EZ-Array K2 Process Data AOI Guide, v4
September 21st, 2023

This document covers the installation and use of Add-On Instruction (AOI) for the Logix Designer software package from Rockwell Automation. This AOI handles cyclic IO-Link Process Data In from a Banner EZ-Array K2 via an IO-Link Master to an Allen-Bradley PLC. The AOI covers parsing and display of the EZ-Array Process Data In. The AOI has one User Defined Tag data type.

Components

Banner_EZAK2_PD_v4.L5X

UDT Packaged with the AOI

Banner_EZAK2_PDI_0_v4
Banner_EZAK2_PDI_128_v4
Banner_EZAK2_PDI_129_v4
Banner_EZAK2_PDI_1_v4
Banner_EZAK2_PDI_2_v4
Banner_EZAK2_PDI_v4

Other AOIs Available Separately

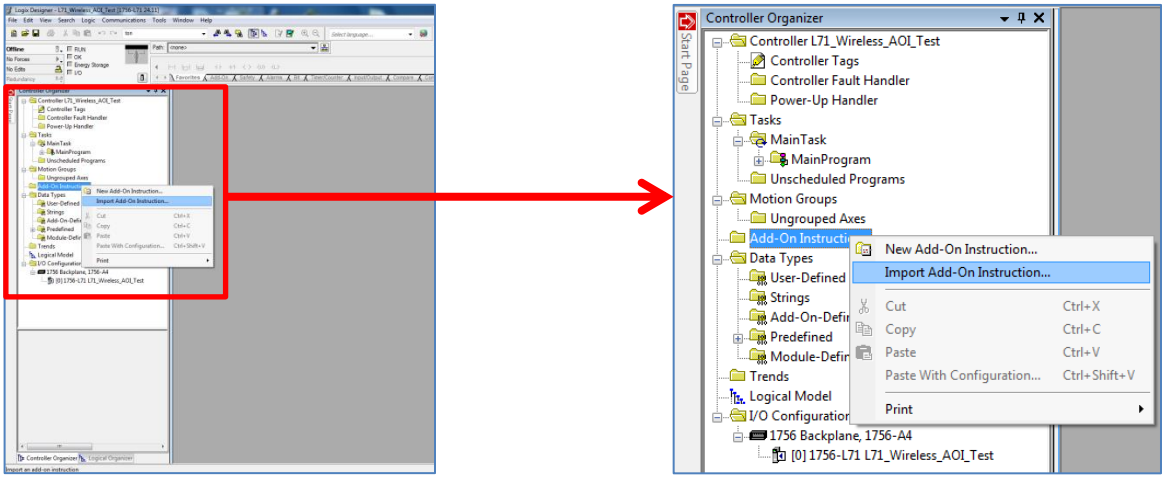
Banner has AOI files for controlling other Banner IO-Link devices and for a variety of IO-Link Masters. Banner also has AOI files for easily handling Banner device Parameter Data.

