SIEMENS

SIMATIC Ident

RFID systems
Ident profile, Add-on instruction for Rockwell systems

Function Manual

01/2019

C79000-G8976-C410-03

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

℗ 01/2019 Subject to change

Preface

Warranty and Liability

The application examples are not binding and do not claim to be complete regarding the configuration and equipment and any eventuality. The application examples do not represent customer-specific solutions. They are only intended to provide help for typical applications. You are responsible for ensuring that the described products are used correctly These application examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these application examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these application examples at any time without prior notice. If there are any deviations between the recommendations provided in these application examples and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this application example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act ("Produkthaftungsgesetz"), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract ("wesentliche Vertragspflichten"). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these application examples or excerpts hereof is prohibited without the expressed consent of the Siemens AG.

Table of contents

Introduction

1

Purpose of this document

As the interface to the communication services the ready made program block "Ident_Profile" is available to you for your user program in the environment of Rockwell controllers along with the Ident blocks (add-on instructions). This manual contains descriptions of the blocks with which you can commission and assign parameters for the various Ident systems via the Ethernet/IP interface.

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 for Rockwell systems and describes the delivery status as of January 2019.

The functions mentioned in this documentation are currently supported by the RF680R and RF685R readers and the RFID 181EIP communication module.

Documentation classification

You can find additional information on the blocks and Ident devices named in this manual on the Siemens Industry Online Support (https://support.industry.siemens.com/cs/ww/en/ps/14970/man) pages or on the DVD "Ident Systems" (6GT2080-2AA20) in the respective manuals:

- RFID 181EIP
- SIMATIC RF600 (system manual and configuration manual)

Specifications

The Ident profile block in the manual is 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.

Naming of the functions

The elements typically referred to as "blocks" at Siemens are called "Add-On Instructions" at Rockwell. In the remainder of this manual the typical Rockwell names are used.

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)

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 constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit
Link: (http://www.siemens.com/industrialsecurity)

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customers' 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 profile can be used in Rockwell controllers with an Ethernet/IP interface. The modules can be configured using the "Studio 5000 Logix Designer" or "RSLogix 5000" programs.

Figure 2-1 Configurable module: RFID 181EIP and RF680R/RF685R

2.2 Functions of the instructions

The "Ident_Profile" instruction serves as the communication interface between an Ident device (e.g. RFID 181EIP) and the user program. The instruction supports the following functions:

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

The Ident profile is a single complex instruction containing all the commands and functions for RFID systems and code reader systems. The "Param_RFID181EIP" (or "Param_RF68xR") instruction is used to configure the Ident device during startup.

The size of the data buffers "TXREF" and "RXREF" can be variable. You can specify the length manually.

Configuring with Studio 5000 Logix Designer 3

3.1 Configuring with Studio 5000 Logix Designer

Note

Serial number in Studio 5000 Logix Designer

Note that the serial number specified in the Studio 5000 Logix Designer does not match the device serial number. The serial number specified in the Logix Designer forms the last 4 bytes of the MAC address of the reader / communication module.

Note

Tested programs

The content described in this section was tested with the programs "Studio 5000 Logix Designer" (V21 bis V28) and "RSLogix 5000" (V20).

3.2 Integrating modules into Studio 5000 Logix Designer

The modules must be integrated into the Studio 5000 Logix Designer via an EDS file. The EDS file is contained in the Add-On Instructions library. You can find the latest file on the pages of the Siemens Industry Online Support (https://support.industry.siemens.com/cs/ww/en/ps/14971/dl). With RF68xR readers, the file is also stored on the readers themselves and on the DVD that is supplied with the RF68xR readers.

Procedure

Follow the steps below to link the EDS file of the respective module into Studio 5000 Logix Designer:

1. Copy the installation file (*.eds) locally to your PC.
2. Open the Studio 5000 Logix Designert and change to the project view.
3. Use the menu command "Tools > EDS Hardware Installation Tool"

   the "EDS Hardware Installation Tool" is opened.
4. Follow the instructions of the tool to link the module into Studio 5000 Logix Designer.

Result: Your hardware catalog in Studio 5000 Logix Designer is now updated.

3.3 Creating a Studio 5000 Logix Designer project

The modules can be integrated into Rockwell automation systems using Studio 5000 Logix Designer. The connection is via Ethernet/IP. Following this, you can configure the RF68xR reader using WBM while you control the work with the module using the add-on instructions of Studio 5000 Logix Designer.

Requirement

The module is connected, has started up and an IP address has been assigned to the device. The Studio 5000 Logix Designer was started.

Procedure

Follow the steps below to create a new project:

1. Start the Studio 5000 Logix Designer.
2. Create a new project.

   Select the controller you are using and configure the project properties.
3. Change to the Project view.
4. Right-click on "Ethernet" in the "Controller Organizer > I/O Configuration" area.
5. In the shortcut menu, select the menu command "New Module...".

   The input screen "Select Module Type" is opened.
6. Select the module (e.g. "RF68xR") and click on the "Create" button.

   The input screen "New Module" opens.
7. Enter the name of the device in the "Name" text box.
8. Enter the IP address of the device in the "IP address" text box.
9. Click the "Change" button.

   The input screen "Module Definition" opens.
10. In the "Size" box change the data word size from "SINT" to "INT".
11. Confirm your entry with "OK".
12. Confirm with "OK".

Configuring the instructions

4

4.1 Importing add-on instructions

To program Ident systems using the Studio 5000 Logix Designer, you require add-on instructions. You can find the add-on instructions on the DVD supplied with the RF68xR readers or on the pages of the Siemens Industry Online Support (https://support.industry.siemens.com/cs/ww/en/ps/14971/dl).

Requirement

The Studio 5000 Logix Designer was started and a project was created.

Procedure

Follow the steps below to import the add-on instructions into the Studio 5000 Logix Designer:

1. Copy the "*.l5x" installation file locally to your PC.

   You can find the file on the DVD supplied with the reader or under the provided link.
2. Open the Studio 5000 Logix Designer and change to the project view.
3. Right-click on the folder "Add-on Instructions" in the "Controller Organizer" area.
4. In the shortcut menu, select the menu command "Import Add-On Instruction...".

   The dialog "Import Add-On Instruction" opens.
5. Go to the "*.L5X" file and select it.
6. Confirm your entry with "OK".

Result: The add-on instructions are imported into the Studio 5000 Logix Designer:

4.2 Overview of the add-on instructions

To program the various identification systems, a library with add-on instructions is available.

The following table provides an overview of the currently available add-on instructions.

Table 4- 1 Overview of the add-on instructions

Position			Symbolic name	Description
Add-On Instructions	Ident instructions	Parameter instructions	Param_RF68xR	You can use these instructions to transfer the parameters required after a restart to the reader or communication module.
			Param_RFID181EIP	
		Basic instructions	Read	Using this instruction, you can easily program communication with the Ident systems. The basic instructions include all the instructions that are often used.
			Write	
			Read_MV	
			Set_MV_Program	
		Extended instructions	Config_Download	Using this instruction, you can easily program communication with the Ident systems. The extended instructions 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. Using the "AdvancedCmd" instruction, it is possible to access other commands from the Ident command set and to execute chained commands.
		Status instructions	Reader_Status	Using the status instructions, you obtain information about the reader or transponder.
			Tag_Status	
		RESET instructions	Reset_MV	Using the reset instructions, you can initialize Ident systems.
			Reset_RF68xR	
			Reset_RF200	
			Reset_RF300	
			Reset_RF600	
			Reset_Univ	
	Ident profile		Ident_Profile	These instructions are available for experts to be able to include complex command structures in their own program sequence. It is also possible to use command repetition and chaining.

Position		Symbolic name	Description
Data types	System data types	IID_CMD_STRUCT	Data type for the Ident profile for setting the command parameters.
		IID_DATA_RF68xR	Data type for parameter data for the reader / communication module and data for synchronizing the instructions for all read points of a reader.
		IID_DATA_RFID181EIP	
	Status data types	IID_READSTAT_ 81_RF2_3_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_ 86_RF300	
		IID_READSTAT_ 87_RF600	
		IID_READSTAT_ 88_RF600	
		IID_READSTAT_ 89_RF68xR	
		IID_READSTAT_ A0_A1_RF600	
		IID_TAG_STATUS_ 04_RF300	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_ 82_RF300	
		IID_TAG_STATUS_ 83_ISO	
		IID_TAG_STATUS_ 84_RF600	
		IID_TAG_STATUS_ 85_RF600	

The Ident profile is a single complex instruction that supports all commands and functions for Ident. The Ident instructions represent a simplified interface of the Ident profile. Each Ident instruction contains a single command of the Ident profile.

Note

Parallel operation using Ident instruction and Ident profile is not possible

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

4.3 General structure of the add-on instructions

Form of the instruction as an example

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

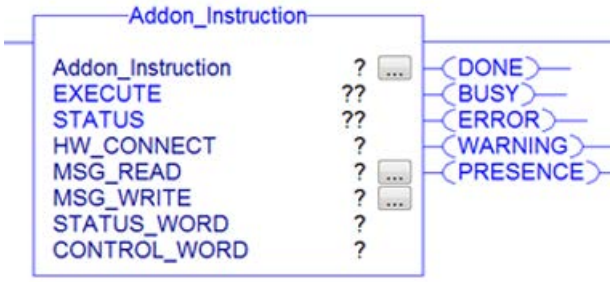


Figure 4-1 Example instructions

Description of the parameters

Table 4- 2 Description of the input and output parameters

Parameter	Description
Input parameters	
EXECUTE	There must be a positive edge at this input before the instruction executes the command.
HW_CONNECT	Global parameter of the type "IID_HW_CONNECT" to address the channel/reader and to synchronize the instructions. This parameter is located in the variables of the type "IID_DATA_*". "HW_CONNECT" must always be transferred to the instructions to address the relevant channel/reader.
MSG_READ	Message variable for communication between controller and CM/reader
MSG_WRITE	Message variable for communication between controller and CM/reader
STATUS_WORD	Cyclic status word which is sent from the CM/reader to the controller.
CONTROL_WORD	Cyclic control word which is sent from the controller to the CM/reader.
Output parameters	
DONE (BOOL)	The job was executed. If the result is positive, this parameter is set.
BUSY (BOOL)	The job is being executed.
ERROR (BOOL)	The job was ended with an error. The error code is indicated in Status.
WARNING (BOOL)	The job was ended with a warning.
STATUS (DINT)	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 instruction is called. This parameter does not occur in specific instructions for optical readers.

4.4 Project preparation

To prepare the project and configure the individual instructions, you must first perform the following steps:

1. Configure the parameter instruction ("Param_*") and address the modules.
2. Assign parameters to the "RESET" instruction.

You can then configure the required instructions accordingly.

Addressing the Ident devices

When working with all instructions, you require a variable of the "IID_HW_CONNECT" data type for addressing the reader / communication modules. The "IID_HW_CONNECT" variable relates to a read point /channel of the reader and is part of the "IID_DATA_*" data type. Setting the command parameters for the Ident profile is handled by the Ident instructions. The Ident profile and the "AdvancedCMD" instruction also require the variable of the "IID_CMD_STRUCT" data type for the configuration of the individual commands. Depending on whether you work with the Ident profile or the Ident instructions, you need to integrate and assign parameters for these data types as described in the following sections.

4.4.1 Configuring instruction

You first need to link the corresponding parameter instruction ("Param_RF68xR" or "Param_RFID181EIP") to the project and create and assign the relevant variables for the parameters "READER_CONFIGDATA" and "MSG_PARAM". Using the parameter instruction, configuration data can be sent to the Ident system.

In addition to this, the instruction resets the required start values of the "IID_HW_CONNECT" variables for the reader channels (read points).

Note that this instruction always needs to be executed first before other instructions can be executed. The instruction must be executed each time the controller or reader/CM is restarted.

Feed for instruction "Param_*"

Proceed as follows to create the "Param_*" instruction:

1. Drag the instruction from the "Controller Organizer" into the project.
2. Create a instance variable.
3. Create a global variable for the "CONFIGDATA" parameter.

   The variable of the data type "IID_DATA_*". The variable is called as an example for the following "RfidDevice01_ConfigData" description.
4. Create a global variable of the "MESSAGE" type at the "MSG_PARAM" parameter.
5. Configure the variable of the parameter "MSG_PARAM" ("...") as follows:

   In the "Configuration" tab:

   – Message Type = CIP generic
   – Service Type = Custom
   – Service Code = 41
   – Class = 80
   – Instance = 1
   – Attribute = 1
   – Source Element = RfidDevice01_ConfigData.Parameter
   – Source Length = 10

   In the "Communication" (only with RF68xR):

   – Large Connection = ✓
6. Select the relevant Ident device for which you want to assign parameters in the "Communication" tab in the text box "Path".
7. If the instruction for the RFID181 EIP communication module has been created, assign the "MOBY_MODE" and "BAUD_RATE" parameters the appropriate values.

   You can find more information on this subject in the section "Param instructions (Page 20)".
8. Start the instruction by calling the "EXECUTE" parameter.

4.4.2 Configuring the reset instruction

In the next step, a reset instruction must be created, configured and executed. You can use a specific reset instruction (such as ""Reset_RF68xR") or the Ident profile with a "WRITE-CONFIG" command to do this. Both procedures require two additional global message variables for the "MSG_READ" and "MSG_WRITE" parameters. These two variables can used with all other instructions.

Before the read point of the reader is operational, the "WRITE-CONFIG" command needs to be executed. This resets the relevant read point.

Feed for instruction "RESET*"

The procedure is described below using a specific "WRITE-CONFIG" instruction. The procedure for using the Ident profile with a "WRITE-CONFIG" command is identical.

Proceed as follows to create the "Reset_*" instruction:

1. Drag the instruction from the "Controller Organizer" into the project.
2. Create a instance variable.
3. Select the variable ("RfidDevice01_ConfigData.CH01" in out example) for the "HW_CONNECT" parameter.
4. Create a global variable for "MSG_READ".
5. Configure the variable of the parameter "MSG_READ" ("...") as follows:

   In the "Configuration" tab:

   - Message Type = CIP generic
   - Service Type = Custom
   - Service Code = 40
   - Class = 80
   - Instance = 1
   - Source Element = RfidDevice01_ConfigData.CH01.Static.buf
   - Destination Element = RfidDevice01_ConfigData.CH01.Static.buf
6. Select the relevant Ident device for which you want to assign parameters in the "Communication" tab in the text box "Path".
7. When using the RF68xR reader:

   Select the "Connected" and "Large Connection" check boxes, if applicable.

   You can find more information on this subject in the section "Param instructions (Page 20)".
8. Create a global variable for "MSG_WRITE".
9. Assign parameters for the variable as follows:

   In the "Configuration" tab:

   - Message Type = CIP generic
   - Service Type = Custom
   - Service Code = 41
   - Class = 80
   - Instance = 1
   - Source Element = RfidDevice01_ConfigData.Static.buf
10. Select the relevant Ident device for which you want to assign parameters in the "Communication" tab in the text box "Path".

11. When using the RF68xR reader:

    If applicable, select the "Connected" and "Large Connection" check boxes if they were activated beforehand in step 7.

12. Select the relevant input word of the reader/CM for the parameter "STATUS_WORD" (<Reader Name>:I.Data[0]).

13. Select the relevant output word of the reader/CM for the parameter "CONTROL_WORD" (<Reader Name>:O.Data[0]).

14. Start the instruction by calling the "EXECUTE" parameter.

4.5 Programming add-on instructions

4.5.1 Param instructions

Using the "Param_RF68xR" or "Param_RFID181EIP" instruction, the configuration data is sent to the Ident system. In addition to this, the instruction resets the required start values of the "HW_CONNECT" variables for the reader channels (read points). Note that this instruction always needs to be executed first before other instructions can be executed. The instruction must be executed each time the controller or module is restarted.

