

S15C-UI Process Data Function

January 9th, 2026

This document covers the installation and use of a function for Siemens' TIA Portal software package. This function handles cyclic IO-Link Process Data In from a Banner S15C-UI sensor via an IO-Link Master to a Siemens PLC. The function covers parsing and display of the S15C-UI sensor Process Data In.

Components

Banner S15C R45C Library v16.zal16

There are two methods for the process data. The first is used when creating a connection to Banner's IO-Link masters. The second set of instructions are for systems using other manufacturer's IO-Link masters.

Installation Instructions

1. Open a project.
2. Go to the Open Global Library option in the Libraries tab in TIA Portal v16 or greater.



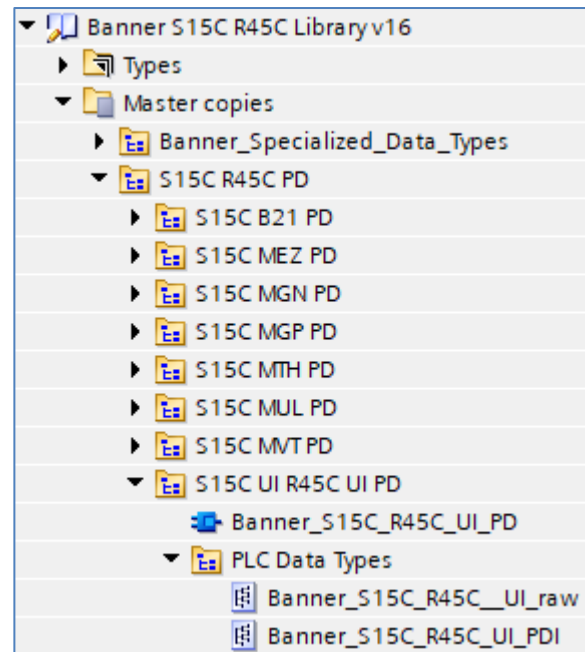
3. Switch the “Files of type” to Compressed libraries. Go to the location of the compressed library.
4. Press the Open button and the library will be uncompressed and opened.
5. The library is now accessible in the Libraries tab in v16 or greater.

Setup of S15C-UI with a Banner DXMR

1. Go to Device and Networks to configure the DXMR. Add the DXMR if it has yet to be added to the system.
2. Open the IO-Link Generic Devices and select the proper module. The 4/4 byte option has been selected for port 1. Make note of the I address for the Slot 2 which represents Port 1. Slot 2 starts are 10. The other number needed is I14. The data for the port start at that point (I14). The previous four bytes represents Port Status, Process Data In Size, and Process Data Out Size.

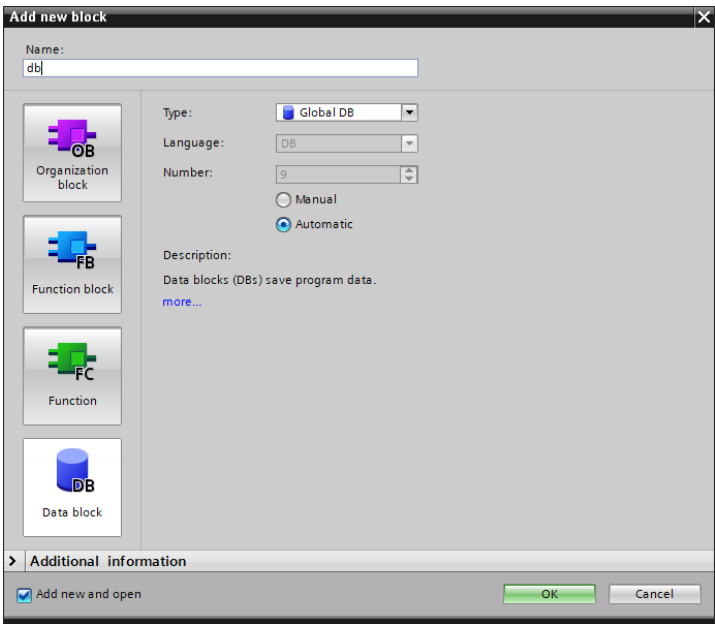
| Module | Rack | Slot | I address | Q address | Type |
|-------------------------------------|------|------|-----------|-----------|-----------------------------------|
| ▼ dxm | 0 | 0 | | | 1-port Device |
| ▶ Interface | 0 | 0 X1 | | | dxm |
| Banner IO-Link Master Info_1 | 0 | 1 | 1...9 | | Banner IO-Link Master Info |
| IO-Link In/Out 4/ 4 Byte + Status_1 | 0 | 2 | 10...17 | 1...18 | IO-Link In/Out 4/ 4 Byte + Status |

