Distributed synchronous operation and isochronous mode via PROFINET IRT

SIMOTION & SINAMICS

Application example • September 2013

Applikationen & Tools

Answers for industry.

SIEMENS

Siemens Industry Online Support

This article is taken from the Siemens Industry Online Support. The following link takes you directly to the download page of this document:

http://support.automation.siemens.com/WW/view/en/38486079

CAUTION

The functions and solutions described in this article confine themselves to the realization of the automation task predominantly. Please take into account furthermore that corresponding protective measures have to be taken up in the context of Industrial Security when connecting your equipment to other parts of the plant, the enterprise network or the Internet. Further information can be found under the Item-ID 50203404.

http://support.automation.siemens.com/WW/view/en/50203404.

If you have any questions concerning this document please e-mail us to the following address:

profinet.team.motioncontrol.i-dt@siemens.com

You can also actively use our Technical Forum from the Siemens Industry Online Support regarding this subject. Add your questions, suggestions and problems and discuss them together in our strong forum community:

http://www.siemens.de/forum-applikationen

S

SIMOTION & SINAMICS PROFINET IRT
- Direct data exchange broadcast
- IRT I-Device

Application example

Warranty and liability

Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These application examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. 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 without prior notice.
If there are any deviations between the recommendations provided in these application examples 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 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 a condition which goes to the root of the contract ("wesentliche Vertragspflichten"). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Table of contents

1 Task

1.1 Overview

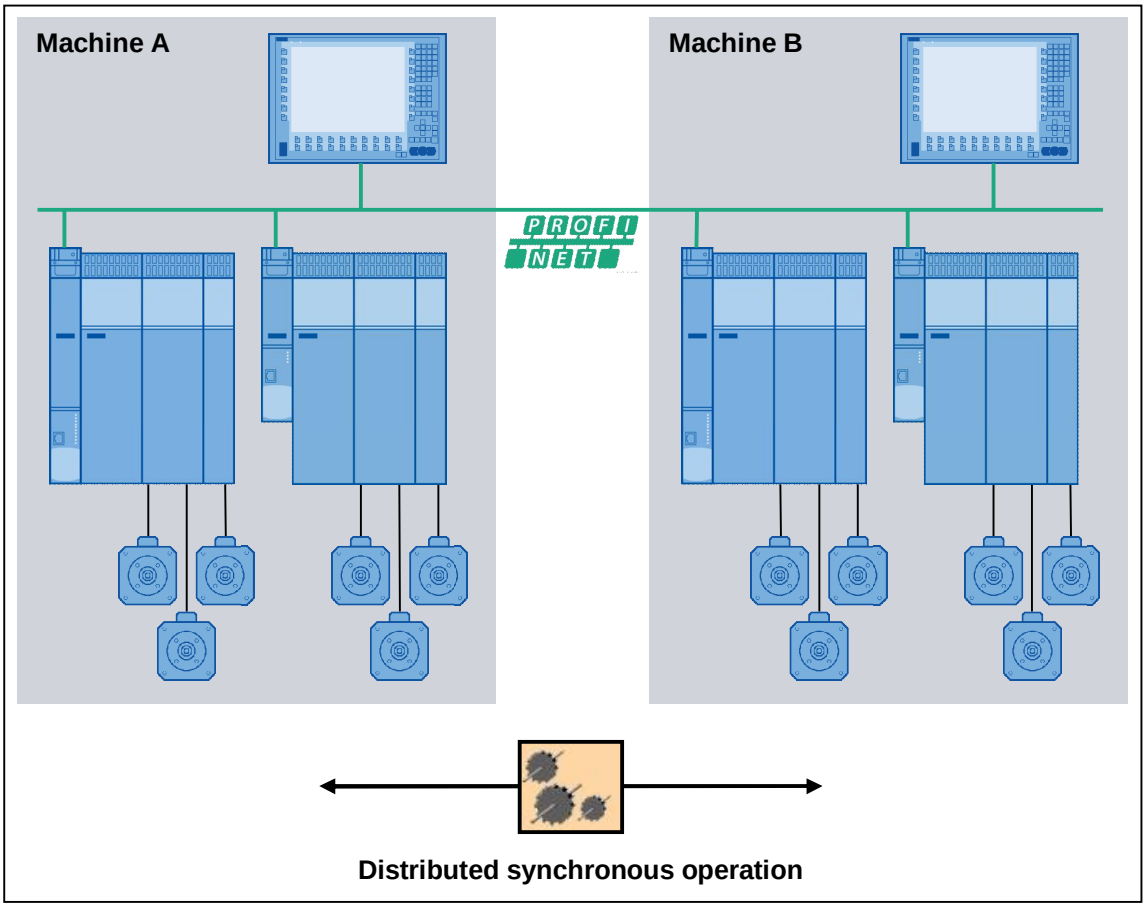
Introduction

Modular machine concepts have several controllers and drives. These components have to operate depending on each other. It means all controllers and drives have to exchange data.

Overview of the automation task

The figure below provides an overview of the automation task.

Figure 1-1



A virtual master in **machine A** calculates e.g. the command position for the slave axes of **machine B**. It means the master axis and following axes are on different controllers. This functionality is also called “distributed synchronous operation“. Therefore an isochronous real-time communication between controller and drives is needed.

Note More information about “distributed synchronous operation” are available in the SIMOTION document “Gear_TechnologyFct.pdf”.
http://support.automation.siemens.com/WW/view/en/61055960

2 Solution

2.1 Solution overview

PROFINET IRT + isochronous mode (High Performance = RTC3, Real Time Class) is needed for an isochronous communication between controllers and drives.

This type of communication is required by SIMOTION CPUs for distributed synchronous operation. Following PROFINET IO functions can be used:

- **Direct data exchange broadcast** or
- **IRT I-Device.**

2.1.1 Direct data exchange broadcast

Direct data exchange broadcast allows isochronous communication of inputs and outputs between more than two SIMOTION CPUs.

For distributed synchronous operation the input and output addresses (distributed synchronous data) are automatically set by the engineering software SIMOTION SCOUT. It is not allowed to modify the addresses.

All participated SIMOTION CPUs must be in the same SIMOTION project!

2.1.2 IRT I-Device

IRT-I-Device transmits the input and output data also in synchronous mode. The difference is that in this case the SIMOTION I-Device can communicate only to one PROFINET IO-Controller.

For distributed synchronous operation the input and output addresses (distributed synchronous data) for the I-Device must be set by the user.

Instead of direct data exchange broadcast, where you have to configure both SIMOTION CPUs in the same project, you have the possibility to configure the IRT I-Device communication via GSD file and using two different SIMOTION projects. In addition you have the possibility to create modular machine concepts by activating or deactivating I-Devices during runtime without any bus failure.

CAUTION	**The IRT I-Device of a SIMOTION CPU with FW V4.3 cannot be operated in isochronous mode when a SIMOTION CPU with FW < V4.2 is used as higher-level I/O controller!**

NOTE	More information about modular machines you will find in the SIMOTION Document “Modular_Machine_Concepts.pdf”. http://support.automation.siemens.com/WW/view/en/61056223

Display

The following figure displays the most important components of the solution:

Figure 2-1

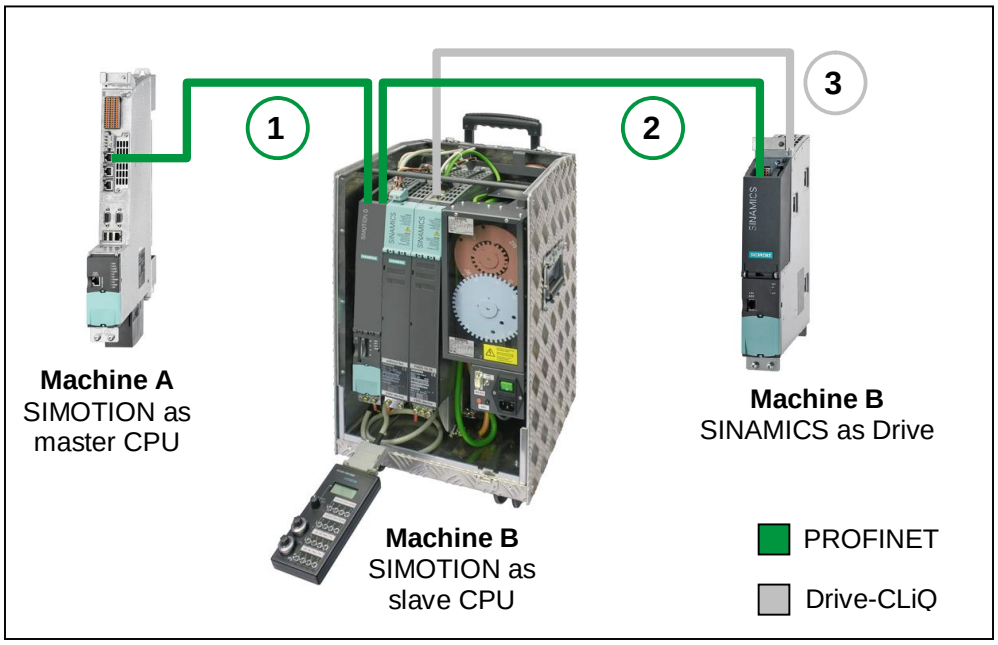


Table 2-1

No.	Connection
1.	PROFINET connection between master and slave CPU (IRT communication)
2.	PROFINET connection between slave CPU and the drive (IRT communication)
3.	Drive-CLiQ connection between SINAMICS control unit and SINAMICS motor module

Delimitation

This application does not include a description of

- the SINAMICS drives and
- the SIMOTION CPU.

Knowledge of these topics is assumed.

2.2 Required Hardware and Software Components

The application was generated with the following components:

Hardware components

Table 2-2

Component	No.	MLFB / order number	Note
Training unit SIMOTION D435	1	6ZB2470-0AE00	V4.3 SP1
SIMOTION D455-2	1	6AU1455-2AD00-0AA0	V4.3 SP1
CBE30 (PN option board for D4x5)	1	6FC5312-0FA00-0AA0	
CU320-2 PN	1	6SL3040-1MA01-0AA0	V4.5.0.6

CAUTION **Motor Modules Booksize (SMM/DMM 6SL31*) with order number smaller than 3 at the end are not released for use with SINAMICS V4.3 or higher.**

CAUTION **SIMOTION D435 CPU with order number lower than 1 at the end is not released for use with PROFINET communication.**

Standard software components

Table 2-3

Component	Count	Order number	Comment
STEP7	1	6ES7810-4CC08-0YA5	V5.5 SP2 HF1
SIMOTION SCOUT	1	6AU1810-0BA42-…	V4.3 SP1 HF1

Sample files and projects

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

- **38486079_DistributedSyncOperation_PN_IRT_V2_2_en.pdf**
  this document.
- **38486079_DistributedSyncOperation_PN_IRT_V2_2.zip**
  > Projects > DistributedSyncOperation_PN_IRT_CCQ_V2_2.zip
  SIMOTION Project with the PROFINET function
  direct data exchange broadcast
- **38486079_DistributedSyncOperation_PN_IRT_V2_2.zip**
  > Projects > DistributedSyncOperation_PN_IRT_I-Device_V2_2.zip
  SIMOTION Project with the PROFINET function
  IRT I-Device

3 Basics

3.1 PROFINET Communication

In addition to the MAC address and IP address, PROFINET uses additionally a device name to identify the PROFINET devices. This device name must be unique within the PROFINET network.

3.1.1 Device Name

During the commissioning phase a device name is assigned to each PROFINET device by the engineering system (HW Config, NetPro, STARTER, SCOUT, Primary Setup Tool). The device name can be assigned by different ways:

- IO-Controller
  - Engineering Software (HW Config, NetPro, STARTER, SCOUT, Primary Setup Tool)
  - By downloading the HW Config
  - By user program (system function `_setNameOfStation()` for SIMOTION)
- IO-Device
  - Engineering Software (HW Config, NetPro, STARTER, SCOUT, Primary Setup Tool)
  - From IO-Controller via topology-based initialization

The device name will be stored in the device (on MMC or CF card). When replacing this device (e.g. in case of malfunction), this device must be initiated using the device name of the replaced device. For this step more possibilities are available:

- By plugging the MMC or CF card (if available)
- Engineering software (HW Config, NetPro, STARTER, SCOUT, Primary Setup Tool)
- Topology-based initialization by the IO-Controller itself. For this the PROFINET interface must be in factory settings.

It means the new device can assume the function of the replaced device without changing the configuration.

3.1.2 Assign Device name via HW Config

Open HW Config and select the PROFINET IO-System. Click in menu under „PLC > Ethernet" onto "Assign Device Name…". The dialog assign device name will appear. All configured device names are in dropdown box listed. All recognized PROFINET devices via Network interface are shown under available devices. IO-Controllers are not shown here, because they will get the device name by downloading the HW Config explicit.

Figure 3-1

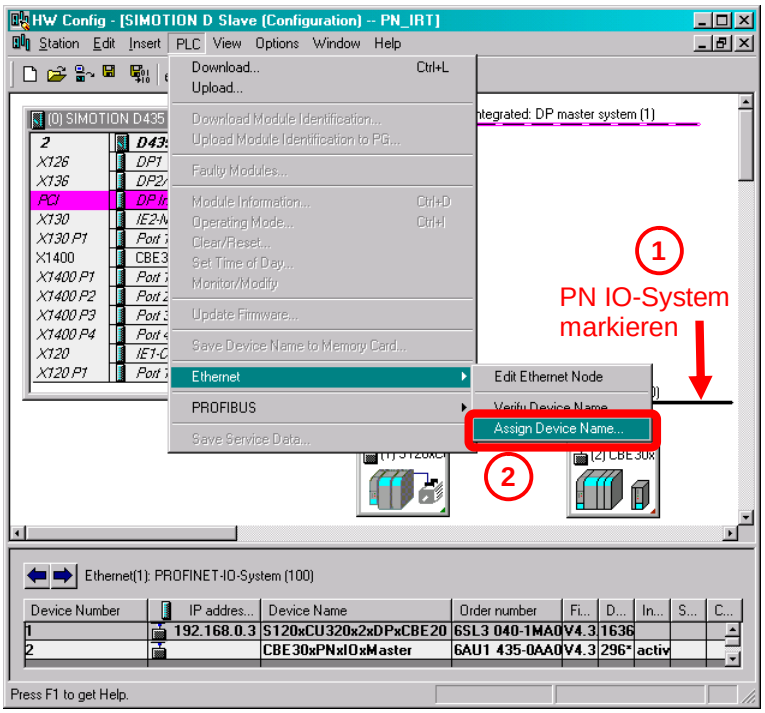
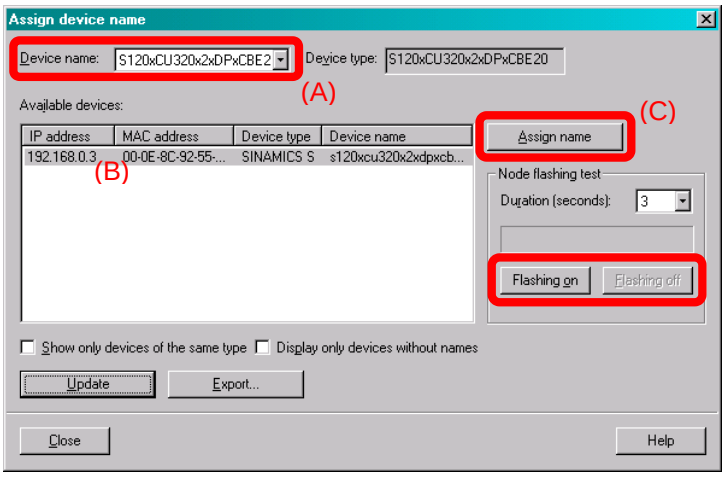


Figure 3-2



Select the configured device name (A) and select the corresponding device (B) and click on “Assign name” (C). The device name will be transferred to the device.

If the device identification is not clear you can activate the flashing of a specified LED. Select the device and click on “Flashing on” to activate the function. Depending on the device type following LED will start flashing:

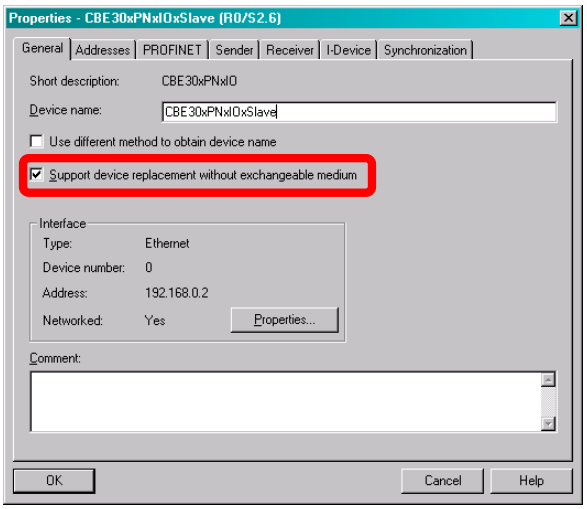
Table 3-1

Device	LED
SIMOTION	SF-LED
SINAMICS S120	RDY-LED

3.1.3 Topology-based initialization

The device name can be assigned by the PROFINET IO-Controller itself. With the checkbox “Support device replacement without exchangeable medium” the PROFINET feature topology-based initialization is activated. This feature is activated by default.

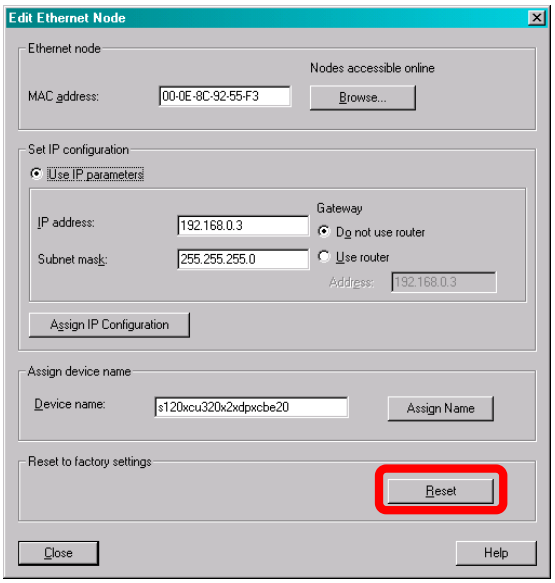
Figure 3-3



This properties window will open by double clicking on the PN interface of the IO-Controller in HW Config.

Please observe that **the PN interface of IO-Device must be in factory settings** to support this function (in this state the interface has the IP address = 0.0.0.0 and an empty device name = „“). To reset the PROFINET interface to factory settings open HW Config and click on „PLC > Ethernet > Edit Ethernet Node > Reset“ (see following figure).

Figure 3-4



3.1.4 Device name assignment rules

The device name has to follow the rules of DNS (Domain Name System). Following possibilities for DNS names are given:

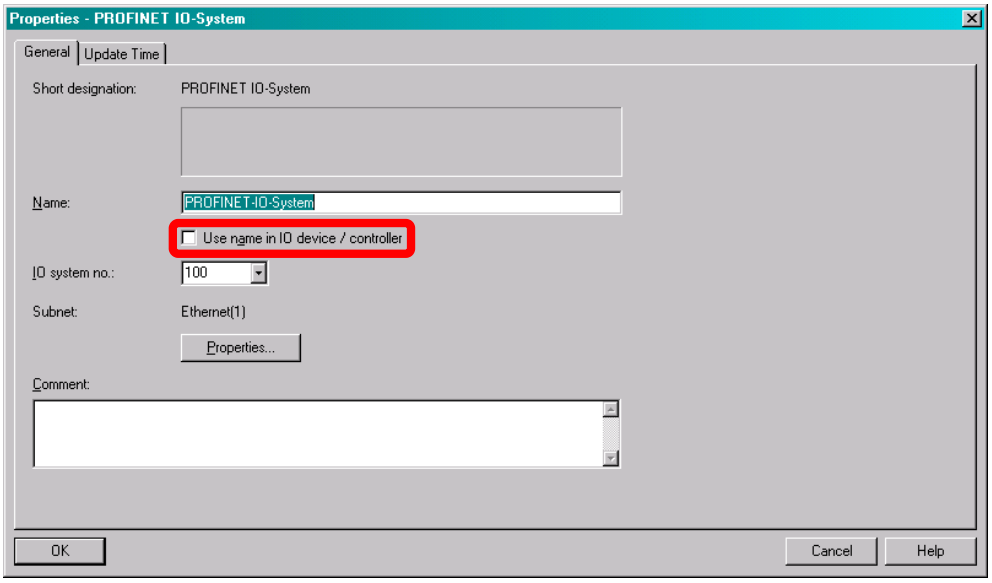
- Letters (a..z),
- Numbers (0..9) and the signs
- Minus (-) and
- Dot (.) are allowed.