Instruction "Param_RF68xR"

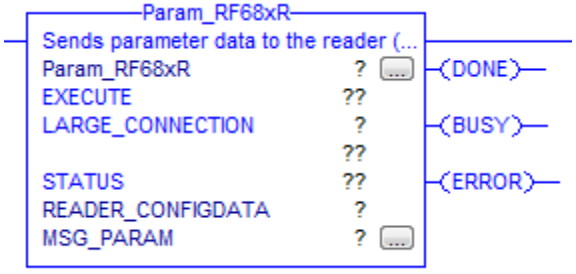


Figure 4-2 Instruction "Param_RF68xR"

The input parameter "LARGE_CONNECTION" is supported only by ControlLogix controllers as of version "5.0". You also require the Logix Designer as of version "21.00.00" and the RSLogix 5000 software as of version "20.00.00".

If the parameter is set (value = "true" or "1"), the maximum frame length is increased from 240 bytes to 1035 bytes. If the parameter is enabled, you must also enable the options "Connected" and "Large Connection" in the parameters "MSG_READ" and "MSG_WRITE" of the message variables.

Instruction "Param_RFID181EIP"

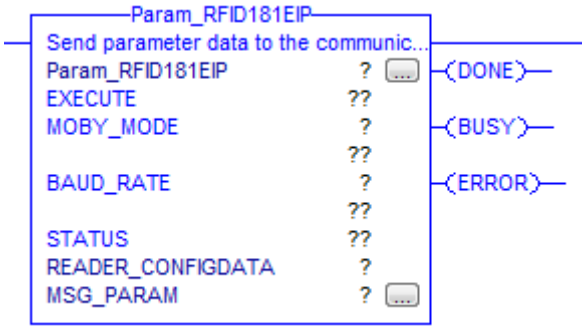


Figure 4-3 Instruction "Param_RFID181EIP"

Table 4- 3 Explanation for the instruction "Param_RFID181EIP"

Parameter	Data type	Default values	Description
MOBY_MODE	SINT	0	MOBY mode for the connected Ident system on the channels of the RFID 181EIP: • 0x05 = RF200/RF300/RF600; MV400/MV500 • 0x06 = RF620R/RF630R (4 bytes UID) • 0x07 = RF620R/RF630R (8 bytes UID)
BAUD_RATE	SINT	0	Transmission speed for serial communication on the RFID 181EIP: • 0x01 = 19.2 kBd • 0x02 = 38.4 kBd • 0x03 = 57.6 kBd • 0x04 = 115.2 kBd

4.5.2 Basic instructions

4.5.2.1 Read

The "Read" instruction 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 instruction reads the data from memory bank 3 (USER area). Specific access to a certain transponder is made with the "EPCID_UID" and "LEN_ID".

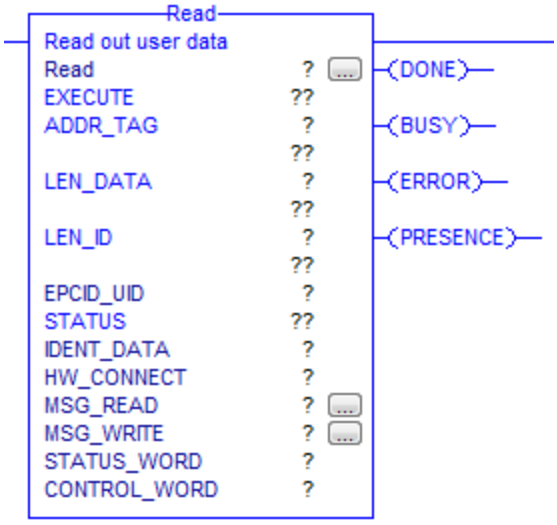


Figure 4-4 Instruction "Read"

Table 4- 4 Explanation for the instruction "Read"

Parameter	Data type	Default values	Description
ADDR_TAG	DINT	0x00	Physical address on the transponder where the read starts. You will find more information on addressing in the section "Transponder addressing (Page 96)". 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	INT	0x00	Length of the data to be read
LEN_ID	SINT	0x00	Length of the EPC-ID/UID Default value: 0x00 ≙ unspecified single tag access (RF200, RF300, RF68xR)

Parameter	Data type	Default values	Description
EPCID_UID	SINT[62]	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	SINT[10]	0	Data buffer in which the read data is stored. Note: When necessary, an array of any size can be transferred to this parameter. Note that the array must be ≥ 10 bytes long (e.g. "SINT[12400]").

[1] You can find additional information on working with optical reader systems in the operating instructions for "SIMATIC MV420 / SIMATIC MV440" and "SIMATIC MV500".

4.5.2.2 Write

The "Write" 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. For RF68xR readers, the instruction writes the data to memory bank 3 (USER area). Specific access to a certain transponder takes place with the optional "EPCID_UID" and "LEN_ID" parameters.

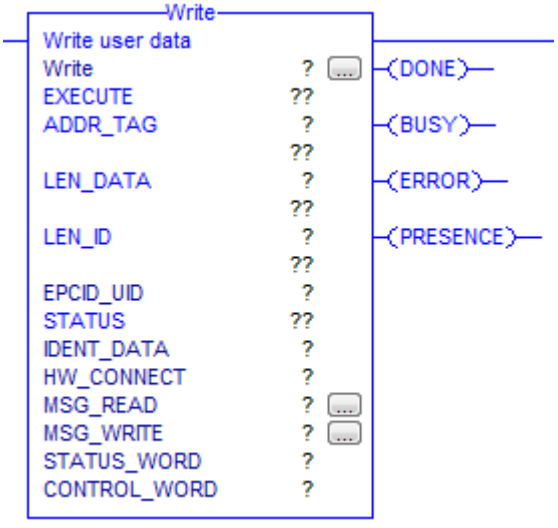


Figure 4-5 Instruction "Write"

Table 4- 5 Explanation for the instruction "Write"

Parameter	Data type	Default values	Description
ADDR_TAG	DINT	0x00	Physical address on the transponder where the write starts. You will find more information on addressing in the section "Transponder addressing (Page 96)". With MV: Address is always 0. [1)]
LEN_DATA	INT	0x00	Length of the data to be written
LEN_ID	SINT	0x00	Length of the EPC-ID/UID Default value: 0x00 ≙ unspecified single tag access (RF200, RF300, RF68xR)
EPCID_UID	SINT[62]	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	SINT[10]	0	Data buffer with the data to be written. With MV: The first byte encodes the corresponding MV command.[1)] Note: When necessary, an array of any size can be transferred to this parameter. Note that the array must be ≥ 10 bytes long (e.g. "SINT[12400]").

[1)] You can find additional information on working with optical reader systems in the operating instructions for "SIMATIC MV420 / SIMATIC MV440" and "SIMATIC MV500".

4.5.2.3 Read_MV

The "Read_MV" instruction reads the read result of an optical reader. The "Read" instruction must be used to read the configuration. The length of the data to be read is calculated automatically by the instruction 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).

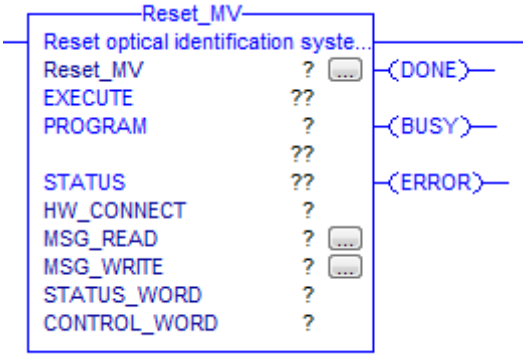


Figure 4-6 Instruction "Read_MV"

Table 4- 6 Explanation for the instruction "Read_MV"

Parameter	Data type	Default values	Description
IDENT_DATA	SINT[10]	0	Read result The length of the read code is located in bytes 0 and 1. Note: When necessary, an array of any size can be transferred to this parameter. Note that the array must be ≥ 10 bytes long (e.g. "SINT[12400]").
LEN_DATA	INT	0x00	Length of the read result ≙ 2 bytes code length + read code

4.5.2.4 Set_MV_Program

Using the "Set_MV_Program" instruction, you can change the program in a camera. The required program number is transferred using the "PROGRAM" parameter.

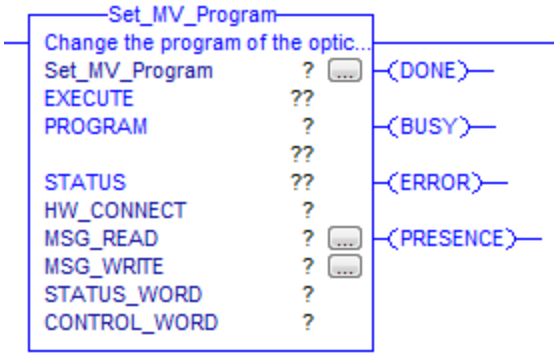


Figure 4-7 Instruction "Set_MV_Program"

Table 4- 7 Explanation for the instruction "Set_MV_Program"

Parameter	Data type	Default values	Description
PROGRAM	SINT	0x01	Program number Range of values: 0x01 ... 0x0F

4.5.3 Extended instructions

4.5.3.1 Config_Upload/-_Download

Using the "Config_Upload" and "Config_Download" instructions, you can read ("Config_Upload") or write ("Config_Download") the configuration of the RFID181 EIP reader and RF68xR reader communications modules connected to this controller.

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 4- 8 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 ID (≙ CONFIG_ID; 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 RF68xR. It is always the same configuration data that is transferred.

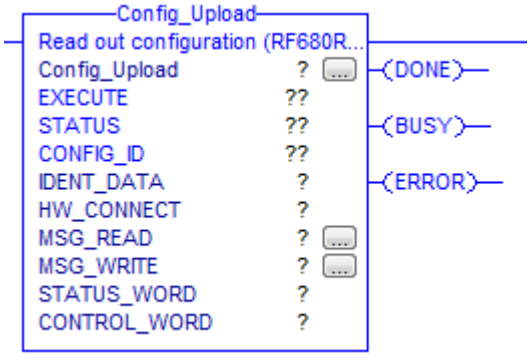


Figure 4-8 Instruction "Config_Upload"

Table 4- 9 Explanation for the instruction "Config_upload"

Parameter	Data type	Default value	Description
IDENT_DATA	SINT[10]	0	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 "IDENT_DATA", we recommend a memory size of 8-16 KB. Note: When necessary, an array of any size can be transferred to this parameter. Note that the array must be ≥ 10 bytes long (e.g. "SINT[12400]").
CONFIG_ID	DINT	--	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.

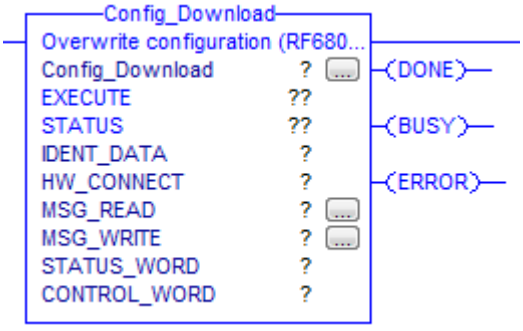


Figure 4-9 Instruction "Config_download"

Table 4- 10 Explanation for the instruction "Config_download"

Parameter	Data type	Default value	Description
IDENT_DATA	SINT[10]	0	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 "IDENT_DATA", we recommend a memory size of 8-16 KB. Note: When necessary, an array of any size can be transferred to this parameter. Note that the array must be ≥ 10 bytes long (e.g. "SINT[12400]").

4.5.3.2 Inventory

The "Inventory" instruction activates the taking of inventories. With a single inventory, acquisition cycles of all antennas with all polarizations are queried. For example, if 2 antennas are connected to a reader and each has 3 polarizations, then an inventory will include 6 acquisition cycles.

With the RF620R and RF630R readers, inventories are always taken as soon as the antenna is turned on.

Special features of the RF68xR reader

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 "0xE7FE0400" (buffer overflow).

The "DURATION" and "DUR_UNIT" parameters are also available for the RF68xR reader. Using the parameters, you can specify the duration of the inventories.

With the 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 instruction 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 having to start the "Inventory" instruction 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.

  "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 having to start the "Inventory" instruction with "EXECUTE".

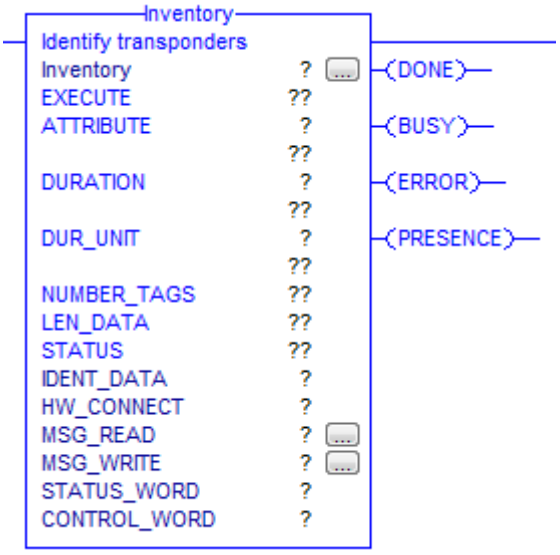


Figure 4-10 Instruction "Inventory"

Table 4- 11 Explanation for the instruction "Inventory"

Parameter	Data type	Default values	Description
ATTRIBUTE	SINT	0x00	Selecting the status mode: • RF620R, RF630R: 0x82 (read out next data record), 0x83, 0x85, 0x90, 0x91, 0x92 • RF68xR: 0x80, 0x81, 0x86, 0x87
DURATION	INT	0x00	RF68xR: period depending on "DUR_UNIT" Period of time or number of inventories or number of "Observed" events Example: • 0x00 ≙ no inventory • 0x01 ≙ one inventory
DUR_UNIT	INT	0x00	RF68xR: unit for "DURATION" • 0x00 ≙ time [ms] • 0x01 ≙ inventories • 0x02 ≙ number of transponders that achieve the "Observed" state
IDENT_DATA	SINT[10]	0	Data buffer for inventory data Note: When necessary, an array of any size can be transferred to this parameter. Note that the array must be ≥ 10 bytes long (e.g. "SINT[12400]").

Parameter	Data type	Default values	Description
NUMBER_TAGS	INT	0	Number of transponders in the antenna field
LEN_DATA	INT	0x00	Length of the valid data During command processing = 0

Result for RF68xR

The number of "TAG_DATA[x]" elements of the data types of the ATTRIBUTE "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 ATTRIBUTE "0x80" and "0x81".

Table 4- 12 ATTRIBUTE "0x80"

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

Table 4- 13 ATTRIBUTE "0x81"

Name	Type	Comment
NUM_MDS	INT	Number of MDS
TAG_DATA	IID_IN_I_81[n]	
TAG_DATA[1]	IID_IN_1_81	
reserved	SINT	
ID_LEN	SINT	EPC length
EPC_ID	SINT[62]	EPC-ID
tagPC	INT	
RSSI	SINT	RSSI value
MaxRSSI	SINT	highest RSSI value
MinRSSI	SINT	lowest RSSI value
channel	SINT	channel; 1..15_ESTI; 1..53:FCC
antenna	SINT	antenna; bit coded; Bit 0 = antenna 1; Bit 1 = antenna 2; Bit 2 = antenna 3; Bit 3 = antenna 4
polarization	SINT	polarizatuin of antenna; 0=undefined; 1=circular; 2=vertical linear; 3=horizontal
time	DINT	timestamp
power	SINT	power in dBm
filterDataAvailable	SINT	0=false; 1=true
Inventoried	INT	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.

