

SIEMENS



Application Example • 03/2017

Engineering and Commissioning of Multiple Use IO Systems

S7-1500, PRONETA, SIMATIC Automation Tool



<https://support.industry.siemens.com/cs/ww/en/view/29430270>

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 the 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. catalogues – the contents of the other documents shall have priority.

We do not accept any liability for the information contained in this document. Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of fundamental contractual obligations (“wesentliche Vertragspflichten”). The 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, systems, machines, and networks.

In order to secure plants, systems, machines, and networks against cyber threats it is necessary to implement (and to maintain continuously) a holistic, state-of-the-art Industrial Security concept. With this in mind, Siemens' products and solutions are only part of such a concept.

It is the client's responsibility to prevent unauthorized access to his plants, systems, machines, and networks. Systems, machines, and components should only be connected with the company's network or the Internet, when and insofar as this is required and the appropriate protective measures (for example, use of firewalls and network segmentation) have been taken.

In addition, Siemens' recommendations regarding appropriate protective action should be followed. For further information on Industrial Security, please visit <http://www.siemens.com/industrialsecurity>.

Siemens' products and solutions undergo continuous development to make them even more secure. Siemens explicitly recommends to carry out updates as soon as the respective updates are available and always only to use the current product versions. The use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates as they occur, subscribe to the Siemens Industrial Security RSS feed at <http://www.siemens.com/industrialsecurity>.

Table of Contents

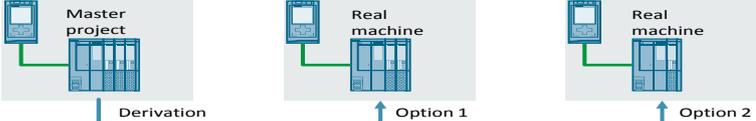
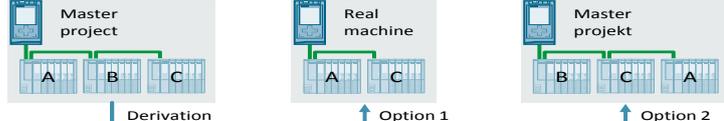
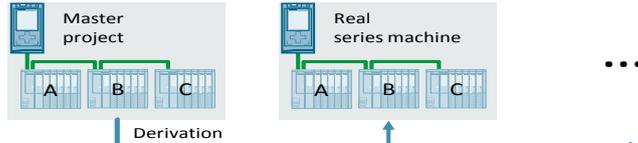
Warranty and Liability	2
Introduction	4
1 Task.....	5
1.1 Task definition	5
1.2 Possible solution	5
2 Principle of the Multiple Use IO System.....	7
2.1 Description of a “Multiple Use IO system”	7
2.2 Implementing a multiple use IO system	9
2.2.1 Implementation using the application program	9
2.2.2 Implementation without the application program using external tools	13
Overview of the SIMATIC Automation Tool	13
Overview of the PRONETA tool	13
2.3 Hardware and software components	14
2.3.1 Validity	14
2.3.2 Components used	14
3 Engineering and Programming	15
3.1 Infrastructure information	15
3.2 Configuring the IO system.....	16
3.3 Configuring the IO controller	19
3.4 Configuring the IO devices	20
3.5 Special case of a multiple use IO system with an HMI panel as IO device	21
4 Commissioning the IO system	26
4.1 Commissioning with application program via TIA Portal / SIMATIC Automation Tool.....	27
4.1.1 Loading process using the TIA Portal	30
4.1.2 Loading process using the SIMATIC Automation Tool	31
4.2 Commissioning without the application program using the SIMATIC Automation Tool / PRONETA	34
4.2.1 Assigning the IP addresses and the device name using the SIMATIC Automation Tool.....	35
4.2.2 Assigning the IP addresses and the device name using PRONETA	36
5 Links & Literature	37
6 History.....	37

Introduction

Definition of the term “Variants Management”

“Variants Management” is a generic term for an innovative machine concept in the series production of modular machines that may be adapted to customer requirements as easy as possible. Yet this presupposes possibilities that are just as flexible. An addressing of all the plant and machine parts performed at a time must be adaptable without great effort and changes in the engineering project.

The table below shows the fields of application of the variants management:

Term	Explanation of the fields of application	In this DOC
<p>Module-level Configuration Control</p>	<p>The module-level Configuration Control allows for flexible configuration levels of distributed and centralised I/O modules within a project. Thus a single STEP 7 project (maximum configuration) may be used for multiple configuration levels of the distributed stations.</p> 	<p>✗</p>
<p>Configuration Control for IO systems</p>	<p>The Configuration Control for IO systems allows for flexible configuration levels and interconnections of stations within an IO system. Thus a single STEP 7 project may be used for several concrete IO system versions as long as they can be derived from the maximum configuration.</p> 	<p>✗</p>
<p>Multiple Use IO systems</p>	<p>“Multiple Use IO systems” means that a single IO system is used for several machines. Thus a PROFINET IO system created in a STEP 7 project may be used for several machines since the IP addresses and the device names are fixed by the IO controller and not directly by the STEP 7 project.</p> 	<p>✓</p>

1 Task

1.1 Task definition

Description

In the series machine production, it is common practice that the PROFINET IO system of a machine (consisting of an IO controller and the IO devices attributed to it) is used in identical configuration in different plants or even within the same automation network. The systems only differ in their network address and their device name.

This is the reason why every machine required its own engineering project until now although the automation components were identical in all of them. Thus a lot of time and money has been spent for engineering and commissioning, added to the lack of flexibility.

Requirements

Based on typical application cases, the automation solution must meet the following requirements:

- One project (configuration and program) shall be loadable without any change to several machines of the same type.
- To connect the machine to an existing network infrastructure, the on-site commissioning shall only require some minor adjustments.

1.2 Possible solution

Description

The innovative function “Multiple Use IO System” ensures flexible automation solutions also in the series machine production.

By setting the IO system to “Multiple Use IO System”, the STEP 7 project becomes a “Series Machine Project”.

STEP 7 performs several configuration settings and checks ensuring thus that the IO system is self-contained and does not have any dependency to components outside the IO system. Parameters which vary from one machine to another (like the network address and the device name) will only be fixed with an external tool or by the application program during the commissioning.

This allows you to cover several identical automation systems with one and the same project.

Advantages

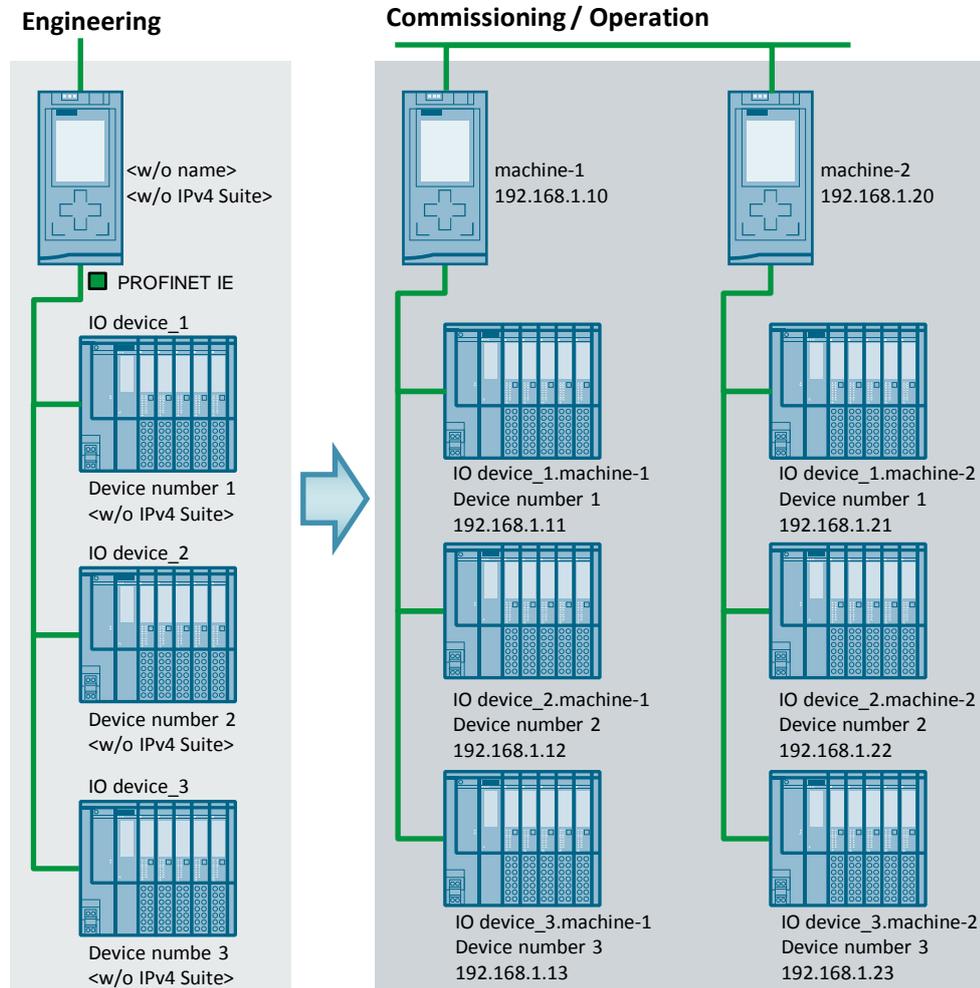
A “Multiple Use IO System” has the following advantages:

- A single project for several machines with identical set-up.
- Fewer adjustments during the on-site commissioning (IP address, device name).
- No programming unit with STEP 7 needed for commissioning; commissioning can also be done with tools like SIMATIC Automation Tool and PRONETA.