3. Drag the Banner_S15C_R45C_UI_raw and Banner_S15C_R45C_PDI to the PLC Data Types area under your PLC.
4. Drag the Banner_S15C_UI_PD to the Program Blocks area.
5. Drag the necessary tags from Banner_Specialized_Data_Types. The tag used in this example is "Banner_4In". These tags represent the full raw process data along with port status information.
6. Go to PLC Tags. Create two tags. One of the tags creates the full data structure while the second represents the raw Process Data from the IO-Link Master. In this example, Tag table_1 was created, the tag "S15C UI IOLM1 01 PDI" was created using a Data Type of "Banner_4In". This naming convention calls out the type of sensor in question as well as the specific IO-Link Master and port number where the sensor is connected. A different IO-Link Master might be named IOLM2 or IOLM3, for instance, and other specific sensors may be connected to different port numbers. The "I" address found in step 2 is tied to this new tag. The second tag uses "S15C_UI_IOLM1_01_inRaw". These are the tags that will be used in the Function block.



| Name | Data type | Address |
|--------------------------|----------------------|---------|
| ▶ S15C UI IOLM1 01 PDI | "Banner_4In" | %I10.0 |
| ▶ S15C UI IOLM1 01 inRaw | "Banner_S15C_UI_raw" | %I14.0 |

7. Go to Program blocks. Add a new Data block if necessary. In this example the new data block is named “db”.



8. In the new data block, create a new tag to represent the parsed Process Data In for our S15C-MUI. The tag name again calls out the type of sensor, the IO-Link Master, and the port number. Use the data type “Banner_S15C_UI_PDI” for the new tag.

| Name | Data type |
|--------------------|----------------------|
| Static | |
| S15C UI IOLM 01 PD | "Banner_S15C_UI_PDI" |
| Measurement | Real |
| Measurement Scale | UDInt |

9. Add the “Banner_S15C_UI_PD” function to an OB ladder. Link the “PDI” to the raw Process Data variable from step 5. Link the “S15C UI IOLM 01 PD” to the parsed Process Data variable from step 7.

The last variable, “PDI Config”, allows the function to correctly interpret the Process Data.

There are two ways to achieve this goal. We can simply type in the correct number for “PDI Config” (see Fig. 1), or we can link this function to the S15C-UI Data Function Block (see Fig. 2). See Appendix A for more information about S15C-UI Process Data.

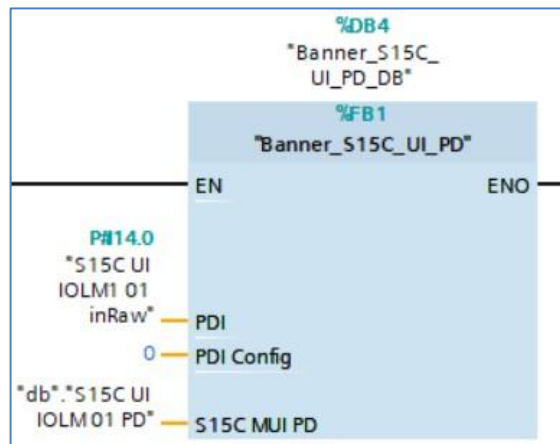


Figure 1: Hand typed correct numbers for PDI Config

NOTE: if you type in the incorrect number, you will get incorrectly displayed Process Data information.

PDI Config: the options here are “0” (Analog Value) and “1” (Digital Measurement). The default is “0”.

