Application Note

FESTO

SBSI Vision Sensor integration with S7-1500 PLC USING TIA PORTAL 15

This Application Note helps the user to integrate SBSI Vision Sensor 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 SBSI Vision Sensor.

Type(s)

Title ..... SBSI Vision Sensor 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

Table of contents

1 Components/Software used

Type/Name	Version Software/Firmware
GSDML File for SBSI Vision Sensor	V1_17_129
Siemens TIA Portal	V 15.1
Vision Sensor Configuration Studio	V 123.2.2
SBS Sensor Type description	

Table 1.1: Components/Software used

2 APPLICATION DESCRIPTION

This document explain how you will integrate the host function blocks of SBS Vision Sensor into SIEMENS TIA Portal V15.

The supported systems are:

- S71500
- S71200

Supported Field Bus :

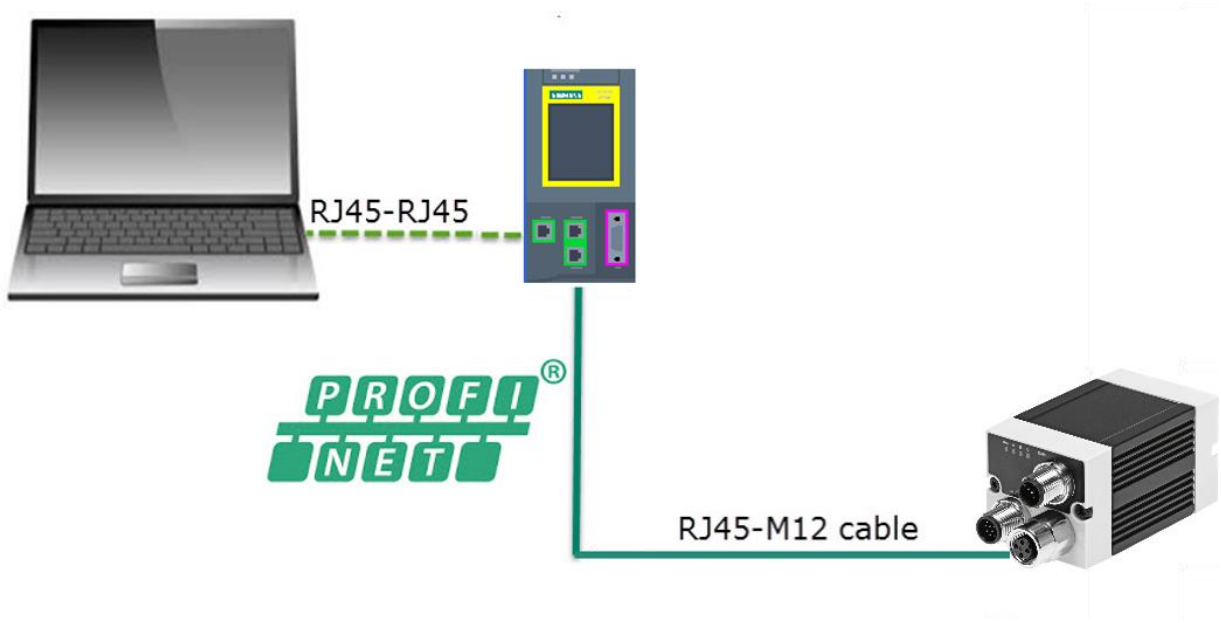
- Profinet IO

The application note has the description for the following:

- SBS vision sensor configuration in vision sensor configuration studio.
- Configuring the needed payload data of the vision sensor in vision configuration studio.
- SBS Vision sensor Setup in SIEMENS TIA Portal.
- Installing the GSDML File for SBS Vision Sensor.
- Adding the SBS Vision sensor to Devices and Networks in TIA Portal.
- Description of the function blocks of SBS Vision sensor.
- Integrating the Function Blocks within the programming environment of TIA Portal.

2.1 Topology

When we want to communicate between S7-1500/1200 PLC and SBS Vision sensor the following topology must be followed.



➔ **NOTE**

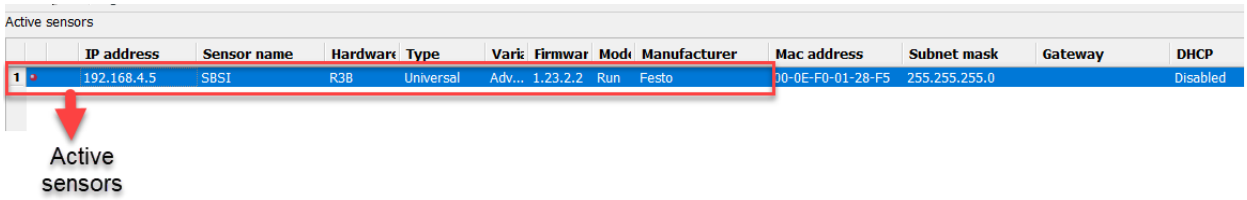
- **The IP address of SBS Vision Sensor , Ethernet Port of S7-1500/1200 and the PC used for programming the PLC must be in the same IP range.**

3 Configuration of SBS Vision Sensor via Vision Sensor Configuration Studio for the use with PROFINET.

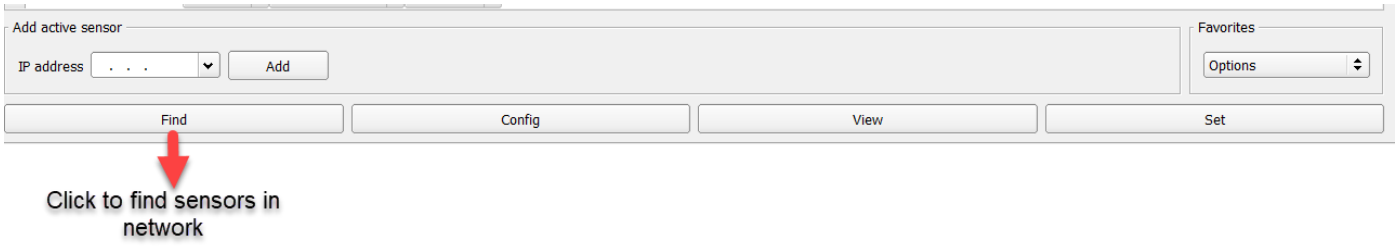
3.1 Settings in Vision Sensor Device Manager.

- Open Vision Sensor Configuration Studio .
- The following screen will be displayed in Vision Sensor device manager.

- At the start of Vision Sensor Device Manager SBSI sensor is listed in the window " **Active sensors**".



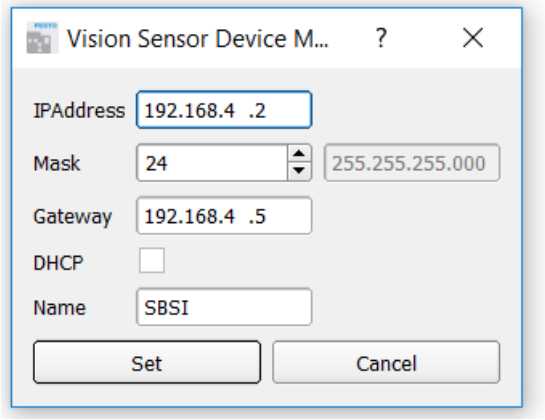
- If it is not listed the sensor which is connected in the network can be found out by clicking the button " **Find**" as shown below.



3.2 Setting of IP address and Name of SBS Vision Sensor.

- Select the desired sensor and Click on **SET** the to get the network settings tab as shown below.

- The network settings tab will be as shown below.



NOTE

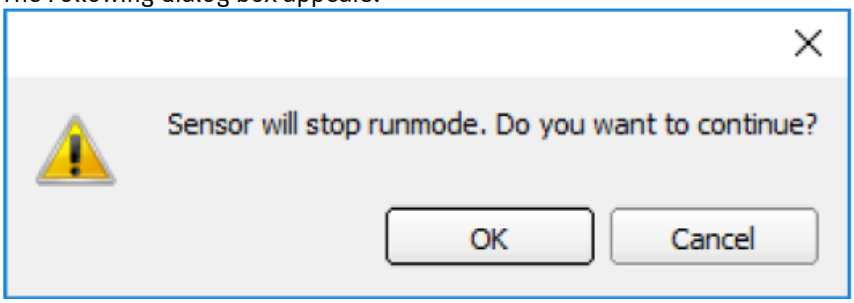
- **These settings are not active before a reboot of the sensor.**

Important Conditions for properly working PROFINET communication

1. The SBS name must be identical in PLC and sensor.
2. The IP address of SBS and PLC must correspond (must be in same address range).
3. IP address and name can be set in different ways:
   - **Via SBS software** (Vision Sensor Device Manager)
   - **Via PLC interface** (TIA portal). Refer **Chapter 6** to get detailed description on how to set SBS IP address and device name from TIA portal.

3.3 Open Vision Sensor Configuration Studio.

- Select the desired SBS Vision sensor. Click on **Config**.
- The Following dialog box appears.



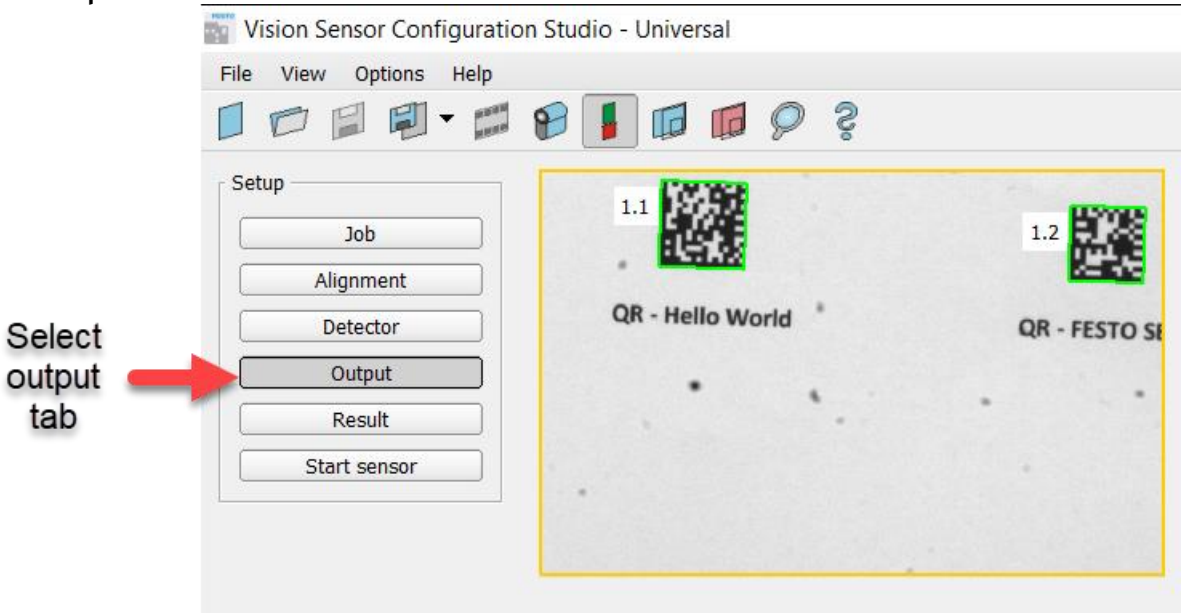
- Confirm the dialog box by clicking **OK** to stop the Vision Sensor Device Manager and to start the configuration in Vision Sensor Configuration Studio.
- The following display can be seen.

NOTE

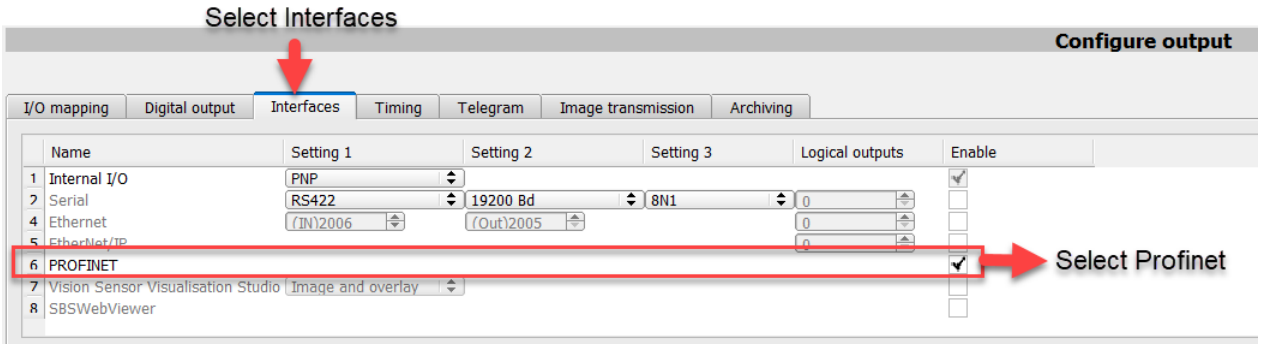
- **In this chapter only settings needed for Profinet communication is explained. The step by step procedure to configure the vision sensor with an example is given in Chapter 10.**

3.4 Select interface “PROFINET”.

- Go to **Output** tab as shown below.

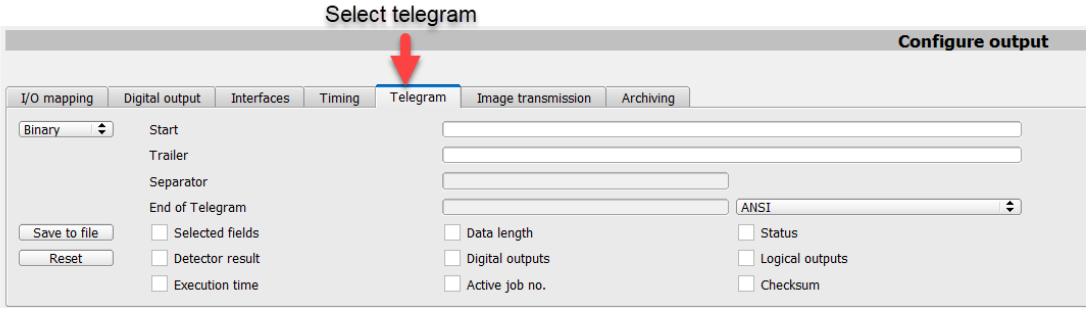


- Go to **Interface ››› PROFINET** as shown below to select Profinet interface.

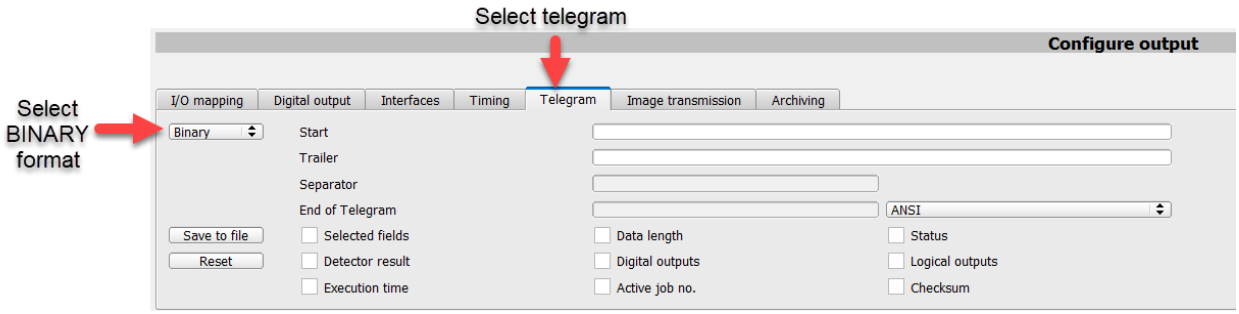


3.5 Definition of the telegram.

- Go to **Output ›››› Telegram** to define the data which has to be transferred via PROFINET interface.



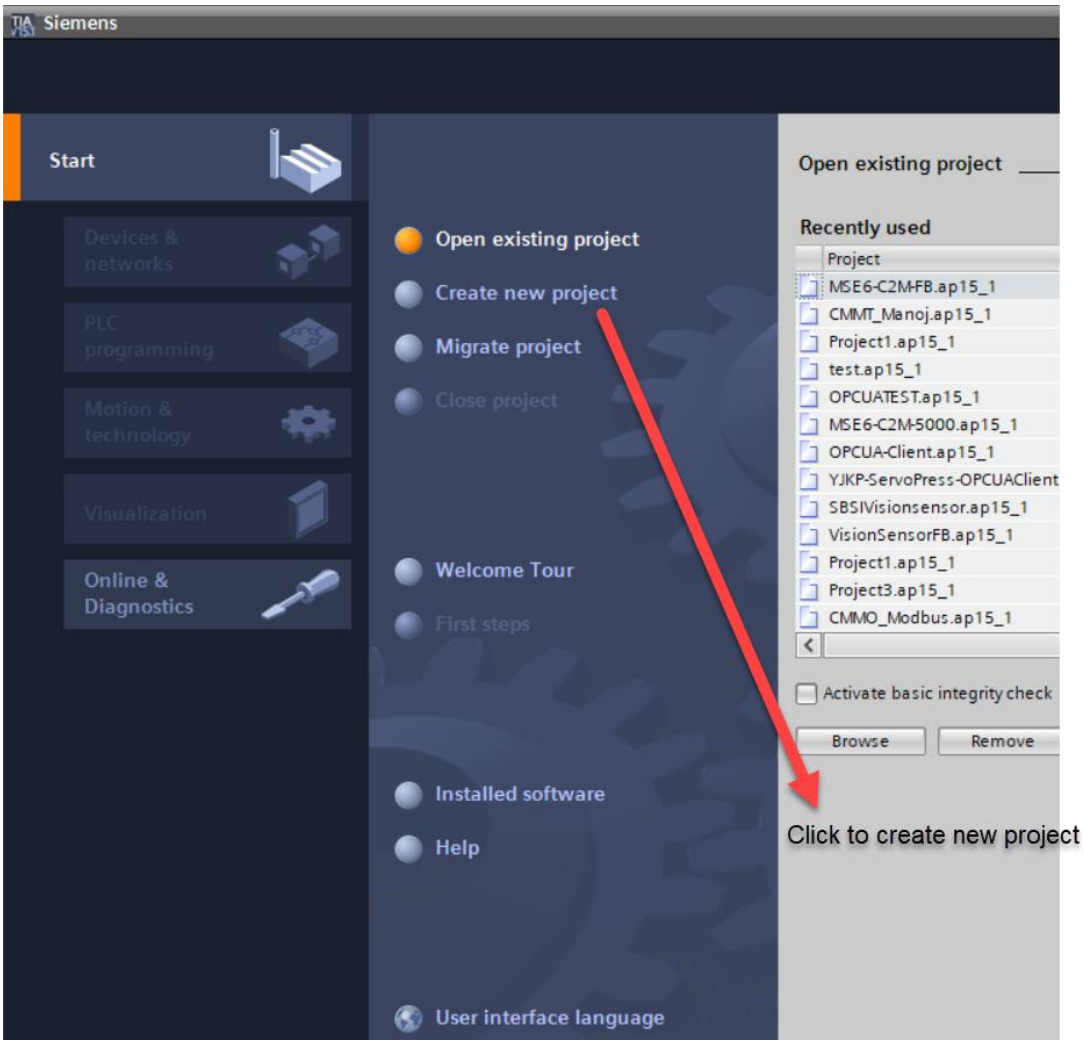
- For use with PROFINET interface the telegrams should be defined with **BINARY** format.



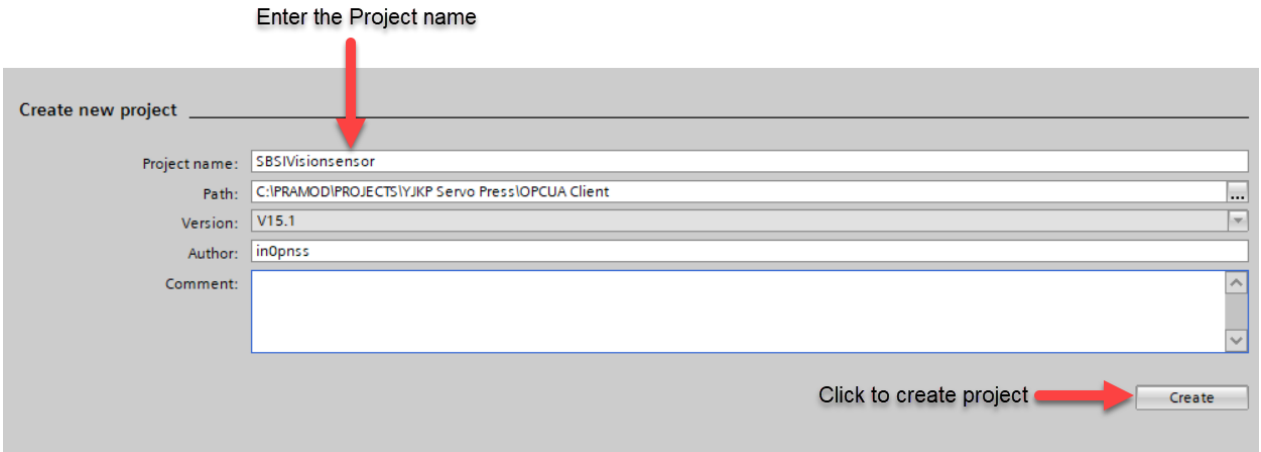
4 S7-1500 PLC CONFIGURATION IN TIA PORTAL

4.1 Creating a new project in TIA Portal

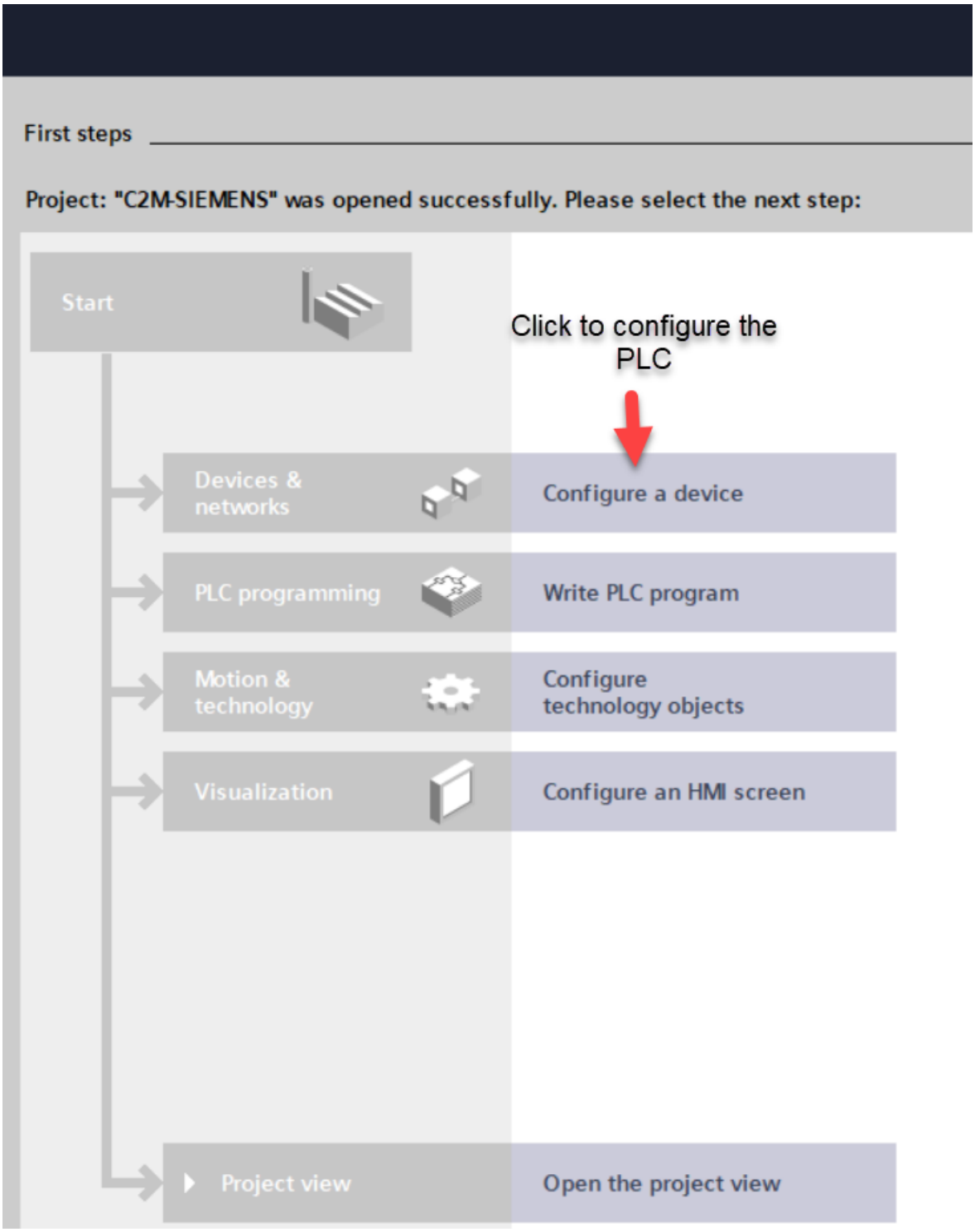
1. Start the TIA Portal V15.1 software.
2. Double click on **Create New Project** to create a new project.



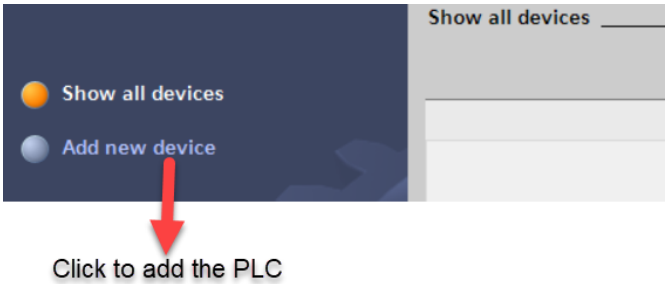
3. Enter the Project name and select the path where the project must be saved in your system. Then click on **Create** to create the project.



