



OPERATION MANUAL

Fieldbus-Controller Model 9251 PROFINET Integration into TIA Portal

© 2025 burster
praezisionsmesstechnik gmbh & co kg
All rights reserved

Manufacturer:

burster
präzisionsmesstechnik gmbh & co kg
Talstr. 1 - 5 P.O. Box 1432
76593 Gernsbach 76587 Gernsbach
Germany Germany

Valid from: **22.04.2025**
Applies to: **9251-VXX03**

Tel.: +49-7224-645-0
Fax.: +49-7224-645-88
Email: info@burster.com
www.burster.com

4675-BA9251PROFINETTIAEN-5799-041529

Table of Contents

Introduction	3
1. Creating new project	4
2. Installation of GSDML files	6
3. Device replacement without exchangeable medium.....	7
4. Creation of network connections	8
5. Create a sample program	12
6. Further Examples	18
5.1 Reading of 'string' data types	19
5.2 Read and Write of 'real' data types	22

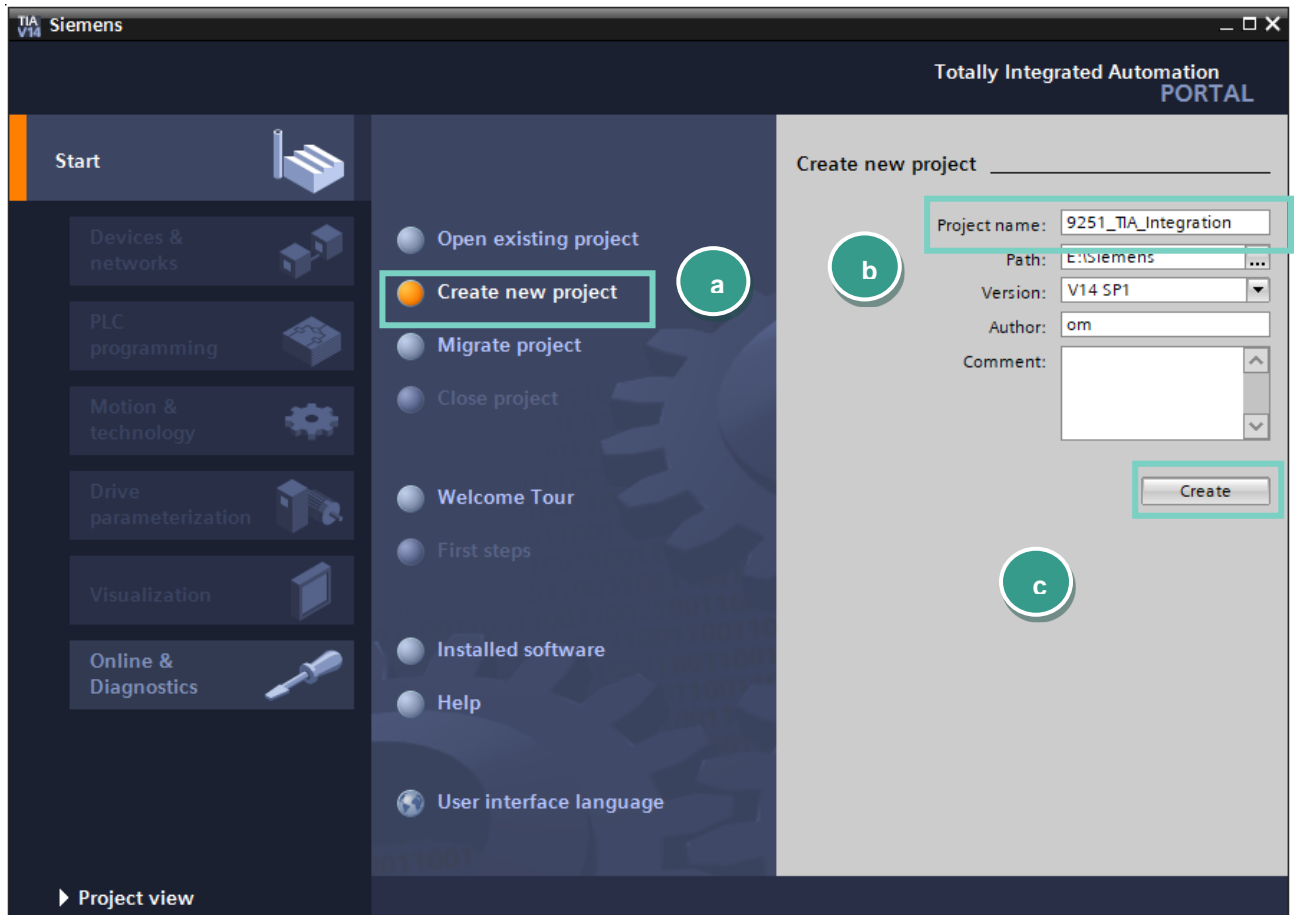
Introduction

This quick start guide describes an approach how you can configure the 9251 via TIA Portal using the example of S7-1511 CPU. Please note that the samples here cannot be directly used in your production line because they have been extremely simplified to reach a better understanding. Therefore, you may have to complete them by checking of status, error, length values etc.

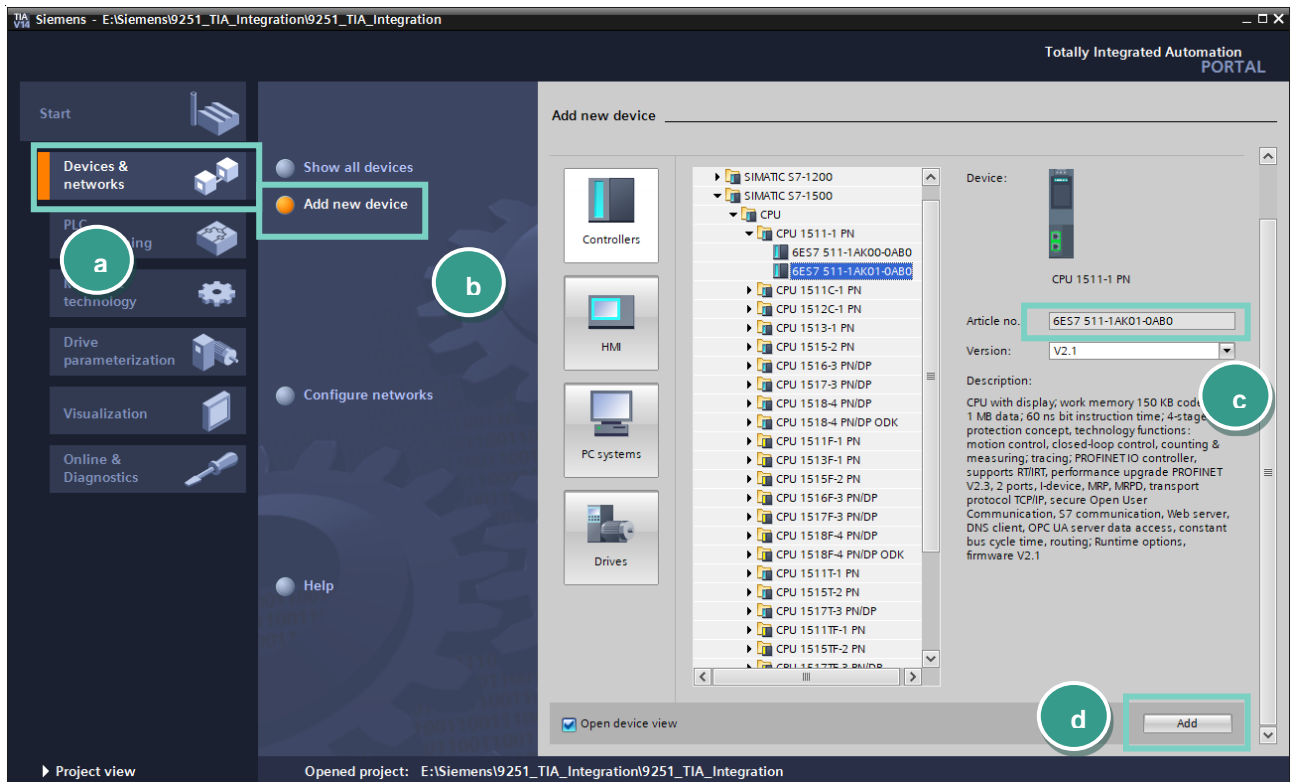
Please also note that you will have to use the 9251 PROFINET manual to get further information about input and output parameters (cyclic as well acyclic data transfer)

1. Creating new project

- Start the **Totally Integrated Automation Portal**, select **Create New Project** (a), assign the project a name (b) and click **Create** (c):



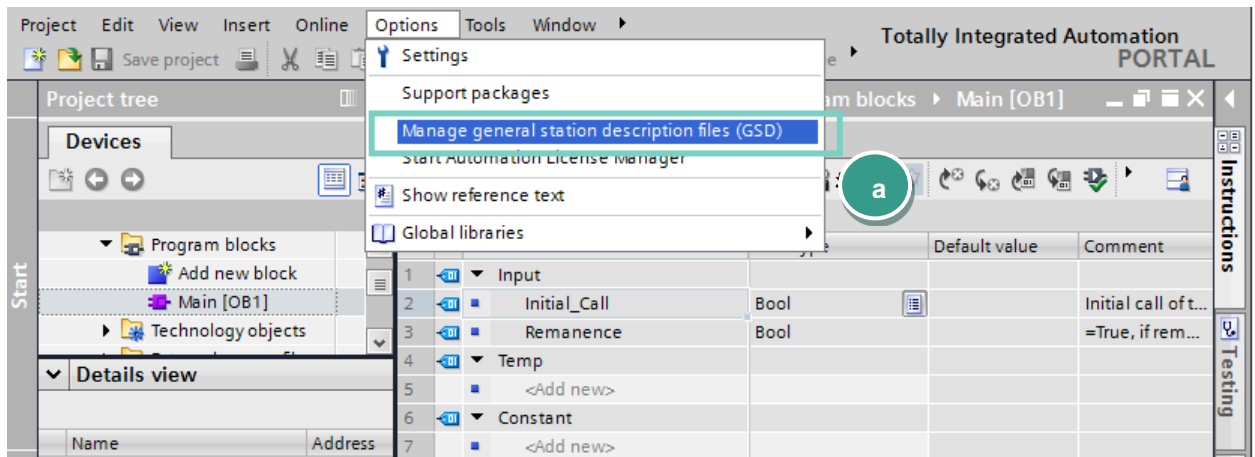
- Go to **Devices & networks** (a) on the left side select **Add new device** (b) and look for your CPU (c). Afterwards click the **Add** button (d).



