

**SIEMENS**



Configuration Example • 08/2015

# I-Device Function in Standard PN Communication

SIMATIC S7-CPU, CP, SIMOTION, SINUMERIK

<https://support.industry.siemens.com/cs/ww/en/view/109478798>

## 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 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 will be excluded. Such an exclusion will 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 compensation for damages due to a breach of a fundamental contractual obligation is, 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 <http://support.automation.siemens.com>.

# Table of Contents

<b>Warranty and Liability .....</b>	<b>2</b>
<b>1 Task and Solution .....</b>	<b>4</b>
1.1 Task Description.....	4
1.2 Possible solution .....	4
1.3 Application area and usage.....	6
<b>2 Basics on the I-Device.....</b>	<b>7</b>
2.1 Description of the functionality .....	7
2.2 Principle of data exchange .....	9
2.3 Diagnostics and alarm behavior.....	10
2.4 Information on topology and use.....	11
2.5 Description of PNIO_SEND and PNIO_RECV.....	12
<b>3 Configuration .....</b>	<b>13</b>
3.1 S7-CPU as I-Device .....	13
3.1.1 Infrastructure information .....	13
3.1.2 Implementing and configuring the devices.....	14
3.1.3 Configuring the I-Device function .....	17
3.1.4 Address assignment and loading .....	20
3.1.5 Testing the I-Device function.....	21
3.2 SIMATIC-NET CP as an I-Device .....	22
3.2.1 Infrastructure information .....	22
3.2.2 Implementing and configuring the devices.....	23
3.2.3 Configuring the I-Device function .....	26
3.2.4 Address assignment and loading .....	29
3.2.5 Testing the I-Device function.....	30
3.3 SIMOTION as I-Device.....	31
3.3.1 Infrastructure information .....	31
3.3.2 Implementing and configuring the devices.....	32
3.3.3 Configuring the I-Device function .....	34
3.3.4 Configuration in SIMOTION SCOUT TIA .....	38
3.3.5 Address assignment and loading .....	42
3.3.6 Testing the I-Device function.....	43
<b>4 Links &amp; Literature .....</b>	<b>44</b>
<b>5 History.....</b>	<b>44</b>

# 1 Task and Solution

## 1.1 Task Description

### Description

Mostly in mechanical engineering, there is often the need to implement a simple deterministic communication between modules or machines to avoid having to use “additional” connection-related means of communication.

### Requirement

The following requirements have to be considered:

- efficient engineering, even for fail-safe applications
- consistent decrease of engineering costs and commissioning time
- Reduction of the required network and system components
- Simplification of the communication of machines and systems
- simple communication between heterogeneous systems

## 1.2 Possible solution

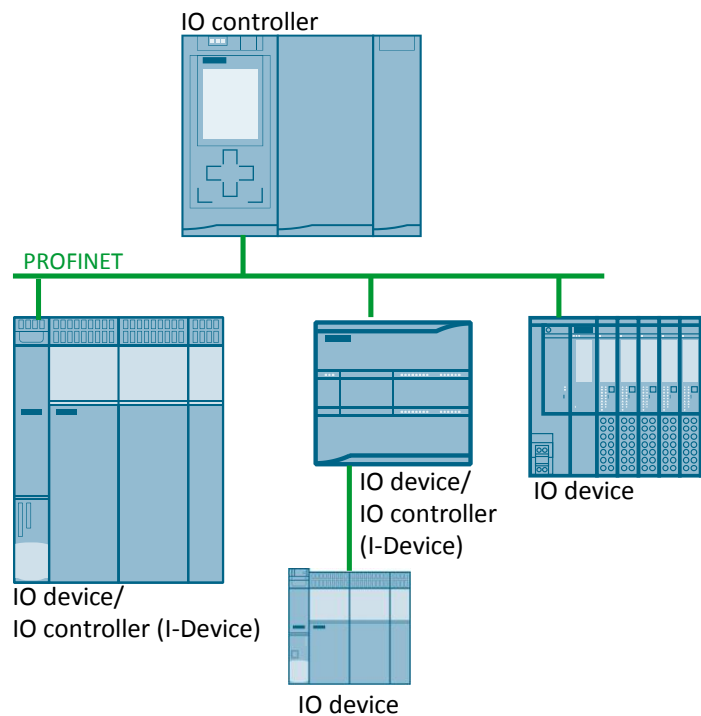
### Description

The I-Device function can be used to exchange data between two controllers very easily.

An “I-Device” is an “intelligent CPU/CP used as an IO device”. This function enables PROFINET to communicate not only with subordinate devices like IO controllers, but also IO communication with other higher-level or central controllers as an IO device.

## Schematic diagram

Figure 1-1



I-Device allows for a very simple and fast communication between two PROFINET IO devices (or PN IO controllers) in the same subnet that can take place simultaneously and on one bus. With I-Device enables access to the IO address via the PROFINET IO protocol without using PN/PN couplers.

## Usable component

The I-Device function is supported by the following modules:

- S7-300 (from V3.2)
- CP 343-1 (Adv.)
- S7-400 (from V6)
- S7-1500
- S7-1200 (from V4)
- ET 200S/ET200SP/ET200pro CPU
- SIMOTION

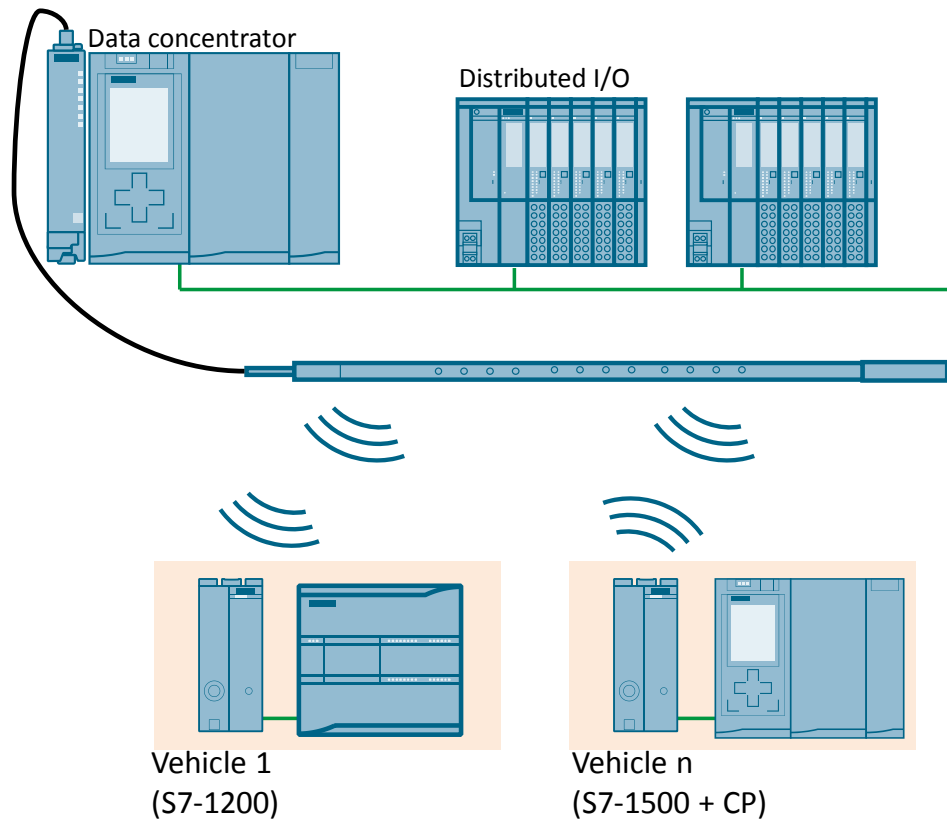
## 1.3 Application area and usage

### Application area

I-Device is ideally suitable for all automation solutions in which several networked controllers are used.

The figure below shows the use of the I-Device function in a carrier control.

Figure 1-2



This system has each carrier equipped with a CPU as I-Device. There is an additional, separate CPU as central controller for all carriers.

Communication to the central PLC is made via I-Device by instancing the carriers in the hardware configuration. This allows for the network to address each carrier in the using a unique device name and the CPU to use a unique IO address range.

### Advantages

- Leaner and more flexible topology
- Simple interconnection of controllers from several projects
- Integration of Siemens controllers with controllers of other manufacturers in a network (via GSD files).
- Elimination of PN/PN couplers
- Consistent communication to all network devices without additional router mechanisms

## 2 Basics on the I-Device

### 2.1 Description of the functionality

#### Explanation of terms

A CPU or a CP with the “Intelligent IO device” configuration is called an I-Device.

An I-Device is like a standard IO device and needs to be handled like one. This means that the I-Device is also connected to a higher-level IO controller.

At a PROFINET interface and in addition to its role as IO device, an I-Device can also be an IO controller.

#### I-Device function

The “I-Device” function enables a CPU or CP to exchange data with an IO controller and can be used as intelligent unit for preprocessing partial processes.

Preprocessing is carried out in the user program of the “I-Device” CPU. The values acquired in central or distributed (PROFINET or PROFIBUS) I/O are preprocessed by the user program and are made available to the IO controller via a PROFINET IO interface of the CPU.

#### Data exchange

The guiding principle of the I-Device method is to use the known process image in a CPU.

From the higher-level IO controller’s view, communicating to an I-Device is similar to communicating to a distributed IO with the usual read and write processes at inputs and outputs.

From the I-Device’s view, data transfer to a higher-level IO controller is also analog to the data transfer to the local or assigned distributed IO via inputs and outputs.

With the use of the process image in the programming, an engineering tool, for example, can directly access all devices in the network – without having to implement additional routing mechanisms.

