

K50 Pro Parameter Data Function Block

11/17/2022

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 K50 Pro Touch and allows the user to easily change K50 Pro Touch Parameter Data.

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

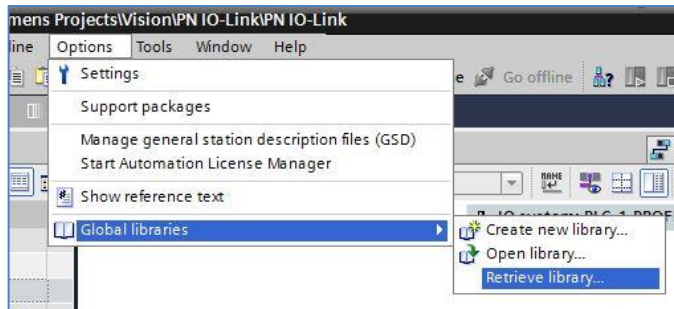
Components

Banner K50 Pro Touch.zal14

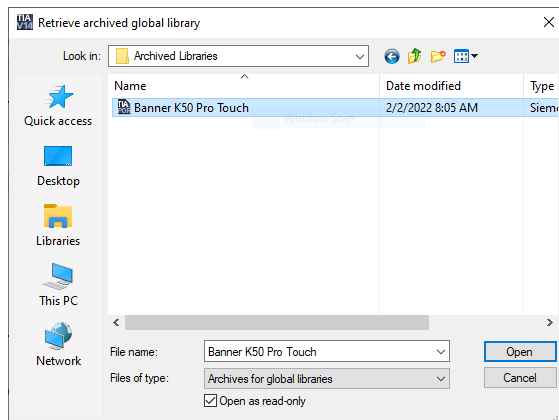
There are two methods for the parameter data. The first is used when creating a connection to Banner's IO-Link masters. The second set of instructions are for systems using other manufacturer's IO-Link masters.

Installation Instructions

1. Open a project.
2. Go to Options > Global Libraries > Retrieve Library.



3. Select the Banner K50 Pro Touch. Click Open.



4. The library is now accessible in the Libraries tab.
5. Go to page 3 for Banner IO-Link Masters, page 8 for K50 Pro for all other IO-Link Master, and to page 15 for K50 Audible for all other IO-Link Masters.

Setup of K50 Pro Touch with a Banner DXMR90-4K

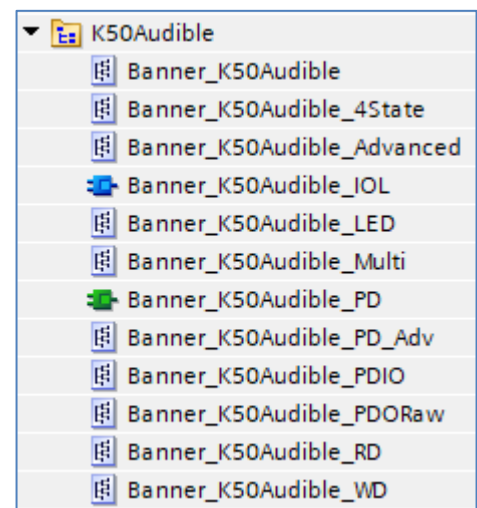
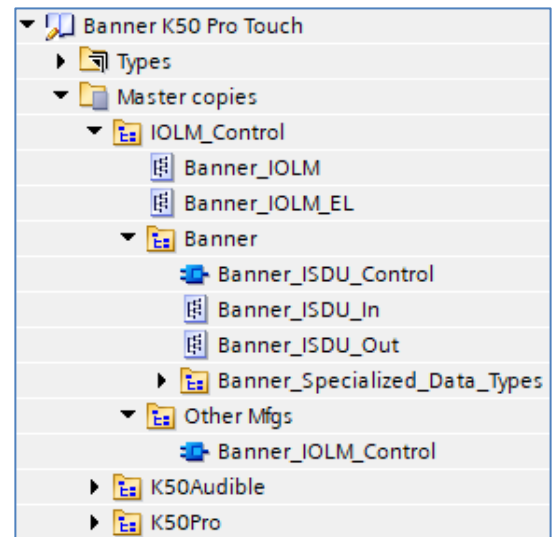
1. Go to Device and Networks to configure the DXMR90-4K. Add the DXMR90-4K if it has yet to be added to the system.
2. Add Banner IO-Link Master Info to Slot 1. This sets the DXMR90-4K for IO-Link mode.

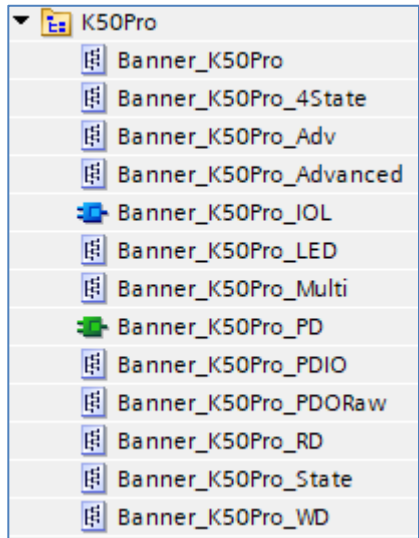
Banner IO-Link Master Info_1	0	1	1...9	Banner IO-Link Master Info
------------------------------	---	---	-------	----------------------------

3. Open the IO-Link Generic Devices and select the proper module. The IO-Link ISDU 190/190 Byte_1 is required for this Function Block. Make note of the I address for the 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. Drag the Banner_IOLM and Banner_IOLM_EL to the PLC Data Types area under your PLC. These are found in the IOLM_Control.
5. Open the Banner folder and drag the Banner_ISDU_Control to the Program blocks area.
6. Now open either the K50 Audible (step 6) or K50 Pro (step 8) folder depending on which unit it being configured.
7. Move the Banner_K50Audible_IOL to the Program blocks area.
8. Finally move the Banner_K50Audible, Banner_K50Audible_Advanced, Banner_K50Audible_RD, and Banner_K50Audible_WD to the PLC Data Types.
9. Move the Banner_K50Pro_IOL to the Program blocks area.
10. Finally move the Banner_K50Pro, Banner_K50Pro_Advanced, Banner_K50Pro_State, Banner_K50Pro_RD, and Banner_K50Pro_WD to the PLC Data Types.





