SIEMENS

Configuration help • 05/2016

Industrial Identification with SIMATIC RF200 and ASM 456

SIMATIC RF / PROFIBUS / SIMATIC S7-300/400/1200/1500

https://support.industry.siemens.com/cs/ww/en/view/109483364

Warranty and Liability

Note

The application examples are not binding and do not claim to be complete with regard to configuration, equipment or any contingencies. The application examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for the correct operation of the described products. These application examples do not relieve you of the responsibility of safely and professionally using, installing, operating and servicing equipment. 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 and without prior notice. If there are any deviations between the recommendations provided in this application example and other Siemens publications – e.g. Catalogs – the contents of the other documents shall 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 fundamental contractual obligations (“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 Siemens AG.

Security information

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

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. Products and solutions from Siemens are only one part of such a concept.

The customer is responsible for preventing unauthorized access to the customer’s plants, systems, machines and networks. Systems, machines and components should be connected to the company network or the Internet only if and to the extent necessary and if appropriate protective action (e.g., use of firewalls and network segmentation) was taken.

In addition, Siemens’ recommendations regarding appropriate protective action should be followed. For more information about industrial security, visit http://www.siemens.com/industrialsecurity.

Siemens’ products and solutions undergo continuous development to make them even more secure. Siemens strongly recommends to perform updates as they become available and use only the latest product versions. Using versions that are out of date or no longer supported can increase the risk of cyber threats.

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

Table of Contents

1 Task

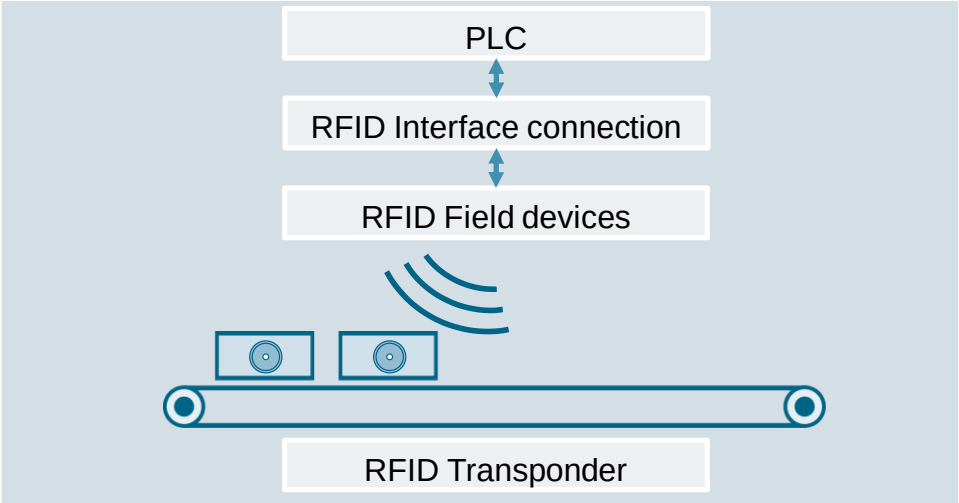
Introduction

To realize an automated identification of products, materials and material carriers for process and production systems, Siemens offers the SIMATIC Ident product portfolio as customized building blocks for this task.
This application example illustrates what should be observed while configuring RFID systems.

Overview of the automation task

The figure below provides an overview of the automation task.

Figure 1-1



For the integration of the SIMATIC Ident modules into a SIMATIC control system, there are a multitude of interface and communication modules.

For the operation of the SIMATIC RFID modules with a SIMATIC S7 control, suitable function blocks for a programming via a STEP 7 TIA Portal are available.

Depending on the preferred transmission mode, the appropriate communication module has to be selected.

Objective of this configuration help

In this application example, the RFID readers shall be connected to the control system via **PROFIBUS.** For this, the required communication module is the **ASM 456.** As RFID system for identification on the basis of transponders, the **SIMATIC RF200** will be used.

The application example shall support you in putting your SIMATIC Ident modules into operation. In this, the main focus will be on the prefabricated blocks of the “SIMATIC Ident” library. Additionally, the configuration of the SIMATIC Ident modules will be addressed.

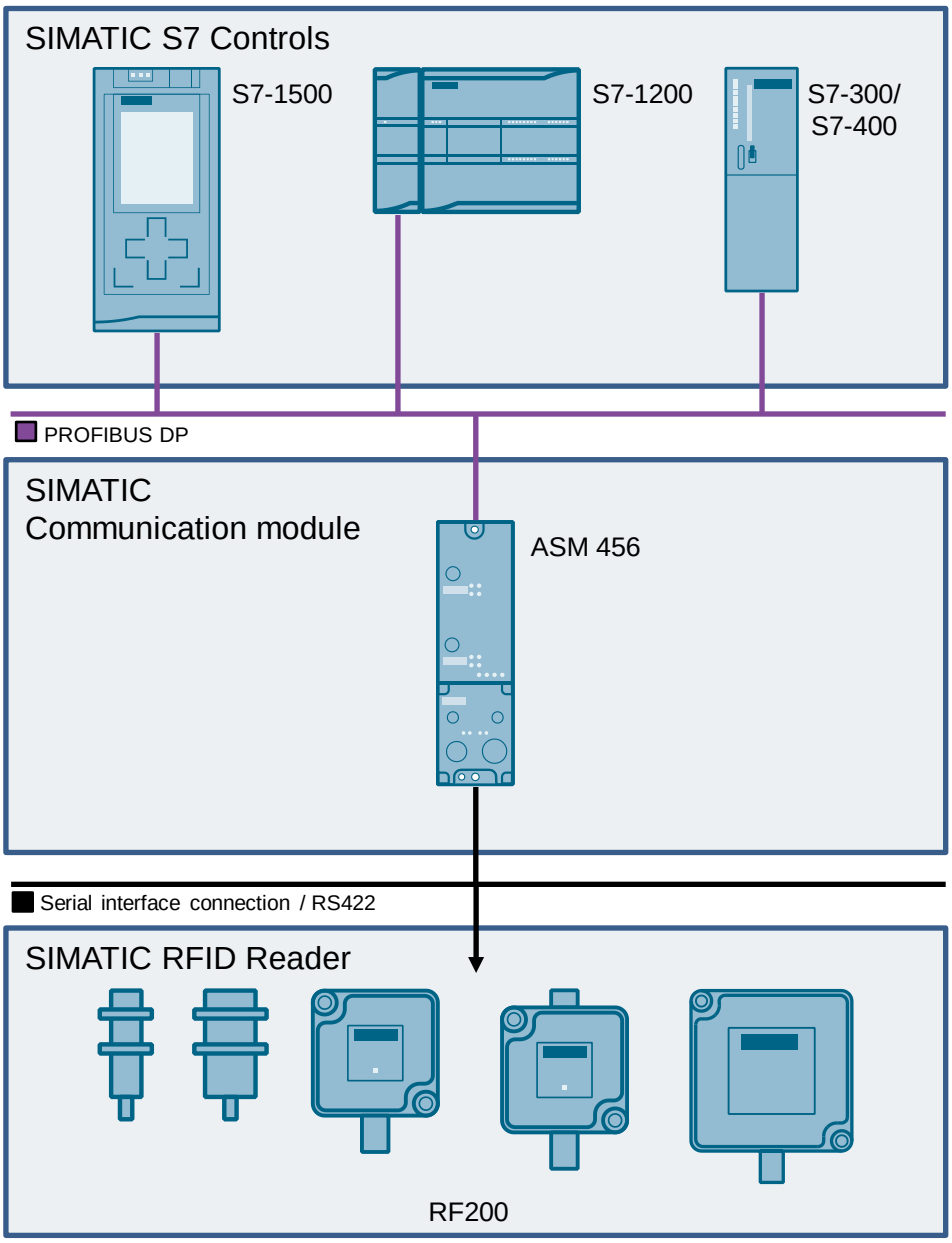
2 Solution

2.1 Overview

Schematic layout

The following figure displays the key components of the solution:

Figure 2-1



Advantages

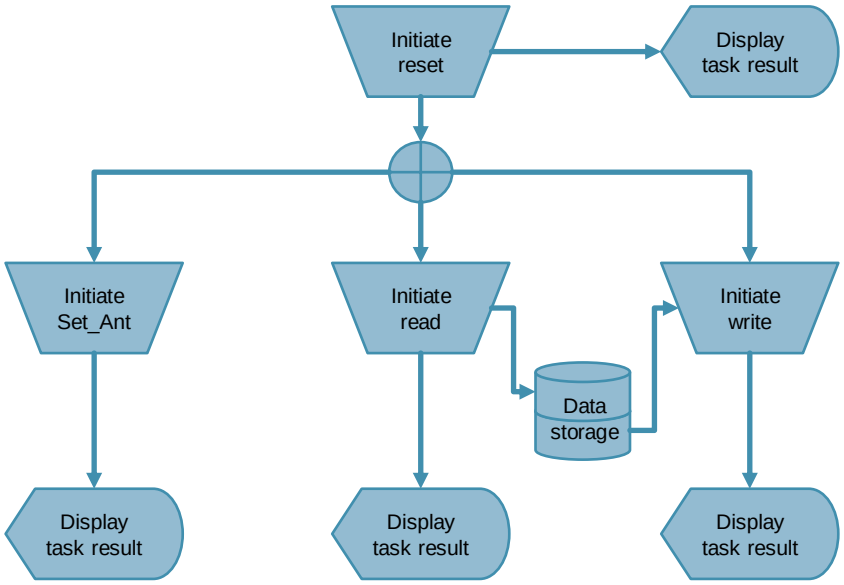
The solution presented here offers the following advantages:

- As a result of the illustrations and explanations contained in this application example, you will save time and costs during the commissioning of your own automation solution.
- You can expand the supplied hardware configuration as required, or integrate it into your project.
- By using the ASM 456, the SIMATIC RF Reader can be integrated into a control system via PROFIBUS. Older automation solutions often require this.
- Using the prefabricated Ident blocks simplifies the operation of the SIMATIC RF Reader significantly.

