Application Note

FESTO

SBRD Vision System integration with S7-1500 PLC USING TIA PORTAL 15

This Application Note helps the user to integrate SBRD Vision System with Siemens S7-1500 PLC using TIA Portal 15. Also this application note briefly describes the Function Blocks which are used for controlling the functionality of SBRD Vision System.

Type(s)

12356

Title ..... SBRD Vision System Integration with S7-1500 PLC Using TIA Portal 15
Version 1.00
Document no. 12356
Original ..... en
Author ..... Festo

Last saved 29.06.2020

Copyright Notice

Legal Notice

Internet: http://www.festo.com
E-Mail: service_international@festo.com

Table of contents

1 Components/Software used

Type/Name	Version Software/Firmware
SBRD controller pn 8067301	Fw version 1.1.04
GSDML File for SBRD Vision System	GSDML-V2.33-Festo-SBRD-Q-20181126
Siemens TIA Portal	V 15.1
Camera Configuration Studio	V 1.1.19323
Festo Field Device Tool	V 2.9.9 or higher

Table 1.1: Components/Software used

2 APPLICATION DESCRIPTION

This document explain the integration of the host function blocks of SBRD Vision System into SIEMENS TIA Portal V15.

The supported systems are:

- S71500
- S71200

Supported Field Bus :

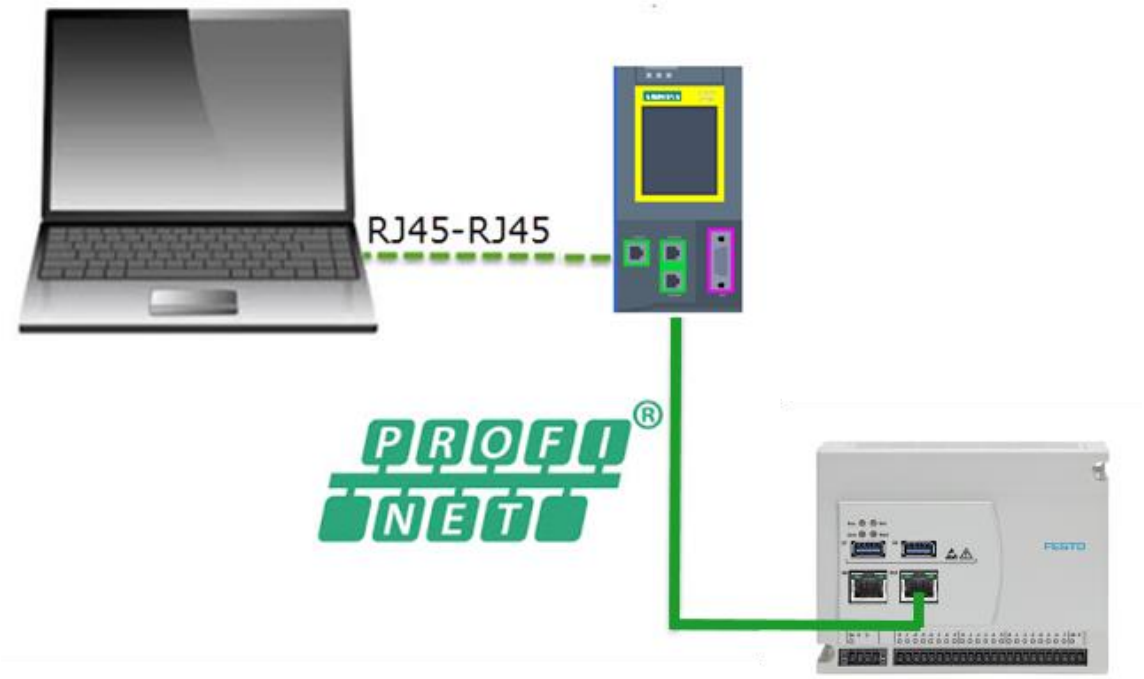
- Profinet IO

The application note has the description for the following:

- SBRD Vision System configuration in Camera Configuration Studio. CCS
- Configuring the needed payload data of the camera in Camera Configuration Studio.
- SBRD Vision System Setup in SIEMENS TIA Portal.
- Installing the GSDML File for SBRD Vision System.
- Adding the SBRD Vision System to Devices and Networks in TIA Portal.
- Description of the function blocks of SBRD Vision System.
- Integrating the Function Blocks within the programming environment of TIA Portal.

2.1 Topology

The following topology is used to communicate between S7-1500/1200 PLC and SBRD Vision System:



➔ **NOTE**

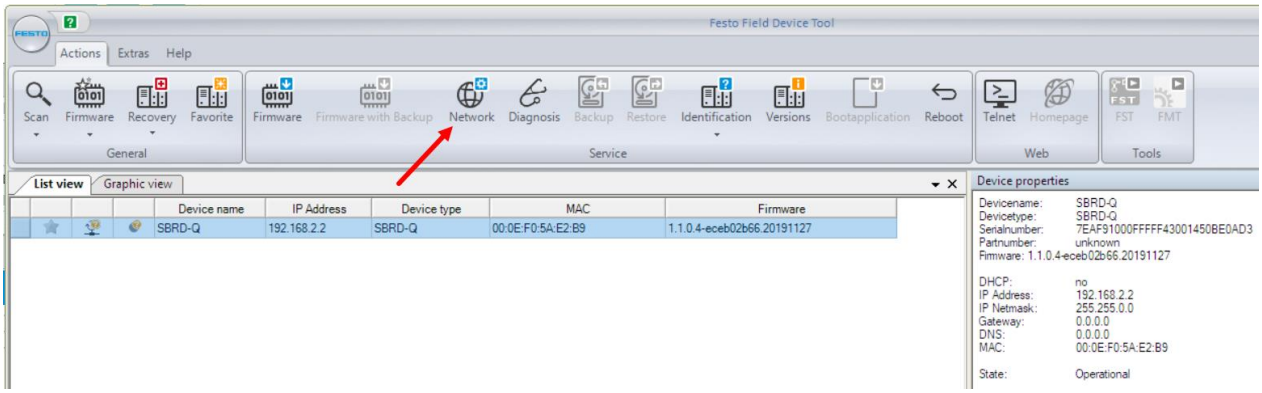
- **Only the RJ45 socket on the right side is designated for the Profinet network.**
- **The IP address of SBRD Vision System, Ethernet Port of S7-1500/1200 and the PC used for programming the PLC, must be in the same IP range.**

3 Configuration of SBRD Vision System with Camera Configuration Studio for PROFINET.

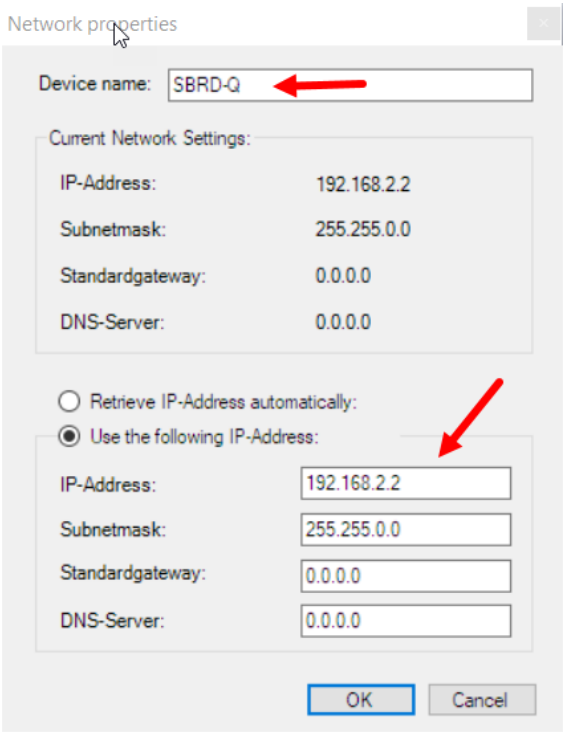
Remark:
The following explanation show only a basic vision project to enable the communication by Profinet. It is not intended to show the capabilities of the CCS software.

3.1 IP configuration of the SBRD Vision System.

- Open Festo Field Device Tool .
- The following screen will be displayed in Festo Field Device Tool when SBRD controller is connected.



- Select the controller and press Network button in the ribbon.



- Now you can edit IP-address, subnet mask and the name of the device.

Important Conditions for properly working PROFINET communication

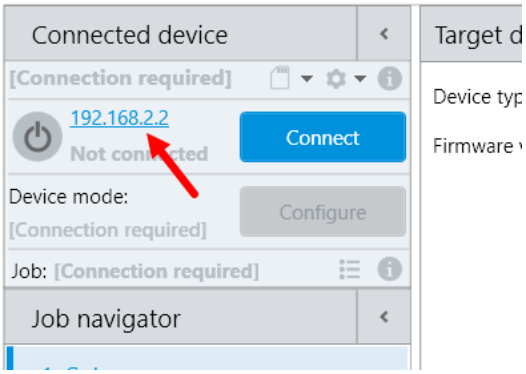
1. The Device name must be identical in PLC and SBRD controller.
2. The IP address of SBRD and PLC must correspond (must be in same address range).
3. IP address and name can be set in different ways:
   - **Via FFT software** (Festo Field Device Tool)

- **Via PLC interface** (TIA portal). Refer **Chapter 6** to get detailed description on how to set SBRD-IP address and device name from TIA portal.

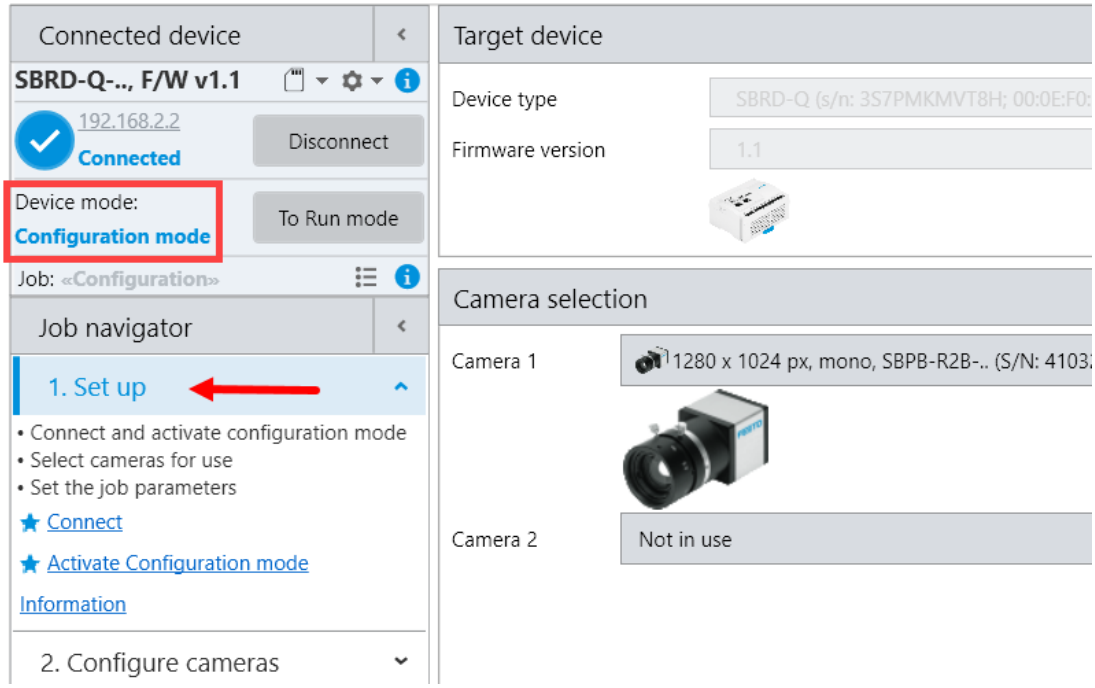
3.2 Create a project with CCS software

- Open CCS and select New Job

- Click on the shown IP-address to open the search process for the connected device.



- Select the device and connect
- If device is connected click on Configure to enter the configuration mode of the system

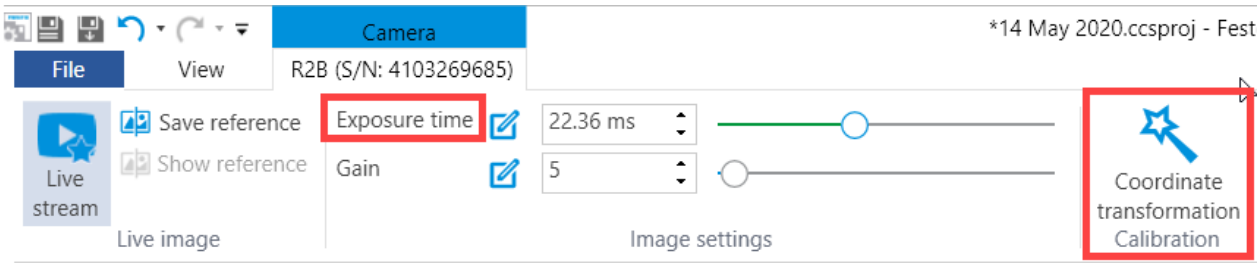


3.3 Job navigator 1. Step Set up

- Follow the steps in Job navigator. 1. Set up: select the camera
- The auto detect function show all the connected cameras. Select the one which is used for the task.

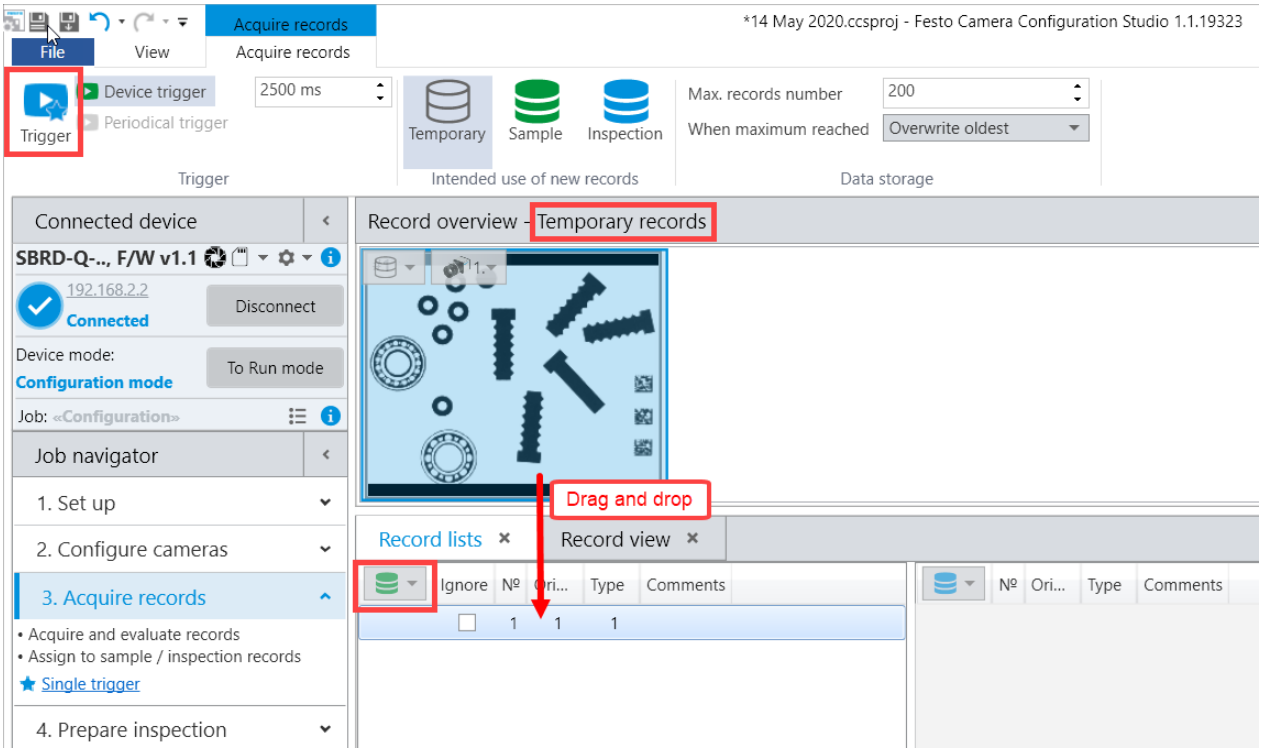
3.4 Step 2 Configure cameras

- Adjust the exposure time and do a calibration of the coordinate system, if it is necessary, to get the results in mm-values. Details explained in the CCS online help (File › Help) chapter 5.



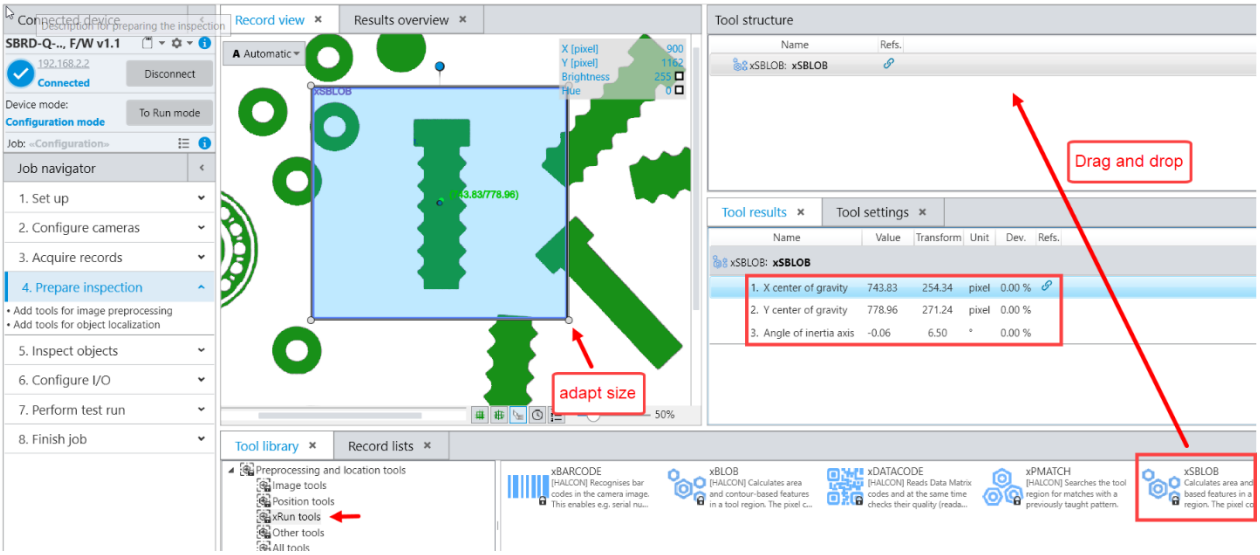
3.5 Step 3 Acquire records

- Place a test part in front of the camera and capture an image with the trigger button. At least 1 image is required.
- Drag and drop one image to the sample part column (green container icon) in the record list.

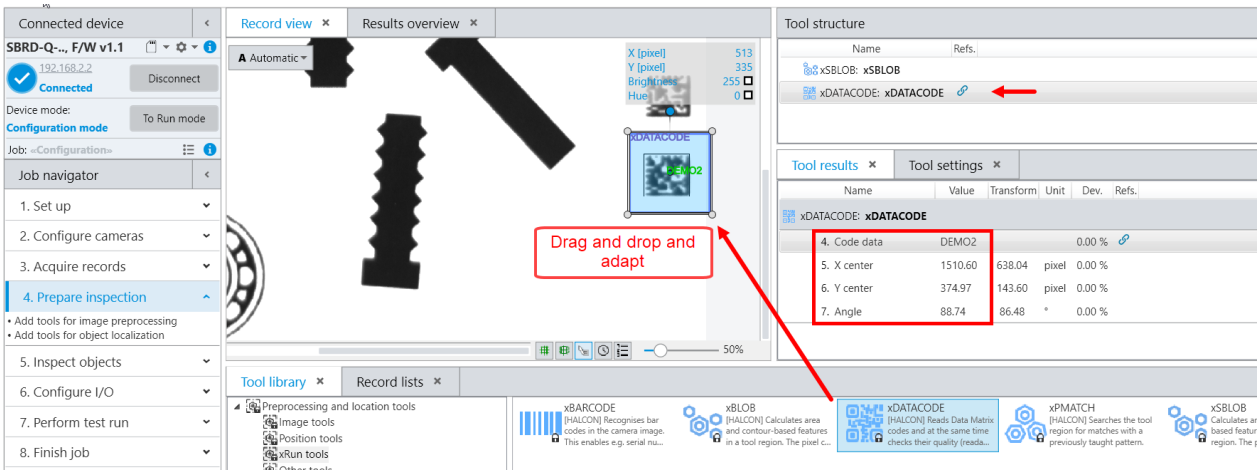


3.6 Step 4 Prepare inspection

- This application note focus on 2 use case. The first use case is to transfer x,y-coordinates. The second use case is to transfer a string variable of a bar code to the PLC.
- Use case 1: apply a tool e.g. xSBLOB to the part in the image. This is possible by double click on the tool or by drag and drop out of the Tool library into the image or the Tool structure.
- Move and adapt the search region of the xSBLOB, that it covers the field of view.
- By default, the xBLOB calculate x,y-coordinate and an angle. This results will be transferred by Profinet to PLC in this example. The settings in the tool could be more specified, but for this basic example it is not necessary. In tool results you can see the calculated coordinates.



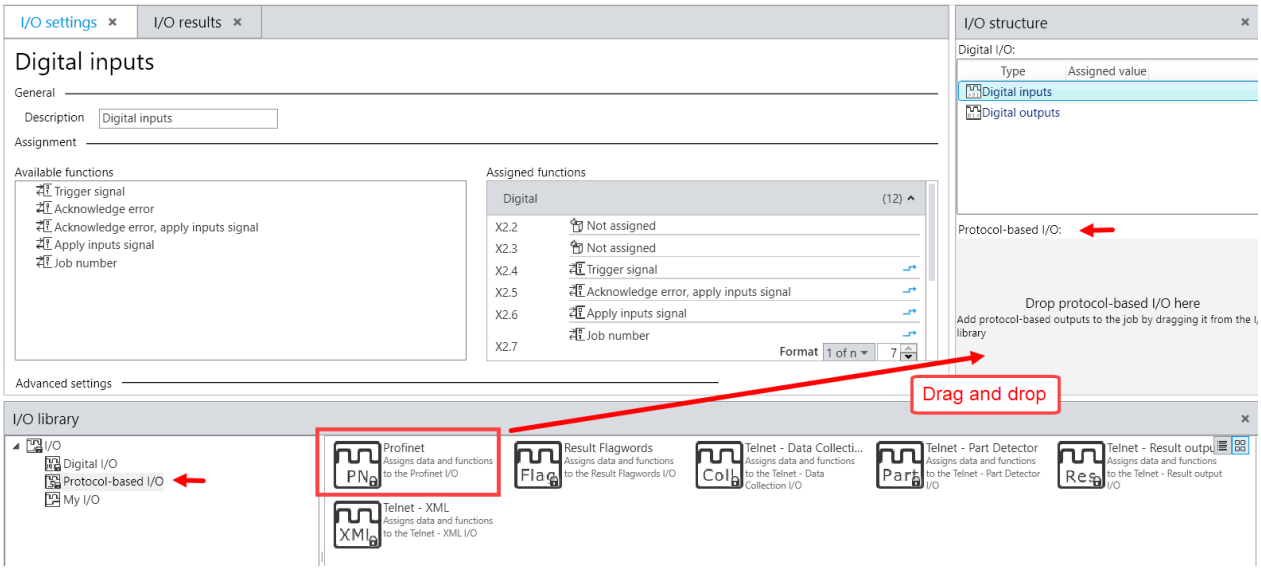
- Use case 2: Select a tool for the code reading. xBarcode or xDatacode as in the image is available.
- Move and adapt the search region of the tool, that it covers the field of view.
- By default, the code reader tool calculate the content, x,y-coordinate and an angle of the code. This results will be transferred by Profinet to PLC in this example. In tool results you can see the code and the coordinates.



