SD50 Process Data AOI Guide, v6
November 17th, 2026

This document covers the installation and use of an Add-On Instruction (AOI) for the Logix Designer software package from Rockwell Automation. This AOI handles cyclic IO-Link Process Data Out to a Banner SD50 device via an IO-Link Master connected to an Allen-Bradley PLC. The AOI covers parsing and display of the SD50 Pro Process Data Out. The AOI has five User Defined Tag data types.

NOTE: AB 5032/5034 IO-Link Masters do not use this Process Data AOI. The 5032 already breaks out the process data automatically. Use those tags.

Components
Banner_SD50_PD_v6_AOI.L5X

Packaged with the AOI
Banner_SD50_PDIO_v6
Banner_SD50_PD_Counter_v6
Banner_SD50_PD_Message_v6
Banner_SD50_PD_Run_v6
Banner_SD50_PD_Timer_v6

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 Process Data.

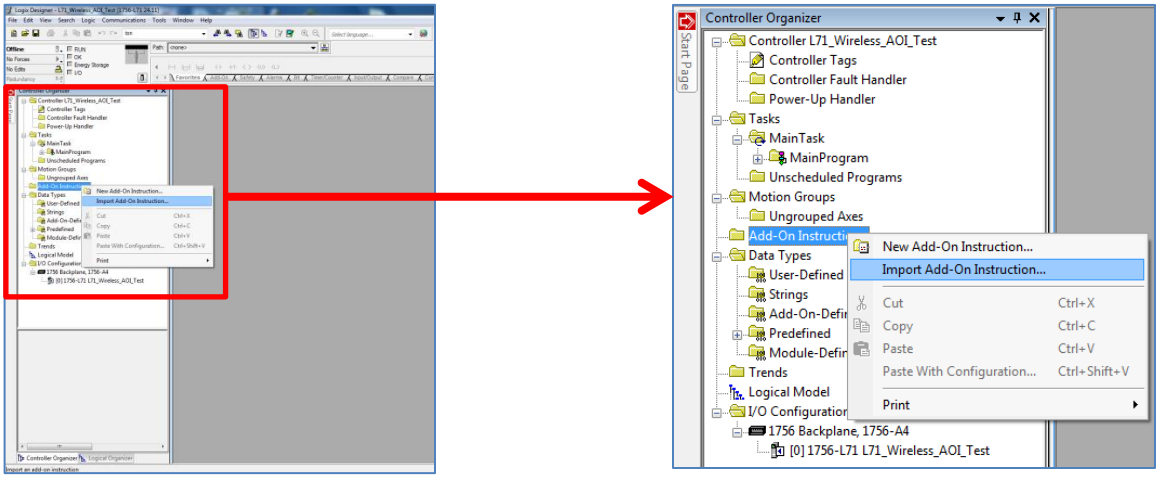
Contents

1. Installation Process 1
2. Configuring the IO-Link Master 2
3. Configuring the AOI 3
4. Using the AOI 6
5. String Construction 7
Appendix A SD50 Process Data 9
Appendix B IO-Link Master Cheat Sheet 12

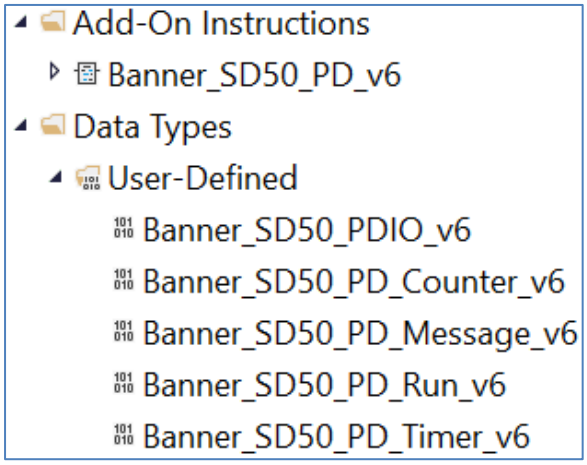
1. Installation Process

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_SD50_PD_v6_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.
6. AOI Installation into the Logix Designer software complete.



