

Banner IOL SD50 Parameter Data Function Block

March 17th, 2026

This document covers the installation and use of a function block for Siemen's TIA Portal software package. This function block handles acyclic IO-Link commands to and from a Banner SD50 IO-Link Device and allows the user to easily change the IO-Link Device Parameter Data.

Each Banner IO-Link Device Parameter Data function block is meant to be used alongside an IO-Link Master Control function block. This paper describes how to set up both blocks.

Components

Banner SD50 Library v16.zal16

There are two methods for parameter 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 manufacturers' IO-Link masters.

Open Global Library 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 IO-Link Device with a Banner DXMR Device

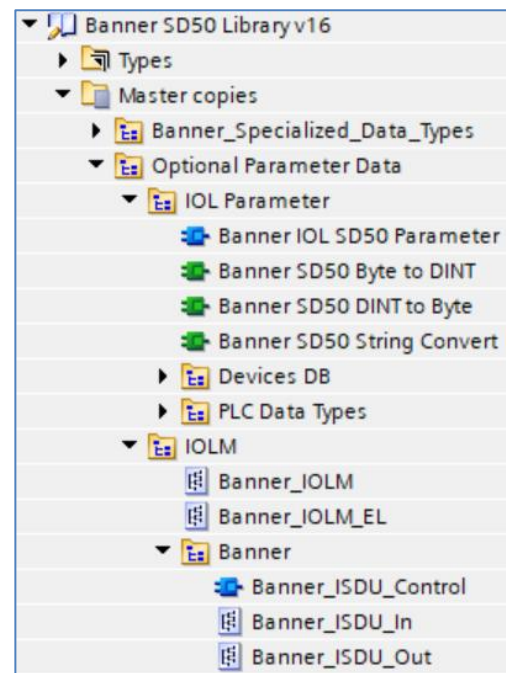
1. Go to Device and Networks to configure the DXMR. Add the DXM it has yet to be added to the system.
2. Add Banner IO-Link Master Info to Slot 1. This sets the DXMR for IO-Link mode.

Banner IO-Link Master Info_1	0	1	1...9	Banner IO-Link Master Info
------------------------------	---	---	-------	----------------------------

3. Open the IO-Link ISDU folder. Select the IO-Link ISDU 190/190 Byte option. Make note of the I address for Slot 10. The inputs data starts at I185 while the outputs data starts at Q185 for this example.

IO-Link ISDU 190/190 Byte_1	0	10	185...380	185...380	IO-Link ISDU 190/190 Byte
-----------------------------	---	----	-----------	-----------	---------------------------

4. Switch to the Libraries Tab. The Banner IOL Parameter Library should already be opened (See previous section if it is not).
5. Expand the "Master copies" folder.
6. Expand the "Optional Parameter Data" folder.
7. Expand the "IOLM" folder.
8. Drag the Banner_IOLM and Banner_IOLM_EL to the PLC Data Types area under your PLC.
9. Open the Banner folder and drag the Banner_ISDU_Control to the "Program blocks" area.
10. Also move the Banner_ISDU_In and Banner_ISDU_Out to the "PLC Data Types" area.
11. Now expand the "IOL Parameter" folder.
12. Move the "Banner IOL SD50 Parameter", "Banner SD50 Byte to DINT", "Banner SD50 DINT to Byte", and "Banner SD50 String Convert" to the Program blocks area.
13. Expand the "PLC Data Types" folder.
14. Move all the data types to the "PLC Data Types" area.
15. Finally open the "Devices DB".
16. Move the SD50 database to the "Program blocks" area.



17. The database will have two items in it. The "SD50 Rules" tells the Function Block how the data is organized. The "SD50 IOLM1 01" is the location the data is saved into by the Function Block. This tag can/should be renamed by the user. If multiple SD50 units are in the system this tag should have copies made of it. One for each SD50 in the system.

Name	Data type
▼ Static	
■ ► SD50 Rules	"Banner IOL Rules Array"
■ ► SD50 IOLM1 01	"Banner IOL SD50 Device"

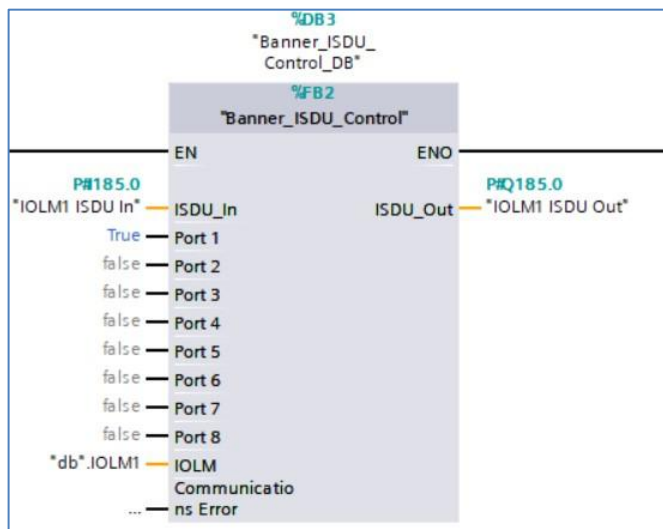
18. Go to PLC Tags. Create two tags. The first tag “IOLM1 ISDU In” and the second tag is “IOLM1 ISDU Out”. Use the %I and %Q values from step 3.

▶ IOLM1 ISDU In	*Banner_ISDU_In*	%I185.0
▶ IOLM1 ISDU Out	*Banner_ISDU_Out*	%Q185.0

19. Go to Program blocks. Add a new Data block if necessary (can use the device db that was imported). In this example the new data block is named “db”.
20. Create a tag with the type of “Banner_IOLM”. This example uses IOLM1.

db	
Name	Data type
▼ Static	
▶ IOLM1	*Banner_IOLM*

21. Next add the “Banner_ISDU_Control” function block to a ladder rung. You will be prompted to make a new data block. Accept this. You now must define the input variables for this function block: ISDU_In, ISDU_Out, and IOLM. Also set which ports the Function Block will interact with by changing the Port # to True. In this example only Port 1 will be used so that is the only one set to True. Only set a Port to True is the sensor/device is present and the parameter data Function block is configured for this device.