The dot divides the device name into labels. The device name can include more labels, for example:

<CPU name>.<Interface number >.<Name of IO-System>…

- *<Interface number>*, if the device has more than one PN interfaces available (e.g. “X150”)
- *<Name of IO-System>*, optional and configured by HW Config (see Figure 3-5).
- Each label must start with a letter and doesn’t end with minus or a dot.
- The maximum length of one label is 63 characters.

Figure 3-5



Observe also following additional rules:

- The maximum total length is 127 characters (incl. minus and dot)
- The device name may only include lower case. On the device upper case are replaced through lower case.
- Do not use umlauts (ä, ö, ü)
- Do not use special characters ! " § $ % & / ( ) = ? * ' _ : ; > < , # + | ~ \ }] [{
- Do not use blanks
- The device name does not start with "port-xyz" (x, y, z = 0..9).
- **Do not use the minus sign on SIMOTION controllers. Up from SIMOTION SCOUT Version 4.3 this limitation is raised.**

3.1.5 IP addresses

PROFINET uses IP addresses for establishing the PROFINET IO communication and for NRT (Non Real Time Communication e. g. TCP, UDP, S7 communication).

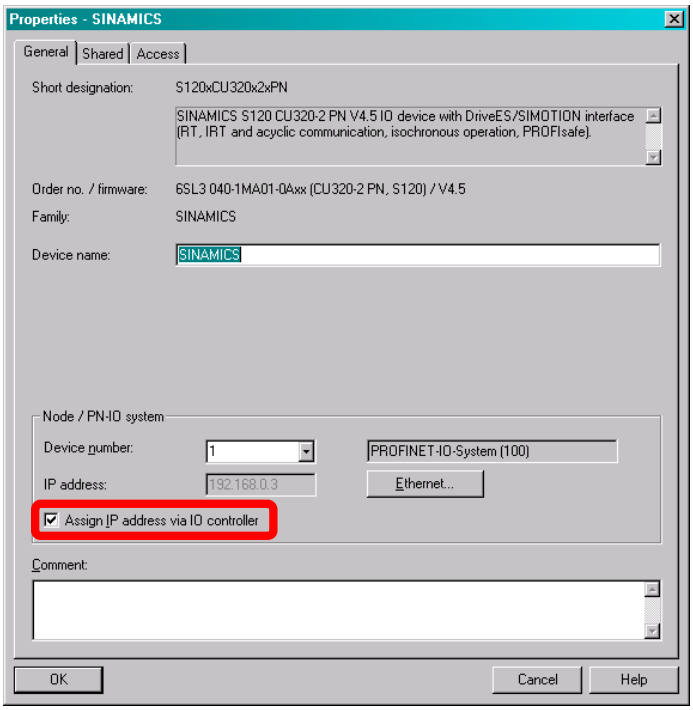
The IP address is also needed to go online to the device. It is recommenced to assign an IP address to each PROFINET IO-Device during startup of a project commissions. To do this you have more possibilities:

- IO-Controller
  - Engineering software (HW Config, NetPro, STARTER, SCOUT, Primary Setup Tool)
  - Download of HW Config, please observe the current active IP address of the device!
- IO-Device
  - Engineering software (HW Config, NetPro, STARTER, SCOUT, Primary Setup Tool)

In a PROFINET IO-System it is possible to receive the IP address from the IO-Controller. This feature is activated by default.
For this a correct working PROFINET connection between IO-Controller and IO-Device is required and the active device name must be the same as in the HW Config. Figure 3-6 shows the configuration of a SINAMICS S120. The device with device name “SINAMICS” will get the IP address 192.168.0.3 and the same subnet mask as the IO-Controller.

If the device doesn’t receive the IP information from IO-Controller it is maybe not possible to go online with the engineering software (e. g. Starter or SCOUT). In this case assign a fixed IP to the device.

Figure 3-6

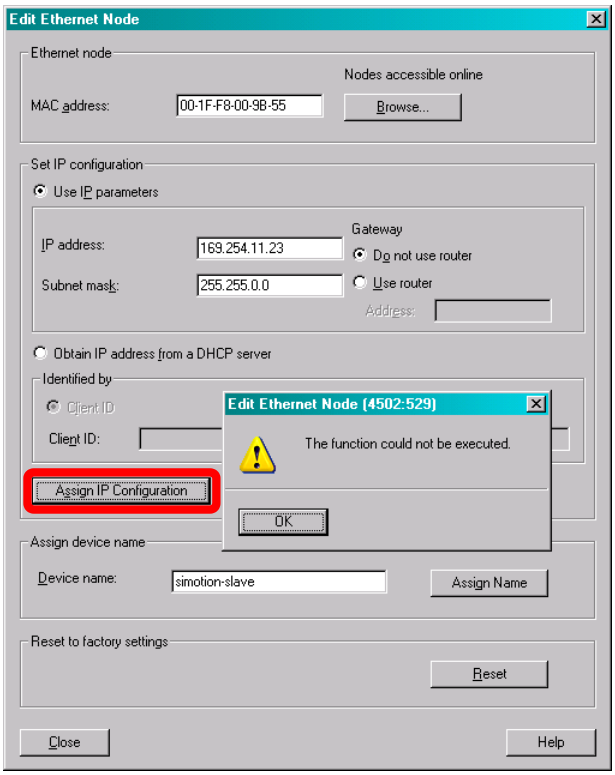


The received IP address from IO-Controller is just a temporary address (till next power OFF/ON of the device). The received address is higher prior as a fixed set IP address via engineering software.
After power ON the IP address must be received from IO-Controller first. If the address will not receive, the IP address set by the engineering system or the default IP address 0.0.0.0 (default setting of the PN interface) will be active.
An IP address assigned by any engineering software is permanently saved.

If you disable the function „Assign IP address via IO-Controller" the fixed IP address will be used. The IO-Controller will not assign the configured IP address.

The IP address in the device must be unique. Please observe that the IP addresses used by PN interfaces (e.g. X150) and the IP addresses of the standard Ethernet interfaces (e.g. X127) must be in different IP subnets. For example: If you assign an IP address 169.254.11.23 to the PN interface X150 so the device will report an error if the standard Ethernet interface X127 is still assigned to 169.254.11.22 (default IP address of X127).

Figure 3-7



3.1.6 Send clock for IRT communication

- The send clock for the IRT communication can be configured to a value ranging between 250 µsec and 4.0 msec.
  - Up from firmware V4.5 of SINAMICS S120 the minimum send clock time of the onboard PN interface is 250 µsec.
  - By using the CBE20 the minimum send clock time is 500 µsec.
  - Please refer also to the SIMOTION documentation “Communication.pdf” Chapter 4.2.2.3 “Overview of the possible bus cycle clocks”.
- Up from SIMOTION V4.1 SP1 the send clock of the isochronous PROFIBUS interfaces must be equal to the servo cycle clock. This also applies to PROFIBUS Integrated (SINAMICS Intergrated). Maybe the servo cycle clock or PROFIBUS send clock must be a multiple of the PROFINET IO send clock.
- Isochronous applications (e.g. ServoTask) are synchronized to the send clock or a multiple of the send clock. The cycle reduction is configured in the SIMOTION SCOUT under “Set system cycle clocks…“.

3.1.7 Isochronous mode

Isochronous mode means that the application (e.g. ServoTask of SIMOTION or OB6X of SIMATIC) is synchronized with the PROFINET IRT send clock. For this PROFINET IRT (High Performance = RTC3) is absolutely needed.

With a SIMOTION controller the synchronization of the application with the communication send clock will be handled as follow:

- As Sync-Master the synchronization will be handled automatically, but if the Sync-Master is configured as IRT I-Device the synchronization must be done manually like an Sync-Slave
- A Sync-Slave must always be synchronized with the send clock. This has to be done in the StartupTask with following system function:

```
i32RetValue := _enableDpInterfaceSynchronizationMode(
  dpInterfaceSyncMode := AUTOMATIC_INTERFACE_SYNCHRONIZATION
  );
```

If a SIMOTION controller is synchronized it can be checked by the LED SY (X150) respectively by the green LED on the CBE30 or CBE30-2.

- Green flashing = not synchronized
- Green continuous = synchronized

A check of the system variable `stateOfDpInterfaceSynchronization = DP_INTERFACES_SYNCHRONIZED` is also possible.

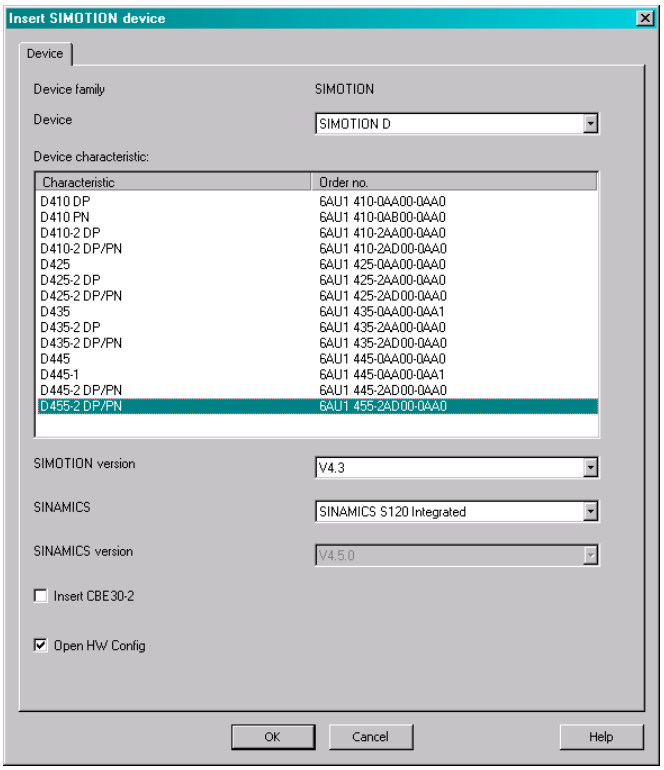
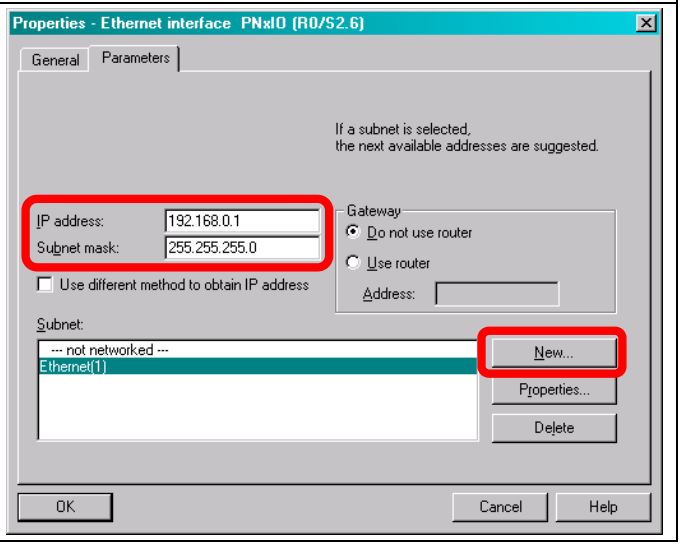
If a SINAMCIS device is synchronized you can check the LED PN (X150) respectively by the green LED on the CBE20.

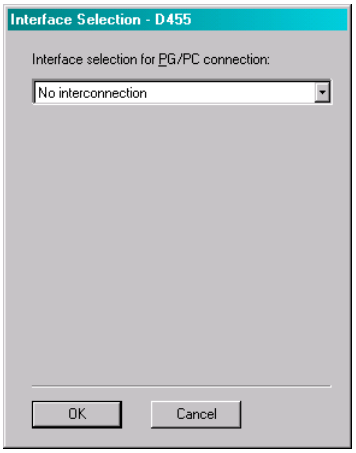
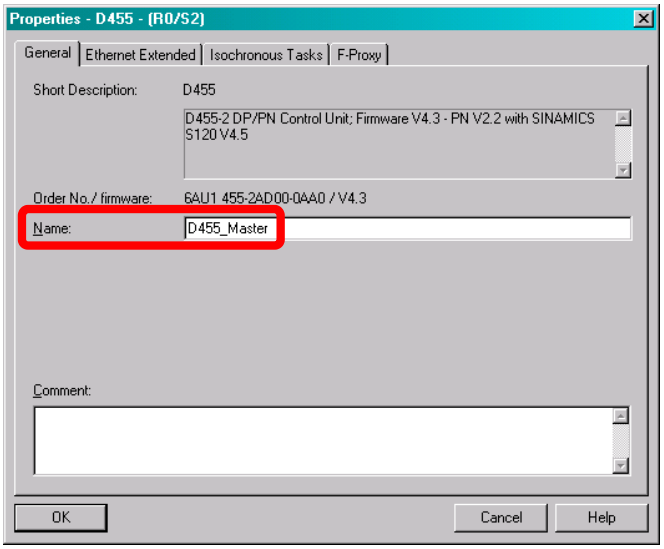
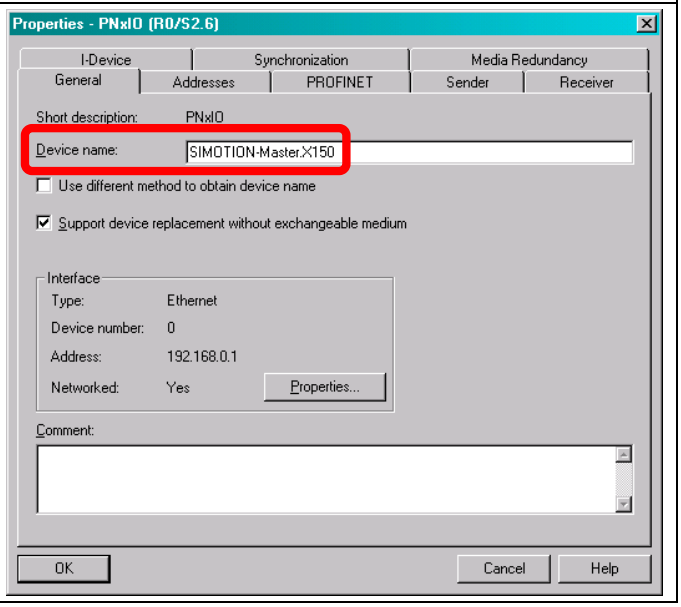
- Green flashing = not synchronized
- Green continuous = synchronized

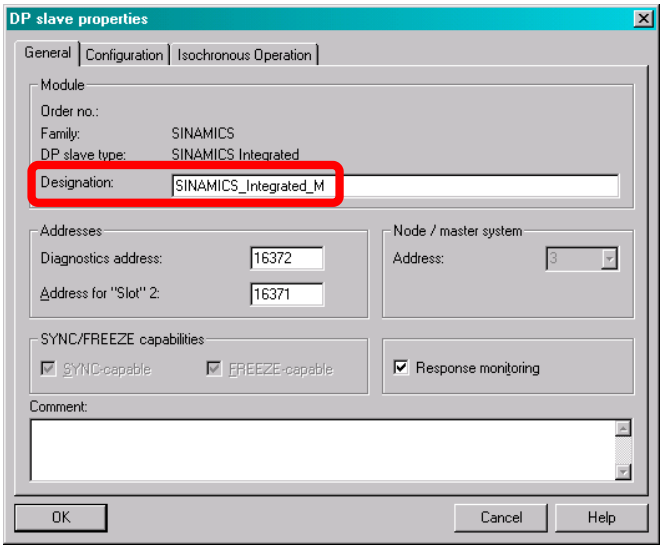
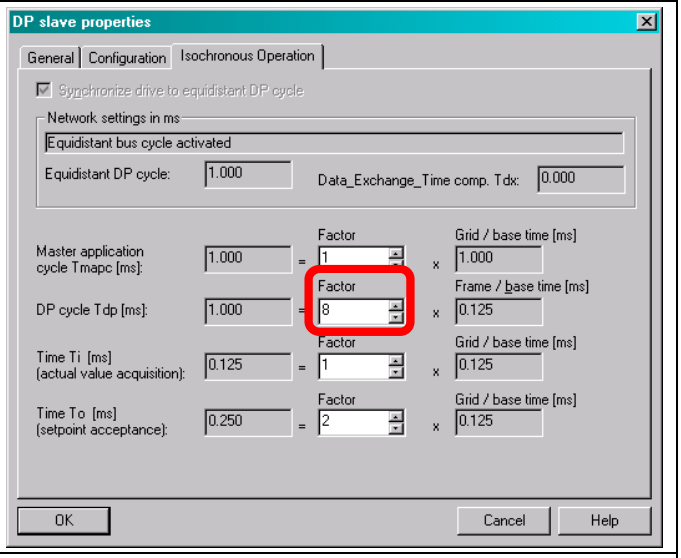
4 Configuration

4.1 HW Config of the master CPU

Table 4-1

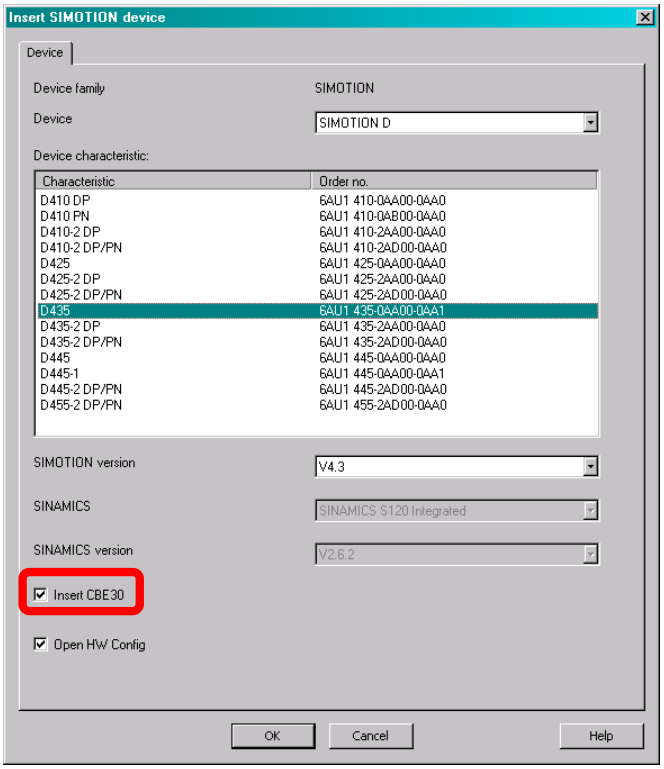
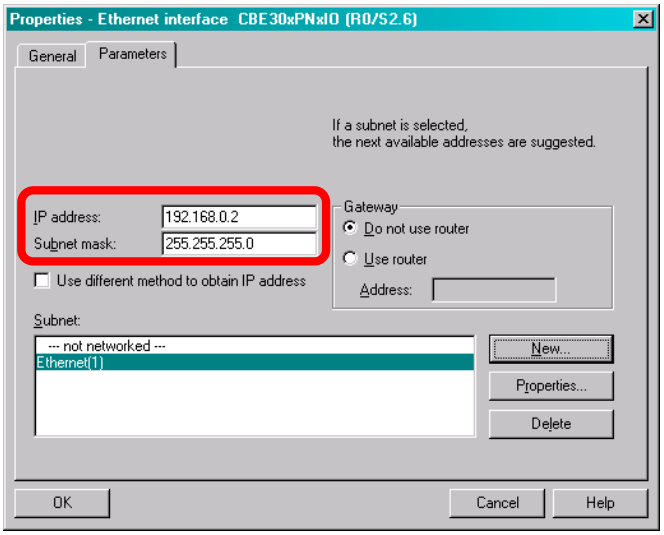
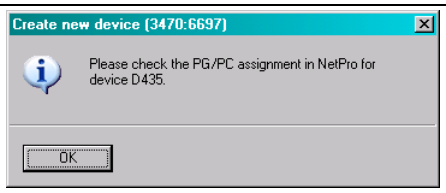
No.	Instruction	
1.	Start SIMOTION SCOUT and create a new project.	
2.	Add a new SIMOTION device and open the HW Config. In the sample project a SIMOTION D455-2 DP/PN V4.3 is used as master CPU (without CBE30-2)	
3.	Set the IP address of the SIMOTION PN interface. In the sample project following addresses are used: IP address: 192.168. 0 .1 Subnet mask: 255.255.255.0 If necessary create a new subnet.	