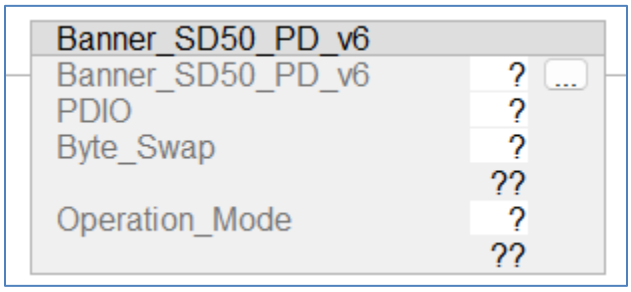
2. Configuring the IO-Link Master

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 the 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_SD50_PD_v6" 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_SD50_PD_v6". Click New Tag. Name the new tag. This example uses the name "SD50_ IOLM1_01_PD_Status". The example naming convention accounts for this being an SD50 Pro device 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_SD50_PD_v6". This custom-made array of registers is specially built to handle the memory needs of this AOI. Click Create to make the tag array.

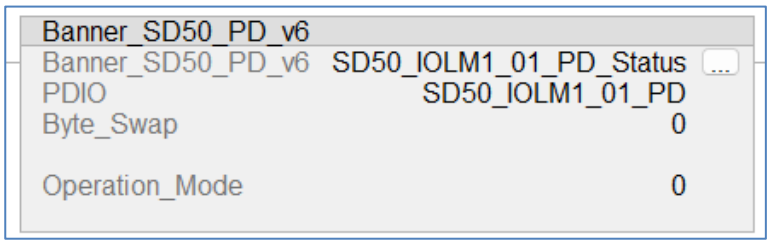
3. Now we will right-click on the question mark on the line labeled "Process_Data" in the AOI. Click on "New Tag". Give the tag a name. This example uses the name "SD50_IOLM1_01_PD". Notice that the Data Type is "Banner_SD50_PDIO_v6". Click Create.

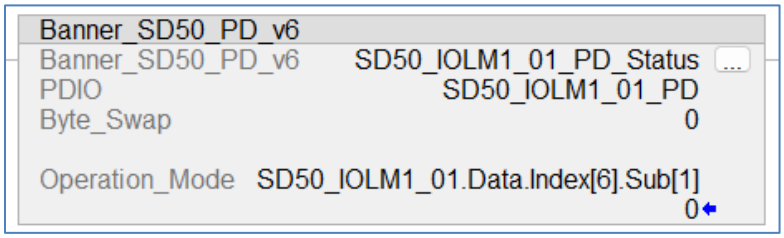
   This array will handle the displaying of the parsed Process Data Out for the SD50 Pro device.

4. The next line in the AOI is a setting to account for byte swapping. In the case of the SD50 Pro, the Process Data Out is 32 bytes long. IO-Link Masters may read each pair of 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.

5. The line labeled “Operation Mode” allows the AOI to know which of four possible Process Data Out definitions is currently in use. The choices for this setting are “0” (Run Mode), “1” (Message mode), “2” (Measurement mode), “3” (Timer mode), and “4” (Counter mode). The default setting is “0” (Run mode). This AOI needs to know which mode selection has been made in the device.

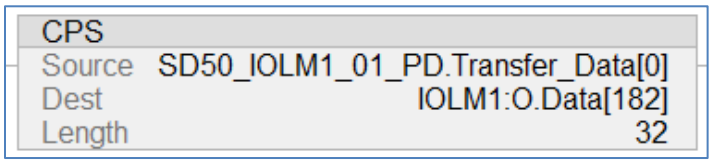
   There are two ways to achieve this goal. We can simply type in the correct number as a constant, or we can link this SD50 Pro Process Data AOI to the SD50 Pro Parameter Data AOI. See Appendix A for more information about SD50 Pro Process Data.



