EZ-Array K2 Process Data Function

11/11/2022

This document covers the installation and use of a function for Siemen's TIA Portal software package. This function handles cyclic IO-Link Process Data In from a Banner EZ-Array K2 via an IO-Link Master to a Siemens PLC. The function covers parsing and display of the EZ-Array K2 Process Data In.

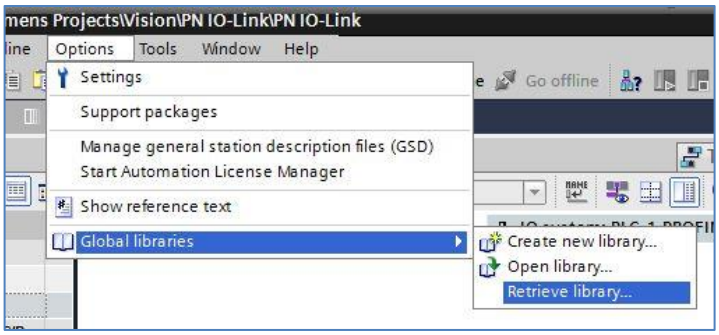
Components

Banner EZ-Array.zal14

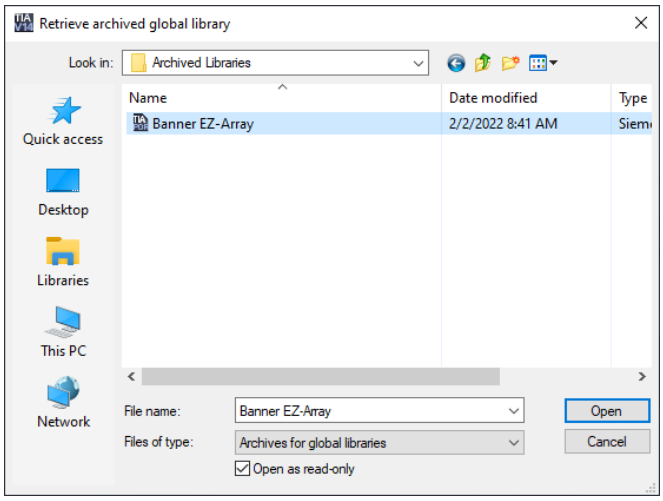
There are two methods for the process data. The first is used when creating a connection to Banner's IO-Link masters. The second set of instructions are for systems using other manufacturer's IO-Link masters.

Installation Instructions

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



3. Select the Banner EZ-Array. Click Open.



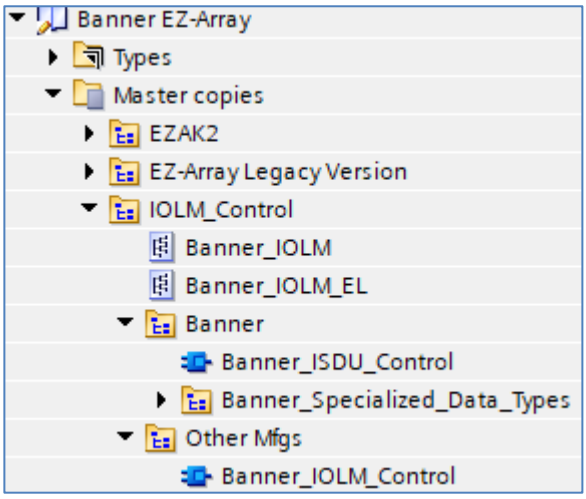
4. The library is now accessible in the Libraries tab.
5. Go to page 3 for Banner IO-Link Masters and to page 6 for all other Io-Link Masters.

Setup of EZAK2 with a Banner DXMR90-4K

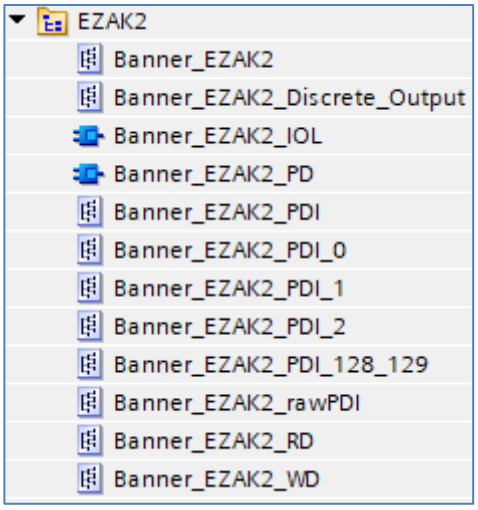
1. Go to Device and Networks to configure the DXMR90-4K. Add the DXMR90-4K if it has yet to be added to the system.
2. Add Banner IO-Link Master Info to Slot 1. This sets the DXMR90-4K for IO-Link mode.
3. Open the IO-Link Generic Devices and select the proper module. The 32/32 byte option has been selected 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 32/32 Byte + Status_1	0	2	10...45	1...46	IO-Link In/Out 32/32 Byte + Status