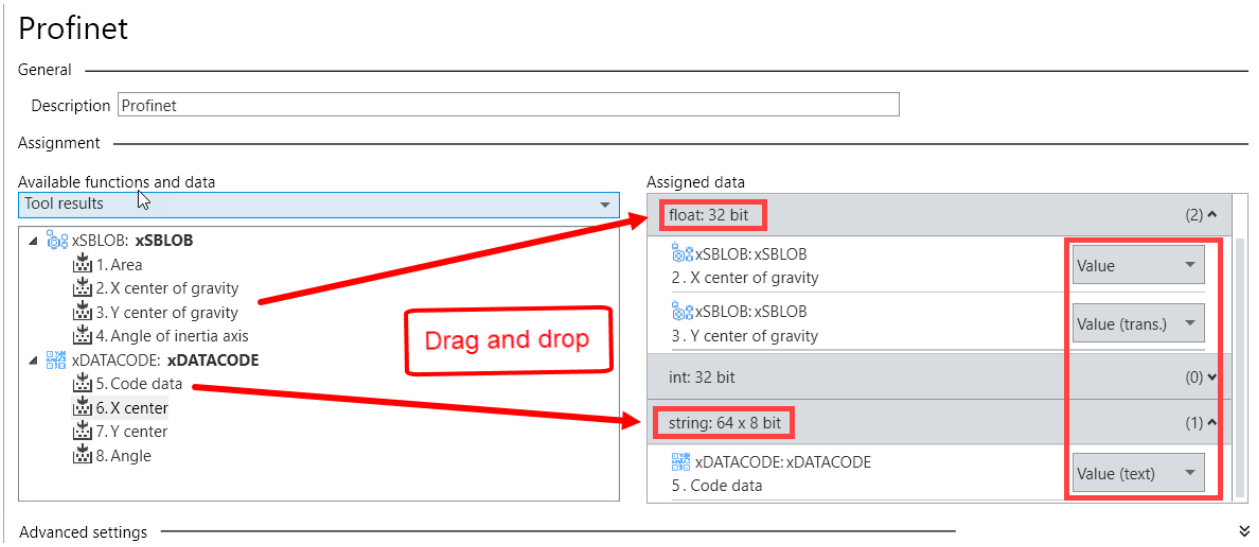
- The next **step 5** is not necessary due to basic vision example.

3.7 Step 6 Configure I/O

- Doubleclick or drag and drop the Profinet telegram out of the I/O Telegram into the field of Protocol-based I/O.



- Then it is possible to assign data to the protocol. Select the particular coordinate and drop it on the suitable data type.
- Once the data is assigned, it is possible to choose different value. For the coordinates it is possible to use the Value(trans), which mean the mm-calibrated value of this coordinate. On value it is the result in pixel.
- For the code data it is important to assign to the string 64 x 8 bit data type. And the value (text) as an ASCII format.

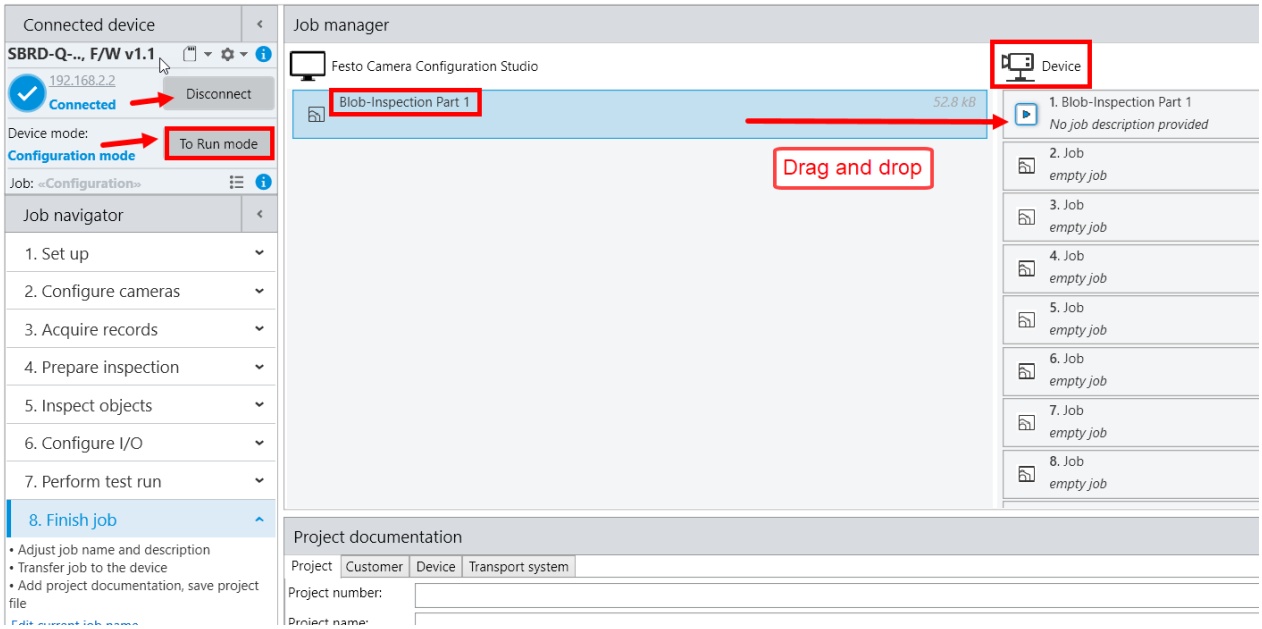


3.8 Step 7 Perform test run

- This is the step where the application and the communication to the PLC could be tested.
- Keep this step for the commissioning of the TIA controller.
- Now the SBRD-controller calculate the results and communicate. (In the other steps the results were calculated by the performance of the PC.)
- The processing time is displayed for evaluation of the application.
- Every trigger by input or Profinet protocol will capture an image and calculate new results.

3.9 Step 8 Finish job

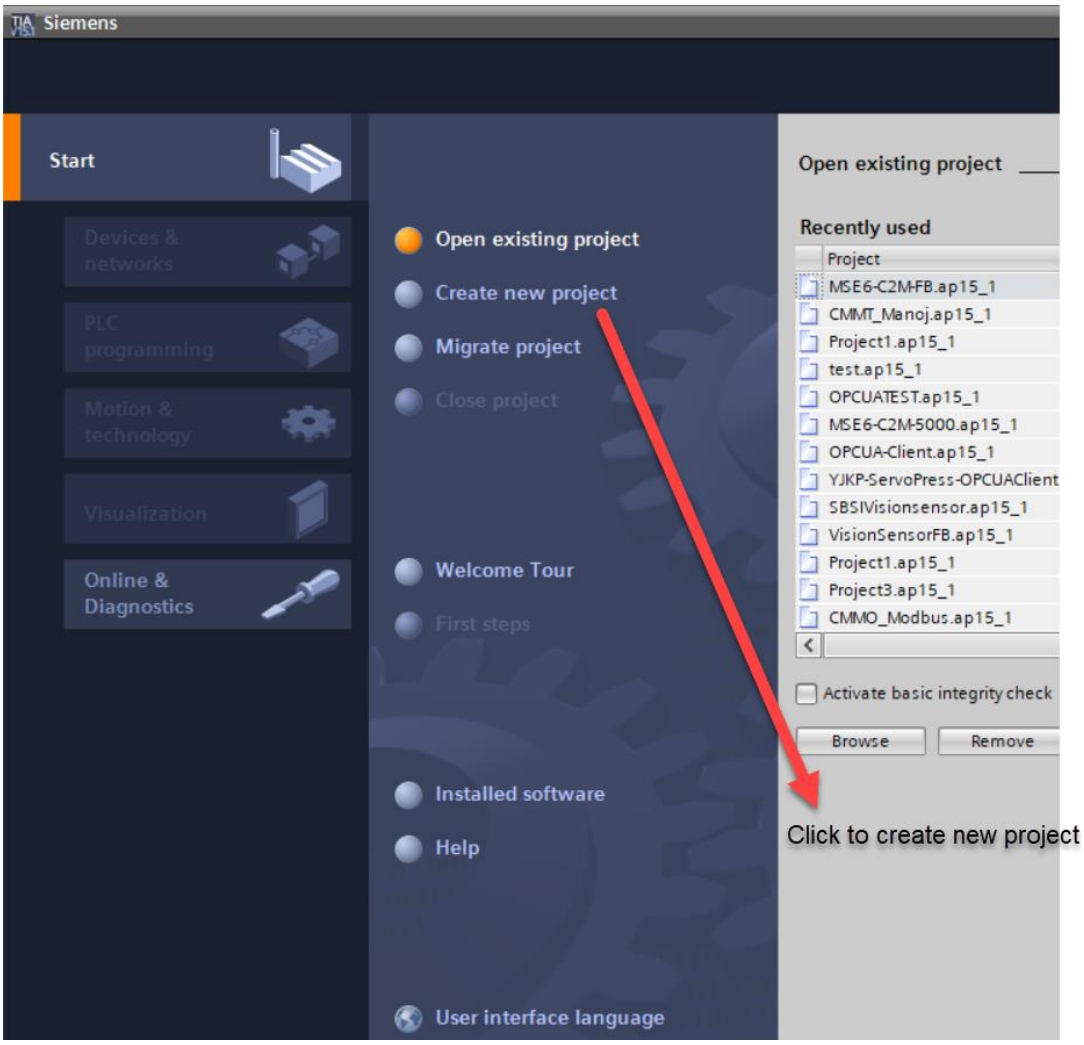
- If the commissioning of the image set-up and the communication by Profinet is successful established, the settings have to be transferred to the persistent job memory on the SBRD-controller.
- Right mouse button on the job icon allow to edit a name for the job
- Drag and drop the job on one of the possible memory.
- Click on job icon in device column to set this memory as the active job. (play button)
- Finish the step and configuration of the controller: Switch to Run mode and disconnect the device.



4 S7-1500 PLC CONFIGURATION IN TIA PORTAL

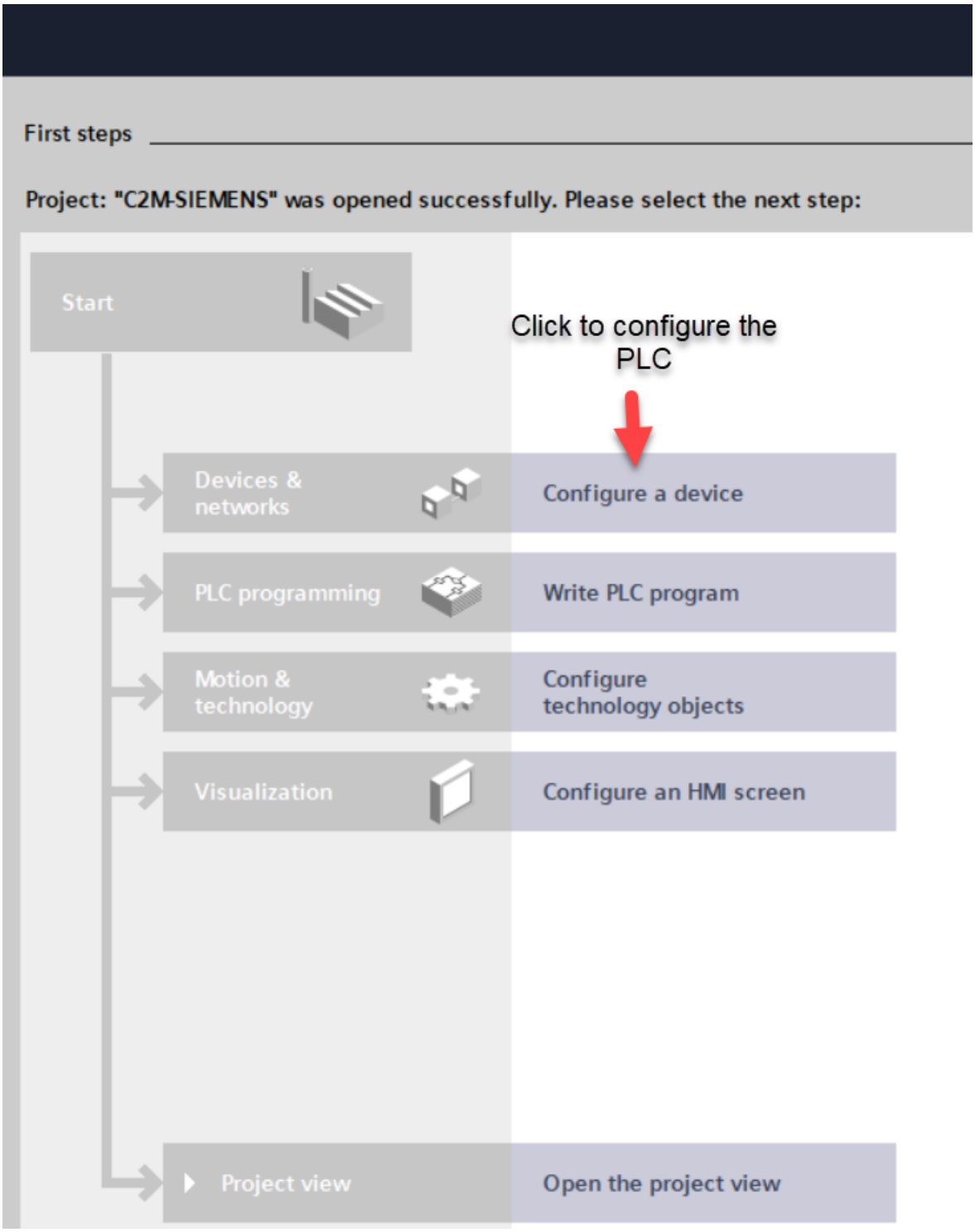
4.1 Creating a new project in TIA Portal

1. Start the TIA Portal V15.1 software.
2. Double click on **Create New Project** to create a new project.

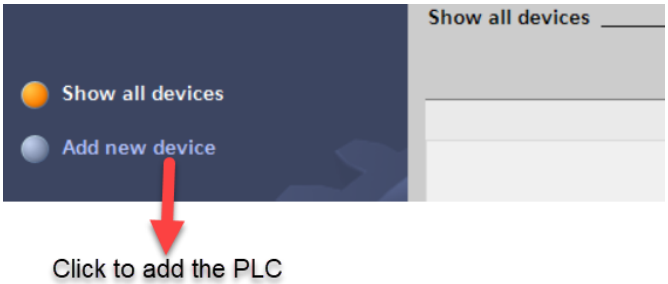


3. Enter the Project name and select the path where the project must be saved in your system. Then click on **Create** to create the project.

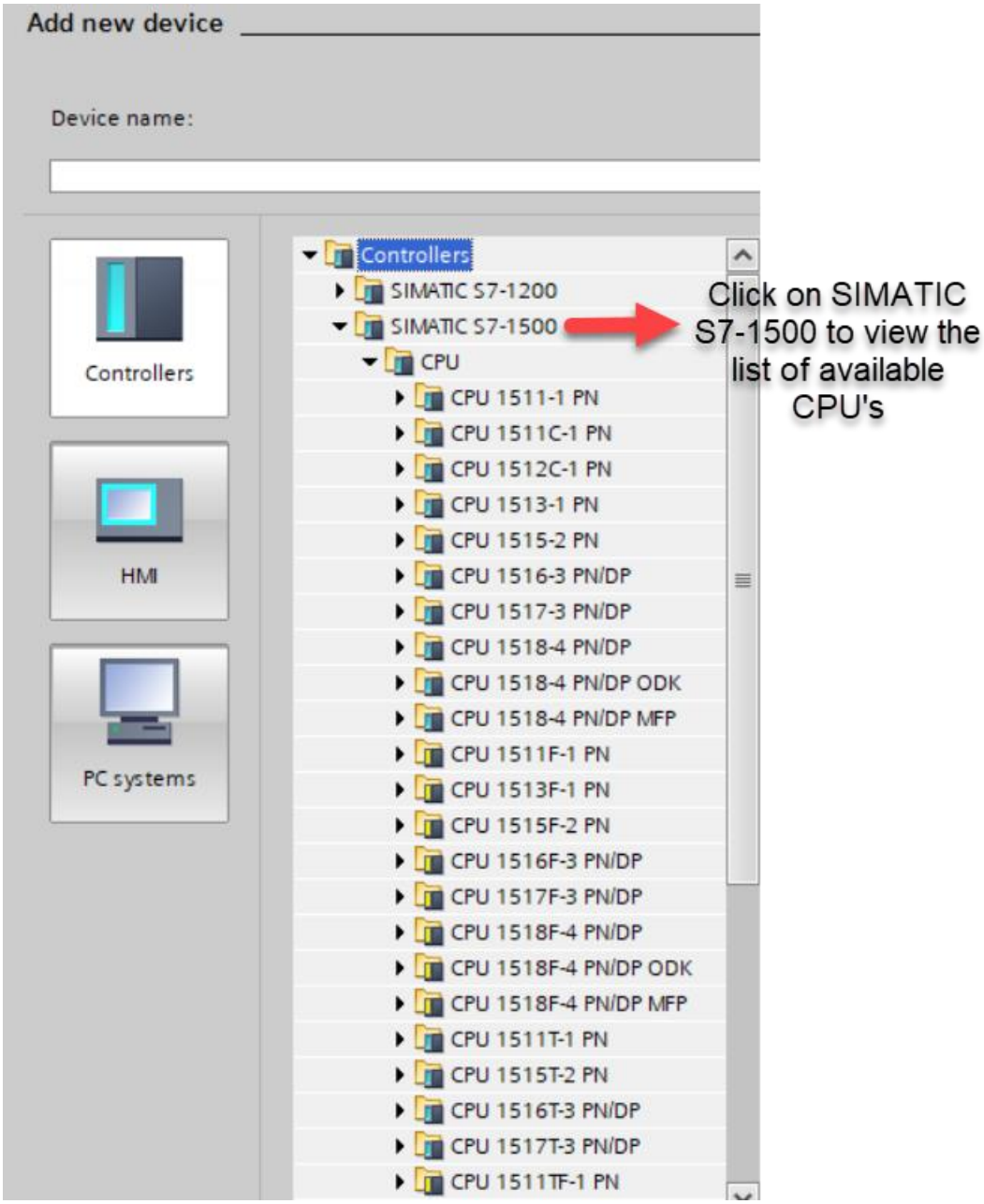
4. Double Click on **Configure a device** to configure the PLC needed**.**



5. Double Click on **Add New Device** to add a new PLC to the project.



6. Click on the PLC needed . In the below example S7-1500 is selected. All the available CPU's under S7-1500 will be displayed.

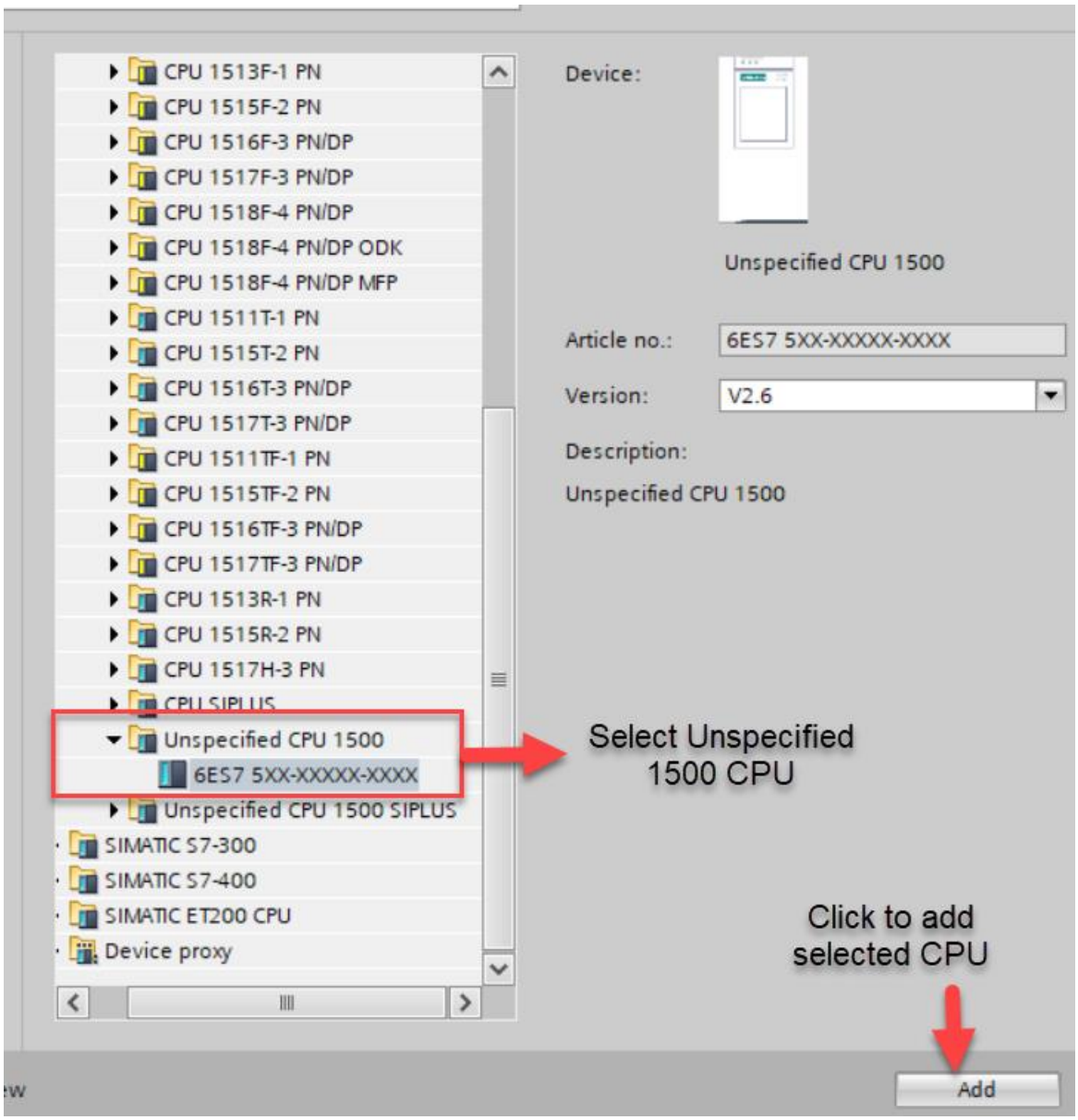


7. Click on **Unspecified CPU S7-1500** and select the PLC below it and click on **Add Device**.

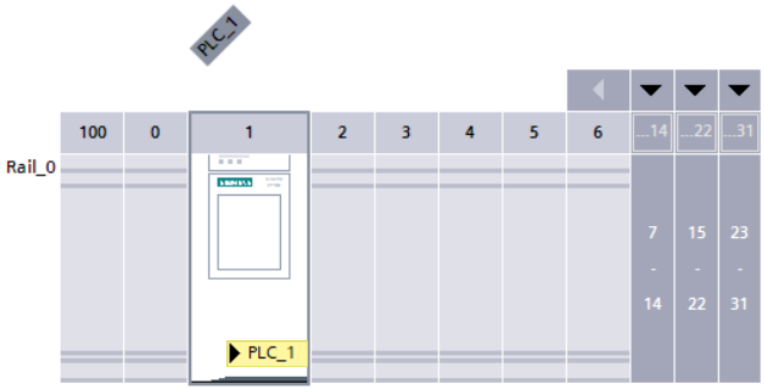
NOTE

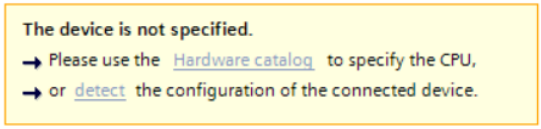
- If we select Unspecified CPU S7-1500, then the actual hardware configurations of the PLC can be read by using the detect hardware configurations option.