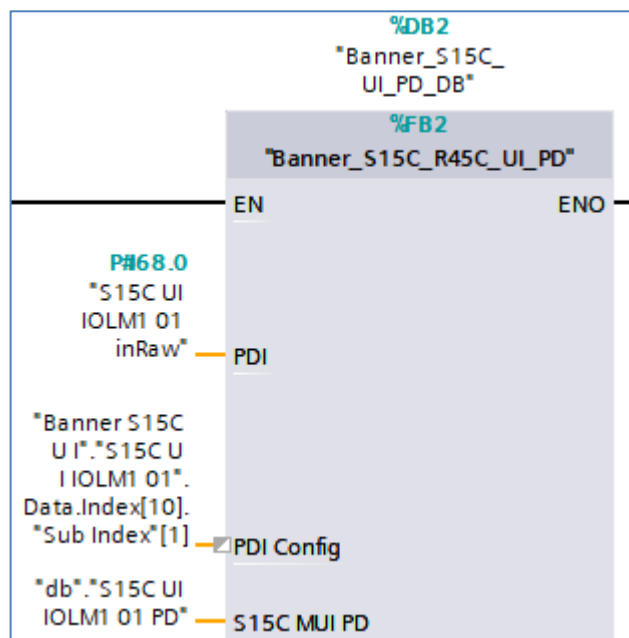
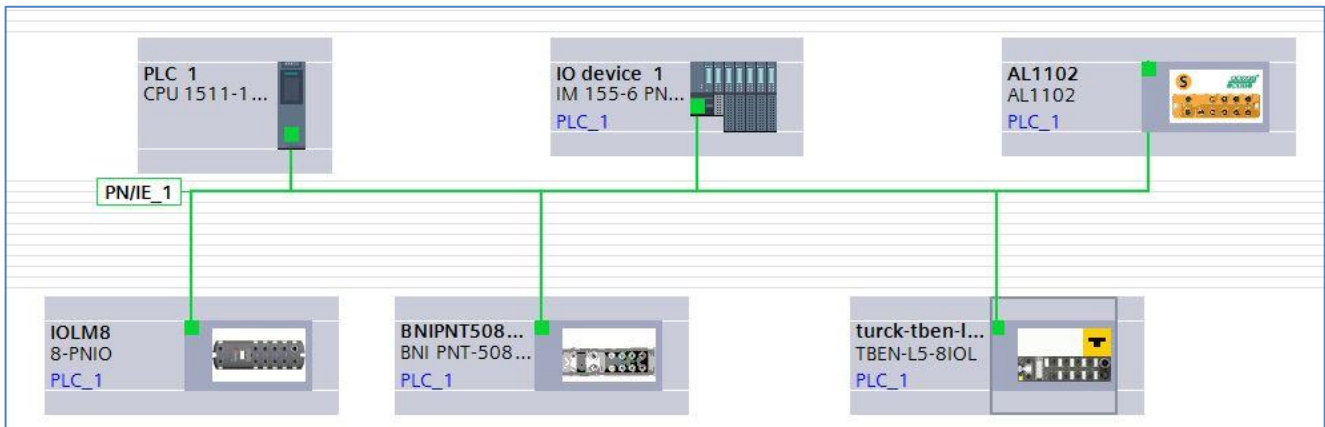
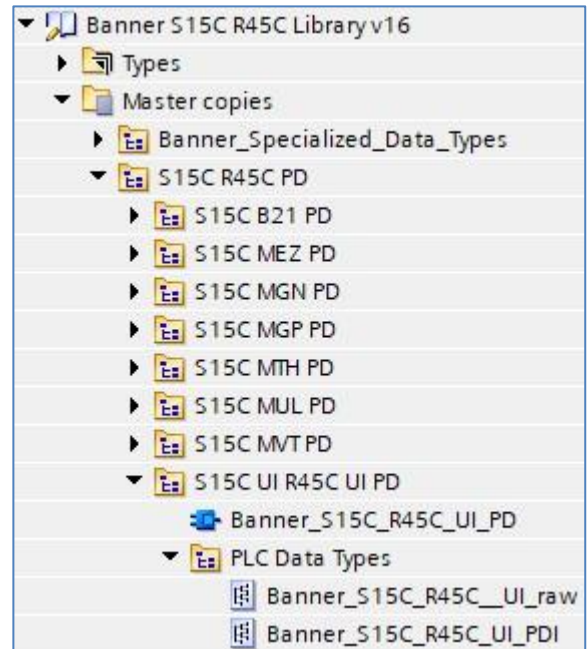


Figure 2: Linking Operational Mode variable to S15C-UI Parameter Data Function Block

10. Process Data setup is complete.
11. Compile and download the configuration to the PLC, then go online. Open the “db” data block and click Monitor all. You should see parsed S15C UI Process Data In.


