SIEMENS
Ingenuity for life

Data Record Communication via I-Device

SIMATIC S7-1500

https://support.industry.siemens.com/cs/ww/de/view/109478798

Siemens Industry Online Support

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/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit http://www.siemens.com/industrialsecurity.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit https://support.industry.siemens.com.

Table of Contents

1 Task

1.1 Overview

Introduction

Communicating via data records allows you to exchange various types of data such as recipes or product information between controllers.

This type of data exchange can also be used with the I-device function. Controllers exchange the data records through the configured transfer areas.

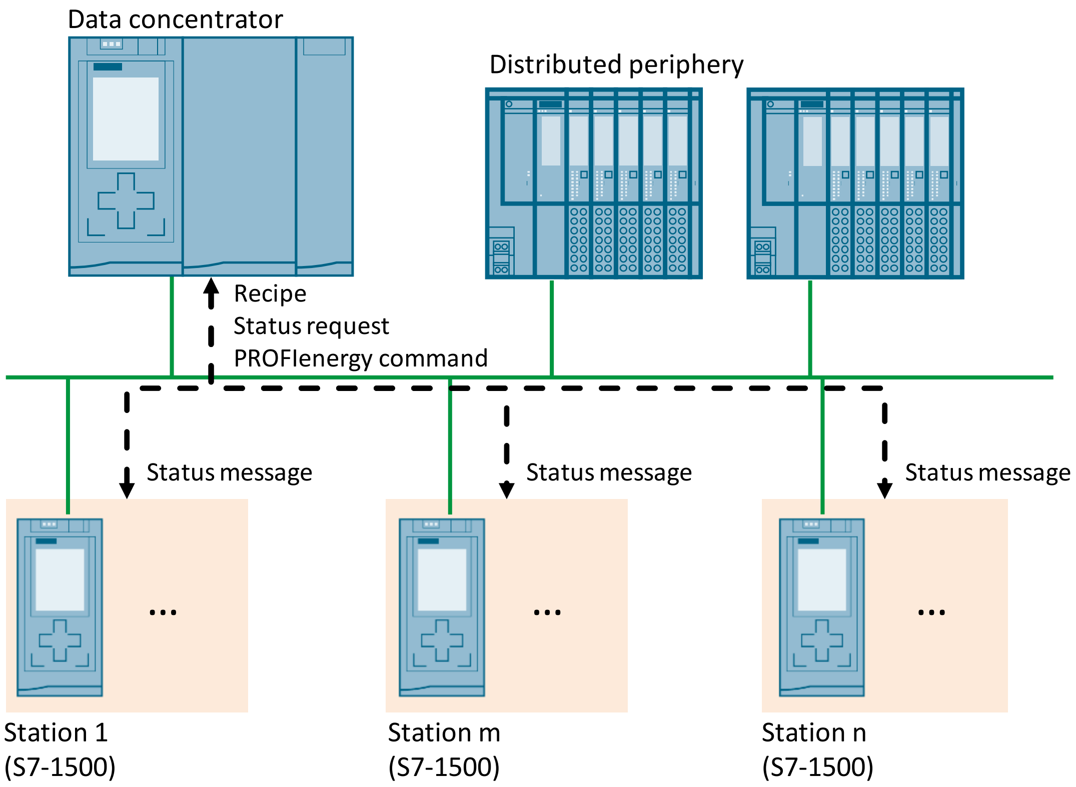
Note For more information about I-device, see here: https://support.industry.siemens.com/cs/de/en/view/109478798.

The application example demonstrates how to configure data record communication between IO controller and I-device using the S7-1500 without additional configuration and hardware effort.

Overview of the automation task

The figure below provides an overview of the automation task.

Figure 1-1



Description

In this system, each station is equipped with a CPU as I-device. Additionally, a separate CPU functions as the central controller for all stations (data concentrator).

Communication to the central IO controller is accomplished by the I-device by instancing the stations in the hardware configuration, allowing each station in the network to be addressed using a unique device name and a unique IO address range for being addressed by the central CPU.

The data concentrator distributes the tasks to the different stations by sending recipes in data records. Additionally, it requests status messages from the individual stations on a regular basis.

Requirements of the automation task

The following table shows the automation requirements for this application example.

Table 1-1

Problem description	Explanation
Sending data records (recipes) from the IO controller to the I-device	The purpose of this example is to show how data records can be sent from an IO controller to an I-device using the existing function blocks.
Requesting data records (status messages) from the I-device with the IO controller	The purpose of this example is to show how an IO controller can request data records from an I-device using the existing function blocks.

Note

It is also possible to send PROFIenergy protocols using the PROFINET I-device function. The "PE_I_DEV” blocks used for this purpose and the associated auxiliary blocks can be found in the standard TIA library under "Extended instruction” > "PROFIenergy”.

For more information, see the TIA help and click here: https://support.industry.siemens.com/cs/de/en/view/109478388.

1.2 Comparing data record communication, open communication and IO communication

General

The PROFINET communication services are divided into two different channels:

- Standard channel:
  - Parameterization and configuration, reading diagnostics data, negotiating the payload channel
  - Open communication (T blocks), data record communication ("RDREC”, "WRREC”, "PRVREC”, "RCVREC”)
- Real-time channel:
  - High-performance payload transfer (cyclic)
  - IO communication (I-device)
  - Clocked payload transfer, hardware support by ERTEC, jitter < 1 µsec

Just like the open communication, the data record communication uses the PROFINET standard channel. The IO communication via I-device runs over the real-time channel.

Comparison

The following table compares the three different communication methods of the I-device function.

Table 1-2

Data record communication	Open communication	IO communication
Data transfer capacity: 32 KB per call	Data transfer capacity: 64KB per call	Data transfer capacity: 1440 bytes per call (max. size of transfer area)
Connection is referenced and established through the HW identifier (automatically by device configuration).	IP addresses have to be parameterized in order to establish the connection.	Connection is obtained and established through the HW identifier (automatically by device configuration).
Low programming effort	Higher programming effort	Very low programming effort
In case of communication, acknowledgment by remote side	In case of communication, acknowledgment can be integrated by user program (ISO level 4)	No acknowledgment of remote side

Contrary to the open communication, the IP address is not parameterized in data record communication. It is determined automatically during IO communication by means of the PROFINET name and the hardware identifier.

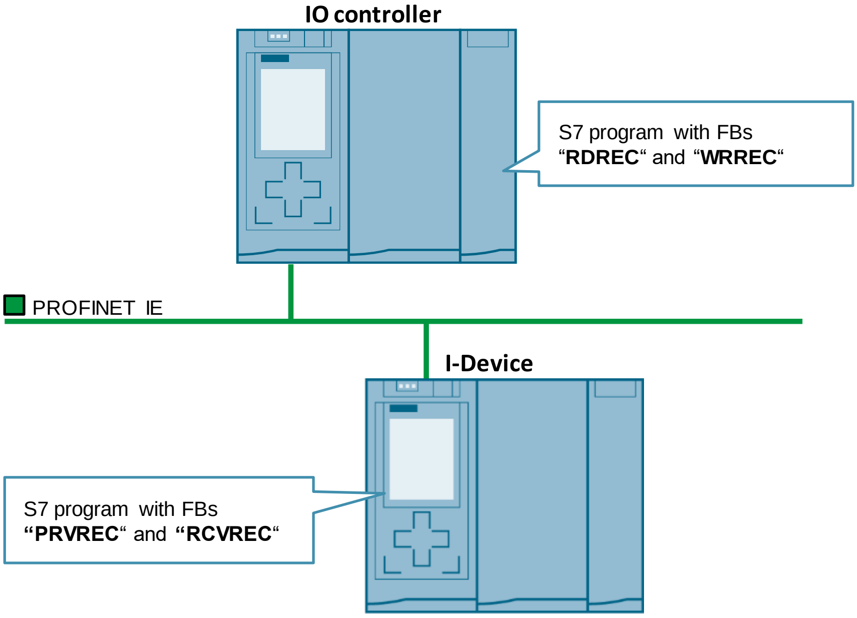
2 Solution

2.1 Overview

Schematic layout

The figure below shows a schematic overview of the components of the solution:

Figure 2-1



Note The function blocks "RDREC”, "WRREC”, "PRVREC” and "RCVREC” are located in the "Extended Instructions” of the TIA portal.

Description

Operating data record communication between IO controller and I-device requires the FBs "RDREC” and "WRREC” in the IO controller and "PRVREC” and "RCVREC” in the I-device to be called at cyclic intervals.

This enables data records to be transferred rapidly between the controllers.

Note The required blocks are also available for the S7-300.