4. Drag the Banner_EZAK2_PDI, Banner_EZAK2_PDI_0, Banner_EZAK2_PDI_1, Banner_EZAK2_PDI_2, Banner_EZAK2_PDI_128_129, Banner_EZAK2_rawPDI to the PLC Data Types area under your PLC. Banner_EZA_PDI is found in the EZAK2 folder in the library. Drag the Banner_EZAK2_PD to the Program Blocks area.

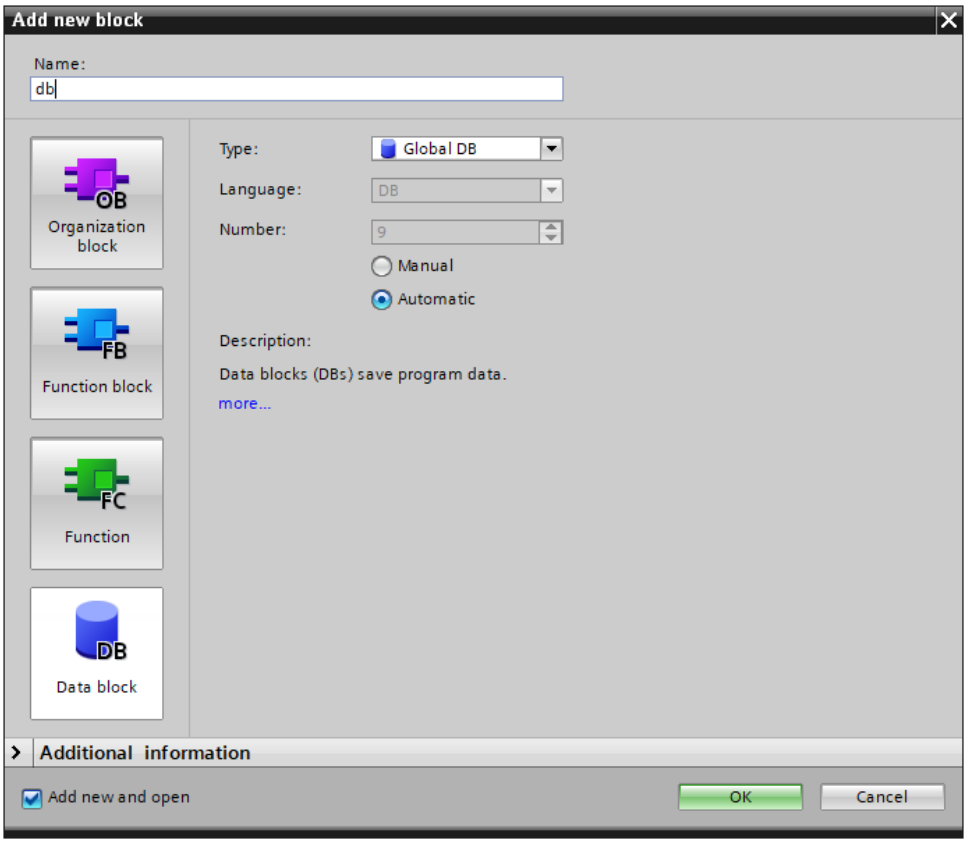


5. Drag the necessary tag from IOLM_Control > Banner > Banner_Specialized_Data_Types. The tag used in this example is “Banner_32in”. This tag represents the full raw process data along with port status information.
6. Go to PLC Tags. Create two tags. One tag is for the full data structure while the second creates a tag to represent the raw Process Data from the IO-Link Master. In this example, Tag table_1 was created, then the tag “EZAK2 IOLM1 01 PDI” was created using a Data Type of “Banner_32In”. 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 is “EZAK2 IOLM1 01 inRaw”. This is the tag that will be used in the Function block.



Name	Data type	Address
EZAK2 IOLM1 01 PDI	"Banner_32In"	%I10.0
EZAK2 IOLM1 01 inRaw	"Banner_EZAK2_rawPDI"	%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 EZAK2. The tag name again calls out the type of sensor, the IO-Link Master, and the port number. Use the data type “Banner_EZAK2_PDI” for the new tag.