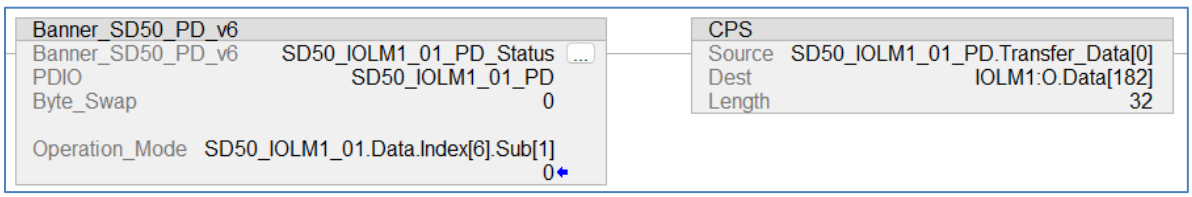


   NOTE: if you type in the incorrect number (i.e. it does not match the device’s current configuration) you will get incorrectly displayed Process Data Out information.

6. The final step required before we download and run the SD50 Pro Process Data AOI involves a File Synchronous Copy (CPS) instruction. A CPS instruction is added to the AOI rung, after the AOI. This CPS instruction is used to copy Process Data Out from the AOI into the raw Process Data Out registers used by the IO-Link Master. See Appendix B for more information. In this example, we will connect the AOI’s “Transfer_Data[0]” to the starting byte location for port 1 in the Process Data Out. In this example, that is byte 182. The size to be copied is 32 bytes.

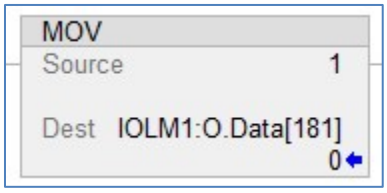


   Here is what the entire rung looks like when completed.



If a Banner IO-Link Master is being used, setup a Move block. Send a 1 to the Activate Outputs array value (see table for each port's value). As an example, if port 1 needs the process data outputs active then send a 1 to 181.

IO-Link Master Port	Activate Outputs
1	181
2	215
3	249
4	283
5	317
6	351
7	385
8	419



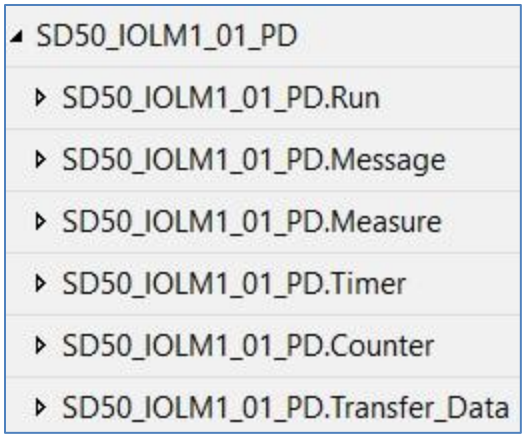
The "Banner_SD50_PD_v6" AOI is now ready for use.

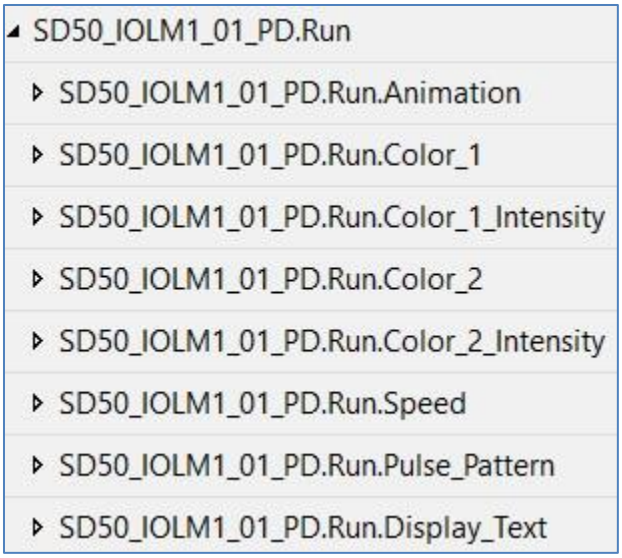
4. Using the AOI

The “Banner_SD50_PD_v6” Add-On Instruction has created a group of tags representing the SD50 Pro Process Data, broken out into its component parts.

Look in the Controller Tags to find the name you used above. This example used the name “SD50_IOLM1_01_PD”. The tag array, seen below, has individual pieces of information instead of unlabeled bits.