Advantages

The solution presented here offers the following advantages:

- Leaner and more flexible topology
- No need for PN/PN couplers
- Consistent communication to all devices in the PN network
- Fast communication between S7-1500 controllers
- Completed communication is acknowledged by the remote side
- Universal usage, since no IP addresses have to be assigned at the block.

Note S7-1500 controllers are capable of processing up to 20 data records in parallel.

Delimitation

This application does not contain a description of:

- IO controllers (especially S7-1500)
- SCL programming
- I-device function
- STEP 7 V15.1, FUP, SCL, Ethernet and TCP/IP

Basic knowledge of these topics is assumed.

2.2 Description of the core functionality

The example project for this application contains two examples.

- Sending data records from the IO controller to the I-device
- Requesting data records from the I-device by the IO controller

Sequence of the core functionality

The following figure shows the schematic sequence of the entire application example:

Figure 2-2

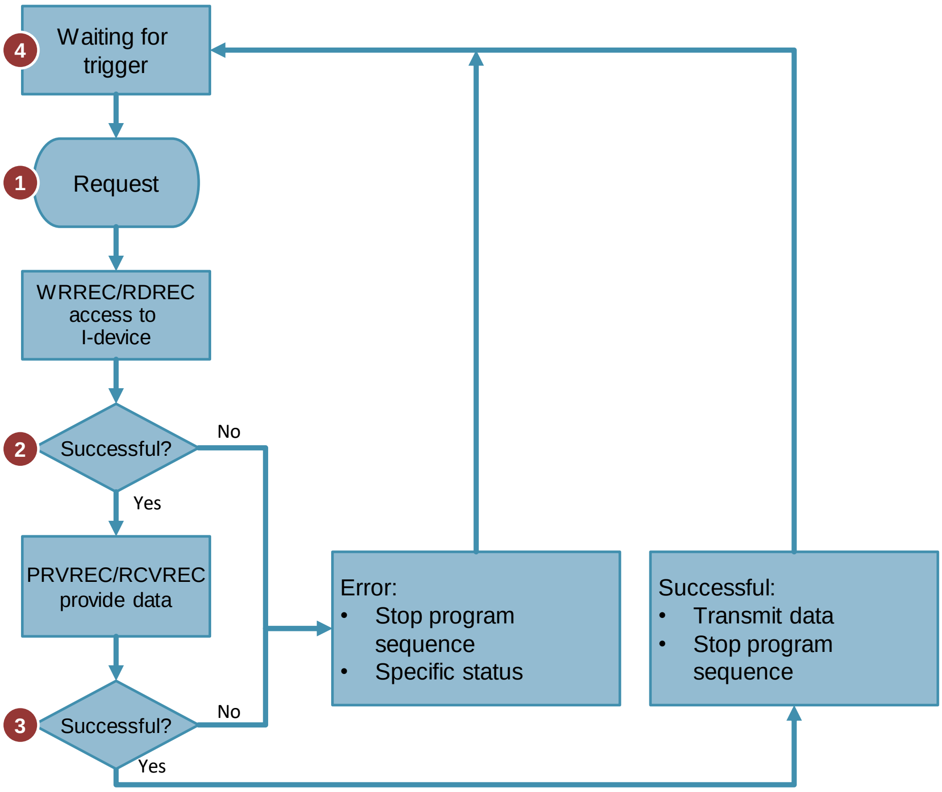


Table 2-1

	Action	Note
1.	The controller initiates the job using the commands WRREC or RDREC that access the I-device's transfer areas and thus make a request.	Is triggered by the user or the user program. The transfer areas are defined by a HW identifier.
2.	Access to the transfer area is checked.	**Error:** the program sequence is stopped and the specific status message is output. **Successful:** the program sequence is continued.
3.	The I-device processes the request and provides the data through PRVREC or stores the data record through RCVREC.	**Error:** the program sequence is stopped and the specific status message is output. **Successful:** the data were transferred successfully and the program sequence was finished.
4.	Waiting for new trigger.	

2.3 Hardware and software components

2.3.1 Validity

This application is valid for

- STEP 7 as of V13 SP1
- S7-1500 V1.8 and higher

Note The I-device blocks used are also available for S7-300 / S7-400 / S7-1500 (FW < V1.8). The controllers cannot be operated by the web server used in this application example.

As an alternative to the web server, the instruction can be controlled by the user program or a watch table.

The function of the user program can also be used with FW<1.8.

2.3.2 Components used

The application was created with the following components:

Hardware components

Table 2-2

Component	No.	Article number	Note
SIMATIC S7-1500 CPU 1511-1 PN	2	6ES7 511-1AK00-0AB0	Alternatively, any other SIMATIC S7-1500 can be used. FW > 1.8 is necessary to be able to operate the application example using the web server.

Software components

Table 2-3

Component	No.	Article number	Note
STEP 7 PROFESSIONAL V15.1	1	6ES7 822-1AA02-0YA5	

Example files and projects

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

Table 2-4

Component	Note
109478798_datacom_IDevice_CODE_V10_U.zip	This zip file contains the STEP 7 project.
109478798_datacom_IDevice_DOKU_ v20_d.pdf	This document.

Compatible web browsers

The following web browsers were tested for the web server:

- Internet Explorer (version 8 to 11)
- Mozilla Firefox (version 22 to 32)
- Google Chrome (version 33 to 38)
- Mobile Safari and Chrome for iOS (iOS 8)
- Android Browser and Android Chrome (JellyBean operating system)

3 Basics

3.1 Read/ write blocks in the IO controller

RDREC

This block allows you to read data records quickly with little configuration effort.

For reading to be successful, the parameters "ID”, "INDEX”, "MLEN” and "RECORD” have to be supplied with a valid value.

Once the parameters have been determined correctly, input parameter "REQ” starts the job.

WRREC

This block allows you to write data records quickly with little configuration effort.

For writing to be successful, the parameters "ID”, "INDEX”, "LEN” and "RECORD” have to be supplied with a valid value.

Once the parameters have been determined correctly, input parameter "REQ” starts the job.

Note For more information on the blocks "RDREC” and "WRREC”, see the TIA help.

3.2 Provision/reception blocks of the I-device

3.2.1 Provision block "PRVREC”

General

The I-device uses this block to provide data records to the IO controller.

First, the instruction checks whether the IO controller has made a request to provide data records through the "RDREC” instruction. If a request from the IO controller has been received, the data record is transferred to the higher-ranking IO controller. If data transfer has been successful, the instruction sends a response to the IO controller to finish the transfer.

To perform these different tasks, the "PRVREC” block has different operating modes that can be specified by the input parameter "MODE”.

Table 3-1

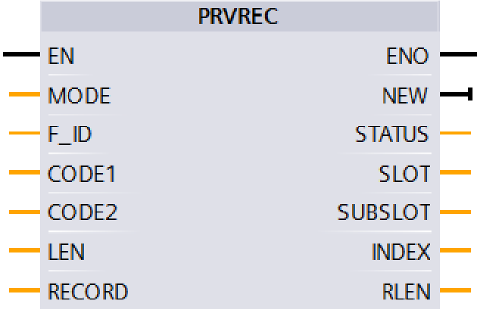
MODE	Explanation
0	Checks whether there is a request of the IO controller for the provision of a data record (NEW=true).
1	Reception of a request to provide a data record for any of the I-device's sub slots.
2	Reception of a request to provide a data record for a specific sub slot of the I-device.
3	Provide data record and send positive response to the higher-ranking IO controller.
4	Send negative response to the higher-ranking IO controller.

In these different operating modes, the block has to be called successively in the user program. For more information on calling "PRVREC”, see chapter 4.1.

Parameter interface

The following figure shows the call of the "PRVREC” block.

Figure 3-1



Subsequently, all parameters for the "PRVREC” block are explained:

Inputs

The following table describes the inputs of the "PRVREC” block.

Table 3-2

Name	Data type	Description
MODE	Int	Defines the operating mode.
F_ID	HW_SUB MODULE	Sub slot in the transfer area of the I-device for the data record to be sent (only for MODE=2).
CODE1	BYTE	In case of successful provision zero (MODE=3). In case of faulty provision "Error Code 1” (MODE=4).
CODE2	BYTE	In case of successful provision zero (MODE=3). In case of faulty provision "Error Code 2” (MODE=4).
LEN	UInt	Maximum length of the data record to be sent in bytes.
RECORD	VARIANT	INOUT: Provided data record. The DB parameters always have to be specified completely for the S7-300/400 (ex.: P#DB13.DBX0.0 Byte 100)

Outputs

The following table describes the outputs of the "PRVREC” block.

