



PROFINET

Integration of 4D*pro* devices into PLC

SICK
Sensor Intelligence.

SICK AG
September 2021

- PROFINET portfolio of identification sensors
- Basic handling on PLC side
 - ▶ Installing the GSDML file
 - ▶ Adding sensors
 - ▶ Assigning PROFINET names
- Handling at SOPAS ET (optional)
- Data channel / using Sick Function Block
- Trouble shooting / Further documentation



PROFINET PORTFOLIO IDENTIFICATION SENSORS



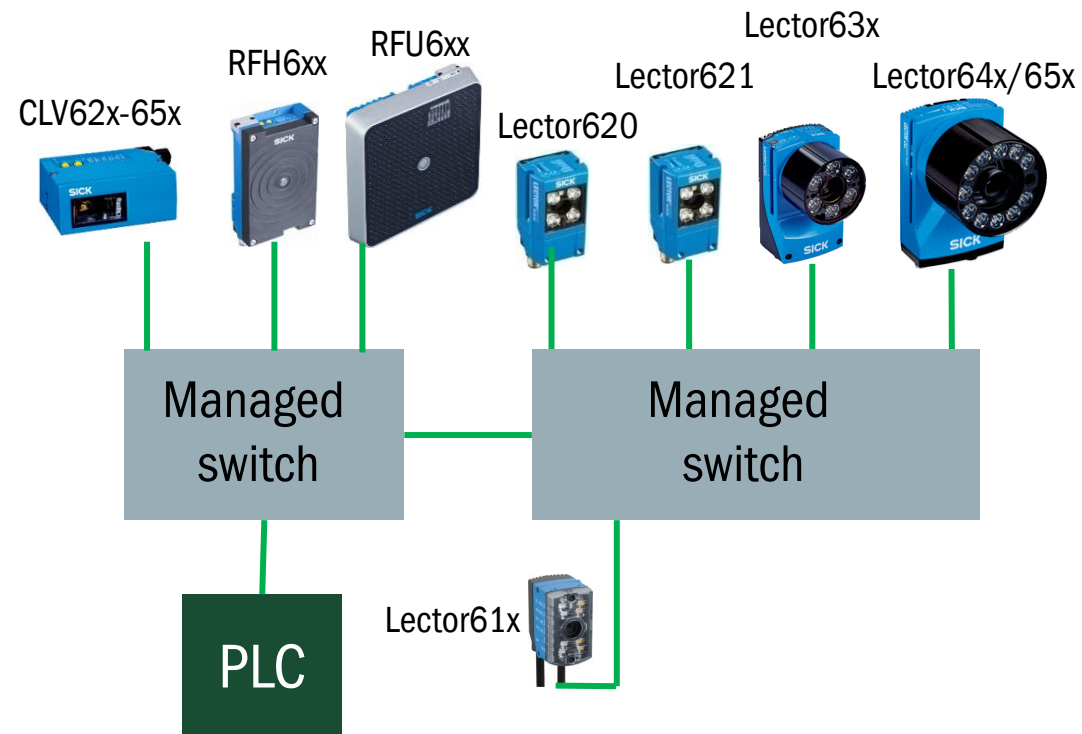
SICK
Sensor Intelligence.

PROFINET on board

SINGLE PORT – DUAL PORT

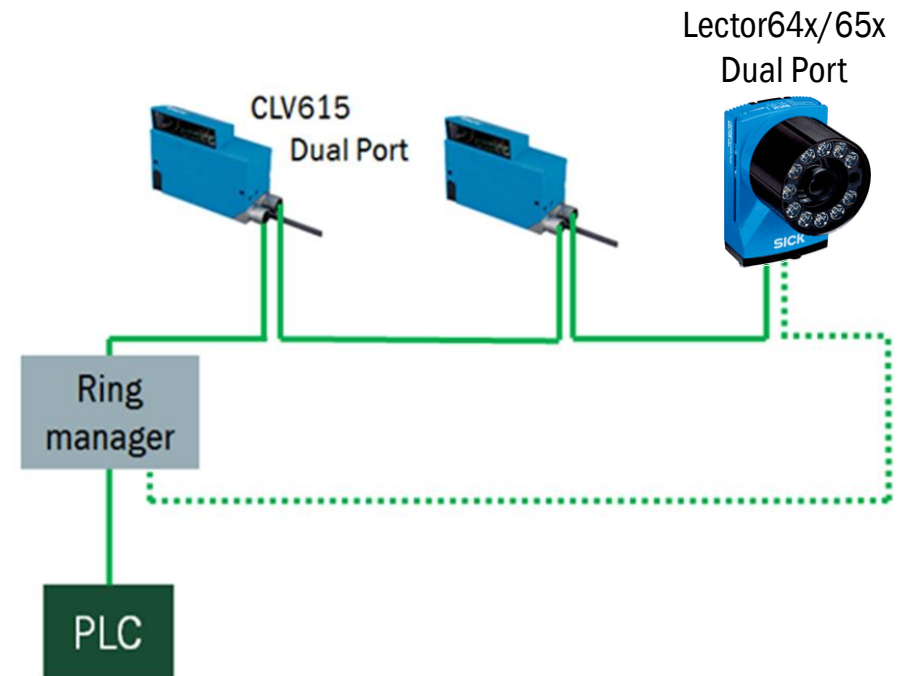
Single port

Star Topology



Dual port

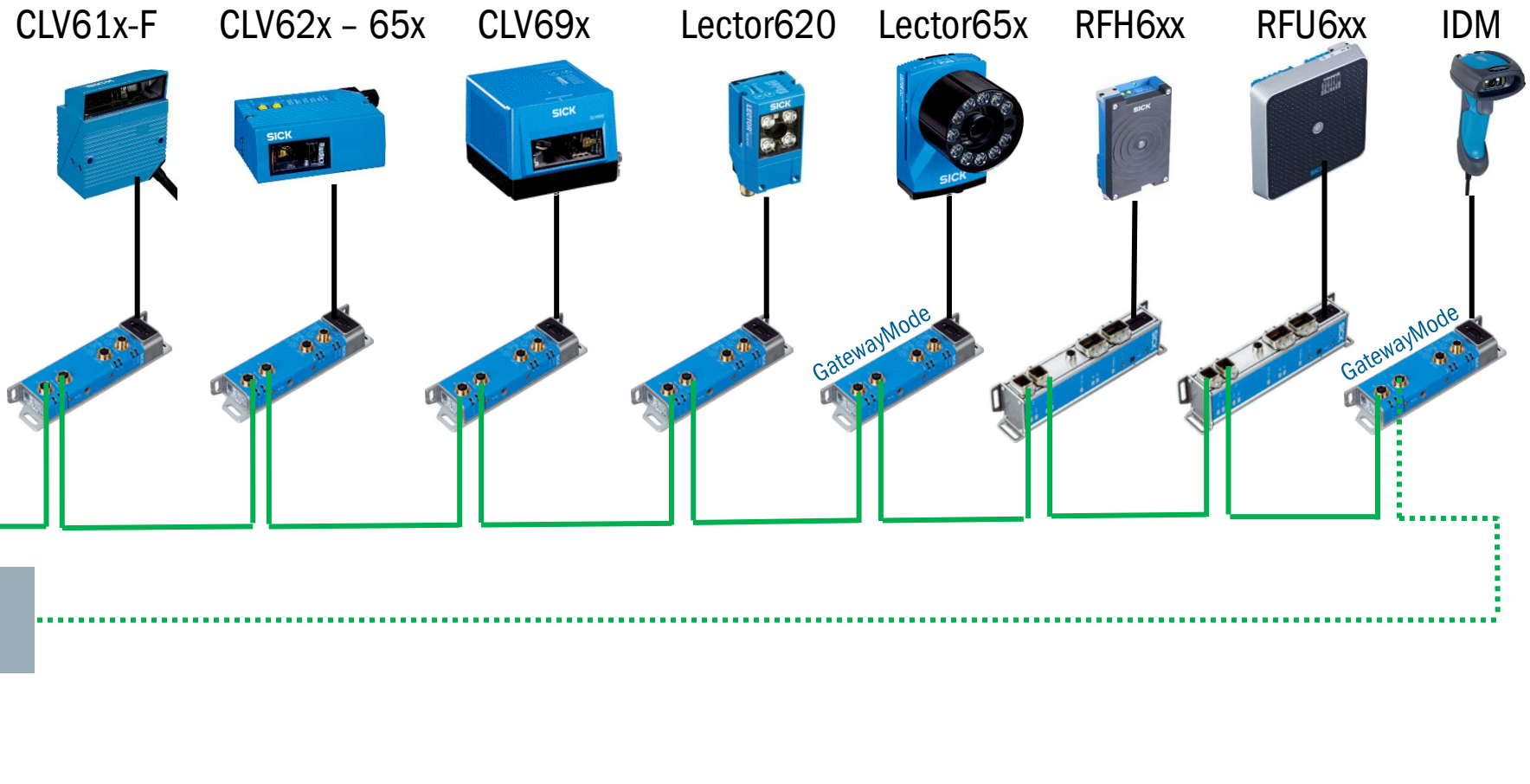
Line/Ring Topology



To run PROFINET on board as 1-port use the Ethernet version of 4Dpro devices

PROFINET via external module

DUAL PORT



Support of further devices on request

HANDLING ON PLC SIDE

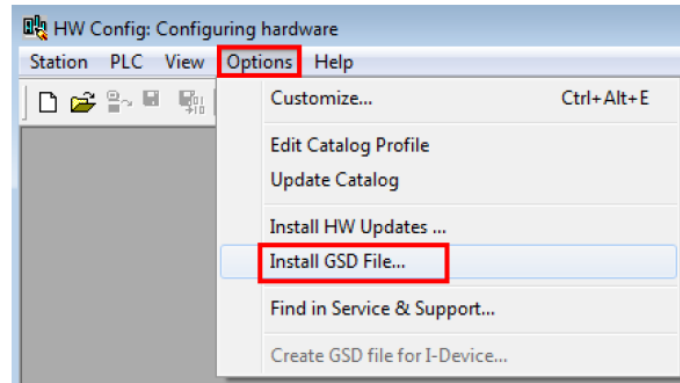


02

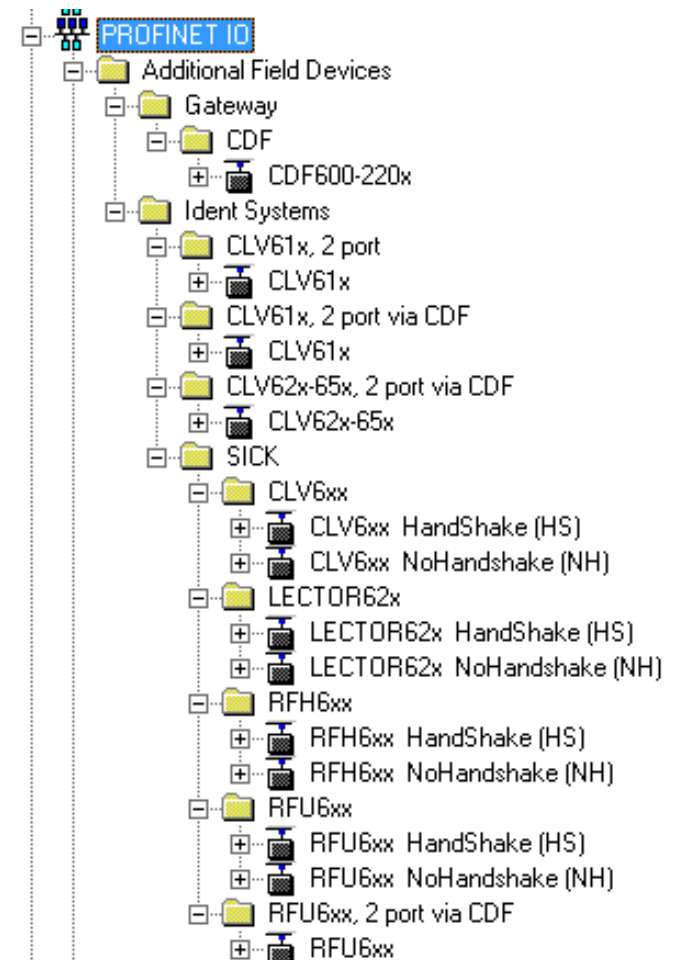
INSTALLING THE GSD FILE

HW-CONFIG (1)

- Install GSDML file at „Options / Install GSD File ...“ and update the catalog



- The 4Dpro devices can be found in the catalog:
 - ▶ PROFINET IO / Additional Field Devices / Gateway/...
 - CDF600-22xx using gateway mode
 - ▶ PROFINET IO / Additional Field Devices / Ident Systems / ...
 - Sensor single port
 - CLV62x-65x, LECTOR6xx, RFH6xx, RFU6xx
 - Sensor dual port
 - CLV61x-D
 - Sensor dual port via CDF600-22xx (proxy mode)
 - CLV61x, CLV62x-65x, RFU6xx

