SIEMENS

SIMATIC Ident

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

Function Manual

07/2020
C79000-G8976-C410-04

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
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

Ⓟ 07/2020 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 June 2020.

The functions mentioned in this documentation are currently supported by the SIMATIC RF600, as well as communication modules SIMATIC RF185C/RF186C/RF188C, RF186CI/RF188CI and the communication module RFID 181EIP.

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/14971/man) pages or on the DVD "Ident Systems" (6GT2080-2AA20) in the respective manuals:

- SIMATIC RF185C, RF186C, RF188C, RF186CI, RF188CI
- SIMATIC RF600 (system manual and configuration manual)
- RFID 181EIP

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: (https://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: (https://support.industry.siemens.com/cs/ww/en/ps/15247/pm)

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: RF18xC/RF18xCI, RFID 181EIP and RF61xR/RF68xR

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 parameter instructions ("Param_RF18xC", "Param_RFID181EIP" or "Param_RF600") are used to configure the respective Ident device during startup.

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). For the RF600 readers and the RF18xC/RF18xCI communication modules, the EDS file can be downloaded via the WBM of the modules ("System > Device Description Files"). For the RF600 readers, the file is also stored on the DVD which is supplied with the 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 Studio 5000 Logix Designer.
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 RF600 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. Add an Ethernet/IP module if necessary.
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. "RF600") and click "Create".

   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 RF600 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 Studio 5000 Logix Designer.
3. Open the "Import Add-On Instruction" dialog ("FILE > Import Component > Add-On Instruction").
4. Go to the "*.L5X" file and select it.
5. Confirm your entry with "OK".

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

NOTICE
Overwriting existing add-in instructions If older versions of the add-in instructions are already included in your project, ensure that all existing instructions ("References > Add-On Instructions") and data types ("References > Data Types") are overwritten.

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_RF600	You can use these instructions to transfer the parameters required after a restart to the reader or communication module.
			Param_RF18xC	
			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_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_Reader	
			Reset_RF200	
			Reset_RF300	
			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_RF600	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_RF18XC	
		IID_DATA_RFID181EIP	
		IID_PARAMETER_RF18XC	Data type for the module and module configuration of the RF18xC/RF18xCI communication modules.
	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 or communication module and to process the data further directly without data type conversions.
		IID_READSTAT_ 86_RF300	
		IID_READSTAT_ 89_RF600	
		IID_READSTAT_ EF_RF300G2	
		IID_CM_STATUS_ A2	
		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_IN_I_80	Data types for the result of "Inventory" with the relevant attribute. The data types help you to interpret the data received from the reader and to process the data further directly without data type conversions.
		IID_IN_I_81	

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.

Note

Copying the data of the "IDENT_DATA" parameter for status data types

Note that the data of the "IDENT_DATA" parameter cannot always be copied to a structured data type with a single copy instruction due to the control architecture ("DINT" limits). If necessary, use several copy instructions.

4.3 General structure of the 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.

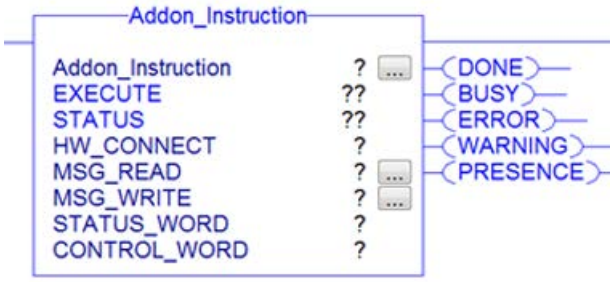


Figure 4-1 Example instructions

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.

Parameter	Description
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

After you have imported the instruction library into your project, you need to perform the following preparatory work:

1. Depending on the Ident devices used, you need to create variables for each Ident device and for each channel of an Ident device.
2. You then need to execute a parameter instruction for these Ident devices to establish the initial connection between Ident device and controller.
3. Finally, you need to execute a reset command for each channel of an Ident device to reset the connected reader and transfer the application-specific settings.

These steps are described in detail in the following sections.

Once the project preparation steps have been completed, you can execute application-specific commands, such as a read or write command.

4.4.1 Creating configuration variables

A configuration variable must be created for each module (CM/reader). You must create these variables in the "Controller Tags".

There is a separate data type for this variable for each type of module:

- Reader SIMATIC RF600: IID_DATA_RF600
- Communication module SIMATIC RF18xC/RF18xCI: IID_DATA_RF18XC
- Communication module RFID 181EIP: IID_DATA_RFID181EIP

These variables contain buffers for the parameters that are transferred during startup using the respective parameter instruction. These buffers (<variable name>.parameter) are needed later in the message variable for configuration of the "Source" parameter.

For the addressing and synchronization of individual channels/reading points of a module, the configuration variable contains a corresponding number of variables of the type "IID_HW_CONNECT". All application instructions as well as the "Ident_Profile"

instruction have a "HW_CONNECT" parameter where the respective "HW_CONNECT" variable must be created for the corresponding channel.

4.4.2 Creating a message variable

A message variable must be created for each module. The parameter assignment is transferred with the aid of this variable and this is required for the "MSG_PARAM" parameter of the corresponding "Param" instruction (e.g. "Param_RF18xC").

Two additional message variables are required for each channel/reading point of a module for sending and receiving Ethernet/IP telegrams. The message variable for receiving is required for the "MSG_Read" parameter of all Ident instructions. The message variable for sending is required for the parameter "MSG_WRITE" of all Ident instructions.

Variables of the "MESSAGE" type can only be created in the "Controller Tags".

4.4.2.1 Configuring the message variable for configuration

The message variable for the transfer of the configuration ("MSG_PARAM" parameter) must be structured as follows:

Table 4- 3 Structure of the message variable for transferring the configuration

Tab	Parameter	Description / Value
Configuration	Message Type	CIP Generic
	Service Type	Custom
	Service Code	0x41
	Source Element	<Configuration variable>.Parameter
Communication	Path	Click on the "Browse" button and select the desired module.

4.4.2.2 Configuring a message variable for receiving a channel

This message variable must be created and configured for each channel of a module. Ensure you select the correct buffer when setting the "Source Element" and "Destination Element" parameters.

The message variable for receiving Ethernet/IP telegrams ("MSG_PARAM" parameter) must be structured as follows:

Table 4- 4 Structure of the message variable for receiving a channel

Tab	Parameter	Description / Value
Configuration	Message Type	CIP Generic
	Service Type	Custom
	Service Code	0x40
	Source Element	<Configuration variable>.CH0*.Static.buf Possible values for CH0*: • RF600: CH01, ... CH04 • RF18xC/RF18xCI: CH01, ... CH04, CM_Configuration • RFID 181EIP: CH01, CH02 The RF18xC/RF18xCI communication modules have a configuration channel in addition to the reader channels. This is called "CM_Configuration" in the configuration variable. <Configuration Variable>.CM_Configuration.Static.buf
	Destination Element	Value must be identical to the value in the "Source Element" parameter
Communication	Path	Click on the "Browse" button and select the desired module.
	Connected	Deactivate both parameters.
	Large Connection	RF600 reader: These parameters can be activated to enable longer telegram lengths. You can find detailed information on this parameter in the section "Param_RF600 (Page 29)".

4.4.2.3 Configuring a message variable for sending a channel

This message variable must be created and configured for each channel of a module. Ensure you select the correct buffer when setting the "Source Element" and "Destination Element" parameters.

The message variable for sending Ethernet/IP telegrams ("MSG_WRITE" parameter) must be structured as follows

Table 4- 5 Structure of the message variable for sending a channel

Tab	Parameter	Description / Value
Configuration	Message Type	CIP Generic
	Service Type	Custom
	Service Code	0x41
	Source Element	Value is identical to the value in the "Source Element" parameter of the message variable for receiving.
Communication	Path	Click on the "Browse" button and select the desired module.
	Connected	Deactivate both parameters.
	Large Connection	RF600 reader: These parameters can be activated to enable longer telegram lengths. You can find detailed information on this parameter in the section "Param_RF600 (Page 29)".

4.4.3 Creating a parameter variable (RF18xC/RF18xCI)

These variables allow you to configure the modules and their channels. The variable must be of the "IID_PARAMETER_RF18XC" type and is required for the "Param_RF18xC" instruction. To do this, create the variable and supply the individual parameters of this variable with the corresponding values. The parameter values of these variables are described in the following tables.

The RF18xC/RF18xCI communication modules are available in several device versions which differ, among other things, in the number of reader interfaces (channels):

- RF185C: 1 channel
- RF186C: 2 channels
- RF188C: 4 channels
- RF186CI: 2 channels + DI/DO
- RF188CI: 4 channels + DI/DO

Each of these channels can be configured independently. The following basic parameters are available:

- General reader
  for RF300, RF200 and MV400/MV500
- RF300 fast protocol
  for RF300 (Gen2)
- Freeport

Parameter values of the "IID_PARAMETER_RF18XC" parameter variable

Table 4- 6 Basic parameters of the variables

Name	Value	Meaning	Description
Device	16#0000 0102	RF185C	Selection of the communication module used (device version)
	16#0000 0202	RF186C	
	16#0000 0302	RF188C	
	16#0000 0402	RF186CI	
	16#0000 0502	RF188CI	
IoMode	2	Input	Mode of the IO interface of the communication module • Input: Single digital input • Output: Single digital output • IO-Link: Mode for connecting an IO-Link device with up to 8 digital inputs/outputs.
	3	Output	
	4	IO-Link	
ReaderSlot01	--	--	Details of these parameters are provided in the table below. The number of reader slots displayed depends on the device version used.
ReaderSlot02	--	--	
ReaderSlot03	--	--	
ReaderSlot04	--	--	

Table 4- 7 Sub-parameters of the "ReaderSlot" parameters

Name	Value	Meaning	Description
ModuleID	16#2001 0002	General reader	Selection of the desired Ident system (ID of the module)
	16#2001 1002	RF300 fast protocol	
	16#2000 8002	Freeport	
BaudRate	16#01	19.2 kBd	Selection depends on the Ident device/system being used. With this parameter, you set the data transmission speed between the communication module and the reader.
	16#03	57.6 kBd	
	16#04	115.2 kBd	
	16#0A	921.6 kBd	
FreeportParameter	--	--	Details on this parameter are provided in the table below.

Table 4- 8 Sub-parameters of the "FreeportParameter"

Name	Value	Meaning	Description
Parity	0	None	Parity selection A sequence of data bits can be expanded by a parity bit. With its value "0" or "1", the parity bit is added to the sum of all bits (data bits and parity bits) to form a defined status. This increases data reliability. • None: Data is sent without a parity bit. • Even: The parity bit is set so that the sum of the data bits (including the parity bit) is even when the signal state is "1". • Odd: The parity bit is set so that the sum of the data bits (including the parity bit) is odd when the signal state is "1". • Fixed value 1: The parity bit is set permanently to the value "1". • Fixed value 0: The parity bit is set permanently to the value "0".
	1	Even	
	2	Odd	
	3	Fixed value 1	
	4	Fixed value 0	
DataBits	7	--	Selection of the number of bits to which a character is mapped.
	8	--	
StopBits	1	--	Selection of the number of stop bits

Name	Value	Meaning	Description
	2	--	that indicate the end of a character. The stop bits are appended to every transferred character during transmission.
Interface	1	RS422	Selection of the interface type that the connected hardware (reader / optical readers) uses.
	2	RS232	
EndDetection	1	On receipt of the end delimiter(s)	Specifies the end detection of a received frame: • On receipt of the end delimiter(s): At the end of the frame, there are one or two defined end delimiters. When data is received, the end of the frame is recognized when the configured end delimiter(s) is/are received. • On receipt of fixed number of characters: The length of the received frames is always the same. When data is received, the end of the frame is recognized when the set number of characters has been received. • After character delay time elapses: The frame has neither a fixed length nor defined end delimiters. The end of a frame is indicated by a gap in the character sequence. The size of this gap is specified by the character delay time.
	2	On receipt of fixed number of characters	
	3	After character delay time elapses	
NumberOfDelimeters	1	--	Requirement: "EndDetection = 1" Selection of the number of end delimiters. A maximum of 2 end delimiters can be configured. When data is received, the end of the frame is recognized when the selected end delimiter combination is received.
	2	--	

Name	Value	Meaning	Description
FirstEndDelimeter	16#00 ... 16#FF	--	Requirement: "EndDetection = 1" Entry of the first end delimiter of a maximum of two end delimiters for the end criteria "On receipt of the end delimiter(s)". The selected end delimiter or the selected end delimiter combination limits the length of the frame. Parameter value depending on the "DataBits" parameter.
SecondEndDelimeter	16#00 ... 16#FF	--	Requirement: "EndDetection = 1" Entry of the second end delimiter of a maximum of two end delimiters for the end criteria "On receipt of the end delimiter(s)". The selected end delimiter combination limits the length of the frame. Parameter value depending on the "DataBits" parameter.
DataLength	1 ... 233	--	Requirement: "EndDetection = 2" Entry of the frame length in bytes for the end criterion "On receipt of fixed number of characters".
DelayTime	1 ... 65535	--	Requirement: "EndDetection = 3" Entry of the time [ms] that can elapse until a frame end is recognized. Select the character delay time dependent on the sending behavior of your communication partner. Depending on the data transmission speed the character delay time is limited to a minimum value. Note that the ASCII driver also pauses between two frames during transmission.