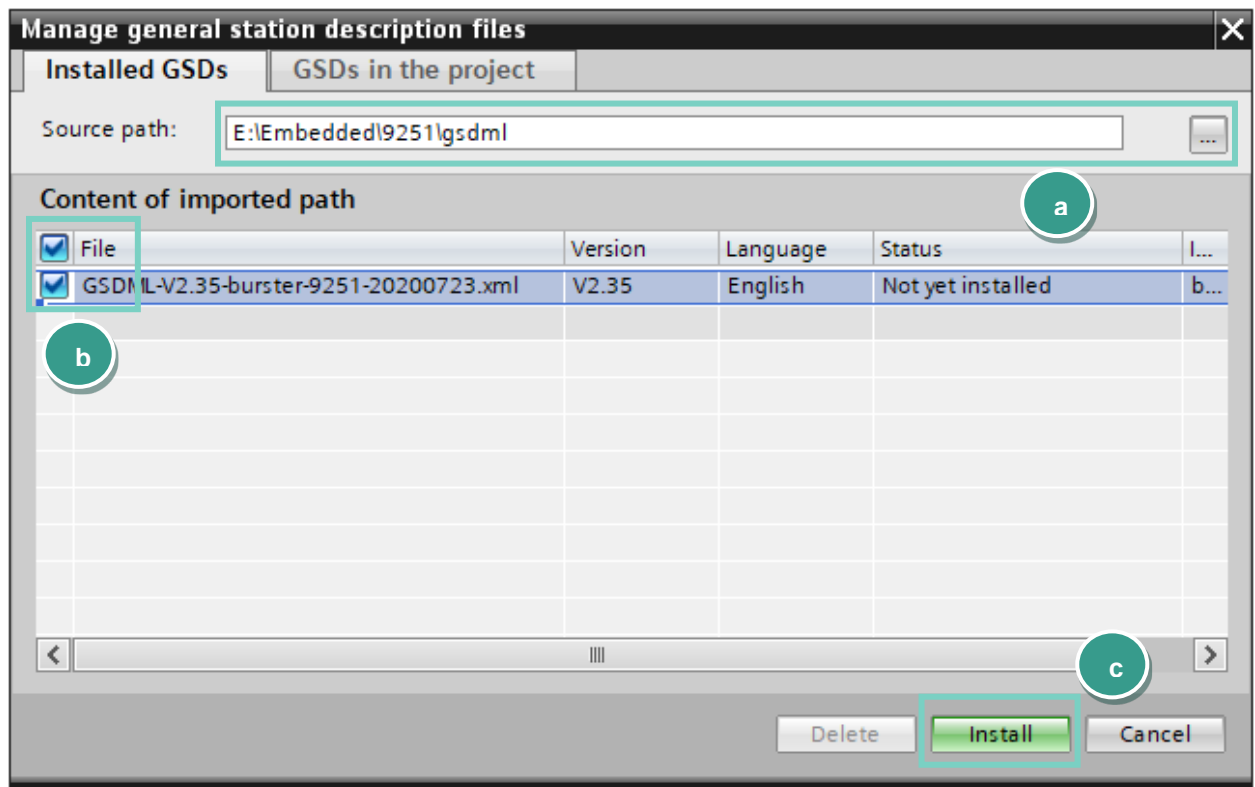
2. Installation of GSDML files

Note: Please make sure that your GSDML file is compatible to the field bus firmware in the 9251. The latest GSDML file is available for download on www.burster.com. Also for compatibility reasons, uninstall all previous GSDML files of particular device if you have any!

- Go to **Options** → **Manage general station description files (GSD)**



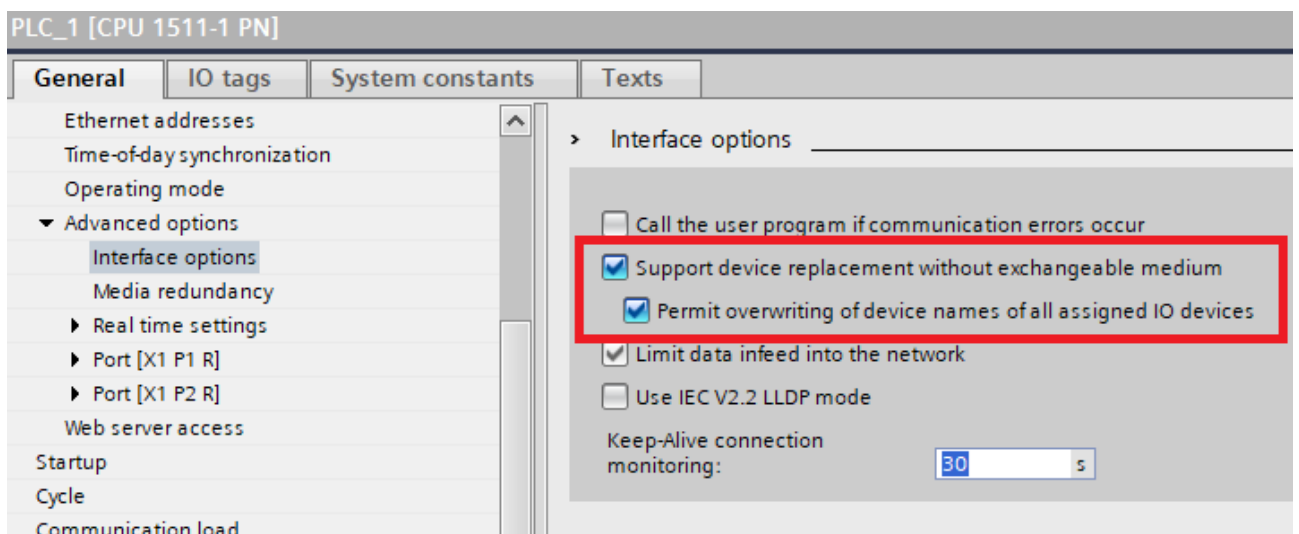
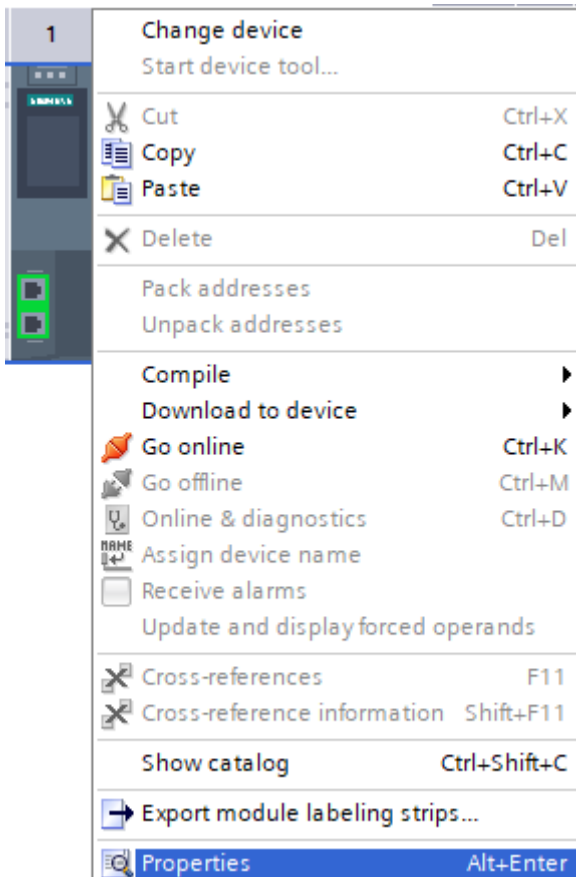
- Navigate to your 9251 GSDML directory (a)(you will find the GSD files for download on www.burster.com Download Area/Software), select the GSD file (b) and click **Install** (c)



3. Device replacement without exchangeable medium

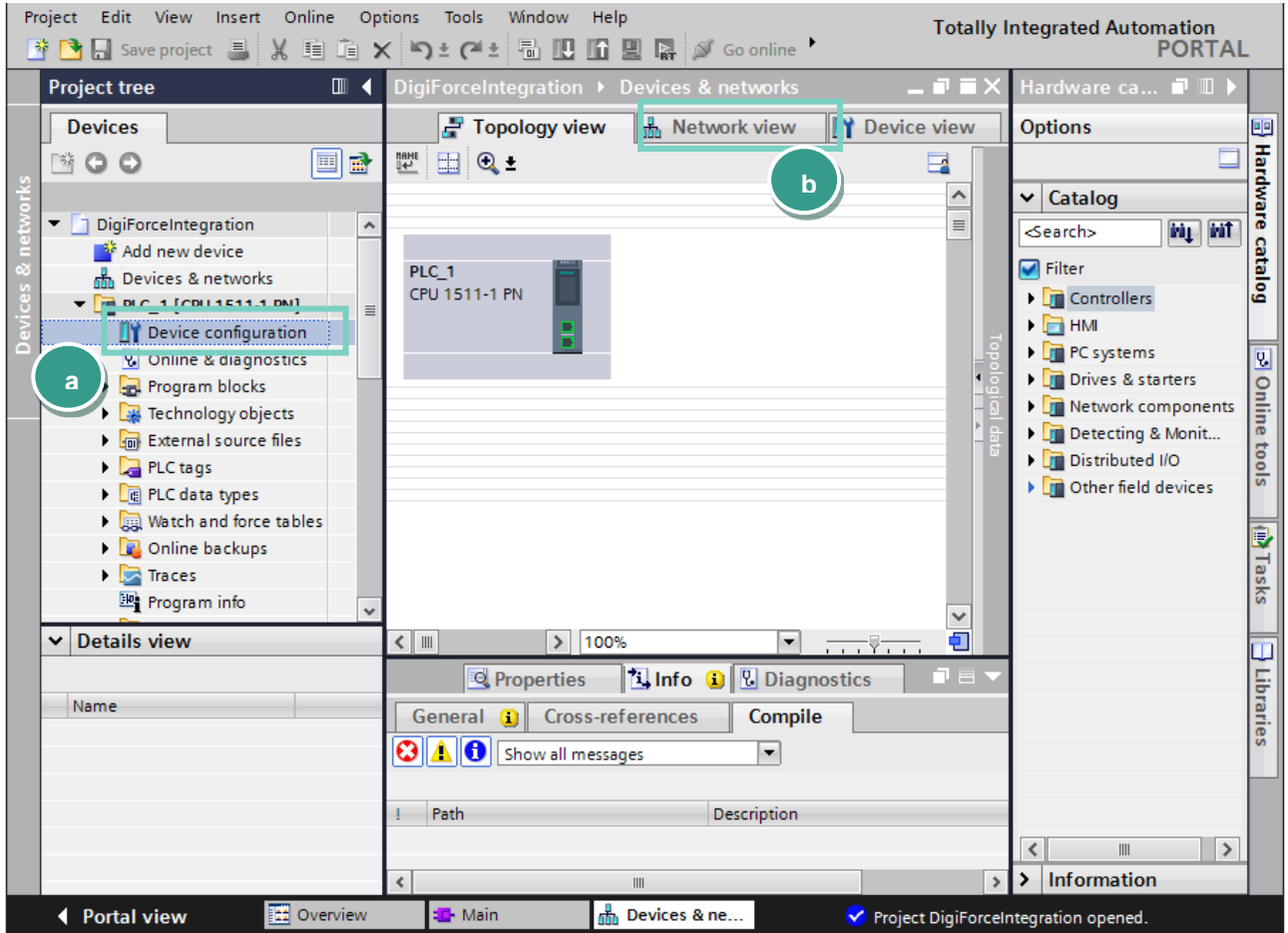
To be able to exchange a device with another one later without setting the PROFINET station name of the new device manually you can allow overwriting of device names by the PLC-Controller.

