Configuring the F-I-Device function with the “SENDDP” and “RCVDP” blocks.

PROFIsafe

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

Warranty and Liability ........................................................................................................... 2
1 Task and solution .................................................................................................................. 4
2 Configuring the F-I-Device function for fail-safe controllers ......................................... 6
  2.1 Description of the differences to the standard F-I-Device ........................................... 6
  2.2 Description of SENDDP and RCVDP ........................................................................ 9
  2.3 Using a “dummy CPU” for cross-project configuration and with a S7-1500 as F-I-Device ................................................................. 12
  2.4 Advantages of using F-I-Devices .................................................................................. 13
    2.4.1 Configuration with a “dummy CPU” .................................................................. 13
    2.4.2 Configuration with a GSD file .......................................................................... 13
3 Configuration ........................................................................................................................ 14
  3.1 Infrastructure information ......................................................................................... 14
  3.2 F-CPU as an F-I-Device in one project .................................................................... 15
    3.2.1 Implementing and configuring the devices ......................................................... 15
    3.2.2 Configuring the F-I-Device function ................................................................. 18
  3.3 F-CPU as an F-I-Device across projects ................................................................... 24
    3.3.1 Implementing and configuring the F-I-Device (Project B) ................................ 24
    3.3.2 Address assignment and loading (Project B) .................................................. 30
    3.3.3 Implementing and configuring the IO controller and the dummy CPU (Project A) .............................................................................. 32
    3.3.4 Address assignment and loading (Project A) .................................................. 41
  3.4 Diagnosis of configured F-I-Device communication ................................................ 43
    3.4.1 Correctly established F-I-Device communication ........................................... 43
    3.4.2 Communication error during F-I-Device communication ................................ 45
4 Links & Literature ............................................................................................................... 47
5 History ................................................................................................................................. 47
# 1 Task and solution

## Description

Even in fail-safe automation systems, there is often the need to implement a deterministic communication between modules or machines that is easy to configure to avoid having to use “additional” connection-related means of communication.

This document describes the following engineering options:

1. Fail-safe communication between two F-CPUs is configured in a joint TIA project.
2. Fail-safe communication between two F-CPUs is configured in separate TIA projects (across projects).

## Note

This document only describes the use of the F-I-Device function in a fail-safe environment and explains the differences to standard environments. A detailed description, the benefits and the area of application of the I-Device function can be found in the Configuration Description “Config_I-Device_Standard” on the article page [2].

© Siemens AG 2015 All rights reserved

I-Device_Safety
Entry-ID: 109478798,  V2.0,  11/2016  4
Diagrammatic representation

Figure 1-1

F-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, even in a fail-safe environment.

The fail-safe communication between the safety program of the F-CPU of an IO controller and the safety program(s) of the F-CPU(s) of one or more F-I-Devices takes place via connections between IO controller and F-I-Device connections (F-CD) (via PROFINET IO, just like in the standard application).

The communication between IO controller and F-I-Device requires no additional hardware.

Usable components

The F-I-Device function in a fail-safe environment is supported by the following modules:

- S7-300 (from V3.2) / S7-300F
- S7-400 (from V6) / S7-400F
- S7-1500 / S7-1500F
- SIMOTION
- ET 200S CPU / ET 200SP CPU / ET 200pro CPU
- SIMATIC CPs (CP 343-1 Adv. IT, CM 1542-1)

Note: To find out which module supports the “F-I-Device” mode or the connection of “F-I-Device” as an “IO controller”, refer to [7].
2 Configuring the F-I-Device function for fail-safe controllers

2.1 Description of the differences to the standard F-I-Device

General

A fail-safe communication between IO controller and F-I-Device transfers a fixed number of data between the safety programs in F-CPUs of IO controllers / F-I-Devices.

The data are sent with the "SENDDP" command and received with the "RCVDP" command and are stored in configured transfer areas in the devices. A transfer area consists of an input and an output address range.

Data exchange

In contrast to the standard I-Device, the fail-safe configuration of the F-I-Device function does not access the inputs and outputs of the configured transfer areas of the IO controller or the F-I-Device directly.

The transfer areas are accessed using the “SENDDP” and “RCVDP” blocks (see Chapter 2.2).

“SENDDP” serves the outputs of the IO controller / F-I-Device and “RCVDP” reads their inputs.

Figure 2-1

Configuration options

Basically, there are two possibilities for configuration:

1. Configuring an F-I-Device within one project.
2. Configuring an F-I-Device that is used in another project or in another engineering system.
Transfer areas

Transfer areas are required for the communication between IO controller and IO device even in a fail-safe environment.