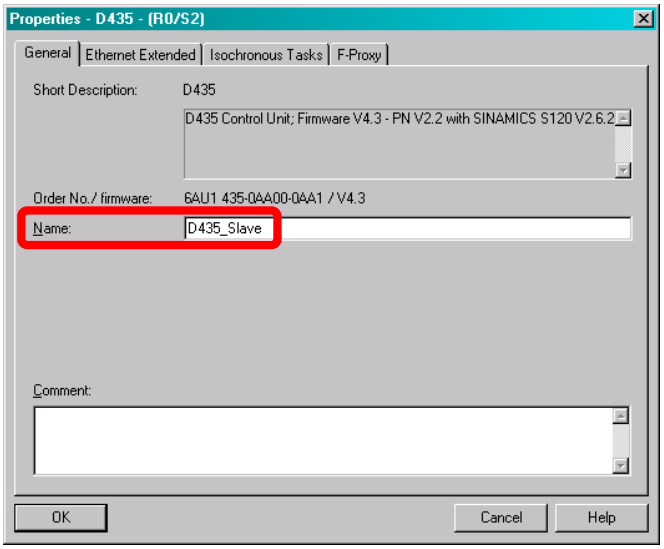
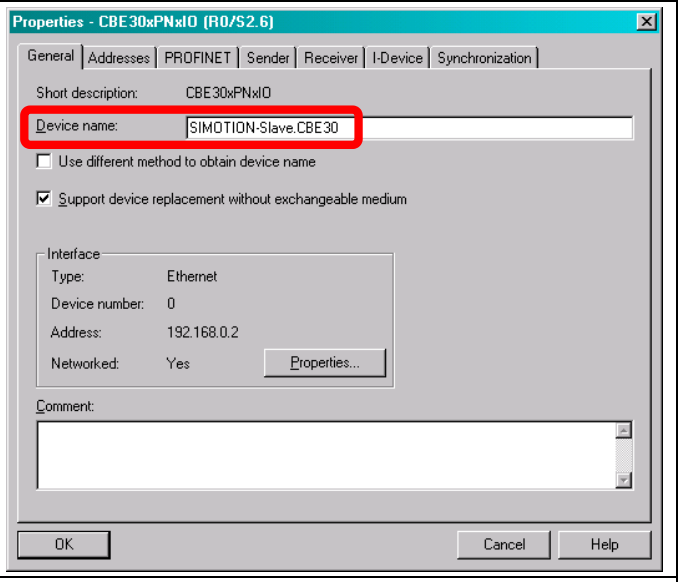
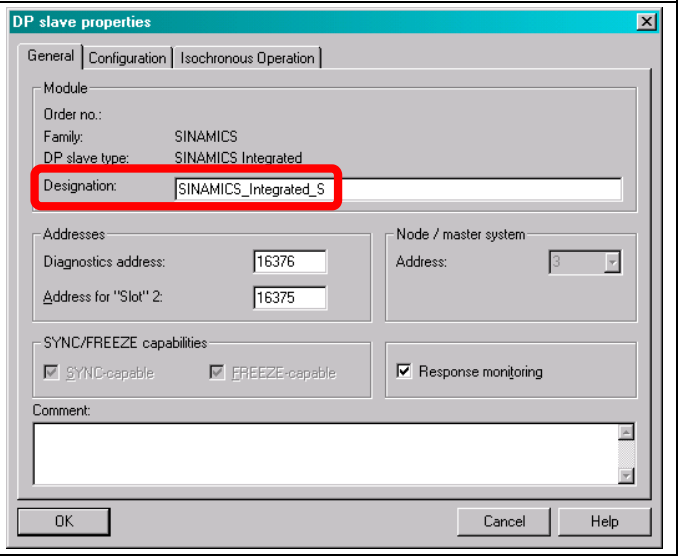
4.	A dialog will open to select a SIMOTION interface, where the engineering system (PG/PC) is connected. In the sample project no interface will be selected first.	
5.	By double click on the CPU (Slot 2) the properties dialog will open. There you can change the name of the CPU. This name is not the PROFINET IO device name (see instruction 6). In the sample project the name of the master CPU is „D455_Master“.	
6.	By double click on the PROFINET IO interface of the SIMOTION CPU the properties dialog will open. There you have to modify the device name. Device name: „SIMOTION-Master.X150“ The Setting „Support device replacement without exchangeable medium” activates the topology-based initialization.	

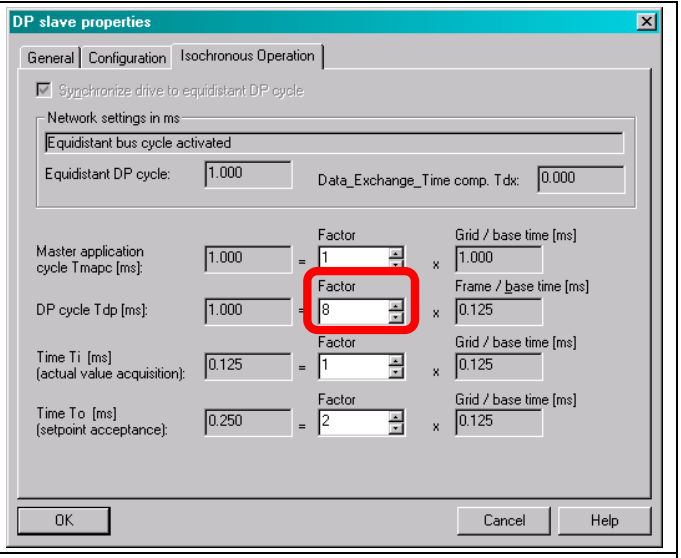
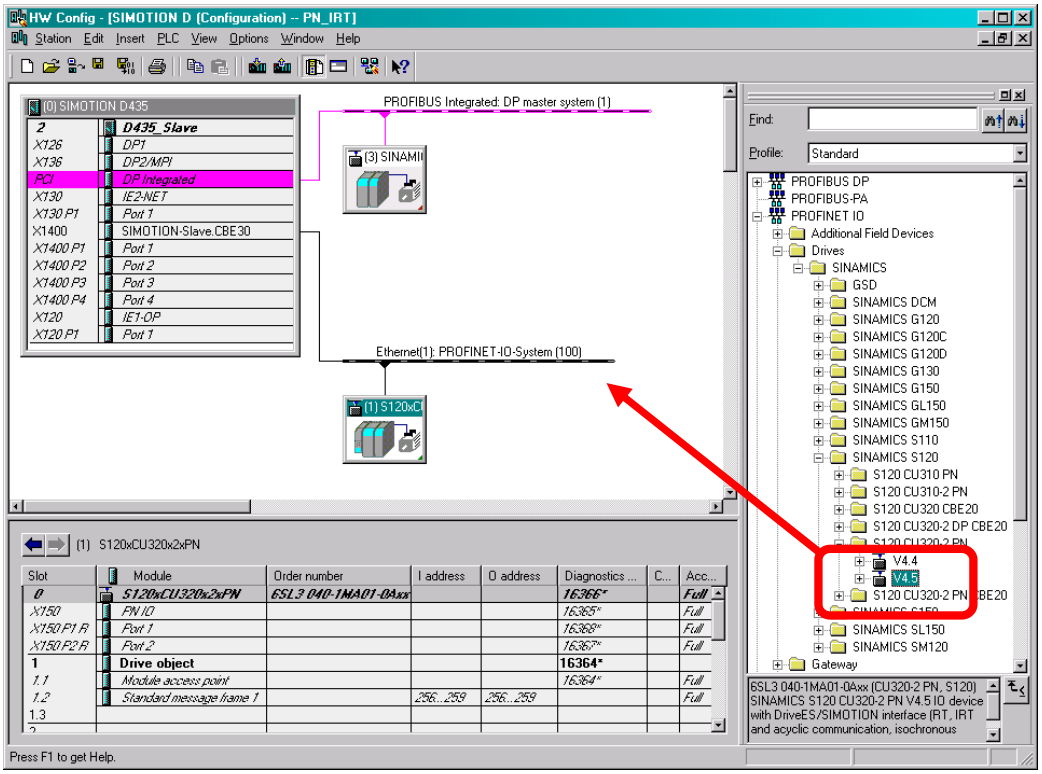
7.	A double click on the SINAMICS_Integrated will open the properties dialog. If required change in the tab “General” the designation of the SINAMICS_Integrated. In the sample project the designation for the master CPU is “SINAMICS_Integrated_M“.	
8.	Change to the tab “Isochronous Operation”. Modify the DP cycle to match the cycle time of the servo cycle clock. In the sample project the servo cycle clock = 1 msec.	
9.	Save and compile HW Config.	

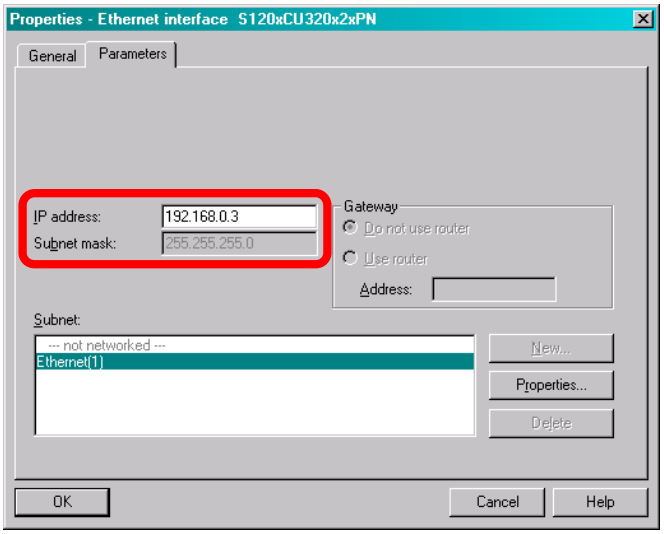
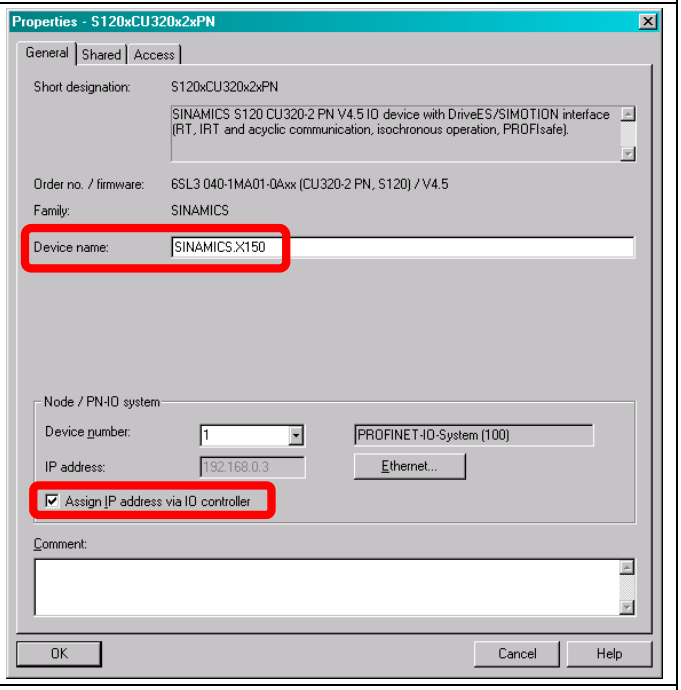
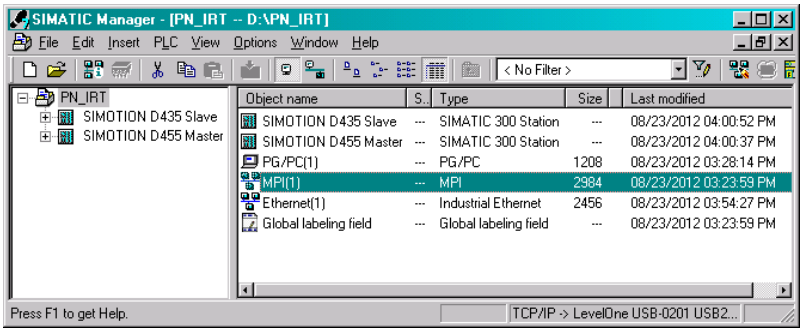
4.2 HW Config of the slave CPU

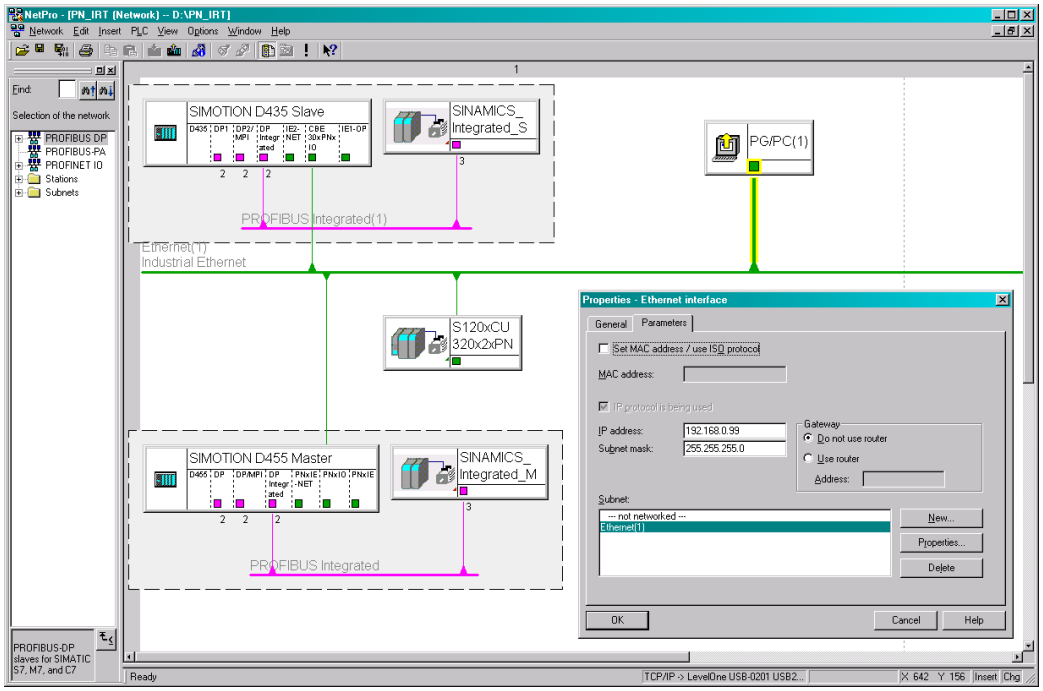
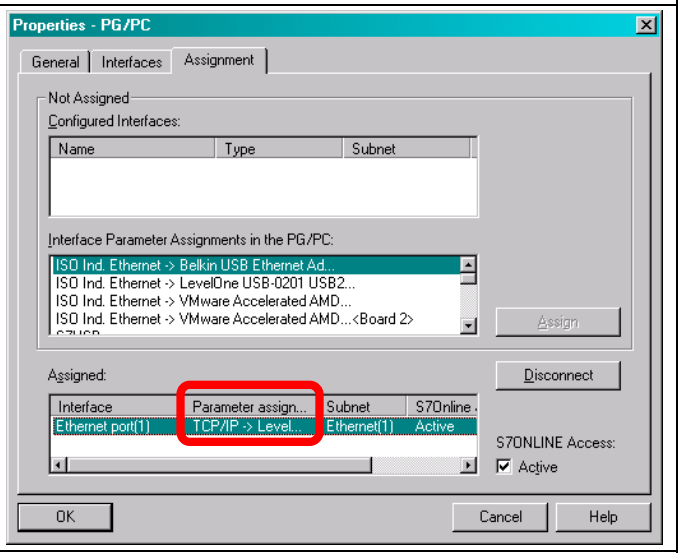
Table 4-2

No.	Instruction	
1.	Change to the SIMOTION SCOUT.	
2.	Add an additional SIMOTION device into the project and open HW Config. In the sample project the training unit with a SIMOTION D435 V4.3 + CBE30 as PN interface is used.	
3.	Select the already configured subnet. Set the IP address of the SIMOTION PN interface. In the sample project following address will be used: IP address: 192.168. 0 .2 Subnet mask: 255.255.255.0	
4.	A message box reminds that you have to check the PG/PC settings in NetPro.	

5.	By double click on the CPU (Slot 2) the properties dialog will open. There you can change the name of the CPU. This name is not the PROFINET IO device name (see instruction 6). In the sample project the name of the salve CPU is „D435_Slave“.	
6.	By double click on the PROFINET IO interface of the SIMOTION CPU the properties dialog will open. There you have to modify the device name. Device name: „SIMOTION-Slave.CBE30“ The Setting „Support device replacement without exchangeable medium” activates the topology-based initialization.	
7.	A double click on the SINAMICS_Integrated will open the properties dialog. If required change in the tab “General” the designation of the SINAMICS_Integrated. In the sample project the designation is changed to „SINAMICS_Integrated_S“ for the slave CPU	

8.	Change to the tab “Isochronous Operation”. Modify the DP cycle to match the servo cycle clock. In the sample project the servo cycle clock = 1 msec (= min. servo cycle clock of the SIMOTION D435).	
9.	Select from the hardware catalogue a SINAMICS drive with PN interface (e.g. SINAMICS S120 CU320-2PN V4.5).	

10.	Set the IP address of the SINAMICS PN interface (X150). In the sample project following address will be used: IP address: 192.168. 0 .3 Subnet mask: 255.255.255.0	
11.	By double click on the PROFINET interface of the SIMOTION CPU the properties dialog will open. There you have to modify the device name. With the setting „Assign IP address via IO-Controller“ the IO-Controller will transfer the configured IP address to the IO-Device with the device name “SINAMCIS.X150”. In the sample project the IO-Device „SINAMICS.X150“ has the IP address 192.168.0.3.	
12.	Save and compile HW Config.	
13.	If required modify with the SIMATIC Manager the name of the SIMOTION CPUs and delete the not required MPI Network. 	