Each operating mode for the Process Data Out has its own tag array. If the SD50 Pro device is in operating mode “0” (Run mode), use the tags found under the “Run” tag, as seen below. If the operating mode is “1” (Message mode), use the corresponding tags in the “Message” tag instead. Measurement, timer, and Counter follow a similar pattern. The example below shows the Run process data. This is the default process data mode.





Enter a String into the “Display Text” tag and that string will appear on the SD50. You can have up to 29 characters in the string. See the following section for tips on constructing strings. Setting the Animation to a value of 1 turns on the color LEDs for the SD50. See the description for the various options. The color is changed by manipulating the Color 1 tag. Check the description column for all possible settings.

5. String Construction

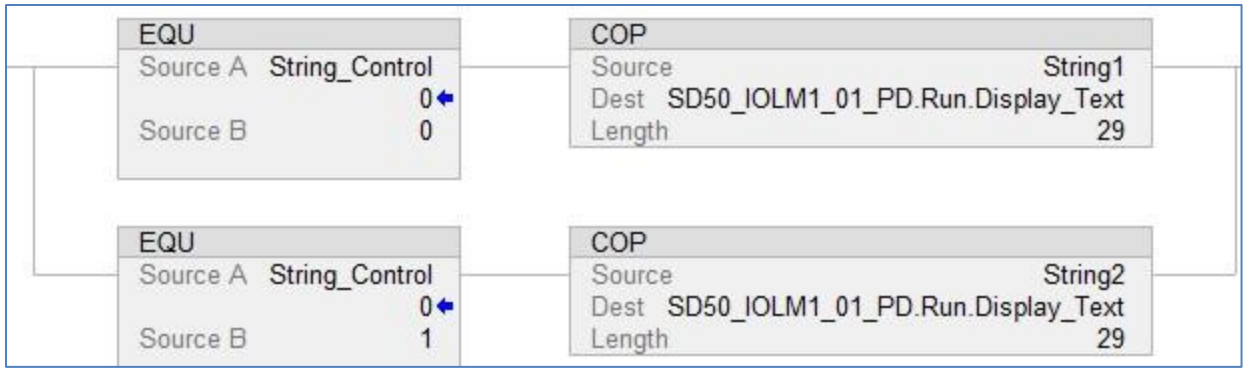
This section will go over how to how String Constants and variable length Strings into the “Display_Text” tag in the Process Data AOI. Care must be taken to ensure proper String length control, or not all the previous String will be overwritten. This is only done when the SD50 is in Run Mode (0).

Static Strings

If swapping between static preconstructed strings use a COP or CPS instructions. Ensure that the length is set to 29 so the entire string is used. This will move the blank/empty string sections also. This will overwrite any previous string values.

Example
Here we have two strings. One is long “Banner Long String Test” while the other is shorter “Short String”. By using a COP command with a length of 29 it is possible to swap between both strings correctly. This is the proper way to get just the string that is currently being sent. If a Move instructions was used the String2 would give “Short StringString Test” if String1 was previously moved. Using the COP as shown below will prevent that from happening.

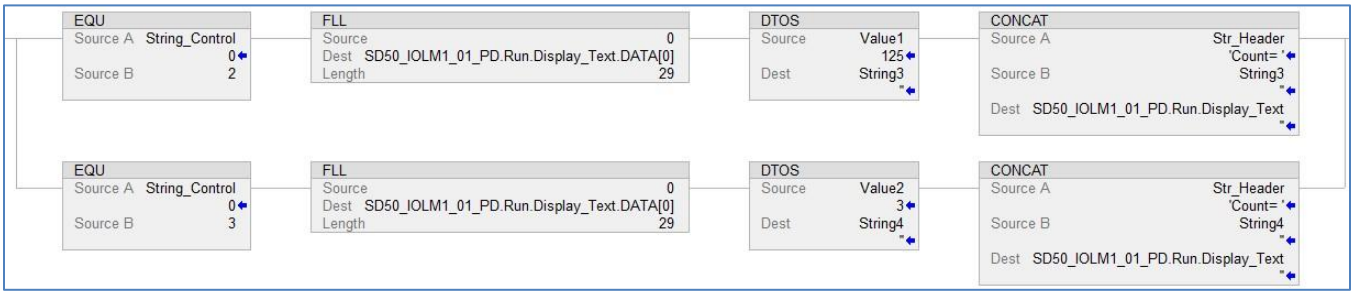




Dynamic Strings

If creating dynamic strings, it is necessary to ensure that the String tag being used is reset before each use. If this is not done it is possible to send extra string information. Use the FLL (File Fill) command with a Length of 29 to the string tag that will store the string that will be dynamically created. Next use the necessary commands to create the string.