These contain the data to be transferred.

These transfer areas are assigned to the “SENDDP” and “RCVDP” blocks, as these are used to exchange data in a fail-safe environment.

Rules:

The transfer area of the data to be sent must start with the same start address for output data and input data (this only applies to the SIMATIC S7-300 / S7-400 controllers).

The output data transfer area requires 12 bytes (consistently), the input data transfer area requires 6 bytes (consistently).

The transfer area of the data to be received must start with the same start address for output data and input data must start with the same start address for output data and input data (this only applies to the SIMATIC S7-300 / S7-400 controllers).

The input data transfer area requires 12 bytes (consistently), the output data transfer area requires 6 bytes (consistently).

Example:

“SENDDP” sends 12 bytes to the partner. These 12 bytes consist of 6 bytes of F-IO data (16 Boolean and 2 INT values) and 6 bytes of F-parameters. “RCVDP” answers these data with an acknowledgment of 6 bytes of F-parameters (see Figure 2.2).

Figure 2.2

Note

The transfer area of the input data to be sent and the transfer area for the output data to be received are created automatically by TIA Portal.
Limits for data transfer

If the data volume to be transferred is bigger than the capacity of the interlinked “SENDDP” / “RCVDP” commands, additional “SENDDP” / “RCVDP” commands can be used. To achieve this, create additional transfer areas.

The table below shows how many output and input data are assigned in fail-safe communication connections.

Table 2-1

<table>
<thead>
<tr>
<th>Security fail-safe communication</th>
<th>Communication connection</th>
<th>Assigned input and output data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>In the IO controller</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Output data</td>
</tr>
<tr>
<td>IO controller - F-I-Device</td>
<td>Send: F-I-Device to IO controller</td>
<td>6 bytes</td>
</tr>
<tr>
<td></td>
<td>Receive: F-I-Device from IO controller</td>
<td>12 bytes</td>
</tr>
</tbody>
</table>

Note

Per F-I-Device, you can create up to 64 transfer areas. Each “SENDDP” / “RCVDP” block requires 2 transfer areas. Per F-I-Device, 32 arbitrary combinations of “SENDDP” / “RCVDP” blocks are possible.
### 2.2 Description of SENDDP and RCVDP

#### General

The data between the IO controller and an F-I-Device are sent fail-safe with the “SENDDP” command and received with the “RCVDP” command. These commands allow for a fail-safe transfer of a fixed amount of BOOL or INT (alternatively DINT) data.

These commands can be found in the “instructions” task card under “Communication”. The “RCVDP” command must be called at the beginning of the main safety block and the “SENDDP” command at the end.

Please note that the send signals are only sent once the “SENDDP” command has been called at the end of processing the respective F-runtime group.

#### Description

The “SENDDP” command sends 16 BOOL data and 2 INT data via PROFIBUS DP/PROFINET from one F-CPU to the other in a fail-safe way. In case of the S7-1500, DINT data are used. The data can then be received from the respective “RCVDP” command.

#### Calling the blocks SENDDP and RCVDP

The figure below shows how to call up the blocks “SENDDP” and “RCVDP”.

![Diagram](image)
### DP_DP_ID parameter for several F-I-Devices

The “DP_DP_ID” parameter of the “SEND_DP” and “RCV_DP” blocks is an ID that is unique in the network of the communicating “SEND_DP” and “RCV_DP” blocks. When using several “SEND_DP” / “RCV_DP” blocks, the “DP_DP_ID” input must be adjusted accordingly.

The figure below shows one option how to configure the blocks.

Figure 2-4

#### Note

Per F-I-Device, you can create up to 64 transfer areas.
Each “SEND_DP” / “RCV_DP” block requires 2 transfer areas.
Per F-I-Device, 32 arbitrary combinations of “SEND_DP” / “RCV_DP” blocks are possible.

The parameters “DP_DP_ID” of the “SEND_DP” and “RCV_DP” blocks need to be unequal to the HW-IDs of the project. Use a different value range for the “DP_DP_ID” parameter, e.g. “DP_DP_ID > 10000.” Further information about the blocks is available here [5].
Using the blocks “SENDDP” and “RCVDP”

The following time diagram shows the behavior of the “SENDDP” and “RCVDP” blocks during initial communication, during communication errors, during troubleshooting and during the manual acknowledgment on “RCVDP”.

Figure 2-5

**Behavior in the event of communication errors**

If a communication error occurs during the fail-safe CPU-CPU communication, the “ERROR” and “SUBS_ON” outputs of both blocks are set.

As long as this communication error is not fixed and acknowledged, the “RCVDP” block outputs the substitute values that are parameterized at the “SUBBO_xx” and “SUBI_xx” or “SUBDI_00” inputs.