4.4.4 Executing the parameter instruction

To establish an initial connection between the Ident device and the controller, a parameter instruction must be executed after each restart of the Ident device or the controller. This transfers the required parameter data to the Ident device and the values for the "HW_Connect" variables of the configuration variable. Each device family has its own parameter instructions:

- Reader SIMATIC RF600: Param_RF600
- Communication module SIMATIC RF18xC/RF18xCI: Param_RF18xC
- Communication module RFID 181EIP: Param_RFID181EIP

Procedure

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

1. Insert the parameter instruction from the "Controller Organizer" into your application program that matches your module.
2. Create a instance variable.
3. Connect the "READER_CONFIGDATA" parameter with the created configuration variable.

   You can find detailed information on this in the section "Creating configuration variables (Page 17)".
4. Connect the "MSG_PARAM" parameter with the created message variable.

   You can find detailed information on this in the section "Configuring the message variable for configuration (Page 18)".
5. Supply the device-specific parameters of the parameter instructions, depending on the Ident device used.
   - Param_RF600: If necessary, set the "LARGE_CONNECTION" parameter.
     You can find detailed information on this in the section "Parameter instructions (Page 29)".
   - Param_RF18xC: Connect the "DEVICE_PARAMETERS" parameter with the created parameter variable.
     You can find detailed information on this in the section "Creating a parameter variable (RF18xC/RF18xCI) (Page 20)".
   - Param_RFID181EIP: Provide the "MOBY_MODE" and "USER_MODE" parameters with the corresponding values.
     You can find detailed information on this in the section "Parameter instructions (Page 29)".
6. Start the parameter instruction by calling the "EXECUTE" parameter.

4.4.5 Executing the reset command

A reset command must be executed for each channel of a communication module or for each reading point of an RF600 reader after the variables have been created and before the actual work with the Ident instructions. To execute a reset command, you can use the respective reset instructions or the "Ident_Profile" instruction with a "Write-Config" command. The following table shows the reset instructions compatible with the respective device families:

Table 4- 9 Compatible reset instructions

Ident device	**Channel**	**Reset_Reader**	**Reset_RF200 Reset_RF300 Reset_MV Reset_Univ**	**Ident-Profile (Write-Config)**
Reader RF600	CH01	✓	--	✓
	CH02	✓	--	✓
	CH03	✓	--	✓
	CH04	✓	--	✓
RF18xC/RF18xCI	CH01	--	✓	✓
	CH02	--	✓	✓
	CH03	--	✓	✓
	CH04	--	✓	✓
	CM_Configuration	✓	--	✓
RFID EIP181	CH01	--	✓	✓
	CH02	--	✓	✓

Procedure

Depending on whether you execute the reset command with a "Reset_*" instruction or the "Ident_Profile" instruction, the procedure differs. Both options are described below.

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

1. Insert a reset instruction from the "Controller Organizer" into your application program.
2. Create an instance variable.
3. Connect the "HW_CONNECT" parameter with the correct "HW_CONNECT" variable from the configuration variable.
   You can find detailed information on this in the section "Assignment table (Page 28)".
4. Connect the "MSG_READ" parameter with the created message variable for receiving.
   You can find detailed information on this in the section "Configuring a message variable for receiving a channel (Page 19)".
5. Connect the parameter "MSG_WRITE" to the created message variable for sending.
   You can find detailed information on this in the section "Configuring a message variable for sending a channel (Page 20)".
6. Select the relevant input word of the reader/CM for the "STATUS_WORD" parameter.
   You can find detailed information on this in the section "Assignment table (Page 28)".

7. Select the relevant output word of the reader/CM for the "CONTROL_WORD" parameter.
   You can find detailed information on this in the section "Assignment table (Page 28)".
8. Supply the instruction-specific reset parameters (e.g. "TAG_TYPE" for "Reset_RF300")
   You can find detailed information on this in the section "RESET instructions (Page 60)".
9. Start the reset instruction by calling the "EXECUTE" parameter.

Proceed as follows to create and execute a reset command with the "Ident_Profile" instruction

1. Insert the "Ident_Profile" instruction from the "Controller Organizer" into your application program.
2. Perform the steps 2 ... 7 as described in the previous instruction.
3. Create an array with 10 elements of the "IID_CMD_STRUCT" type (command variable).
4. Set a "Write-Config" command for the first command ("<command variable>.CMD[0]").
   You can find detailed information on this in the section "Commands (Page 76)".
5. Create an array of the type "SINT" for the parameter "IDENT_DATA_TXREF" (transmit data variable).
   In this array, you need to enter the data that you want to send to the Ident device. The array can consist of 10 ... 65535 elements.
6. Create an array of the type "SINT" for the parameter "IDENT_DATA_RXREF" (receive data variable).
   In this array, the instruction enters the result received from the Ident device. The array can consist of 10 ... 65535 elements.
7. If necessary, enter the corresponding reset data in the transmit data variable (from step 5).
   You can find detailed information on this in the section "Commands (Page 76)".
8. Select the command just created ("<command variable>.CMD[0]") and set the parameter "CMDSEL = 0".
9. Start the instruction by calling the "INIT" parameter.

4.4.6 Assignment table

In order to be able to use the Ident instructions, the "HW_CONNECT", "STATUS_WORD" and "CONTROL_WORD" parameters must be supplied for the respective channel or for the reading point. The assignment of these parameters is described in the following table:

Table 4- 10 Parameter assignment for the parameters "HW_CONNECT", "STATUS_WORD" and "CONTROL_WORD"

Channel [1)]	**HW_CONNECT**	**STATUS_WORD**	**CONTROL_WORD**
Channel 1	<Configuration variable> .CH01	<Device name> :I.Data[0]	<Device name> :O.Data[0]
Channel 2	<Configuration variable> .CH02	<Device name> :I.Data[1]	<Device name> :O.Data[1]
Channel 3	<Configuration variable> .CH03	<Device name>:I.Data[2]	<Device name> :O.Data[2]
Channel 4	<Configuration variable> .CH04	<Device name> :I.Data[3]	<Device name> :O.Data[3]
CM Configuration Module	<Configuration variable> .CM_Configuration	<Device name> :I.Data[4]	<Device name> :O.Data[4]

1) The number of available channels depends on the Ident device and device versions used.

Additional parameter assignments

By using the cyclic input words, you can also query information on the states of the digital inputs/outputs as well as the reader diagnostics.

You can find detailed information on the digital inputs/outputs in the section "Digital inputs/outputs (Page 97)", on the reader diagnostics in the section "Structure of the input word for the reader diagnostics (Page 114)".

4.5 Programming instructions

4.5.1 Parameter instructions

The configuration data is sent to the Ident system using the "Param_RF600", "Param_RF18xC" or "Param_RFID181EIP" instruction. 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.

4.5.1.1 Param_RF600

You can use the "Param_RF600" instruction to transfer the configuration data to an RF600 reader.

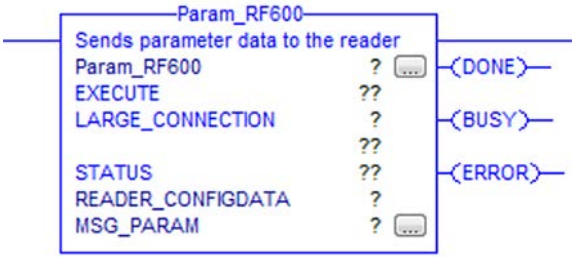


Figure 4-2 "Param_RF600" instruction

Table 4- 11 Explanation for the instruction "Param_RF600"

Parameter	Data type	Default values	Description
LARGE_ CONNECTION	BOOL	0	With this variable, the telegram length of the Ethernet/IP telegrams can be changed: • 240 bytes = false (default) • 1035 bytes = true You can find detailed information on this parameter below this table.
READER_ CONFIGDATA	IID_DATA_ RF600	--	The configuration variable contains the buffer for the configuration data and the "HW_CONNECT" variables which are required for operation of the individual channels. You can find detailed information on creating variables in the section "Creating configuration variables (Page 17)".
MSG_PARAM	MESSAGE	--	The message variable transfers the configuration data from the buffer of the configuration variable to the Ident device. You can find detailed information on creating and configuring variables in the section "Configuring the message variable for configuration (Page 18)".

"LARGE_CONNECTION" parameter

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.

4.5.1.2 Param_RF18xC

You can use the "Param_RF18xC" instruction to transfer the configuration data to an RF18xC/RF18xCI communication module.

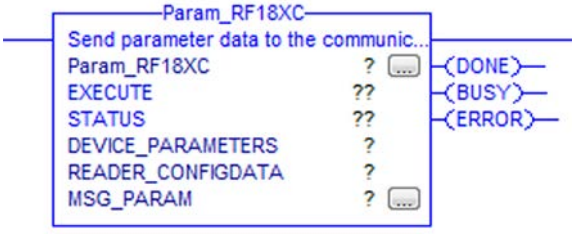


Figure 4-3 "Param_RF18xC" instruction

Table 4- 12 Explanation for the instruction "Param_RF18xC"

Parameter	Data type	Default values	Description
DEVICE_ PARAMETERS	IID_PARAMETER_ RF18XC	--	The channels of the communication module can be configured with this variable. You can find detailed information on the creation and configuration of variables in the section "Creating a parameter variable (RF18xC/RF18xCI) (Page 20)".
READER_ CONFIGDATA	IID_DATA_ RF18XC	--	The configuration variable contains the buffer for the configuration data and the "HW_CONNECT" variables which are required for operation of the individual channels. You can find detailed information on creating variables in the section "Creating configuration variables (Page 17)".
MSG_PARAM	MESSAGE	--	The message variable transfers the configuration data from the buffer of the configuration variable to the Ident device. You can find detailed information on the creation and configuration of variables in the section "Configuring the message variable for configuration (Page 18)".

4.5.1.3 Param_RFID181EIP

The "Param_RFID181EIP" instruction can be used to transfer the configuration data to an RFID181 EIP communication module.

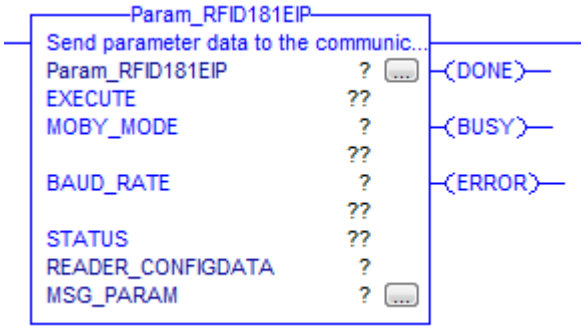


Figure 4-4 Instruction "Param_RFID181EIP"

Table 4- 13 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; MV400/MV500
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
READER_ CONFIGDATA	IID_DATA_ RFID181EIP	--	The configuration variable contains the buffer for the configuration data and the "HW_CONNECT" variables which are required for operation of the individual channels. You can find detailed information on creating variables in the section "Creating configuration variables (Page 17)".
MSG_PARAM	MESSAGE	--	The message variable transfers the configuration data from the buffer of the configuration variable to the Ident device. You can find detailed information on the creation and configuration of variables in the section "Configuring the message variable for configuration (Page 18)".

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 RF600 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".

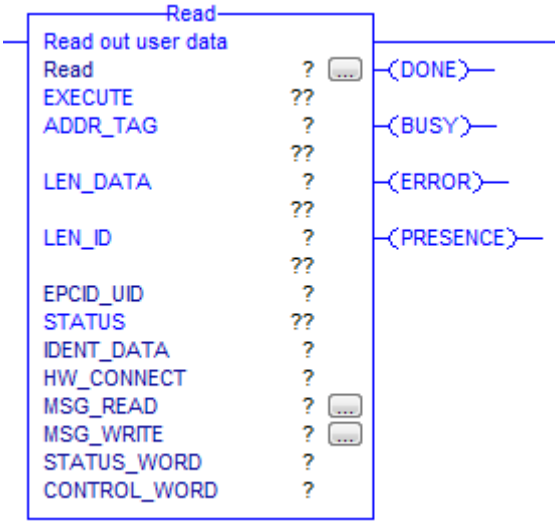


Figure 4-5 Instruction "Read"

Table 4- 14 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 98)". 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, RF61xR, 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")
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 RF600 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.

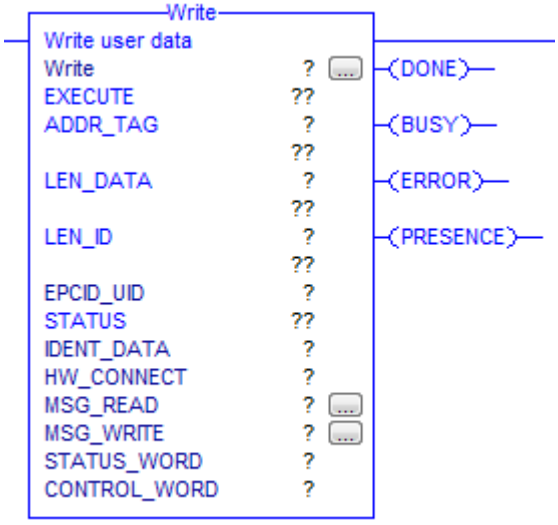


Figure 4-6 Instruction "Write"

Table 4- 15 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 98)". 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, RF61xR, 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")
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).