This saves the time needed to do the hardware configurations from the hardware catalog.



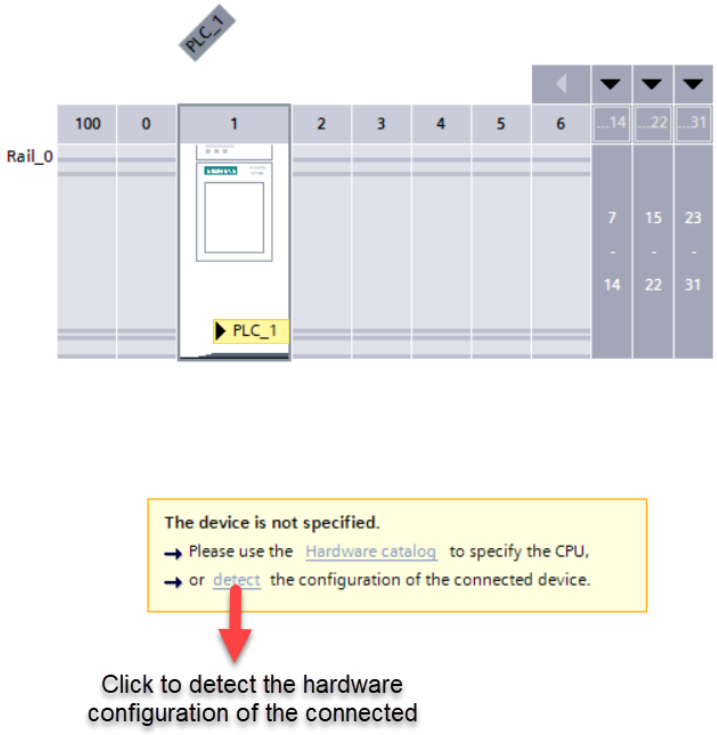
8. The project view will be as shown below.



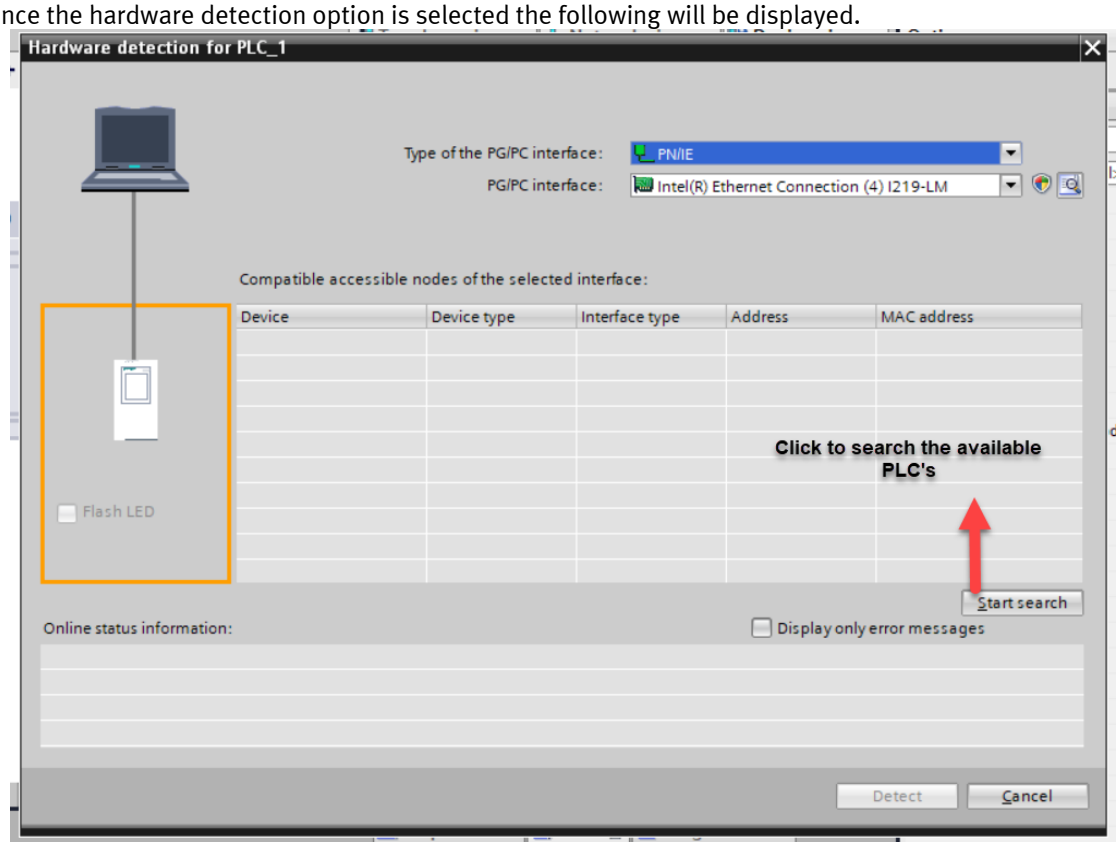


4.2 Detecting the actual hardware configuration of the PLC connected in the network

1. Click on **Detect the configuration of the connected PLC** option to retrieve the PLC Configuration.

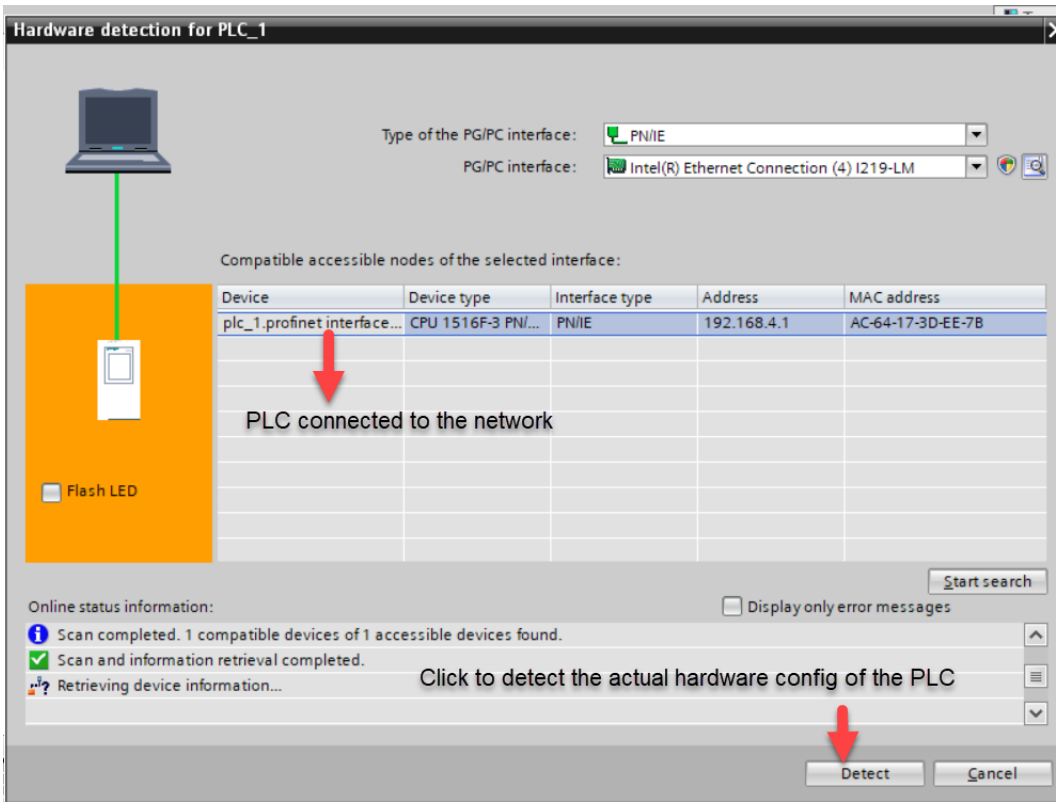


2. Once the hardware detection option is selected the following will be displayed.

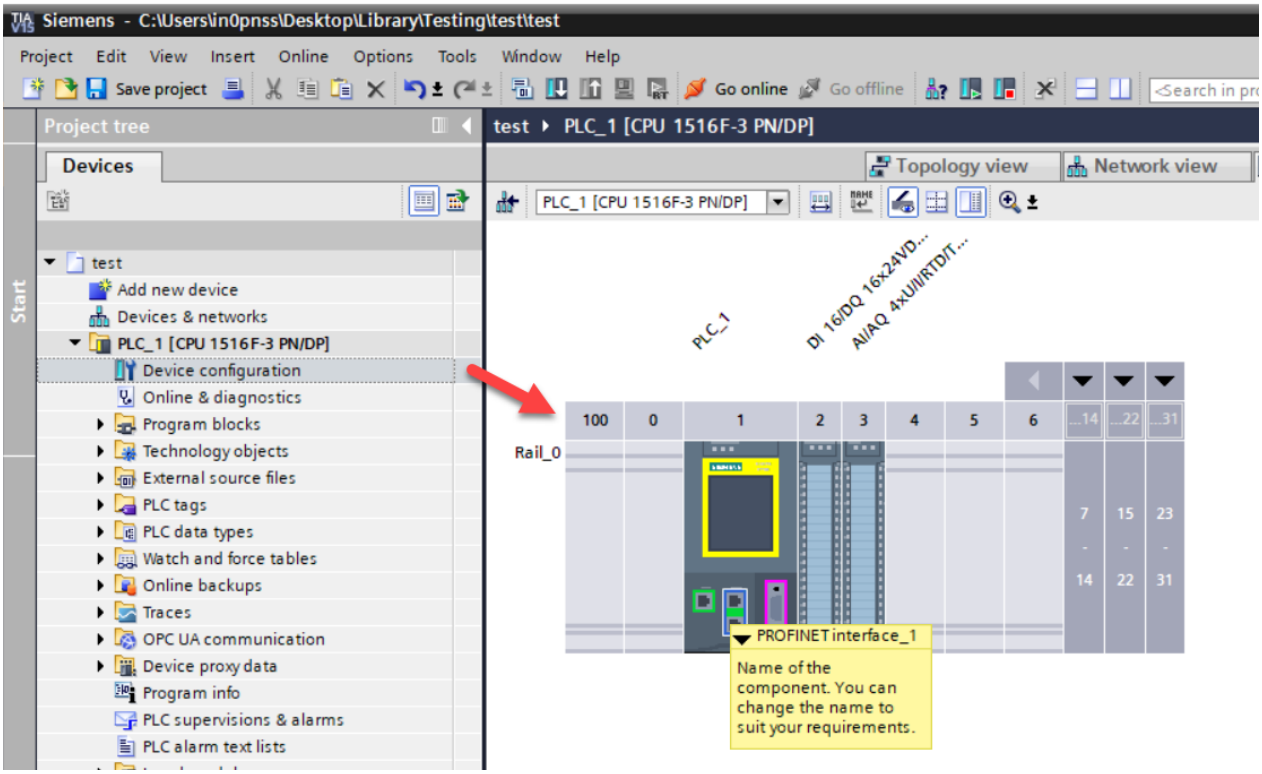


3. Click **Start Search** to find the PLC's which are available in the network as shown in the above image.

4. Once the search is completed the PLC in the network will be showed as shown below:
   Click **Detect** to retrieve the hardware configurations of the PLC connected in the network.



5. Select the PLC which is communicating with the SBRD controller from the PLC's found in the search.
6. Click **Detect** to retrieve the hardware configuration of the PLC.
7. Once the hardware configuration is retrieved from PLC, go to Device configuration .

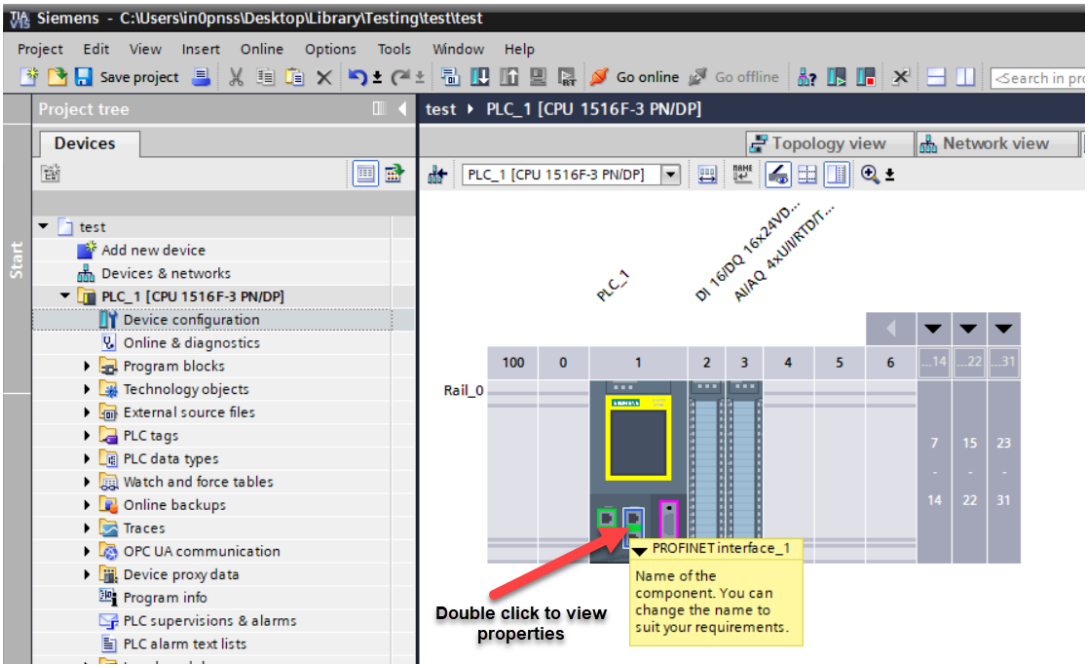


4.3 Configuration of the IP parameters of the profinet interface of the PLC.

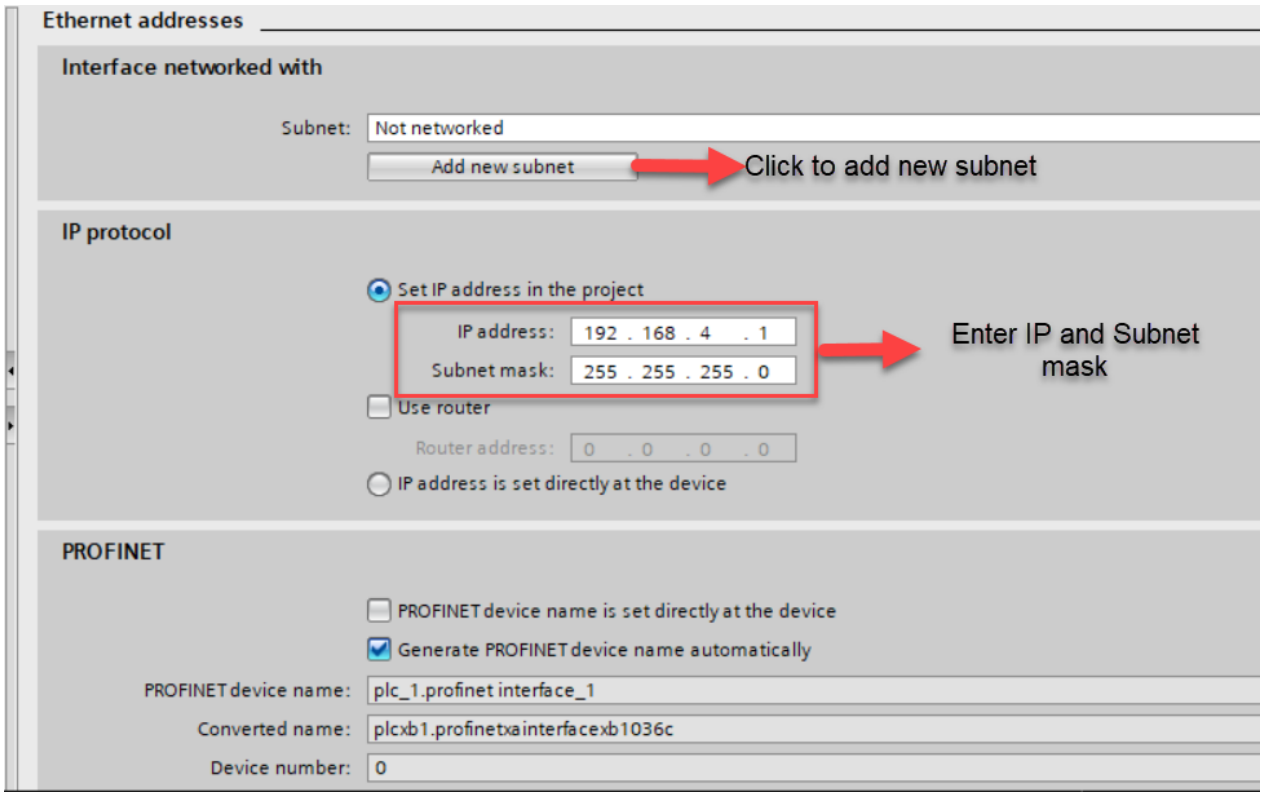
NOTE

- **The Profinet Interface address of PLC , SBRD-Vision System and the PC used for programming the PLC must be in the same range.**

1. Double click on **the Profinet interface_1**(refer above image) to view the properties of the Profinet interface.



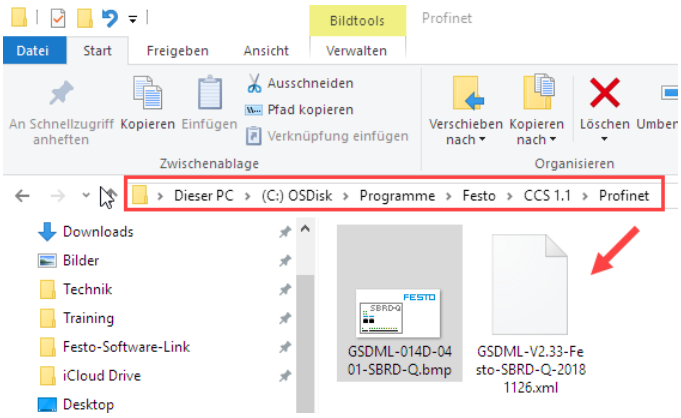
2. Go to **General › Ethernet address** to change the IP parameters of Profinet interface_1.