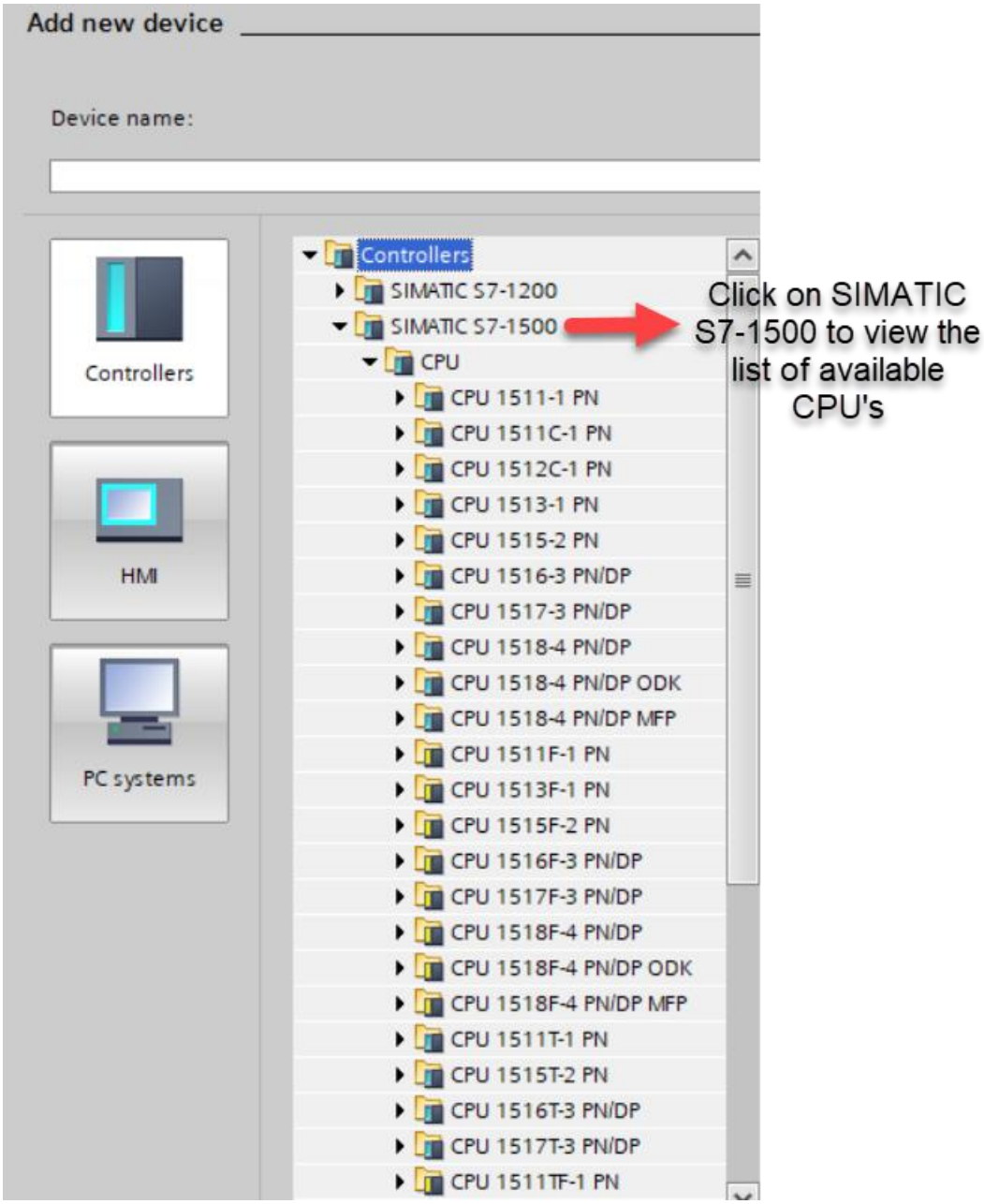
4. Double Click on **Configure a device** to configure the PLC needed.



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.

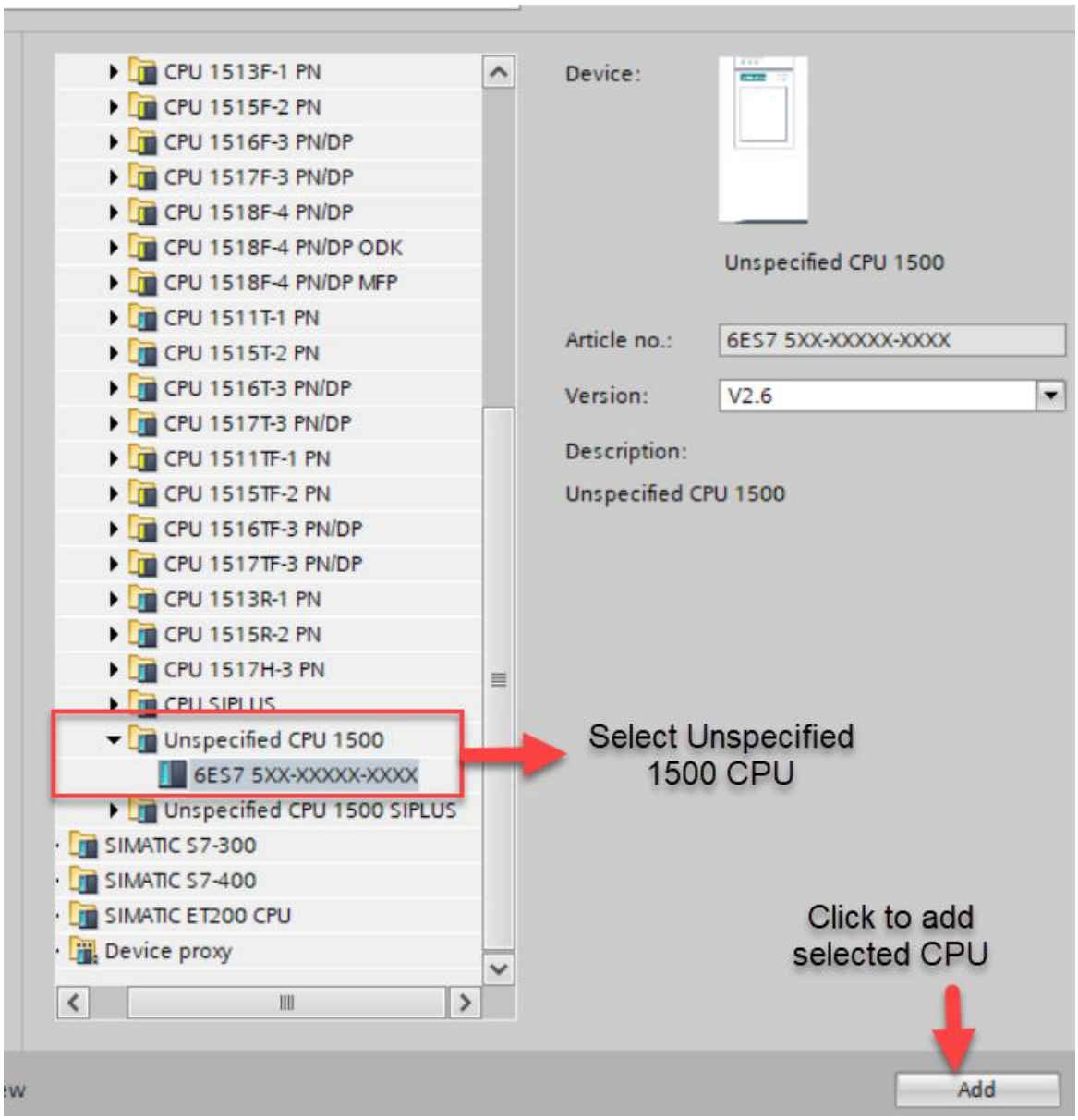


7. Click on **Unspecified CPU S7-1500** and select the PLC below it and click on **Add Device**.

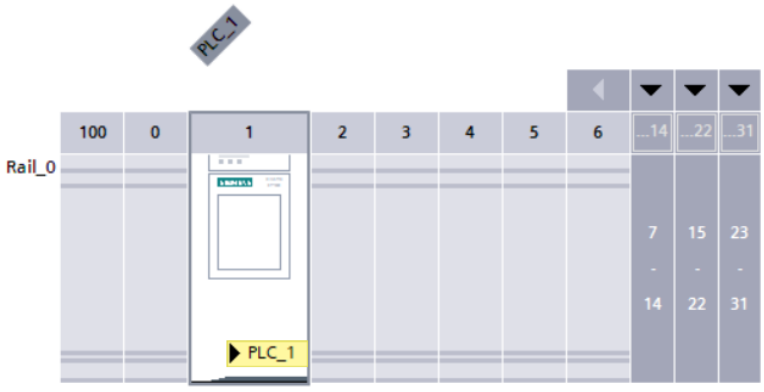
NOTE

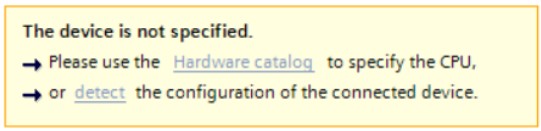
- If we select Unspecified CPU S7-1500, then the actual hardware configurations of the PLC can be read by using the detect hardware configurations option.

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



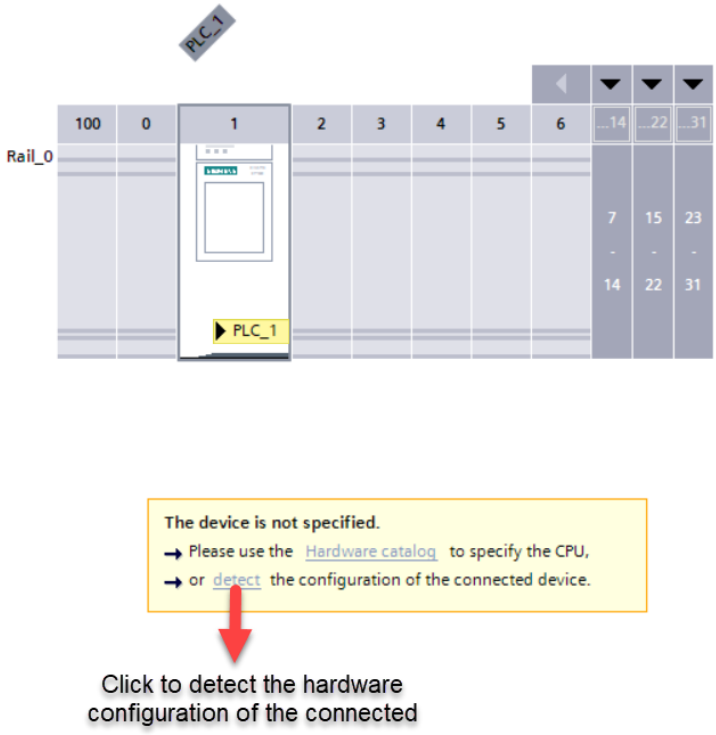
8. The project view will be as shown below.



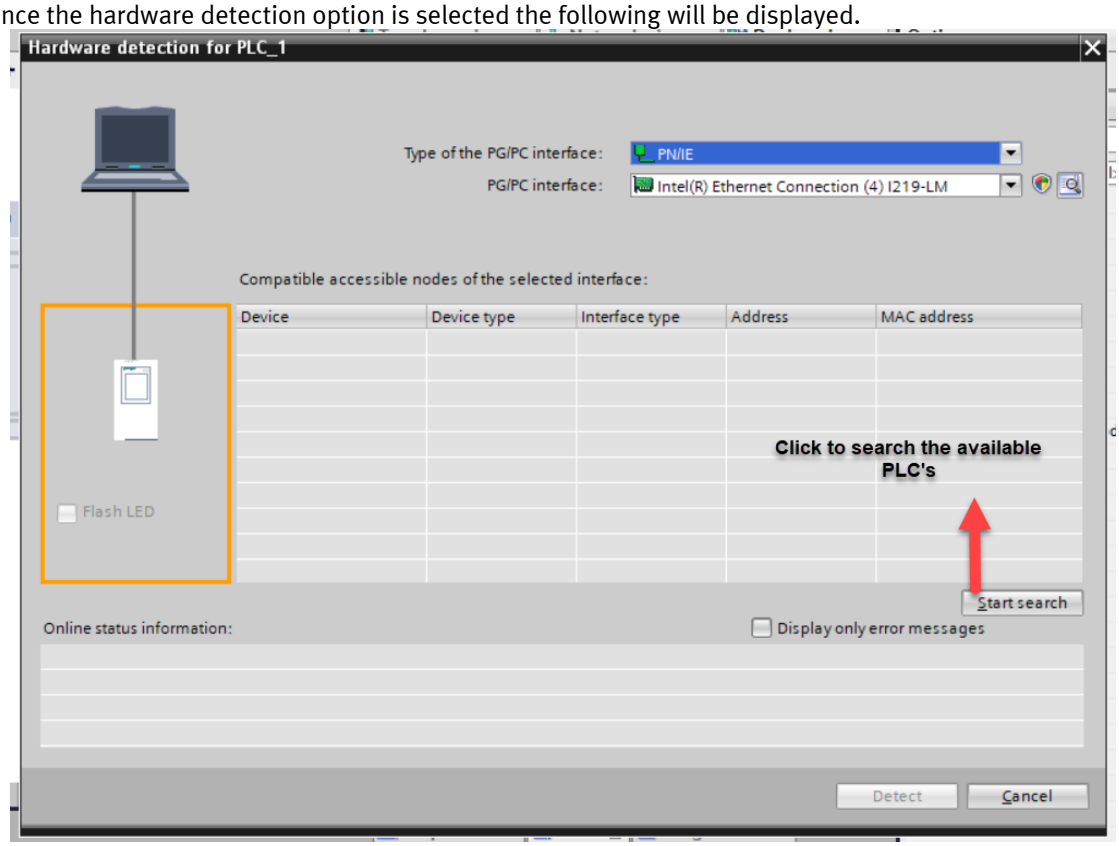


4.2 Detecting the actual hardware configuration of the PLC connected in the network

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



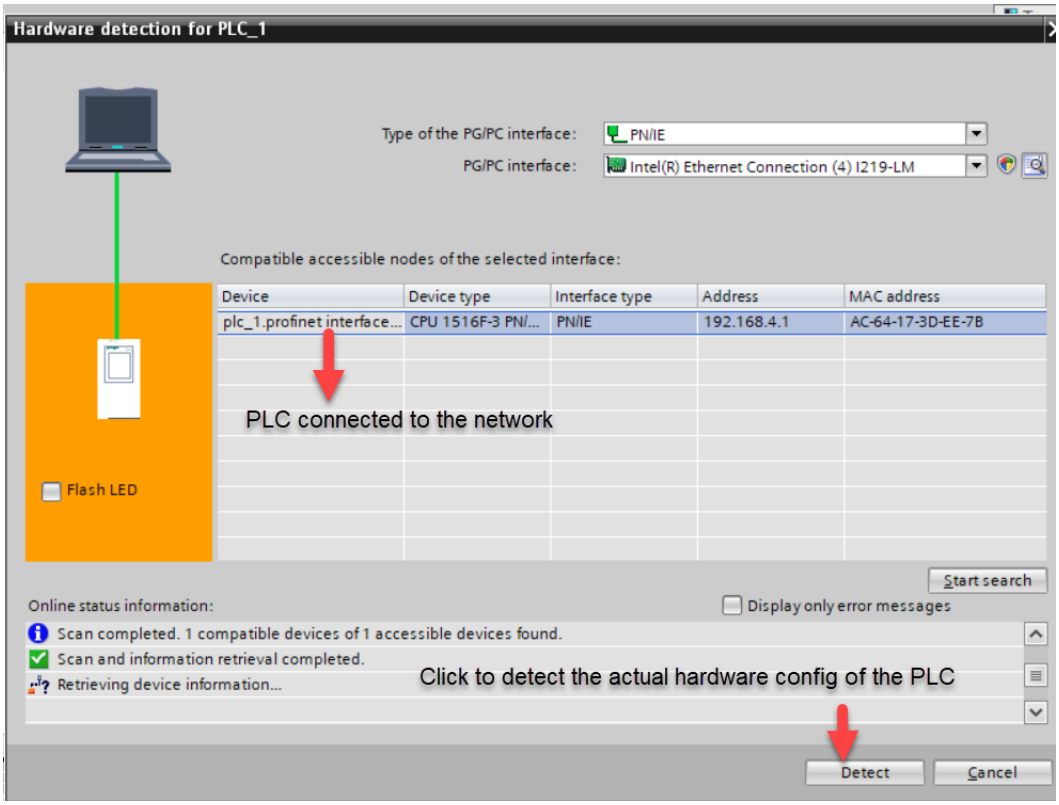
2. Once the hardware detection option is selected the following will be displayed.



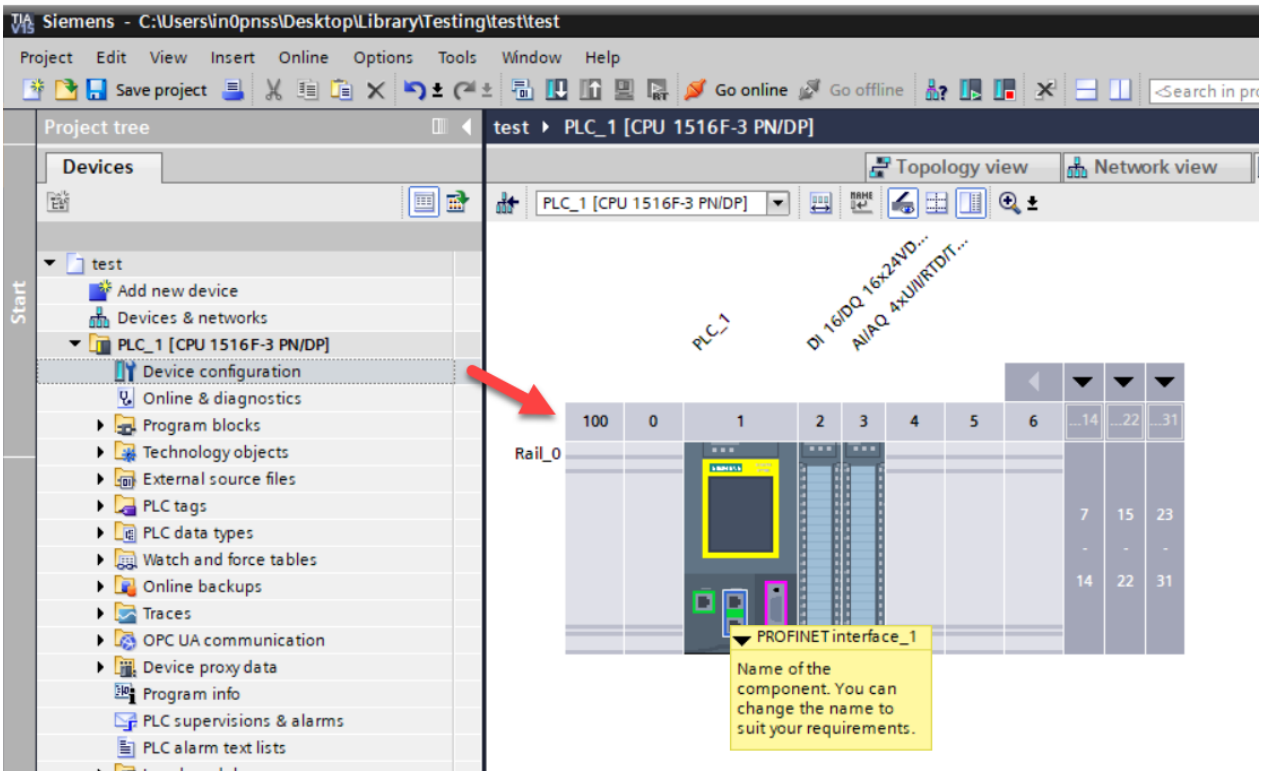
3. Click **Start Search** to find the PLC's which are available in the network as shown in the above image.

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 SBSI Vision sensor 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 .

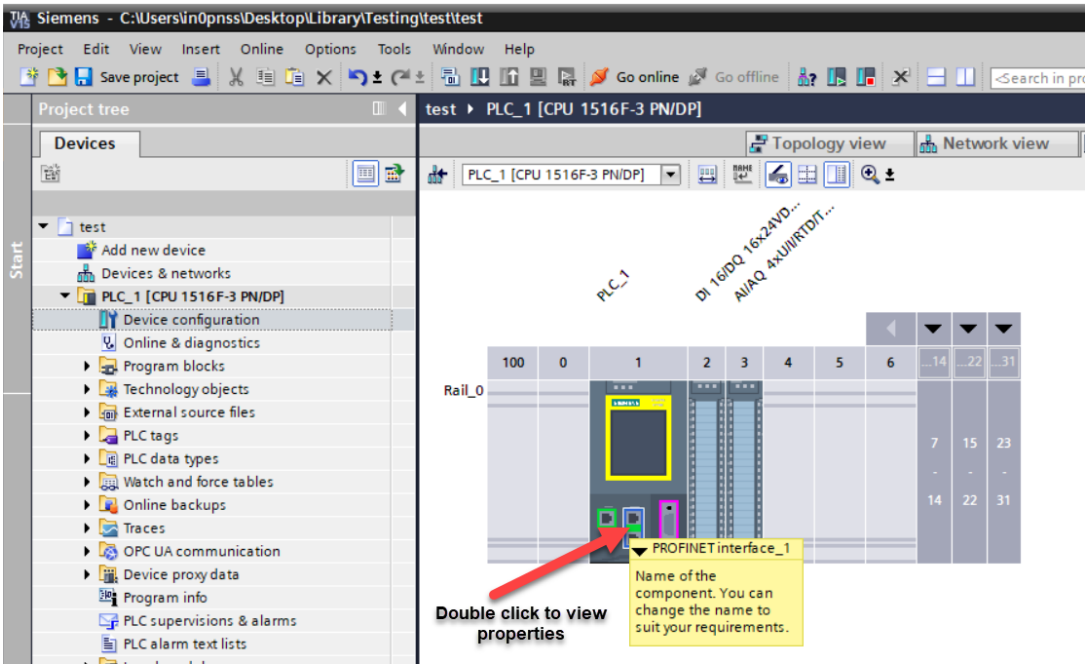


4.3 Configuration of the IP parameters of the profinet interface of the PLC.

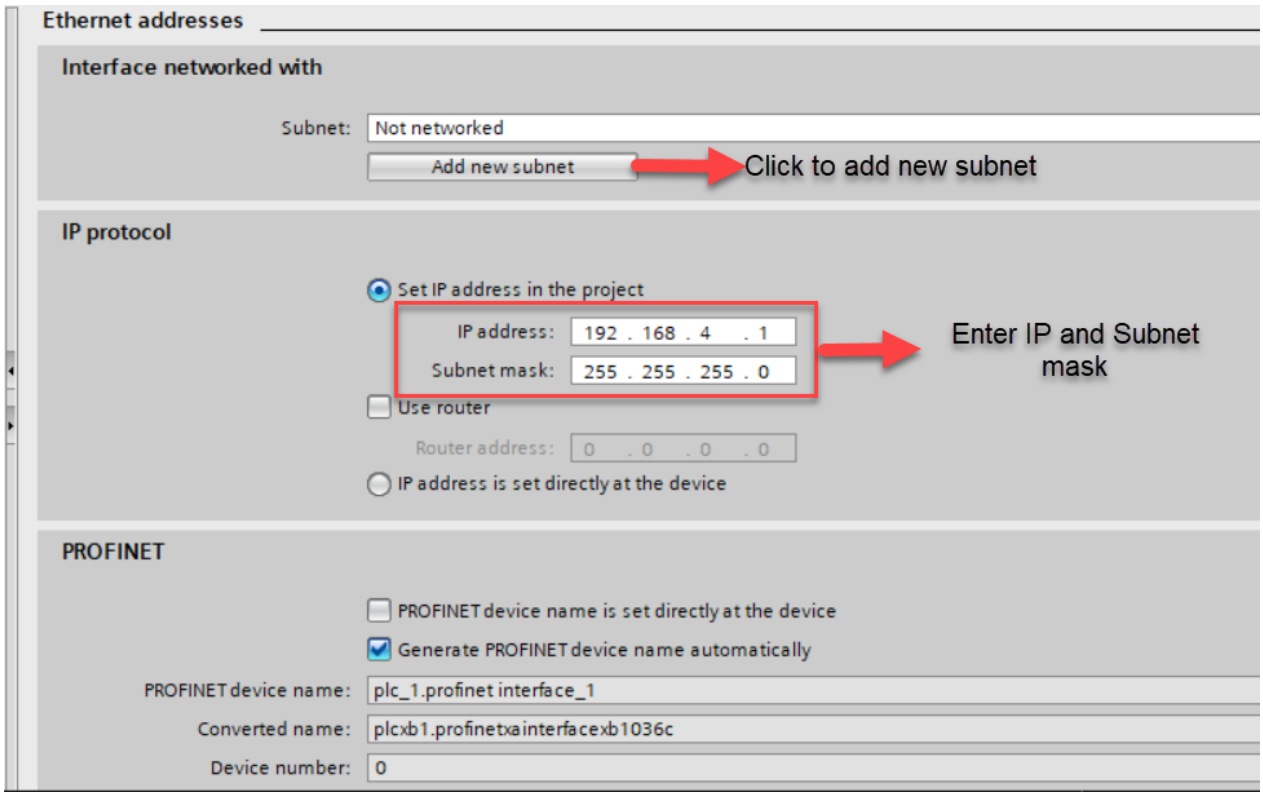
NOTE

- **The Profinet Interface address of PLC , SBS Vision Sensor 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.