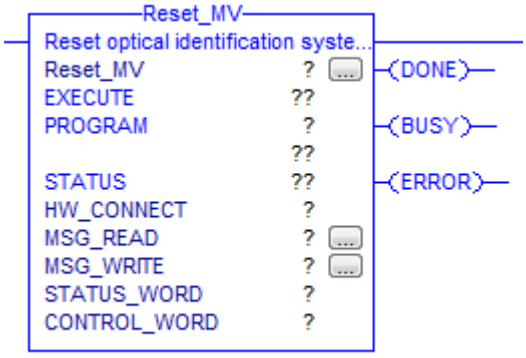


Figure 4-7 Instruction "Read_MV"

Table 4- 16 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.

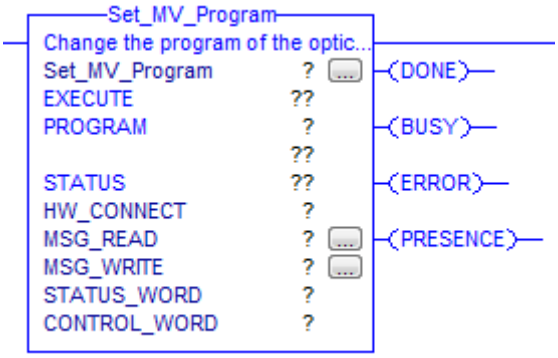


Figure 4-8 Instruction "Set_MV_Program"

Table 4- 17 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 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 allows you to optionally execute a series of concatenated commands with only one single call of the instruction. 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 95)".

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

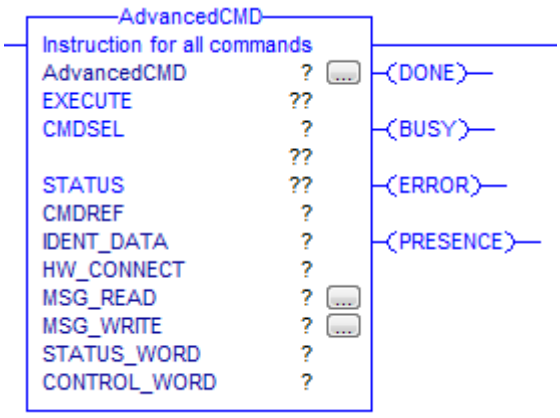


Figure 4-9 Instruction "AdvancedCMD"

Table 4- 18 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 72)" • "Command structure (Page 74)"
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]").

4.5.3.2 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 RF600 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. When multiple readers has the same configuration, you only have to save the configuration data once per reader type. A single backup can be copied to all identically configured readers during the download.

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. You can also use the device status to check the serial number to see if the device has been replaced.

The configuration data has the following structure:

Table 4- 19 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 RF600. It is always the same configuration data that is transferred. The configuration data of the RF18xC/RF18xCI communications module can be transferred exclusively via the "CM configuration_1" module.

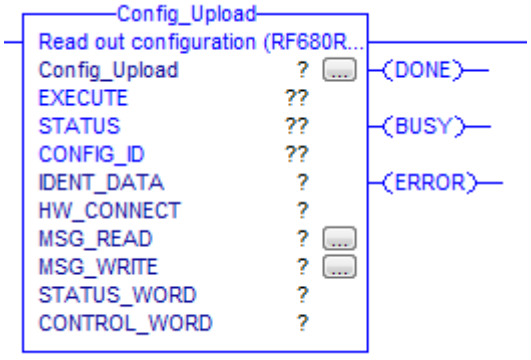


Figure 4-10 Instruction "Config_Upload"

Table 4- 20 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. We recommend a memory size of 4 KB for a standard configuration. 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.

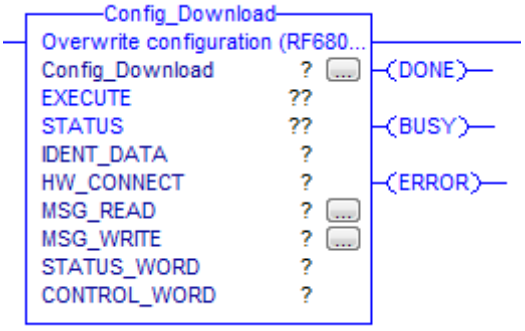


Figure 4-11 Instruction "Config_download"

Table 4- 21 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. We recommend a memory size of 4 KB for a standard configuration. 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]").

4.5.3.3 Inventory

The "Inventory" instruction activates the taking of inventories for RF600 readers (RF61xR/RF68xR). 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.

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).

You can specify the duration of the inventories using the "DURATION" and "DUR_UNIT" parameters.

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".

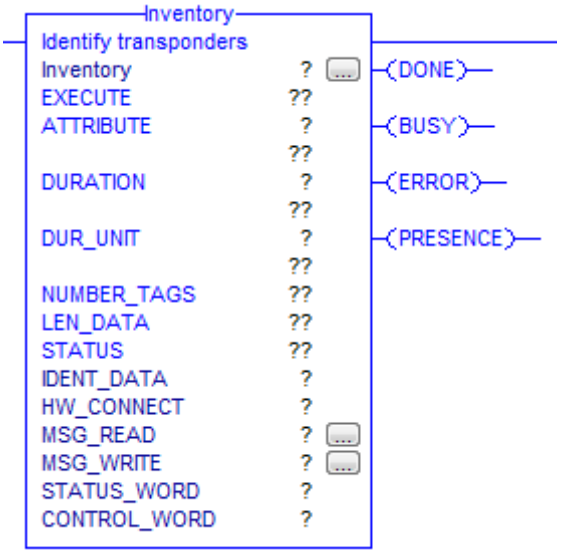


Figure 4-12 Instruction "Inventory"

Table 4- 22 Explanation for the instruction "Inventory"

Parameter	Data type	Default values	Description
ATTRIBUTE	SINT	0x00	Selecting the status mode: • RF600: 0x80, 0x81, 0x86, 0x87
DURATION	INT	0x00	Duration dependent on "DUR_UNIT" Period of time or number of inventories or number of "Observed" events Example: • 0x00 ≙ no inventory • 0x01 ≙ one inventory
DUR_UNIT	INT	0x00	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]").
NUMBER_TAGS	INT	0	Number of transponders in the antenna field
LEN_DATA	INT	0x00	Length of the valid data During command processing = 0

Results for RF600

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". Since ATTRIBUTE "0x86" and "0x87" do not return any data, their structure is not further executed.

Table 4- 23 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- 24 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; 0 = antenna 1; 1 = antenna 2; 2 = antenna 3; 3 = antenna 4
polarization	SINT	polarizatuin of antenna; 0 = undefined; 1 = circular; 2 = vertical linear; 3 = horizontal

Name	Type	Comment
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.

4.5.3.4 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.

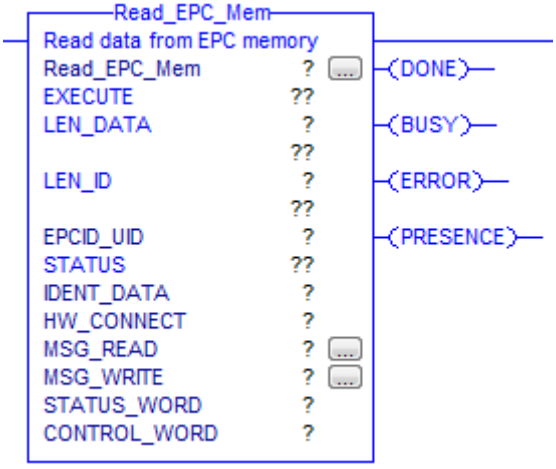


Figure 4-13 Instruction "Read_EPC_Mem"

Table 4- 25 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 (RF600)
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")
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.

4.5.3.5 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.

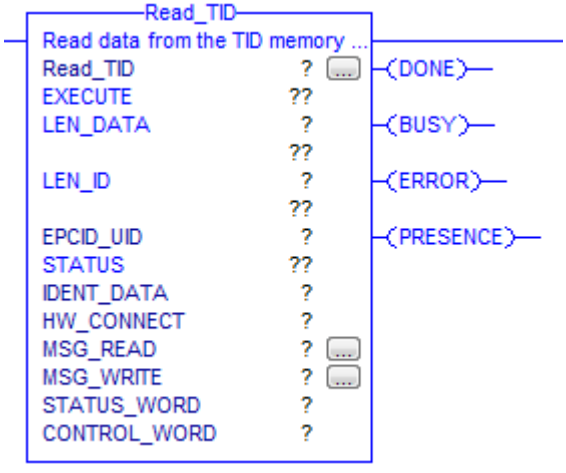


Figure 4-14 Instruction "Read_TID"

Table 4- 26 Explanation for the instruction "Read_TID"

Parameter	Data type	Default values	Description
LEN_DATA	INT	0x00	Length of the TID memory to be read
LEN_ID	SINT	0x00	Length of the EPC-ID/UID Default value: 0x00 ≙ unspecified single tag access (RF600)
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")
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]").

4.5.3.6 Read_UID

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

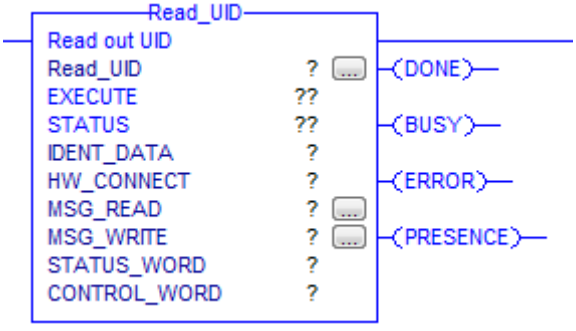


Figure 4-15 Instruction "Read_UID"

Table 4- 27 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]").

4.5.3.7 Set_Ant

Using the "Set_Ant" instruction, you can switch antennas on or off. The "Set_Ant_RF300" instruction can also be used for RF200.

Set_Ant_RF300

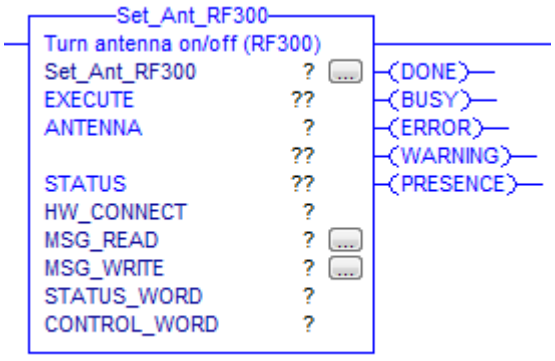


Figure 4-16 Instruction "Set_Ant_RF300"

Table 4- 28 Explanation for the instruction "Set_Ant_RF300"

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

4.5.3.8 Set_Param

Using the "Set_Param" instruction, you can change UHF parameters (e.g. the antenna power) on an RF61xR/RF68xR, as well as the date/time parameters of an RF18xC/RF18xCI during runtime. Note that changes to the date/time parameters of an RF18xC/RF18xCI can only be made via the "CM_Configuration" module.

Note

Settings saved only temporarily

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

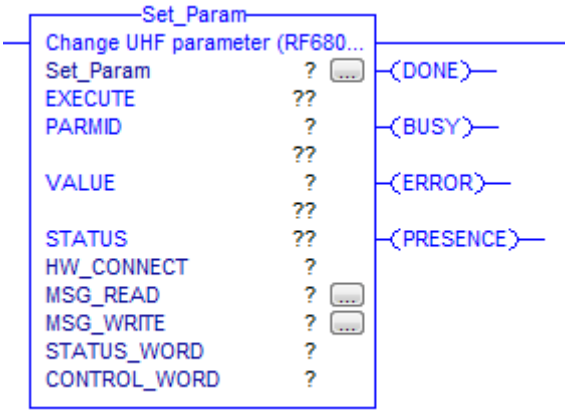


Figure 4-17 Instruction "Set_Param"

Table 4- 29 Explanation for the instruction "Set_Param"

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

Values for "PARMID" and "VALUE"

Depending on the Ident device used, the following parameters can be read out with the aid of this block.

Table 4- 30 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.

4.5.3.9 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.

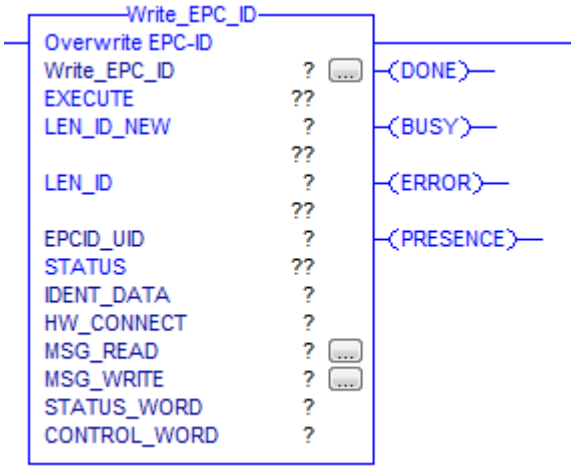


Figure 4-18 Instruction "Write_EPC_ID"

Table 4- 31 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]").

4.5.3.10 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.

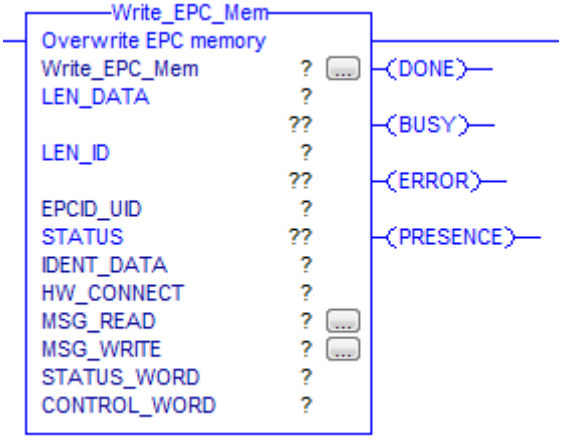


Figure 4-19 Instruction "Write_EPC_Mem"

Table 4- 32 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 (RF600)
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")
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]").