The transmitted data from “SENDDP” that are at the “SD_BO_xx” and “SD_I_xx” or “SD_DI_00” inputs are output again only when no further communication errors are detected (“ACK_REQ = TRUE) and when the error has been acknowledged manually at the “ACK_REI” input.
2.3 Using a “dummy CPU” for cross-project configuration and with a S7-1500 as F-I-Device

Description

If the two communication partners are in separate projects, fail-safe communication between an IO controller and an F-I-Device must be realized by configuring a “dummy CPU”.

The “dummy CPU” represents the “F-I-Device” in the “IO controller” project. Another “dummy CPU” represents the “IO controller” in the “F-I-Device” project.

It is not possible to use a GSDML file of the F-I-Device in TIA Portal V13 SP1 for fail-safe communication with a S7-1500.

Solution

Configuring project B (“F-I-Device”):
CPU, configured in “F-I-Device” mode with:
- Safety program and communication blocks “SENDDP” and “RCVDP”
- complete hardware configuration of a normal “F-I-Device” (transfer areas, network configuration)

“Dummy CPU” as representative of the “IO controller” from project A

Configuring project A (“IO controller”):
CPU, configured as “IO controller” with:
- Safety program and communication blocks “SENDDP” and “RCVDP”
  “Dummy CPU” which represents the “F-I-Device” CPU from project B
- Identical HW configuration like the “F-I-Device”
- IO controller assignment

Note

The transfer areas (HW identifier, name, size), IP address and PROFINET name of the “dummy CPU” and the “F-I-Device” must be identical.

The “dummy CPU” does not have to exist as a real hardware component, it only serves for configuration purposes.
2.4 Advantages of using F-I-Devices

2.4.1 Configuration with a “dummy CPU”

If the F-I-Device is to be configured with the help of a “Dummy CPU”, the following possible combinations are available:

Figure 2-7

<table>
<thead>
<tr>
<th>F-I-Device – Dummy CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>S7-1500 internal</td>
</tr>
<tr>
<td>S7-1500 internal</td>
</tr>
<tr>
<td>CM 1542-1</td>
</tr>
<tr>
<td>S7-300 Internal</td>
</tr>
<tr>
<td>CP343-1</td>
</tr>
<tr>
<td>S7-400 Internal</td>
</tr>
<tr>
<td>CP443-1</td>
</tr>
</tbody>
</table>

Failsafe not possible

2.4.2 Configuration with a GSD file

If the F-I-Device is to be configured with the help of a GSD file, the following possible combinations are available:

Figure 2-8

<table>
<thead>
<tr>
<th>F-I-Device – GSD file</th>
</tr>
</thead>
<tbody>
<tr>
<td>S7-1500 internal</td>
</tr>
<tr>
<td>S7-1500 internal</td>
</tr>
<tr>
<td>CM 1542-1</td>
</tr>
<tr>
<td>S7-300 Internal</td>
</tr>
<tr>
<td>CP343-1</td>
</tr>
<tr>
<td>S7-400 Internal</td>
</tr>
<tr>
<td>CP443-1</td>
</tr>
</tbody>
</table>

Failsafe not possible
3 Configuration

3.1 Infrastructure information

Software package

Install STEP 7 Professional V13 SP1 and the options package STEP 7 Safety on a PC/PG.

Required devices/components:

Use the following components for the configuration:

- Two CPU 1516F-3 PN/DP
- 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 S7-1500
- A PG/PC with the configuration tools “STEP 7 Professional V13 SP1” and “STEP 7 Safety 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 F-I-Device functionality, you can also use another fail-safe 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.
3.2  F-CPU as an F-I-Device in one project

3.2.1  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 1516F-3 PN/DP) and “F-I-Device” (CPU 1516F-3 PN/DP).

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 CPU 1516F-3 PN/DP (“F-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.
3.2.2 Configuring the F-I-Device function

Prerequisite

The device view of the CPU 1516F-3 PN/DP of the F-I-Device 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 F-I-Device itself or by the higher-level IO controller. Activate the checkbox for this solution.

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

**Result**
The CPU 1516F-3 PN/DP (“F-I-Device”) has now been configured as F-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 F-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. Select F-CD for fail-safe communication.

3. The addresses are pre-assigned automatically. Create two fail-safe transfer areas (F-CD) for this solution with a length of 12 bytes.

   ![Transfer area configuration](image)

   **Note**
   If required, you can adjust the addresses to your environment and specify the length of the transfer area to be transferred consistently. The start addresses of IO controller and F-I-Device can be chosen freely.

