this reader/channel.
LAST_CMD_INIT	BOOL	This bit indicates that the last command to be executed was a reset. • True = last command was reset • False = last command was not reset This bit is reset at the next command start

You will find the status variables on the following path: "IID_HW_CONNECT variable/Static/*Name*".

A.2 Service & Support

Industry Online Support

In addition to the product documentation, the comprehensive online information platform of Siemens Industry Online Support at the following Internet address:
Link 1: (https://support.industry.siemens.com/cs/de/en/)

Apart from news, there you will also find:

- Project information: Manuals, FAQs, downloads, application examples etc.
- Contacts, Technical Forum
- The option submitting a support query:
  link 2: (https://support.industry.siemens.com/My/ww/en/requests)
- Our service offer:

  Right across our products and systems, we provide numerous services that support you in every phase of the life of your machine or system - from planning and implementation to commissioning, through to maintenance and modernization.

You will find contact data on the Internet at the following address:
Link 3: (http://w3.siemens.com/aspa_app)

RFID homepage

For general information about our identification systems, visit RFID home page (http://www.siemens.com/ident/rfid).

Online catalog and ordering system

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

SITRAIN - Training for Industry

The training offer includes more than 300 courses on basic topics, extended knowledge and special knowledge as well as advanced training for individual sectors - available at more than 130 locations. Courses can also be organized individually and held locally at your location.

You will find detailed information on the training curriculum and how to contact our customer consultants at the following Internet address:

Link: (http://sitrain.automation.siemens.com/sitrainworld/)