WLF LC25C Process Data AOI Guide, v4
January 11th, 2024

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 WLF/LC25C device via an IO-Link Master connected to an Allen-Bradley PLC. The AOI covers parsing and display of the Process Data Out. The AOI has two User Defined Tag data types.

Components

Banner_WLf_LC25C_PD_v4_AOI.L5X

UDT Packaged with the AOI

Banner_WLF_LC25C_PDIO_v4
Banner_WLS_CC_v4

Other AOIs Available Separately

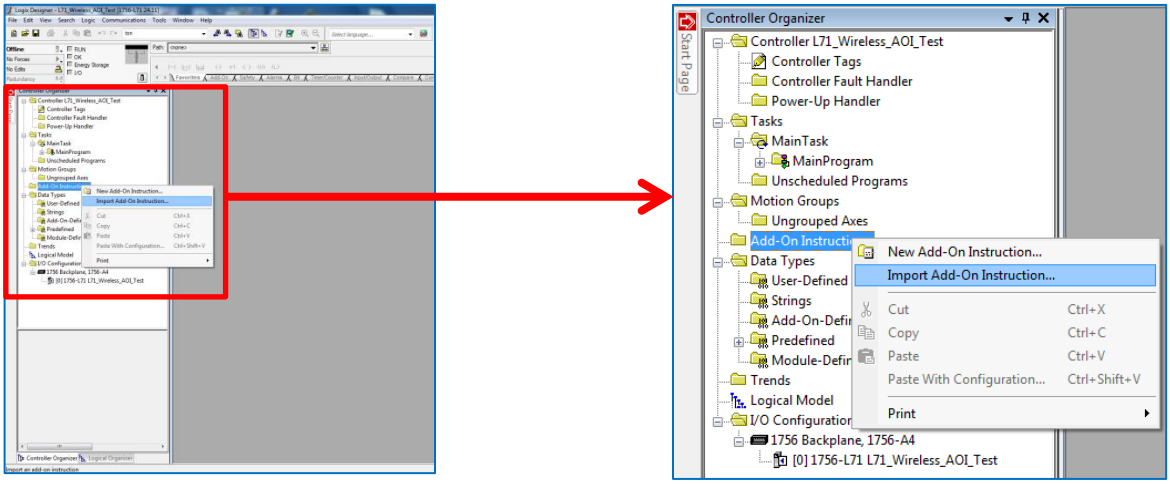
Banner has AOI files for controlling other Banner IO-Link devices and for a variety of IO-Link Masters. Banner also has AOI files for easily handling Banner device Parameter Data.

