OPERATION MANUAL

Resistomat 2311
PROFINET Integration into TIA Portal

Valid from: **01.02.2023**
Applies to: **Resistomat 2311-VXX03**

4426-2311PROFINETIAEN-5999-081530

Contents

1 Introduction

This quick start guide describes an approach how you can configure the RESISTOMAT® 2311 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 beed extremely simplified to reach a better understanding. Therefore, you may have to complete them by checking of status, error, length values etc.

Note: *Please also note that you will have to use the RESISTOMAT® 2311 PROFINET manual to get futher information about input and output parameters (cyclic as well acyclic data transfer).*

2   Creating new project

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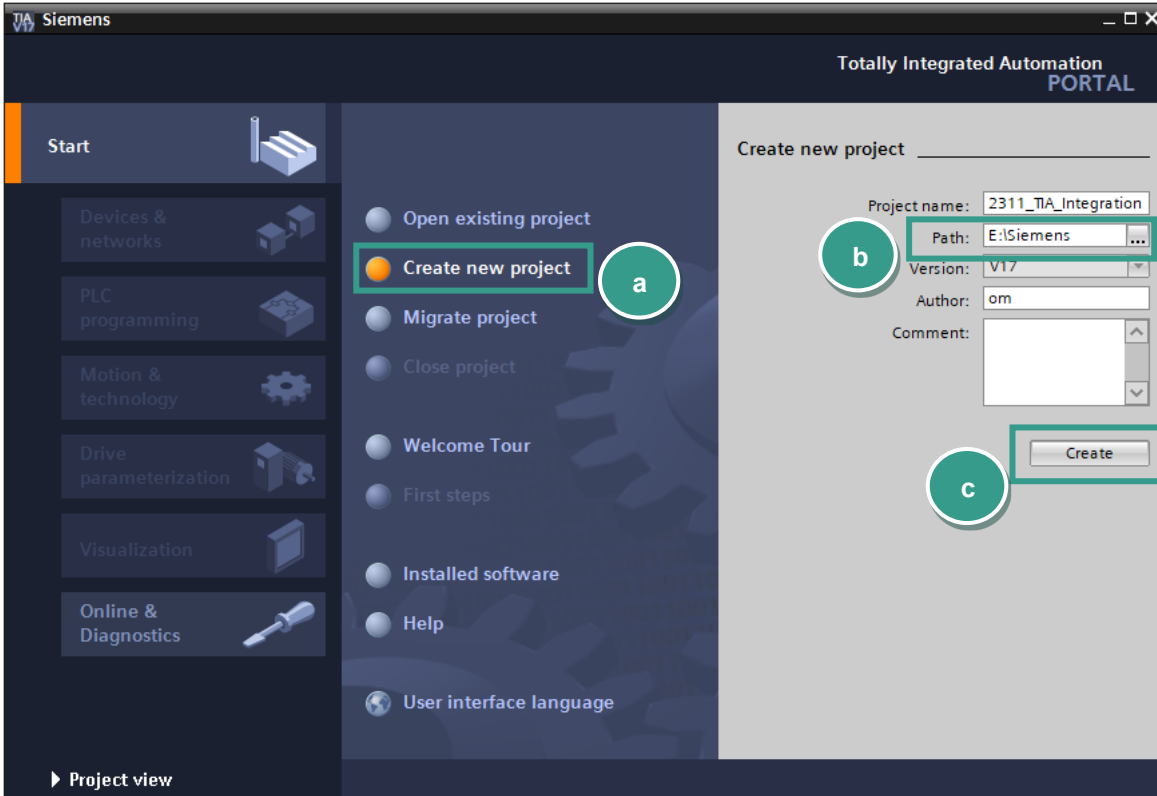

2. Go to ***Devices & networks*** (a) on the left side select ***Add new device*** *(b)* and look for yor CPU (c). Afterwards click the **Add** button (d).
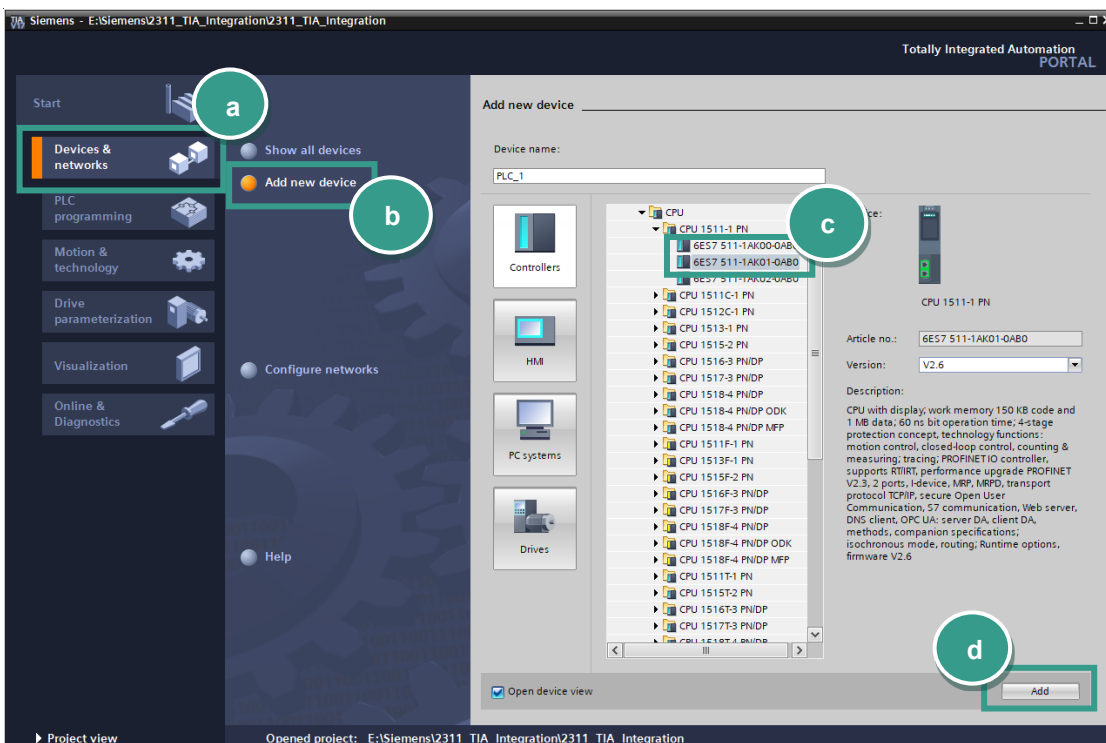
3 Installation of GSDML files

Note: Please make sure that your GSDML file is compatible to the field bus firmware in the Resistomat 2311. Also for compatibility reasons, uninstall all previous GSDML files of particular device if you have any!