****K50 Pro will be used for the rest of the document. Use K50 Audible in place of K50 Pro if that model is required.****

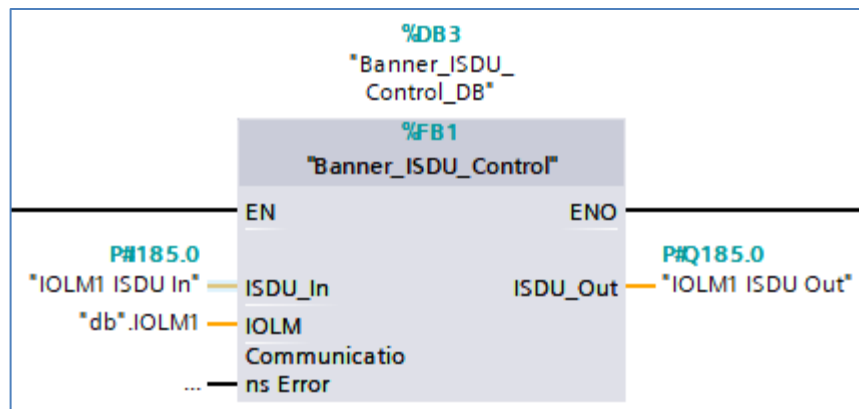
11. Go to PLC Tags. Create two tags. The first tag “IOLM1 ISDU In” and the second tag is “IOLM1 ISDU Out”. Use the %I and %Q values from step 2.

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

12. Go to Program blocks. Add a new Data block if necessary. In this example the new data block is named “db”.
13. Create a tag with the type of “Banner_IOLM”. This example uses IOLM1.
14. Expand the IOLM1 tag, then expand the Port Controlled section. The Port Controlled tag array determines which of the ports has a function block-controlled Banner IO-Link device plugged into it. Each Port Controlled array tag with **true** as the start value is considered to have such a device connected. Correctly setting this array allows the Device and IO-Link Master function blocks to control the device on that port. Errors will occur if a port without an IO-Link device is set to true.

db			
	Name	Data type	Start value
[-DI]	[-] IOLM1	"Banner_IOLM"	
[-DI]	[-] Port Controlled	Array[1..8] of Bool	
[-DI]	[-] Port Controlled[1]	Bool	true
[-DI]	[-] Port Controlled[2]	Bool	true
[-DI]	[-] Port Controlled[3]	Bool	true
[-DI]	[-] Port Controlled[4]	Bool	false
[-DI]	[-] Port Controlled[5]	Bool	false
[-DI]	[-] Port Controlled[6]	Bool	false
[-DI]	[-] Port Controlled[7]	Bool	false
[-DI]	[-] Port Controlled[8]	Bool	false
[-DI]	[-] Port Activate	Array[1..8] of Bool	
[-DI]	[-] Port Read Request	Array[1..8] of Bool	
[-DI]	[-] Port Write Request	Array[1..8] of Bool	
[-DI]	[-] Port RW Complete	Array[1..8] of Bool	
[-DI]	[-] Port Device Read	Array[1..8] of Bool	
[-DI]	[-] Transfer Data	Array[0..231] of B...	
[-DI]	[-] Wr_Length	UInt	0
[-DI]	[-] Rd_Length	UInt	0
[-DI]	[-] IO-Index	Int	0
[-DI]	[-] Reset	Bool	false

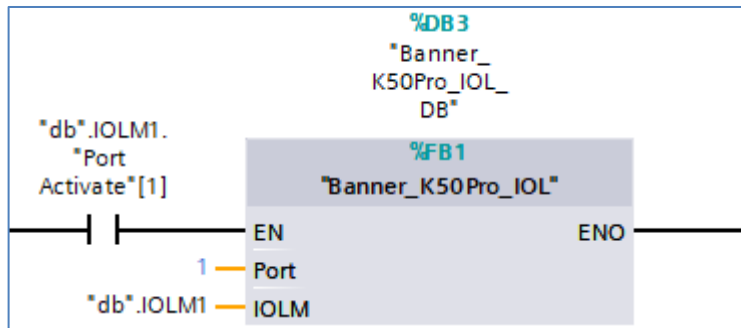
11. Next add the "Banner_ISDU_Control" function block to an OB ladder. You will be prompted to make a new data block. You now must define three input variables for this function block: ISDU_In, ISDU_Out, and IOLM.











































12. Link the IOLM variable to the database IOLM tag created in step 10. While ISDU_In are linked to variables created in step 7.
13. Now add the "Banner_K50Pro_IOL" 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 10.

As a final step, the Port Activate bit is added on the same rung as the K50 Pro Touch 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 shine.



14. The K50 Pro Touch Parameter Data function block is now set up. Compile the project and download it to the PLC. Go online, then open the K50 Pro Touch data block. When the function block starts out, it does an initial global read of all K50 Pro Touch information. The Read Data section of the data block shows this information.

Banner_K50Pro_IOL_DB				
	Name	Data type	Start value	Monitor value
7	  K50Pro Data	*Banner_K50Pro*		
8	  Initial Global Read	Bool	false	TRUE
9	  Command	Int	0	0
10	  Read Data	*Banner_K50Pro_RD*		
11	  Master Cycle Time	USInt	0	56
12	  Min Cycle Time	USInt	0	50
13	  M-Sequence Capability	USInt	0	9
14	  IO-Link Version ID	USInt	0	17
15	  Process Data Input Length	USInt	0	16
16	  Process Data Output Length	USInt	0	137
17	  Vendor ID Combined	UDInt	0	451
18	  Vendor ID 1	USInt	0	1
19	  Vendor ID 2	USInt	0	195
20	  Device ID Combined	UDInt	0	393220
21	  Device ID 1	USInt	0	6
22	  Device ID 2	USInt	0	0
23	  Device ID 3	USInt	0	4
24	  Serial Number	String	"	'18-10-1521:09:34'
25	  Write Data	*Banner_K50Pro_W...		
26	  Reset	Bool	false	FALSE

The Write Data section of the data block shows all the writeable parameter data for the K50 Pro Touch. Simply change a value here and it will be automatically written to the sensor via a one-time acyclic write through the IO-Link Master.

It is almost like the K50 Pro Touch is a PROFINET-speaking device!

