OPERATION MANUAL

DIGIFORCE® 9307
PROFINET Integration into TIA Portal

Manufacturer:
burster
präzisionsmesstechnik gmbh & co kg
Talstraße 1 - 5
D-76593 Gernsbach
Germany

Postfach 1432
D-76593 Gernsbach
Germany

Tel.: (+49) 07224 645-0
Fax.: (+49) 07224 645-88
E-Mail:info@burster.com
www.burster.com

Valid from: **01.08.2018**
Applies to: **DIGIFORCE® 9307-V0X03**

2766-BA9307PROFINTIAEN-5770-071525

Table of Contents

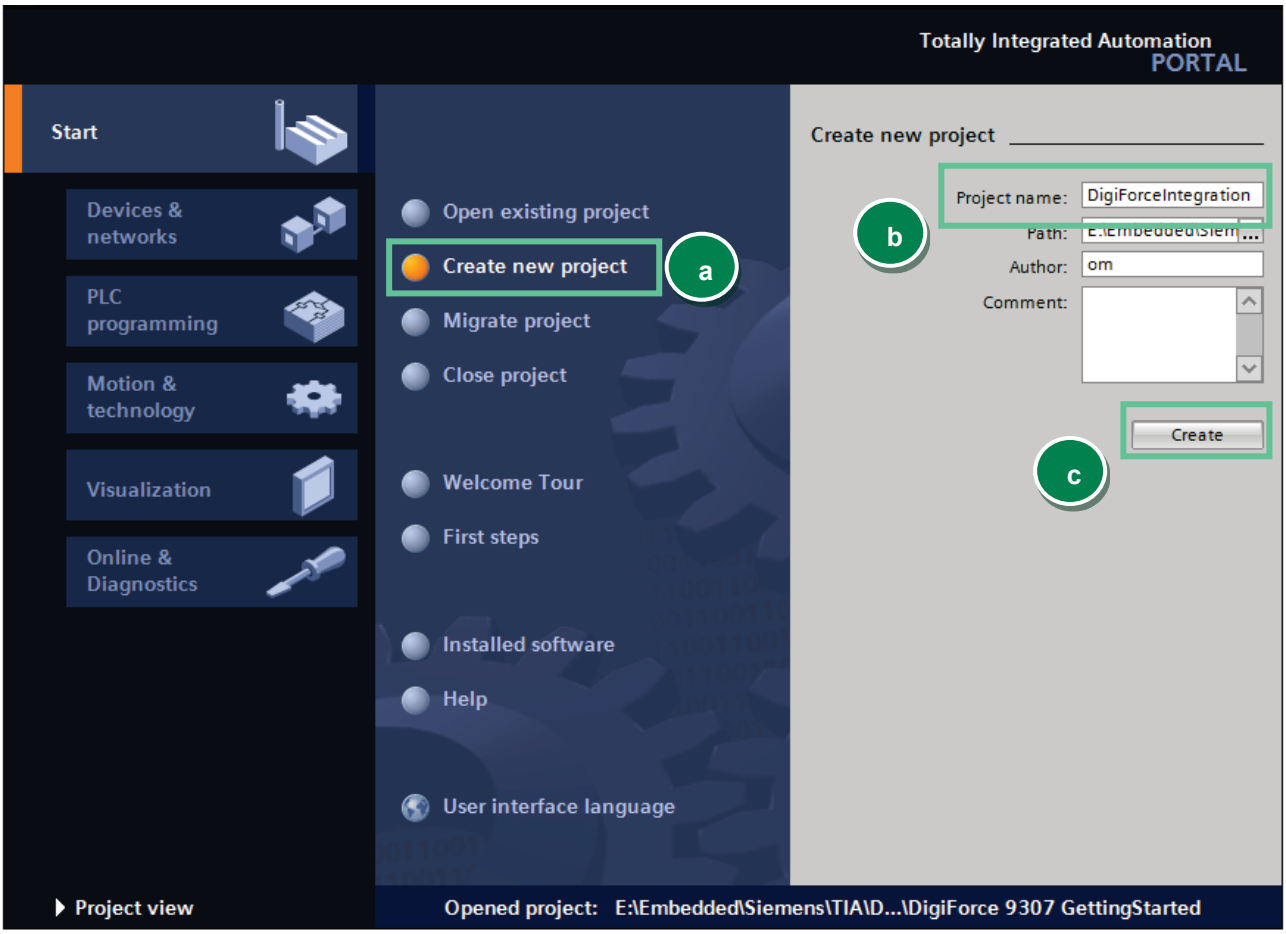
Introduction

This quick start guide describes an approach how you can configure the DIGIFORCE® 9307 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.

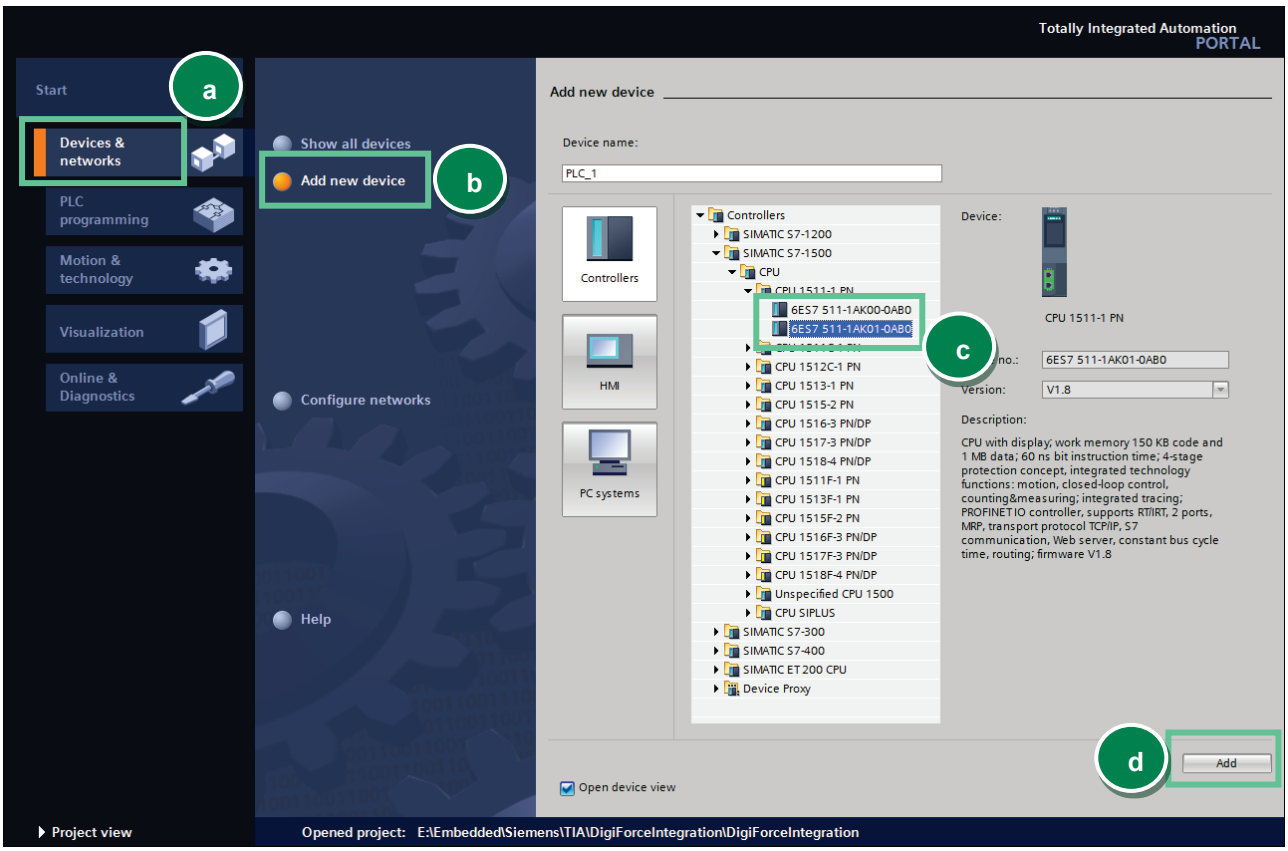
Please also note that you will have to use the DIGIFORCE® 9307 PROFINET manual to get futher information about input and output parameters (cyclic as well acyclic data transfer)