14.	Modify in NetPro the setting for the PG/PC object to get online to the devices for downloading.  In the sample project the engineering system (PG/PC) is using the IP address 192.168.0.99	
15.	To go online to the SINAMICS_Integrated the correct Ethernet interface “TCP/IP → …” must be assigned.	
16.	Save and compile NetPro	

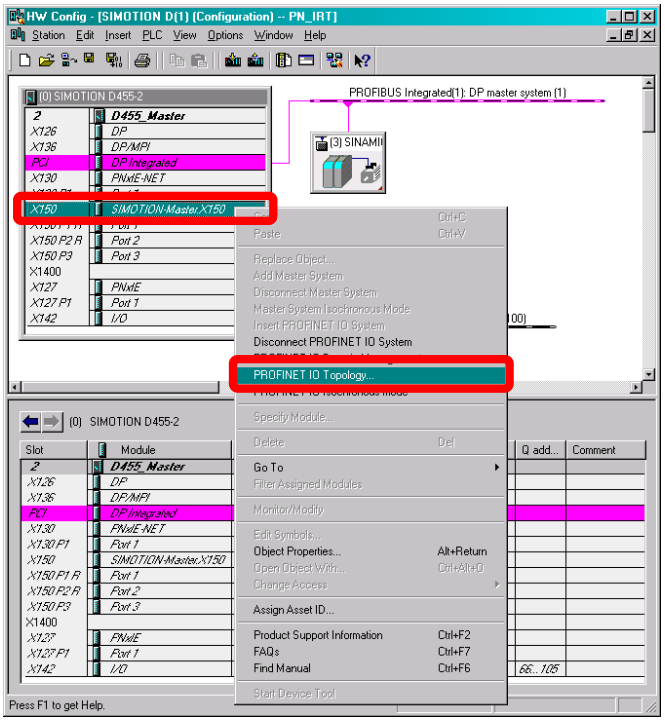
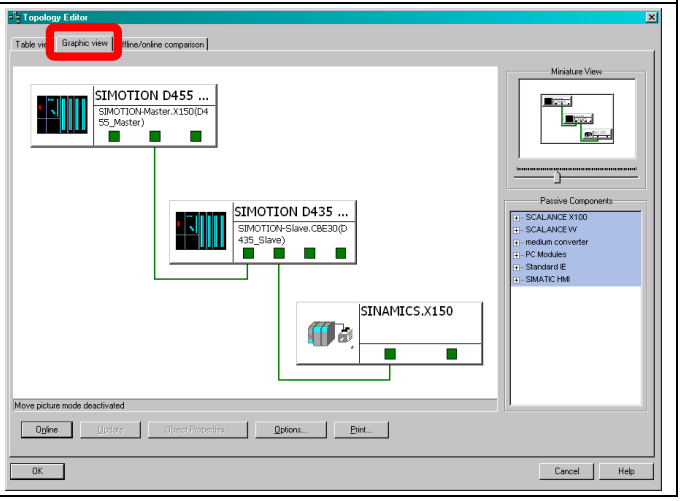
4.3 Configuration of PROFINET IRT

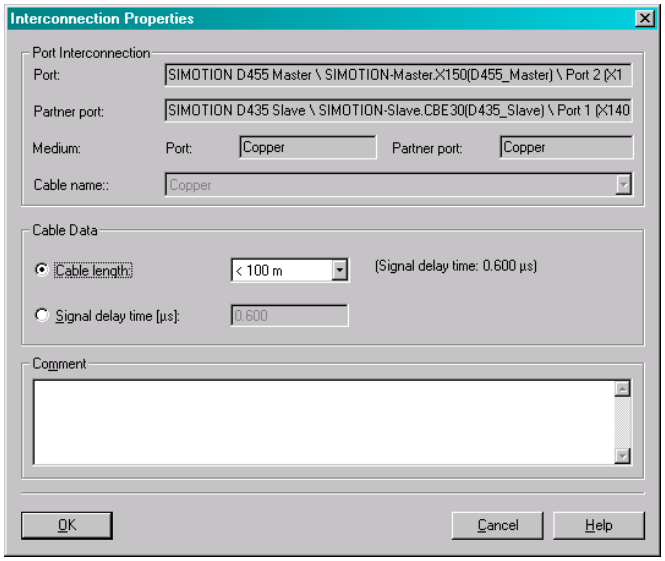
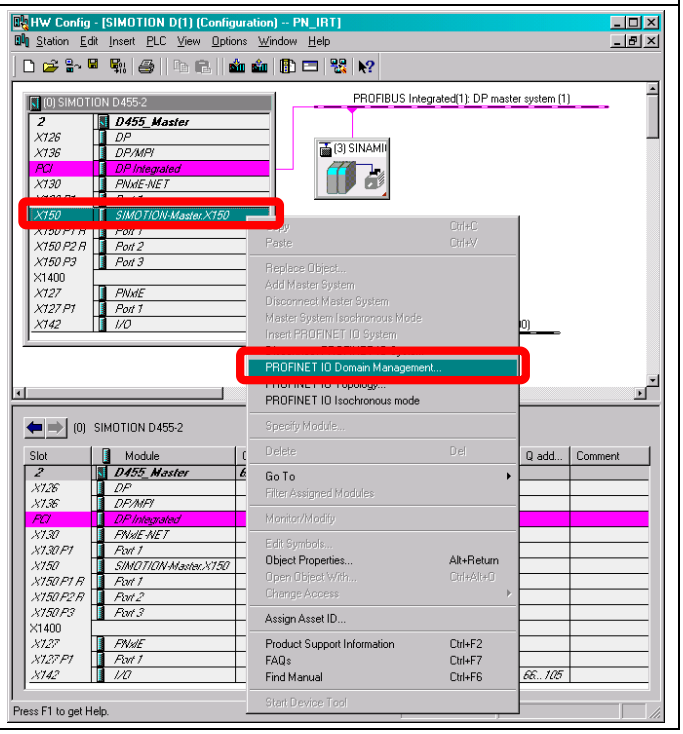
For using PROFINET IRT (High Performance = RTC3)

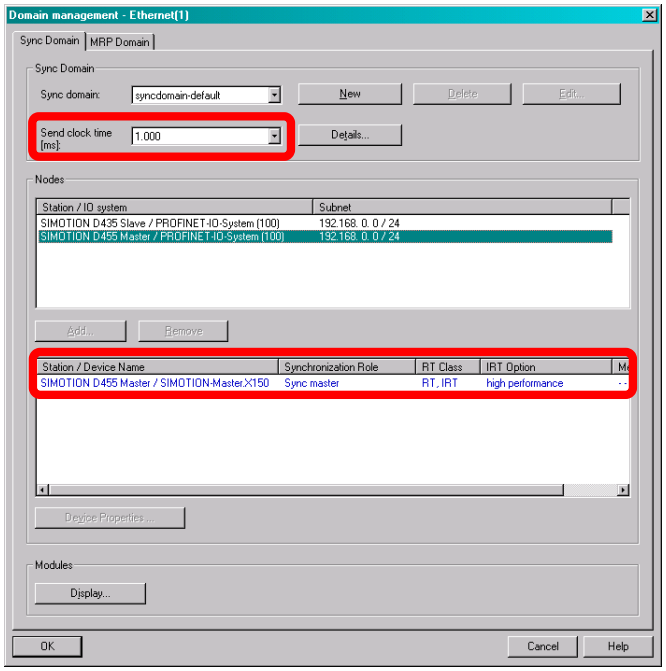
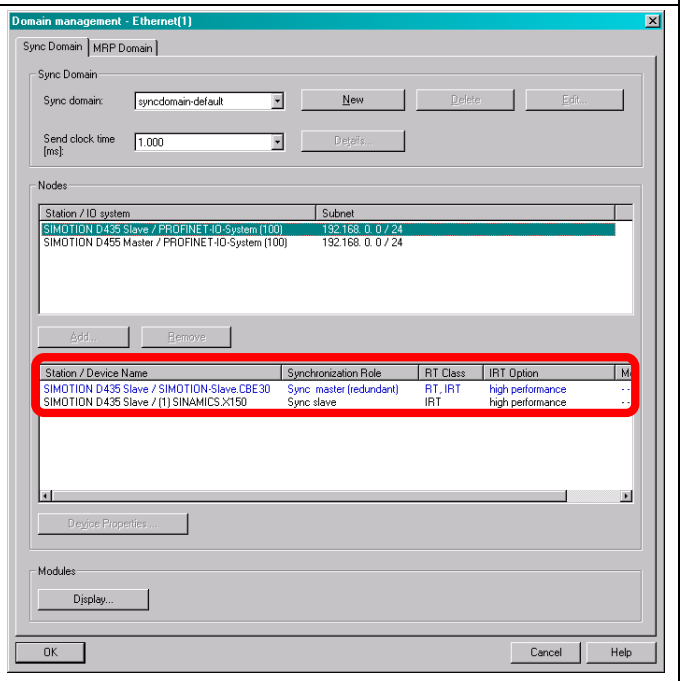
- the PROFINET topology and
- the Sync-Domain (Sync-Master and Sync-Slave)

have to be configured.

Table 4-3

No.	Instruction	
1.	Open HW Config, click with the right mouse button on the PN interface (e.g. X150) and select „PROFINET IO Topology…“.	
2.	Change to the graphic view to get a comfortable possibility to edit the PROFINET topology. If the engineering system is connected to the PN network the current topology can be compared with the configured topology (tab „Offline/online comparison“). If the current and the configured topology is different, the device will flash a red LED at “PN” respectively “OPT” on SIMOTION device.	

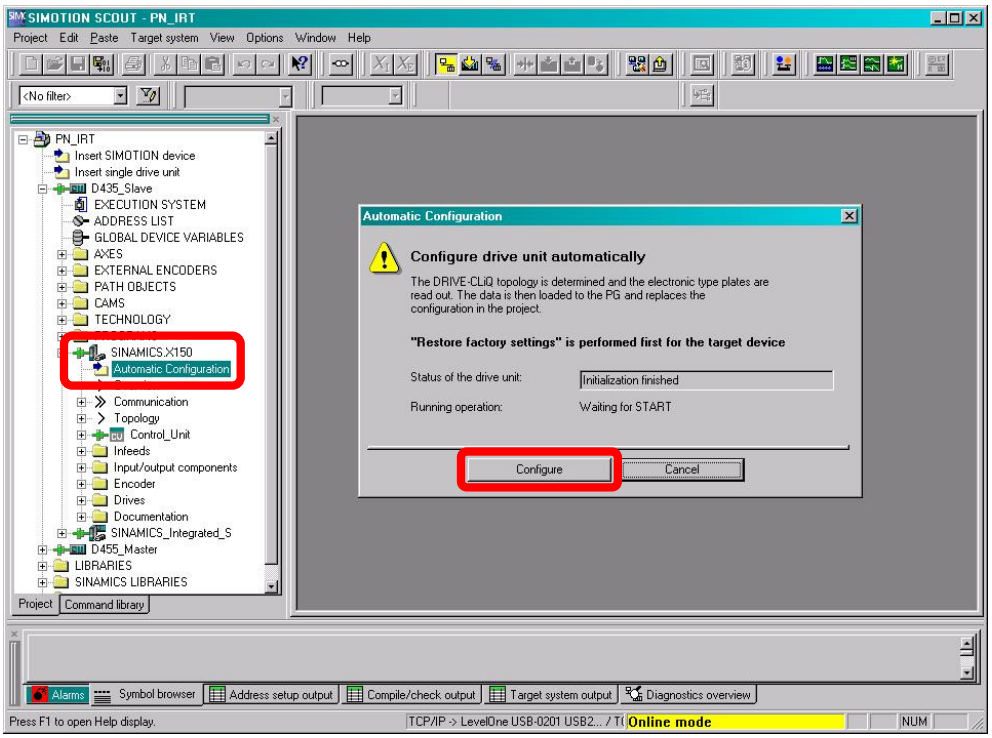
3.	The used cable length respectively the signal delay time must be adjusted (especially for fiber optic or slip rings). For copper cables use the default setting (Cable length < 100 m).	
4.	Click with the right mouse button on the PN interface (e.g. X150) and select „PROFINET IO Domain Management…“.	

5.	Define the PROFINET IRT send clock time and the Sync-Domain as shown. Each Sync-Domain requires an IO-Controller or SCALANCE X200 IRT switch defined as Sync-Master. All other devices are configured as Sync-Slave. The send clock time can be between 250 µsec and 4.0 msec depending on the used PROFINET devices. Select in section “Nodes” the corresponding device and define the role as Sync-Master by double click on it. In the sample project the master CPU will be the Sync-Master.	
6.	Select under “Nodes” the slave CPU and define all other devices of this Sync-Domain as Sync-Slave (High Performance = RTC3). It means all PROFINET devices in the same IRT line, between the Sync-Master and the last IRT device must be configured as Sync-Slave. A SIMOTION controller can also be a **Sync-Master (redundant)** (see sample project). In this case the Sync-Master (redundant) takes the role for synchronizing all IRT devices. This function is designed for commissioning of machine parts. **Please observe that is not possible to switch to often between Sync-Master and Sync-Master (redundant). If required a reset of the PN interface is needed (power OFF/ON)!**	 **Note:** If the slave CPU is configured as **Sync-Slave** instead of **Sync-Master (redundant)** the CPU can be disconnected from the Sync-Master how offent you want!
7.	Save and compile Hardware config.	
8.	Download first both Hardware configurations and then download the SIMOTION projects into the CPUs.	

4.4 Commissioning of the drive

In SIMOTION SCOUT the commissioning of the drive has to be made. These steps are shown only particular.

Figure 4-1



SERVO_02

- Connector: X1
- Drive Object (DO) type: Servo
- Motor with Drive-CLiQ (1FK7022-5AK71-1LG0)
- Without motor holding brake
- SIEMENS standard telegram 105, PZD-10/10

SERVO_03

- Connector: X2
- Drive Object (DO) type: Servo
- Motor without Drive-CLiQ (1FK7022-5AK71-1AG0)
- Without Motor holding brake
- SIEMENS standard telegram 105, PZD-10/10

Note

A signal for (p0864) has to be configured. In the sample project the constant “1” is used for “Infeed operation“ (p0864).

In real application the constant “1” should not be used for “Infeed operation” (p0864).

CAUTION

If the configuration includes a Single Line Module for 1AC 230V (training unit SIMOTION D435) then you have to modify the parameters for both Servo module as followed:

p0210: 345V
p1244[0]: 401V
p1248[0]: 240V

See also the Entry ID 27038754:
„Change/Replace Motor Module in SINAMICS S120 Demo Case“
http://support.automation.siemens.com/WW/view/en/27038754

4.5 Axis and I/O configuration

4.5.1 … on the master CPU

Axis ”MasterAxis”

- Position axis
- Rotary
- Virtual
- Modulo axis: 0..360° (Axis > „Mechanics“)

I/O variables (ADDRESS LIST)

For data exchange between master and slave CPU user data have to be transmitted (e. g. control or status data). Therefore you have to define in SIMOTION SCOUT for both CPUs I/O variables in “ADDRESS LIST”. On the master CPU following I/O variables are required:

Table 4-4

Variable	Address	Type	Bit	Note
`ib16ReceiveData`	**P**`IW0`	`WORD`	0	→ Axis „SlaveAxisRed“ is synchronized
			1	TRUE → Axis „SlaveAxisBlue“ is synchronized
			2..15	reserved
`qb16SendData`	**P**`QW0`	`WORD`	0	TRUE starts (FALSE stops) the synchronous operation of axis „SlaveAxisRed“
			1	TRUE starts (FALSE stops) the synchronous operation of axis „SlaveAxisBlue“
			2..15	reserved

Note

The I/O variables are assigned to the process image of the “BackgroundTask”, but by referring the variable with the syntax **P** (not %) the access will be **directly**. In the sample project the I/O variables are copied into a user process image.

Fixed Process image of the „BackgroundTask“ (64 Bytes):
- Input range: `%IB0..%IB63`
- Output range: `%QB0..%QB63`

4.5.2 … on the slave CPU

Axis „SlaveAxisRed“

- Synchronous operation
- Rotary
- Electrical axis
- Mode: Standard (position control)
- Drive: SERVO_02
- Encoder: Absolute encoder, cyclic absolute (Encoder pulses per revolution 512, Data width 21)
- Modulo axis: 0..360° (axis > „Mechanics“)

Axis „SlaveAxisBlue“

- Synchronous operation
- Rotary
- Electrical axis
- Mode: Standard (position control)
- Drive: SERVO_03
- Encoder type: incremental encoder: Sin/Cos incremental (Encoder pulses per revolution 2048). The encoder is connected via drive-CLiQ using SMC20.
- Modulo axis: 0..360° (axis > „Mechanics“)

I/O variables (ADDRESS LIST)

On the slave CPU following IO variables have to be defined.

Table 4-5

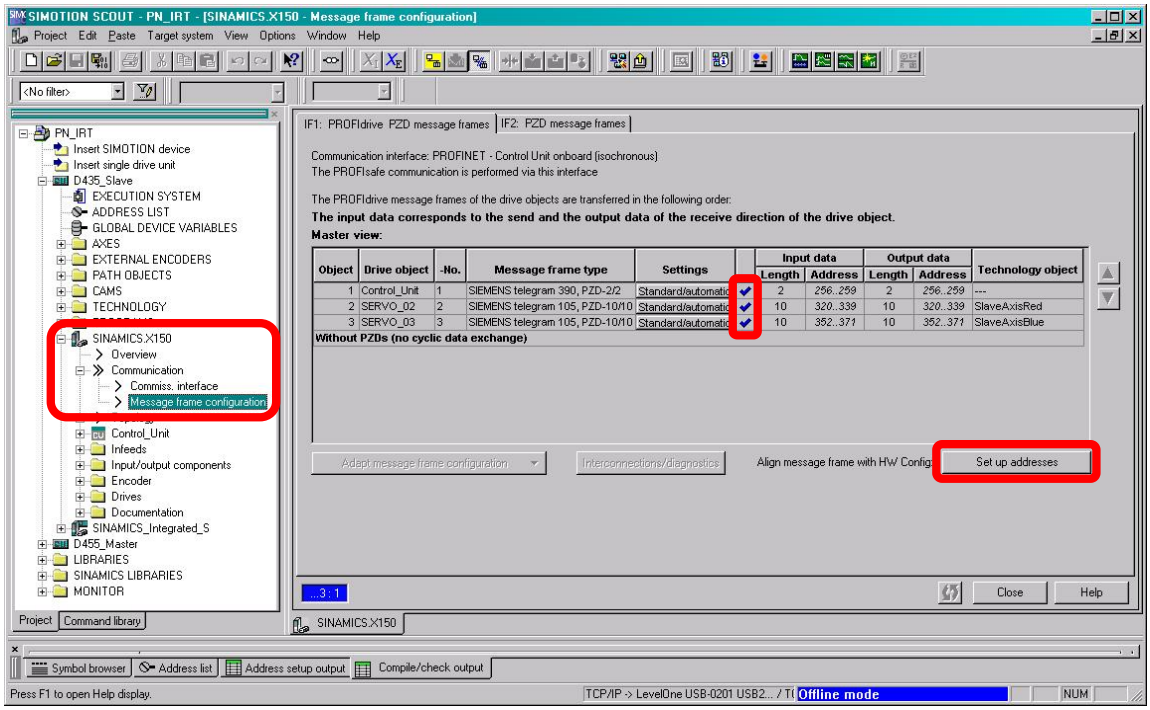
Variable	Address	Type	Bit	Note
ib16ReceiveData	PIW0	WORD	0	TRUE starts (FALSE stops) the synchronous operation of axis „SlaveAxisRed“
			1	TRUE stops (FALSE stops) the synchronous operation of axis „SlaveAxisBlue“
			2..15	reserved
qb16SendData	PQW0	WORD	0	TRUE → Axis „SlaveAxisRed“ is synchronized
			1	TRUE → Axis „SlaveAxisBlue“ is synchronized
			2..15	reserved

(see the note at chapter “I/O variables (ADDRESS LIST)” of the master CPU)

Synchronization with HW Config

During the configuring of the axis the engineering system creates automatically the communication between axis and the drive, because the setting “Standard/Automatic” is active (see chapter 4.4).
Following dialog appears after clicking on “Communication > Message frame configuration”. If necessary the telegram configuration must be aligned with the HW Config.