5 Adding GSDML File of SBRD Vision System to TIA Portal.

5.1 GSDML file

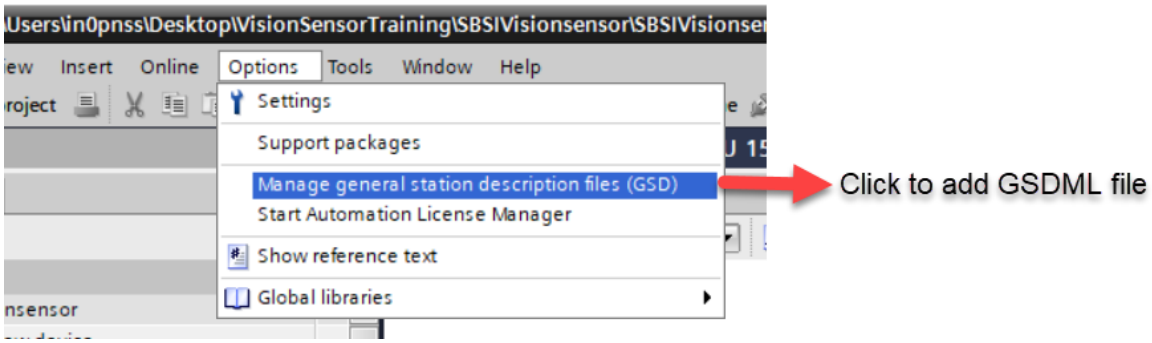
The Profinet GSDML file can be found in the location where the CCS- Software is installed.
C:\Program Files\Festo\CCS 1.1\Profinet



5.2 Adding the GSD File to TIA Portal

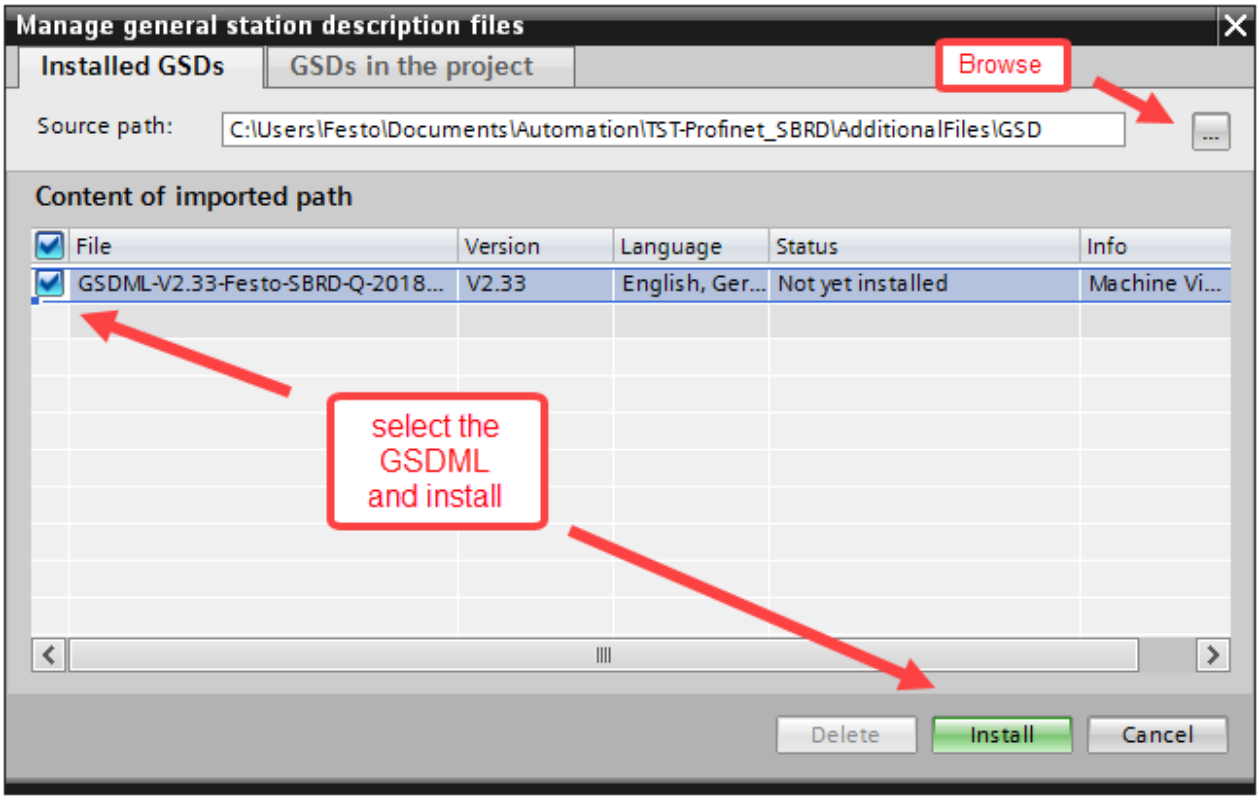
The GSDML file of SBS Vision Sensor must be added to the TIA portal.

1. Click on **Options ›› Manage General Station Description Files (GSD)**

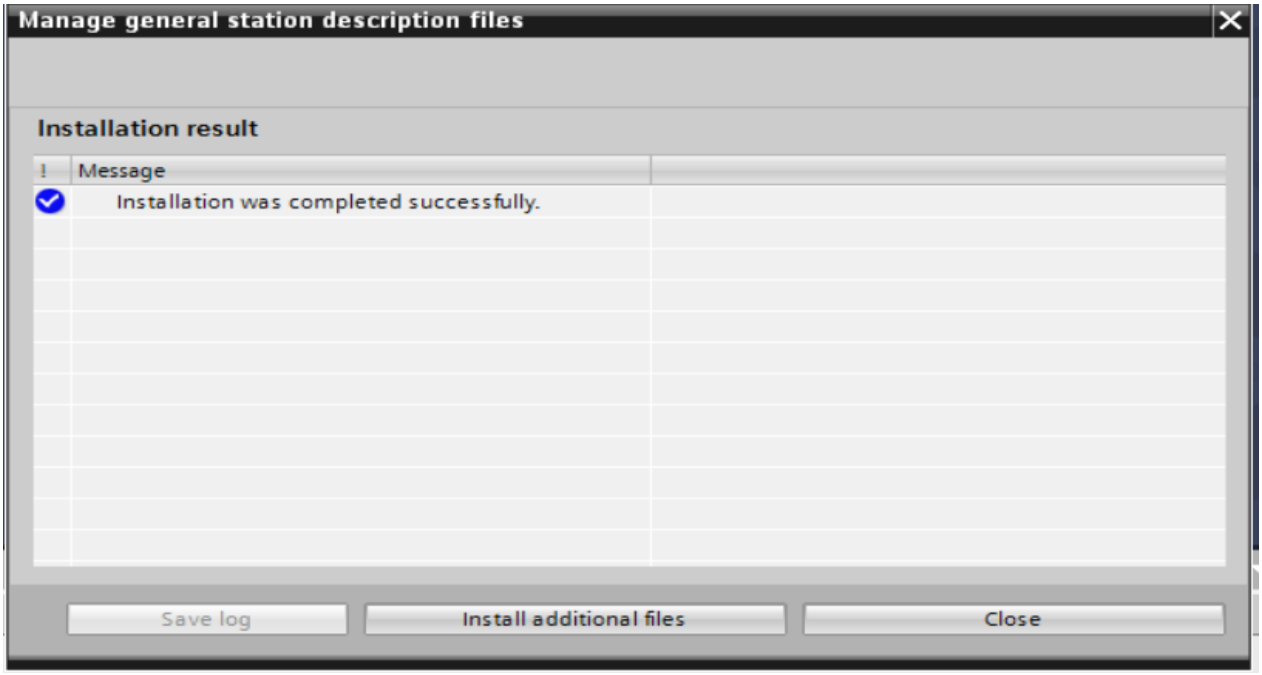


2. Browse to the folder where the file is located.

3. Once the GSDML file has been selected, click on **install** to start installing the GSDML file.



4. Once the GSDM file is installed the following pop up will be displayed.

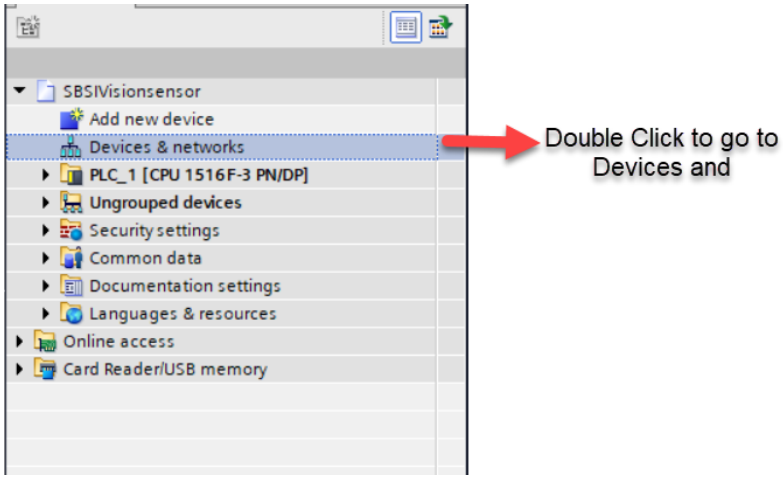


5. Click on **CLOSE** button to finish the installation and update the hardware catalogue.

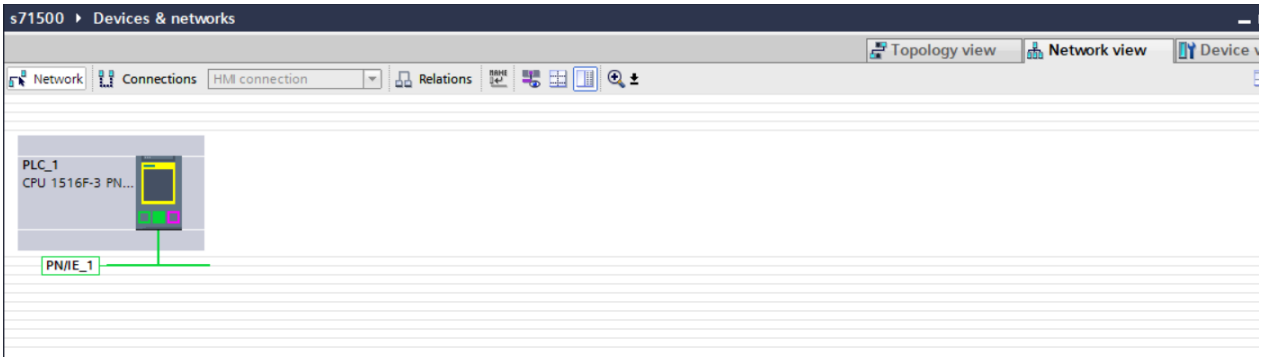
6 Configuration of SBRD Vision System in TIA Portal.

6.1 Adding the installed SBRD Vision System to Network View.

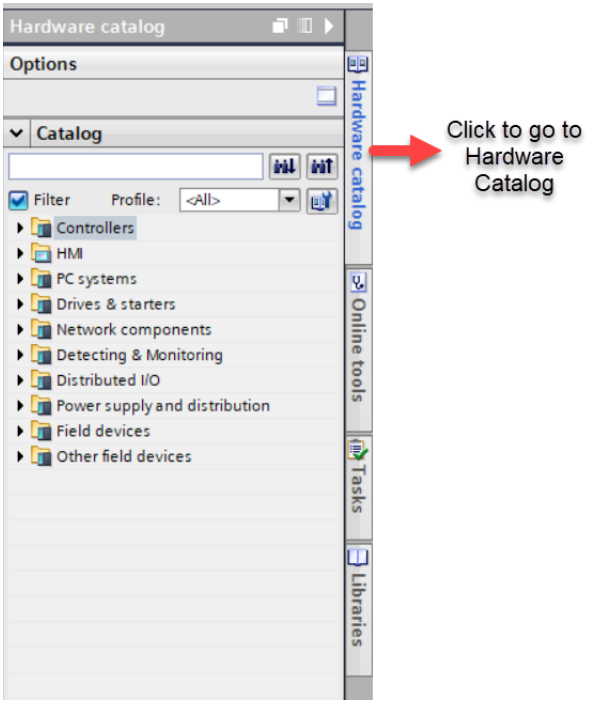
1. Double on **Devices and Networks** .



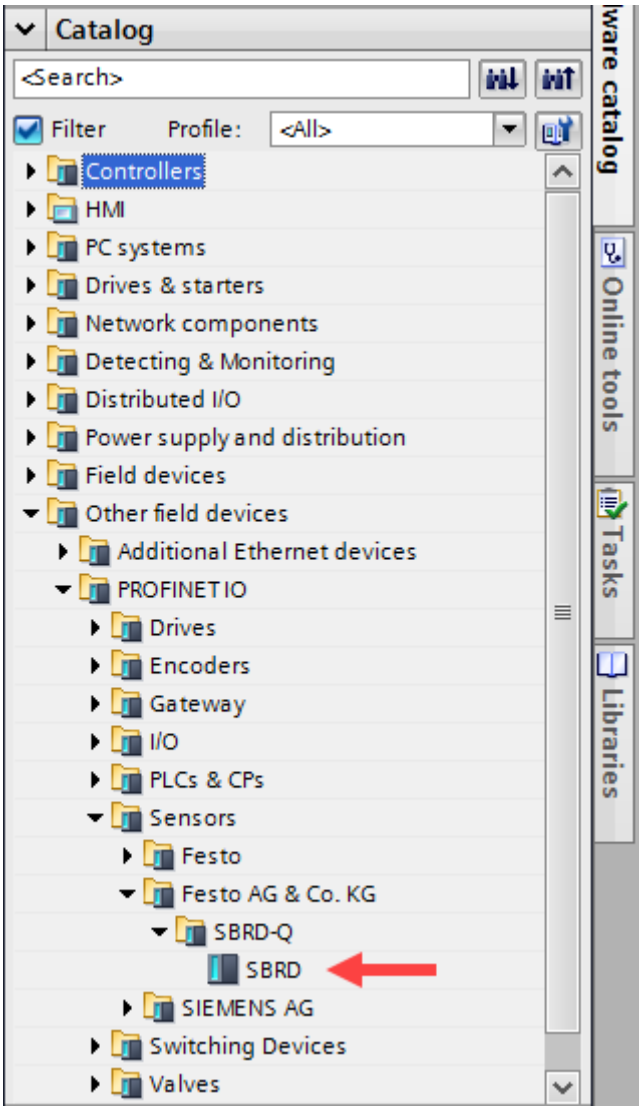
2. The network view will be displayed as shown below.



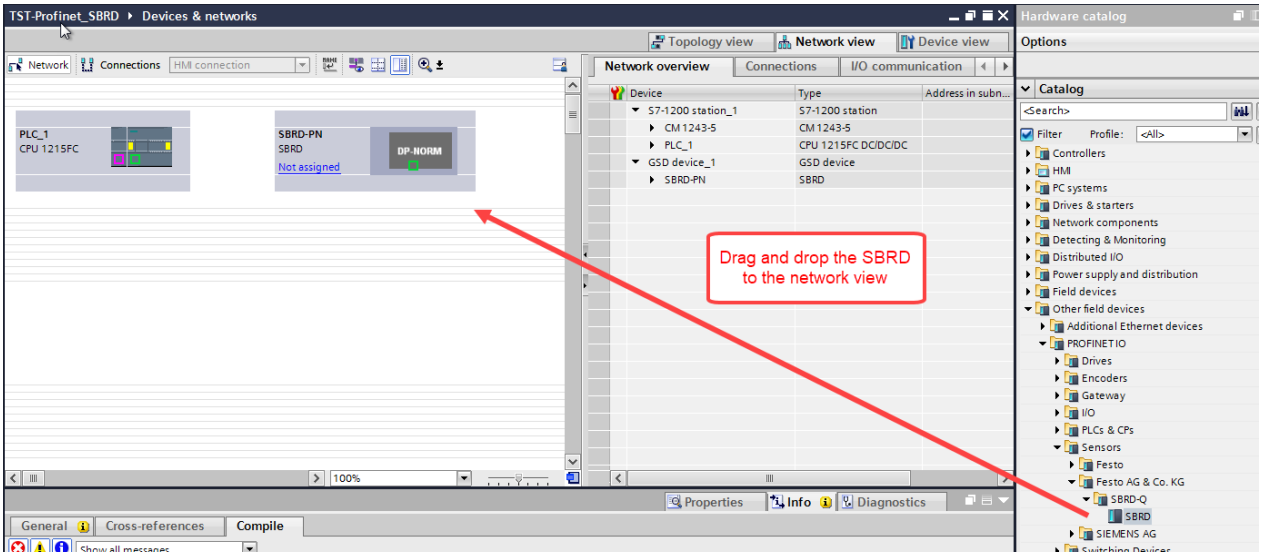
3. Go to **Hardware Catalog** as shown below.



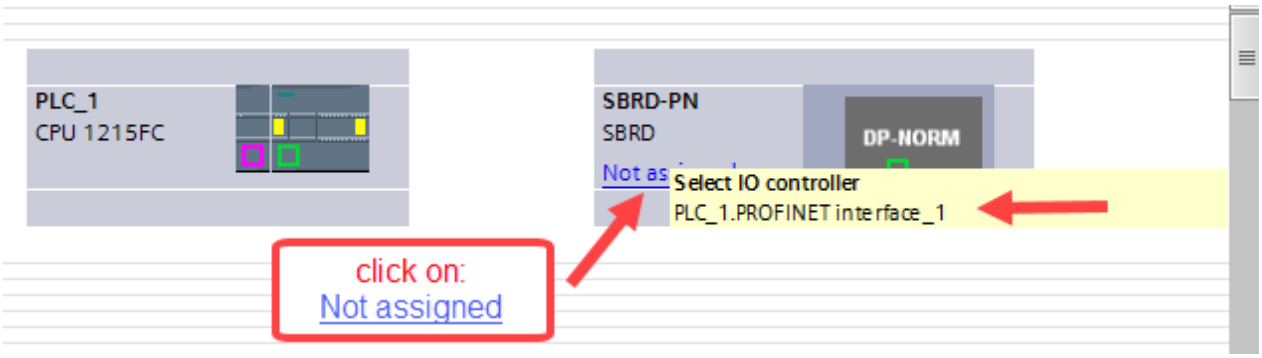
4. Under Hardware Catalog, Go to
Other Field Devices ›››Profinet IO ››› Sensors ››› Festo AG&Co. KG ››› SBRD.



5. Drag and drop the SBRD to the network view as shown below.

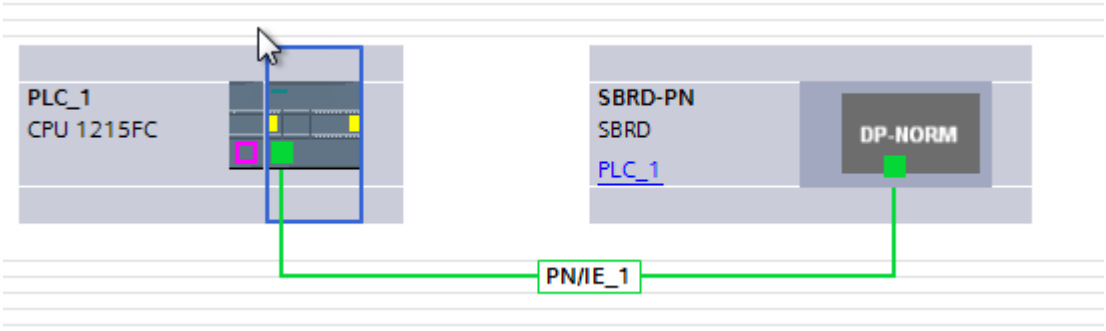


6. Connect the S7-1500 PLC to SBRD controller.
7. Double click on **Not Assigned** as shown below.



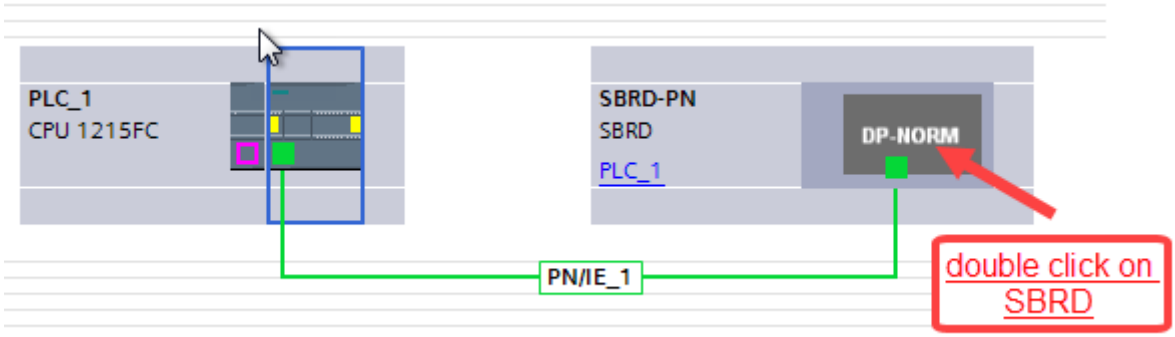
8. Select the Profinet interface. If there are more then select the correct one, where the controller is physically conneted.

After the profinet interface is selected the connection in the device view will look as shown below:

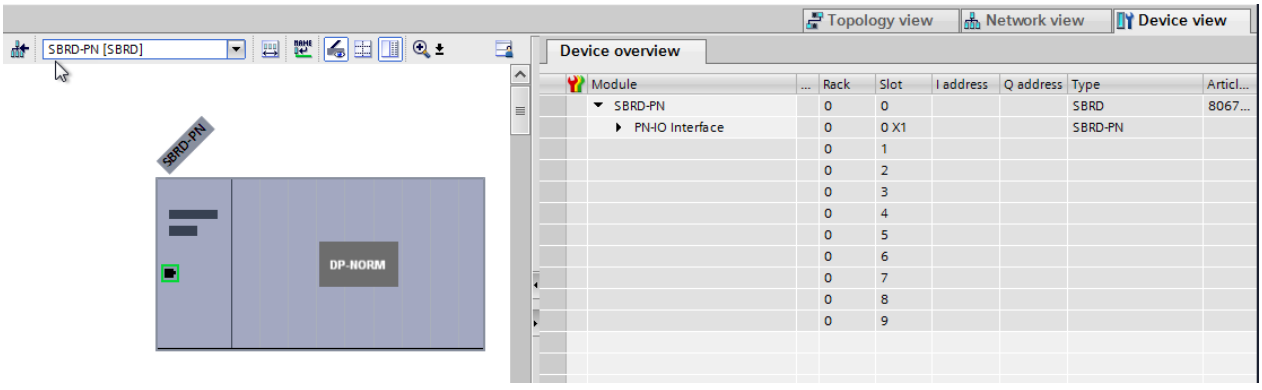


6.2 Network Configuration of Profinet Interface SBRD Vision System in TIA Portal.

1. Go to **Devices and Networks** and Double click **SBRD**.

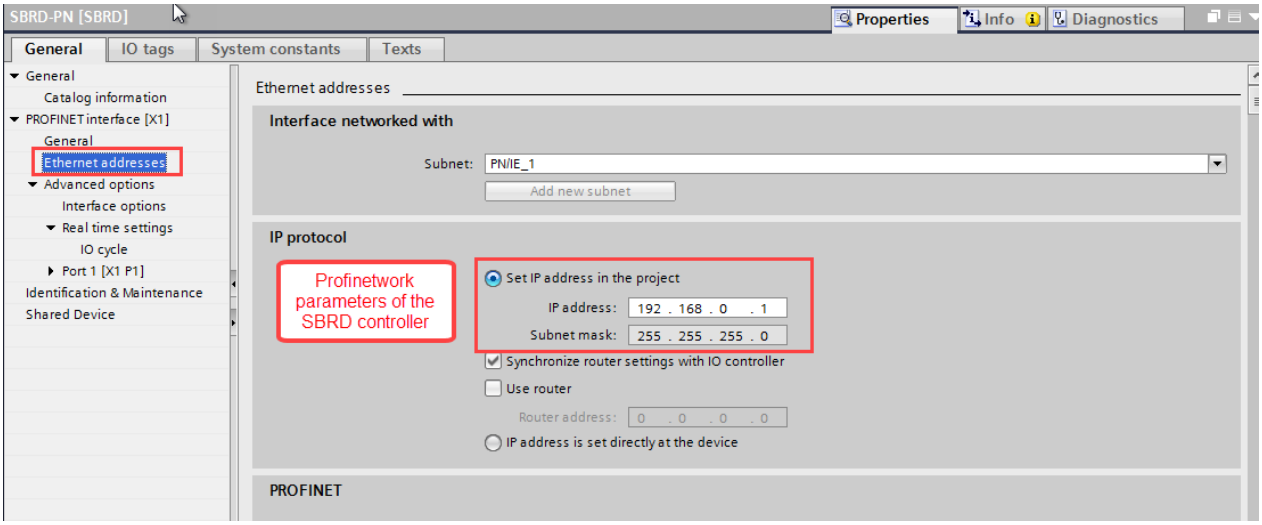


2. Once you double click the following display will be seen.



3. Click on **SBRD** to view the properties .

4. The Network parameters can be viewed under **Ethernet Addresses** tab as shown below.

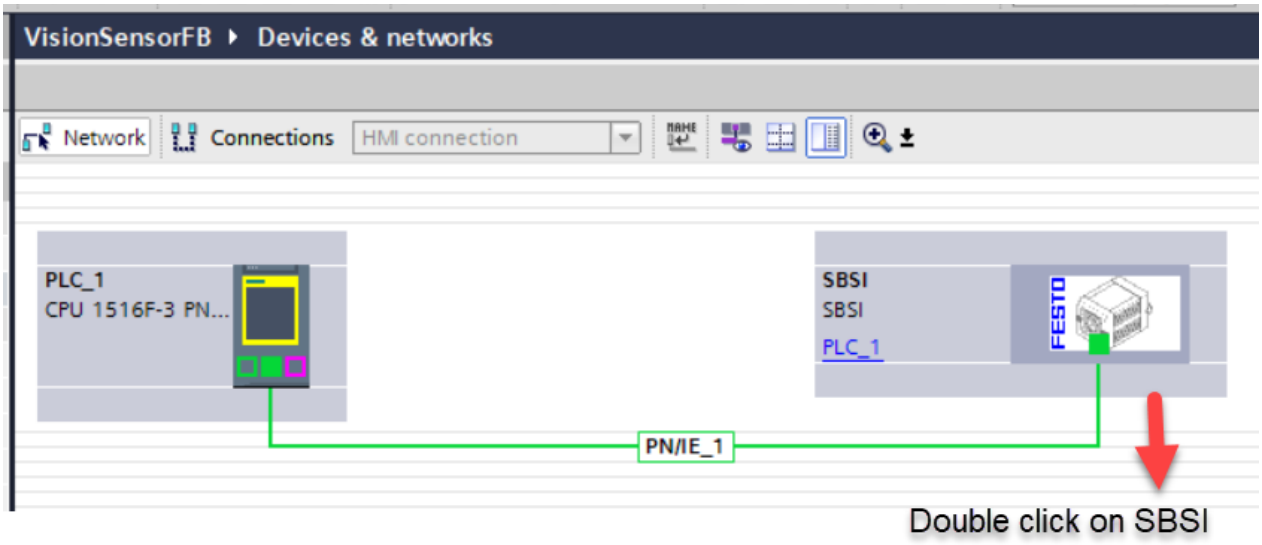


NOTE

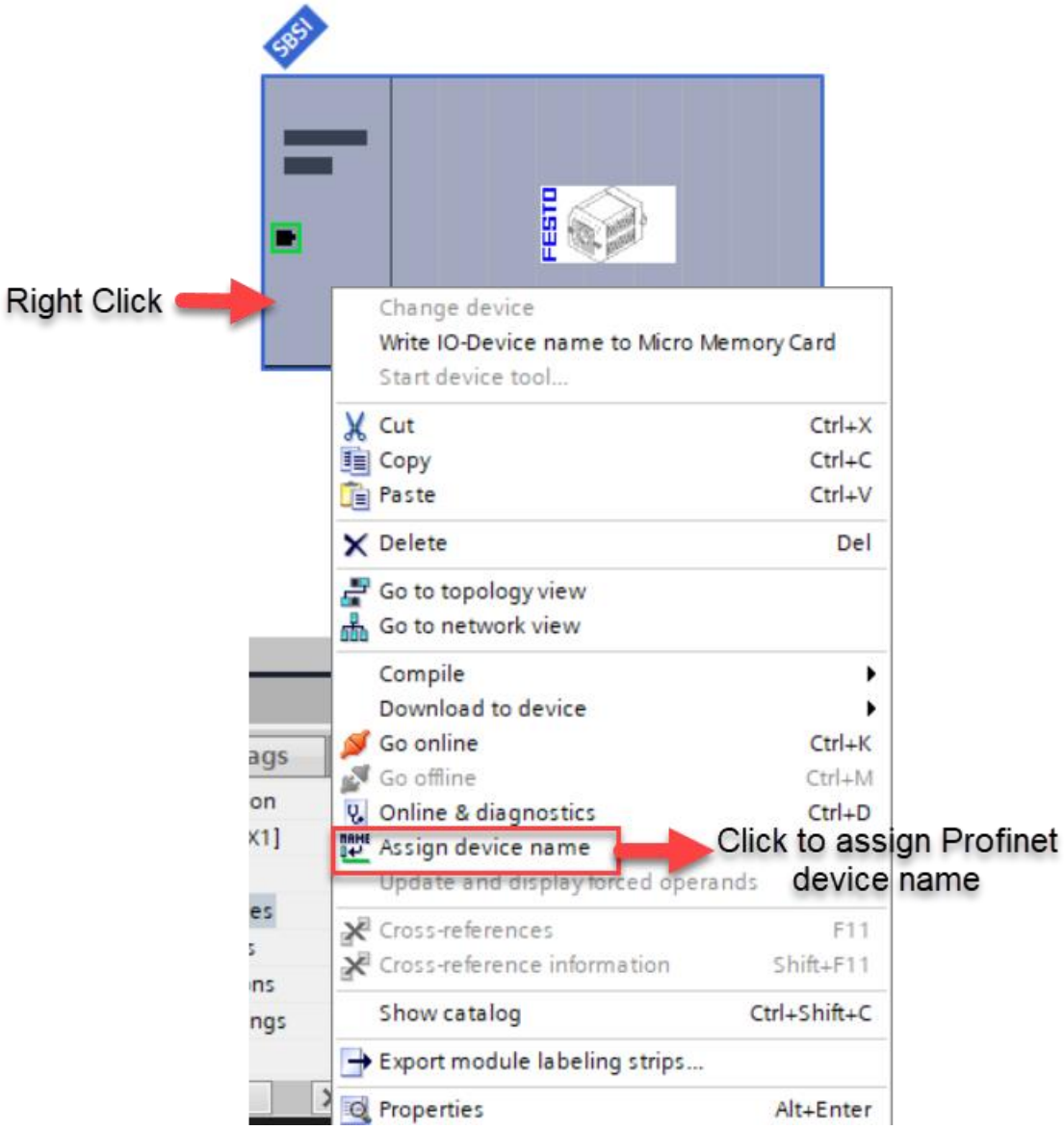
- The IP address of the Profinet interface **SBRD , S7-1500 PLC and** IP address of your PC must be in the same range.

6.3 Assigning Profinet name to SBRD Vision System.

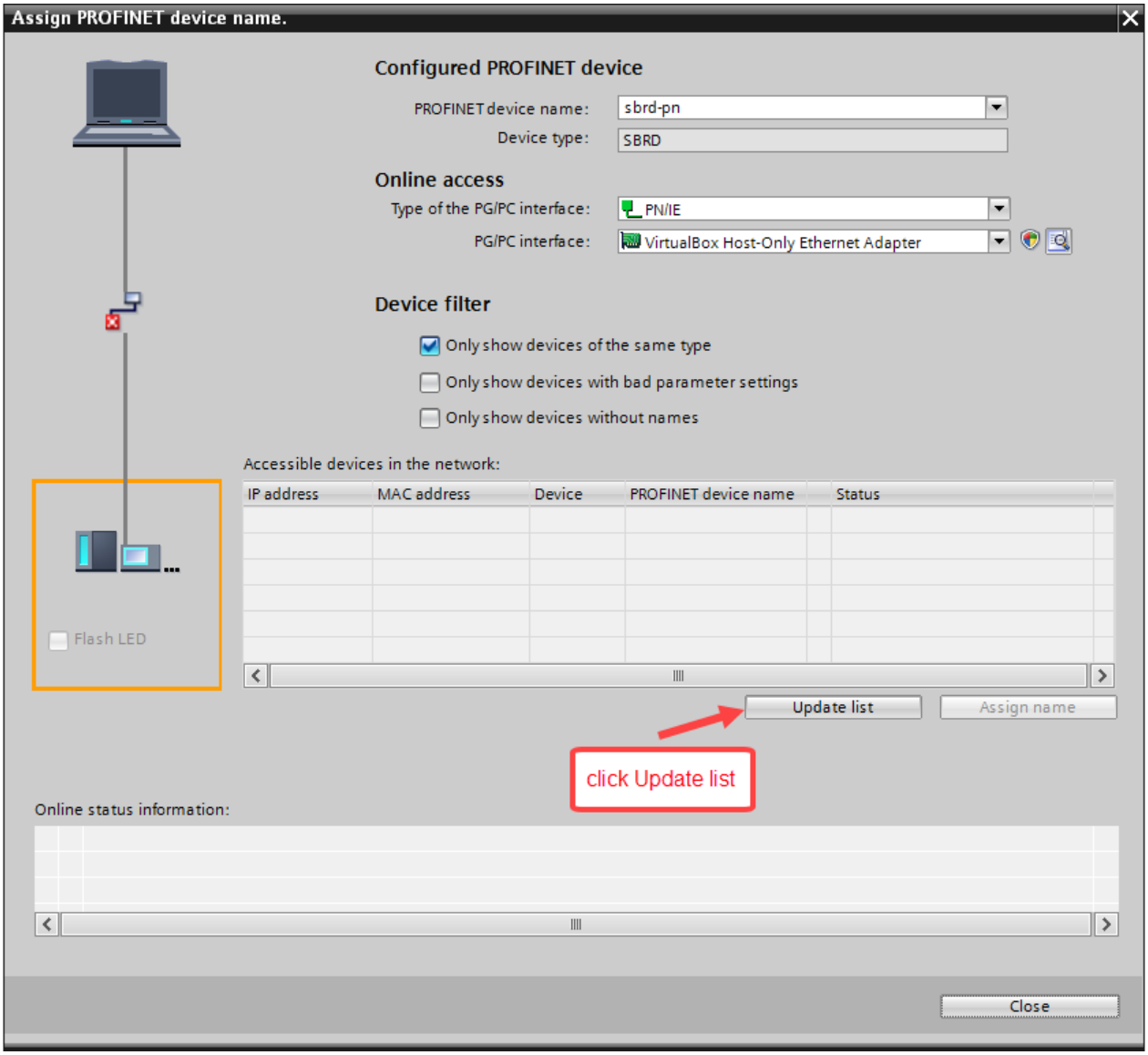
1. Go to **Devices and Networks** . Double click **SBRD** to go to **device view**.



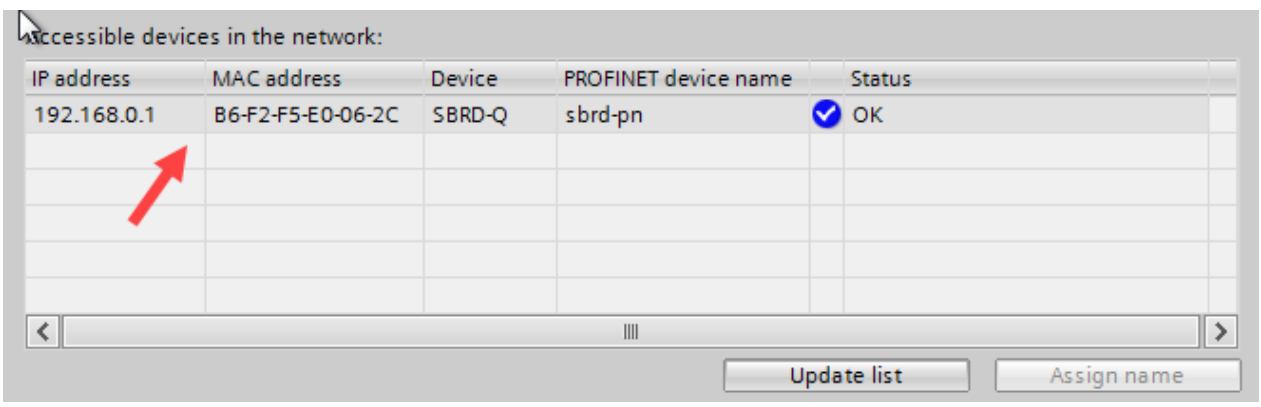
2. Right click on **MSE** ›› Click **Assign Device Name** as shown below.



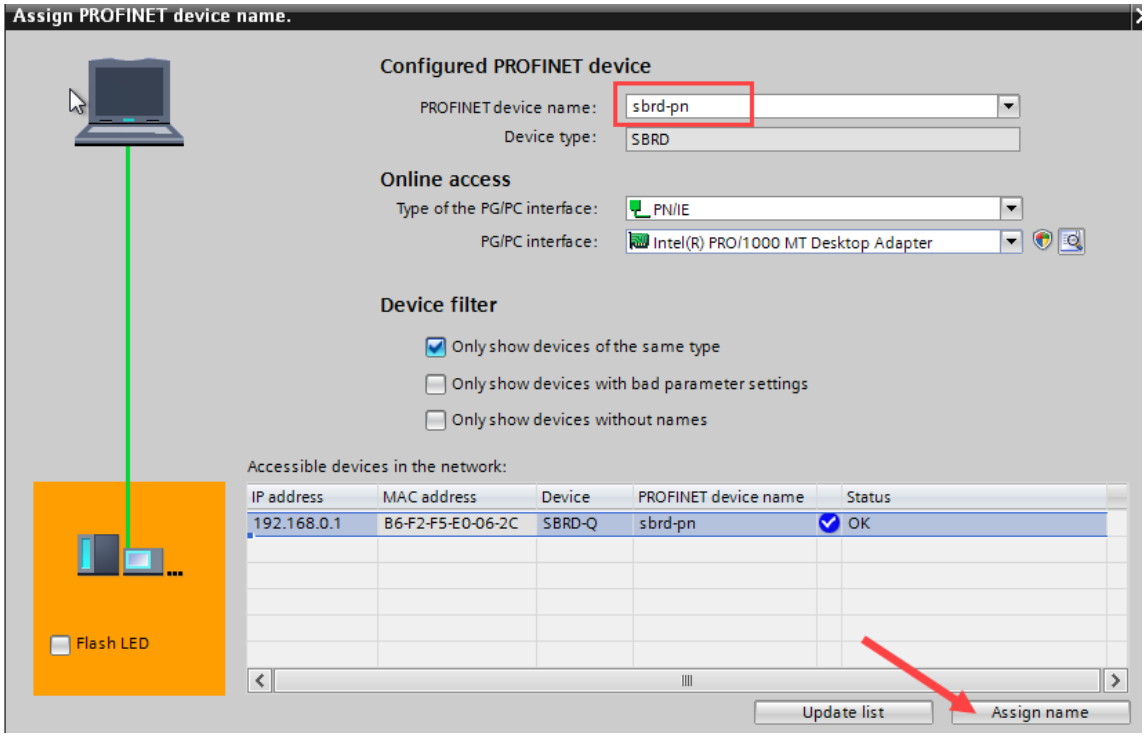
3. The wizard for assigning Profinet Device name looks as shown below.
   Search the available devices in the Profinet network by clicking **Update List.**



4. Once the search is completed the available profinet devices in the network will be displayed.



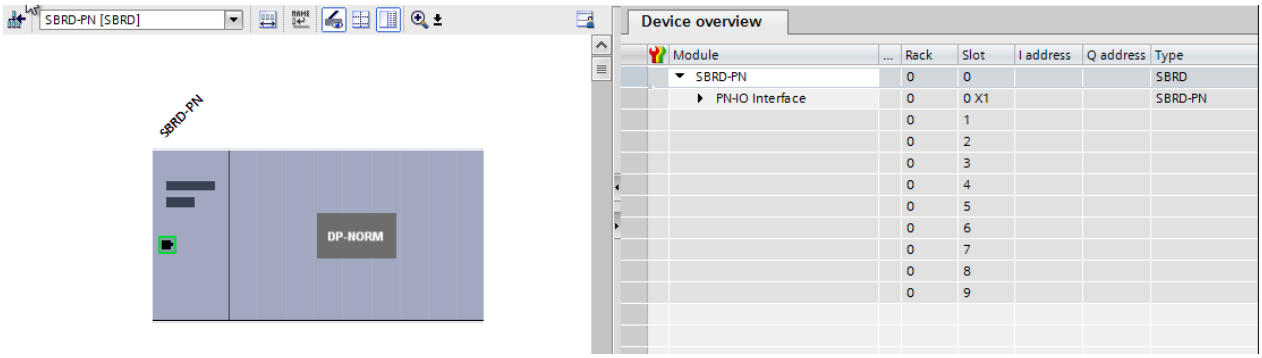
5. Click **Assign Name** to assign the profinet device name to the project.



6.4 Adding Data Module to SBRD.

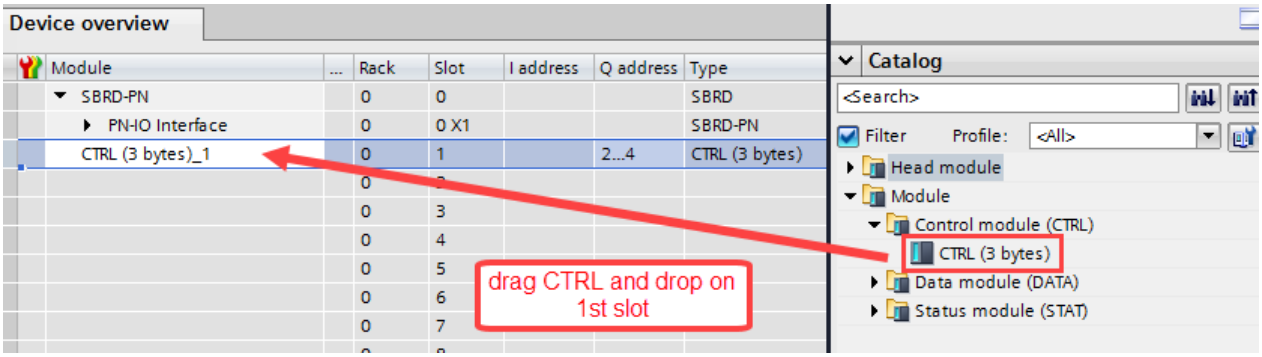
1. Go to **Devices and Networks** . Double click **SBRD** to go to **device view**.

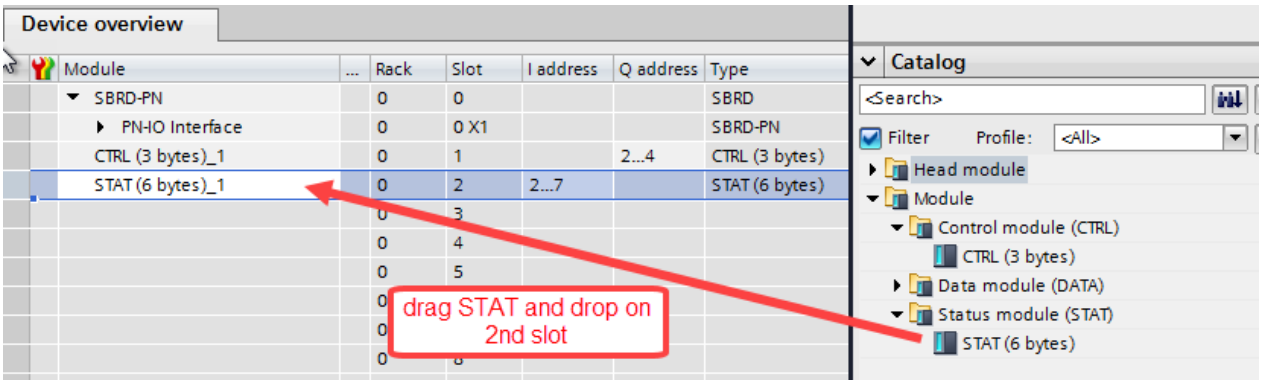
- Device view appears as shown below.