Contents

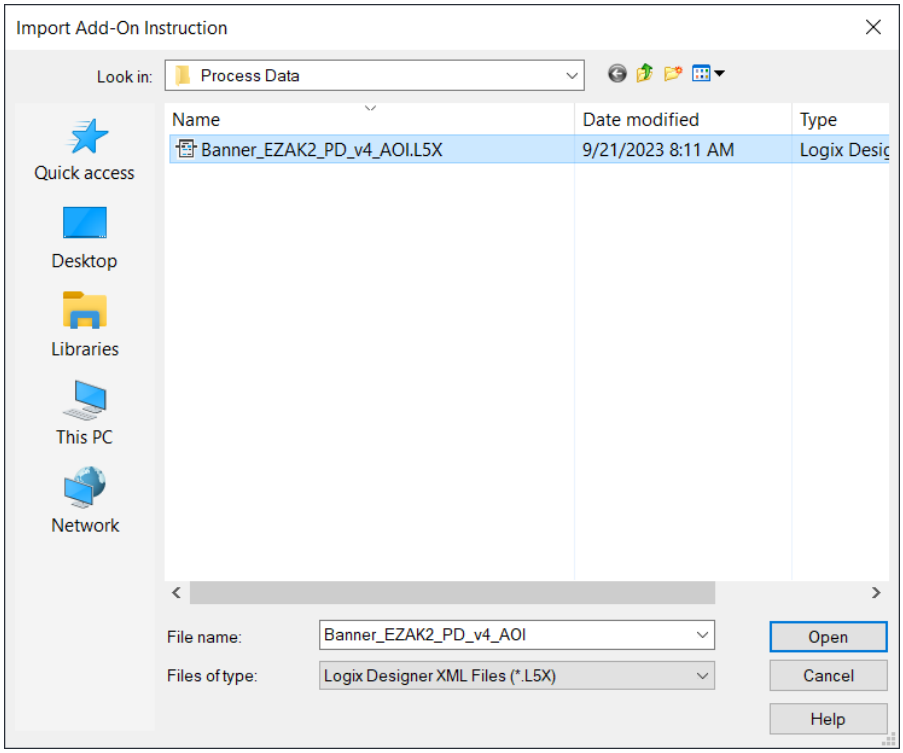
1. Installation Process

This section describes how to install the AOI in Logix Designer software.

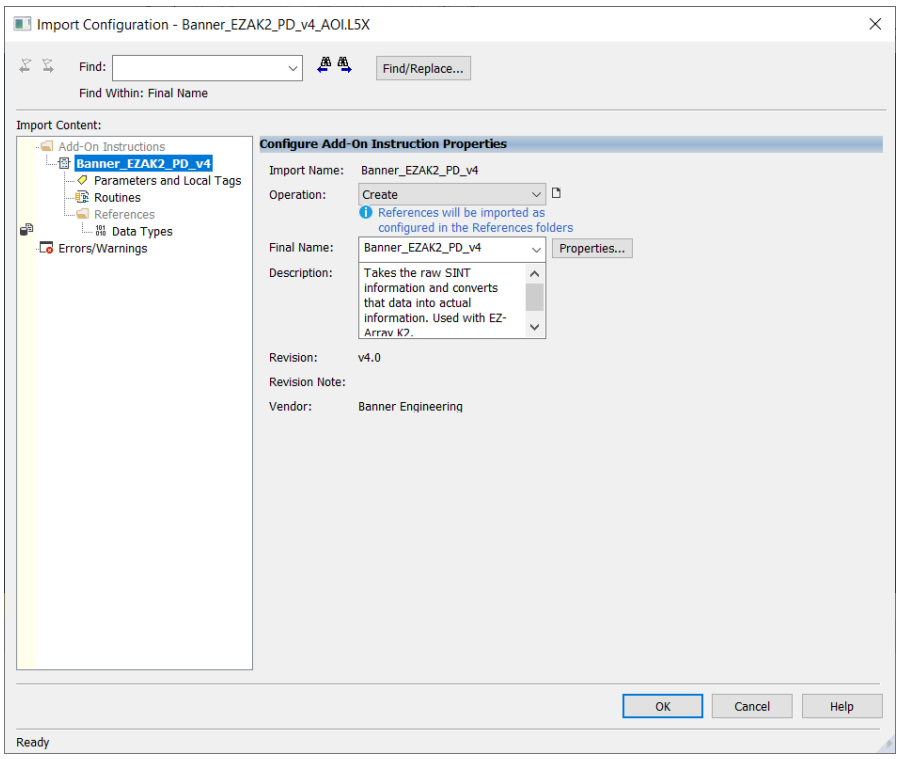
1. Open a project.
2. In the Controller Organizer window, right-click on the Add-On Instruction folder. Select the Import Add-On Instruction option.

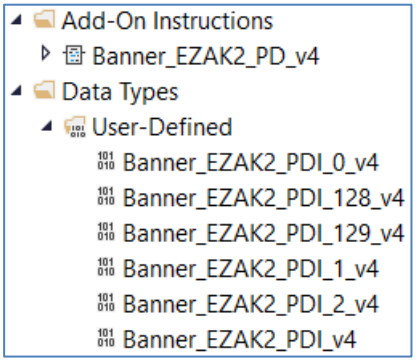


3. Navigate to the correct file location and select the AOI to be installed. In this example the "Banner_EZAK2_PD_v4_AOI.L5X" file will be selected. Click the Open button.



4. The Import Configuration window will pop up. The default selection will create all the necessary items for the AOI. Click the OK button to complete the import process.





5. The AOI is added to the Controller Organizer. window and should look like the picture at left.
6. AOI installation into the Logix Designer software complete.