Banner_K50Pro_IOL_DB				
	Name	Data type	Start value	Monitor value
7	▼ K50Pro Data	"Banner_K50Pro"		
8	Initial Global Read	Bool	false	TRUE
9	Command	Int	0	0
10	► Read Data	"Banner_K50Pro_RD"		
11	▼ Write Data	"Banner_K50Pro_W..."		
12	System Command	USInt	0	0
13	Parameter Access Lock	Bool	false	FALSE
14	Data Storage Lock	Bool	false	FALSE
15	Local Parameterization Lock	Bool	false	FALSE
16	Local User Interface Lock	Bool	false	FALSE
17	Operating Mode	USInt	0	1
18	Custom Intensity	USInt	0	100
19	Custom Flash Rate	USInt	0	15
20	Restrict to Gamut	USInt	0	0
21	Touch Sensitivity	USInt	0	1
22	Function	Bool	false	FALSE
23	Mute Enable	Bool	false	FALSE
24	OnDelay	UInt	0	0
25	Output State	Bool	false	TRUE
26	Off Delay Type	Bool	false	FALSE
27	Off Delay	UInt	0	0
28	► State Parameters	Array[1..4] of "Ban..."		
29	Custom 1 - Red	USInt	0	255

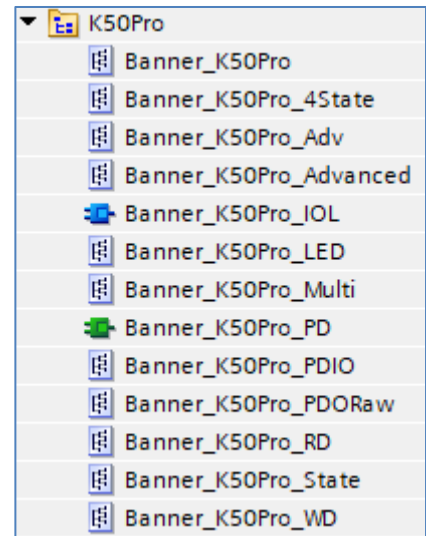
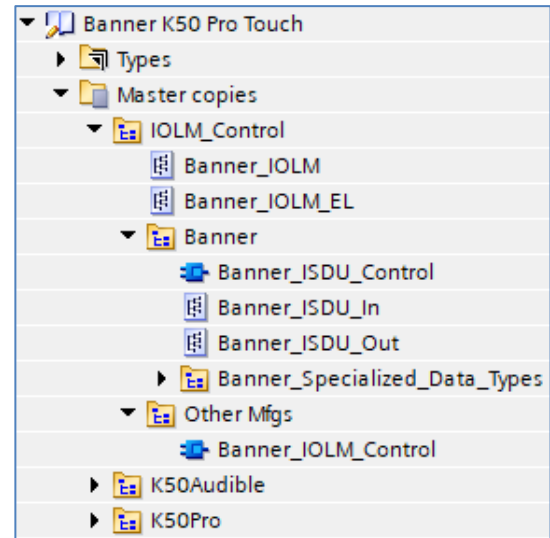
Setup of K50 Pro Touch with other IO-Link Masters

Additional Component Needed

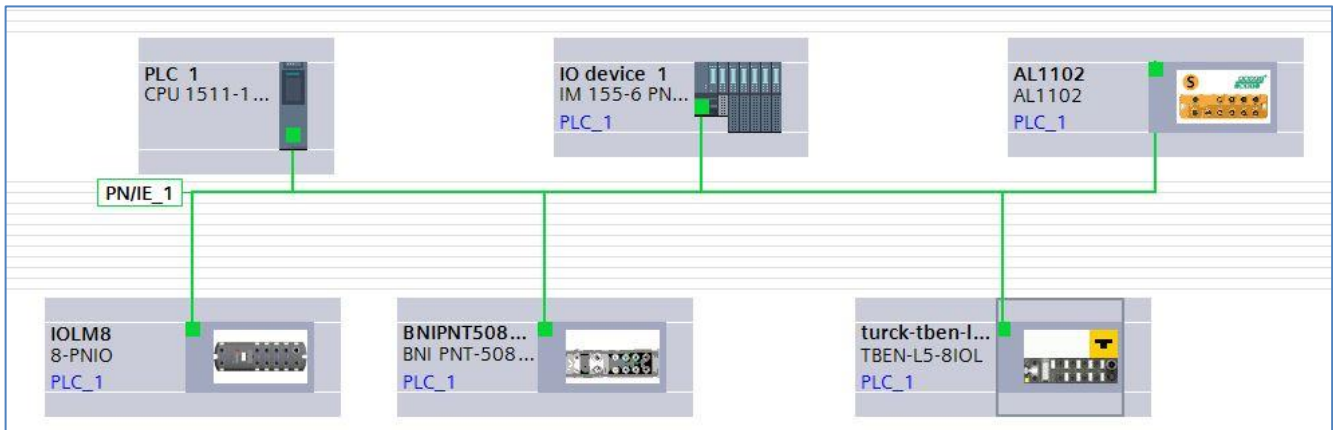
Siemens IO_LINK_DEVICE function block

Installation Instructions

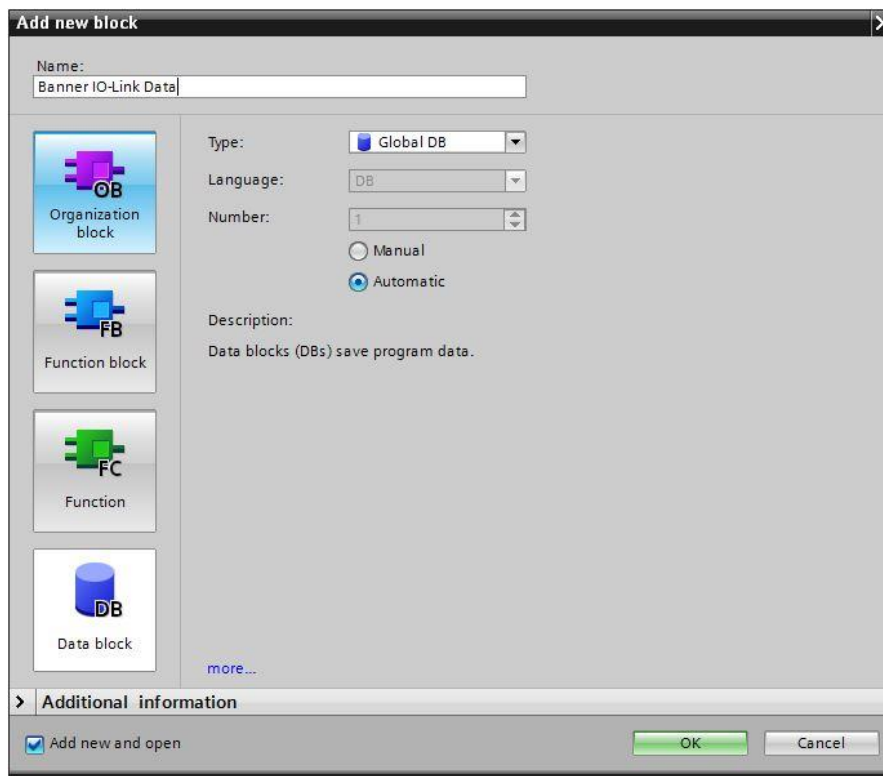
1. The Banner K50 Pro Touch library will now be in the Global Library List. Expand the Master copies section. The K50 Pro Touch folder contains elements for both Process Data and Parameter Data connections to a K50 Pro Touch device. As Parameter Data is the focus of this paper, we will concern ourselves with these six items: Banner_K50Pro, Banner_K50Pro_Advanced, Banner_K50Pro_IOL, Banner_K50Pro_RD, Banner_K50Pro_State, and Banner_K50Pro_WD.
2. Drag Banner_K50Pro_IOL to the Program Blocks area under your PLC.
3. Drag the Banner_K50Pro, Banner_K50Pro_Advanced, Banner_K50Pro_RD, Banner_K50Pro_State, and Banner_K50Pro_WD to the PLC Data Types area under your PLC.
4. We also must prepare for setting up the IO-Link Master. Go to the IOLM_Control section of the Banner IO-Link Library List.
5. Drag Banner_IOLM and Banner_IOLM_EL to the PLC Data Types area under your PLC.
6. Drag the Banner_IOLM_Control to the Program Blocks area under your PLC.
7. Finally, we must bring the Siemens-made IO_LINK_DEVICE function block specific to your PLC into our project. 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.