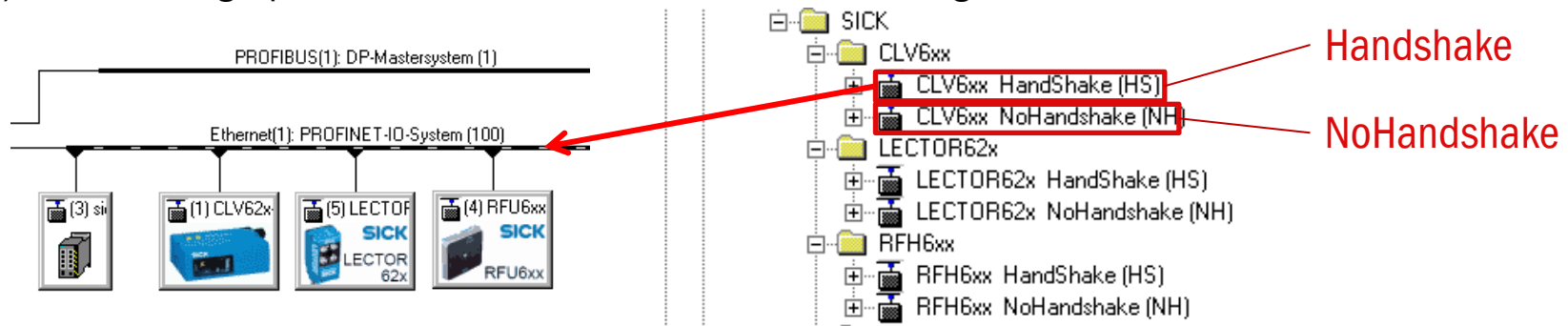


INSTALLING THE GSD FILE

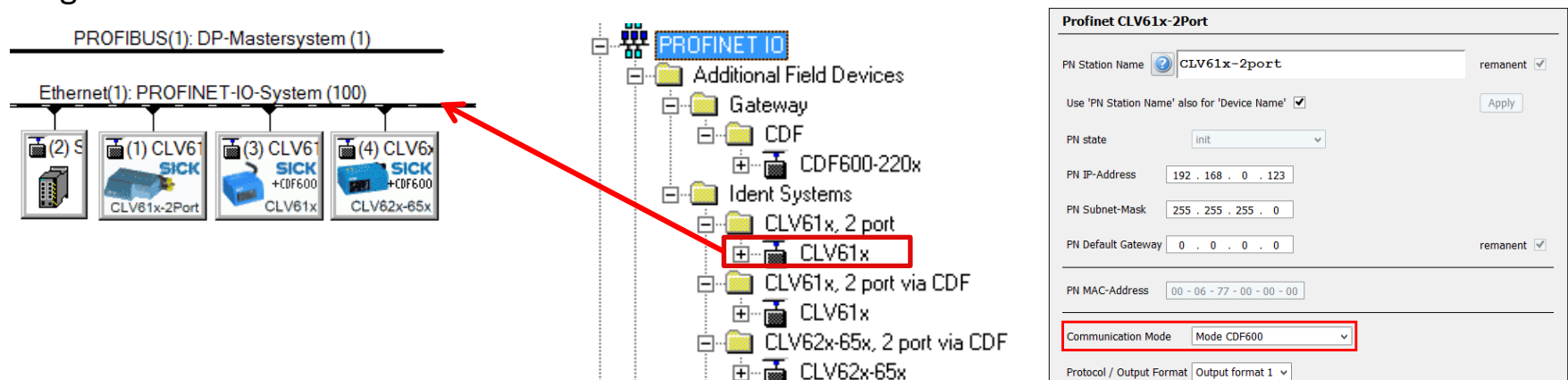
HW-CONFIG (2) → HANDSHAKE SELECTION

- Insert the required device either as Handshake (HS) or NoHandshake (NH) by using drag & drop on the PROFINET line
 - Dependent on sensor:

A) PROFINET single port: Choose one of the two lines in the tree to get either Handshake or NoHandshake mode



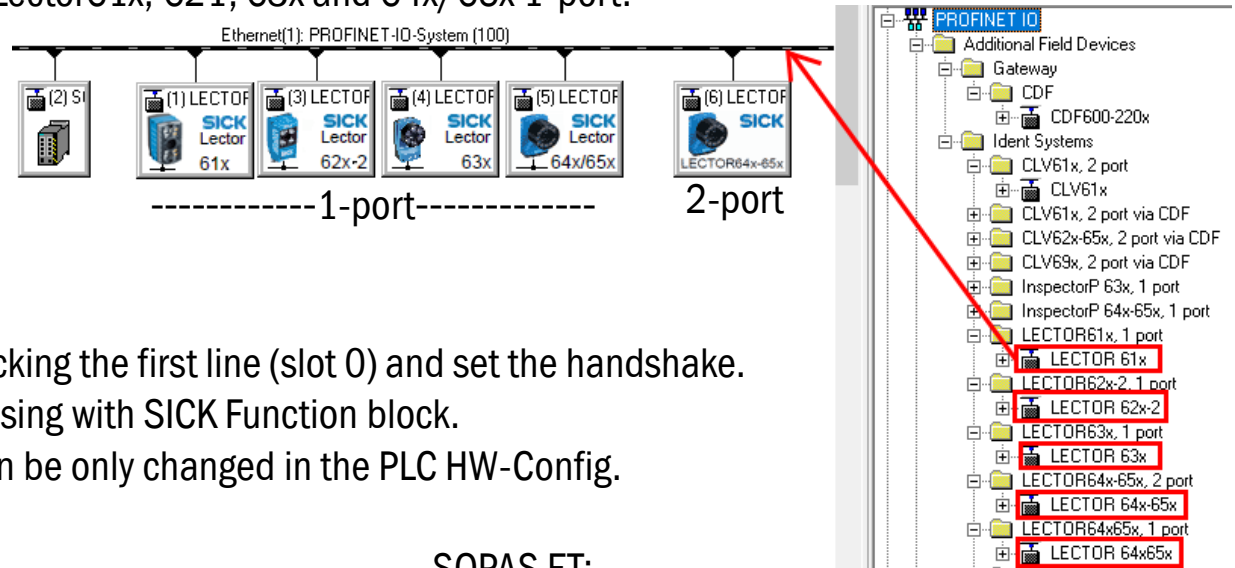
B) PROFINET dual port: Insert the shown device. The handshake setting is done in the device parameterization, or using a GSD Parameterization module. Default is with handshake.



INSTALLING THE GSD FILE

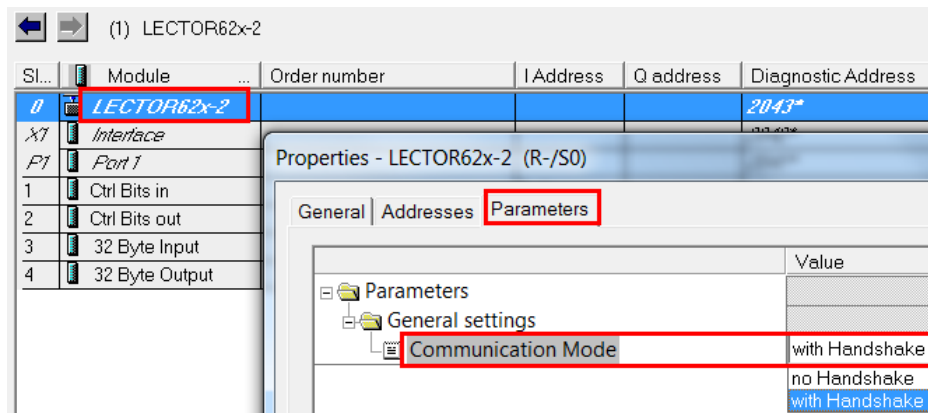
HW-CONFIG (3) → HANDSHAKE SELECTION

C) PROFINET at Lector64x/65x 2-port and Lector61x, 621, 63x and 64x/65x 1-port:

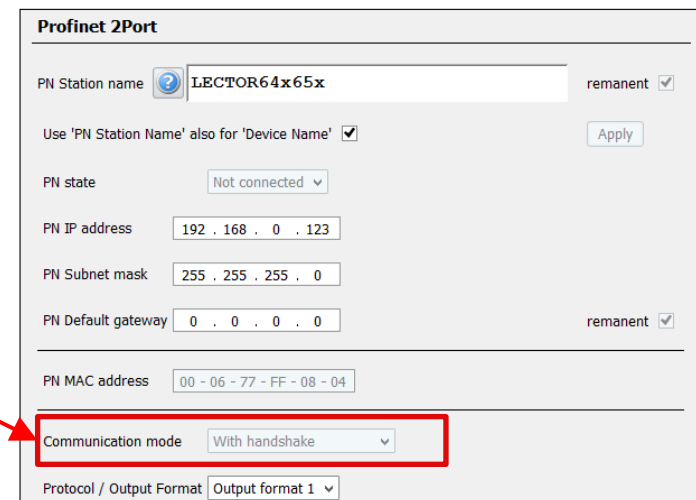


The handshake setting is done by double clicking the first line (slot 0) and set the handshake. Default is with handshake, which is fine for using with SICK Function block. The setting is also shown in SOPAS, but it can be only changed in the PLC HW-Config.

PLC HW-Config:



SOPAS ET:

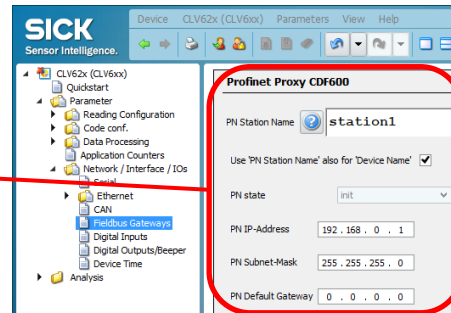


PROFINET GSD FILES FOR CDF600 PROXY MODE

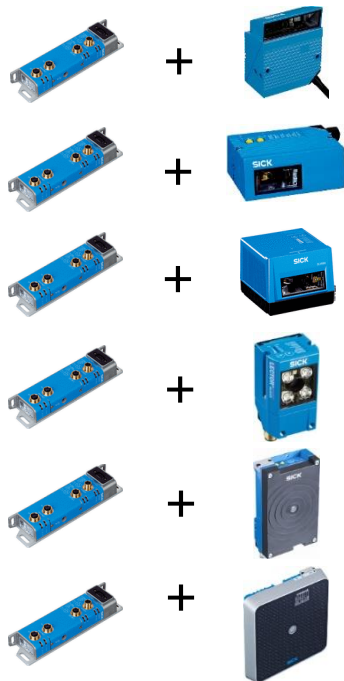
FOR CDF V1.20/1.30 (OLD HW) AND V2.00 (NEW HW)

Mode = 0 → Proxy-Mode:

- CDF integrated in SOPAS
- Trigger via ExtIn1
- Trigger via Trigger bit
- Trigger via Command (via Function Block)



These GSD Files can be used at CDF firmware V1.20/1.30 (old hardware) and V2.00 (new hardware). CDF with V2.00 (new hardware) can be used in existing installations.



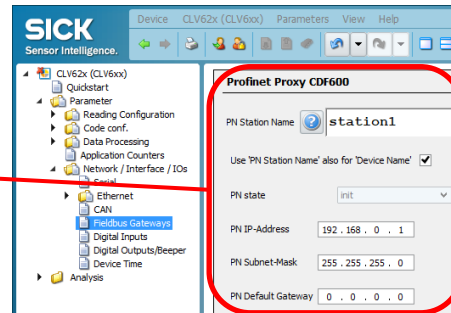
Device	GSD-File: „via CDF“
CLV61x-Fxxx / V1.21 or higher	GSDML-V2.3-SICK-CLV61x_via_CDF600-20150312.xml
CLV62x-65x / V5.26 or higher	GSDML-V2.3-SICK-CLV62xCLV65x_via_CDF600-20150312.xml
CLV69x / V1.60 or higher	GSDML-V2.3-SICK-CLV69x_via_CDF-20150312.xml
Lector620 / V2.10	GSDML-V2.3-SICK-Lector62x_via_CDF-20150312.xml
RFH6xx / V3.11 or higher	GSDML-V2.3-SICK-RFH6xx_via_CDF600-20150312.xml
RFU6xx / V1.62 or higher	GSDML-V2.3-SICK-RFU6xx_via_CDF600-20150312.xml

PROFINET GSD FILES FOR CDF600 PROXY MODE

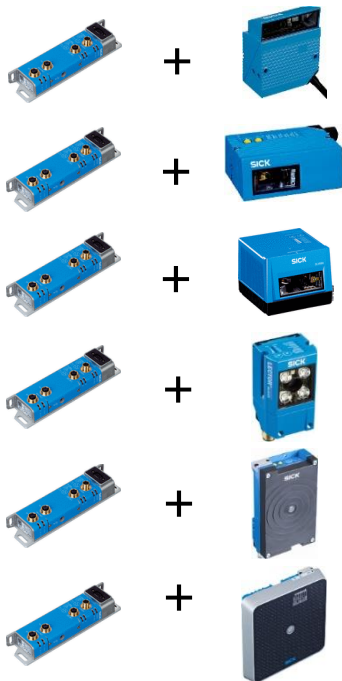
FOR CDF V2.00 (NEW HARDWARE)

Mode = 0 → Proxy-Mode:

- CDF integrated in SOPAS
- Trigger via ExtIn1
- Trigger via Trigger bit
- Trigger via Command (via Function Block)



These GSD Files can be used at CDF firmware V2.00 (new hardware) and higher.
These files are recommended for new installations.



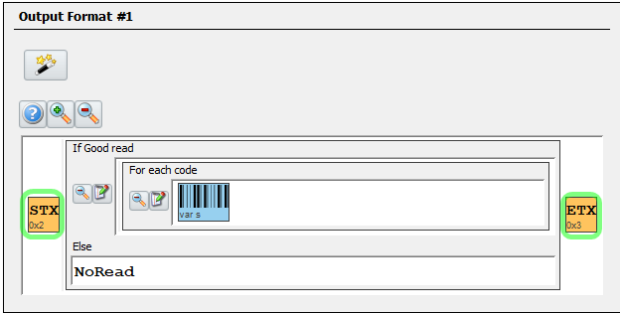
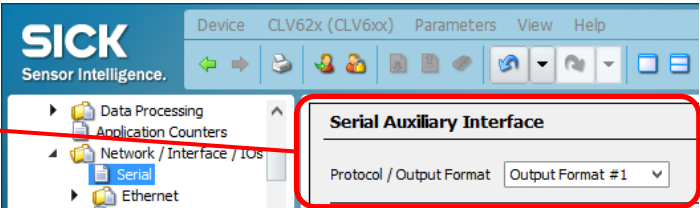
Device	GSD-File: „via CDF-V2“
CLV61x-Fxxx / V1.21 or higher	GSDML-V2.41-SICK-CLV61x_via_CDF600-20210706.xml
CLV62x-65x / V5.26 or higher	GSDML-V2.41-SICK-CLV62xCLV65x_via_CDF600-20210706.xml
CLV69x / V1.60 or higher	GSDML-V2.41-SICK-CLV69x_via_CDF-20210706.xml
Lector620 / V2.10	GSDML-V2.41-SICK-Lector620_via_CDF-20210706.xml
RFH6xx / V3.11 or higher	GSDML-V2.41-SICK-RFH6xx_via_CDF600-20210706.xml
RFU6xx / V1.62 or higher	GSDML-V2.41-SICK-RFU6xx_via_CDF600-20210706.xml

PROFINET GSD FILES FOR CDF600 GATEWAY MODE

FOR CDF V1.20/1.30 (OLD HW) AND V2.00 (NEW HW)

Mode = 2 → Gateway-Mode:

- Sensor via serial AUX connected, Pin 2+3
- Trigger via Command (via Function Block)



Output format must be STX / ETX framed

Can be used for any sensor with serial RS232 interface with 57600 Baud, 8, n, 1, STX/ETX framing. Depending on sensor additional wiring is needed.