Figure 4-2

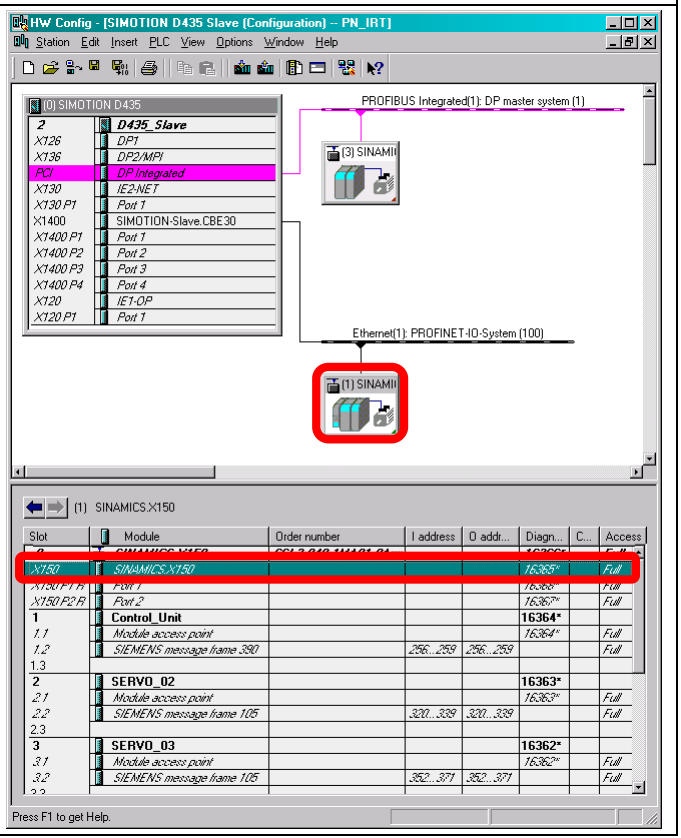


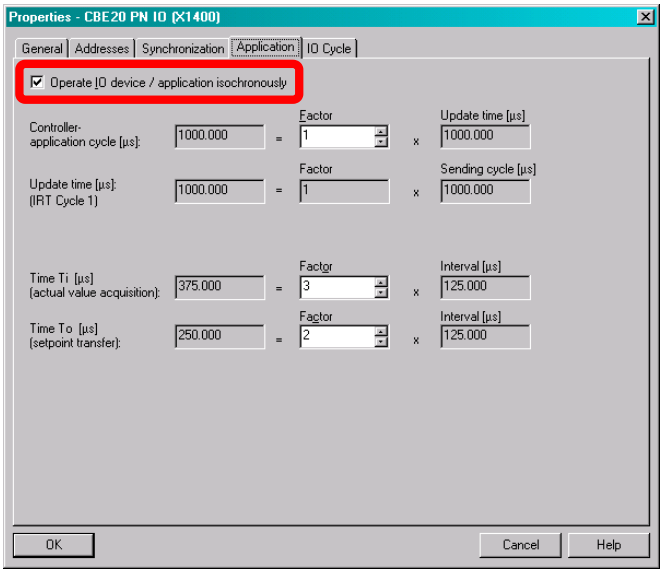
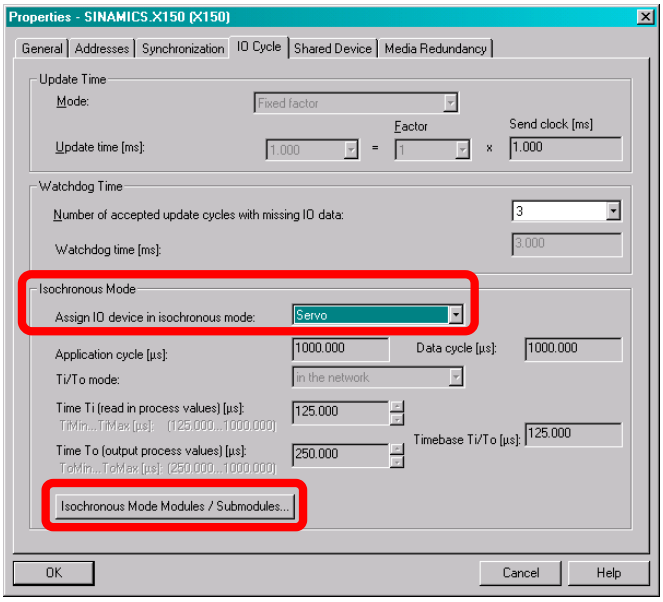
On electrical axis as default “Dynamic Servo control“ (DSC) is activated. In this case automatically the PROFdrive telegram “SIEMENS Telegram 105” will be configured. This telegram type needs isochronous mode, see following chapter.

4.6 Activate isochronous mode on the drive

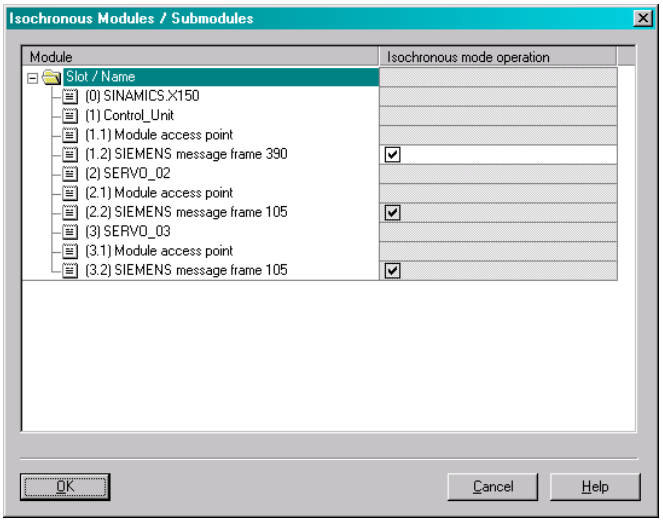
With PROFIdrive telegrams including control word 2 (STW2) and status word 2 (ZSW2) the application must (can) be synchronized to the communication send clock (e. g. Standard Telegram 2, 3, … or SIEMENS Telegram 105, 106, …).

Table 4-6

No.	Instruction	
1.	Open HW Config of the slave CPU „D435-Slave“ to activate the isochronous mode.	
2.	Open the properties dialog of the PN interface by double click.	

3.	If you use a SIMOTION Version <= V4.1 you have to enable the feature “Operate IO device / application isochronously” in the tab “Application”.	SIMOTION device <= V4.1 
	With a SIMOTION device > V4.1 the activation of the isochronous mode is in the tab “IO Cycle”. Here the IO-Device (SINAMICS drive) will be synchronized to the “Servo”. It means the drive will now work in isochronous mode to the ServoTask of the SIMOTION CPU. By clicking on “Isochronous Mode Modules / Submodules…” the submodules can be selected for isochronous mode. As default all modules are assigned. This setting should not be changed.	SIMOTION device > V4.1  **Note:** Please observe that the summary of Ti + To is not equal to the send clock time. Otherwise the drive must copy input and output data in one cycle. This will increase the load of the Control Unit (CU)!

4.6 Activate isochronous mode on the drive

4.	Activate for all PROFIdrive modules the isochronous mode operation.	
5.	Save and compile HW Config.	

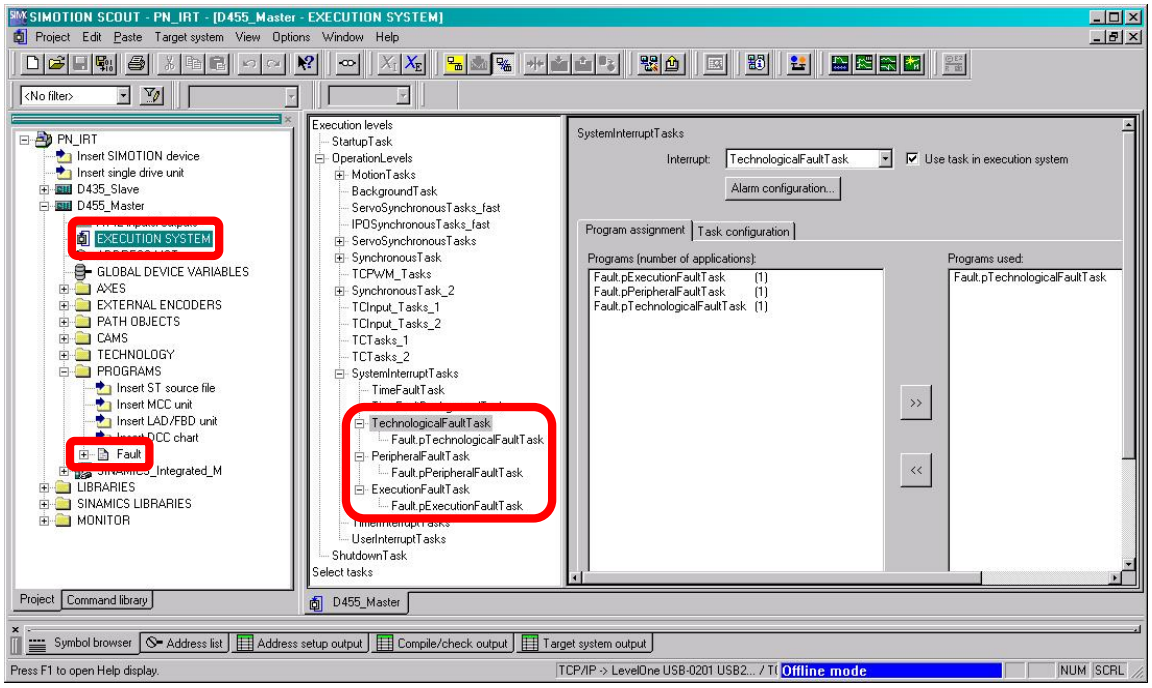
4.7 Description of the sample program

4.7.1 ST unit Fault

To prevent the SIMOTION controller from changing into the STOP mode in case of execution, peripheral or technological fault, a program must be executed in the

- ExecutionFaultTask,
- PeripheralFaultTask and
- TechnologyFaultTask.

Figure 4-3



In the sample project exists a ST unit `Fault` with the programs

- `pTechnologyFaultTask`,
- `pPeripheralFaultTask` and
- `pExecutionFaultTask`.

In case of a peripheral fault a global variable gboError will be set in the pPeripheralFaultTask. This flag will be used to acknowledge the technological faults after unplugging and plugging PROFINET cables.

This acknowledgement will be handled by the “BackgroundTask” of the slave CPU “D435_Slave” (at the end of ST unit `SlaveAxes`). On the master CPU “D455_Master” only a virtual axis exists, therefore no acknowledgement is needed.

4.7.2 ST unit `InterfaceSynchronization`

The PROFINET interface of the slave CPU “D435_Slave” (Sync-Slave) must be synchronized in the user program.

For this, the program `pStartupTask` calls the system function `_enableDpInterfaceSynchronizationMode()`. The program is located in the ST unit `InterfaceSynchronization` and has to be assigned to the “StartupTask”.

Note The Sync-Master must not be synchronized by the user application.
Only if the Sync-Master is an IRT I-Device it has to be synchronized in the user program!

4.7.3 ST unit `MasterAxis` on the master CPU

Global variables for operating the sample application

Table 4-7

Variable	Type	Note
`gsMaster.sMasterAxis.boMove`	`BOOL`	Start and stop the master axis
`gsMaster.sMasterAxis.r64CommandVelocity`	`LREAL`	Setpoint speed for the master axis
`gsSlave.sAxisRed.boEnableGearing`	`BOOL`	De-/Synchronization of the slave axis (red) to the master axis
`gsSlave.sAxisRed.boSynchronized`	`BOOL`	Status of the slave axis (red)
`gsSlave.sAxisBlue.boEnableGearing`	`BOOL`	De-/Synchronization of the slave axis (blue) to the master axis
`gsSlave.sAxisBlue.boSynchronized`	`BOOL`	Status of the slave axis (blue)

Function block `FBMoveAxis` to control the master axis

This function block supports a position controlled movement of an axis. The input `enable` is used to start (`TRUE`) and stop (`FALSE`) the movement. The setpoint speed is specified at the input `commandVelocity`. The setpoint speed can be modified during operation. Further dynamic parameters are hard-coded in the function block.

This function block will be called in the program `pBackgroundTask`. This program must be executed in the “BackgroundTask”.

Table 4-8

Globale variables		I/O variables
`gsSlave.sAxisRed.boEnableGearing`	→	`qboEnableGearingAxisRed`
`gsSlave.sAxisRed.boSynchronized`	←	`iboAxisSynchronizedRed`
`gsSlave.sAxisBlue.boEnableGearing`	→	`qboEnableGearingAxisBlue`
`gsSlave.sAxisBlue.boSynchronized`	←	`iboAxisSynchronizedBlue`

4.7.4 ST unit SlaveAxes on the slave CPU

Function block FBSyncAxis to control the slave axis (gearing axis)

This function block allows to de-/synchronize a gearing axis. The input enable is used to start (TRUE) and stop (FALSE) the synchronous operation.
The de-/synchronization starts immediately, depending on the dynamic parameters which are hard-coded in the function block.

This function will be called in program pBackgroundTask. This program must be executed in the “BackgroundTask”.

Table 4-9

I/O-Variablen	Funktion
iboEnableGearingAxisRed	De-/synchronization of the slave axis (red)
iboEnableGearingAxisBlue	De-/synchronization of the slave axis (blue)
qboAxisSynchronizedRed	Status of the slave axis (red)
qboAxisSynchronizedBlue	Status of the slave axis (blue)

As already mentioned at the end of this unit in the program pBackgourndTask the automatically acknowledgement of a technological fault is implemented.

Acknowledgement of technological faults after sign-of-life failure

After sign-of-life failure the Alarm 20005 will appear. This alarm will be acknowledged if the cyclic communication between drive object (DO) of the SINAMICS and the technology object (TO) of the SIMOTION is established again.

```
_toAxis.actorMonitoring.cyclicInterface = ACTIVE
```

Acknowledgement of technological faults after disconnection to the master CPU

If the PROFINET cable between the master and slave CPU is removed during distributed synchronous operation, all synchronous objects get the technological alarm 50201. This alarm is acknowledged if the cyclic communication between drive object (DO) of the SINAMICS and the technology object (TO) of the SIMOTION is established again.

_toAxis.actorMonitoring.cyclicInterface = ACTIVE

Note For the command value calculation (interpolation) of the distributed synchronous operation maybe the number of level overflows in the IPO-Task or the IPO cycle time has to be increased depending on the CPU type.

4.8 Configuration of the direct data exchange broadcast

In following cases the direct data exchange broadcast (German: Controller-Controller-Querverkehr) will be used for communication between more than one SIMOTION CPUs:

- **Distributed synchronous operation**
  Data transmission between master and slave axes (located on different CPUs)
- **User data**
  Data transmission between more than one SIMOTION CPUs via I/O variables

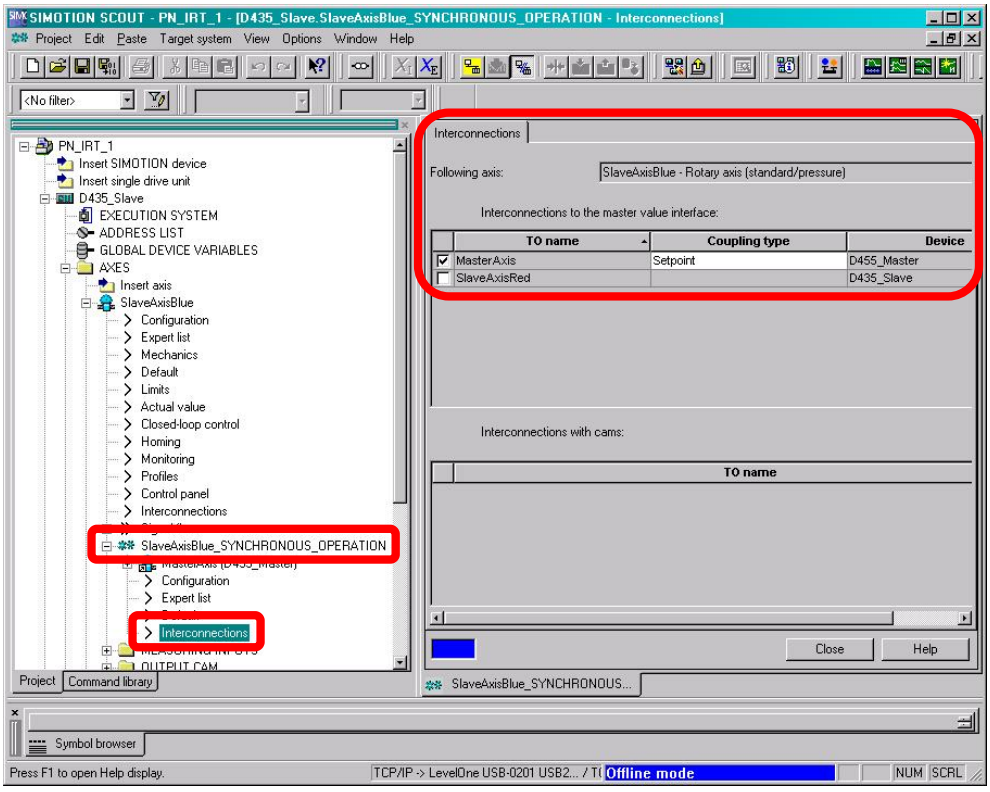
In total an amount of 3072 input bytes and 3072 output bytes per SIMOTION CPU can be transmitted via direct data exchange broadcast.

Distributed synchronous operation

The communication data for distributed synchronous operation are automatically created by the SIMOTION SCOUT when defining a master and slave axis.
The HW Config doesn't need to be changed for this.

In the sample project both gearing axis „SlaveAxisRed" and „SlaveAxisBlue" are synchronized to the setpoint value of the virtual axis "MasterAxis" of CPU "D455_Master" (distributed synchronous operation, see following figure).

Figure 4-4



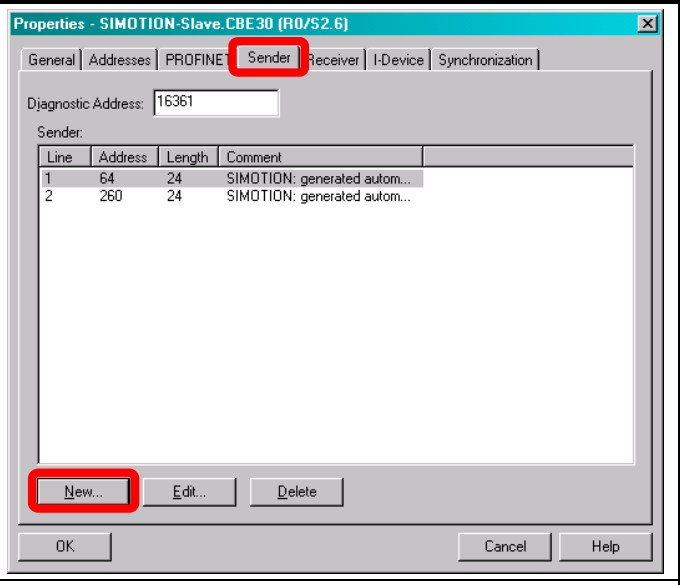
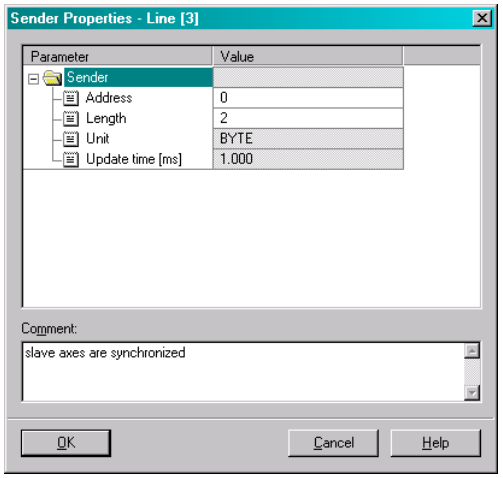
24 bytes are required for each synchronous operation. This will be allocated in HW Config via direct data exchange broadcast (see following figures).

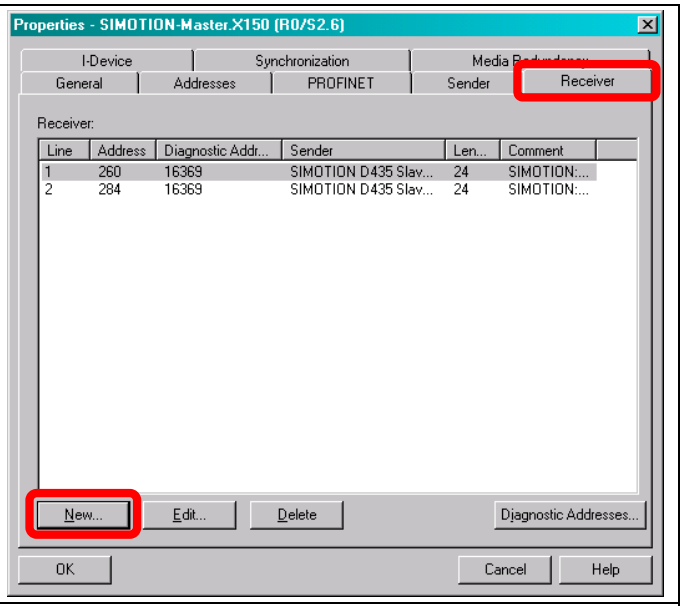
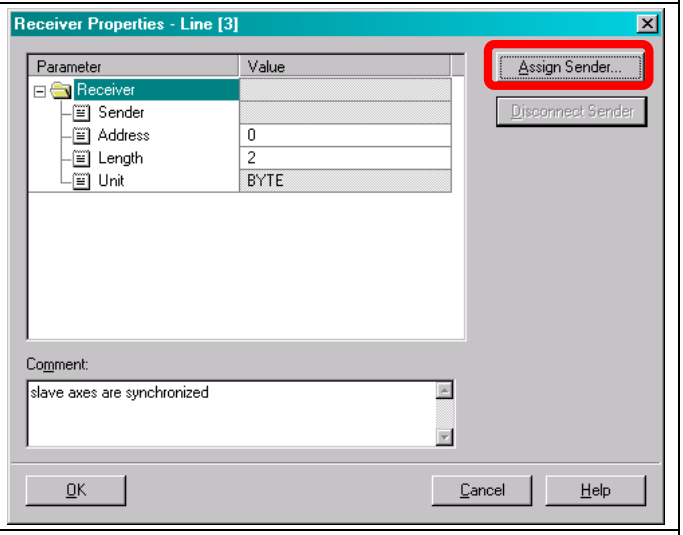
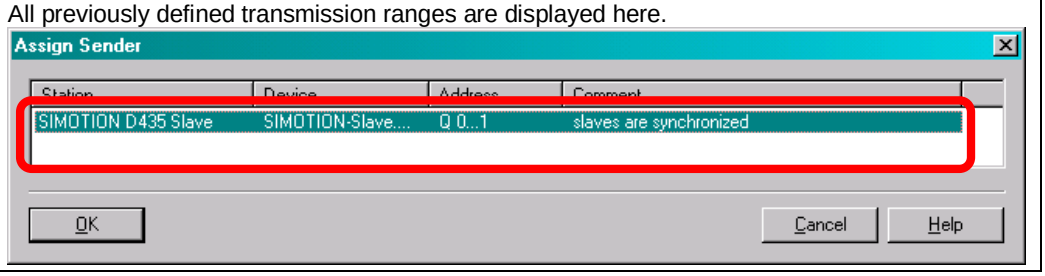
CAUTION **The 24 bytes input and output data created by the SIMOTION SCOUT may not be changed.**

User data

User data transferred via direct data exchange broadcast must be configured in HW Config.