Results for RF620R, RF630R

The RF620R/RF630R data types of the ATTRIBUTE "0x82, 0x83, 0x85, 0x90, 0x91, 0x92" are not included in the library of the add-on instructions. If required, you can create these data types. You can find information on this in the manual "Ident profile and Ident blocks, standard function for Ident systems (https://support.industry.siemens.com/cs/ww/en/ps/14971/man)".

4.5.3.3 Read_EPC_Mem

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

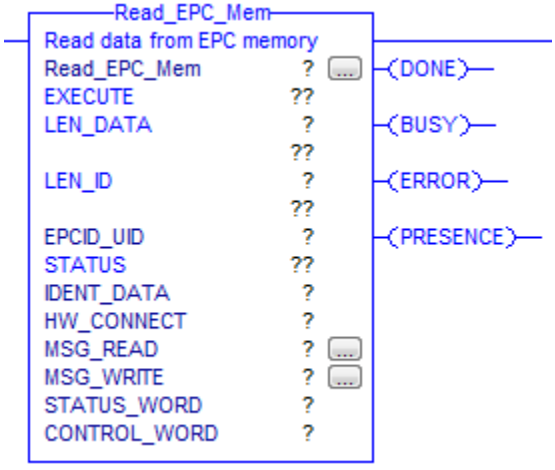


Figure 4-11 Instruction "Read_EPC_Mem"

Table 4- 14 Explanation for the instruction "Read_EPC_Mem"

Parameter	Data type	Default values	Description
LEN_DATA	INT	0x00	Length of the EPC memory to be read out (1 ... 62 bytes)
LEN_ID	SINT	0x00	Length of the EPC-ID/UID Default value: 0x00 ≙ unspecified single tag access (RF68xR)
EPCID_UID	SINT[62]	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	SINT[10]	0	Data buffer in which the read EPC memory data is stored. Note: For Variant, currently only a "SINT" with a variable length can be created. For Any, other data types/UDTs can also be created.

4.5.3.4 Read_TID

The "Read_TID" instruction 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.

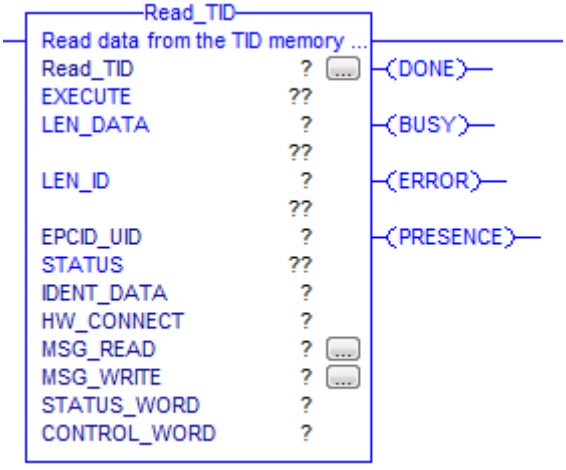


Figure 4-12 Instruction "Read_TID"

Table 4- 15 Explanation for the instruction "Read_TID"

Parameter	Data type	Default values	Description
LEN_DATA	INT	0x00	Length of the EPC memory to be read out (1 ... 62 bytes)
LEN_ID	SINT	0x00	Length of the EPC-ID/UID Default value: 0x00 ≙ unspecified single tag access (RF68xR)
EPCID_UID	SINT[62]	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	SINT[10]	0	Read TID Note: When necessary, an array of any size can be transferred to this parameter. Note that the array must be ≥ 10 bytes long (e.g. "SINT[12400]").

4.5.3.5 Read_UID

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

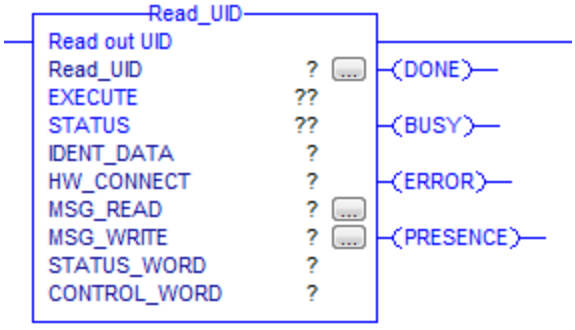


Figure 4-13 Instruction "Read_UID"

Table 4- 16 Explanation for the instruction "Read_UID"

Parameter	Data type	Default value	Description
IDENT_DATA	SINT[10]	0	UID Note: When necessary, an array of any size can be transferred to this parameter. Note that the array must be ≥ 10 bytes long (e.g. "SINT[12400]").

4.5.3.6 Set_Ant

Using the "Set_Ant" instruction, you can switch antennas on or off. There are different instructions for RF300 and RF600. The "Set_Ant_RF300" instruction can also be used for RF200. The "Set_Ant_RF600" instruction relates only to the RF620R and RF630R readers.

Set_Ant_RF300

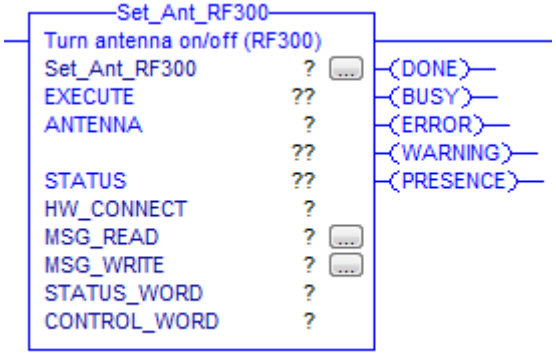


Figure 4-14 Instruction "Set_Ant_RF300"

Table 4- 17 Explanation for the instruction "Set_Ant_RF300"

Parameter	Data type	Default values	Description
ANTENNA	BOOL		0 = turn antenna off 1 = turn antenna on

Set_Ant_RF600

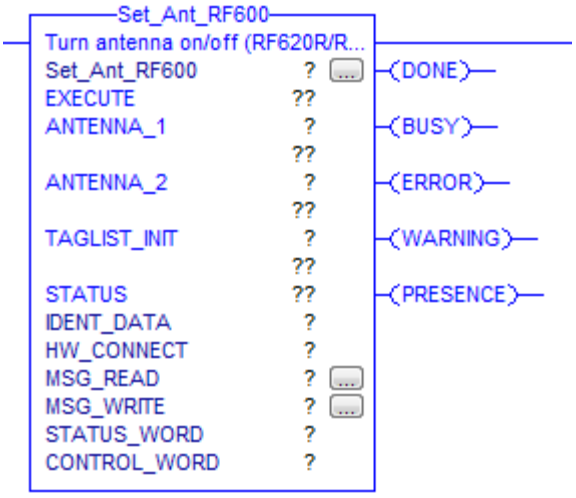


Figure 4-15 Instruction "Set_Ant_RF600"

Table 4- 18 Explanation for the instruction "Set_Ant_RF600" (RF620R/RF630R)

Parameter	Data type	Default values	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

4.5.3.7 Set_Param

Using the "Set_Param" instruction, you can change UHF parameters during runtime (e.g. the antenna power).

Note

Settings saved only temporarily

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

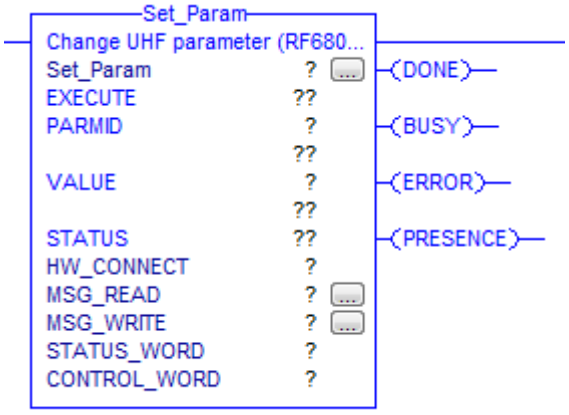


Figure 4-16 Instruction "Set_Param"

Table 4- 19 Explanation for the instruction "Set_Param"

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

All parameters listed in the table below are valid for RF68xR.

Table 4- 20 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	

PARMID (hex)	PARMID (ASCII)	Parameter	VALUE
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/-62). • 32: Tx: 40 Kbps / Rx: 40 Kbps / FM0 • 33: Tx: 40 Kbps / Rx: 62:5 Kbps / Miller4 • 34: Tx: 40 Kbps / Rx: 75 Kbps / Miller4 • 35: Tx: 80 Kbps / Rx: 62:5 Kbps / Miller4 • 37: Tx: 80 Kbps / Rx: 400 Kbps / FM0 • 65: Tx: 40 Kbps / Rx: 40 Kbps / ISO 18000-62
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; Set the internal reader clock, whereby 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 01/01/1990; Setting of the internal reader clock, whereby only the date is changed and not the time.

4.5.3.8 Write_EPC_ID

The "Write_EPC_ID" instruction 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 instruction 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.

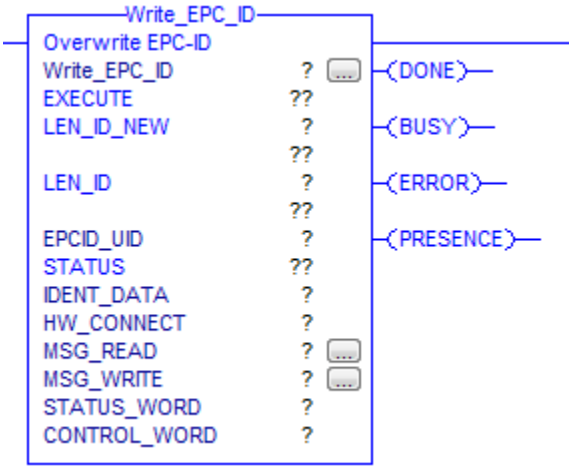


Figure 4-17 Instruction "Write_EPC_ID"

Table 4- 21 Explanation for the instruction "Write_EPC_ID"

Parameter	Data type	Default values	Description
LEN_ID_NEW	SINT	0x00	Length of the current EPC-ID
LEN_ID	SINT	0x00	Length of the previous EPC-ID
EPCID_UID	SINT[62]	0	Previous EPC ID
IDENT_DATA	SINT[10]	0	Current EPC ID Note: When necessary, an array of any size can be transferred to this parameter. Note that the array must be ≥ 10 bytes long (e.g. "SINT[12400]").

4.5.3.9 Write_EPC_Mem

The "Write_EPC_Mem" instruction 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.

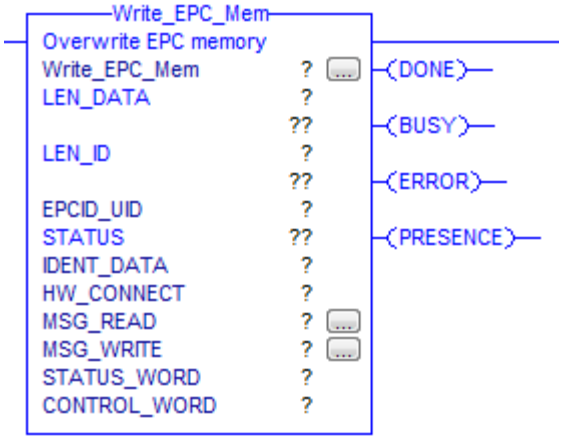


Figure 4-18 Instruction "Write_EPC_Mem"

Table 4- 22 Explanation for the instruction "Write_EPC_Mem"

Parameter	Data type	Default values	Description
LEN_DATA	INT	0x00	Length of the EPC memory to be overwritten (1 ... 62 bytes)
LEN_ID	SINT	0x00	Length of the EPC-ID/UID Default value: 0x00 ≙ unspecified single tag access (RF68xR)
EPCID_UID	SINT[62]	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	SINT[10]	0	Data buffer with the EPC memory data to be overwritten. Note: When necessary, an array of any size can be transferred to this parameter. Note that the array must be ≥ 10 bytes long (e.g. "SINT[12400]").

4.5.3.10 AdvancedCMD

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

This instruction gives you the option of sending the command as a chained command. The instruction provides a CMD buffer for 10 commands for this. 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 93)".

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

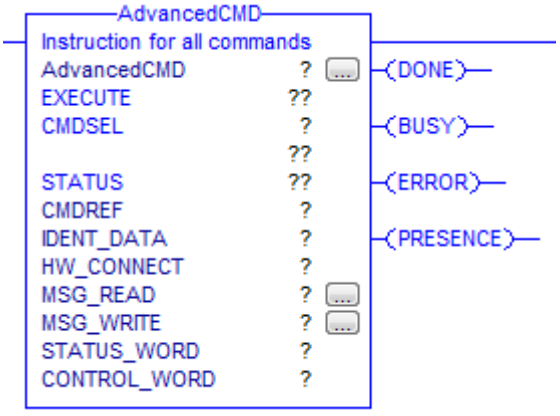


Figure 4-19 Instruction "AdvancedCMD"

Table 4- 23 Explanation for the instruction "AdvancedCMD"

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	IID_CMD_STRUCT [10]	--	You will find a detailed description of the parameter in the sections: • "Overview of the commands (Page 66)" • "Command structure (Page 68)"
IDENT_DATA	SINT[10]	0	Buffer for data to be written or read. Note: When necessary, an array of any size can be transferred to this parameter. Note that the array must be ≥ 10 bytes long (e.g. "SINT[12400]").

4.5.4 Status instructions

4.5.4.1 Reader_Status

The "Reader_Status" instruction reads status information from the reader or communications module ("CM configuration_1" module) For the various reader families, there are different status modes that you can select using the "ATTRIBUTE" parameter.

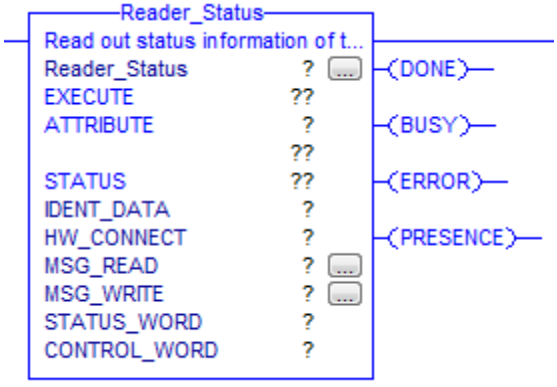


Figure 4-20 Instruction "Reader_Status"

Table 4- 24 Explanation for the instruction "Reader_status"

Parameter	Data type	Default values	Description
ATTRIBUTE	SINT	0x81	Identifier of the status modes / possible entries: • RF200: 0x81 • RF300: 0x81, 0x86 • RF620R, RF630R: 0x87, 0x88, 0xA0, 0xA1 • RF68xR: 0x89
IDENT_DATA	SINT[10]	0	Event values depending on attributes Note: When necessary, an array of any size can be transferred to this parameter. Note that the array must be ≥ 10 bytes long (e.g. "SINT[12400]").

Results

Apply the correct data type that is assigned to the ATTRIBUTE value at the "IDENT_DATA" input of the instruction so that the data can be correctly interpreted. Only an Array of Byte can be created at the "Ident Data" parameter . The conversion to the corresponding data type must be done manually.