Setup of S15C-MUI with other IO-Link Masters

1. The Banner IO-Link Library will now be in the Global Library List. Expand the Master copies section.
2. Drag Banner_S15C_UI_PD to the Program Blocks area under your PLC.
3. Drag Banner_S15C_UI_PDI and Banner_S15C_UI_raw to the PLC Data Types area under your PLC.
4. Go to Devices and networks to configure the system as necessary. Below is an example of what a configuration might look like. This example shows 5 different IO-Link Masters connected to the same PLC.

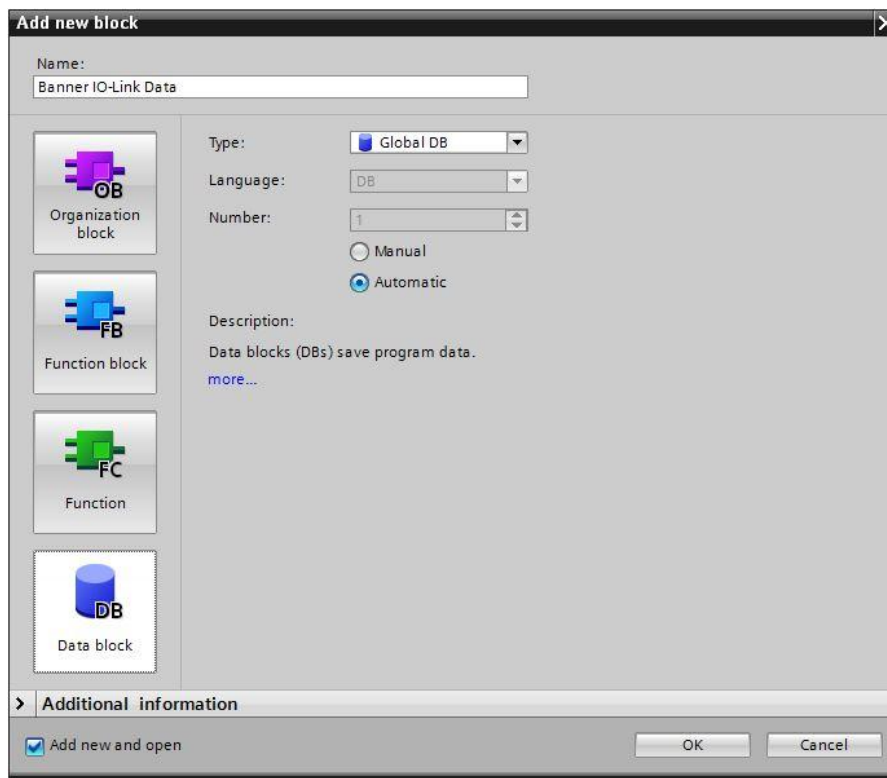


5. Click on the relevant device and configure the IO-Link Master as necessary. Refer to the documentation for the IO-Link Master. Recall that a S15C-UI requires 4 bytes of space for the Process Data In.
6. Record the "I" address where this S15C-UI Process Data In is to be stored, as the address will be required in the next step. In this example, 4 bytes of Process Data In for port 1 on the IO-Link Master will be stored in I68 through I71.

- Go to PLC Tags. Add a new tag table, if desired, then create a new tag to represent the raw Process Data from the IO-Link Master. In this example the tag “S15C UI IOLM1 01 iRaw” was created using a Data Type of “Banner_S15C_UI_raw”. This naming convention calls out the type of sensor in question as well as the specific IO-Link Master and port number where the sensor is connected. A different IO-Link Master might be named IOLM2 or IOLM3, for instance, and other specific sensors may be connected to different port numbers. The “I” address found in step 9 is tied to this new tag.

| | Name | Data type | Address |
|---|---|----------------------|---------|
| 1 |  S15C UI IOLM1 01 iRaw | "Banner_S15C_UI_raw" | %I68.0 |