#### Configuration options

Basically, there are two possibilities for configuration:

1. Configuring an I-Device within a project.
2. Configuring an I-Device that is used in another project or in another engineering system.

When configuring an I-Device for another project or for another engineering system, STEP 7 enables you to do this by exporting a configured I-Device in a GSD file. The GSD file can be imported in the other project or the other engineering system like other GSD files. This allows not only for communication within the project, but also communication across projects and data exchange between different manufacturers in a tried and tested way.

#### Variants of I-Devices

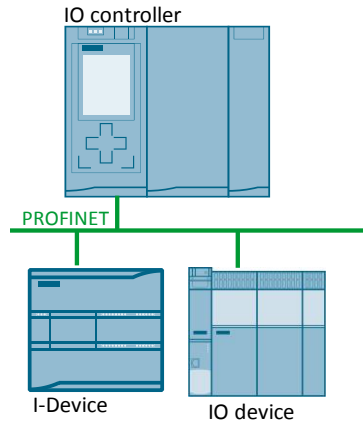
There are the following variants of I-Devices:

- The CPU has only central modules.
- The CPU has central modules and its own IO system with distributed IO.

### I-Device without its own IO system

The I-Device does not have its own distributed IO. Configuration and parameterization of the I-Device as an IO device is carried out like it would be the case in a distributed IO system.

Figure 2-1



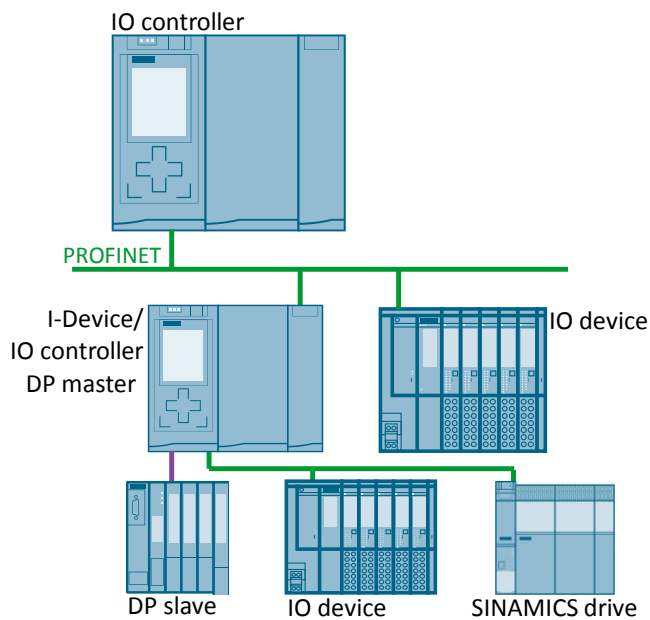
### I-Device with its own IO system

A CPU can be assigned to a (higher-level) IO controller as an I-Device and at the same time create its own IO system with distributed IO.

The subordinate IO system can again contain I-Devices. This allows for hierarchically structured IO systems.

An I-Device can, besides its role as an IO controller, be DP Master for a subordinate PROFIBUS system via a PROFIBUS interface.

Figure 2-2





## 2.2 Principle of data exchange

### General

The guiding principle of the I-Device method is to use the process image in a CPU.

Data exchange between a higher-level IO controller and an I-Device is therefore based on the standard IO controller / IO device relationship.

Communication data are provided in so-called transfer areas.

### Transfer areas

Transfer areas are the IO areas which are used to exchange data between the I-Device and the higher-level IO controller.

Each transfer area contains an information unit that is consistently exchanged between the IO controller and I-Device.

These areas are defined when configuring the I-Device function and are then, amongst others, included in the GSD file that is created afterwards.

The transfer areas are restricted as follows:

Table 2-1

Description	Data
Max. size of the transfer memory for inputs and outputs per AR (gross values, net minus user data supplement)	1440 bytes (inputs) 1440 bytes (outputs)
Max. user data consistency length (max. size of a transfer area)	1024 bytes
Number of transfer areas (can be distributed to the max. number of supported ARs)	64

### Note

S7-1200, S7-300, S7-400 and ET 200S CPU as I-Device support up to two Application Relations (AR), S7-1500 as I-Device a maximum of four ARs.

### NOTICE

The address range of the configured transfer areas directly influences the usable bandwidth of the I-Device:

Bandwidth of the transfer areas + bandwidth of the subordinate IO system = total bandwidth used at I-Device

A transfer area address range that is too large results in higher demand of bandwidth and can therefore lead to higher update times.

## 2.3 Diagnostics and alarm behavior

S7-CPU's have multiple diagnostics and alarm functions that can, for example, signal errors or failures of subordinate IO systems. These diagnostic messages reduce down-times and make localization and rectification of errors easier.

### Diagnostic capabilities

The higher-level IO controller and the I-Device CPU have the following diagnostic mechanisms available:

- OB 83 (insert/remove)
- OB 85 (program sequence error; only S7-300/400)
- OB 86 (rack failure)
- OB 122 (I/O access errors)

In addition, TIA Portal offers commands (only S7-1200/1500) for local troubleshooting and diagnostics:

- GET\_ERROR (retrieve error locally)
- GET\_DIAG (read diagnostics information of a module)

### IO diagnostic messages

The IO diagnostic messages can be processed in the user program of the I-Device CPU and are transferred from there via transfer areas to the higher-level IO controller.

The IO controller must individually evaluate these diagnostics data in the user program.

### Specifics at startup of the higher-level IO controller

- When an IO device comes back after a shutdown, normally, OB 86 in the IO controller is called.
- The returned-to-station message of a higher-level IO controller in the I-Device is divided into two parts:
  1. Calling OB 86: The initial values for the I-Device outputs are set. The input values, however, are not valid yet; they become valid once OB 86 in the higher-level IO controller is called up.
  2. Calling OB 83 for each input transfer area; this call shows the validity of an input transfer area. The startup of the I-Device is only completed once OB 83 for the input transfer areas has been called. Depending on the situation, this step can be carried out with considerable delay or not at all.

### Online view of the diagnostics

The list of available participants shows the I-Device like a normal CPU.

## 2.4 Information on topology and use

### Topology with I-Device

If I-Devices (with or without its own IO system) is used in a PROFINET network, it is advisable to keep the bandwidth required for communication as small as possible.

The (subordinate) IO system of the I-Device should depend on the IO system of the IO controller.

This avoids an overlap of the communication paths between IO controller and its IO devices with the communication paths between I-Device CPU and its subordinate IO system.

The following topology recommendations can therefore be made:

- **I-Device with one port:**  
Connect an I-Device with only one port to a PROFINET switch independent from the higher-level IO system. Connect the subordinate IO system to another port of the switch.
- **I-Device with two ports:**  
If the I-Device has two ports, connect it to a PROFINET switch independent from the higher-level IO system. Use the second port for its own (subordinate) IO system.
- **I-Device with three and more ports:**  
Connect an I-Device with three or more ports via the I-Device with one or both ports with the higher-level IO system in line topology. Use the third port for its own (subordinate) IO system.

### Rules for RT and IRT communication

IO systems with I-Devices are also suitable for creating real-time applications with RT and IRT communication. The following rules must be followed:

- Both higher-level and subordinate IO system must support RT communication. You can use RT communication for both IO systems at the same time.
- IRT communication can be combined with RT communication - but not in both IO systems. IRT is possible where the IO interface is parameterized.

## 2.5 Description of PNIO\_SEND and PNIO\_RECV

The PNIO\_SEND and PNIO\_RECV commands are used for data transfer in the CP operating modes PROFINET IO controller or PROFINET IO device for S7-300 modules. The following section only describes the use as a PROFINET IO Device.

### Use as a PROFINET IO Device

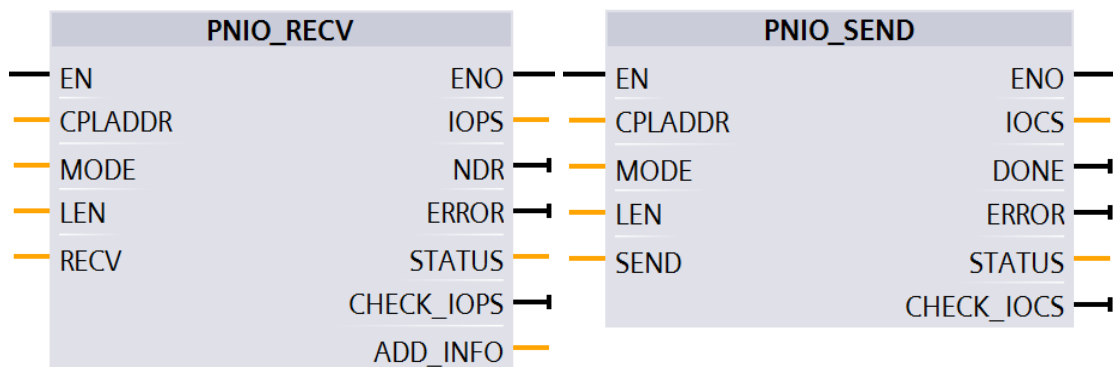
When used as a PROFINET IO device, the commands have the following tasks:

- **PNIO\_SEND:**  
The command reads the pre-processed CPU process inputs in the PROFINET IO device and transfers them to the PROFINET IO controller (configured input address). In addition, the command delivers the IOCS of the PROFINET IO controller as a status message.
- **PNIO\_RECV:**  
The command takes the data transferred by the PROFINET IO controller (configured output address) and the IOPS of the PROFINET IO controller and writes them in the CPU data areas in the PROFINET IO device reserved for the process outputs.

### Call in the user program

The figure below shows the call of the PNIO\_RECV and PNIO\_SEND blocks in the user program.