Table 4- 25 ATTRIBUTE "0x81" (data type "IID_READSTAT_81_RF2_3_U")

Name	Type	Comment
status_info	SINT	SLG status mode
hardware	SINT	Type of hardware
hardware_version	INT	Version of hardware
loader_version	INT	Version of loader
firmware	SINT	Type of firmware
firmware_version_HB	SINT	Version of firmware
firmware_version_LB	SINT	
driver	SINT	Type of driver
driver_version	INT	Version of driver
interface	SINT	Type of interface (RS 232/RS 422)
baud	SINT	Baudrate
reserved1	SINT	Reserved
reserved2	SINT	Reserved
reserved3	SINT	Reserved
distance_limiting_SLG	SINT	Distance limiting of SLG
multitag_SLG	SINT	Multitag SLG
field_ON_control_SLG	SINT	Field ON control
field_ON_time_SLG	SINT	Field On time
sync_SLG	SINT	Synchronization with SLG
status_ant	SINT	Status of antenne
stand_by	SINT	Time of standby after command
MDS_control	SINT	Presence mode

Table 4- 26 ATTRIBUTE "0x86" (data type "IID_READSTAT_86_RF300")

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

Table 4- 27 ATTRIBUTE "0x89" (data type "IID_READSTAT_89_RF68xR") compatible with RF68xR

Name	Type	Comment
status info	SINT	SLG status mode(Subcommand)
hardware version	SINT	Version of hardware
firmware version	SINT[4]	Version of firmware
config ID	DINT	Unix timestamp
inventory_status	INT	0=inventory not active; 1=inventory active; 2=presence mode active
sum of filtered tags	INT	All filtered Tags
filtered smoothing	INT	Filtered Tags trough Smoothing
filtered blacklist	INT	Filtered Tags trough Blacklist
filtered data-filter	INT	Filtered Tags trough Data-Filter
filtered RSSI threshold	INT	Filtered Tags trough RSSI Threshold
filtered RSSI delta	INT	Filtered Tags trough RSSI Delta

Table 4- 28 ATTRIBUTE "0x87" (data type "IID_READSTAT_87_RF600")

Name	Type	Comment
status info	SINT	SLG status mode
hardware	SINT	Type of hardware
hardware version	INT	Version of hardware
reserved0	INT	
firmware	SINT	Type of firmware
firmware version HB	SINT	Version of firmware highbyte
firmware version LB	SINT	Version of firmware lowbyte
driver	SINT	Type of driver
current_time_hour	SINT	Hours [1]
current time min	SINT	Minutes
current time sec	SINT	Seconds
reserved1	SINT	
SLG version	SINT	SLG version
baud	SINT	Baudrate
reserved2	SINT	
distance limiting SLG	SINT	Selected transmit power
multitag SLG	SINT	Multitag SLG
field ON control SLG	SINT	Selected comunication typ
field ON time SLG	SINT	Selected channel
expert mode	SINT	Expert mode
status_ant	SINT	Status of antenna [2]
scanning_time_SLG	SINT	Radio communication profile (country specific radio standart)
MDS control	SINT	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"/"WRITE-CONFIG". With "INIT"/"WRITE-CONFIG" of the RF620R, the default value is "1" (int. antenna on), with the RF630R, it is "3" (antennas 1 and 2 on).

Table 4- 29 ATTRIBUTE "0x88" (data type "IID_READSTAT_88_RF600")

Name	Type	Comment
status info	SINT	SLG status mode(Subcommand)
hardware	SINT	Type of hardware
hardware version	INT	Version of hardware
reserved word1	INT	Reserved
firmware	SINT	Type of firmware
firmware version HB	SINT	Version of firmware (High-Byte)
firmware version LB	SINT	Version of firmware (Low-Byte)
driver	SINT	Type of driver
current_time_hour	SINT	Hours 1)
current_time_minute	SINT	Minutes 1)
current_time_sec	SINT	Seconds 1)
current_time_reservByte	SINT	
SLG version	SINT	SLG-Version
baud	SINT	Baudrate
reserved byte1	SINT	Reserved
distance limiting SLG	SINT	Selected transmit power
multitag SLG	SINT	Multitag SLG
field ON control SLG	SINT	Selected communication type
field ON time SLG	SINT	Selected channel
expert mode	SINT	Expert mode
status_ant	SINT	Status of antenna 2)
scanning_time_SLG	SINT	Radio communication profile (country specific radio standart)
MDS control	SINT	Presence mode
blink pattern	SINT	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	INT	Reserved
reserved word3	INT	Reserved
reserved word4	INT	Reserved
filtered max rssi	SINT	Maximum RSSI value of a tag, of all filtered tags
reserved byte2	SINT	Reserved
filtered tags rssi	SINT	Number of tags, filtered out by the RSSI threshold
reserved byte3	SINT	Reserved
filtered tags black list	INT	Number of tags, filtered out via Black-List
filtered tags epc data	INT	Number of tags, filtered out via EPC Data Filter
filtered tags smoothing	INT	Number of tags in Tag List of status Not-Observed
itf ph1 max detect	INT	Number of reads of a Tag, filtered out via ITF-phase 1
itf ph1 tags detect	INT	Number of tags, filtered out via ITF-phase 1
itf ph2 max detect	INT	Number of reads of a Tag, filtered out via ITF-phase 2
itf ph2 tags detect	INT	Number of tags, filtered out via ITF-phase 2
filtered_istm_min_dist	INT	Minimum distance of tags according to sorting criterion of ISTM
filtered istm tags	INT	Number of tags, filtered out via ISTM algorithm
last error	SINT	error code of the last occuring error (last command)

Name	Type	Comment
reserved_byte4	SINT	Reserved
error_command1	INT	Last command (has lead to error code) "last error"
error_command2	INT	Last command (has lead to error code) "last error"
error_command3	INT	Last command (has lead to error code) "last error"
reserved_word5	INT	Reserved
reserved_array_byte	SINT[30]	

Table 4- 30 ATTRIBUTE "0xA0" and "0xA1" (data type "IID_READSTAT_A0_A1_RF600")

Name	Type	Comment
reserved	SINT	
Status_info	SINT	Status-Info, SLG-Status SubCommand 20/21
number_tags_frame	SINT	Number of Tags in this frame
number_tags_next_frames	SINT	Number of Tags in the next frames
reserved_byte1	SINT	Reserved
reserved_byte2	SINT	Reserved
reserved_byte3	SINT	Reserved
reserved_byte4	SINT	Reserved
reserved_byte5	SINT	Reserved
reserved_byte6	SINT	Reserved
Black_List_ID	IID_IN_Blacklist[13]	EPC-ID Length
Black_List_ID[1]	IID_IN_Blacklist	
EPC_Length	SINT	EPC-ID Length
Antenna	SINT	Antenna = Default 3
Filtered_Tag	INT	Number of times - EPC-ID filtered out via BlackList
EPC	SINT[12]	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

4.5.4.2 Tag_Status

The "Tag_Status" 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.

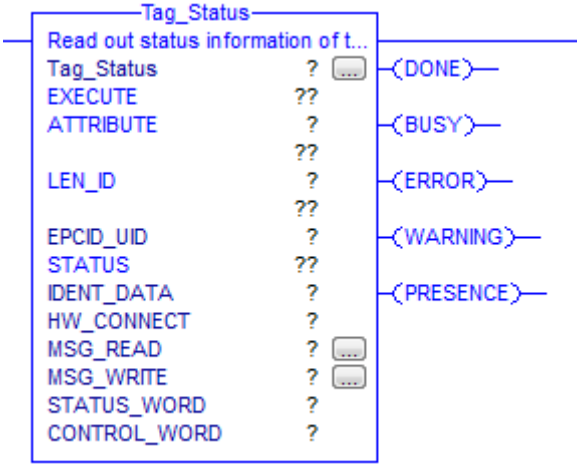


Figure 4-21 Instruction "Tag_Status"

Table 4- 31 Explanation for the instruction "Tag_Status"

Parameter	Data type	Default values	Description
ATTRIBUTE	SINT	0x00	Identifier of the status modes / possible entries: • RF200: 0x83 • RF300: 0x04, 0x82, 0x83 [1)] • RF620R, RF630R: 0x84, 0x85
LEN_ID	SINT	0x00	Length of the EPC-ID/UID Default value: 0x00 ≙ unspecified single tag access (RF68xR)
EPCID_UID	SINT[62]	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	SINT[10]	0	Event values depending on attributes Note: When necessary, an array of any size can be transferred to this parameter. Note that the array must be ≥ 10 bytes long (e.g. "SINT[12400]").

[1)] With RF300 first generation with the attribute "0x83" the values can only be queried with ISO transponders. With RF300 second generation with this attribute the values can be queried with all transponder types. In "General Mode" only this attribute functions.

Results

Table 4- 32 ATTRIBUTE "0x04" ("IID_TAG_STATUS_04_RF300" data type)

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

Table 4- 33 ATTRIBUTE "0x82" ("IID_TAG_STATUS_82_RF300" data type)

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

Table 4- 34 ATTRIBUTE "0x83" ("IID_TAG_STATUS_83_ISO" data type)

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

Table 4- 35 ATTRIBUTE "0x84" ("IID_TAG_STATUS_84_RF600" data type)

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

[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 4- 36 ATTRIBUTE "0x85" ("IID_TAG_STATUS_85_RF600" data type)

Name	Type	Comment
status_info	SINT	MDS status mode
antenna	SINT	Antenna which has observed the MDS
channel	SINT	Channel
UID	SINT[8]	
DT_glimpsed_1	SINT	Time elasped between acknowledgement and first read in [ms]1 Highbyte
DT_glimpsed_2	SINT	Time elasped between acknowledgement and first read in [ms]2
DT_glimpsed_3	SINT	Time elasped between acknowledgement and first read in [ms]3
DT_glimpsed_4	SINT	Time elasped between acknowledgement and first read in [ms]4 Low-Byte
reserved1	SINT	
reserved2	SINT	
reserved3	SINT	
reserved4	SINT	
last_observed_hour	SINT	Last observed time hour
last_observed_min	SINT	Last observed time minutes [1)]
last_observed_sec	SINT	Last observed time seconds [1)]
last observed EPC length	SINT	Last observed time EPC length
EPC ID Byte	SINT[62]	EPC-ID
reads HB	SINT	Number of Reads of MDS in Inventory (1 - 65535)
reads LB	SINT	Number of Reads of MDS in Inventory (1 - 65535)
RSSI	SINT	Current RSSI value of MDS [2)]
mean RSSI	SINT	Mean RSSI value of MDS
max RSSI	SINT	Max RSSI value of MDS
min RSSI	SINT	Min RSSI value of MDS
min POWER	SINT	Min Power value of MDS
current_POWER	SINT	Current Power value of MDS [3)]
reserved5	SINT[137]	

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 "Reads" value 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

4.5.5 RESET instructions

The reset instructions described in this section are required if you want to operate the RF68xR readers or the RFID181 EIP communication modules with a Rockwell controller.

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

4.5.5.1 Reset_RF68xR

Using the "Reset_RF68xR" instruction, you can reset the RF680R and RF685R read points. Note that a separate reset command must be sent for each read point. The read point is reset to the settings configured on the reader with the WBM. The "Reset_RF68xR" instruction does not have any device-specific parameters and is executed using the "EXECUTE" parameter.

Using the "Reset_RF68xR" instruction, you can interrupt any active Ident instruction at any time. The instructions are then ended with "DONE = true" and "ERROR = false".

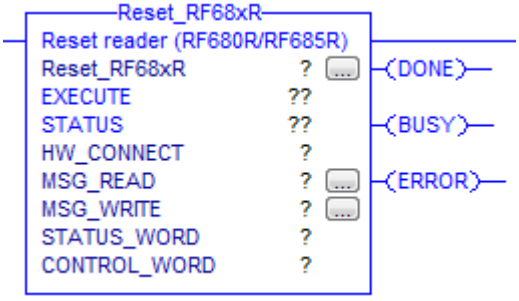


Figure 4-22 Instruction "Reset_RF68xR"

4.5.5.2 Reset_MV

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

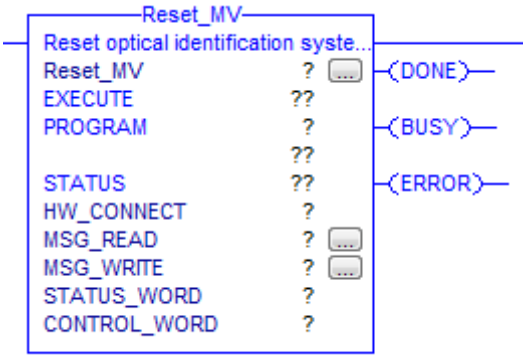


Figure 4-23 Instruction "Reset_MV"

Table 4- 37 Explanation for the instruction "Reset_MV"

Parameter	Data type	Default values	Description
PROGRAM	SINT		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 MV400)

4.5.5.3 Reset_RF200

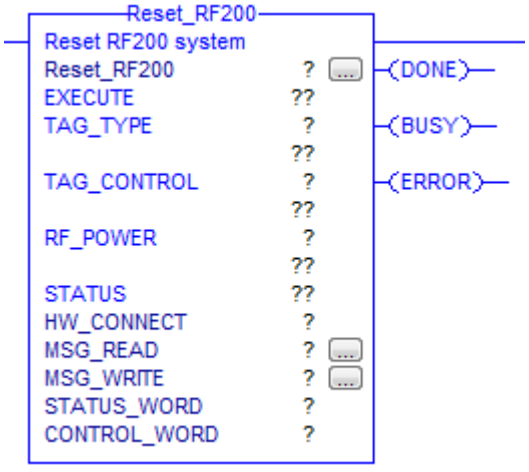


Figure 4-24 Instruction "Reset_RF200"

Table 4- 38 Explanation for the instruction "preset_RF200"

Parameter	Data type	Default values	Description
TAG_CONTROL	SINT	1	Presence check
TAG_TYPE	SINT	1	Transponder type: • 1 = every ISO transponder • 3 = MDS D3xx - optimization
RF_POWER	SINT	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.

4.5.5.4 Reset_RF300

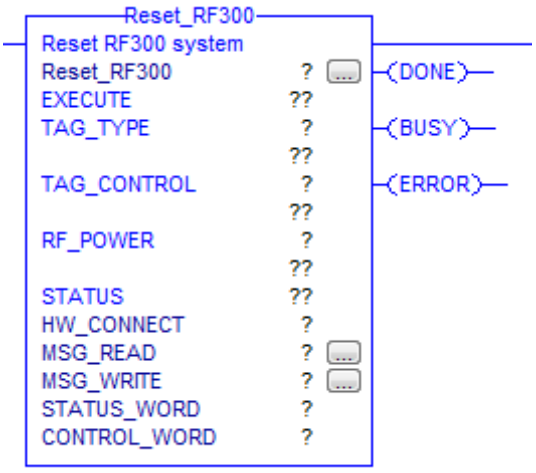


Figure 4-25 Instruction "Reset_RF300"

Table 4- 39 Explanation for the instruction "preset_RF300"

Parameter	Data type	Default values	Description
TAG_CONTROL	SINT	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	SINT	0	Transponder type: • 1 = every ISO transponder • 0 = RF300 transponder
RF_POWER	SINT	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. This setting is not necessary with the RF380R readers of the 2nd generation (6GT2801-3BAx0) because the power limits are optimized automatically depending on the reader-transponder distance.

4.5.5.5 Reset_RF600

Note that this reset instruction is only compatible with the RF620R and RF630R readers.