4.5.4 Status instructions

4.5.4.1 Reader_Status

The "Reader_Status" instruction reads status information from the reader or communication module (RF18xC/RF18xCI with the "CM configuration_1" module). For the various reader families, there are different status modes that you can select using the "ATTRIBUTE" parameter.

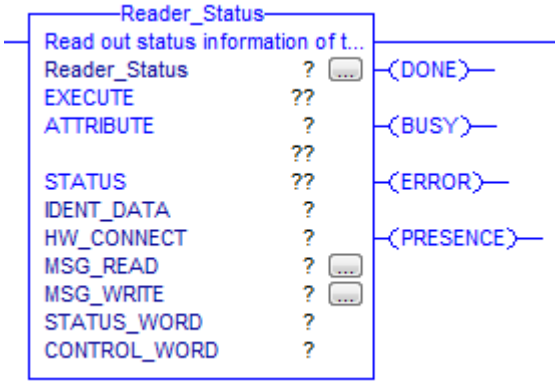


Figure 4-20 Instruction "Reader_Status"

Table 4- 33 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 • RF300 (Gen2): 0xEF • RF61xR, RF68xR: 0x89
			• RF18xC, RF18xCI: 0xA2
IDENT_DATA	SINT[10]	0	Result 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- 34 ATTRIBUTE "0x81" (data type "IID_READSTAT_81_RF2_3_U")

Name	Type	Comment
status info	SINT	Reader status mode
hardware	SINT	Type of hardware
hardware_version	INT	Version of hardware 0 = RF310R, RF340R, RF350R; 1 = RF380R, RF260R; 2 = RF310R (ISO), RF210R, RF220R; 3 = RF380R (ISO), RF240R; 4 = RF340R (ISO), RF350R (ISO), RF250R; 5 = RF310R (ISO); 8 = MV400/MV500; A = RF310R Gen2, RF290R; B = RF340R Gen2; C = RF350R Gen2; D = RF380R Gen2; M = RF280R RS422; N = RF280R RS232
loader version	INT	Version of loader
firmware	SINT	Type of firmware
firmware version HB	SINT	Version of firmware
firmware version LB	SINT	Version of firmware
driver	SINT	Type of driver
driver version	INT	Version of driver
interface	SINT	Type of interface 0x01 = RS422; 0x02 = RS232
baud	SINT	Baudrate 0x01 = 19.2 kBd; 0x03 = 57.6 kBd; 0x05 = 115.2 kBd
reserved1	SINT	Reserved
reserved2	SINT	Reserved
reserved3	SINT	Reserved
distance limiting SLG	SINT	Distance limiting of Reader
multitag SLG	SINT	Multitag Reader
field ON control SLG	SINT	Field ON control
field ON time SLG	SINT	Field On time
sync SLG	SINT	Synchronization with Reader
status_ant	SINT	Status of antenne 0x01 = on; 0x02 = off
stand by	SINT	Time of standby after command
MDS control	SINT	Presence mode

Table 4- 35 ATTRIBUTE "0x86" (data type "IID_READSTAT_86_RF300")

Name	Type	Comment
status info	SINT	Reader 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]	Reserved

Table 4- 36 ATTRIBUTE "0x6F" (data type "IID_READSTAT_EF_RF300G2")

Name	Type	Comment
status_info	SINT	Reader status mode
hardware	SINT	Type of hardware
hardware_version	INT	Version of hardware 0 = RF310R, RF340R, RF350R; 1 = RF380R, RF260R; 2 = RF310R (ISO), RF210R, RF220R; 3 = RF380R (ISO), RF240R; 4 = RF340R (ISO), RF350R (ISO), RF250R; 5 = RF310R (ISO); A = RF310R Gen2, RF290R; B = RF340R Gen2; C = RF350R Gen2; M = RF280R RS422; N = RF280R RS232
reserved	INT	Reserved
loader	SINT	Type of loader
loader version 1	SINT	Version of loader
loader version 2	SINT	Version of loader
loader version 3	SINT	Version of loader
loader version 4	SINT	Version of loader
firmware	SINT	Type of firmware
firmware version 1	SINT	Version of firmware
firmware version 2	SINT	Version of firmware
firmware version 3	SINT	Version of firmware
firmware version 4	SINT	Version of firmware
reserved1	DINT	Reserved
driver	SINT	Type of driver
driver version 1	SINT	Version of driver
driver version 2	SINT	Version of driver
reserved2	INT	Reserved

Table 4- 37 ATTRIBUTE "0xA2" (data type "IID_CM_STATUS_A2") compatible with RF18xC/RF18xCI

Name	Type	Comment
status_info	SINT	CM device status
hardware version	SINT	Version of hardware
firmware version	SINT[4]	Version of firmware
config ID	DINT	Config ID as unix timestamp
serial number	SINT[16]	Serial number
device_type	SINT	Type of device 0x00 = RF185C; 0x01 = RF186C; 0x02 = RF188C; 0x03 = RF186CI; 0x04 = RF188CI
upload request	SINT	Upload request for configuration data
active_connections	SINT	Active connections 0x00 = PLC; 0x01 = OPC UA; 0x02 = XML
io_mode	SINT	Mode of in-/outputs 0x00 = Disabled; 0x01 = Single Input; 0x02 = Single Output; 0x03 = Support of IO link devices/hubs
low_voltage	SINT	Low voltage warning 0x00 = OK; 0x01 = low voltage
reserved0	SINT	Reserved
reserved1	SINT	Reserved
number serial interfaces	SINT	Number of serial interfaces supported by CM
serial_interface_data	IID IN SERIAL INTERFACE[4]	
connection_status	SINT	Status of communication between reader and CM 0x00 = down; 0x01 = up

Name	Type	Comment
reader_type	SINT	Type of connected reader 0 = RF310R, RF340R, RF350R; 1 = RF380R, RF260R; 2 = RF310R (ISO), RF210R, RF220R; 3 = RF380R (ISO), RF240R; 4 = RF340R (ISO), RF350R (ISO), RF250R; 5 = RF310R (ISO); 8 = MV400/MV500; A = RF310R Gen2, RF290R; B = RF340R Gen2; C = RF350R Gen2; D = RF380R Gen2; M = RF280R RS422; N = RF280R RS232
hardware_error	SINT	Hardware error
reserved	SINT	Reserved

Table 4- 38 ATTRIBUTE "0x89" (data type "IID_READSTAT_89_RF600") compatible with RRF61xR/F68xR

Name	Type	Comment
status info	SINT	Reader 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

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.

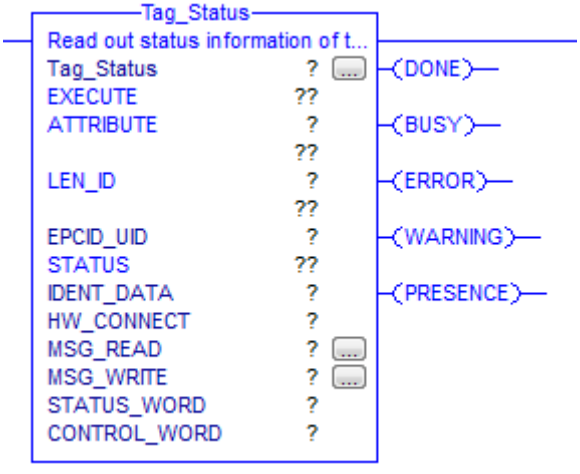


Figure 4-21 Instruction "Tag_Status"

Table 4- 39 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]
LEN_ID	SINT	0x00	Length of the EPC-ID/UID Default value: 0x00 ≙ unspecified single tag access (RF600)
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")
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- 40 ATTRIBUTE "0x04" ("IID_TAG_STATUS_04_RF300" data type)

Name	Type	Comment
reserved	SINT	
status info	SINT	Tag status mode
UID	SINT[8]	
MDS_type	SINT	Type of Tag 0x01 = Tag without FRAM; 0x02 = Tag with FRAM 8K; 0x03 = Tag with FRAM 32K; 0x04 = Tag with FRAM 64K; 0x05 = Tag with FRAM 128K; 0x06 = Tag with FRAM 256K
Lock state	SINT	Write Protection Status EEPROM
Reserved1	SINT[6]	

Table 4- 41 ATTRIBUTE "0x82" ("IID_TAG_STATUS_82_RF300" data type)

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

Table 4- 42 ATTRIBUTE "0x83" ("IID_TAG_STATUS_83_ISO" data type)

Name	Type	Comment
reserved	SINT	
status info	SINT	Tag status mode
UID	SINT[8]	
MDS_Type	BYTE	Type of Tag 0x00 = unknown; 0x03 = ISO: MDS D3xx, Infineon; 0x04 = ISO: MDS D4xx, Fujitsu; 0x05 = ISO: MDS D1xx, NXP; 0x06 = ISO: MDS D2xx, TI; 0x07 = ISO: MDS D261, STM; 0x08 = ISO: MDS D5xx, Fujitsu - 8k; 0x11 = RF300 - 0k; 0x12 = RF300 - 8k; 0x13 = RF300 - 32k; 0x14 = RF300 - 64k; 0x21 = MIFARE Classic - 1k, NXP; 0x22 = MIFARE Classic 1k, In-fineon; 0x23 = MIFARE Classic 4k, NXP; 0x41 (A) = P2P RF310R Gen2; 0x42 (B) = P2P RF340R Gen2; 0x43 (C) = P2P RF350R Gen2; 0x44 (D) = P2P RF380R Gen2
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

4.5.5 RESET instructions

The reset instructions described in this section are required if you want to operate the 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_Reader

Using the "Reset_Reader" instruction, you can reset the RF600 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_Reader" instruction does not have any device-specific parameters and is executed using the "EXECUTE" parameter. This instruction is also required to reset the "CM_Configuration" module of a RF18xC/RF18xCI communication module.

You can use the "Reset_Reader" instruction to interrupt any active Ident instruction at any time. The instructions are then ended with "DONE = true" and "ERROR = false".

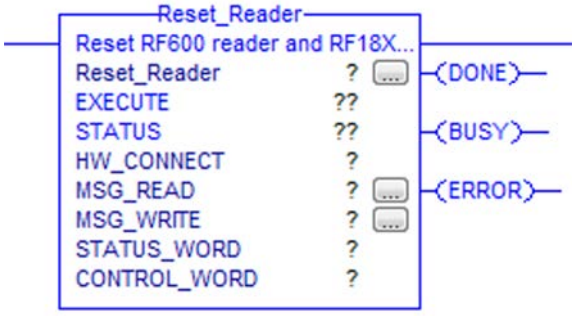


Figure 4-22 "Reset_Reader" instruction

4.5.5.2 Reset_MV

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

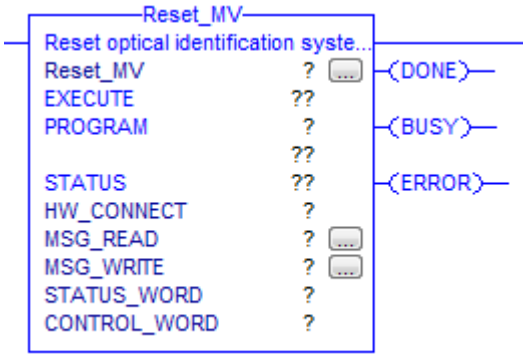


Figure 4-23 Instruction "Reset_MV"

Table 4- 43 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

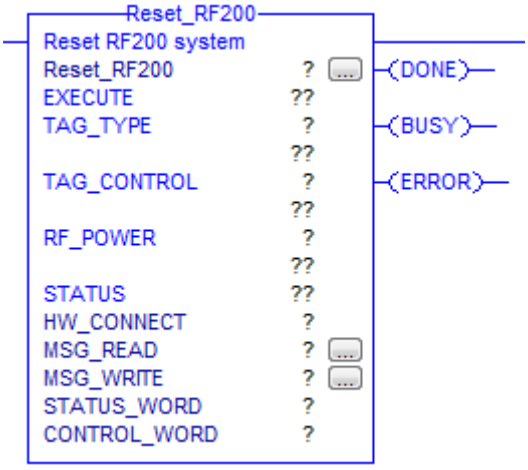


Figure 4-24 Instruction "Reset_RF200"

Table 4- 44 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

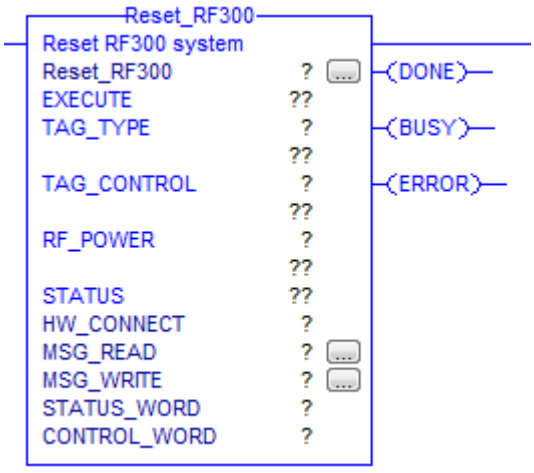


Figure 4-25 Instruction "Reset_RF300"

Table 4- 45 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_Univ

The "Reset_Univ" instruction is a universal reset statement that can be used to reset identification systems.