1. Creating new project

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



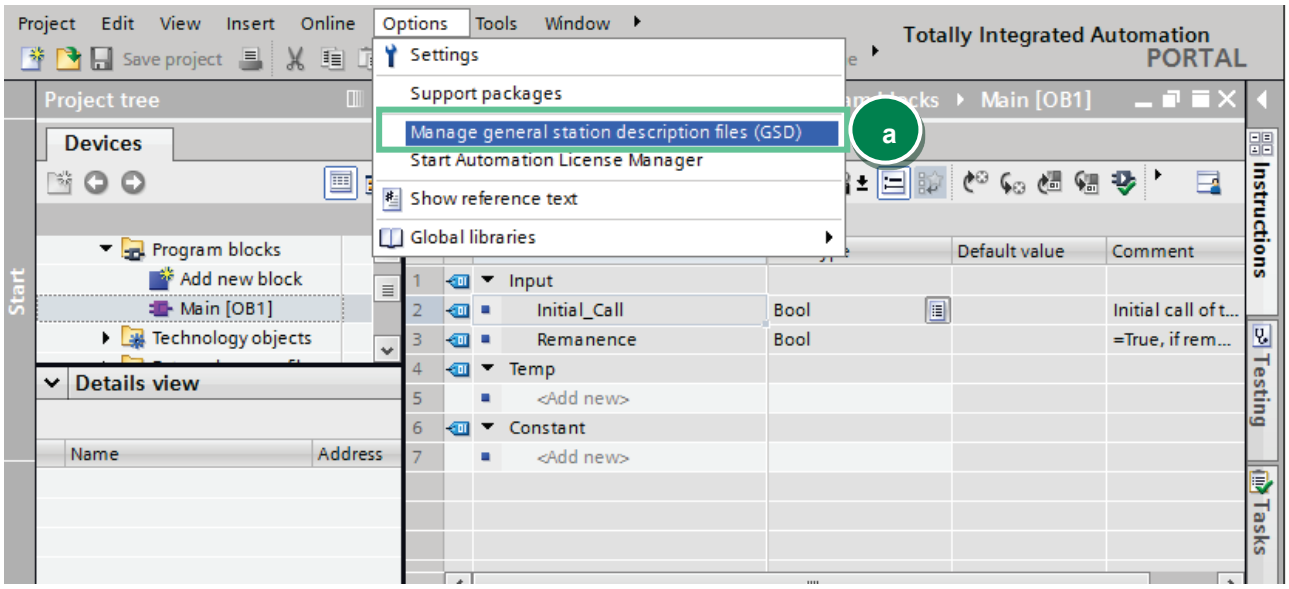
➢ Go to ***Devices & networks*** (a) on the left side select ***Add new device*** *(b)* and look for yor CPU (c). Afterwards click the **Add** button (d).



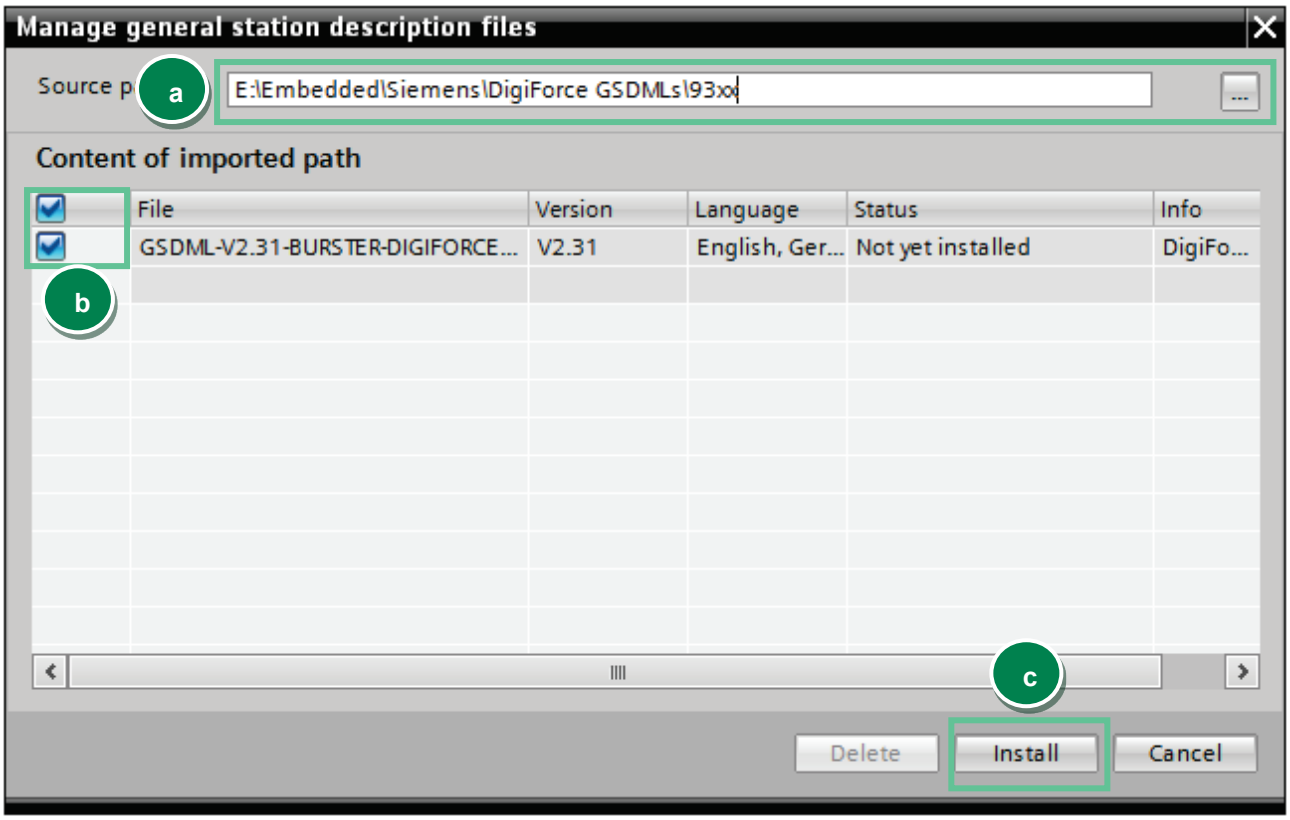
2. Installation of GSDML files

Note: Please make sure that your GSDML file is compatible to the field bus firmware in the DIGIFORCE® 9307. 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)***