4. Create an additional transfer area by clicking in the second field. Change the address range direction with a click on the arrow symbol.

   ![Additional transfer area](image)

   **Note**
   For fail-safe communication, the length of the transfer area cannot be changed, as “SENDDP” and “RCVDP” can send and receive only 12 bytes. The start addresses of IO controller and F-I-Device can be chosen freely.
### 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.

### Note

Set the transfer areas in the real F-I-Device and in the “dummy CPU” in the same order. If the subslot numbers (see marking in fig. above) between F-I-Device and “dummy CPU” differ, the transfer areas must be reset in the same order.

### Calling the blocks SENDDP and RCVDP

Carry out the following steps in the “IO controller” and in the “F-I-Device”.

1. Open the FB1 “Main_Safety_RTG1” (which was generated automatically).
2. Call the “RCVDP” block in the first FB1 network.
   The “RCVDP” block requires a single-instance data block.
3. Call the “SENDDP” block in the first network of FB1.
   The “SENDDP” block requires a single-instance data block.

### Assignment of inputs

The following inputs of the two blocks can be pre-defined to enable communication with each other.

1. Input “DP_DP_ID”:
   The ID of the respective associated “SENDDP” and “RCVDP” must be unique in the network in order to be able to establish communication.
   This means: “DP_DP_ID” of “SENDDP” in the “IO controller” and “DP_DP_ID” of “RCVDP” in the “F-I-Device” must be identical.
   The same applies for “DP_DP_ID” of “RCVDP” in the “IO controller” and “DP_DP_ID” of “SENDDP” in the “F-I-Device”.

---

3 Configuration

---

© Siemens AG 2015. All rights reserved.
2. “LADDR” input:
   For the S7-1500, this requires that you enter the HW identifier of the respective used transfer area which is used by the block.

   Communication path:
   - “IO controller” (“SSENDP”) to “F-I-Device” (“RCVDP”) (1)
   - “F-I-Device” (“SSENDP”) to “IO controller” (“RCVDP”) (2)

   Figure 3-1

   Note
   Use the HW identifiers which are stored in the system constants to assign the “LADDR” input for the S7-1500.
   After a transfer area has been generated, its HW identifier is stored in the system constants and can be used symbolically.
   “PLC tags” > “Show all tags” > “System constants”

   For the S7-300/S7-400, it is required that you enter the start address of the respective transfer area which is used by the block.

   Communication path:
   - “IO controller” (“SSENDP”) (1) to “F-I-Device” (“RCVDP”) (2)
   - “F-I-Device” (“SSENDP”) (3) to “IO controller” (“RCVDP”) (4)

   Figure 3-2

   Note
   Use the start address of the respective transfer area to assign the “LADDR” input for the S7-300/S7-400.
   You can find them, for example, in the settings of the PROFINET interface of the CPU, under “Operating mode” > “I-Device communication”. 
3 Configuration

3. “TIMEOUT” input
Configure the “TIMEOUT” inputs of the “RCVDP” and “SENDDP” commands with the desired monitoring time. It is not necessary to configure the monitoring time.

**Note**
Further information on monitoring time can be found here [6].

4. “SD_BO_00” – “SD_BO_15” and “SD_I_00” and “SD_I_01”:
These inputs of the “SENDDP” block are written in the transfer areas. They can be changed during the safety program. Enter the data to be sent in these user program inputs.
3.3 F-CPU as an F-I-Device across projects

A fail-safe communication across projects between a SIMATIC S7-1500 as IO controller and a SIMATIC S7-1500 as F-I-Device can be realized by configuring “dummy CPUs”.

3.3.1 Implementing and configuring the F-I-Device (Project B)

Preparation

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

Add two new devices using the project navigation. Select the same CPU for the “Dummy CPU” which you use for the “IO controller” in project A. As F-I-Device, select the CPU used by you.

In order to be able to keep both devices apart regarding their function in this solution, change the internal project name to “Dummy CPU” (CPU 1516F-3 PN/DP) and “F-I-Device” (CPU 1516F-3 PN/DP).

Configuring the “dummy CPU”

Proceed as follows:

1. Select CPU 1516F-3 PN/DP (“dummy CPU”) 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.

Configuring the F-I-Device

Changing the operating mode
The device view of the CPU 1516F-3 PN/DP of the F-I-Device is now active and the properties of the selected interface are displayed in the inspector window.

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 Configuration

3. Assign the F-I-Device to the “dummy CPU” (IO controller).

4. 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 F-I-Device itself or by the higher-level IO controller. Activate the checkbox for this solution.