Figure 2-3



**Note** Further information about the blocks is available in the system manual [\[4\]](#).

## 3 Configuration

### 3.1 S7-CPU as I-Device

#### 3.1.1 Infrastructure information

##### Software package

Install STEP 7 Professional V13 SP1 on a PC/PG.

##### Required devices/components:

To set up the environment, use the following components:

- A CPU 1516-3 PN/DP
- A CPU 1214C
- Two SIMATIC memory cards
- One or two 24V power supplies with cable connector and terminal block plug (the modules can also be operated with a shared power supply)
- DIN rail with mounting accessories for the S7-1x00
- A PG/PC with the configuration tool "STEP 7 Professional V13 SP1" installed
- The necessary network cables, TP cables (twisted pair) according to the IE FC RJ45 standard for Industrial Ethernet

##### Note

Provided the module supports I-Device functionality, you can also use another PROFINET CPU.

The configuration described below explicitly refers only to the components listed in "Required devices/components".

##### Setting up the infrastructure

Connect all the components involved in this solution via the integrated PROFINET interface.

### 3.1.2 Implementing and configuring the devices

#### Preparation

Open the TIA Portal configuration software and create a new project.

#### Implementing the devices

Add two new devices using the project navigation. Select the CPU you are using. In order to be able to keep both devices apart regarding their function in this solution, change the internal project name to "IO controller" (CPU 1516-3 PN/DP) and "I-Device" (CPU 1214C).

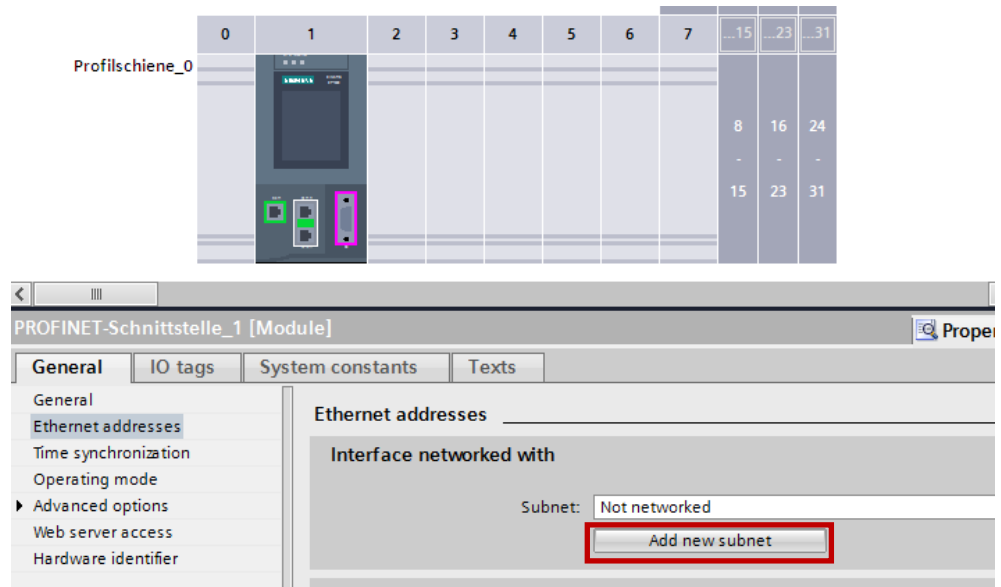
#### Configuring the devices

To set up an automation system, the individual hardware components must be configured, parameterized and connected to each other. Use the device and network view to do this.

Proceed as follows:

1. Select CPU 1516-3 PN/DP (IO controller) in the project tree and open its folder.
2. Open the device view of the component by double clicking "Device configuration".
3. In the graphical view, select the component interface to be networked. The properties of the selected interface are displayed in the inspector window.

4. Select the parameter group "Ethernet addresses" and, under "Interface networked with", click "Add new subnet".

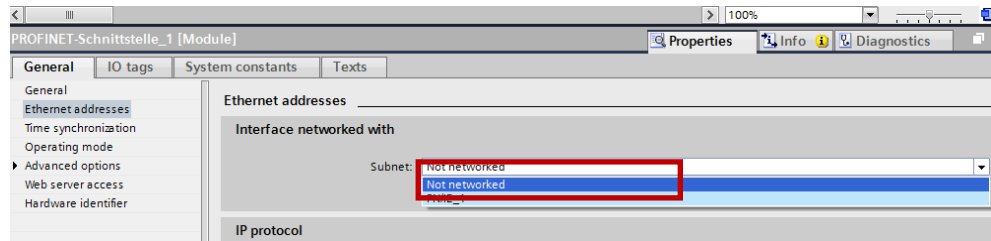


#### Result

The interface is now connected to a new subnet of the suitable subnet type. When this is done, the address parameters of the interface are set consistently automatically.

5. Select CPU 1214C (I-Device) in the project tree and open its folder.
6. Open the device view of the component by double clicking "Device configuration".
7. In the graphical view, select the component interface to be networked. The properties of the selected interface are displayed in the inspector window.

8. Select the parameter group “Ethernet addresses” and, under “Interface networked with”, select the subnet to be connected from the “Subnet” drop-down list.



#### Result

The interface and the selected subnet are now connected. When this is done, the address parameters of the interface are set consistently automatically.



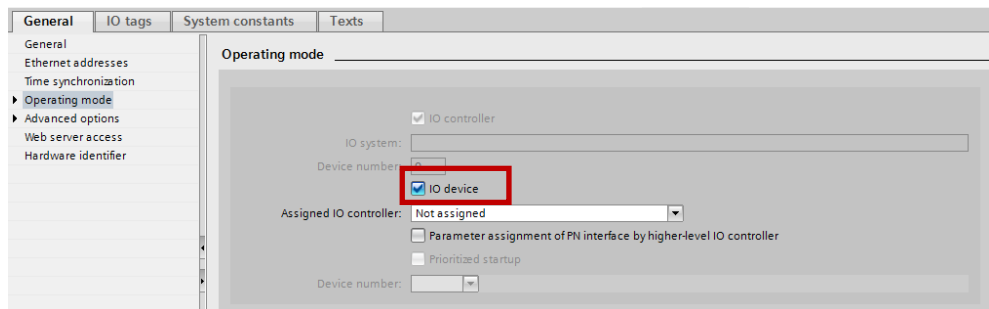
### 3.1.3 Configuring the I-Device function

#### Prerequisite

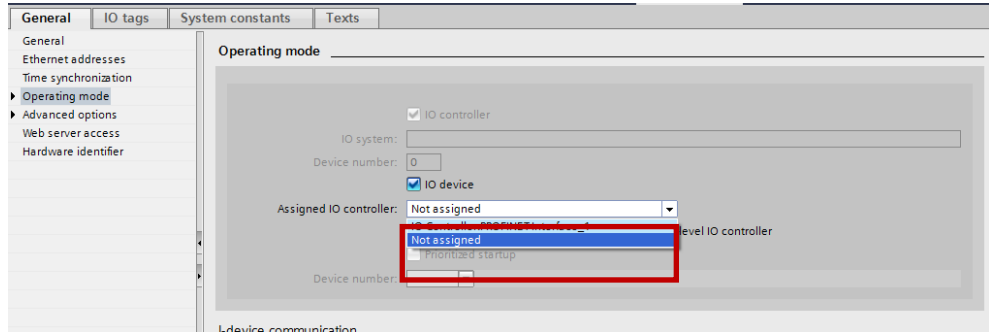
The device view of the CPU 1214C is now active and the properties of the selected interface are displayed in the inspector window.

#### Changing the operating mode

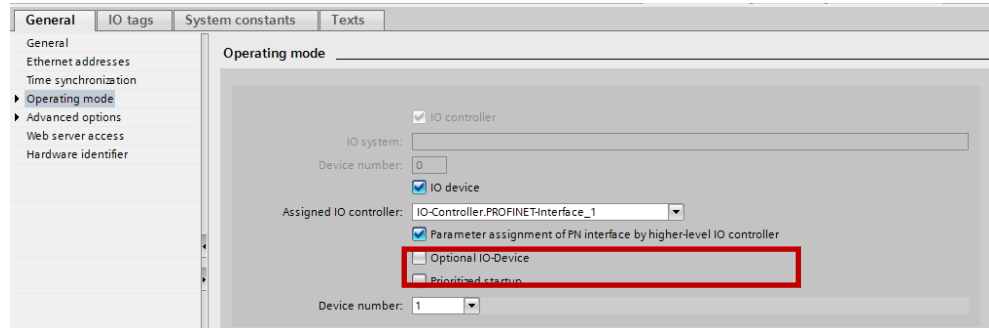
1. In the inspector window, select the parameter group “Operating mode” and tick the “IO device” checkbox.



2. You can select the IO controller from the “Assigned IO controller” drop-down list. Then, the networking and the IO system between both devices is displayed in network view.



3. Use the “Parameter assignment of PN interface by higher-level IO controller” checkbox to determine whether the interface and its ports will be parameterized by the I-Device itself or by the higher-level IO controller. Activate the checkbox for this solution.



**Note** If you use the I-Device with a subordinate IO system, the PROFINET interface (e.g. port parameters) of the I-Device cannot be configured by the higher-level IO controller.