Schematic diagram

The Figure below shows the principle of a “Multiple Use IO System” as a graph.

Figure 1-1



Master Project:

- A single project
- Controller: Flexible Address
- Devices: Addresses adaptable at controller

Machine modules:

- Controller: Address assignment by external tool (SAT, PRONETA) or application program
- Devices: Addresses adapted from controller address

A “Multiple Use IO System” enables you to commission with a single project several machine modules of the same type with different network parameters. This essentially reduces the engineering effort and the length of the commissioning for such kind of plants.

Usable components

The components below support the function “Multiple use IO system”:

- S7-1500 CPU as of Firmware version V1.5 or higher as IO controller
- ET 200SP CPU as of Firmware version V1.6 or higher as IO controller
- STEP 7 as of V13 or higher
- Distributed I/O systems with PROFINET interface

2 Principle of the Multiple Use IO System

2.1 Description of a "Multiple Use IO system"

Concept

The automation components for a machine comprise a PROFINET IO system consisting of an IO controller (PROFINET interface of a CPU) and the IO devices attributed to it.

By setting the IO system to "Multiple Use IO System", the STEP 7 project becomes a "Series Machine Project". With this setting, STEP 7 performs certain configuration settings and checks. These settings ensure that the IO system is self-contained and does not have any dependency to components outside the IO system.

Application cases

The application cases below are typical for the multiple use IO system:

- The customer uses several machines in identical set-up (PROFINET IO system).
- Several customers use the machine in identical set-up in different plants.

Principle

The "Multiple Use IO System" is designed so as to allow loading one and the same project to several CPUs of the S7-1500. Afterwards, you can set for every CPU the related IP address and the device name.

The CPU, based on the topology engineered, then automatically starts assigning a unique device name as well as an IP address to any of the devices it is allocated to, starting from its own IP address.

The device names have the following structure: **Device name.Controller name**

The IP address consists of: **Controller IP address + device number.**

Rules

The rules below apply to a "Multiple use IO system":

- An IO device must not be configured as a Shared Device.
- The topology of the IO devices and the IO controller must be engineered.
- If an IO device is operated as an I device (CPU as "intelligent IO" device) within a multiple use IO system, the following rules shall apply:
 - If the I device has a lower-level IO system, it must not be connected to the same PROFINET interface as the higher-level IO controller.
 - The option "Parametrisation of the PN interface by higher-level controller" must be set at the PROFINET interface of the I device.

Note

If the I device is engineered through a PROFINET GSD, STEP 7 cannot check the compliance with this rule. You must ensure yourself that it is compliant.

- If MRP (Media Redundancy Protocol) is configured, all the IO devices at the multiple use IO system must belong to the same MRP domain.
- If IRT (Isochronous Real Time) is configured:
 - All the IO devices at the multiple use IO system must belong to the same Sync domain.
 - The Sync domain must not contain further IO devices.
- IO /PB Links cannot be operated as IO device at a multiple use IO system with STEP 7 V13.

Requirements

The requirements below must be met to prevent a series machine project with a multiple use IO system from having interdependencies with other devices outside the machine:

- A series machine project consists of one IO controller and the related IO devices. Consequently, only configure one CPU as IO controller and the related IO devices in the series machine project.
- Do not use bilateral but only unilateral connections (unspecified connections) for the communication.

Note

For further information on the multiple use IO system, please refer to the PROFINET manual ([13](#)).

2.2 Implementing a multiple use IO system

There are two options to implement the function “Multiple use IO system”:

- Within the application program using the “T_CONFIG” instruction.
- Without the application program:
 - Using the SIMATIC Automation Tool.
 - With PRONETA.

The chapters below give a detailed description of the two options.

2.2.1 Implementation using the application program

Description of the “T_CONFIG” instruction

The “T_CONFIG” module enables you to change the device name and the IP addresses of the IO controller.

General

The “T_CONFIG” instruction is used for the program-controlled configuration of the integrated PROFINET interface of the CPU or of a CP/CM, respectively.

The following settings can be done using the “T_CONFIG” instruction:

- IP protocol settings
 - IP address
 - Subnet mask
 - Router address
- PROFINET settings
 - PROFINET device name

The table below shows the input parameters of the “T_CONFIG” block.

Table 2-1

Parameters	Data type	Description
REQ	Boolean	Edge-triggered control parameter
INTERFACE	HW_Interface	HW flag of the PROFINET interface to be configured
CONF_DATA	Variant	Pointer to the configuration data record of the “Multiple Use IO system”

The table below shows the output parameters of the “T_CONFIG” block.

Table 2-2

Parameters	Data type	Description
DONE	Boolean	TRUE as soon as the instruction is completed.
BUSY	Boolean	TRUE if the instruction is active.
ERROR	Boolean	TRUE if the instruction is completed with error.
STATUS	DWord	Status of the instruction.
ERR_LOC	DWord	Information on the error location: 0: Error during execution or parametrisation. >0: Structure or content error in „CONF_DATA“

Mode of Operation

“T_CONFIG” is an instruction working in asynchronous mode. The complete processing of the task covers several cycles. However, only one task can be active at any time.

The task starts as soon as a positive edge is acquired at input “REQ”.

The output parameters “STATUS”, “BUSY”, “DONE”, and “ERROR” display the status of the task.

Note

The CPU restarts (warm start) once the task successfully executed.

“T_CONFIG” must be called up in OB 100 in a loop as long as one of the outputs “DONE” or “ERROR” is set.

Requirements

The following requirements must be met in the use of the “T_CONFIG” instruction:

- In the hardware configuration, make sure that you have set that the IP addresses and the device names are assigned by the application program.
- Configuring the PROFINET interface:
 - To change the IP address parameters, the option “IP address is set directly at device” must be activated.
 - To change the PROFINET device names, the option “PROFINET device name is set directly at the device” must be activated.
- The configuration data must be stored to the following system data types and delivered to the “CONF_DATA” parameter:
 - “IF_CONF_V4”: IP Address, subnet mask, router address.
 - “IF_CONF_NOS”: Device names of the IO devices pertaining to the IO system.

Note

The TIA portal does not suggest tags of the data type “IF_CONF_V4” and “IF_CONF_NOS”. They must be input manually.

Set-up of the “CONF_DATA” structure

The “CONF_DATA” structure contains the following information:

- Header: „IF_CONF_HEADER“:

Table 2-3

Name	Data type	Description
FieldType	UInt	Field type: Must always have the value "0".
FieldId	UInt	Field ID: Must always have the value "0".
SubfieldCount	UInt	Number of system data types used (“IF_CONF_V4” and “IF_CONF_NOS”): 1. Only one of the system data types is used. 2. Either system data types are used.

- Structure for IP addresses and subnets parametrisation (“IF_CONF_V4”):

Table 2-4

Name	Data type	Description
Id	UInt	System data type flag. The start value of this parameter must not be changed (“30”).
Length	UInt	Length of the system data type “IF_CONF_V4”.
Mode	UInt	Validity of addressing: 1: Permanent validity of the configuration data. 2: Temporary validity of the configuration data (erasing the permanent configuration data)
InterfaceAddress	IP_V4	IP address of the PROFINET interface.
SubnetMask	IP_V4	Subnet mask of the PROFINET interface.
DefaultRouter	IP_V4	Router address of the PROFINET interface.

- Structure for the PROFINET device names parametrisation (“IF_CONF_NOS”):

Table 2-5

Name	Data type	Description
Id	UInt	System data type flag. The start value of this parameter must not be changed (“40”).
Length	UInt	Length of the system data type “IF_CONF_NOS”. The absolute length information value for the “Length” parameter results from: - 6 bytes for the parameter ID, Length, and Mode. - Up to 240 bytes for the device name (“NOS”). For a dynamic length, use the default value (“246”) and make sure to put the value “0” after the device name.
Mode	UInt	Validity of addressing: 1: Permanent validity of the configuration data. 2: Temporary validity of the configuration data (erasing the permanent configuration data)
NOS	Array [1..240] of byte	Device name (Name of Station). For the device name, the rules and restrictions below must be complied with:

Rules and restrictions for the “NOS” parameter

For the “NOS” parameter, the rules below apply:

- The array must be allocated from the first byte on. If the first byte is set to “0”, the device name is erased.
- The minimum length is 2 bytes; the maximum length is 240 bytes.
- If the device name is shorter than indicated in the “Length” parameter, the device name must be followed by a “zero byte”.
- If the device name is longer than indicated in the “Length” parameter, the device name is only written to the indicated length.

The restrictions below apply for the device name:

- The name must be indicated in ASCII code.
- Only lower case letters, figures, hyphens, or dots may be used for the name.
 - The name must not begin or end with a hyphen.
 - The name must not begin with figures.
 - The name must not have the form x.x.x.x (x = 0, 1, ..., 999).
 - The name must not begin with the string “port-xyz” or “port-xyz-abcde” (a, b, c, d, e, x, y, z = 0, 1, ..., 9)
 - A name part between two dots may be 63 characters long as a maximum.
 - No special characters like mutated vowels, brackets, underline character, diagonal slash, space character, etc.