2.2 Description of the core functionality

Sequence of the core functionality

Figure 2-2



A reset (initialization) via the Ident blocks puts the RFID reader into operation mode. After executing the reset successfully, the user can select between the different RFID operations and monitor their status.

2.3 Hardware and software components

This application example was created with the following components:

Hardware components

Table 2-1

Component	Qty	Article number	Note
SIMATIC S7-300 CPU 317-2 PN/DP	1	6ES7317-2EK14-0AB0	Alternatively, a different S7-300/400-CPU may also be used
SIMATIC S7-1200 CPU 1214 DC/DC/DC	1	6ES7214-1AG40-0XB0	Alternatively, a different S7-1200-CPU may also be used
Communication module CM 1243-5	1	6GK7243-5DX30-0XE0	Only required in conjunction with a S7-1200
SIMATIC S7-1500 CPU 1516-3 PN/DP	1	6ES7516-3AN01-0AB0	Alternatively, a different S7-1500-CPU may also be used
ASM 456	1	6GT2002-0ED00	
ET200ECO connection block ECOFAST	1	6ES7194-3AA00-0AA0	Alternatively, the ET200 ECO connection block M12 7/8” may also be used (6ES7194-3AA00-0BA0)
SIMATIC NET, PROFIBUS ECOFAST HYBRID CABLE	1	6XV1830-7BH30	
SIMATIC RF220R	1	6GT2821-2AC10	Alternatively, a different SIMATIC RF200 reader may also be used.
MDS D424	n	6GT2600-4AC00	Alternatively, different ISO transponders may also be used
KAB_KONF_ASM 456, RF170C-READER_2M	1	6GT2891-0FH20	

Note This document describes the automation solution with different S7-control types. To put the application example into operation, all you need is one of the controls named above.

Hinweis When using a SIMATIC S7-400, only the function block "Ident_Profile" works.

Software components

Table 2-2

Component	Qty	Article number	Note
STEP 7 TIA Portal	1	6ES7822-0A.03-..	V13 SP1 or higher

Example files and projects

The following list includes all files and projects that are used in this example.

Table 2-3

Component	Note
109483364_SIMATIC_RF200_ASM456_S7-300.zip	This zip file contains a STEP 7 project for example project for SIMATIC S7-300.
109483364_SIMATIC_RF200_ASM456_S7-1200.zip	This zip file contains a STEP 7 project for example project for SIMATIC S7-1200.
109483364_SIMATIC_RF200_ASM456_S7-1500.zip	This zip file contains a STEP 7 project for example project for SIMATIC S7-1500.
109483364_SIMATIC_RF200_ASM456_en.pdf	This document.

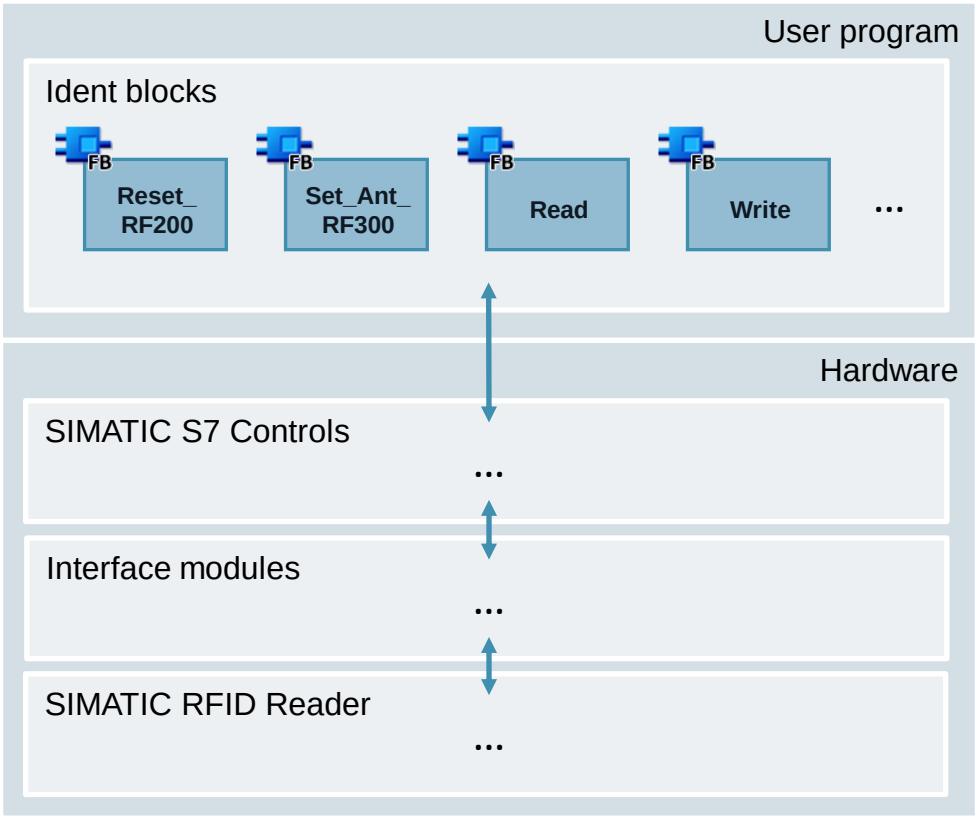
3 PLC program

The following chapters describe the PLC program of the configuration help.

3.1 General overview

The following figure schematically shows the interaction of the user program and the SIMATIC hardware:

Figure 3-1



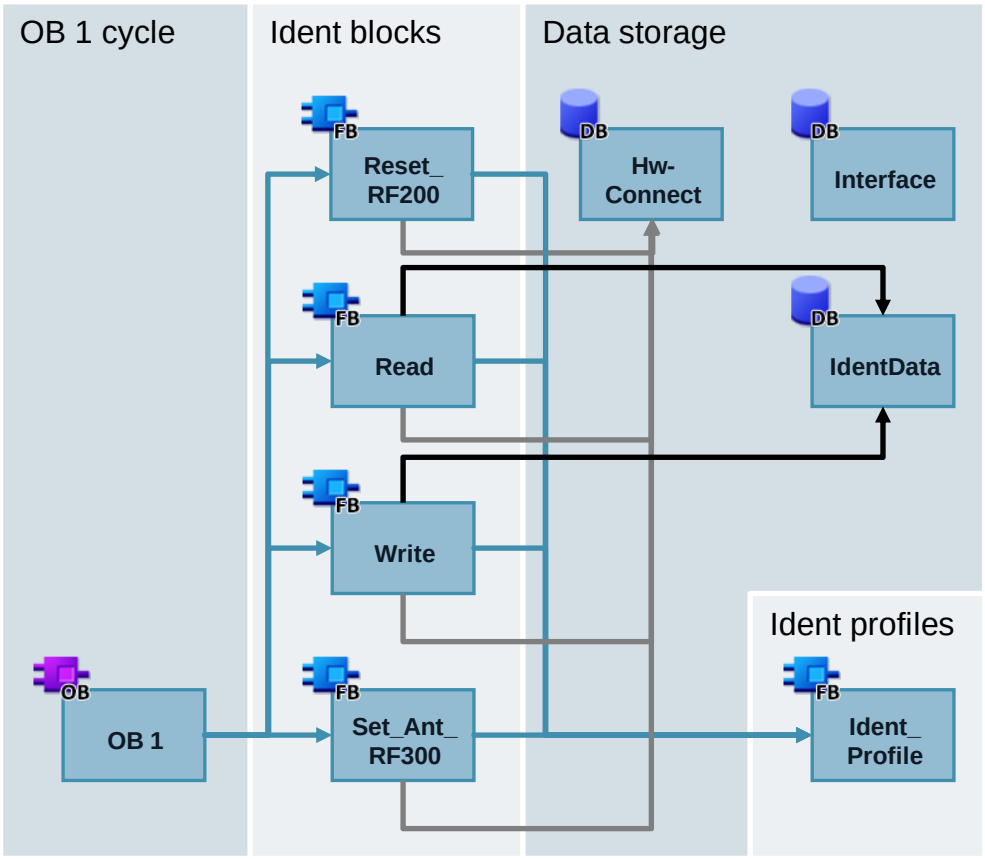
The Ident blocks from the “SIMATIC Ident” library integrated into the TIA Portal, control the RFID operations of the RFID readers. The command telegrams created by the Ident blocks are transmitted via the communication modules to the SIMATIC RFID readers by the SIMATIC S7 control. Over the same path, the response telegrams of the RFID readers are sent back to the user program, where they are available for the user via the block outputs.
In this, it does not matter for the Ident blocks, whether the communication modules are connected to the control system proprietarily, or via PROFIBUS or PROFINET.