Table 3-3

Name	Data type	Description
NEW	Bool	Requesting a new data record from the higher ranking IO controller.
STATUS	DWORD	Specific status of the order.
SLOT	HW_SUB MODULE	Identical with "F_ID”.
SUBSLOT	HW_SUB MODULE	Identical with "F_ID”.
INDEX	UInt	Number of the data record to be sent (in case of pre-built data records).
RLEN	UInt	Real length of the data record to be sent.

3.2.2 Receive block "RCVREC”

This block enables receiving I-device data records of the IO controller.

First, the instruction checks whether the IO controller has made a request to receive data records through the "WRREC” instruction. If a request from the IO controller has been received, the data record is received by the higher-ranking IO controller. If data reception has been successful, the instruction sends a response to the IO controller to finish the transfer.

To perform these different tasks, the "RCVREC” block has different operating modes that can be specified by the input parameter "MODE”.

Table 3-4

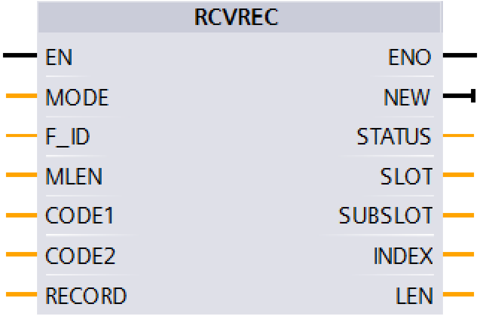
MODE	Explanation
0	Checks whether there is a request of the IO controller for the reception of a data record (NEW=true).
1	Receiving a data record for any sub slot of the I-device.
2	Receiving a data record for a specific sub slot of the I-device.
3	Send positive response to the higher-ranking IO controller.
4	Send negative response to the higher-ranking IO controller.

In three of these different operating modes, the block has to be called successively in the user program. For more information on calling "RCVREC”, see chapter 4.1.

Parameter interface

The following figure shows the call of the "RCVREC” block.

Figure 3-2



Subsequently, all parameters for the "RCVREC” block are explained:

Inputs

The following table describes the inputs of the "RCVREC” block.

Table 3-5

Name	Data type	Description
MODE	Int	Defines the operating mode.
F_ID	HW_SUB MODULE	Sub-slot in the transfer area of the I-device for the data record to be received (only for MODE=2).
CODE1	BYTE	In case of successful provision zero (MODE=3). In case of faulty provision "Error Code 1” (MODE=4).

Name	Data type	Description
CODE2	BYTE	In case of successful provision zero (MODE=3). In case of faulty provision "Error Code 2" (MODE=4).
MLEN	UInt	Maximum length of the data record to be received in bytes.
RECORD	VARIANT	INOUT: Target area for the received data record. The DB parameters always have to be specified completely for the S7-300/400 (ex.: P#DB13.DBX0.0 Byte 100)

Outputs

The following table describes the outputs of the "RCVREC" block.

Table 3-6

Name	Data type	Description
NEW	Bool	TRUE when requesting the reception of a data record.
STATUS	DWORD	Specific status of the order.
SLOT	HW_SUB MODULE	Identical with "F_ID".
SUBSLOT	HW_SUB MODULE	Identical with "F_ID".
INDEX	UInt	Number of the data record received (in case of pre-built data records).
LEN	UInt	Real length of the data record to be received.

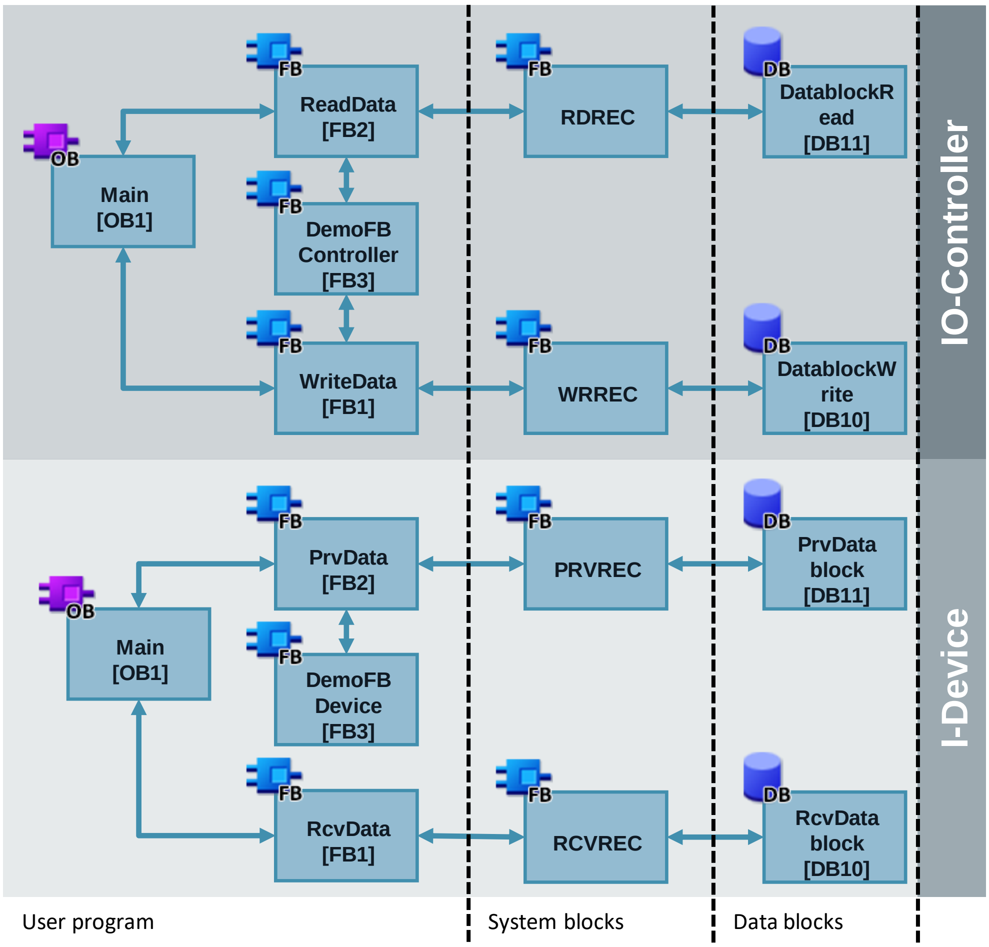
4 Mode of Operation

4.1 General overview

Program structure

The figure below shows the program structure of the S7 user program.

Figure 4-1



Description

In the first example, data records are sent from an IO controller to an I-device. This is accomplished in the controller by using the "WRREC" block. This block sends a request to receive data to the I-device through the I-device's transfer area. The request is processed in the user program of the I-device by means of the "RCVREC" instruction.

In the second example, an IO controller requests data records from an I-device. The job is initiated on the side of the IO controller through the "RDREC" block. This block sends a request to provide data to the I-device over the I-device's transfer area. The request is processed in the user program of the I-device by means of the "PRVREC" instruction.

The blocks can send either pre-built data records or own data records. In this application example, two DBs respectively are used on the side of the IO controller ("DatablockRead", "DatablockWrite") and on the side of the I-device ("PrvDatablock", "RcvDatablock").

Note Read and write processes can only be started from the IO controller.

Program details on data record request

To process the jobs of the IO controller, the system block "PRVREC” has to be called in different modes.

The following illustration shows the sequence of requesting a data record from the "I-device” through the "IO controller”:

Figure 4-2

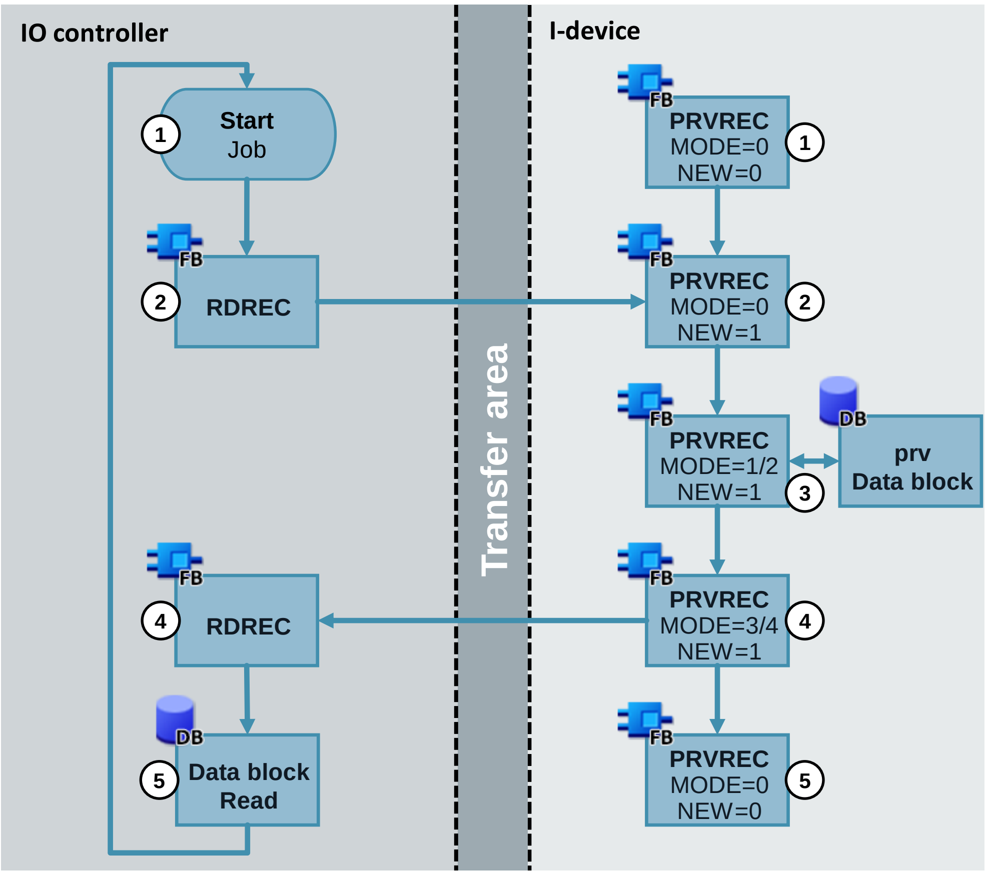


Table 4-1

No.	IO controller	I-device
1.	The user starts the job through the user program or the tag table.	"PRVREC” is in mode 0 and is waiting for requests by the IO controller.
2.	"RDREC” requests a data record. Communication with the I-device and thus with "PRVREC” is accomplished through the transfer area of the I-device.	"PRVREC” receives a data record request.
3.		"PRVREC” writes to the output parameters and transfers the data record selected previously into the parameter "RECORD”.

No.	IO controller	I-device
4.	"RDREC" receives a positive/negative response from "PRVREC" and transfers the data record to its "RECORD" parameter if the response is positive.	"PRVREC" sends a positive/negative response to the IO controller.
5.	The data record was transferred to the target area specified previously.	The block is again in mode 0 and is waiting for requests by the IO controller.

Program details of writing a data record

To process the jobs of the IO controller, the system block "RCVREC" has to be called in different modes.

The following illustration shows the sequence of sending a data record from the IO controller to the I-device:

Figure 4-3

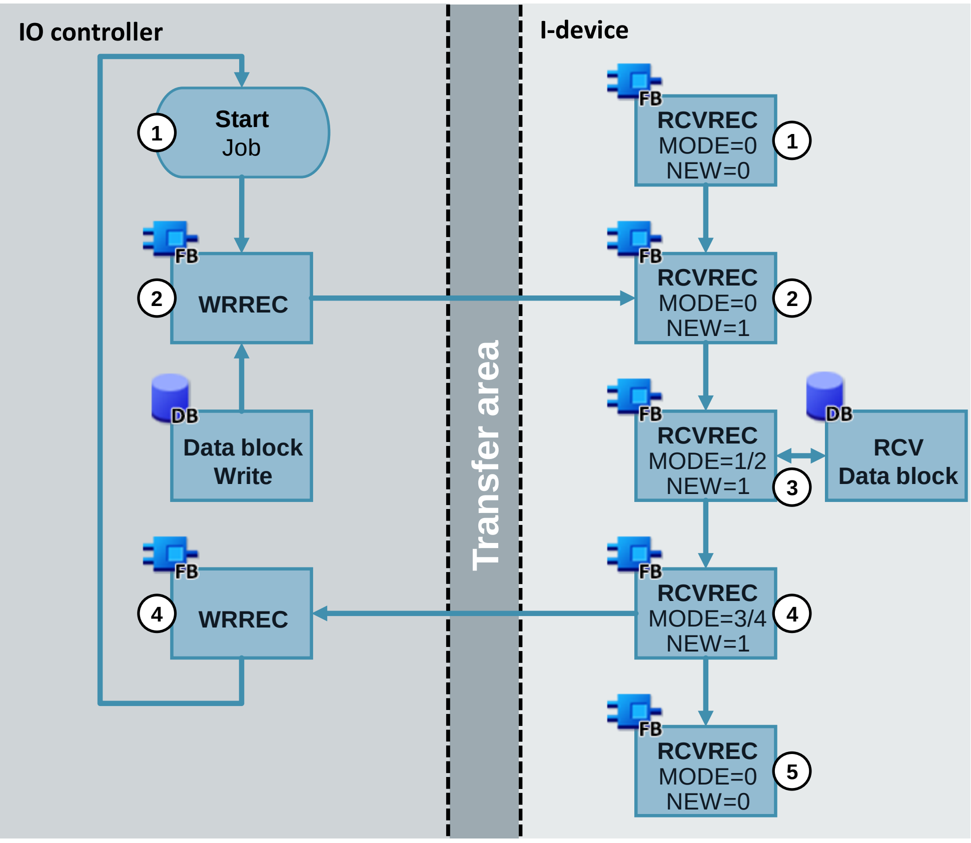


Table 4-2

No.	IO controller	I-device
1.	The user starts the job through the user program or the tag table.	"RCVREC" is in mode 0 and is waiting for an order by the "IO controller".
2.	"WRREC" sends a data record. Communication with the I-device and thus with "RCVREC" is accomplished through the transfer area of the I-device.	"RCVREC" receives the data record that was sent.
3.		"RCVREC" writes to the output parameters and transfers the data record selected previously to the

No.	IO controller	I-device
		parameter "RECORD”. The data record was transferred to the specified target area.
4.	"WRREC” receives a positive / negative response from "RCVREC”.	"RCVREC” sends a positive / negative response to the IO controller.
5.		The block is again in mode 0 and is waiting for orders by the IO controller.

Program details for "demoFBxx”

The function blocks "demoFBxx” simulate the changing data records and additionally measure the time of the read and write processes.

Jobs of the "demoFBxx” function block in the IO controller:

- Measuring the time needed for the read process (output "timeRead”) or write process (output "timeWrite”).
- Changing the data of the data record ("DatablockWrite”) to be sent after each write action.

Jobs of the "demoFBDevice” function block in the I-device:

- Changing the data of the provided data record ("PrvDatablock”) after each read process.

Note

The application example can also be executed without the function blocks "demoFBxx”.

Prerequisite

The prerequisite for this application example is the configuration of a communication between IO controller and I-device.

Click here to get to the configuration guide.

4.2 Program description of the IO controller

4.2.1 Program details on "WriteData"

General

The "WriteData" function block is the user's interface to the "WRREC" system block.

The user can freely define the source area, the index and the length of the data record to be written of the transfer area used. The hardware identifier of the transfer area is fixed in the device configuration. The outputs provide information on the current status of the block.

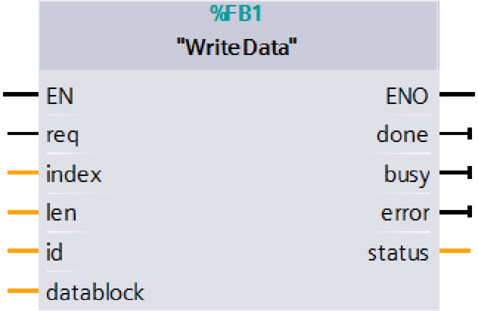
The write process is started once the user has made a request.

In this application example, the DB "DatablockWrite" was created for the source area of the data record to be written. It is possible to send pre-built records or own data records.

Parameter interface

The following figure shows the call of the "WriteData" block.

Figure 4-4



Subsequently, the parameters of the "WriteData" block are explained.

Inputs

The following table describes the inputs of the "WriteData" block.

Table 4-3

Name	Data type	Description
req	Bool	Job triggered at positive edge
index	DInt	Number of the data record to be sent (used in case of pre-built data records / irrelevant for own data records).
len	UInt	Maximum length of the data record to be sent.
id	HW_IO	Hardware identifier of the I-device's transfer area used.
datablock	Variant	Pointer to the source area of the data record to be sent.

Outputs

The following table describes the outputs of the "WriteData” block.