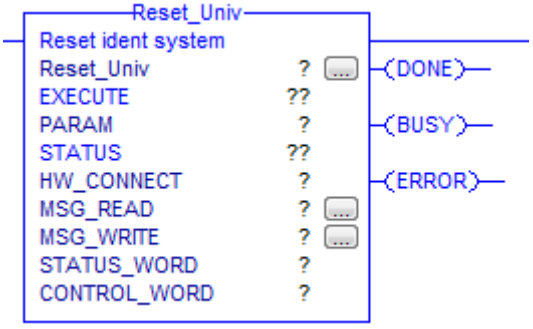


Figure 4-26 Instruction "RESET_Univ"

Table 4- 46 Explanation for the instruction "RESET_Univ"

Parameter	Data type	Default values	Description
PARAM	SINT [16]		Data for reset frame You can find information on this in the section "Connection of serial devices via Freeport protocol (Page 119)". The data to be set here can be made available by Support when necessary for special settings.

Table 4- 47 Structure of the "PARAM" parameter

Byte	1	2...5	6	7...8	9	10	11	12	13...14	15	16
Value (RF200, RF300, MOBY U/D)	0x04	0x00	0x0A	0x00	scan-ning_ time	param	option_1	distance_ limiting	No. of tran-sponders	field_on_ control	field_on_ time
Value (Free-port)	0x04	0x00	0x0A	0x00	0x00	0x00	option_1	0x00	0x00	0x00	0x00

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 instructions (Page 29)" 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.

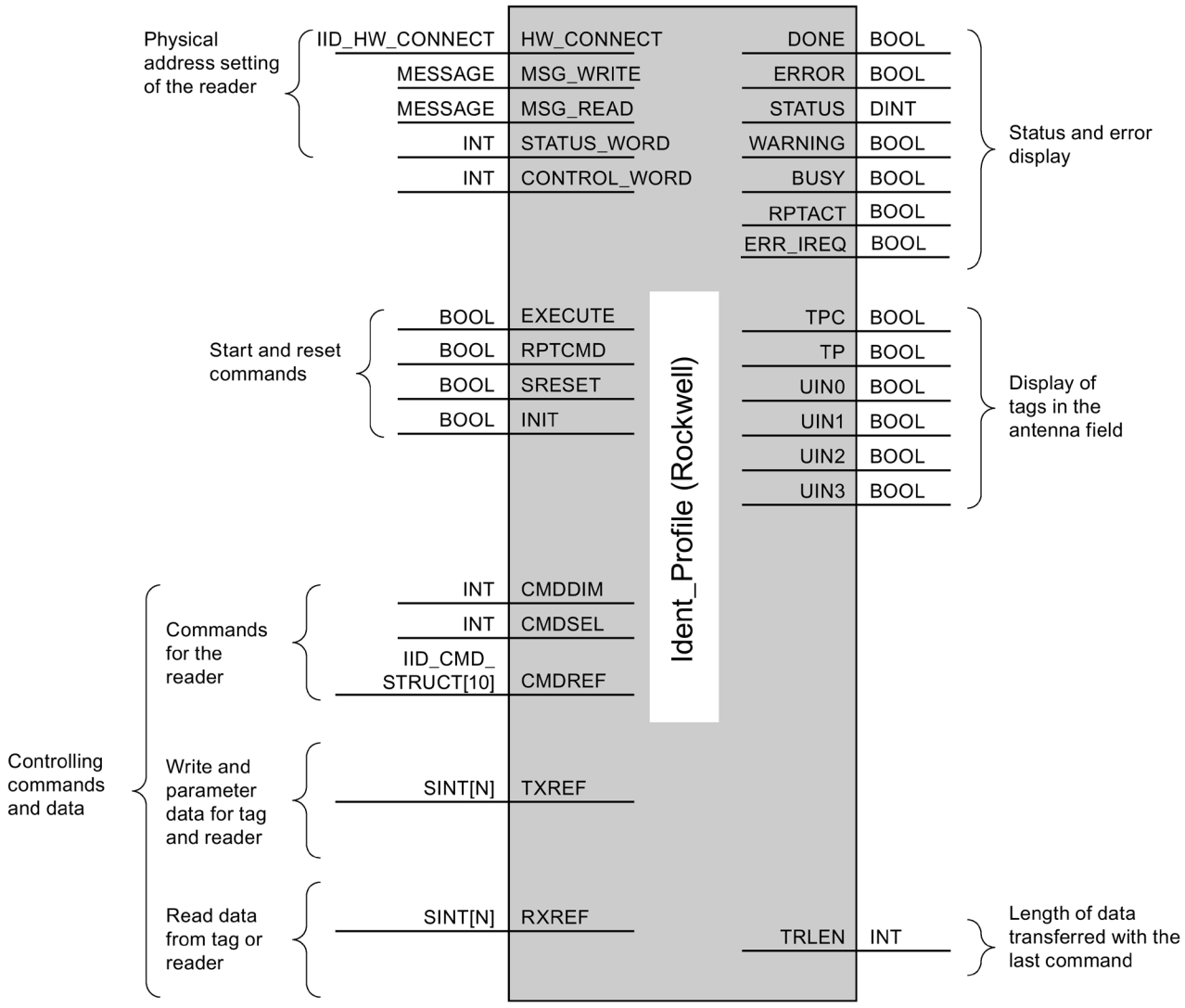


Figure 4-27 The input parameters of the Ident profile

Note

Working with multiple channels

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

Interface description

Table 4- 48 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 instructions. 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 the communication 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"; 0 ⇒ 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 72)".
TXREF	SINT[n]	--	Reference to global memory area for send data. The memory area can be shared with other instances of the instruction. The value "n" of the individual instructions is variable and can range in size from 10 bytes ... 32 KB.
RXREF	SINT[n]	--	Reference to global memory area for receive data. The memory area can be shared with other instances of the instruction. The value "n" of the individual instructions is variable and can range in size from 10 bytes ... 32 KB.

Table 4- 49 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 "Diagnostics and 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 instruction processes 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

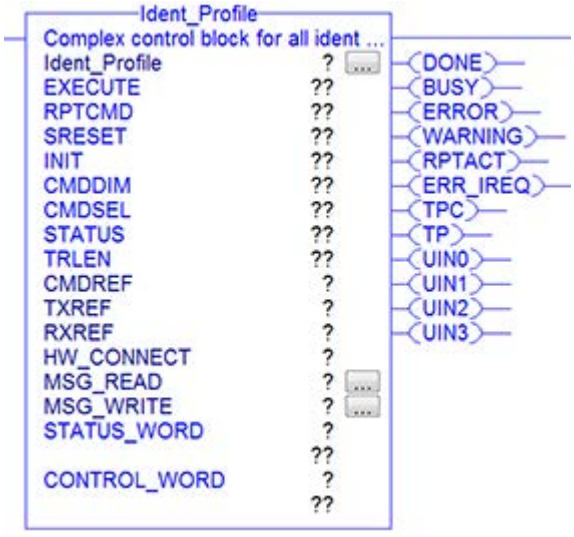


Figure 4-28 Instruction "Ident_Profile"

You must assign the following parameters of the instruction:

- Ident_Profile
- CMDREF
- IDENT_DATA_TXREF
- IDENT_DATA_RXREF
- HW_CONNECT

- MSG_READ
- MSG_WRITE
- STATUS_WORD
- CONTROL_WORD

Example of parameter assignment

Ident_Profile

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. You can find detailed information on assigning parameters in the section "Assignment table (Page 28)".

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...
- Ouput 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 4-29 Automatically created input and output variables

MSG_WRITE and MSG_READ

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

Use the variables as described in the section "Creating a message variable (Page 18)".

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 64)".

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]

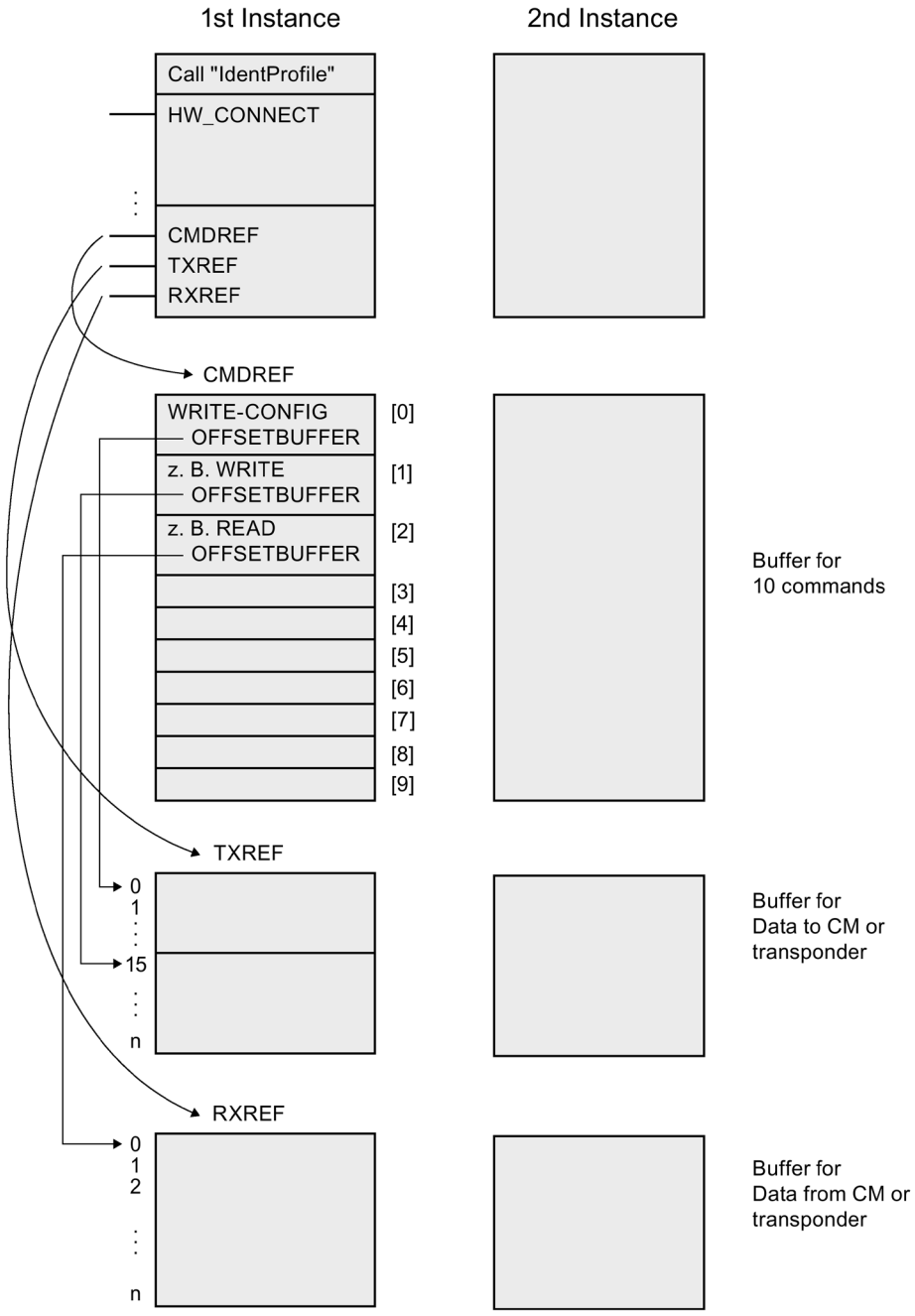


Figure 4-30 Data structure example of the Ident profile

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" instruction.

Table 4- 50 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	Writes a new EPC-ID to the transponder.
KILL-TAG	0x6A	‘j‘	EPCID_UID, LEN_ID, PSWD	The transponder is permanently deactivated.
LOCK-TAG-BANK	0x79	‘y‘	EPCID_UID, LEN_ID, PSWD, ACTION, MASK	The corresponding memory area of the transponder is blocked as specified.
EDIT-BLACKLIST	0x7A	‘z‘	EPCID_UID, LEN_ID, MODE	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	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- 51 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 72)".)
ATTRIBUTES	SINT	0	Sub command name for several commands (e.g. "READER-STATUS", "INVENTORY", etc.)
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, 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")
LEN_DATA	DINT	0	Amount of data to be read/written in bytes
ADR_TAG	DINT	0	Physical start address on the transponder
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
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.

Parameter		Data type	Default value	Description
EXT_UHF		STRUCT	--	Structure for additional parameters
	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- 52 PHYSICAL-READ

Parameter	Value / description
CMD	0x70
OFFSETBUFFER	Offset in the "RXREF" receive buffer
LEN_DATA	Length of received data
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")
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- 53 PHYSICAL-WRITE

Parameter	Value / description
CMD	0x71
OFFSETBUFFER	Offset in the "TXREF" send buffer
LEN_DATA	Length of the data to be written
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")

Parameter	Value / description
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- 54 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 • RF300 Gen2: 0xEF • RF600: 0x89
	• RF18xC/RF18xCI, RF166C: 0xA2
RXREF	Received status data You will find the data structure of the status modes in the section "Reader_Status (Page 54)".

Table 4- 55 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
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")

Parameter	Value / description
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 57)".

Table 4- 56 INVENTORY

Parameter	Value / description
CMD	0x69
OFFSETBUFFER	Offset in the "RXREF" receive buffer
ATTRIBUTES	Identifier of the status modes / possible entries: • 0x80 ≙ inventory with brief transponder information • 0x81 ≙ inventory with a lot of transponder information • 0x86 ≙ Presence mode on • 0x87 ≙ Presence mode off
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
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	Data received You will find the data structure of the status modes in the section "Inventory (Page 41)".