Example
Here a FLL is used to blank the string. Next a value is converted from a DINT to a String and then this is added to the String “Count= “ to create the finalized string.



Appendix A SD50 Process Data

The SD50 has 32 bytes of Process Data Out, mapped into 5 different modes, as shown below.

This Process Data is mapped to a specific group of EtherNet/IP registers. The 256-bits of Process Data encode many separate pieces of information.

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

First is Run mode (mode 0). This controls the basic on/off/flash/animation state for the SD50. It also allows for the settings the Display Text.

ProcessDataOut "Process Data Out Run Mode" id=V_Pd_OutRunMode

bit length: 256
data type: 256-bit Record

subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	0	4-bit UInteger	0 = Off, 1 = Steady, 2 = Flash, 3 = Two Color Flash, 4 = Half/Half Steady, 5 = Half/Half Flash, 6 = Intensity Sweep, 7 = Two Color Sweep					Animation	The Animation type
2	4	4-bit UInteger	0 = Green, 1 = Red, 2 = Orange, 3 = Amber, 4 = Yellow, 5 = Lime Green, 6 = Spring Green, 7 = Cyan, 8 = Sky Blue, 9 = Blue, 10 = Violet, 11 = Magenta, 12 = Rose, 13 = White, 14 = Custom 1, 15 = Custom 2					Color 1	The main color of the Animation. Custom Colors are defined in Parameter data
3	8	3-bit UInteger	0 = High, 1 = Medium, 2 = Low, 3 = Off, 4 = Custom					Color 1 Intensity	The Intensity of Color 1, Custom Intensity defined in Parameter Data
4	11	2-bit UInteger	0 = Slow, 1 = Standard, 2 = Fast, 3 = Custom					Speed	The speed of the Animation
5	13	3-bit UInteger	0 = Normal, 1 = Strobe, 2 = Three Pulse, 3 = SOS, 4 = Random					Pulse Pattern	The pattern of Animation
6	16	4-bit UInteger	0 = Green, 1 = Red, 2 = Orange, 3 = Amber, 4 = Yellow, 5 = Lime Green, 6 = Spring Green, 7 = Cyan, 8 = Sky Blue, 9 = Blue, 10 = Violet, 11 = Magenta, 12 = Rose, 13 = White, 14 = Custom 1, 15 = Custom 2					Color 2	The secondary color of the Animation. Only used if Animation has two colors. Custom Colors are defined in Parameter data
7	20	3-bit UInteger	0 = High, 1 = Medium, 2 = Low, 3 = Off, 4 = Custom					Color 2 Intensity	The Intensity of Color 2, Custom Intensity defined in Parameter Data
8	24	29-octet String UTF-8						Display Text	Run Mode Display Text

Here is the information for Message mode (mode 1). This setup allows for similar light control as Run mode. Instead of manually setting a character string a number is set into Message 1 and 2. The messages are prepopulated in the IO-Link parameter data.