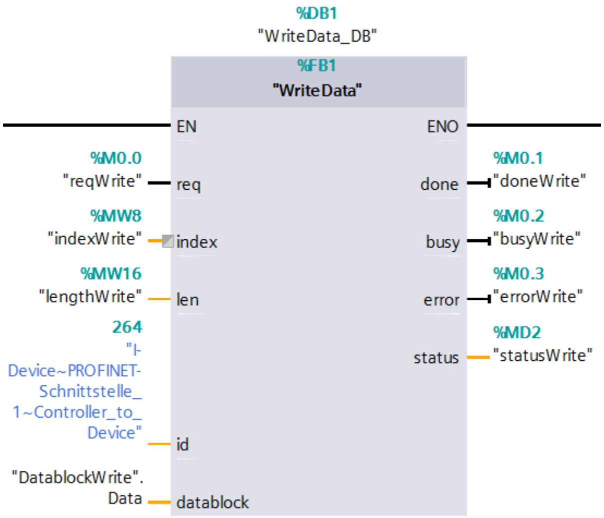
Table 4-4

Name	Data type	Description
done	Bool	TRUE if job was completed successfully.
busy	Bool	TRUE while the job is running.
error	Bool	TRUE if the job has failed.
status	DWord	Outputs the job status as a HEX code

Block call in this application example

The following figure shows the call of the "WriteData” block.

Figure 4-5



Note

Use the HW identifier stored in the system constants for the input "id”. If a transfer area was created, its HW identifier will be stored in the system constants and can be used symbolically.

"PLC tags” > "Show all tags” > "System constants”

4.2.2 Program details on "ReadData”

General

The "ReadData” function block is the user's interface to the "RDREC” system block.

As for the "WriteData” block, the user can freely define the target area, the index and the length of the requested data record and the hardware identifier of the sub-module used. The outputs provide information on the current status of the block.

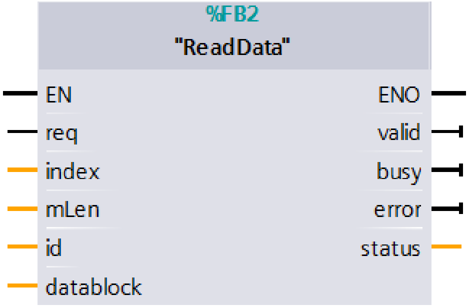
The data record is requested once the user has made a request.

In this application example, the DB "DatablockRead” was created for the target area of the data record to be written. It is possible to receive pre-built records or own data records.

Parameter interface

The following table describes the inputs of the "ReadData” block.

Figure 4-6



Subsequently, the parameters of the "ReadData” block are explained.

Inputs

The following table describes the inputs of the "ReadData” block.

Table 4-5

Name	Data type	Description
req	Bool	Job triggered at positive edge
index	DInt	Number of the data record to be received (used in case of pre-built data records / irrelevant for own data records).
mLen	UInt	Maximum length of the data record to be received.
id	HW_IO	Hardware identifier of the I-device's transfer area used.
datablock	Variant	Pointer to the target area of the data record to be received.

Outputs

The following table describes the outputs of the "ReadData” block.

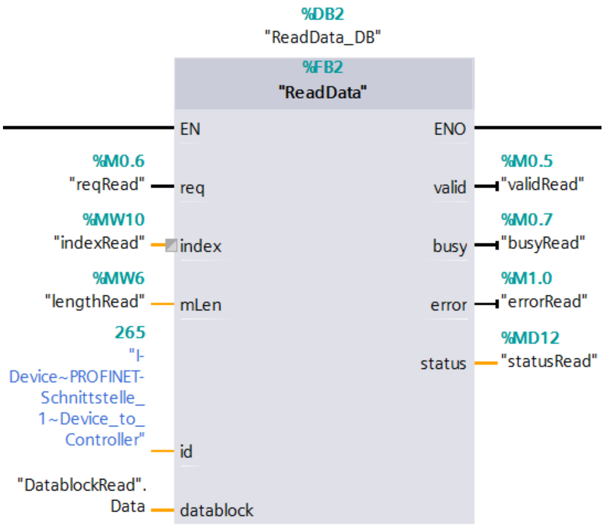
Table 4-6

Name	Data type	Description
valid	Bool	TRUE if job was completed successfully.
busy	Bool	TRUE while the job is running.
error	Bool	TRUE if the job has failed.
status	DWord	Outputs the job status as a HEX code

Block call in this application example

The following figure shows the call of the "ReadData” block.

Figure 4-7



Note

Use the HW identifier stored in the system constants for the input "id”. If a transfer area was created, its HW identifier will be stored in the system constants and can be used symbolically.

"PLC tags” > "Show all tags” > "System constants”

4.3 Program description of the I-device

4.3.1 Details on "PrvData”

General

The "PrvData” function block is the user's interface to the "PRVREC” system block. The different modes of "PRVREC” are passed through using a sequencer.

The user can define the maximum length and the source area of the source area to be sent. The outputs provide information on the current status of the block and on the data record to be sent.

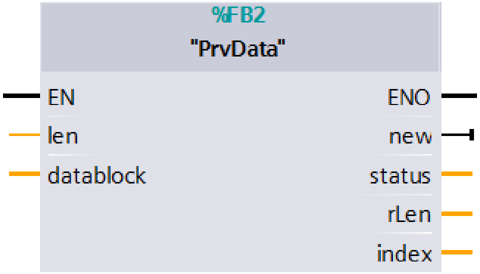
The data record is provided once the IO controller has requested data through the "ReadData” block.

In this application example, the DB "PrvDatablock” was created for the source area of the provided data record. It is possible to provide pre-built data records or own data records.

Parameter interface

The following table describes the inputs of the "PrvData” block.

Figure 4-8



Subsequently, the parameters of the "PrvData” block are explained.

Inputs

The following table describes the inputs of the "PrvData” block.

Table 4-7

Name	Data type	Description
len	UInt	Maximum length of the data record to be provided.
datablock	Variant	Pointer to the source area of the data record provided.

Outputs

The following table describes the outputs of the "PrvData” block.

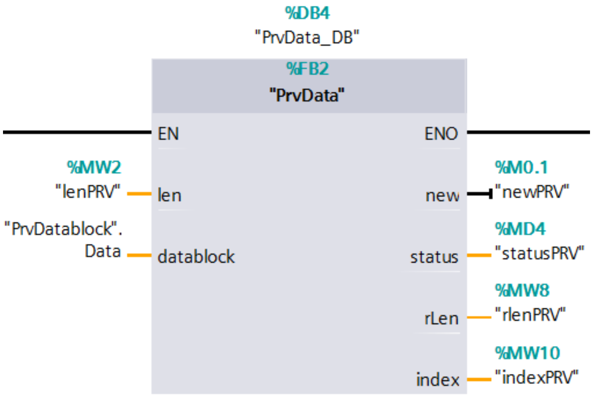
Table 4-8

Name	Data type	Description
new	Bool	TRUE if a new data record was requested.
status	DWord	Outputs the job status as a HEX code
rLen	UInt	Specifies the length of the provided data record.
index	UInt	Specifies the number of the provided data record (used in case of pre-built data records / irrelevant for own data records).

Block call in this application example

The following figure shows the call of the "PrvData” block.

Figure 4-9



4.3.2 Details on "RcvData”

General

The "RcvData” function block is the user's interface to the "RCVREC” system block.

The user can define the maximum length and the target area of the data record to be received. The outputs provide information on the current status of the block and on the data record received.

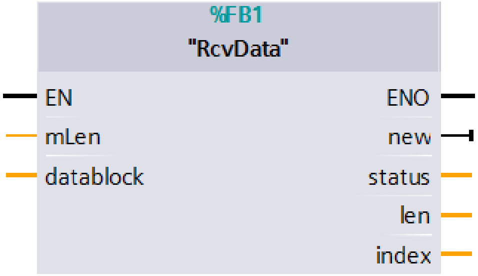
The data record is accepted once a data record has been sent by the IO controller through the "WriteData” block.

In this application example, the DB "RcvDatablock” was created for the target area of the data record to be received. It is possible to receive pre-built records or own data records.

Parameter interface

The following table describes the inputs of the "RcvData” block.

Figure 4-10



Subsequently, the parameters of the "RcvData” block are explained.

Inputs

The following table describes the inputs of the "RcvData” block.

Table 4-9

Name	Data type	Description
len	UInt	Maximum length of the data record to be received.
datablock	Variant	Pointer to the target area of the data record received.

Outputs

The following table describes the outputs of the "RcvData” block.