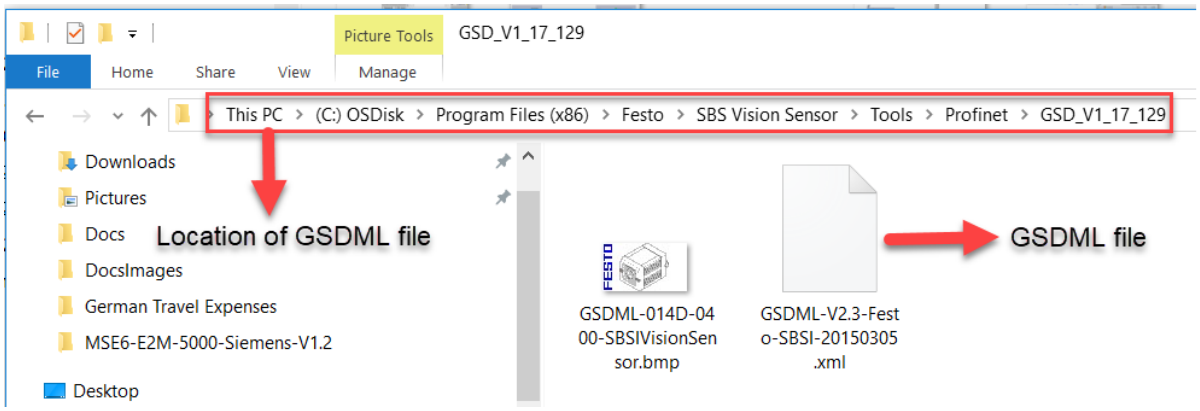
2. Go to **General › Ethernet address** to change the IP parameters of Profinet interface_1.



5 Adding GSDML File of SBS Vision sensor to TIA Portal.

5.1 GSDML file location in the installation folder

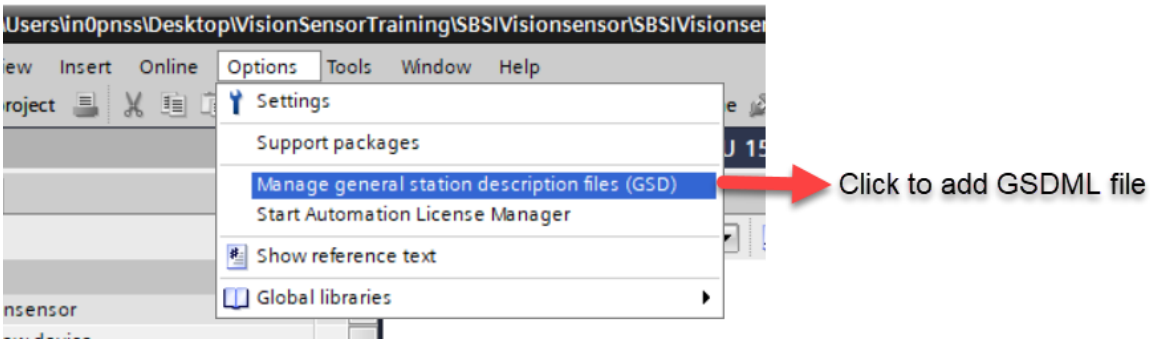
1. The Profinet GSDML file can be found in the location where the SBS Vision sensor software in the system.
   It can be found in the below location:
   C:\Program Files (x86)\Festo\SBS Vision Sensor\Tools\Profinet\GSD_V1_17_129



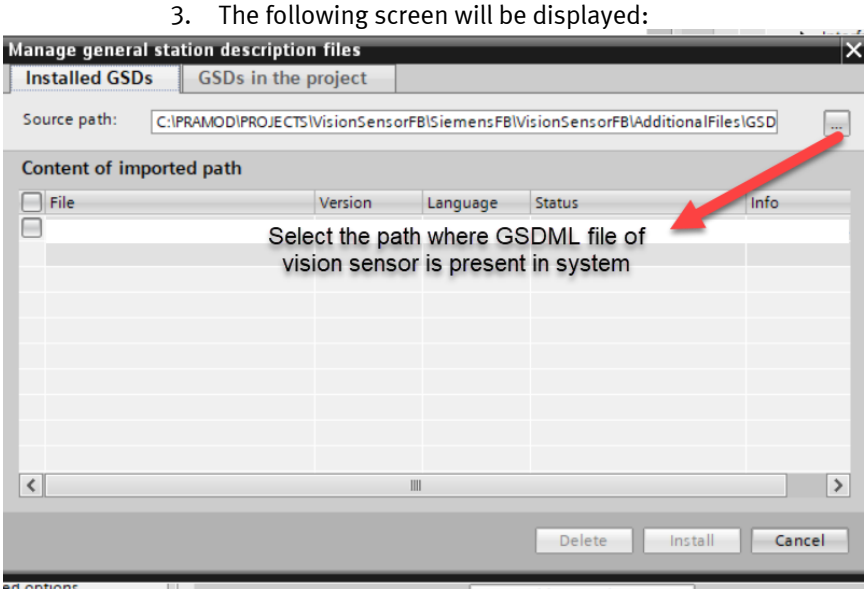
5.2 Adding the GSD File to TIA Portal

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

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

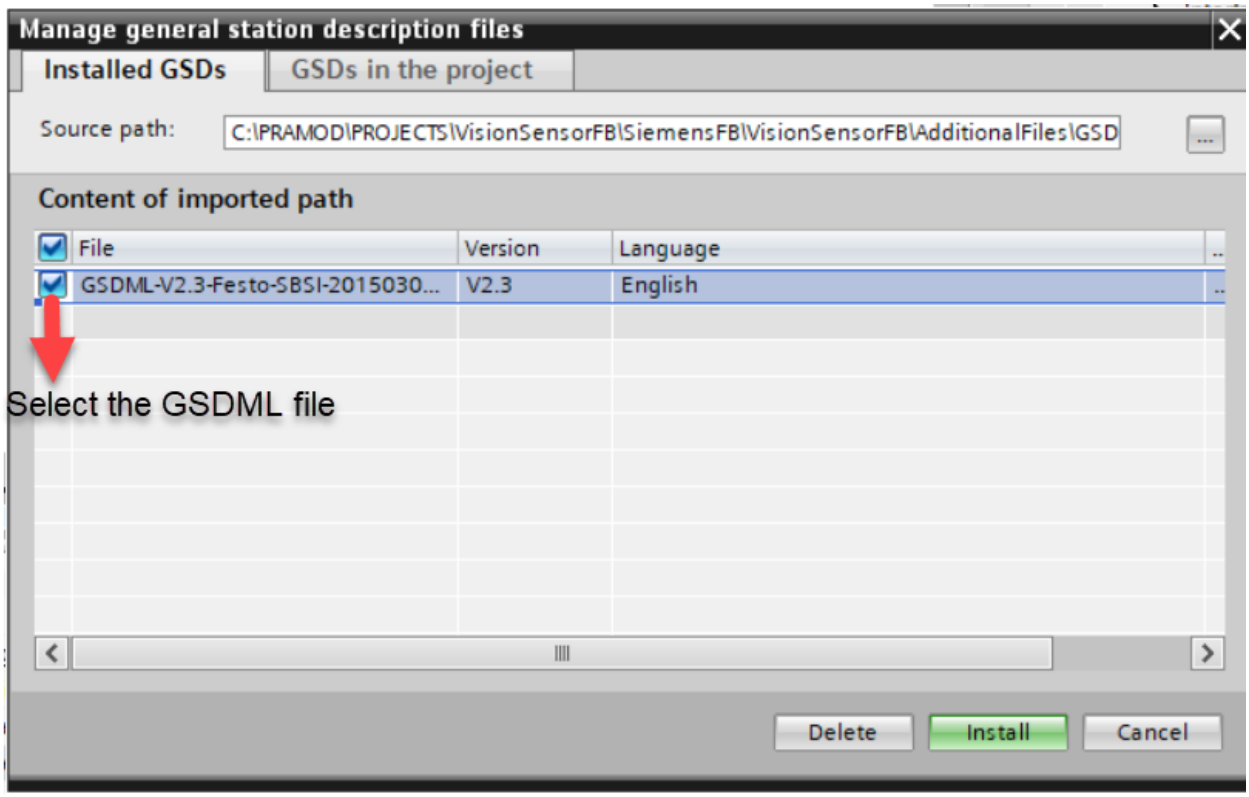


2. Click on **Manage general station description files(GSD)** as shown in above image.
3. The following screen will be displayed:

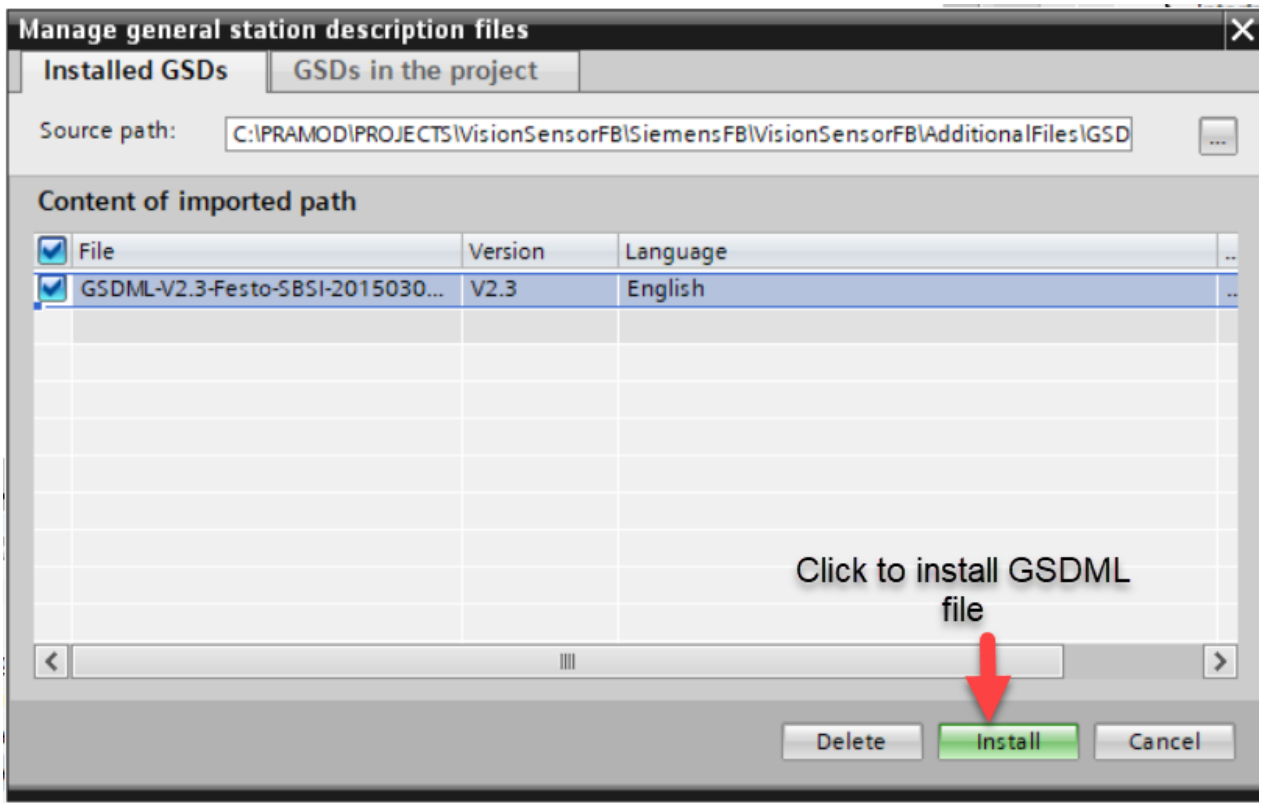


4. As shown in the above image select the path where the GSDML file is saved in your system.

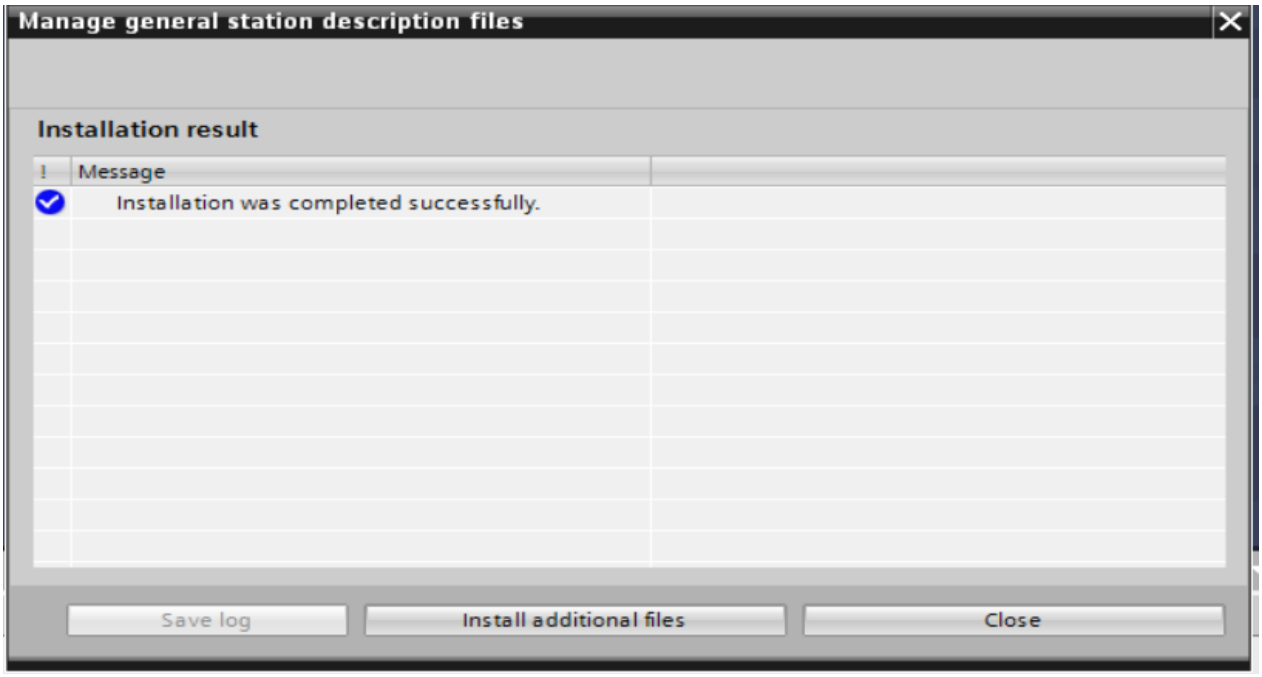
5. Once the appropriate file has been chosen the following screen will be displayed:



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



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

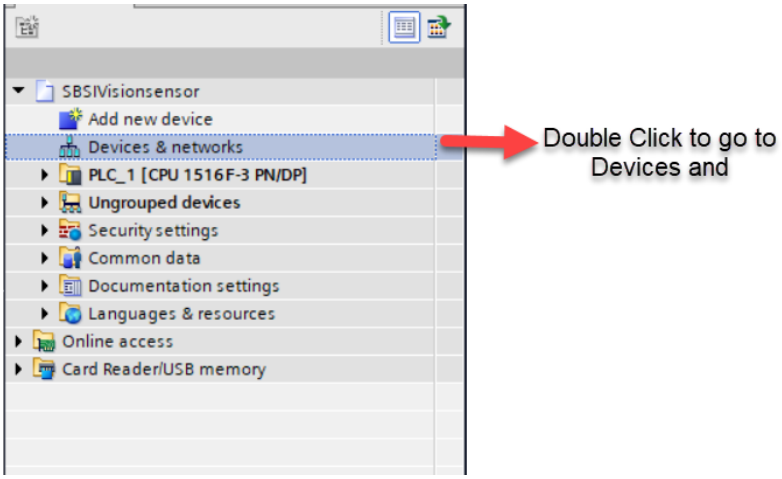


8. Click on **CLOSE** button to finish the installation and update the hardware catalogue.

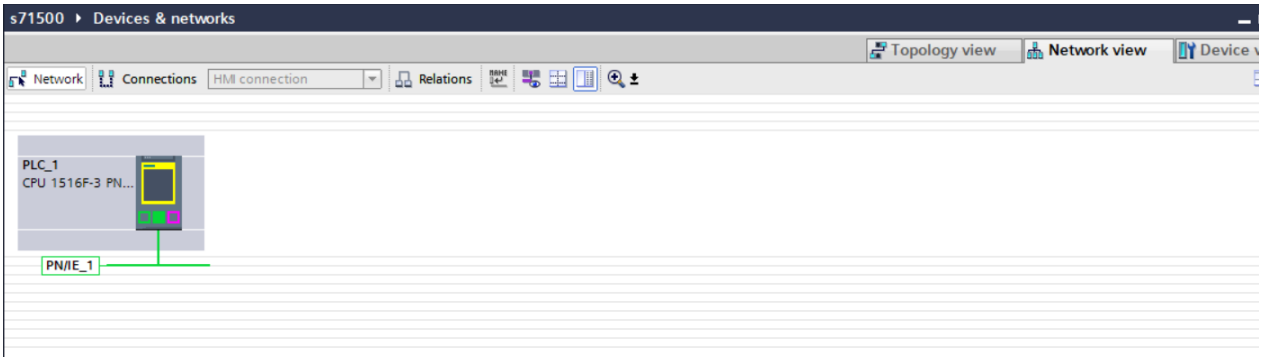
6 Configuration of SBS Vision Sensor in TIA Portal.

6.1 Adding the installed SBS Vision Sensor to Network View.

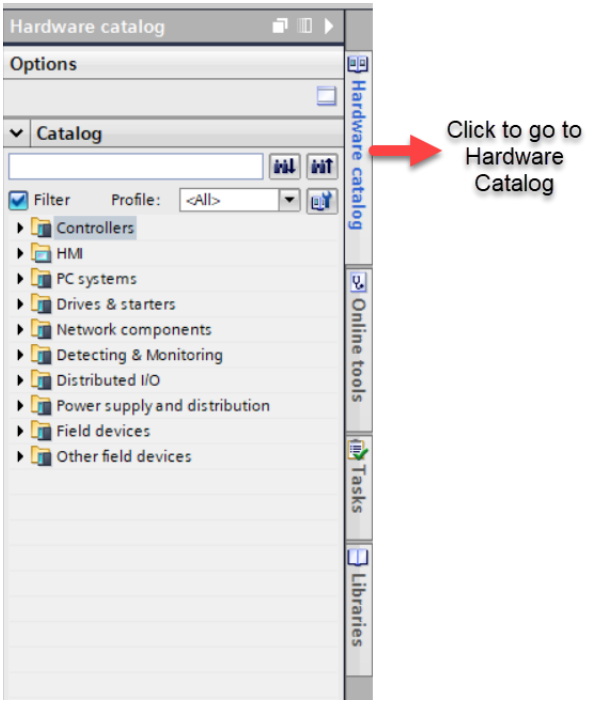
1. Double on **Devices and Networks** .



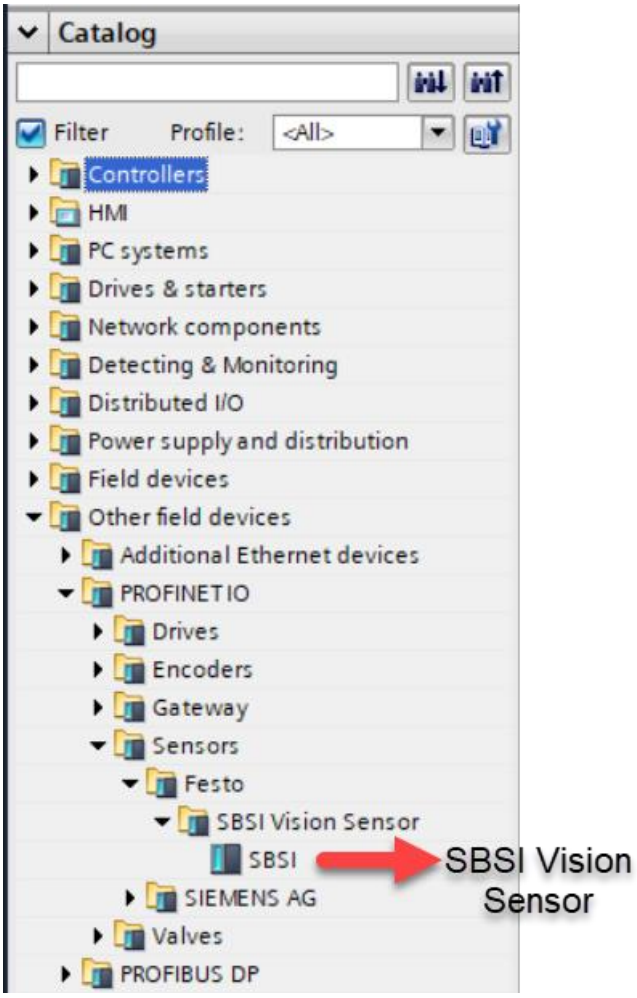
2. The network view will be displayed as shown below.



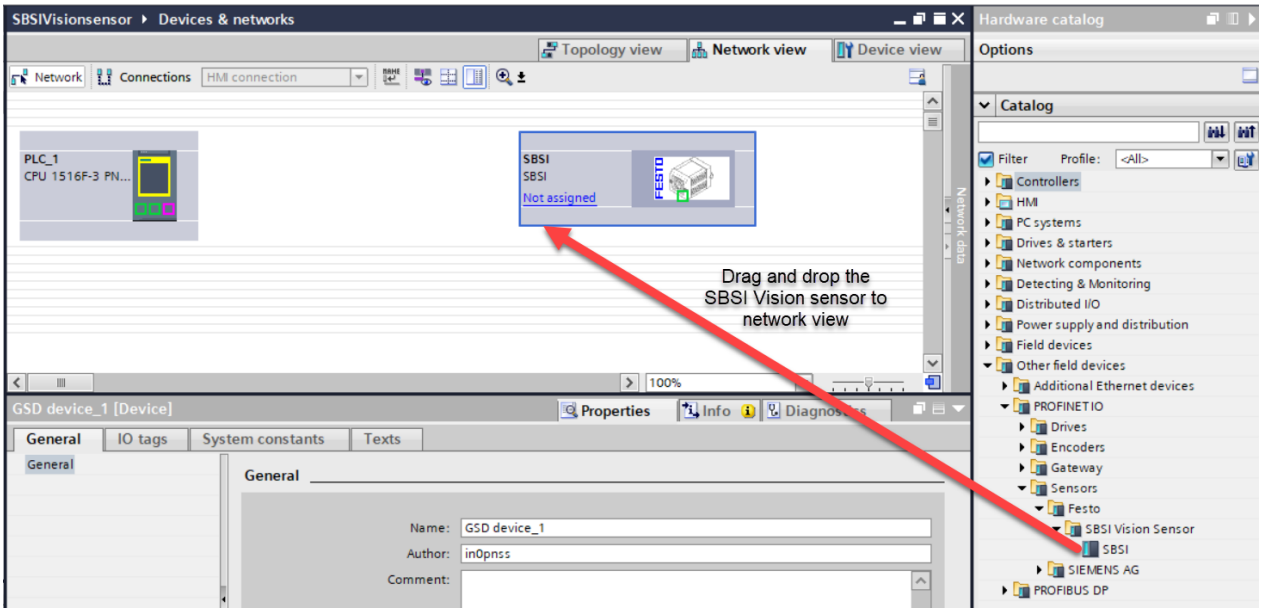
3. Go to **Hardware Catalog** as shown below.



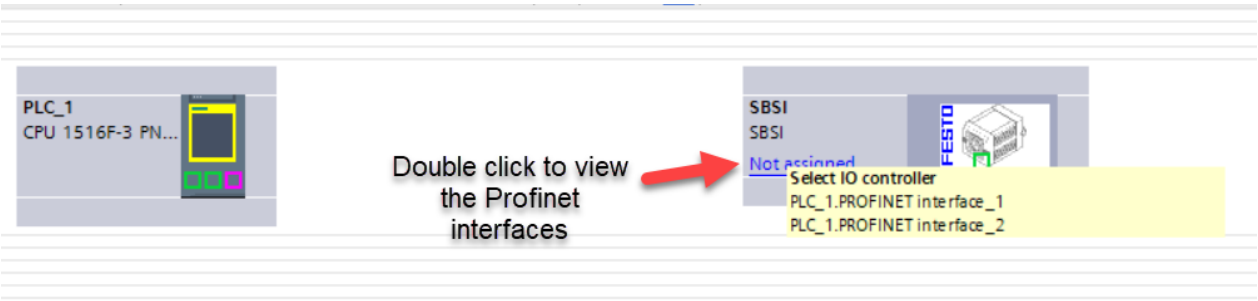
4. Under Hardware Catalog, Go to **Other Field Devices ›››Profinet IO ››› Sensors ››› Festo ››› SBSI Vision Sensor ››› SBSI.**



5. Drag and drop the SBSI Vision sensor to the network view as shown below.



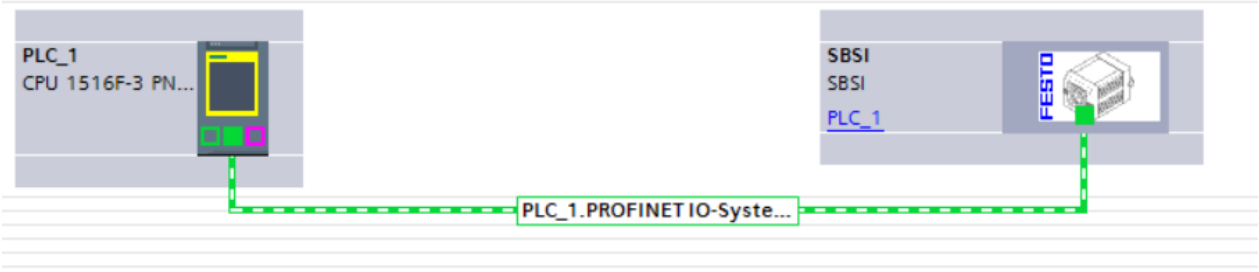
6. Connect the S7-1500 PLC to SBSI Vision Sensor.
7. Double click on **Not Assigned** as shown below.