Go to the the properties of your PLC Controller and then to *Interface options* and *Permit overwriting of device names...*

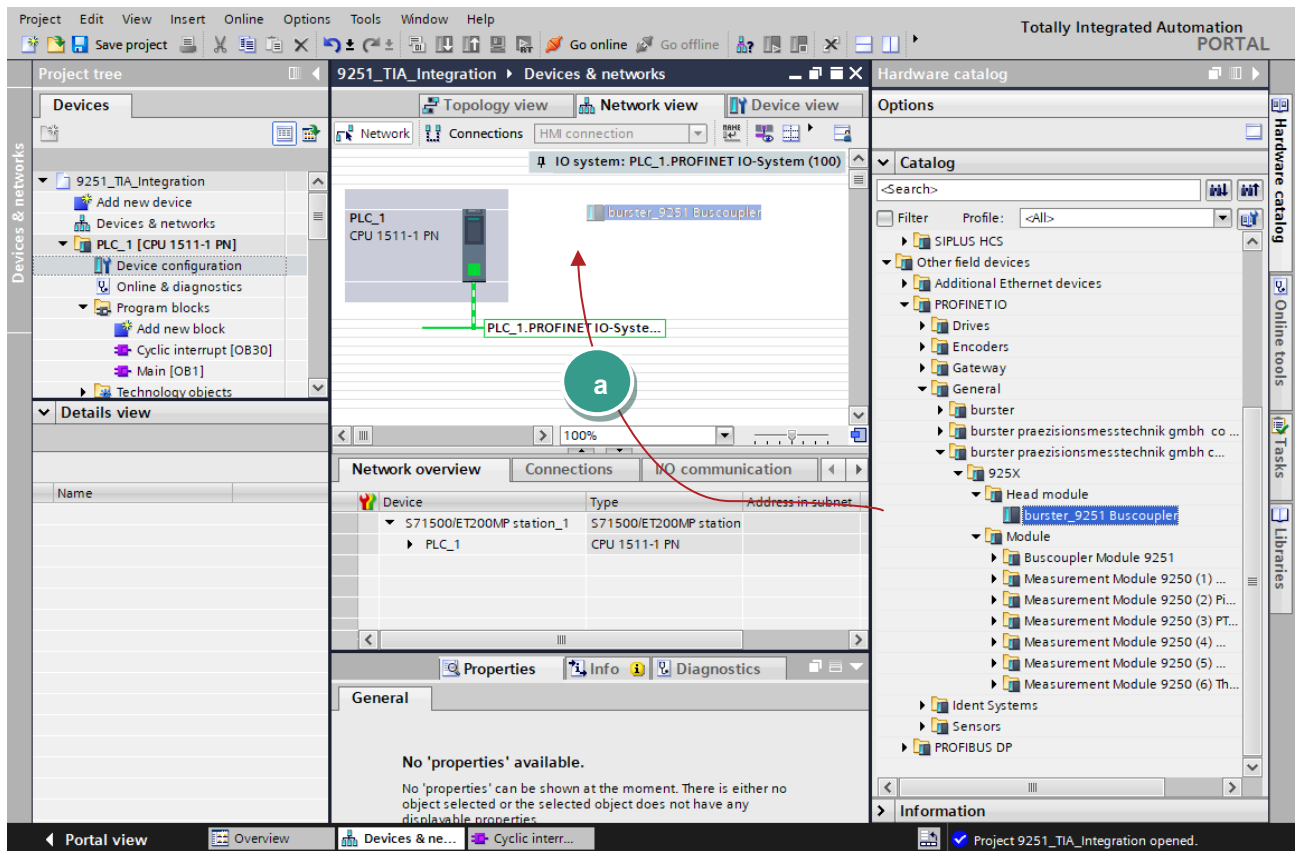


4. Creation of network connections

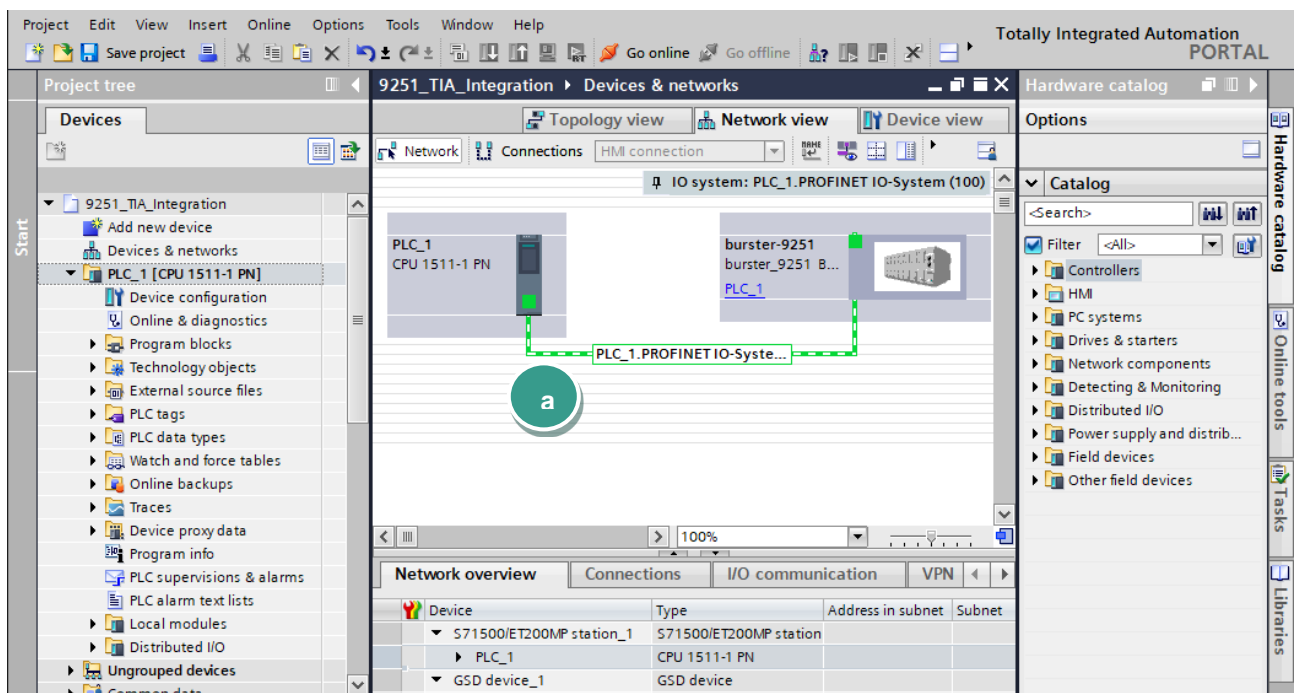
- Double click **Device Configuration** (a) in the project tree und switch to **Network view** (b) :



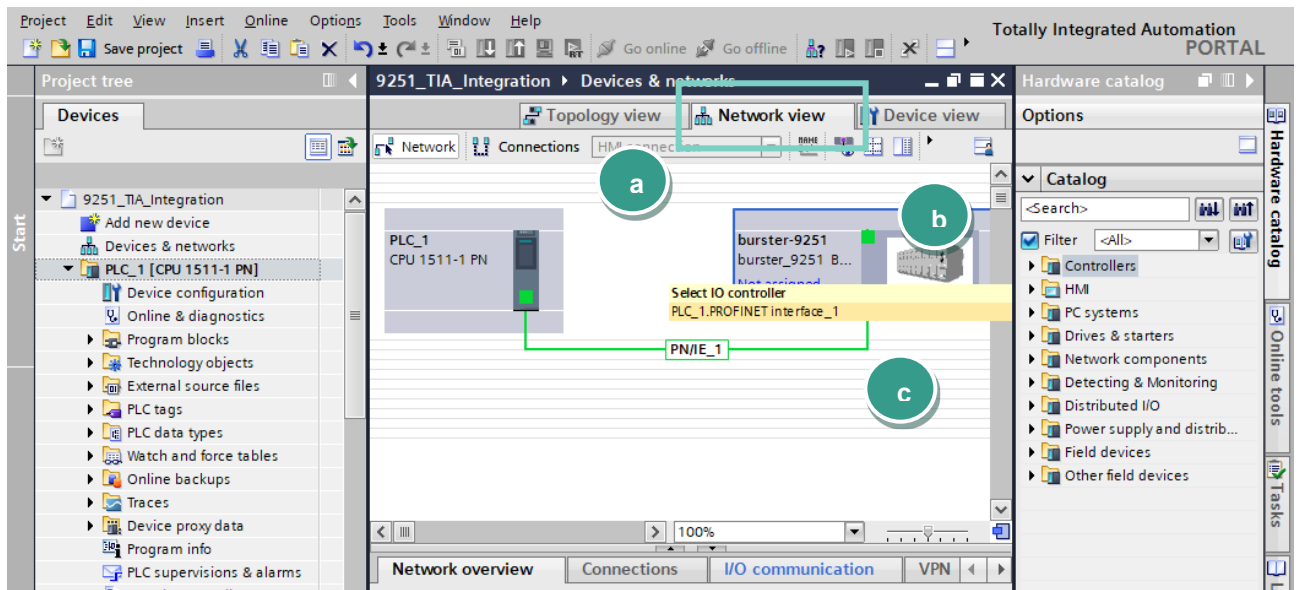
- Now select the **burster_9251_Buscoupler** device in the catalog and drag & drop it into the working area (a):