Note The block “Set_Ant_RF300” applies for SIMATIC RF200 as well as for RF300 modules.

Program overview

The following figure shows the call hierarchy of the user program in this application example:

Figure 3-2



To demonstrate the use of the Ident blocks in a simple way, they are called directly in the OB 1 cycle in this application example. Hence, a minimal PLC program is created.
Every Ident-block is responsible for a specific RFID functionality.
The read and to be written transponder data are stored in the data block “IdentData”.
The data block “Interface” contains all tags to be controlled in the application example.
All Ident blocks call the fundamental block “Ident_Profile” internally. Therefore, the Ident blocks are only an encapsulation of the complex Ident_Profile block in order to facilitate the realization of the RFID functionalities for the user.

The names of the blocks describe their fundamental core functionality. The following Ident blocks are available for the SIMATIC RF200 product portfolio:

Table 3-1

Ident block name	Implemented into this example
Reset_RF200	✓
Reset_Reader	✘
Set_ant_RF300	✓
Read	✓

Ident block name	Implemented into this example
Write	✓
Read_UID	✘
AdvancedCMD	✘
Reader_Status	✘
Tag_Status	✘

In this application example, only the four implemented blocks are considered, since they are relevant for the fundamental operation of the SIMATIC RF-readers.

3.2 Fundamentals of using the Ident blocks

To use the Ident blocks, the following fundamentals have to be observed:

Calling the blocks

If an Ident block is called in the TIA Portal, all required software components are automatically generated for the data block and added to your project. These include user-defined data types (UDTs), the fundamental “Ident_Profile” data block and various system function blocks (SFCs and SFBs).
You can find the Ident blocks in the TIA Portal under “Instructions“ > “Optional packages“ > “SIMATIC Ident“.

Addressing the blocks

All Ident blocks have an INOUT parameter called “HW_CONNECT”. Via this interface, the blocks have to be informed about the address information of the configured interface module/reader. For this, upon calling an Ident block, the UDT “IID_HW_CONNECT” is automatically generated and stored for use in the TIA project tree under “PLC data types”.

You can find further information about using the UDT “IID_HW_CONNECT” in the next chapter 3.3 Explanation of “HW_CONNECT“.

Edge-triggered execution of the blocks

All Ident blocks have an input parameter called “EXECUTE”. Via this interface, the execution of the respective RFID functionality of the blocks is started. It should be noted that the input interface “EXECUTE” reacts to positive edges only. This means that the blocks have to be run with a logical 0 (”false”) in the EXECUTE parameters, before they can register switching to 1 (”true”) in a later run of the CPU cycle.
This is why it is important to set the EXECUTE parameter to 0 (”false”) after a successful or unsuccessful processing of the Ident blocks, in order to be able to restart the block at a later time.

Also see chapter “7. Tips and tricks”.

Resetting the blocks

Before a SIMATIC RFID reader can start, a reset has to be performed. For this, special reset blocks are available.

Command buffer for transponder operations

All transponder operations (e.g. “Read” and “Write”) are buffered in the SIMATIC RFID readers. By this, you can send a read task to the reader, even if there is no transponder in his RF field. The buffered task will be executed only when a transponder passes through the RF field. To abort a current buffered task, you have to perform a reset.

3.3 Explanation of “HW_CONNECT”

The input parameter “HW_CONNECT” of the Ident blocks serves for addressing the configured interface modules/readers and requires the UDT “IID_HW_CONNECT” as data type. Upon calling an Ident block, this data type will be automatically generated for you. It can be found in the project tree under “PLC data types”.

Details on UTD “IID_HW_CONNECT”

The structure of the UDT “IID_HW_CONNECT” is shown in the following figure:

Figure 3-3

		Name	Data type	Start value
1		Static		
2		hwConnect	IID_HW_CONNECT	
3		HW_ID	Word	16#0
4		CM_CHANNEL	Int	1
5		LADDR	DWord	16#0
6		Static	IID_IN_SYNC	

The following three fields in the UDT are intended for addressing the interface modules/readers:

- “HW_ID”
- “CM_CHANNEL“
- “LADDR“

In this, the “HW_ID” (to be noted only with S7-1200/1500) describes the HW identifier of the RFID communication channel of the interface module. The “CM_CHANNEL” describes the slot of the RFID reader on the communication module (1 or 2). “LADDR” describes the I/O address of the RFID communication channel.

Using the UDT “IID_HW_CONNECT”

The following description explains the procedure for creating an addressing tag:

1. Create a tag of the type “IID_HW_CONNECT”.
2. Enter the correct address values into the tag (see below).
3. Connect this tag with the input parameter “HW_CONNECT” of the Ident blocks.

Note For each SIMATIC RFID reader in your project, you have to create a separate address tag.

Reading the correct address values

The following figure shows, where you can read the correct values for addressing the interface modules/readers:

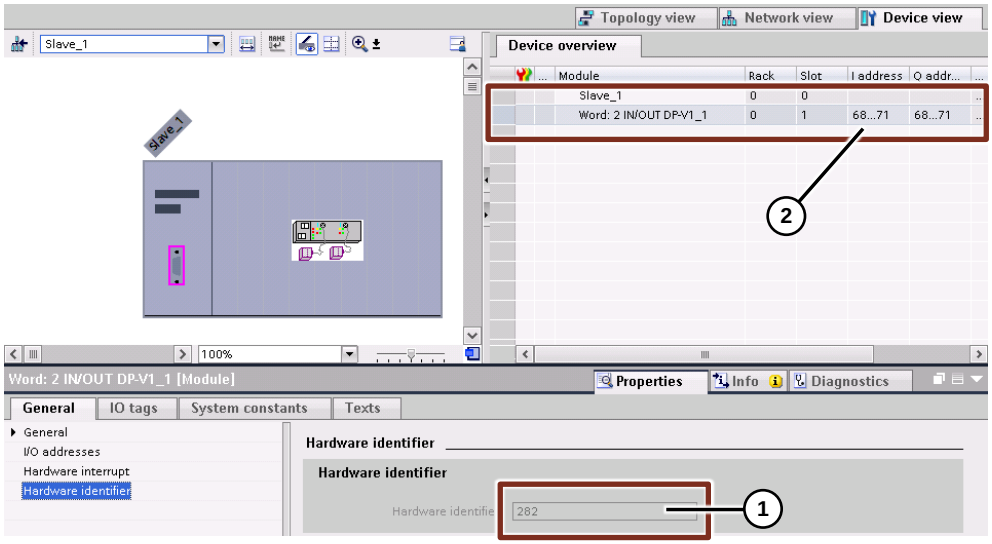


Table 3-2

No.	Description
1.	Value for “HW_ID“ of the UDT (only for S7-1200/1500)
2.	Value for “LADDR“ of the UDT

The following description explains the procedure for reading the address values:

1. In the TIA Portal, navigate to “Device view”.
2. Select your configured interface module.
3. In the window “Device overview”, select the module “Word 2 IN/OUT DP-V1_1“ (at ASM456).
4. In the inspector window, click on “Properties“ > “Hardware identifier“
5. In the field “Hardware identifier”, you can find the value that has to be entered into the field “HW_ID” **(1)** in the UDT (in this example: 282).
6. In the fields “I address” and “Q address” in the device view of the selected module, you can find the value for the field “LADDR” **(2)** of the UDT (in this example: 68). Be sure that identical values are assigned to both, the E and the A address.
7. In the field “CM_CHANNEL”, enter if the reader is connected to port 1 or 2 of the interface module.

3.4 Explanation of the Ident blocks

This chapter explains the main Ident blocks.