8. 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.























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



10. 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 "IOLM5". A different IO-Link Master might be called IOLM1 or IOLM2, for instance.

Banner IO-Link Data				
		Name	Data type	Start value
1		▼ Static		
2		■ ► IOLM5	"Banner_IOLM"	

11. Expand the IOLM5 tag, then expand the Port Controlled section. The Port Controlled tag array determines which of the ports has a function block-controlled Banner IO-Link device plugged into it. Each Port Controlled array tag with **true** as the start value is considered to have such a device connected. Correctly setting this array allows the Device and IO-Link Master function blocks to control the device on that port. Errors will occur if a port without an IO-Link device is set to true.

Banner IO-Link Data					
		Name	Data type	Start value	Monitor value
2		■ ▼ IOLM5	"Banner_IOLM"		
3		■ ▼ Port Controlled	Array[1..8] of Bool		
4		■ Port Controlled[1]	Bool	false	
5		■ Port Controlled[2]	Bool	true	
6		■ Port Controlled[3]	Bool	false	
7		■ Port Controlled[4]	Bool	false	
8		■ Port Controlled[5]	Bool	false	
9		■ Port Controlled[6]	Bool	false	
10		■ Port Controlled[7]	Bool	false	
11		■ Port Controlled[8]	Bool	false	
12		■ ► Port Activate	Array[1..8] of Bool		
13		■ ► Port Read Request	Array[1..8] of Bool		
14		■ ► Port Write Request	Array[1..8] of Bool		
15		■ ► Port RW Complete	Array[1..8] of Bool		
16		■ ► Port Device Read	Array[1..8] of Bool		
17		■ ► Transfer Data	Array[0..231] of Byte		
18		■ Wr_Length	Int	0	
19		■ Rd_Length	Int	0	
20		■ IO-Index	Int	0	
21		■ Reset	Bool	false	

12. 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 three input variables for this function block: CAP, ID Control state, and IOLM.

Defining an input variable for the fourth 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”.

Link the “IOLM” input variable to the tag created in step 13.

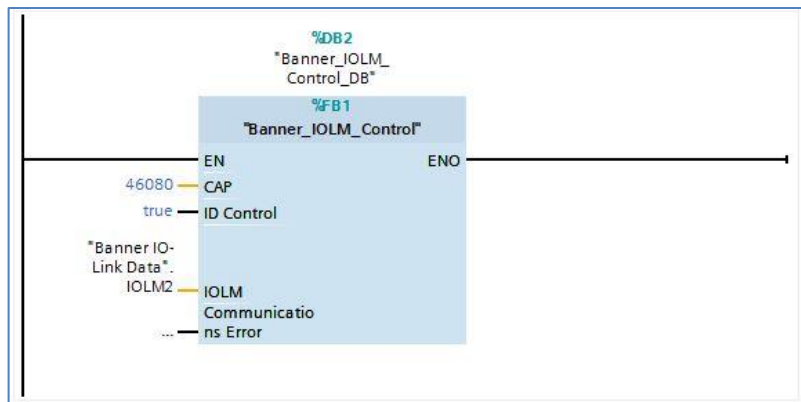


Figure 1: An example using an ifm IO-Link Master

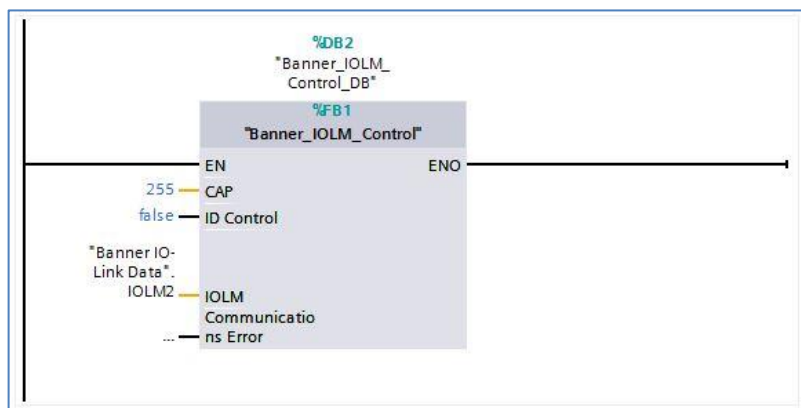


Figure 2: An example using a Balluff IO-Link Master

- 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

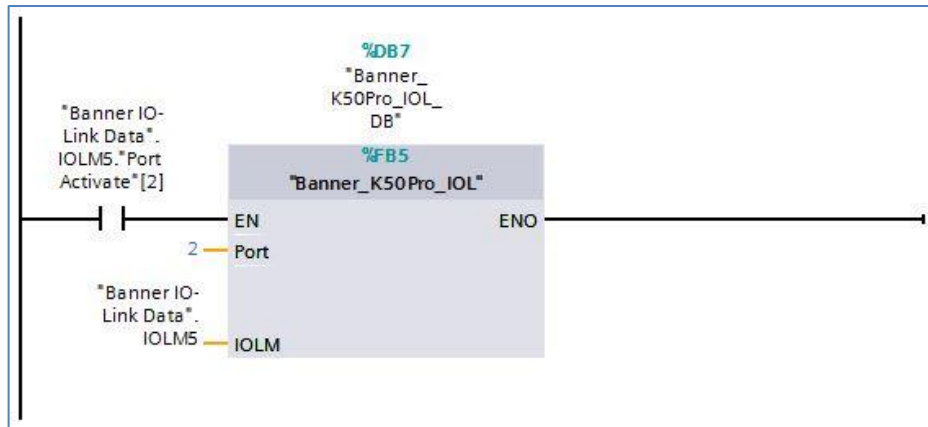
Figure 3: The ID_Array when using an ifm IO-Link Master; device attached to port 6

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)

- Now add the "Banner_K50Pro_IOL" 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 13.