Table 4-10

No.	Instruction	
1.	Double click the PROFINET interface of the SIMOTION controller to open its properties. All send data for the direct data exchange broadcast is listed under the tab “Sender”. Here is also the data for both synchronous operations (24 bytes each). **This configuration may not be manipulated.** Create additional user data by clicking on “New”.	
2.	In the sample project the user data is created as a WORD starting from the address 0. Max. 254 bytes can be created per data block. Further max. 3 KBytes can be configured for each sender. Under “Comment” a comment for each data block can be entered.	

3.	After defining all send data the receive data can be configured. Open for this the HW Config of the receiving CPU. Here is also the input data for both synchronous operations (24 bytes each). **This configuration may not be manipulated.** Create additional user data by clicking on “New”.	
4.	Press “Assign Sender…” to select the previously defined send data of another CPU. In the sample project the user data is created as a WORD starting from the address 0. Under “Comment” a comment for each data block can be entered.	
5.	All previously defined transmission ranges are displayed here. 	

Note First configure all send data of the controllers. After that the receive data can be created by using the previously defined send data of the other controllers.

Note A SIMOTION CPU supports max. 3 KBytes of data for each direction via direct data exchange broadcast, including the 24 bytes for each synchronous operation.

4.9 Configuration of the IRT I-Device

Alternatively to direct data exchange broadcast the IRT I-Device can be used to transfer data in isochronous mode between 2 SIMOTION CPUs:

- **Distributed synchronous operation**
  Data transmission between master and slave axes (located on different CPUs)
- **User data**
  Additional data transmission between SIMOTION CPUs via I/O variables

In total an amount of 1440 input bytes and 1440 output bytes per SIMOTION CPU can be transmitted via I-Device communication. Please observe that an I-Device can be assigned only to one PROFINET IO-Controller.

The benefit by using the IRT I-Device instead of direct data exchange broadcast is the possibility to use two different SIMOTION projects (configuration via GSD file). In this case more engineers are able to work independent with different SIMOTION projects.

CAUTION

Please observe that it is not possible to add on the SIMOTION PN interface, which is already configured as IRT I-Device, any additional IRT IO-Device. In this case only RT IO-Devices are possible. The CPU doesn't support IRT I-Device and IRT IO-Controller at the same time!

Due to this issue the master CPU in the sample project must be configured as IRT I-Device, because on the slave CPU additional IRT IO-Devices are connected.

With SIMOTION D4x5-2 up from firmware V4.3 two PROFINET interfaces (X150 and CBE30-2) are supported. In this case the SIMOTION CPU can be used one PN interface (CBE30-2) as IRT I-Device and on the other interface (X150) as IRT IO-Controller for additional IRT I-Devices.
More information are available in the entry "Use of two PROFINET interfaces with SIMOTION"
(see http://support.automation.siemens.com/WW/view/en/59396321)

Distributed synchronous operation

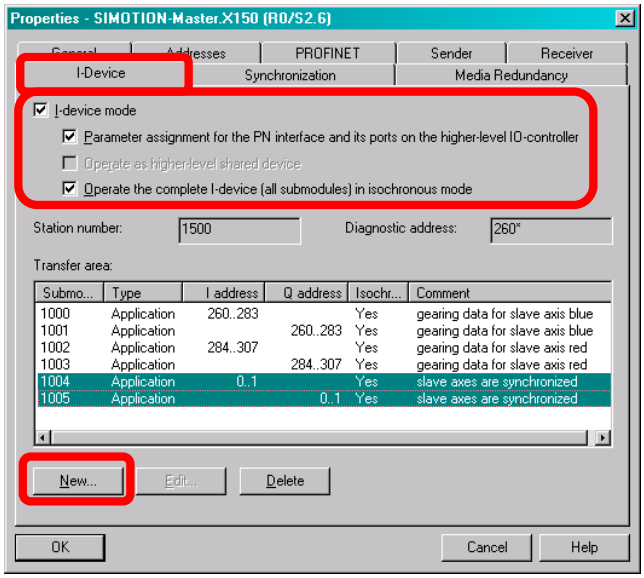
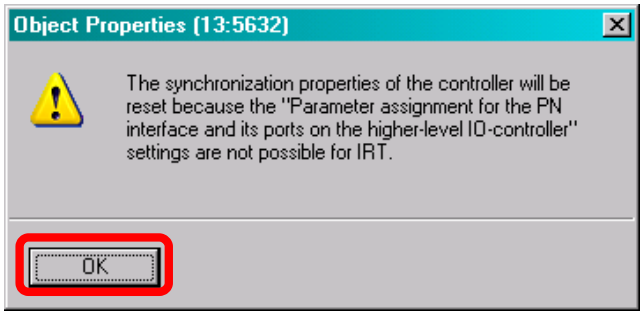
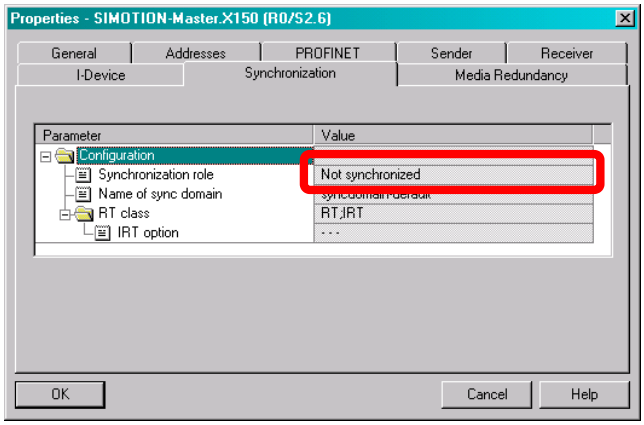
If distributed synchronous operation is configured via IRT I-Device and not configured by direct data exchange broadcast, the 24 bytes input and output data per synchronous operation are not configured automatically. By using IRT I-Device the configuration has to be done manually like for the user data.

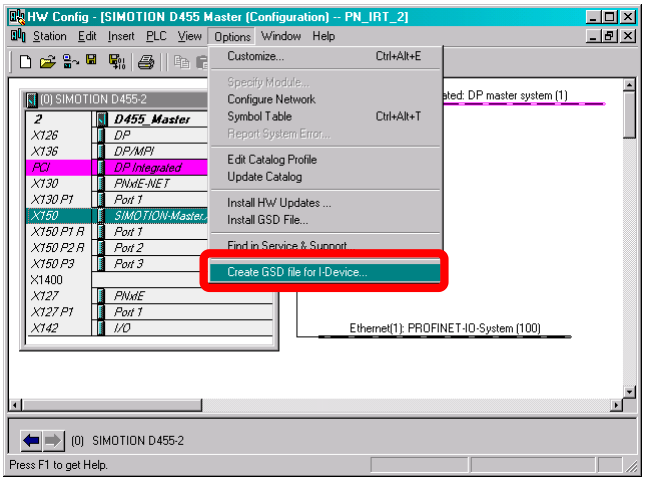
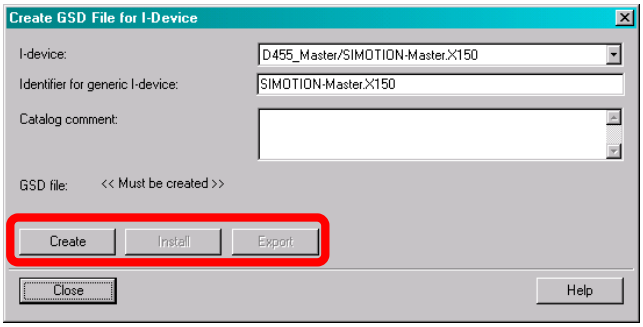
User data

The user data must be configured manually in HW Config for the IRT I-Device communication.

4.9.1 HW Config of the IRT I-Device

Table 4-11

No.	Instruction	
1.	Open HW Config of master CPU “D455_Master”	
2.	By double click on the PROFINET interface the properties dialog will open. Select in the tab “I-Device” the option • “I-device mode“, • “Parameter assignment for the PN interface and its ports on the higher-level IO-Controller” and • “Operate the complete I-device in isochronous mode” Click on “New...“ and create all input- and output modules as shown on the right side. **For each** synchronous operation 24 bytes for input and output data are required. In the sample project the data for the blue axis are assigned at byte address 260 to 283, for red axis at byte address 284 to 307. In addition 2 bytes of user data are created in the sample project for each direction.	 If the checkbox “Parameter assignment for the PN interface and its ports on the higher-level IO-Controller” is activated maybe following dialog must be confirmed, because the PN interface can only be a RT IO-Controller, not IRT. 
3.	Change to the tab “Synchronization“ and if required change the synchronization role to „Not synchronized“. In this case the SIMOTION CPU operates on this PN interface as RT IO-Controller. Confirm the dialog by clicking on „OK“.	

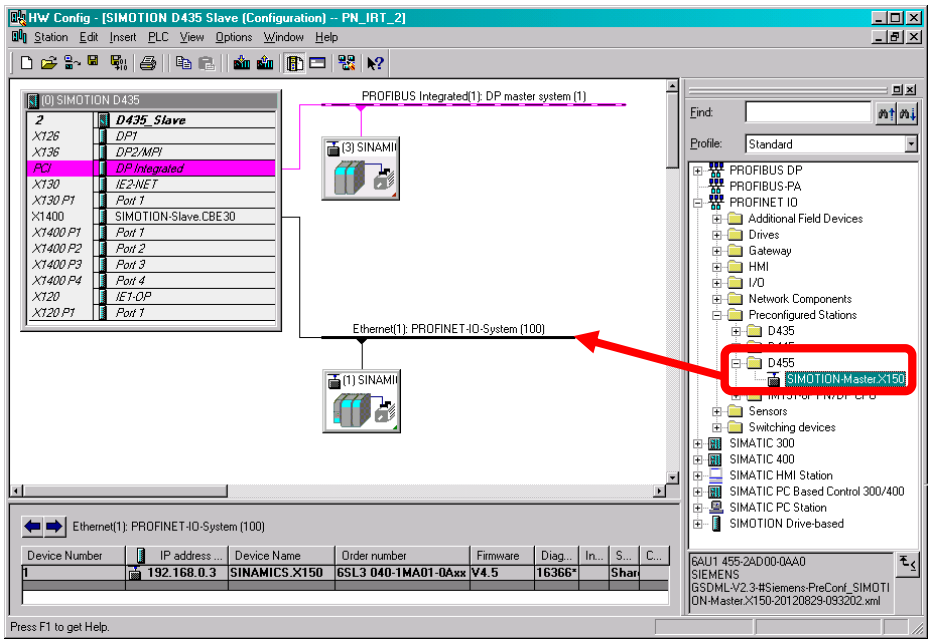
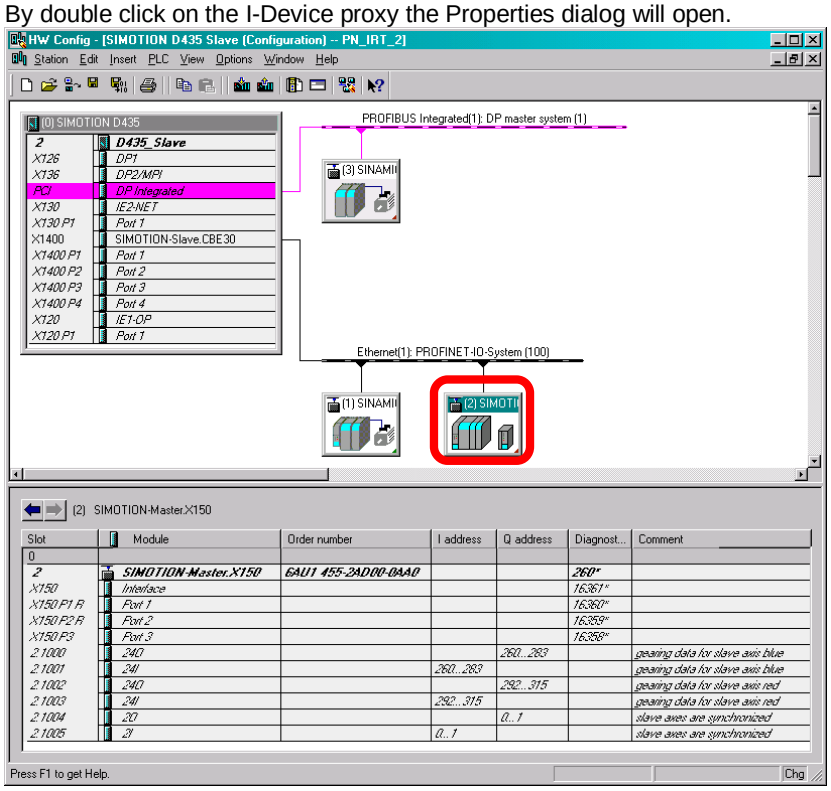
No.	Instruction	
4.	Save and compile after finishing the configuration of the I-Device.	
5.	Now the GSD file for the I-Device must be created. Select for this in the menu “Options > Create GSD file for I-Device…”.	
6.	Click on “Create“ to generate the file. The generated file must be imported in the engineering system, which is used for the higher-level IO-Controller configuration. The imported GSD File will be shown in the hardware catalogue in following directory „PROFINET IO > Preconfigured Stations > D455“.	 **Export** The higher-level IO-Controller and the I-Device can be located in different SIMOTION projects. The export function can be used to transfer the GSD file to another engineering system. The function “Options > Install GSD file” can be used to import the file in an engineering system.

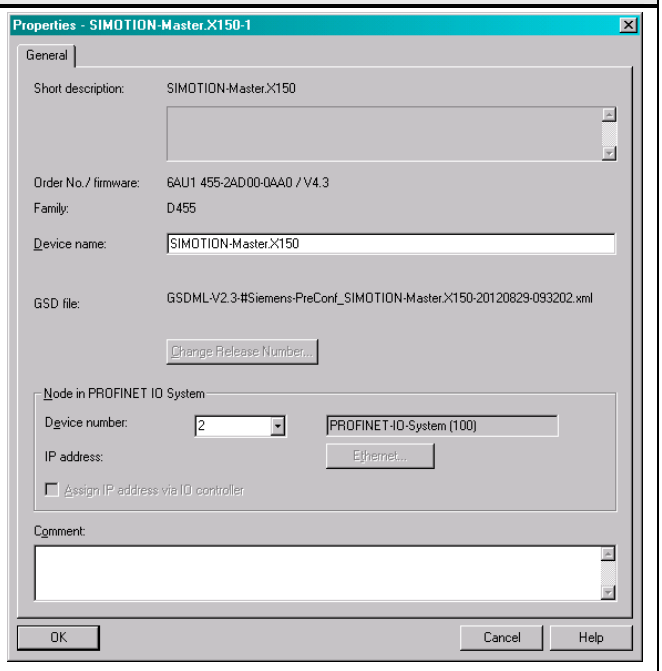
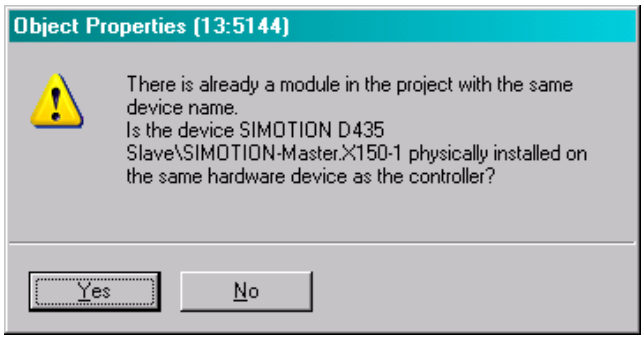
Note If the I-Device configuration is changed afterwards, it has to be created and installed again. In the HW Config of the higher-lever IO-Controller the I-Device proxy has to be updated. After that the addressing of the inputs and outputs must be checked again.

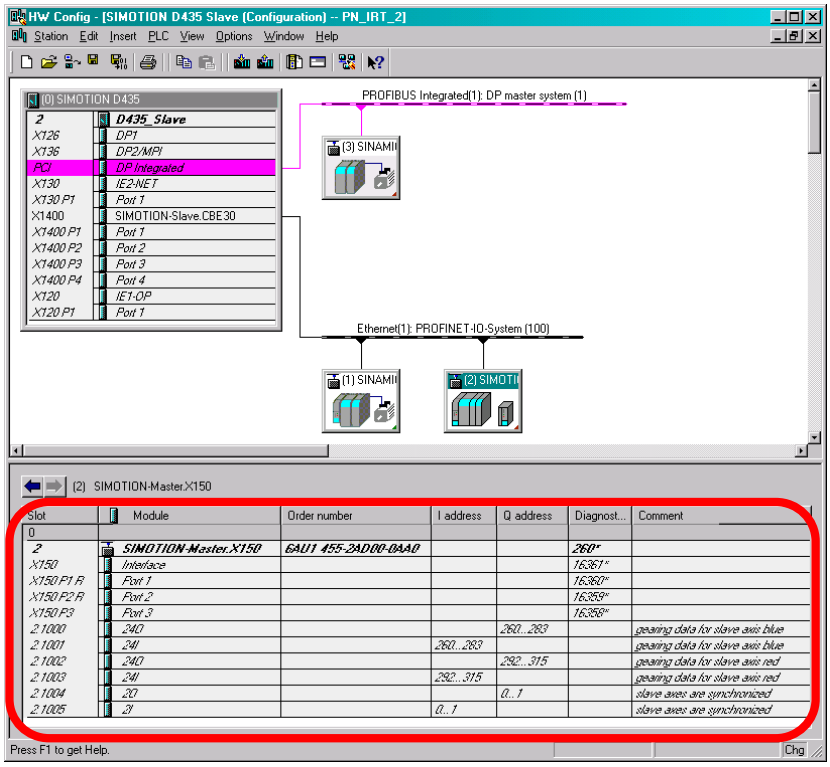
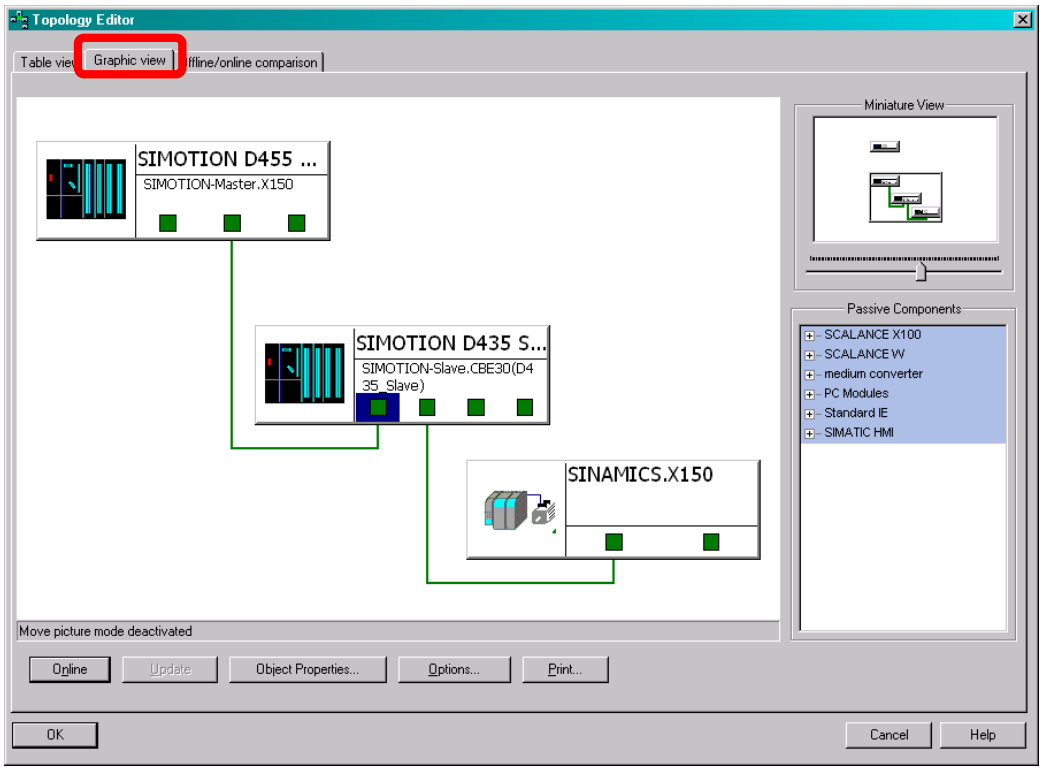
Note If the configuration of the higher-level IO-Controller is done by another engineering system, the GSD file of the I-Device must be exported and installed in the other engineering system.