General output parameters of the Ident blocks

All Ident blocks share the following output parameters:

Table 3-3

Parameters	Description
DONE	Task successfully completed.
BUSY	Task running.
ERROR	Task was completed with an error.
STATUS	Error message for a task with errors (ERROR = true)
PRESENCE	Transponder presence in RF field

3.4.1 Functionality “Reset_RF200“

The block “Reset_RF200” puts the reader into operation mode. If the block is executed, all settings at the input parameters of the data block are transmitted to the reader.
Also, by executing the block, the current tasks of the reader are aborted and reader errors are acknowledged.

Program details form block

Figure 3-4

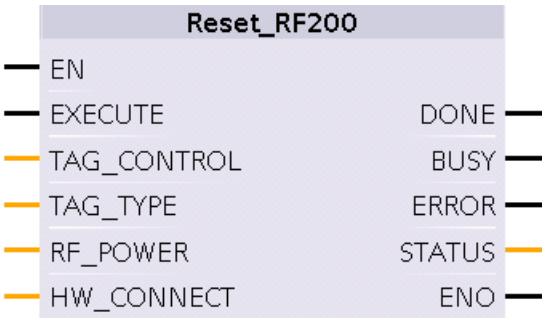


Table 3-4

Parameters	Description
EXECUTE	Starts the reset task
TAG_CONTROL	Presence check on/off
TAG_TYPE	Transponder type: • 1 = ISO transponder • 3 = MDS D3xx optimization
RF_POWER	Transmission power (only for RF290R): HF power from 0,5 W to 5 W in 0,25 W-steps (values range: 0x02 ... 0X14); default value 0x04 ≙ 1 W
HW_CONNECT	Address tag

3.4.2 Functionality “Set_Ant_RF300“

The block “SET_ANT_RF300” serves to switch the RF field of the reader on and off. This block is mainly used for preventing field overlays between adjacent readers.

Program details form block

Figure 3-5

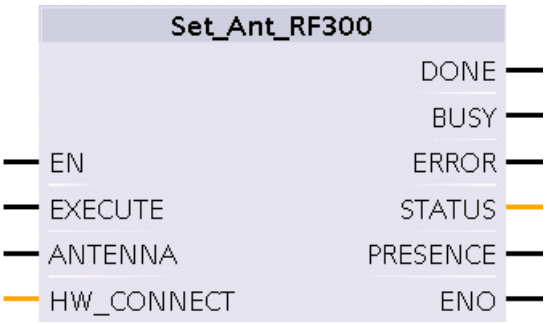


Table 3-5

Parameters	Description
EXECUTE	Starts the antenna task
ANTENNA	RF antenna on/off • 1 = RF field on • 0 = RF field off
HW_CONNECT	Address tag

3.4.3 Functionality “Read”

With the block “Read”, you can read process data from a transponder. Via the parameters “ADDR_TAG” and “LEN_DATA”, the storage area to be read is defined. The read data are made available via the parameter “IDENT_DATA”.

Program details for block

Figure 3-6

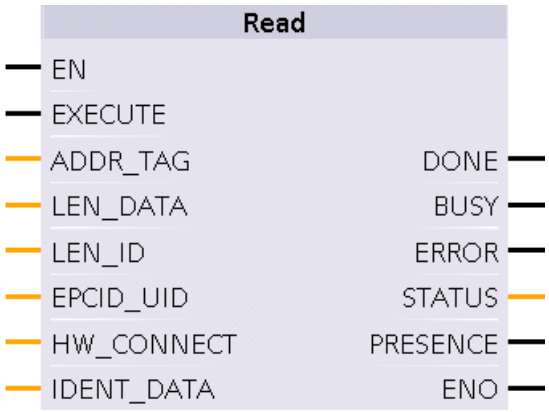


Table 3-6

Parameters	Description
EXECUTE	Starts the read task
ADDR_TAG	Physical address from where the transponder data is read
LEN_DATA	Length of the data to be read
LEN_ID	(Only for RF600)
EPCID_UID	(Only for RF600)
HW_CONNECT	Address tag
IDENT_DATA	Data buffer for the transponder data read; Only one array of byte with a variable length is allowed

3.4.4 Functionality “Write”

With the block “Write” you can write process data on a transponder. Via the parameters “ADDR_TAG” and “LEN_DATA”, the storage area to be written is defined. The data to be written are made available via the parameter “IDENT_DATA”.

Program details for block

Figure 3-7

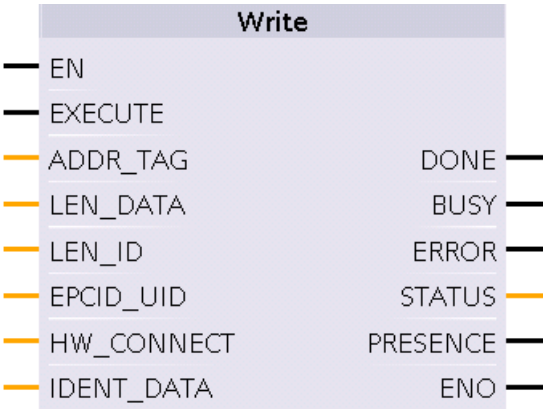


Table 3-7

Parameters	Description
EXECUTE	Starts the write task
ADDR_TAG	Physical address from where the transponder data is written
LEN_DATA	Length of the data to be written
LEN_ID	(Only for RF600)
EPCID_UID	(Only for RF600)
HW_CONNECT	Address tag
IDENT_DATA	Data buffer for the transponder data read; Only one array of byte with a variable length is allowed

3.5 Explanation of block “Ident_Profile”

The block “Ident_profile” forms the basis of all Ident blocks. The simplified Ident blocks execute the Ident profile for the respective RFID functionality with a fitting parameter configuration. The blocks therefore represent a simplified interface of the Ident profile.

Note Provided that the Ident blocks can display the requested RFID functionalities, it is not recommended to work with the block “Ident_Profile” directly.

The Ident profile is based on the standardized “Proxy Ident Function Block” protocol. The specifications of the protocol can be obtained via the PROFIBUS user organization (\10\).

3.6 Diagnostics and error messages

In case of an error, the Ident blocks supply different hex codes via the output parameter “STATUS”. The meaning of the codes can be found in the manual “SIMATIC Ident RFID systems Ident profile and Ident blocks, standard function for Ident systems” (\3\) under chapter 4 “Error messages”.

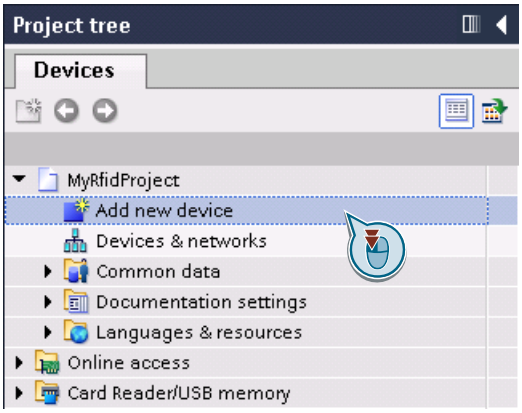
4 Configuration and Settings

This chapter describes the necessary configuration steps to create a project with SIMATIC S7-CPU, ASM 456 and SIMATIC RF200 reader.

Note Follow the instructions in this chapter, if you use alternative hardware. With identical hardware (see chapter "2.3 Hardware and software components"), the following configuration steps do not have to be made as they are included in this project. Continue with chapter "5 Installation and Commissioning".

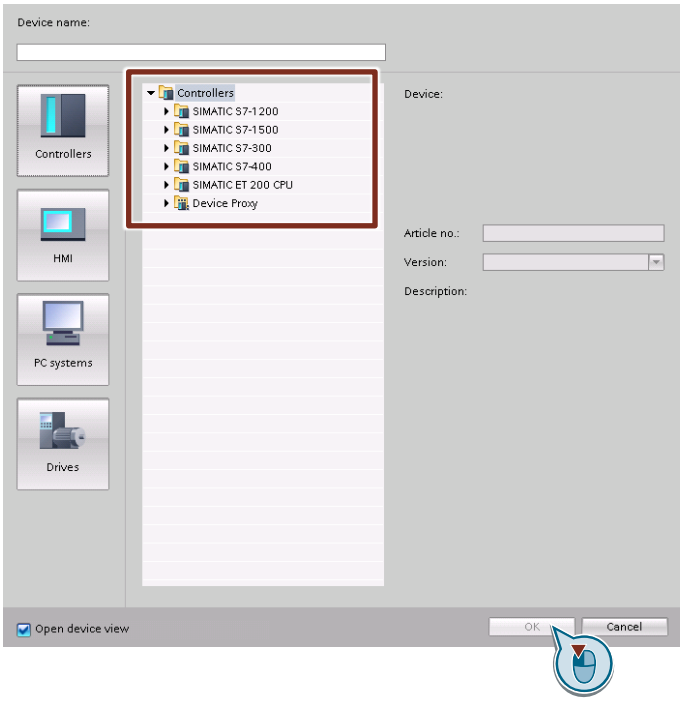
1. Open your TIA Portal V13 SP1.
2. Create a new project.
3. In the TIA Portal project tree, double-click on "Add new device".