CDF600-22xx
15-pin HD socket device:

External RS-232 device:

TxD, pin 2 ----->----- RxD
RxD, pin 3 ----<----- TxD
Gnd, pin 5 ----->----- Gnd



Device	GSD-File: CDF Gateway Mode
CDF600-22xx / V1.20/1.30 (old hardware) or V2.00 (new hardware)	GSDML-V2.3-SICK-CDF600-20150312.xml
CDF600-22xx / V2.00 or higher (new hardware)	GSDML-V2.41-SICK-CDF600-20210706.xml (recommended for new installations)

PROFINET ON BOARD GSD FILES AT LECTOR6XX FAMILY

ICON – GSD FILE - DIRECTORY



Lector61x: >= V4.0.0 / GLS611: >= V4.3.0

SICK_Lector61x_1P_Profinet_minV4.0.0:
GSDML-V2.34-SICK-Lector61x_1P-20191122.xml

HW Catalog: Profinet\Additional devices\Ident systems:

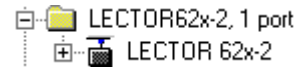


GSD-Parameterization
not available

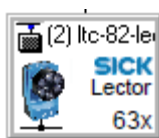
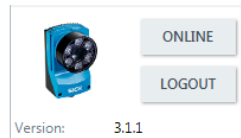


Lector621: >= V3.1.1 / GLS621: >= V3.2.0

SICK_Lector62x-2_1P_Profinet_minV3.1.1:
GSDML-V2.34-SICK-Lector62x-2_1P-20190118.xml



GSD-Parameterization
not available

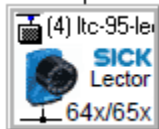
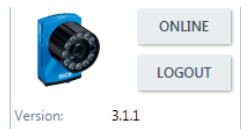


Lector63x: >= V3.1.1

SICK_Lector63x_1P_Profinet_minV3.1.1
GSDML-V2.34-SICK-Lector63x_1P-20190118.xml

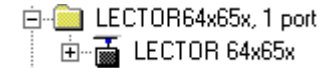


GSD-Parameterization
not available

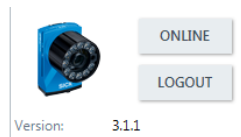


Lector64x/65x: >= V3.1.1

SICK_Lector64x65x_1P_Profinet_minV3.1.1
GSDML-V2.34-SICK-Lector64x65x_1P-20190118.xml

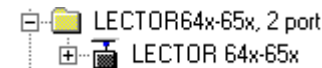


GSD-Parameterization
not available



Lector64x/65x 2port:

SICK_Lector64x65x-2port_Profinet_minV142
GSDML-V2.31-SICK-Lector64x65x-20151103.xml

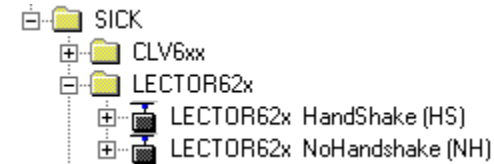


GSD-Parameterization
not available



Lector620 V2.10:

SICK_Lector62x_Profinet_minV210
GSDML-V2.3-SICK-Lector62x-20140929.xml

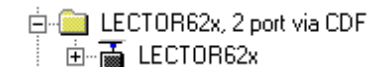


GSD-Parameterization
available



Lector620 V2.10:

SICK_Lector62x_Profinet_via_CDF_minV210
GSDML-V2.3-SICK-Lector62x_via_CDF-20150312.xml



GSD-Parameterization
available

see sick.com for latest GSD files

PROFINET 1PORT AT LECTOR6XX FAMILY

Lector61x/621:



Profinet on P3: 4 pin, M12, D-coded

Profinet 1-port (P3)

Profinet enabled ☒

Profinet

PN Station name remanent ☐

Use 'PN Station Name' also for 'Device Name' ☒

PN state

PN IP address

PN Subnet mask

PN Default gateway remanent ☐

PN MAC Address

Communication mode

Protocol / Output Format

Plant designation

Location designation

Installation date

Additional information

Lector63x:



Profinet on P3: M12, 8-pin, X-coded

Profinet 1-port (P3)

Profinet enabled ☒

Profinet

PN Station name remanent ☐

Use 'PN Station Name' also for 'Device Name' ☒

PN state

PN IP address

PN Subnet mask

PN Default gateway remanent ☐

PN MAC Address

Communication mode

Protocol / Output Format

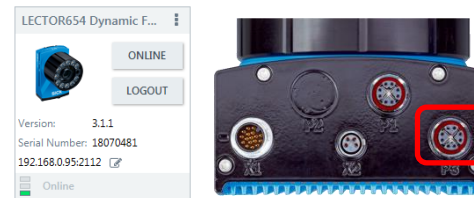
Plant designation

Location designation

Installation date

Additional information

Lector64x/65x:



Profinet on P3: M12, 8-pin, X-coded

Profinet 1-port (P3)

Profinet enabled ☒

Profinet

PN Station name remanent ☐

Use 'PN Station Name' also for 'Device Name' ☒

PN state

PN IP address

PN Subnet mask

PN Default gateway remanent ☐

PN MAC Address

Communication mode

Protocol / Output Format

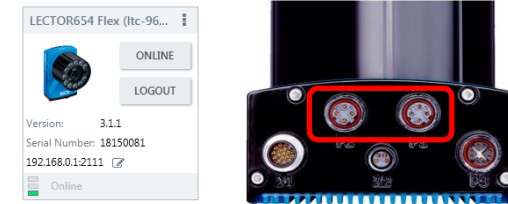
Plant designation

Location designation

Installation date

Additional information

Lector64x/65x-2port:



Profinet on P1/2: M12, 4-pin, D-coded

Profinet 2-port (P1:P2)

Profinet

PN Station name remanent ☒

Use 'PN Station Name' also for 'Device Name' ☒

PN state

PN IP address

PN Subnet mask

PN Default gateway remanent ☐

PN MAC Address

Communication mode

Protocol / Output Format

Plant designation

Location designation

Installation date

Additional information

At Lector64x/65x 1port P1 must be set to different Subnet as P3, also when not used.

Lector64x/65x 2port: Profinet fix selected

LECTOR 64x/65x DUAL PORT PROFINET ON BOARD

INTERFACE OPTIONS



Lector64x/65x with Dual Port Profinet on board:

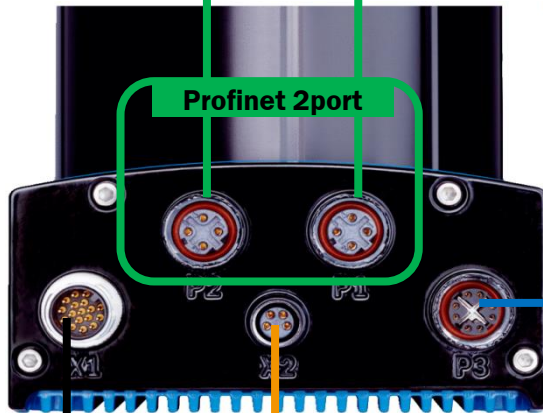
1071472 V2D642R-MCXXH6

1075975 V2D652R-MCXXH6

1083896 V2D654R-MCXXH6



Profinet, 2 Ports (P1, P2)



Ethernet (P3) Gigabit

USB - SOPAS (X2)

Serial Aux + Host, Dig. IO, Power (X1)



Dual Port Profinet
for Result Data and Command handling,
e.g. via SICK Function Block PN/DP



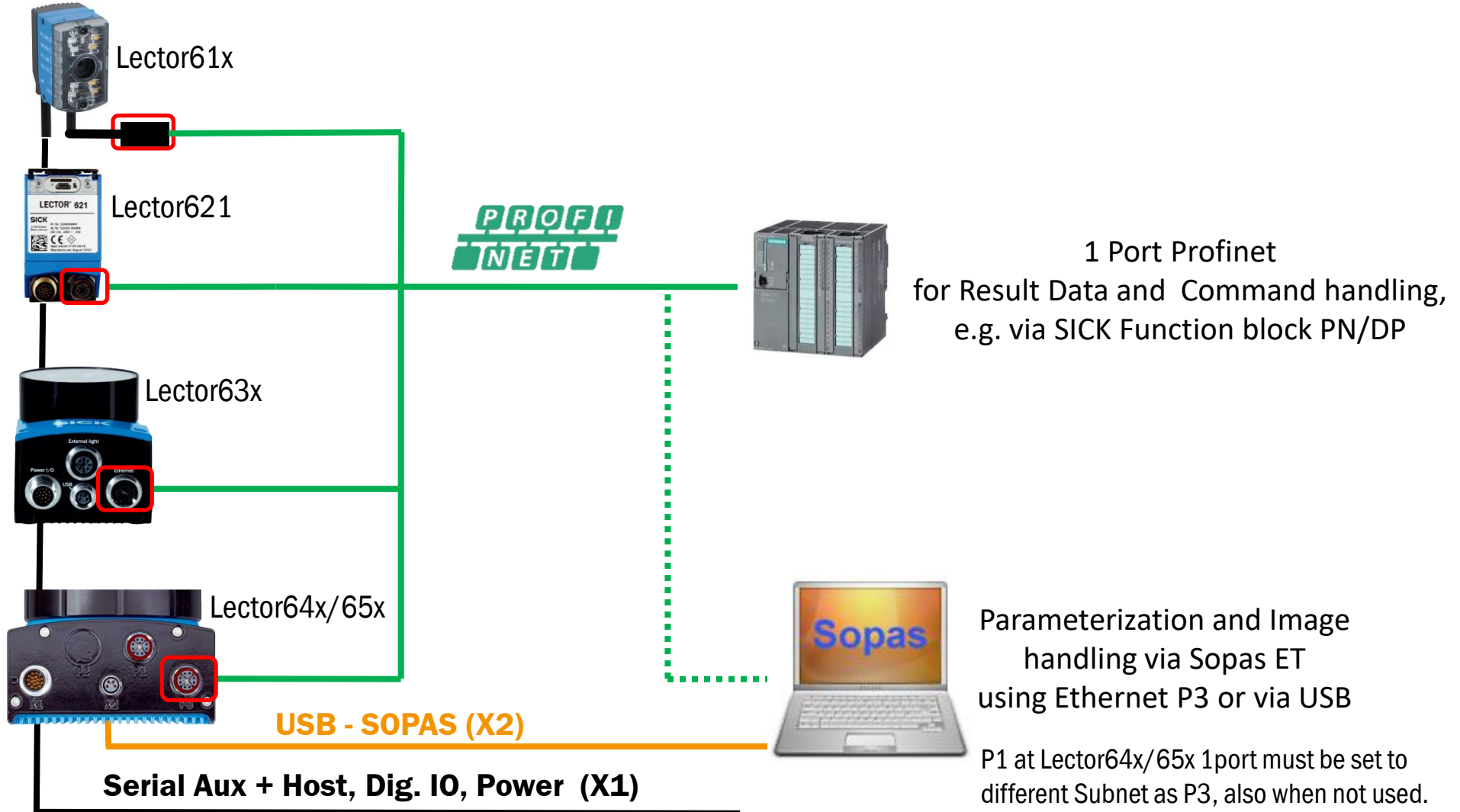
Parameterization and Image
handling via Sopas ET,
only via Ethernet P3 or USB

P3 at Lector64x/65x Dual Port can be set to
same Subnet as Profinet P1/P2, when needed.
P3 must have then different IP as the Profinet
port P1/P2.

LECTOR 61X, 621-65X 1 PORT PROFINET ON BOARD

INTERFACE OPTIONS

Lector61x with V4.0.0 and Lector621, 63x, 64x, 65x with V3.1.1 or higher: Profinet, 1 Port on P3



CDF IN GATEWAY MODE - AS ALTERNATIVE TO CONNECT ANY 4DPRO SENSOR TO PROFINET



As alternative, or when Profinet is not integrated, or 2port is needed, any 4Dpro sensor with Serial Auxport can be connected using CDF600-22xx in Gateway mode to Profnet.

Set Mode switch to 2 and use GSD from CDF600-22xx.

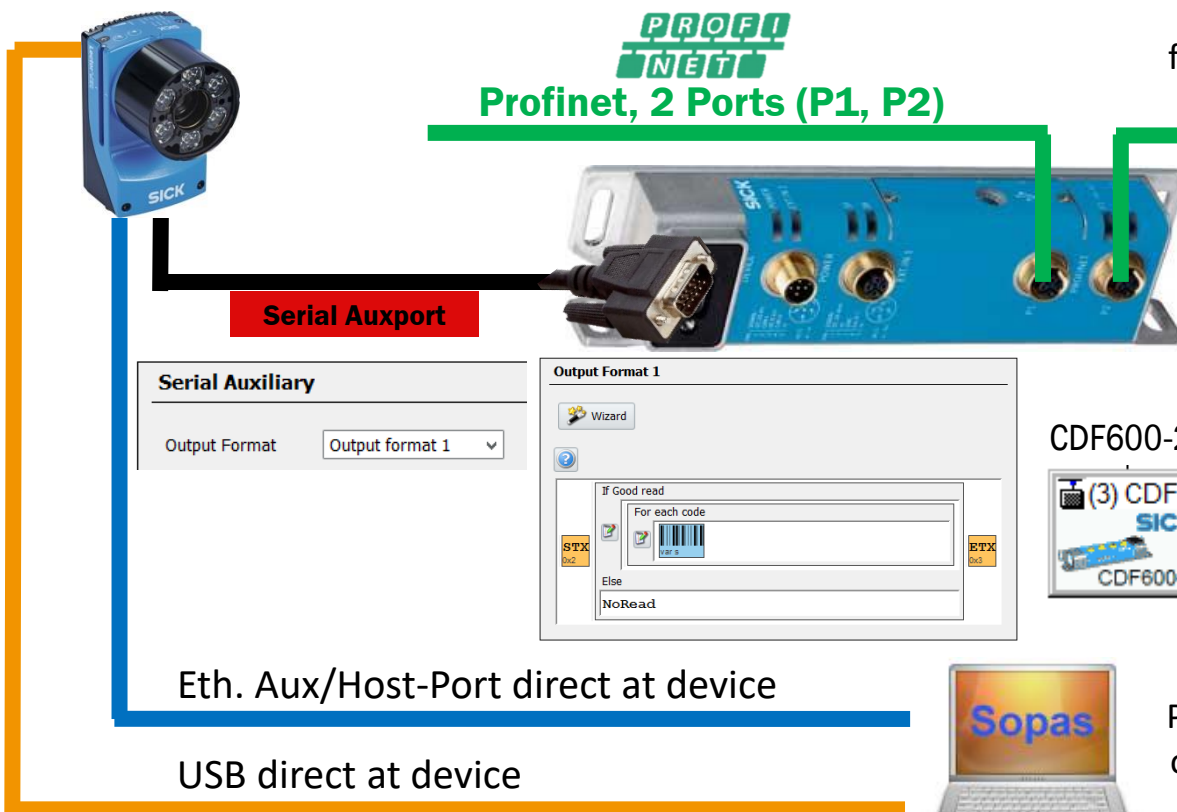
Ser Aux must be set to Output format 1 using an STX ETX framing.