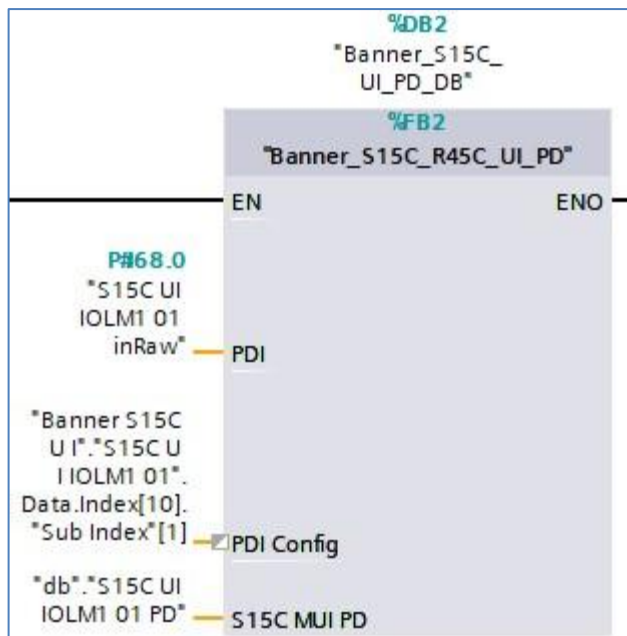
- Go to Program blocks. Add a new Data block if necessary. In this example the new data block is named “Banner IO-Link Data”.



9. In the new data block, create a new tag to represent the parsed Process Data for our S15C-UI. The tag name again calls out the type of sensor, the IO-Link Master, and the port number. Use the data type “Banner_S15C_UI_PDI” for the new tag.

| | Name | Data type |
|---|---------------------|----------------------|
| 1 | Static | |
| 2 | S15C UI IOLM1 01 PD | "Banner_S15C_UI_PDI" |

10. Add the “Banner_S15C_UI_PD” function to an OB ladder. Link the “PDI” to the raw Process Data variables from step 10. Link the “S15C MUI PD” to the parsed Process Data variable from step 12. For the “PDI Config” can either have the value manually entered or link to the PDI Config value form the Parameter Data Function Block. See Appendix A for the manual option. Linking to the Parameter Data is shown in the below image.



11. Process Data setup is complete.
12. Compile and download the configuration to the PLC, then go online. Open the “Banner IO-Link Data” data block and click Monitor all. Expand “S15C UI IOLM1 01 PD”.

| | Name | Data type |
|---|---------------------|----------------------|
| 1 | Static | |
| 2 | S15C UI IOLM1 01 PD | "Banner_S15C_UI_PDI" |
| 3 | Measurement | Real |
| 4 | Measurement Scale | UDInt |

Appendix A**S15C-UI Process Data**

The S15C-UI has 4 bytes of Process Data In, as shown below.

ProcessData id=PD_ProcessDataWithSignal (condition V_Vendor_Specific_Configuration.1 == 0)

ProcessDataIn "Process Data Input" id=PD_ProcessDataIn

bit length: 32

data type: 32-bit Record (subindex access not supported)

| subindex | bit offset | data type | allowed values | default value | acc. restr. | mod. other var. | excl. from DS | name | description |
|----------|------------|----------------|----------------|---------------|-------------|-----------------|---------------|-------------------|------------------------------|
| 1 | 0 | 32-bit Integer | | | | | | Measurement Value | The measurement device value |

| | | | | | |
|-------------|---------|---------|--------|-------|--|
| octet | 0 | 1 | 2 | 3 | |
| bit offset | 31 - 24 | 23 - 16 | 15 - 8 | 7 - 0 | |
| subindex | 1 | 1 | 1 | 1 | |
| element bit | 31 - 24 | 23 - 16 | 15 - 8 | 7 - 0 | |

ProcessData id=PD_ProcessDataWithDMS (condition V_Vendor_Specific_Configuration.1 == 1)

ProcessDataIn "Process Data Input" id=PD_ProcessDataInDMS

bit length: 32

data type: 32-bit Record (subindex access not supported)

| subindex | bit offset | data type | allowed values | default value | acc. restr. | mod. other var. | excl. from DS | name | description |
|----------|------------|----------------|----------------|---------------|-------------|-----------------|---------------|-------------------|------------------------------|
| 1 | 16 | 16-bit Integer | | | | | | Measurement Value | The measurement device value |
| 2 | 8 | 8-bit Integer | | | | | | Measurement Scale | The measurement device scale |

| | | | | | |
|-------------|---------|---------|--------|-------|--|
| octet | 0 | 1 | 2 | 3 | |
| bit offset | 31 - 24 | 23 - 16 | 15 - 8 | 7 - 0 | |
| subindex | 1 | 1 | 2 | //// | |
| element bit | 15 - 8 | 7 - 0 | 7 - 0 | | |

This function intelligently parses this Process Data into its component pieces.