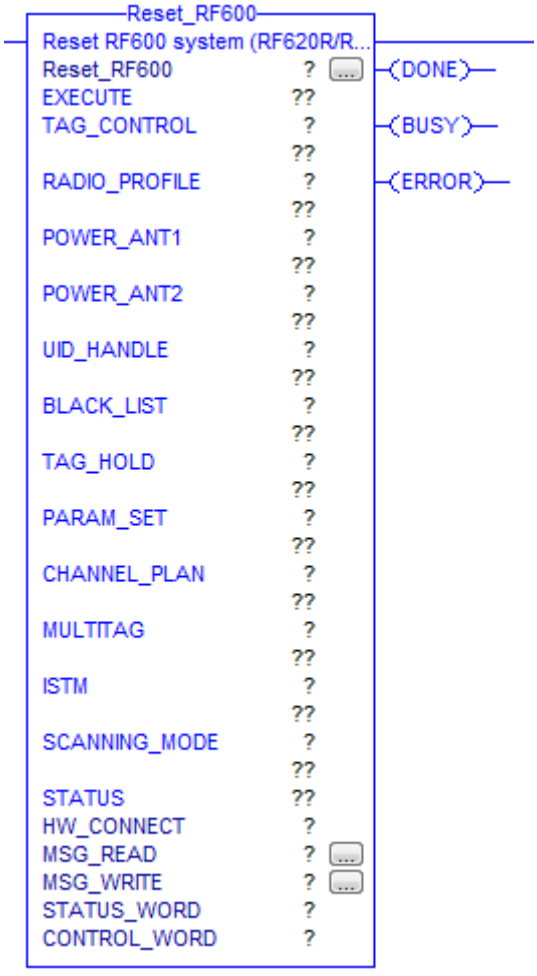


Figure 4-26 Instruction "preset_RF600" (RF620R/RF630R)

Table 4- 40 Explanation for the instruction "preset_RF600" (RF620R/RF630R)

Parameter	Data type	Default values	Description
TAG_CONTROL	BOOL	True	Presence check
RADIO_PROFILE	SINT	0x01	Scanning time: Wireless profile according to EPC Global (range of values: 0x01 ... 0x09 depending on the reader variant)
POWER_ANT1	SINT	0x00	Transmit power for antenna 1 or internal antenna (range of values: 0x00 ... 0x0F)
POWER_ANT2	SINT	0x00	Transmit power for antenna 2 or external antenna (range of values: 0x00 ... 0x0F)

Parameter	Data type	Default values	Description
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
PARAM_SET	SINT	0x00	Field_ON_Control (0 = fast; value range: 0x00, 0x02)
CHANNEL_PLAN	SINT	0x0F	Field_ON_Time (value range: 0x00 ... 0x0F; ETSI only)
MULTITAG	SINT	0x01	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.

4.5.5.6 Reset_Univ

The "Reset_Univ" instruction is a universal reset statement that can be used to reset identification systems. Use this instruction only in consultation with Support.

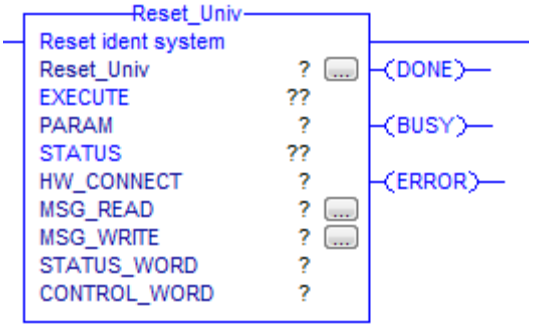


Figure 4-27 Instruction "RESET_Univ"

Table 4- 41 Explanation for the instruction "RESET_Univ"

Parameter	Data type	Default values	Description
PARAM	SINT [16]		Data for reset frame The data to be set here can be made available by Support when necessary for special settings.

Table 4- 42 Structure of the "PARAM" parameter

Byte	1	2...5	6	7...8	9	10	11	12	13...14	15	16
Value	0x04	0x00	0x0A	0x00	scan-ning_ time	param	option_ 1	dis-tance_ limiting	No. of tran-spond-ers	field_ on_ control	field_ on_ time

4.6 Programming the Ident profile

4.6.1 Structure of the Ident profile

Note

Parallel operation using Ident instructions and Ident profile is not possible

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

The instructions described in the section"Programming add-on instructions (Page 20)" represent a simplified interface of the Ident profile. If the functionality available with the instructions 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 instruction containing all the functionality of the Ident instructions. The Ident profile was developed specially for trained users who want to configure complex functions with a single instruction. For untrained users, we recommend using the Ident instructions.

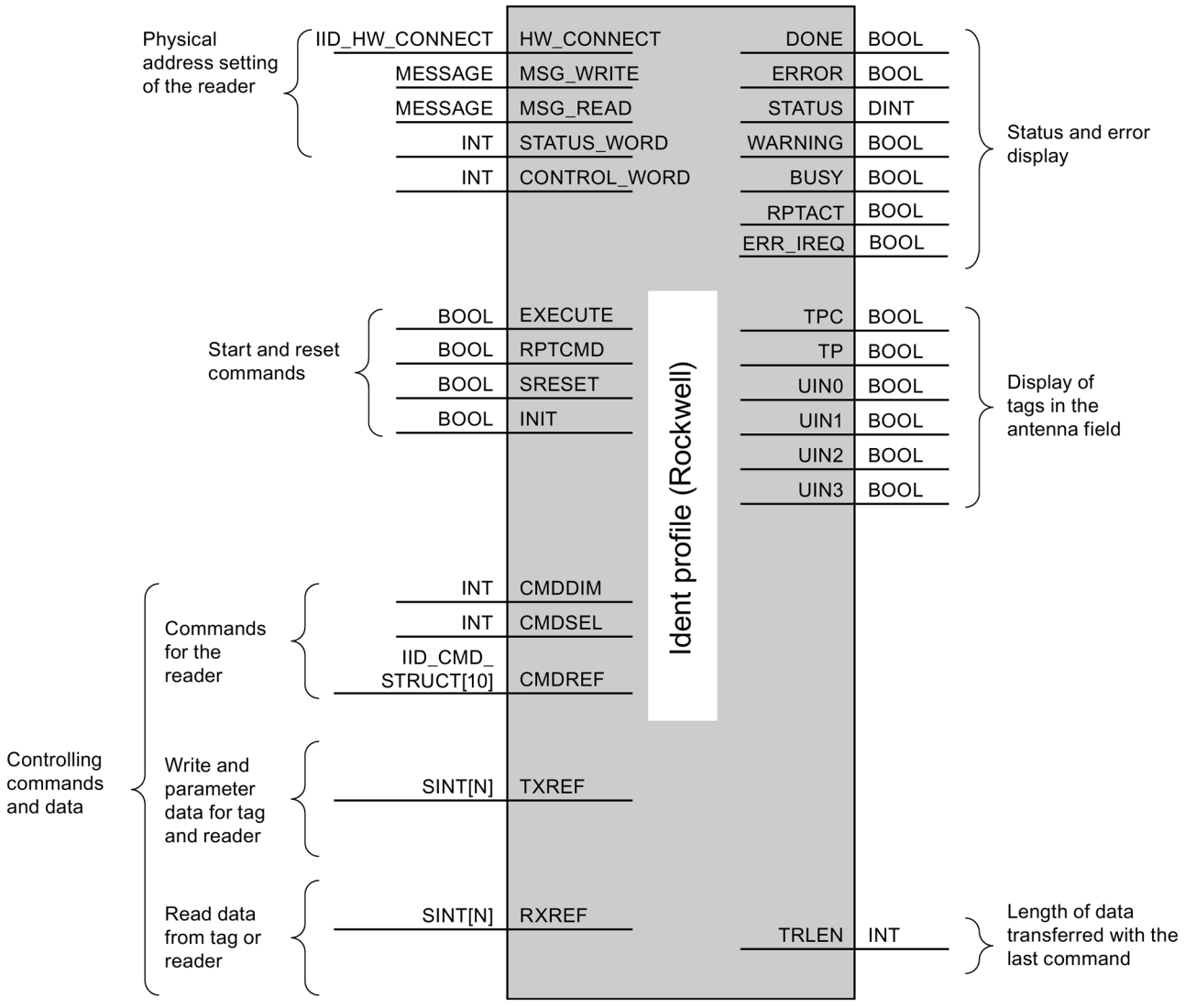


Figure 4-28 The input parameters of the Ident profile

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 4- 43 Input parameter

Input parameter	Data type	Default value	Meaning
HW_CONNECT	HW_CONNECT	--	Global parameter of type "IID_HW_CONNECT" to address the communication module and reader and to synchronize the blocks. This parameter is located in the variables of the type "IID_DATA_ *". "HW_CONNECT" must always be transferred to the instruction to address the relevant channel/reader.
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 restart 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	IID_CMD_STRUCT[n]	--	Command field The field can hold up to 10 commands. The commands are complex variables of the type "CMD_STRUCT". You can find more information on "CMDREF" in the section "Commands of the Ident profile (Page 66)".
TXREF	SINT[n]	--	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 10-32 KB in size.
RXREF	SINT[n]	--	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 10-32 KB in size.

Table 4- 44 Output parameter

Output parameter	Data type	Default value	Meaning
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	DINT	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 103)".
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 acknowledgment 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. on startup of the reader, connection interruption or antenna errors with RF68xR or configuration changes)
TPC	BOOL	FALSE	Transponder Presence Changed (only when 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.

Output parameter	Data type	Default value	Meaning
UIN0	BOOL	FALSE	With RFID readers, the number of transponders in the antenna field is indicated. UIN0 ... UIN3 can be interpreted as binary values. Example: UIN3 = 0; UIN2 = 0; UIN1 = 1; UIN0 = 1 ⇒ 3 transponders With optical reader devices, the various states of the reader are displayed. • UIN0: Corresponds to "IN_OP" bit of the reader • 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 = 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.

4.6.2 Assigning parameters for the instruction

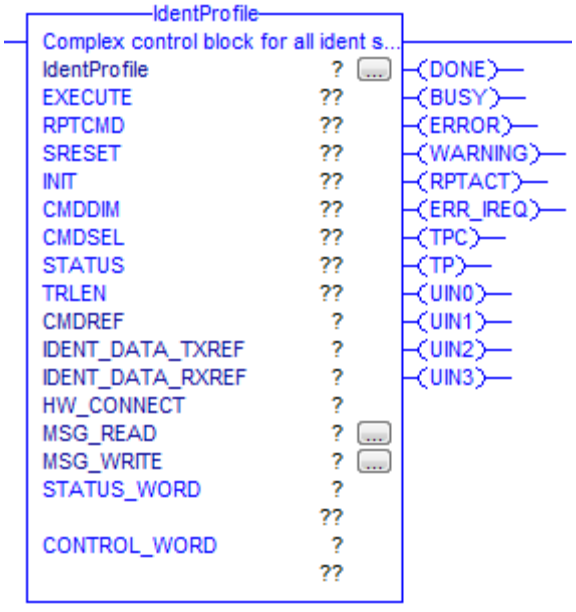


Figure 4-29 Instruction "IDENT_PROFIL"

You must assign the following parameters of the instruction:

- IdentProfile
- CMDREF
- IDENT_DATA_TXREF

- IDENT_DATA_RXREF
- HW_CONNECT
- MSG_READ
- MSG_WRITE
- STATUS_WORD
- CONTROL_WORD

Example of parameter assignment

IdentProfile

Create a instance variable.

CMDREF

Assign a variable to the "CMDREF" parameter. This variable has the "IID_CMD_STRUCT[10]" data type.

HW_CONNECT

Assign a variable to the "HW_CONNECT" parameter. This variable is part of the variables of the "IID_DATA_*" type.

IDENT_DATA_TXREF

Assign a variable to the "IDENT_DATA_RXREF" parameter. This variable has the "SINT[n]" data type.

IDENT_DATA_RXREF

Assign a variable to the "IDENT_DATA_RXREF" parameter. This variable has the "SINT[n]" data type.

STATUS_WORD and CONTROL_WORD

Assign the cyclic input word to the STATUS_WORD parameter and the cyclic output word for the channel to the CONTROL_WORD parameter. Per physical module, variables are automatically created for the cyclic words.

The "I" stands for input and the "O" for output as shown in the following figure. These are assigned as follows:

- Input for STATUS_WORD: ...I.Data...
- Output for CONTROL_WORD: ...O.Data...

The element numbers are assigned as follows:

- Element 0 for channel 1: ...Data[0]
- Element 1 for channel 2: ...Data[1]

RFID_181EIP_01:C	{...}	{...}		AB:ETHERNET_MODULE:C:0
RFID_181EIP_01:I	{...}	{...}		AB:ETHERNET_MODULE_INT_4Bytes:I:0
RFID_181EIP_01:I.Data	{...}	{...}	Decimal	INT[2]
RFID_181EIP_01:I.Data[0]	0		Decimal	INT
RFID_181EIP_01:I.Data[1]	0		Decimal	INT
RFID_181EIP_01:O	{...}	{...}		AB:ETHERNET_MODULE_INT_4Bytes:O:0
RFID_181EIP_01:O.Data	{...}	{...}	Decimal	INT[2]
RFID_181EIP_01:O.Data[0]	0		Decimal	INT
RFID_181EIP_01:O.Data[1]	0		Decimal	INT

Figure 4-30 Automatically created input and output variables

MSG_WRITE and MSG_READ

Assign a "MESSAGE" variable to the parameters "MSG_WRITE" and "MSG_READ".

Configure the parameters as described in the section "RESET instructions (Page 53)".

4.6.3 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 58)".

The relationship between the three "IN/OUT" parameters is described in greater detail below:

- CMDREF (command buffer):

  IID_CMD_STRUCT[10]

- TXREF (send buffer):

  SINT[n]

- RXREF (receive buffer):

  SINT[n]

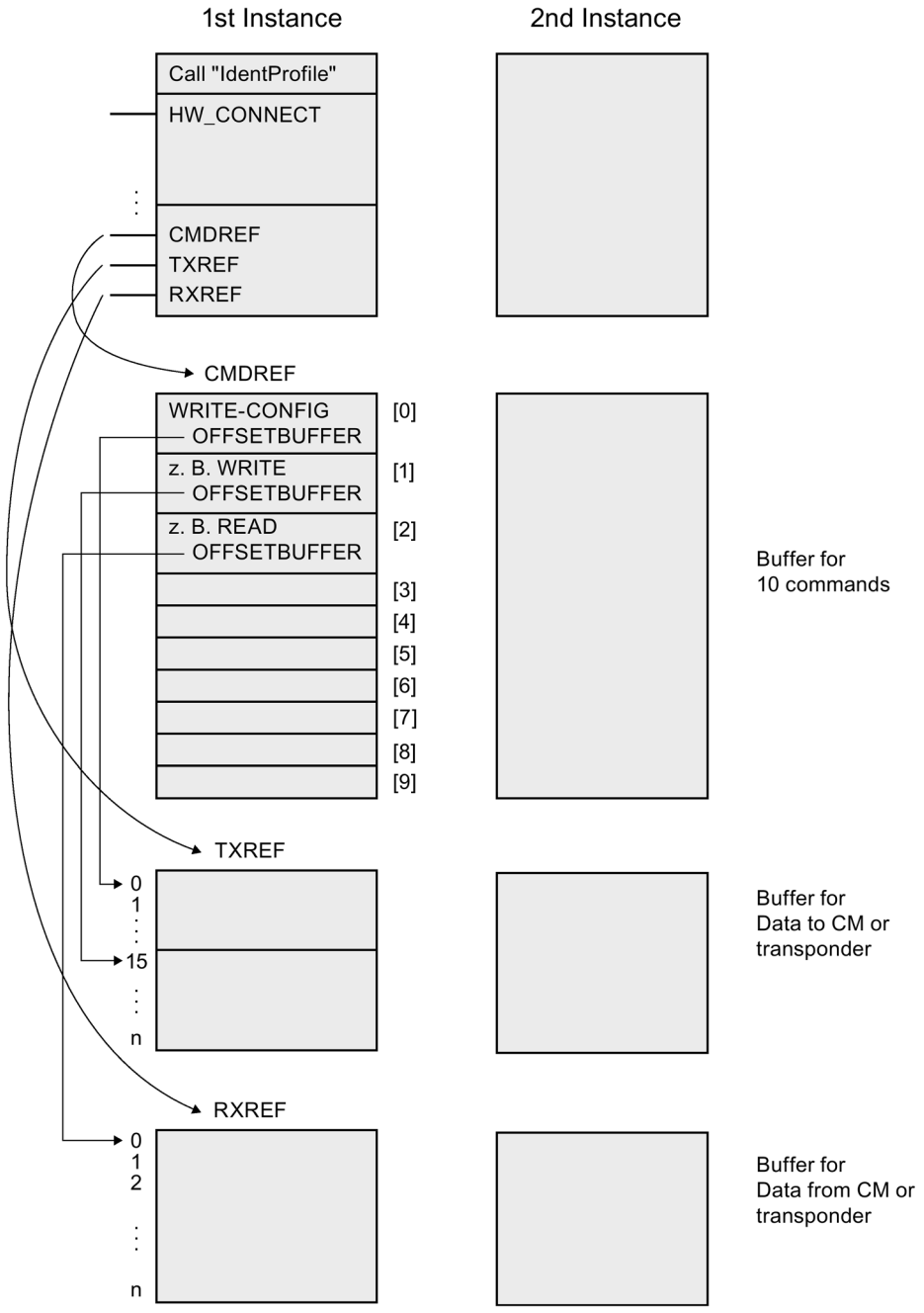


Figure 4-31 Data structure example of the Ident profile

Explanation of the data structure example

- CMDREF[0]:

  Command "WRITE-CONFIG", OFFSETBUFFER = 0

  At the "CMDREF[0]" point you need to set the "WRITE-CONFIG" command so that the "INIT/Reset" is correctly executed.