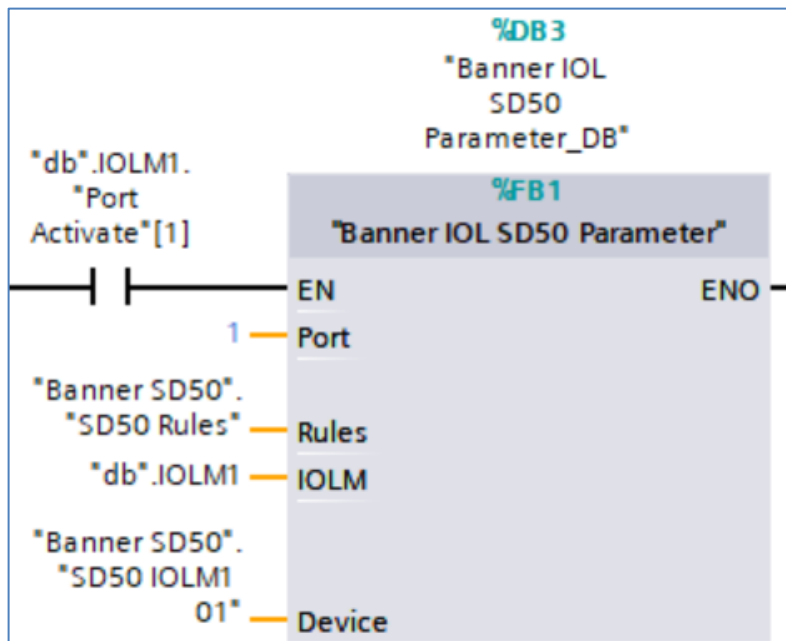


22. Link the IOLM variable to the database IOLM tag created in step 20. While ISDU_In and ISDU_Out are linked to variables created in step 18.

23. Now add the “Banner IOL SD50 Parameter” function block to a ladder rung. You will be prompted to make a new data block. Accept this. Type in the port number for the device, then link the “IOLM” variable to the IO-Link master variable created in step 20.

The Rules and Device need to be linked to the tags from the “Banner Q5X2K” database pulled in during step 17.

As a last step, the Port Activate (which is part of the IOLM tag from step 19) bit is added on the same rung as the “Banner IOL Parameter” function block to ensure orderly behavior. The IO-Link Master function block will cycle through all ports, giving each connected device function block a time to operate.



24. Setup of Parameter Data for a Banner DXMR is complete.
25. Go to Page 11 for information on how to use the Function Block.

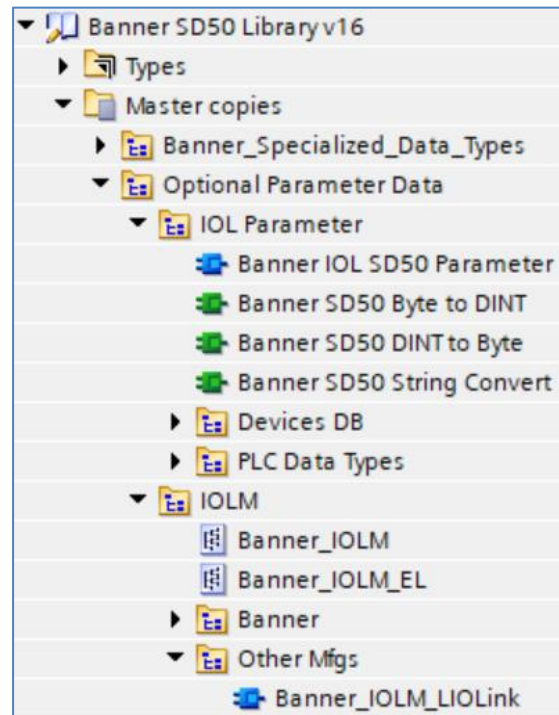
Setup of Banner IOL Parameter with other IO-Link Masters

Additional Component Needed

Siemens LIOLink V7.2 Library for TIA Portal V16+ (downloadable on Siemens website)

Installation Instructions

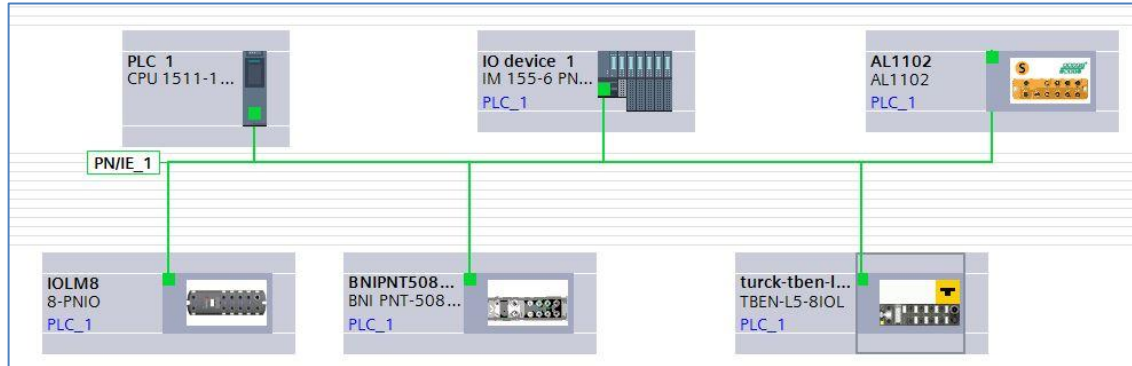
1. The Banner IOL Parameter Library will now be in the Global Library List. Expand the “Master copies” folder.
2. Expand the “Optional Parameter Data” folder.
3. Open IOLM folder.
4. Drag the Banner_IOLM and Banner_IOLM_EL to the “PLC Data types” folder.
5. Expand the “Other Mfgs” folder.
6. Move the Banner_IOLM_LIOLink to the area under “Program Blocks”.
7. Now expand the “IOL Parameter” folder.
8. Move the “Banner IOL SD50 Parameter”, “Banner SD50 Byte to DINT”, “Banner SD50 DINT to Byte”, and “Banner SD50 String Convert” to the Program blocks area.
9. Expand the “PLC Data Types” folder.
10. Move all the data types to the “PLC Data Types” area.
11. Finally open the “Devices DB”.
12. Move the SD50 database to the “Program blocks” area.
13. The database will have two items in it. The “SD50 Rules” tells the Function Block how the data is organized. The “SD50 IOLM1 01” is the location the data is saved into by the Function Block. This tag can/should be renamed by the user. If multiple SD50 units are in the system this tag should have copies made of it. One for each SD50 in the system.