1. Go to Options→Manage general station description files (GSD) (a)
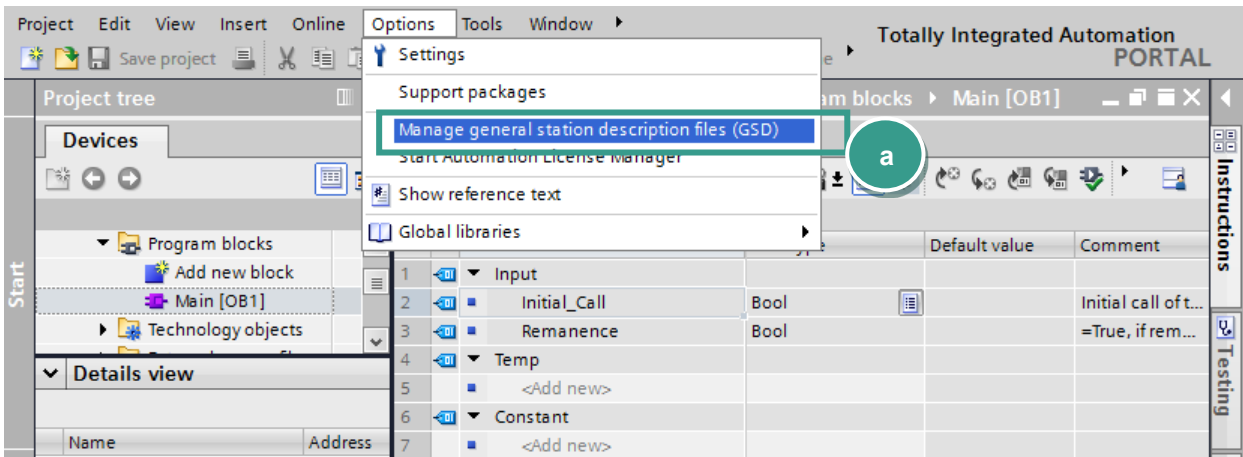
2. Navigate to your Resistomat 2311 GSDML directory (a) (you will find the GSD on www.burster.com/en/download-area), select the GSDML file (b) and click **Install** (c)
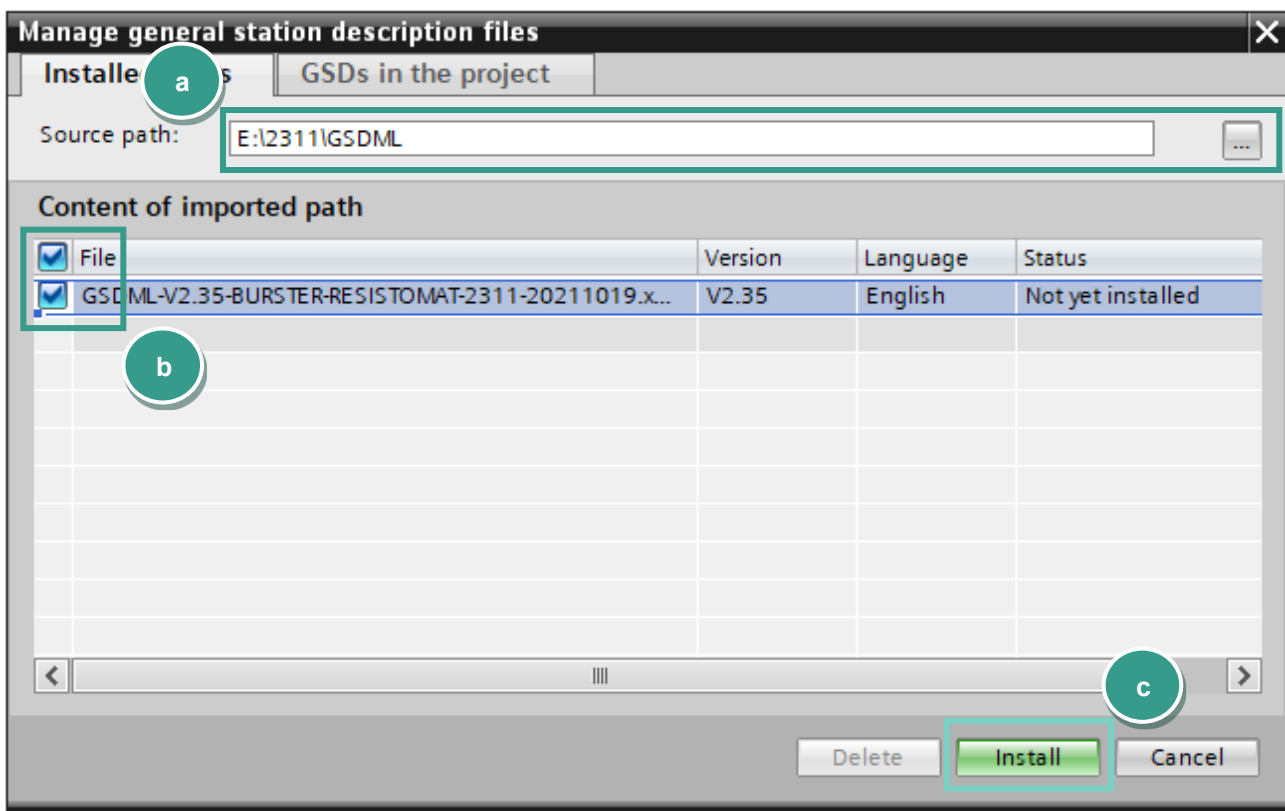
4 Creation of network connections

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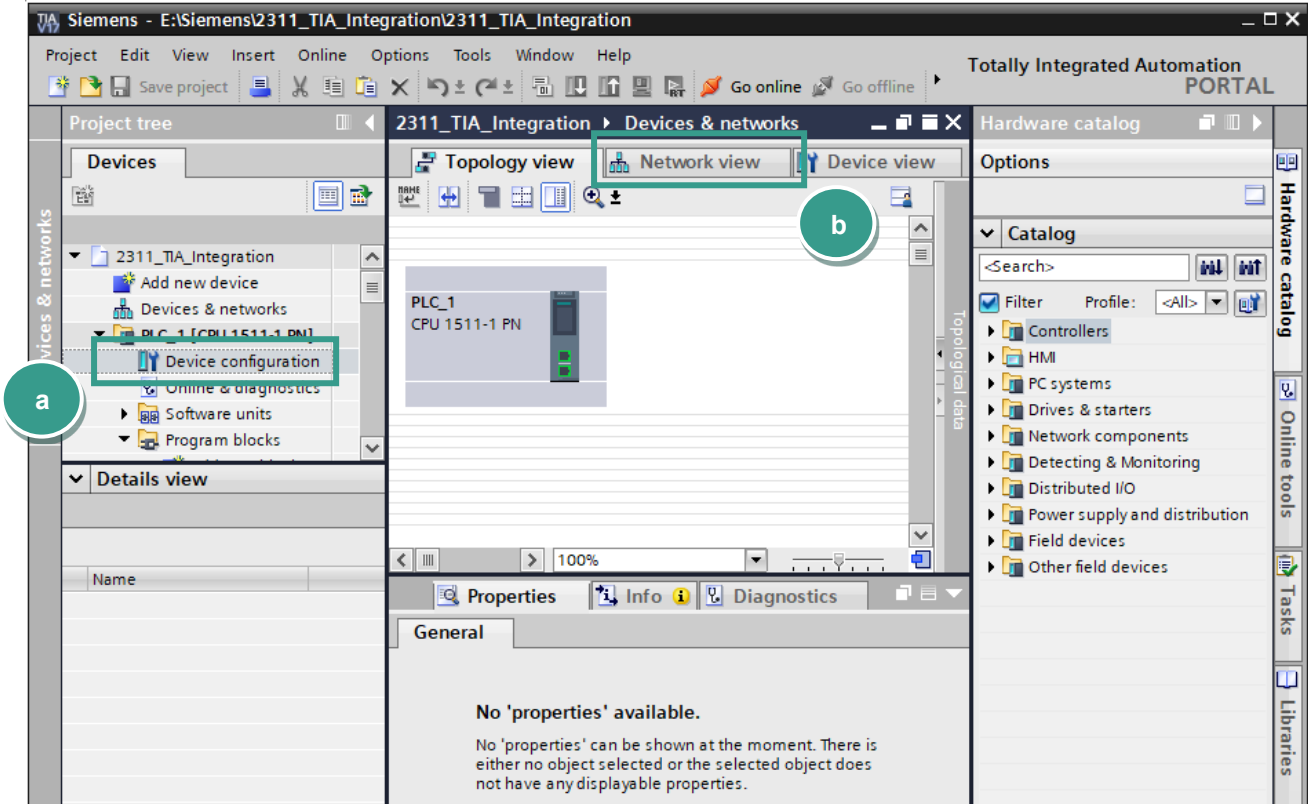