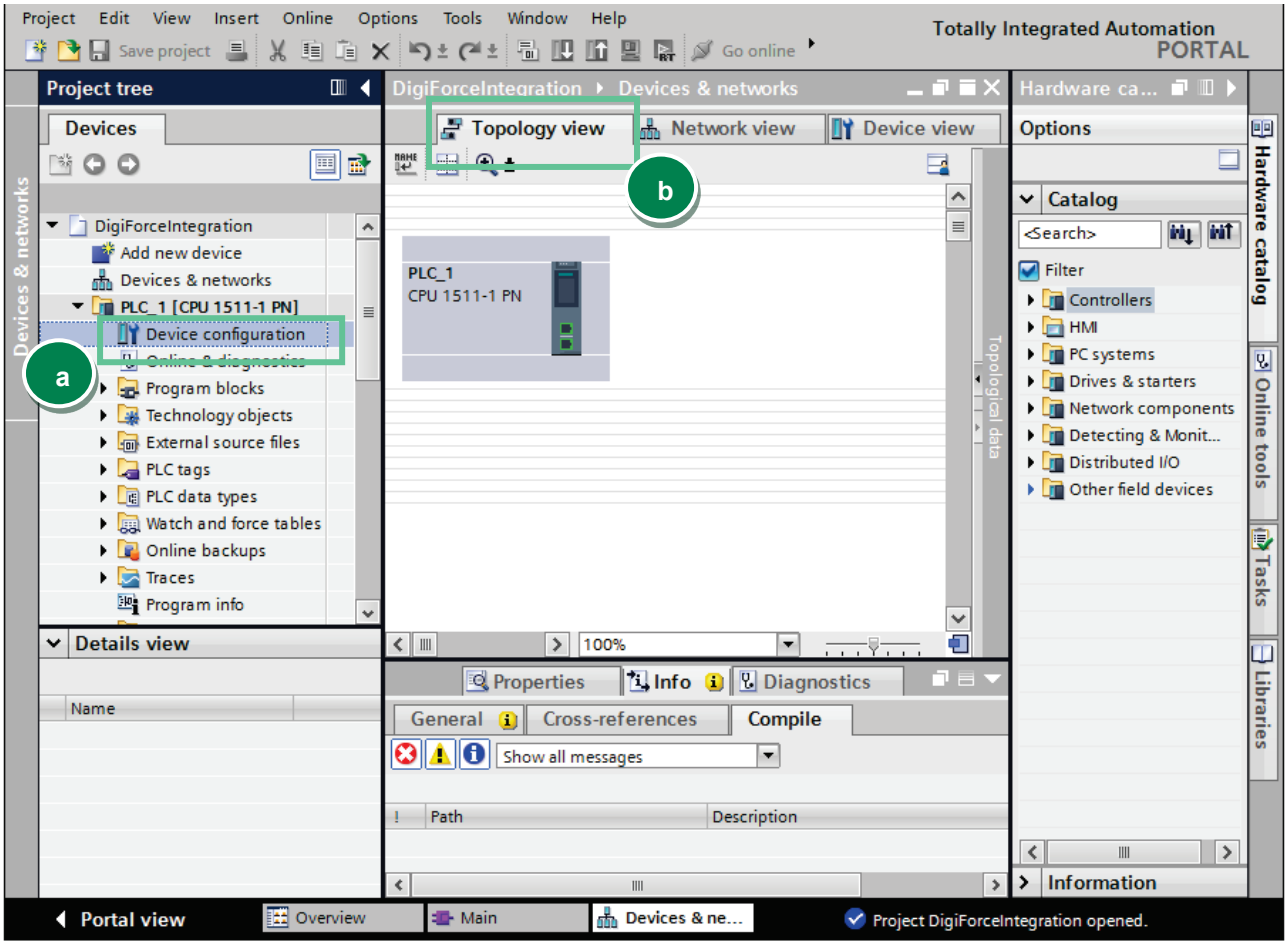


➢ Navigate to your DIGIFORCE® 9307 GSDML directory (a)(you will find the GSDML files on burster DVD that you got with your DIGIFORCE® 9307 device or on www.burster.com), select the GSDML file (b) and click ***Install*** (c)

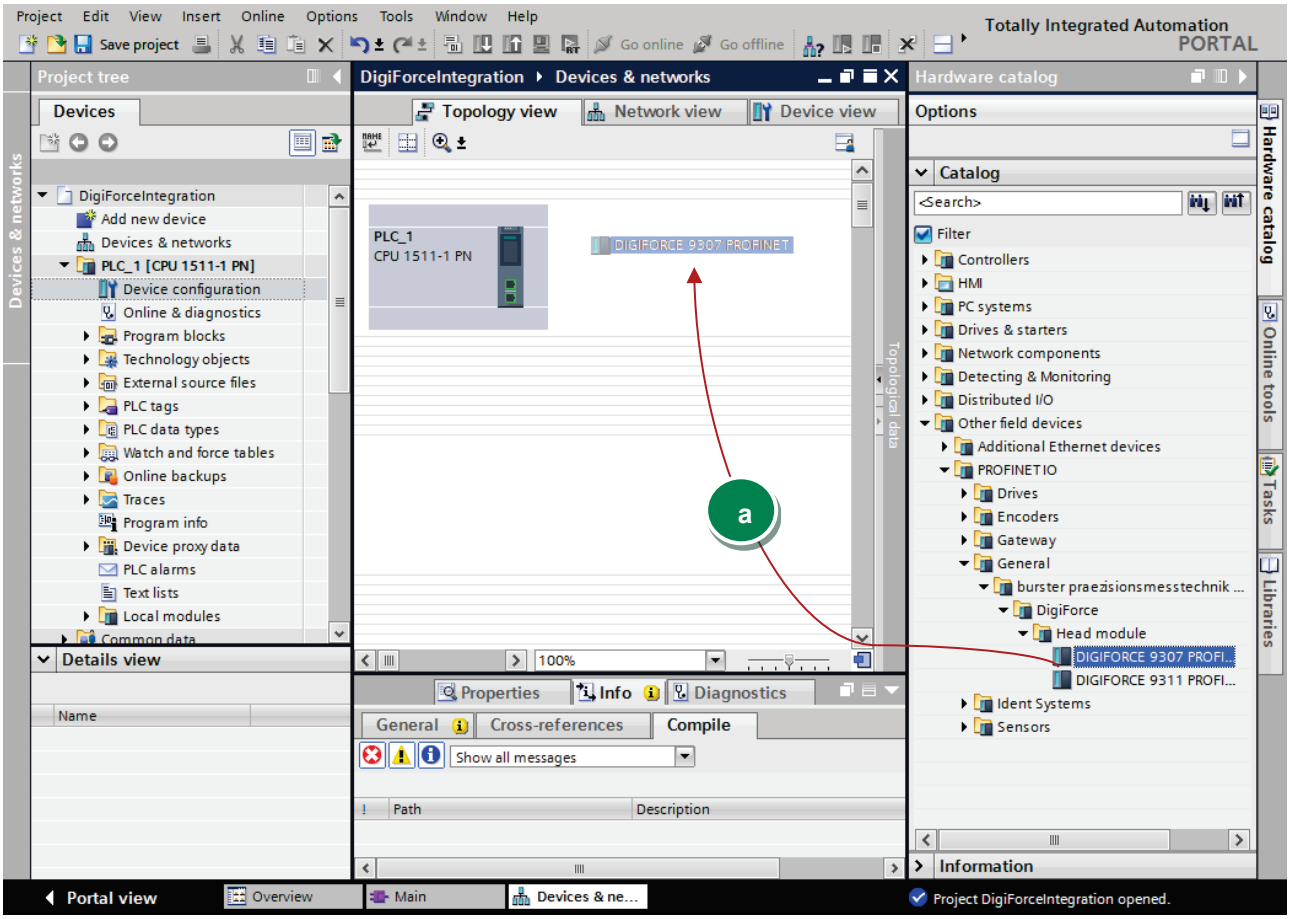


3. Creation of network connections

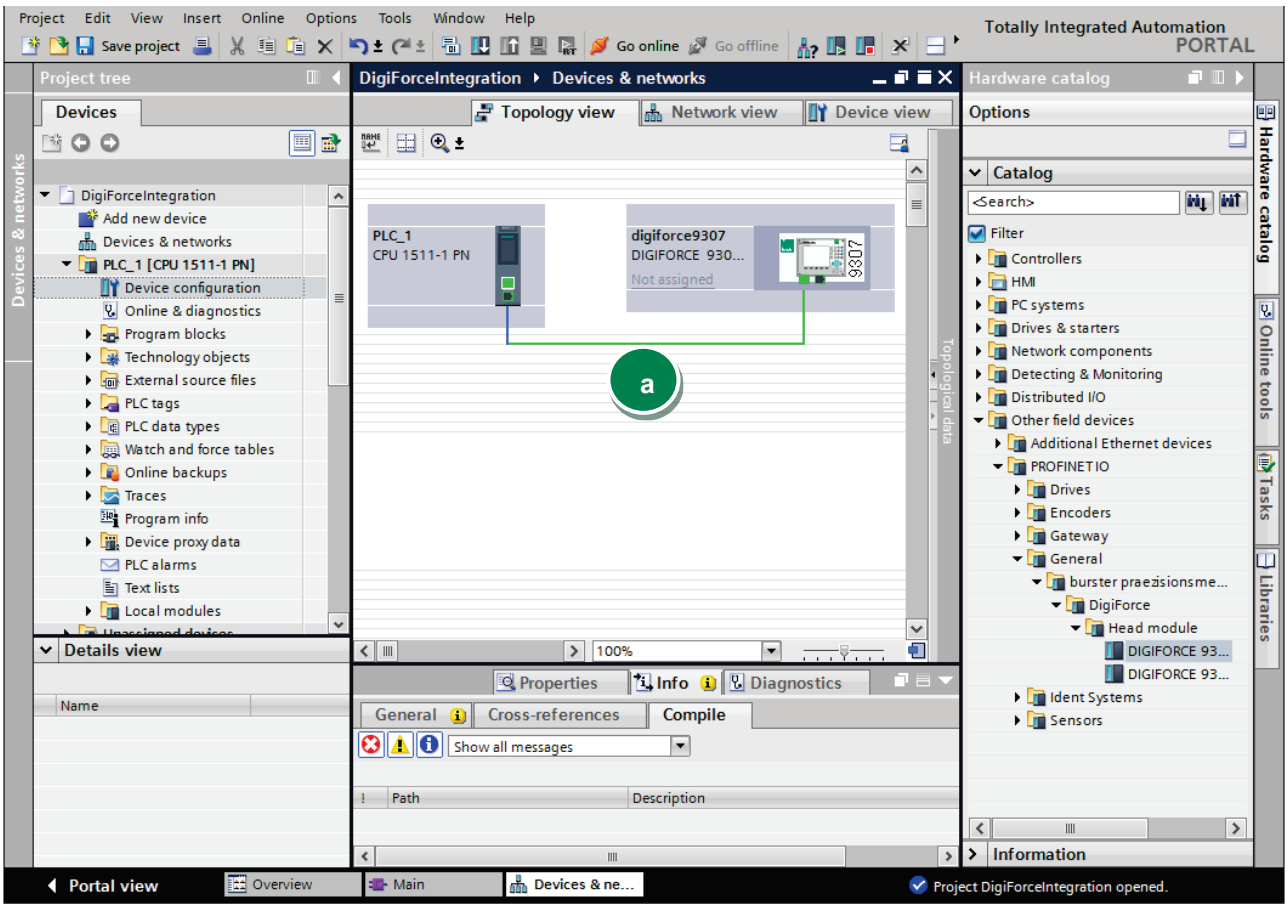
➢ Double click ***Device Configuration*** (a) in the project tree und switch to **Topology view** (b):