Name	Data type
▼ Static	
■ ► SD50 Rules	"Banner IOL Rules Array"
■ ► SD50 IOLM1 01	"Banner IOL SD50 Device"

14. Now we must bring the Siemens-made IO_LINK_DEVICE function block or LIOLink function block specific to your PLC into our project. This example will use IO_LINK_DEVICE. This can be found in a Siemens IO-Link Library. See their website for more details. Once that library is retrieved and opened, drag IO_LINK_DEVICE to the Program Blocks area under your PLC.

15. 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.



16. Go to Program blocks. Add a new Data block if necessary. In this example the new data block is named “db”.
17. In the new data block, create a new tag to represent the IO-Link Master, using the data type “Banner_IOLM”. This example uses the tag name “IOLM1”. A different IO-Link Master might be called IOLM2 or IOLM3, for instance.

db	
Name	Data type
Static	
IOLM1	*Banner_IOLM*

18. Next add the “Banner_IOLM_Control” function block to an OB ladder. You will be prompted to make a new data block. You now must define the input variables for this function block.

Defining an input variable for the last input, Communications Error, is optional.

The Client Access Point (CAP) varies, depending on the specific IO-Link Master used.

IO-Link Master	CAP
Balluff (BNI PNT-508-105-Z015)	255
Control (IOLMPN8P)	255
ifm (AL1102)	46080
Siemens (CM 4xIO-Link)	227
Turck (TBEN-L5-8IOL)	251

The ID Control state variable should be “true” if using an IO-Link Master from ifm; otherwise, it should be set to “false”.

All Ports that will be accessed by the Function Block set them to “true”. In this example only Port 1 will be set to “true”. Only set to “true” if sensor/device is present and the parameter data function block is configured for this device.

Link the “IOLM” input variable to the tag created in step 17.

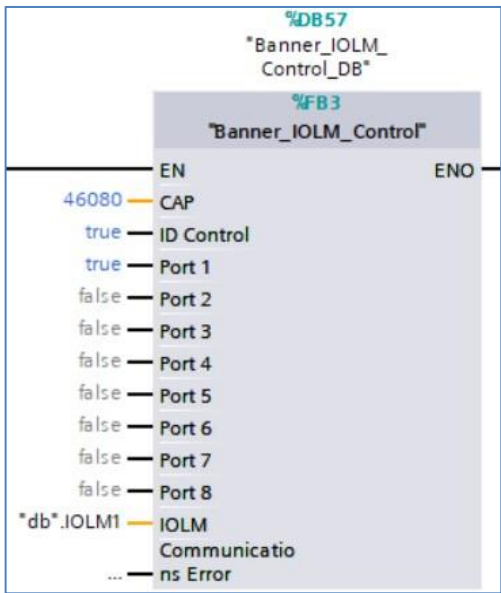


Figure 1: An example using an ifm IO-Link Master

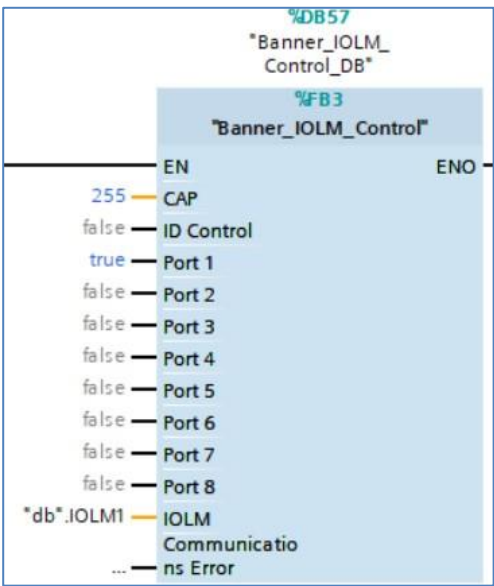


Figure 2: An example using a Balluff IO-Link Master

19. The ID Control true/false state is linked to an array called “ID_Array”, found in “Banner_IOLM_Control_DB”. This array contains the Hardware ID property of the PROFINET configuration.

See Appendix A for more information on how to find the correct value for your specific IO-Link Master.

In the case of an IO-Link Master from ifm, each port has a different Hardware ID and each number must be entered into the correct place. The example shown in Figure 3 is of an IO-Link device connected to port 6 of an ifm IO-Link Master. The ifm IO-Link Master’s port 6 Hardware ID is entered into the “ID_Array[6]” slot. This full array of different Hardware IDs, based on port used, is used when the “ID Control” variable is set to true (i.e. only when the IO-Link Master is from ifm).

IO-Link Masters from other vendors use a single Hardware ID value for all ports. In this case, the Hardware ID is entered into the “ID_Array[1]” slot of the array, regardless of the port to which the device is connected. This array is ignored (but the [1] slot is still important) when the “ID Control” variable is set to false.

ID_Array	Array[1..8] of HW_IO	
ID_Array[1]	HW_IO	0
ID_Array[2]	HW_IO	0
ID_Array[3]	HW_IO	0
ID_Array[4]	HW_IO	0
ID_Array[5]	HW_IO	0
ID_Array[6]	HW_IO	279
ID_Array[7]	HW_IO	0
ID_Array[8]	HW_IO	0

Figure 3: The ID_Array when using an ifm IO-Link Master; device attached to port 6

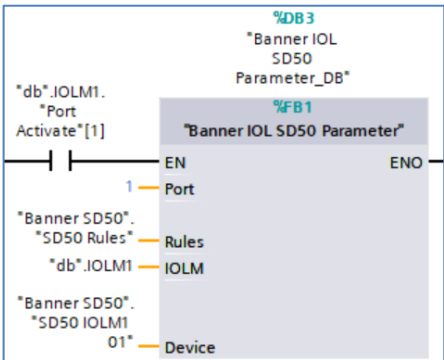
ID_Array	Array[1..8] of HW_IO	
ID_Array[1]	HW_IO	309
ID_Array[2]	HW_IO	0
ID_Array[3]	HW_IO	0
ID_Array[4]	HW_IO	0
ID_Array[5]	HW_IO	0
ID_Array[6]	HW_IO	0
ID_Array[7]	HW_IO	0
ID_Array[8]	HW_IO	0

Figure 4: The ID_Array when using a Balluff IO-Link Master; device attached to any port (only ID_Array[1] is used)

20. Now add the “Banner IOL SD50 Parameter” function block to an OB ladder. You will be prompted to make a new data block. Type in the port number for the device, then link the “IOLM” variable to the IO-Link master variable created in step 17.