Trigger is only possible by SOPAS command, as easiest using SICK Function Block.

**PROFI
NET**
Profnet, 2 Ports (P1, P2)

Dual Port Profinet
for Result Data and Command handling,
e.g. via Sick Function Block PN/DP

➔ Trigger only via SOPAS command,
e.g. via Sick Function Block

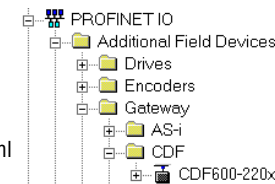


HW Catalog: Profinet\Additional devices:

CDF600-22xx Gateway Mode (switch = 2)



SICK_CDF600-220x_Profinet_minV120
GSDML-V2.3-SICK-CDF600-20150312.xml or
GSDML-V2.41-SICK-CLV61x_via_CDF600-20210706.xml
see sick.com for latest GSD files



Parameterization and Image handling via SOPAS ET
only when connected **directly** to the SOPAS device
using Ethernet or USB.



USING HANDSHAKE OR NO-HANDSHAKE

GUIDELINE

■ **Handshake Mode:** recommended

- ▶ SICK function block recommended
- ▶ Can send and receive also longer telegrams
- ▶ Flow control via handshake (Confirmed Messaging protocol)
- ▶ Trigger options: **
 - Fieldbus bit (Ctrl-Bits Out)
 - Hardware input at the sensor
 - Command

Recommended when long data telegrams must be received, or when sending of commands is needed.

For RFU6xx and RFH6xx it is in general recommended to use SICK function blocks. These function block allows to use powerful read and write tag functions.

■ **No Handshake Mode:**

- ▶ SICK function block not allowed (no program for handshake needed)
- ▶ Can handle only short telegrams which fits inside the “xx Byte input” module range
- ▶ Can be only used to receive single telegrams (no flow control)
- ▶ Trigger options: **
 - Fieldbus bit (Ctrl-Bits Out)
 - Hardware input at the sensor

Recommended when just triggering and data receiving and no command sending to the sensor is needed, e.g. at CLV6xx and Lector6xx

** Limited trigger options by using CDF600-2 gateway mode

EXAMPLE FOR NO-HANDSHAKE – CLV61X-DUAL PORT USING DIRECT DATA ACCESS

HW-Config: Insert device and use GSD Parametrization to set No-Handshake and trigger via Fieldbus

HW Config Table:

Slot	Module	Order number	I address	Q address
0	CLV61x-2Port			
X1	Interface			
X1 A	Port 1			
X1 A	Port 2			
1	Ctrl Bits in		0...1	
2	Ctrl Bits out			0...1
3	32 Byte Input		256...287	
4	32 Byte Output			256...287
5	01_Start Remote Config ~			
6	47_Communication Mode			
7	99_End Remote Config <<~			

General parameters:

- General parameters: Load Defaults, Codabar, Code 39, UPC / EAN / GTIN, 2/5 Interleaved, Code 93, Code 128, Pharma, Start Object Trigger, GS1 DataBar
- General parameters: Application Defaults
- General parameters: Communication Mode: CDF600 no Handshake
- General parameters: End of Remote Config: Don't save parameters perm.

Result data:

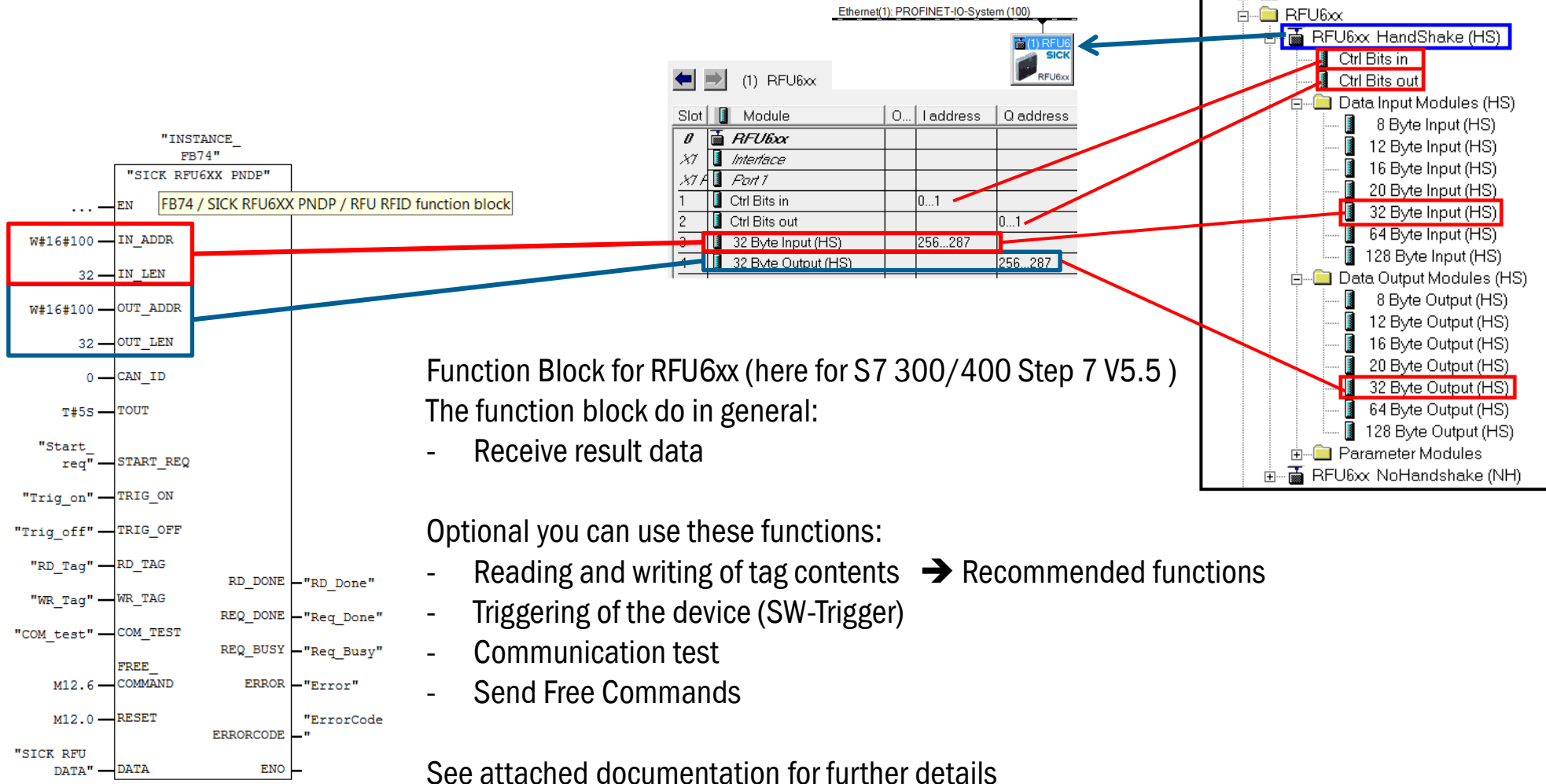
Address	Symbol	Display format	Status value	Modify val
1	// Trigger using Fieldbus-Bit			
2	Q 1.0 "2_Trigger"	BOOL	true	
4	// Receive Count + Data:			
5	IB 256 "Input Bits / Heartbeat"	BIN	2#0000_0100	
6	IB 257 "Receive_Count"	DEC	4	
7	IB 258 "Transmit_Count"	DEC	0	
8	IB 259 "Receive-Length-Low"	DEC	8	
9	IB 260 "Receive-Length-High"	DEC	0	
10	IB 261 "Rec-Data, Byte 1"	CHARACTER	'1'	
11	IB 262 "Rec-Data, Byte 2"	CHARACTER	'2'	
12	IB 263 "Rec-Data, Byte 3"	CHARACTER	'3'	
13	IB 264 "Rec-Data, Byte 4"	CHARACTER	'4'	
14	IB 265 "Rec-Data, Byte 5"	CHARACTER	'5'	
15	IB 266 "Rec-Data, Byte 6"	CHARACTER	'6'	
16	IB 267 "Rec-Data, Byte 7"	CHARACTER	'7'	
17	IB 268 "Rec-Data, Byte 8"	CHARACTER	'8'	
18	IB 269 "Rec-Data, Byte 9"	CHARACTER	B#16#00	
19	IB 270 "Rec-Data, Byte 10"	CHARACTER	B#16#00	

Trigger: Output Bit Q1.0 triggers the scanner:
1 = Trigger on, 0 = Trigger off

Result data:
The user data are visible in the area outlined in blue.
The "Receive Count" counter increments with each block of data received.
On overflow, the 0 is skipped: 1 to 255, 1 to 255, ...
In the example, eight data bytes were received, with the content "12345678".
No handshake program is needed in this No-Handshake mode.

EXAMPLE FOR HANDSHAKE WITH RFU630 USING SICK S7 FUNCTION BLOCK (NEED HANDSHAKE MODE)

HW-Config: Insert device in handshake mode and use parameterization
via SOPAS ET to be flexible for fine tuning local at the device



ADDING SENSORS

HW-CONFIG / SET DEVICE NAME

- Select the **Device name** for indentifying the device in PROFINET by double clicking on the device icon
- SOPAS ET also displays this name when scanning the network

- The **IP address** can be controlled by the PLC (IO controller). By default this is active.

The PLC sets the IP address in the so called „temporary“ mode. This means it is not allowed for the device to save this IP address. At next power cycle it must start with IP 0.0.0.0. The PLC will set the IP address again.

Properties - CLV61x

General | Identification

Short description: CLV61x
CLV61x - Barcodescanner; 2 port PROFINET interface via CDF600; GSDML V2.3; Firmware V2.0.0

Order no./ firmware: / V2.0

Family:

Device name: CLV61x

GSD file: GSDML-V2.3-SICK-CLV61x_via_CDF600-20131029.xml
Change Release Number...

Node in PROFINET IO system

Device number: 1 PROFINET-IO-System (100)

IP address: 192.168.0.51 Ethernet...

☒ Assign IP address via IO controller

Comment:

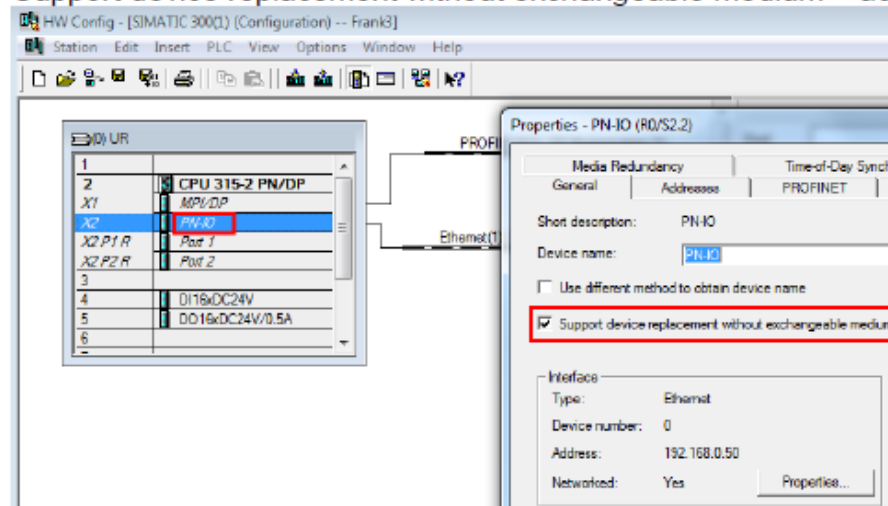
OK Cancel Help

ADDING SENSORS

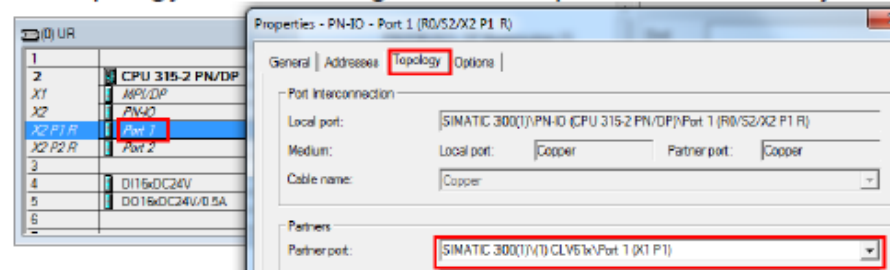
HW-CONFIG / SET PROFINET NAME AUTOMATICALLY

The PN name can be automatically assigned by the PROFINET IO Controller (PLC), if:

- a) This has been activated in the CPU (is activated by default):
Support device replacement without exchangeable medium = active



- b) The topology has been taught-in. A PN partner is then firmly assigned to each used port.



- c) The switches used must support this function, e.g., Scalance X200 series.