- CMDREF[1]:

  Command "WRITE", OFFSETBUFFER = 15

- CMDREF[2]:

  Command "READ", OFFSETBUFFER = 0

If the "CMDREF[1]" command is selected, the Write command is started and the data to be written is fetched from the "TXREF" parameter starting with element 15. If the "CMDREF[2]" command is selected, the read data is stored starting at element 0 in the "RXREF" parameter.

4.6.4 Commands of the Ident profile

4.6.4.1 Overview of the commands

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

Table 4- 45 Commands of the Ident profile

Command	Command code		Parameters used	Description
	HEX	ASCII		
PHYSICAL-READ	0x70	'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	0x71	'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	0x74	't'	OFFSETBUFFER, ATTRIBUTES	Reads out the status of the reader.
TAG-STATUS	0x73	's'	OFFSETBUFFER, EPCID_UID, LEN_ID, ATTRIBUTES	Reads out the status of a transponder.
INVENTORY	0x69	'i'	OFFSETBUFFER, ATTRIBUTES, DURATION, DUR_UNIT	Requests a list of all currently accessible transponders within the antenna range.
FORMAT	0x66	'f'	OFFSETBUFFER, EPCID_UID, LEN_ID, LEN_DATA	Initializes the transponder.

Command	Command code		Parameters used	Description
	HEX	ASCII		
PUT	0x65	'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.
WRITE-ID	0x67	‘g‘	OFFSETBUFFER, EPCID_UID, LEN_ID, NEW-LEN_ID, PSWD	RF68xR: Writes a new EPC-ID to the transponder.
KILL-TAG	0x6A	‘j‘	EPCID_UID, LEN_ID, PSWD	RF68xR: The transponder is permanently deactivated.
LOCK-TAG-BANK	0x79	‘y‘	EPCID_UID, LEN_ID, PSWD, ACTION, MASK	RF68xR: The corresponding memory area of the transponder is blocked as specified.
EDIT-BLACKLIST	0x7A	‘z‘	EPCID_UID, LEN_ID, MODE	RF68xR: The black list is processed. The current transponder can be added, all identified transponders added, individual transponders deleted or all transponders deleted.
GET-BLACKLIST	0x6C	‘l‘	OFFSETBUFFER	RF68xR: The entire TagIDs are read out from the black list.
READ-CONFIG	0x61	'a'	--	Reads out the parameters from the communications module/reader.
WRITE-CONFIG	0x78	'x'	LEN_DATA, CONFIG	Sends new parameters to the communications module/reader.

4.6.4.2 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 programmed. The "CMDSEL" parameter specifies which command [0...9] is started with "EXECUTE".

Remember that the first element in the buffer is always reserved for "INIT". This means that if "INIT" is set, "CMDSEL" must be set to "0" and element "0" 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 4- 46 Command structure of the parameters

Parameter	Data type	Default value	Description
CMD	SINT	0	Command code (compare the table in the section "Commands of the Ident profile (Page 66)".)
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	SINT[62]	0	Buffer for up to 62 bytes EPC-ID 2 - 62-byte EPC-ID is entered at the start of the buffer (length is set by "LEN_ID") Default value: 0x00 ≙ unspecified single tag access
LEN_DATA	DINT	0	Amount of data to be read/written in bytes
ADR_TAG	DINT	0	Physical start address on the transponder
ATTRIBUTES	SINT	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	SINT	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	SINT	0	Length of the valid data in the "EPCID_UID" field.
	MEM_BANK	SINT	3	Memory bank on the transponder • 0x00 = RESERVED • 0x01 = EPC • 0x02 = TID • 0x03 = USER
	PSWD	DINT	0	Password for transponder access 0x00 ≙ no password
	EDIT_BLACKLIST_MODE	SINT	0	Mode • 0x00 = add EPC-ID • 0x01 = add all "Observed" transponders • 0x02 = delete EPC-ID • 0x03 = delete all
	INVENTORY_DURATION	DINT	0	Duration Period of time or number of inventories or number of "Observed" events Example: • 0x00 ≙ no inventory • 0x01 ≙ one inventory
	INVENTORY_DUR_UNIT	DINT	0	Unit for "DURATION" • 0x00 = time [ms] • 0x01 = inventories • 0x02 = number of "Observed" events
	LOCK-TAG-BANK_ACTION	DINT	0	Lock-Action (see "EPC Specification")
	LOCK-TAG-BANK_MASK	DINT	0	Lock-Mask (see "EPC Specification")

4.6.4.3 Commands

All relevant parameters and parameter values for the individual commands are listed below. Parameters that are not listed receive the default value specified in the previous section.

Table 4- 47 PHYSICAL-READ

Parameter	Value / description
CMD	0x70
OFFSETBUFFER	Offset in the "RXREF" receive buffer
LEN_DATA	Length of received data RF300 Gen1: Max. 57,085 bytes per read operation
ADDR_TAG	Address on the transponder
CHAINED	• True = chained • False = not chained
EPCID_UID	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_ID	Length of the EPC-ID (2-62 bytes) 0x00 ≙ unspecified single tag access
MEM_BANK	Memory bank • 0x00 ≙ Reserved • 0x01 ≙ EPC • 0x02 ≙ TID • 0x03 ≙ USER
PSWD	Password 0x00 ≙ no password
RXREF	Read data

Table 4- 48 PHYSICAL-WRITE

Parameter	Value / description
CMD	0x71
OFFSETBUFFER	Offset in the "TXREF" send buffer
LEN_DATA	Length of the data to be written RF300 Gen1: Max. 57,085 bytes per write operation
ADDR_TAG	Address on the transponder
CHAINED	• True = chained • False = not chained
EPCID_UID	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_ID	Length of the EPC-ID (2-62 bytes) 0x00 ≙ unspecified single tag access
MEM_BANK	Memory bank • 0x00 ≙ Reserved • 0x01 ≙ EPC • 0x02 ≙ TID • 0x03 ≙ USER
PSWD	Password 0x00 ≙ no password
TXREF	Data to be written

Table 4- 49 READER-STATUS

Parameter	Value / description
CMD	0x74
OFFSETBUFFER	Offset in the "RXREF" receive buffer
ATTRIBUTES	Identifier of the status modes / possible entries: • RF200: 0x81 • RF300: 0x81, 0x86 • RF620R, RF630R: 0x87, 0x88, 0xA0, 0xA1 • RF68xR: 0x89
RXREF	Received status data You will find the data structure of the status modes in the section "Reader_Status (Page 44)".

Table 4- 50 TAG-STATUS

Parameter	Value / description
CMD	0x73
OFFSETBUFFER	Offset in the "RXREF" receive buffer
ATTRIBUTES	Identifier of the status modes / possible entries: • RF200: 0x83 • RF300 (RF300T): 0x04, 0x82 • RF300 (ISO): 0x83 • RF300 Gen2: 0x83 • RF600 (R620R, RF630R): 0x84, 0x85
EPCID_UID	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_ID	Length of the EPC-ID/UID
RXREF	Received status data You will find the data structure of the status modes in the section "Tag_Status (Page 49)".

Table 4- 51 INVENTORY

Parameter	Value / description
CMD	0x69
OFFSETBUFFER	Offset in the "RXREF" receive buffer
ATTRIBUTES	Identifier of the status modes / possible entries: RF68xR: • 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: • 0x00 ≙ list of all tags with UID
INVENTORY_DURATION	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

Parameter	Value / description
INVENTORY_DUR_UNIT	Only for 0x80 and 0x81: Unit for "DURATION" • 0x00 ≙ time [ms] • 0x01 ≙ inventories • 0x02 ≙ number of transponders that achieve the "Observed" state
RXREF	Only with 0x80 and 0x81 for RF68xR: 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 29)".

Table 4- 52 FORMAT (not with RF200 and RF68xR)

Parameter	Value / description
CMD	0x66
OFFSETBUFFER	Offset in the "TXREF" send buffer
LEN_DATA	Length of the parameter data to be sent (15 bytes)
EPCID_UID	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_ID	Length of the EPC-ID/UID
TXREF	Parameter data to be written

Table 4- 53 Structure of the data attachment for the "FORMAT" command with normal addressing

Byte	1...8	9	10	11	12	13	14	15
Value	0x00	0x06	0x03	0x00	INIT-Wert	0x00	MSB	LSB

Table 4- 54 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 "0x00"
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 4- 55 Memory sizes of the transponders

Transponder type			Memory size	INIT duration
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 4- 56 PUT (not with RF68xR)

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

Table 4- 57 Data structure of the PUT command

Put_SET_ANT		Switches the antenna of the reader off and on. 1 | 2 | 3 'N' | 'A' | Mode
	Mode	RF200/RF300: • 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 | 2 | 3 ... 10 | 11 'N' | 'K' | UID | Mode
	UID	UID of the transponder
	Mode	• 0x00 ≙ end processing of the transponder • 0x01 ≙ processing pause of the transponder
	Length	11

Table 4- 58 WRITE-ID (only with RF68xR)

Parameter	Value / description
CMD	0x67
OFFSETBUFFER	Offset in the "TXREF" send buffer
EPCID_UID	Previous EPC-ID 0x00 ≙ unspecified single tag access
LEN_ID	Length of the previous EPC-ID (2-62 bytes) 0x00 ≙ unspecified single tag access
LEN_DATA	Length of the new EPC-ID
PSWD	Password 0x00 ≙ no password
TXREF	New EPC-ID

Table 4- 59 KILL-TAG (only with RF68xR)

Parameter	Value / description
CMD	0x6A
EPCID_UID	EPC-ID 0x00 ≙ unspecified single tag access
LEN_ID	Length of the previous EPC-ID (2-62 bytes) 0x00 ≙ unspecified single tag access
PSWD	Password must be ≠ 0x00

Table 4- 60 LOCK-TAG-BANK (only with RF68xR)

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

Table 4- 61 EDIT-BLACKLIST (only with RF68xR)

Parameter	Value / description
CMD	0x7A
EDIT_BLACKLIST_MODE	• 0x00 ≙ add EPC-ID • 0x01 ≙ Add all "Observed" transponders • 0x02 ≙ Delete EPC-ID • 0x03 ≙ delete all
EPCID_UID	EPC-ID 0x00 ≙ unspecified single tag access [1]
LEN_ID	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 4- 62 GET-BLACKLIST (only with RF68xR)

Parameter	Value / description
CMD	0x6C
OFFSETBUFFER	Offset in the "RXREF" receive buffer
RXREF	Read black list IDs

Table 4- 63 Result of GET-BLACKLIST

Name	Type	Comment
NUM_IDS	INT	Number of MDS
TAG_DATA	IID_IN_I_80[n]	
TAG_DATA[1]	IID_IN_I_80	
Reserved	SINT	
ID_Len	SINT	Length of EPC ID
EPC_ID	SINT[62]	EPC-ID
TAG_DATA[2]	IID_IN_I_80	
...	...	
TAG_DATA[n]	IID_IN_I_80	

Table 4- 64 READ-CONFIG

Parameter	Value / description
CMD	0x61
OFFSETBUFFER	Offset in the "RXREF" receive buffer
RXREF	Read reset parameters

Table 4- 65 WRITE-CONFIG

Parameter	Value / description
CMD	0x78
OFFSETBUFFER	Offset in the "TXREF" send buffer
LEN_DATA	Length of the parameter data
CONFIG	• 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
TXREF	Configuration data to be sent

Structure of the configuration data attachment of "WRITE-CONFIG"

For RF68xR:

- 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 CM/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 CM/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 38)". LEN_DATA = 0x08

For RF200, RF300, RF620R, RF630R

Table 4- 66 For RF200, RF300, RF620R, RF630R when "CONFIG = 0x03"

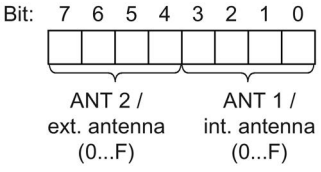
Byte	1	2...5	6 [1]	7...8	9	10	11	12	13...14	15	16
Value	0x04	0x00	0x0A 0x05	0x00	scanning_ time	param	op- tion_1	distance_ limiting	No. of tran- sponders	field_on_ control	field_on_ time

[1] 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 4- 67 Bytes of the "PARAM" parameter

Byte	Value	RFID system	Meaning				
Byte 9	scanning_ time	RF200, RF300	0x00 (reserved)				
		RF600	"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 variant is applicable to your country before you select a wireless profile.				
			Value	Description	ETSI	FCC	CMIIT
			0x00	No standard selected; the error "0x15" is output	--	--	--
			0x01	Reader works with the default wireless profile. Value of the default wireless profile:	ETSI new	FCC	China
			0x02	ETSI new: EU, EFTA, Turkey; 4-channel plan	X	--	--
			0x03	ETSI old: EU, EFTA, Turkey; readers commissioned after December 31, 2009, must not be operated with this setting.	X	--	--
			0x04	FCC: e.g. USA, Canada, Mexico	--	X	--
			0x05	Reserved	--	--	--
			0x06	China	--	--	
			0x07	Thailand	--	X	--
			0x08	Brazil	--	X	--
			0x09	South Korea	--	X	--
			0xC0	India	X	--	--
			Note: If you select country profiles other than those defined for the particular reader variant, the error message "09" is acknowledged.				
Byte 10	param	RF200, RF300	Setting for the RFID mode and presence check				
			Bit	Description			
			7 ... 5	Presence check and transponder control: • 0x00 = Operation without presence check • 0x01 = Operation with presence check Antenna is permanently switched on. • 0x04 = Operation without presence check Antenna is switched off. The antenna is only switched on when one of the following commands is sent: Read, Write, Init, Tag-Status			
			4	To be assigned the value "0x00".			
			3 ... 0	RFID mode: • 0x05 = Without multitag mode with communications module ASM 475, ASM 456, RF170C, RF180C • 0x06 = With multitag mode with communications module ASM 475, ASM 456, RF170C, RF180C			