The “STATUS” parameter outputs the error code “C080_9400” if an inadmissible character is used.

2.2.2 Implementation without the application program using external tools

Overview of the SIMATIC Automation Tool

General

The SIMATIC Automation Tool is used to configure, operate, service, and document a controller program created and checked with the TIA Portal software.

Within a network consisting of many devices, the SIMATIC Automation Tool can execute the jobs of a group of devices at a time. This simplifies the process and saves time during the commissioning.

This free tool is available from the Siemens Industry Online Support ([\4](#)).

Functions

The most important functions of the SIMATIC Automation Tool are listed below:

- Scanning the network for available devices and listing of such devices.
- Loading of IP address, gateway, device name, programs or firmware to the available devices. This function is used for a multiple use IO system.
- Overall reset of a CPU memory.
- Setting a CPU to the RUN or STOP condition

Note

For further information on the download, the system requirements, the installation, the functions, and much more of the SIMATIC Automation Tool, please refer to the Siemens Industry Online Support ([\4](#)).

Overview of the PRONETA tool

General

PRONETA is a tool intended for quick analysis and configuration of PROFINET networks as well as testing of distributed ET 200 I/O systems. This free tool is available from the Siemens Industry Online Support ([\5](#)).

Functions

PRONETA includes the following functions:

- Network analysis:
The network analysis gives a quick overview of the devices installed within the PROFINET network and on how they are interconnected. The different network parameters (IP address, device name) are displayed and can be edited here. This function is used for a multiple use IO system.
- I/O Test
The I/O test is used to check the wiring of a distributed I/O device even before a CPU is installed.

Note

For further information on the download, the system requirements, the installation, the functions, and much more of the PRONETA tool, please refer to the Siemens Industry Online Support ([\5](#)).

2.3 Hardware and software components

2.3.1 Validity

This engineering example is valid for

- S7-1500 CPU as of Firmware version V1.5 or higher as IO Controller
- ET 200SP CPU as of Firmware version V1.6 or higher as IO Controller
- STEP 7 as of V14 or higher
- Distributed I/O systems with PROFINET interface

2.3.2 Components used

This engineering example was created using the following components:

Hardware components

Table 2-6

Component	Qty	Article Code	Note
CPU 1511-1 PN	1	6ES7511-1AK01-0AB0	Alternatively, you can use all the CPUs described in Chapter 2.3.1.
ET 200SP IM155-6PN ST	3	6ES7155-6AU00-0BN0	Alternatively, IO devices with PROFINET interface can also be used. The module set-up of the interface module is not relevant for the function shown.
HMI TP900 COMFORT	1	6AV2124-0JC01-0AX0	Alternatively, any other SIMATIC HMI COMFORT panel can be used.

Software components

Table 2-7

Component	Article Code	Note
STEP 7 PROFESSIONAL 14	6ES7822-1AA04-0YA5	Alternatively, a smaller package is also possible.
WinCC Professional 14	6AV210-.....4-0	Included in the TP900 Comfort Starter Kit.

Example files and projects

The list below includes all the files and projects that are used in the present engineering example.

Table 2-8

Component	Note
29430270_MultipleIOSys_PROJ_v20.zip	<This zip file contains the STEP 7 project.>
29430270_MultipleIOSys_DOCU_v20_d.pdf	The present document.

3 Engineering and Programming

3.1 Infrastructure information

Software package

Install STEP 7 Professional V14 on your PC/PG.

Setting up the infrastructure

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

Note

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

Implementing and configuring the devices

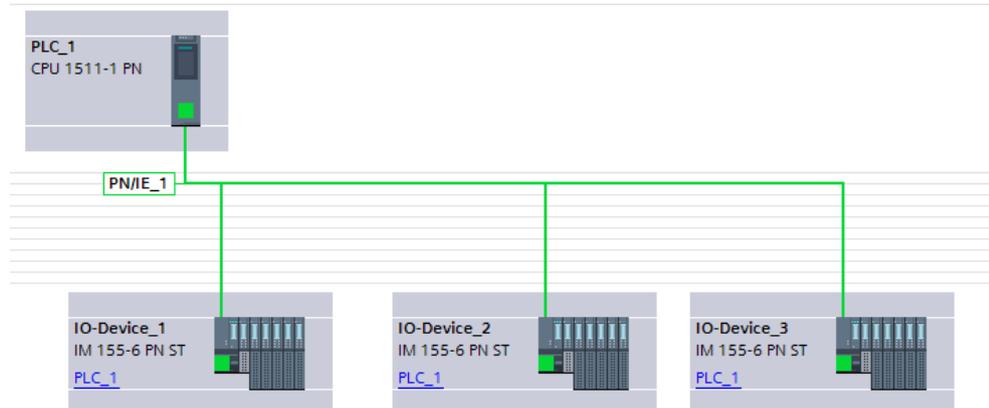
Open the TIA Portal configuration software and create a new project. Create a hardware configuration with the available controller you wish to use and the IO devices.

Create a new PROFINET network and connect all the components involved to this network.

Result

The interfaces of both the IO devices and the IO controller are now connected to the selected subnet. The address parameters of the interface are thus automatically set in a consistent manner.

Figure 3-1



3.2 Configuring the IO system

Requirements

The following requirements must be met:

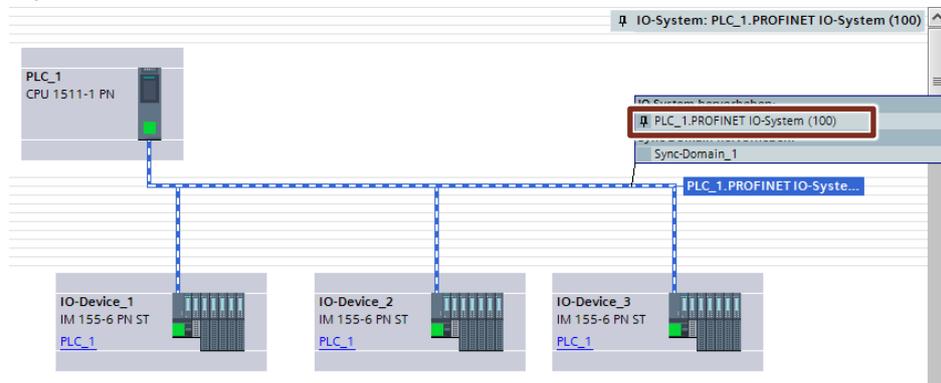
- The IO controller is an S7 1500-CPU as of FW \geq V1.5 or an ET 200SP-CPU as of FW \geq V1.6.
- The same PROFINET IO system was assigned to both the IO controller and the IO devices.

Configuring the IO system

The steps below explain the settings to do at the PROFINET IO system:

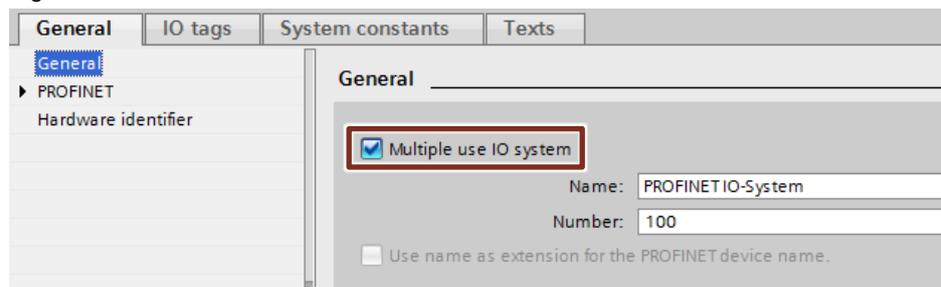
1. Select the IO system whose properties you wish to edit in the inspector window.

Figure 3-2



2. Check the “Multiple use IO system” check box in the “General” area of the inspector window.

Figure 3-3

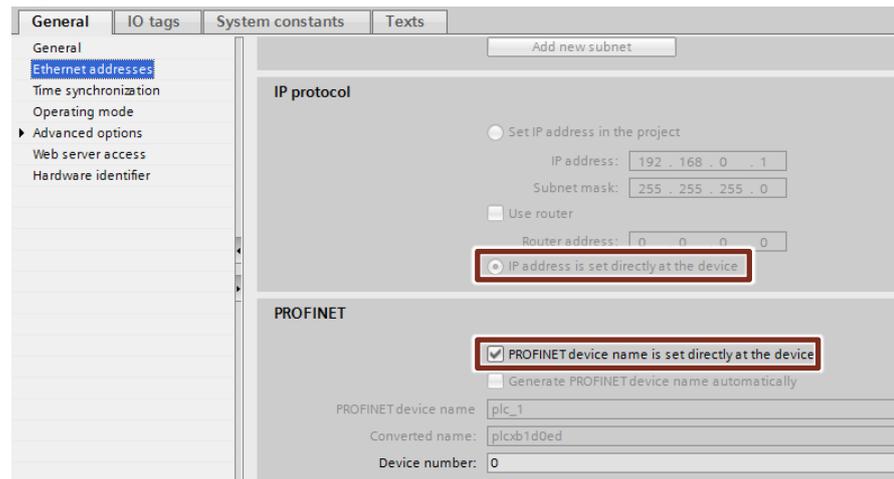


Result

Do the settings below at the devices in the STEP 7 IO system:

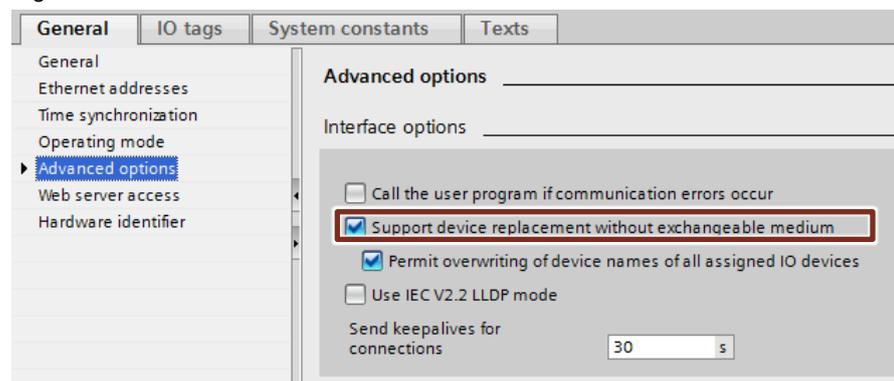
- IO controller:
 - The option “PROFINET device name is set directly at the device” is set. Initially, the IO controller does not have a PROFINET device name.
 - The option “IP address is set directly at the device” is set. Initially, the IO controller does not have an IP address.