Contents

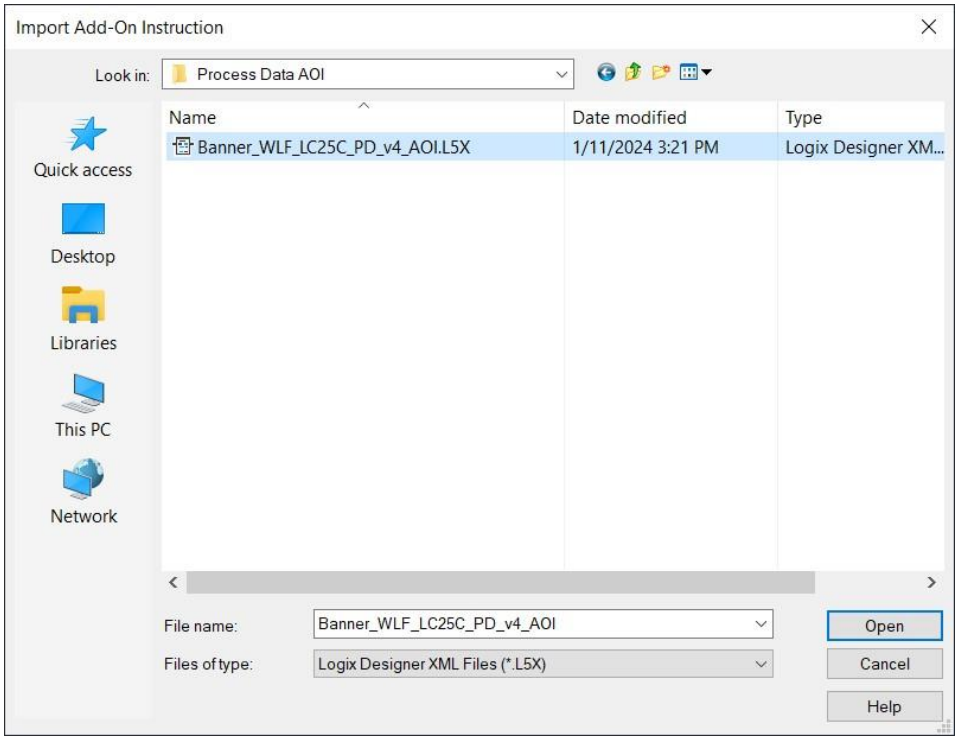
1. Installation Process

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

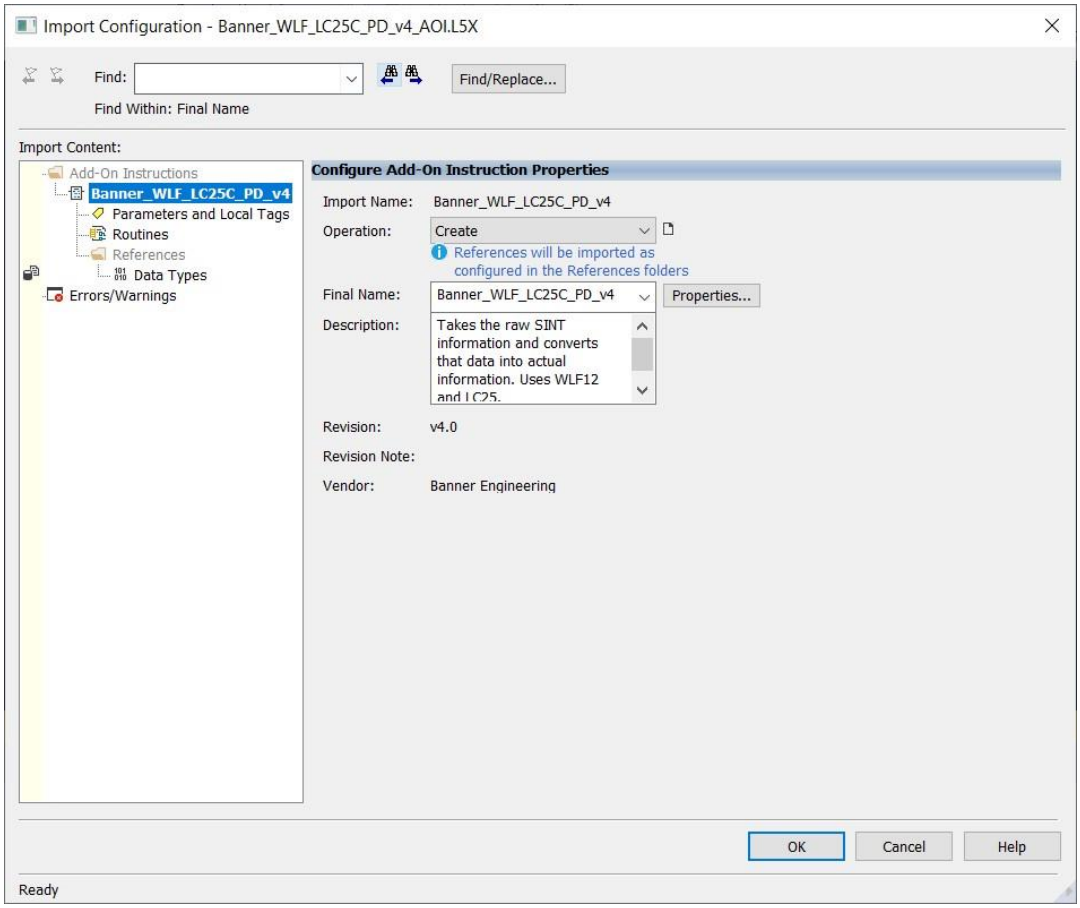
1. Open a project.
2. In the Controller Organizer window, right-click on the Add-On Instruction folder. Select the Import Add-On Instruction option.



3. Navigate to the correct file location and select the AOI to be installed. In this example the "Banner_WLF_LC25C_PD_v4.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.

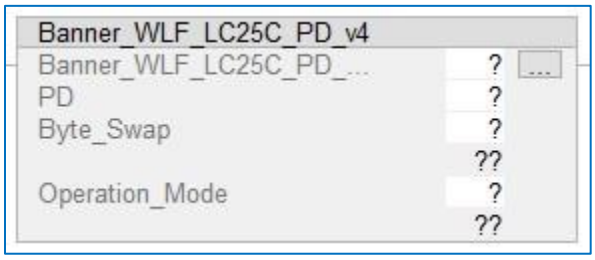
2. Configuring the IO-Link Master

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.