The Rules and Device need to be linked to the tags from the “Banner SD50” database pulled in during step 13.

As a final step, the Port Activate bit is added on the same rung as the Banner IOL Parameter function block to ensure orderly behavior. The IO-Link Master function block will cycle through all ports, giving each connected device function block a time to operate.



21. Setup of Parameter Data for IO-Link Master is complete.
22. Go to next page for information on how to use the Function Block.

Using the Banner IOL SD50 Parameter Function Block

The Banner IOL SD50 Parameter function block will automatically gather data for all IO-Link devices in the system when first powered on. Parameter data is an acyclic process and can take some time to complete. The initial read is complete when the “Initial Global Read” tag is set to “true”. This flag can be set to false to request another full global read of all parameter data for an IO-Link device. The flag is found in the tag that is part of the pulled in database from the library. The tag is in the “Data type” “Banner IOL SD50 Device”. There should be one tag of “Banner IOL SD50 Device” for each device in the system. Rename them as needed for the system.

Name	Data type	Monitor value
▼ Static		
■ ▶ SD50 Rules	"Banner IOL Rules Array"	
■ ▼ SD50 IOLM1 01	"Banner IOL SD50 Device"	
■ Initial Global Read	Bool	TRUE
■ Command	USInt	0

The Data section in “Banner IOL SD50 Device” should be expanded. Now the Index should also be expanded. Now the Index array is fully visible. Start at index[2] each line is labeled. These labels represent the IO-Link Indices in the device. Read only indices will have a “ro” in the comment. Write only has a “wo”. Everything with neither a “ro” or a “wo” is Read Write capable. See the image below for an example of this.

▼ SD50 IOLM1 01	"Banner IOL SD50 Device"		
■ Initial Global Read	Bool	TRUE	
■ Command	USInt	0	
■ ▼ Data	"Banner IOL Device Index"		
■ ▼ Index	Array[0..34] of "Banner IOL De..."		
■ ▶ Index[0]	"Banner IOL Device SubIndex"		
■ ▶ Index[1]	"Banner IOL Device SubIndex"		
■ ▶ Index[2]	"Banner IOL Device SubIndex"		Direct Parameters (ro)
■ ▶ Index[3]	"Banner IOL Device SubIndex"		Standard Command (wo)
■ ▶ Index[4]	"Banner IOL Device SubIndex"		Device Access Locks
■ ▶ Index[5]	"Banner IOL Device SubIndex"		Serial Number (ro)
■ ▶ Index[6]	"Banner IOL Device SubIndex"		Operating Mode
■ ▶ Index[7]	"Banner IOL Device SubIndex"		Additional Settings
■ ▶ Index[8]	"Banner IOL Device SubIndex"		Display Settings
■ ▶ Index[9]	"Banner IOL Device SubIndex"		Timer Settings
■ ▶ Index[10]	"Banner IOL Device SubIndex"		Measure General Configuration
■ ▶ Index[11]	"Banner IOL Device SubIndex"		Measure Base Configuration
■ ▶ Index[12]	"Banner IOL Device SubIndex"		Measure Threshold 1 Configuration

A Global Read can be started by either entering a 1 into the Command or setting the “Initial Global Read” to false. A singular Index read is started by entering the index number into Command. As an example, if “Additional Settings” should be read then entering a seven into the Command does the Read operation for that Index. The Data in the index is now updated. Expand the index to see the data.

The Write operation requires a few steps to complete. Start by expanding the Index that will be updated. For this example, “Additional Settings” will be used.

▼ Index[7]	*Banner IOL Device SubIndex*		Additional Settings
▼ Sub Index	Array[0..34] of DInt		
■ Sub Index[0]	DInt	0	
■ Sub Index[1]	DInt	100	Custom Intensity: 0 to 100%
■ Sub Index[2]	DInt	15	Custom Flash Rate: 5 to 200 (0.5 to 20)
■ Sub Index[3]	DInt	100	Custom Display Intensity: 0 to 100%
■ Sub Index[4]	DInt	15	Custom Display Scroll Speed: 0 to 255

Change all the “Sub Index” values that need to be updated. This example changes Custom Intensity from 100 to 90, and Custom Flash from 15 to 20.

▼ Index[7]	*Banner IOL Device SubIndex*		Additional Settings
▼ Sub Index	Array[0..34] of DInt		
■ Sub Index[0]	DInt	0	
■ Sub Index[1]	DInt	90	Custom Intensity: 0 to 100%
■ Sub Index[2]	DInt	20	Custom Flash Rate: 5 to 200 (0.5 to 20)
■ Sub Index[3]	DInt	100	Custom Display Intensity: 0 to 100%
■ Sub Index[4]	DInt	15	Custom Display Scroll Speed: 0 to 255

The Command needs to be set for the Write command to be updated. Take the index number and add 40 to it. This is the value that needs to be entered into the Command value. Here the value 47 (40 + 7) is entered and the device is updated.

▼ SD50 IOLM1 01	*Banner IOL SD50 Device*	
■ Initial Global Read	Bool	TRUE
■ Command	USInt	47

The Command value will be set back to 0 after the operation is completed. When the Command is set back to 0 look at the “Communications Error” tag from the “Banner_IOLM_Control” or “Banner_ISDU_Control” database. This tells the user if the write operation was successful or not.

▼ SD50 IOLM1 01	*Banner IOL SD50 Device*	
■ Initial Global Read	Bool	TRUE
■ Command	USInt	0

Communications Error	Bool	false	TRUE
----------------------	------	-------	------

The SD50 uses String in many of the Parameter Data Indexes. These Indexes unfortunately only support DINT data types. To allow for transfer of String data into these DINTs the String element is used. This String data is located directly under the Data tag.