Figure 3-4



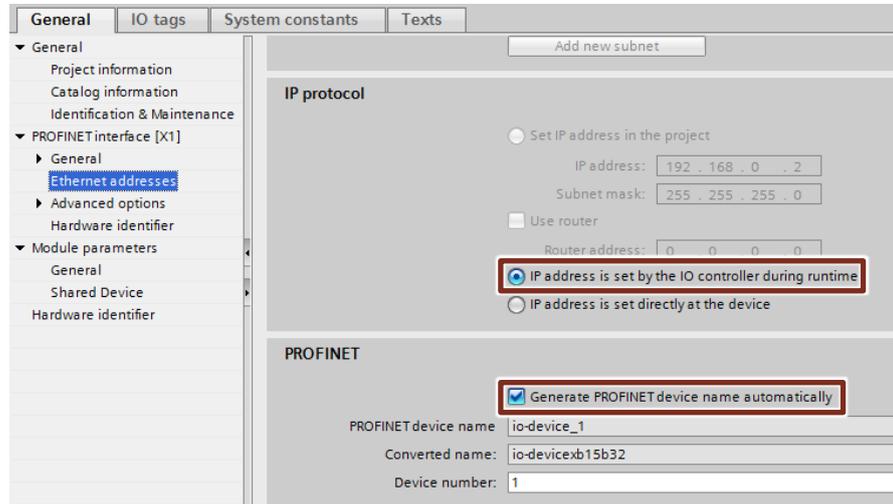
- The option “Support device replacement without exchangeable medium” is activated. This facilitates an automatic commissioning. The commissioning engineer must no longer assign the device name and the IP address. Based on the target topology and the other settings, the IO controller assigns to the IO devices the device name and the IP address during start-up.

Figure 3-5



- IO-Devices:
 - The option “Generate PROFINET device name automatically” is active.
 - The option “IP address is set by the IO controller during runtime” is active. Initially, the IO devices do not have an IP address.

Figure 3-6



- The device name is automatically assigned to the IO devices and is used on site to disambiguate the IP address.

3.3 Configuring the IO controller

Prerequisites

The following requirements must be met:

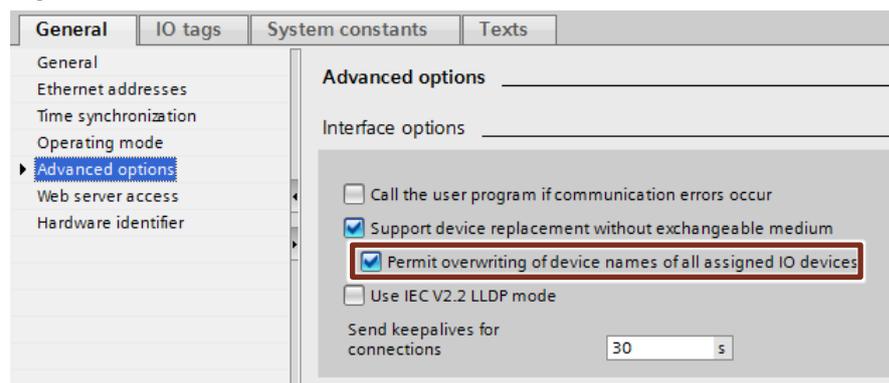
- The S7-1500 CPU used has firmware \geq V1.5.
- The IO controller is assigned to a subnet.

Configuring the IO controller

Further settings are necessary at the IO controller to configure a “Multiple use IO system”:

1. Open the device view of the IO controller by double clicking “Device configuration”.
2. Click the PROFINET interface of the IO device to be connected. Access the “Advanced options”.
3. Check the option “Permit overwriting of device names of all assigned IO devices”.

Figure 3-7



Result

The CPU is now able to overwrite the device names and the IP address of the IO devices assigned to it.

3.4 Configuring the IO devices

Requirements

The following requirements must be met:

- The interface modules used are equipped with a PROFINET interface.
- The IO system is defined as “Multiple use IO system”.

Configuring the interface modules

To configure a multiple use IO system, you must not do any settings at the IO device as long as the option “multiple use IO system” is activated for the IO system (see [Chapter 3.3](#)).

Note

A multiple use IO system is perfect for the use in the Configuration Control because the huge number of options and variants for series machines ensure maximum flexibility and customizability. For an engineering instruction of the Configuration Control, please refer to the entry sheet ([12](#)).

3.5 Special case of a multiple use IO system with an HMI panel as IO device

Requirements

The requirements below must be met if an HMI panel shall be operated as IO device in a multiple use IO system:

- SIMATIC Comfort Panel
- The IO system and the IO controller are configured as described in [Chapter 3.3](#) and [3.4](#).

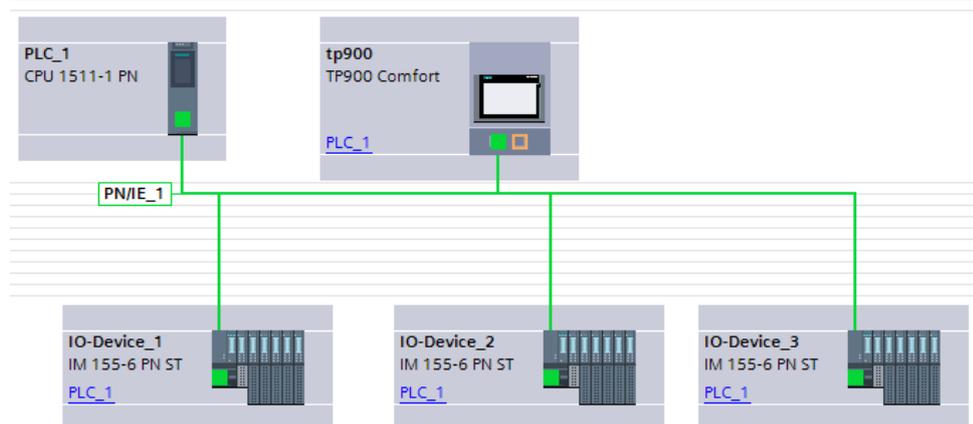
Software package

Install WinCC Professional V14 on your PC/PG.

Characteristic feature

A panel can also be built in as a device to which an IP address and a device name are assigned in this case. Consequently, the HMI communication from the panel to the CPU points to the IP address configured for the CPU in the project. This is the reason why this address does not work any longer.

Figure 3-8



Solution

To be able to communicate, the HMI panel must establish a new communication to the CPU.

The HMI panel can determine its own IP address with a script. Taking this one as basis, it is possible to calculate the IP address of the CPU with the help of the device number of the HMI panel. The HMI panel obtains its device number from the loaded TIA Portal project.

CPU IP address = Panel IP address - Panel device number

The IP address of the HMI panel is determined. The script is called up at the moment when the HMI communication between CPU and HMI panel is aborted (event 140001). If the IP address has not changed after completion of the script, the "event 140001" occurs again and the script is executed once more.

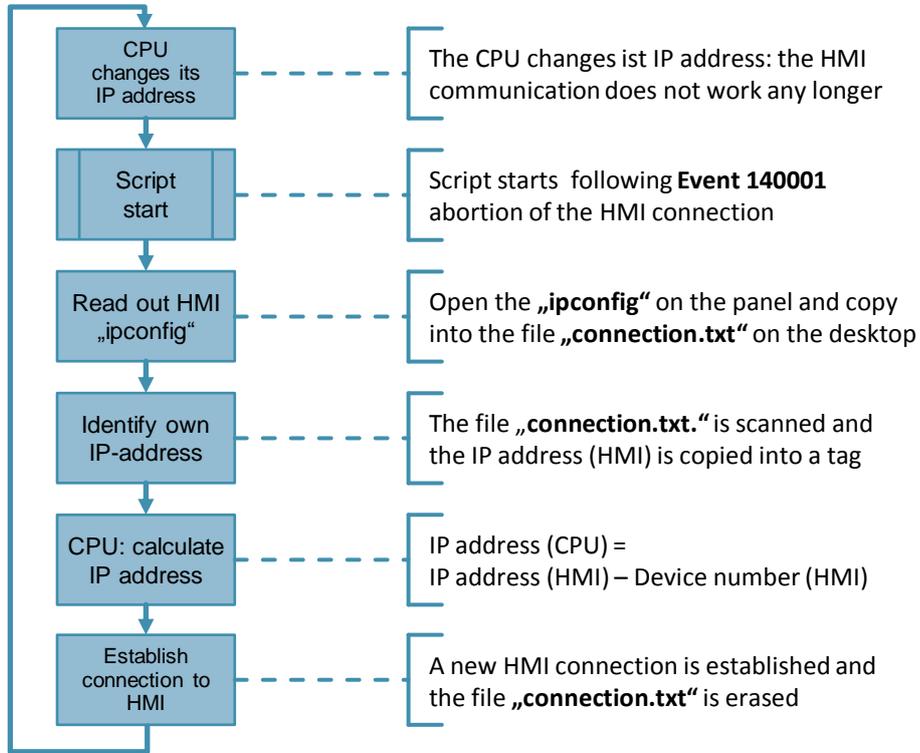
Note

You need a SIMATIC Comfort Panel for this solution.