Table 4- 57 FORMAT (not with RF200 and RF600)

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")

Parameter	Value / description
LEN_ID	Length of the EPC-ID/UID
TXREF	Parameter data to be written

Table 4- 58 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- 59 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- 60 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
752 bytes	MOBY E	EEPROM	02EF	approx. 1.5 s

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

Table 4- 61 PUT (not with RF600)

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- 62 Data structure of the PUT command

Put_SET_ANT		Switches the antenna of the reader off and on. 1: 'N' 2: 'A' 3: Mode
	Mode	RF200/RF300: • 0x01 ≙ antenna on • 0x02 ≙ antenna off
	Length	3

Table 4- 63 WRITE-ID (only with RF600)

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- 64 KILL-TAG (only for RF600)

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- 65 LOCK-TAG-BANK (only for RF600)

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- 66 GET-BLACKLIST (only for RF600)

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- 67 GET-BLACKLIST (only for RF600)

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

Table 4- 68 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- 69 READ-CONFIG

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

Table 4- 70 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 RF600 and RF18xC/RF18xCI:

- 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:

  This command allows you to change UHF parameters (e.g. the antenna power) on an RF61xR/RF68xR, as well as the date/time parameters of an RF166C or RF18xC/RF18xCI during runtime.

Byte	Name
0 ... 3	Parameter identification "PARMID" (4 bytes)
4 ... 7	Parameter value "VALUE" (4 bytes)

You can find possible values for parameter identification and value in section "Set_Param (Page 49)".
LEN_DATA = 0x08

For RF200, RF300 and Freeport

Table 4- 71 For RF200, RF300 and Freeport with "CONFIG = 0x03".

Byte	1	2...5	6 [1]	7...8	9	10	11	12	13...14	15	16
Value (RF200, RF300)	0x04	0x00	0x0A 0x05	0x00	scan-ning_ time	param	option_1	distance_ limiting	No. of tran-sponders	field_on_ control	field_on_ time
Value (Free-port)	0x04	0x00	0x0A	0x00	0x00	0x00	option_1	0x00	0x00	0x00	0x00

[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.

The parameter values of bytes 9 ... 16 are described below.

Table 4- 72 Parameter values of bytes 9 ... 16

Byte	Value	RFID system	Meaning	
Byte 9	scanning_time	RF200	Bits 0...7: Reserved (to be assigned the value "0").	
		RF300	Bit 0...1: Reserved (to be assigned the value "0"). Bit 2...4: Modulation depth • 0: 0% (default) • 1: 7% • 2: 10% • 3: 15% • 4: 20% • 5 25% • 6: 30% • 7: 100% Bit 5...7: Input attenuation • 0: Default (default attenuation stored in the firmware, depending on the reader type and air interface protocol) • 1: 0 dB • 2: 5 dB • 3: 10 dB • 4: 15 dB • 7: -5 dB	
Byte 10	param	RF200, RF300	Setting for the RFID mode and presence check	
			Bit	Description

Byte	Value	RFID system	Meaning	
			7 ... 5	This parameter switches the presence check on or off on the reader. Possible values: • 0: Operation without presence check[1)] Antenna is permanently switched on. • 1: Operation with presence check Antenna is permanently switched on. • 2: RF300 (2nd Generation): Operation without presence check and with semaphore procedure (RF300, ISO 14443 [MIFARE Classic, MOBY E]) Antenna is permanently switched on. • 3: RF300 (2nd Generation): Operation with presence check and with semaphore procedure (RF300, ISO 14443 [MIFARE Classic, MOBY E]) Antenna is permanently switched on. • 4: 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 • 6: RF300 (2nd Generation): Operation without presence check and with semaphore procedure (RF300, ISO 14443 [MIFARE Classic, MOBY E]) Antenna is switched off. The antenna is only switched on when one of the following commands is sent: Read, Write, Init, Tag-Status You can find information on the semaphore procedure in the product information "Input parameters for the RF300 system".
				[1)]Note on LED behavior with MOBY I migration: In "None present" mode, the LED of the reader stays blue until initial access (Read/Write/Init) to the transponder takes place. The LED flashes green as long as no other access is made or command is executed. As soon as a command is sent, the LED lights up green or orange, depending on whether or not a transponder is within the antenna range.

<table>
<tr><th>Byte</th><th>Value</th><th>RFID system</th><th colspan="3">Meaning</th></tr>
<tr><td rowspan="8"></td><td rowspan="8"></td><td rowspan="8"></td><td>4</td><td colspan="2">RF300, 2nd Generation
For RF300 2nd Generation, this parameter enables ECC mode for the transponder types RF300 and ISO 14443 (MIFARE Classic, MOBY E). The value "0" must be entered for all other systems.
Possible values:
• 0: ECC mode is switched off.
• 1: ECC mode is switched on.
In ECC mode, the reader can detect bit errors on the transponder with a high degree of probability. If possible, corrected data is returned during read access (the data on the transponder remains unchanged). When write access occurs, the affected data on the transponder is corrected - if possible. ECC mode can only be used for transponders that have been completely initialized with the ECC bit set and are thus ECC formatted. The transponder is divided into blocks of 16 bytes for this, whereby 14 bytes are reserved for user data and 2 bytes for ECC information (CRC). This leads to a reduction of the usable memory volume by approx. 1/8th.
If bit errors have been detected and corrected, the warning "0xF0FE0002" is issued in the "STATUS" output parameter. If the bit error could not be corrected, the error "0xE1FE0700" is output.
Note: Note that only the transponders of type RF300 and ISO 14443 (e.g. MOBY E) are supported in this mode.</td></tr>
<tr><td colspan="3">Values of bits 3 ... 0</td></tr>
<tr><td>Values of bits 3 ... 0</td><td>Operating mode</td><td>Description</td></tr>
<tr><td>0</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>1</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>5</td><td>RF200, RF300</td><td>Single tag mode</td></tr>
<tr><td colspan="3">Note: Note that after a parameter change the CM must be restarted.</td></tr>
<tr><td rowspan="5">Byte 11</td><td rowspan="5">option_1</td><td rowspan="3">RF200, RF300, Freeport</td><td colspan="3">This byte is bit-coded. As default, it has the value "B#16#0". With this byte, special controllers can be implemented on the CM/reader.</td></tr>
<tr><td>Bit</td><td colspan="2">Description</td></tr>
<tr><td>1</td><td colspan="2">Possible values:
• 0 = The flashing of the "ERR" LED in case of an error is not reset.
• 1 = Flashing of the "ERR" LED in the event of an error 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. With Freeport only the flashing of the "ERR" LED on the communication module is reset.</td></tr>
<tr><td rowspan="2">RF300 (2nd generation)</td><td colspan="3">The following values are permitted:</td></tr>
<tr><td>Value</td><td colspan="2">Description</td></tr>
</table>

<table>
<tr><th>Byte</th><th>Value</th><th>RFID system</th><th colspan="3">Meaning</th></tr>
<tr><td></td><td></td><td></td><td>0x00</td><td colspan="2">The flashing "ERR" LED in the event of an error can only be reset by switching off the power supply to the reader.</td></tr>
<tr><td></td><td></td><td></td><td>0x01</td><td colspan="2">The flashing of the "ERR" LED in the event of an error is reset after the successful execution of a command. This only affects communication errors (e.g. 0xE1FE0200, 0xE2FE0100); other errors are not reset with this.</td></tr>
<tr><td></td><td></td><td></td><td>0x02</td><td colspan="2">The flashing of the "ERR" LED in the event of an error is reset by a "init_run" or "WRITE-CONFIG" with "Init" (RESET).</td></tr>
<tr><td></td><td></td><td></td><td>0x03</td><td colspan="2">The flashing of the "ERR" LED in the event of an error is reset by a "init_run" or "WRITE-CONFIG" with "Init" (RESET) or after the successful execution of a command.</td></tr>
<tr><td rowspan="3">Byte 12</td><td rowspan="3">distance_limiting</td><td>RF200</td><td colspan="3">0x00 (reserved)</td></tr>
<tr><td>RF300 (RF350R, RF380R only)</td><td colspan="3">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%.</td></tr>
<tr><td>Bit 0...3: Transmit power
(Readers of the 1st generation)
The following values are permitted:
• 2: 0.5 W
• 3: 0.75 W
• 4: 1.0 W
• 5 1.25 W (default)
• 6: 1.5 W
• 7: 1.75 W
• 8: 2.0 W</td><td colspan="2">Bit 4...7: Antenna type
(RF350R readers of the 2nd generation)
To improve communication stability. The following values are permitted:
• 0: Unspecified (default)
• 1: ANT 1
• 2: ANT 3
• 3: ANT 3S
• 4: ANT 8
• 5 ANT 12
• 6: ANT 12 (6GT2398-1CC10 with integrated antenna connecting cable 0.6 m)
• 7: ANT 18
• 8: ANT 18 (6GT2398-1CA10 with integrated antenna connecting cable 0.6 m)
• 9: ANT 30</td></tr>
</table>

Byte	Value	RFID system	Meaning		
Bytes 13...14	No. of transponders	RF300	To be assigned the value "0x0001".		
Byte 15	field_on_control	RF200, RF300	To be assigned the value "0x00".		
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 operation is basically guaranteed with every ISO-compatible transponder. All ISO 15693 transponders (MDS D) specified in the system manual "SIMATIC RF300" support these ISO 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 (MDS D3xx; Infineon)	e.g. MDS D324, D339
			0x04	ISO 15693 (MDS D4xx, Fujitsu - 2 KB)	e.g. MDS D421, D422, D423, D424, D425, D426, D428, D460
			0x05	ISO 15693 (MDS D1xx, NXP)	e.g. MDS D100, D124, D126, D139, D150, D165
			0x06	ISO 15693 (MDS D2xx, TI)	e.g. MDS D200
			0x07	ISO 15693 (MDS D261, STM)	e.g. MDS D261
			0x08[1)]	ISO 15693 (MDS D5xx, Fujitsu - 8 kB)	e.g. MDS D521, D522, D524, D525, D528
			0x10[1)]	RF300 (RF3xxT)	For all transponders of the type "RF3xxT"
			0x20[1)]	ISO 14443 (MIFARE Classic MOBY E, E6xx)	e.g. MDS E600, E611, E623, E624

Byte	Value	RFID system	Meaning		
			0x2F[1)]	ISO 14443 (MIFARE Classic)	Open mode for changing the key material on the MIFARE Classic transponder. You can find information on open mode in the product information "Input parameters for the RF300 system".
			0x31[1)]	General mode (ISO, RF300, MOBY E)	Activation of the "General Mode" for processing the transponder types RF300, ISO 15693 and ISO 14443 (MIFARE Classic, MOBY E). With this setting, operation is basically guaranteed with every compatible transponder.
			[1)] Applies only to readers of the 2nd generation.		
			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.		

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- 73 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

Note that the communication reset without configuration data (CONFIG = 0x01) can only be used if you operate the reader without communications module via PROFINET IO. Reset without configuration data corresponds to the "INIT" command without program selection.

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

Table 4- 74 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- 75 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- 76 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- 77 Command data area "TXREF" command identifier 0x02 (activate read program number)

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

Table 4- 78 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- 79 Command data area "TXREF" command identifier 0x04 (activate read match string)

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

Table 4- 80 Command data area "TXREF" command identifier 0x05 (set DISA bit)

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

Table 4- 81 Command data area "TXREF" command identifier 0x06 (reset DISA bit)

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

Table 4- 82 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- 83 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- 84 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.

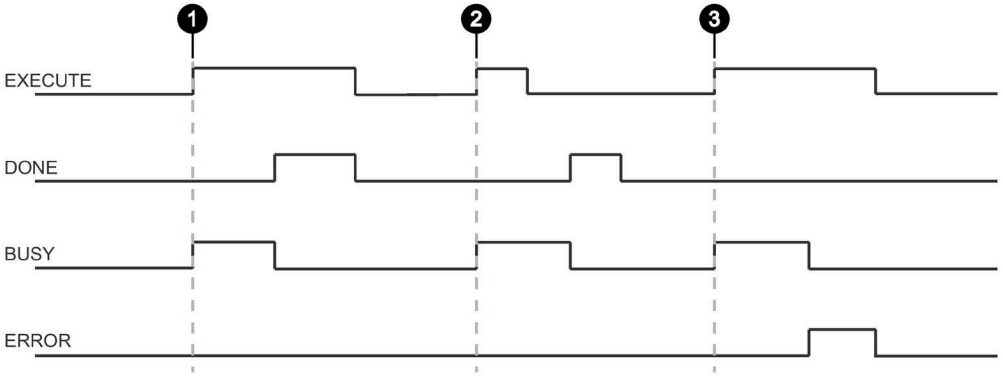
   The content of "CMDREF" = [0] is reserved for initialization. It is executed when the "INIT" input of the Ident profile is set and "CMDSEL" is = [0].
2. Transfer the data to be written to the send data buffer "TXBUF".
3. Select the previously written command (SINT[n]) with the "CMDSEL" parameter.
4. 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 4-31 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

You can send chained commands with the Ident profile as well as the "Advanced" instruction. 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 instructions, 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 instruction 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 RF600 readers as well as RF18xC/RF18xCI communication modules.

Overview of the commands

Table 4- 85 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.
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‘	RF600: Writes a new EPC-ID to the transponder.
KILL-TAG	0x6A	‘j‘	RF600: The transponder is permanently deactivated.
LOCK-TAG-BANK	0x79	‘y‘	RF600: Defines a password for transponder access.