2. Now select the Resistomat 2311 device in the catalog and drag & drop it into the working area (a):
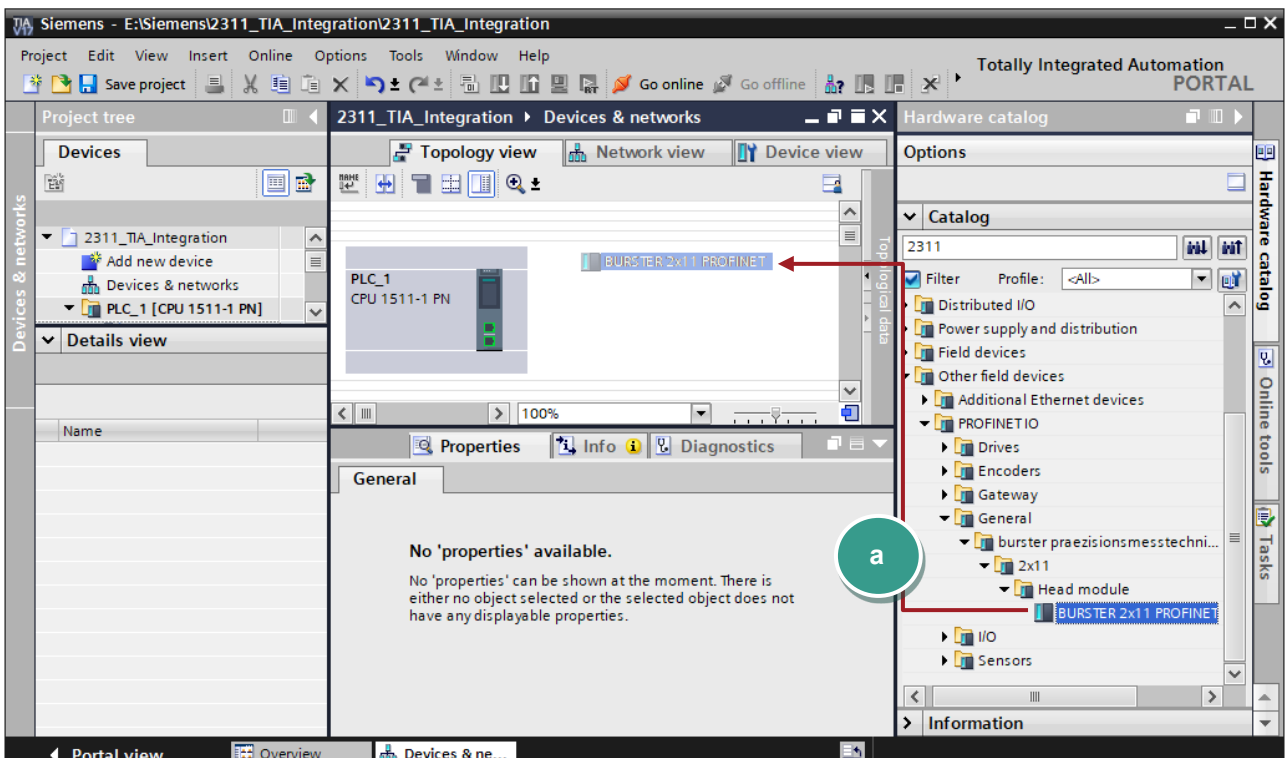
3. Please select an ethernet port on the S7 and hold the left mouse button down to connect the S7 with Resistomat 2311 (a):
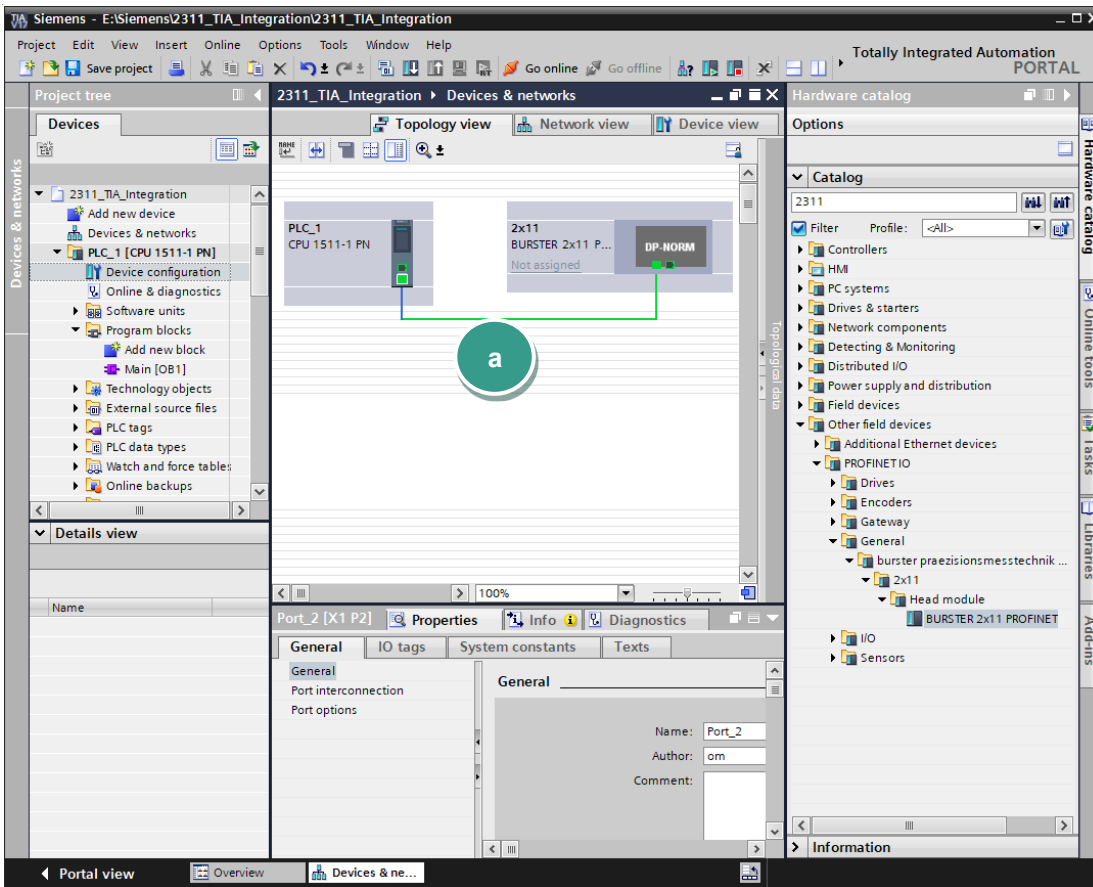
4. Change now to **Network view** (a) to assign a controller to the Resistomat 2311. Click on the link "Not assigned" (b) of Resistomat 2311 and select your controller (c):
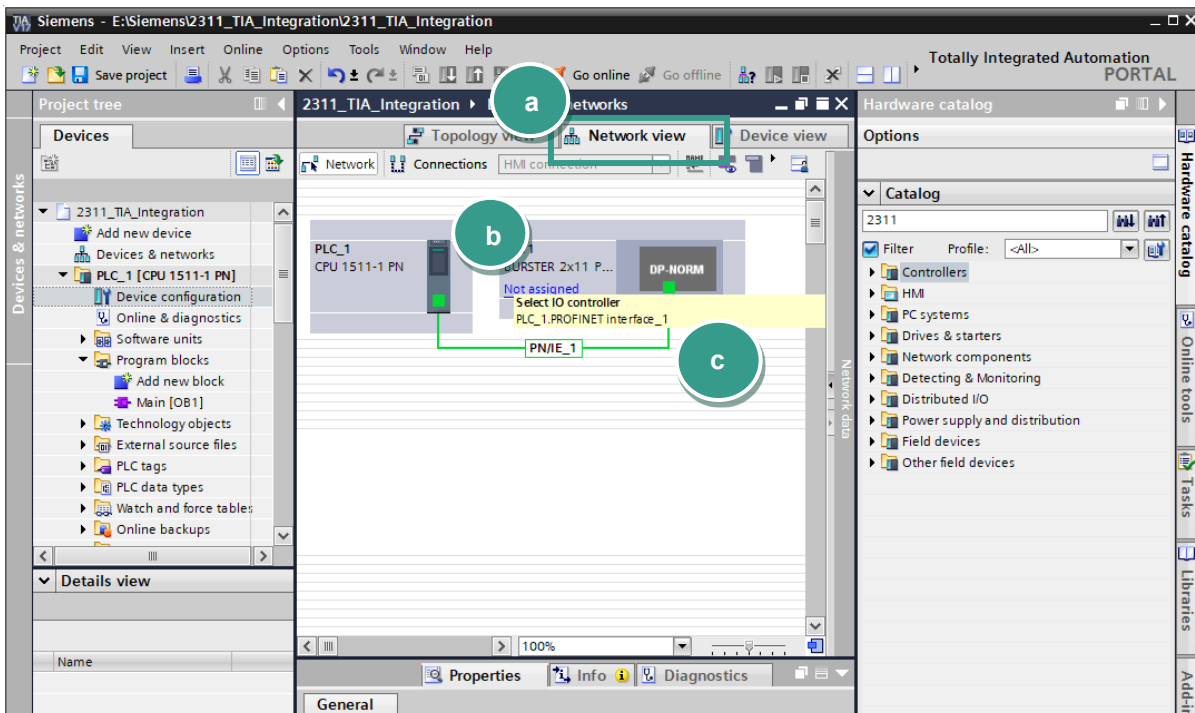
IMPORTANT: Check if devices are also connected physically to the right ports. You find the port number assignment in the section *Port-Identification of* **Resistomat 2311 PROFINET** manual