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

Result
The CPU 1516F-3 PN/DP (“F-I-Device”) has now been configured as F-I-Device and takes on the role of an IO device in the PROFINET network.
Creating a transfer area

Note

For information on the rules for transfer areas in fail-safe communication, refer to Chapter 2.1.

Transfer areas are the IO areas which are used to exchange data between the F-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 communication relationship type. Select F-CD for fail-safe communication.

3. The addresses are pre-assigned automatically. Create two fail-safe transfer areas (F-CD) for this solution with a length of 12 bytes.

Note

For fail-safe communication, the length of the transfer area cannot be changed, as “SENDDP” and “RCVDP” can send and receive only 12 bytes. The start addresses of IO controller and F-I-Device can be chosen freely.

4. Create an additional transfer area by clicking in the second field. Change the address range direction with a click on the arrow symbol.

Note

Each communication path between the “RCVDP” and “SENDDP” blocks requires such a transfer area.
3 Configuration

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.

<table>
<thead>
<tr>
<th>General</th>
<th>Device tags</th>
<th>System constants</th>
<th>Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Device tags</td>
<td>System constants</td>
<td>Tests</td>
</tr>
<tr>
<td>Ethernet</td>
<td>Address/1</td>
<td>Time synchronization</td>
<td>Operating mode</td>
</tr>
<tr>
<td>Real-time settings</td>
<td>Web server access</td>
<td>Hardware identifier</td>
<td>F-I-Device in IO</td>
</tr>
</tbody>
</table>

Note
Set the transfer areas in the real F-I-Device and in the “dummy CPU” in the same order. If the subslot numbers (see marking in fig. above) between F-I-Device and “dummy CPU” differ, the transfer areas must be reset in the same order.

Calling the blocks SENDDP and RCVDP

Carry out the following steps in the “F-I-Device” (project B):

1. Call the “RCVDP” block in the first FB1 network.
   The “RCVDP” block requires a single-instance data block.

2. Call the “SENDDP” block in the first network of FB1.
   The “SENDDP” block requires a single-instance data block.

Assignment of inputs

The following inputs of the two blocks can be pre-defined to enable communication with each other.

1. Input “DP_DP_ID”:
   The ID of the respective associated “SENDDP” and “RCVDP” must be unique in the network in order to be able to establish communication.
   This means: “DP_DP_ID” of “SENDDP” in the “IO controller” and “DP_DP_ID” of “RCVDP” in the “F-I-Device” must be identical.
   The same applies for “DP_DP_ID” of “RCVDP” in the “IO controller” and “DP_DP_ID” of “SENDDP” in the “F-I-Device”.

Note
An example for the assignment of the input “DP_DP_ID” is available in Chapter 2.2.
2. **“LADDR” input:**

For the S7-1500, this requires that you enter the HW identifier of the respective used transfer area which is used by the block.

Communication path:
- “IO controller“ ("SENNDDP") to “F-I-Device“ ("RCVDP") (1)
- “F-I-Device“ ("SENNDDP") to “IO controller“ ("RCVDP") (2)

**Note**

Use the HW identifiers which are stored in the system constants to assign the “LADDR” input.

After a transfer area has been generated, its HW identifier is stored in the system constants and can be used symbolically.

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

For the S7-300/S7-400, it is required that you enter the start address of the respective transfer area which is used by the block.

Communication path:
- “IO controller“ ("SENNDDP") (1) to “F-I-Device“ ("RCVDP") (2)
- “F-I-Device“ ("SENNDDP") (3) to “IO controller“ ("RCVDP") (4)

**Note**

Use the start address of the respective transfer area to assign the “LADDR” input for the S7-300/S7-400.

You can find them, for example, in the settings of the PROFINET interface of the CPU, under "Operating mode" > “I-Device communication".
3. “TIMEOUT” input:
   Configure the “TIMEOUT” inputs of the “RCVDP” and “SENDDP” commands with the desired monitoring time.

   **Note**
   Further information on monitoring time can be found here [6].

4. “SD_BO_00” – “SD_BO_15” and “SD_I_00” and “SD_I_01”:
   These inputs of the “SENDDP” block are written in the transfer areas. They can be changed during the safety program. Enter the data to be sent in these user program inputs.

### 3.3.2 Address assignment and loading (Project B)

**Note**
Before you can compile or load the project, you need to set a password for the safety CPU.

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 F-I-Device must be given the configured device name.

**Note**
As the “dummy CPU” and “F-I-Device” are stored in different projects, you need to make sure that the PROFINET name is identical.