As a final step, the Port Activate bit is added on the same rung as the K50 Pro Touch 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 shine.

















































15. The K50 Pro Touch Parameter Data function block is now set up. Compile the project and download it to the PLC. Go online, then open the K50 Pro Touch data block. When the function block starts out, it does an initial global read of all K50 Pro Touch information. The Read Data section of the data block shows this information.

Banner_K50Pro_IOL_DB				
	Name	Data type	Start value	Monitor value
7	▼ K50Pro Data	"Banner_K50Pro"		
8	Initial Global Read	Bool	false	TRUE
9	Command	Int	0	0
10	▼ Read Data	"Banner_K50Pro_RD"		
11	Master Cycle Time	USInt	0	56
12	Min Cycle Time	USInt	0	50
13	M-Sequence Capability	USInt	0	9
14	IO-Link Version ID	USInt	0	17
15	Process Data Input Length	USInt	0	16
16	Process Data Output Length	USInt	0	137
17	Vendor ID Combined	UDInt	0	451
18	Vendor ID 1	USInt	0	1
19	Vendor ID 2	USInt	0	195
20	Device ID Combined	UDInt	0	393220
21	Device ID 1	USInt	0	6
22	Device ID 2	USInt	0	0
23	Device ID 3	USInt	0	4
24	Serial Number	String	"	'18-10-1521:09:34'
25	► Write Data	"Banner_K50Pro_W..."		
26	Reset	Bool	false	FALSE

16. The Write Data section of the data block shows all the writeable parameter data for the K50 Pro Touch. Simply change a value here and it will be automatically written to the sensor via a one-time acyclic write through the IO-Link Master.

It is almost like the K50 Pro Touch is a PROFINET-speaking device!

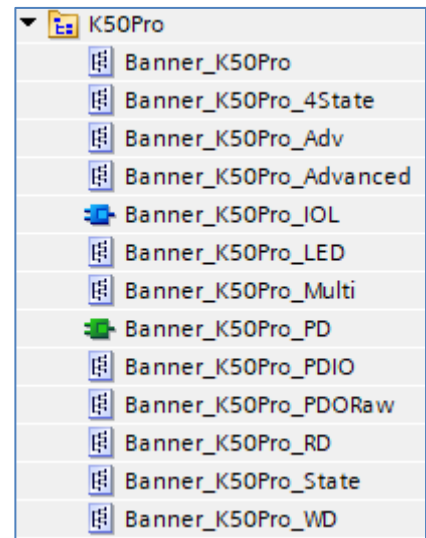
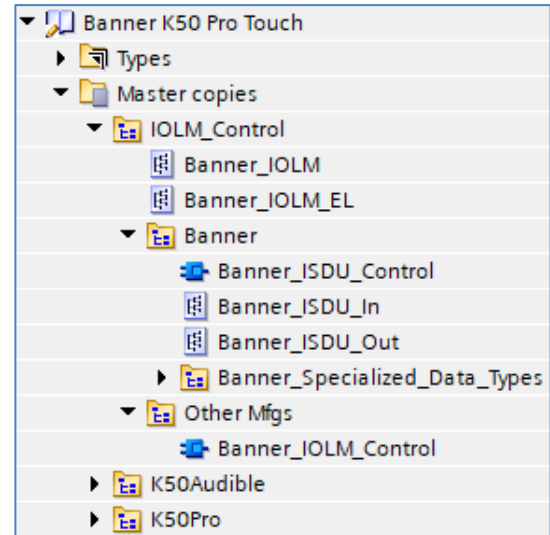
Banner_K50Pro_IOL_DB				
	Name	Data type	Start value	Monitor value
7	  K50Pro Data	"Banner_K50Pro"		
8	  Initial Global Read	Bool	false	TRUE
9	  Command	Int	0	0
10	  Read Data	"Banner_K50Pro_RD"		
11	  Write Data	"Banner_K50Pro_W..."		
12	  System Command	USInt	0	0
13	  Parameter Access Lock	Bool	false	FALSE
14	  Data Storage Lock	Bool	false	FALSE
15	  Local Parameterization Lock	Bool	false	FALSE
16	  Local User Interface Lock	Bool	false	FALSE
17	  Operating Mode	USInt	0	1
18	  Custom Intensity	USInt	0	100
19	  Custom Flash Rate	USInt	0	15
20	  Restrict to Gamut	USInt	0	0
21	  Touch Sensitivity	USInt	0	1
22	  Function	Bool	false	FALSE
23	  Mute Enable	Bool	false	FALSE
24	  OnDelay	UInt	0	0
25	  Output State	Bool	false	TRUE
26	  Off Delay Type	Bool	false	FALSE
27	  Off Delay	UInt	0	0
28	  State Parameters	Array[1..4] of "Ban..."		
29	  Custom 1 - Red	USInt	0	255