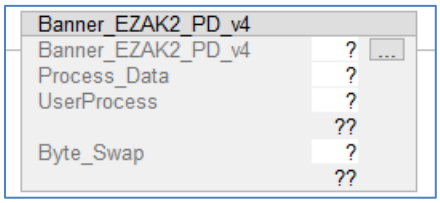
2. Configuring the IO-Link Master

Make an EtherNet/IP connection to the IO-Link Master.

Create an Ethernet communications module for the IO-Link Master device. The controller tags generated include Input (I) and Output (O) Assembly Instances. Each Assembly has a corresponding tag array. Creating this Class 1 EtherNet/IP implicit IO connection will provide PLC access to the IO-Link device Process Data. Each port on the IO-Link Master is given a dedicated group of I and O registers. See the relevant IO-Link Master User's Guide for more information.

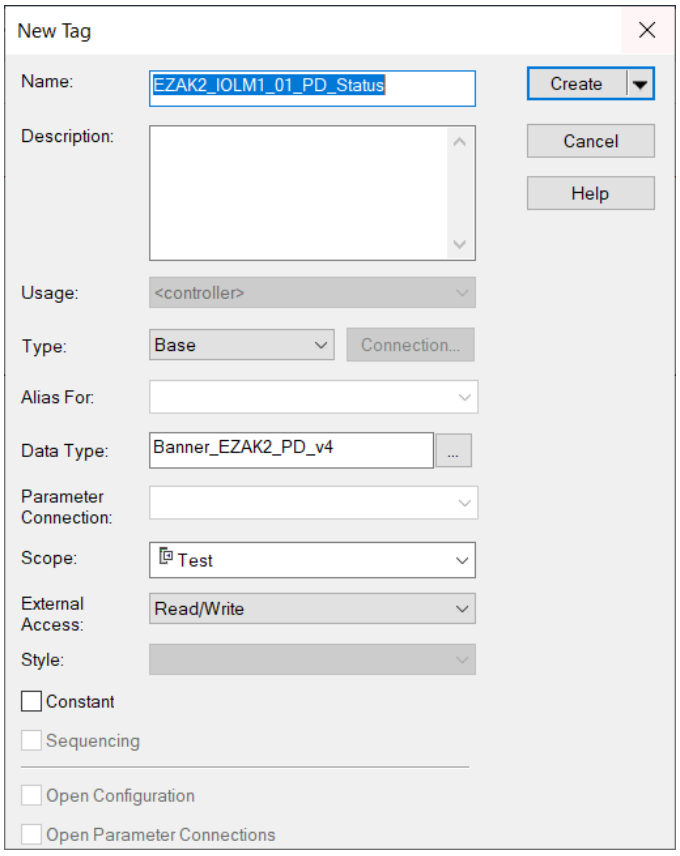
3. Configuring the AOI

1. Add the "Banner_EZAK2_PD_v4" AOI to your ladder logic program. For each of the question marks shown in the instruction we need to create and link a new tag array. The AOI includes a new type of User Defined Tags (UDT): a custom array of tags meant specifically for this AOI.