5. Change the device name in Device view → Properties → General → Ethernet addresses:
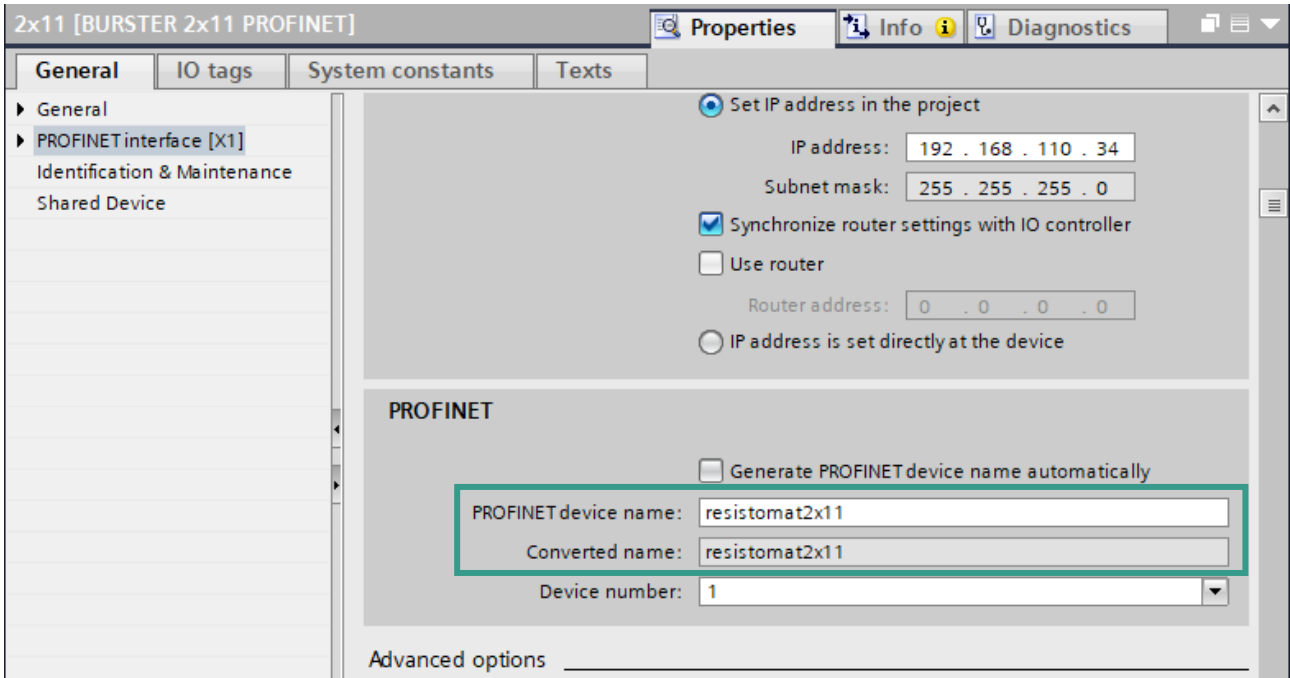
5 Perform a measurement and read the measurement result

In this section, you will learn how to create a simple program to start and stop a measurement periodically and display the measured resistance value with corresponding unit in a watch table. You will need to refer to sections *PLC inputs* and *PLC outputs* of the **RESISTOMAT® 2311**.

PROFINET manual to understand the meaning of inputs and outputs bytes.