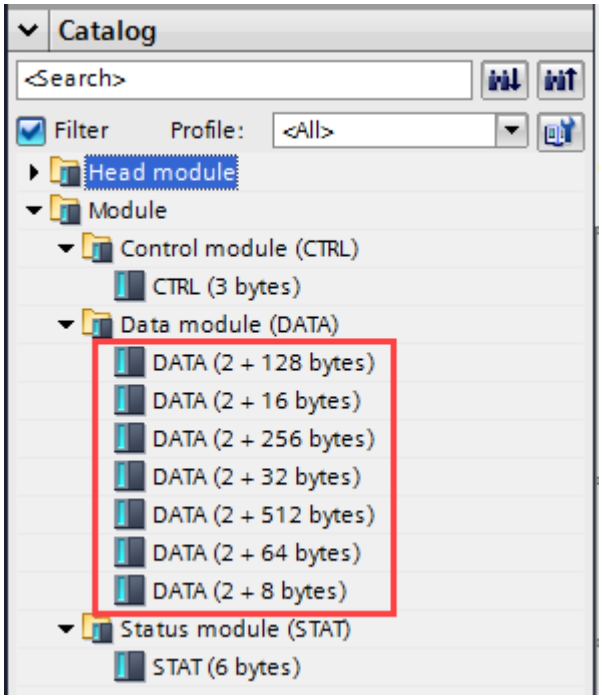
- By default there are no bytes addressed! The SBRD Profinet interface provide control, status and data modules. Now the particular modules have to be adjusted according to following order:
  1. Ctrl
  2. Stat
  3. Data

Drag and drop the modules:

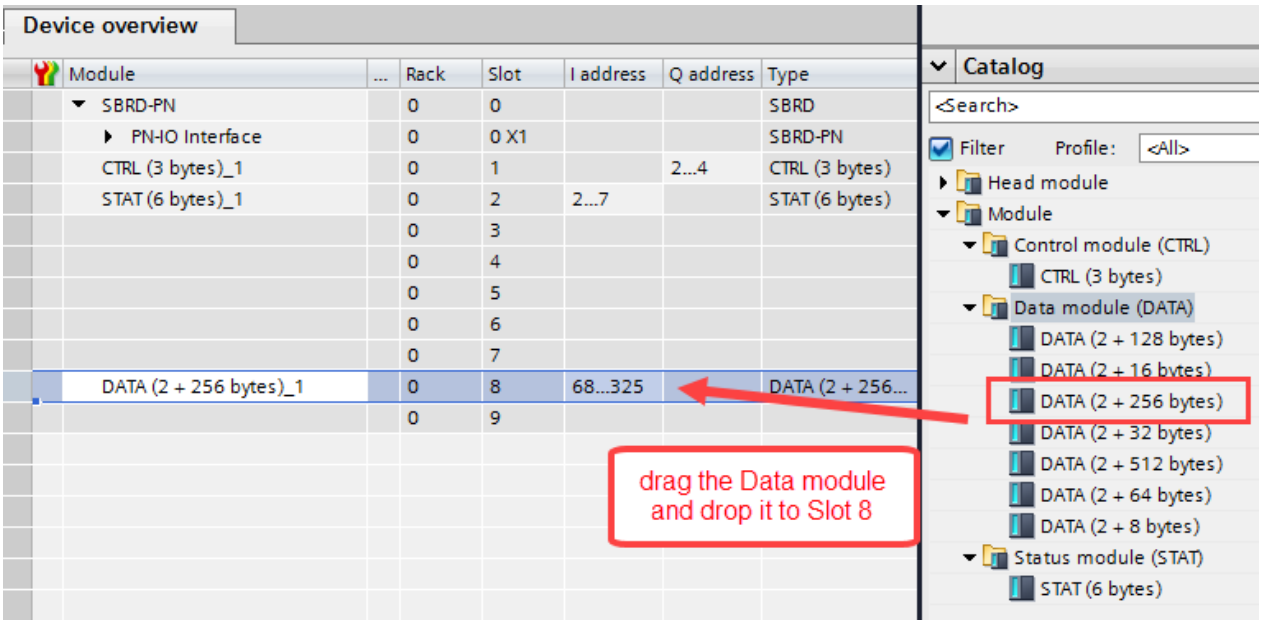




- Based on the amount of data in Profinet dataoutput, which is configured in CCS, the appropriate size of data module has to be assigned.
- The available options are :

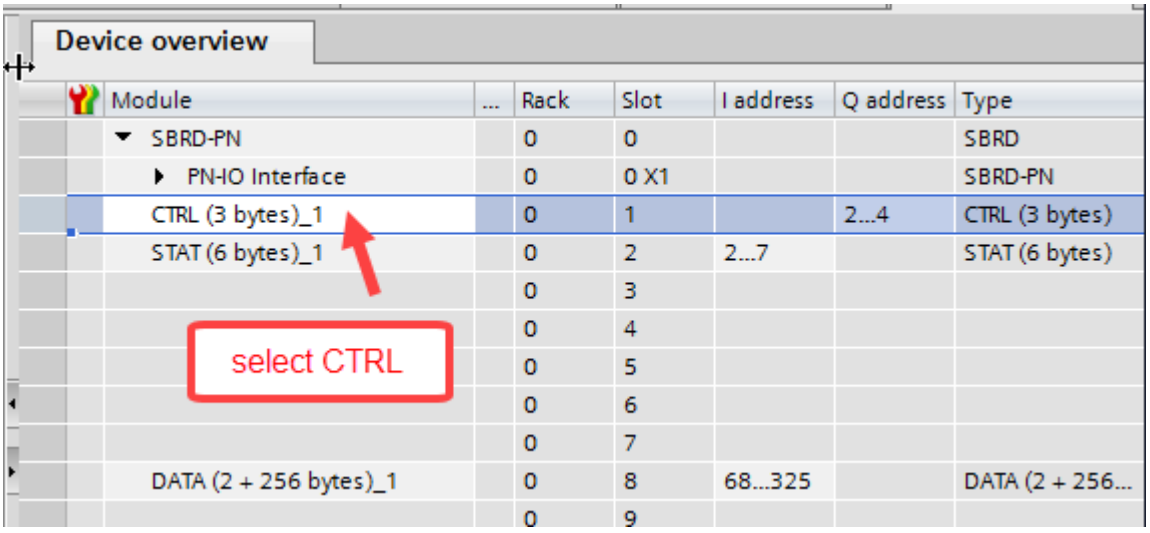


- Drag and drop the needed data module to the Device Overview as shown below.
  In our example we have considered DATA(2+ 256 bytes).

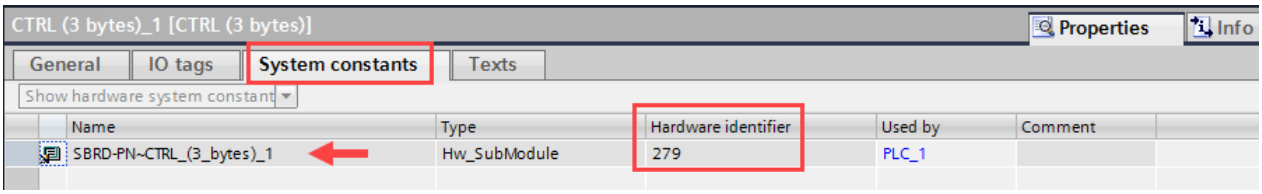


6.5 Identifying Hardware ID of the Sub modules of SBSI Vision Sensor.

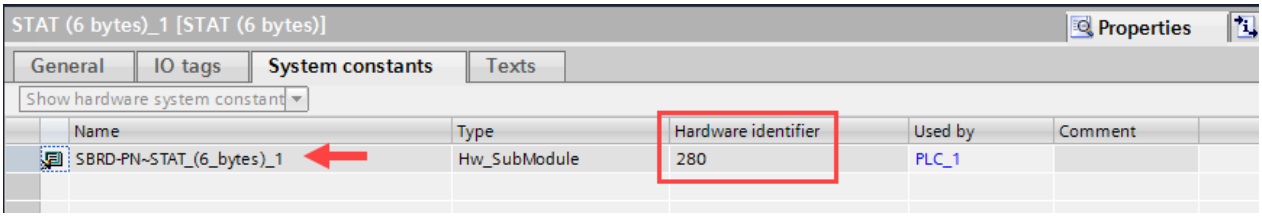
- In **device view** , select **CTRL module** as shown below.



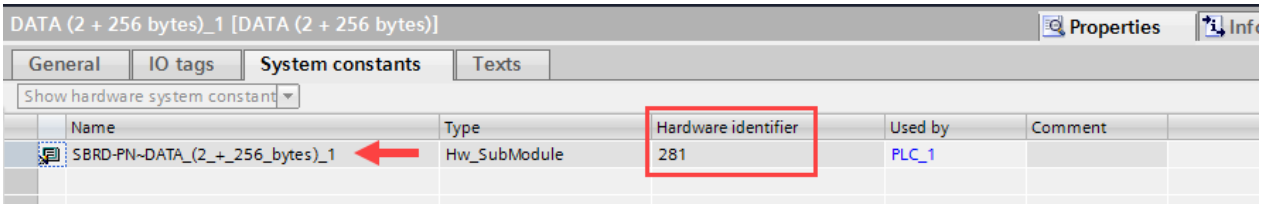
- Select **System Constants** as shown below from the Properties tab of **CTRL** module.



- In **device view** , select **STAT module** as shown below.



- In **device view** , select **DATA module** as shown below.

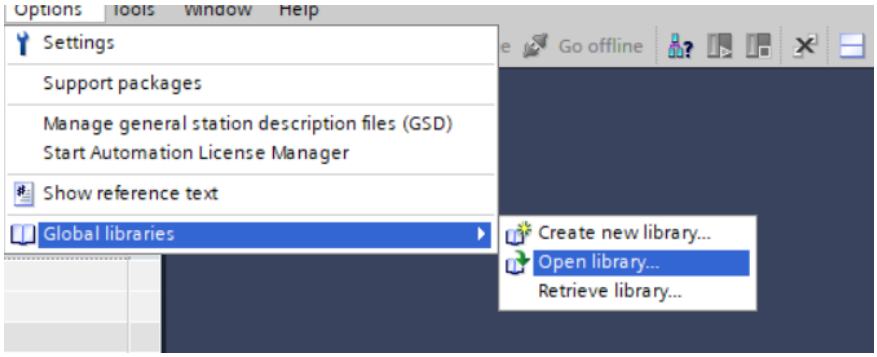


7 Linking VisionSensor library to the project.

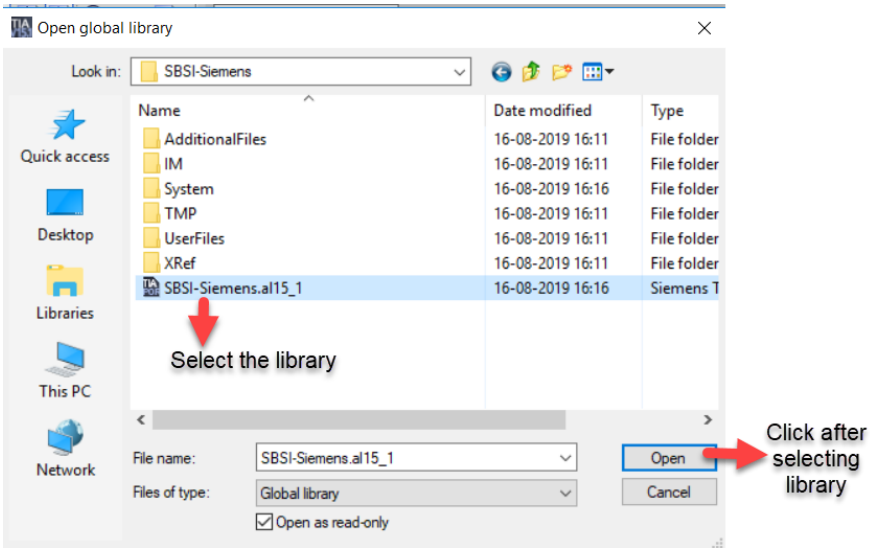
1. Unzip the library enclosed in the application note into a folder of your choice.

   IMPORTANT:
   The library was initially intended for the SBSI Vision Sensor from Festo. The functionally of the Profinet is fully identical. So in the following description the SBSI name is equal for the implementation for the SBRD Vision System!

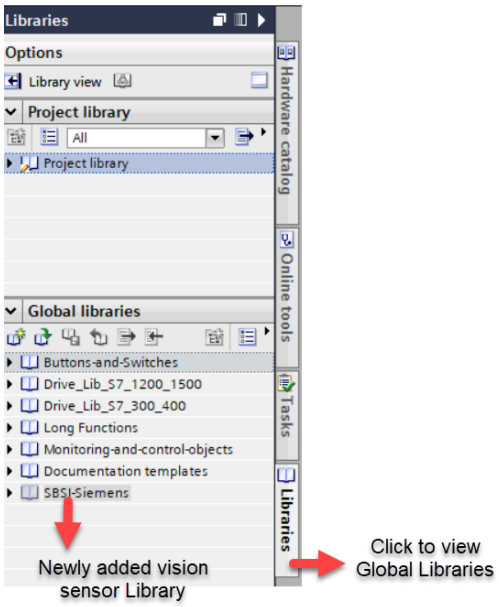
2. Click on **Options › Global Libraries › Open Library** as shown below**.**



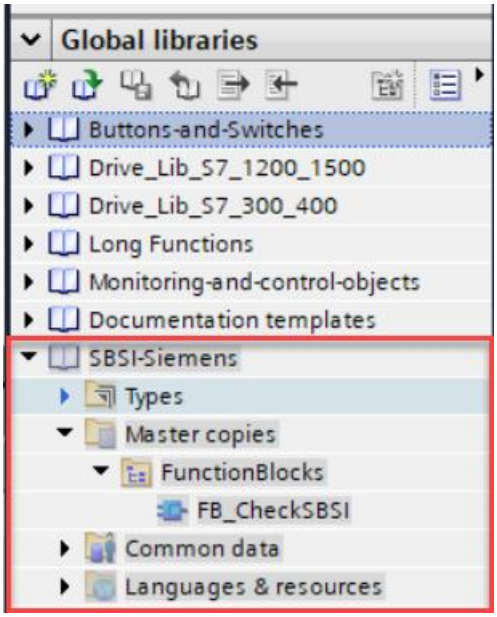
3. Choose the path where the Library SBSI Siemens is saved in your system. Select the library and click **OPEN.**



4. The opened library is now available in **“ Libraries › Global Libraries”.**



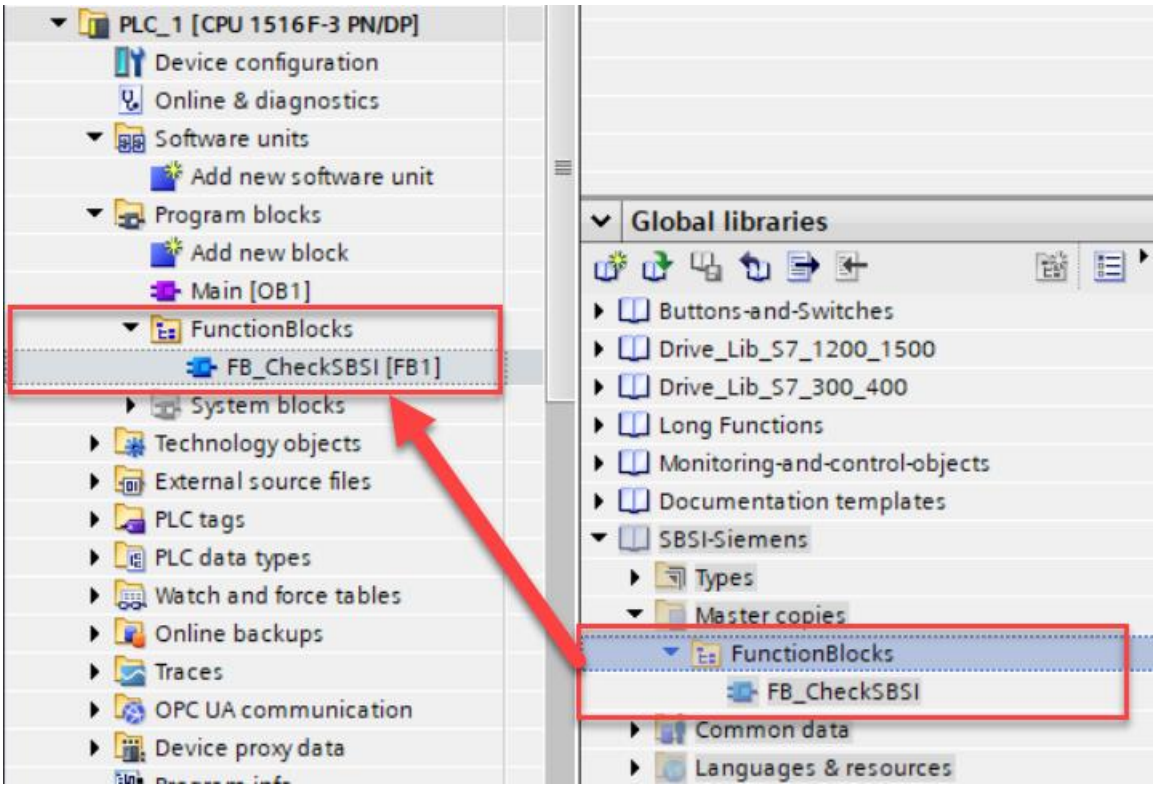
5. Open the **SBSI-Siemens** Library to view the content.



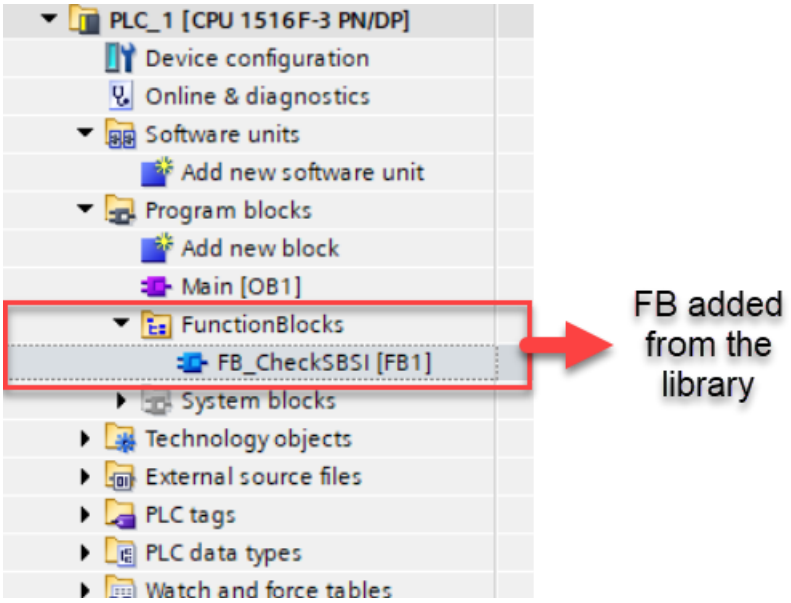
6. The Library has the following sub folders:

- Function Blocks : It has the Function Block named **FB_CheckSBSI** which is used to control the SBRD basic functions.

7. Drag and drop the folder **Function Blocks** to the folder **Program Blocks** in your project.
   Drag and drop the folder Function Blocks to the project as shown below.



8. Once the folders from the library has been copied to the project, the **Project Tree** will look as shown below.



8 DESCRIPTION OF SBSI VISION-Siemens Library FUNCTION BLOCKS

The library has 1 Function Blocks within it:

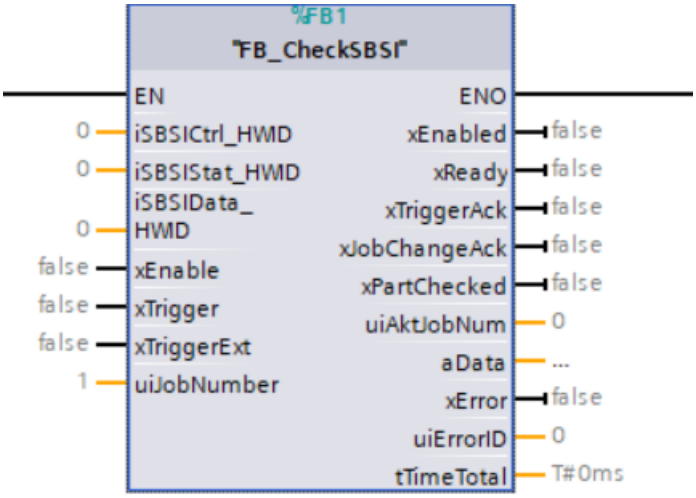
1. FB_CheckSBSI

8.1 FB_CheckSBSI

The Functions block has the following features:

- It allows the user to do Vision Sensor control operations. Using this Function Block the user can control the triggering of the SBRD.
- Using the Function Block the job number of the vision sensor can be changed.
- It gives the various status information of the SBRD operation like **Active job number**, **Error information** occurred during the operation, **Job change acknowledgement**, **Trigger acknowledgement**.
- It allows outputs the data output result data of the active job.
- **Attention**: The communication fail and the FB is blocked, when a job no. without Profinet setting is being selected!

The below image shows the schematic view of the FB_CheckSBSI block.



The following tables gives a detailed explanation of the inputs and outputs of the function block.

INPUT DATA

NAME	DATA TYPE	DESCRIPTION
iSBSICtrl_HWID	HW_ANY	Hardware ID of the **Ctrl** Module of SBRD.
iSBSIStat_HWID	HW_ANY	Hardware ID of the **Stat** Module of SBRD.
iSBSIData_HWID	HW_ANY	Hardware ID of the **Data** Module of SBRD.
xEnable	BOOL	**FALSE** – Disable the Function Block. **TRUE** – Enable the Function Block.
xTrigger	BOOL	**TRUE** – Rising Edge (Low ==› High) triggers the Vision Sensor.

NAME	DATA TYPE	DESCRIPTION
xTriggerExt	BOOL	**TRUE** – Hardware Trigger or free run enabled. **FALSE** – Hardware Trigger or free run disabled.
uiJobNumber	BOOL	Gives the active job number of the SBRD. If the value doesn't equal with the active job number ,then the request for job change will be triggered automatically.

Table 5.1: FB_CheckSBSI Input Data

OUTPUT DATA

NAME	DATA TYPE	DESCRIPTION
xEnabled	BOOL	**TRUE** – Function Block is enabled **FALSE**- Function Block is disabled
xReady	BOOL	**FALSE** – SBRD is not ready for next evaluation cycle. **TRUE** – SBRD is ready for next evaluation cycle.
xTriggerAck	BOOL	**FALSE** – No acknowledge for a successful trigger to SBRD. **TRUE** – Acknowledge for a successful trigger to SBRD.
xPartChecked	BOOL	**FALSE** – SBRD is busy in operation. **TRUE** – SBRD is waiting for next command.
aData	ARRAY[0..257] OF BYTES	1. **BYTE 0** – Gives the Image ID of the job being executed. Image ID is incremented with each job execution independent from trigger source. 2. **BYTE 1** – Bit0 of this byte has following values. • **Bit0** is **1** means Data Overrun = Data truncated. • **Bit0** is **0** means No Data overrun. 3. **BYTE 2 to BYTE 257** - Data as defined in CCS-Configuration Studio in I/O Configuration
xError	BOOL	**FALSE** – No Error during SBRD operation. **TRUE** – Error during SBRD operation.