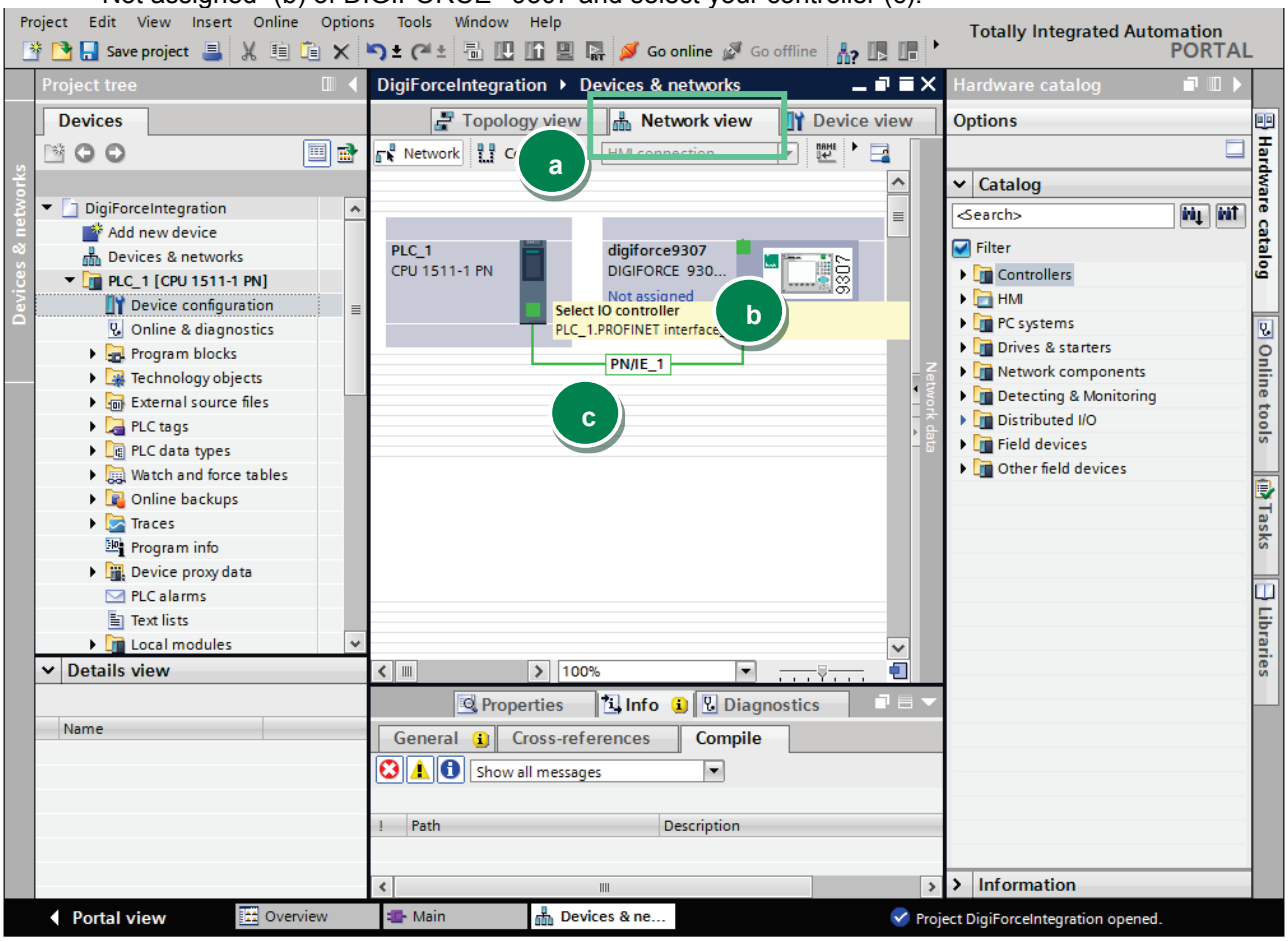
➢ Now select the DIGIFORCE® 9307 device in the catalog and drag & drop it into the working area (a):



➢ Please select an ethernet port on the S7 and hold the left mouse button down to connect the S7 with DIGIFORCE® 9307:



➢ Change now to **Network view** (a) to assign a controller to the DIGIFORCE® 9307. Click on the link "Not assigned" (b) of DIGIFORCE® 9307 and select your controller (c):

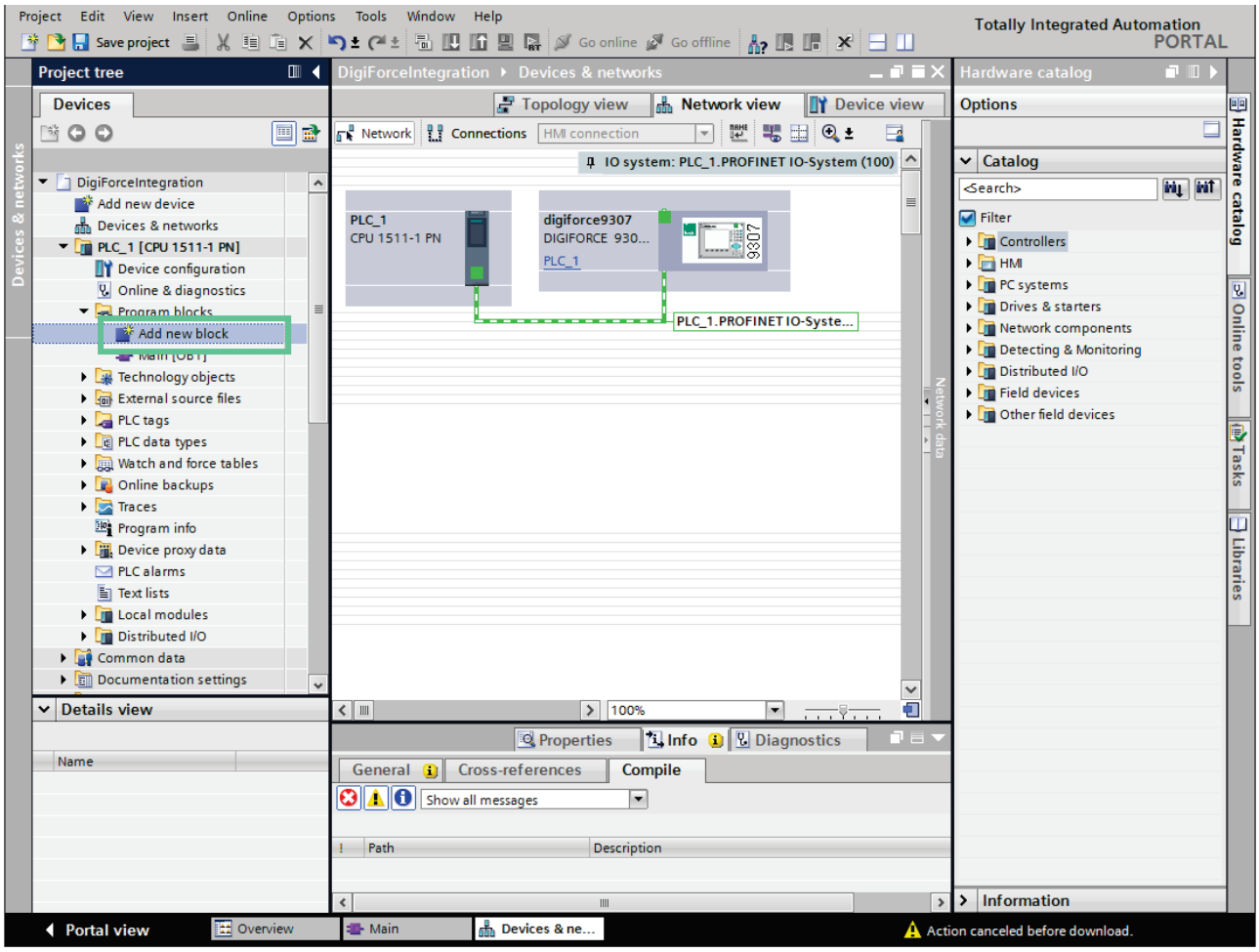


➢ Check if devices also connected physically to the right ports. You find the port number assignment in the section *5.3 Port-Identification* of **DIGIFORCE® 9307 PROFINET** manual.