Example of command structure

Table 4- 86 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

Readers RF615R, RF680R and RF685R

The 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 made 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 RSLogix 5000 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- 87 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.

Communications modules RF186CI and RF188CI

The CI device versions SIMATIC RF186CI and RF188CI have an I/O interface. You can use this interface either as a single digital input or output or for the connection of an IO-Link module (e.g. 6ES7143-5AH00-0BL0). The mode is set via the "IO_MODE" parameter in the parameter variable (see section "Creating a parameter variable (RF18xC/RF18xCI) (Page 20)").

- "Input" mode: The interface can be used as a single digital input.
- "Output" mode: The interface can be used as a single digital output.
- "IO-Link" mode: An IO-Link module with up to 8 digital inputs/outputs can be connected.

The digital inputs can be accessed via the cyclic input word "<Device name>:I.Data[5]" of an RF18xCI module. The digital outputs can be set via the cyclic output word "<Device name>:O.Data[5]" of an RF18xCI module.

You can find a detailed description of the individual modes in the "IO Function" section of the operating manual "SIMATIC RF185C, RF186C, RF188C, RF186CI, RF188CI (https://support.industry.siemens.com/cs/ww/en/ps/25532/man)".

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 following tables show the address space of the individual transponder parameters. The "ADDR_TAG" and "LEN_DATA" parameters must be assigned according to this address space.

Address spaces of the ISO 15693 transponder (MDS D)

Address spaces of the ISO 15693 transponder (MDS D) are illustrated below. These transponders can be used in the RF200, RF300 and MOBY D Ident systems.

The UID of the transponders is permanently addressed and can only be read out in full.

- Start address of the UID: 0xFFF0
- Length of the UID: 0x0008

Table 4- 88 Address spaces of the ISO 15693 transponder (MDS D)

User memory	Transponder type	Chip	Address area (hex, 16-bit)
112 bytes EEPROM	MDS D1xx	NXP I-Code SLI	0x0000 ... 0x006F
256 bytes EEPROM	MDS D2xx	Texas Inst. Tag-it HFI STM ST LRI 2K	0x0000 ... 0x00FF
992 bytes EEPROM	MDS D3xx	Infineon my-d	0x0000 ... 0x03DF
2000 bytes FRAM	MDS D4xx	Fujitsu MB89R118	0x0000 ... 0x07CF
8192 bytes FRAM	MDS D5xx	Fujitsu MB89R112	0x0000 ... 0x1FFF

Address space of the RF300 transponder (RF3xxT)

The address spaces of the RF300 transponder (RF3xxT) are illustrated below. These transponders can be used in the RF300 Ident System. Note that the EEPROM user memory for RF300 can be used either as R/W memory or as an OTP memory. You can find more information on this in the "SIMATIC RF300" system manual.

The UID of the transponders is permanently addressed and can only be read out in full.

- Start address of the UID: 0xFFF0
- Length of the UID: 0x0008

Table 4- 89 Address space of the RF300 transponder (RF3xxT)

User memory	Transponder type	Address area (hex, 16-bit)
20 bytes EEPROM	RF320T	0xFF00 ... 0xFF13
8 KB FRAM/EEPROM	RF340T / RF360T	FRAM: 0x0000 ... 0x1FFC
		EEPROM: 0xFF00 ... 0xFF13
32 KB FRAM/EEPROM	RF330T / RF340T / RF350T / RF360T / RF370T / RF380T	FRAM: 0x0000 ... 0x7FFC
		EEPROM: 0xFF00 ... 0xFF13
64 KB FRAM/EEPROM	RF370T	FRAM: 0x0000 ... 0xFEFC
		EEPROM: 0xFF00 ... 0xFF13

General notes on the meaning of the OTP memory for RF300 transponders

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.

Address mapping of the OTP memory on RF300 transponders

EEPROM memory (R/W) and OTP memory are only available once in the transponders.

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

Data can be read via the EEPROM address or the OTP address.

Table 4- 90 Address mapping of OTP memory

EEPROM memory (R/W)		OPT memory (written 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 "0xFF80" to "0xFF93" activates the write protection (OTP function) on the EEPROM user memory. This operation is not reversible. Activation of write protection must always take place in ascending order without gaps, starting at address "0xFF80".

Address spaces of the MIFARE transponder (MDS E)

The address spaces of the MIFARE transponders (MDS E) are illustrated below. These transponders can be used in the RF300 Ident System.

The UID of the transponders is permanently addressed and can only be read out in full.

- Start address of the UID: 0xFFF0
- Length of the UID: 0x0008

Table 4- 91 Address spaces of the MIFARE transponder (MDS E)

User memory	Transponder type	Address area (hex, 16-bit)
752 bytes EEPROM	MDS E600 / MDS E611 / MDS E623 / MDS E624	0x0000 ... 0x02EF

Address spaces of the RF600 transponders

With the RF61xR/RF650R/RF68xR readers, 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.

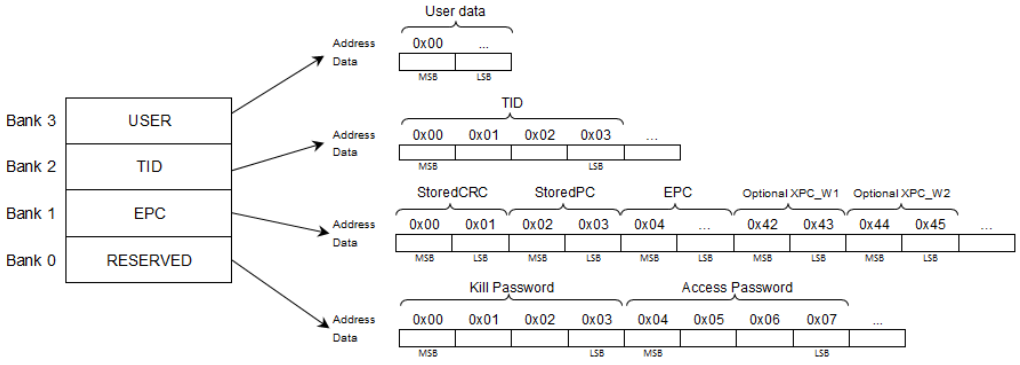


Figure 4-32 Memory configuration

Table 4- 92 Address spaces of the RF600 transponders

Chip type	Transponder	USER User memory	EPC		TID (read only)	RESERVED (passwords)
		Area / length	Area / length (default and max.)	Access	Area / length	Area / length
Alien Higgs 3	RF630L variants RF640L variants RF690L variants	16/64 bytes[1]	96 / 480 bits	read/ write	8 bytes	8 bytes
Alien Higgs 4	RF630L variants	16 bytes	96 / 128 bits	read/ write	24 bytes	8 bytes
Impinj Monza 4 QT[2]	RF630L variants RF620T RF625T	64 bytes	96 / 128 bits	read/ write	4 bytes	8 bytes
NXP G2iL	RF630L variants	0 bytes	96 / 128 bits	read/ write	8 bytes	8 bytes
NXP G2iM[3]	RF630L variants RF680L variants	64 bytes	96 / 256 bits	read/ write	26 bytes	8 bytes
NXP G2XM	RF630L variants RF680L variants RF610T RF630T RF640T	64 bytes	96 / 240 bits	read/ write	8 bytes	8 bytes
NXP UCODE 7xm-2k	RF630L variants RF642L variants RF682L variants RF645T	256 bytes	96 / 448 bits	read/ write	12 bytes	8 bytes
NXP UCODE DNA	RF682T	384 bytes	96 / 224 bits	read/ write	12 bytes	8 bytes

1) The EPC memory area of the Alien Higgs 3 chips can be increased at the cost of the user memory (USER). You can find additional information on this in the relevant transponder sections of the "SIMATIC RF600" system manual.

2) Uses User Memory Indicator (UMI).

3) The TID area contains a writable area of 112 bits that can be used when necessary.

All RF600 transponders and smart labels provide a Kill password and a Lock function.

Diagnostics and error messages

5

5.1 Error messages from the Ident profile / Ident instructions

5.1.1 Structure of the "STATUS" output parameter

An error condition in the Ident profile is always present if "ERROR = TRUE" appears in the output parameter. 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 3 (least significant byte)	Warnings In this byte, each bit has a separate meaning.
Byte 2	Error code
Byte 1	Error number 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 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 • F0 - Warnings

5.1.2 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**
0xE1FE0100	Memory of the transponder cannot be written to • Transponder memory is defective. • Transponder EEPROM was written too frequently and has reached the end of its service life. • Transponder is write protected (Memory Lock)
0xE1FE0200	Presence error The transponder has moved out of the transmission window of the reader. The command was executed only partially. Read command: "IDENT_DATA" has no valid data. Write command: The transponder that has just left the antenna field contains an incomplete data record. Possible causes: • Operating distance from reader to transponder is not being maintained. • Configuration error: The data record to be processed is too large (in dynamic mode). • With timeout: No transponder in the antenna field
0xE1FE0300	Address error The address area of the transponder has been exceeded. Possible causes: • Start address at command start is incorrectly set. • Transponder is not the correct type. • The area to be written to is write-protected. Freeport: Address ≠ "0x0000", "0x0002", "0xFFFF"
0xE1FE0400	Initialization error Transponder is unable to execute the initialization command. Possible causes: • Transponder is defective.
0xE1FE0500	The transponder memory is full.

Error message (hex)	**Description**
0xE1FE0600	Error in transponder memory • The transponder has never been written to or has lost the contents of its memory due to battery failure. Possible causes / action to be taken: – Replace transponder (if battery bit is set). – Re-initialize transponder. • RF300: Data storage was not completed correctly for the semaphore procedure.
0xE1FE0700	• Password error • RF300: ECC error Possible causes: – The bit error could not be corrected. – Transponder data has been lost (transponder defective). – The transponder was not initialized with the ECC driver. – The EEPROM memory of the transponder has reached its end of life and the data has been lost. – The transponder (with FRAM memory) has been stored too hot and has lost information. It is possible that the transponder can still be used after initialisation.
0xE1FE0800	The transponder in the antenna field does not have the expected UID / EPC ID or has no UID / EPC ID.
0xE1FE0900	The command is not supported by the transponder.
0xE1FE0A00	The transponder is read/write-protected.
0xE1FE8100	The transponder is not responding.
0xE1FE8200	The transponder password is incorrect. Access is denied.
0xE1FE8300	The verification of the written transponder data has failed. Possible causes: • Transponder is defective. • Transponder is in the limit area.
0xE1FE8400	General transponder error Contact Support if needed.
0xE1FE8500	The transponder has too little power to execute the command. Possible causes: • Transponder is in the limit area. • The transmission/radiated power of the reader or antenna is too low.

Error message (hex)	**Description**
0xE2FE0100	Field disturbance on reader Possible causes: • The reader is receiving interference pulses from the environment. – External interference field. The interference field can be detected with the "inductive field indicator" of the mobile reader. – The distance between two readers is too small and does not correspond to the configuration guidelines. – The connecting cable to the reader is defective or 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 errors 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 acknowledgment 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, i.e.: – The transponder is located exactly in the limit area of the transmission window. – The transponder is defective and consumes too much power. – The EEPROM transponder was incorrectly configured by "FORMAT". • RF600: – No ETSI channel is available. – An incorrect communication standard was selected in the "INIT" command. – The expert parameter is incorrect. – The performance check of the ETSI wireless profile is faulty. • Freeport: Negative acknowledgment or bad frame from the reader
0xE2FE0200	• More transponders are located in the transmission window than can be processed at the same time by the reader. • There is more than one transponder in the transmission window.
0xE2FE8100	There is no transponder with the required EPC ID/UID in the transmission window or there is no transponder at all in the antenna field.
0xE2FE8200	The requested data is not available.
0xE2FE8300	CRC error in reader-transponder communication.
0xE2FE8400	The selected antenna is not enabled.
0xE2FE8500	The selected frequency is not enabled.
0xE2FE8600	The carrier signal is not activated.
0xE2FE8700	There is more than one transponder in the transmission window.
0xE2FE8800	General radio protocol error

Error message (hex)	**Description**
0xE4FE0100	Short circuit or overload of the 24 V outputs • The reader is using too much current. • The reader cable is causing a short-circuit. Possible consequences: • The affected output is turned off • All outputs are turned off when total overload occurs • A reset can only be performed by turning the 24 V voltage off and on again • and then starting "Reset_Reader"
0xE4FE0300	• Error in the connection to the reader; the reader is not answering. – The cable between the communications module and reader is wired incorrectly or there is a cable break – The 24 V supply voltage is not connected or is not on or has failed briefly – Automatic cutout on the communications module has responded – Hardware defect – Another reader is in the vicinity and is active – Execute "INIT"/"WRITE-CONFIG" after correcting the error • The antenna of the reader is turned off. A tag command to the communications module was started in this status. – Turn on the antenna with the command "Antenna on/off." – The antenna is turned on (off) and has received an additional turn-on (turn-off) command • The mode in the "SET_ANT" command is unknown • The antenna on the reader is turned off or the antenna cable is defective
0xE4FE0400	• The buffer on the communications module or reader is not adequate to store the command temporarily. • Freeport: New data was received before existing data was retrieved. – Scan mode: The latest data is lost. With the "Write" command, the oldest acknowledgments are lost without an error message being generated.
0xE4FE0500	The buffer on the communications module or reader is not adequate to store the data temporarily.
0xE4FE0600	The command is not permitted in this status or is not supported. Possible causes: • The "INIT" command has been concatenated. • RF600: Command repetition was started without "Presence mode".