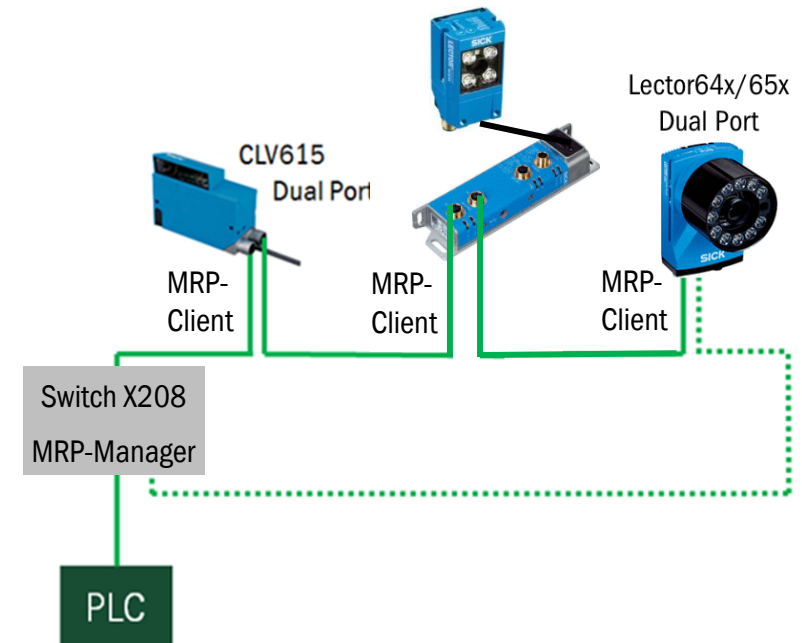
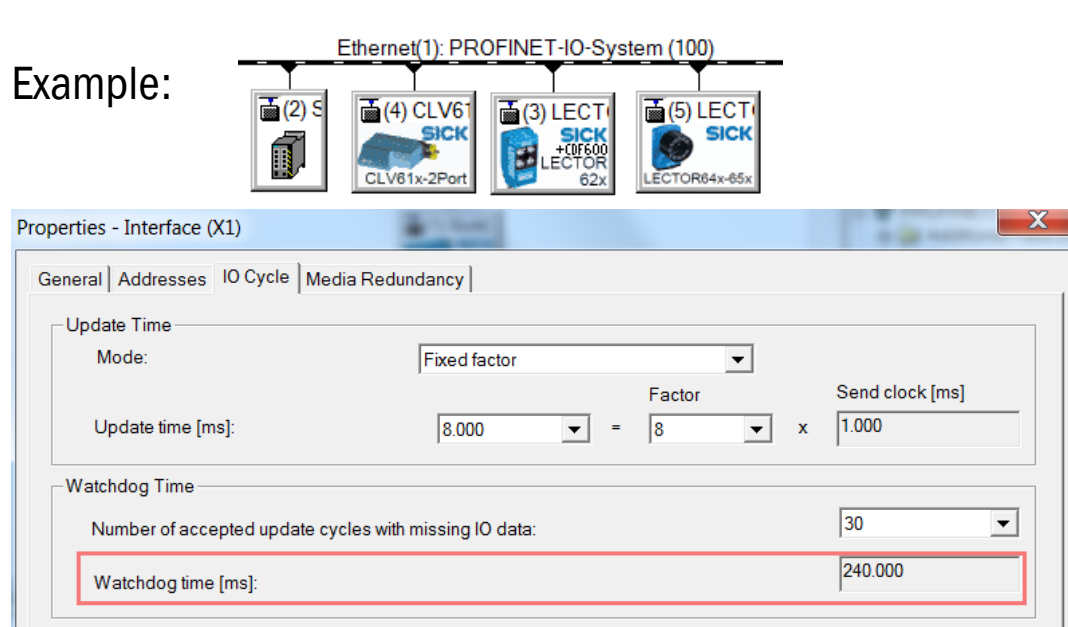
If a PN device is connected with an **empty name** and its type matches, the PN name stored on the PROFINET IO Controller (PLC) is assigned.

SETUP OF A RING TOPOLOGY BASED ON “MRP”

HW-CONFIG / RING PROTOCOL - MRP

- In an MRP Ring one device has to be MRP-Manager (e.g. switch). The other devices must be MRP-Clients.
- The reconfiguration time for MRP is 200 ms. With the MRP protocol, it is possible to detect and to reconfigure a defective ring within 200 ms (for up to 50 nodes).
- Accordingly, a watchdog time of > 200 ms has to be selected for each PROFINET device in the ring.
- This can be reached by increasing the update time or the accepted update cycles without IO data.
- For further details see e.g. Siemens Application example Entry ID: [109739614](#)

Example:



ADDITION OF DATA MODULES

HW-CONFIG / NECESSARY REQUIREMENT

- Add one input and one output module. The size can be selected (data size = 8 ... 128 bytes)
Using handshake mode up to 4.000 byte data telegrams can be handled
- The HW config must contain 4 modules (not less): Ctrl-Bits In/Out and Data Byte In/Out
- Depending on device type these 4 modules are automatically inserted when the device is inserted in PROFINET. If not insert these 4 modules by using drag & drop

The screenshot displays the SIMATIC Manager HW Config interface. The top window shows the rack configuration with slots 1-6. Slot 2 contains the CPU 315-2 PN/DP. Slot 3 contains the SICK CLV61x module. The bottom window shows the module configuration table. Red arrows point from the module configuration table to the module selection tree on the right.

Slot	Module	Order number	I address	Q address	Diagnostic address:	Comment
0	CLV61x				2042	
X1	Interface				2042	
X1 P1	Port 1				2042	
X1 P2	Port 2				2042	
1	Ctrl Bits in		0...1			
2	Ctrl Bits out			0...1		
3	32 Byte Input		256...287			
4	32 Byte Output			256...287		
5						
6						
7						

The module selection tree on the right shows the following structure:

- PROFIBUS-PA
- PROFINET IO
 - Additional Field Devices
 - Gateway
 - Ident Systems
 - CLV61x, 2 port via CDF
 - CLV61x
 - Ctrl Bits in
 - Ctrl Bits out
 - Data Input Modules
 - 8 Byte Input
 - 12 Byte Input
 - 16 Byte Input
 - 20 Byte Input
 - 32 Byte Input
 - 64 Byte Input
 - 128 Byte Input
 - Data Output Modules
 - 8 Byte Output
 - 12 Byte Output
 - 16 Byte Output
 - 20 Byte Output
 - 32 Byte Output
 - 64 Byte Output
 - 128 Byte Output
 - Parameter Modules

CTRL-BITS-IN @ PROFINET ON BOARD

HW - CONFIG / NECESSARY REQUIREMENT / CTRL-BITS-IN / INPUTS AT PLC

By using the Ctrl-Bits-In, the PLC can monitor the digital inputs and outputs of the ID sensor.

For logical assignment of the Ctrl-Bits-In see below table. The available I/O hardware can vary depending on the type of ID sensor.

Address	Bit	Name	Meaning	CLV62x-65x	Lector61x	Lector620	Lector621-65x	Lector64x-65x 2port	RFH6xx	RFU61x	RFU62x-65x
Add.+1	D0	Device Ready	Status of the attached ID sensor	X	X	X	X	X	X	X	X
Add.+1	D1	System Ready **	The connected ID sensor is OK (device ready) and the monitored CAN devices are accessible (display in the system status "Net monitoring OK"). **	X	X	---	X	X	X	---	---
Add.+1	D2	Good Read *	Status of the last read results *	X	X	X	X	X	X	X	X
Add.+1	D3	No Read *	Status of the last read results *	X	X	X	X	X	X	X	X
Add.+1	D4	External Output 1	Status of External Output 1 (via CMC600)	X	--- 1)	X	X	X	X	---	X
Add.+1	D5	External output 2	Status of External Output 2 (via CMC600)	X	--- 1)	X	X	X	X	---	X
Add.+1	D6	Result 1	Status of Result Output 1	X	X	X	X	X	X	---	X
Add.+1	D7	Result 2	Status of Result Output 2	X	X	X	X	X	X	---	X
Add.	D8	External Input 1	Status of External Input 1 (via CMC600)	X	---	X	X	X	X	---	X
Add.	D9	External input 2	Status of External Input 2 (via CMC600)	X	---	X	X	X	X	---	X
Add.	D10	Sensor 1	Status of Sensor Input 1	X	X	X	X	X	X	X	X
Add.	D11	Sensor 2	Status of Sensor Input 2	X	X	X	X	X	X	---	X
Add.	D12	-	Reserved								
Add.	D13	-	Reserved								
Add.	D14	-	Reserved								
Add.	D15	Toggle	Inverted after each read cycle.	X	X	X	X	X	X	X	X

Observe byte order: D0 stands for the least significant bit of Add.+1, X = supported, --- = not supported

* The bits are valid when the "Toggle" bit changes.

** Not supported by every ID sensor.

--- 1) Lector61x can not control the CMC600 due to missing serial Aux port. But this bit can be used as logical bit in Ctrl-Bits-In without CMC600. For Ctrl-Bits at CDF600 please refer to TI of CDF600.

CTRL-BITS-OUT @ PROFINET ON BOARD

HW - CONFIG / NECESSARY REQUIREMENT / CTRL-BITS-OUT / OUTPUTS AT PLC

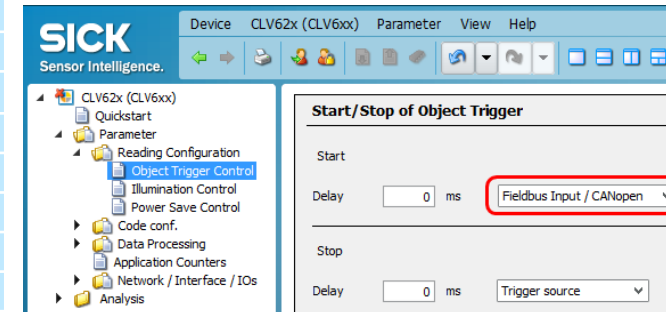
By using the Ctrl-Bits-Out, various functions can be activated in the ID sensor. For this to be allowed, the respective function must be configured in the ID sensor to “**Fieldbus Input**”. For logical assignment of the Ctrl-Bits-In see below table.
The available I/O hardware can vary depending on the type of ID sensor.

Address	Bit	Name	Meaning	CLV62x-65x	Lector61x	Lector620	Lector621-65x	Lector64x-65x 2port	RFH6xx	RFU61x	RFU62x-65x
Add.+1	D0	Trigger	Object trigger for the ID sensor	X	X	X	X	X	X	X	X
Add.+1	D1	Sensor Idle	Attached ID sensor is put into sleep mode **	X	---	---	---	---	---	---	---
Add.+1	D2	Teach In 1	The Teach In 1 teach-in process is triggered **	X	X	X	X	X	---	---	---
Add.+1	D3	Teach In 2	The Teach In 2 teach-in process is triggered **	X	X	X	X	X	---	---	---
Add.+1	D4	External Output 1	Controls external output 1 (via CMC)	X	---	X	X	X	X	---	X
Add.+1	D5	External output 2	Controls external output 2 (via CMC)	X	---	X	X	X	X	---	X
Add.+1	D6	Result 1	Controls Result output 1 **	---	X	X	X	X	---	---	X
Add.+1	D7	Result 2	Controls Result output 2 **	---	X	X	X	X	---	---	X
Add.	D8	-	Reserved								
Add.	D9	-	Reserved								
Add.	D10	-	Reserved								
Add.	D11	-	Reserved								
Add.	D12	Distance_Config_0	Controls bit 0, dynamic (focus) switchover **	X	---	---	---	---	---	---	---
Add.	D13	Distance_Config_1	Controls bit 1, dynamic (focus) switchover **	X	---	---	---	---	---	---	---
Add.	D14	Distance_Config_2	Controls bit 2, dynamic (focus) switchover **	X	---	---	---	---	---	---	---
Add.	D15	Distance_Config_3	Controls bit 3, dynamic (focus) switchover **	X	---	---	---	---	---	---	---

Example:

Set CLV62x-65x to read cycle start via “**Fieldbus Input**” and read cycle stop via “Read Cycle Source”.

→ The ID sensor can be triggered by the PLC using the bit **D0 “Trigger”**:



Observe byte order: D0 stands for the least significant bit of Add.+1, X = supported, --- = not supported

** Not supported by every ID sensor

For Ctrl-Bits at CDF600 please refer to TI of CDF600.

GSDML PARAMETERIZATION

HW-CONFIG / OPTIONALLY, DEPENDING ON TYPE

- Insert the required parameterization modules
- Always start with „Start Remote Config“, end with „End Remote Config“.

The screenshot shows the SIMATIC Manager HW Config interface. The main window displays the hardware configuration of a SICK CLV6xx sensor. The hardware rack configuration is shown on the left, with modules like CPU 315-2PN/DP, MPI/DP, PN-IO, and Port 1/2. The PROFIBUS and Ethernet networks are also shown. The 'Parameter' tab is selected, showing the '01_Start Remote Config' and '99_End Remote Config' modules. Red arrows point from these modules in the main window to their respective parameter configuration windows on the right.

Eigenschaften - 01_Start Remote Config >>>>>>>>> - (R-/55)

Parameter	Wert
General Parameter	
Load Defaults	Application Defaults
Codabar	No
Code 39	Yes
UPC EAN	No
2/5 Interleaved	Yes
Code 93	No
Code 128	Yes
Pharma	No
Start Object Trigger	Sensor 1
GS1 DataBar	No

Eigenschaften - 25_Code 128 Family - (R-/57)

Parameter	Wert
General Parameter	
Multiple Reads	3
Code 128 Active	Yes
EAN 128 Active	No
EAN 128 FC1 First Pos. 1	49
EAN 128 FC1 First Pos. 2	49
EAN 128 FC1 First Pos. 3	49
EAN 128 FC1 Within Code 1	0
EAN 128 FC1 Within Code 2	0
EAN 128 FC1 Within Code 3	0
Length Mode	Free
Length Interval Min	0
Length Interval Max	0
Length Fix 1	0
Length Fix 2	0
Length Fix 3	0
Length Fix 4	0
Length Fix 5	0

Eigenschaften - 99_End Remote Config >>>>>>>>> - (R-/59)

Parameter	Wert
General Parameter	
End of Remote Config	Save parameters permanently

Note: The GSDML configuration is always dominant and overwrites existing SOPAS ET parameters with each power cycling of the device or the IO Controller. The GSDML Configuration is not supported by all devices. E.G. the CLV690 do NOT support this function.

CHECK/SET PROFINET NAME VIA PLC TOOLS

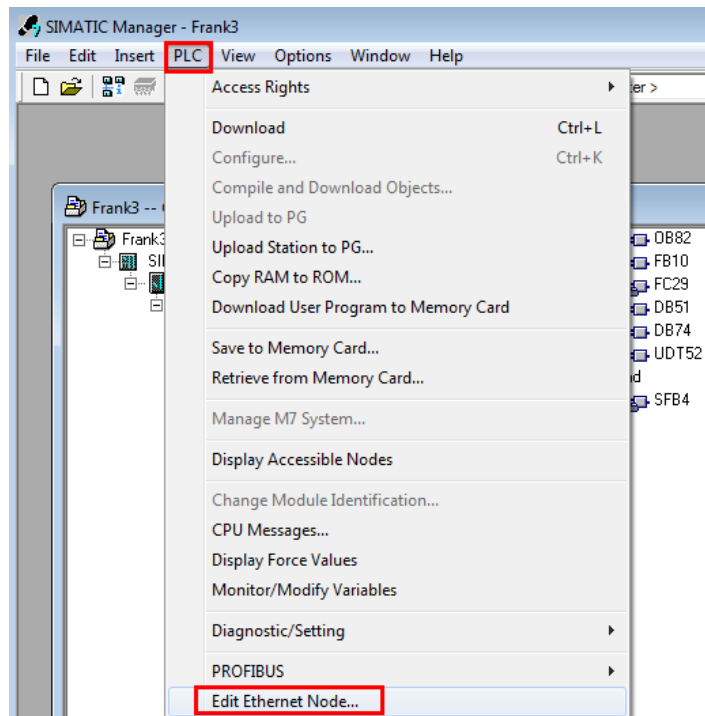
START FROM HW-CONFIG OR S7 MANAGER

Tools for checking and assigning PN name and IP address from the PLC side:
You can use this tool in the S7 programming system to search for a PROFINET device and to check or change the PROFINET device name and IP address.