Name	Data type
Static	
EZAK2 IOLM1 01 PD	"Banner_EZAK2_PDI"
Array Measurement	"Banner_EZAK2_PDI_0"
Straight Scan	"Banner_EZAK2_PDI_1"
Edge Scan	"Banner_EZAK2_PDI_2"
Channel State OR	"Banner_EZAK2_PDI_128_129"
Channel State AND	"Banner_EZAK2_PDI_128_129"

9. Add the “Banner_EZAK2_PD” function to an OB ladder. Link the “Raw PDI” to the raw Process Data variable from step 5. Link the “PDI” to the parsed Process Data variable from step 7.

   The last variable, “UserPD”, allow the function to correctly interpret the Process Data. In the case of the EZ-Array K2, there are five user-selected modes for the Process Data. This function needs to know what choice has been made in the EZA K2 for this User Process Data variable.

   There are two ways to achieve this goal. We can simply type in the correct number for User Process Data (see Fig. 1), or we can link this EZA K2 Process Data Function to the EZA K2 Parameter data function block (see Fig. 2). See Appendix A for more information about EZAK2 Process Data.

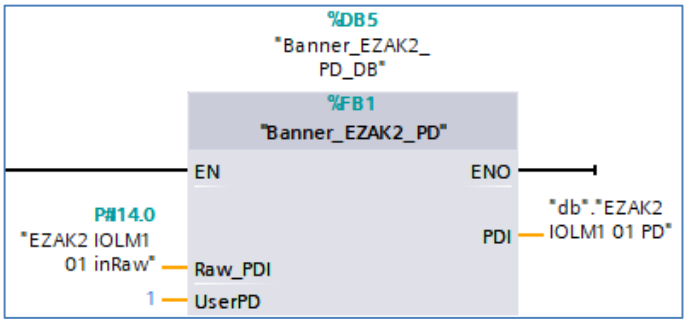


Figure 1: Hand type correct number for User Process Data

NOTE: if you type in the incorrect number (i.e. it does not match the tower light’s current Operational Mode configuration) you will get incorrectly displayed Process Data Out information.

UserPD: the options here are “0” (Active Measurements Only), “1” (Straight Scan Measurements), “2” (Single/Double Edge Measurements), “128” (Channel States/Reduced States using OR), and “129” (Channel States/Reduced States using AND). The default is “0”.

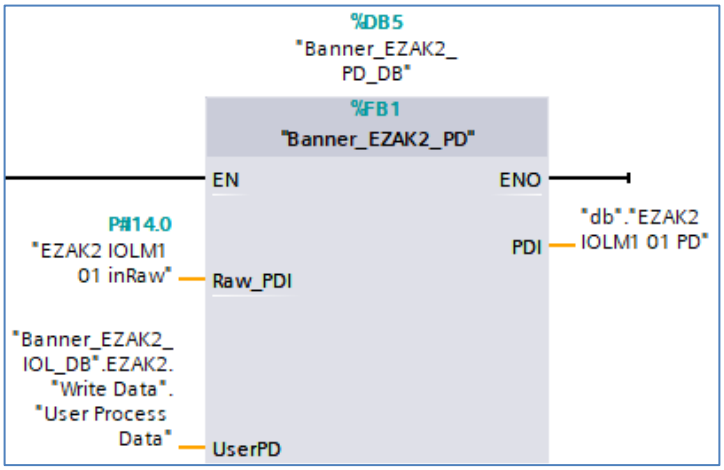


Figure 2: Linking User Process Data variable to EZAK2 Parameter Data Function Block

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 EZA K2, like that shown below.

EZAK2 IOLM1 01 PD	"Banner_EZAK2_PDI"		
Array Measurement	"Banner_EZAK2_PDI_0"		
Measurement 1	UInt	0	1
Measurement 2	UInt	0	93
Straight Scan	"Banner_EZAK2_PDI_1"		
Edge Scan	"Banner_EZAK2_PDI_2"		
Channel State OR	"Banner_EZAK2_PDI_128_129"		
Channel State AND	"Banner_EZAK2_PDI_128_129"		

User Process Data 0 – Active Measurements

EZAK2 IOLM1 01 PD	"Banner_EZAK2_PDI"			
Array Measurement	"Banner_EZAK2_PDI_0"			
Straight Scan	"Banner_EZAK2_PDI_1"			
FBB	UInt	0	26	First Beam Blocked
LBB	UInt	0	43	Last Beam Blocked
TBB	UInt	0	18	Total Beams Blocked
Transitions	UInt	0	2	
CBB	UInt	0	18	Contiguous Beams Blocked
FBM	UInt	0	1	First Beam Made
LBM	UInt	0	120	Last Beam Made
TBM	UInt	0	102	Total Beams Made
CBM	UInt	0	77	Contiguous Beams Made
MBB	UInt	0	34	Middle Beam Blocked
CFBB	UInt	0	26	Contiguous First Beam Blocked
CLBB	UInt	0	43	Contiguous Last Beam Blocked