Proceed as follows:

1. The device view of the CPU 1516F-3 PN/DP of the F-I-Device is now active and the properties of the selected interface are displayed in the inspector window.
3 Configuration

2. In the inspector window, select the parameter group “Ethernet addresses” and deactivate the checkbox “Generate PROFINET device name automatically”).

![Image of PROFINET configuration window]

3. Under “PROFINET device name”, enter “F-I-Device” as a name.

![Image of PROFINET configuration window]

**Note**
The “IO controller” supplies the “F-I-Device” with the IP address and the device name of the “dummy CPU”. This is the reason why you need to make sure that “dummy CPU” and “F-I-Device” have an identical hardware configuration.

**Loading the project**

To load the project data, select CPU 1516F-3 PN/DP (“F-I-Device”) in the project navigation and load the project into the module.

Further information on loading can also be found in the TIA Portal Online Help or in the system manual, see [4].
3.3.3 Implementing and configuring the IO controller and the dummy CPU (Project A)

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 used for the “IO controller”. Make sure you use the same CPU as a “dummy CPU” as the “F-I-Device” in the other project.

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 1516F-3 PN/DP) and “Dummy CPU” (CPU 1516F-3 PN/DP).

Configuring the IO controller
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.
3 Configuration

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.

Configuring the “dummy CPU”

1. Select CPU 1516F-3 PN/DP (“dummy CPU”) 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", select the subnet to be connected from the “Subnet” drop-down list.

Assign the Ethernet interface the same IP address as the “F-I-Device” from project B.

![Diagram showing Ethernet addresses configuration]

**Result**
The interface and the selected subnet are now connected.

**Note**
Make sure that device names and IP address correspond to the configuration of the F-I-Device from project B. If necessary, adjust these parameters.

**Changing the operating mode**
The device view of the CPU 1516F-3 PN/DP of the “dummy CPU” is now active and the properties of the selected interface are displayed in the inspector window.

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

![Diagram showing Operating mode configuration]
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.

The “dummy CPU” is assigned to the network of the “IO controller”.

![Network configuration diagram]

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 F-I-Device itself or by the higher-level IO controller. Activate the checkbox for this solution.

![Parameter assignment diagram]

**Note**

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

**Result**

The CPU 1516F-3 PN/DP (“dummy CPU”) has now been configured as F-I-Device and takes on the role of an IO device in the PROFINET network.
Creating a transfer area

**Note**

The rules for transfer areas in fail-safe communication are available [here](#).

Transfer areas are the IO areas which are used to exchange data between the F-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 communication relationship type. Select F-CD for fail-safe communication.

3. The addresses are pre-assigned automatically. Create two fail-safe transfer areas (F-CD) for this solution with a length of 12 bytes.

**Note**

The addresses must correspond to the addresses of the F-I-Device (project B).

**Note**

For fail-safe communication, the length of the transfer area cannot be changed, as “SENDDP” and “RCVDP” can send and receive only 12 bytes.

Make sure the IO addresses of controller and F-I-Device start with the same byte.
4. Create an additional transfer area by clicking in the second field. Change the address range direction with a click on the arrow symbol.

**Note**

Each communication path between the “RCVD” and “SENDDP” blocks requires such a transfer area.

**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.

**Note**

Set the transfer areas in the real F-I-Device and in the “dummy CPU” in the same order. If the subslot numbers (see fig. above) between F-I-Device and “dummy CPU” differ, the transfer areas must be reset in the same order.

Now, project A has an “IO controller” and a “dummy CPU” configured as “F-I-Device”.

The “dummy CPU” can now be used to represent the “F-I-Device” from project B. Now, the specific requirements and various programs can be initialized in both projects.
### Comparison of the “F-I-Device” and “dummy CPU” configuration

The configuration of “F-I-Device” and “dummy CPU” are identical except for the assignment of the subnet and the IO controller.

The figures below show the sections to be configured in the HW configuration of the two CPUs.

**Table 3-1**

<table>
<thead>
<tr>
<th>Excerpt from “F-I-Device”</th>
<th>Excerpt from “dummy CPU”</th>
</tr>
</thead>
<tbody>
<tr>
<td>The figures below show an excerpt from the configuration of the “Ethernet addresses” of “dummy CPU” and “F-I-Device”. IP addresses and device names of both components are identical.</td>
<td>IP addresses and device names of both components are identical.</td>
</tr>
<tr>
<td>The “F-I-Device” is not assigned to a subnet, as the respective subnet is in the other project.</td>
<td>The “dummy CPU” is assigned to the same subnet as the “IO controller”. This means the communication with the F-I-Device has been established.</td>
</tr>
<tr>
<td>The figures below show an excerpt from the configuration of the “Operating mode” of “dummy CPU” and “F-I-Device”. Operating modes and transfer areas of both components are identical.</td>
<td>The “dummy CPU” is assigned to the IO controller which enables communication between IO controller and “F-I-Device”.</td>
</tr>
</tbody>
</table>

The “F-I-Device” is assigned to a “dummy CPU” (IO controller) which enables communication between “IO controller” and “F-I-Device”.