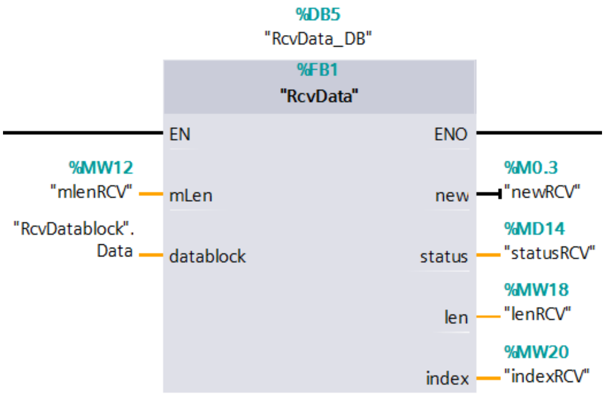
Table 4-10

Name	Data type	Description
new	Bool	TRUE when requesting the reception of a data record.
status	DWord	Outputs the job status as a HEX code
rLen	UInt	Specifies the length of the received data record.
index	UInt	Specifies the number of the received data record (used in case of pre-built data records / irrelevant for own data records).

Block call in this application example

The following figure shows the call of the "RcvData” block.

Figure 4-11



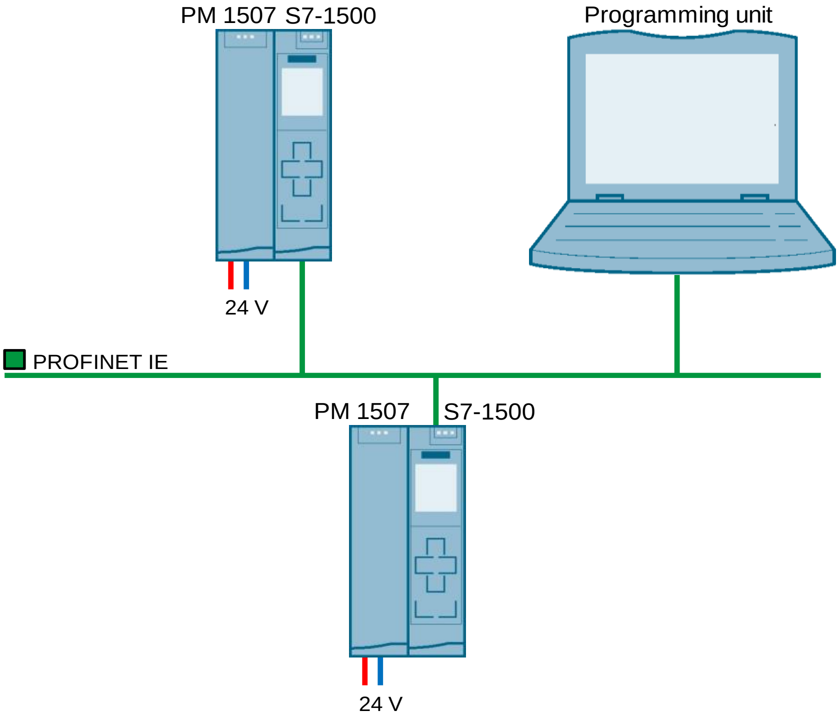
5 Installation and Commissioning

5.1 Hardware and Software Installation

Hardware installation

The figure below shows the hardware configuration of the application.

Figure 5-1



ATTENTION	Always follow the installation guidelines for all components.

Table 5-1

No.	Action
1.	Connect the controllers to a 24V power supply.
2.	Connect the programming unit to a 230V power supply.
3.	Connect each component with an Ethernet cable (RJ45).

Installing the software

This chapter describes the steps necessary for installing the example code.

Note It is recommended to run the latest versions of any installed software.

Table 5-2

No.	Action	Remarks
1.	Install STEP 7 V15.1 on your programming unit.	Follow the instructions of the program.

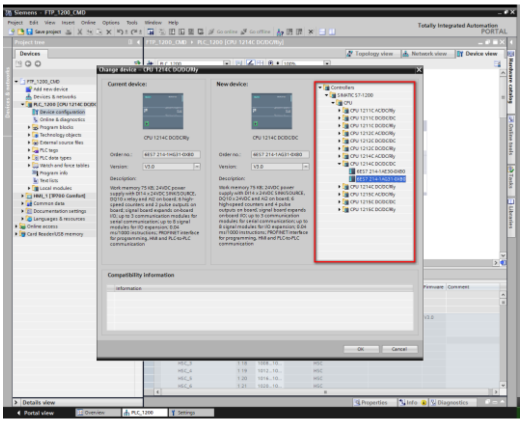
5.2 Commissioning

5.2.1 IO controller and I-device

IO controller

The table below shows how to commission the IO controller:

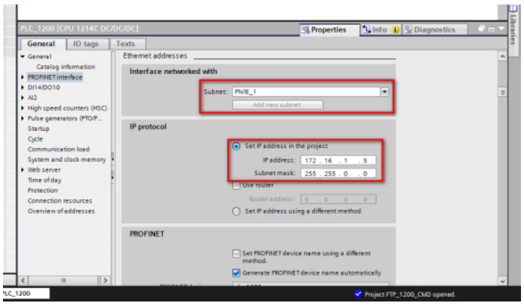
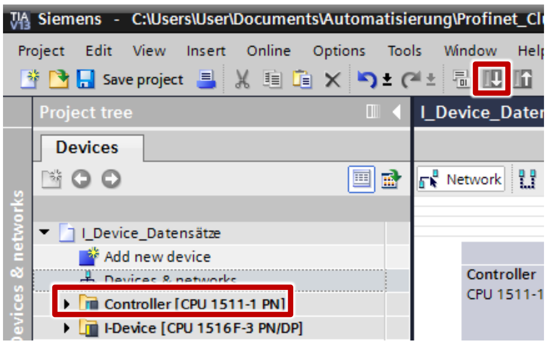
Table 5-3

No.	Action	Picture / Note
1.	Load the application example to your programming unit and unzip the archive.	Click here for the code to the application example.
2.	Open the example project	109478798_datacom_IDevice_CODE_V10_U
3.	Open the "Device Configuration" of the controller ("IO controller"). When using the same controller as in the example, proceed with step 5.	SIMATIC S7-1500 CPU 1511-1 PN
4.	Right-click the controller and click "Change device". Select your S7-1500 from the tree and confirm with "OK".	

Change device - CPU 1214C DC/DC/Rly

Current device: CPU 1214C DC/DC/Rly
Order no.: 6ES7 214-1HG31-0XB0
Version: V3.0
Description: Work memory 75 KB; 24VDC power supply with DI14 x 24VDC SINK/SOURCE, DQ10 x relay and AI2 on board; 6 high-speed counters and 2 pulse outputs on board; signal board expands on-board I/O; up to 3 communication modules for serial communication; up to 8 signal modules for I/O expansion; 0.04 ms/1000 instructions; PROFINET interface for programming, HMI and PLC-to-PLC communication

New device: CPU 1214C DC/DC/DC
Order no.: 6ES7 214-1AG31-0XB0
Version: V3.0
Description: Work memory 75 KB; 24VDC power supply with DI14 x 24VDC SINK/SOURCE, DQ10 x 24VDC and AI2 on board; 6 high-speed counters and 4 pulse outputs on board; signal board expands onboard I/O; up to 3 communication modules for serial communication; up to 8 signal modules for I/O expansion; 0.04 ms/1000 instructions; PROFINET interface for programming, HMI and PLC-to-PLC communication

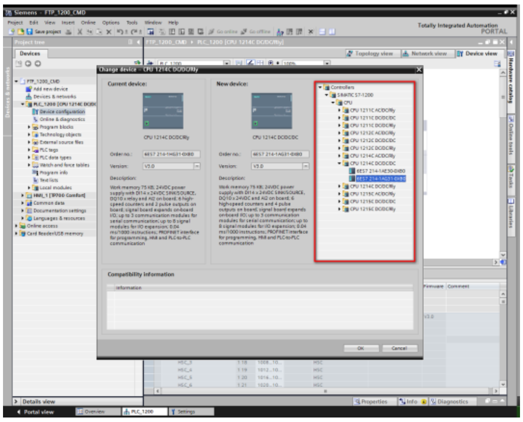
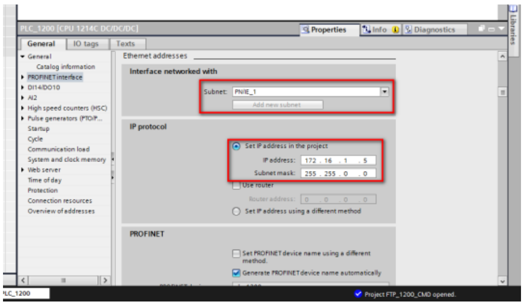
No.	Action	Picture / Note
5.	Right-click the controller and click "Properties". In the subnavigation of the inspector window, select "PROFINET interface". Enter an IP address and the subnet mask under "Ethernet addresses". Make sure that these match the IP address and subnet mask of the programming unit. Connect the controller with a subnet.	 PLC_1200 [CPU 1214C DC/DC/DC] Ethernet addresses Interface networked with Subnet: PN/IE_1 IP protocol ● Set IP address in the project IP address: 172 . 16 . 1 . 5 Subnet mask: 255 . 255 . 0 . 0 □ Use router Router address: 0 . 0 . 0 . 0 ○ Set IP address using a different method PROFINET □ Set PROFINET device name using a different method. ☑ Generate PROFINET device name automatically Project FTP_1200_CMD opened. Properties Info Diagnostics Libraries General IO tags Texts General Catalog information PROFINET interface DI14/DO10 AI2 High speed counters (HSC) Pulse generators (PTO/P... Startup Cycle Communication load System and clock memory Web server Time of day Protection Connection resources Overview of addresses Add new subnet PLC_1200
6.	In the project navigation, select the controller and download the program into the controller. Configuration of the IO controller is complete.	 Siemens - C:\Users\User\Documents\Automatisierung\Profinet_Cl TIA V13 Project Edit View Insert Online Options Tools Window Help Save project Project tree Devices I_Device_Datensätze Add new device Devices & networks Controller [CPU 1511-1 PN] I-Device [CPU 1516F-3 PN/DP] Devices & networks I_Device_Daten Network Controller CPU 1511-1