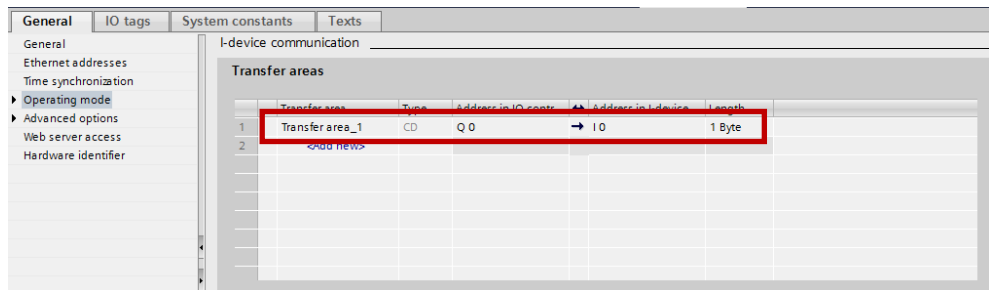
#### Result

The CPU 1214C has now been configured as I-Device and takes on the role of an IO device in the PROFINET network.

### Creating a transfer area

Transfer areas are the IO areas which are used to exchange data between the I-Device and the higher-level IO controller.

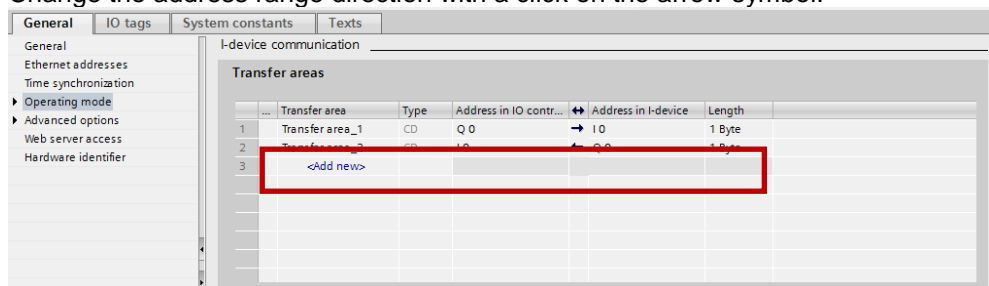
1. Go to the “I-Device communication” section.  
Click the first field of the “Transfer areas” column. STEP 7 creates a predefined name which you are able to change.
2. Select the type of communication relationship: currently, only CD or F-CD for “Data exchange controller device” can be selected.
3. The addresses are pre-assigned automatically. Leave this default unchanged for this solution.



**Note**

If required, you can adjust the addresses to your environment and specify the length of the transfer area to be transferred consistently.

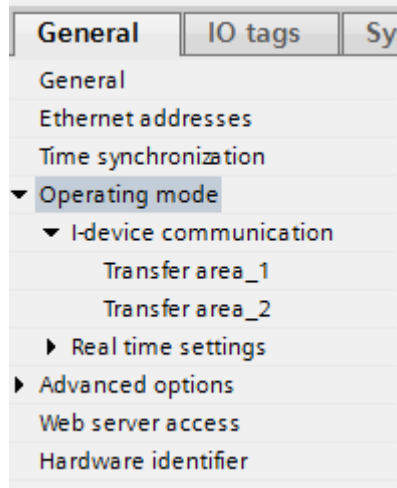
4. Create an additional transfer area by clicking in the second field.  
Leave this default setting of name, type, address and length unchanged as well.  
Change the address range direction with a click on the arrow symbol.



5. Save the project.

### Result

One individual entry for each transfer area is created below the “Operating mode” parameter group. Select one of these entries to adjust or correct and comment on the details of the transfer area.



### 3.1.4 Address assignment and loading

Connect the PG to a free port at a controller to assign the device name and load the project data.

The PG interface must be set to TCP/IP and must be in the same IP band as the controllers.

#### Assigning a device name

For PROFINET communication, the I-Device must be given the configured device name.

Proceed as follows:

1. In STEP 7, select the respective IO device using the MAC address in the “Accessible devices” dialog.
2. Click on “Assign Name” to load the configured device name into the IO device.
3. The IO controller recognizes the IO device using its device name and automatically assigns the configured IP address to the IO device.

#### Loading the project

To load the project data, select both controllers one after the other in the project navigation and load the project into the respective module.

Further information on loading can also be found in the TIA Portal Online Help or in the system manual, see [V4](#).

### 3.1.5 Testing the I-Device function

After the previous chapters, the I-Device function is now complete and both controllers have established a PROFINET communication relationship.

You can very easily test the functionality using transfer areas.

1. Create a tag table in both CPUs.
2. Add the following IO ranges to the tag tables:  
Output byte 0 (“%QB0”)  
Input byte 0 (“%IB0”)

	Name	Address	Display format	Monitor value	Modify value
1		%QB0	Hex	16#00	16#00
2		%IB0	Hex	16#00	

3. Set the output byte 0 of the I-Device CPU to the desired value. The input byte 0 on the IO controller side takes on this value.

	Name	Address	Display format	Monitor value	Modify value
1		%QB0	Hex	16#05	16#05
2		%IB0	Hex	16#00	

## 3.2 SIMATIC-NET CP as an I-Device

### 3.2.1 Infrastructure information

#### Software package

Install STEP 7 Professional V13 SP1 on a PC/PG.

#### Required devices/components:

To set up the environment, use the following components:

- A CPU 1511-1 PN
- A CPU 315-2 DP
- A CP 343-1 lean
- A SIMATIC MEMORY CARD and a Micro Memory Card (MMC)
- One or two 24V power supplies with cable connector and terminal block plug (the modules can also be operated with a shared power supply)
- DIN rail with mounting accessories for the S7-1500 and S7-300
- A PG/PC with the configuration tool "STEP 7 Professional V13 SP1" installed
- The necessary network cables, TP cables (twisted pair) according to the IE FC RJ45 standard for Industrial Ethernet

#### Note

You can also use another PROFINET CPU as IO controller.

Provided the module supports I-Device functionality, you can also use another PROFINET CP.

The configuration described below explicitly refers only to the components listed in "Required devices/components".

#### Setting up the infrastructure

Connect the CPU 315-2 DP with the CP 343 Lean via a backplane bus.

Connect the CPU 1511-1 PN with a PN interface of the CP 343 Lean.

### 3.2.2 Implementing and configuring the devices

#### Preparation

Open the TIA Portal configuration software and create a new project.

#### Implementing the devices

Add two new devices using the project navigation. Select the CPU you are using. In order to be able to keep both devices apart regarding their function in this solution, change the internal project name to "IO controller"(CPU 1511-1 PN) and "I-Device" (CPU 315-2 DP). Add the CP 343-1 Lean to the CPU 315-2 DP (I-Device).

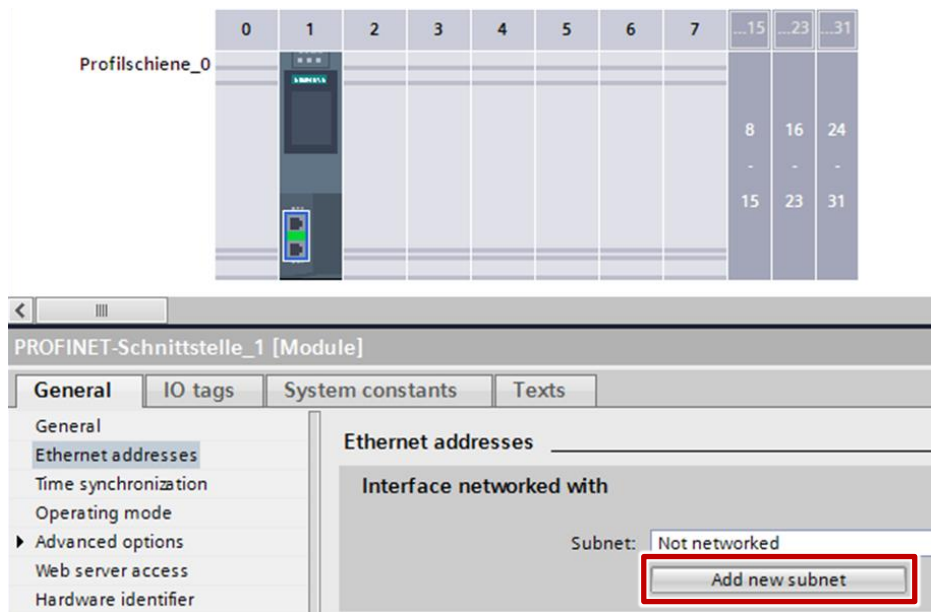
#### Configuring the devices

To set up an automation system, the individual hardware components must be configured, parameterized and connected to each other. Use the device and network view to do this.

Proceed as follows:

1. Select CPU 1511-1 PN/DP (IO controller) in the project tree and open its folder.
2. Open the device view of the component by double clicking "Device configuration".
3. In the graphical view, select the component interface to be networked. The properties of the selected interface are displayed in the inspector window.

4. Select the parameter group "Ethernet addresses" and, under "Interface networked with", click "Add new subnet".



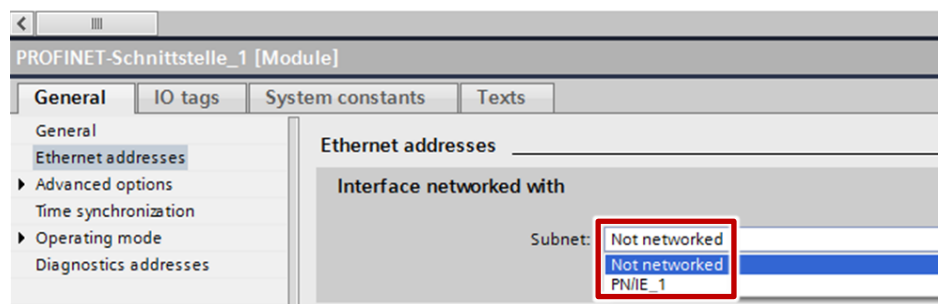
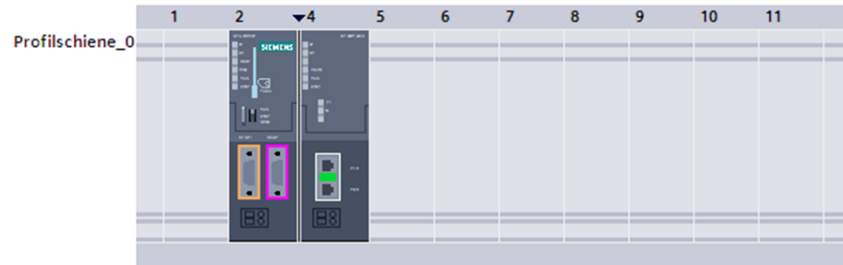
#### Result

The interface is now connected to a new subnet of the suitable subnet type. When this is done, the address parameters of the interface are set consistently automatically.