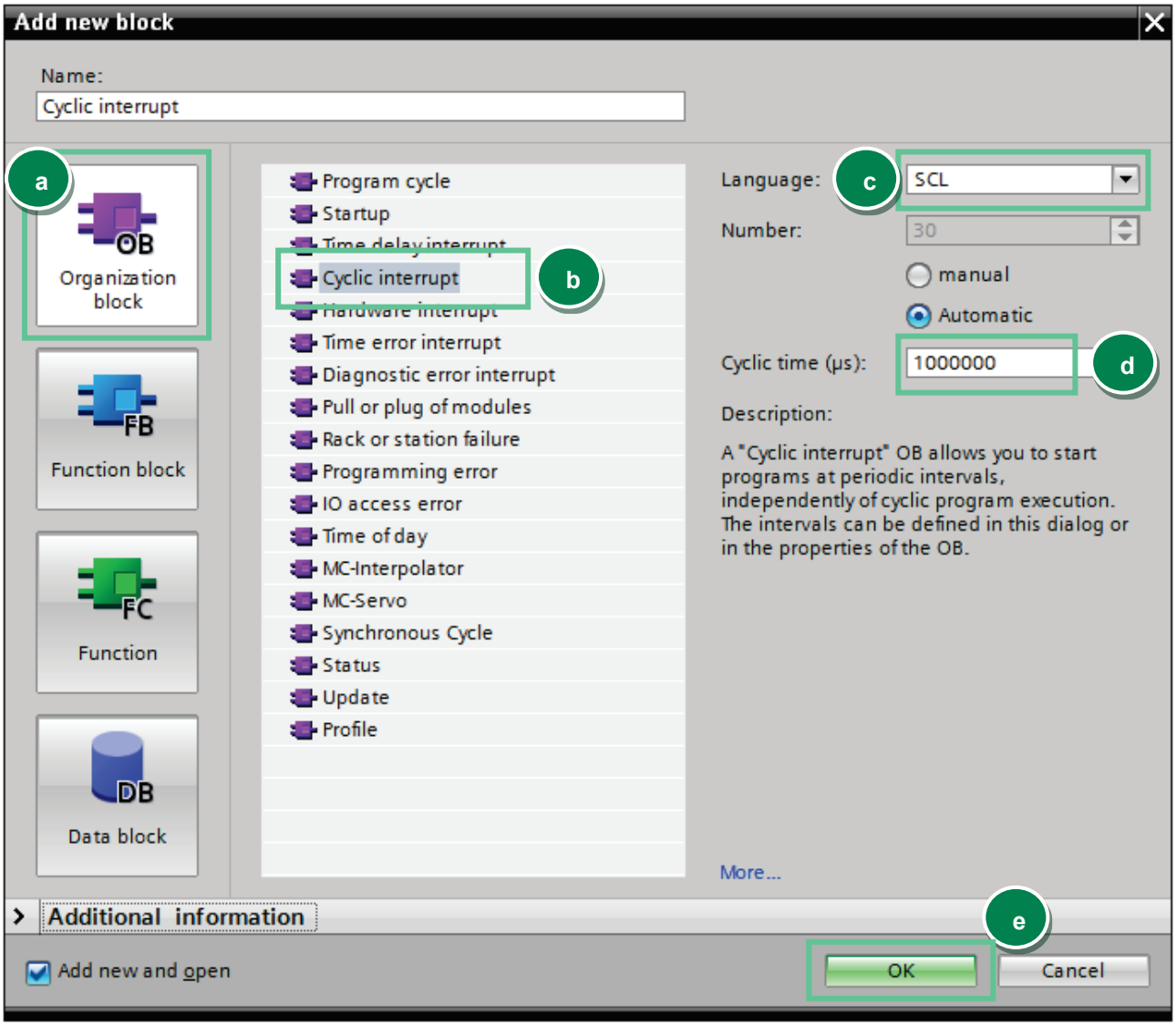
4. Create a sample program:

In this section, you will learn how to create a simple program to start and stop a measurement periodically. You will need to refer to sections *8.2 PLC inputs* and *8.3 PLC outputs* of **DIGIFORCE® 9307 PROFINET** manual to understand the meaning of inputs and outputs bytes.

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



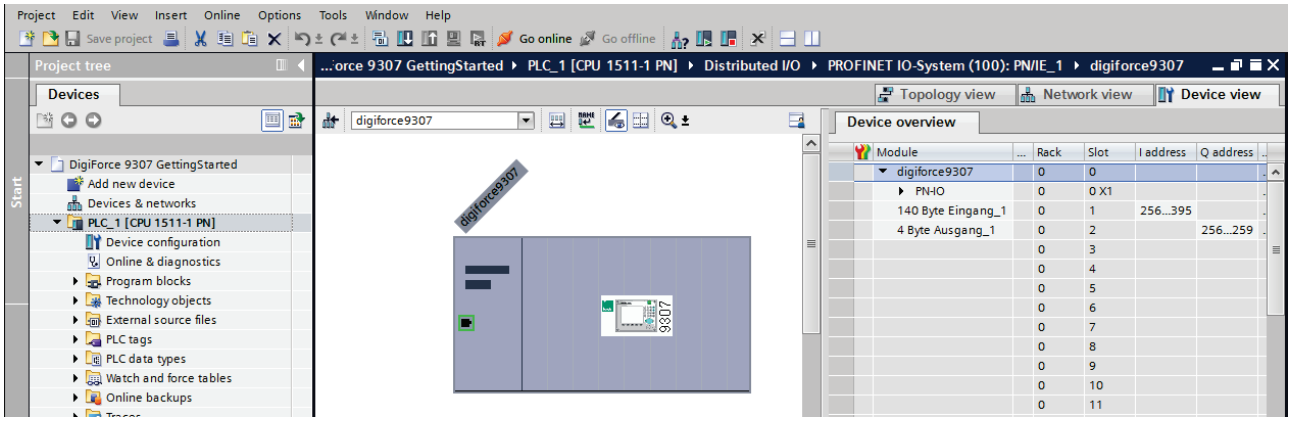
➢ Select in the new window ***Organization block*** (a) and then ***Cyclic interrupt*** (b)**.** As language set SCL (c), change the cyclic time to 1.000.000 µs (d) and click OK (e):



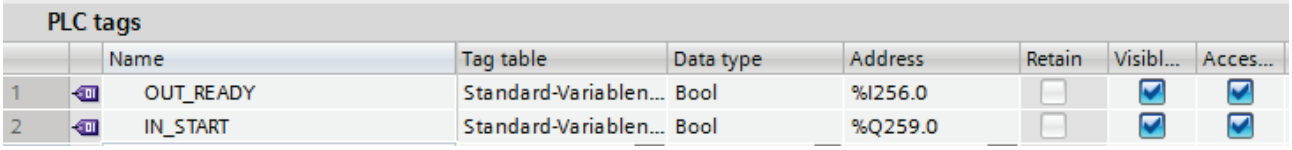
➢ Type in the following source code in the code fild of the new block:

```
IF %Q259.0 = TRUE THEN          // is IN_START (measurement start) set?
  %Q259.0 := FALSE;             // IN_START (measurement start) is set, then reset it
ELSE                            // IN_START is not set
  IF %I256.0 = FALSE THEN       // is OUT_READY  (DigiForce ready for measurement) set?
    RETURN;                     // If not -
  END_IF;                       // return
  %Q259.0 := TRUE;              // set IN_START(measurement start)
END_IF;
```

Please note: the addresses may be different. You have to check them in the **Device view->Device overview** of the DIGIFORCE® 9307.