- Select an Ethernet port on the S7 and hold the left mouse button down to connect the S7 with 9251:

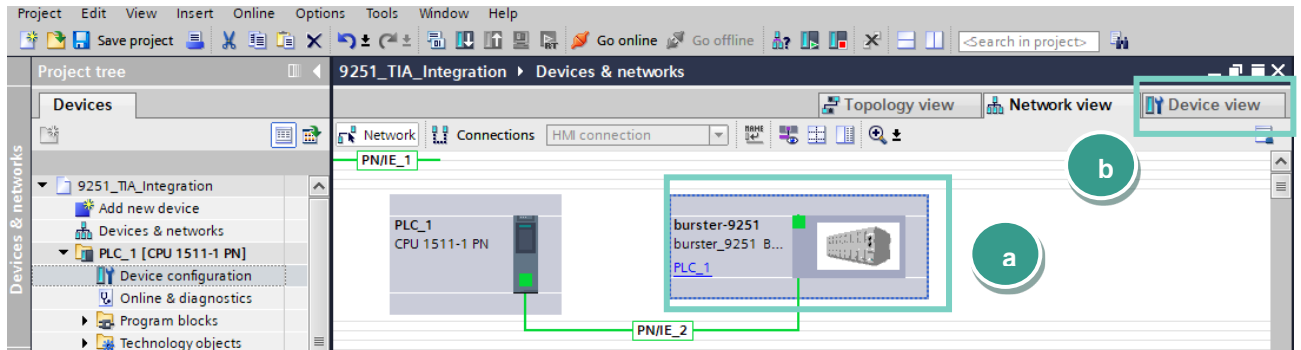


- If device is not automatically assigned, change now to **Network view** (a) to assign a controller to the 9251. Click on the link "Not assigned" (b) of 9251 and select your controller (c):



Note: Check if devices also connected physically to the right ports. You find the port number assignment directly on the device front side.

- Next, click on the *burster-9251* device (a) and then switch to **Device view** (b)

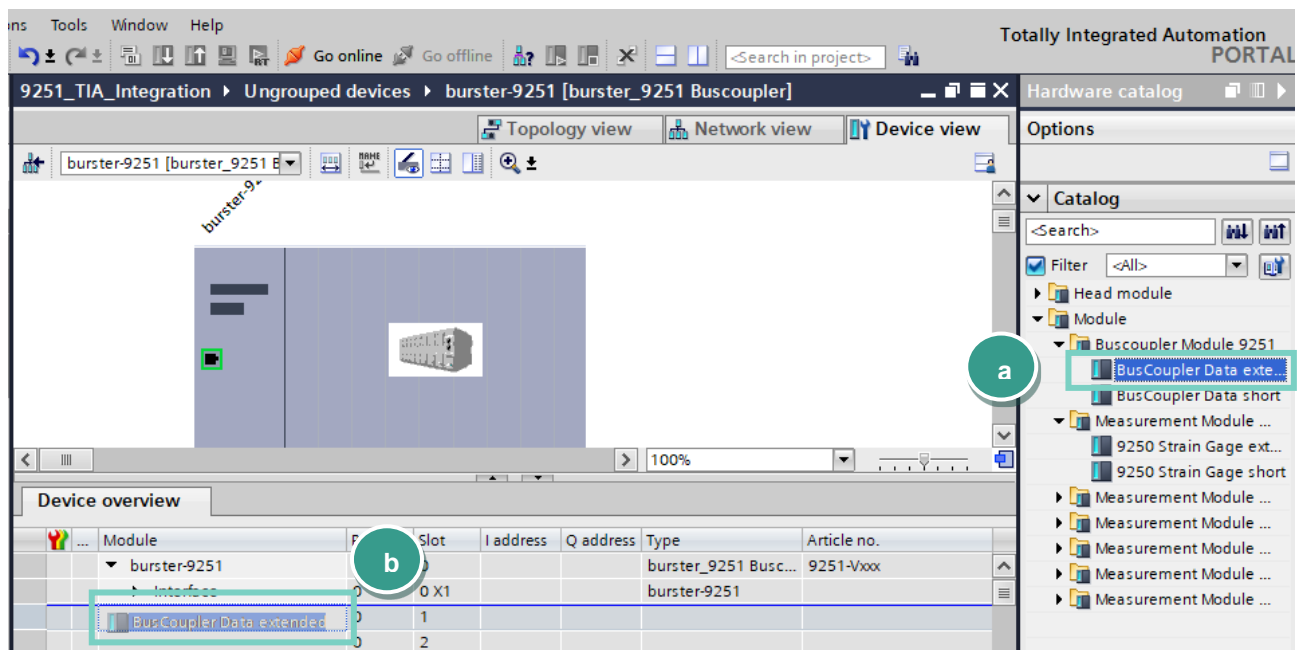


- Now, remove the entry **BusCoupler Data short 1** (a) from the **Device overview**

Device overview							
...	Module	Rack	Slot	I address	Q address	Type	Article no.
▼	burster-9251	0	0			burster_9251 Busc...	9251-Vxxx
	▶ Interface	0	0 X1			burster-9251	
	▼ BusCoupler Data short 1	0	1			BusCoupler Data sh..	
				0...7		BusCoupler Data sh..	
					0...15	BusCoupler CTRL	

Change device	
Start device tool...	
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Delete	Del

- in addition, insert the entry **BusCoupler Data extended** (a) into the **Device overview** table (b), you can drag it or just double click on the entry:



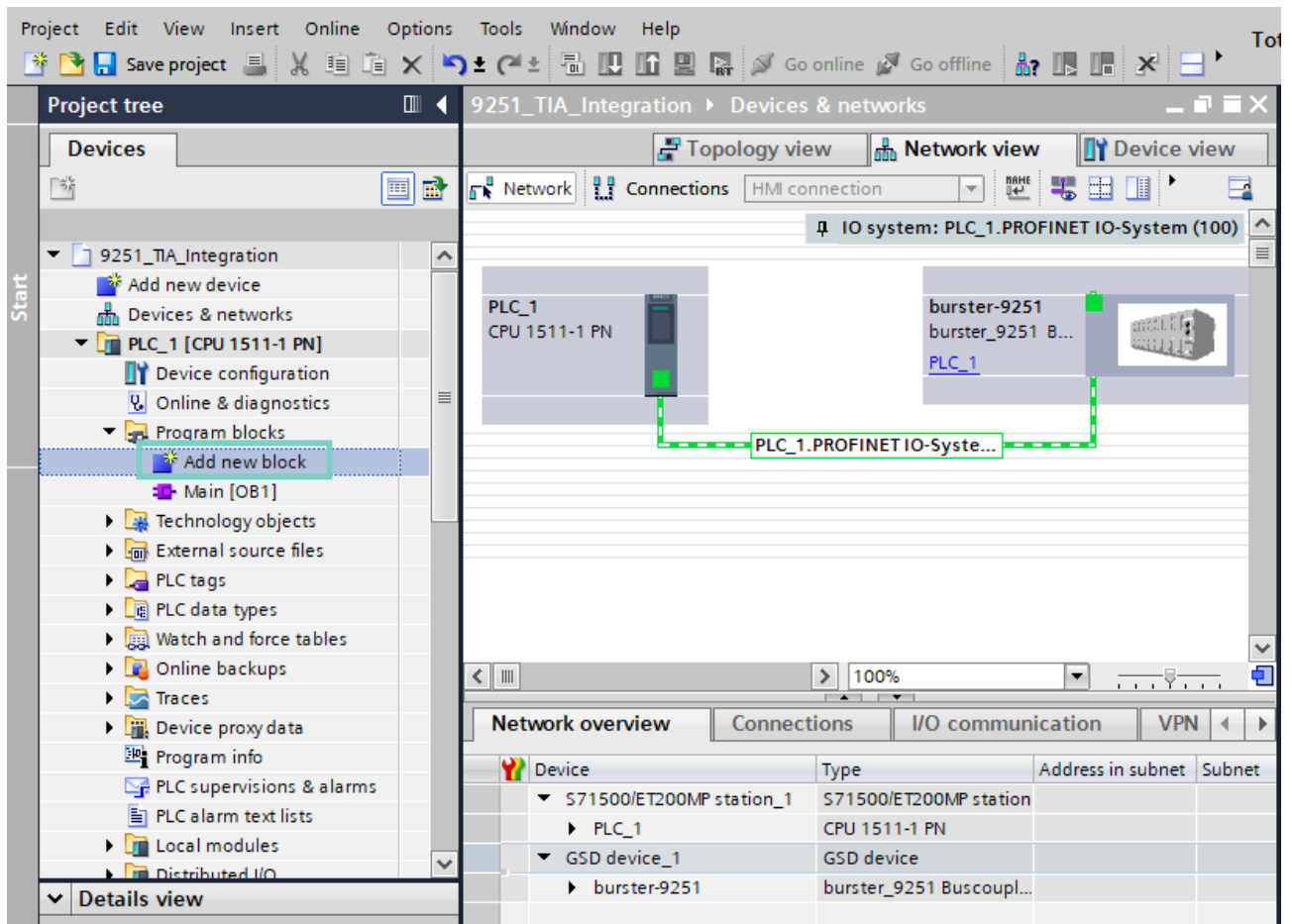
Note: In comparison to the “short” version, allows the **extended** version the receiving of 32 last measurement values, so you can expand this example to show all 32 last values.

If you have a modular device with several 9250 modules and would like to communicate with those modules, you just have to drag the modules (depending on how many modules are connected) the module “Measurement Module 9250 (1) Strain Gauge” (short or extended) from the module tree into the table Device overview.