Sequence of the script

The Figure below shows the sequence of the script:

Figure 3-9

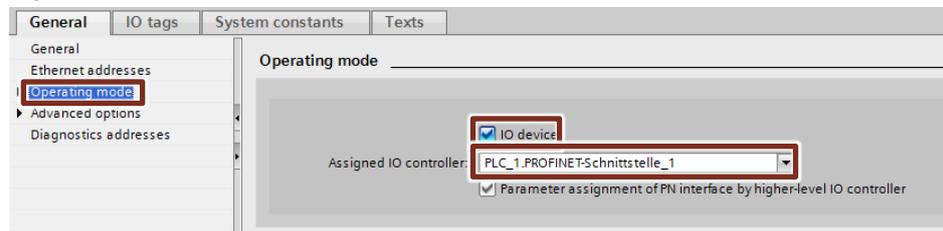


Configuring the HMI panel

The HMI must be configured as follows:

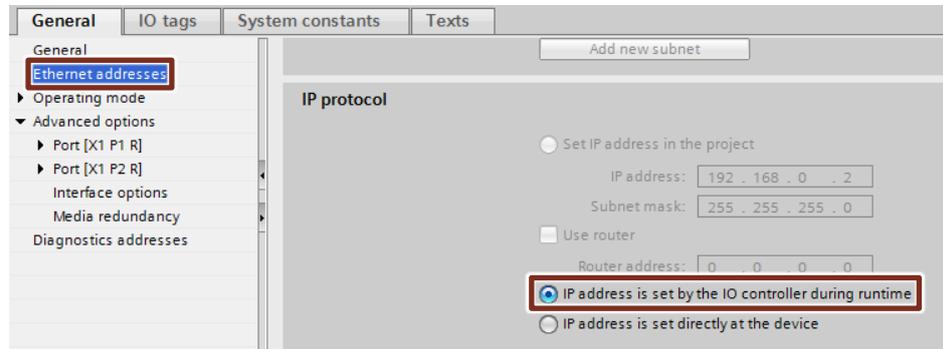
1. In the graphical view, select the HMI panel interface to be networked. The properties of the selected interface are displayed in the inspector window.
2. Select the parameter group “Operating mode”, activate the option “IO device”, and assign the controller to the HMI under “Assigned IO controller”.

Figure 3-10



3. Select the parameter group “Ethernet addresses” and activate the option “IP address is set by the IO controller during runtime”.

Figure 3-11

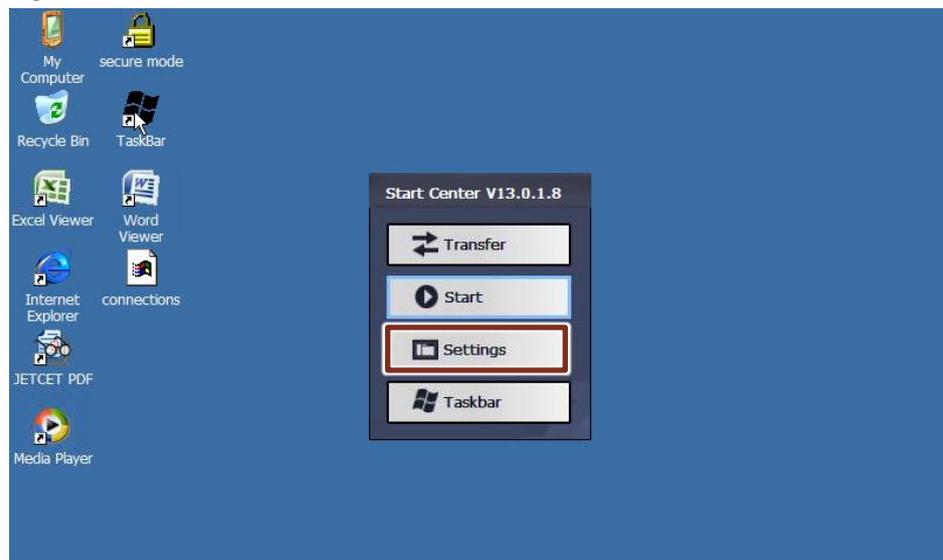


In addition, you must enable the PROFINET directly at the HMI. Disabled by default, PROFINET would not be activated by loading the configuration into the panel.

Proceed as follows:

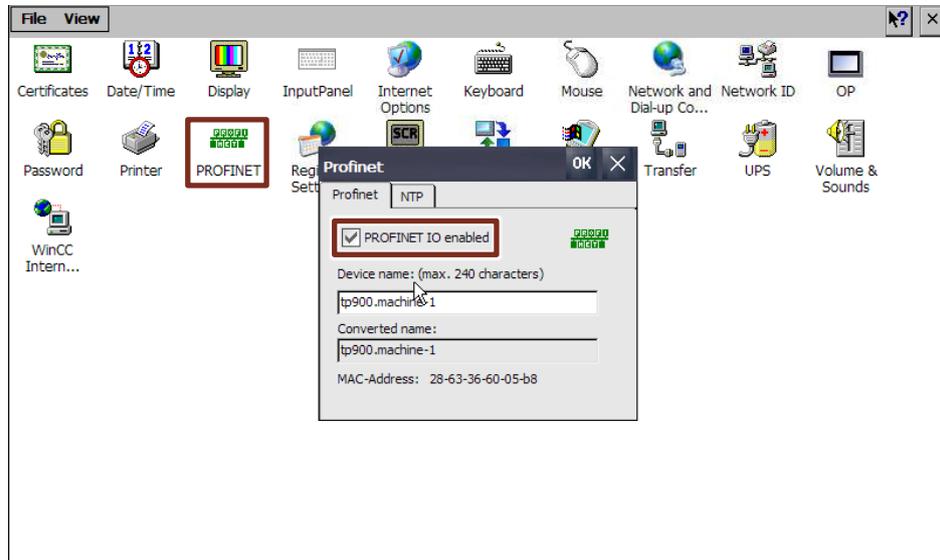
1. Terminate the runtime running on your HMI panel if you have already started one.
2. Open the “Settings” in the HMI.

Figure 3-12



3. Open the “PROFINET” menu. The “Profinet” pop-up window opens. Enable the “PROFINET IO enabled” option.

Figure 3-13



Result

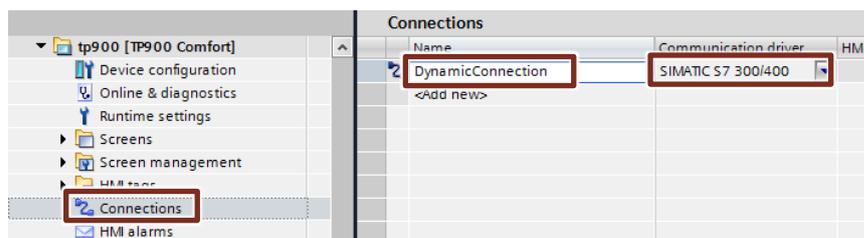
The HMI is now declared as an IO device and has been assigned to the IO controller. The IO controller assigns now the IP address and the device name.

Inserting a script

The HMI communication discontinues after having changed the IP address and the device name. It is necessary to execute a script to re-establish the HMI communication. The script with the new connection data re-establishes the HMI communication. You need the VB scripts “SetConnection” and “GetStringParts” joined to the documentation for this purpose.

Proceed as follows:

1. Insert a connection under “Connections”. Enter your name and activate “SIMATIC S7 300/400” as “Communication driver”.



2. Copy the scripts “SetConnection” and “GetStringParts” from the joined example project („29430270_MultipleIOSys_PROJ_v10.zip“). You will find the scripts under “Scripts” > “VB scripts”.