▼ SD50 IOLM1 01	"Banner IOL SD50 Device"	
■ Initial Global Read	Bool	TRUE
■ Command	USInt	0
■ ▶ Data	"Banner IOL Device Index"	
■ ▶ String	Array[1..20] of String	

Expand the String tag and you gain access to all String parameter data elements. The data is shown as an ascii string instead of a DINT number.

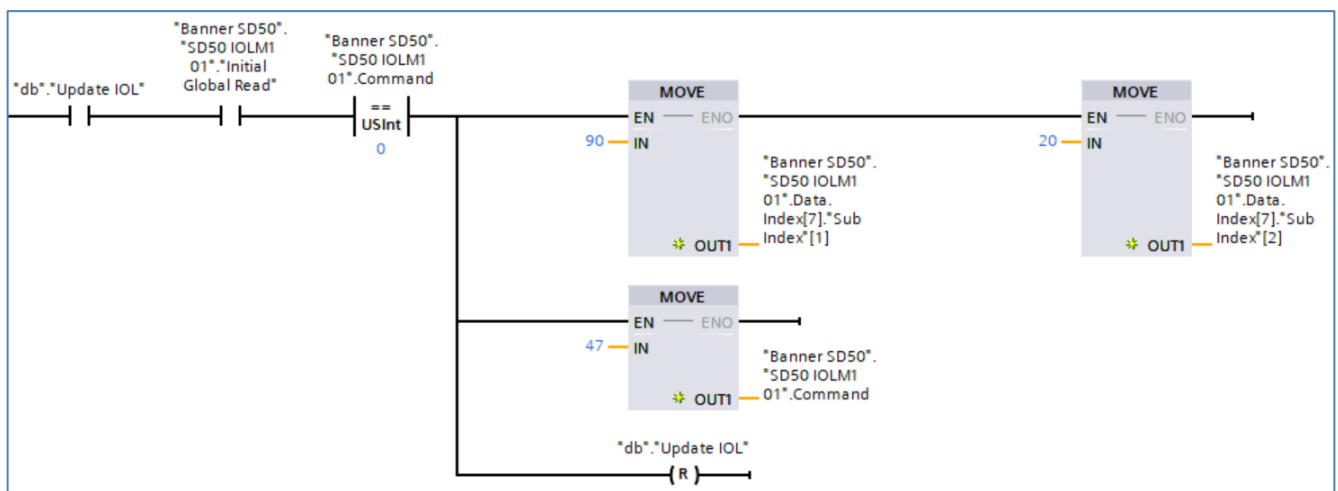
▼ String	Array[1..20] of String		
■ String[1]	String	'Time='	Index 10 - MTC Mode Data Label
■ String[2]	String	's'	Index 10 - MTC Mode Value Label
■ String[3]	String	'Base'	Index 11 - Override String
■ String[4]	String	'Thresh 1'	Index 12 - Override String
■ String[5]	String	'Thresh 2'	Index 13 - Override String
■ String[6]	String	'Thresh 3'	Index 14 - Override String
■ String[7]	String	'Thresh 4'	Index 15 - Override String
■ String[8]	String	'Reset'	Index 16 - Message 1
■ String[9]	String	'Fault'	Index 17 - Message 2

When one of the Strings needs to be adjusted, change the string here (String Data element). Next modify any additional items in the Index (under Data) and then set the command to write the updates. As an example if the MTC Mode Data Label needs to be changed to 'Time2=', start by changing String[1] to 'Time2='. If no additional items in Index 10 need to be changed then set Command to 50 to process the update. If other items need to be changed set those parameters in Data.Index[10] before setting Command to 50.

The previous section went through the steps of how to manually read and write the data. This section will show an example of how the update could happen when programmed. The user will need to come up with some logic that determines when the update should occur. This logic should not be able to be continuously activated.

***NOTE:** Parameter Data is meant to be updated infrequently. The data is stored in EEPROM that has a limited number of writings available. Exceeding this limit can cause the IO-Link device to error out. Each Index data element has a separate counter for this. If the application requires very quick Index writes, contact Banner Engineering to discuss. Never need to worry about Reads, however.

Example Logic for IO-Link Parameter Data Update



In this example the “db”.Update IOL” represents the logic that triggers the parameter data update. Here it is just a simple Boolean value; however, it will likely be more complex in the actual system. Next is a normally open contact that is checking that the “Initial Global Read” has been set to true. This ensures that the system has read all the parameter data correctly. If the parameter data has yet to be read then the data cannot be update yet. Next is an Equal comparison check. This looks at the Command variable for the function block. If Command is already processing a command/operation wait until that is completed before trying to do another command/operation. Finally, all the data for one Parameter Data Index can be adjusted as needed. In this example the Sub-Index[1] (Custom Intensity) and Sub-Index[2] (Custom Flash Rate) are updated to the necessary values for the system. Command has a 47 sent to it which represent Index 7 (remember writes are activated by adding 40 to the Index number). The Boolean used to activate the routine it turned off. It may be necessary to handle this in another way depending on the logic used to activate this process. The Banner IOL SD50 Parameter Function Block will update the IO-Link device to the new parameters.

Appendix A IO-Link Master Hardware ID Numbers

The Hardware ID number used in “ID_Array” in the “Banner_IOLM_Control_DB” function block is not trivial to find. Each manufacturer uses the Hardware Identifier of a slightly different subcomponent as the value required for our purposes. Furthermore, the particular Hardware Identifier numbers will change based on the number of devices in your configuration. These pictures show which subcomponent’s Hardware ID is relevant to the function block.

In each case, click on the hardware device from the “Devices & Networks” view. Click on “Properties”, then click on “System Constants” to see the screen shots below.

Balluff

Use the Hardware Identifier from the “BNI_PNT-508-105-Z015_1” Hw_SubModule. Type this number into the [1] slot of the ID_Array found in the “Banner_IOLM_Control_DB” data block.