The “dummy CPU” is assigned to the IO controller which enables communication between IO controller and “F-I-Device”.

---

**Excerpt from “F-I-Device”**

<table>
<thead>
<tr>
<th>Ethernet addresses</th>
<th>IP protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface networked with</td>
<td></td>
</tr>
<tr>
<td>Subnet</td>
<td>Add new subnet</td>
</tr>
<tr>
<td>not networked</td>
<td></td>
</tr>
<tr>
<td>102.149.0.1</td>
<td></td>
</tr>
<tr>
<td>Add new subnet</td>
<td></td>
</tr>
</tbody>
</table>

**Excerpt from “dummy CPU”**

<table>
<thead>
<tr>
<th>Ethernet addresses</th>
<th>IP protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface networked with</td>
<td></td>
</tr>
<tr>
<td>Subnet</td>
<td>Add new subnet</td>
</tr>
<tr>
<td>not networked</td>
<td></td>
</tr>
<tr>
<td>102.100.0.1</td>
<td></td>
</tr>
<tr>
<td>Add new subnet</td>
<td></td>
</tr>
</tbody>
</table>

---

**Excerpt from “F-I-Device”**

<table>
<thead>
<tr>
<th>Operating mode</th>
<th>Transfer areas</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Excerpt from “dummy CPU”**

<table>
<thead>
<tr>
<th>Operating mode</th>
<th>Transfer areas</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

© Siemens AG 2015. All rights reserved
Calling the blocks SENDDP and RCVDP

Carry out the following steps in the “IO controller” (project A)

Open the FB1 “Main_Safety_RTG1”.

1. Call the “RCVDP” block in the first FB1 network.
   The “RCVDP” block requires a single-instance data block.

2. Call the “SENDDP” block in the first network of FB1.
   The “SENDDP” block requires a single-instance data block.

Assignment of inputs

The following inputs of the two blocks can be pre-defined to enable communication with each other.

1. Input “DP_DP_ID”:
   The ID of the respective associated “SENDDP” and “RCVDP” must be unique in the network in order to be able to establish communication.

   This means: “DP_DP_ID” of “SENDDP” in the “IO controller” and “DP_DP_ID” of “RCVDP” in the “F-I-Device” must be identical.

   The same applies for “DP_DP_ID” of “RCVDP” in the “IO controller” and “DP_DP_ID” of “SENDDP” in the “F-I-Device”.

   Note: An example for the assignment of the input “DP_DP_ID” is available here.

2. “LADDR” input:
   This requires that you enter the HW identifier of the respective used transfer area which is used by the block.

Communication path:

- “IO controller” (“SENDDP”) to “F-I-Device” (“RCVDP”) (1)
- “F-I-Device” (“SENDDP”) to “IO controller” (“RCVDP”) (2)

Figure 3-7
3 Configuration

**Note**
Use the HW identifiers which are stored in the system constants to assign the “LADDR” input. After a transfer area has been generated, its HW identifier is stored in the system constants and can be used symbolically.

In device view, select the “F-I-Device” or the “IO controller”. Under “System constants”, you can find the respective HW identifier “PLC tags” > “Show all tags” > “System constants”

For the S7-300/S7-400, it is required that you enter the start address of the respective transfer area which is used by the block.

Communication path:
- “IO controller” (“SENDDP”) (1) to “F-I-Device” (“RCVDP”) (2)
- “F-I-Device” (“SENDDP”) (3) to “IO controller” (“RCVDP”) (4)

**Figure 3-8**

**Note**
Use the start address of the respective transfer area to assign the “LADDR” input for the S7-300/S7-400.

You can find them, for example, in the settings of the PROFINET interface of the CPU, under “Operating mode” > “I-Device communication”.

3. “TIMEOUT” input
Configure the “TIMEOUT” inputs of the “RCVDP” and “SENDDP” commands with the desired monitoring time.

**Note**
Further information on monitoring time can be found here [6].

4. “SD_BO_00” – “SD_BO_15” and “SD_I_00” and “SD_I_01”:
These inputs of the “SENDDP” block are written in the transfer areas. They can be changed during the safety program. Enter the data to be sent in these user program inputs.
3.3.4 Address assignment and loading (Project A)

Note
Before you can compile or load the project, you need to set a password for the safety CPU.