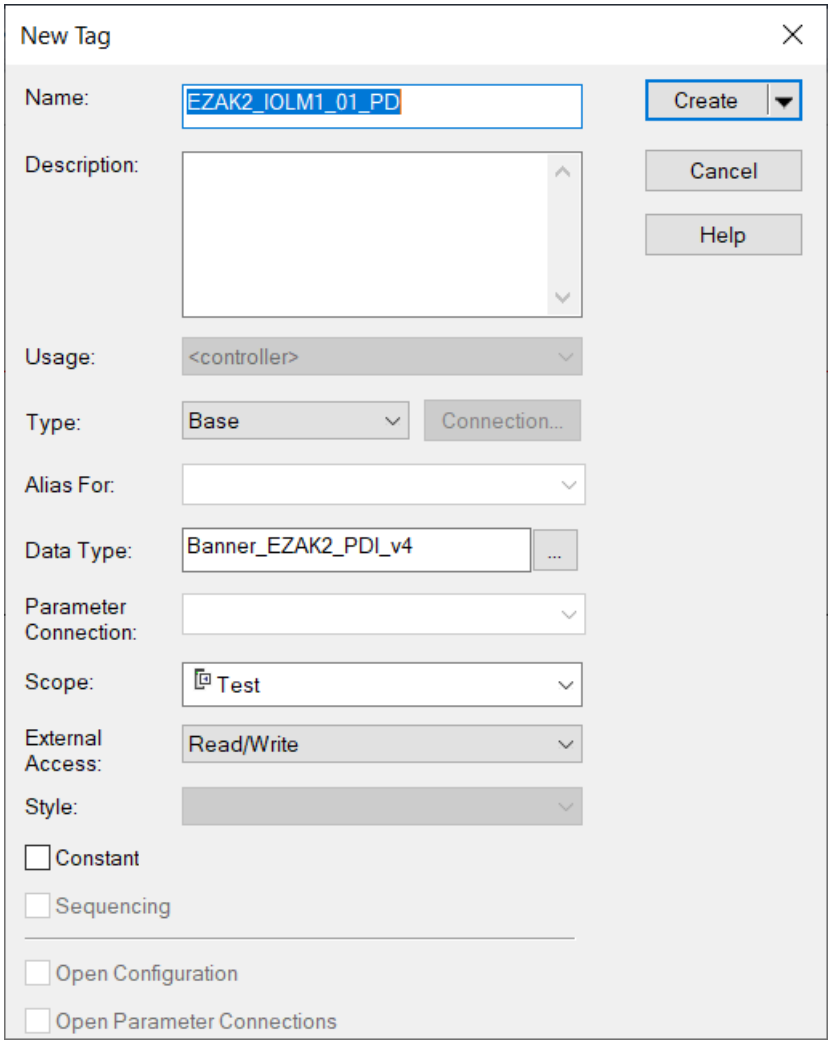


2. In the AOI, right-click on the question mark on the line labeled "Banner_EZA_PD_v4". Click New Tag. Name the new tag. This example uses the name "EZAK2_IOLM1_01_PD_Status". The example naming convention accounts for this being an EZ-Array connected to IO-Link Master #1, port #1, in our program. More masters could be named IOLM2, IOLM3, and different sensors could be connected at other port numbers, etc.

   Note that the Data Type is the User-Defined Data Type (UDT) entitled "Banner_EZA_PD_v4". This custom-made array of registers is specially built to handle the memory needs of this AOI. Click Create to make the tag array.

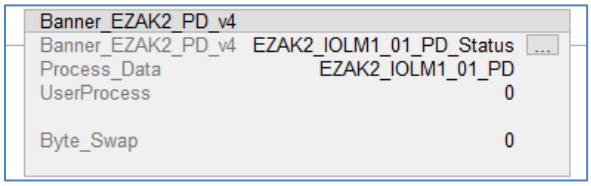


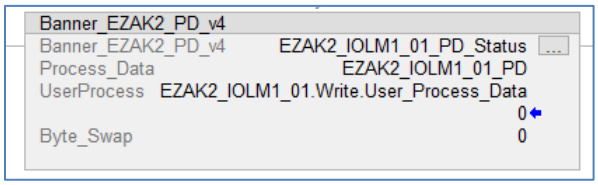
3. Now we will right-click on the question mark on the line labeled “Process Data” in the AOI. Click on “New Tag”. Give the tag a name. This example uses the name “EZAK2_IOLM1_01_PD”. Notice that the Data Type is “Banner_EZAK2_PDI_v4”. Click Create.



4. The next line in the AOI tells the AOI which mode the EZ-Array K2 is currently reporting process data. The array has five different process data layouts. By default, the array is configured for Active Measurement Process Data (mode 0). This AOI needs to know what choices have been made in the sensor for this option.

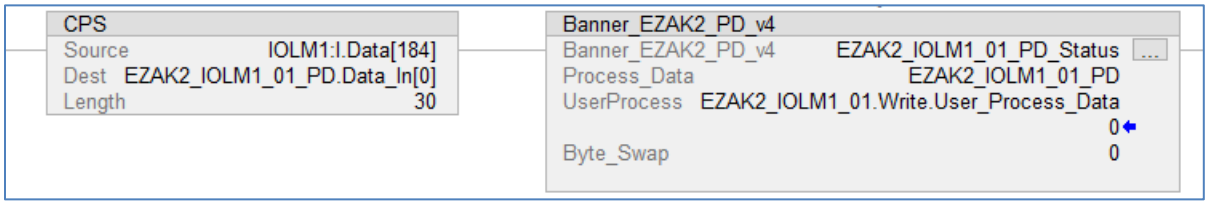
   There are two ways to achieve this goal. We can simply type in the correct number for this selection, or we can link this EZAK2 Process Data AOI to the EZAK2 Parameter Data AOI. See Appendix A for more information about EZAK2 Process Data In.