User Process Data 1 – Straight Scan Measurements

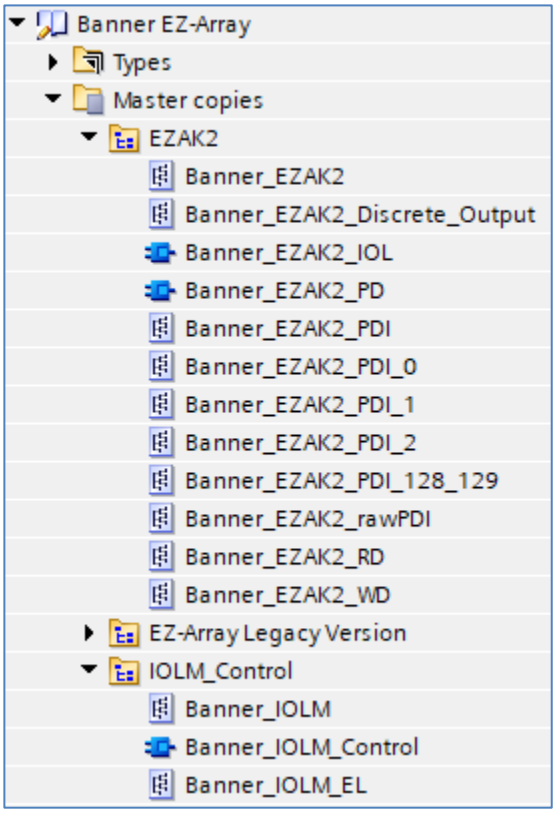
EZAK2 IOLM1 01 PD	"Banner_EZAK2_PDI"			
Array Measurement	"Banner_EZAK2_PDI_0"			
Straight Scan	"Banner_EZAK2_PDI_1"			
Edge Scan	"Banner_EZAK2_PDI_2"			
FBB	UInt	0	26	First Beam Blocked
LBB	UInt	0	43	Last Beam Blocked
TBB	UInt	0	18	Total Beams Blocked
CBB	UInt	0	18	Contiguous Beams Blocked
MBB	UInt	0	34	Middle Beam Blocked
OD	UInt	0	18	Outer Diameter
ID	UInt	0	0	Inner Diameter
CLBB	UInt	0	43	Contiguous Last Beam Blocked
CFBB	UInt	0	26	Contiguous First Beam Blocked
Object 1 FBB	UInt	0	0	
Object 1 LBB	UInt	0	0	
Object 2 FBB	UInt	0	0	
Object 2 LBB	UInt	0	0	
Object 3 FBB	UInt	0	0	
Object 3 LBB	UInt	0	0	

User Process Data 2 – Single/Double Edge Measurements

EZAK2 IOLM1 01 PD	"Banner_EZAK2_PDI"		
Array Measurement	"Banner_EZAK2_PDI_0"		
Straight Scan	"Banner_EZAK2_PDI_1"		
Edge Scan	"Banner_EZAK2_PDI_2"		
Channel State OR	"Banner_EZAK2_PDI_128_129"		
Beam State	Array[1..15] of UInt		
Beam State[1]	UInt	0	0
Beam State[2]	UInt	0	65024
Beam State[3]	UInt	0	2047
Beam State[4]	UInt	0	0
Beam State[5]	UInt	0	0
Beam State[6]	UInt	0	0
Beam State[7]	UInt	0	0
Beam State[8]	UInt	0	0
Beam State[9]	UInt	0	0
Beam State[10]	UInt	0	0
Beam State[11]	UInt	0	0
Beam State[12]	UInt	0	0
Beam State[13]	UInt	0	0
Beam State[14]	UInt	0	0
Beam State[15]	UInt	0	0
Channel State AND	"Banner_EZAK2_PDI_128_129"		

User Process Data 128 Channel States “OR” & 129 Channel States “AND”