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 real F-I-Device must be given the configured device name.

Note
As the “dummy CPU” represents the real “F-I-Device” from project B, their device names must be identical.

Proceed as follows:

1. The device view of the CPU 1516F-3 PN/DP of the “dummy CPU” is now active and the properties of the selected interface are displayed in the inspector window.

2. In the inspector window, select the parameter group “Ethernet addresses” and deactivate the checkbox “Generate PROFINET device name automatically”). Figure 3-9
3. Under “PROFINET device name”, enter “F-I-Device” as a name.

Figure 3-10

Note
The “IO controller” supplies the real “F-I-Device” from project B the IP address and the device name of the “dummy CPU” when starting up. This is the reason why you need to make sure that “dummy CPU” and “F-I-Device” have an identical hardware configuration.

Loading the project

Note
Before you can compile or load the project, you need to set a password for the safety CPU.

To load the project data, select CPU 1516F-3 PN/DP (“IO controller”) in the project navigation and load the project into the module.

Further information on loading can also be found in the TIA Portal Online Help or in the system manual, see V4L.
3.4 **Diagnosis of configured F-I-Device communication**

After the previous chapters, the configuration of the F-I-Device function in either one or two projects is now complete and both controllers have established a PROFINET communication relationship.

3.4.1 **Correctly established F-I-Device communication**

The following chapter shows the input and output parameters of the “SENDDP” and “RCVDP” blocks in case of an established F-I-Device communication.

**Established communication at “SENDDP” block**

The following chapter shows the values of the inputs and outputs of the “SENDDP” block in case of a functioning F-I-Device communication.

Figure 3-11

For a functioning connection, the “ERROR” and “SUBS_ON” outputs at the “SENDDP” block and the “ERROR”, “SUBS_ON” and “ACK_REQ” outputs at the “RCVDP” block need to be “FALSE”.

The values for the input parameters “DP_DP_ID”, “TIMEOUT” and “LADDR” (HW-ID) have only been assigned as an example.
Established communication at “RCVDP” block

The following figure shows the values of the inputs and outputs of the “RCVDP” block in case of a functioning F-I-Device communication.

Figure 3-12

Note

For a functioning connection, the “ERROR”, “SUBS_ON” and “ACK_REQ” outputs at the “RCVDP” block need to be “FALSE”.

The values for the input parameters “DP_DP_ID”, “TIMEOUT” and “LADDR” (HW-ID) have only been assigned as an example.
3.4.2 Communication error during F-I-Device communication

The following chapter shows the input and output parameters of the “SENDDP” and “RCVDP” blocks in case of a communication error during the F-I-Device communication.

Note

In the case of communication errors, please refer to the chapter “Behavior in the event of communication errors” from Chapter 2.2.

Communication error at “SENDDP” block

The following chapter shows the values of the inputs and outputs of the “SENDDP” block in case of a communication error.

Note

The values for the input parameters “DP_DP_ID”, “TIMEOUT” and “LADDR” (HW-ID) have only been assigned as an example.

In the case of communication errors, please refer to the chapter “Behavior in the event of communication errors” from Chapter 2.2.
Communication error at “RCVDP” block

The following figure shows the values of the inputs and outputs of the “RCVDP” block in case of a functioning F-I-Device communication.

Figure 3-13

Note

The values for the input parameters “DP_DP_ID”, “TIMEOUT” and “LADDR” (HW-ID) have only been assigned as an example.

In the case of communication errors, please refer to the chapter “Behavior in the event of communication errors” from Chapter 2.2
## 4 Links & Literature

Table 4-1

<table>
<thead>
<tr>
<th>Topic</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Siemens Industry Online Support</td>
</tr>
<tr>
<td>2</td>
<td>Download page of the entry</td>
</tr>
<tr>
<td>3</td>
<td>PROFINET with: STEP 7 V13</td>
</tr>
<tr>
<td>4</td>
<td>STEP 7 Professional V13 SP1</td>
</tr>
<tr>
<td>5</td>
<td>Description of SENDDP and RCVDP</td>
</tr>
<tr>
<td>6</td>
<td>Monitoring time for fail-safe communication</td>
</tr>
<tr>
<td>7</td>
<td>Overview of the I-Device function support</td>
</tr>
</tbody>
</table>

## 5 History

Table 5-1

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.0</td>
<td>08/2015</td>
<td>First version</td>
</tr>
<tr>
<td>V2.0</td>
<td>11/2016</td>
<td>Adding check lists, selection help, more detailed description of “SENDDP” /”RCVDP” and transfer areas.</td>
</tr>
</tbody>
</table>