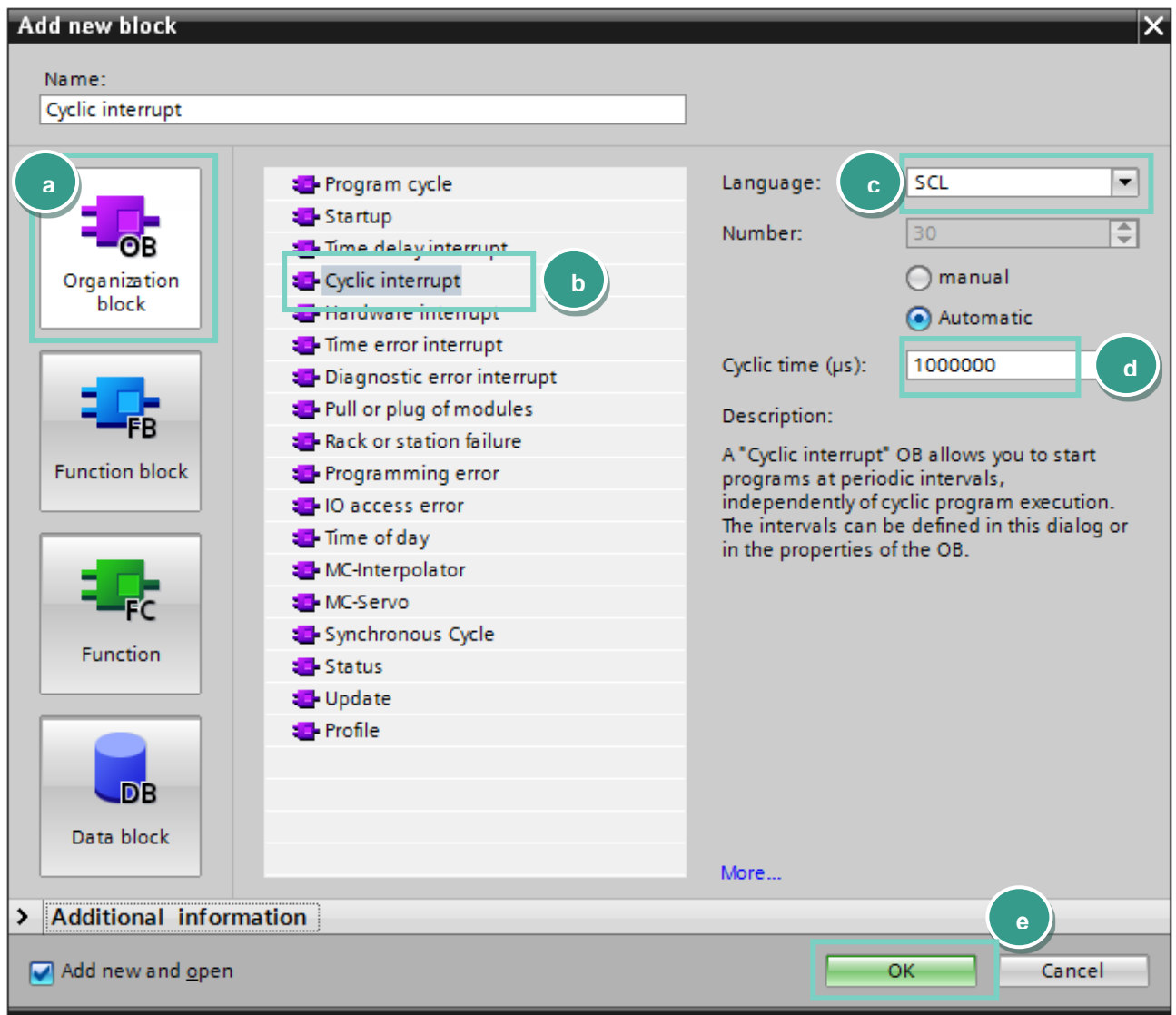
5. Create a sample program

In this section, you will learn how to create a simple program to read a measurement value from the analogue input of the bus controller. You will need to refer *Cyclic Input Data* of the **9251 PROFINET Interface Documentation** manual to understand the meaning of input bytes.

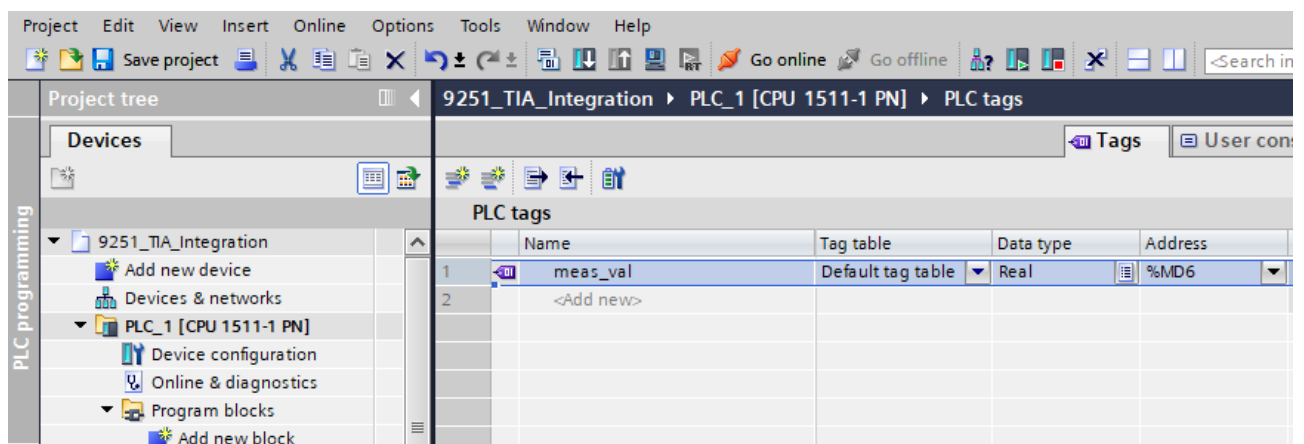
- Expand the tree node **Program blocks** in the **Project tree** and double click **Add new block**:



- Select in the new window **Organization block** (a) and then **Cyclic interrupt** (b). As language set SCL (c), change the cyclic time to 1.000.000 µs (d) and click OK (e):



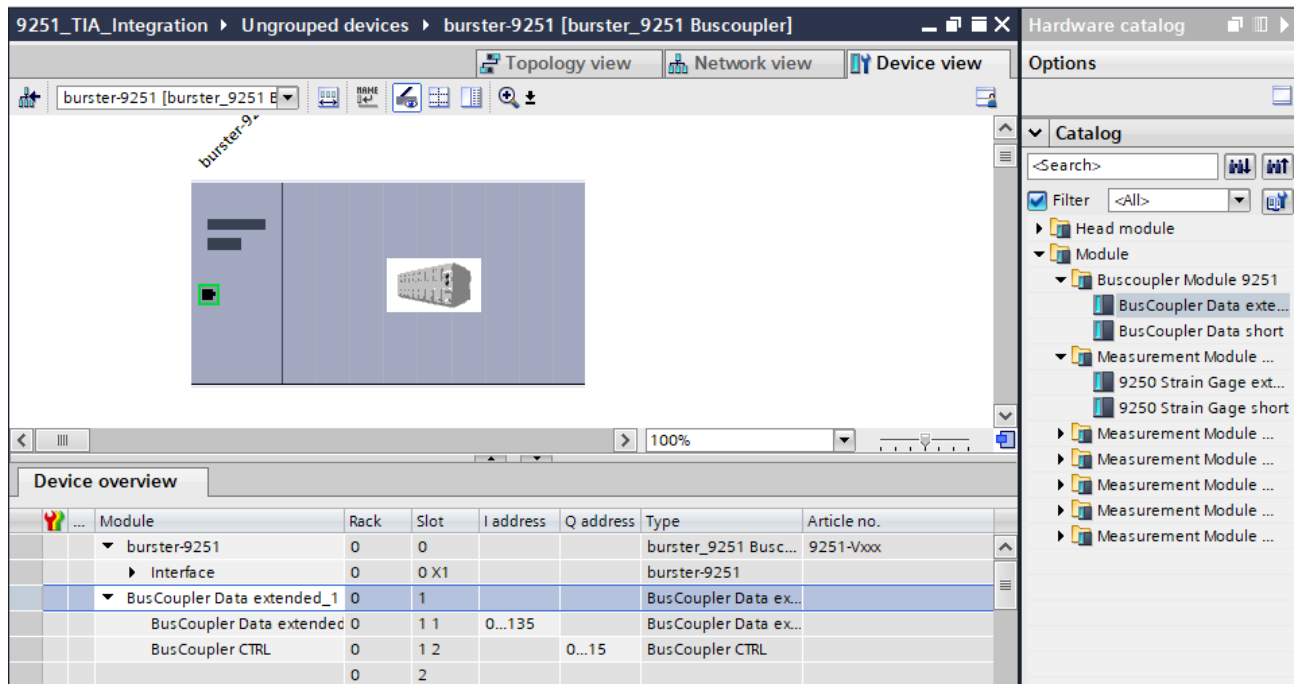
- Declare a new variable **meas_val** under **PLC Tags**



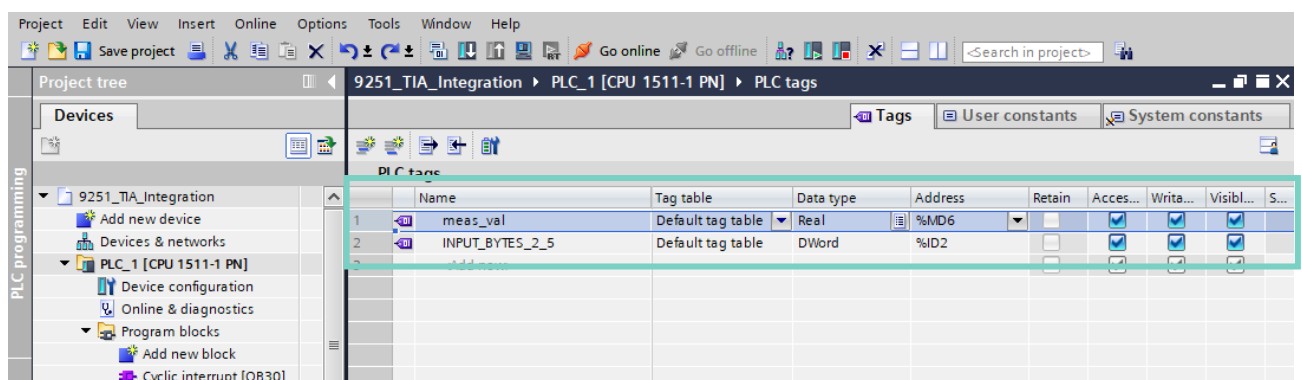
- then type in the following source code in the code field of the new block:

```
"meas_val" := DWORD_TO_REAL("INPUT_BYTES_2_5");
```