Figure 4-1



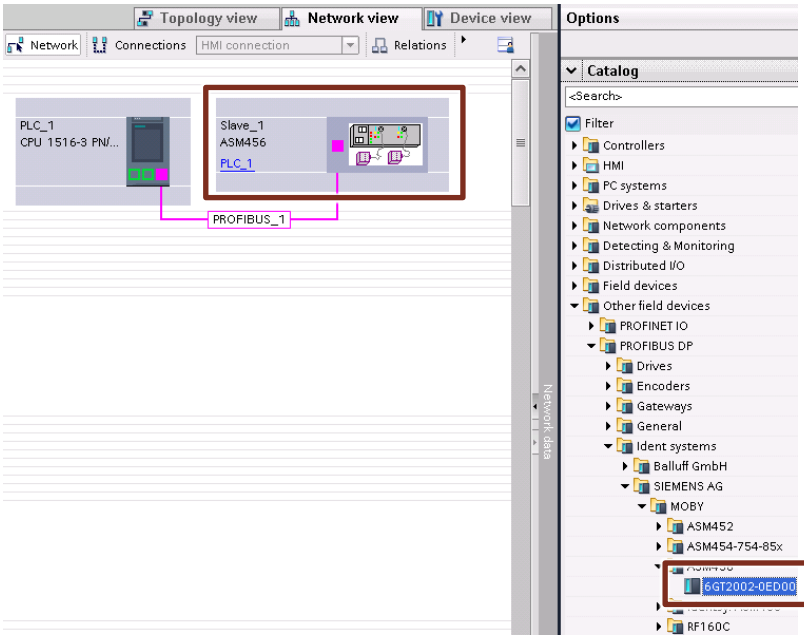
4. In the appearing dialog box, select the S7-CPU you use. Make sure that the CPU has a PROFIBUS port or add a PROFIBUS-CP to the CPU. Confirm with “OK”.

Figure 4-2



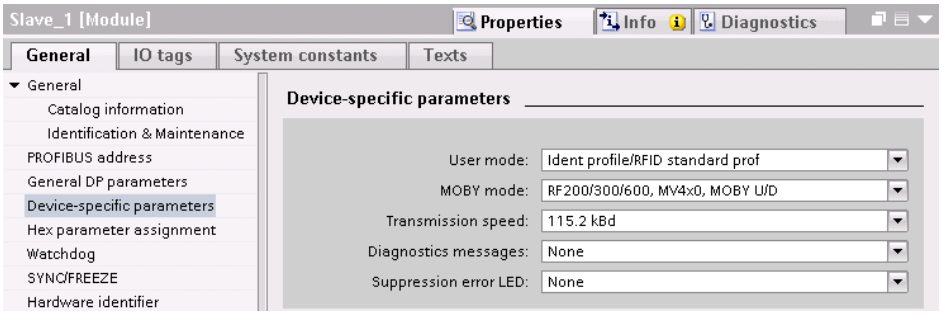
5. Go to “Network view”, in the “Hardware Catalog”, click on “Other field devices“ > “PROFIBUS DP“ > “Ident Systems“ > “Siemens AG“ > “Moby“ > “ASM456“. Per “Drag&Drop”, drag the interface module (”6GT2002-0ED00“) from the Hardware Catalog into your work area and connect the PROFIBUS port with the CPU.

Figure 4-3



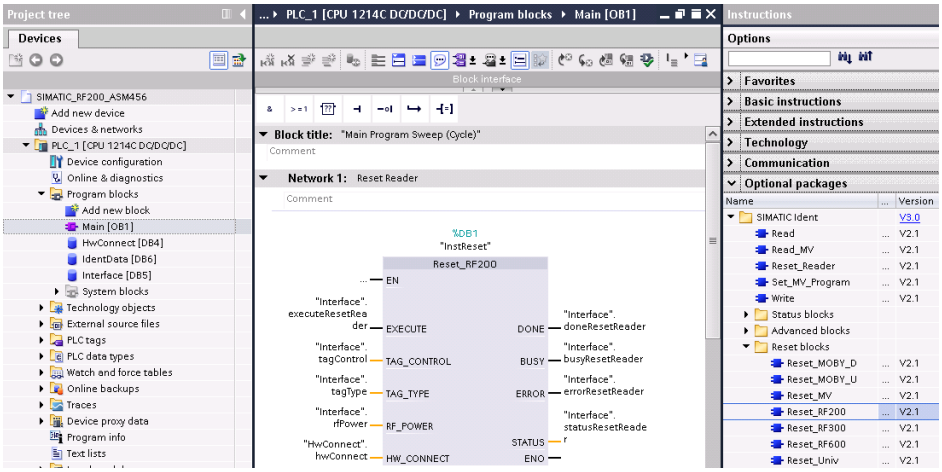
6. Double-click on the ASM456 and navigate to Properties of the ASM in the inspector window.
7. Select “Device-specific parameters”. Set the existing parameters as follows:
   - User Mode: Ident profile/RFID standard profile
   - MOBY Mode: RF200/300/600, MV4x0, Moby U/D
   - Transmission speed: 115,2 kBaud
   - Diagnostic messages: none
   - Suppression error LED: none

   Figure 4-4



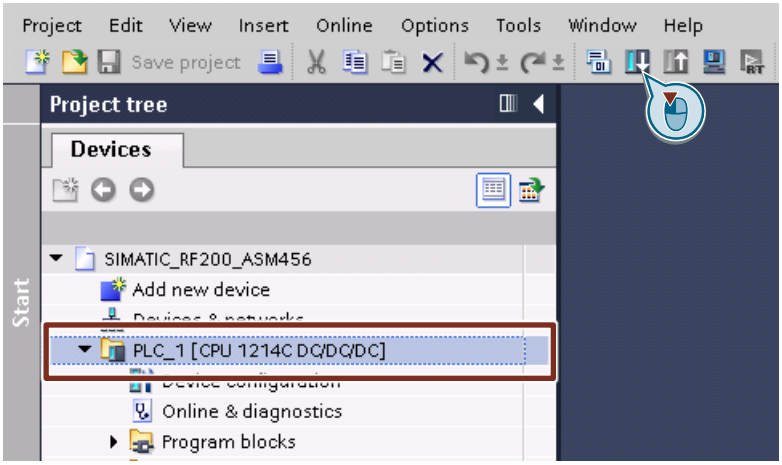
8. Open an FC, FB or OB by calling the Ident blocks.
9. In the TIA Portal, navigate to “Instructions“ > “Optional packages“ > “SIMATIC Ident“ Drag the Ident blocks you require into the block previously opened by you:

   Figure 4-5



10. Assign the required values/tags to the parameter interfaces of the Ident blocks.
11. Load the user program and the hardware configuration into your S7-CPU. For this, select the S7-CPU in your project tree and then click on “Download to device”.

Figure 4-6



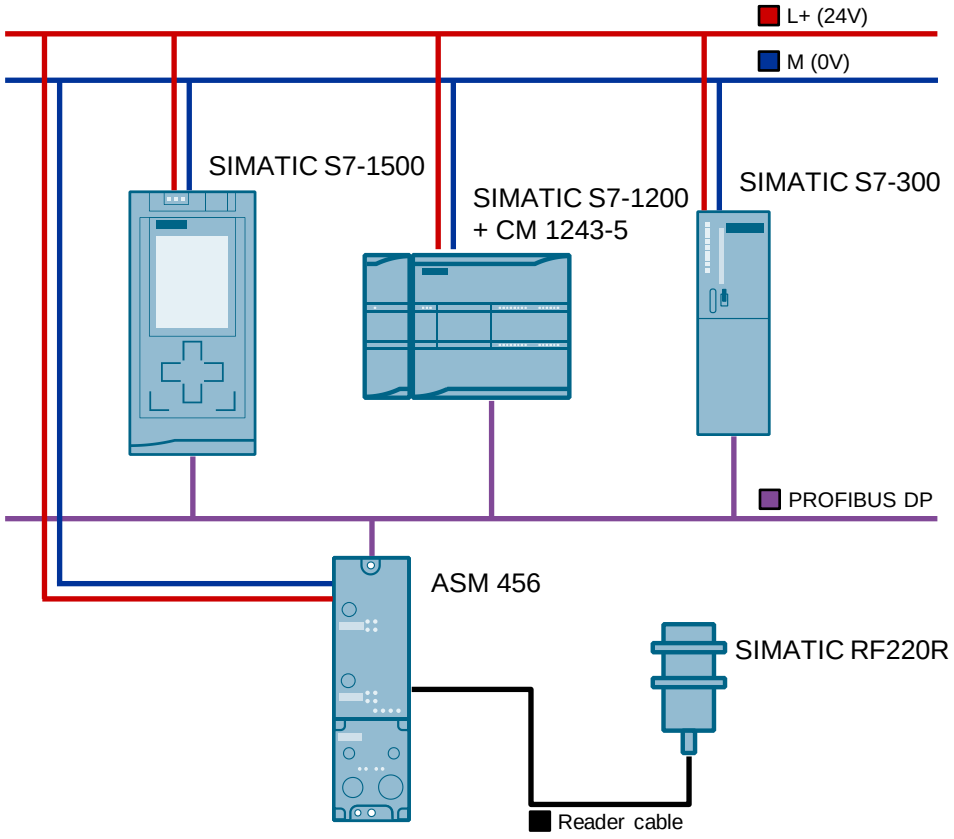
5 Installation and Commissioning

This chapter describes the steps necessary for starting the example, using the code from the download and the components from the hardware list.