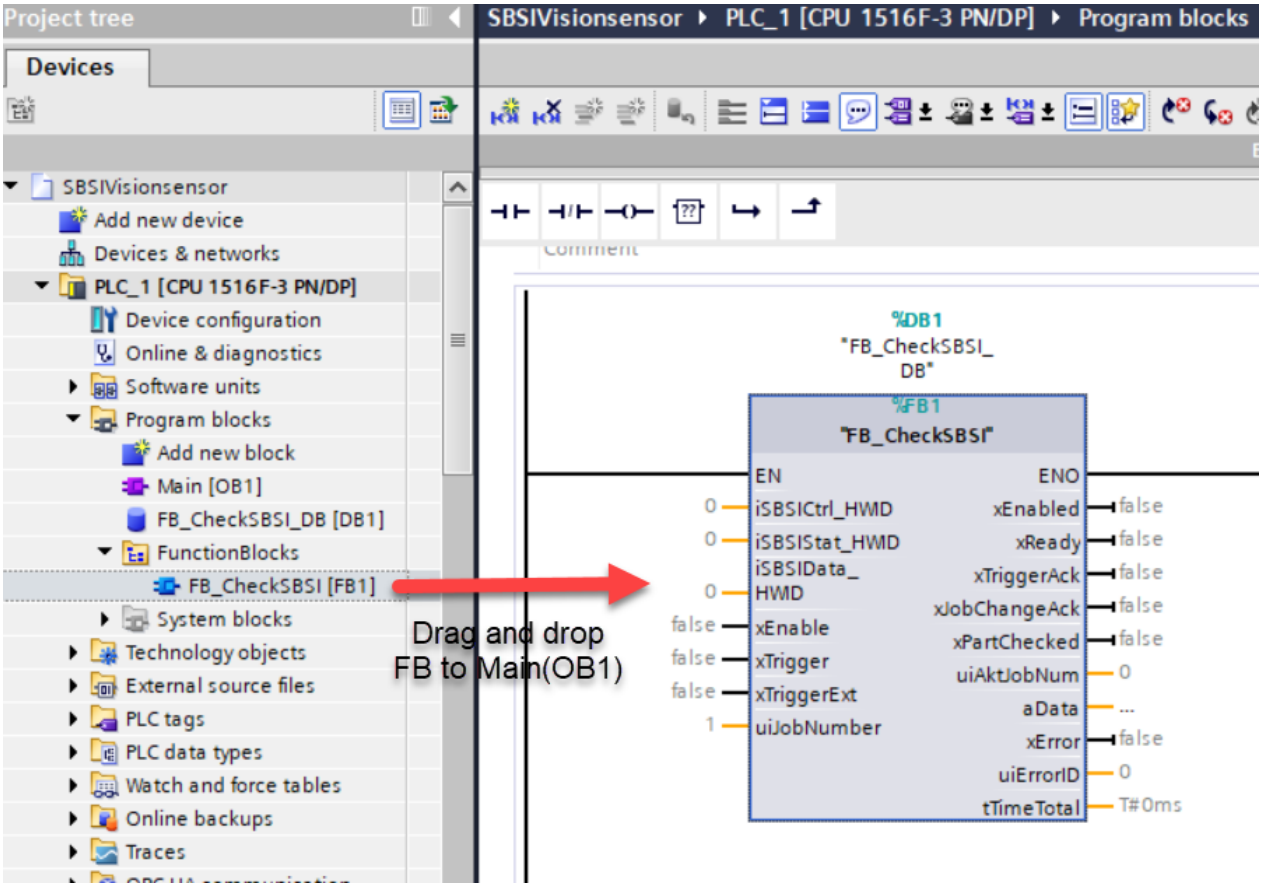
NAME	DATA TYPE	DESCRIPTION
uiErrorId	BOOL	Gives the ID of the error occured. It has following values. **0 –** No Error. **1 –** Failure Trigger request **2 –** Failure Change Job. **3 –** Failure switch to run. **5 –** Failure Profinet Not active in Job. **15 –** System Error
tTimeTotal	TIME	Time period for which the SBRD is busy.

Table 5.2: FB_CheckSBSI Output Data

9 INTEGRATION OF FUNCTION BLOCKS INTO PROJECT

9.1 FB_CheckSBSI

Drag and drop the FB_CheckSBSI function block to the TIA portal to the Main(OB1) as shown below.



NOTE

Refer Table 5.2 in Chapter 8.1 to get detailed description of the Inputs and Outputs to be configured for the Function Block.

The following are the inputs, the user has to configure properly :

1. **iSBSICtrl_HWID :** The hardware ID of the **Ctrl module** must be given to this input **.**

NOTE

- **Refer Chapter 6.5 to get detailed description on how to identify the Hardware ID of iSBSICtrl_HWID module in TIA Portal.**

2. **iSBSIStat_HWID :** The hardware ID of the **Stat module** must be given to this input **.**

NOTE

- **Refer Chapter 6.5 to get detailed description on how to identify the Hardware ID of iSBSIStat_HWID module in TIA Portal.**

3. **iSBSIData_HWID :** The hardware ID of the **Data module** must be given to this input **.**

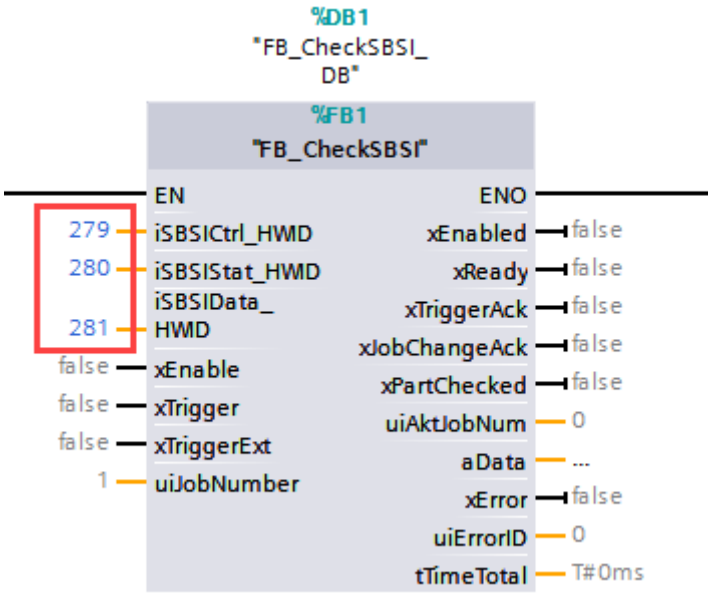
NOTE

- **Refer Chapter 6.5 to get detailed description on how to identify the Hardware ID of iSBSIData_HWID module in TIA Portal.**

4. **xEnable :**
   In order to make the Function Block operational make this input **TRUE.**
   In order to disable the operations of this Function Block make this input **FALSE.**

5. **uiJobNumber:**
   The job which the Vision Sensor has to execute has to be given in this input.
   If this value doesn't equal with the Active job number of the vision sensor then the request for changing job number will be triggered automatically.

6. After the Hardware ID's of the modules are added the Function Block overview will appear as shown below.



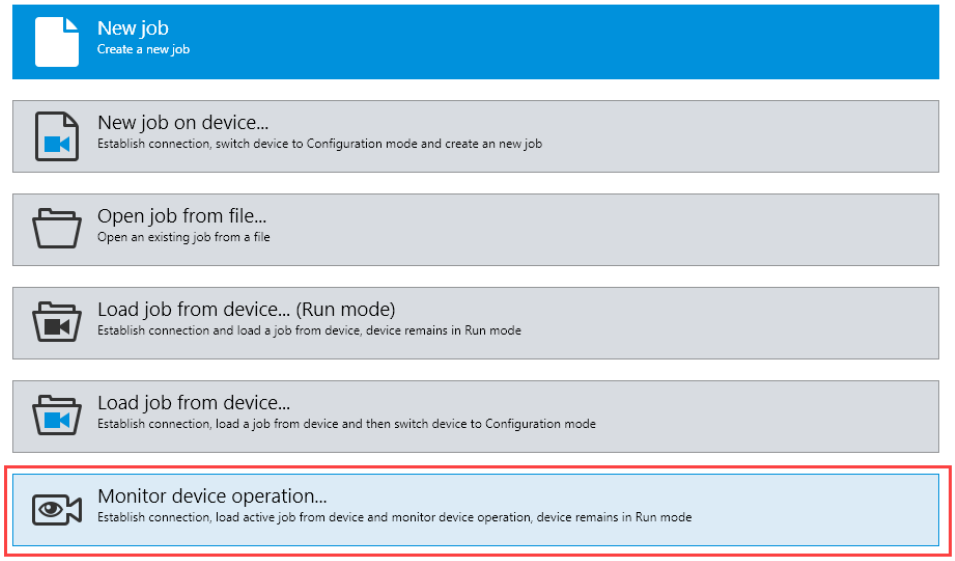
10 FUNCTION BLOCK EXECUTION WITH AN EXAMPLE

This chapter will explain the following in detail:

1. To demonstrate the FB execution with an example where we have to detect a position and a datamatrix code using a SBRD Vision System. As configured in chapter 3.
2. Prepare the SBRD Vision System in to monitor the results in CCS Configuration software.
3. Creating watch tables in TIA portal to monitor the following :
   - Data sent from the SBRD in the telegram.
   - Status of the SBRD.
4. Mapping the result data configured in vision configuration studio with the array of data obtained as an output of the Function Block **FB_CheckSBSI**.

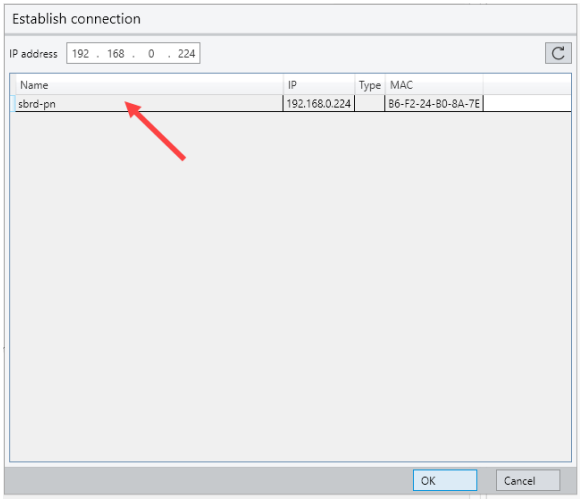
10.1 Prepare the SBRD Vision System to monitor the results in CCS-software..

1. Open CCS software in Monitor device operation ...



2. Select the connected device:

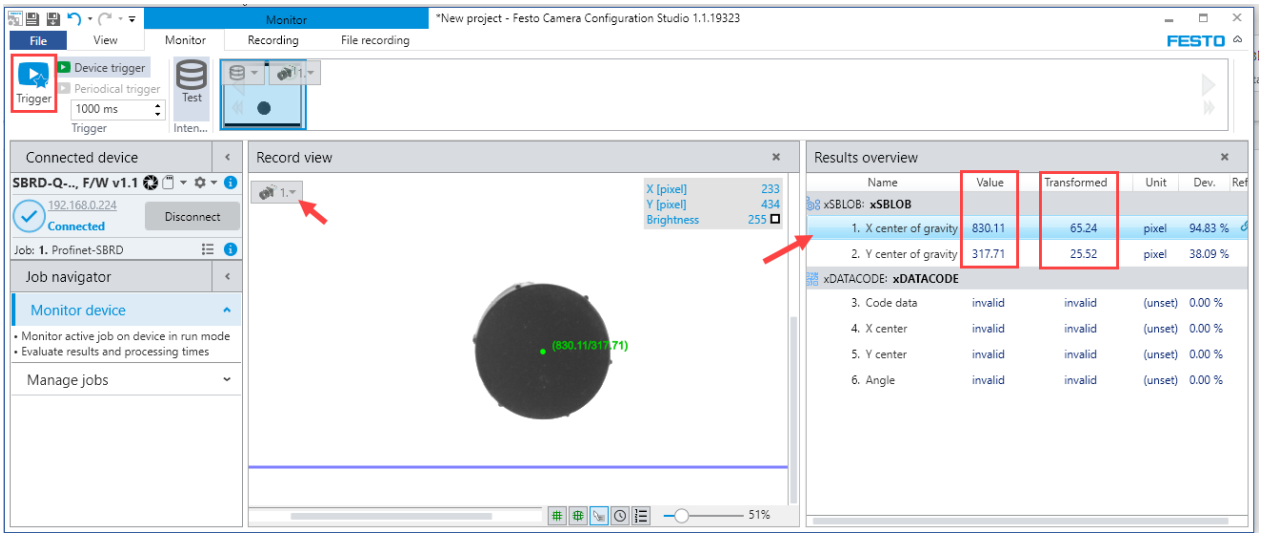
› O.K



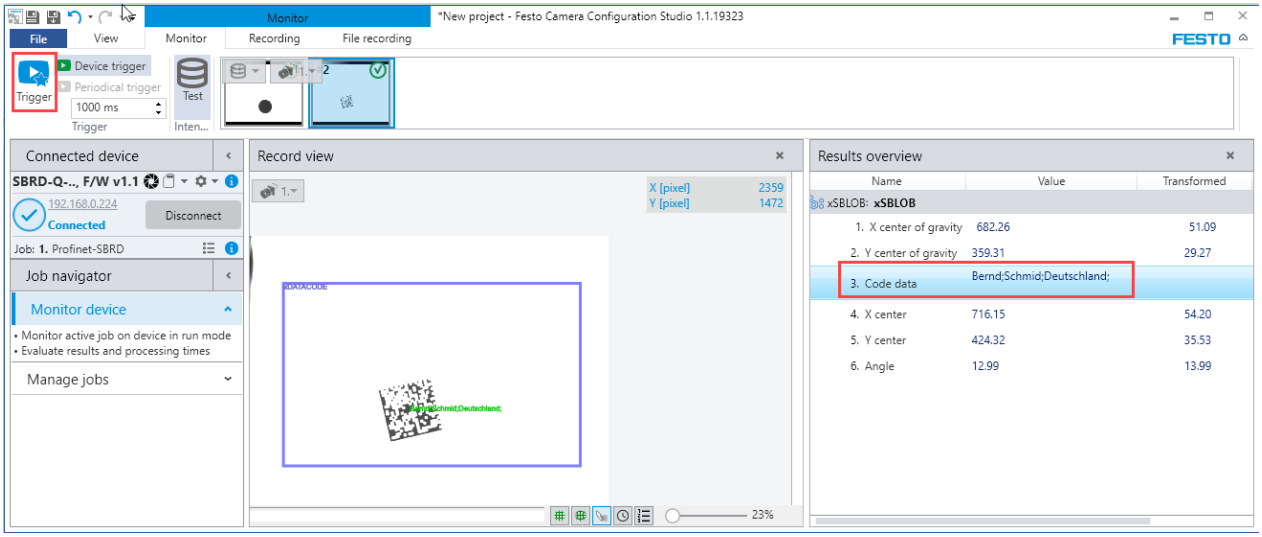
3. Image Trigger

- Click on Trigger button
- Select the right camera e.g 1. In Record view
- Select the xSBlob tool in Results overview

Then the display show the image with the results. There were 2 result format available. The Value which is in pixel and the Transformed, which is in mm.
These results will be compared with the data on the Siemens controller in the following description.

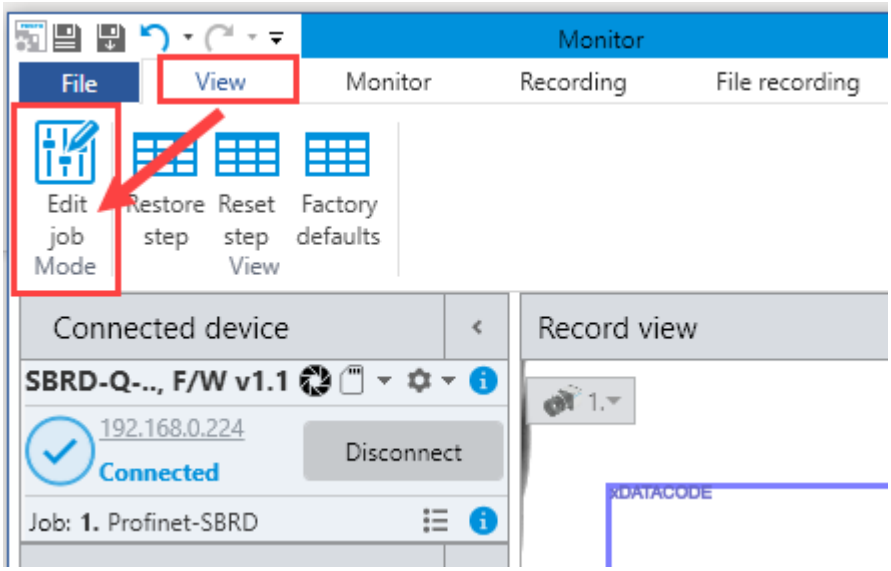


Place a sample of a data matrix code into the viewing area. Do again a trigger and check the results.



If this tests were successful, keep the SBRD and the CCS-software in this status!

If the results were not o.k, then the configuration has to be improved and adjusted.
Then change to Edit job Mode:

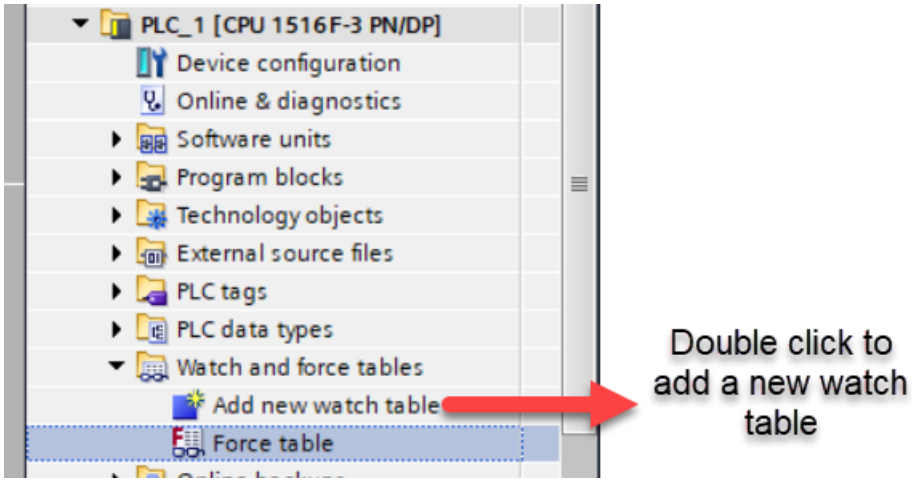


Then the settings of the camera and the inspection could be configured, as shown in chapter 3.

10.2 Example description in TIA Portal

1. Integrate the Function Block **FB_CheckSBSI** as explained in **Chapter 9.**

2. Create a watch table.

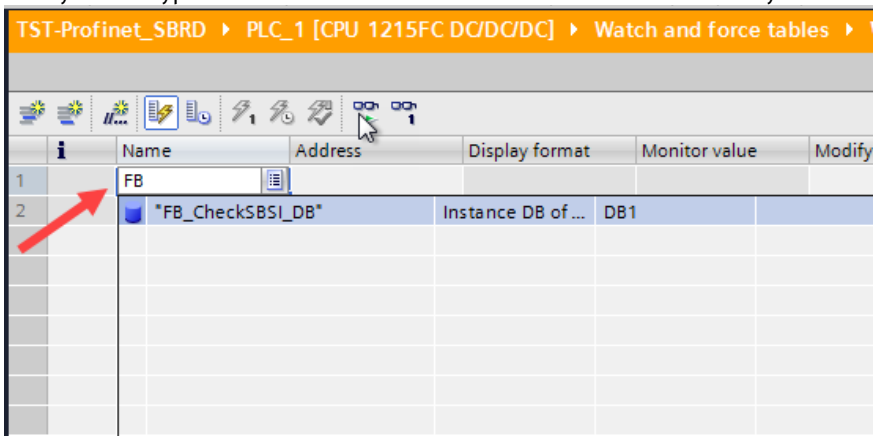


3. Give a name to the watch table. Here in our example we have named the watch table as SBRD-Dataoutput.

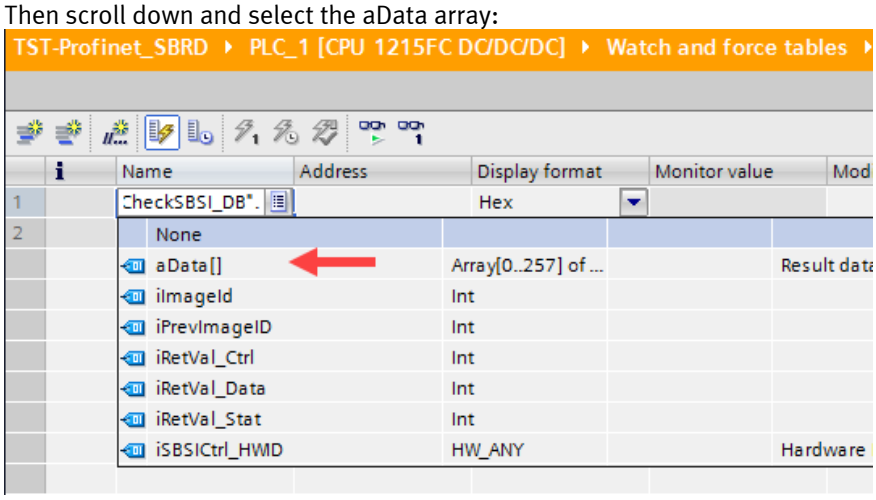
4. Add the **aData** output of the **FBCheckSBSI.** Add all the 258 array elements of aData to the watch table for testing purpose.

5. Add the first element of the array aData to the watch table as shown below.

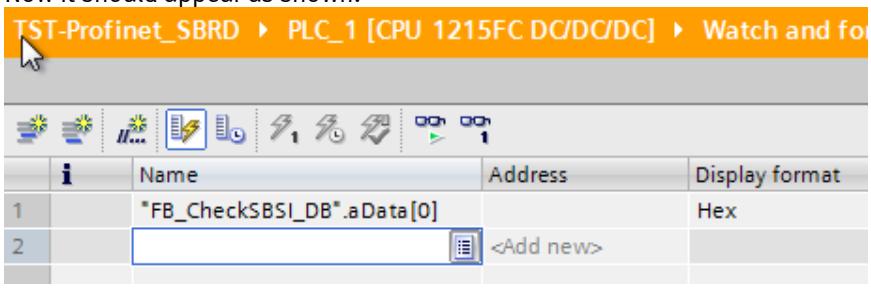
First you can type the name of the FB and then select the shown entry



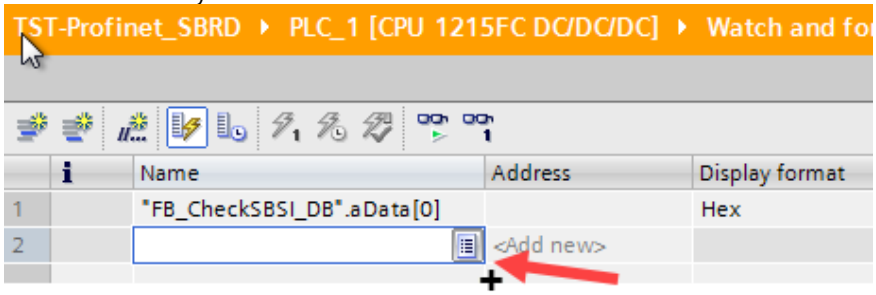
Then scroll down and select the aData array:



Now it should appear as shown:



6. Click the corner of the 1st element and when a **"+"** mark appears drag the cursor down until all 256 elements of the array are added to the watch table.