5. Select CPU 315-2 DP (I-Device) in the project tree and open its folder.
6. Open the device view of the component by double clicking "Device configuration".
7. In the graphical view, select the component interface of the CP 343-1 Lean to be networked.  
The properties of the selected interface are displayed in the inspector window.



8. Select the parameter group “Ethernet addresses” and, under “Interface networked with”, select the subnet to be connected from the “Subnet” drop-down list.



#### Result

The interface and the selected subnet are now connected. When this is done, the address parameters of the interface are set consistently automatically.

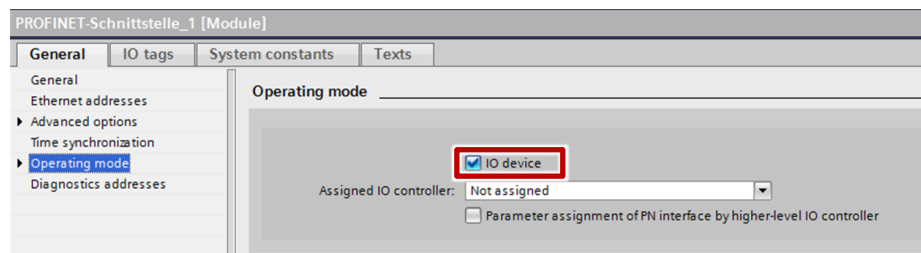
### 3.2.3 Configuring the I-Device function

#### Prerequisite

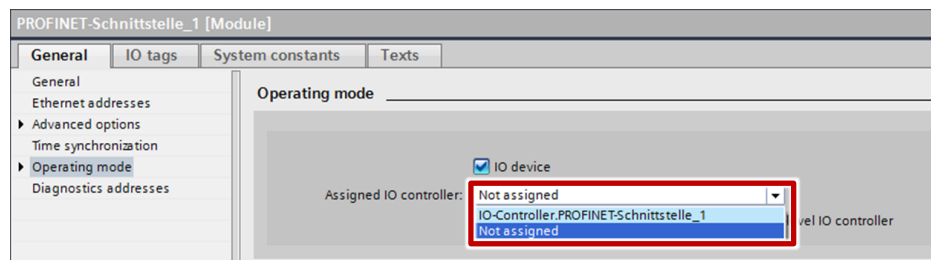
The device view of CPU 315-2 DP is now active and the properties of the selected interface are displayed in the inspector window. Select CP 343-1 Lean for the following steps.

#### Changing the operating mode

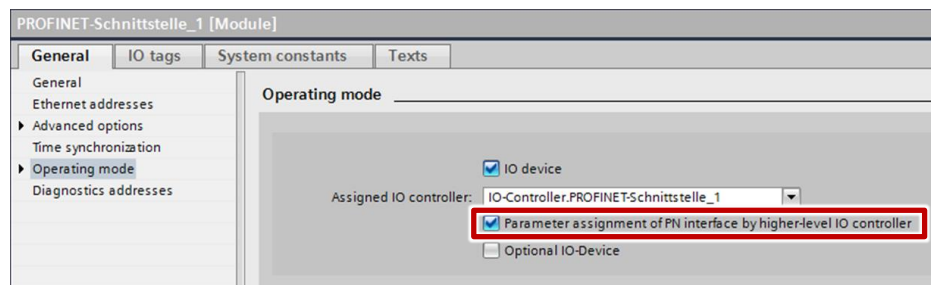
1. In the inspector window, select the parameter group “Operating mode” and tick the “IO device” checkbox.



2. You can select the IO controller from the “Assigned IO controller” drop-down list. Then, the networking and the IO system between both devices is displayed in network view.



3. Use the “Parameter assignment of PN interface by higher-level IO controller” checkbox to determine whether the interface and its ports will be parameterized by the I-Device itself or by the higher-level IO controller. Activate the checkbox for this solution.



**Note** If you use the I-Device with a subordinate IO system, the PROFINET interface (e.g. port parameters) of the I-Device cannot be configured by the higher-level IO controller.

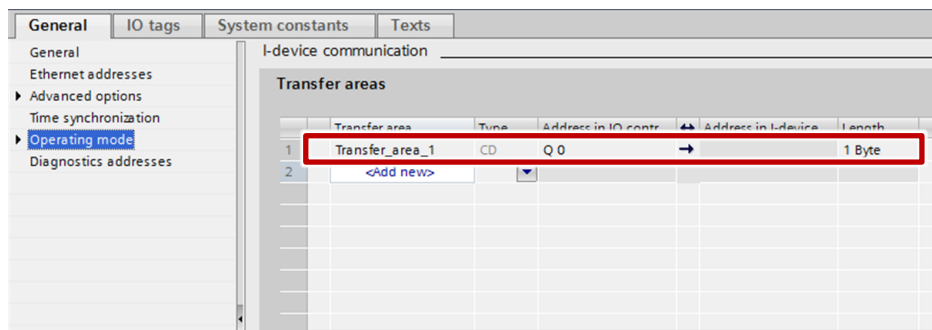
#### Result

The CP 343-1 Lean has now been configured as I-Device and takes on the role of an IO device in the PROFINET network.

#### Creating a transfer area

Transfer areas are the IO areas which are used to exchange data between the I-Device and the higher-level IO controller.

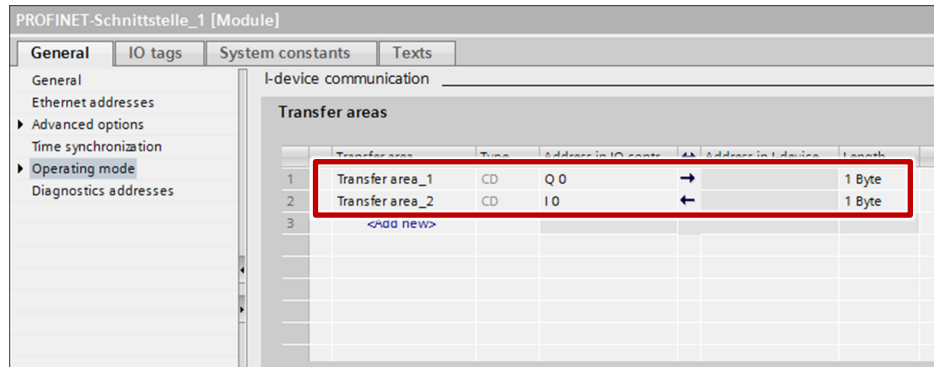
1. Go to the “I-Device communication” section.  
Click the first field of the “Transfer areas” column. STEP 7 creates a predefined name which you are able to change.
2. Select the type of communication relationship: currently, only CD or F-CD for “Data exchange controller device” can be selected.
3. The addresses are pre-assigned automatically. Leave this default unchanged for this solution.



#### Note

If required, you can adjust the addresses to your environment and specify the length of the transfer area to be transferred consistently.

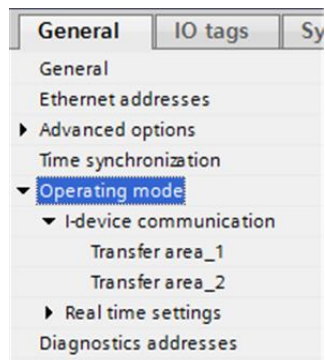
4. Create an additional transfer area by clicking in the second field. Leave this default setting of name, type, address and length unchanged as well. Change the address range direction with a click on the arrow symbol.



5. Save the project.

#### Result

One individual entry for each transfer area is created below the “Operating mode” parameter group. Select one of these entries to adjust or correct and comment on the details of the transfer area.



#### Calling PNIO\_RECV and PNIO\_SEND

Communication within the I-Device is made by calling the program blocks PNIO\_RECV and PNIO\_SEND (“Instructions > Communication > PROFINET IO”) in TIA Portal) in the CPU user program.

1. Call the PNIO\_RECV and PNIO\_SEND blocks in the user program of CPU 315-2 DP.
2. Enter the required data in the block interfaces (see TIA Portal help).

### 3.2.4 Address assignment and loading

Connect the PG to a free port at the S7-1511-1 PN or CP 343-1 Lean to assign the device name and load the project data.  
The PG interface must be set to TCP/IP and must be in the same IP band as the controller or CP.

#### Assigning a device name

For PROFINET communication, the I-Device must be given the configured device name.

Proceed as follows:

1. In STEP 7, select the respective IO device using the MAC address in the "Accessible devices" dialog.
2. Click on "Assign Name" to load the configured device name into the IO device.
3. The IO controller recognizes the IO device using its device name and automatically assigns the configured IP address to the IO device.

#### Loading the project

To load the project data, select both controllers one after the other in the project navigation and load the project into the respective module.  
Further information on loading can also be found in the TIA Portal Online Help or in the system manual, see [41](#).

### 3.2.5 Testing the I-Device function

After the previous chapters, the I-Device function is now complete. The S7-1511-1 PN controller and the CP 343-1 Lean have established a PROFINET communication relationship.