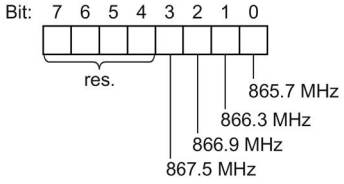
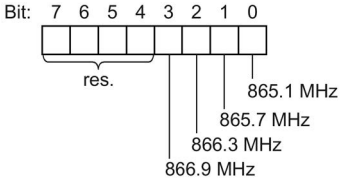
<table>
<tr><th>Byte</th><th>Value</th><th>RFID system</th><th colspan="3">Meaning</th></tr>
<tr><td rowspan="11"></td><td rowspan="11"></td><td rowspan="8"></td><td colspan="3">Value of bit 3 ... 0</td></tr>
<tr><td>Value of bit 3 ... 0</td><td>Operating mode</td><td>Description</td></tr>
<tr><td>0x00</td><td>Reserved</td><td>reserved for the setting with the switch or GSD parameter assignment</td></tr>
<tr><td>0x01</td><td>MOBY I</td><td>If RF300 readers of the 2nd generation are to be operated in MOBY I mode, the value = 0x01 must be set.
Short "INIT" (only the "param" and "option_1" parameters are transferred to the reader).</td></tr>
<tr><td>0x05</td><td>RF200, RF300 - without multitag mode</td><td>ASM 475, ASM 456, RF170C, RF180C</td></tr>
<tr><td>0x06</td><td>MOBY U - with mulititag mode (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>0x07</td><td>RF300 - with mulititag mode (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="5">RF600</td><td colspan="3">RFID mode setting</td></tr>
<tr><td>Bit</td><td colspan="2">Description</td></tr>
<tr><td>7 ... 5</td><td colspan="2">Presence check and transponder control:
• 0x00 = Operation without presence check
• 0x01 = Operation with presence check
Antenna is permanently switched on.</td></tr>
<tr><td>4</td><td colspan="2">To be assigned the value "0x00".</td></tr>
<tr><td>3 ... 0</td><td colspan="2">RFID mode:
• 0x04 = ISTM mode
• 0x05 = Single tag mode
• 0x06 = With single tag mode (UID = 0x00), 4 bytes UID of the 8 byte handle ID or
With multitag mode, 4-byte UID as handle ID for accessing the transponder with EPC-ID of any length
• 0x07 = With single tag mode (UID = 0x00), 8 bytes UID or
With multitag mode, 8-byte UID of bytes 5-12 of 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>
</table>

<table>
<tr><th>Byte</th><th>Value</th><th>RFID system</th><th colspan="2">Meaning</th></tr>
<tr><td rowspan="6">Byte 11</td><td rowspan="6">option_1</td><td rowspan="3">RF200,
RF300</td><td colspan="2">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.</td></tr>
<tr><td>Bit</td><td>Description</td></tr>
<tr><td>1</td><td>0x01 = Flashing of the "ERR" LED is reset by a "WRITE-CONFIG". With RF200/RF300, this resets the flashing of the "ERR" LED on the communications module and on the reader.</td></tr>
<tr><td rowspan="3">RF600</td><td colspan="2">This byte is bit-coded. As default, it has the value "B#16#0".</td></tr>
<tr><td>Bit</td><td>Description</td></tr>
<tr><td>1</td><td>0x01 = Flashing of the "ERR" LED of the communications module is reset by a "WRITE-CONFIG".</td></tr>
<tr><td>2</td><td>• 0x00 = OFF
• 0x01 = ON</td></tr>
<tr><td rowspan="12">Byte 12</td><td rowspan="12">distance_
limiting</td><td>RF200</td><td colspan="2">0x00 (reserved)</td></tr>
<tr><td rowspan="10">RF300
(only
RF380R)</td><td colspan="2">Note: This parameter is intended for trained users. Siemens recommends that untrained users use the default value.
Readers of the 1st generation: With this parameter you can change the transmit power (output power) of the RF380R reader of the 1st generation (6GT2801-3AB10). When doing this, remember that the change to the transmit power will affect the detection in the limit range (upper/lower operating distance), as well as the minimum distance that is to be maintained between adjacent RF380Rs.
Settings outside the specified range have the effect that the default value (1.25 W) will be set. In this case, for reasons of compatibility, there is no error message.
Readers of the 2nd generation: This setting is not necessary with the RF380R readers of the 2nd generation (6GT2801-3BAx0) because the power limits are optimized automatically depending on the reader-transponder distance. For reasons of compatibility, this setting can nevertheless be made. Note that the values "0x02", "0x03" and "0x04" bring about a reduction of the power of approximately 50%.
The following settings are possible:</td></tr>
<tr><td>Value</td><td>Transmit power</td></tr>
<tr><td>0x02</td><td>0.5 W</td></tr>
<tr><td>0x03</td><td>0.75 W</td></tr>
<tr><td>0x04</td><td>1.0 W</td></tr>
<tr><td>0x05</td><td>1.25 W (default)</td></tr>
<tr><td>0x06</td><td>1.5 W</td></tr>
<tr><td>0x07</td><td>1.75 W</td></tr>
<tr><td>0x08</td><td>2.0 W</td></tr>
<tr><td>RF600</td><td colspan="2">The transmit power of the reader is set with "distance_limiting".
 

By default, ANT 1 is used with the preset transmit power.</td></tr>
</table>

<table>
<tr><th>Byte</th><th>Value</th><th>RFID system</th><th colspan="6">Meaning</th></tr>
<tr><td rowspan="9"></td><td rowspan="9"></td><td rowspan="9"></td><td>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>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>0x00</td><td>18 / (65)</td><td>18 / (65)</td><td>20 / (105)</td><td>18 / (65)</td><td>18 / (65)</td></tr>
<tr><td>0x01</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></tr>
<tr><td>0x09</td><td>27 / (500)</td><td>27 / (500)</td><td>29 / (795)</td><td>27 / (500)</td><td>27 / (500)</td></tr>
<tr><td>0x0A</td><td>27 / (500)</td><td>28 / (630)</td><td>30 / (1000)</td><td>28 / (630)</td><td>27 / (500)</td></tr>
<tr><td>0x0B
(...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>No. of transponders</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>
<tr><td>Byte 15</td><td>field_on_control</td><td>RF200, RF300</td><td colspan="6">0x00 (reserved)</td></tr>
<tr><td></td><td></td><td rowspan="4">RF600</td><td colspan="6">"field_ON_control" sets the communications speed (fast/slow) and Tag Hold (ON/OFF).</td></tr>
<tr><td>Bit</td><td colspan="5">Description</td></tr>
<tr><td>0 ... 1</td><td colspan="5">Speed
Reader parameter assignments that have been optimized depending on the application are made available via the speed:
• 0x00 = Fast detection
• 0x02 = Slower, more reliable detection</td></tr>
<tr><td>2 ... 4</td><td colspan="5">Reserved</td></tr>
<tr><td></td><td></td><td>5</td><td colspan="5">Tag hold
• 0x00 = OFF
• 0x01 = ON</td></tr>
</table>

Byte	Value	RFID system	Meaning		
			6	ScanningMode (relevant for multitag mode): • 0x00 = Normal multitag mode without ScanningMode (including "repeat_command") • 0x01 = ScanningMode 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 0x01, the reader in multitag mode is prepared for the use of "ScanningMode".	
			7	Reserved	
Byte 16	field_on_time	RF200	Transponder type		
			0x01	Any ISO transponder	
		RF300	Selection of the transponder types used With the value "0x01"/"0x31" (ISO 15693 general), the readers of the 2nd generation always use the ISO commands with which the highest performance can be achieved for the particular transponder. With readers of the 1st generation, the value "0x01" activates the general ISO mode with rudimentary ISO commands. 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:		
			Value	Transponder type	Description
			0x00	RF300 (RF3xxT)	For all transponders of the type "RF3xxT"
			0x01	ISO 15693 general	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 15693 (Infineon, MDS D3xx)	e.g. MDS D324, D339
			0x04	ISO 15693 (Fujitsu - 2 kB, MDS D4xx)	e.g. MDS D421, D422, D423, D424, D425, D426, D428, D460
			0x05	ISO 15693 (NXP, MDS D1xx)	e.g. MDS D100, D124, D126, D139, D150, D165
			0x06	ISO 15693 (TI, MDS D2xx)	e.g. MDS D200
			0x07	ISO 15693 (STM, MDS D261)	e.g. MDS D261
			0x08[1)]	ISO 15693 (Fujitsu - 8 kB, MDS D5xx)	e.g. MDS D521, D522, D524, D525, D528
			0x10[1)]	RF300 (RF3xxT)	For all transponders of the type "RF3xxT"

<table>
<tr><th>Byte</th><th>Value</th><th>RFID system</th><th colspan="3">Meaning</th></tr>
<tr><td rowspan="6"></td><td rowspan="6"></td><td rowspan="5"></td><td>0x20[1)]</td><td>ISO 14443 (MOBY E, E6xx)</td><td>e.g. MDS E600, E611, E623, E624</td></tr>
<tr><td>0x31[1)]</td><td>General mode (ISO, RF300, MOBY E)</td><td>Activation of the so-called "General mode" for processing the transponder types RF300, ISO 15693 and ISO 14443. 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 that individual settings or transponder families can be combined (e.g. ISO 15693 general + RF300). In this case, the relevant hex values need to be combined (ISO 15693 general [0x01] + RF300 [0x10] = 0x11).
Note:
• ISO 15693: 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.
• ISO 15693: With "TAG_TYPE(ftim) = 0x01", the readers of the 1st generation support the ISO commands "Inventory", "ReadSingleBlock", "WriteSingleBlock", "LockBlock," "GetSystemInformation" and for multitags also "Select" and "ResetToReady".
• If a previously unknown transponder cannot be identified based on the parameters above, an error message is generated (error_MOBY= "0x0D").
• Invalid parameters are rejected with an error message ("error_MOBY= 0x15").</td></tr>
<tr><td colspan="3">[1)] Applies only to readers of the 2nd generation.</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 colspan="2">Changing the channel assignment in the India wireless profile ("scanning_time = 0xC0"):
 
</td></tr>
</table>

Byte	Value	RFID system	Meaning
			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

4.6.4.4 Expanded commands for optical reader systems (MV400/MV500)

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 4- 68 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 4- 69 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/MV500 have further commands that can be transferred with the "PHYSICAL-WRITE" command.

Table 4- 70 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:
			• 0x02	• 0x01 = Program change
			• 0x01	• 0x02 = Activate read program number
			• Match string length + 3	• 0x03 = Write match string
			• 0x01	• 0x04 = Activate read match string
			• 0x01	• 0x05 = Set DISA bit
			• 0x01	• 0x06 = Reset DISA bit
			• Total length of the XMATCH user data + 4	• 0x07 = Write trigger-synchronized match string (XMATCH)
			• 0x07	• 0x08 = Set Digital Out

Table 4- 71 Command data area "TXREF" command identifier 0x01 (program change)

Address	Value	Meaning
0x0000	0x01	"Program change" command identifier
0x0001	0x00 ... 0x0F	Number of the program

Table 4- 72 Command data area "TXREF" command identifier 0x02 (activate read program number)

Address	Value	Meaning
0x0000	0x02	"Read program number" command identifier

Table 4- 73 Command data area "TXREF" command identifier 0x03 (write match string)

Address	Value	Meaning
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 4- 74 Command data area "TXREF" command identifier 0x04 (activate read match string)

Address	Value	Meaning
0x0000	0x04	Command identifier "Read match string"

Table 4- 75 Command data area "TXREF" command identifier 0x05 (set DISA bit)

Address	Value	Meaning
0x0000	0x05	Command identifier "Set DISA bit"

Table 4- 76 Command data area "TXREF" command identifier 0x06 (reset DISA bit)

Address	Value	Meaning
0x0000	0x06	Command identifier "Reset DISA bit"

Table 4- 77 Command data area "TXREF" command identifier 0x07 (XMATCH)

Address	Value	Meaning
0x0000	0x07	Command identifier "XMATCH"
0x0001	0x00	Reserved
0x0002	You will find detailed information in the manual "SIMATIC MV420 / SIMATIC MV440" or "SIMATIC MV500".	XMATCH user data
…		
0xN		

Table 4- 78 Command data area "TXREF" command identifier 0x08 (set Digital Out)

Address	Value	Meaning
0x0000	0x08	Command identifier "Set Digital Out"
0x0001	0x01...0x04	Number of the logical external signal. Corresponds to "EXT_1", "EXT_2", "EXT_3" and "EXT_4".
0x0002	0x00...0x02	Level of the signal • 0x00: Set level statically to "low". • 0x01: Set level statically to "high". • 0x02: Set level for configured pulse time to "high".
0x0003	0x01...0x07	Type of logic operation • 0x01: Logical "OR" • 0x02: Logical "AND" • 0x03: Logical "Exclusive OR" • 0x04: no logic operation • 0x05: Logical "OR not" • 0x06: Logical "AND not" • 0x07: Logical "Exclusive OR not"
0x0004	0x00...0x05	Logical signal linked to. If the logic operation type is 0x4, the parameter has no significance. • 0x00: Logical signal "IN_OP" • 0x01: Logical signal "TRD" • 0x02: Logical signal "RDY" • 0x03: Logical signal "READ" • 0x04: Logical signal "MATCH" • 0x05: Logical signal "NOK"
0x0005	0x00	Reserved, must be 0x00 to retain upwards compatibility.
0x0006	0x00	Reserved, must be 0x00 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 4- 79 PHYSICAL-READ

CMD	OFFSET BUFFER	ADDR_TAG	LEN_DATA	RXREF
0x70	Offset in the "RXREF" send buffer	0x0000	Length of the data to be fetched from the reader:	Data fetched from the reader:
			• ≥ code length +2	• Code data
			• = 0x01	• Program number
			• ≥ Match string length +2	• Match string

4.6.4.5 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.

4.6.4.6 Editing commands

Follow the steps below to edit the commands:

1. Write the "CMDREF" (SINT[n]) parameter with the required commands.

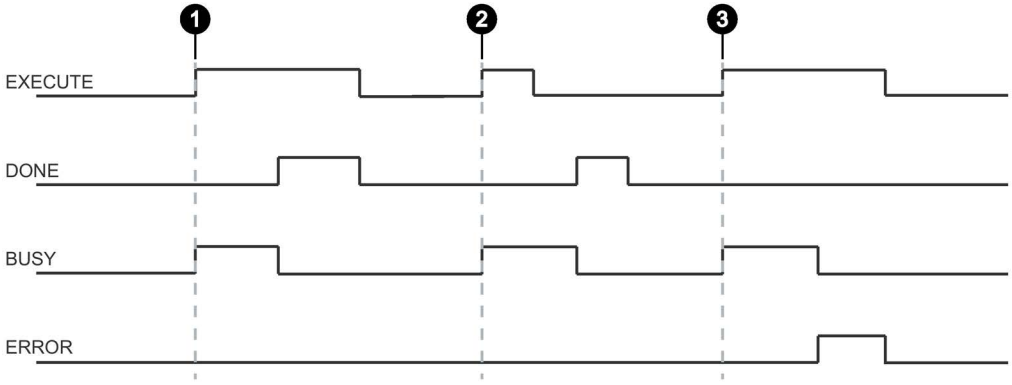
   The content of "CMDREF" = [0] is reserved for initialization. It is executed when the "INIT" input of the Ident profile is set and "CMDSEL" is = [0].
2. Transfer the data to be written to the send data buffer "TXBUF".
3. Select the previously written command (SINT[n]) with the "CMDSEL" parameter.
4. 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 The function/instruction is started by setting EXECUTE (EXECUTE = 1). 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 4-32 General sequence of the Ident profile

4.6.4.7 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 controller
- Switching on the power supply of the CM/reader
- Plugging the reader into the CM
- Interruption in Ethernet/IP communication
- An error message by the "STATUS" parameter

4.6.4.8 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 ("SINT[n]" of "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.

At the time of publication of this manual, chaining is supported only by the RF68xR readers.