1. Add the variables **out_start**, **in_ready** and **in_error** as shown below
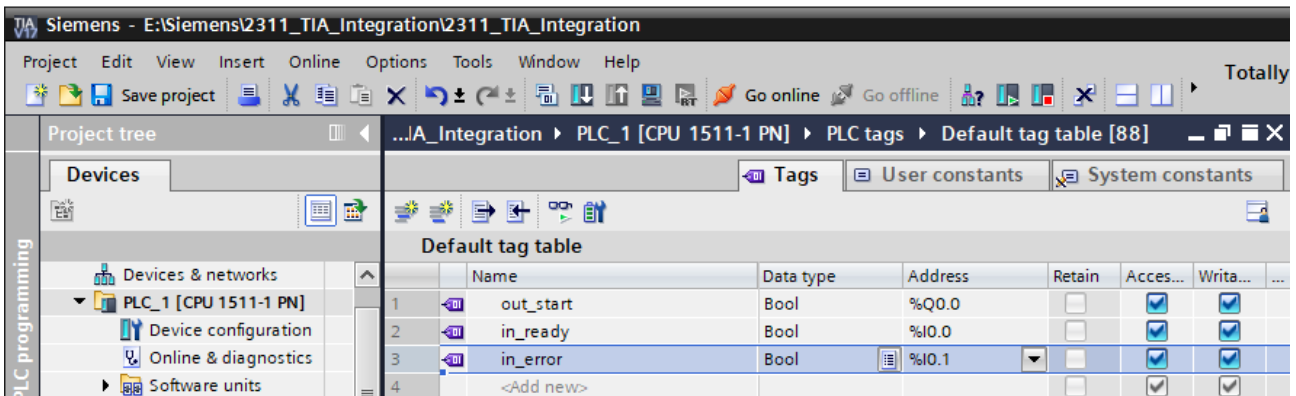
Note: **Please note**: the addresses may be different. You have to check them in the **Device view →** **Device overview** of the Resistomat® 2311:
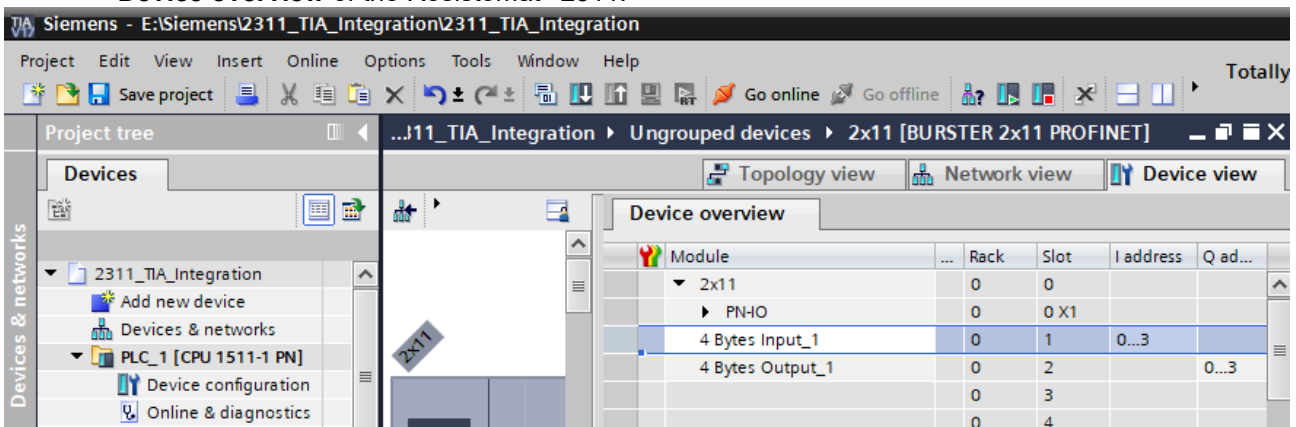
2. Expand the tree node *Program blocks* in the Project tree and double click *Add new block:*
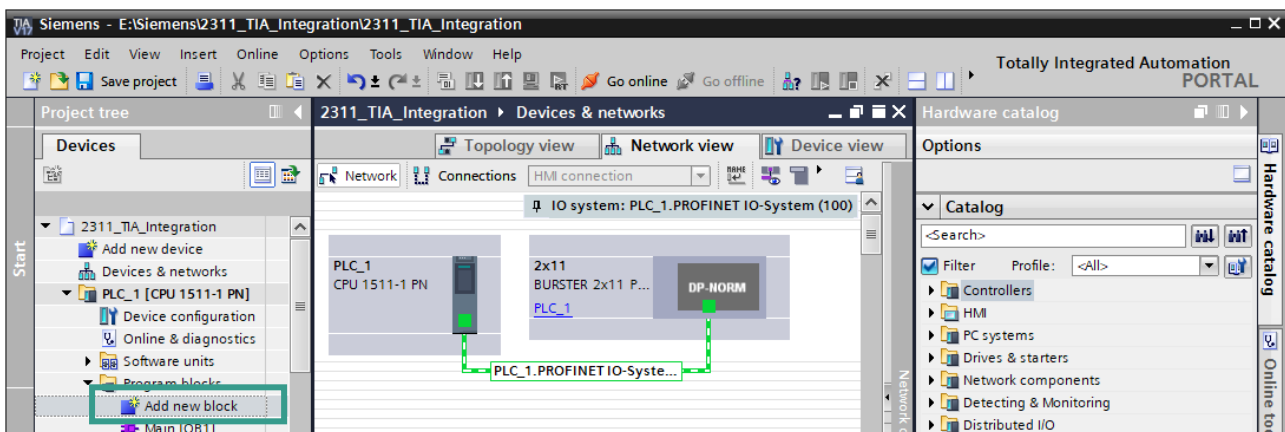
3. Select in the new window **_Organization block_** (a) → **_Cyclic interrupt_** (b)**.** Set the cyclic time to 3.000.000 µs (c)**.** As language set SCL (d) and click OK:
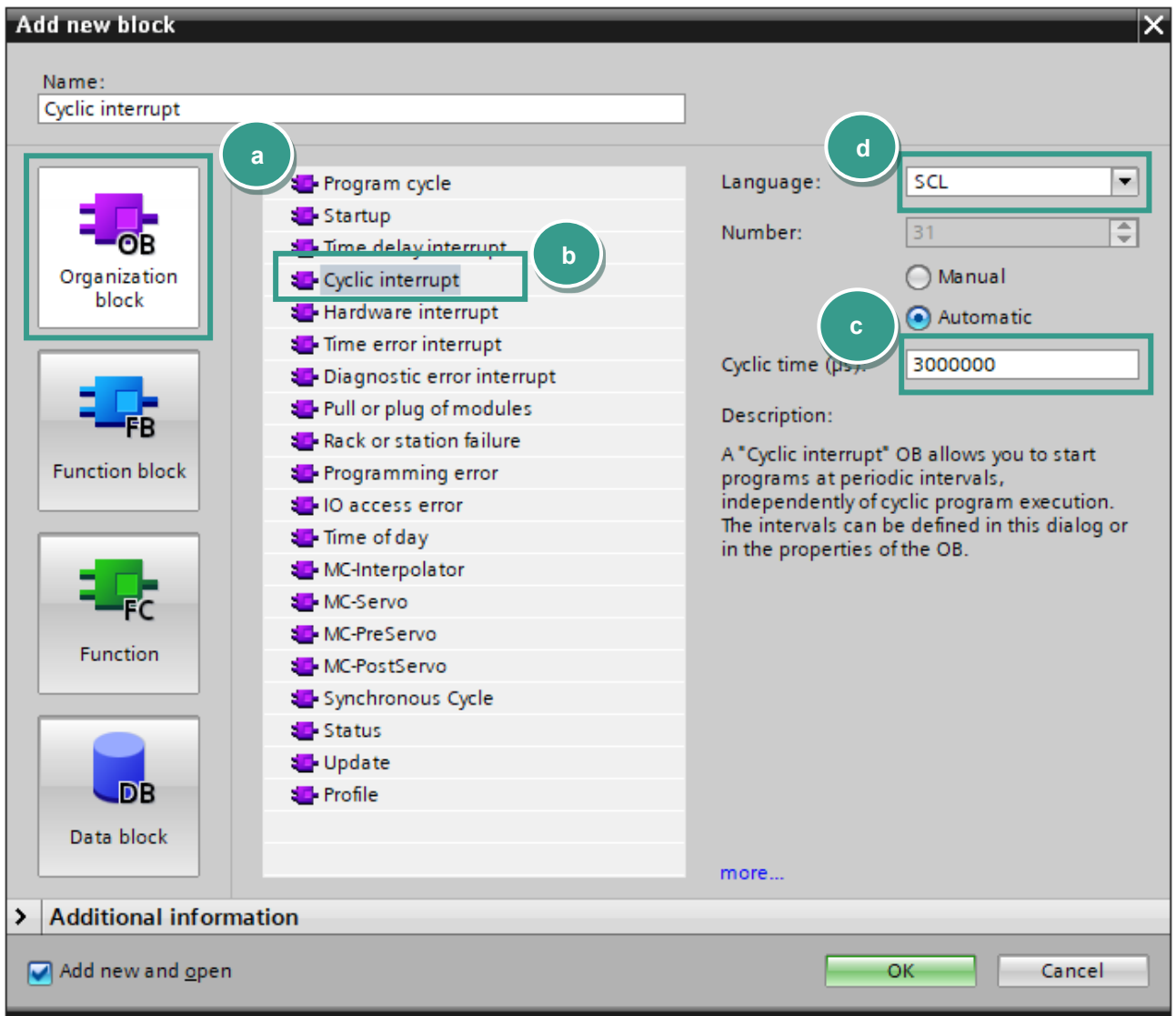
4. Type in the following source code in the code field of the new block:

```
IF "out_start" = TRUE THEN            // Is measurement start set?
    "out_start" := FALSE;             // Then reset it
    IF "in_error" = FALSE THEN        // No error?
        "read_meas_value"();          // Read the measurement value
    END_IF;
ELSE
    IF "in_ready" = TRUE THEN         // Is 2311 ready for a new measurement?
        "out_start" := TRUE;          // Start a new measurement
    END_IF;
END_IF;
```