8. If the SBSI Vision sensor is connected to Profinet Interface _2(X2) , then select PLC_1.PROFINET interface_2.

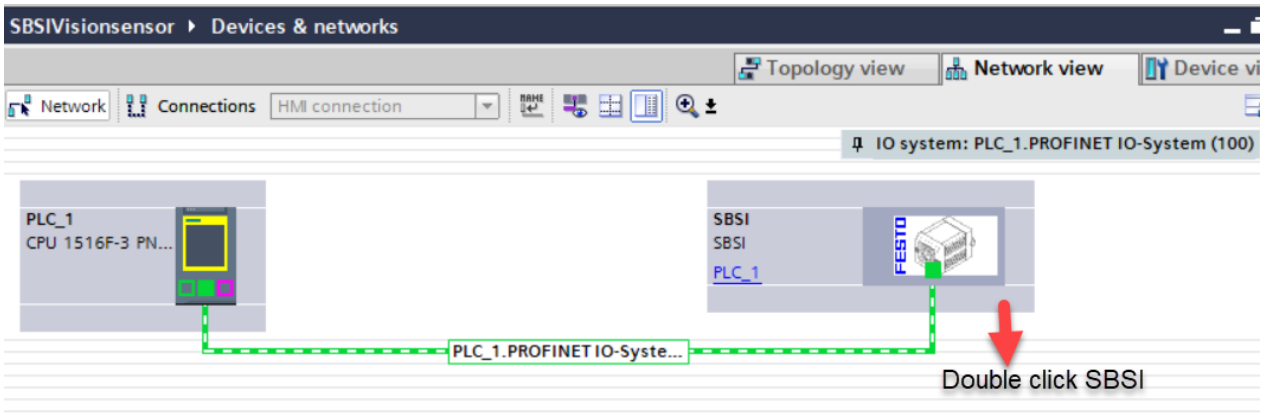
   If the SBSI Vision sensor is connected to Profinet Interface _1(X1) , then select PLC_1.PROFINET inter face_1.

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

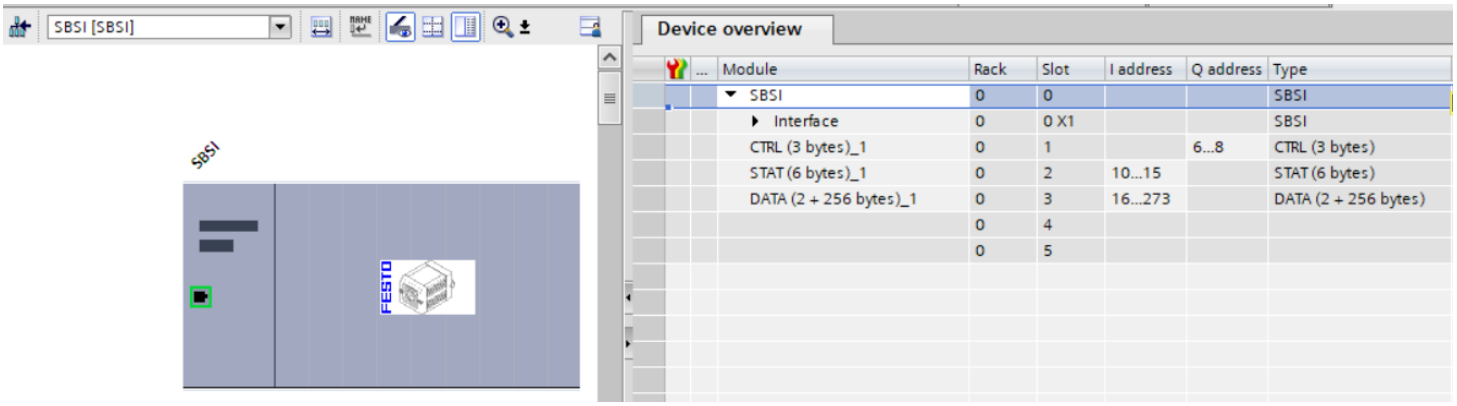


6.2 Network Configuration of Profinet Interface SBSI Vision Sensor in TIA Portal.

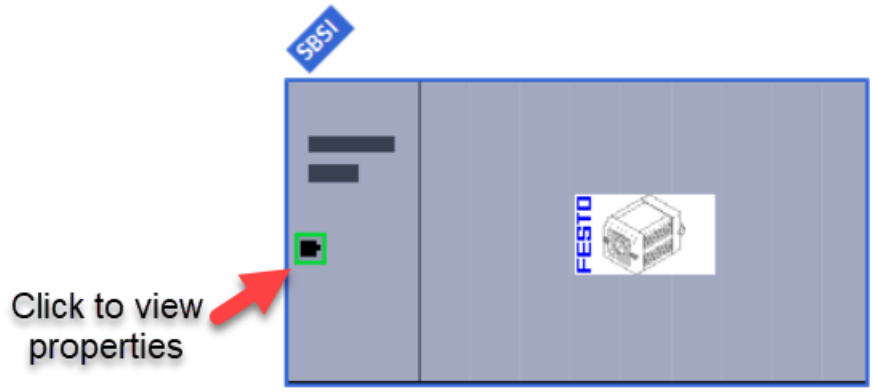
1. Go to **Devices and Networks** and Double click **SBSI Vision Sensor**.



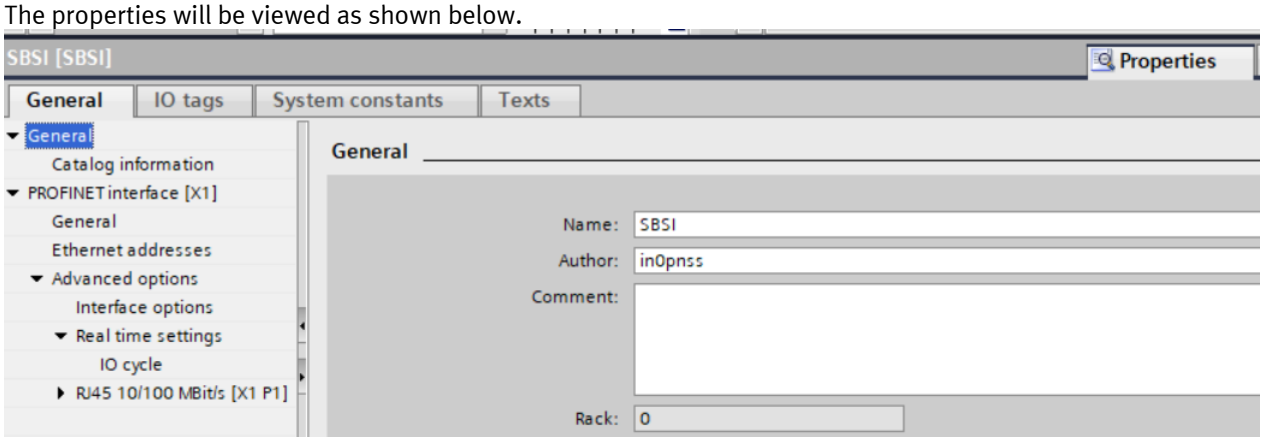
2. Once you double click the following display will be seen.



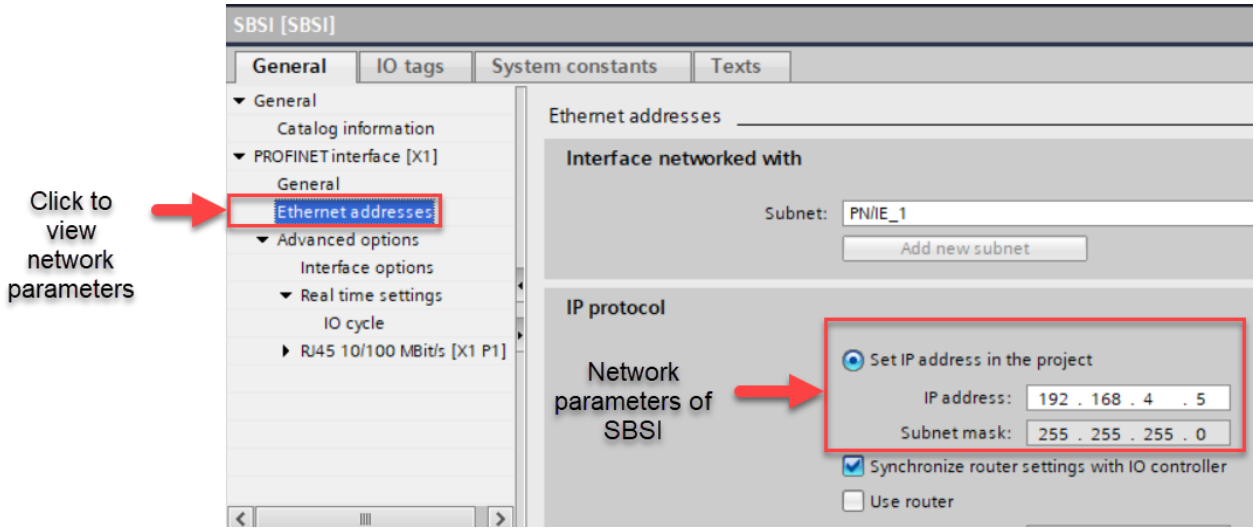
3. Click on **SBSI** to view the properties .



4. The properties will be viewed as shown below.



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

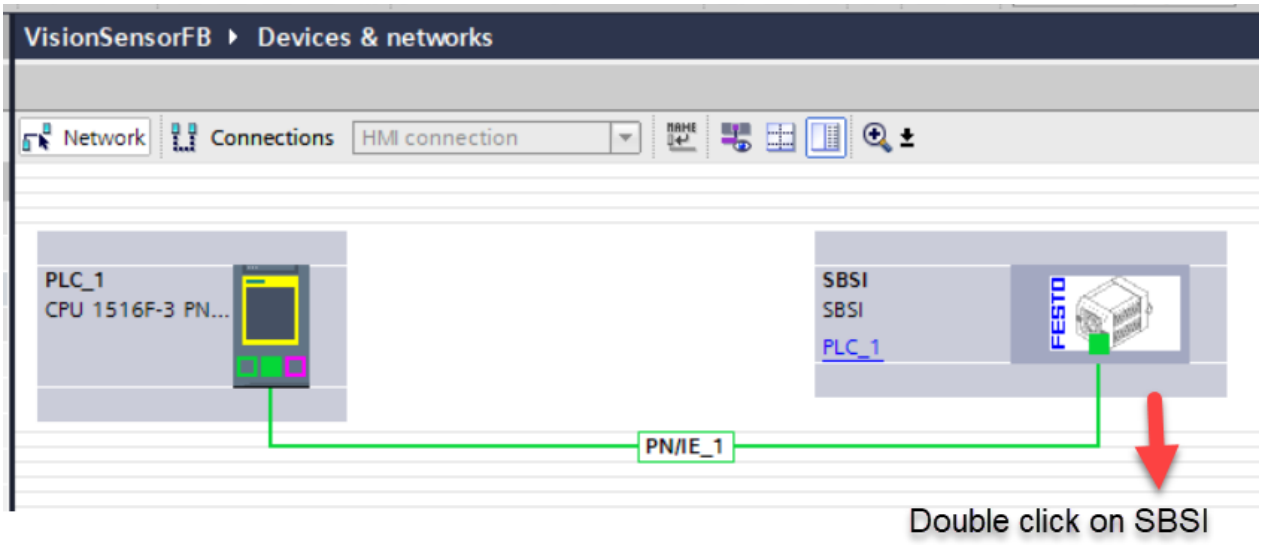


NOTE

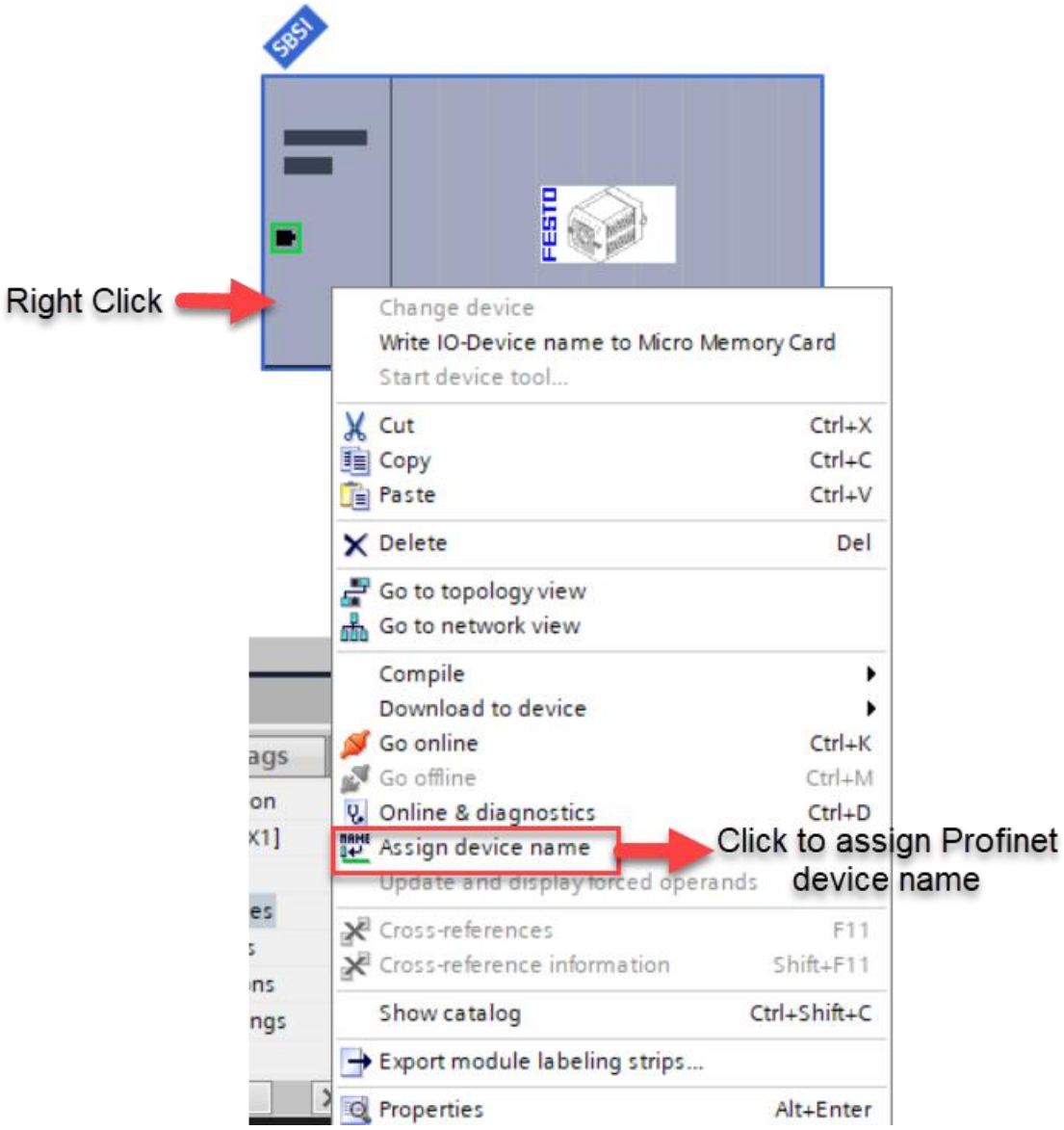
- The IP address of the Profinet interface **SBS Vision Sensor , S7-1500 PLC and** IP address of your PC must be in the same range.

6.3 Assigning Profinet name to SBSI Vision Sensor.

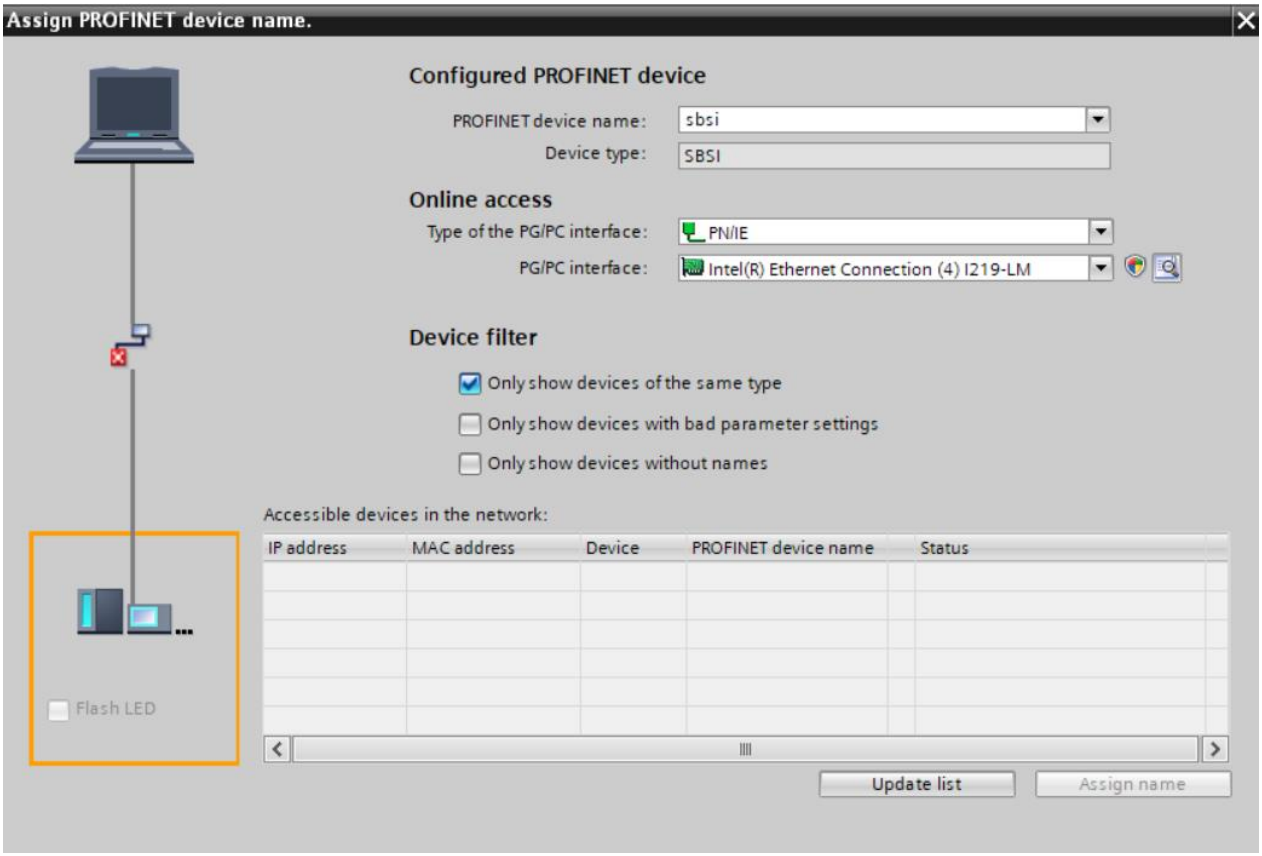
1. Go to **Devices and Networks** . Double click **SBSI** to go to **device view**.



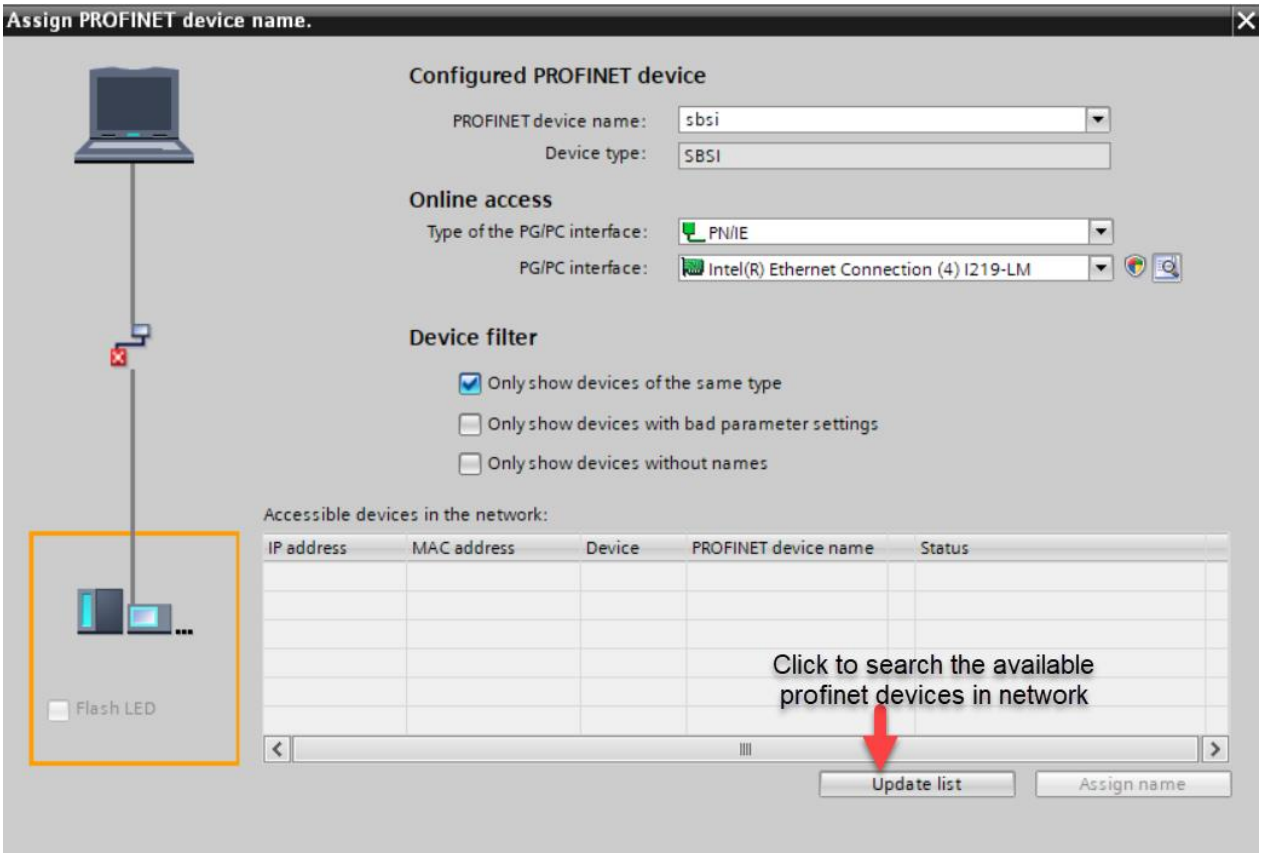
2. Right click on **MSE** ›› Click **Assign Device Name** as shown below.



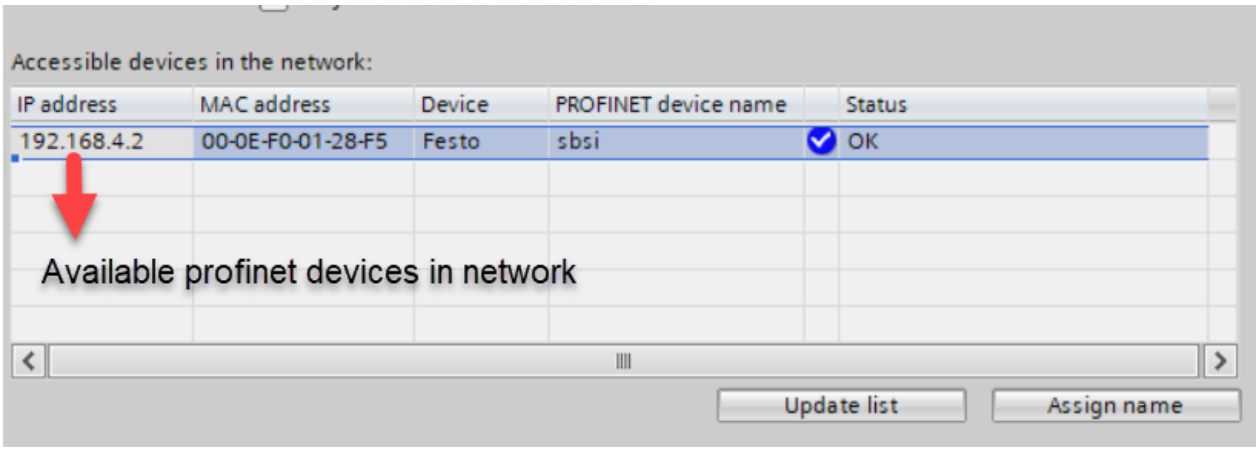
3. The wizard for assigning Profinet Device name looks as shown below.