5.1 Installing the hardware

The figure below shows the hardware configuration of the application.

Figure 5-1



Note The installation guidelines for SIMATIC S7-1500 (\6\), SIMATIC S7-1200 (\7\), SIMATIC S7-300 (\8\), SIMATIC ASM 456 (\5\) and SIMATIC RF200 (\4\) should always be observed.

5.2 Installing the software (download)

This chapter describes the steps to install the sample code.

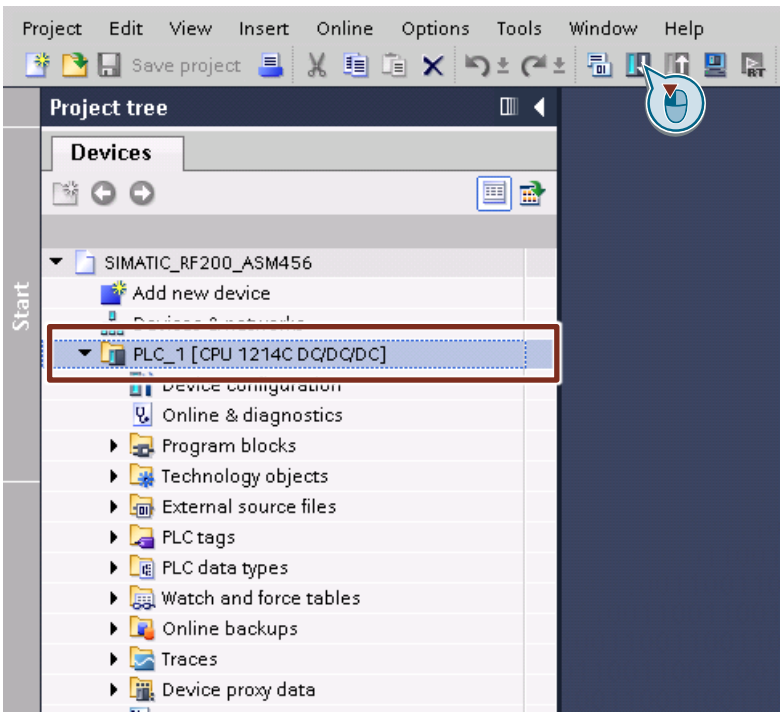
1. Install TIA Portal V13 SP1 or higher.
2. The Step 7 V13 SP1 project is available on the HTML page from which you downloaded this document (\2\). Save the project "109483364_SIMATIC_RF200_ASM456_S7-xxx.zip" (depending on CPU) on your hard drive.
3. Unzip the file.
4. In the TIA Portal, open the unzipped project: "Project" > "Open" > "Browse".
5. The project is now available to you.

5.3 Commissioning

This chapter describes the steps to start up the sample code.

1. Wire your hardware.
2. Open the application example in the TIA Portal.
3. If you use alternative hardware, follow the settings described in chapter "4 Configuration and Settings".
4. In your programming device, assign an IP address from the IP strip of the configured S7-CPU of the application example, or adjust the IP address of the CPU according to your preferences.
5. Load the user program to your S7-CPU. For this, select the S7-CPU in your project tree and then click on "Download to device".

Figure 5-2



6 Operating the application example

The application example is controlled via a monitoring table.
This chapter describes the operation of the application example.

6.1 Description of the monitoring table

The following figure shows the monitoring table for operating the application example:

Figure 6-1

	i	Name	Address	Display format
1	// Reset Reader			
2		"Interface".executeResetReader		Bool
3		"Interface".tagControl		Hex
4		"Interface".tagType		Hex
5		"Interface".rfPower		Hex
6		"Interface".doneResetReader		Bool
7		"Interface".errorResetReader		Bool
8		"Interface".statusResetReader		Hex
9	// Read TAG			
10		"Interface".executeRead		Bool
11		"Interface".addrTagRead		Hex
12		"Interface".lenDataRead		Hex
13		"Interface".doneRead		Bool
14		"Interface".errorRead		Bool
15		"Interface".statusRead		Hex
16	// Write TAG			
17		"Interface".executeWrite		Bool
18		"Interface".addrTagWrite		Hex
19		"Interface".lenDateWrite		Hex
20		"Interface".doneWrite		Bool
21		"Interface".errorWrite		Bool
22		"Interface".statusWrite		Hex
23	// Set Ant			
24		"Interface".executeAnt		Bool
25		"Interface".antennaAnt		Bool
26		"Interface".doneAnt		Bool
27		"Interface".errorAnt		Bool
28		"Interface".statusAnt		Hex

The monitoring table is divided into four areas. Each area belongs to one of the Ident blocks called in the OB 1:

- Reset Reader: “Reset_RF200“
- Read Tag: “Read”
- Write Tag: “Write”
- Set Ant: “Set_Ant_RF300“

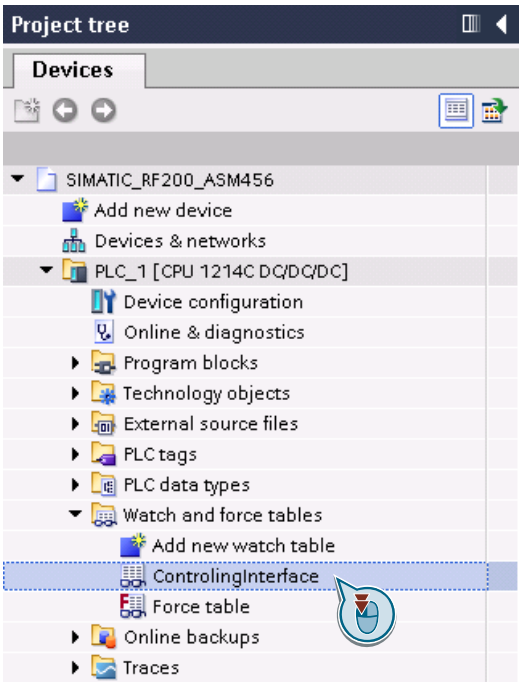
Within the areas, the tags to be controlled and to be monitored are entered from the data block “Interface”. These tags are directly linked to the Ident blocks.

6.2 Operating the monitoring table

This chapter describes the steps to operate the sample code.

1. Open the data block “IdentData” and fill the tag “writeData” at your own discretion. This data will be written to the transponders in the later course. Load the data block into the S7-CPU.
2. Open the monitoring table “ControllingInterface” from the project tree with a double-click.

Figure 6-2



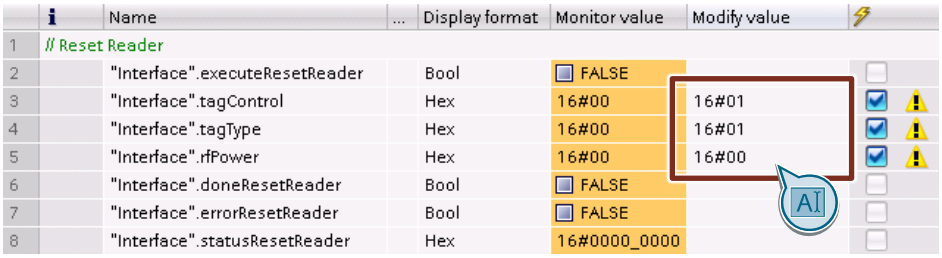
3. In the monitoring table, click on “Monitor all”.

Figure 6-3

6.2.1 Executing a reset command

1. Enter the desired reset parameters into the tags from the area “Rest Reader” into the fields of the column “Modify Value”.