Setup of EZ-Array K2 with other IO-Link Masters

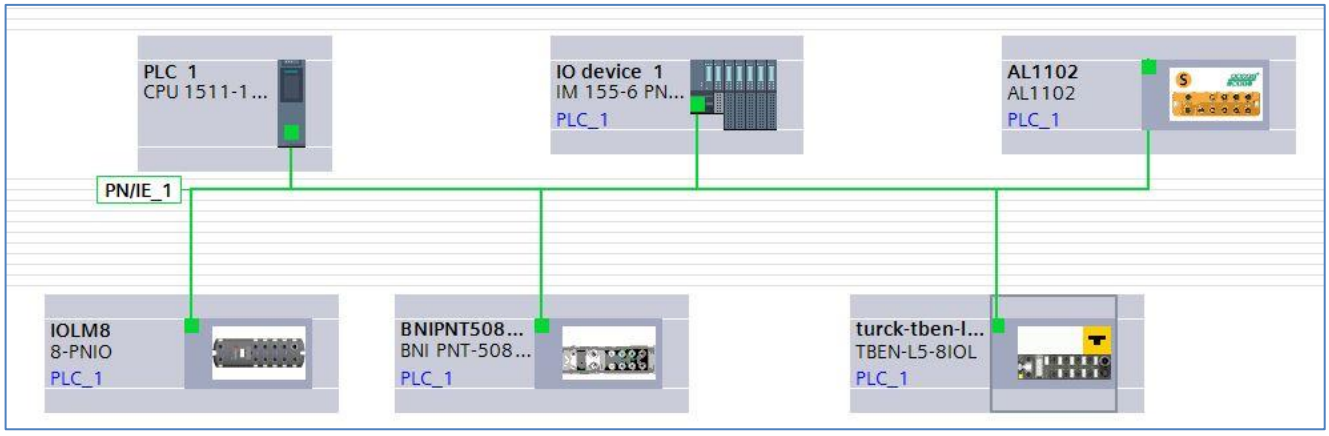


1. The Banner EZ-Array library will now be in the Global Library List. Expand the Master copies section. The EZ-Array folder contains elements for both Process Data and Parameter Data connections to an EZ-Array. As Process Data is the focus of this paper, we will concern ourselves with these two items: Banner_EZAK2_PD, Banner_EZAK2_PDI, Banner_EZAK2_PDI_0, Banner_EZAK2_PDI_1, Banner_EZAK2_PDI_2, Banner_EZAK2_PDI_128_129, and Banner_EZAK2_rawPDI.

2. Drag Banner_EZAK2 _PD to the Program Blocks area under your PLC.

3. Drag the Banner_EZAK2_PDI, Banner_EZAK2_PDI_0, Banner_EZAK2_PDI_1, Banner_EZAK2_PDI_2, Banner_EZAK2_PDI_128_129, and Banner_EZAK2_rawPDI 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. The EZ-Array K2 requires 30 bytes of space for the Process Data.

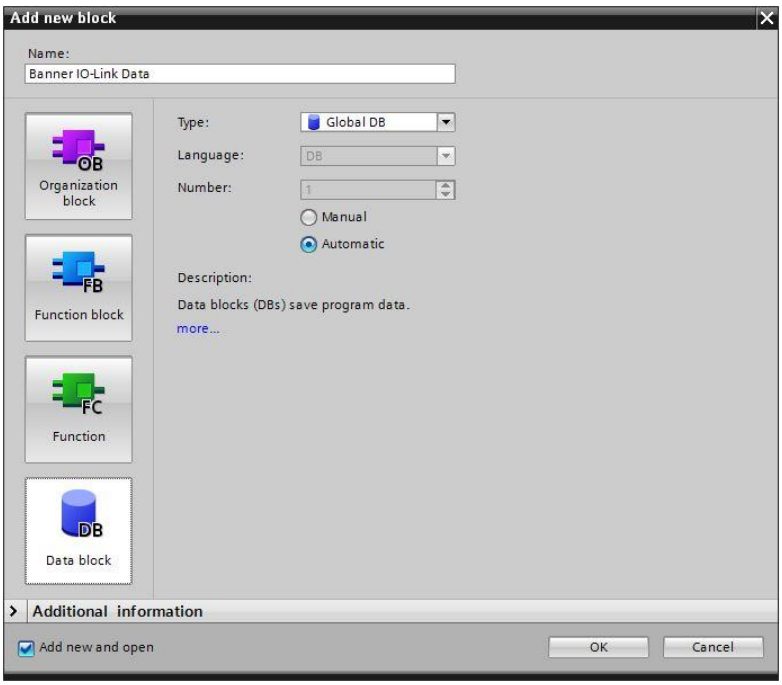
6. Record the "I" address where this EZ-Array Process Data is to be stored, as the address will be required in the next step. In this example, 2 bytes of Process Data In for port 2 on the IO-Link Master will be stored in I68 to I99.