ProcessDataOut "Process Data Out Message Mode" id=V_Pd_OutMessageMode

bit length: 256
data type: 256-bit Record

subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	224	4-bit UInteger	0 = Off, 1 = Steady, 2 = Flash, 3 = Two Color Flash, 4 = Half/Half Steady, 5 = Half/Half Flash, 6 = Intensity Sweep, 7 = Two Color Sweep					Animation	The Animation type
2	228	4-bit UInteger	0 = Green, 1 = Red, 2 = Orange, 3 = Amber, 4 = Yellow, 5 = Lime Green, 6 = Spring Green, 7 = Cyan, 8 = Sky Blue, 9 = Blue, 10 = Violet, 11 = Magenta, 12 = Rose, 13 = White, 14 = Custom 1, 15 = Custom 2					Color 1	The main color of the Animation. Custom Colors are defined in Parameter data
3	232	3-bit UInteger	0 = High, 1 = Medium, 2 = Low, 3 = Off, 4 = Custom					Color 1 Intensity	The Intensity of Color 1, Custom Intensity defined in Parameter Data
4	235	2-bit UInteger	0 = Slow, 1 = Standard, 2 = Fast, 3 = Custom					Speed	The speed of the Animation
5	237	3-bit UInteger	0 = Normal, 1 = Strobe, 2 = Three Pulse, 3 = SOS, 4 = Random					Pulse Pattern	The pattern of Animation
6	240	4-bit UInteger	0 = Green, 1 = Red, 2 = Orange, 3 = Amber, 4 = Yellow, 5 = Lime Green, 6 = Spring Green, 7 = Cyan, 8 = Sky Blue, 9 = Blue, 10 = Violet, 11 = Magenta, 12 = Rose, 13 = White, 14 = Custom 1, 15 = Custom 2					Color 2	The secondary color of the Animation. Only used if Animation has two colors. Custom Colors are defined in Parameter data
7	244	3-bit UInteger	0 = High, 1 = Medium, 2 = Low, 3 = Off, 4 = Custom					Color 2 Intensity	The Intensity of Color 2, Custom Intensity defined in Parameter Data
8	248	4-bit UInteger	1..13 = Message Selection (1-13)					Message Selection 1	Message Mode Message Selection 1 (1-13)
9	252	4-bit UInteger	1..13 = Message Selection (1-13)					Message Selection 2	Message Mode Message Selection 2 (1-13)

Here is Measure mode (mode 2).

ProcessDataOut "Process Data Out Measure Mode" id=V_Pd_OutMeasureMode

bit length: 256
data type: 256-bit Record

subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	240	16-bit UInteger						Measure Mode Value	Value describing the level of the device, range determined in Measure Mode Parameter Data

Here is the information for Timer mode (mode 3).

ProcessDataOut "Process Data Out Timer Mode" id=V_Pd_OutTimerMode

bit length: 256
data type: 256-bit Record

subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	254	Boolean						Run Timer	Run the timer in timer mode
2	255	Boolean						Reset Timer	Reset the timer in timer mode

Here is the information for Counter mode (mode 4).

ProcessDataOut "Process Data Out Counter Mode" id=V_Pd_OutCounterMode

bit length: 256
data type: 256-bit Record

subindex	bit offset	data type	allowed values	default value	acc. restr.	mod. other var.	excl. from DS	name	description
1	253	Boolean						Increment Count	Increment the counter value when in counter mode
2	254	Boolean						Decrement Count	Decrement the counter value when in counter mode
3	255	Boolean						Reset Count	Reset the count in counter mode

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"

Port	Allen-Bradley*		Lumberg		Balluff		Turck		ifm		Banner	
	PDI	PDO	PDI	PDO	PDI	PDO	PDI	PDO	PDI	PDO	PDI	PDO
1	I.Ch0Data[0]	O.Ch0Data[0]	14	4	8	6	6	4	190	46	184	182
2	I.Ch1Data[0]	O.Ch1Data[0]	68	36	56	38	38	36	222	78	218	216
3	I.Ch2Data[0]	O.Ch2Data[0]	122	68	104	70	70	68	254	110	252	250
4	I.Ch3Data[0]	O.Ch3Data[0]	176	100	152	102	102	100	286	142	286	284
5	I.Ch4Data[0]	O.Ch4Data[0]	230	132	200	134	134	132	318	174	320	318
6	I.Ch5Data[0]	O.Ch5Data[0]	284	164	248	166	166	164	350	206	354	352
7	I.Ch6Data[0]	O.Ch6Data[0]	338	196	296	198	198	196	382	238	388	386
8	I.Ch7Data[0]	O.Ch7Data[0]	392	228	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
Lumberg	0
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 and DXMR110-8K) 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