5. The final setting in the AOI is a setting to account for byte swapping. In the case of the EZ-Array K2, the Process Data is thirty bytes long. IO-Link Masters may read those bytes in either order, so this AOI must be ready to perform a byte swap. Enter a “0” or a “1” to toggle this setting. See Appendix B for more information.

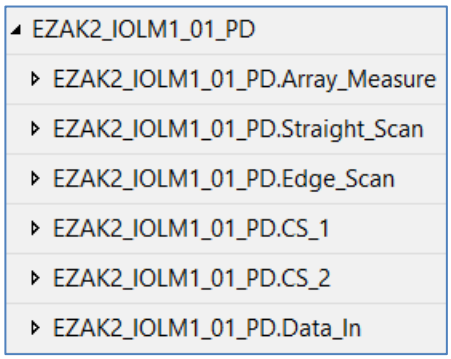
6. The last step requires using the CPS or COP instruction to move the raw byte data into an array that the AOI will use to parse the bytes. Link the DEST to the process data element EZAK2_IOLM1_01_PD.Data_In[0]. The Source is linked the IO-Link Master array element. For this example, a Banner Master is being used. Use Appendix B Table 1 to determine the location in the data. Lastly the length needs to have a value of 30.



The “Banner_EZAK2_PD_v4” AOI is now ready for use.

4. Using the AOI

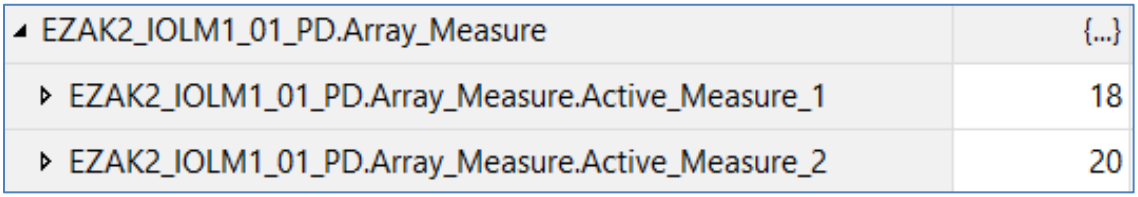
The “Banner_EZAK2_PD_v2” Add-On Instruction has created a group of tags representing the EZ-Array Process Data. In this example the tag was called EZAK2_IOLM3_01_PD. If that tag is expanded five different sections are shown. These sections represent the User Process Data (UPD) settings (0, 1, 2, 128, & 129).



- Array Measure: UPD 0
- Straight Scan: UPD 1
- Edge Scan: UPD 2
- CS_1: UPD 128
- CS_2: UPD 129

Array Measure – 0

The Array Measure process data gives two pieces of information called “Active_Measure_1” and “Active_Measure_2”. These represent the two measurement modes configured by the parameter data. By default, “Active_Measure_1” is configured for Total Beams Blocked (TBB), while “Active_Measure_2” is configured for First Beam Blocked (FBB). In this example, the data shows 18 and 20. This means that 18 beams are blocked in the array while the beam 20 is the first beam blocked out of the array. This information gives the perfect positioning for when a singular object is being sent through the array at one time.



Straight Scan – 1

The Straight Scan process data gives all the measurement modes that are available while the EZ-Array is configured in a straight can mode. Before activating this process data ensure that the scan type is set as 1 (this is the default value).

◢ EZAK2_IOLM1_01_PD.Straight_Scan	{...}
▷ EZAK2_IOLM1_01_PD.Straight_Scan.FBB	20
▷ EZAK2_IOLM1_01_PD.Straight_Scan.LBB	37
▷ EZAK2_IOLM1_01_PD.Straight_Scan.TBB	18
▷ EZAK2_IOLM1_01_PD.Straight_Scan.Transitions	2
▷ EZAK2_IOLM1_01_PD.Straight_Scan.CBB	18
▷ EZAK2_IOLM1_01_PD.Straight_Scan.FBM	1
▷ EZAK2_IOLM1_01_PD.Straight_Scan.LBM	150
▷ EZAK2_IOLM1_01_PD.Straight_Scan.TBM	132
▷ EZAK2_IOLM1_01_PD.Straight_Scan.CBM	113
▷ EZAK2_IOLM1_01_PD.Straight_Scan.MBB	28
▷ EZAK2_IOLM1_01_PD.Straight_Scan.CFBB	20
▷ EZAK2_IOLM1_01_PD.Straight_Scan.CLBB	37

Edge Scan – 2

The Edge Scan process data gives all the measurement modes that are available while the EZ-Array is configured in either Single Edge or Double Edge mode.