Module	...	Rack	Slot	I address	Q address	Type
IOLM8		0	0			8-PNIO
Interface		0	0 X1			IOLM8
IO-Link In 32 bytes_1		0	1	68...99		IO-Link In 32 bytes

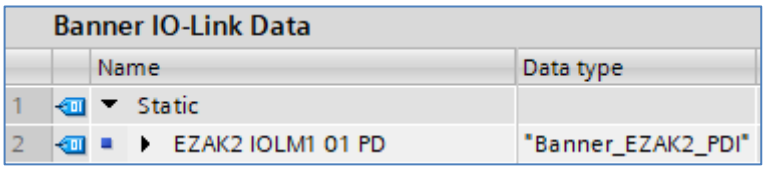
7. Go to PLC Tags. Add a new tag table, then create a new tag to represent the raw Process Data from the IO-Link Master. In this example, Tag table_1 was created, then the tag "EZAK2 IOLM1 01 rawPD" was created using a Data Type of "Banner_EZAK2_rawPDI". This naming convention calls out the type of sensor 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		EZAK2 IOLM1 01 rawPD	"Banner_EZAK2_rawPDI"	%I68.0

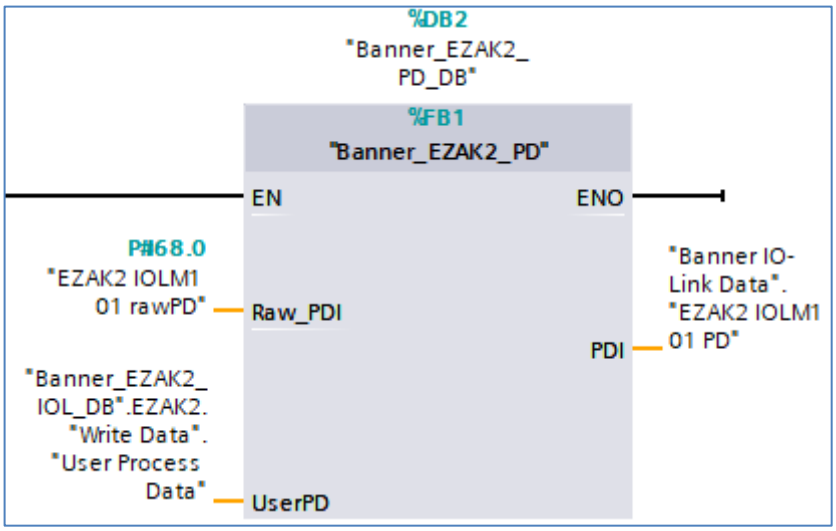
8. Go to Program blocks. Add a new Data block if necessary. In this example the new data block is named "Banner IO-Link Data".



9. In the new data block, create a new tag to represent the parsed Process Data In for our EZ-Array. The tag name again calls out the type of sensor, the IO-Link Master, and the port number. Use the data type “Banner_EZAK2_PDI” for the new tag.



10. Add the “Banner_EZAK2_PD” function block to an OB ladder. Link the “Raw_PDI” to the raw Process Data variable from step 10. Link the “PDI” to the parsed Process Data variable from step 12. Finally, the UserPD needs to either have a number entered or link it to the Parameter data.



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. You should see parsed EZ-Array Process Data In, like one of the options shown below.

EZAK2 IOLM1 01 PD	"Banner_EZAK2_PDI"		
Array Measurement	"Banner_EZAK2_PDI_0"		
Measurement 1	UInt	0	1
Measurement 2	UInt	0	93
Straight Scan	"Banner_EZAK2_PDI_1"		
Edge Scan	"Banner_EZAK2_PDI_2"		
Channel State OR	"Banner_EZAK2_PDI_128_129"		
Channel State AND	"Banner_EZAK2_PDI_128_129"		

User Process Data 0 – Active Measurements

EZAK2 IOLM1 01 PD	"Banner_EZAK2_PDI"			
Array Measurement	"Banner_EZAK2_PDI_0"			
Straight Scan	"Banner_EZAK2_PDI_1"			
FBB	UInt	0	26	First Beam Blocked
LBB	UInt	0	43	Last Beam Blocked
TBB	UInt	0	18	Total Beams Blocked
Transitions	UInt	0	2	
CBB	UInt	0	18	Contiguous Beams Blocked
FBM	UInt	0	1	First Beam Made
LBM	UInt	0	120	Last Beam Made
TBM	UInt	0	102	Total Beams Made
CBM	UInt	0	77	Contiguous Beams Made
MBB	UInt	0	34	Middle Beam Blocked
CFBB	UInt	0	26	Contiguous First Beam Blocked
CLBB	UInt	0	43	Contiguous Last Beam Blocked