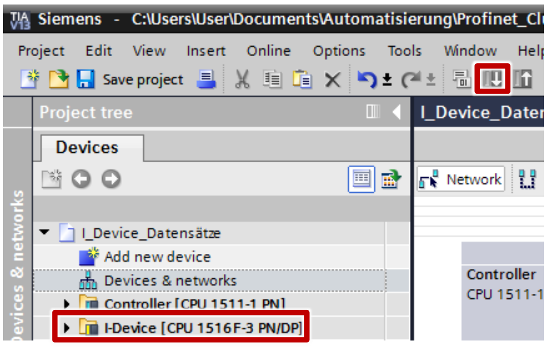
I-device

Note Make sure that the I-device was parameterized as an I-device and that the transfer areas were created so that the necessary preconditions are satisfied.

The table below shows how to commission the I-device:

Table 5-4

No.	Action	Picture / Note
1.	Open the "Device Configuration" of the controller ("I-Device"). When using the same controller as in the example, proceed with step 3.	SIMATIC S7-1500 CPU 1511-1 PN
2.	Right-click the controller and click "Change device". Select your S7-1500 from the tree and confirm with "OK".	 Change device - CPU 1214C DC/DC/Rly Current device: CPU 1214C DC/DC/Rly, Order no.: 6ES7 214-1HG31-0XB0, Version: V3.0 New device: CPU 1214C DC/DC/DC, Order no.: 6ES7 214-1AG31-0XB0, Version: V3.0 Controllers > SIMATIC S7-1200 > CPU > CPU 1214C DC/DC/DC > 6ES7 214-1AG31-0XB0 OK / Cancel
3.	Right-click the controller and click "Properties". In the subnavigation of the inspector window, select "PROFINET interface". Enter an IP address and the subnet mask under "Ethernet addresses". Make sure that these values match the IP address and subnet mask of the programming unit and of the IO controller. Connect the controller with the same subnet as the IO controller.	 PLC_1200 [CPU 1214C DC/DC/DC] Ethernet addresses Interface networked with Subnet: PN/IE_1 IP protocol ● Set IP address in the project IP address: 172 . 16 . 1 . 5 Subnet mask: 255 . 255 . 0 . 0 □ Use router Router address: 0 . 0 . 0 . 0 ○ Set IP address using a different method PROFINET □ Set PROFINET device name using a different method. ☑ Generate PROFINET device name automatically Project FTP_1200_CMD opened.

No.	Action	Picture / Note
4.	In the project navigation, select the controller and download the program into the controller. The configuration of the I-device is complete.	

Siemens - C:\Users\User\Documents\Automatisierung\Profinet_Cl
Project Edit View Insert Online Options Tools Window Help
Save project
Project tree
Devices
I_Device_Dater
Network
Devices & networks
I_Device_Datensätze
Add new device
Devices & networks
Controller [CPU 1511-1 PN]
I-Device [CPU 1516F-3 PN/DP]
Controller
CPU 1511-1

5.2.2 Modifications at the S7 program

The hardware identifiers of the transfer areas that enable the communication between the IO controller and the I-device are defined in the user program.

They are located in OB1 "Main" of the IO controller at the corresponding input parameters "id" of the blocks "WriteData" and "ReadData".

If you use different transfer areas than in the application example, you can adapt these parameters.

Modifications to the user program

The maximum size of the sent or requested data records is set to 1024 bytes in this application example. You can adjust the size by increasing or decreasing the transfer areas of the I-device.

6 Operating the Application Example

General

The web server of the controllers is used to operate the application example.

Note You need firmware version 1.8 and TIA Portal STEP 7 V15.1 for S7-1500 controllers!

Alternatively, you can control and monitor the application example directly in the TIA Portal using a watch table.

To use the web server, all you need is a PG/PC and a web browser.

The web server allows monitoring and managing the CPU through authorized users over a network. Evaluations and diagnoses are thus possible over large distances and without STEP 7.

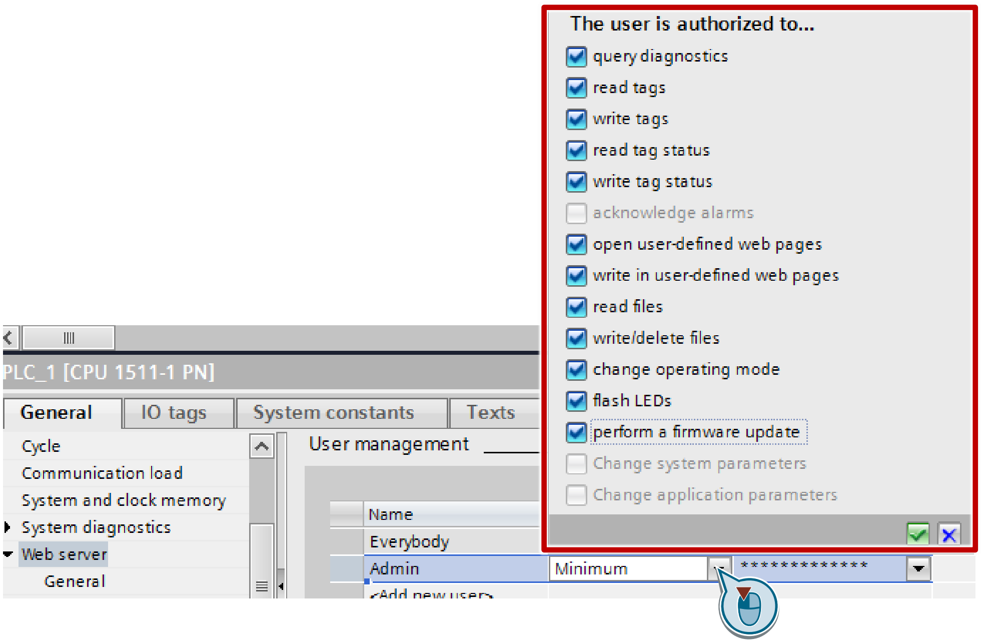
Note For more information on the web server, see here \3\.

Rights assignment on the web server

The watch table on the web server can only be controlled by a user who has the corresponding rights.

You can assign the rights in the controller settings at "Web server" > "User management".

Figure 6-1



The user is authorized to...

- ☑ query diagnostics
- ☑ read tags
- ☑ write tags
- ☑ read tag status
- ☑ write tag status
- □ acknowledge alarms
- ☑ open user-defined web pages
- ☑ write in user-defined web pages
- ☑ read files
- ☑ write/delete files
- ☑ change operating mode
- ☑ flash LEDs
- ☑ perform a firmware update
- □ Change system parameters
- □ Change application parameters

Name		
Everybody		
Admin	Minimum	*************
<Add new user>		

Operation

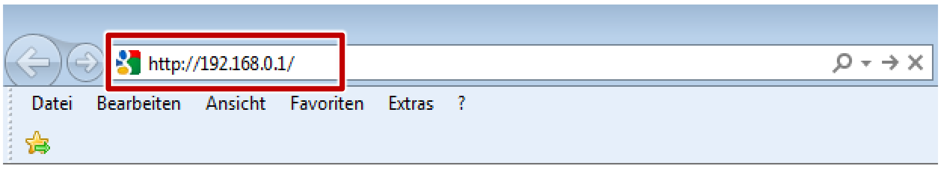
Note The following illustrations show the operation from the IO controller.