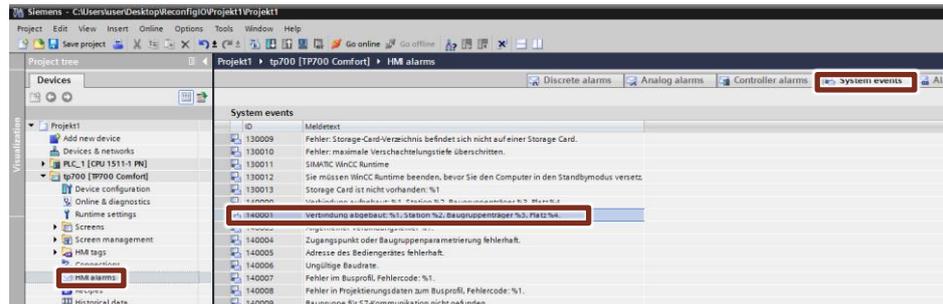
3. Insert the scripts “SetConnection“ and „GetStringParts“ under “Scripts“ > “VB scripts“ into your project. In line # 66 of the script “SetConnection“, change the name to the one that you have assigned to your connection in step 1.

```

64 temparr = Split(IPAddr, ".")
65 tempIP = (((temparr(0) * 256 + temparr(1)) * 256 + temparr(2)) * 256 + temparr(3)) - DeviceNumber
66 ChangeConnector "DynamicConnection" tempIP, 1, 0
67
68 fso.DeleteFile(path)
69 Set fso = Nothing 'release memory
70
71 End Sub
    
```

4. In the menu “HMI alarms“, access the “System events“ tab. Assign the function “SetConnection“ to the event “140001“ under “Events“ and enter the device number.

Figure 3-14



Result

The necessary scripts are added and will be called up at event 140001 (“HMI connection aborted”) so that a new HMI connection will establish.

4 Commissioning the IO system

General

Once all the IO system components correctly configured in the project, the different machines involved may be commissioned with a Master project.

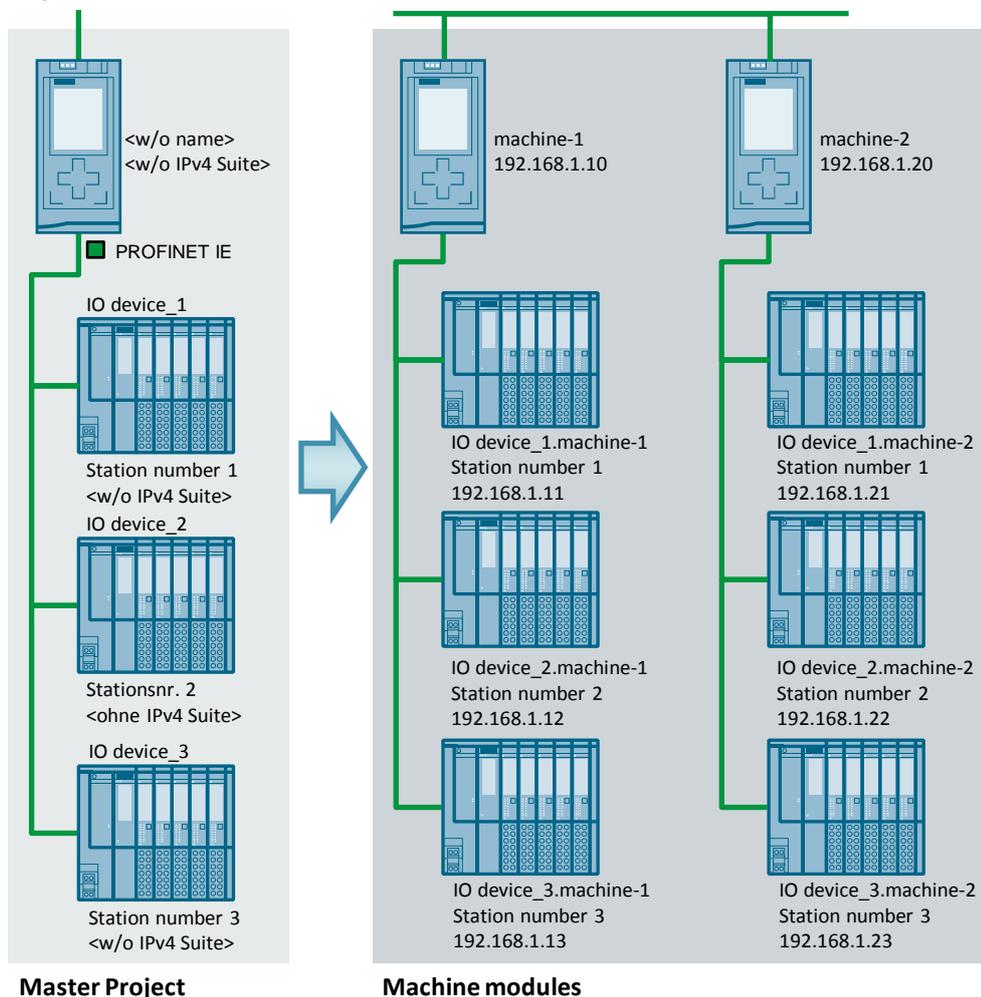
There are two options to commission the machines:

- Commissioning with the application program using the “T_CONFIG” instruction via TIA Portal or SIMATIC Automation Tool.
- Commissioning without the application program but with an external tool like PRONETA / SIMATIC Automation Tool.

Example

The Figure below shows the engineered example to be commissioned.

Figure 4-1



The following sections explain the different commissioning options.

4.1 Commissioning with application program via TIA Portal / SIMATIC Automation Tool

Requirements

The requirements below must be met to commission the multiple use IO system using the application program:

- The IO system is configured as “Multiple use IO system”.
- The IO controller is qualified for overwriting the device names of all the IO devices assigned.
- The IO devices are correctly configured and the IP address and the device names are assigned by the IO controller.
- STEP 7 as of V13 or higher

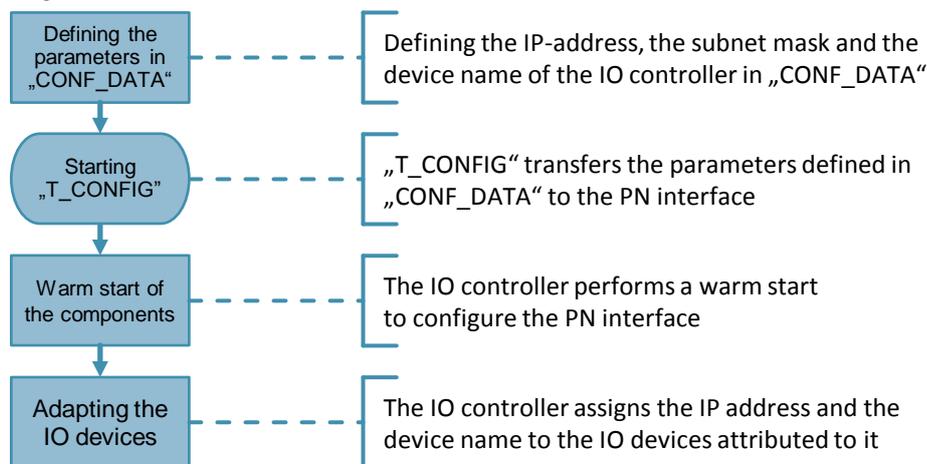
General

When commissioning the machines involved by means of the application program, the “T_CONFIG” instruction will assign to the IO controller its IP address, subnet address and device name.

For this purpose, the data record “CONF_DATA” must be adapted to the desired machine data for any machine.

The Figure below shows the principle how the parameters are assigned:

Figure 4-2



Preparing the “CONF_DATA” structure

The structure “CONF_DATA” must be parametrised in order that the IO controller and the IO devices get the correct IP address and device names. Two “CONF_DATA” structures are necessary for the present example:

- „CONF_DATA“ for module 1 with „machine_1“.
- „CONF_DATA“ for module 2 with „machine_2“

The header "IF_CONF_HEADER" is identical for either module and parametrised as follows:

Table 4-1

Name	Parametrisation	Description
FieldType	0	The value must always be "0".
FieldId	0	The value must always be "0".
SubfieldCount	2	Either system data type "IF_CONF_V4" and "IF_CONF_NOS" is used.

As module 1 and module 2 use different IP addresses, the system data type "IF_CONF_V4" must be adapted to every module.

In the following, we will show the structure for module 1 with the IP address 192.168.0.10. Module 2 is parametrised analogously but with the IP address 192.168.0.20:

Table 4-2

Name	Parametrisation	Description	
Id	30	The start value of this parameter must not be changed ("30").	
Length	18	Length of the system data type "IF_CONF_V4" in bytes.	
Mode	1	The configuration data are permanently valid.	
Interface Address	1	192	IP address of the IO controller's PROFINET interface. Parametrisation for "machine-1": 192.168.0.10 Parametrisation for „machine-2“: 192.168.0.20
	2	168	
	3	0	
	4	10	
Subnet Mask	1	255	Subnet masks of the IO controller's PROFINET interface. Parametrisation for "machine-1": 255.255.255.0 Parametrisation for "machine-2": 255.255.255.0
	2	255	
	3	255	
	4	0	

As module 1 and module 2 use different device names, the system data type "IF_CONF_NOS" must be adapted to every module.

In the following, we will show the structure for module 1 with the device name "machine-1". Module 2 is parametrised analogously but with the device name "machine-2":

Table 4-3

Name	Parametrisation	Description	
Id	40	System data type flag. The start value of this parameter must not be changed ("30").	
Length	15	Length of the system data type "IF_CONF_NOS". - 6 bytes for the parameter ID, Length, and Mode. - 9 bytes for the device name ("NOS"). For a dynamic length, use the default value ("246") and make sure to put the value "0" after the device name.	
Mode	1	The configuration data are permanently valid.	
NOS	1	'm'	Device name (Name of Station).