User Process Data 1 – Straight Scan Measurements

EZAK2 IOLM1 01 PD	"Banner_EZAK2_PDI"			
Array Measurement	"Banner_EZAK2_PDI_0"			
Straight Scan	"Banner_EZAK2_PDI_1"			
Edge Scan	"Banner_EZAK2_PDI_2"			
FBB	UInt	0	26	First Beam Blocked
LBB	UInt	0	43	Last Beam Blocked
TBB	UInt	0	18	Total Beams Blocked
CBB	UInt	0	18	Contiguous Beams Blocked
MBB	UInt	0	34	Middle Beam Blocked
OD	UInt	0	18	Outer Diameter
ID	UInt	0	0	Inner Diameter
CLBB	UInt	0	43	Contiguous Last Beam Blocked
CFBB	UInt	0	26	Contiguous First Beam Blocked
Object 1 FBB	UInt	0	0	
Object 1 LBB	UInt	0	0	
Object 2 FBB	UInt	0	0	
Object 2 LBB	UInt	0	0	
Object 3 FBB	UInt	0	0	
Object 3 LBB	UInt	0	0	

User Process Data 2 – Single/Double Edge Measurements

EZAK2 IOLM1 01 PD	"Banner_EZAK2_PDI"		
Array Measurement	"Banner_EZAK2_PDI_0"		
Straight Scan	"Banner_EZAK2_PDI_1"		
Edge Scan	"Banner_EZAK2_PDI_2"		
Channel State OR	"Banner_EZAK2_PDI_128_129"		
Beam State	Array[1..15] of UInt		
Beam State[1]	UInt	0	0
Beam State[2]	UInt	0	65024
Beam State[3]	UInt	0	2047
Beam State[4]	UInt	0	0
Beam State[5]	UInt	0	0
Beam State[6]	UInt	0	0
Beam State[7]	UInt	0	0
Beam State[8]	UInt	0	0
Beam State[9]	UInt	0	0
Beam State[10]	UInt	0	0
Beam State[11]	UInt	0	0
Beam State[12]	UInt	0	0
Beam State[13]	UInt	0	0
Beam State[14]	UInt	0	0
Beam State[15]	UInt	0	0
Channel State AND	"Banner_EZAK2_PDI_128_129"		

User Process Data 128 Channel States “OR” & 129 Channel States “AND”

Appendix A EZ-Array K2 Process Data

The EZ-Array K2 has 30 bytes of Process Data In, as shown below. The image shows how the data is organized for the various Process Data. The Parameter Data item User Process controls which of the data is displayed.

octet	0	1	2	3	4	5	6	7
bit offset	239 - 232	231 - 224	223 - 216	215 - 208	207 - 200	199 - 192	191 - 184	183 - 176
subindex	1	1	2	2	3	3	4	4
element bit	15 - 8	7 - 0	15 - 8	7 - 0	15 - 8	7 - 0	15 - 8	7 - 0

octet	8	9	10	11	12	13	14	15
bit offset	175 - 168	167 - 160	159 - 152	151 - 144	143 - 136	135 - 128	127 - 120	119 - 112
subindex	5	5	6	6	7	7	8	8
element bit	15 - 8	7 - 0	15 - 8	7 - 0	15 - 8	7 - 0	15 - 8	7 - 0

octet	16	17	18	19	20	21	22	23
bit offset	111 - 104	103 - 96	95 - 88	87 - 80	79 - 72	71 - 64	63 - 56	55 - 48
subindex	9	9	10	10	11	11	12	12
element bit	15 - 8	7 - 0	15 - 8	7 - 0	15 - 8	7 - 0	15 - 8	7 - 0

octet	24	25	26	27	28	29		
bit offset	47 - 40	39 - 32	31 - 24	23 - 16	15 - 8	7 - 0		
subindex	13	13	14	14	15	15		
element bit	15 - 8	7 - 0	15 - 8	7 - 0	15 - 8	7 - 0		

When the User Process is set as 0 (Active Measurements Only) the below data is displayed.

ProcessData id=PD_ProcessDataMeasurement1 (condition V_UserProcess = 0)

ProcessDataIn "Array Measurement" id=PD_ProcessDataInMeasurement1

bit length: 240
data type: 240-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	224	16-bit UInteger						Active Measurement 1	
2	208	16-bit UInteger						Active Measurement 2	

When the User Process is set as 1 (Straight Scan Measurements) the below data is displayed.