EZAK2_IOLM1_01_PD.Edge_Scan	{...}
EZAK2_IOLM1_01_PD.Edge_Scan.FBB	20
EZAK2_IOLM1_01_PD.Edge_Scan.LBB	37
EZAK2_IOLM1_01_PD.Edge_Scan.TBB	18
EZAK2_IOLM1_01_PD.Edge_Scan.CBB	18
EZAK2_IOLM1_01_PD.Edge_Scan.MBB	28
EZAK2_IOLM1_01_PD.Edge_Scan.OD	18
EZAK2_IOLM1_01_PD.Edge_Scan.ID	0
EZAK2_IOLM1_01_PD.Edge_Scan.CFBB	20
EZAK2_IOLM1_01_PD.Edge_Scan.CLBB	37
EZAK2_IOLM1_01_PD.Edge_Scan.Object_1_FBB	0
EZAK2_IOLM1_01_PD.Edge_Scan.Object_1_LBB	0
EZAK2_IOLM1_01_PD.Edge_Scan.Object_2_FBB	0
EZAK2_IOLM1_01_PD.Edge_Scan.Object_2_LBB	0
EZAK2_IOLM1_01_PD.Edge_Scan.Object_3_FBB	0
EZAK2_IOLM1_01_PD.Edge_Scan.Object_3_LBB	0

Channel State 1 – 128

The Channel State 1 process data gives all the beam states. This sets each bit individually. These bits represent a single beam for Arrays of length up to 1200. When the length is 1500 or greater each bit represents 2 beams. These beams are OR'd together to get the value (see table for possible results). When looking at the beam states in the example below it is possible to see that the first beam blocked is beam 20. Beams 20 through 37 are blocked.

Beam 1	Beam 2	Result
0	0	0
0	1	1
1	0	1
1	1	1

◢ EZAK2_IOLM1_01_PD.CS_1	{...}
◢ EZAK2_IOLM1_01_PD.CS_1.Beams	{...}
▷ EZAK2_IOLM1_01_PD.CS_1.Beams[0]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_1.Beams[1]	2#1111_1111_1111_1000
▷ EZAK2_IOLM1_01_PD.CS_1.Beams[2]	2#0000_0000_0001_1111
▷ EZAK2_IOLM1_01_PD.CS_1.Beams[3]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_1.Beams[4]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_1.Beams[5]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_1.Beams[6]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_1.Beams[7]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_1.Beams[8]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_1.Beams[9]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_1.Beams[10]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_1.Beams[11]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_1.Beams[12]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_1.Beams[13]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_1.Beams[14]	2#0000_0000_0000_0000

Channel State 2 – 129

The Edge Scan process data gives all the beam states. This sets each bit individually. These bits represent a single beam for Arrays of length up to 1200. When the length is 1500 or greater each bit represents 2 beams. These beams are AND'd together to get the value (see table for possible results). When looking at the beam states in the example below it is possible to see that the first beam blocked is beam 20. Beams 20 through 37 are blocked.

Beam 1	Beam 2	Result
0	0	0
0	1	0
1	0	0
1	1	1

◢ EZAK2_IOLM1_01_PD.CS_2	{...}
◢ EZAK2_IOLM1_01_PD.CS_2.Beams	{...}
▷ EZAK2_IOLM1_01_PD.CS_2.Beams[0]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_2.Beams[1]	2#1111_1111_1111_1000
▷ EZAK2_IOLM1_01_PD.CS_2.Beams[2]	2#0000_0000_0001_1111
▷ EZAK2_IOLM1_01_PD.CS_2.Beams[3]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_2.Beams[4]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_2.Beams[5]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_2.Beams[6]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_2.Beams[7]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_2.Beams[8]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_2.Beams[9]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_2.Beams[10]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_2.Beams[11]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_2.Beams[12]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_2.Beams[13]	2#0000_0000_0000_0000
▷ EZAK2_IOLM1_01_PD.CS_2.Beams[14]	2#0000_0000_0000_0000

Appendix A EZ-Array Process Data In

The EZ-Array has 30 bytes of Process Data In for five different configurations. These different configurations are shown below.