Name	Parametrisation	Description
2	,a'	Parametrisation for "machine-1": "machine-1" Parametrisation for "machine-2": "machine-2" When parametrising the device name, follow the rules for device names (see Chapter 2.2).
3	'c'	
4	,h'	
5	'i'	
6	,n'	
7	'e'	
8	,-'	
9	,1'	

Note

For further information on the parameters of the "CONF_DATA" structure, please refer to [Chapter 2.2.1](#).

The structure "CONF_DATA" must be parametrised individually for any IO controller.

Application program calling "T_CONFIG"

The following steps are necessary within the application program:

1. Add OB 100 "Startup" with the programming language "SCL" to your application program.
2. Call the "T_CONFIG" instruction in the controller's OB 100 "Startup" and execute it in a loop as long as the IO controller is parametrised ("T_CONFIG" parameter "Done" = "true").
3. Set the "Req" parameter to "true" and the "Interface" parameter to the HW tag of the controller.

Use the CPU system tags for the interconnection of "Interface":
"PLC tags" > "Show all tags" > "System constants"

The Figure below shows the call of the "T_CONFIG" instruction.

Figure 4-3

```

1 WHILE NOT "T_CONFIG_DB".Done DO
2   "T_CONFIG_DB"(Req := true,
3     Interface := "Local-PROFINET-Schnittstelle_1",
4     Conf_Data := "CONF_DATA".ConfData);
5 END_WHILE;
```

4.1.1 Loading process using the TIA Portal

Use a RJ45 socket to connect your programming unit to the IO controller's PROFINET interface or to the network to which the IO controller is connected and load the project into the CPU.

Result

The application program containing the hardware configuration is loaded into the controller. The controller restarts after the loading process is complete. OB 100 "Startup" is executed and thus also the "T_CONFIG" block. The IO controller's PROFINET interface is parametrised according to the structure in "CONF_DATA". Finally, the IO controller overwrites the IP addresses and device names of the IO devices assigned to it.

Note

This commissioning method requires that the project with the adapted data structure is individually loaded into any IO controller. For this purpose, repeat steps 1 thru 4 for any IO controller.

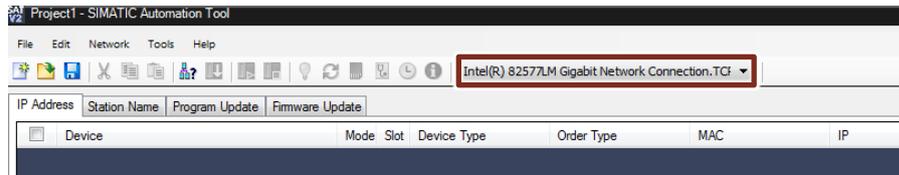
4.1.2 Loading process using the SIMATIC Automation Tool

Preparing the SIMATIC Automation Tool

Some settings at the SIMATIC Automation Tool are necessary before you can load the projects into the IO controllers.

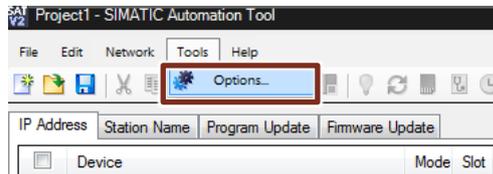
1. Set the network adapter to the one you are using here.

Figure 4-4



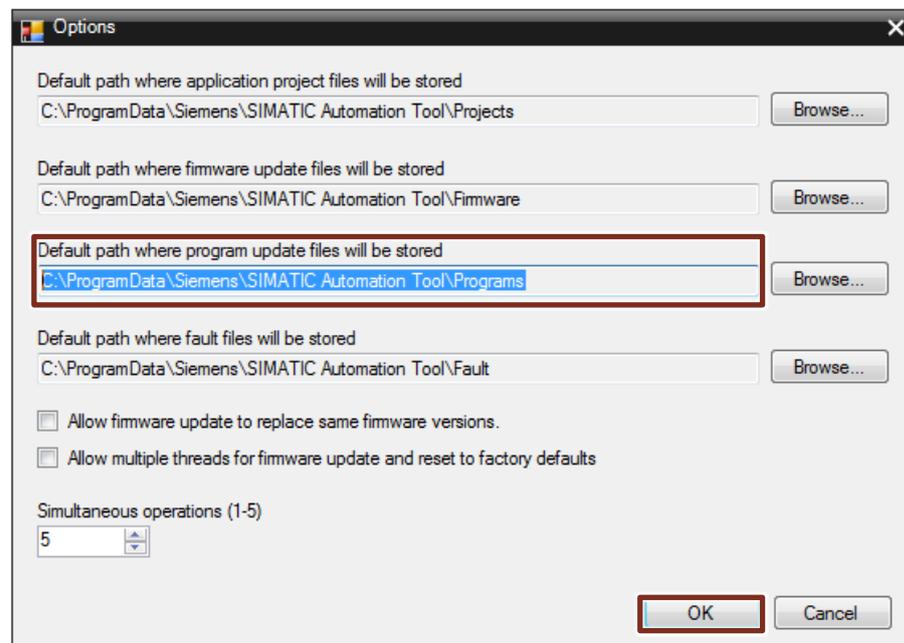
2. Fix the folder where the SIMATIC Automation Tool stores the program updates. For this purpose, open the SIMATIC Automation Tool options ("Tools > "Options").

Figure 4-5



3. The window in the Figure below opens. Here you can see the location where the projects, firmware updates, programs and error files used by the SIMATIC Automation Tool are stored. Click "OK" to accept the set data.

Figure 4-6

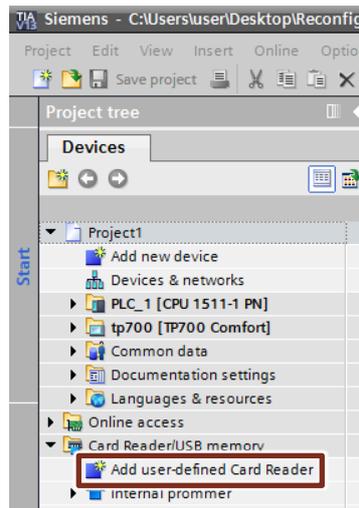


Preparing the Master project for SIMATIC Automation Tool

To use a program with the SIMATIC Automation Tool, the project must at first be transferred from the TIA Portal software to a SIMATIC memory card, an USB stick, or a hard disk partition, respectively. Proceed as follows:

1. Use the function “Add user-defined Card Reader” under “Card Reader/USB memory” to create a storage location for the program data of the Master project.

Figure 4-7



2. Select the same folder where the SIMATIC Automation Tool stores the program updates as the storage location. You will find this folder in the SIMATIC Automation Tool under “Tools” > “Options” > „Default path where program update files will be stored“ (see [Figure 4-8](#)).
3. Move the CPU of the Master project into the new tab using Drag&Drop.

Note

Use an own folder for any program within the storage location of the SIMATIC Automation Tool. So you can select here the folder from which the program shall be loaded into the controller.

Result

The project was stored to the same folder where the SIMATIC Automation Tool stores the program updates. The SIMATIC Automation Tool can now load the project stored into the different CPUs.

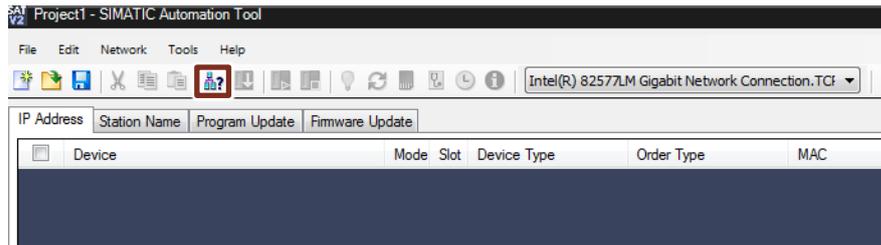
Loading the projects using the SIMATIC Automation Tool

This method requires that every IO controller gets its own project with its own “CONF_DATA” structure created and stored for the SIMATIC Automation Tool.

The SIMATIC Automation Tool is set and the Master project is correctly configured and stored. The program can now be loaded to the IO controllers using the SIMATIC Automation Tool. Proceed as follows:

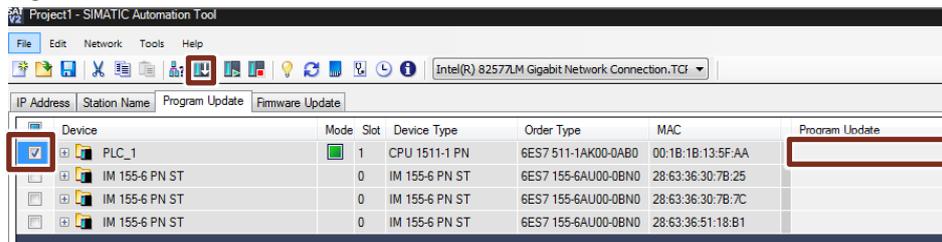
1. Scan the network with the SIMATIC Automation Tool using the button “Scan devices on network”.

Figure 4-8



2. Within the SIMATIC Automation Tool, access the “Program Update” tab. Select the IO controllers to which the program shall be loaded as well as the program which shall be loaded into the controller. Finally, load the programs to the selected IO controllers.

Figure 4-9



Result

The configuration of the IO system, the IO controller, and the IO devices is loaded now. Neither IP addresses nor device names are assigned yet.