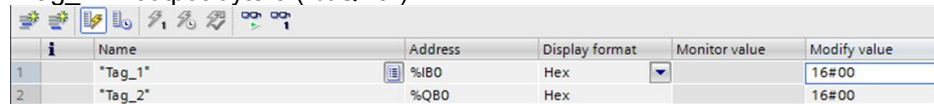
You can very easily test the functionality using transfer areas.

1. Create a tag table in both CPUs.
2. Add the flags or tags of a data block, which you have assigned to the "Send" input of the PNIO\_SEND program block before, into the tag tables.

In this case:

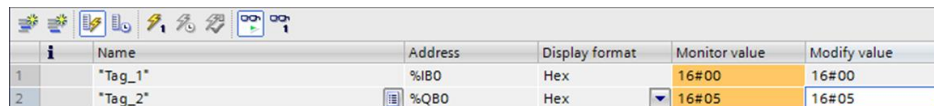
"Tag\_1" = input byte 0 ("%IB0")

"Tag\_2" = output byte 0 ("%QB0")



	i	Name	Address	Display format	Monitor value	Modify value
1		"Tag_1"	%IB0	Hex		16#00
2		"Tag_2"	%QB0	Hex		16#00

3. Set the output byte of the I-Device CPU, which you have assigned to the "Send" input of the PNIO\_SEND program block, to the desired value. The input byte 0 on the IO controller side takes on this value.



	i	Name	Address	Display format	Monitor value	Modify value
1		"Tag_1"	%IB0	Hex	16#00	16#00
2		"Tag_2"	%QB0	Hex	16#05	16#05

## 3.3 SIMOTION as I-Device

### 3.3.1 Infrastructure information

#### Software package

Install STEP 7 Professional V13 SP1 and SIMOTION SCOUT V4.4 on a PC/PG.

#### Required devices/components:

To set up the environment, use the following components:

- A CPU 1516-3F PN/DP
- A SIMOTION D455-2 DP/PN (S120)
- A SIMATIC MEMORY CARD
- A CompactFlash Card
- One or two 24V power supplies with cable connector and terminal block plug (the modules can also be operated with a shared power supply)
- DIN rail with mounting accessories for the S7-1500
- A PG/PC with the configuration tools "STEP 7 Professional V13 SP1" and "SIMOTION SCOUT TIA V4.4" installed
- The necessary network cables, TP cables (twisted pair) according to the IE FC RJ45 standard for Industrial Ethernet

#### Note

You can also use another PROFINET CPU as IO controller.

Provided the module supports I-Device functionality, you can also use another SIMOTION controller.

The configuration described below explicitly refers only to the components listed in "Required devices/components".

#### Setting up the infrastructure

Connect a PN interface of CPU 1516F-3 PN/DP with the X150 interface of the SIMOTION D455-2 DP/PN.

### 3.3.2 Implementing and configuring the devices

#### Preparation

Open the TIA Portal configuration software and create a new project.

#### Implementing the devices

Add two new devices using the project navigation. Select the CPU you are using.

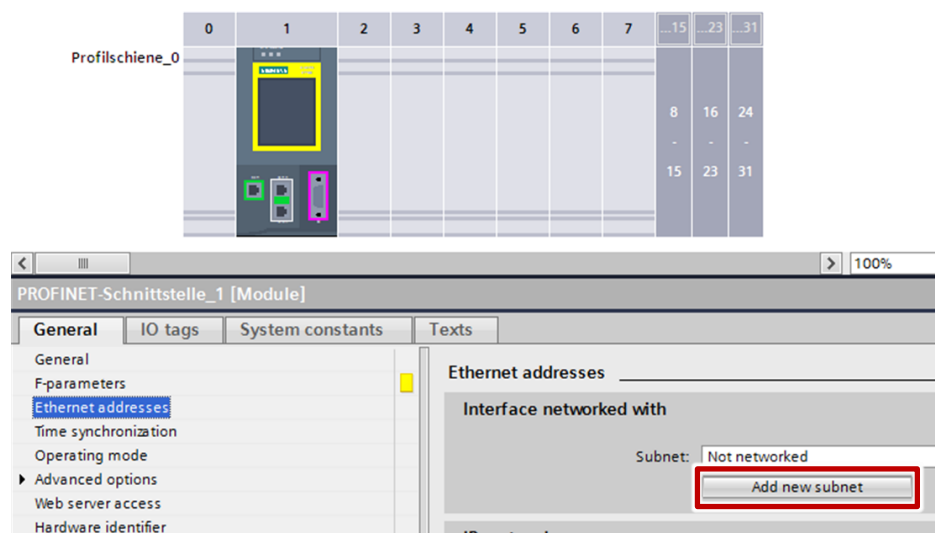
In order to be able to keep both devices apart regarding their function in this solution, change the internal project name to "IO controller" (CPU 1516-3F PN) and "I-Device" (SIMOTION D455-2 DP/PN).

#### Configuring the devices

To set up an automation system, the individual hardware components must be configured, parameterized and connected to each other. Use the device and network view to do this.

Proceed as follows:

1. Select CPU 1516F-3 PN/DP (IO controller) in the project tree and open its folder.
2. Open the device view of the component by double clicking "Device configuration".
3. In the graphical view, select the component interface to be networked. The properties of the selected interface are displayed in the inspector window.
4. Select the parameter group "Ethernet addresses" and, under "Interface networked with", click "Add new subnet".

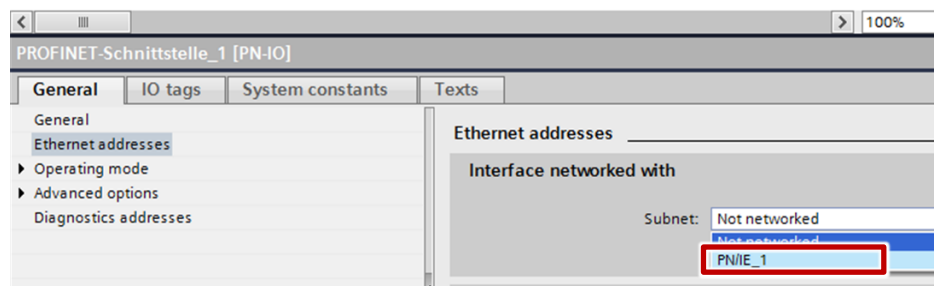


#### Result

The interface is now connected to a new subnet of the suitable subnet type. When this is done, the address parameters of the interface are set consistently automatically.



5. Select SIMOTION D455-2 DP/PN in the project tree and open its folder.
6. Open the device view of the component by double clicking “Device configuration”.
7. In the graphical view, select the SIMOTION D455-2 DP/PN interface to be networked.  
The properties of the selected interface are displayed in the inspector window.
8. Select the parameter group “Ethernet addresses” and, under “Interface networked with”, select the subnet to be connected from the “Subnet” drop-down list.



#### Result

The interface and the selected subnet are now connected. When this is done, the address parameters of the interface are set consistently automatically.

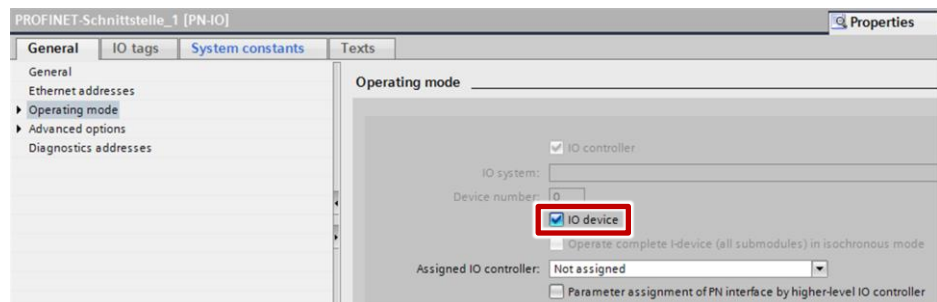
### 3.3.3 Configuring the I-Device function

#### Prerequisite

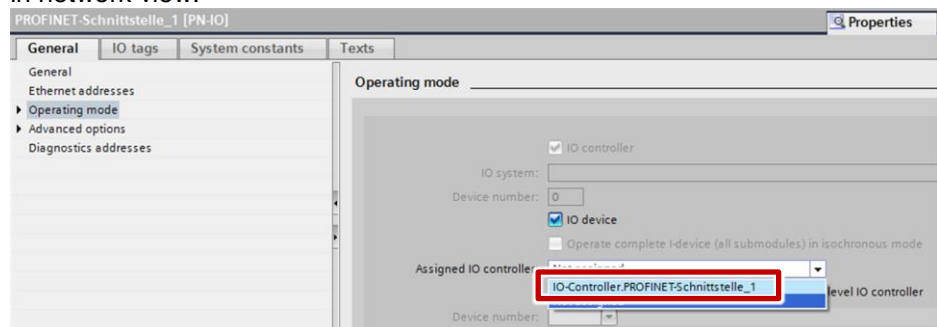
The device view of the SIMOTION D455-2 DP/PN is now active and the properties of the selected interface are displayed in the inspector window.