7. Once the array elements are added the watch table looks as shown below.

TST-Profinet_SBRD ▸ PLC_1 [CPU 1215FC DC/DC/DC] ▸ Watch and force tables ▸ Watch table_1

	i	Name	Address	Display format	Monitor val...	Modify va...		Comment
1		"FB_CheckSBSI_DB".aData[0]		Hex				Image ID
2		"FB_CheckSBSI_DB".aData[1]		Hex				Reserved
3		"FB_CheckSBSI_DB".aData[2]		Hex				
4		"FB_CheckSBSI_DB".aData[3]		Hex				
5		"FB_CheckSBSI_DB".aData[4]		Hex				
6		"FB_CheckSBSI_DB".aData[5]		Hex				
7		"FB_CheckSBSI_DB".aData[6]		Hex				
8		"FB_CheckSBSI_DB".aData[7]		Hex				
9		"FB_CheckSBSI_DB".aData[8]		Hex				
10		"FB_CheckSBSI_DB".aData[9]		Hex				
11		"FB_CheckSBSI_DB".aData[10]		Hex				
12		"FB_CheckSBSI_DB".aData[11]		Hex				
13		"FB_CheckSBSI_DB".aData[12]		Hex				
14		"FB_CheckSBSI_DB".aData[13]		Hex				

8. The first 2 data in the payload are reserved for internal data. (see manual)
So configure the display format of Byte2 to Byte (x) as **Hex** for all values as configured in the CCS dataoutput for Profinet. For all String data as configured in the CCS dataoutput set the display format to **Char**.

TST-Profinet_SBRD ▸ PLC_1 [CPU 1215FC DC/DC/DC] ▸ Watch and force tables ▸ Watch table_1

	i	Name	Address	Display format	Monitor val...	Modify va...		Comment
1		"FB_CheckSBSI_DB".aData[0]		Hex				Image ID
2		"FB_CheckSBSI_DB".aData[1]		Hex				Reserved
3		"FB_CheckSBSI_DB".aData[2]		Hex				x Coordinate
4		"FB_CheckSBSI_DB".aData[3]		Hex				x Coordinate
5		"FB_CheckSBSI_DB".aData[4]		Hex				x Coordinate
6		"FB_CheckSBSI_DB".aData[5]		Hex				x Coordinate
7		"FB_CheckSBSI_DB".aData[6]		Hex				y Coordinate
8		"FB_CheckSBSI_DB".aData[7]		Hex				y Coordinate
9		"FB_CheckSBSI_DB".aData[8]		Hex				y Coordinate
10		"FB_CheckSBSI_DB".aData[9]		Hex				y Coordinate
11		"FB_CheckSBSI_DB".aData[10]		Character				Begin Data matrix Code
12		"FB_CheckSBSI_DB".aData[11]		Character				
13		"FB_CheckSBSI_DB".aData[12]		Character				
14		"FB_CheckSBSI_DB".aData[13]		Character				
15		"FB_CheckSBSI_DB".aData[14]		Character				
16		"FB_CheckSBSI_DB".aData[15]		Character				

Information about the memory allocation for the dataoutput gives the CCS. (I/O Configuration)

Connected device

[Connection required]

192.168.0.224

Not connected

Connect

Device mode:

[Connection required]

Configure

Job: [Connection required]

Job navigator

1. Set up
2. Configure cameras
3. Acquire records
4. Prepare inspection
5. Inspect objects
6. Configure I/O

I/O settings | I/O results

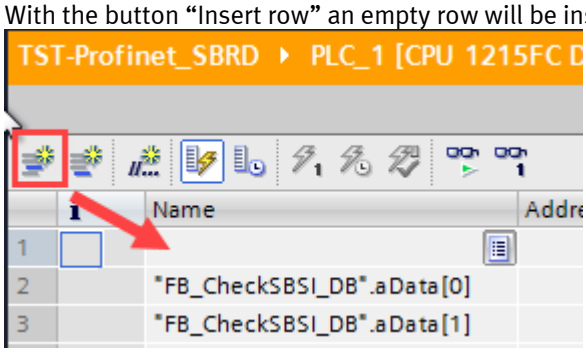
Data sections:

Name	Type	Position [B	Data (hex)	Data
Image ID	byte	0	04	4
Result status (:1 data overrun)	byte	1	01	1
X center of gravity	float32	2..5	D4 5C 61 42	56.3407
Y center of gravity	float32	6..9	EA CF F8 41	31.1015
Code data	string	10..73	42 65 72 6E 64 3B 53 63 68 6D 69	Bernd;Schmid;Deutschland;

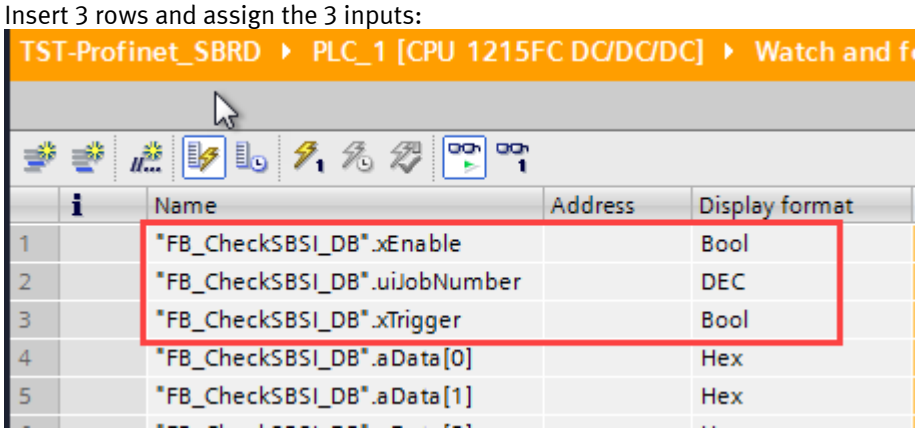
Data:

```
04 01 D4 5C 61 42 EA CF F8 41 42 65 72 6E 64 3B 53 63 68 6D 69 64 3B 44 65 75 74 73 63 68 6C 61 6E 64 3B
0A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00
```

9. For test purpose the function block could be controlled by the watch table. The inputs XEnable, uiJob-Number and xTrigger have to be assigned to the watch table.
With the button “Insert row” an empty row will be inserted:



Insert 3 rows and assign the 3 inputs:



Test:

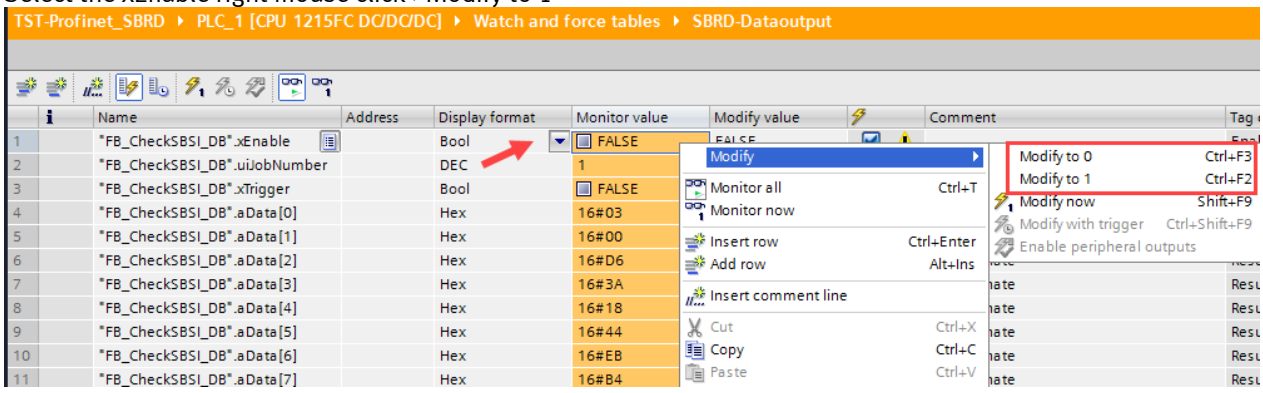
1. Download the configuration to the PLC and set to Run mode.

   Step in the watch table and activate: Monitor all. 

   SBRD in Monitor mode and inspection part is in the field of view.

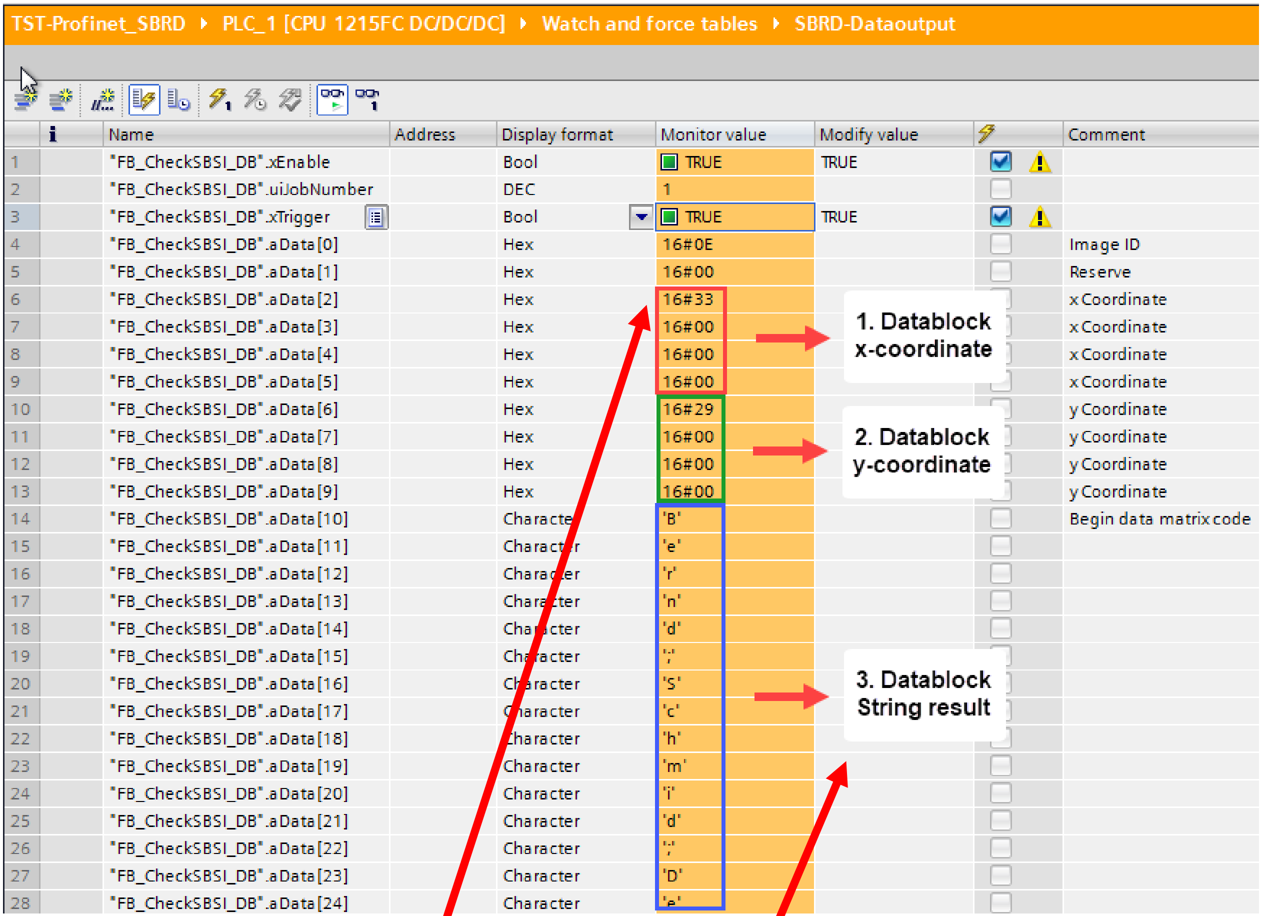
2. Enable the Function Block. This is done by making **xEnable = TRUE**.

   Select the xEnable right mouse click › Modify to 1

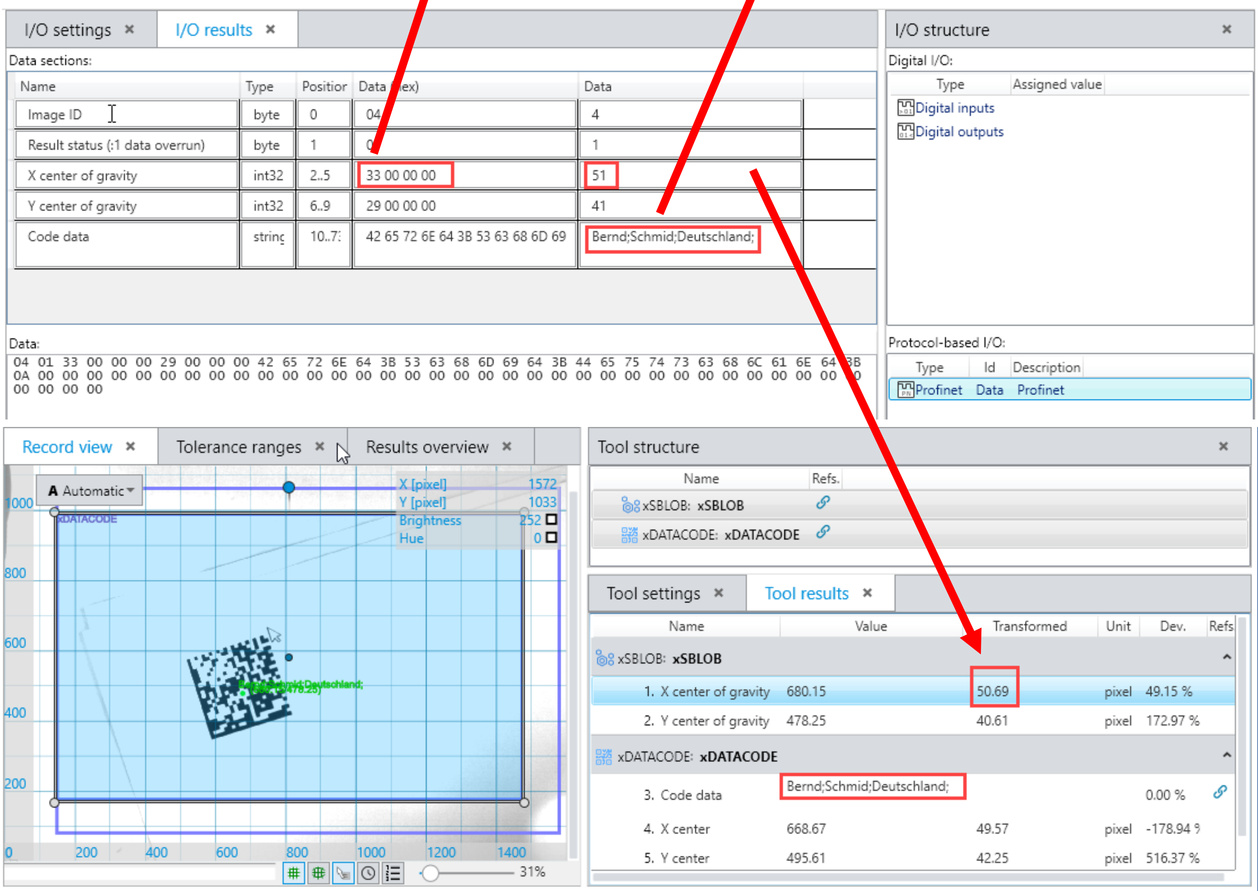


3. Toggle input **xTrigger = 1.** In the same way as xEnable

4. Observe the data received in the aData output of the FB.



- The online payload data values of TIA portal match with the Results of the CCS data output I/O results:

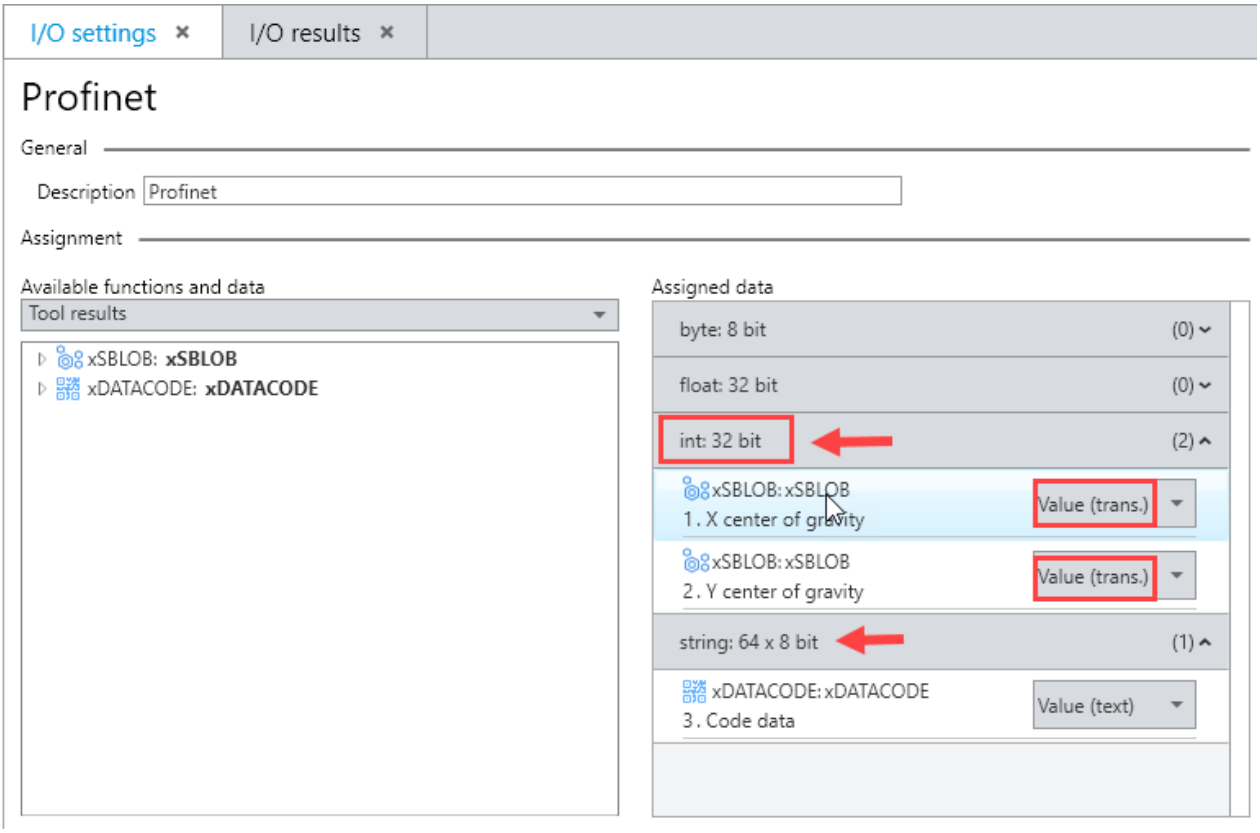


The data 33h equals to 51 dec which is rounded due to int.

10.3 Payload Data Mapping between TIA Portal and Camera Configuration Studio.

This chapter explains how the dataoutput configured in CCS software and the data appearing in TIA portal (**aData output of the FB**)
The topic is that Festo device send the data in different Byte order than the Siemens work.

1. Consider in the example above we have configured the result value as 32 bit int. data format for the coordinates and string for the code.



With the “small” value (trans) it was easy to compare the data. If the numbers were bigger we have to care for the Byte order! Siemens work on different Byte oder (Motorola) than our Festo Devices! (Intel)

In next example we configure the value in pixel.

The data block in Tia show:

"FB_CheckSBSI_DB".aData[2]	Hex	16#A4		x Coordinate
"FB_CheckSBSI_DB".aData[3]	Hex	16#02		x Coordinate
"FB_CheckSBSI_DB".aData[4]	Hex	16#00		x Coordinate
"FB_CheckSBSI_DB".aData[5]	Hex	16#00		x Coordinate

The Question is what is the correct Byte order?

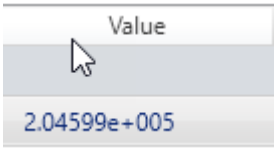
A4 02 **or** 02 A4

02 A4 is the correct answer = 676 › So the lowest Byte appear first!

Other example:

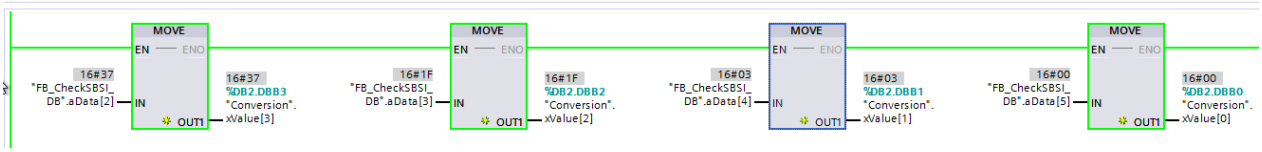
"FB_CheckSBSI_DB".aData[2]		Hex	16#37		x Coordinate
"FB_CheckSBSI_DB".aData[3]		Hex	16#1F		x Coordinate
"FB_CheckSBSI_DB".aData[4]		Hex	16#03		x Coordinate
"FB_CheckSBSI_DB".aData[5]		Hex	16#00		x Coordinate

37 1F 03 00 › 00 03 1F 37 = 204599 in CCS ›

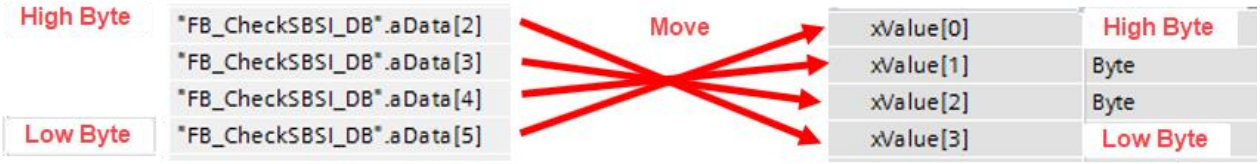


2. Conversion of the Byte order

To see the values in the correct form, the sequence of the Bytes within one data block has to be changed. There may different way to do this. Here we show very simple with some MOVE operation.

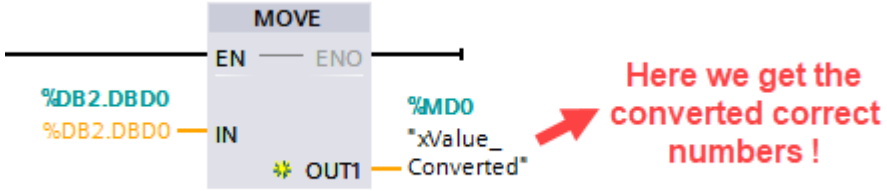


he idea is to move the single Byte from the a.Data to an Datablock array with 4 Bytes. The first Byte will be moved to the highest array position. The second to the second highest and so on...
Then the Datablock could be interpreted as an double word number with the correct value!



This is very important especially when the data are transferred from the camera in Float 32 (REAL). Then the interpretation of the Hex numbers, and verify with the Decimal display in the CCS software, are not easy!
Float 32 is the common data format for the values of coordinates, because of the possibility of comma values!

At last the array from the data block (DB02) move into a “Memory Double Word” MD0 which could be interpreted as a Real.



The End