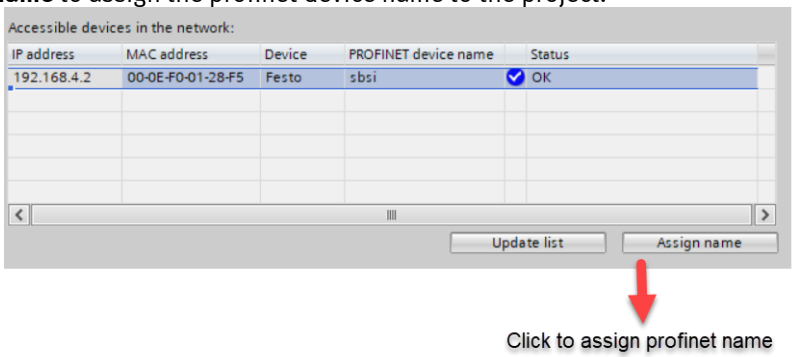
4. Search the available devices in the Profinet network by clicking **Update List.**



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

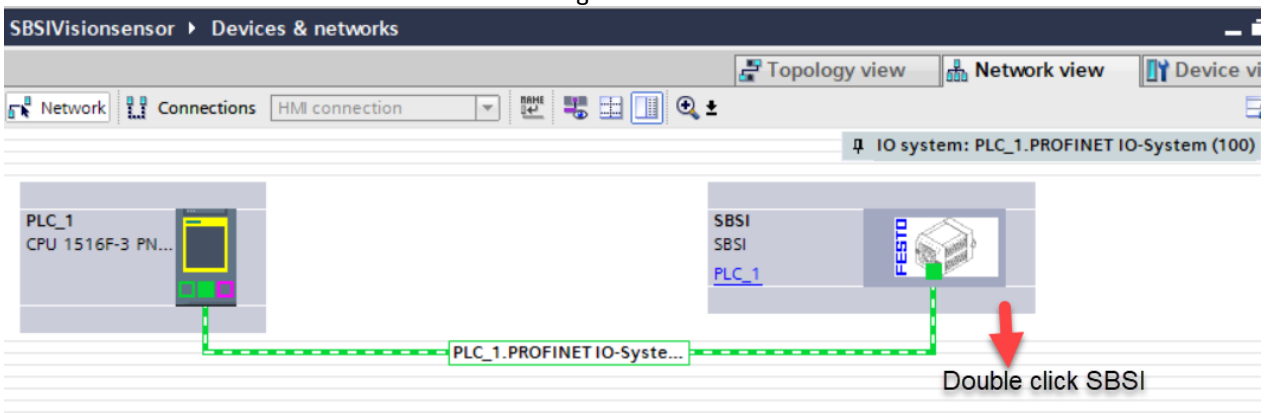


6. Click **Assign Name** to assign the profinet device name to the project.

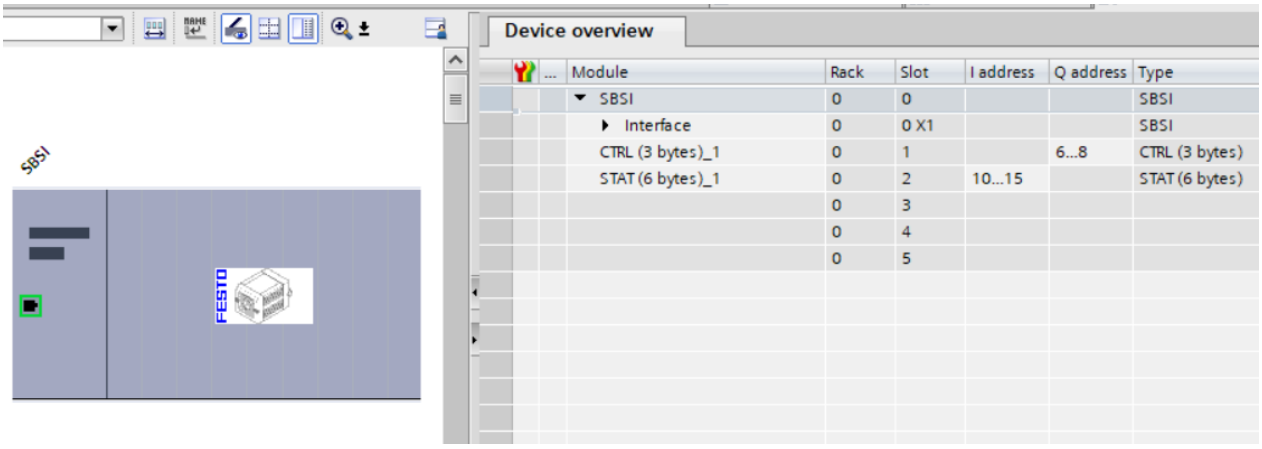


6.4 Adding Data Module to SBSI.

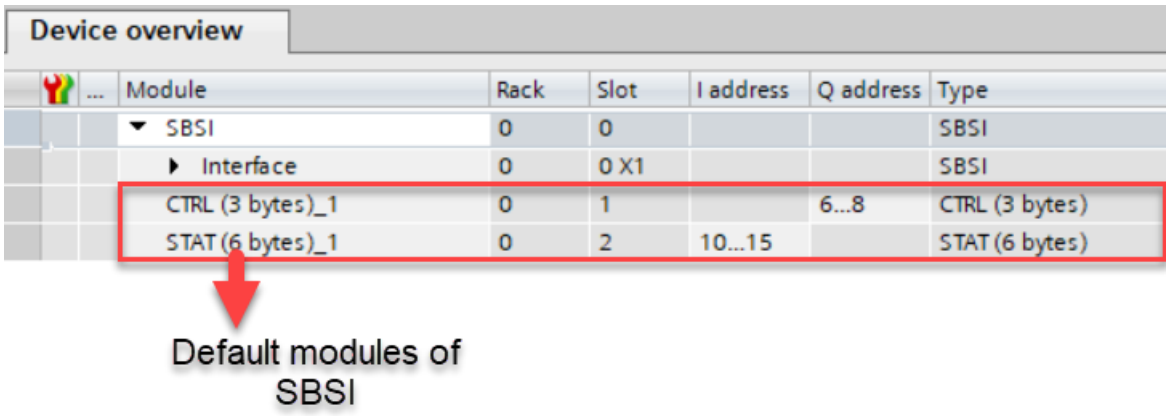
1. Go to **Devices and Networks** . Double click **SBSI** to go to **device view**.



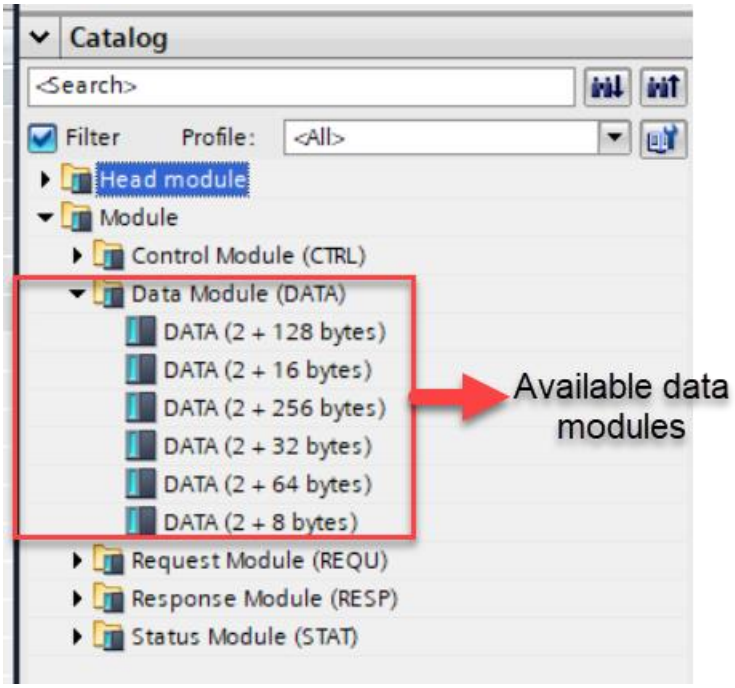
- Device view appears as shown below.



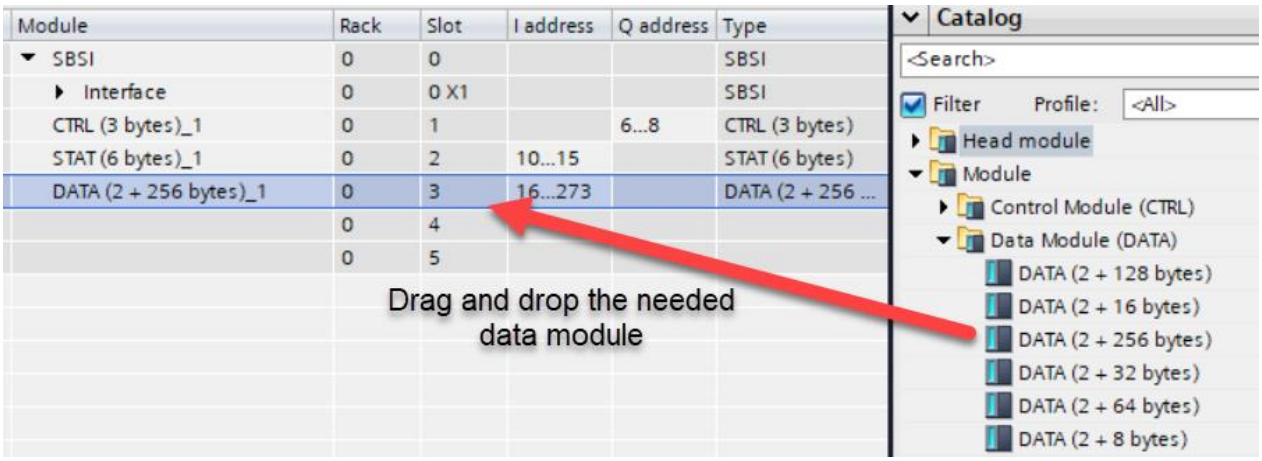
- By default CTRL and STAT bytes will be there in the SBSI configuration as shown below.



- Based on the payload selection the telegram in Vision sensor configuration studio, Select the appropriate data module.
- The available options are :

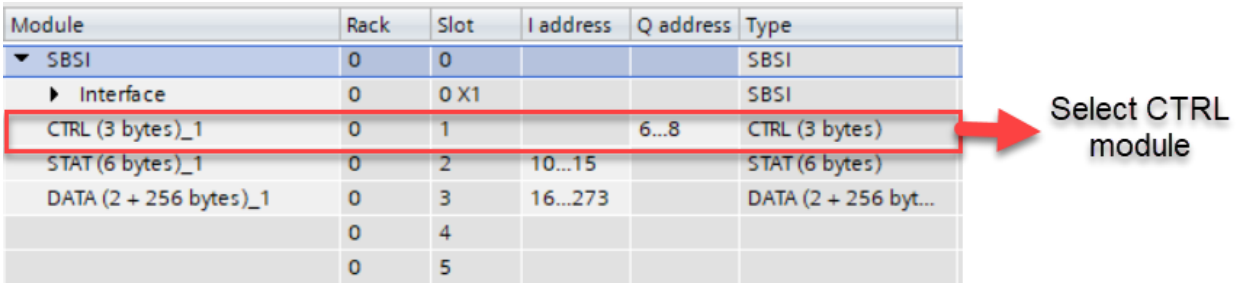


- Drag and drop the needed data module to the Device Overview as shown below.
  In our example we have considered DATA(2+ 256 bytes).

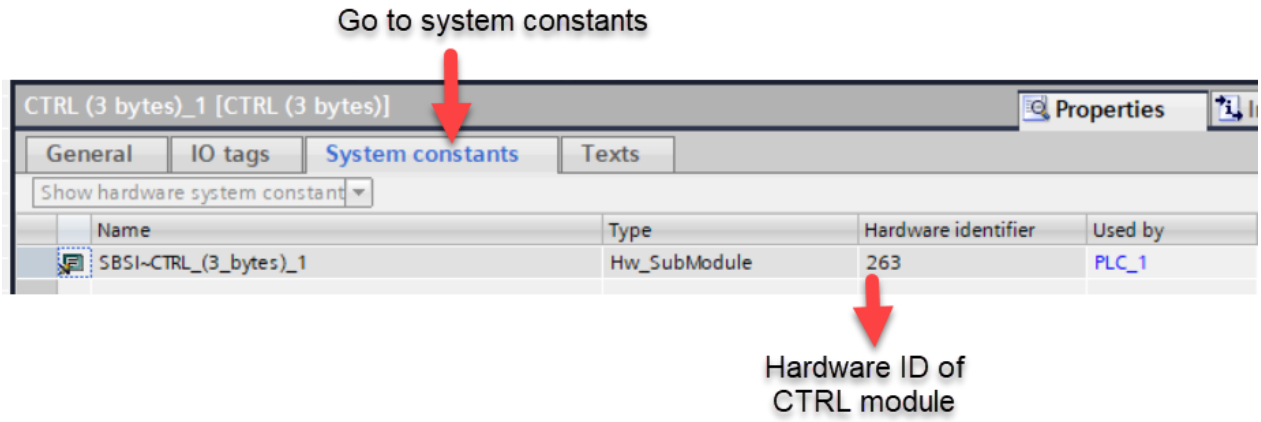


6.5 Identifying Hardware ID of the Sub modules of SBSI Vision Sensor.

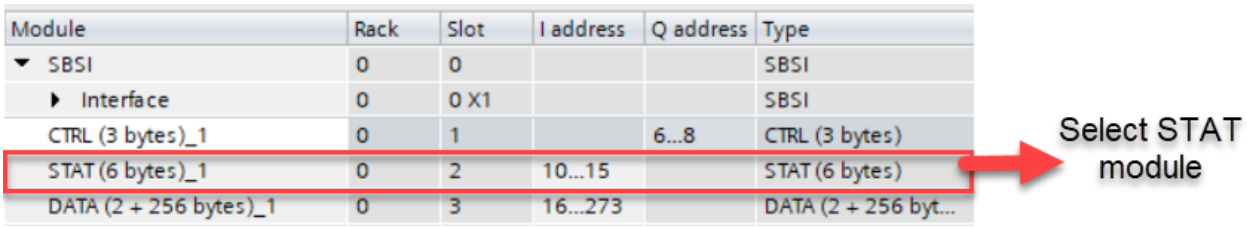
- In **device view** , select **CTRL module** as shown below.



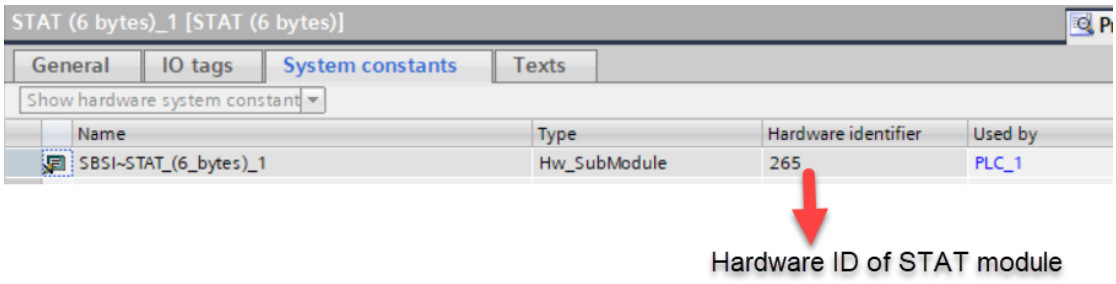
- Select **System Constants** as shown below from the Properties tab of **CTRL** module.



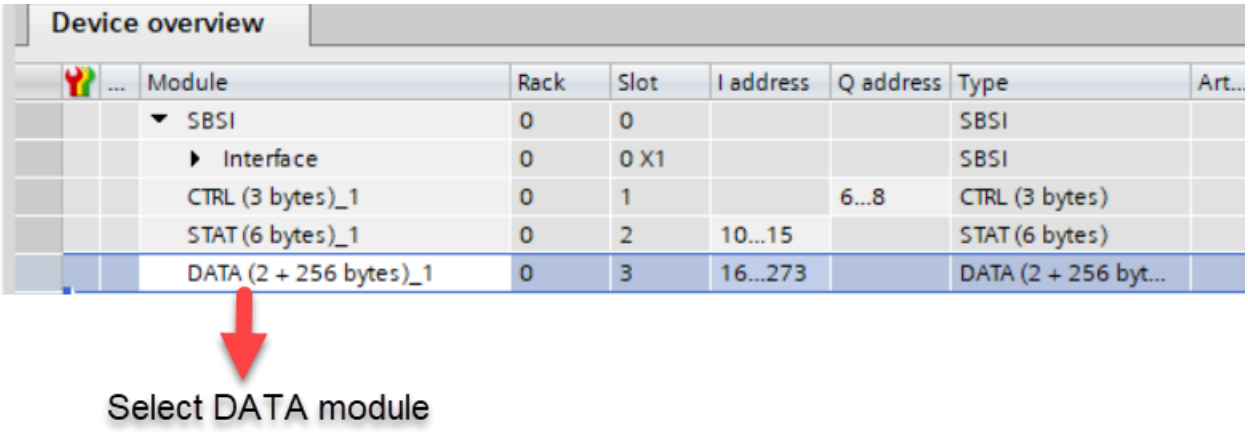
- In **device view** , select **STAT module** as shown below.



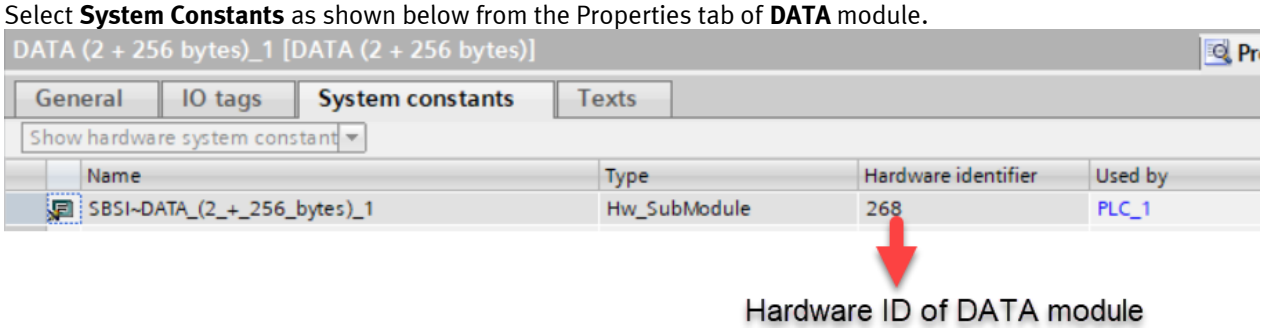
- Select **System Constants** as shown below from the Properties tab of **CTRL** module.



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



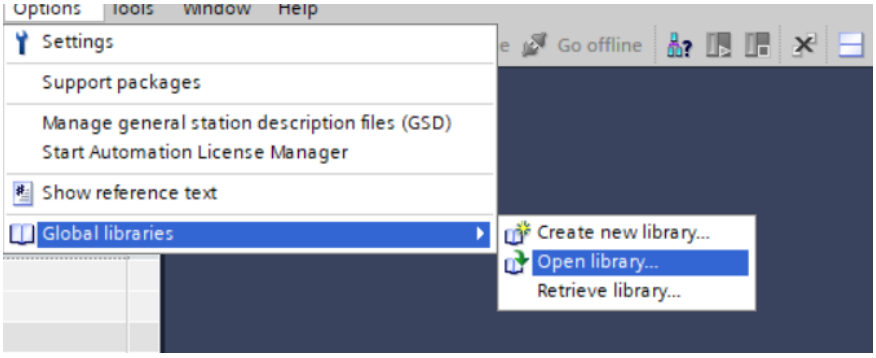
- Select **System Constants** as shown below from the Properties tab of **DATA** module.



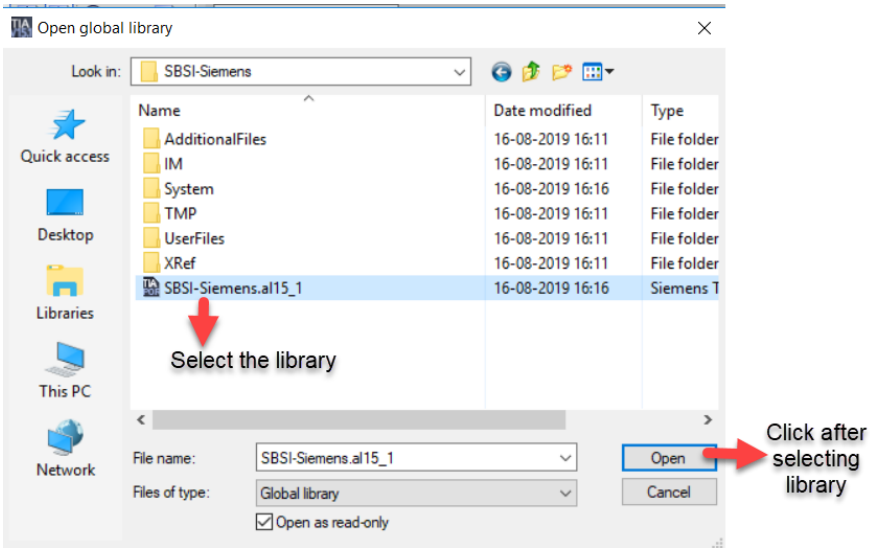
7 Linking VisionSensor library to the project.

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

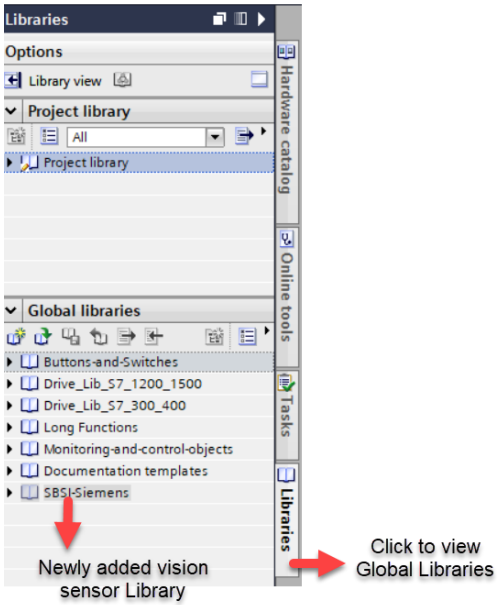
2. Click on **Options › Global Libraries › Open Library** as shown below.



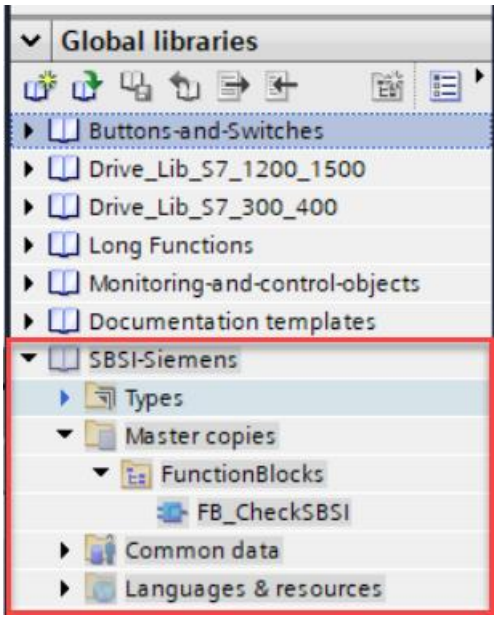
3. Choose the path where the Library is saved in your system. Select the library and click **OPEN.**



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



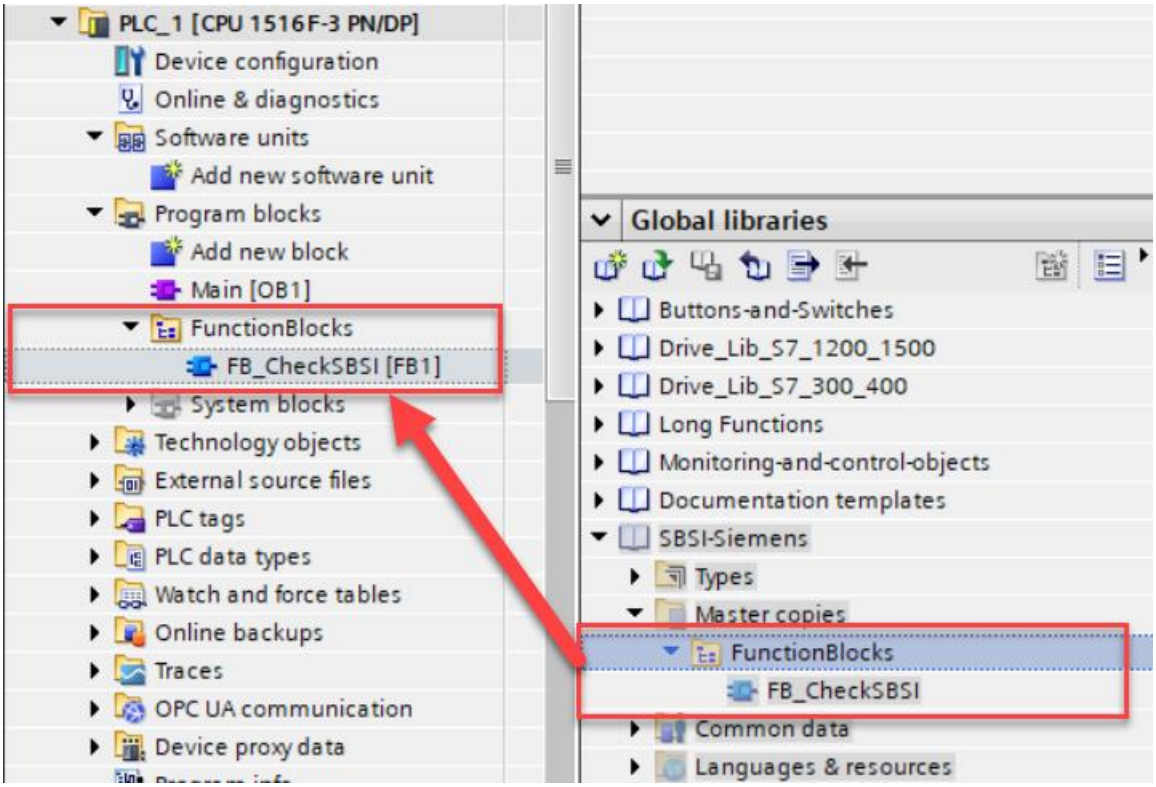
5. Open the **SBSI-Siemens** Library to view the content.



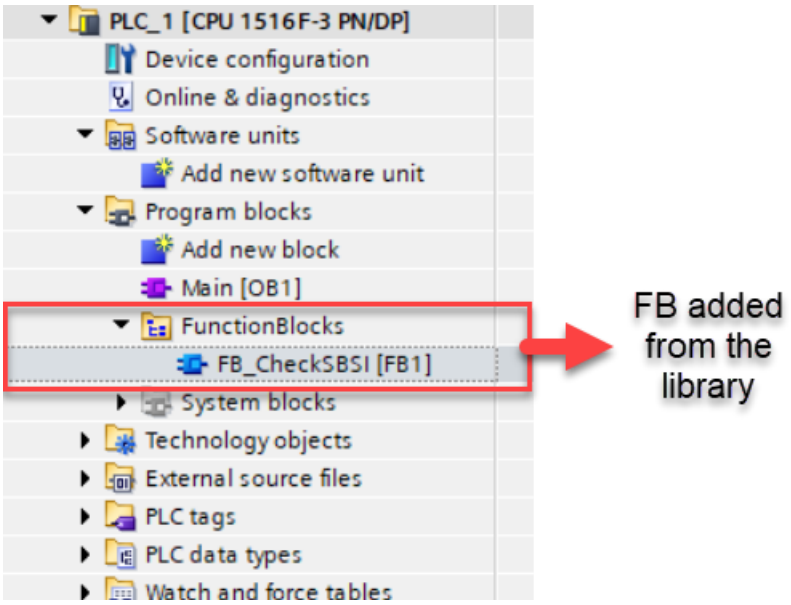
6. The Library has the following sub folders:

- Function Blocks : It has the Function Block named **FB_CheckSBSI** which is used to control the SBSI Vision Sensor 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.