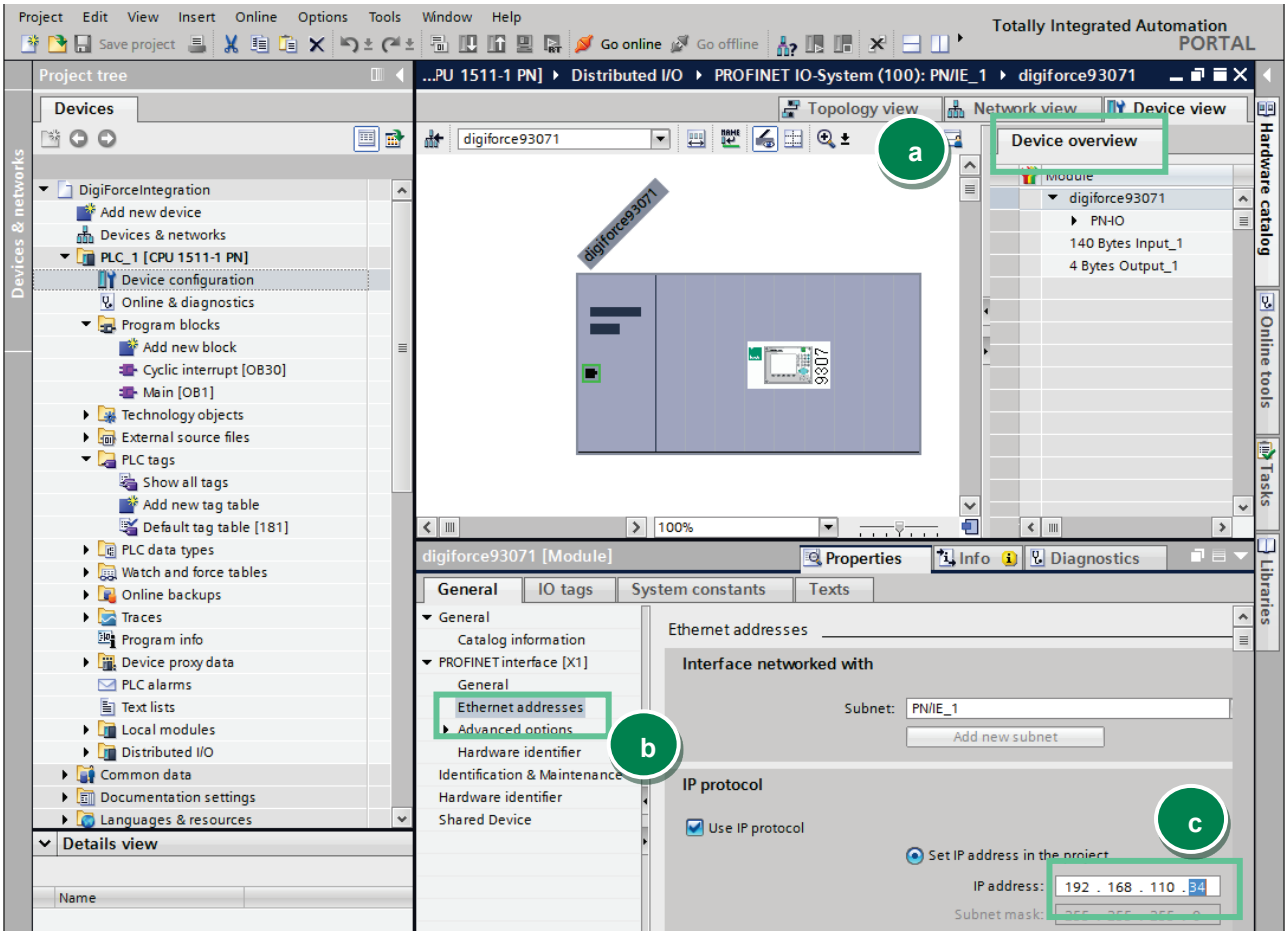
You will also see that the TIA-Editor replaces the input/output addresses with tags. You can change the tags names in PLC Tag table:



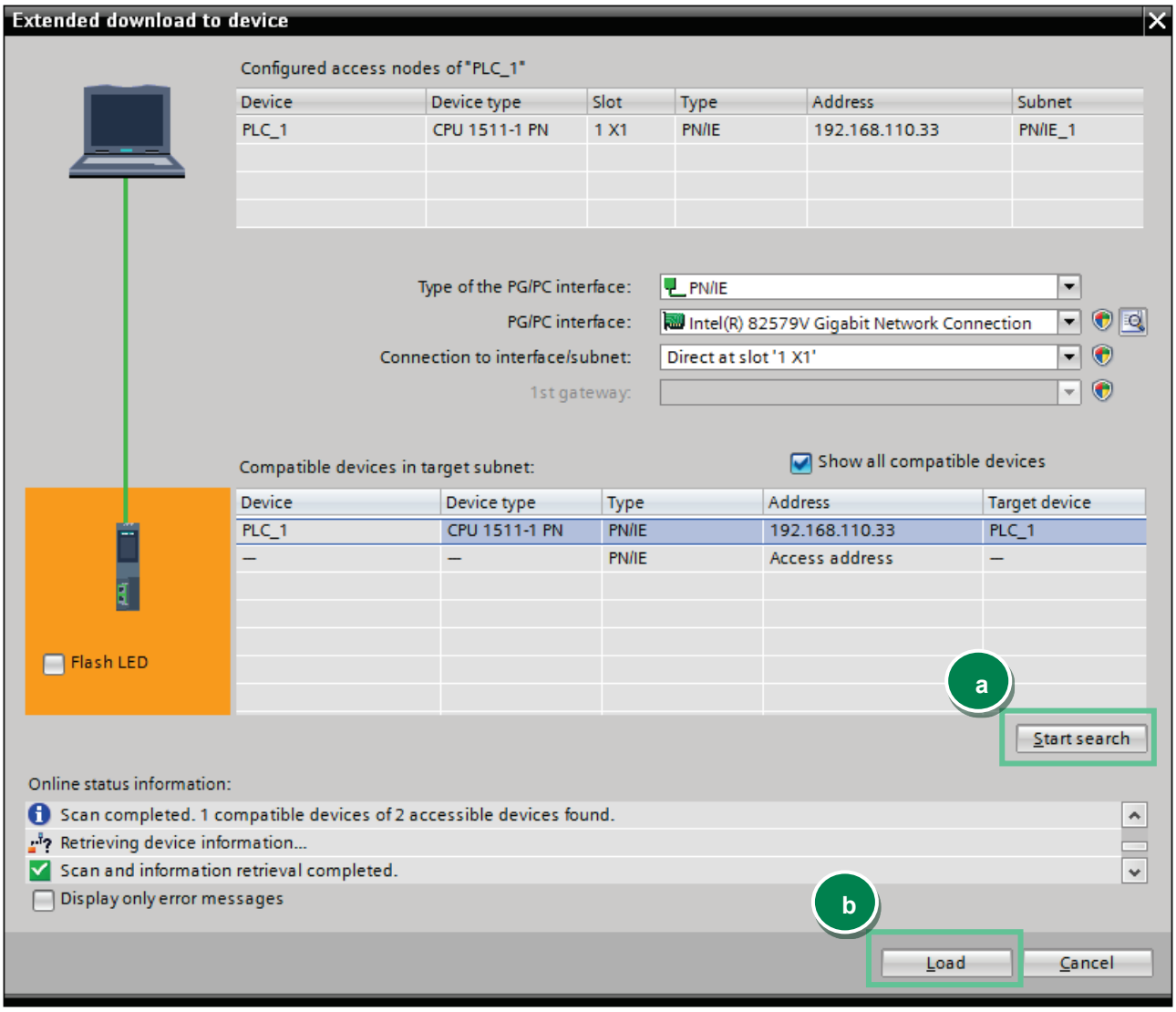
PLC tags

	Name	Tag table	Data type	Address	Retain	Visibl...	Acces...
1	OUT_READY	Standard-Variablen...	Bool	%I256.0		☑	☑
2	IN_START	Standard-Variablen...	Bool	%Q259.0		☑	☑

➢ Before you load the project into the CPU you have to set the IP addresses of your controller and your device. To do this please go to **Device view** (a) and select ***Ethernet addresses*** (b) in **General** tab. Set now a desired IP-Address (c):



➢ To load the configuration into the CPU select it first and then 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):