Figure 6-4



	Name	...	Display format	Monitor value	Modify value	
1	// Reset Reader					
2	"Interface".executeResetReader		Bool	FALSE		
3	"Interface".tagControl		Hex	16#00	16#01	
4	"Interface".tagType		Hex	16#00	16#01	
5	"Interface".rfPower		Hex	16#00	16#00	
6	"Interface".doneResetReader		Bool	FALSE		
7	"Interface".errorResetReader		Bool	FALSE		
8	"Interface".statusResetReader		Hex	16#0000_0000		

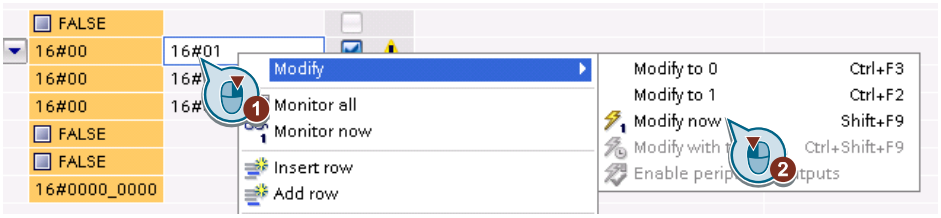
In this example, the values are set as follows:

- tagControl: 16#01 (presence check on)

- tagType: 16#01 (ISO transponder)
- rfPower: 16#00 (only for reader RF290R)

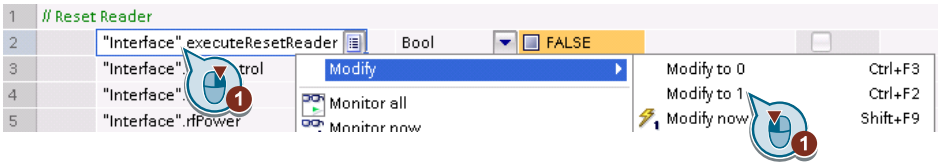
2. Right-click on any line of the monitoring table. Go to “Modify” and then left-click on “Modify now”. By this, you transmit the values entered in Step 1 directly into the data block of the S7-CPU.

Figure 6-5



3. Perform a reset now. For this, right-click on the line of the tag “executeResetReader”, select “Modify” and left-click on “Modify to 1”.

Figure 6-6



4. Via the tags “doneResetReader”, “errorResetReader” and “statusResetReader”, you can check, whether the task has been carried out successfully or not.

Figure 6-7

1	// Reset Reader				
2		"Interface".executeResetReader	Bool	TRUE	TRUE
3		"Interface".tagControl	Hex	16#01	16#01
4		"Interface".tagType	Hex	16#01	16#01
5		"Interface".rfPower	Hex	16#00	16#00
6		"Interface".doneResetReader	Bool	TRUE	
7		"Interface".errorResetReader	Bool	FALSE	
8		"Interface".statusResetReader	Hex	16#0000_0000	

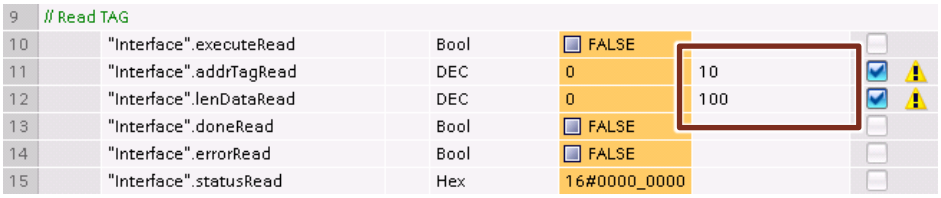
A faulty task only occurs, if the tag “errorResetReader” is set. In this case, a hex error code will be automatically written into the “statusResetReader” tag. Via this error code, you can identify the cause of error.

After a successful task, you can see the LED status indicator on the reader flashing continuously in green.

6.2.2 Executing a read command

1. Enter the desired read parameters into the tags from the area “Read TAG” into the fields of the column “Modify Value”.

Figure 6-8

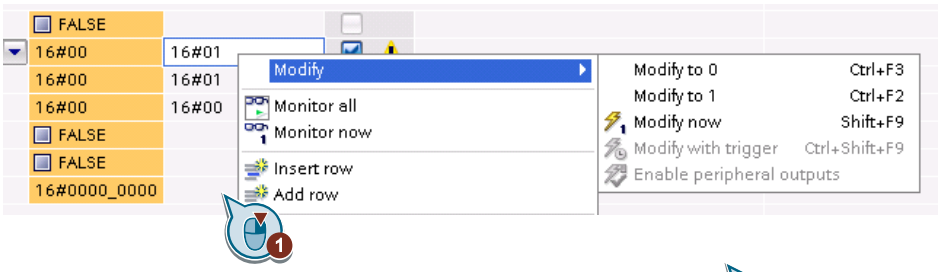


In this example, the values are set as follows:

- addrTagRead: 10 (start address, from where on the transpo shall be read)
- lenDataRead: 100 (length of the data to be read)

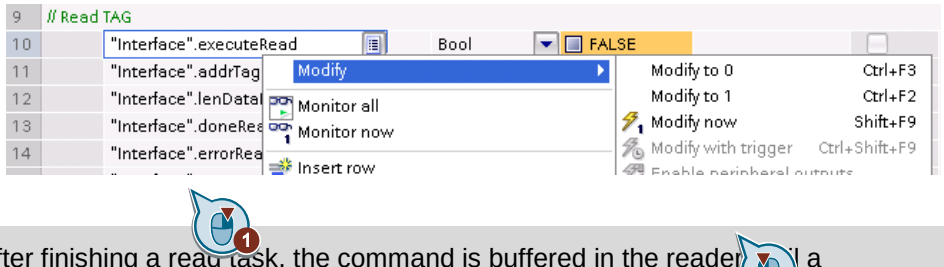
2. Right-click on any line of the monitoring table. Go to “Modify” and then left-click on “Modify now”. By this, you transmit the values entered in Step 1 directly into the data block of the S7-CPU.

Figure 6-9



3. Perform a read now. For this, right-click on the line of the “exe Read” tag, select “Modify” and left-click on “Modify to 1”.

Figure 6-10



Note

After finishing a read task, the command is buffered in the reader il a transponder passes through the RF field. Only then will the outpu meters DONE, ERROR and STATUS be set.

4. Via the tags “doneRead”, “errorRead” and “statusRead”, you can check, whether the task has been successfully carried out or not.

Figure 6-11

9	// Read TAG				
10		"Interface".executeRead	Bool	TRUE	TRUE
11		"Interface".addrTagRead	DEC	10	10
12		"Interface".lenDataRead	DEC	100	100
13		"Interface".doneRead	Bool	TRUE	
14		"Interface".errorRead	Bool	FALSE	
15		"Interface".statusRead	Hex	16#0000_0000	

A faulty task only occurs, if the tag “errorRead” is set. In this case, a hex error code will be automatically written into the “statusRead” tag. Via this error code, you can identify the cause of error.

After a successful task, the transponder data read can be found in the data block “IdentData” under the “readData” tag.

6.2.3 Executing a write command

1. Enter the desired write parameters into the tags from the area “Write TAG” into the fields of the column “Modify Value”.

Figure 6-12

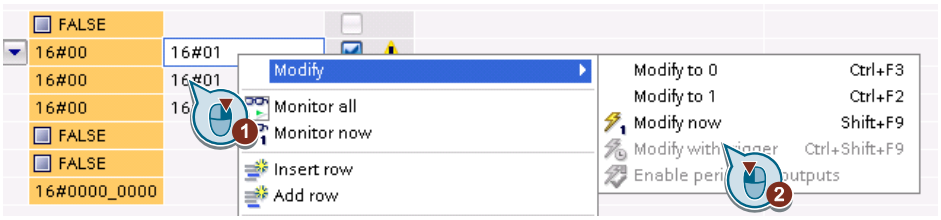
16	// Write TAG					
17		"Interface".executeWrite	Bool	FALSE		
18		"Interface".addrTagWrite	DEC	0	10	
19		"Interface".lenDateWrite	DEC	0	100	
20		"Interface".doneWrite	Bool	FALSE		
21		"Interface".errorWrite	Bool	FALSE		
22		"Interface".statusWrite	Hex	16#0000_0000		

In this example, the values are set as follows:

- addrTagWrite: 10 (start address, from where on the transponder shall be written on)
- lenDataWrite: 100 (length of the data to be written)

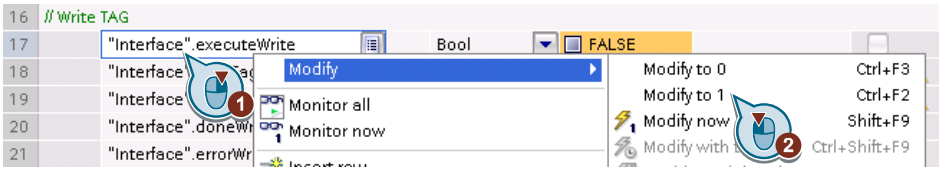
2. Right-click on any line of the monitoring table. Go to “Modify” and then left-click on “Modify now”. By this, you transmit the values entered in Step 1 directly into the data block of the S7-CPU.