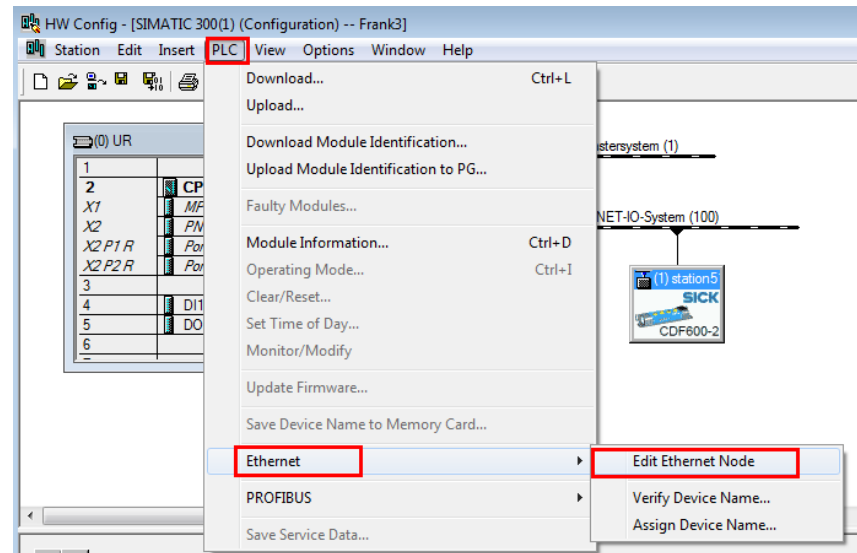
Starting from S7 manager: PLC > Edit Ethernet Nodes

Starting from S7 HW Config: PLC > Ethernet > Edit Ethernet Nodes

Start in S7 manager:



Or start in S7 hardware config:



Using “Browse” you can search for all PROFINET devices. At the selected device the PN-Name or the IP can be set. The communication is done via MAC address. So can be done also when PN name and IP is not correct or double defined.

Either way, the following window appears:

The 'Edit Ethernet Node' window is divided into several sections:

- Ethernet node:** Contains a 'MAC address' field and a 'Browse...' button. A label 'Nodes accessible online' is present.
- Set IP configuration:** Includes radio buttons for 'Use IP parameters' (selected) and 'Obtain IP address from a DHCP server'. Under 'Use IP parameters', there are fields for 'IP address' and 'Subnet mask', and a 'Gateway' section with a 'Do not use router' radio button (selected) and a 'Use router' option with an 'Address' field.
- Identified by:** Includes radio buttons for 'Client ID' (selected), 'MAC address', and 'Device name'. A 'Client ID' field is present.
- Assign device name:** Includes a 'Device name' field and an 'Assign Name' button.
- Reset to factory settings:** Includes a 'Reset' button.

At the bottom are 'Close' and 'Help' buttons.

You can scan the network using "Browse":

The 'Browse Network - 3 Nodes' window displays a table of discovered nodes. The table has columns for IP address, MAC address, Device type, Name, and Subnet mask. The third row is highlighted in blue.

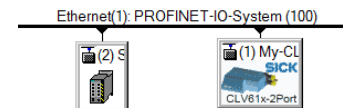
!	IP address	MAC address	Device type	Name	Subnet mask
	192.168.0.50	00-1B-1B-2D-7E-F7	S7-300	pn-io
	192.168.0.49	00-1B-1B-4D-77-17	SCALANCE X-200	scalance-x208
	192.168.0.51	00-06-77-06-0F-80	CDF600-220x	station51

You can search for a device (MAC address) and assign this a device name or an IP address.

VERIFY PROFINET NAME VIA PLC TOOLS

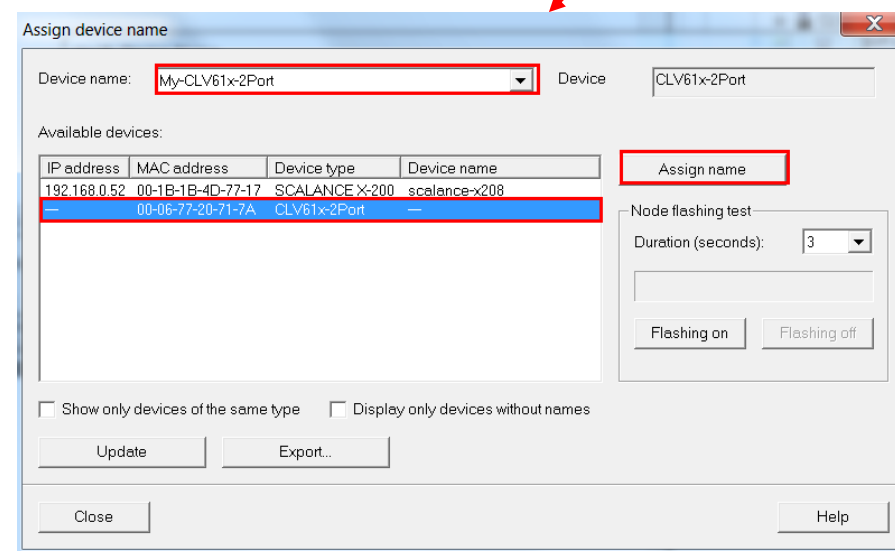
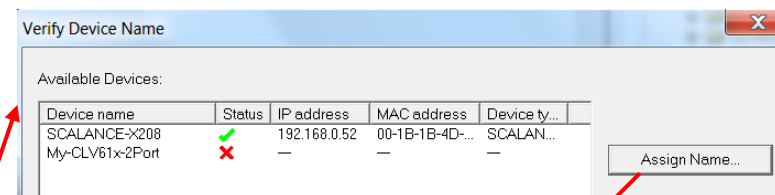
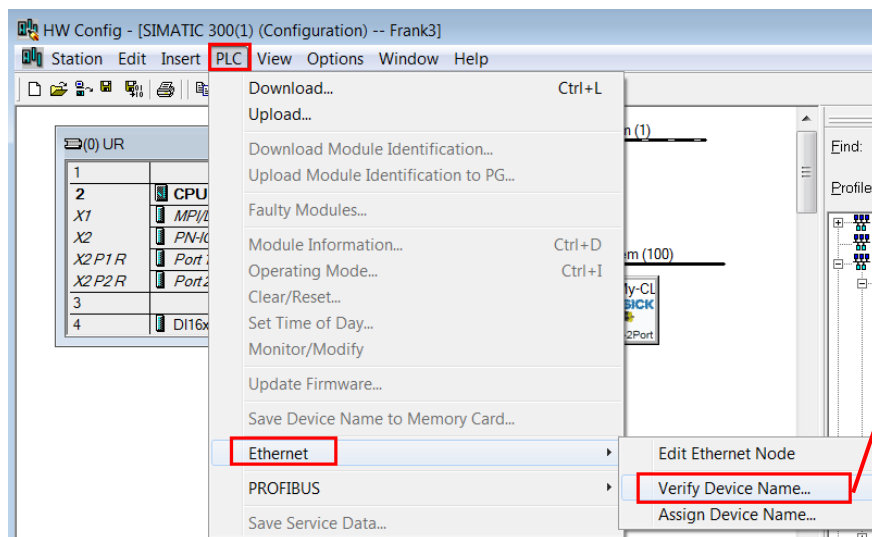
EASIEST WAY TO SET THE PN-NAME TO A DEVICE

Select at first the PN-Line, until it is **solid** (not dotted) displayed:



Then start the verify tool in the HW-Config with “PLC > Ethernet > Verify Device Name”.

Then the network is scanned if all defined names are present. If not, you can press the button “Assign Name”. Then above the missing name(s) shown and below the devices found in the network are shown. Then you can select the correct device and assign the PN-Name just with one click, without typing it manually again.



Note:

The green status ok means only that the **PN-Name is matching**.
You have to verify your own that the correct GSD file is used.

CONFIGURATION VIA SOPAS ET



03

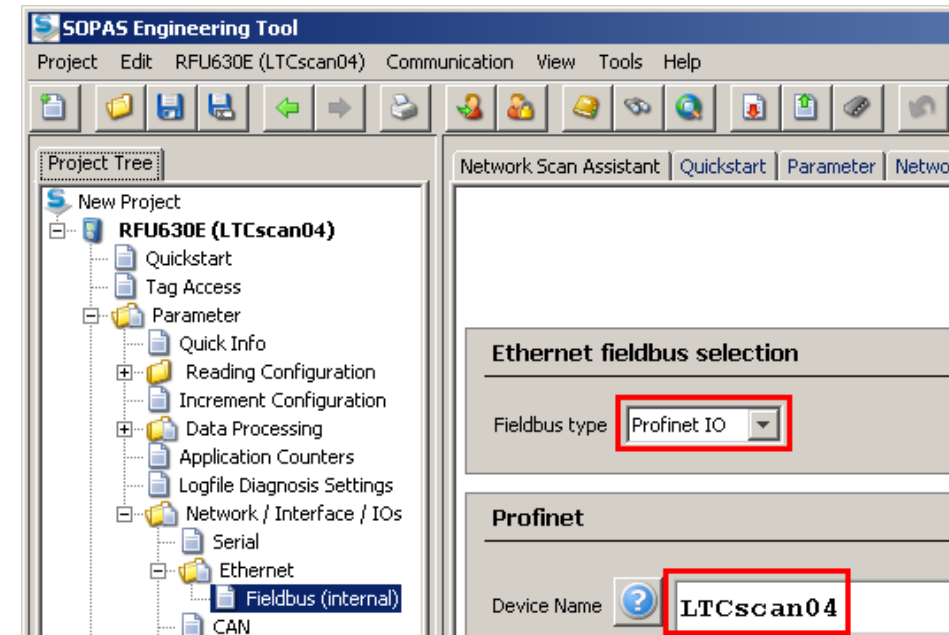
The devices can be fully configured via PLC.

But additional you can use SOPAS ET to configure or check the devices.

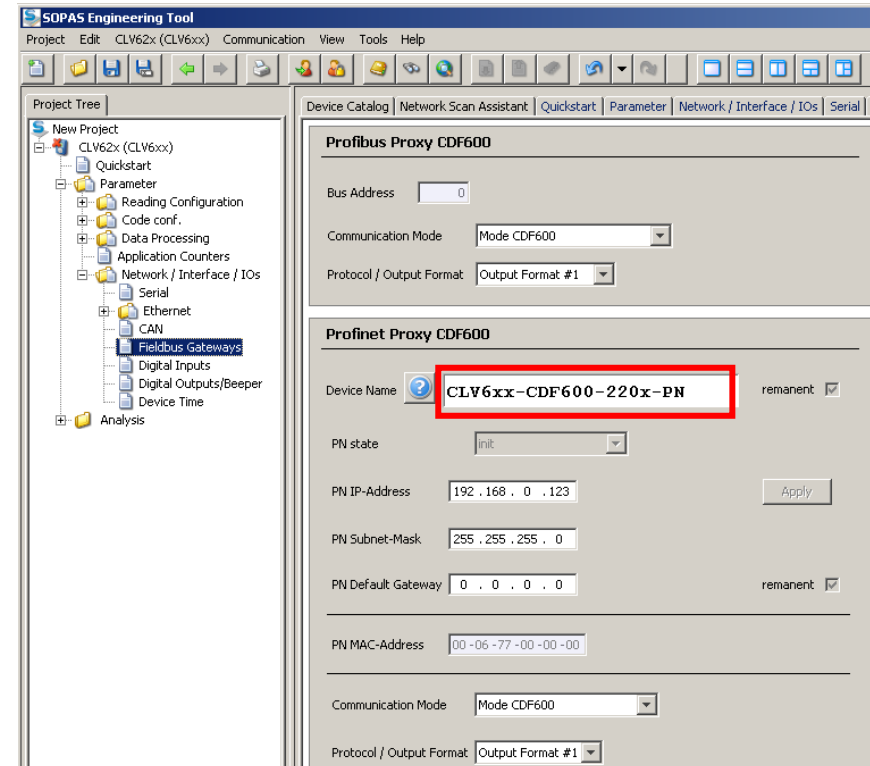
ACTIVATE PROFINET

SET DEVICE NAME – AT PROFINET-INTERN

- In SOPAS ET you have to activate PROFINET by selecting „Parameter / Network Interfaces IO / Ethernet / Fieldbus (internal)“ the fieldbus type to „PROFINET“
- Choose the **device name** to the same name that you have defined in the PLC
- If it is not done by the PLC you also can set the IP address of the device
- To activate the new parameterization save parameter permanent and power cycle the device



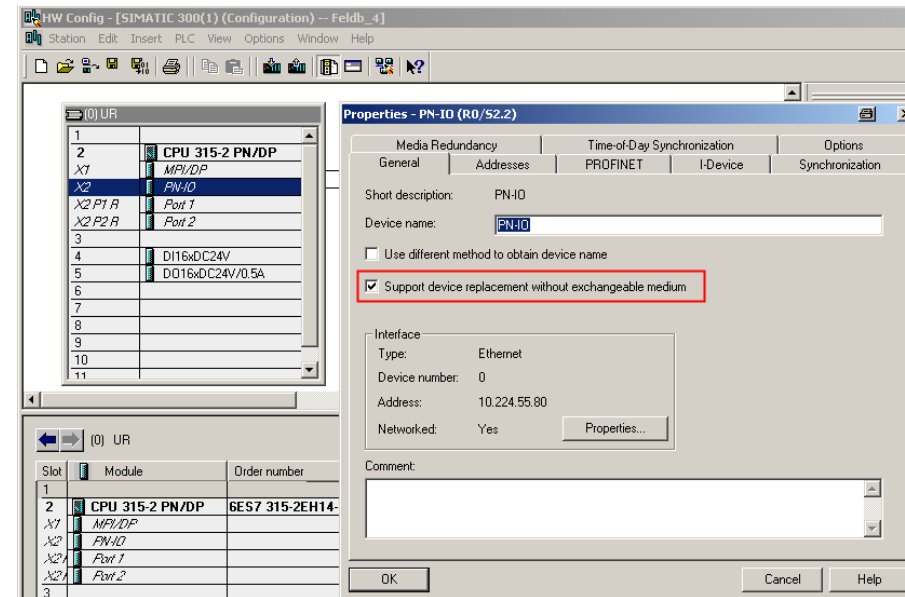
- When a CLV62x-65x (V5.25 or higher) is connected to CDF600-220x (Mode 0) it shows in SOPAS ET the page „PROFINET Proxy CDF600“ at „Parameter / Network Interfaces / IO / Fieldbus Gateways“.
- Choose the **device name** to the same name that you have defined in the PLC
- If it is not done by the PLC you also can set the IP address of the device
- To activate the new parameterization **save parameter permanent** and **power cycle** the device



- SICK 4Dpro single port devices are delivered with PROFINET not activated
- When you insert a new device with default parameterization in a PROFINET network, it recognizes that it is in a PROFINET environment installed, and activates PROFINET automatically
- It deletes the device name and activates PROFINET
- Then it can be found from HW-Config via „Edit Ethernet nodes“ and a name and/or IP can be manually set.