5. Add a new function block and name it **read_meas_value**:
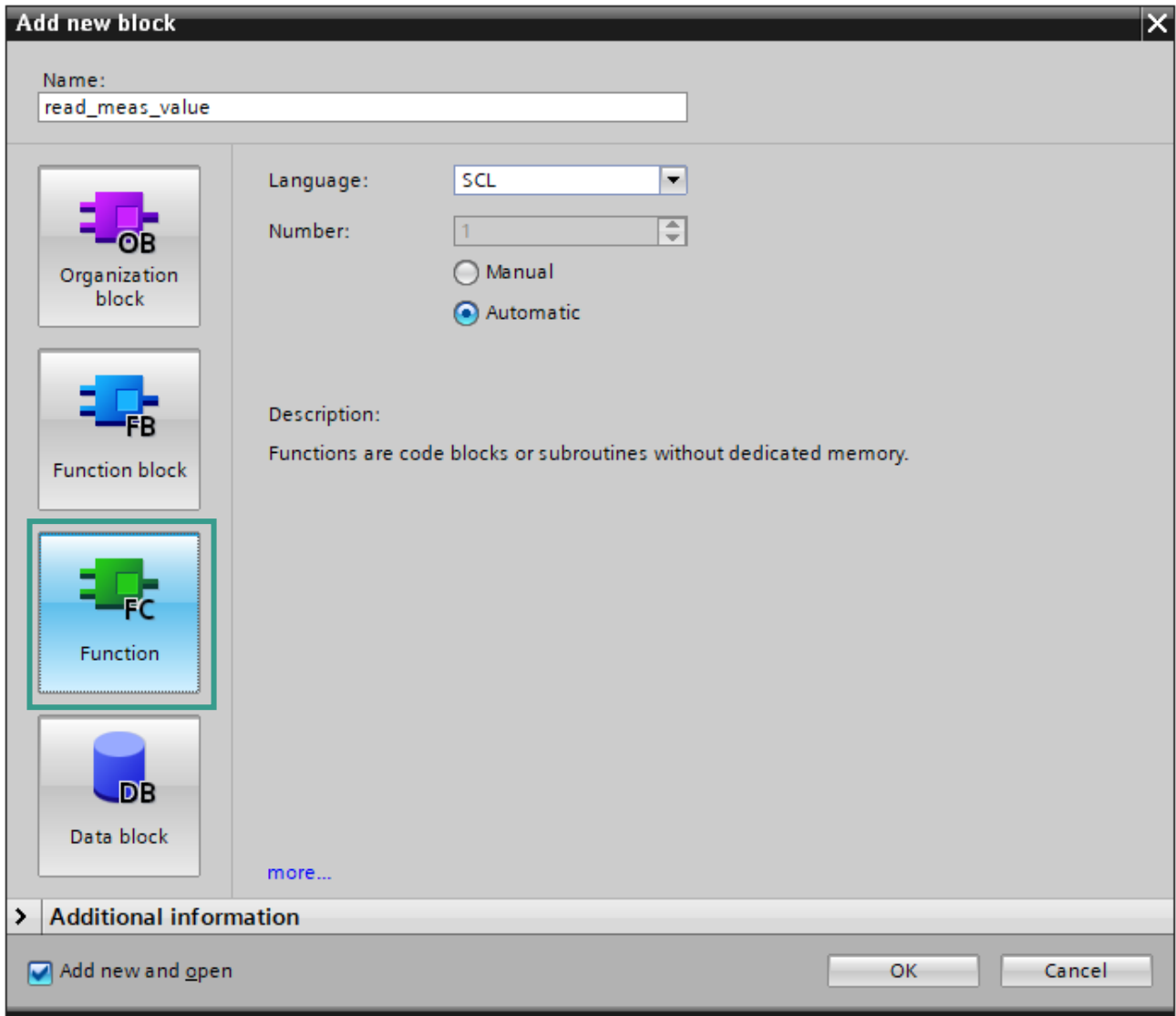
You will use this block to read the measurement value string, containing the measurement resistance value and the corresponding unit. For this acyclic operation, you will also need an instance of **RDREC** block.

Variables:

		Name	Data type	Default value	Comment
12	■	Done	Bool		
13	■	BytesWritten	UInt		
14	■	LenRead	UInt		
15	■	Data	Bool		
16	■	MeasStatus	UInt		
17	■ ▶	MeasValueAsByteArray	Array[0..63] of Byte		
18	▼	Constant			
19	■	<Add new>			
20	▼	Return			
21	■	read_meas_value	Void		

read_meas_value

6. Type in the following source code in the new function block:

Sourcecode:

```
"Data_block_1".MeasValueAsString := '---';          // Init Measurement value string

REPEAT
   "RDREC_DB"(REQ := TRUE,
        ID := 296,                                   // 296: HW-ID for slot 48
        INDEX := 12,                                 // Index 12: Measurement Status
        MLEN := 2,                                   // Max. lenth of data to read
        VALID => #Valid,                             // New data received and valid
        BUSY => #Busy,                               // Read not completed yet
        ERROR => #Error,                             // Read error
        STATUS => #Status,                           // Read status
        LEN => #LenRead,                             // Number of bytes was read from device
        RECORD := #MeasStatus);                      // Measurement status
UNTIL NOT #Busy
END_REPEAT;

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

REPEAT
   "RDREC_DB"(REQ := TRUE,
        ID := 296,                                   // 296: HW-ID for slot 48
        INDEX := 15,                                 // Index 15: measured resistance value
        MLEN := 64,                                  // Max. lenth 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 were read
        RECORD := #MeasValueAsByteArray);            // Array[0..63] of Byte
UNTIL NOT #Busy
END_REPEAT;

Chars_TO_Strg(Chars := #MeasValueAsByteArray,        // Byte array
        pChars := 0,                                 // Start position
        Cnt := #BytesWritten,                        // Number of bytes copied
        Strg => "Data_block_1".MeasValueAsString);   // Destination string
```
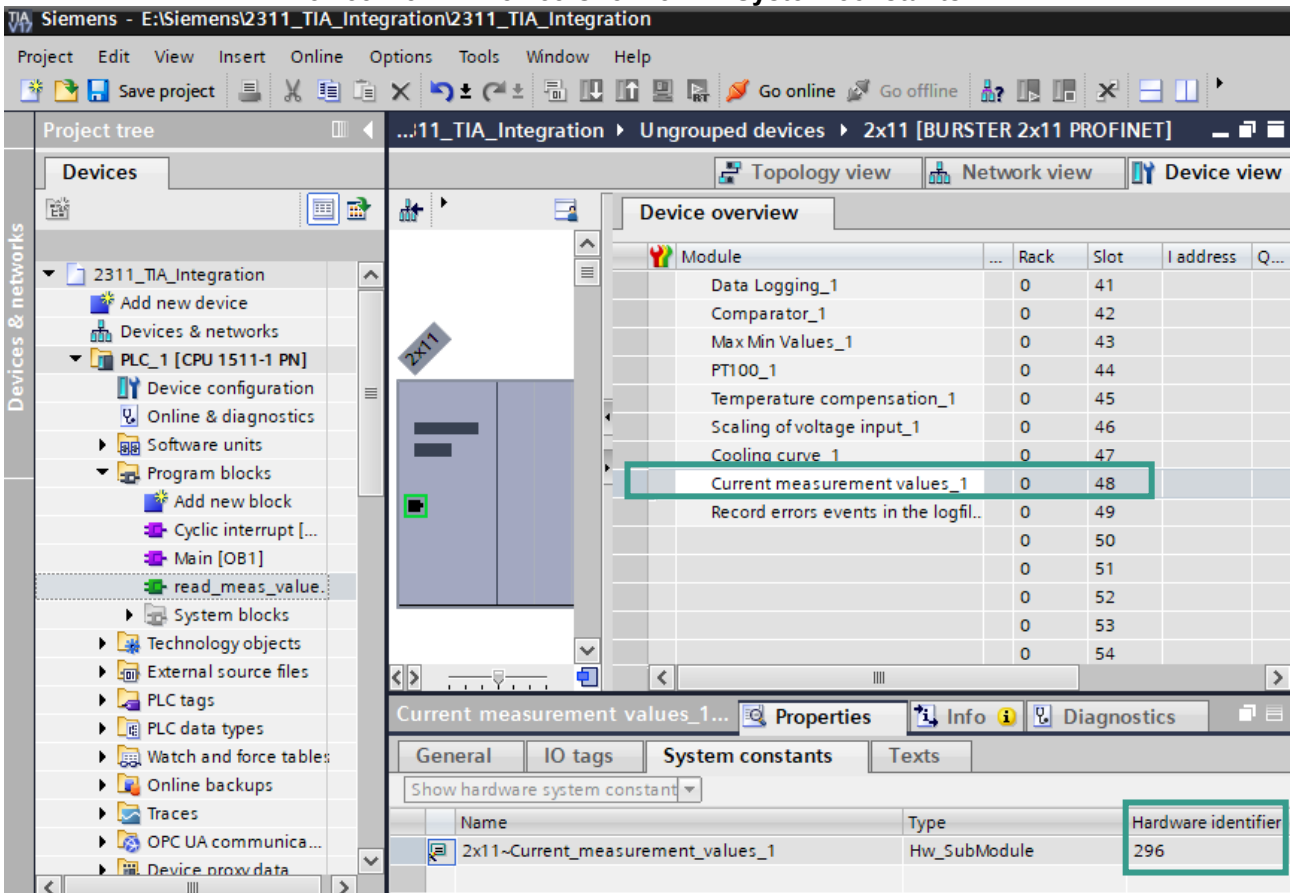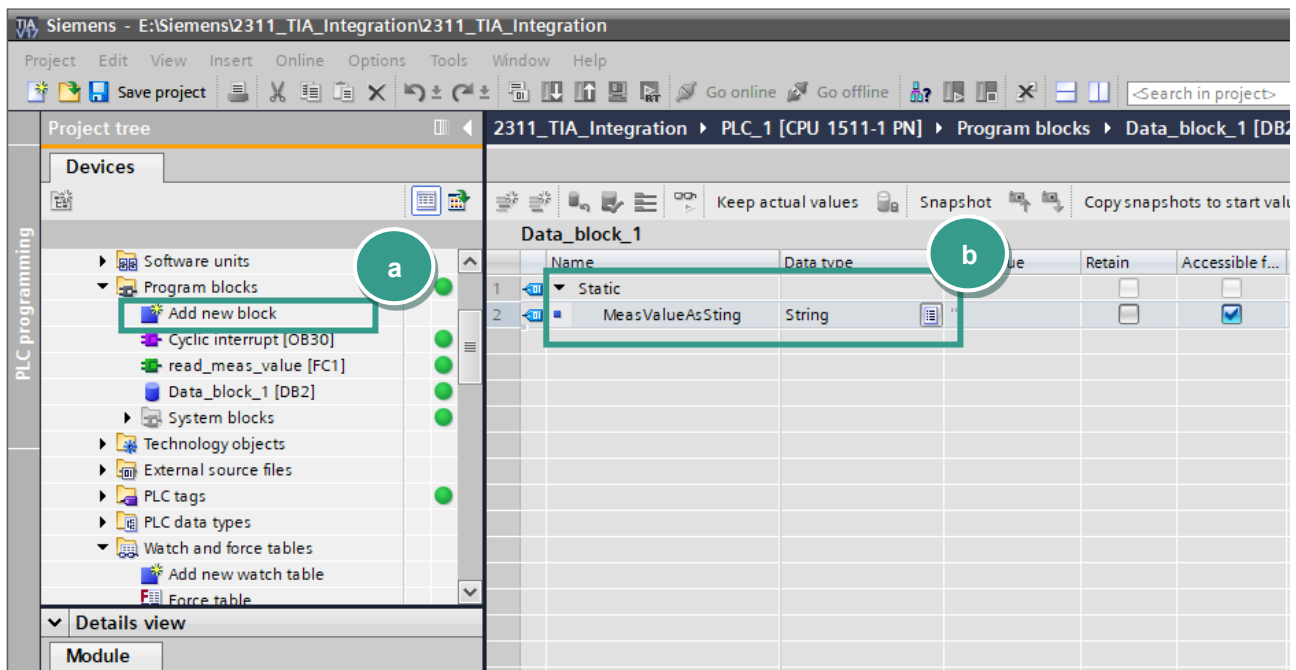
Note: Your Hardware identifier for the slot 48 (Current measurement values) may be different. You can check it in the **Device view → Device Overview → System constants**:

7. Add a new data block via **Add new block** (a) → **Data block** and insert the string variable '*MeasValueAsString*' into it (b):

8. Before you load the project into the CPU you have to set the IP addresse 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) assigned to your in section *IP-Protocol*:
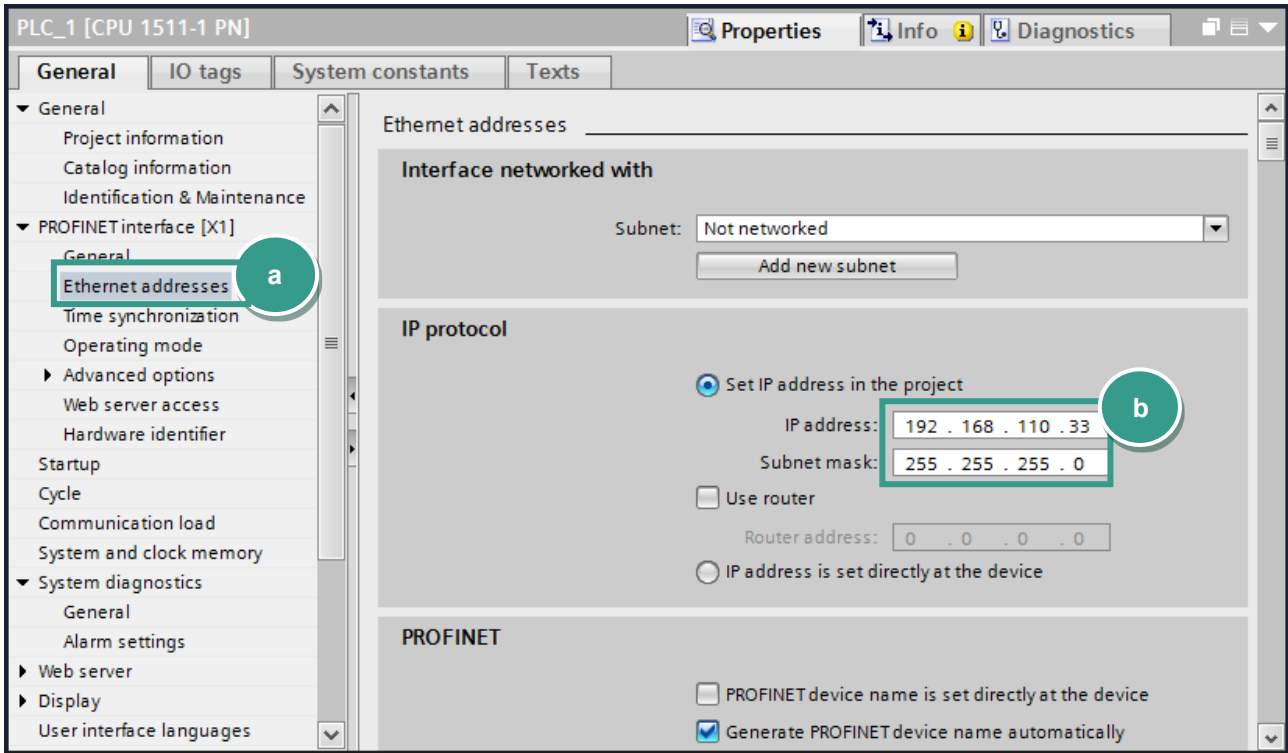
9. To load the configuration into the CPU select it first go to *Online → Download to device* and click on *Start search* (a) to look for your controller. Then select the controller and click on *Load* (b):

RESISTOMAT® 2311 PROFINET
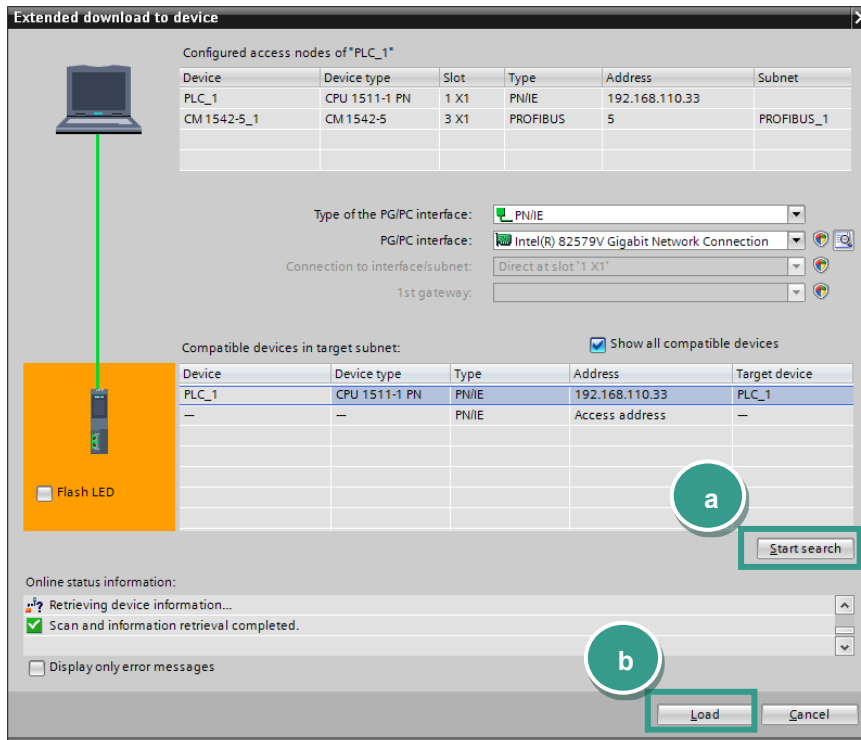
The PLC checks the 2311 device ready status and starts a new measurement. Then it stops the measurement after 3 seconds, read the measurement results, starts a new measurement again and so on.