Setup of K50 Pro Touch with other IO-Link Masters

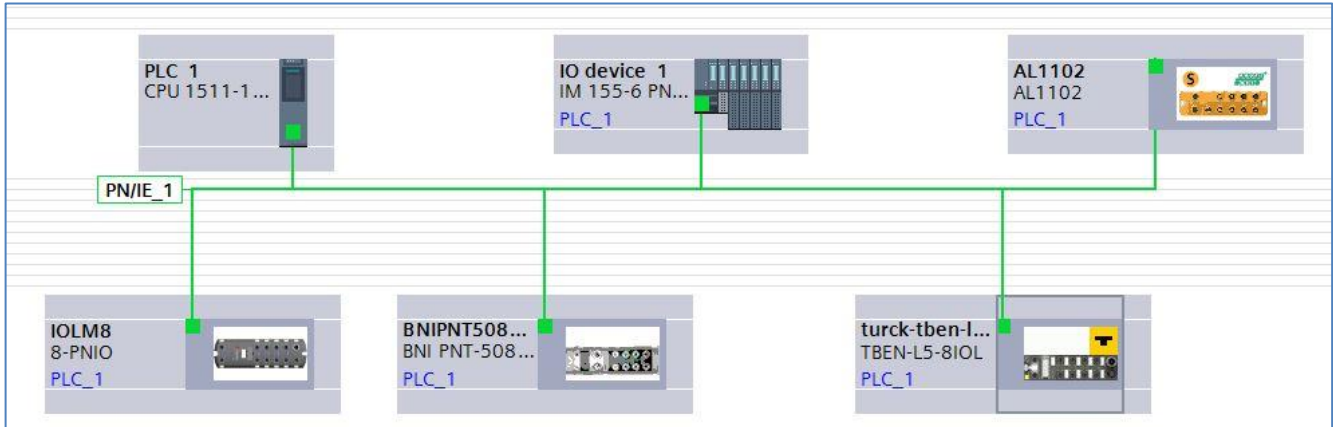
Additional Component Needed

*Siemens IO_LINK_DEVICE function block***Installation Instructions**

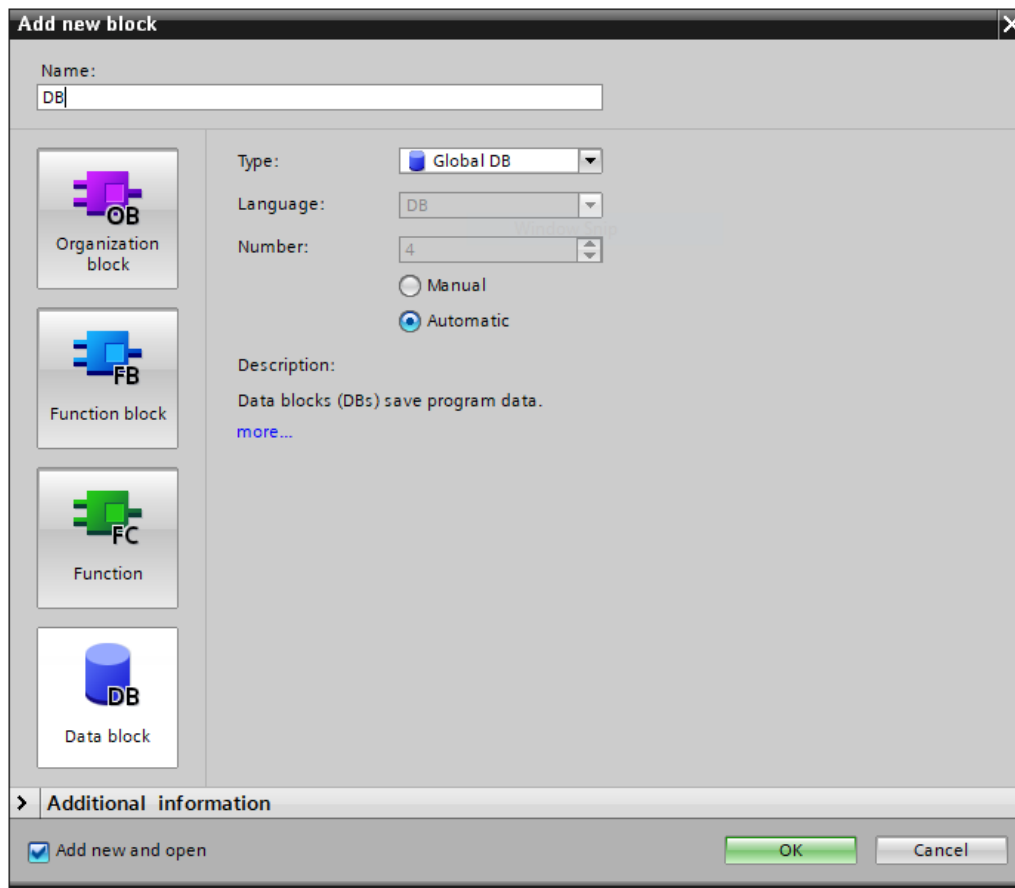
1. The Banner K50 Pro Touch library will now be in the Global Library List. Expand the Master copies section. The K50Audible folder contains elements for both Process Data and Parameter Data connections to a K50 Pro Audible device. As Parameter Data is the focus of this paper, we will concern ourselves with these six items:
Banner_K50Audible,
Banner_K50Audible_Advanced,
Banner_K50Audible_IOL, Banner_K50Audible_RD,
and Banner_K50Audible_WD.
2. Drag Banner_K50Audible_IOL to the Program Blocks area under your PLC.
3. Drag the Banner_K50Audible, Banner_K50Audible_Advanced, Banner_K50Audible_RD, and Banner_K50Audible_WD to the PLC Data Types area under your PLC.
4. We also must prepare for setting up the IO-Link Master. Go to the IOLM_Control section of the Banner IO-Link Library List.
5. Drag Banner_IOLM and Banner_IOLM_EL to the PLC Data Types area under your PLC.
6. Drag the Banner_IOLM_Control to the Program Blocks area under your PLC.
7. Finally, we must bring the Siemens-made IO_LINK_DEVICE function block specific to your PLC into our project. 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.



8. 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.



9. Go to Program blocks. Add a new Data block if necessary. In this example the new data block is named "DB".



10. 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.

Name	Data type
▼ Static	
■ ► IOLM1	"Banner_IOLM"

11. Expand the IOLM1 tag, then expand the Port Controlled section. The Port Controlled tag array determines which of the ports has a function block-controlled Banner IO-Link device plugged into it. Each Port Controlled array tag with **true** as the start value is considered to have such a device connected. Correctly setting this array allows the Device and IO-Link Master function blocks to control the device on that port. Errors will occur if a port without an IO-Link device is set to true.

Name	Data type	Start value
▼ Static		
■ ▼ IOLM1	"Banner_IOLM"	
■ ▼ Port Controlled	Array[1..8] of Bool	
■ Port Controlled[1]	Bool	true
■ Port Controlled[2]	Bool	false
■ Port Controlled[3]	Bool	false
■ Port Controlled[4]	Bool	false
■ Port Controlled[5]	Bool	false
■ Port Controlled[6]	Bool	false
■ Port Controlled[7]	Bool	false
■ Port Controlled[8]	Bool	false
■ ► Port Activate	Array[1..8] of Bool	
■ ► Port Read Request	Array[1..8] of Bool	
■ ► Port Write Request	Array[1..8] of Bool	
■ ► Port RW Complete	Array[1..8] of Bool	
■ ► Port Device Read	Array[1..8] of Bool	
■ ► Transfer Data	Array[0..231] of Byte	
■ Wr_Length	UInt	0
■ Rd_Length	UInt	0
■ IO-Index	Int	0
■ Reset	Bool	false

12. 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 three input variables for this function block:

CAP, ID Control state, and IOLM.

Defining an input variable for the fourth 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”.

Link the “IOLM” input variable to the tag created in step 13.