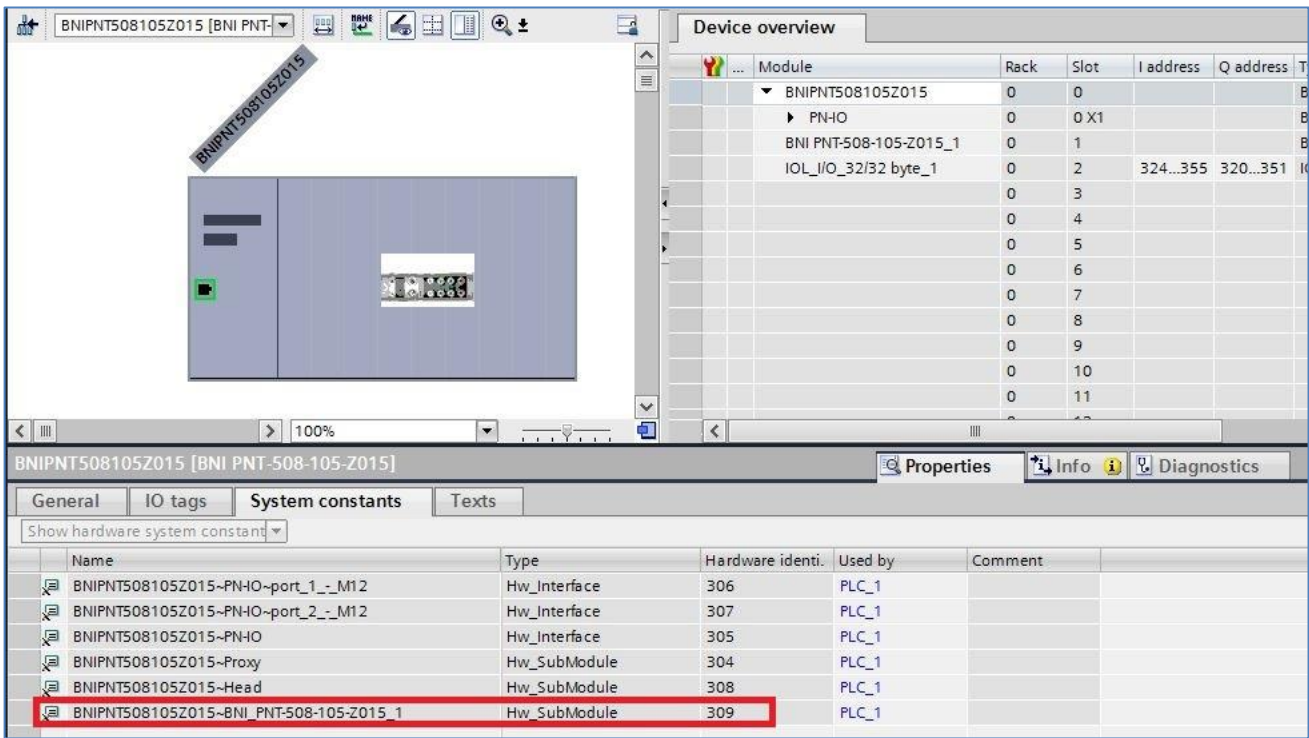


Figure 2: Balluff BNI005H. Type this value into the “ID_Array[1]” location.

Control

Use the Hardware Identifier from the “Head” Hw_SubModule. Type this number into the [1] slot of the ID_Array found in the “Banner_IOLM_Control_DB” data block.

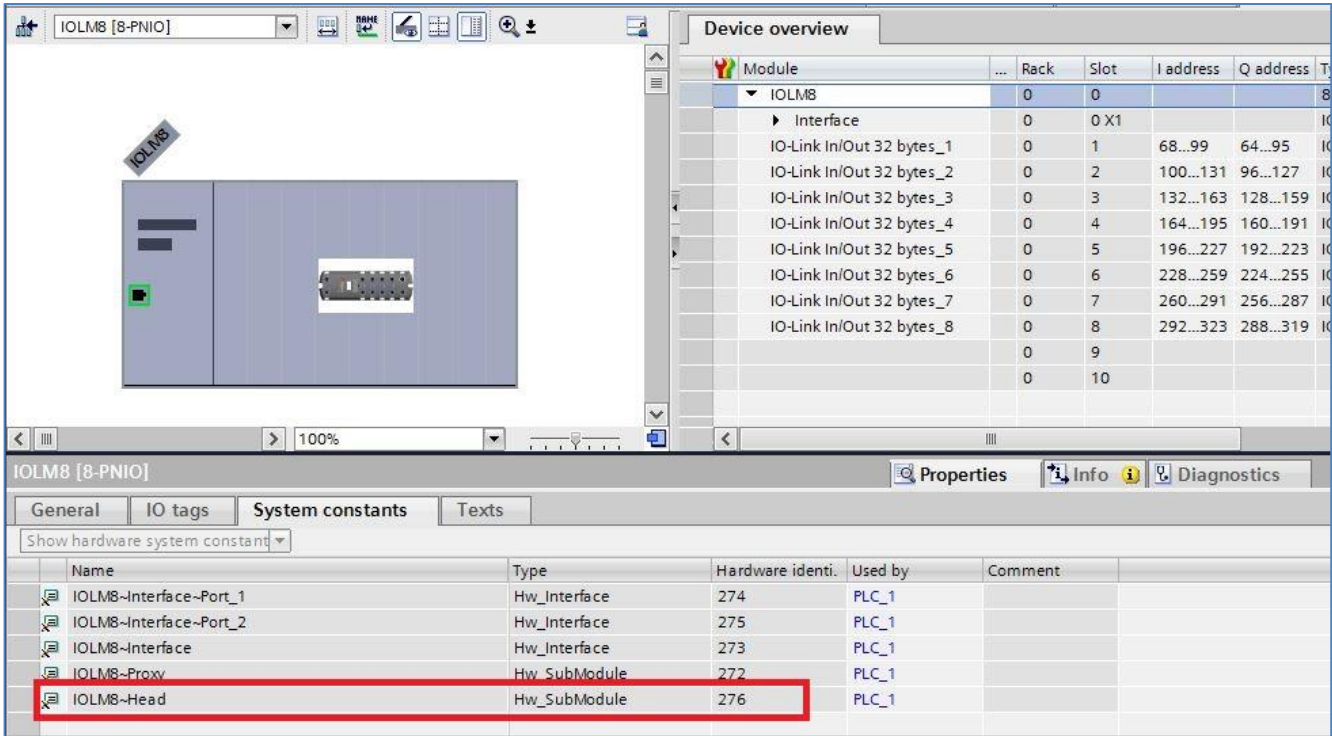


Figure 3: Control IOLM8 PNIO. Type this value into the “ID_Array[1]” location.