PLC_1 [CPU 1516F-3 PN/DP]
Device configuration
Online & diagnostics
Software units
Add new software unit
Program blocks
Add new block
Main [OB1]
FunctionBlocks
FB_CheckSBSI [FB1]
System blocks
Technology objects
External source files
PLC tags
PLC data types
Watch and force tables
FB added from the library

8 DESCRIPTION OF SBSIVISION-Siemens Library FUNCTION BLOCKS

The library has 1 Function Blocks within it:

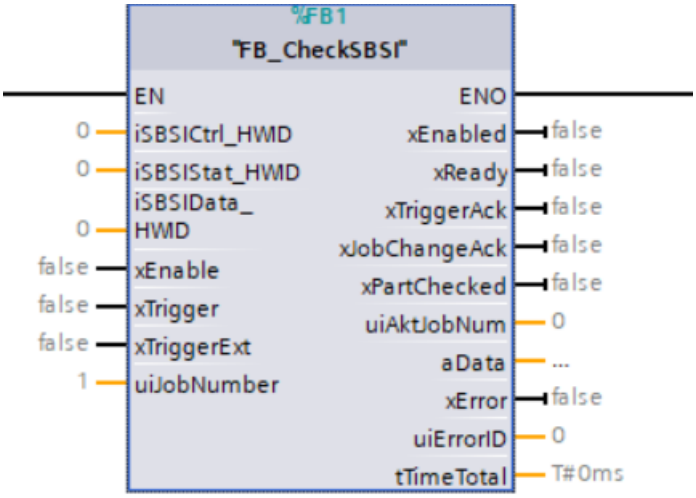
1. FB_CheckSBSI

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 SBSI Vision Sensor.
- Using the Function Block the job number of the vision sensor can be changed.
- It gives the various status information of the vision sensor operation like **Active job number**, **Error information** occurred during the operation, **Job change acknowledgement**, **Trigger acknowledgement**.
- It allows outputs the payload result data of the active job.

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 SBSI Vision Sensor.
iSBSIStat_HWID	HW_ANY	Hardware ID of the **Stat** Module of SBSI Vision Sensor.
iSBSIData_HWID	HW_ANY	Hardware ID of the **Data** Module of SBSI Vision Sensor.
xEnable	BOOL	**FALSE** – Disable the Function Block. **TRUE** – Enable the Function Block.
xTrigger	BOOL	**TRUE** – Rising Edge (Low ==› High) triggers the Vision Sensor.
xTriggerExt	BOOL	**TRUE** – Hardware Trigger or free run enabled. **FALSE** – Hardware Trigger or free run disabled.

NAME	DATA TYPE	DESCRIPTION
uiJobNumber	BOOL	Gives the active job number of the vision sensor. 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** – SBSI Vision Sensor is not ready for next evaluation cycle. **TRUE** – SBSI Vision Sensor is ready for next evaluation cycle.
xTriggerAck	BOOL	**FALSE** – No acknowledge for a successful trigger to SBSI Vision Sensor. **TRUE** – Acknowledge for a successful trigger to SBSI Vision Sensor.
xPartChecked	BOOL	**FALSE** – SBSI VisionSensor is busy in operation. **TRUE** – SBSI VisionSensor 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 Vision Configuration Studio in **Output/Telegram/Payload.**
xError	BOOL	**FALSE** – No Error during SBSI Vision Sensor operation. **TRUE** – Error during SBSI Vision Sensor operation.

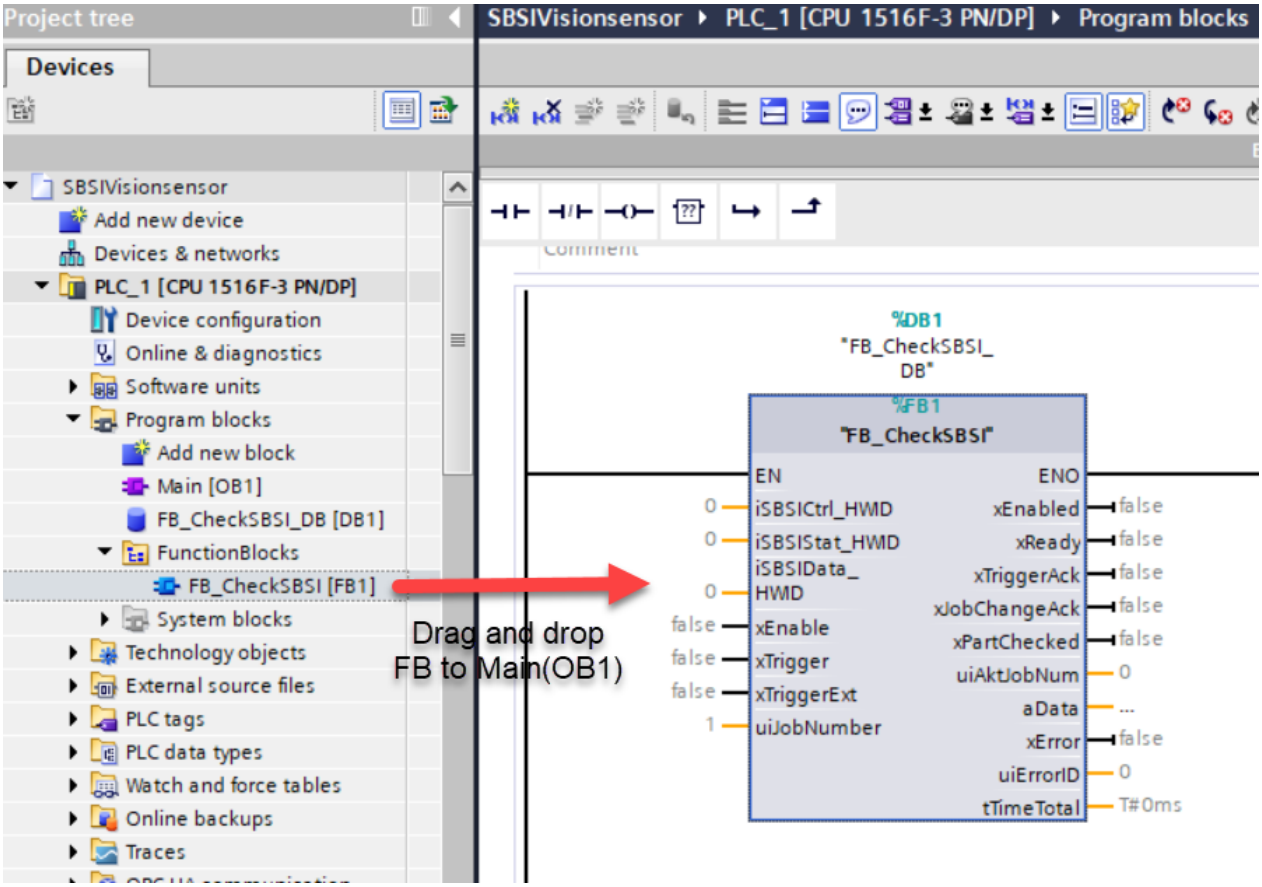
NAME	DATA TYPE	DESCRIPTION
uiErrorId	BOOL	Gives the ID of the error occured. It has following values. **0 –** No Error. **1 –** Failure Trigger request **2 –** Failure Change Job. **3 –** Failure switch to run. **5 –** Failure Profinet Not active in Job. **15 –** System Error
tTimeTotal	TIME	Time period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

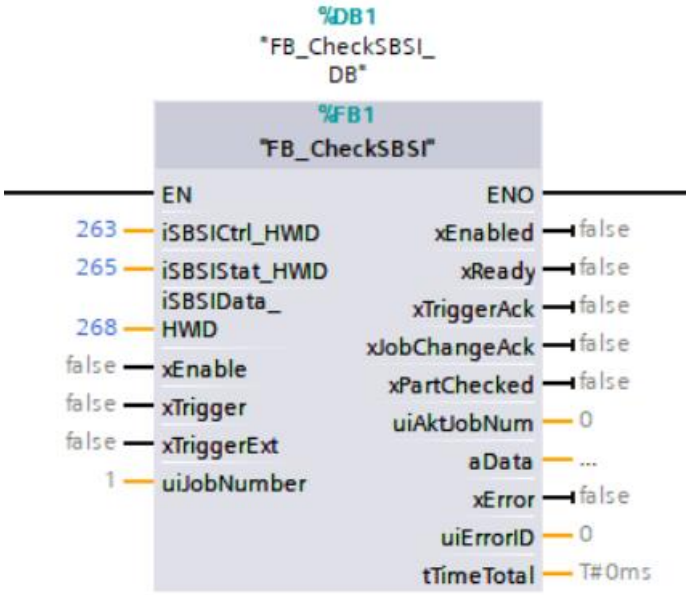
NOTE

- **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.



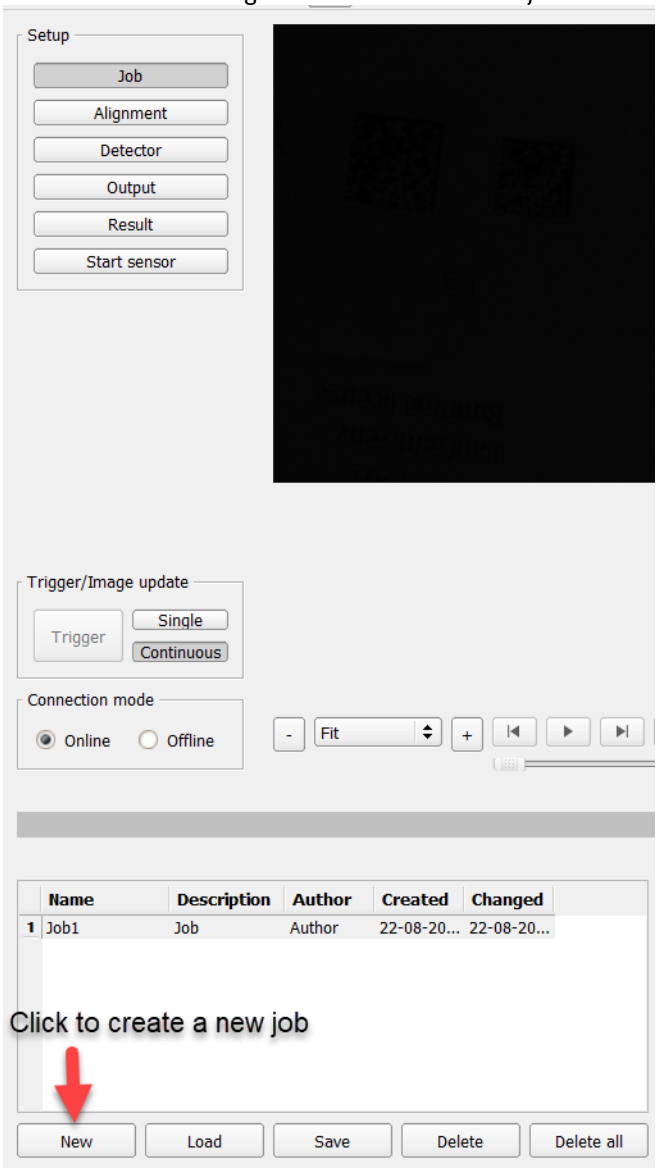
10 FUNCTION BLOCK EXECUTION WITH AN EXAMPLE

This chapter will explain the following in detail:

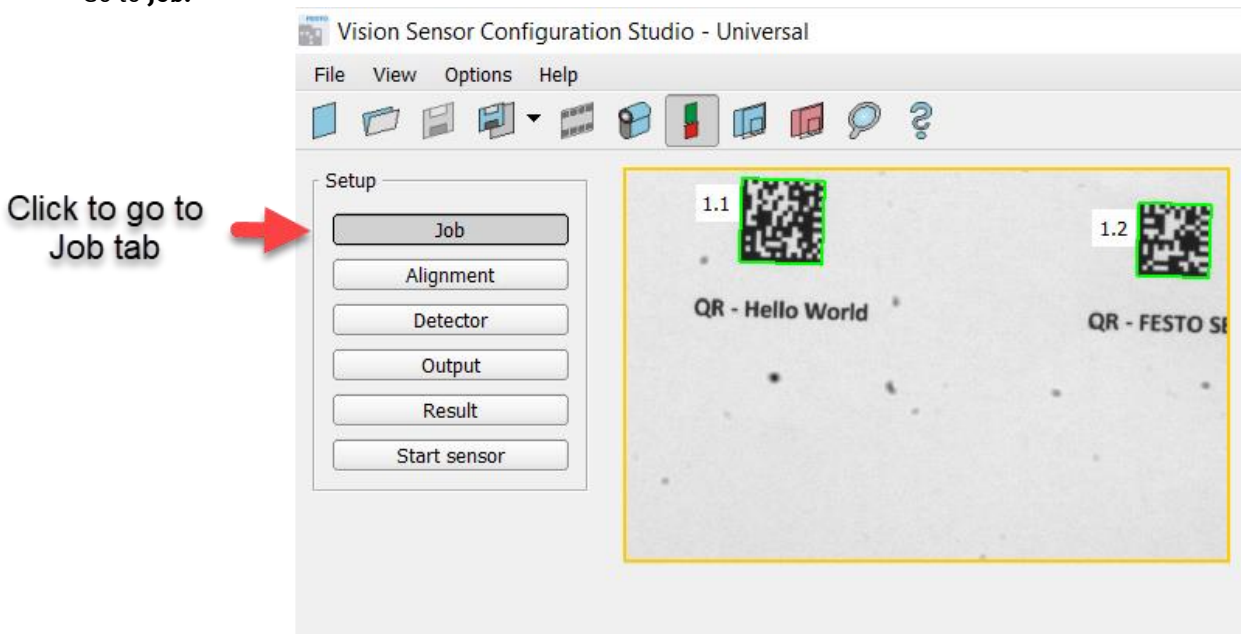
1. To demonstrate the FB execution with an example where we have to detect a barcode using a SBS vision sensor.
2. Configuration of the Vision Configuration Studio to detect a sample of barcode.
3. Configuration of telegram in Vision Configuration Studio.
4. Configuration of data to be communicated with the Siemens PLC in the telegram.
5. Creating watch tables in TIA portal to monitor the following :
    - Data sent from the SBS Vision Sensor in the telegram.
    - Status of the SBS Vision Sensor.
6. 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 Example configuration in Vision Configuration Studio.

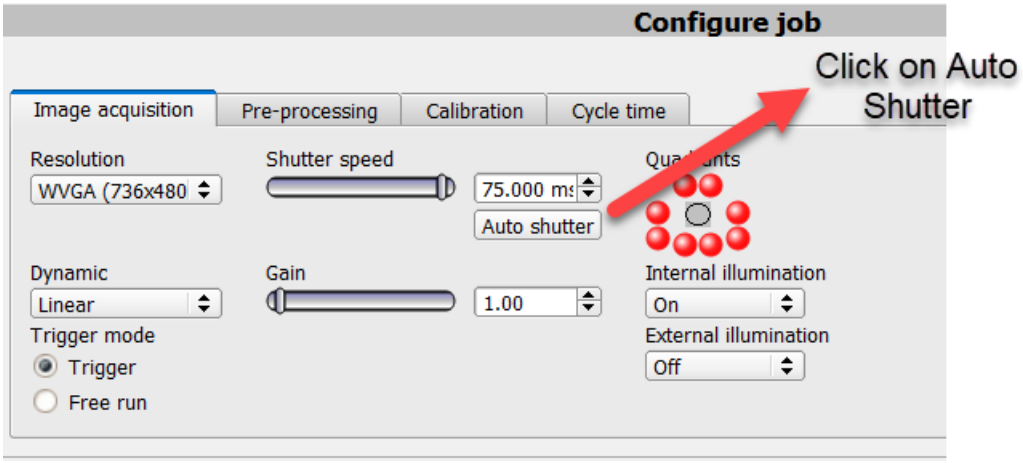
1. **Creating a new Job.**
    - Create a new Job named **Job1**. Go to **Job ›› New.**
    - Refer the image below to create a new job.



2. **Image Acquisition settings**.
   - Go to **Job.**

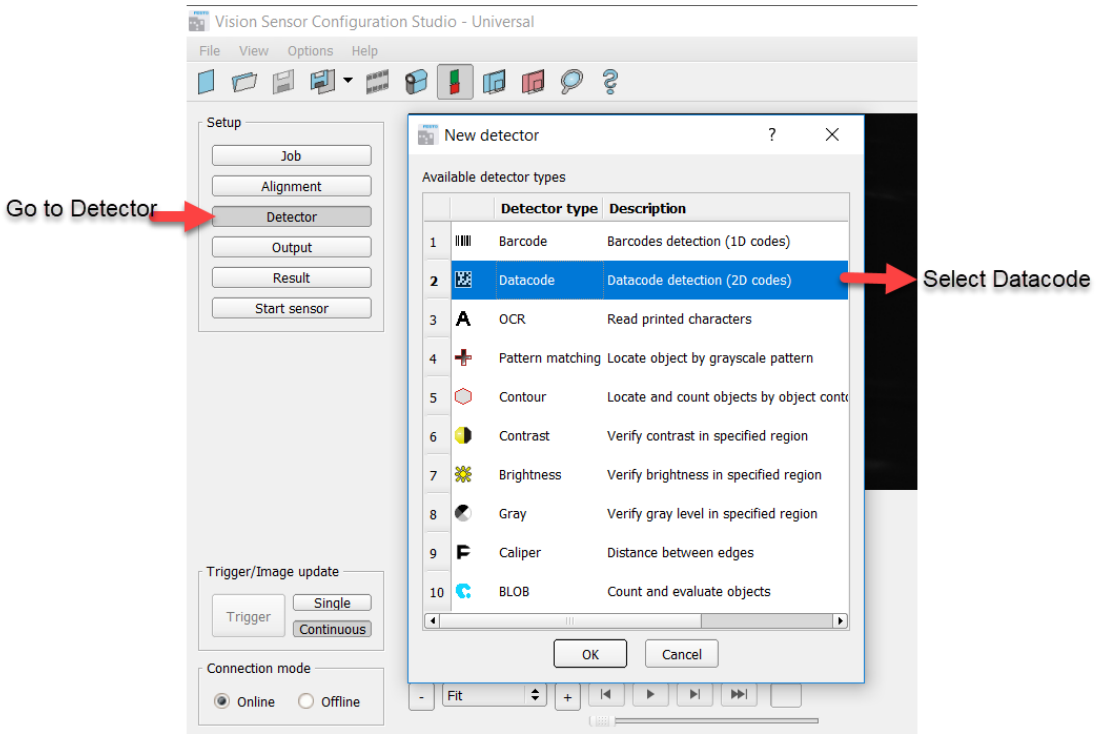


   - Click **Auto Shutter** under image acquisition tab.

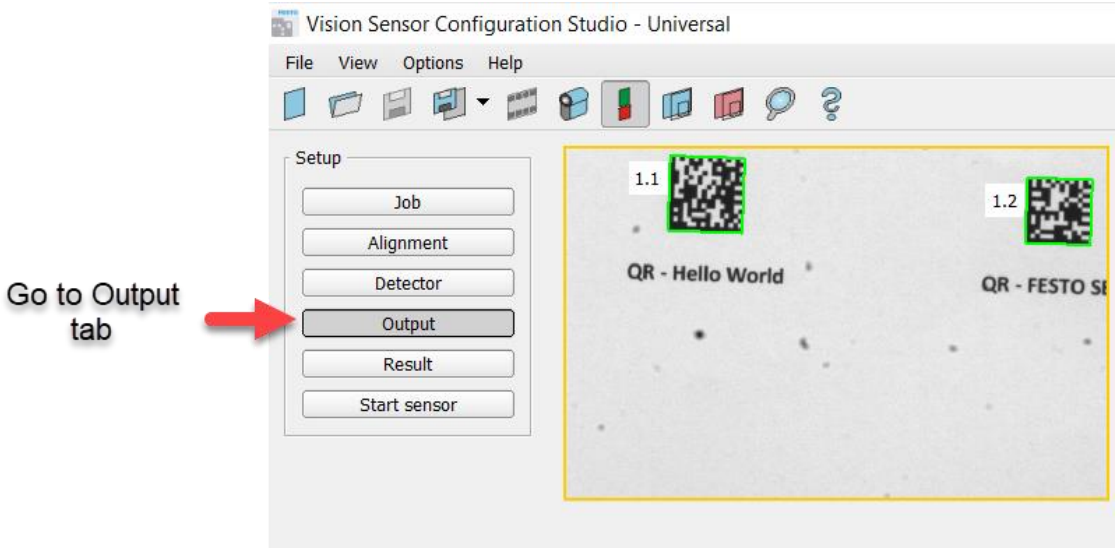


   - In the image window the bar code sample which the vision sensor focuses on will be displayed as shown below.

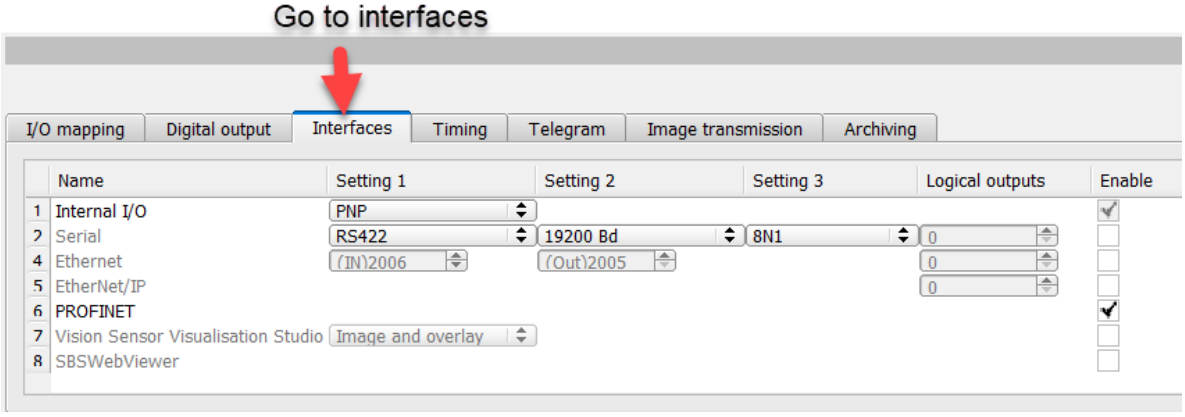
3. **Detector Selection.**
   - Go to **Detector** . Select **Data Code** as shown below.



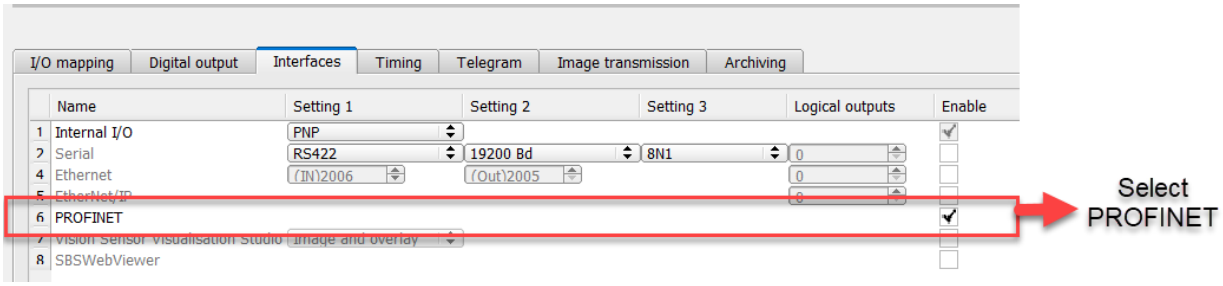
4. **Interface Settings**.
   - Go to **Output** tab.



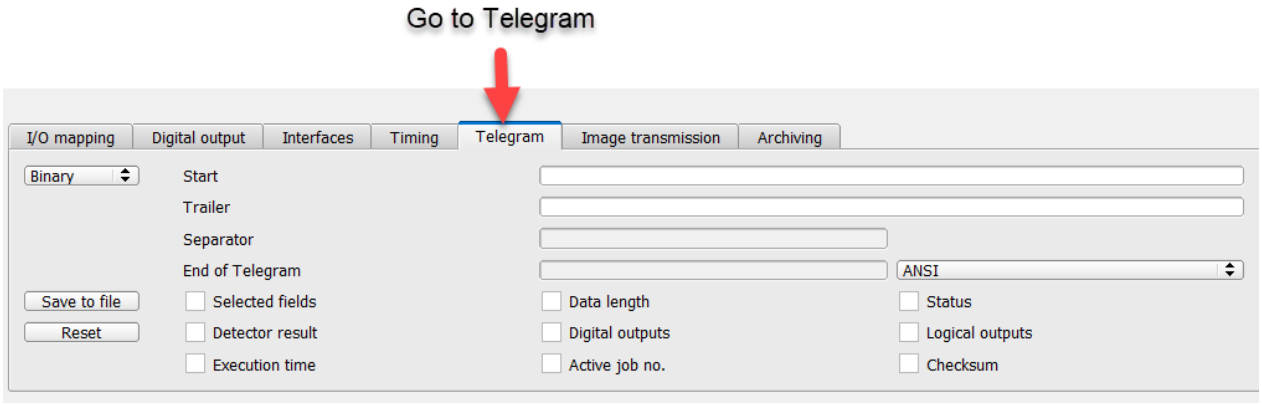
   - Under Output tab, Go to **Interfaces** as shown below.



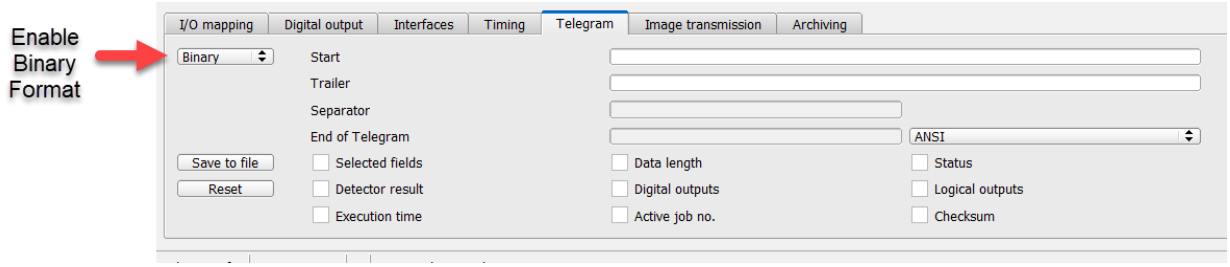
- Enable **PROFINET** under interfaces tab as shown below.