#### Changing the operating mode

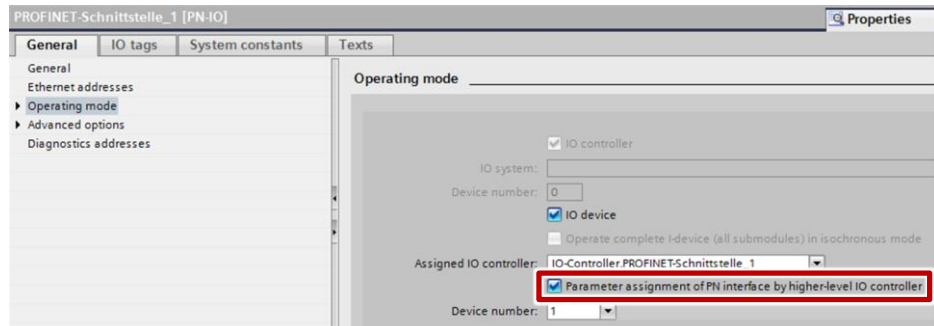
1. In the inspector window, select the parameter group “Operating mode” and tick the “IO device” checkbox.



2. You can select the IO controller from the “Assigned IO controller” drop-down list. Then, the networking and the IO system between both devices is displayed in network view.



- Use the “Parameter assignment of PN interface by higher-level IO controller” checkbox to determine whether the interface and its ports will be parameterized by the I-Device itself or by the higher-level IO controller. Activate the checkbox for this solution.



**Note** If you use the I-Device with a subordinate IO system, the PROFINET interface (e.g. port parameters) of the I-Device cannot be configured by the higher-level IO controller.

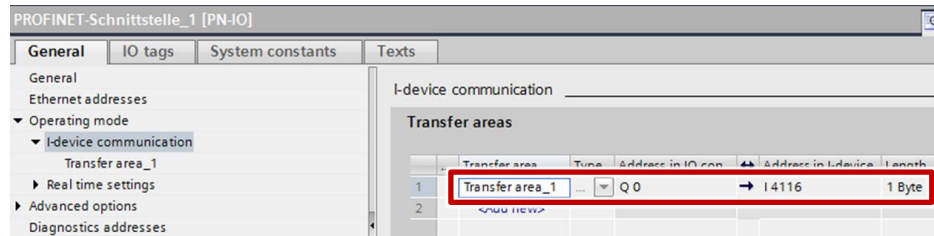
#### Result

The SIMOTION D455-2 DP/PN has now been configured as I-Device and takes on the role of an IO device in the PROFINET network.

### Creating a transfer area

Transfer areas are the IO areas which are used to exchange data between the I-Device and the higher-level IO controller.

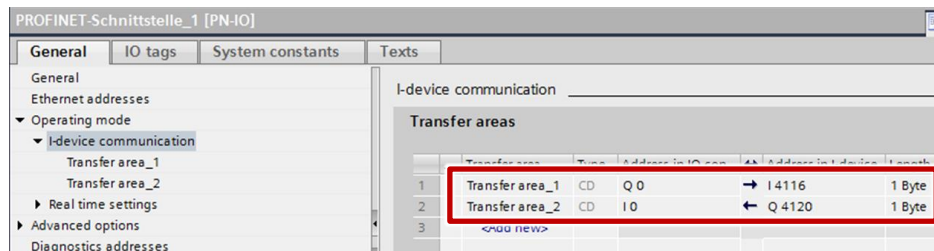
4. Go to the “I-Device communication” section.  
Click the first field of the “Transfer areas” column. STEP 7 creates a predefined name which you are able to change.
5. Select the type of communication relationship: currently, only CD or F-CD for “Data exchange controller device” can be selected.
6. The addresses are pre-assigned automatically. Leave this default unchanged for this solution.



#### Note

If required, you can adjust the addresses to your environment and specify the length of the transfer area to be transferred consistently.

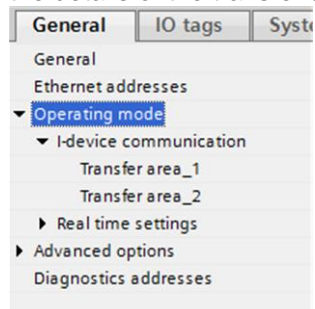
7. Create an additional transfer area by clicking in the second field.  
Leave this default setting of name, type, address and length unchanged as well.  
Change the address range direction with a click on the arrow symbol.



8. Save the project.

#### Result

One individual entry for each transfer area is created below the “Operating mode” parameter group. Select one of these entries to adjust or correct and comment on the details of the transfer area.



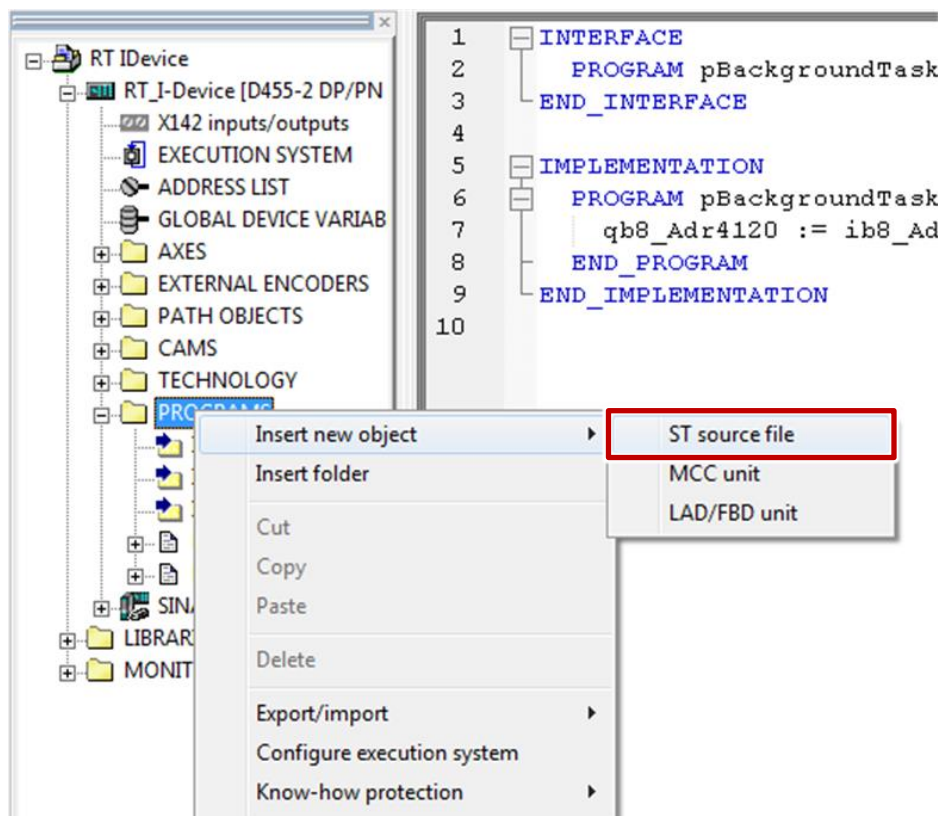
### 3.3.4 Configuration in SIMOTION SCOUT TIA

In order to enable the I-Device function of the SIMOTION D455-2 DP/PN, the following steps must be configured. To do this, open SIMOTION SCOUT TIA V4.4.

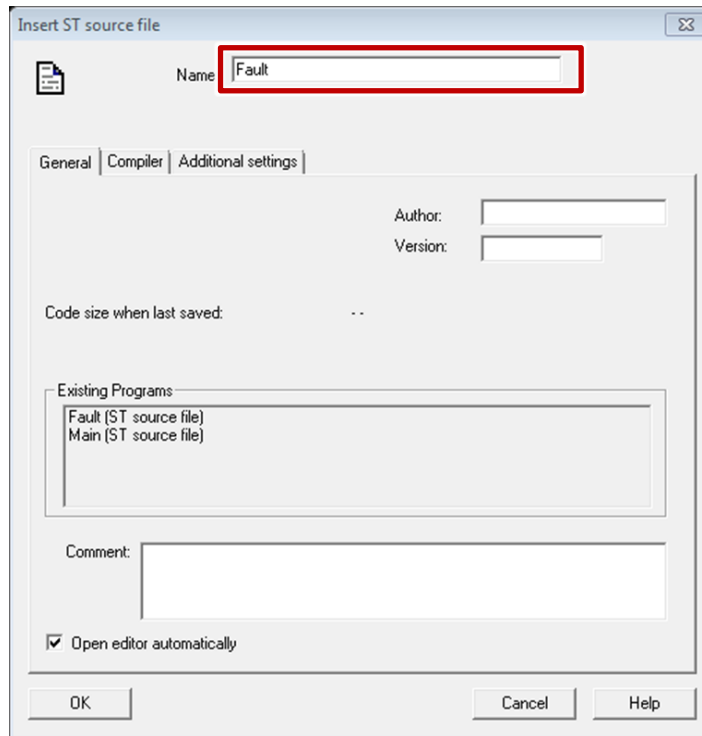
#### Defining SystemInterruptTask

The SIMOTION D455-2 DP/PN will go to fault status if a “SystemInterruptTask” is triggered. This is why you must define routines for these “SystemInterruptTasks”.