Turck

Use the Hardware Identifier from the “Basic_1” Hw_SubModule. Type this number into the [1] slot of the ID_Array found in the “Banner_IOLM_Control_DB” data block.

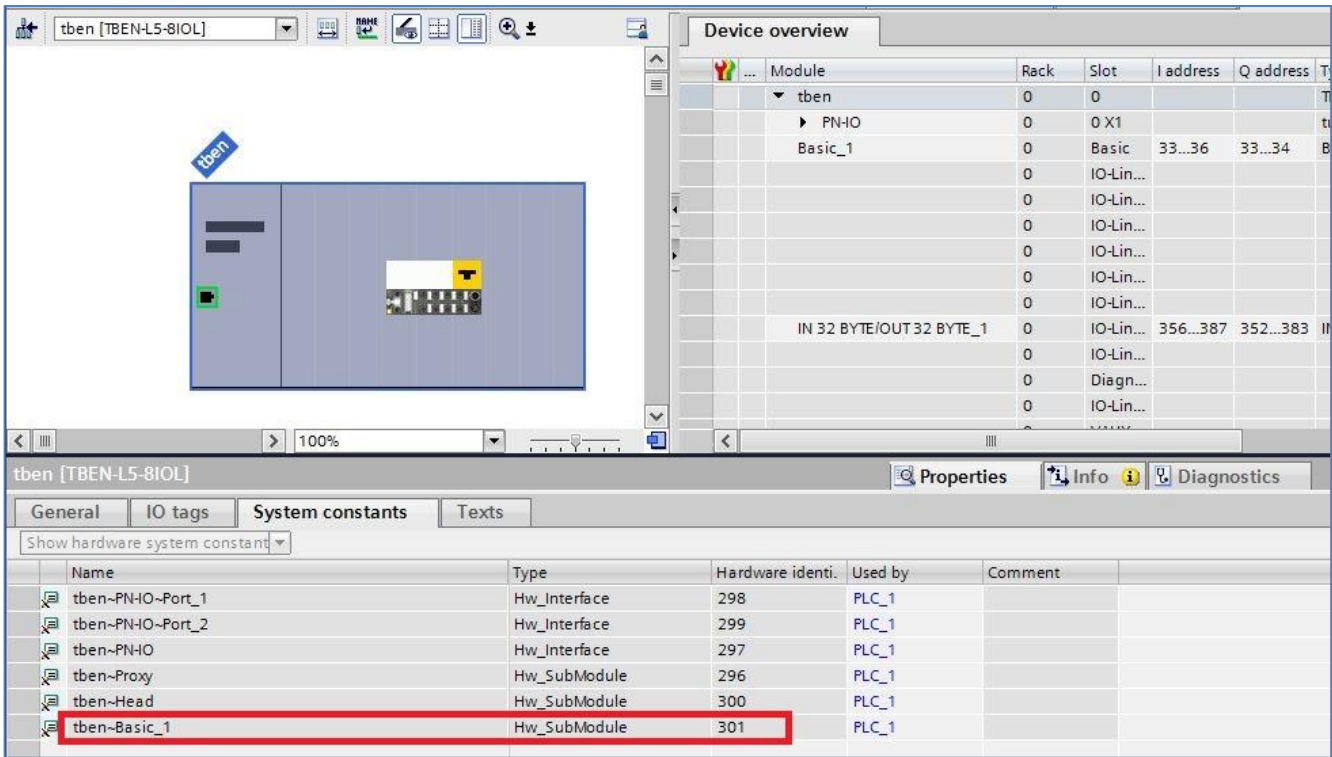


Figure 3: Turck TBEN-L5-8IOL. Type this value into the “ID_Array[1]” location.

Siemens

Use the Hardware Identifier from the “CM_4xIO-Link_1” Hw_SubModule. Type this number into the [1] slot of the ID_Array ID_Array found in the “Banner_IOLM_Control_DB” data block.