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	

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

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	

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

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	

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	

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	

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		

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	

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	

This Process Data is mapped to a specific group of EtherNet/IP registers. User Process Data 0, 1, and 2 take the raw values and divide them by 4. User Process Data 128 and 129 represents a beam state. These two User Process Data types stores a single beams state per bit for smaller length arrays. Longer arrays require two beams to be represents by a singular bit. This is accomplished by either “ORing” or “ANDing” the two beams together. Value 128 “ORs” the beams together, while 129 “ANDs” the beams together.

Appendix B IO-Link Master Cheat Sheet

Different IO-Link Masters behave differently in several ways. For one, the register locations where Process Data is stored varies. For another, some IO-Link Masters require byte-swapping and/or word-swapping. The tables below aim to define some of these differences. Note that these numbers are when using all default settings. IO-Link Masters can change the register locations to which Process Data is mapped in response to non-default, optional settings. See relevant IO-Link Master documentation for more information.

PDI (Process Data In) is found in the IO-Link Master's T->O (PLC "Input") Assembly Instance.

PDO (Process Data Out) is found in the IO-Link Master's O->T (PLC "Output") Assembly Instance.

Table 1. First Register of Process Data "SINT0"

	Allen-Bradley*		Comtrol		Balluff		Turck		ifm		Banner	
Port	PDI	PDO	PDI	PDO	PDI	PDO	PDI	PDO	PDI	PDO	PDI	PDO
1	I.Ch0Data[0]	O.Ch0Data[0]	4	0	8	6	6	4	190	46	184	182
2	I.Ch1Data[0]	O.Ch1Data[0]	40	32	56	38	38	36	222	78	218	216
3	I.Ch2Data[0]	O.Ch2Data[0]	76	64	104	70	70	68	254	110	252	250
4	I.Ch3Data[0]	O.Ch3Data[0]	112	96	152	102	102	100	286	142	286	284
5	I.Ch4Data[0]	O.Ch4Data[0]	148	128	200	134	134	132	318	174	320	318
6	I.Ch5Data[0]	O.Ch5Data[0]	184	160	248	166	166	164	350	206	354	352
7	I.Ch6Data[0]	O.Ch6Data[0]	220	192	296	198	198	196	382	238	388	386
8	I.Ch7Data[0]	O.Ch7Data[0]	256	224	344	230	230	228	414	270	422	420

*see relevant Banner Allen-Bradley IO-Link Master AOI Guide and Allen-Bradley User Guides for more information on using device IODD files to aid in integration.

Note: Murr IO-Link Masters have configurable process data. Refer to the Murr IO-Link Master Instruction Manual for Process Data mappings.

Table 2. Byte-Swap

IO-Link Master	Byte Swap
Allen-Bradley	0
Comtrol	1
Balluff	0
Turck	1
ifm	1
Murr	0
Banner	0

Specific hardware used in both tables (all default settings):

- Allen-Bradley Armor Block I/O IO-Link Master (1732E-8IOLM12R)
- Comtrol 8-EIP IO-Link Master (99608-8)
- Balluff BNI006A (BNI EIP-508-105-Z015)
- Turck TBEN-L5-8IOL
- ifm AL1122
- Murr Impact67 E DIO 12 DIO4/IOL4 4P (Art.-No. 55144)

Banner IO-Link Masters (DXMR90-4K) have a port status register. The register gives the status of the port. It gives information on if the port has an IO-Link device connected and if Process Data is valid. This is optional information but is useful for troubleshooting. The data comes into the PLC as bytes while the literature shows the value as a word. The table below gives the upper and lower byte data location in the PLC. The upper byte includes bits 15 through 8, while the lower byte has bits 7 through 0.

IO-Link Master Port	Upper Bits 15 - 8	Lower Bits 7 - 0
1	182	183
2	216	217
3	250	251
4	284	285
5	318	319
6	352	353
7	386	387
8	420	421

Port Status:

Bit0 = Connected?
Bit1 = Process Data Valid?
Bit2 = Event Pending?
Bit3 = Ready for ISDU?
Bit4 = Pin4 SIO State
Bit5 = Pin2 SIO State

Bit6-7 = Pin4 Mode:
SDCI Mode = 0
SIO Input Mode = 1
SIO Output Mode = 2

Bit8-10 = Pin2 Mode:
Disabled = 0
Input Normal = 1
Output = 2
Diagnostic Input = 3
Inverted Input = 4