Overview of the commands

Table 4- 80 Overview of the commands with which chaining is possible

Command	Command code		Description
	HEX	ASCII	
PHYSICAL-READ	0x70	'p'	Reads data from a transponder / optical reader system by specifying the physical start address, the length and the password.
PHYSICAL-WRITE	0x71	'q'	Writes data to a transponder / optical reader system by specifying the physical start address, the length and the password.
READER-STATUS	0x74	't'	Reads out the status of a communications module. This command must not be the last command within the chain.
TAG-STATUS	0x73	's'	Reads out the status of a transponder.

Command	Command code		Description
	HEX	ASCII	
INVENTORY	0x69	'i'	Requests a list of all currently accessible transponders within the antenna range.
FORMAT	0x66	'f'	Initializes the transponder.
PUT	0x65	'e'	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.
WRITE-ID	0x67	‘g‘	RF68xR: Writes a new EPC-ID to the transponder.
KILL-TAG	0x6A	‘j‘	RF68xR: The transponder is permanently deactivated.
LOCK-TAG-BANK	0x79	‘y‘	RF68xR: Defines a password for transponder access.

Example of command structure

Table 4- 81 Example of a command structure with 3 commands (without EPC-ID)

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

With chaining, the entire "IID_CMD_STRUCT" buffer ("IID_CMD_STRUCT[0...n]") 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[0]" reserved for "INIT"

In the Ident profile, the "IID_CMD_STRUCT[0]" parameter is normally reserved for "INIT". If you want to use "IID_CMD_STRUCT[0]" for another command, make sure that the reset parameters are written into this parameter when there is an "INIT".

4.7 Digital inputs/outputs

The RF650R/RF68xR readers have four digital inputs/outputs, while the RF615R reader has one digital input/output. The outputs can be configured in the WBM. The request and control are via the controller. You can control the addresses of the inputs/outputs via a 1-WORD subsegment of the process image of the reader/CM. You can enter the addresses using STEP 7 or Studio 5000 Logix Designer in the properties of the reader in the parameter "Digital inputs/outputs". You can access the digital inputs/outputs via the byte with the lower value address.

The structure of this byte and the assignment to the digital inputs/outputs is shown below:

Table 4- 82 Assignment of the digital inputs/outputs

Bit	3	2	1	0
Input byte	DI 3	DI 2	DI 1	DI 0
Output byte	DO 3	DO 2	DO 1	DO 0

All other bits of the input/output byte are reserved.

4.8 Transponder addressing

Addressing

Addressing of the transponders is linear from address "0x0000" (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

System	Addressing	16-bit hexadecimal number
RF200, RF300	**MDS D139 (I-Code 1; 44 bytes)**	
	Start address	0x0000
	End address	0x002B
	ID-Nr.: (fixed-coded, can only be read as a whole)	
	Start address	0xFFF0
	Length	0x0008
	ISO-MDS (I-Code SLI; 112 bytes)	
	Start address	0x0000
	End address	0x006F
	ID-Nr.: (fixed-coded, can only be read as a whole)	
	Start address	0xFFF0
	Length	0x0008
	ISO MDS (Tag-it HF-I; 256 bytes)	
	Start address	0x0000
	End address	0x00FF
	ID-Nr.: (fixed-coded, can only be read as a whole)	
	Start address	0xFFF0
	Length	0x0008
	ISO MDS (my-d SRF55V10P; 992 bytes)	
	Start address	0x0000
	End address	0x03DF
	ID-Nr.: (fixed-coded, can only be read as a whole)	
	Start address	0xFFF0
	Length	0x0008
	ISO-MDS (MB 89R118B, 2000 bytes)	
	Start address	0x0000
	End address	0x07CF
	ID-Nr.: (fixed-coded, can only be read as a whole)	

System	Addressing	16-bit hexadecimal number
	Start address	0xFFF0
	Length	0x0008

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	0xFF00
	End address	0xFF13
	ID-Nr.: (fixed-coded; can only be output as a whole)	
	Start address	0xFFF0
	Length	0x0008
	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	0xFF00
	End address	0xFF13
	R/W memory (FRAM)	
	Start address	0x0000
	End address	0x1FFC
	Id-Nr.: (fixed-coded, can only be read out as a whole)	
	Start address	0xFFF0
	Length	0x0008
	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	0xFF00
	End address	0xFF13
	R/W memory (FRAM)	
	Start address	0x0000
	End address	0x7FFC
	ID-Nr.: (fixed-coded; can only be output as a whole)	
	Start address	0xFFF0
	Length	0x0008
	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))	

System	Addressing	16-bit hexadecimal number
	Start address	0xFF00
	End address	0xFF13
	R/W memory (FRAM)	
	Start address	0x0000
	End address	0xFEFC
	ID-Nr.: (fixed-coded; can only be output as a whole)	
	Start address	0xFFF0
	Length	0x0008

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:

- 0xFF80
- 0xFF84
- 0xFF88
- 0xFF8C
- 0xFF90

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**
0xFF00	1 .. 20	0xFF80	4,8,12,16,20
0xFF01	1 .. 19		
0xFF02	1 .. 18		
0xFF03	1 .. 17		
0xFF04	1 .. 16	0xFF84	4,8,12,16
0xFF05	1 .. 15		
0xFF06	1 .. 14		
0xFF07	1 .. 13		
0xFF08	1 .. 12	0xFF88	4,8,12
0xFF09	1 .. 11		
0xFF0A	1 .. 10		
0xFF0B	1 .. 9		
0xFF0C	1 .. 8	0xFF8C	4.8
0xFF0D	1 .. 7		
0xFF0E	1 .. 6		
0xFF0F	1 .. 5		
0xFF10	1 .. 4	0xFF90	4
0xFF11	1 .. 3		
0xFF12	1 .. 2		
0xFF13	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

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
RF642L	NXP / UCODE 7xm-2k	00 - FF 256 bytes	FF00-FF38 / 448 bits FF00-FF0B / 96 bits	read/ write	FFC0-FFCB 12 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

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
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
RF682T	NXP UCode DNA	00 - 17F 384 bytes	FF00-FF38 / 448 bits FF00-FF0B / 96 bits	read/ write	FFC0-FFCB 12 bytes	FF80-FF87 8 bytes	LOCKED	Yes

1) The user area also applies to the readers RF61xR/RF650R/RF68xR 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 RF61xR/RF650R/RF68xR

With the new readers RF61xR/RF650R/RF68xR, 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.

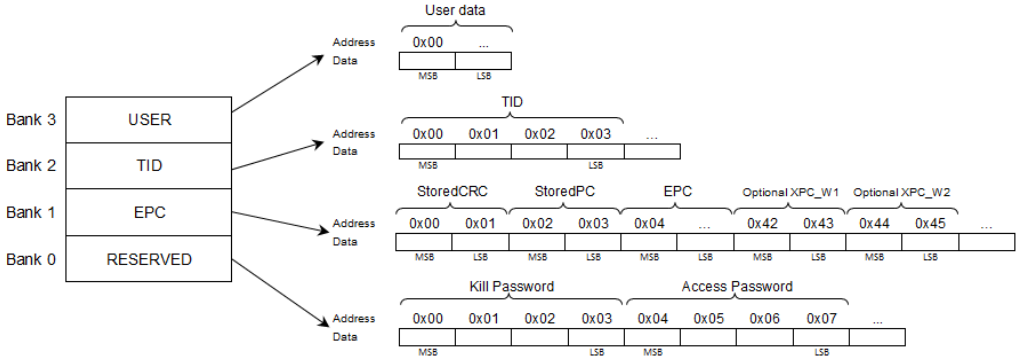


Figure 4-33 Memory configuration

Error messages

5

5.1 Structure of the "STATUS" output parameter

There is always an error status in the "Ident profile" instruction 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 5- 1 Bytes of the "STATUS" output parameter

Byte	Meaning
Byte 0	Instruction numbers • Cx - Error in fieldbus communication • 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 followinig meaning: • 0x00 - no error, no warning • 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.

5.2 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 5- 2 Error messages from communications module/reader or from the Ident profile via the STATUS output parameter

Error message (hex)	Description
0xE1FE01	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 • RF620R/RF630R: Transponder is write protected (Memory Lock)
0xE1FE02	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
0xE1FE03	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
0xE1FE04	Only during initialization: Transponder is unable to execute the initialization command • Transponder is defective
0xE1FE06	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
0xE1FE07	Password error RF620R/RF630R: Incorrect password
0xE1FE08	The transponder in the antenna field does not have the expected UID or has no UID.
0xE1FE0A	The transponder is read/write-protected.
0xE1FE81	The transponder is not responding.
0xE1FE82	The transponder password is incorrect. Access is denied.
0xE1FE83	The verification of the written transponder data has failed.

Error message (hex)	Description
0xE1FE84	General transponder error
0xE1FE85	The transponder has too little power to execute the command.
0xE2FE01	• 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 • 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
0xE2FE02	• More transponders are located in the transmission window than can be processed at the same time by the reader. • RF620R/RF630R: Transponder power supply close to limit. Increase the antenna power or reduce the distance to the transponder.
0xE2FE81	There is no transponder with the required EPC-ID in the transmission window or there is no transponder at all in the antenna field.
0xE2FE82	The requested data is not available.
0xE2FE83	The transponder signals a CRC error.
0xE2FE84	The selected antenna is not enabled.
0xE2FE85	The selected frequency is not enabled.
0xE2FE86	The carrier signal is not activated.
0xE2FE87	There is more than one transponder in the transmission window.

Error message (hex)	Description
0xE2FE88	General radio protocol error
0xE4FE01	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"
0xE4FE03	• 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
0xE4FE04	The buffer on the communications module or reader is not adequate to store the command temporarily.
0xE4FE05	The buffer on the communications module or reader is not adequate to store the data temporarily.
0xE4FE06	The command is not permitted in this status or is not supported.
0xE4FE07	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 baud rate was switched over but power has not yet been cycled
0xE4FE81	Reserved
0xE4FE8A	General error
0xE4FE8B	No or bad configuration data was transferred.

Error message (hex)	Description
0xE4FE8C	• 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 / Ethernet/IP error occurred – Execute "INIT"
0xE4FE8D	• 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
0xE4FE8E	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"
0xE5FE01	Incorrect sequence number order (SN) on the reader/communications module
0xE5FE02	Incorrect sequence number order (SN) in the Ident profile Possible cause: User mode "RFID standard profile" is not set in the device configuration.
0xE5FE04	Invalid data block number (DBN) on the reader/communications module
0xE5FE05	Invalid data block number (DBN) in the Ident profile
0xE5FE06	Invalid data block length (DBL) on the reader/communications module
0xE5FE07	Invalid data block length (DBL) in the Ident profile
0xE5FE08	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.
0xE5FE09	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).
0xE5FE0A	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.

Error message (hex)	Description
0xE5FE0B	Incorrect sequence of acknowledgement frames (TDB / DBN)
0xE5FE0C	Synchronization error (incorrect increment of AC_H / AC_L and CC_H / CC_L in the cyclic control word). "INIT" had to be executed
0xE6FE01	Unknown command Ident profile is sending an uninterpretable command to the communications module. • The transponder has signaled an address error.
0xE6FE02	Invalid command index CI
0xE6FE03	• Bad parameter assignment of the communications module or reader – Check "INPUT" parameter in the Ident profile. – Check the parameter assignment in RSLogix. – "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 – 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 instruction was selected – Regardless of the selected reader system, use the "Reset_Reader" instruction.
0xE6FE04	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.
0xE6FE05	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.
0xE6FE06	The reset timer has expired.
0xE6FE81	Reserved
0xE6FE82	Reserved
0xE6FE83	Reserved
0xE6FE84	Reserved

Error message (hex)	Description
0xE6FE85	Reserved
0xE6FE86	The inventory command failed.
0xE6FE87	Read access to the transponder has failed.
0xE6FE88	Write access to the transponder has failed.
0xE6FE89	Writing the EPC-ID on the transponder has failed.
0xE6FE8A	Enabling write protection on the transponder has failed.
0xE6FE8B	The "Kill" command failed.
0xE7FE01	In this status, only the "Reset_Reader" command ("WRITE-CONFIG") is permitted.
0xE7FE02	The "CMD" command code or the value in "CMD SEL" is not permitted.
0xE7FE03	The "LEN_DATA" parameter of the command is too long. It does not match the global data reserved in the send data buffer (TXBUF).
0xE7FE04	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.
0xE7FE05	This error tells you that only an "INIT" command is permitted as the next command. All other commands are rejected.
0xE7FE06	Wrong index (outside range of "101 ... 108" and "-20401 ... -20418")
0xE7FE07	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".
0xE7FE08	Timeout during "INIT" (60 seconds according to "TC3WG9")
0xE7FE09	Command repetition is not supported.
0xE7FE0A	Error during the transfer of the PDU (Protocol Data Unit).
0xFxFExx	An "FxFExxh" error is identical to the corresponding "ExFExxh" error (see "ExFExxh" error). Byte 3 contains additional warning information.

5.3 Errors from Ethernet/IP

The transport layer of the bus system being used (Ethernet/IP) is signaling an error. For precise troubleshooting and analysis, a tracer can be useful. For Ethernet/IP, the open source software "Wireshark" can be used. The system diagnostics of Ethernet/IP can provide further information about the cause of the error.

With error messages output via the "STATUS" parameter of the Ident profile that are not contained in the section "Errors from the communications module/reader (Page 104)" you need to read out the following data ("Controller Tags") of the instance "WRREC" or "RDREC". The "MESSAGE" instruction outputs two error codes. One standard error code that is displayed via "STATUS" and the instance "WRREC" or "RDREC" in the "ERR" variable and an extended error code that is displayed in the "EXERR" variable.

RFID_01_CH00	{...}	{...}		IID_CHANNEL
RFID_01_CH00.CHANNEL_AOI	{...}	{...}		IDENT_PROF
RFID_01_CH00.CMD_ARRAY	{...}	{...}		IID_CMD_STR
RFID_01_CH00.HW_CONNECT	{...}	{...}		IID_HW_CONN
RFID_01_CH00.TXBUFFER	{...}	{...}	Decimal	SINT[2048]
RFID_01_CH00.RXBUFFER	{...}	{...}	Decimal	SINT[2048]
RFID_01_CH00_RDREC	{...}	{...}		MESSAGE
RFID_01_CH00_RDREC.Flags	16#02a0		Hex	INT
RFID_01_CH00_RDREC.EW	0		Decimal	BOOL
RFID_01_CH00_RDREC.ER	0		Decimal	BOOL
RFID_01_CH00_RDREC.DN	1		Decimal	BOOL
RFID_01_CH00_RDREC.ST	0		Decimal	BOOL
RFID_01_CH00_RDREC.EN	1		Decimal	BOOL
RFID_01_CH00_RDREC.TO	0		Decimal	BOOL
RFID_01_CH00_RDREC.EN_CC	1		Decimal	BOOL
RFID_01_CH00_RDREC.ERR	16#0000		Hex	INT
RFID_01_CH00_RDREC.EXERR	16#0000_0000		Hex	DINT

Figure 5-1 Tags of the "MESSAGE" instruction

You will find a description of these error codes in the help of the "RSLogix" configuration software.

5.4 Warnings

Byte 3 of the "STATUS" output parameter indicates warnings if byte 0 of the "STATUS" (instruction numbers) has the value "Fxh" or "Dxh".

Table 5- 3 Possible warnings when working with the Ident profile

Bytes 0...2	Byte 3	Meaning
FxFExxh	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
last_error_status	DINT	This variable contains the last instruction status if an error occurs. This value is always overwritten if a new error occurs with the block.
last_error_timestamp	DINT[7]	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/)