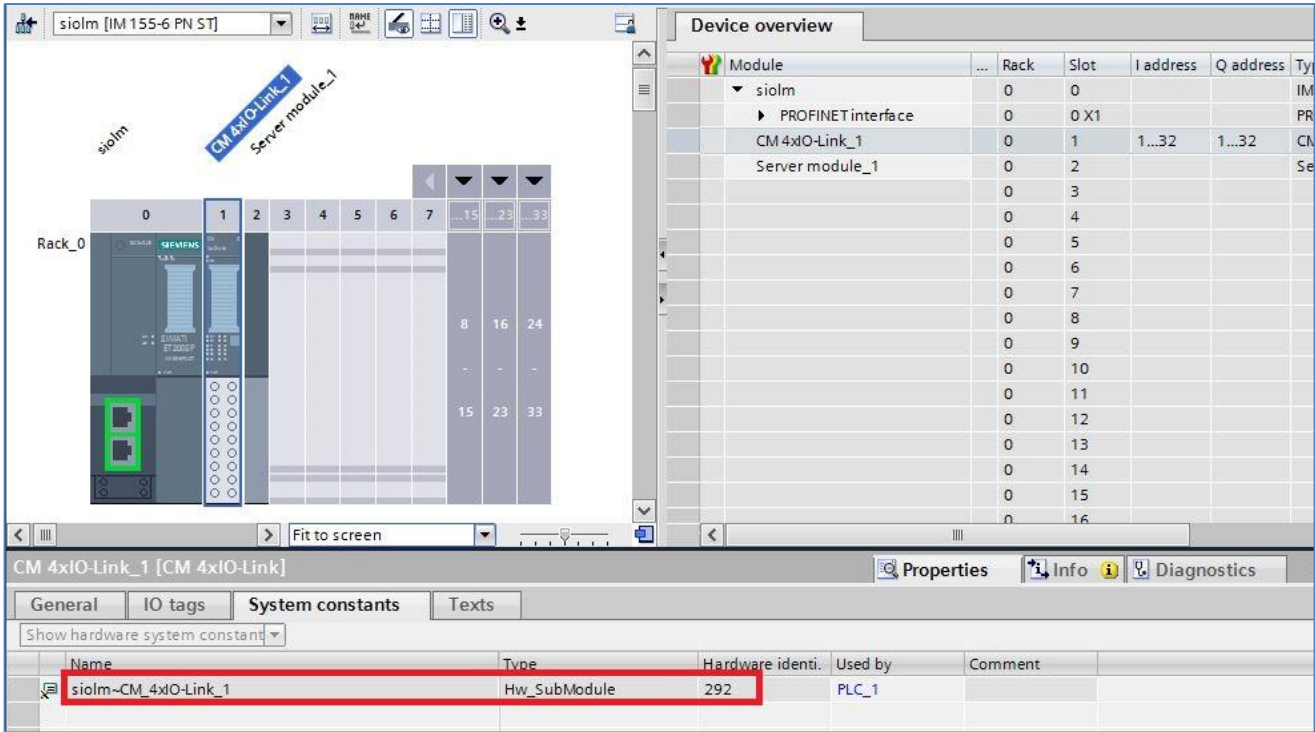
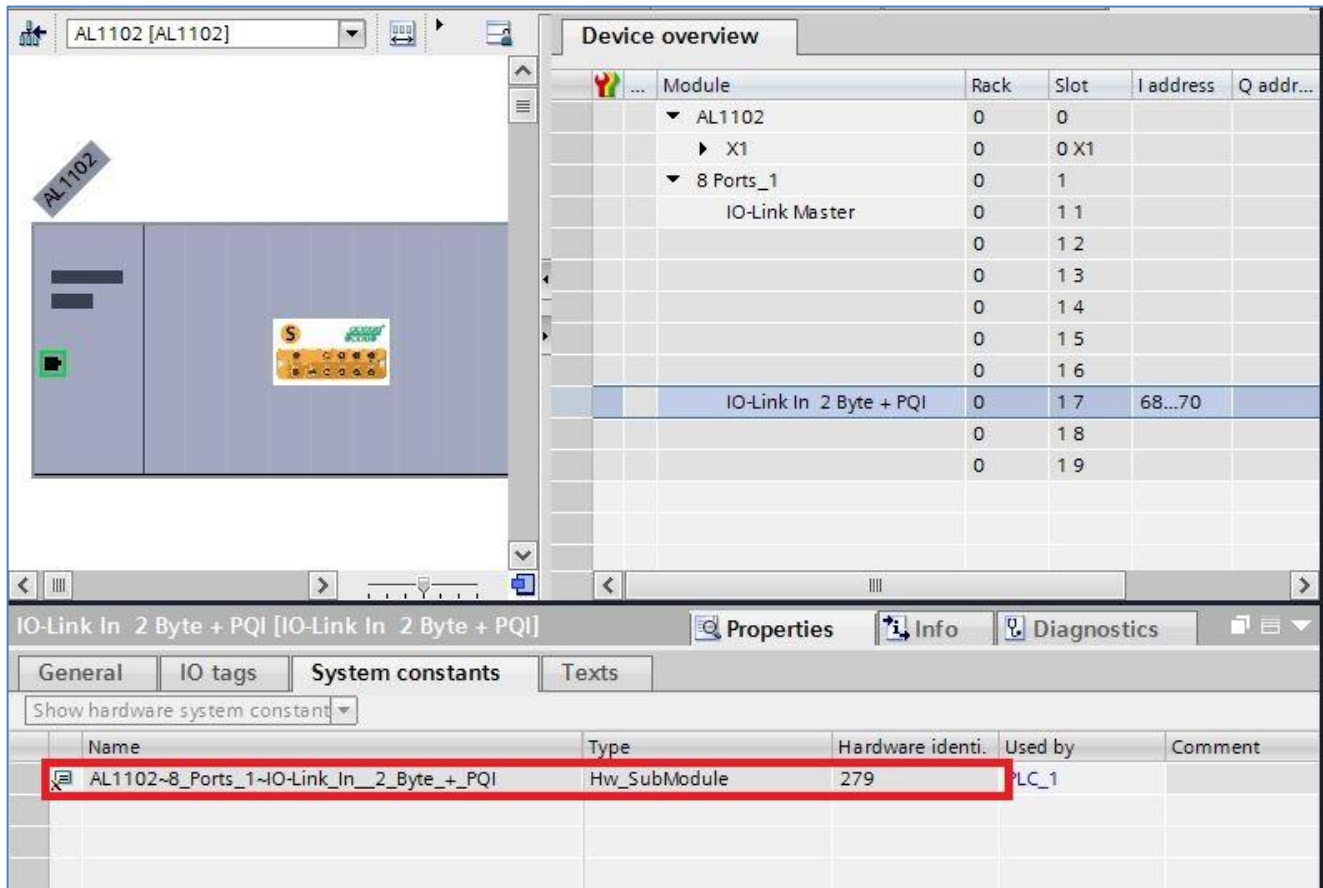


Figure 4: Siemens CM 4xIO-Link Master on ET-200SP. Type this value into the “ID_Array[1]” location.

ifm

Use the Hardware Identifier from the port to which the IO-Link Device you wish to control is connected Hw_SubModule. Each port is a different Hardware identifier. You will need to populate the ID_Array, found in the “Banner_IOLM_Control_DB” data block, with the correct values. In the example below, port 6 on the master has a Hardware ID of “279”. Thus, the [6] entry in the ID_Array variable should be set to “279”.



The screenshot displays the 'Device overview' window for the AL1102 module. The table below shows the hardware identifiers for each port. The 'IO-Link In 2 Byte + PQI' module is highlighted, showing a hardware identifier of 279.

Module	Rack	Slot	I address	Q addr...
AL1102	0	0		
X1	0	0 X1		
8 Ports_1	0	1		
IO-Link Master	0	1 1		
	0	1 2		
	0	1 3		
	0	1 4		
	0	1 5		
	0	1 6		
IO-Link In 2 Byte + PQI	0	1 7	68...70	
	0	1 8		
	0	1 9		

The 'Properties' window shows the 'System constants' tab. The table below shows the hardware identifiers for each port.

Name	Type	Hardware identi.	Used by	Comment
AL1102~8_Ports_1~IO-Link_In__2_Byte_+_PQI	Hw_SubModule	279	LC_1	

Figure 5: ifm AL1102: each port on the ifm IO-Link Master has its own Hardware ID. Type these values into the correct “ID_Array[x]” location, where ‘x’ is the port number in question ([6] here, as the ports are labeled 2 through 9).