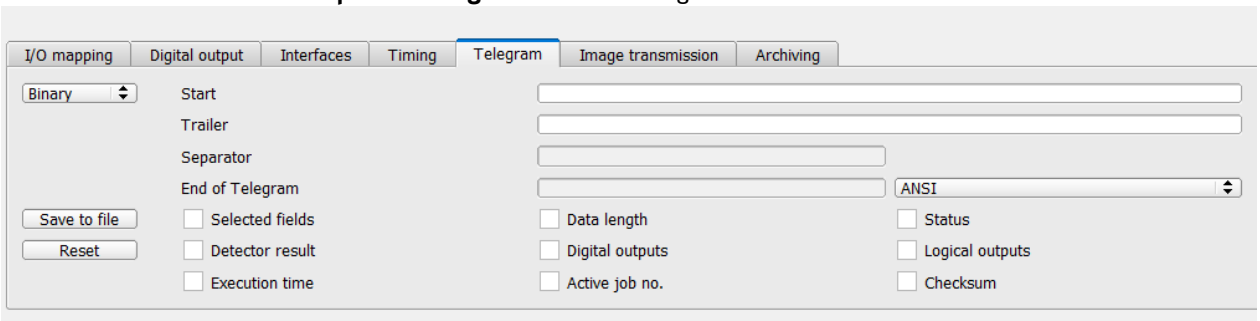
5. **Telegram settings.**
   - Under Output , Go to **Telegram**.



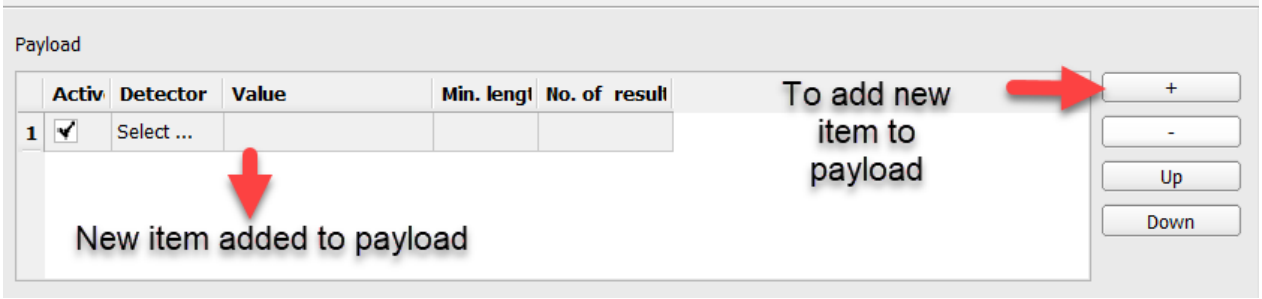
- Ensure that **Binary Format** is enabled.



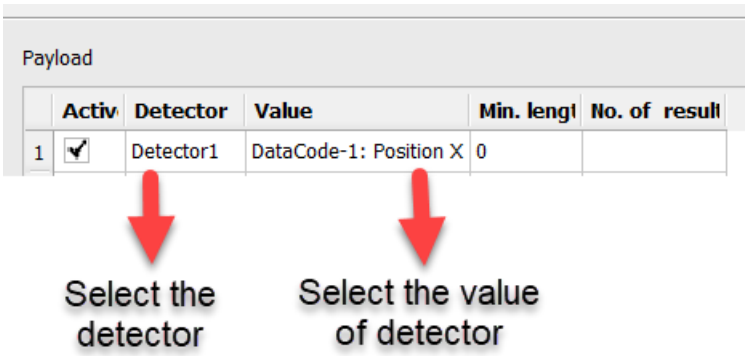
6. **Define the telegram to send the data needed to the PLC.**
   - Go to **Output ››› Telegram**. The following tab will be visualised.



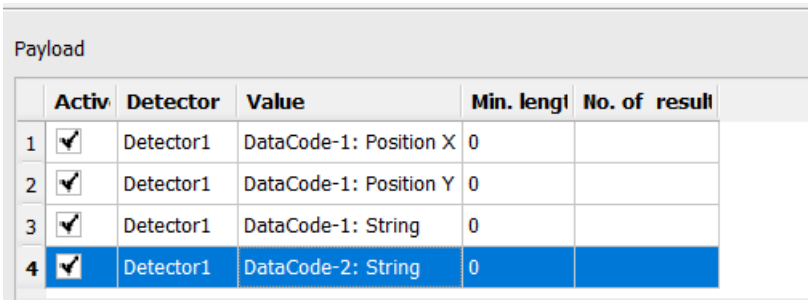
7. **Configure the payload to be sent to the PLC.**
   - Click + button to add a new payload data as shown below.



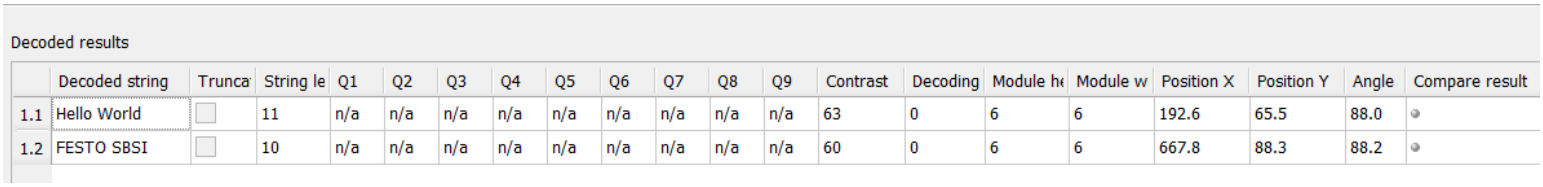
- Select the detector whose data has to be added to the payload.
- Select the value of the detector which has to be added to the payload.



- Repeat the process to add other Detector values also to the telegram.
- In our example we have considered the following data to the telegram.

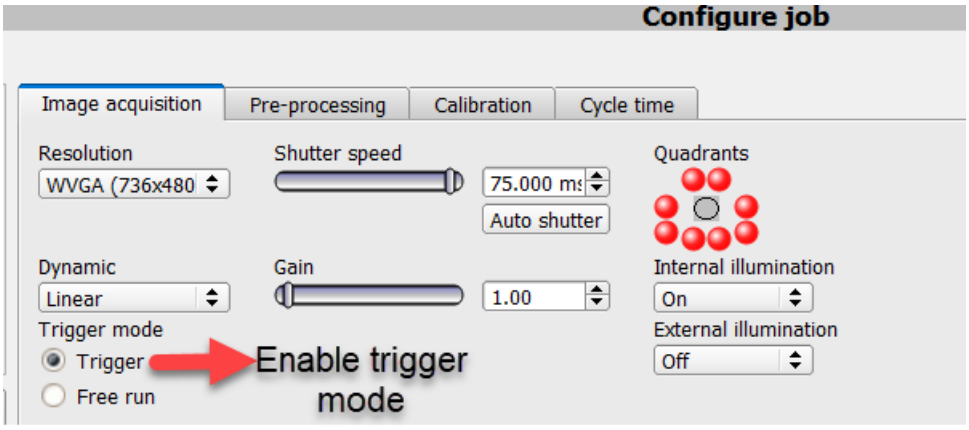


8. **Go to the Result tab to check the result data.**



9. **After the configurations are done ensure that the Vision Sensor is in Trigger Mode and not in Free Run mode.**

10. **Go to Job ›› Image Acquisition ››› Trigger Mode.**

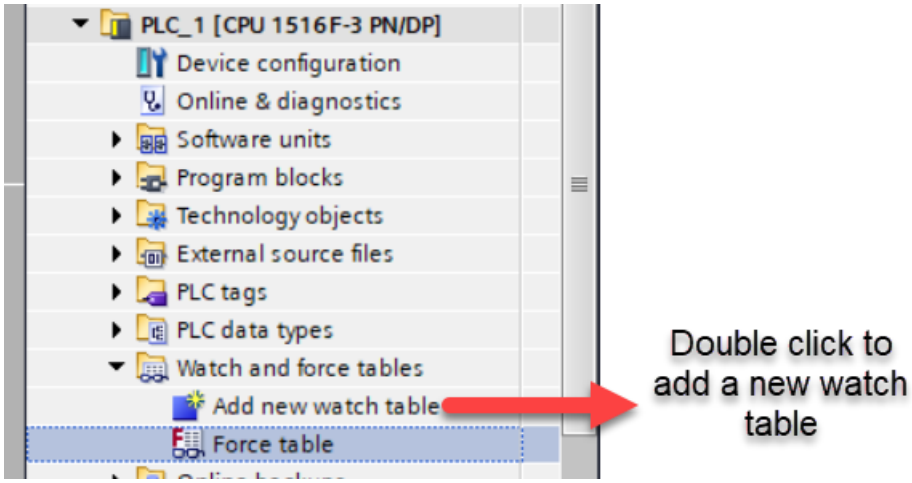


11. Click on **Start sensor** to download the configurations to the SBS vision sensor.

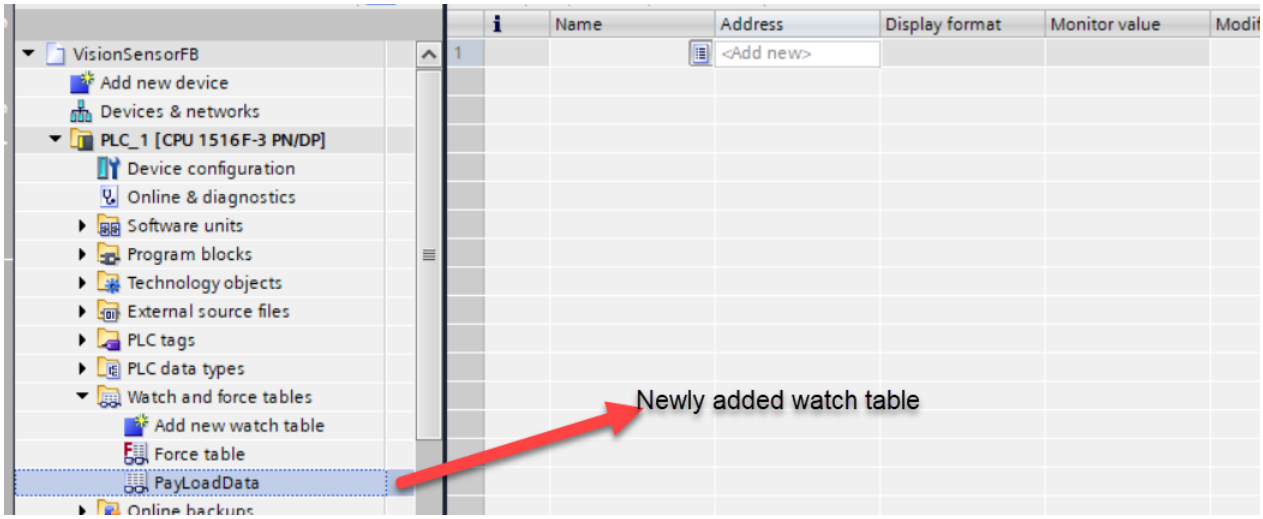
10.2 Example description in TIA Portal

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

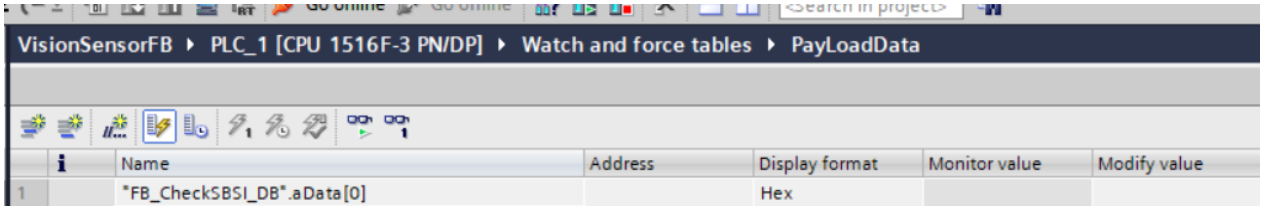
2. Create a watch table.



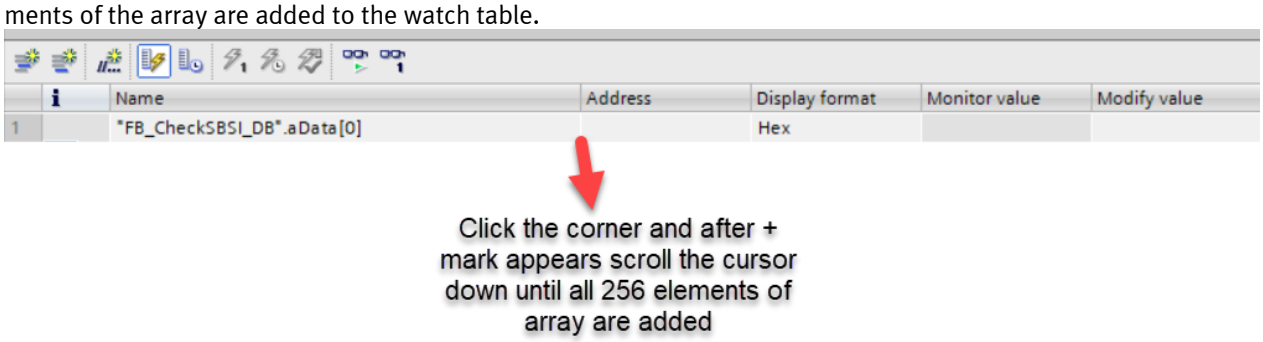
3. Give a name to the watch table. Here in our example we have named the watch table as PayLoad Data.



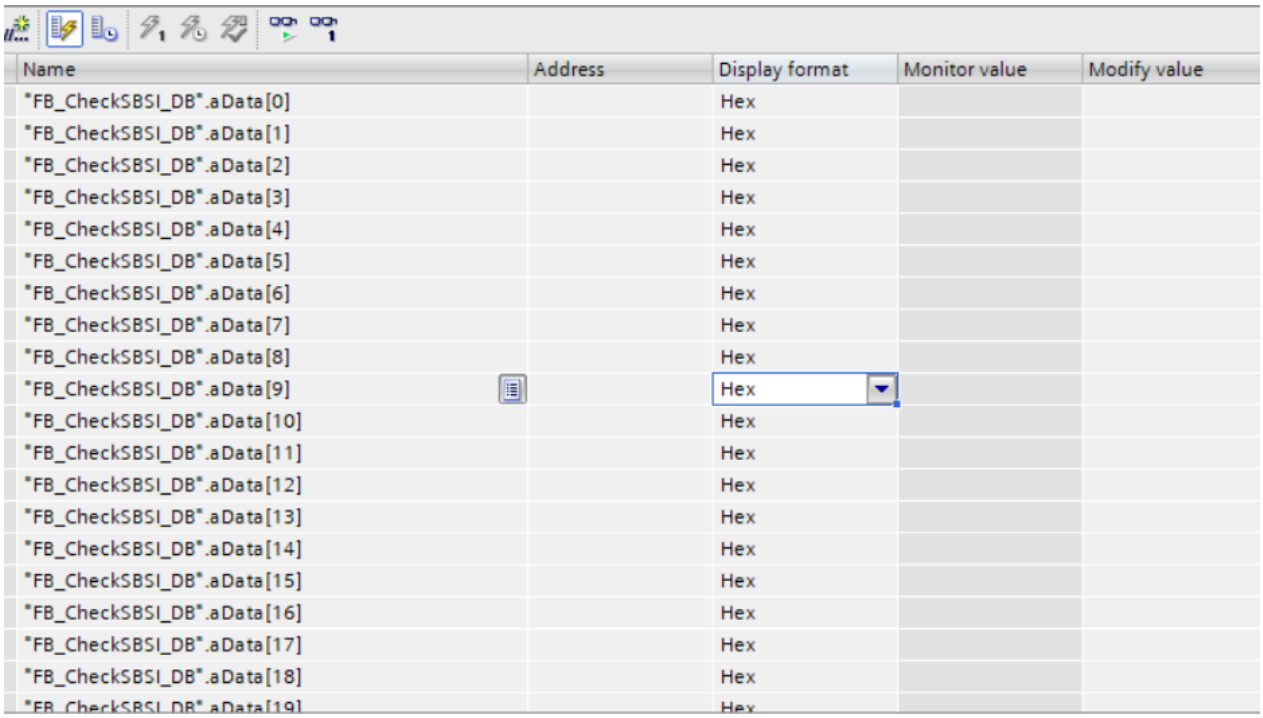
4. Add the **aData** output of the **FBCheckSBSI.** Add all the 258 array elements of aData to the watch table for testing purpose.
5. Add the first element of the array aData to the watch table as shown below.



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.



Name	Address	Display format	Monitor value	Modify value
"FB_CheckSBSI_DB".aData[0]		Hex		
"FB_CheckSBSI_DB".aData[1]		Hex		
"FB_CheckSBSI_DB".aData[2]		Hex		
"FB_CheckSBSI_DB".aData[3]		Hex		
"FB_CheckSBSI_DB".aData[4]		Hex		
"FB_CheckSBSI_DB".aData[5]		Hex		
"FB_CheckSBSI_DB".aData[6]		Hex		
"FB_CheckSBSI_DB".aData[7]		Hex		
"FB_CheckSBSI_DB".aData[8]		Hex		
"FB_CheckSBSI_DB".aData[9]		Hex		
"FB_CheckSBSI_DB".aData[10]		Hex		
"FB_CheckSBSI_DB".aData[11]		Hex		
"FB_CheckSBSI_DB".aData[12]		Hex		
"FB_CheckSBSI_DB".aData[13]		Hex		
"FB_CheckSBSI_DB".aData[14]		Hex		
"FB_CheckSBSI_DB".aData[15]		Hex		
"FB_CheckSBSI_DB".aData[16]		Hex		
"FB_CheckSBSI_DB".aData[17]		Hex		
"FB_CheckSBSI_DB".aData[18]		Hex		
"FB_CheckSBSI_DB".aData[19]		Hex		

8. The first 2 data in the payload are of the **Floating data type** (Double word). So configure the display format of Byte2 to Byte9 as **Hex**.

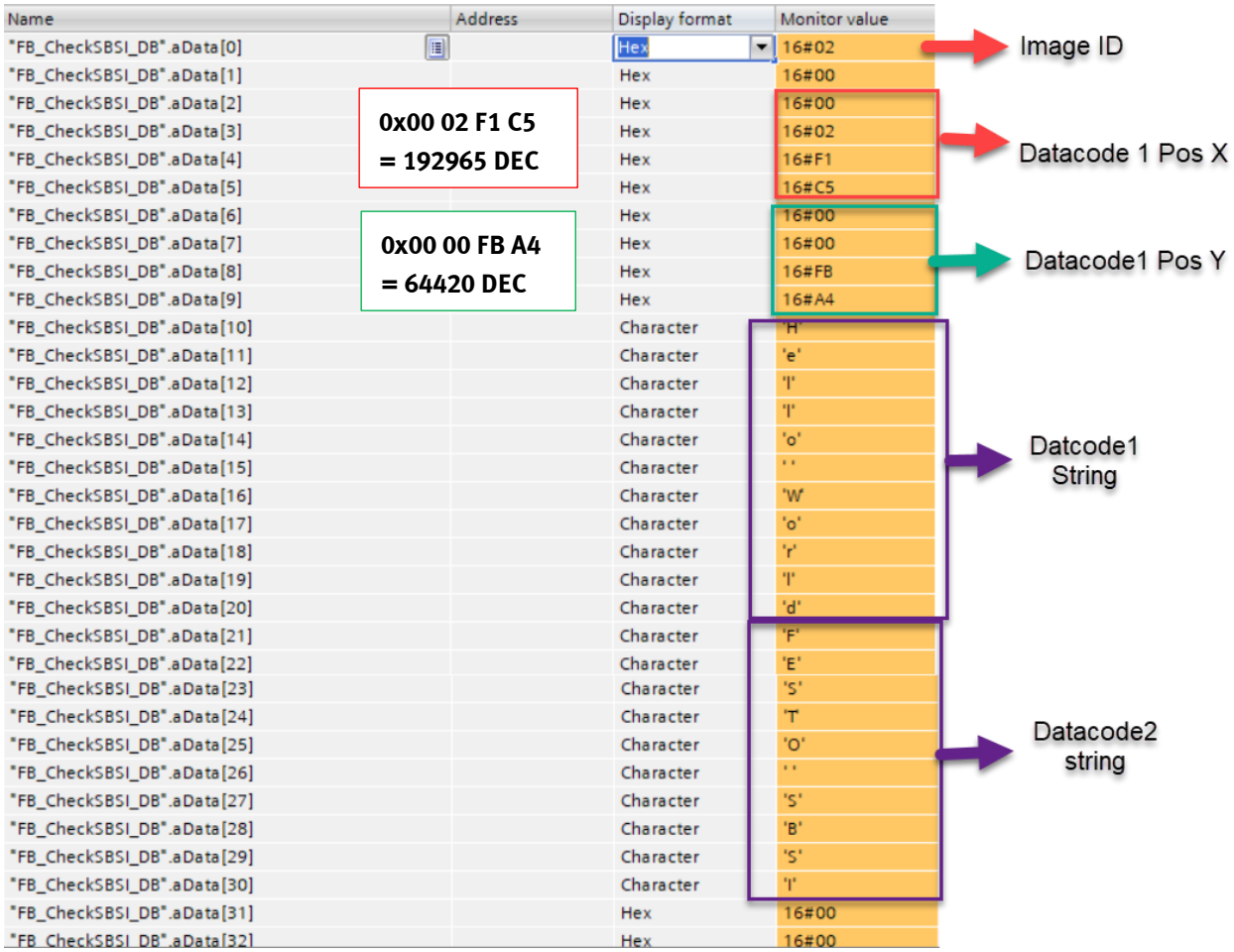
   The next 2 data in the payload are of **String** type . So configure the display format in Watch Table as **Char** from Byte10 onwards.

"FB_CheckSBSI_DB".aData[0]		Hex
"FB_CheckSBSI_DB".aData[1]		Hex
"FB_CheckSBSI_DB".aData[2]		Hex
"FB_CheckSBSI_DB".aData[3]		Hex
"FB_CheckSBSI_DB".aData[4]		Hex
"FB_CheckSBSI_DB".aData[5]		Hex
"FB_CheckSBSI_DB".aData[6]		Hex
"FB_CheckSBSI_DB".aData[7]		Hex
"FB_CheckSBSI_DB".aData[8]		Hex
"FB_CheckSBSI_DB".aData[9]		Hex
"FB_CheckSBSI_DB".aData[10]		Character
"FB_CheckSBSI_DB".aData[11]		Character
"FB_CheckSBSI_DB".aData[12]		Character
"FB_CheckSBSI_DB".aData[13]		Character
"FB_CheckSBSI_DB".aData[14]		Character
"FB_CheckSBSI_DB".aData[15]		Character
"FB_CheckSBSI_DB".aData[16]		Character
"FB_CheckSBSI_DB".aData[17]		Character
"FB_CheckSBSI_DB".aData[18]		Character
"FB_CheckSBSI_DB".aData[19]		Character
"FB_CheckSBSI_DB".aData[20]		Character
"FB_CheckSBSI_DB".aData[21]		Character
"FB_CheckSBSI_DB".aData[22]		Character
"FB_CheckSBSI_DB".aData[23]		Character
"FB_CheckSBSI_DB".aData[24]		Character
"FB_CheckSBSI_DB".aData[25]		Character
"FB_CheckSBSI_DB".aData[26]		Character
"FB_CheckSBSI_DB".aData[27]		Character
"FB_CheckSBSI_DB".aData[28]		Character
"FB_CheckSBSI_DB".aData[29]		Character
"FB_CheckSBSI_DB".aData[30]		Character

9. Enter **iSBSICtrl_HWID, iSBSIStat_HWID, iSBSIData_HWID.** Refer **Chapter 6.5** to identify the hardware ID's of Ctrl , Stat and Data modules of SBSI.

10. Enable the Function Block. This is done by making **xEnable = TRUE**. Show how to toggle at least one variable. Through Watch table, for example.

11. Toggle input **xTrigger = 1.**
Observe the data received in the aData output of the FB. Monitor the watch table output.



- The online payload data values of TIA portal match with the Results of the vision configuration studio (**Refer Step 8 in Section 10.1**).