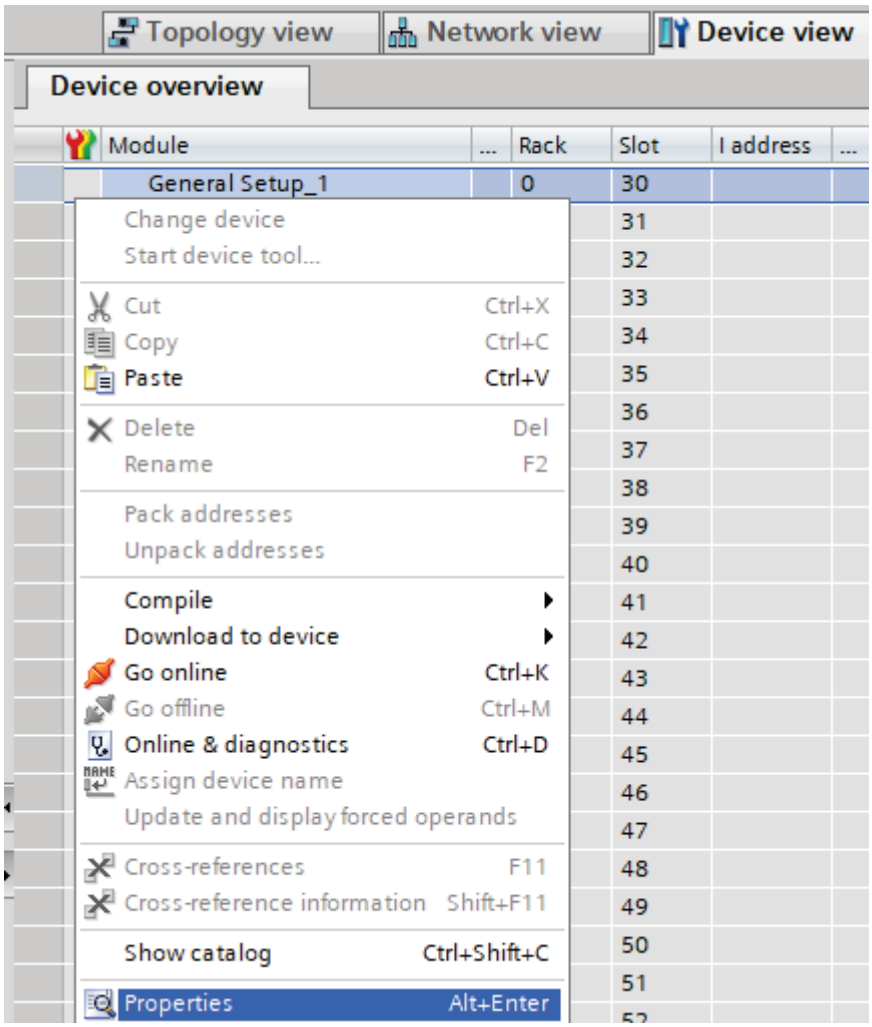


The DIGIFORCE® 9307 starts now a new measurement waits a second, and stops the measurement, waits a second and strarts the measurement again and so on.

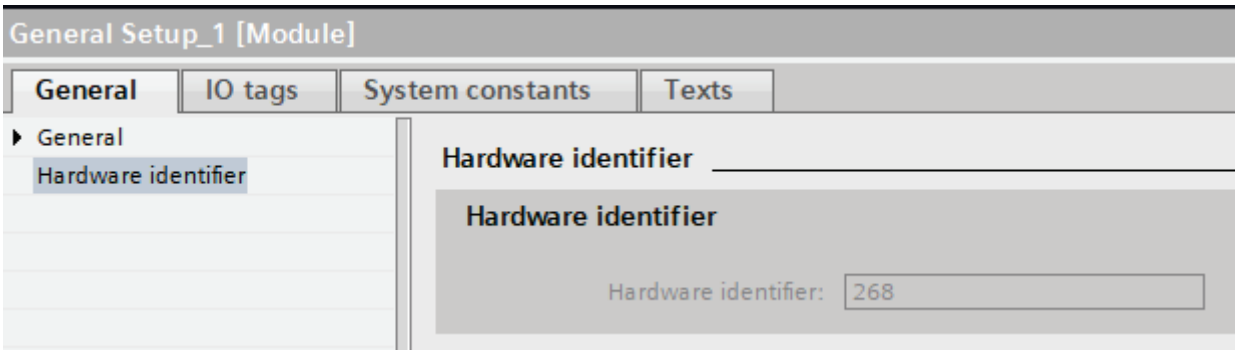
Note: Make sure that PROFINET Control is enabled in DIGIFORCE® 9307. For details, see chapter *5.5 Configuration menu in DIGIFORCE® 9307* of **DIGIFORCE® 9307 PROFINET** manual.

5. Further Examples

In the followed examples, a *Hardware-ID* is used to access a certain slot. To find this, please select a DIGIFORCE® 9307 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. *General Setup* and select ***Properties***:



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



5.1 Reading and Writing of string data types

In this example, we perform a read access on slot 30/Subslot 1/index 10 to get the device type of DIGIFORCE® 9307 and then we will set the first nine characrters of this string as DIGIFORCE® 9307 station name on Slot 30/Subslot 1/Index 19. For these acyclic operations, you will need an instance of RDREC und WRREC blocks. You can see the new station name in the **info menu** of DIGIFORCE® 9307.

PLC parameters table:

5		Valid	Standard-Variablen...	Bool
6		Busy	Standard-Variablen...	Bool
7		Error	Standard-Variablen...	Bool
8		Status	Standard-Variablen...	DWord
9		Done	Standard-Variablen...	Bool
10		lenRead	Standard-Variablen...	UInt

Sourcecode:

```
REPEAT
"RDREC_DB"(REQ:=TRUE,
        ID := 268,              // 268: HW-ID for General Setup (see introduction of 'Further examples')
        INDEX:=10,              // Index 10: Device Detection
        MLEN:=18,               // 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 was read from device
        RECORD:= #data);        // Array[0..18] of Byte
UNTIL NOT #Busy
END_REPEAT;

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

REPEAT
"WRREC_DB"(REQ:=TRUE,
        ID := 268,              // HW-ID for General Setup (see introduction of 'Further examples')
        INDEX:=19,              // Index 19: Station Name
        LEN:=9,                 // Length of data to write
        DONE=>#Done,            // Write done
        BUSY=>#Busy,            // Write not completed yet
        ERROR=>#Error,          // Error
        STATUS=>#Status,        // State
        RECORD:=#data);   // Write the data has being read in RDREC_DB (first 9 bytes)
UNTIL NOT #Busy AND #Done
END_REPEAT;
```

Example 2: Writing of serial number SN1 into device order sheet

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:

PLC parameters table:

Name	Data type	Default value
serial	String	
bytesWritten	UInt	
serialAsByteArray	Array[0..64] of Byte	
Busy	Bool	
Error	Bool	
Status	DWord	
Done	Bool	

Sourcecode:

```
#serial := 'SN123456789';

Strg_TO_Chars(Strg:= #serial,                    // Serial as String
          pChars:= 0,                            // Position in serialAsByteArray
          Cnt => #bytesWritten,                  // Number of Bytes have been written to serialAsByteArray
          Chars:= #serialAsByteArray);

REPEAT
  "WRREC_DB"(REQ := TRUE,
          ID := 268,                             // HW-ID General Setup (see introduction of 'Further examples')
          INDEX := 86,                           // Index 86: Order sheet - Serial number 1
          LEN := INT_TO_UINT(LEN(#serial)),      // Length of serial
          DONE => #Done,                         // Write done
          BUSY => #Busy,                         // Write not completed yet
          ERROR => #Error,                       // Error
          STATUS => #Status,                     // State
          RECORD := #serialAsByteArray);
UNTIL NOT #Busy AND #Done
END_REPEAT;
```

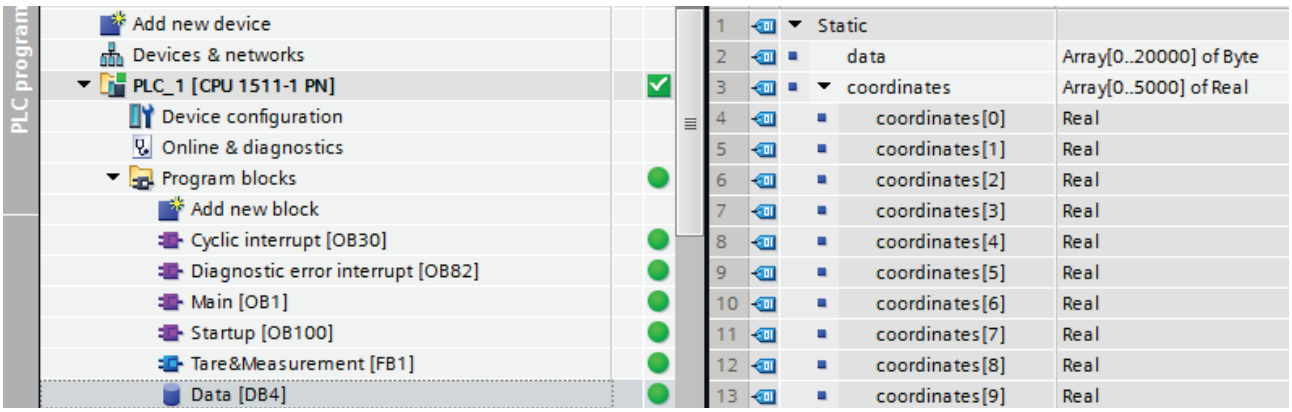
5.2 Retrieving of measurement results

Note: The reading of all X- or Y-Coordinates of a curve an once is only supported in the DIGIFORCE® 9307 PROFINET Firmware V16.0.1 and higher.