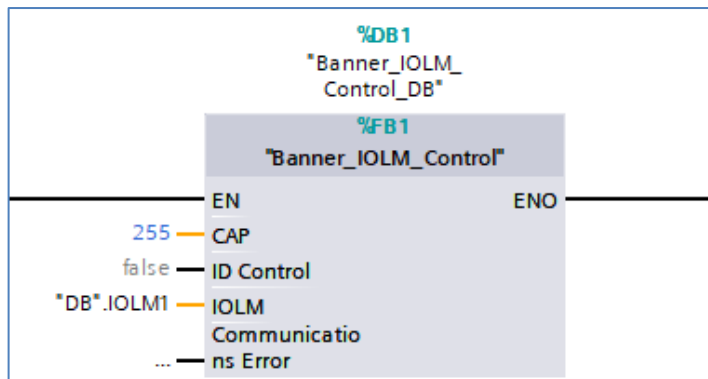


Figure 5: An example using an ifm IO-Link Master

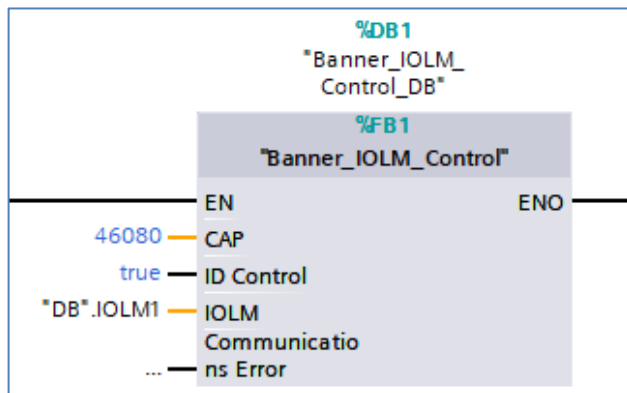


Figure 6: An example using a Balluff IO-Link Master

13. 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

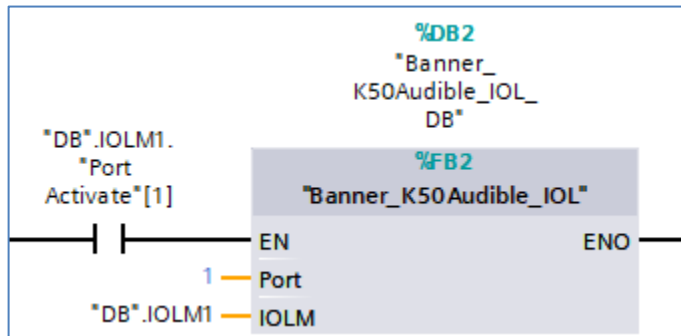
Figure 7: The ID_Array when using an ifm IO-Link Master; device attached to port 6

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

14. Now add the “Banner_K50Audible_IOL” 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 13.

As a final step, the Port Activate bit is added on the same rung as the K50 Pro Audible 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 shine.



15. The K50 Pro Audible Parameter Data function block is now set up. Compile the project and download it to the PLC. Go online, then open the K50 Pro Audible data block. When the function block starts out, it does an initial global read of all K50 Pro Audible information. The Read Data section of the data block shows this information.

Name	Data type	Start value	Monitor value
▼ Static			
■ ▼ Data	*Banner_K50Audibl...		
■ Initial Global Read	Bool	false	TRUE
■ Command	Int	0	0
■ ▼ Read Data	*Banner_K50Audibl...		
■ Master Cycle Time	USInt	0	65
■ Min Cycle Time	USInt	0	50
■ M-Sequence Capability	USInt	0	9
■ IO-Link Version ID	USInt	0	17
■ Process Data Input Len...	USInt	0	16
■ Process Data Output Le..	USInt	0	137
■ Vendor ID Combined	UDInt	0	451
■ Vendor ID 1	USInt	0	1
■ Vendor ID 2	USInt	0	195
■ Device ID Combined	UDInt	0	393228
■ Device ID 1	USInt	0	6
■ Device ID 2	USInt	0	0
■ Device ID 3	USInt	0	12
■ Serial Number	String	"	'22-03-0412:14:57'

16. The Write Data section of the data block shows all the writeable parameter data for the K50 Pro Audible. Simply change a value here and it will be automatically written to the sensor via a one-time acyclic write through the IO-Link Master.

It is almost like the K50 Pro Audible is a PROFINET-speaking device!

Name	Data type	Start value	Monitor value
■ ▼ Write Data	*Banner_K50Audibl...		
■ System Command	USInt	0	0
■ Parameter Access Lock	Bool	false	FALSE
■ Data Storage Lock	Bool	false	FALSE
■ Local Parameterization ..	Bool	false	FALSE
■ Local User Interface Lock	Bool	false	FALSE
■ Operating Mode	USInt	0	3
■ Custom Intensity	USInt	0	100
■ Custom Flash Rate	USInt	0	15
■ Restrict to Gamut	USInt	0	0
■ Touch Sensitivity	USInt	0	1
■ Function	Bool	false	FALSE
■ Mute Enable	Bool	false	FALSE
■ OnDelay	UInt	0	0
■ Output State	Bool	false	TRUE
■ Off Delay Type	Bool	false	FALSE
■ Off Delay	UInt	0	0
■ ► State Parameters	Array[1..4] of *Ban...		

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.

The screenshot displays the Siemens SIMATIC Manager hardware configuration interface. The main window shows a rack of modules for the device 'BNIPNT508105Z015 [BNI PNT-508-105-Z015]'. The 'Device overview' table on the right lists the modules and their addresses. The 'System constants' table at the bottom lists the hardware identifiers for each module. The entry 'BNI_PNT-508-105-Z015_1' with hardware identifier 309 is highlighted in red.

Module	Rack	Slot	I address	Q address
BNIPNT508105Z015	0	0		
PN-IO	0	0 X1		
BNI PNT-508-105-Z015_1	0	1		
IOL_I/O_32/32 byte_1	0	2	324...355	320...351
	0	3		
	0	4		
	0	5		
	0	6		
	0	7		
	0	8		
	0	9		
	0	10		
	0	11		

Name	Type	Hardware identi.	Used by	Comment
BNIPNT508105Z015-PN-IO-port_1-_M12	Hw_Interface	306	PLC_1	
BNIPNT508105Z015-PN-IO-port_2-_M12	Hw_Interface	307	PLC_1	
BNIPNT508105Z015-PN-IO	Hw_Interface	305	PLC_1	
BNIPNT508105Z015-Proxy	Hw_SubModule	304	PLC_1	
BNIPNT508105Z015-Head	Hw_SubModule	308	PLC_1	
BNIPNT508105Z015-BNI_PNT-508-105-Z015_1	Hw_SubModule	309	PLC_1	

Figure 9: 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.