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

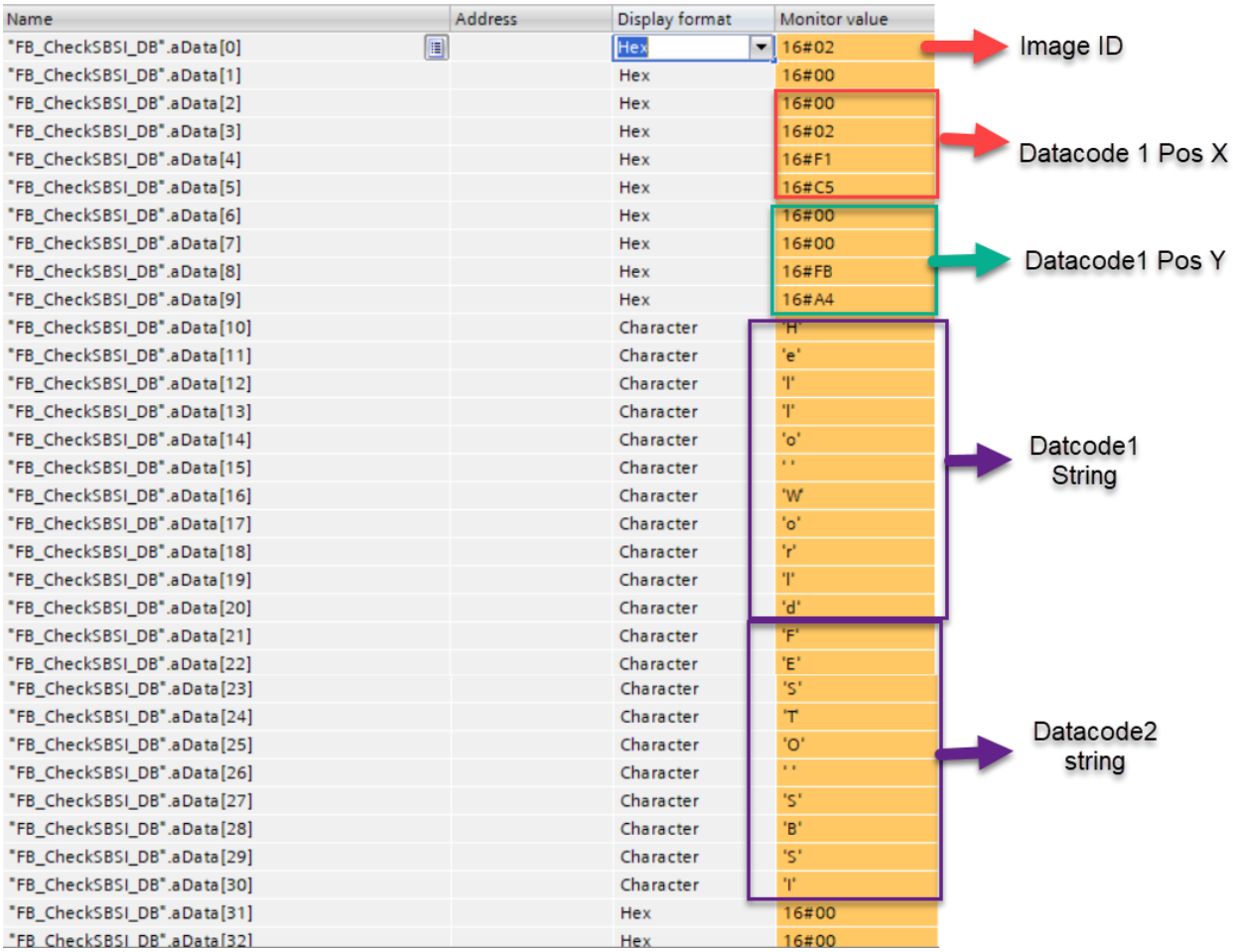
This chapter explains how the payload configured in Vision Configuration Studio and the data appearing in TIA portal(**aData output of the FB**) are mapped with each other.

1. **Consider we have the following payload configuration in Vision Sensor Configuration Studio.**

Payload

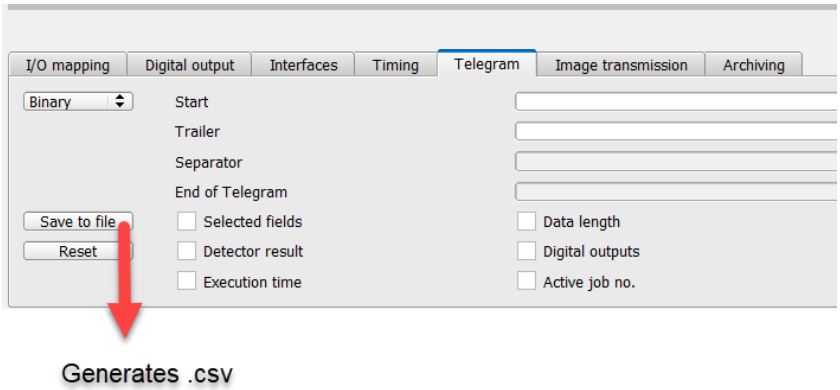
	Activ	Detector	Value	Min. lengt	No. of result
1	☑	Detector1	DataCode-1: Position X	0	
2	☑	Detector1	DataCode-1: Position Y	0	
3	☑	Detector1	DataCode-1: String	0	
4	☑	Detector1	DataCode-2: String	0	

2. **TIA Portal online values of the payload data received from SBS Vision sensor.**

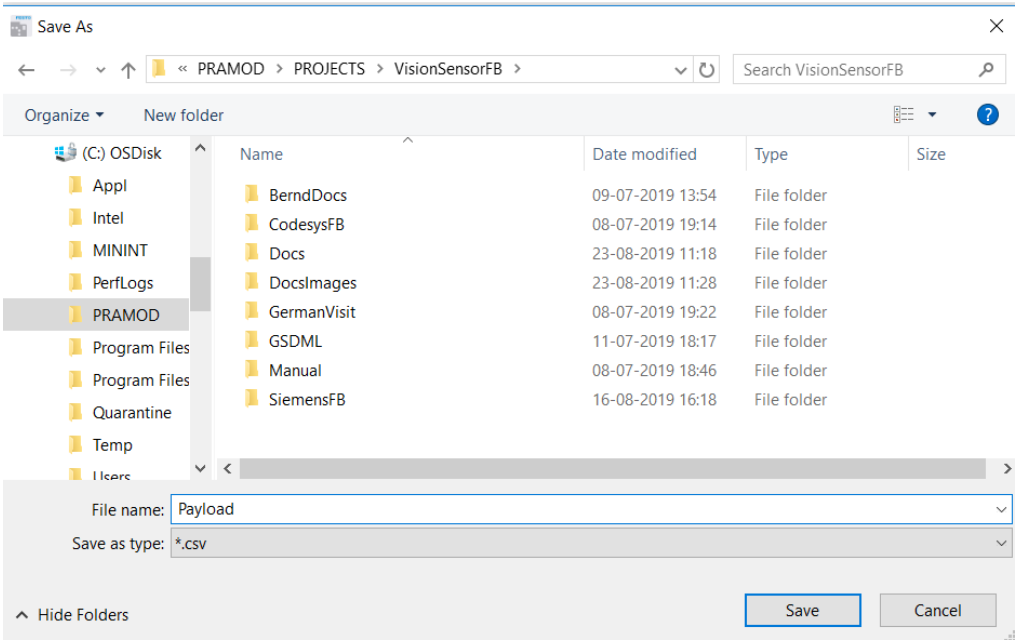


3. **Setup of payload data in Vision Configuration studio as .csv file.**

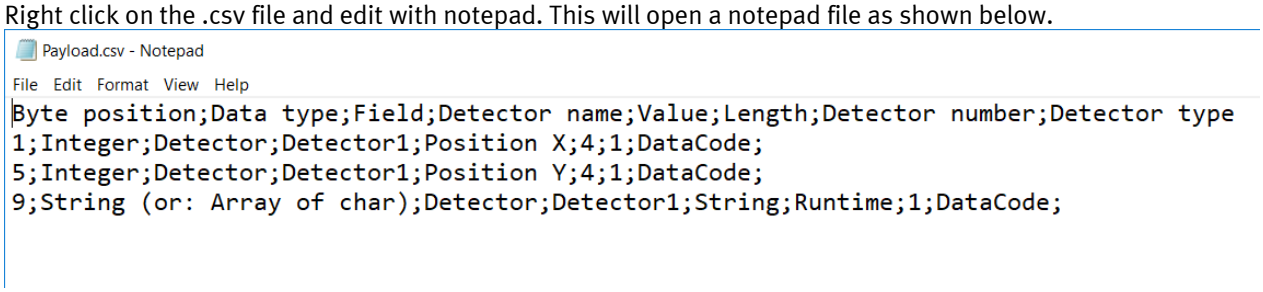
   - **Go to Output ››› Telegram ››› Save to File.**



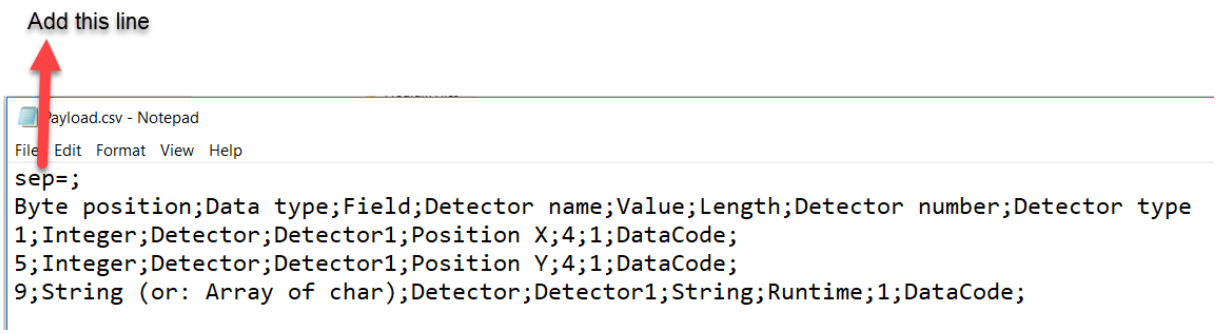
- Select the location where the .csv file has to be saved.



- Click on **Save** to save the file.
- Right click on the .csv file and edit with notepad. This will open a notepad file as shown below.



- Edit the Notepad file by adding **sep=;.**
  Now the edited notepad file will look as shown below.

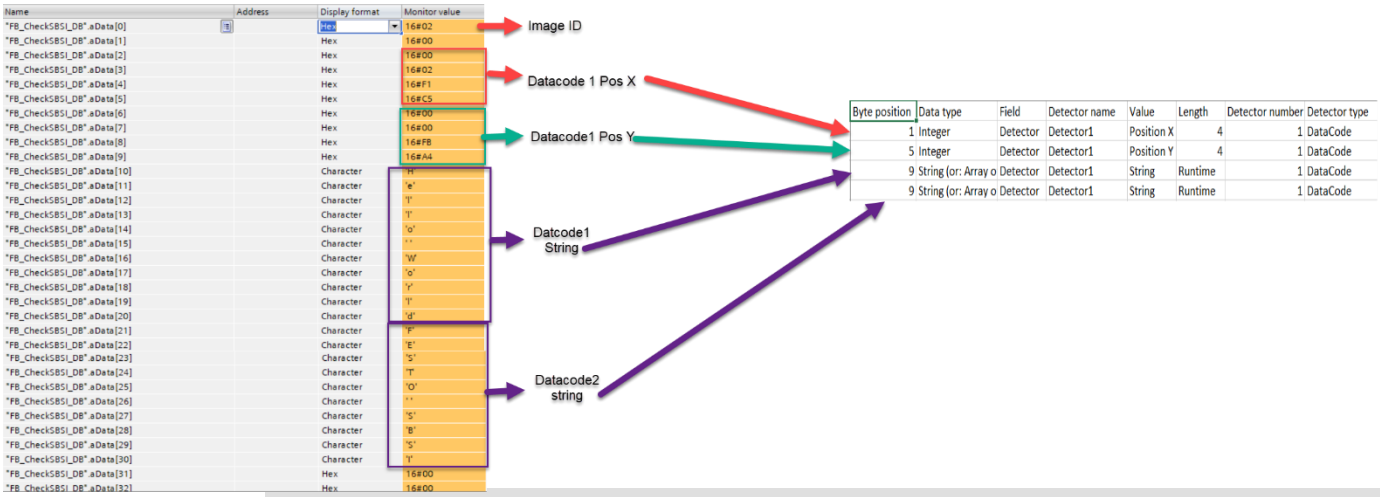


- **SAVE** and close the file.
- Now open the excel sheet. It will have the following view.

- The payload data can be seen in the above excel sheet. The order in which the data was configured in Vision sensor configuration studio in the same order data is displayed in the excel sheet.

Byte position	Data type	Field	Detector name	Value	Length	Detector number	Detector type
1	Integer	Detector	Detector1	Position X	4	1	DataCode
5	Integer	Detector	Detector1	Position Y	4	1	DataCode
9	String (or: Array o	Detector	Detector1	String	Runtime	1	DataCode
9	String (or: Array o	Detector	Detector1	String	Runtime	1	DataCode

- The following image shows how the payload data between Vision Sensor configuration studio and TIA portal.



NOTE

- **Wherever string data is configured in the payload its always better to add them at the end. This is because the string length can be variable and if its put in the middle it can create difficulty in mapping.**

It is recommended when transferring multiple strings to add the String Length into the Payload.

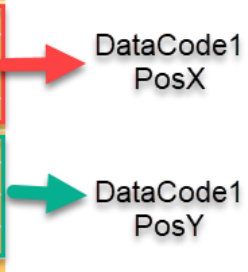
10.4 Payload Data conversion in TIA Portal.

- The payload data which is of the **Floating data type** appears as an array of 4 bytes in the Siemens Profinet data. So these array of bytes must be converted into a single variable of the type **DWORD** type.
- The below section explains how these array of bytes are converted into a single **DWORD** value.

10.4.1 Converting floating values of payload data into a single Double Word.

1. In our example following payload data will appear in the Profinet Data as shown below.

"FB_CheckSBSI_DB".aData[0]	Hex	16#03
"FB_CheckSBSI_DB".aData[1]	Hex	16#00
"FB_CheckSBSI_DB".aData[2]	Hex	16#00
"FB_CheckSBSI_DB".aData[3]	Hex	16#02
"FB_CheckSBSI_DB".aData[4]	Hex	16#EF
"FB_CheckSBSI_DB".aData[5]	Hex	16#BF
"FB_CheckSBSI_DB".aData[6]	Hex	16#00
"FB_CheckSBSI_DB".aData[7]	Hex	16#00
"FB_CheckSBSI_DB".aData[8]	Hex	16#FC
"FB_CheckSBSI_DB".aData[9]	Hex	16#AC
"FB_CheckSBSI_DB".aData[10]	Character	'H'
"FB_CheckSBSI_DB".aData[11]	Character	'e'
"FB_CheckSBSI_DB".aData[12]	Character	'l'
"FB_CheckSBSI_DB".aData[13]	Character	'l'
"FB_CheckSBSI_DB".aData[14]	Character	'o'
"FB_CheckSBSI_DB".aData[15]	Character	' '
"FB_CheckSBSI_DB".aData[16]	Character	'W'
"FB_CheckSBSI_DB".aData[17]	Character	'o'
"FB_CheckSBSI_DB".aData[18]	Character	'r'
"FB_CheckSBSI_DB".aData[19]	Character	'l'
"FB_CheckSBSI_DB".aData[20]	Character	'd'
"FB_CheckSBSI_DB".aData[21]	Character	'F'
"FB_CheckSBSI_DB".aData[22]	Character	'E'
"FB_CheckSBSI_DB".aData[23]	Character	'S'
"FB_CheckSBSI_DB".aData[24]	Character	'T'
"FB_CheckSBSI_DB".aData[25]	Character	'O'
"FB_CheckSBSI_DB".aData[26]	Character	' '
"FB_CheckSBSI_DB".aData[27]	Character	'S'
"FB_CheckSBSI_DB".aData[28]	Character	'B'
"FB_CheckSBSI_DB".aData[29]	Character	'S'
"FB_CheckSBSI_DB".aData[30]	Character	'I'



2. Here in our example 2 data's namely **DataCode1 Pos X and DataCode1 Pos Y** are of the Floating Data type.
3. **DataCode1 Pos X** value appears as an array of 4 Bytes starting from aData[2] to aData[5]. So this array of bytes has to be converted into a single DWORD.
4. **DataCode1 Pos Y** value appears as an array of 4 Bytes starting from aData[6] to aData[9]. So this array of bytes has to be converted into a single DWORD.
5. Under Global variables create 2 variables namely **DataCode1.PosX** and **DataCode1.PosY** of the type **DINT**.

Data_block_1

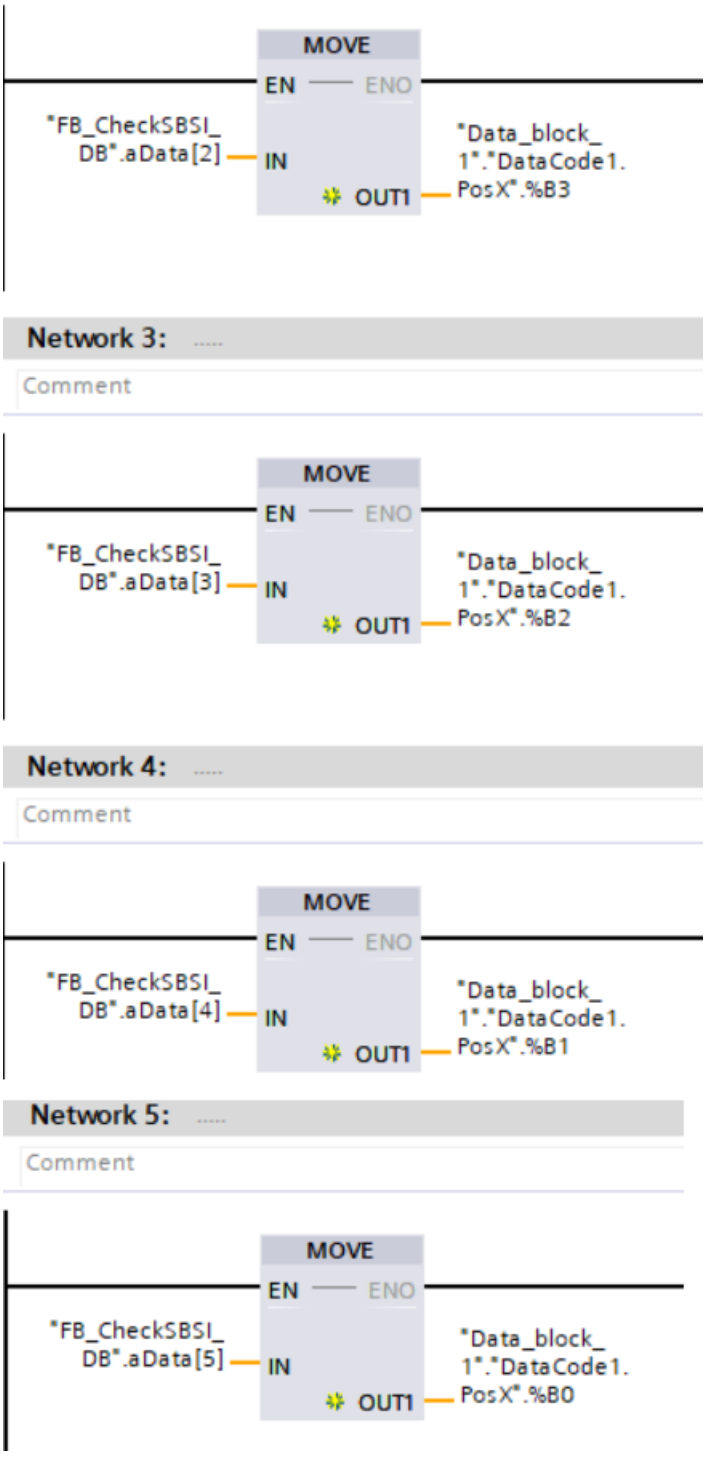
Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
Static							
DataCode1.PosX	DInt	0					
DataCode1.PosY	DInt	0					

6. The array of bytes in the payload data (aData[2] to aData[5] will be mapped to the DINT variable **DataCode1.PosX** as shown below.
   - aData[2] ---- Byte 3 of **DataCode1.PosX**
   - aData[3] ---- Byte 2 of **DataCode1.PosX**
   - aData[4] ---- Byte 1 of **DataCode1.PosX**
   - aData[5] ---- Byte 0 of **DataCode1.PosX**

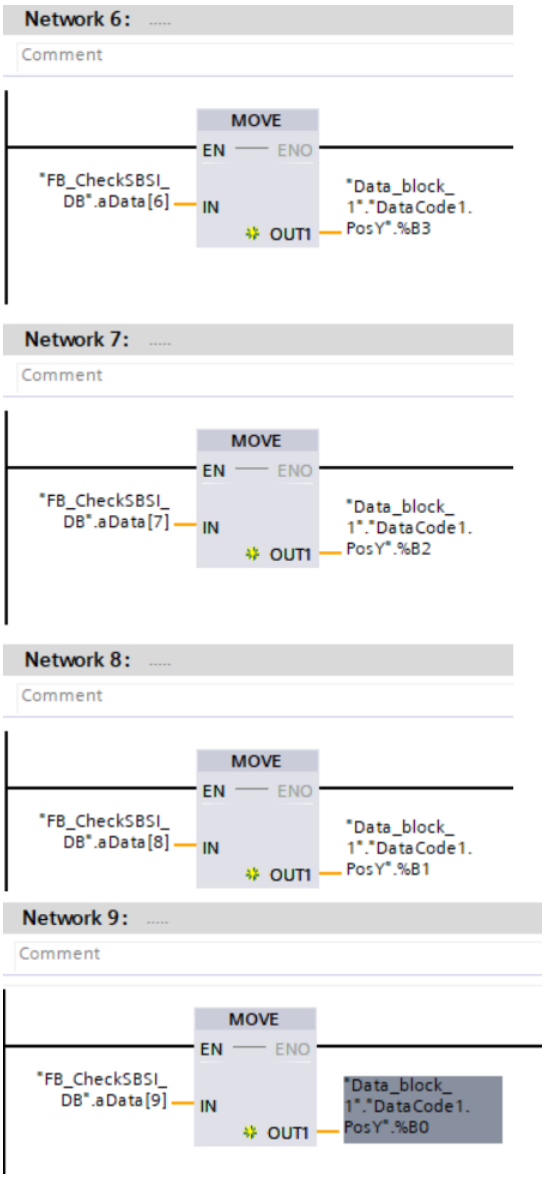
7. The array of bytes in the payload data (aData[2] to aData[5] will be mapped to the DINT variable **DataCode1.PosX** as shown below.
   - aData[6] ---- Byte 3 of **DataCode1.PosY**
   - aData[7] ---- Byte 2 of **DataCode1.PosY**
   - aData[8] ---- Byte 1 of **DataCode1.PosY**
   - aData[9] ---- Byte 0 of **DataCode1.PosY**

8. Write a logic to do the mapping as shown in **Step 6** and **Step 7.**

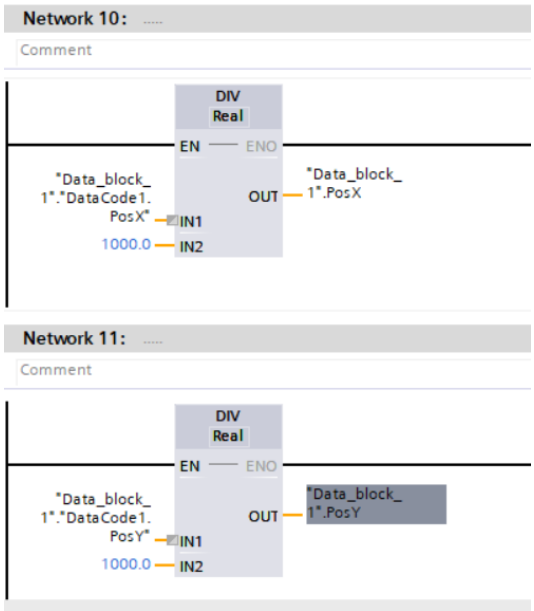
Mapping of DataCode1.PosX



Mapping of DataCode1.PosX



9. The payload data values are multiplied by a **x1000** factor. So if we have to get the value which is exact to Vision Configuration studio we have to divide by **1000**.
   Write a logic for this scaling as shown below.



10. Add the variables DataCode1.PosX and DataCode1.PosY to Watch table.

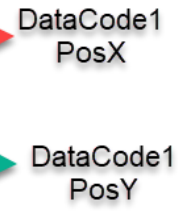
"Data_block_1"."DataCode1.PosX"		DEC
"Data_block_1"."DataCode1.PosY"		DEC

11. Download the project and Go Online.

12. Once the Function Block is executed observe the values of the watch table to monitor the payload data.

13. In our example the payload data appears in the watch table as shown below.

"FB_CheckSBSI_DB".aData[0]		Hex	16#06
"FB_CheckSBSI_DB".aData[1]		Hex	16#00
"FB_CheckSBSI_DB".aData[2]		Hex	16#00
"FB_CheckSBSI_DB".aData[3]		Hex	16#02
"FB_CheckSBSI_DB".aData[4]		Hex	16#F1
"FB_CheckSBSI_DB".aData[5]		Hex	16#18
"FB_CheckSBSI_DB".aData[6]		Hex	16#00
"FB_CheckSBSI_DB".aData[7]		Hex	16#00
"FB_CheckSBSI_DB".aData[8]		Hex	16#FC
"FB_CheckSBSI_DB".aData[9]		Hex	16#57
"FB_CheckSBSI_DB".aData[10]		Character	'H'
"FB_CheckSBSI_DB".aData[11]		Character	'e'
"FB_CheckSBSI_DB".aData[12]		Character	'l'
"FB_CheckSBSI_DB".aData[13]		Character	'l'
"FB_CheckSBSI_DB".aData[14]		Character	'o'
"FB_CheckSBSI_DB".aData[15]		Character	' '
"FB_CheckSBSI_DB".aData[16]		Character	'W'
"FB_CheckSBSI_DB".aData[17]		Character	'o'
"FB_CheckSBSI_DB".aData[18]		Character	'r'
"FB_CheckSBSI_DB".aData[19]		Character	'l'
"FB_CheckSBSI_DB".aData[20]		Character	'd'
"FB_CheckSBSI_DB".aData[21]		Character	'F'
"FB_CheckSBSI_DB".aData[22]		Character	'E'
"FB_CheckSBSI_DB".aData[23]		Character	'S'
"FB_CheckSBSI_DB".aData[24]		Character	'T'
"FB_CheckSBSI_DB".aData[25]		Character	'O'
"FB_CheckSBSI_DB".aData[26]		Character	' '
"FB_CheckSBSI_DB".aData[27]		Character	'S'
"FB_CheckSBSI_DB".aData[28]		Character	'B'
"FB_CheckSBSI_DB".aData[29]		Character	'S'
"FB_CheckSBSI_DB".aData[30]		Character	'I'



14. The PosX and PosY value of Datacode1 coming as an array of bytes is grouped to a DINT variable as explained before . This value in Watch table will appear as shown below.

"Data_block_1"."DataCode1.PosX"		DEC	192792
"Data_block_1"."DataCode1.PosY"		DEC	64599

15. To get the same value as in Vision Configuration studio we have to divide the above value by 1000. The scaled values will be displayed in watch table as shown below.

"Data_block_1".PosX		Floating-point nu...	192.792
"Data_block_1".PosY		Floating-point nu...	64.599

16. This value matches with the PosX and PosY value obtained in Vision Configuration studio.