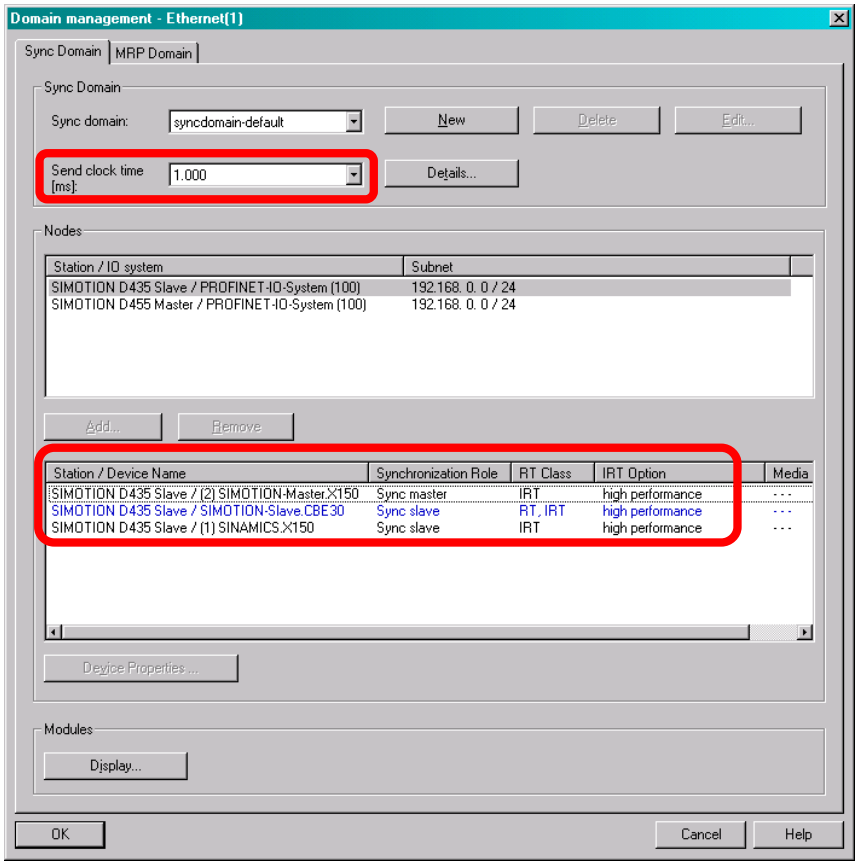
4.9.2 HW Config of the higher-lever IO-Controller

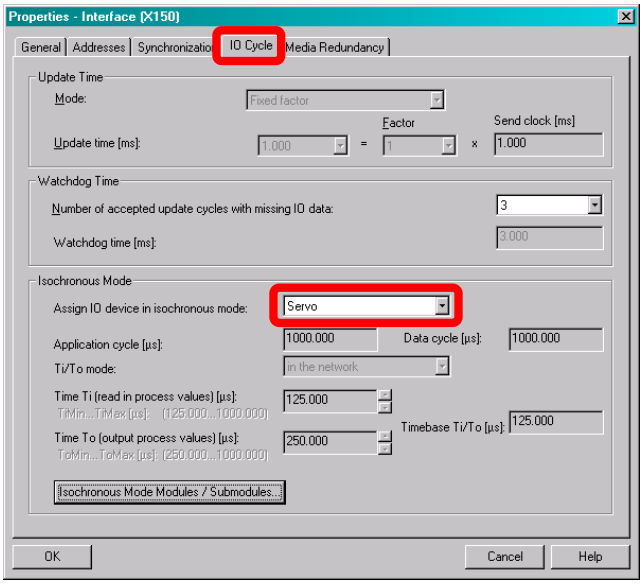
Table 4-12

No.	Instruction
1.	Open HW Config of the slave CPU “D435_Slave”. Add the I-Device via Drag&Drop to the PROFINET IO-System. You will find the imported I-Device in following directory „PROFINET IO > Preconfigured Stations > D455“. 
2.	By double click on the I-Device proxy the Properties dialog will open. 

No.	Instruction	
3.	Please observe that the device name of the I-Device is the same as the device name of the corresponding CPU. If the I-Device Proxy and the corresponding CPU are in the same project, so the engineering system will add automatically „-1“ at the end of the device name. This is caused by the rule, that a device name must be unique in a PROFINET IO-System. But if the device is an I-Device proxy the name must be the same as on the corresponding CPU.	
	After changing the device name following message must be confirmed with “Yes”, because it is the same hardware. In the sample project the device name of the CPU is „SIMOTION-Master.X150“.	

No.	Instruction
4.	The I/O addresses can be modified by double click on the modules. The user data starts at address 0. 
5.	The I-Device is used in isochronous mode via **PROFINET IRT**, so the PROFINET IO topology must be configured. To open the topology editor click with the right mouse button on the PROFINET IO interface and select „PROFINET IO Topology…“. 

No.	Instruction
6.	Edit the PROFINET Domain Management by clicking with right mouse button on the PROFINET IO interface and select „PROFINET IO Domain Management …“. Define the Sync-Master and Sync-Slave to activate **PROFINET IRT**. In the sample projekt the IRT I-Device (CPU „D455_Master“) is the Sync-Master. Set all other devices to Sync-Slave. The send clock time is set to 1 msec. Depending on the used devices a send clock time between 250µsec up to 4 msec is possible. 

No.	Instruction	
7.	Double click on the PN interface of the I-Device to open the properties dialog. Change to the tab "Application" (FW <= V4.1) or "IO-Cycle" (FW > V4.1) and activate the **Isochronous mode**. With this setting the I-Device will work synchronous to the PROFINET IO send clock and to the ServoTask of the SIMOTION CPU. For this all sub modules have to operate in isochronous mode (see „Isochronous Mode Modules / Submodules…"). Confirm the dialog with "OK".	 This picture is only valid for firmware > V4.1.

NOTE Please note, that the IRT I-Device of a SIMOTION CPU with **FW V4.3** cannot be operated in isochronous when a SIMOTION CPU with **FW < V4.2** is used as higher-level I/O controller.

4.9.3 Proxy Objects for distributed synchronous Operation

With IRT I-Device the synchronous operation data are not automatically configured. Therefore following proxy objects exist.

Proxy object types

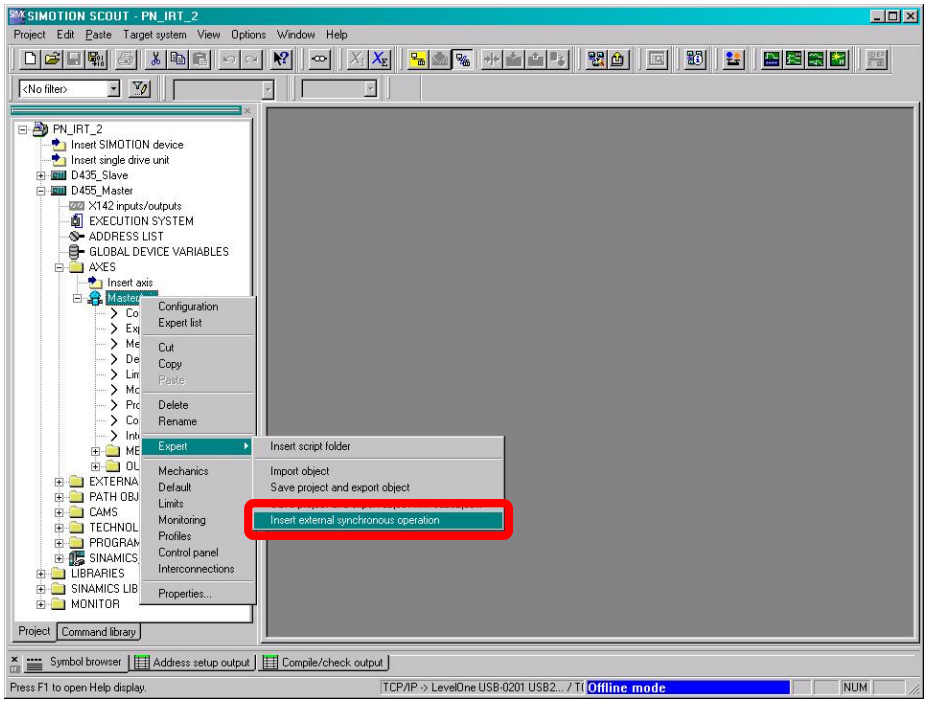
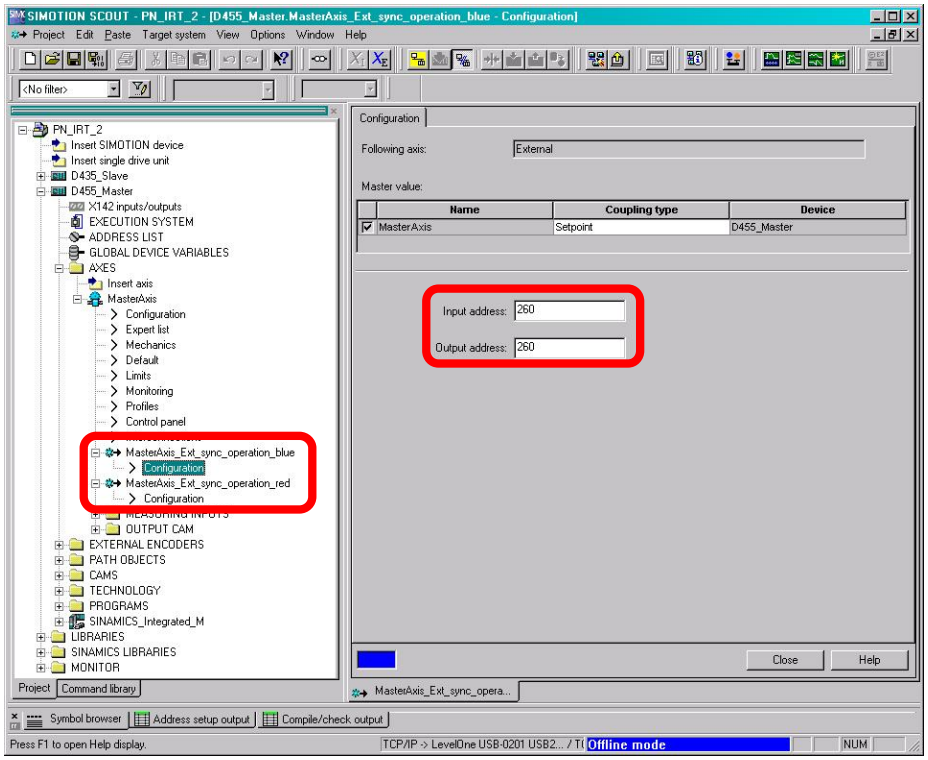
There are two different types of proxy objects:

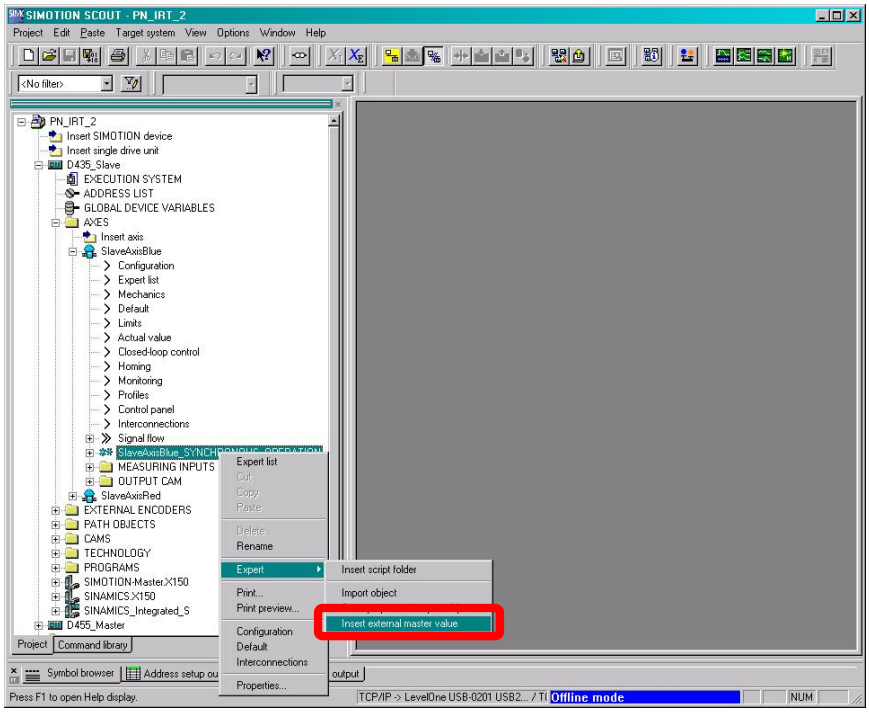
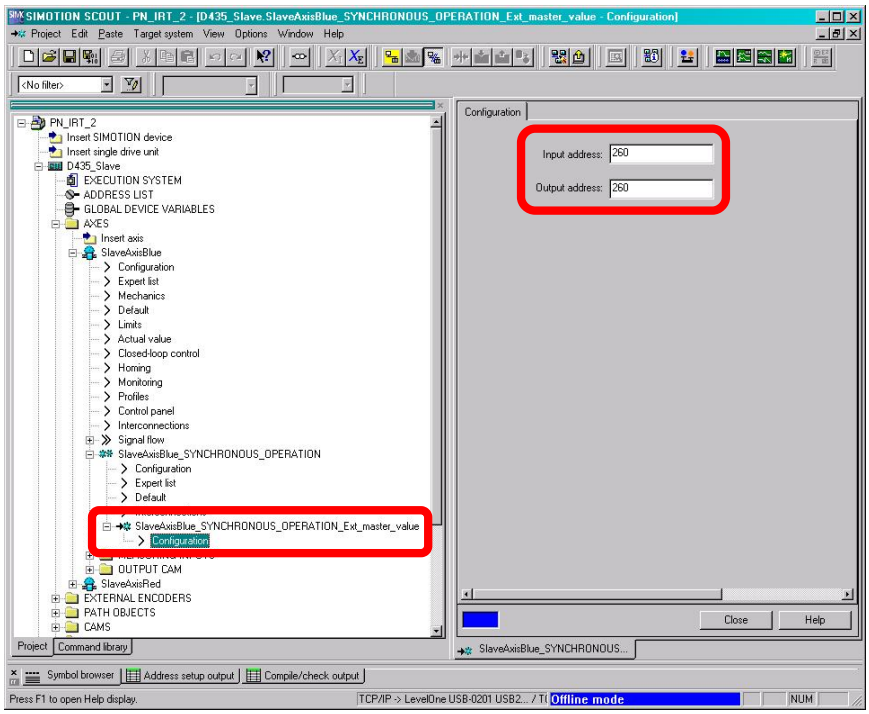
- **External synchronous operation (ExternalFollowingObjectType)**
  A proxy object for external synchronous operation can be created under the following technology object types, i.e. interconnected with them:
  - External encoder
  - Synchronized axis
  - Positioning axis
  - Path axis
- **External master value (ExternalMasterType)**
  A proxy object for an external master value can only be created under a synchronous object, i.e. interconnected with it.

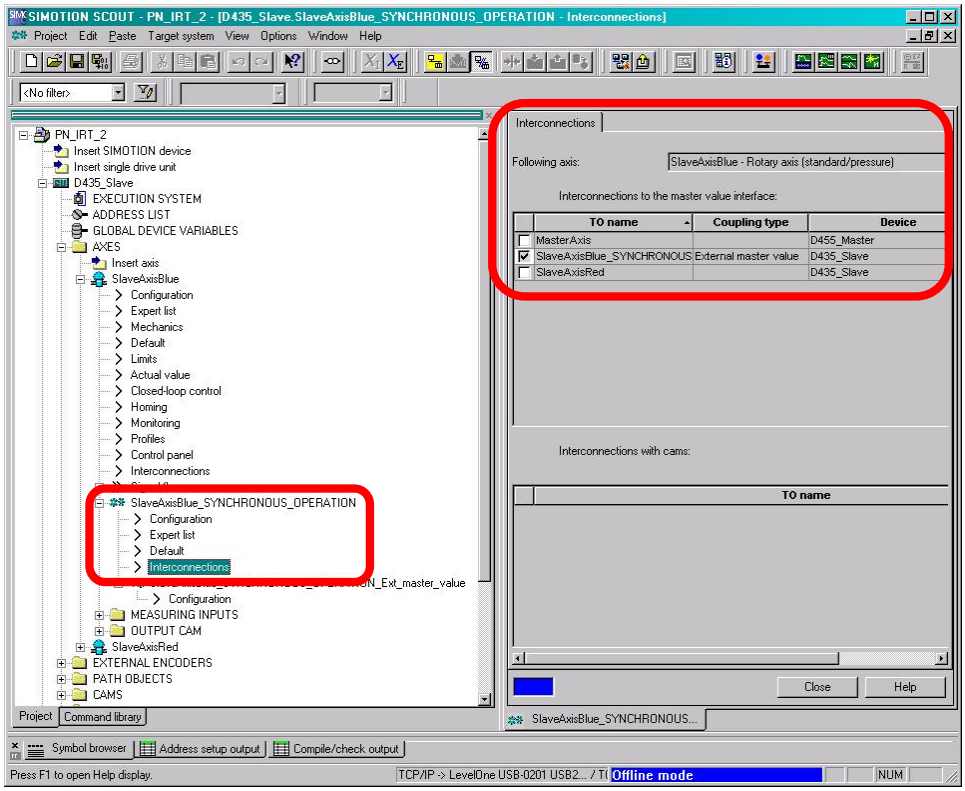
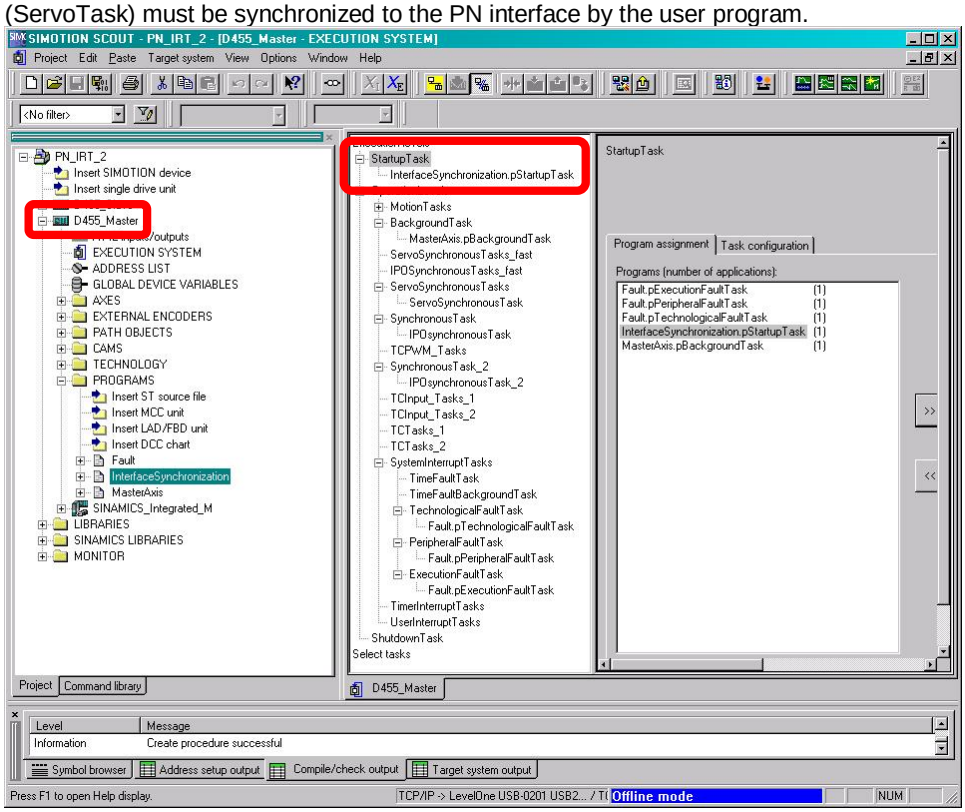
Creating of a proxy object

The proxy objects are created in SIMOTION SCOUT, see following instructions.