Note: Make sure that PROFINET Control is enabled in RESISTOMAT® 2311. For details, see chapter *Configuration menu in Resistomat 2311* of *the* **RESISTOMAT® 2311 PROFINET** manual.

1. To control the measurement value just add the variable **"Data_block_1".MeasValueAsString** to the watch table and click on the button **Monitor all** (a)
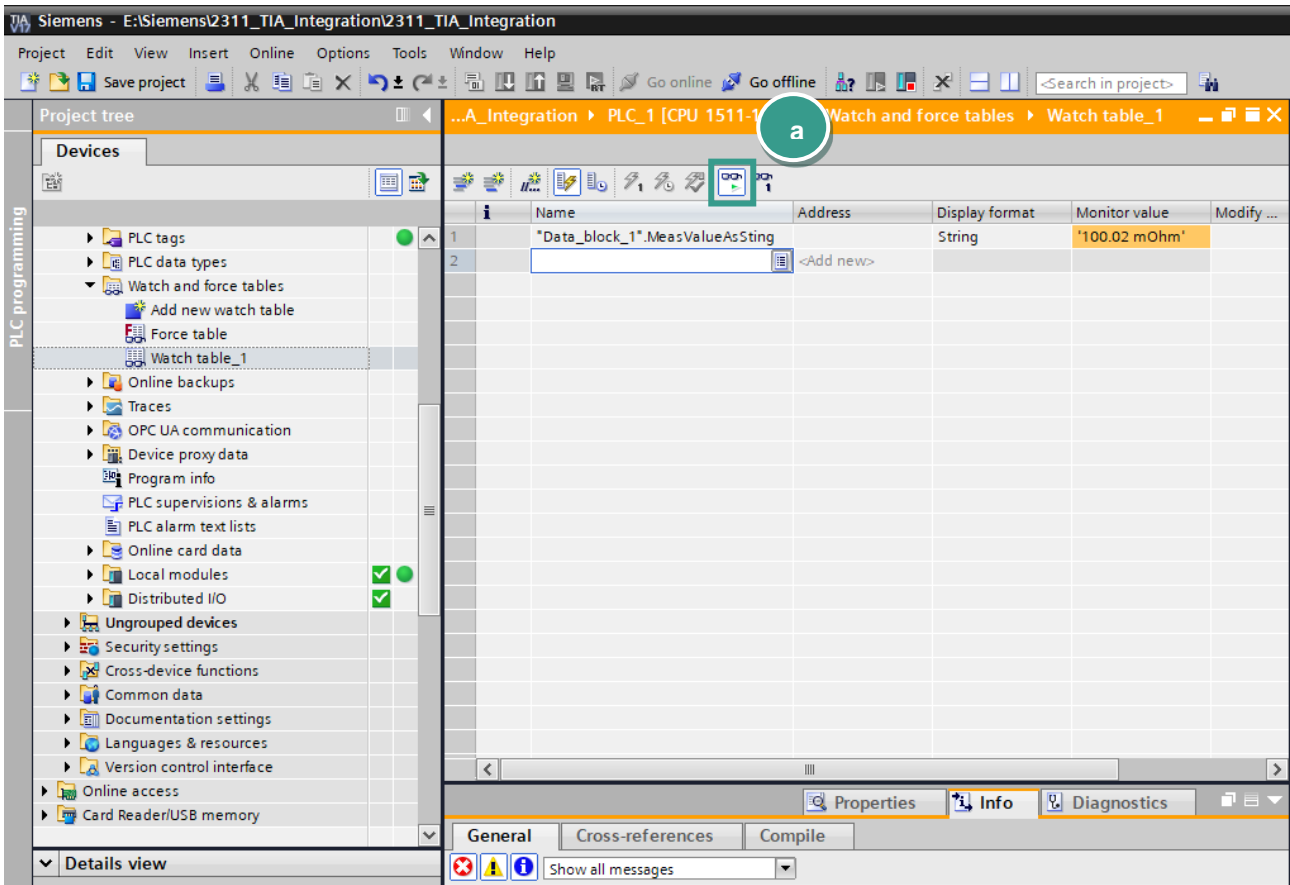
6 Further Examples

6.1 Writing of string data types

Note: Datatype **String** in TIA Portal contains two additional bytes, which represent the length of the string. To avoid these two bytes being sent use the function 'Strg_TO_Chars' to convert the String to a byte array as shown below:

Example 1: Reading Device ID and write it as station name to device

In this example, we perform a write access on slot 30/Subslot 1/index 16 to set the station name of Resistomat 2311. For this acyclic operation, you will need an instance of a WRREC block. You can see the new station name in the **info menu** of Resistomat 2311.

PLC parameters table:

		Name	Data type	Default value	Comment
4		▼ Temp			
5		NumBytesWritten	UInt		
6		▶ abStationName	Array[0..64] of Byte		
7		Busy	Bool		
8		Error	Bool		
9		Status	DWord		
10		Done	Bool		
11		▼ Constant			

Sourcecode:

```
Strg_TO_Chars(Strg := 'Station 3',          // Station name as a string
        pChars := 0,                        // Position in byte array
        Cnt => #NumBytesWritten,            // Number of bytes written
        Chars := #abStationName);           // Station name as byte array

REPEAT
    "WRREC_DB"(REQ := TRUE,
        ID := 268,                          // 268: HW-ID for General Settings
        INDEX := 16,                        // Index 16: Station Name
        LEN := #NumBytesWritten,            // Length of data to write
        DONE => #Done,                      // Write done
        BUSY => #Busy,                      // Write not completed yet
        ERROR => #Error,                    // Error
        STATUS => #Status,                  // Status
        RECORD := #abStationName);          // Bytes to write
UNTIL NOT #Busy AND #Done
END_REPEAT;
```

RESISTOMAT® 2311 PROFINET

Example 2: Writing of serial number 1 into device order sheet

PLC parameters table:

		Name	Data type	Default value	Comment
4		▼ Temp			
5		▪ NumBytesWritten	UInt		
6		▪ ▶ abSerial	Array[0..64] of Byte		
7		▪ Busy	Bool		
8		▪ Error	Bool		
9		▪ Status	DWord		
10		▪ Done	Bool		
11		▼ Constant			

Sourcecode:

```
Strg_TO_Chars(Strg := 'SN123456789',          // Serial as a string
        pChars := 0,                          // Position in byte array
        Cnt => #NumBytesWritten,              // Number of bytes written
        Chars := #abSerial);                  // Station name as byte array

REPEAT
   "WRREC_DB"(REQ := TRUE,
        ID := 268,                            // 268: HW-ID for General Settings
        INDEX := 76,                          // Index 76: Serial number 1
        LEN := #NumBytesWritten,              // Length of data to write
        DONE => #Done,                        // Write done
        BUSY => #Busy,                        // Write not completed yet
        ERROR => #Error,                      // Error
        STATUS => #Status,                    // Status
        RECORD := #abSerial);                 // Bytes to write
UNTIL NOT #Busy AND #Done
END_REPEAT;
```

You can see the written serial in the **Order sheet** menu of Resistomat 2311.