Error message (hex)	**Description**
0xE4FE0700	Startup message from reader/communication module The reader or communications module was off and has not yet received a "Reset_Reader" ("WRITE-CONFIG") command. Possible causes / action to be taken: • Execute the "INIT" command. • The same physical address in the "IID_HW_CONNECT" parameter is being used more than once. Check your "IID_HW_CONNECT" parameter assignments. • Check the connection to the reader. • The device has not yet been restarted after the change to the transmission speed.
0xE4FE8100	The specified tag field of the transponder is unknown.
0xE4FE8A00	General error
0xE4FE8B00	No or bad configuration data/parameters were transferred. Possible causes: • You are accessing a read point that is not configured.
0xE4FE8C00	• Communication error between Ident profile and communications module. Handshake error. Possible causes / action to be taken: – 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
0xE4FE8D00	• Firmware error Possible causes: The firmware update was not run completely. • Internal communication error of the communication module/reader Possible causes / action to be taken: – Connector contact problem on the communications module / reader – Hardware of the communication module/reader has a defect; → Send in communication module/reader for repair. – After eliminating the error, execute the "INIT" command. • Internal monitoring error of the communication module/reader Possible causes / action to be taken: – Program execution error on the communications module / reader – Switch supply voltage to communication module/reader off and on again. – After eliminating the error, execute the "INIT" command.
0xE4FE8E00	The current command was aborted by the "WRITE-CONFIG" ("INIT" or "SRESET") command for the bus connector was pulled. Possible causes: • Communication with the transponder was aborted by "INIT". • This error can only be reported if there is an "INIT" or "SRESET".

Error message (hex)	**Description**
0xE5FE0100	Incorrect sequence number order (SN) on the reader/communications module
0xE5FE0200	Incorrect sequence number order (SN) in the Ident profile Possible cause: User mode "Ident profile/RFID standard profile" is not set in the device configuration.
0xE5FE0400	Invalid data block number (DBN) on the reader/communications module
0xE5FE0500	Invalid data block number (DBN) in the Ident profile
0xE5FE0600	Invalid data block length (DBL) on the reader/communications module
0xE5FE0700	Invalid data block length (DBL) in the Ident profile
0xE5FE0800	The previous command is still active or the buffer is full. A new command was sent to the reader or communication module although the last command is still active. • The active command can only be aborted with "INIT". • Before a new command can be started, "DONE bit = 1" must be set (exception "INIT"). • Two Ident profile calls were assigned the same "HW_ID", "CM_CHANNEL" and "LADDR" parameter settings. • Two Ident profile calls are using the same pointer. • After eliminating the error, execute the "INIT" command. • When working with command repetition (e.g., fixed code transponder), no data is being fetched from the transponder. The data buffer on the reader/communications module has overflowed. Transponder data has been lost.
0xE5FE0900	The reader or communications module executes a hardware reset ("INIT_ACTIVE" set to "1"). "INIT" is expected by the Ident profile (bit 15 in the cyclic control word).
0xE5FE0A00	The "CMD" command code and the relevant acknowledgment do not match. This can be a software error or synchronization error that cannot occur in normal operation.
0xE5FE0B00	Incorrect sequence of acknowledgement frames (TDB / DBN)
0xE5FE0C00	Synchronization error (incorrect increment of "AC_H / AC_L" and "CC_H / CC_L" in the cyclic control word). "INIT" had to be executed.
0xE5FE8100	Communication error between reader and communication module Access denied
0xE5FE8200	Communication error between reader and communication module Resource is occupied
0xE5FE8300	Communication error between reader and communication module Functional error of the serial interface
0xE5FE8400	Communication error between reader and communication module Other faults/errors
0xE6FE0100	Unknown command The Ident profile sends an uninterpretable command to the reader. Possible causes: • The "AdvancedCmd" block was supplied with an incorrect "CMD". • The "CMD" input of the "AdvancedCmd" block was overwritten. Freeport: Incorrect command parameters or unauthorized command chaining (incorrect length specification)
0xE6FE0200	Invalid command index (CI)

Error message (hex)	**Description**
0xE6FE0300	• The communication module or reader was configured incorrectly. Possible causes / action to be taken: – Check the "INPUT" parameters in the Ident profile. – Check the hardware configuration. – "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. Possible causes / action to be taken: – 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 Possible causes / action to be taken: – Bad data in the "AdvancedCmd" or "IID_CMD_STRUCT" (e.g. "WRITE" command with length = 0); check "AdvancedCmd" or "IID_CMD_STRUCT" and execute "INIT". – Reader/communications module hardware defective: The reader or communications module receives bad data with "INIT". – Inconsistent length specifications in the command • Wrong reset block was selected. Possible causes / action to be taken: – Use the reset block suitable for the Ident device.
0xE6FE0400	Presence error A transponder has passed through the transmission window of a reader without being processed. • This error message is not reported immediately. Instead, the reader or communication module waits for the next write / read command. This command is replied to immediately with this error and the write/read command is not executed. The next command is executed normally again by the reader/communication module. • You can reset this error status using an "INIT".
0xE6FE0500	An error has occurred that makes a Reset_Reader ("WRITE-CONFIG" with "Config = 3") necessary. Possible causes / action to be taken: • The "WRITE-CONFIG" command is incorrect. • After eliminating the error, execute an "INIT". • Check the parameter "IID_HW_CONNECT".
0xE6FE8100	A parameter is missing.
0xE6FE8200	The parameter has an invalid format.
0xE6FE8300	The parameter type is invalid.
0xE6FE8400	Unknown parameter
0xE6FE8500	The command or the frame has an invalid format.
0xE6FE8600	The "Inventory" command failed.

Error message (hex)	**Description**
0xE6FE8700	Read access to the transponder has failed.
0xE6FE8800	Write access to the transponder has failed.
0xE6FE8900	Writing the EPC ID/UID on the transponder has failed.
0xE6FE8A00	Enabling write protection on the transponder has failed.
0xE6FE8B00	The "Kill" command failed.
0xE7FE0100	In this state, only the "Reset_*" command ("WRITE-CONFIG" with "CMDSEL =0" and "CMD = 0x78") is permitted.
0xE7FE0200	The command code "CMD" or the value in "CMDSEL" is not permitted. Possible cause / action to be taken: • The "CMDSEL" or "CMDDIM" parameter was set incorrectly. Check the parameters. • The value of the parameter "CMDSEL" must be between 0 and 9 and may never be higher than the value of the "CMDDIM" parameter (max. 10).
0xE7FE0300	The value of the "LEN_DATA" parameter of the command is too long and does not match the global data reserved within the send data buffer (TXBUF).
0xE7FE0400	The receive data buffer (RXBUF) or the send data buffer (TXBUF) is too small, the buffer created at TXBUF/RXBUF does not have the correct data type or the parameter "LEN_DATA" has 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".
0xE7FE0500	Only an "INIT" command is permitted as the next command. All other commands are rejected.
0xE7FE0600	Wrong data record index of an acyclic data record Permitted index is in the ranges "101 ... 108" and "-20401 ... -20418".
0xE7FE0700	The reader or communication module does not respond to "INIT" ("INIT_ACTIVE" is expected in the cyclic status message). The next steps: • Check the address parameter "LADDR".
0xE7FE0800	Timeout during "INIT" (60 seconds)
0xE7FE0900	Command repetition is not supported.
0xE7FE0A00	Error during the transfer of the PDU (Protocol Data Unit).
0xFxFExx00	An "FxFExxxx" error is identical to the corresponding "ExFExxxx" error (see "ExFExxxx" error). Byte 3 contains additional warning information.

5.1.3 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_STP
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.

5.1.4 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	The ECC mode has corrected a 1-bit error.
	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

5.2 Diagnostics and error messages of the "Param_RF18xC" instruction

5.2.1 Structure of the input word for the reader diagnostics

You can query the status of the connected readers using the cyclic input word "<Device name>:I.Data[6]". The number of bits depends on the number of connected readers:

- Bits 0 and 1: Reader 1
- Bits 2 and 3: Reader 2
- Bits 4 and 5: Reader 3
- Bits 6 and 7: Reader 4

Table 5- 4 Status of the connected readers

Least significant bit	Most significant bit	Description
0	0	= 0 Undetermined status. The connection may be interrupted or the parameter assignment is not current.
0	1	= 1 Reader not connected
1	0	= 2 The connected reader does not match the configured reader or the connected reader matches the configured reader but does not respond to the cable monitoring.
1	1	= 3 The connected reader matches the configured reader.

5.2.2 Error messages of the "Param_RF18xC" instruction

An error condition of the "Param_RF18xC" instruction is always present when "ERROR = TRUE" appears in the output parameter. 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- 5 Structure of the error codes

Byte	Description
Byte 0	0xC0: Error ID for errors of a "MESSAGE" instruction
Byte 1	Error code of the "MESSAGE" instruction 0x1F: Parameter error Information on other error codes is available in the help for the "MESSAGE" instructions in the Studio 5000 Logix Designer or in RSLogix 5000
Byte 2 Byte 3	Extended error code of the "MESSAGE" instruction Device-specific error messages in case of incorrect parameter assignment Information on other extended error codes is available in the help for the "MESSAGE" instructions in the Studio 5000 Logix Designer or in RSLogix 5000

The following errors can occur during parameter assignment of the module.

Table 5- 6 Error messages

Error messages	Description
0xC01F7001	The device ID ("Device" parameter) is invalid or does not match the device ID.
0xC01F7002	The frame length of the parameter frame is incorrect.
0xC01F7003	The structure of the parameter frame (number of sequences) is incorrect.
0xC01F7006	Invalid slot
0xC01F7101	An invalid module ID ("ReaderSlot01 > ModuleId" parameter) has been entered for the reader at channel 1.
0xC01F7201	An invalid module ID ("ReaderSlot02 > ModuleId" parameter) has been entered for the reader at channel 2.
0xC01F7301	An invalid module ID ("ReaderSlot03 > ModuleId" parameter) has been entered for the reader at channel 3.
0xC01F7401	An invalid module ID ("ReaderSlot04 > ModuleId" parameter) has been entered for the reader at channel 4.

Appendix

A

A.1 Internal status parameter

Status variables

Every Ident instruction 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 instruction has a time stamp and an error memory to be able to better understand previous problems.

These variables are stored in the respective instance variable of the instruction.

Table A- 1 Status variables in the instance variable

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 instruction.
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 = An instruction 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 Connection of serial devices via Freeport protocol

A.2.1 Compatible Ident devices

The following Ident devices can be operated using the communication modules via the Freeport protocol and the RS232 interface:

- SIMATIC MV320
- SIMATIC RF1040R/RF1070R

You can also connect any serial device to the communication module because the communication module can use the Freeport protocol to communicate with serial devices (e.g. bar code scanner, intelligent sensor or other automation components with serial connection). For communication modules with four serial interfaces, only two interfaces support RS232.

A.2.2 Functions and commands

You can communicate with the Ident devices using the add-on instructions. The "Write" command handles the sending of data and the "Read" command the receipt of data.

The following table shows the block-specific commands that you can execute with serial field devices.

Table A- 3 Overview of the commands

Command	Description
Read	Read data • Address "0x0000" The length of the valid data is entered in the first 2 bytes of the result. The user data is entered starting at the third byte. Max. net data 227 bytes; no error message if the length is exceeded • Address “0x0002” No length information in the result; max. net data 229 bytes
Write	Sending data The address must be specified as “0xFFFF”. Max. net data 194 bytes (without length information); no error message if the length is exceeded
Reset_Univ	Reset the communication between CM and device and empty the buffer. The 11th byte ("option_1") of parameter "PARAM" can be occupied by "0x00" or "0x02". The value "0x04" must be entered in the first byte, and "0x0A" in the 6th byte.

You can find detailed information on the commands and error messages of the RF1040R/RF1070R in the "Programming" section of the "SIMATIC RF1000" manual. The system job frames displayed must be stored in the data area for the "Write" command. The answer message frames are output in the data block following a "Read“ command.

Notes on the commands and functions

Reset_Univ

Communication with the CM is initialized with the Reset command. The Reset command does not have reader-specific parameters and triggers the deletion of the buffers.

After the "Reset_Univ" the communication module is ready for operation.

Read

When the "Read" command is started, the data that the CM has already received is transferred to the controller.

If the "Read" command is started before the CM has received data, the command remains active until the CM has received data that is then transferred to the controller. An active "Read" command can be canceled with a "Reset_*"/"Write_Config" command.

The presence bit ("TP") indicates that the CM has received new data and that this data is available for fetching with a "Read" command. After data transmission to the controller, the presence bit is reset unless new data has already been received.

When doing this, note the buffer size and the buffer overflow. If the receive buffer of the CM is full, existing data is overwritten by newly received data.

Service & Support

B

Industry Online Support

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

Apart from news, you will also find the following there:

- Project information: Manuals, FAQs, downloads, application examples etc.
- Contacts, Technical Forum
- The option to submit a support request:
  Link: (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: (https://www.automation.siemens.com/aspa_app/?ci=yes&lang=en)

"Industrial Identification" homepage

You can find the latest general information about our identification systems on the Internet at our Homepage (www.siemens.com/ident).

Online catalog and ordering system

The online catalog and the online ordering system can also be found on the Industry Mall home page (https://mall.industry.siemens.com).

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: (https://new.siemens.com/global/en/products/services/industry/sitrain.html)