Table 4-13

No.	Instruction
1.	Select the master axis in SIMOTION SCOUT and click with the right mouse button on “Expert > Insert external synchronous operation“. Add for each synchronous operation an **ExternalFollowingObject.** In the sample project one proxy object for the blue axis and one for red axis. 
2.	Define the input and output addresses depending on the HW Config (see Table 4-11 Action 2). In the sample project the start address of the input and output data for the blue axis is “260“ and for the red axis “284“. 

No.	Instruction
3.	After configuration of the external synchronous operations on the master axis the external master value must be defined at the slave axes. Click with the right mouse button on the synchronous object of the corresponding gearing axis (in the sample “SlaveAxisBlue_SYNCHRONOUS_OPERATION”) and then on “Expert > Insert external master value”. 
4.	Define the input and output addresses depending on the HW Config (Table 4-12 Action 4). In the sample project the start address of the input and output data for the blue axis is “260“ and for the red axis “292“. 

No.	Instruction
5.	Automatically on the synchronous operation object an interconnection to the external master value was created. 
6.	Please observe if the IRT I-Device has the role of the Sync-Master, the SIMOTION application (ServoTask) must be synchronized to the PN interface by the user program. 

5 Startup of the application

In the archive **DistributedSyncOperation_PN_IRT_CCQ_V2_1.zip** is a SIMOTION sample project, which includes the configuration of

- a distributed synchronous operation via
- **direct data exchange broadcast.**
  (German: Controller-Controller-Querverkehr)

In the archive **DistributedSyncOperation_PN_IRT_I-Device_V2_1.zip** is a SIMOTION sample project, which includes the configuration of

- a distributed synchronous operation via
- **IRT I-Device**.

These archives can be unpacked with the SIMOTION SCOUT or SIMATIC Manager.

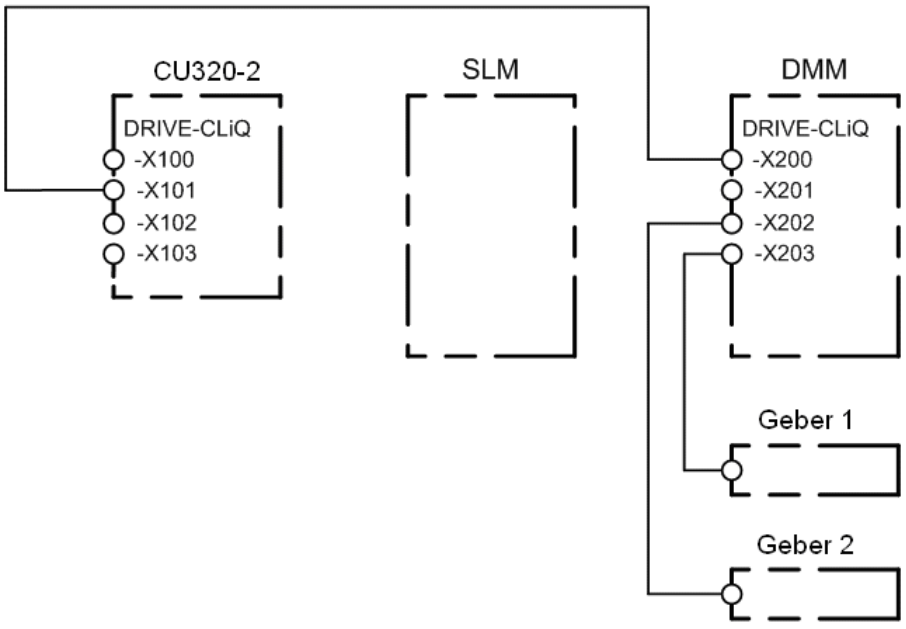
In both projects a distributed synchronous operation will be shown between a virtual master axis and two real slave axes. The drive is a SINAMICS S120 CU320-2 PN, which is connected via PROFINET IRT to a SIMOTION CPU.

Furthermore it shows how to transfer user data (control and status data) in isochronous mode between SIMOTION controllers via direct data exchange broadcast or IRT I-Device.

5.1 Preparation

5.1.1 Drive-CLiQ topology

Figure 5-1



5.1.2 IP addresses and device names

Table 5-1

Device	Device name	IP address
D455_Master (X150)	**SIMOTION-Master.X150**	192.168.0.1
D435_Slave (CBE30)	**SIMOTION-Slave.CBE30**	192.168.0.2
CU320-2 PN (X150)	**SINAMICS.X150**	192.168.0.3
PG/PC		192.168.0.99

The subnet mask is for all devices 255.255.255.0.

Note Upper characters of the configured device name are replaced by lower characters during the node initialization (e.g. HW Config download).

Note In case of a SIMOTION CPU the name of the PROFINET interface will be used as device name. Via double click on the PN interface X150 or CBE30 or CBE30-2 in HW Config the device name can be changed.

In case of a SINAMICS device the device name can be found by double click on the device in HW Config (not the PN interface). The device name is also shown in SIMOTION SCOUT or STARTER.

5.1.3 Windows settings of the PG/PC interface

The IP address of the PC Ethernet network adapter must be adapted under “Control Panel > Network connections”.

Click with the right mouse button on the network adapter and select “Properties“.

Select “Internet Protocol (TCP/IP)” and click on “Properties” to get the IP settings of this Ethernet network adapter. Set IP and subnet mask shown in Table 5-1 for the PG/PC.

5.2 Startup

Table 5-2

No.	Instruction
1	Set the IP addresses of each device to download the Configuration into the devices (see also chapter 3 “Basics”)
2	Download the corresponding SIMOTION project (direct data exchange broadcast or IRT I-Device) into all required devices.
3	Set the device name of the SINAMICS S120 „SINAMICS.X150“, or reset the PN interface to factory settings (topology-based initialization).
4	Switch both CPU to RUN state.
5	Load the user defined SIMOTION IT Web site into the master CPU “D455-Master”. The Web sites are located in folder “38486079_DistributedSyncOperation_PN_IRT_V2_1.zip > HTML > MBS”. The data content of this folder must be transferred to the folder “Files” of the SIMOTION master CPU “D455_Master”.

6 Operation of the application

6.1 Overview

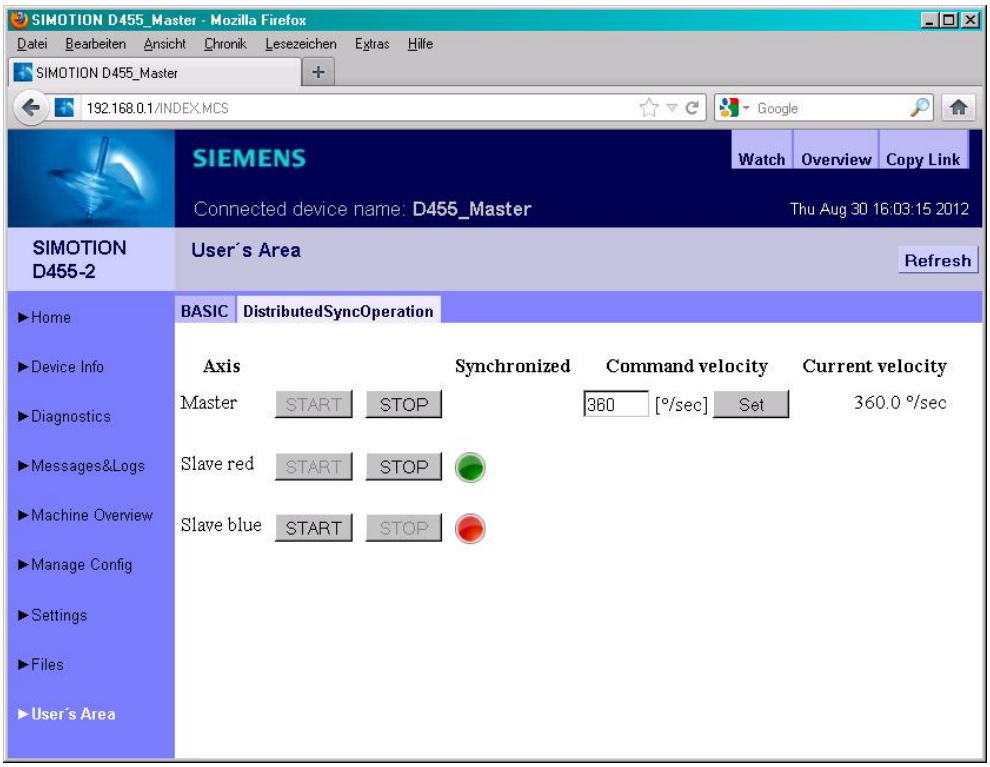
The controlling of the slave axes can be done by the global variables of the master CPU “D455_Master”. The value of these variables will be transferred via I/O variables to the slave CPU “D435_Slave”.

6.2 Test of the distributed synchronous operation

6.2.1 ... by using the user defined SIMOTION IT website

With the website „DistributedSyncOperation.mbs” all axes can be controlled via changing the velocity and start or stop the synchronous operation. Each axis can be controlled separately.

Figure 6-1



6.2.2 ... by direct control of the global variables

The symbol browser allows to control the variables of the ST unit `MasterAxis`. The global variables have following functionality.

Figure 6-2

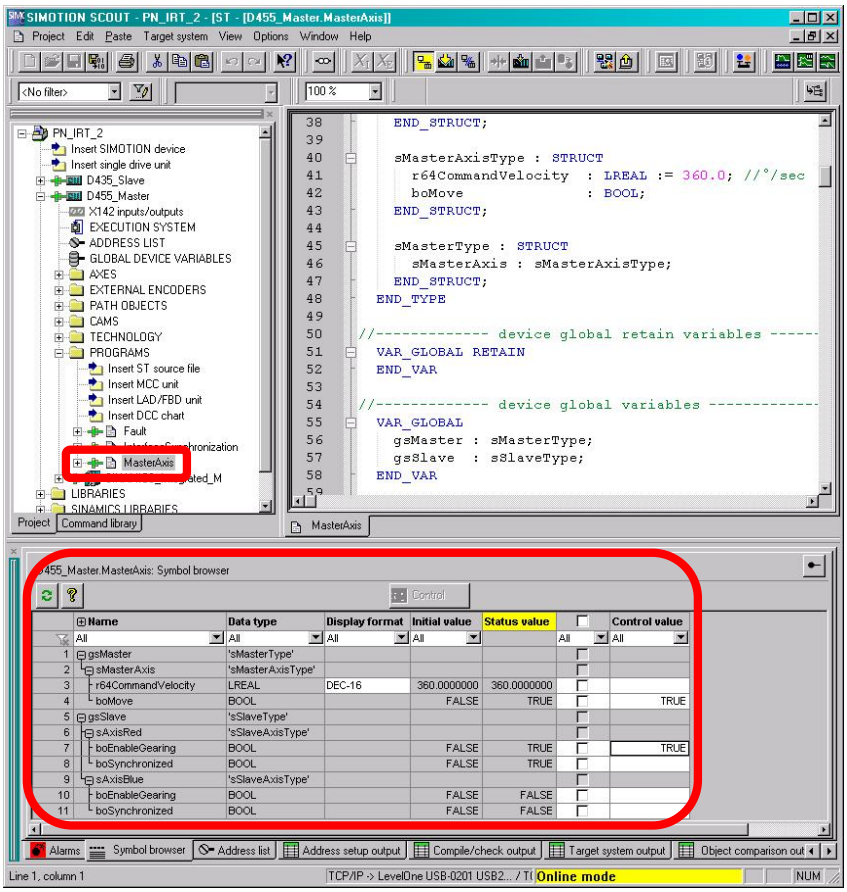


Table 6-1

No.	Instruction
1.	Go online
2.	Both controllers must be in the RUN state.
3.	Define the command velocity, e. g. `gsMaster.sMasterAxis.r64CommandVelocity = 360 °/sec`
4.	Enable and start virtual master axis `gsMaster.sMasterAxis.boMove = TRUE`
5.	Enable and synchronize slave axis red *SlaveAxisRed*`.gsSlave.sAxisRed.boEnableGearing = TRUE`
6.	The variable `boSynchronized` is set as feedback signal.
7.	Enable and synchronize slave axis blue *SlaveAxisBlue*`.gsSlave.sAxisBlue.boEnableGearing = TRUE`
8.	The variable `boSynchronized` is set as feedback signal.

7 Further Notes, Tips and Tricks, etc.

7.1 Multiple use of an I-Device GSD file

Using an I-Device GSD file, several I-Device substitutes, with the same GSD file, can be connected to the higher-level I/O controller.

This means that when using a GSD file several times, each associated I-Device substitute on the I-Device side uses the same I/O addresses in order to communicate with the higher-level I/O controller. It goes without saying that at the higher-level I/O controller, the I/O addresses at the I-Device substitutes are different.

The multiple use of an I-Device GSD file is used to implement modular machine concepts and offers the following advantages:

1. On the one hand, the individual machine modules (I-Devices) internally operate with the same I/O addresses.
2. On the other hand, a machine module (I-Device) can be deactivated and activated by application as required (e.g. for partial commissioning purposes).

In order to do this, each machine module requires its own controller (e.g. SIMOTION controller) and the complete machine a higher-level controller (e.g. SIMATIC CPU, SIMOTION CPU).

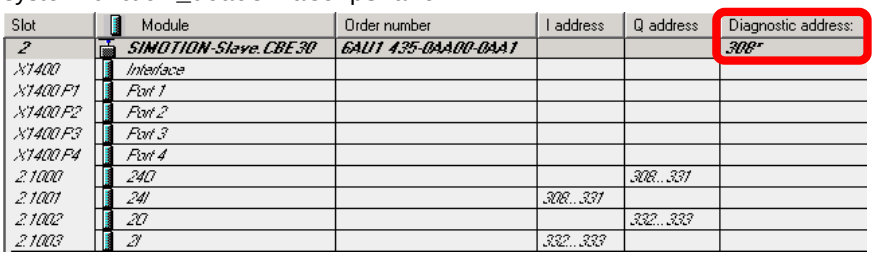
NOTE Further information about modular machines you will find in the SIMOTION document “Modular_Machine_Concepts.pdf”.

http://support.automation.siemens.com/WW/view/en/61056223

7.2 Deactivation of I-Devices by application

Table 7-1

No.	Instruction
1.	Open the HW configuration of the higher-level I/O controller and select the I-Device substitute afterwards.
2.	For the deactivation of an I-Device by application, its diagnostic address has to be assigned to the input parameter `logicalAddressOfStation` of the SIMOTION system function `_deactivateDpSlave`.



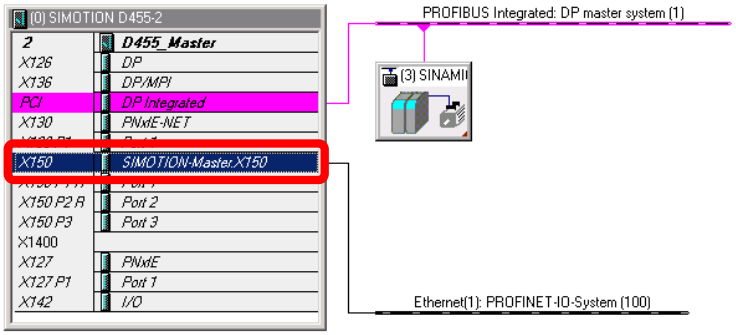
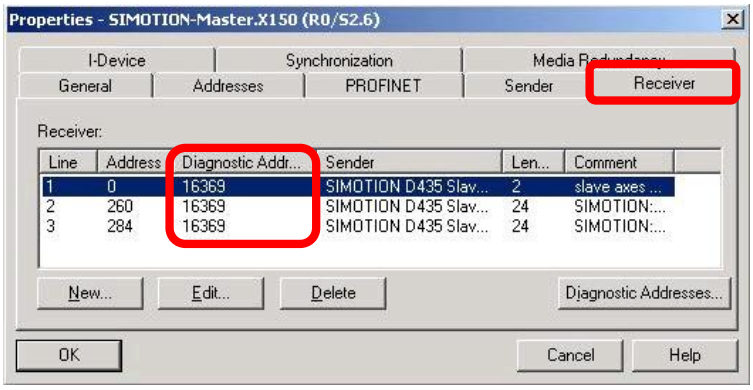
Slot	Module	Order number	I address	Q address	Diagnostic address:
2	SIMOTION-Slave.CBE30	6AU1 435-0AA00-0AA1			308*
X1400	Interface				
X1400 P1	Port 1				
X1400 P2	Port 2				
X1400 P3	Port 3				
X1400 P4	Port 4				
2.1000	240			308...331	
2.1001	241		308...331		
2.1002	20			332...333	
2.1003	21		332...333		

7.3 Deactivation of synchronous operation relationships by application

Beside the deactivation of whole machine modules (I-Devices) also individual synchronous operation relationships can be deactivated by application when direct data exchange broadcast is used.

Therefore there is the possibility to deactivate and activate unused SIMOTION CPUs participating on the direct data exchange broadcast, without getting any bus failure.

Table 7-2

No.	Instruction
1.	Open the HW configuration of the relative master CPU and afterwards the properties of the used PROFINET interface. 
2.	From the view of the master CPU all slave CPUs are displayed in the tab `Receiver`, with which synchronous operation data is exchanged. 
3.	For deactivating the desired synchronous operation relationship by application, the diagnostic address of the appropriate slave CPU has to be used. It has to be assigned to the input parameter `logicalAddressOfDpStation` of the SIMOTION system function `_deactivateDpSlave`.

NOTE Further information about the SIMOTION system function _deactivateDpSlave you will find in the SIMOTION online help as well as in the SIMOTION document “RefList_Device.pdf”.

http://support.automation.siemens.com/WW/view/en/61230890

8 Related literature

8.1 Bibliography

This list is not complete and only represents a selection of relevant literature.

Table 8-1

	Subject	Title
/1/	SIMOTION	Gear_TechnologyFct.pdf http://support.automation.siemens.com/WW/view/en/61055960
/2/	SIMOTION	Modular_Machine_Concepts.pdf http://support.automation.siemens.com/WW/view/en/61056223
/3/	SIMOTION	Communication.pdf http://support.automation.siemens.com/WW/view/en/61055999

8.2 Internet link specifications

This list is not complete and only represents a selection of relevant information.

Table 8-2

	Subject	Title
\1\	Direct link to internet document	http://support.automation.siemens.com/WW/view/en/38486079
\2\	Siemens Industry Online Support	http://support.automation.siemens.com

9 Contact

Siemens AG

Industry Sector
I DT MC PMA APC
Frauenauracher Straße 80
D - 91056 Erlangen
mailto: profinet.team.motioncontrol.i-dt@siemens.com

10 History

Table 10-1

Version	Date	Modifications
V1.1	05/2007	Publication
V1.2	08/2007	Correction, References
V2.0	11/2010	New release with new hardware and firmware
V2.1	01/2013	New revision / implementation of IRT I-Device
V2.2	09/2013	New revision / implementation of chapter “Further Notes, Tips and Tricks, etc.”