The web server is activated both on the IO controller and the I-device.

This application example is controlled from the IO controller and is thus operated from there as well. The following steps show how to start the different modes:

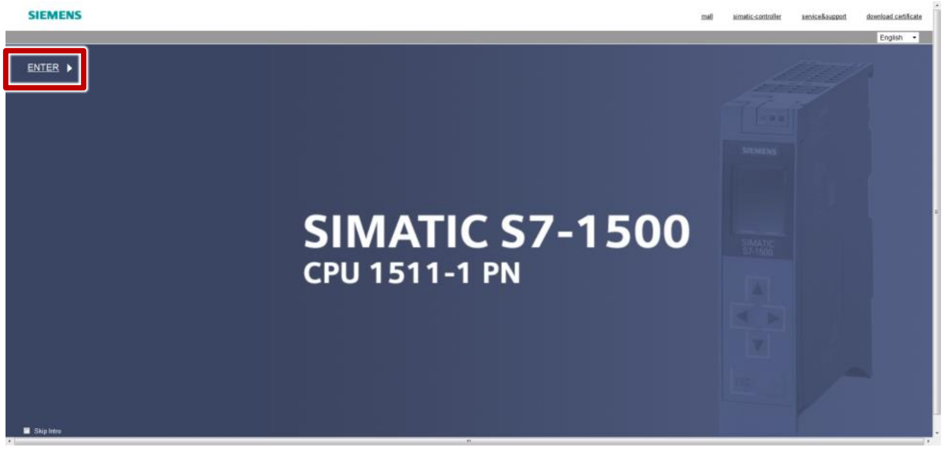
1. Start your web browser.
2. Enter the IP address of the IO controller in the browser address bar.

Figure 6-2

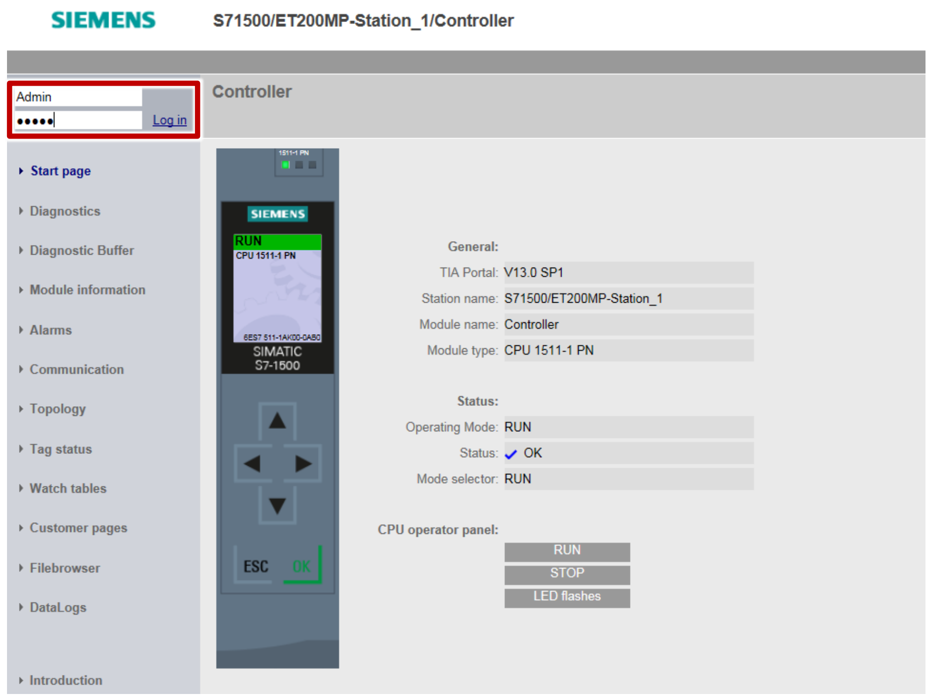


3. Click "ENTER” on this screen to continue.

Figure 6-3



4. Login with the following data: User name: "Admin” password: "admin”.



S71500/ET200MP-Station_1/Controller

Admin
•••••

Controller

General:

TIA Portal: V13.0 SP1

Station name: S71500/ET200MP-Station_1

Module name: Controller

Module type: CPU 1511-1 PN

Status:

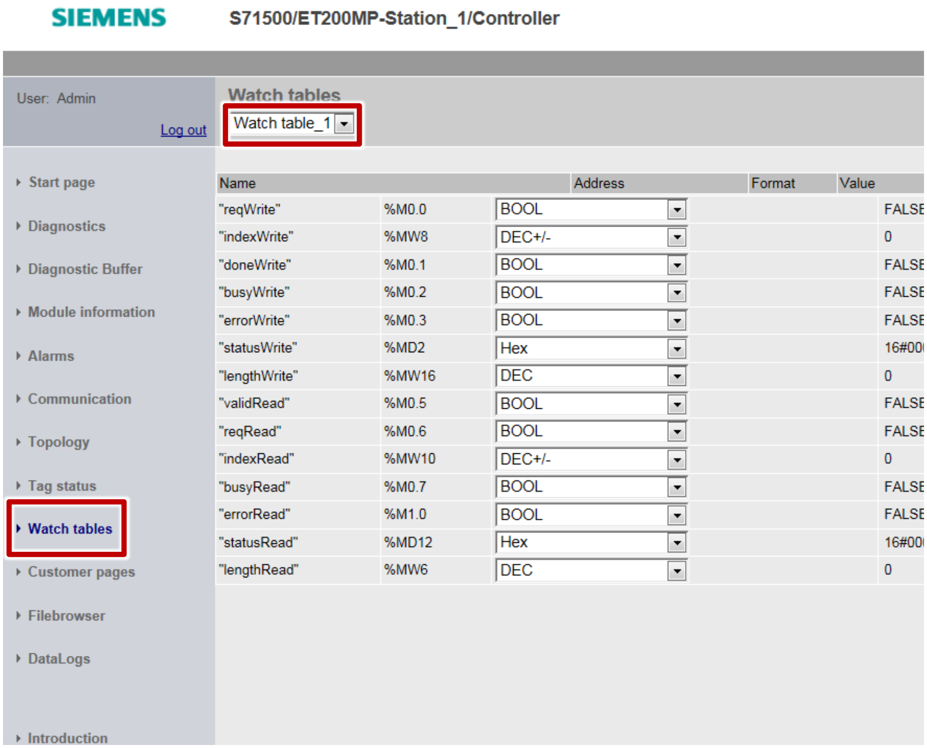
Operating Mode: RUN

Status: ✓ OK

Mode selector: RUN

You have logged on to the web server and can use the web server functions based on your rights.

5. Click "Watch tables” to see the added watch tables. Here you can now operate the application example and start jobs.



Name	Address	Format	Value
"reqWrite"	%M0.0	BOOL	FALSE
"indexWrite"	%MW8	DEC+/-	0
"doneWrite"	%M0.1	BOOL	FALSE
"busyWrite"	%M0.2	BOOL	FALSE
"errorWrite"	%M0.3	BOOL	FALSE
"statusWrite"	%MD2	Hex	16#00
"lengthWrite"	%MW16	DEC	0
"validRead"	%M0.5	BOOL	FALSE
"reqRead"	%M0.6	BOOL	FALSE
"indexRead"	%MW10	DEC+/-	0
"busyRead"	%M0.7	BOOL	FALSE
"errorRead"	%M1.0	BOOL	FALSE
"statusRead"	%MD12	Hex	16#00
"lengthRead"	%MW6	DEC	0

The following table describes how to operate the different scenarios.

Table 6-1

No.	Description
1.	Specify the used data record number at the variable "indexWrite” / "indexRead”. When using own data records, set the variable to 0.
2.	Specify the length of the data record used at the variable "lengthWrite” / "lengthRead”.
3.	Set the variable "reqWrite” / "readWrite” to TRUE to start a read/write process.
4.	If the read/write process has been completed successfully, the variable "doneWrite” / "validRead” is TRUE. If an error has occurred, the variable "errorWrite” / "errorRead” is TRUE and the status of the error that has occurred is output at the variable "statusWrite” / "statusRead”.

7 Related Literature

Table 7-1

	Topic	Title
\1\	Siemens Industry Online Support	http://support.industry.siemens.com
\2\	Download page of the entry	https://support.industry.siemens.com/cs/ww/de/view/109478798https://support.industry.siemens.com/cs/ww/en/view/109478798
\3\	Web server	https://support.industry.siemens.com/cs/de/en/view/59193560

8 History

Table 8-1

Version	Date	Modifications
V1.0	09/2015	First version
V2.0	07/2019	Update TIA Portal V15.1