4.2 Commissioning without the application program using the SIMATIC Automation Tool / PRONETA

Requirements

The requirements below must be met to commission the multiple use IO system using PRONETA:

- The Master project is engineered, without faults, and operational.
- PRONETA or SIMATIC Automation Tool are installed on the programming unit / PC.
- The controllers used are compatible with the SIMATIC Automation Tool (see [4](#)).

General

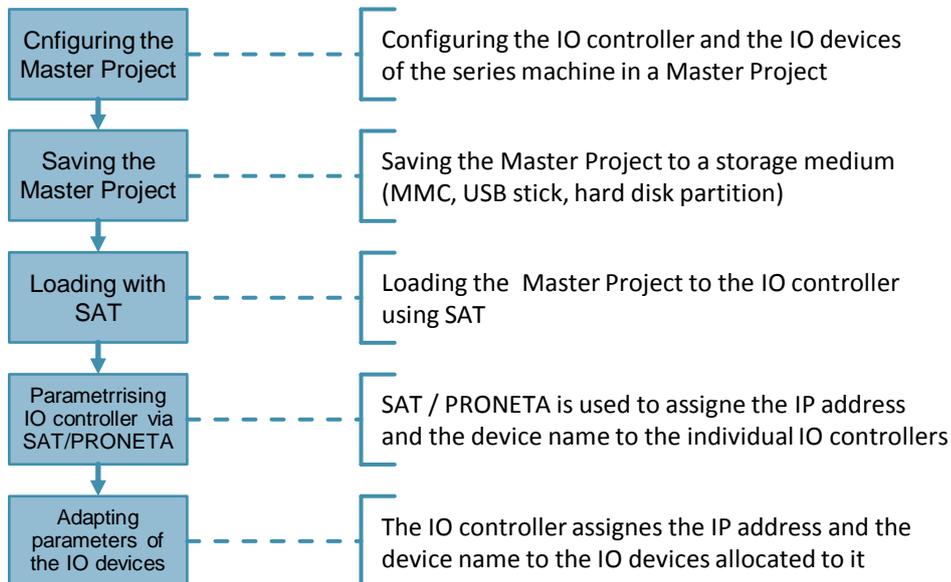
Using the SIMATIC Automation Tool, the Master project can be loaded to all available stations as long as they are compatible with the SIMATIC Automation Tool. As soon as an address is assigned to the IO controller, the latter assigns then an IP address and a device name to the IO devices allocated to it.

For this purpose, the IO system, the IO controller and the IO devices in the Master project must be configured like in [Chapters 3.3, 3.4, and 3.5](#).

After the Master project has been loaded to the different IO controllers of the individual machines, they still do not have an IP address or a device name. These parameters can be assigned using optionally the SIMATIC Automation Tool (SAT), PRONETA, or an HMI panel.

The Figure below shows the principle how the parameters are assigned:

Figure 4-10



Preparing the SIMATIC Automation Tool

Some settings at the SIMATIC Automation Tool are necessary before you can load the Master project into the IO controllers (see [Chapter 4.1.2](#)).

Preparing the Master project for SIMATIC Automation Tool

To use a program with the SIMATIC Automation Tool, the project must at first be transferred from the TIA Portal software to a SIMATIC memory card, an USB stick, or a hard disk partition, respectively (see [Chapter 4.1.2](#)).

Loading the program using the SIMATIC Automation Tool

The SIMATIC Automation Tool is set and the Master project is correctly configured and stored. The program can now be loaded to the IO controllers using the SIMATIC Automation Tool (see [Chapter 4.1.2](#)).

Result

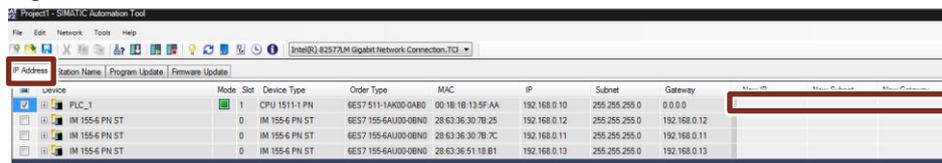
The configuration of the IO system, the IO controller, and the IO devices is loaded now. Neither IP addresses nor device names are assigned yet.

4.2.1 Assigning the IP addresses and the device name using the SIMATIC Automation Tool

After the Master project has been loaded to the IO controllers, you have to assign the IP address, the subnet mask, and the device name to the IO controllers. These parameters can be configured using the SIMATIC Automation Tool or PRONETA, respectively. Proceed as follows:

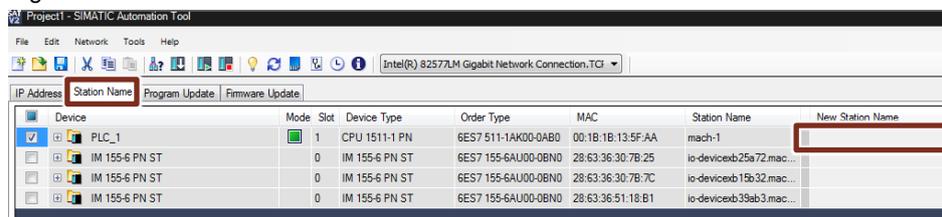
1. Within the SIMATIC Automation Tool, access the “IP address” tab. Assign the desired IP addresses and subnet masks to the IO controllers.

Figure 4-11



2. Within the SIMATIC Automation Tool, access the “Station name” tab. Assign the desired device name to the IO controllers.

Figure 4-12



Result

The IP address and the device name are assigned to the IO controllers. Thus, the IO controller may now assign an IP address and a device name to the IO devices attributed to the PROFINET IO system.

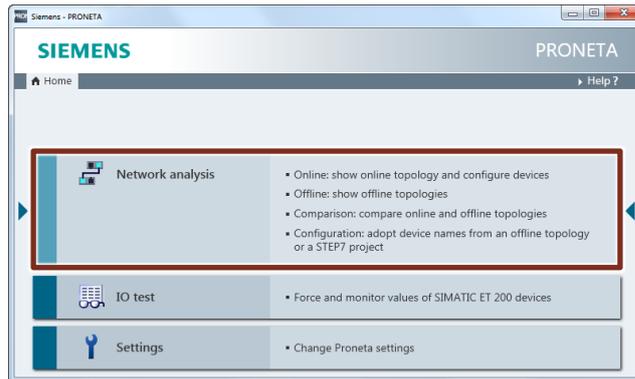
4.2.2 Assigning the IP addresses and the device name using PRONETA

Alternatively to SIMATIC Automation Tool, you can also use the network analysis and configuration tool PRONETA to assign IP addresses and device names.

Proceed as follows:

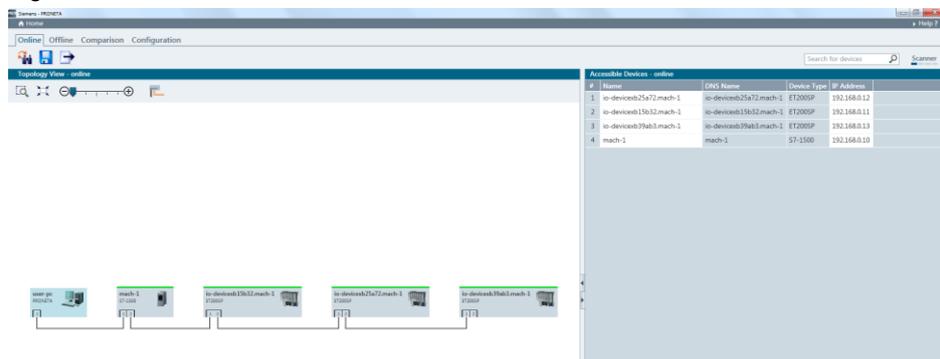
1. Connect your PC / programming unit to the PROFINET network.
2. Start PRONETA and click "Network analysis" to open the network analysis.

Figure 4-13



3. On the left side you are seeing now the online topology (Topology view) and on the right side an overview of the accessible users (device table) of your PROFINET network.

Figure 4-14



4. Change the IP address and the device names of the IO controller by modifying the related entries in the device table.

Figure 4-15

Accessible Devices - online				
#	Name	DNS Name	Device Type	IP Address
1	io-devicexb25a72.mach-1	io-devicexb25a72.mach-1	ET200SP	192.168.0.12
2	io-devicexb15b32.mach-1	io-devicexb15b32.mach-1	ET200SP	192.168.0.11
3	io-devicexb39ab3.mach-1	io-devicexb39ab3.mach-1	ET200SP	192.168.0.13
4	mach-1	mach-1	S7-1500	192.168.0.10

Result

The IP address and the device name are assigned to the IO controllers. Thus, the IO controller may now assign an IP address and a device name to the IO devices attributed to the PROFINET IO system.

5 Links & Literature

Table 5-1

	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Download page of the entry https://support.industry.siemens.com/cs/ww/en/view/29430270
\3\	Manual PROFINET with STEP 7 V14 https://support.industry.siemens.com/cs/ww/de/view/109742272
\4\	Entry regarding the SIMATIC Automation Tool https://support.industry.siemens.com/cs/ww/de/view/98161300
\5\	PRONETA download page https://support.industry.siemens.com/cs/ww/en/view/67460624

6 History

Table 6-1

Version	Date	Revision
V1.0	06/2016	First version
V2.0	03/2017	Upgrade to STEP 7 V14