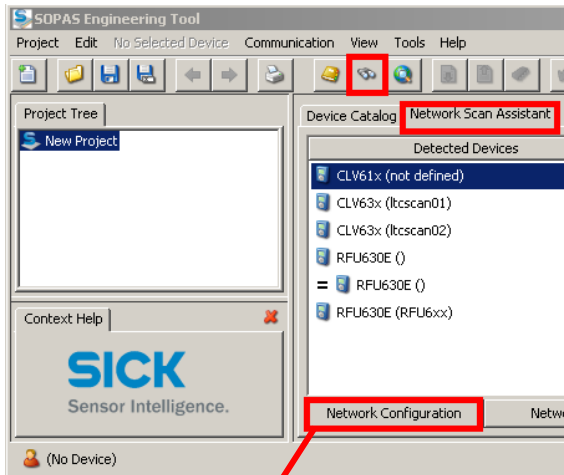
The screenshot displays two sections of the HW-Config software interface. The top section, titled "Ethernet fieldbus selection", contains a "Fieldbus type" dropdown menu currently set to "Profinet IO". The bottom section, titled "Profinet", features a "Device Name" label next to a red rectangular input field, with a blue question mark icon positioned to the left of the field.

- When the „Device replacement without exchangeable medium“ is completely configured in the PLC, and the used environment does support this, the device can get the PROFINET name automatically
- That means:
 - ▶ The feature must be activated in the PLC
 - ▶ The layout (topology) of the network must be trained or completely configured
 - ▶ The switches must support this (managed switches)
- When a 4Dpro device with default setting is inserted and the systems starts up, the device:
 - ▶ detects PROFINET
 - ▶ deletes the device name and activates PROFINET
 - ▶ gets its name from the PLC/PN-Controller
 - ▶ gets the IP
 - ▶ gets the sensor parameter via GSD-Parameterization (when available)
- ➔ So the new device is completely integrated in the PN network

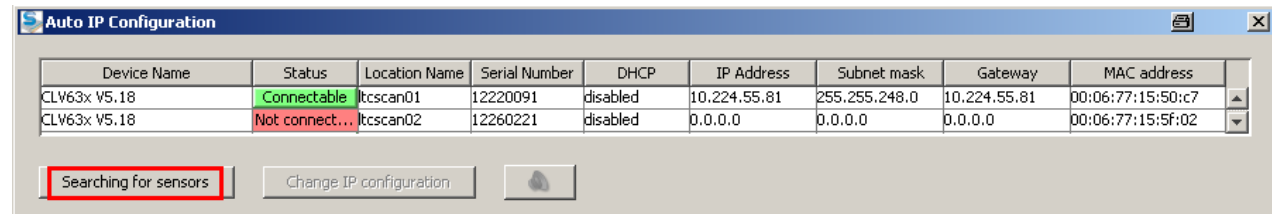
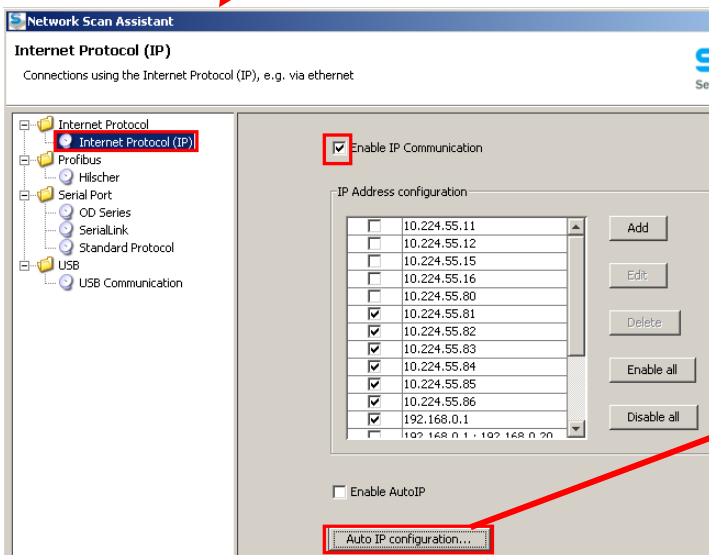


SEARCHING FOR A DEVICE

USING AUTO IP-SCAN OF SOPAS ET V2.X.X

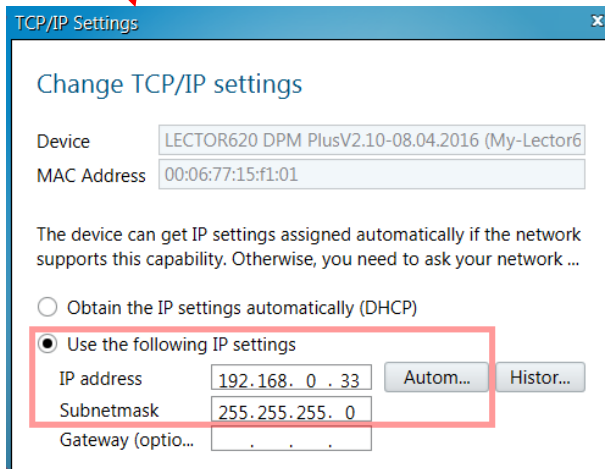
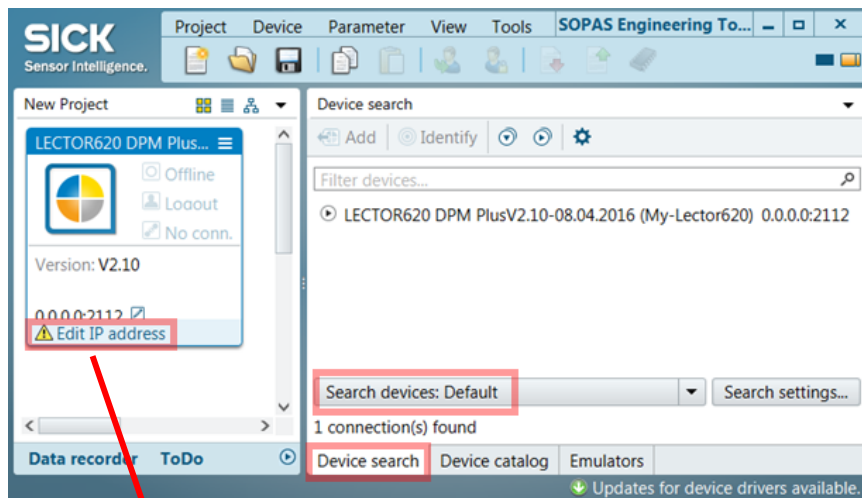


- With the device connecting at PROFINET (Ethernet) you can scan it from SOPAS ET. The PC must have a direct connection to the Ethernet network.
- Choose „View / Network Scan Assistant“ and press „Network configuration“.
- Enable „IP Communication“ press the button „Auto IP-Configuration“.
- With „Searching for sensors“ start the Auto IP Scan.
- Auto IP scan is MAC addressed based and will find any **SICK Ethernet device** independent if PROFINET is enabled or not, and also if the IP address is wrong or 0.



- By double clicking the device you can set the IP address new.
- With a valid IP address you can connect and parametrize a device with SOPAS ET

SEARCHING FOR A DEVICE USING SOPAS ET V3.3.2 OR HIGHER



- With the device connecting at PROFINET (Ethernet) you can scan it from SOPAS ET. The PC must have a direct connection to the Ethernet network.
- Then search for the devices via Ethernet, e.g. use “Default” search. Then all selected **Sick Ethernet devices** can be found with a valid IP-Address.
- The devices may be then seen also with IP 0.0.0.0 (because IP was set temporary) and can be not direct connected. But when you click on “Edit IP address” you can change the IP address.
- With a valid IP address you can connect and parametrize the device with SOPAS ET.

Note:

The IP is normally set by the PLC (IO-Controller). So if possible connect the device to the PLC with a valid name to get an IP address and use this. But if needed, you may set also via Sopas. When next time the PLC is connected the IP maybe overwritten again.

DATA CHANNEL



04

Byte Layout CLV6xx HS-Mode:

Address	Inputs (data CLV → PLC)		Output (data PLC → CLV)
1	Binary Inputs		Binary Outputs
2	ReceiveCount (counter)	→	ReceiveCountBack (counter)
3	TransmitCountBack (counter)	←	TransmitCount (counter)
4	ReceiveLength Lowbyte		TransmitLength Lowbyte
5	ReceiveLength Highbyte		TransmitLength Highbyte
6	ReceiveData, Byte 1		TransmitData, Byte 1
7	ReceiveData, Byte 2		TransmitData, Byte 2
...
n	ReceiveData, Byte n - 5		TransmitData, Byte n - 5

- There are data modules (choose only one per direction) available with data lengths between 8 .. 128 bytes. In handshake mode, the first 5 bytes are used for handing over the data while the rest contains the data itself. Therefore, one module with the length of 32 bytes can contain 27 bytes of user data.
- Receive data:
 - ▶ CLV6xx puts the data into the field „ReceiveData“ and writes the length into the field „ReceiveLength“. Additionally, the value „ReceiveCount“ is being incremented to show that new data is available.
- Byte handshake:
 - ▶ The PLC must answer to show that the data has been correctly received. This is done by copying the value „Receive Count“ to the output module into the value „ReceiveCountBack“ within 10 seconds. You see, the second input byte must be copied to the second output byte. (§ bytelandshake §).
 - ▶ CLV6xx recognizes that both values are identical it can send the next data.
 - ▶ The value „ReceiveCount“ can be 1...255. The zero is skipped in the standard operation.
 - ▶ If CLV6xx shows the value 0, this is an error indication. In this case, it must be started with 0 and the PLC must answer with 0.
 - ▶ The PLC normally should copy the value „ReceiveCount“ to the value „ReceiveCountBack“ without any condition.

DATA TRANSMISSION – EXAMPLE 1

DATA RECEIVE CLV6XX

- Input: 16 bytes, Output: 16 bytes, data telegram up to 11 bytes → Single telegram
- Remarks:
 - ▶ Time 1: The first data “CLV6xx-Data” (11 bytes) has been received and transmitted to the PLC with ReceiveCount = 1.
 - ▶ Time 2: The PLC recognizes and acknowledges this by copying ReceiveCount to ReceiveCountBack. Now, the CLV6xx can deliver the data.
 - ▶ Time 3: The second data “123456789” (9 bytes) has been received by CLV6xx and transmitted to the PLC with ReceiveCount = 2.
 - ▶ Time 4: The recognizes this and acknowledges this by copying ReceiveCount to ReceiveCountBack.
- The data length that can be transported depends on the input module. Length of input module minus 5 → max. data length

Time:				1	2	3	4
// Input Byte - 16 Byte (Result Data received from CDM / Sensor)							
EB 50	"Stat-In-Bits"	BIN	2#0000_0100	2#0000_0100	2#0000_0100	2#0000_0000	2#0000_0000
EB 51	"Rec-Cnt"	DEC	1	1	2	2	2
EB 52	"Tr-Cnt-back"	DEC	0	0	0	0	0
EB 53	"Rec-Len-Low"	DEC	11	11	9	9	9
EB 54	"Rec-Len-High"	DEC	0	0	0	0	0
EB 55	"Rec-Data-1"	CHARACTER	'C'	'C'	'1'	'1'	'1'
EB 56	"Rec-Data-2"	CHARACTER	'L'	'L'	'2'	'2'	'2'
EB 57	"Rec-Data-3"	CHARACTER	'V'	'V'	'3'	'3'	'3'
EB 58	"Rec-Data-4"	CHARACTER	'6'	'6'	'4'	'4'	'4'
EB 59	"Rec-Data-5"	CHARACTER	'x'	'x'	'5'	'5'	'5'
EB 60	"Rec-Data-6"	CHARACTER	'x'	'x'	'6'	'6'	'6'
EB 61	"Rec-Data-7"	CHARACTER	'L'	'L'	'7'	'7'	'7'
EB 62	"Rec-Data-8"	CHARACTER	'D'	'D'	'8'	'8'	'8'
EB 63	"Rec-Data-9"	CHARACTER	'a'	'a'	'9'	'9'	'9'
EB 64	"Rec-Data-10"	CHARACTER	't'	't'	B#16#00	B#16#00	B#16#00
EB 65	"Rec-Data-11"	CHARACTER	'a'	'a'	B#16#00	B#16#00	B#16#00
// Output bytes - 16 Byte (e.g. Commands send to CDM / Sensor)					Sensor)	Sensor)	Sensor)
AB 50	"Stat-Out-Bits"	BIN	2#0000_0000	2#0000_0000	2#0000_0000	2#0000_0000	2#0000_0000
AB 51	"Rec-Cnt-Back"	DEC	0	1	1	2	2
AB 52	"Tr-Cnt"	DEC	0	0	0	0	0
AB 53	"Tr-Len-Low"	DEC	0	0	0	0	0
AB 54	"Tr-Len-High"	DEC	0	0	0	0	0
AB 55	"Tr-Data-1"	CHARACTER	B#16#00	B#16#00	B#16#00	B#16#00	B#16#00
AB 56	"Tr-Data-2"	CHARACTER	B#16#00	B#16#00	B#16#00	B#16#00	B#16#00
AB 57	"Tr-Data-3"	CHARACTER	B#16#00	B#16#00	B#16#00	B#16#00	B#16#00
AB 58	"Tr-Data-4"	CHARACTER	B#16#00	B#16#00	B#16#00	B#16#00	B#16#00
AB 59	"Tr-Data-5"	CHARACTER	B#16#00	B#16#00	B#16#00	B#16#00	B#16#00
AB 60	"Tr-Data-6"	CHARACTER	B#16#00	B#16#00	B#16#00	B#16#00	B#16#00
AB 61	"Tr-Data-7"	CHARACTER	B#16#00	B#16#00	B#16#00	B#16#00	B#16#00
AB 62	"Tr-Data-8"	CHARACTER	B#16#00	B#16#00	B#16#00	B#16#00	B#16#00
AB 63	"Tr-Data-9"	CHARACTER	B#16#00	B#16#00	B#16#00	B#16#00	B#16#00
AB 64	"Tr-Data-10"	CHARACTER	B#16#00	B#16#00	B#16#00	B#16#00	B#16#00
AB 65	"Tr-Data-11"	CHARACTER	B#16#00	B#16#00	B#16#00	B#16#00	B#16#00