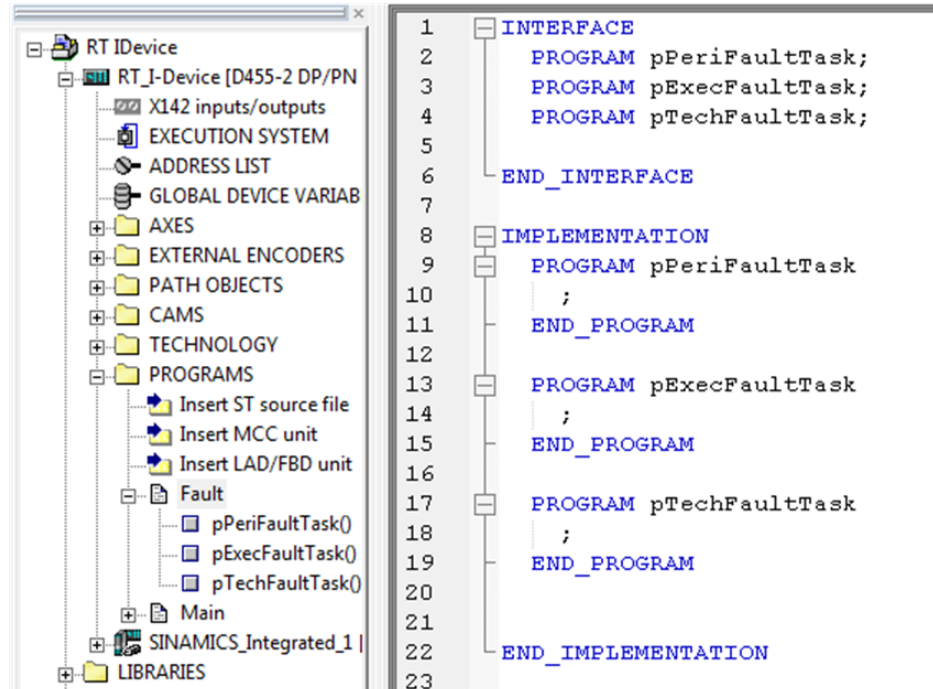
1. To do this, right-click in the tree view on “PROGRAMS” under “Insert new object” and select an “ST source file”.



- In the subsequent window, select the name for the ST source. (Here: "Fault")



- Open the ST source "Fault" and add the following program.



- Repeat step 1 and 2 with the name "Main".

- In tree view, open the “ADDRESS LIST” and add the desired input and output areas (here: input range: “ib8\_Adr14116” and output area: “qb8\_Adr4120”).

Name	I/O address	Read o	Data type	Array l	Process image	Strategy	Display	Substitute
ib8_Adr4116	PB 4116		BYTE	1		Substitut...	HEX	16#00
qb8_Adr4120	PQB 4120		BYTE	1		Substitut...	HEX	16#00

- Open the ST source “Main” and add the following program.

```

1  INTERFACE
2      PROGRAM pBackgroundTask;
3  END_INTERFACE
4
5  IMPLEMENTATION
6      PROGRAM pBackgroundTask
7          qb8_Adr4120 := ib8_Adr4116;
8      END_PROGRAM
9  END_IMPLEMENTATION
10
    
```

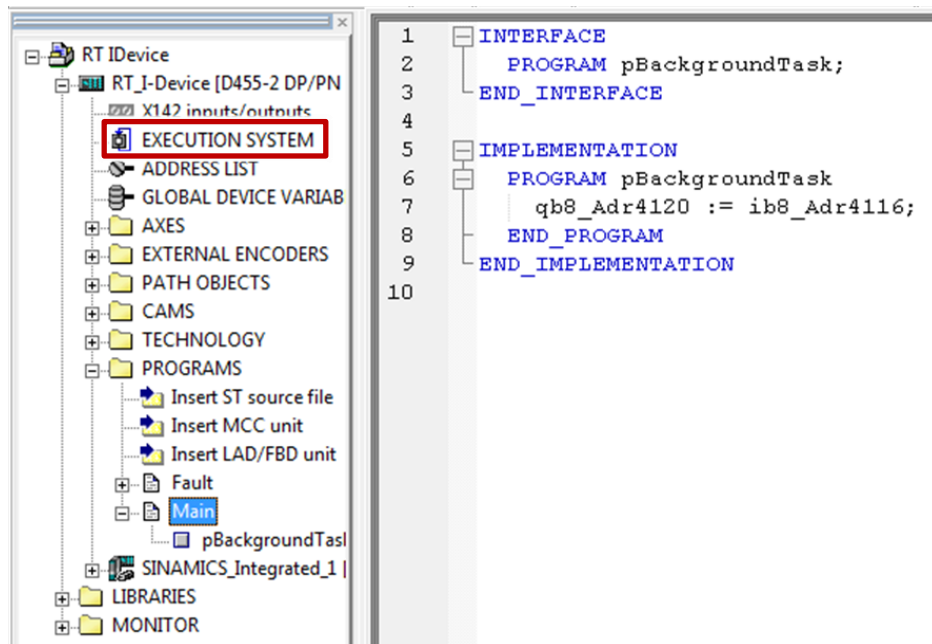
**Note** In this user program, the output area is identical to the input area of the SIMOTION D455-2 DP/PN.



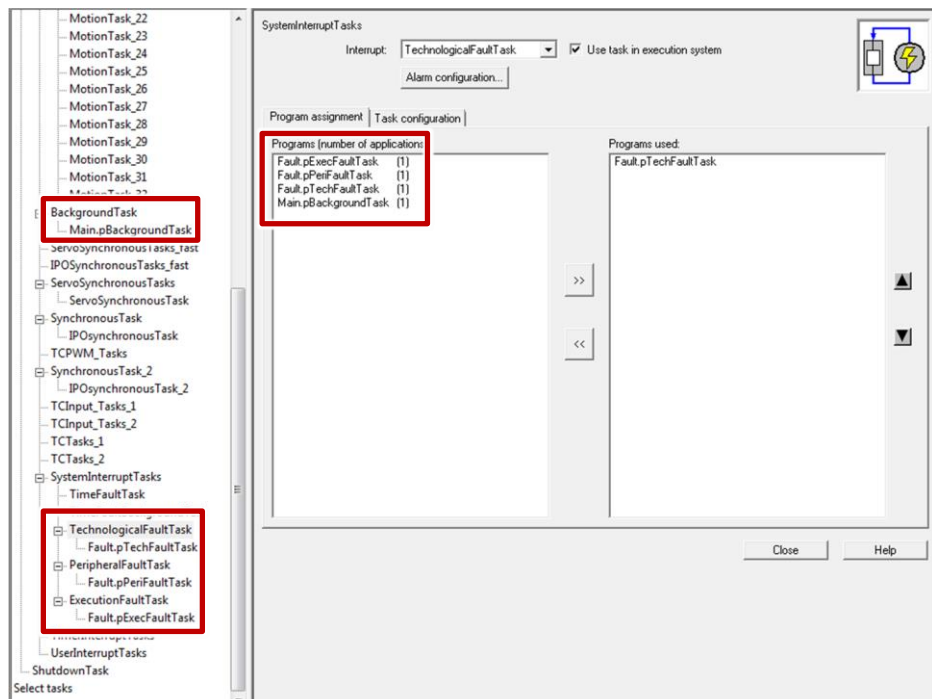
### Assigning the programs to the execution system

The programs created now need to be assigned to their task.

1. To do this, open the “EXECUTION SYSTEM” from the tree view.



2. Drag and drop the programs created here to their respective tasks as shown in the figure below.



3. All required settings have been made. Save and compile the project in SIMOTION SCOUT TIA.

### 3.3.5 Address assignment and loading

Connect the PG to a free port at CPU 1516-3 PN/DP or SIMOTION D455-2 DP/PN to assign the device name and load the project data.

The PG interface must be set to TCP/IP and must be in the same IP band as the controller or CP.

#### Assigning a device name

For PROFINET communication, the I-Device must be given the configured device name.

Proceed as follows:

1. In STEP 7, select the respective IO device using the MAC address in the “Accessible devices” dialog.
2. Click on “Assign Name” to load the configured device name into the IO device.
3. The IO controller recognizes the IO device using its device name and automatically assigns the configured IP address to the IO device.

#### Loading a project - TIA Portal

To load the project data, select both controllers one after the other in the project navigation and load the project into the respective module.

Further information on loading can also be found in the TIA Portal Online Help or in the system manual, see [V4](#).

#### Loading a project - SIMOTION SCOUT TIA

Load the SIMOTION D455-2 DP/PN in SIMOTION SCOUT TIA.

Further information on loading can also be found in the SIMOTION SCOUT TIA Online Help.

### 3.3.6 Testing the I-Device function

After the previous chapters, the I-Device function is now complete. The S7-1511-1 PN controller and the SIMOTION D455-2 DP/PN have established a PROFINET communication relationship.

You can very easily test the functionality using transfer areas.

1. Create a tag table in the IO controller CPU.
2. Add the following flags to the tag tables.

In this case:

“Tag\_1” = output byte 0 (“%QB0”)

“Tag\_2” = input byte 0 (“%IB0”)

	Name	Adresse	Anzeigeformat	Beobachtungswert	Steuerwert
	*Tag_1*	%QB0	Hex	16#00	
	*Tag_2*	%IB0	Hex	16#00	
		<Hinzufügen>			

3. Set the output byte 0 of the IO controller to the desired value. The input byte 0 on the IO controller side takes on this value, as it is mirrored in the I-Device user program (see [here](#)).

	Name	Adresse	Anzeigeformat	Beobachtungswert	Steuerwert
1	*Tag_1*	%QB0	Hex	16#02	16#02
2	*Tag_2*	%IB0	Hex	16#02	
3		<Hinzufügen>			

## 4 Links & Literature

Table 4-1

	Topic	Title
\1\	Siemens Industry Online Support	<a href="http://support.automation.siemens.com">http://support.automation.siemens.com</a>
\2\	Download page of the entry	<a href="https://support.industry.siemens.com/cs/ww/en/view/109478798">https://support.industry.siemens.com/cs/ww/en/view/109478798</a>
\3\	PROFINET with STEP 7 V13	<a href="https://support.industry.siemens.com/cs/en/en/view/49948856">https://support.industry.siemens.com/cs/en/en/view/49948856</a>
\4\	STEP 7 Professional V13 SP1	<a href="https://support.industry.siemens.com/cs/en/en/view/109011420">https://support.industry.siemens.com/cs/en/en/view/109011420</a>

## 5 History

Table 5-1

Version	Date	Modifications
V1.0	08/2015	First version