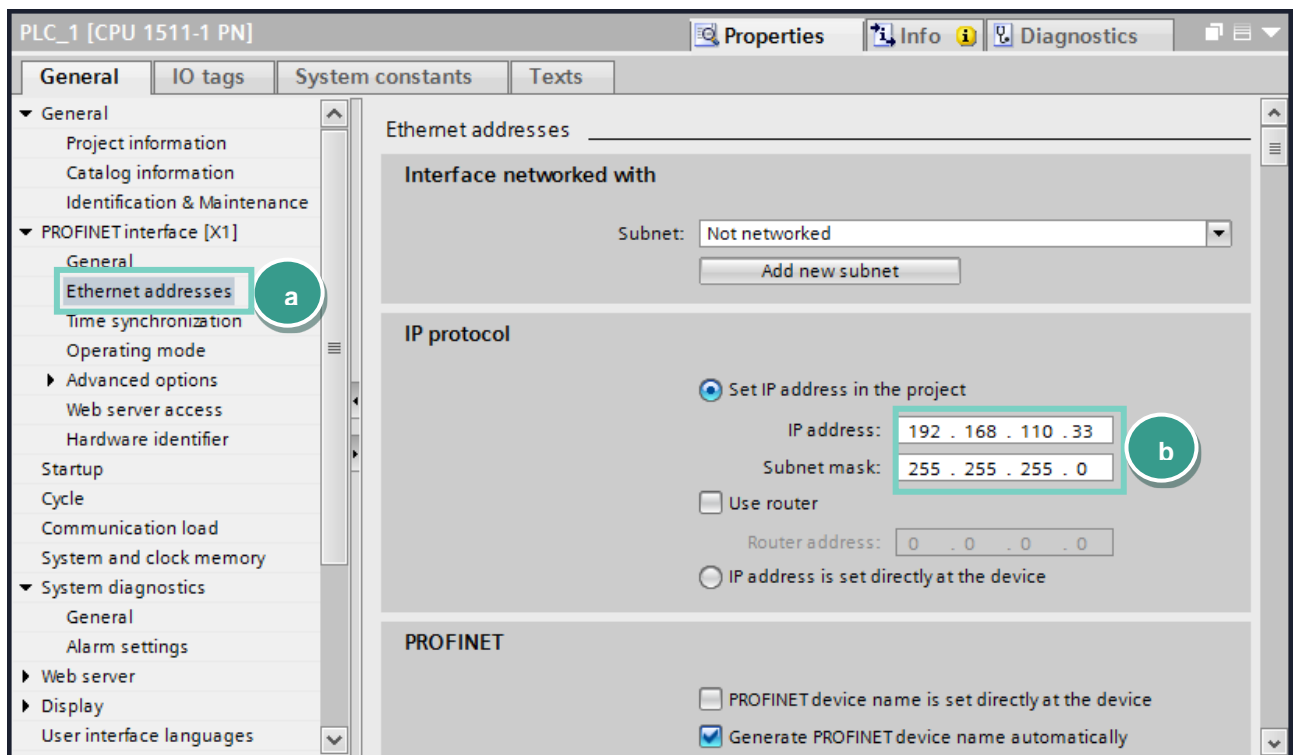
Please note: the addresses may be different. You have to check them in the **Device view** → Device overview of the 9251.



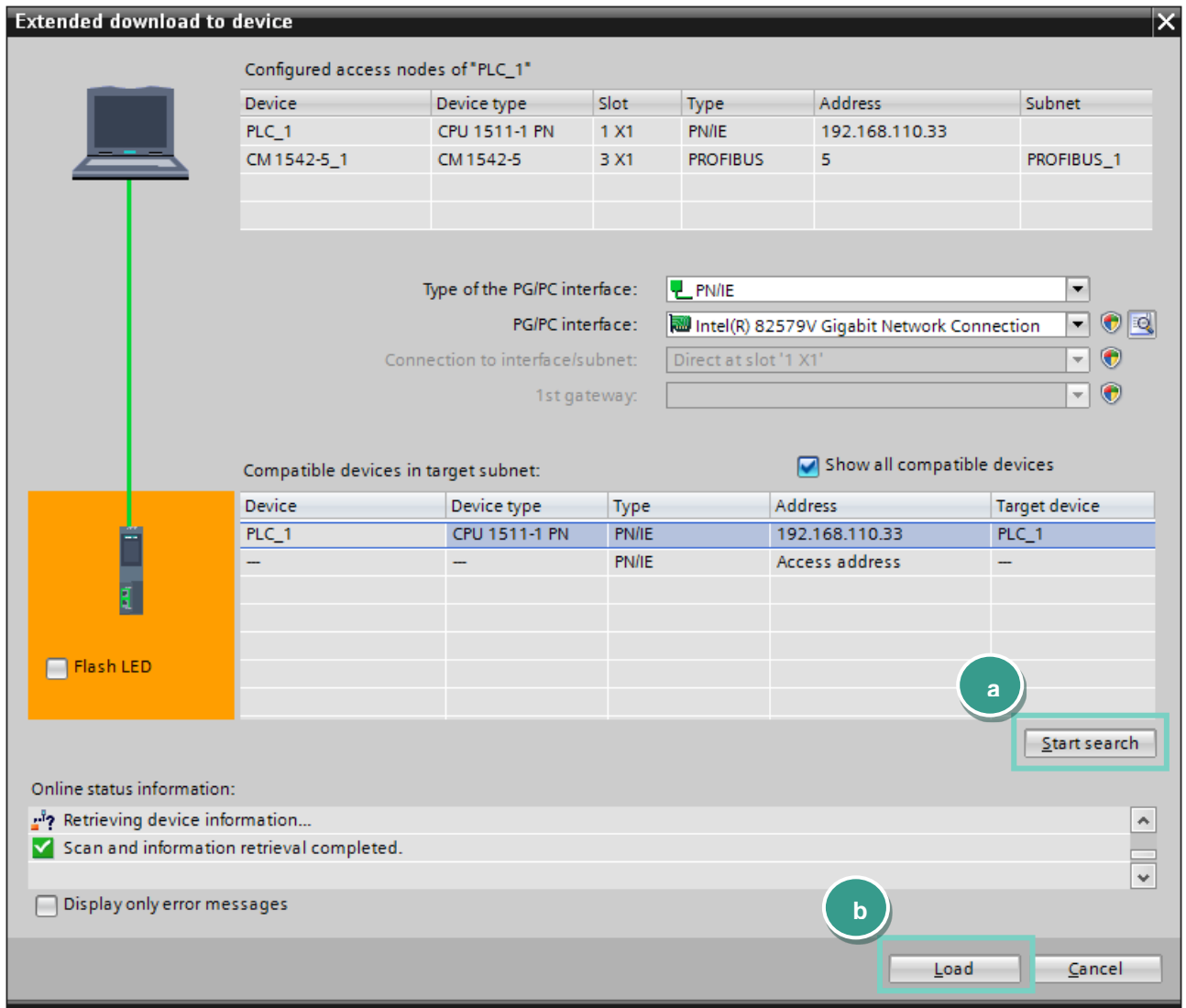
You will also see that the TIA-Editor replaces the input addresses with tags. You can change the tags names in PLC Tag table (e.g. to **IN_BYTES_2_5**):



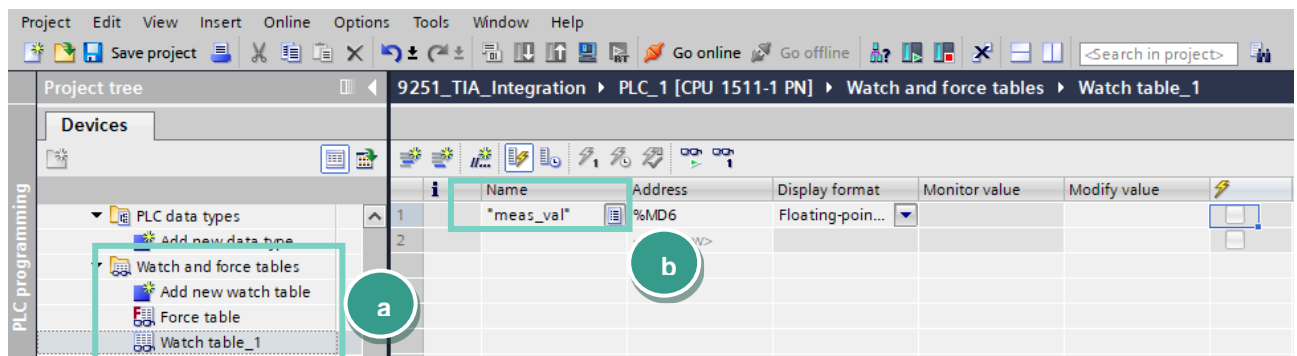
- Before you load the project into the CPU you have to set the IP address of your CPU. To do this please go to **Device view** and select **Ethernet addresses** (a) in **General** tab. Set now the IP-Address and a subnet mask(b) for your PLC:



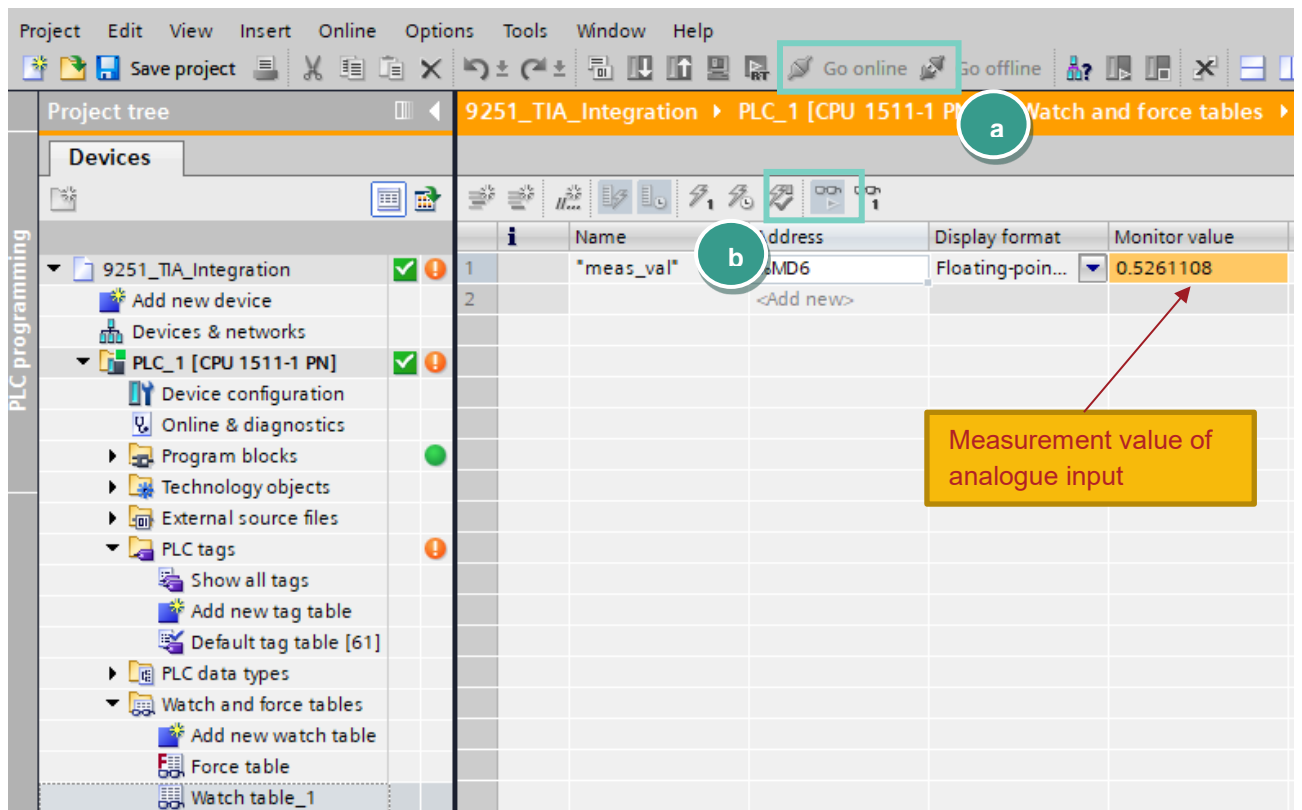
- To load the configuration into the CPU select it first go to **Online** → **Download to device** and click on **Start search** (a) to look for your controller. Then select the controller and click on **Load** (b):



To watch the measurement value, go to **Watch and force tables** (a) → **Add new watch table** (a) und add the variable **meas_val** to the table (b):

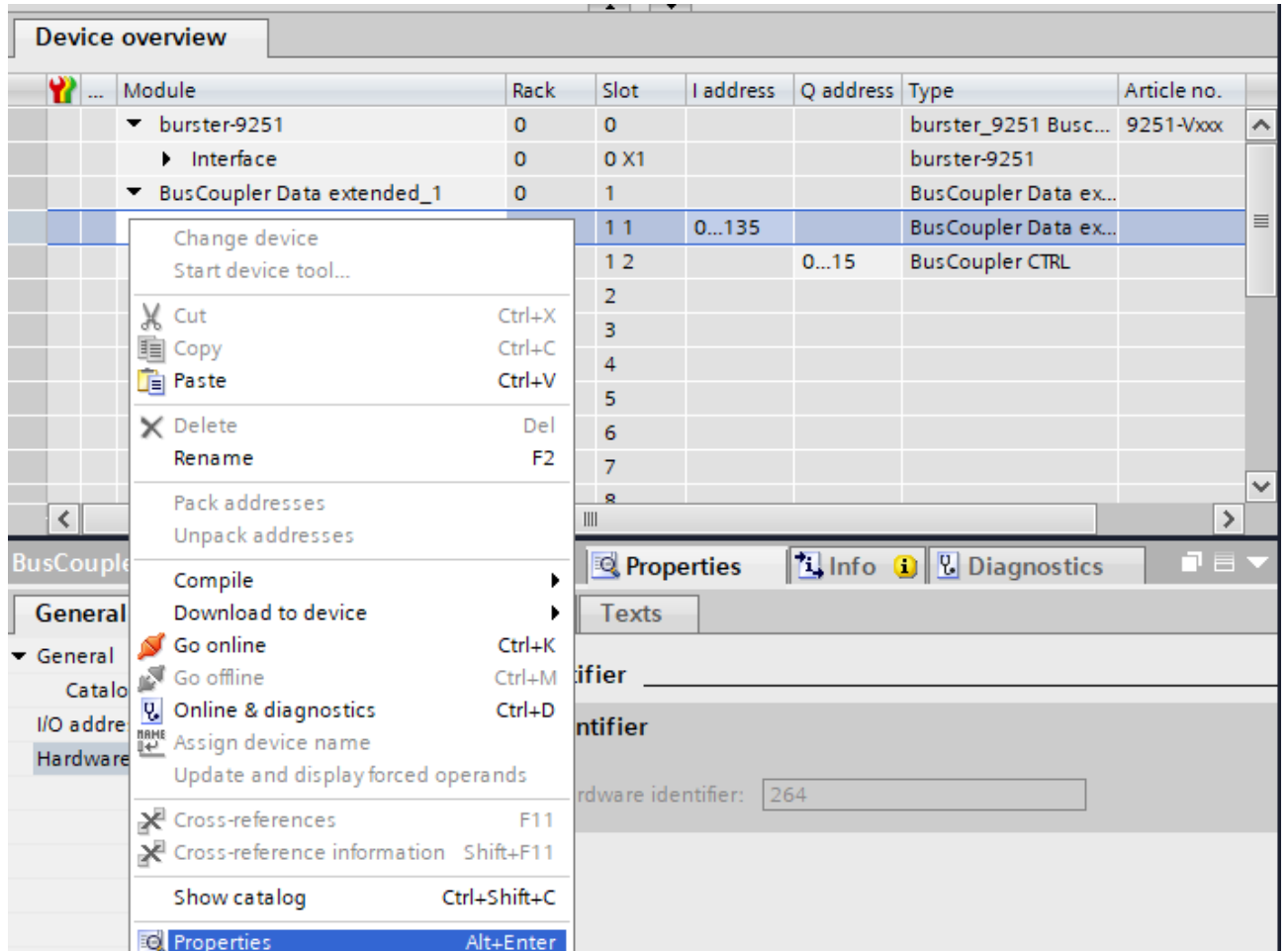


Now click on “Go online” (a) and then “Monitor all” (b) to watch the value of **meas_val**

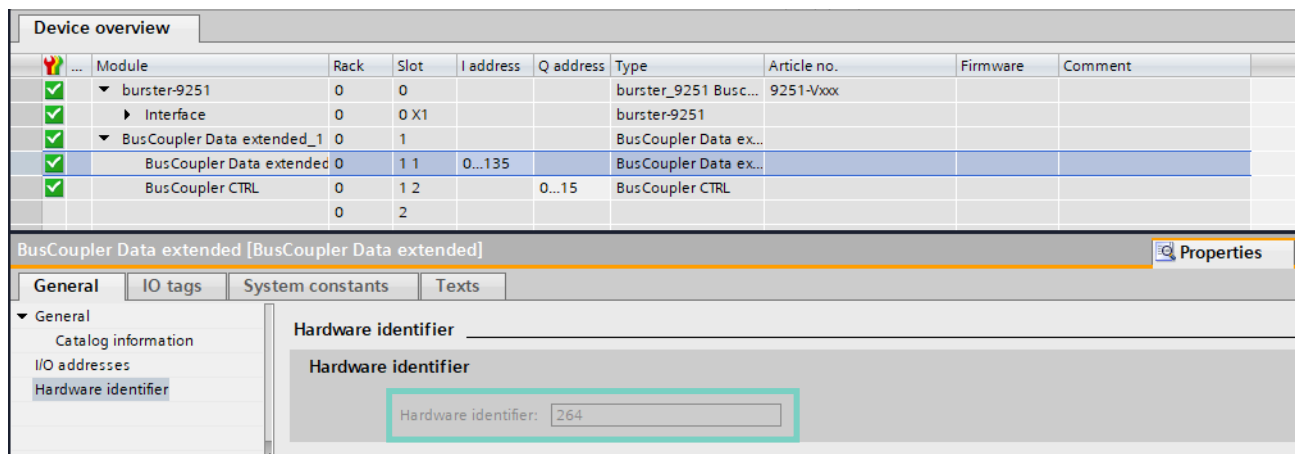


6. Further Examples

In the followed examples, a *Hardware-ID* is used to access a certain slot. To find this, please select a **burster-9251** device in **Topology view** or **Network view** and then switch to **Device view**. Click with the right mouse button on the desired module, e.g. *BusCoupler Data extended* and select **Properties**:



You will see the hardware identifier in the tab **General**:

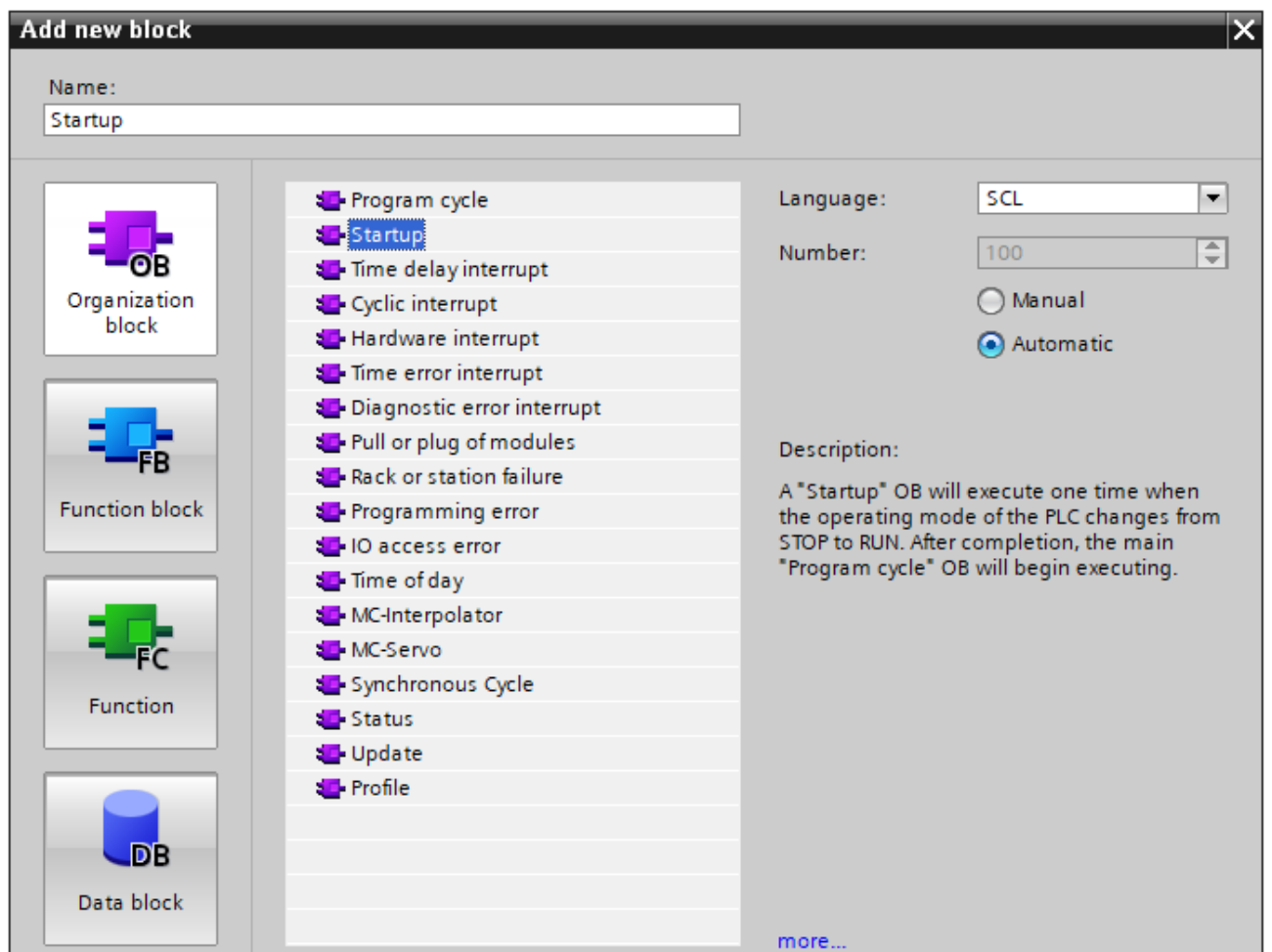
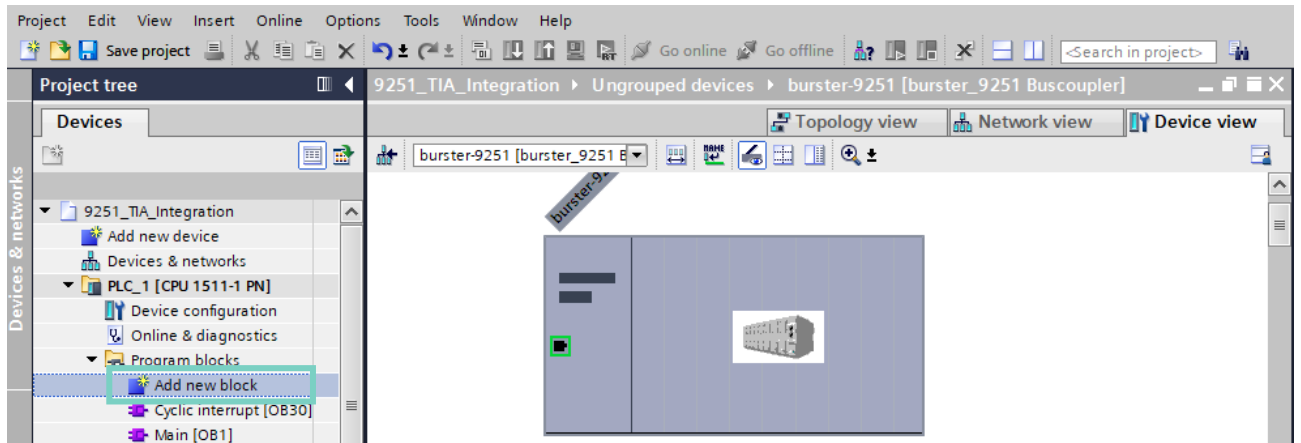


5.1 Reading of 'string' data types

Example 1: Reading the serial number of the 9251

In this example, we perform a read access on index 15 to get the serial number of 9251. For these acyclic operations, you will need an instance of a RDREC block.

Add a **Startup block** to the Program blocks using **Add new block**:



- Add variables to the **Startup** block:

		Startup				
	Name	Data type	Default value	Supervision	Comment	
4	Temp					
5	Valid	Bool				
6	Busy	Bool				
7	Error	Bool				
8	Status	DWord				
9	Done	Bool				
10	lenRead	UInt				
11	serial	Array[0..19] of Byte				
12	<Add new>					
13	Constant					

- Then add a new **Data block**:

Add new block

Name:

Type:

Language:

Number:

Manual

Automatic

Description:
Data blocks (DBs) save program data.
[more...](#)

Organization block (OB)

Function block (FB)

Function (FC)

Data block (DB)

- In addition, insert the variable **device_serial** into this new block:

		Data_block_1						
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Static							
2	device_serial	String	<input type="text" value=""/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	<Add new>			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- Use the following source code to get the serial number from the device:

Source code:

```

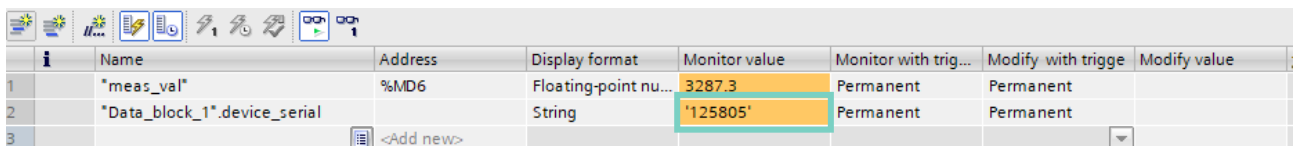
REPEAT
"RDREC_DB"(REQ:=TRUE,
    ID:=264,           // 263: HW-ID (see introduction of 'Further examples')
    INDEX:=15,        // Index 15: Serial
    MLEN:=20,         // Max. length of data to read
    VALID=>#Valid,    // New Data received and valid
    BUSY=>#Busy,      // Read not completed yet
    ERROR=>#Error,    // Error
    STATUS=>#Status,  // State
    LEN=>#lenRead,    // Number of bytes was read from device
    RECORD:= #serial); // Array[0..19] of Byte
UNTIL NOT #Busy
END_REPEAT;

IF #Error = TRUE OR #Status <> 0 THEN
    RETURN;
END_IF;

Chars_TO_Strg(Chars:=#serial,
    pChars:=0,
    Cnt:=19,
    Strg=>"Data_block_1".device_serial);
    
```

Example 2: Reading of serial number

- Add now the variable to your **Watch table** and set the PLC into the **RUN** mode:



	Name	Address	Display format	Monitor value	Monitor with trig...	Modify with trigge	Modify value
1	*meas_val*	%MD6	Floating-point nu...	3287.3	Permanent	Permanent	
2	*Data_block_1*.device_serial		String	'125805'	Permanent	Permanent	
3	<Add new>						

Note: Datatype **String** in TIA Portal contains two additional bytes, which represent the length of the string. To cut off these two bytes, use the function '*Chars_TO_Strg*' to convert the byte array to a String data type.

5.2 Read and Write of 'real' data types

Example 3: Set and Get the Limit A – Lower Value

This example shows you how to write and read the Limit A – Lower Value.

- Add the needed variables to the **parameters table**:

Startup_Limit_A_LOW					
	Name	Data type	Default value	Supervision	Comment
4	Temp				
5	Valid	Bool			
6	Done	Bool			
7	Busy	Bool			
8	Error	Bool			
9	Status	DWord			
10	LenRead	UDInt			
11	limit_a_low_write	Real			

- Add the variable **limit_a_low** to a data block as shown in the example 1 above

Data_block_1								
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	device_serial	String	"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	limit_a_low	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	<Add new>			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- Use the following source code to write the value 0.525 as Limit A Lower Value and then read it back:

```
#limit_a_low_write := 0.525;

REPEAT
  "WRREC_DB"(REQ := TRUE,
    ID := 264, // Write access
    INDEX := 6, // Hardware-ID (see section 'Further examples')
    LEN := 4, // Index 6 to set the Limit A lower
    DONE => #Done, // Length in bytes to write
    BUSY => #Busy,
    ERROR => #Error,
    STATUS => #Status,
    RECORD := #limit_a_low_write);
UNTIL NOT #Busy AND #Done
END_REPEAT;

IF #Error = TRUE OR #Status <> 0 THEN // If write failed → return
  RETURN;
END_IF;

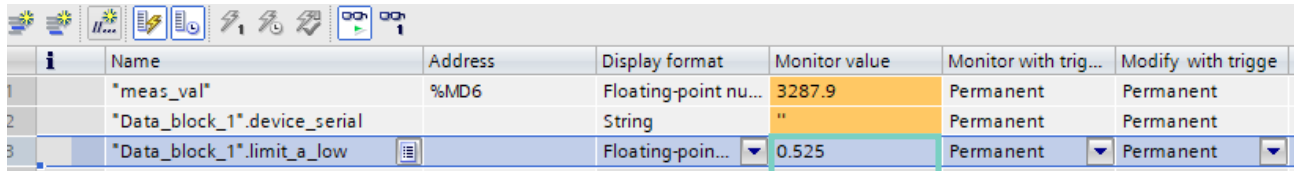
REPEAT
  "RDREC_DB"(REQ := TRUE,
    ID := 264, // Read access
    INDEX := 6, // Hardware-ID (see section 'Further examples')
    MLEN := 4, // Index 6 to set the Limit A lower
    VALID => #Valid, // Max. length of bytes to read
    BUSY => #Busy,
    ERROR => #Error,
    STATUS => #Status,
```

```

        LEN => #lenRead,
        RECORD := "Data_block_1".limit_a_low);           // Number of bytes read
UNTIL NOT #Busy AND #Done                             // Limit A lower value
END_REPEAT;
    
```

Example 3: Writing and Reading of limit a lower value

- Check that the variable **limit_a_low** is identical to the value of the variable **limit_a_low_write**:



	Name	Address	Display format	Monitor value	Monitor with trig...	Modify with trigge
1	"meas_val"	%MD6	Floating-point nu...	3287.9	Permanent	Permanent
2	"Data_block_1".device_serial		String	"	Permanent	Permanent
3	"Data_block_1".limit_a_low		Floating-poin...	0.525	Permanent	Permanent