ProcessData id=PD_ProcessDataMeasurement2 (condition V_UserProcess = 1)

ProcessDataIn "Straight Scan Measurements" id=PD_ProcessDataInMeasurement2

bit length: 240
data type: 240-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	224	16-bit UInteger						First Beam Blocked	
2	208	16-bit UInteger						Last Beam Blocked	
3	192	16-bit UInteger						Total Beam Blocked	
4	176	16-bit UInteger						Transitions	
5	160	16-bit UInteger						Contiguous Beam Blocked	
6	144	16-bit UInteger						First Beam Made	
7	128	16-bit UInteger						Last Beam Made	
8	112	16-bit UInteger						Total Beam Made	
9	96	16-bit UInteger						Contiguous Beam Made	
10	80	16-bit UInteger						Middle Beam Blocked	
11	64	16-bit UInteger						Contiguous First Beam Blocked	
12	48	16-bit UInteger						Contiguous Last Beam Blocked	
13	32	16-bit UInteger						No Measurement	
14	16	16-bit UInteger						No Measurement	
15	0	16-bit UInteger						No Measurement	

When the User Process is set as 2 (Single/Double Edge Measurements) the below data is displayed.

ProcessData id=PD_ProcessDataMeasurement3 (condition V_UserProcess = 2)

ProcessDataIn "Edge Scan Measurements" id=PD_ProcessDataInMeasurement3

bit length: 240
data type: 240-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	224	16-bit UInteger						First Beam Blocked	
2	208	16-bit UInteger						Last Beam Blocked	
3	192	16-bit UInteger						Total Beam Blocked	
4	176	16-bit UInteger						Contiguous Beam Blocked	
5	160	16-bit UInteger						Middle Beam Blocked	
6	144	16-bit UInteger						Outer Diameter	
7	128	16-bit UInteger						Inner Diameter	
8	112	16-bit UInteger						Contiguous First Beam Blocked	
9	96	16-bit UInteger						Contiguous Last Beam Blocked	
10	80	16-bit UInteger						Object 1 First Beam Blocked	
11	64	16-bit UInteger						Object 1 Last Beam Blocked	
12	48	16-bit UInteger						Object 2 First Beam Blocked	
13	32	16-bit UInteger						Object 2 Last Beam Blocked	
14	16	16-bit UInteger						Object 3 First Beam Blocked	
15	0	16-bit UInteger						Object 3 Last Beam Blocked	

When the User Process is set as 128 (Channel States “OR”) the below data is displayed.

ProcessData id=PD_ProcessDataChannelState1 (condition V_UserProcess = 128)

ProcessDataIn "Channel State" id=PD_ProcessDataInChannelState1

bit length: 240
data type: 240-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	224	16-bit UInteger						Beam State	
2	208	16-bit UInteger						Beam State	
3	192	16-bit UInteger						Beam State	
4	176	16-bit UInteger						Beam State	
5	160	16-bit UInteger						Beam State	
6	144	16-bit UInteger						Beam State	
7	128	16-bit UInteger						Beam State	
8	112	16-bit UInteger						Beam State	
9	96	16-bit UInteger						Beam State	
10	80	16-bit UInteger						Beam State	
11	64	16-bit UInteger						Beam State	
12	48	16-bit UInteger						Beam State	
13	32	16-bit UInteger						Beam State	
14	16	16-bit UInteger						Beam State	
15	0	16-bit UInteger						Beam State	

When the User Process is set as 129 (Channel States “AND”) the below data is displayed.

ProcessData id=PD_ProcessDataChannelState2 (condition V_UserProcess = 129)

ProcessDataIn "Channel State" id=PD_ProcessDataInChannelState2

bit length: 240
data type: 240-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	224	16-bit UInteger						Beam State	
2	208	16-bit UInteger						Beam State	
3	192	16-bit UInteger						Beam State	
4	176	16-bit UInteger						Beam State	
5	160	16-bit UInteger						Beam State	
6	144	16-bit UInteger						Beam State	
7	128	16-bit UInteger						Beam State	
8	112	16-bit UInteger						Beam State	
9	96	16-bit UInteger						Beam State	
10	80	16-bit UInteger						Beam State	
11	64	16-bit UInteger						Beam State	
12	48	16-bit UInteger						Beam State	
13	32	16-bit UInteger						Beam State	
14	16	16-bit UInteger						Beam State	
15	0	16-bit UInteger						Beam State	

This Process Data is mapped to a specific group of PROFINET addresses.

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