The screenshot displays the SIMATIC Manager interface for the IOLM8 [8-PNIO] device. The 'Device overview' table on the right lists the modules and their addresses. The 'System constants' table at the bottom lists the hardware identifiers, with the 'IOLM8-Head' entry highlighted in red.

Module	Rack	Slot	I address	Q address	Type
IOLM8	0	0			IO-Link Master
Interface	0	0 X1			IO-Link Interface
IO-Link In/Out 32 bytes_1	0	1	68...99	64...95	IO-Link In/Out
IO-Link In/Out 32 bytes_2	0	2	100...131	96...127	IO-Link In/Out
IO-Link In/Out 32 bytes_3	0	3	132...163	128...159	IO-Link In/Out
IO-Link In/Out 32 bytes_4	0	4	164...195	160...191	IO-Link In/Out
IO-Link In/Out 32 bytes_5	0	5	196...227	192...223	IO-Link In/Out
IO-Link In/Out 32 bytes_6	0	6	228...259	224...255	IO-Link In/Out
IO-Link In/Out 32 bytes_7	0	7	260...291	256...287	IO-Link In/Out
IO-Link In/Out 32 bytes_8	0	8	292...323	288...319	IO-Link In/Out
IO-Link In/Out 32 bytes_9	0	9			IO-Link In/Out
IO-Link In/Out 32 bytes_10	0	10			IO-Link In/Out

Name	Type	Hardware identi.	Used by	Comment
IOLM8~Interface~Port_1	Hw_Interface	274	PLC_1	
IOLM8~Interface~Port_2	Hw_Interface	275	PLC_1	
IOLM8~Interface	Hw_Interface	273	PLC_1	
IOLM8~Proxy	Hw_SubModule	272	PLC_1	
IOLM8~Head	Hw_SubModule	276	PLC_1	

Figure 10: 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.

The screenshot displays the Siemens STEP 7 HW Config interface. The main window shows a rack configuration with a Turck TBEN-L5-8IOL module. The 'Device overview' table on the right lists the modules and their addresses. The 'System constants' table at the bottom lists the hardware identifiers for each module, with 'tben-Basic_1' highlighted in red.

Module	Rack	Slot	I address	Q address
tben	0	0		
PN-HO	0	0 X1		
Basic_1	0	Basic	33...36	33...34
	0	IO-Lin...		
	0	IO-Lin...		
	0	IO-Lin...		
	0	IO-Lin...		
	0	IO-Lin...		
IN 32 BYTE/OUT 32 BYTE_1	0	IO-Lin...	356...387	352...383
	0	IO-Lin...		
	0	Diagn...		
	0	IO-Lin...		

Name	Type	Hardware identi.	Used by	Comment
tben-PN-HO-Port_1	Hw_Interface	298	PLC_1	
tben-PN-HO-Port_2	Hw_Interface	299	PLC_1	
tben-PN-HO	Hw_Interface	297	PLC_1	
tben-Proxy	Hw_SubModule	296	PLC_1	
tben-Head	Hw_SubModule	300	PLC_1	
tben-Basic_1	Hw_SubModule	301	PLC_1	

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.

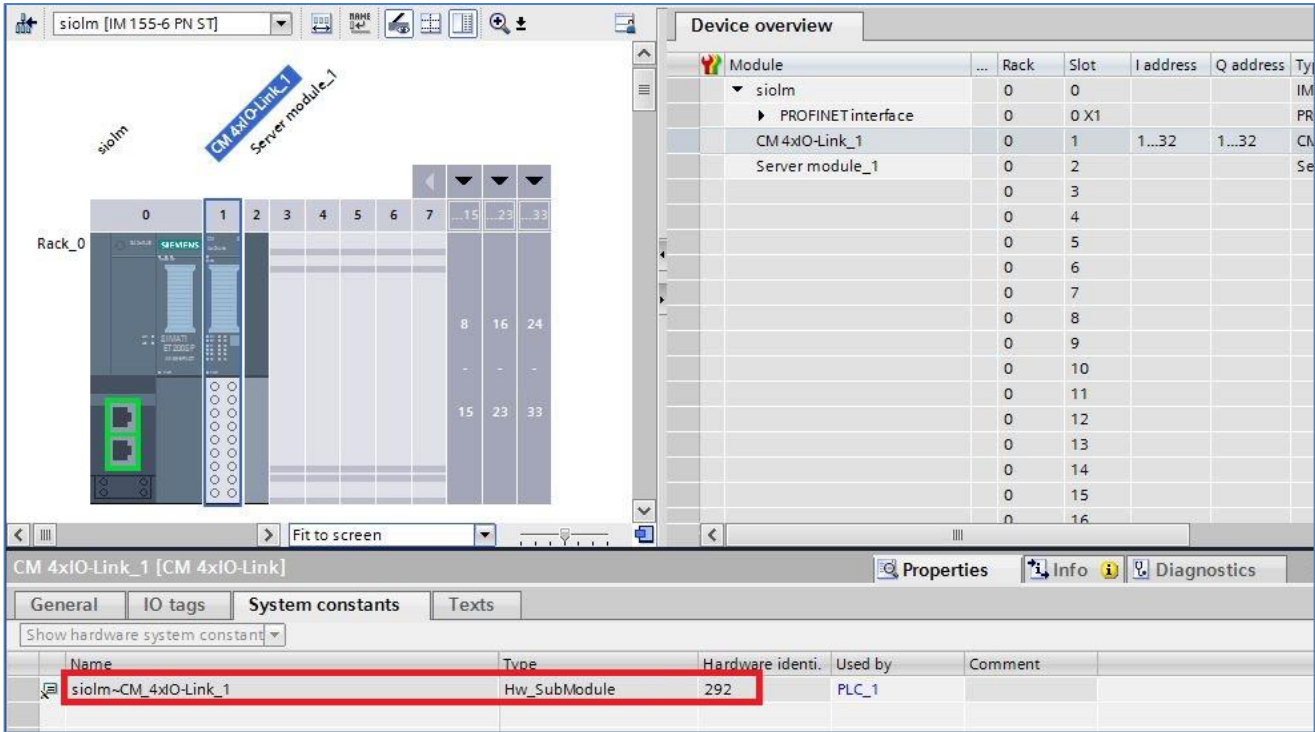


Figure 4: Siemens CM 4xIO-Link Master on ET-200SP. Type this value into the “ID_Array[1]” location.

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 SIMATIC Manager interface. On the left, a rack diagram shows the AL1102 module. The main window is divided into two panes. The top pane, titled 'Device overview', shows a tree structure of the hardware configuration. The bottom pane, titled 'Properties', shows the 'System constants' tab with a table of hardware identifiers.

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		

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).