Figure 6-13



3. Perform a write now. For this, right-click on the line of the “executeWrite” tag, select “Modify” and left-click on “Modify to 1”.

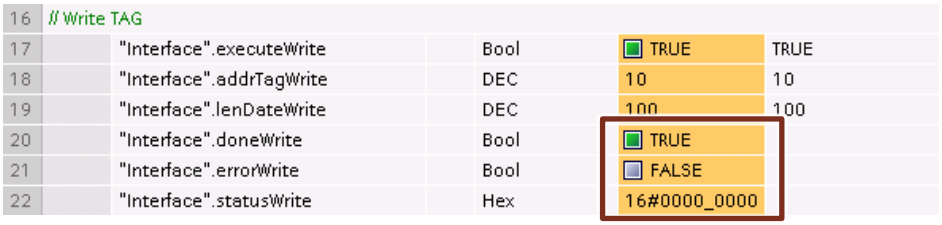
Figure 6-14



Note After finishing a write task, the command is buffered in the reader until a transponder passes through the RF field. Only then will the output parameters DONE, ERROR and STATUS be set.

4. Via the tags “doneWrite”, “errorWrite” and “statusWrite”, you can check, whether the task has been successfully carried out or not.

Figure 6-15



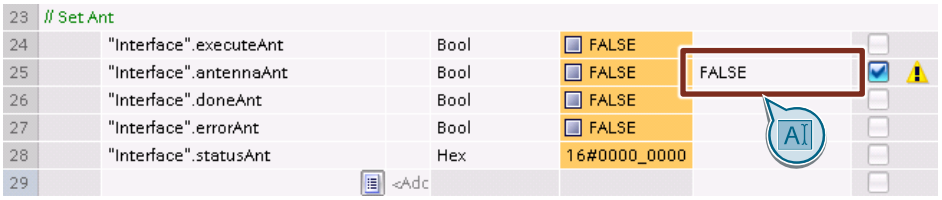
A faulty task only occurs, if the tag “errorWrite” is set. In this case, a hex error code will be automatically written into the “statusWrite” tag. Via this error code, you can identify the cause of error.

After a successful task, you can verify the written transponder data by reading it from the transponder via the “Read” command.

6.2.4 Executing an antenna command

1. Enter the desired antenna parameters into the tags from the area “Set Ant” into the fields of the column “Modify Value”.

Figure 6-16

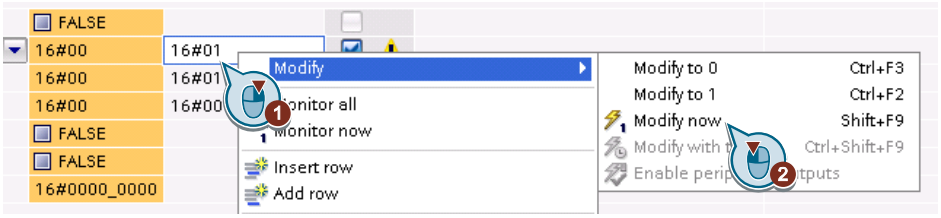


In this example, the value for antennaAnt is set to FALSE (0), to switch off the antenna of the reader.

Note After a reset, the antennas of the SIMATIC RFID readers are switched on by default. If you execute an antenna command with the values TRUE (1) in the parameter ANTENNA after a reset, you will get an error since an already active antenna cannot be reactivated.

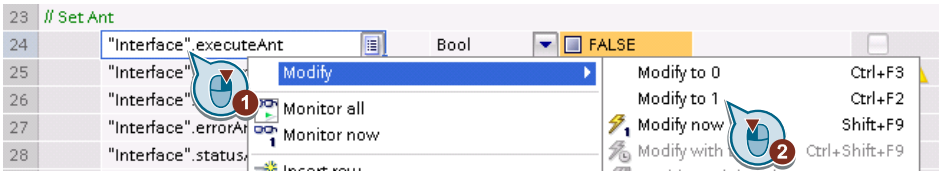
2. Right-click on any line of the monitoring table. Go to “Modify” and then left-click on “Modify now”. By this, you transmit the values entered in Step 1 directly into the data block of the S7-CPU.

Figure 6-17



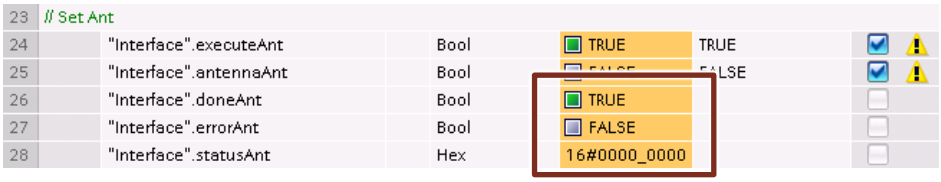
3. Perform an antenna command now. For this, right-click on the line of the “executeAnt” tag, select “Modify” and left-click on “Modify to 1”.

Figure 6-18



4. Via the tags “doneAnt”, “errorAnt” and “statusAnt”, you can check, whether the task has been successfully carried out or not.

Figure 6-19



A faulty task only occurs, if the tag “errorAnt” is set. In this case, a hex error code will be automatically written into the “statusAnt” tag. Via this error code, you can identify the cause of error.

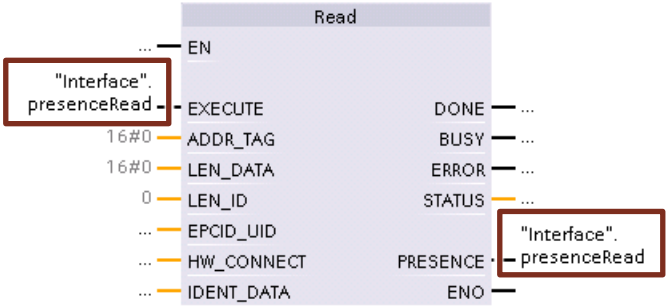
7 Tips and tricks

In this chapter, you will find additional tips and tricks to optimize your SIMATIC RFID applications.

Reading/writing the transponder automatically when transponder is present

Use the output parameter “PRESENCE” of the Ident blocks, to automatically control the input parameter “EXECUTE” when the transponder is present. For this, set the same tag for “PRESENCE” and “EXECUTE”:

Figure 7-1



If you interconnect the blocks in the way illustrated above, a transponder operation is started automatically as soon as a transponder passes the RF field of the readers.

Note

This method is only possible, if a reset command with the parameter “TAG_CONTROL” = true has been previously executed.

With this method, a CPU cycle elapses before a transponder operation is started after a transponder passes the RF field.

Optimizing the processing time of the transponder

Use the command buffer of the SIMATIC RFID reader to minimize the effective processing time for transponder operations. This is particularly helpful, when you realize applications with a fast transponder sequence.

For this, start a transponder operation (e.g. “Read” and “Write”) via the input parameter “EXECUTE”, as soon as the output parameter “DONE” from the previous transponder operation is set.

Executing chained commands

Via the blocks “Ident_Profile” and “AdvancedCMD”, chained commands can be transmitted to the reader. By sending multiple command telegrams in one call, the communication load on the fieldbus can be reduced. Furthermore, the processing time is shortened.

Visit our guide on SIMATIC Ident

Via the “Industrial Identification with SIMATIC Ident in the Siemens Industry Online Support“ guide, (\9\) you can access a number of other application examples and information on the subject of SIMATIC Ident.

8 Related Literature

Table 8-1

	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Download page of the entry https://support.industry.siemens.com/cs/ww/en/view/109483364
\3\	SIMATIC Ident RFID systems Ident profile and Ident blocks, standard functions for Ident systems https://support.industry.siemens.com/cs/ww/de/view/106368029
\4\	SIMATIC Ident RFID systems SIMATIC RF200 https://support.industry.siemens.com/cs/ww/de/view/47189592
\5\	SIMATIC Ident RFID systems connection module ASM 456 - Operating manual https://support.industry.siemens.com/cs/ww/de/view/21230460
\6\	SIMATIC S7-1500, ET 200MP Automation system https://support.industry.siemens.com/cs/ww/de/view/59191792
\7\	SIMATIC S7 S7-1200 Automation system https://support.industry.siemens.com/cs/ww/de/view/109478121
\8\	SIMATIC S7-300 CPU 31xC and CPU 31x: Installation https://support.industry.siemens.com/cs/ww/de/view/13008499
\9\	Industrial Identification with SIMATIC Ident in the Siemens Industry Online Support https://support.industry.siemens.com/cs/ww/de/view/109477926
\10\	Homepage of the PROFIBUS user organization http://www.profibus.com/

9 History

Table 9-1

Version	Date	Modifications
V1.0	05/2016	First version