This example shows you how to read the X-Coordinates of the current curve:

PLC parameters tables:

4	Temp		
5	Valid	Bool	
6	Done	Bool	
7	Busy	Bool	
8	Error	Bool	
9	Status	DWord	
10	i	Int	
11	lastIndex	DWord	
12	lenRead	UInt	
13	measVal	DWord	
14	tmp	DWord	



Sourcecode:

```
REPEAT
    "WRREC_DB"(REQ := TRUE,                    // Write access to index 10 to prepare the curve
              ID := 454,                       // HW-ID (see introduction of 'Further examples')
              INDEX := 10,                     // Index
              LEN := 2,                        // Length in bytes to write
              DONE => #Done,
              BUSY => #Busy,
              ERROR => #Error,
              STATUS => #Status,
              RECORD := "Data".data);          // Any 2 bytes to prepare the curve
UNTIL NOT #Busy AND #Done
END_REPEAT;

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

REPEAT
```

```
    "RDREC_DB"(REQ := TRUE,                          // Read the number of curve values
            ID := 454,                               // HW-ID (see introduction of 'Further examples')
            INDEX := 10,                             // Index
            MLEN := 4,                               // Max. length to read
            VALID => #Valid,
            BUSY => #Busy,
            ERROR => #Error,
            STATUS => #Status,
            LEN => #lenRead,                         // Number of bytes read
            RECORD := #lastIndex);                   // Number of values in the curve - 1
UNTIL NOT #Busy
END_REPEAT;

#lastIndex := SHR(IN := #lastIndex, N := 16);        // upto and including DIGIFORCE® 9307 field bus
                                                     firmware FW-2018.1.0 we have to use DWord to get
                                                     U16 Types from DIGIFORCE® 9307 und shift left
                                                     the result by 2 bytes
IF #Error = TRUE OR #Status <> 0 OR #lenRead <> 2 OR
#lastIndex = 0 THEN                                  // If read failed -> return
    RETURN;
END_IF;
REPEAT
    "RDREC_DB"(REQ := TRUE,
            ID := 454,                               // Read access to read out curve coordinates
            INDEX := 11,                             // HW-ID(see introduction of 'Further examples')
            MLEN := 20000,                           // Index
            VALID => #Valid,                         // Max. length to read
            BUSY => #Busy,
            ERROR => #Error,
            STATUS => #Status,
            LEN => #lenRead,
            RECORD := "Data".data);                  // Number of bytes read
UNTIL NOT #Busy                                      // Array to store the read coordinates
END_REPEAT;

IF #Error = TRUE OR #Status <> 0 OR #lenRead < 4
THEN                                                 // If read failed -> return
    RETURN;
END_IF;

FOR #i := 0 TO DWORD_TO_INT(#lenRead - 1) BY 4 DO
    #measVal := 0;                                   // Write bytes to DWORD and convert to Real
    #tmp := BYTE_TO_DWORD("Data".data[#i]);
    #measVal := #measVal + SHL(IN := #tmp, N := 24);
    #tmp := BYTE_TO_DWORD("Data".data[#i + 1]);      // Shift left the value by 24 bit
    #measVal := #measVal + SHL(IN := #tmp, N := 16);
    #tmp := "Data".data[#i + 2];                     // Shift left the value by 16 bit
    #measVal := #measVal + SHL(IN := #tmp, N := 8);
    #measVal := #measVal + "Data".data[#i + 3];      // Shift left the value by 8 bit
    "Data".coordinates[#i / 4] :=
DWORD_TO_REAL(#measVal);                             // Convert to Real and store in MeasValues[] Array
END_FOR;
```

5.3 Changing of window limits

This example shows you how to enable Evaluation Window 1 and set its coordinates.

Note: You have to write all four window limits and then confirm them with index 15. It is not possible to change only one single limit, e.g. xMax.

PLC parameters table:

Name	Data type	Default value
Temp		
Busy	Bool	
Error	Bool	
Status	DWord	
onOff	UInt	
xMin	Real	
xMax	Real	
yMin	Real	
yMax	Real	
event	Byte	
Done	Bool	
Constant		

Sourcecode:

```
#onOff := 1;                    // Activate Window 1
#event := 1;                    // Acknowledgement for indices 11, 12, 13,14

#xMin := 1.5;                   // Xmin coorrdinate of window 1
#xMax := 3.0;                   // Xmax coorrdinate of window 1
#yMin := 2.5;                   // Ymin coorrdinate of window 1
#yMax := 4.0;                   // Ymax coorrdinate of window 1

REPEAT
  "WRREC_DB"(REQ := TRUE,
        ID := 298,              // HW-ID for Evaluation Window 1 (see introduction of 'Further examples')
        INDEX := 10,            // Index 10: switch on window 1
        LEN := 2,               // Length of UINT16
        DONE => #Done,          // Write done
        BUSY => #Busy,          // Write not completed yet
        ERROR => #Error,        // Error
        STATUS => #Status,      // State
        RECORD := #onOff);
UNTIL NOT #Busy AND #Done
END_REPEAT;

REPEAT
  "WRREC_DB"(REQ := TRUE,
        ID := 298,              // HW-ID for Evaluation Window 1 (see introduction of 'Further examples')
        INDEX := 11,            // Index 11: Window 1 limit Xmin
        LEN := 4,               // Length of UINT16
        DONE => #Done,          // Write done
        BUSY => #Busy,          // Write not completed yet
        ERROR => #Error,        // Error
        STATUS => #Status,      // State
```

```
            RECORD := #xMin);
UNTIL NOT #Busy AND #Done
END_REPEAT;

REPEAT
   "WRREC_DB"(REQ := TRUE,
            ID := 298,                  // HW-ID for Evaluation Window 1 (see introduction of 'Further examples')
            INDEX := 12,                // Index 12: Window 1 limit Xmax
            LEN := 4,                   // Length of Real
            DONE => #Done,              // Write done
            BUSY => #Busy,              // Write not completed yet
            ERROR => #Error,            // Error
            STATUS => #Status,          // State
            RECORD := #xMax);
UNTIL NOT #Busy AND #Done
END_REPEAT;

REPEAT
   "WRREC_DB"(REQ := TRUE,
            ID := 298,                  // HW-ID for Evaluation Window 1 (see introduction of 'Further examples')
            INDEX := 13,                // Index 13: Window 1 limit Ymin
            LEN := 4,                   // Length of Real
            DONE => #Done,              // Write done
            BUSY => #Busy,              // Write not completed yet
            ERROR => #Error,            // Error
            STATUS => #Status,          // State
            RECORD := #yMin) ;
UNTIL NOT #Busy AND #Done
END_REPEAT;

REPEAT
   "WRREC_DB"(REQ := TRUE,
            ID := 298,                  // HW-ID for Evaluation Window 1 (see introduction of 'Further examples')
            INDEX := 14,                // Index 14: Window 1 limit Ymax
            LEN := 4,                   // Length of Real
            DONE => #Done,              // Write done
            BUSY => #Busy,              // Write not completed yet
            ERROR => #Error,            // Error
            STATUS => #Status,          // State
            RECORD := #yMax);
UNTIL NOT #Busy AND #Done
END_REPEAT;

REPEAT
   "WRREC_DB"(REQ := TRUE,
            ID := 298,                  // HW-ID for Evaluation Window 1 (see introduction of 'Further examples')
            INDEX := 15,                // Index 15: adopt values entered into indices 11, 12, 13,14
            LEN := 1,                   // Length of Real
            DONE => #Done,              // Write done
            BUSY => #Busy,              // Write not completed yet
            ERROR => #Error,            // Error
            STATUS => #Status,          // State
            RECORD := #event);
UNTIL NOT #Busy AND #Done
END_REPEAT;
```