DATA TRANSMISSION– EXAMPLE 2

DATA RECEIVE CLV6XX: LONG TELEGRAMS WITH BLOCKING

- Input: 16 byte, Output: 16 bytes, Data “CLV6xx-12345” (12 bytes) → separation into two blocks
- Remarks:
 - ▶ Time 1: First data “CLV6xx-1234” (first 11 bytes) has been received and transmitted to the PLC with ReceiveCount = 1. ReceiveLength is 12 byte.
 - ▶ Time 2: The PLC recognizes this and detects that the length (12 bytes) is longer then the input module length. The PLC interprets that there must be more blocks. Again the PLC acknowledges by copying ReceiveCount to ReceiveCountBack.
 - ▶ Time 3: CLV6xx directly sends the next block with the rest of data. Here, 1 byte with content „5“ with ReceiveCount = 2 und ReceiveLength = 1.
 - ▶ Time 4: The PLC has recognized the following block and acknowledges this by copying ReceiveCount to ReceiveCountBack. Now, the transmission of the complete telegram is finished.
- The data length that can be transported in **one** block depends on the input module
→ max. data length = length of input module minus 5
Max. telegram length is 4000 bytes (separated into blocks)

Time:				1	2/3	4
// Input Byte - 16 Byte (Result Data received from CDM / Sensor)						
EB 50	"Stat-In-Bits"	BIN	2#0000_0000	2#0000_0000	2#0000_0100	
EB 51	"Rec-Cnt"	DEC	1	2	2	
EB 52	"Tr-Cnt-back"	DEC	0	0	0	
EB 53	"Rec-Len-Low"	DEC	12	1	1	
EB 54	"Rec-Len-High"	DEC	0	0	0	
EB 55	"Rec-Data-1"	CHARACTER	'C'	'5'	'5'	
EB 56	"Rec-Data-2"	CHARACTER	'L'	B#16#00	B#16#00	
EB 57	"Rec-Data-3"	CHARACTER	'V'	B#16#00	B#16#00	
EB 58	"Rec-Data-4"	CHARACTER	'6'	B#16#00	B#16#00	
EB 59	"Rec-Data-5"	CHARACTER	'x'	B#16#00	B#16#00	
EB 60	"Rec-Data-6"	CHARACTER	'x'	B#16#00	B#16#00	
EB 61	"Rec-Data-7"	CHARACTER	'I'	B#16#00	B#16#00	
EB 62	"Rec-Data-8"	CHARACTER	'1'	B#16#00	B#16#00	
EB 63	"Rec-Data-9"	CHARACTER	'2'	B#16#00	B#16#00	
EB 64	"Rec-Data-10"	CHARACTER	'3'	B#16#00	B#16#00	
EB 65	"Rec-Data-11"	CHARACTER	'4'	B#16#00	B#16#00	
// Output bytes - 16 Byte (e.g. Commands send to CDM / Sensor)					Sensor)	Sensor)
AB 50	"Stat-Out-Bits"	BIN	2#0000_0000	2#0000_0000	2#0000_0000	
AB 51	"Rec-Cnt-Back"	DEC	0	1	2	
AB 52	"Tr-Cnt"	DEC	0	0	0	
AB 53	"Tr-Len-Low"	DEC	0	0	0	
AB 54	"Tr-Len-High"	DEC	0	0	0	
AB 55	"Tr-Data-1"	CHARACTER	B#16#00	B#16#00	B#16#00	
AB 56	"Tr-Data-2"	CHARACTER	B#16#00	B#16#00	B#16#00	
AB 57	"Tr-Data-3"	CHARACTER	B#16#00	B#16#00	B#16#00	
AB 58	"Tr-Data-4"	CHARACTER	B#16#00	B#16#00	B#16#00	
AB 59	"Tr-Data-5"	CHARACTER	B#16#00	B#16#00	B#16#00	
AB 60	"Tr-Data-6"	CHARACTER	B#16#00	B#16#00	B#16#00	
AB 61	"Tr-Data-7"	CHARACTER	B#16#00	B#16#00	B#16#00	
AB 62	"Tr-Data-8"	CHARACTER	B#16#00	B#16#00	B#16#00	
AB 63	"Tr-Data-9"	CHARACTER	B#16#00	B#16#00	B#16#00	
AB 64	"Tr-Data-10"	CHARACTER	B#16#00	B#16#00	B#16#00	
AB 65	"Tr-Data-11"	CHARACTER	B#16#00	B#16#00	B#16#00	

DATA TRANSMISSION– EXAMPLE 3

SEND COMMANDS CLV6XX: LONG TELEGRAMS WITH BLOCKING

- Input: 16 bytes, output: 16 bytes, Data telegram
“2MC10j1234567890” with 16 bytes → separated into 2 blocks
- This command sets the condition „Match 1“ to code 128 with content „1234567890“ for CLV6xx. The command does not create an answer.
- Remarks:
 - ▶ Time 1: First data is sent by setting TransmitCount = 1 and TransmitLength = 16.
 - ▶ Time 2: CLV6xx recognizes first data and acknowledges by copying TransmitCount to TransmitCountBack.
 - ▶ Time 3: Next data is sent by setting TransmitCount = 2 and TransmitLength = 5.
 - ▶ Time 4: CLV6xx recognizes the next data and acknowledges by copying TransmitCount to TransmitCountBack.
- TransmitCount must start with 1 when 255 has been reached. 0 is skipped.
Max. data length is 4000 bytes (separated into several blocks).

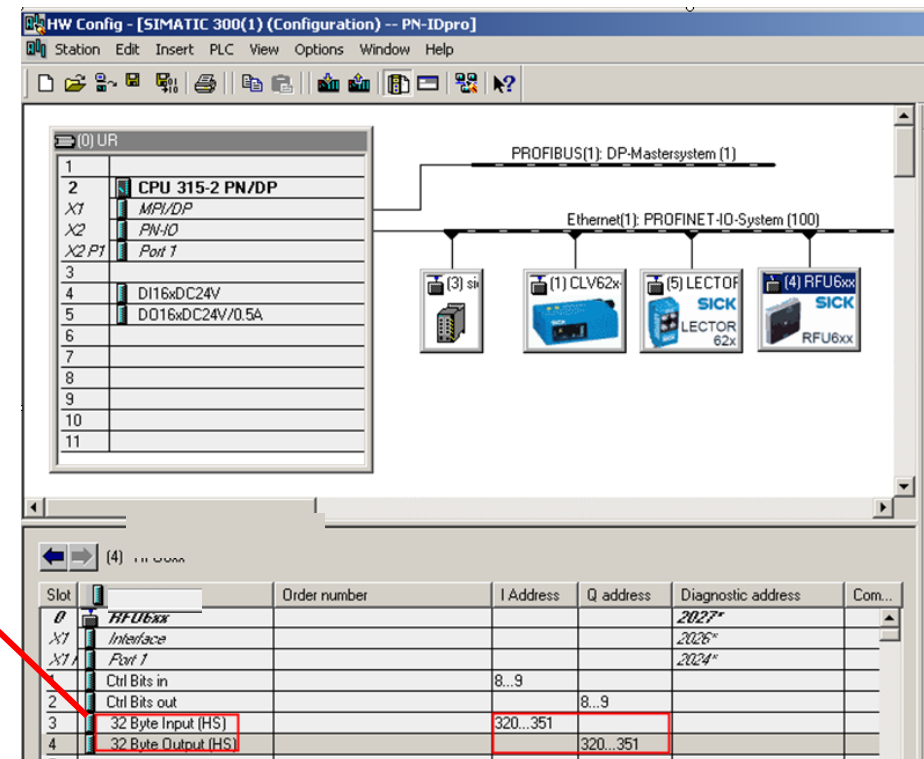
			Time:	1/2	3/4
// Input Byte - 16 Byte (Result Data received from CDM / Sensor)					
EB 50	"Stat-In-Bits"	BIN	2#0000_0100	2#0000_0000	
EB 51	"Rec-Cnt"	DEC	0	0	
EB 52	"Tr-Cnt-back"	DEC	1	2	
EB 53	"Rec-Len-Low"	DEC	1	1	
EB 54	"Rec-Len-High"	DEC	0	0	
EB 55	"Rec-Data-1"	CHARACTER	'1'	'1'	
EB 56	"Rec-Data-2"	CHARACTER	B#16#00	B#16#00	
EB 57	"Rec-Data-3"	CHARACTER	B#16#00	B#16#00	
EB 58	"Rec-Data-4"	CHARACTER	B#16#00	B#16#00	
EB 59	"Rec-Data-5"	CHARACTER	B#16#00	B#16#00	
EB 60	"Rec-Data-6"	CHARACTER	B#16#00	B#16#00	
EB 61	"Rec-Data-7"	CHARACTER	B#16#00	B#16#00	
EB 62	"Rec-Data-8"	CHARACTER	B#16#00	B#16#00	
EB 63	"Rec-Data-9"	CHARACTER	B#16#00	B#16#00	
EB 64	"Rec-Data-10"	CHARACTER	B#16#00	B#16#00	
EB 65	"Rec-Data-11"	CHARACTER	B#16#00	B#16#00	
// Output bytes - 16 Byte (e.g. Commands send to CDM / Sensor)					
AB 50	"Stat-Out-Bits"	BIN	2#0000_0000	2#0000_0000	
AB 51	"Rec-Cnt-Back"	DEC	0	0	
AB 52	"Tr-Cnt"	DEC	1	2	
AB 53	"Tr-Len-Low"	DEC	16	5	
AB 54	"Tr-Len-High"	DEC	0	0	
AB 55	"Tr-Data-1"	CHARACTER	'2'	'6'	
AB 56	"Tr-Data-2"	CHARACTER	'M'	'7'	
AB 57	"Tr-Data-3"	CHARACTER	'C'	'8'	
AB 58	"Tr-Data-4"	CHARACTER	'1'	'9'	
AB 59	"Tr-Data-5"	CHARACTER	'0'	'0'	
AB 60	"Tr-Data-6"	CHARACTER	'J'	B#16#00	
AB 61	"Tr-Data-7"	CHARACTER	'1'	B#16#00	
AB 62	"Tr-Data-8"	CHARACTER	'2'	B#16#00	
AB 63	"Tr-Data-9"	CHARACTER	'3'	B#16#00	
AB 64	"Tr-Data-10"	CHARACTER	'4'	B#16#00	
AB 65	"Tr-Data-11"	CHARACTER	'5'	B#16#00	

USING FUNCTION BLOCK PROFIBUS/PROFINET FOR CLV6XX/LECTOR, RFU AND RFH

- When a S7 Function block PROFIBUS / PROFIBUS is used the I/O address of the input and output bytes must be inserted. It handles the handshake mode automatically.
- There exists one for CLV6xx/Lector, RFH and RFU.
Can be download at : <http://www.sick.com/software> and select Software category „Function block“ (for further details see attached documentation)
- Available function blocks for: (further versions on request)
S7 300/400 V5.5 Profinet / Profibus
S7 300/400 TIA Profinet / Profibus
S7 1200/1500 TIA Profinet / Profibus

```
CALL "LECTOR_CLV6XX", "INSTANCE_FB52"
IN_ADDR :=W#16#140 // input address: 140 hex = 320 dec
IN_LEN :=32 // length = 32 byte
OUT_ADDR :=W#16#140 // output address: 140 hex = 320 dec
OUT_LEN :=32 // length = 32 byte
CAN_ID :=B#16#0
RESET :="Reset"
START_REQ :="Start_Req"
TRIG_ON :="Trigon"
TRIG_OFF :="Trigoff"
MATCH_CODE :="Machcode"
COM_TEST :="Com_Test"
SAVE_PERM :="Save_Perm"
FREE_COMMAND:= "Free_command"
DATA := "LECTOR_CLV_DATA".Data
RD_RESULT :=P#DB51.DBX0.0 BYTE 100
RD_DONE := "RD_Done"
RD_COUNT := "RD_Count"
RD_LEN := "RD_Len"
REQ_DONE := "REQ_Done"
REQ_BUSY := "REQ_Busy"
ERROR := "Error"
ERRORCODE := "Errorcode"
```

CLV6xx/Lector for S7
300/400 V5.5



FURTHER INFORMATION



05

PROFINET NOT RUNNING – TROUBLE SHOOTING

POINTS TO BE CLARIFIED

- Is the SW-Version of Sensor ok? In general latest update is recommended

CLV61x Dual Port :	V1.34 or higher
CLV62x-65x PN intern:	V5.21 or higher
RFU6xx PN intern:	V1.40 or higher
RFH6xx PN intern::	V3.00 or higher
Lector620 PN intern:	V2.10 or higher
Lector61x PN 1port:	V4.0.0 or higher
Lector621, 63x, 64x, 65x PN 1port:	V3.1.1 or higher
CDF600-220x (Gateway):	V1.20 or higher
CLV61x-Fxxx PN via CDF-220x:	V1.21 or higher
CLV62x-65x PN via CDF-220x:	V5.26 or higher
RFU6xx PN via CDF-220x:	V1.62 or higher
- Is PROFINET in the device activated?
- Is the PROFINET name ok?
- Is the Device Ready LED blinking red /green when the device is power up? It indicates that PROFINET is active, but not online. E.g. it must start blinking when no Ethernet cable is plugged in.
- Is the name and IP correct shown when it is scanned via Sopas Auto IP scan?
- Is the name and IP correct shown when it is scanned via PLC HW-Config Ethernet Nodes?
- Is the PLC hardware config correct with at least four modules? Ctrl-Bits In/Out and Data Bytes In/Out. Do not delete Data Bytes Out !
- Is the correct GSDML-File used ? Is PN intern used, or PN via CDF ?
- Is the handshake mode correct selected ?
- When a function block is used: Are the In and Out addresses and lengths are ok ?

- Operating instruction
 - Detailed information about the device in general
 - > see downloads www.sick.com, select the product family
- Technical information
 - Detailed introduction how to integrate the device into PROFINET
 - > see GSDML package
- GSDML documentation (device depending)
 - Detailed introduction about parameterization via GSDML modules
 - > see GSDML package
- S7 Function block (available for S7 300/400 and S7 1200/1500 TIA)
 - ➔ optional to handle handshake mode automatically
 - > can be downloaded at <http://www.sick.com/software>, select software category „Function block“

MANY THANKS FOR YOUR ATTENTION.