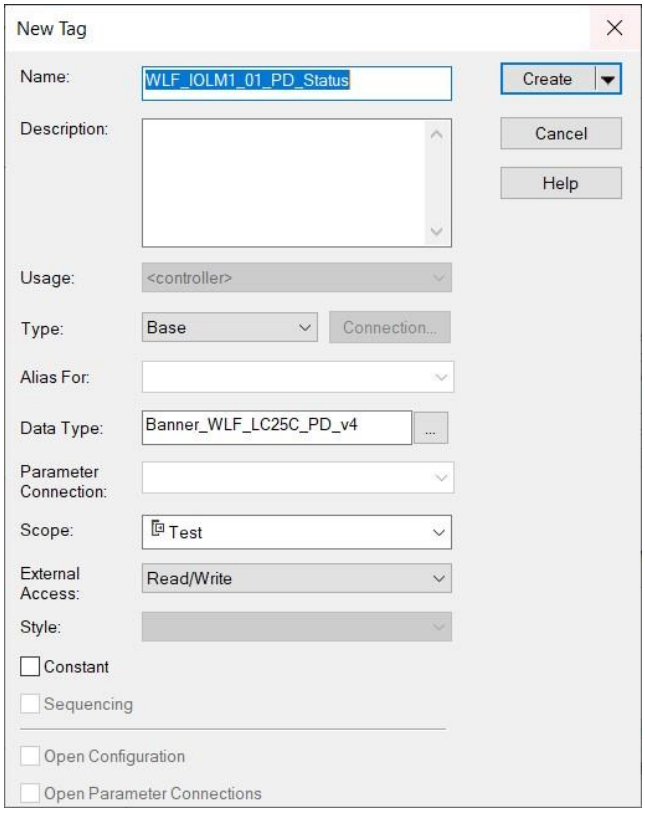
3. Configuring the AOI

1. Add the "Banner_WLF_LC25C_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.



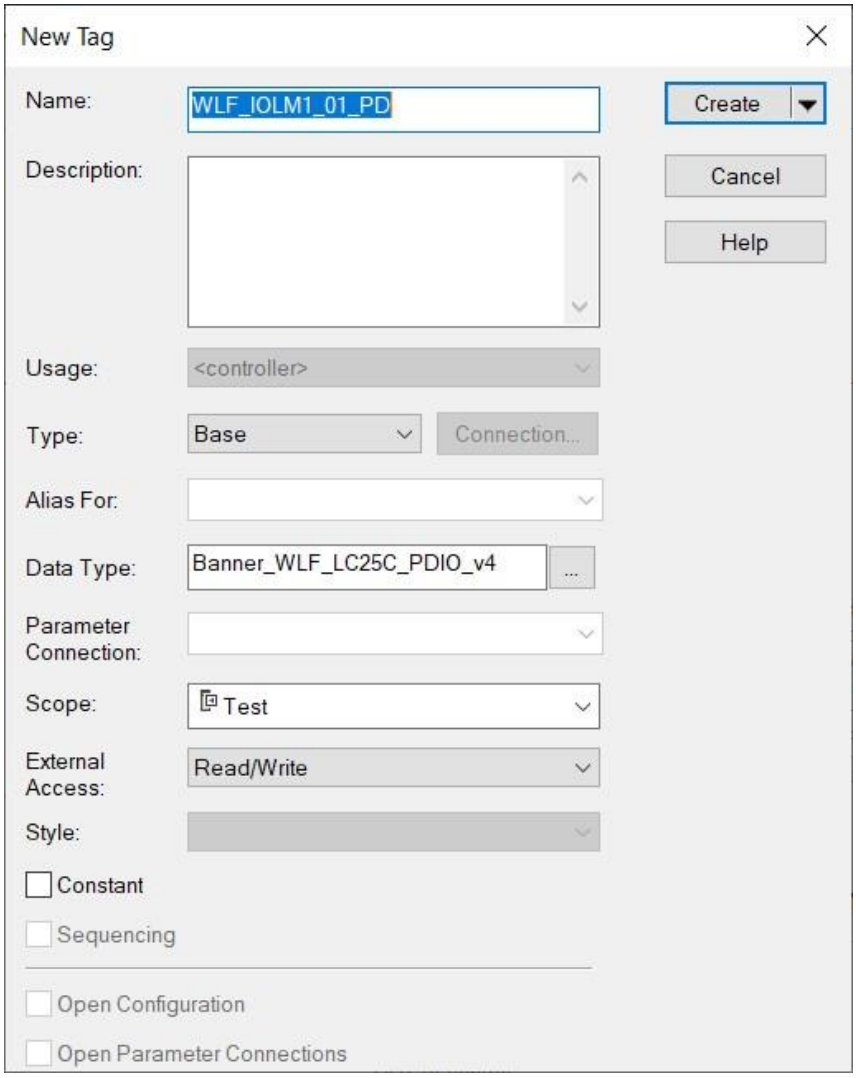
2. In the AOI, right-click on the question mark on the line labeled "Banner_WLF_LC25C_PD_v4". Click New Tag. Name the new tag. This example uses the name "WLF_ IOLM1_01_PD_Status". The example naming convention accounts for this being an WLF device connected to IO-Link Master #1, port #1, in our program. 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_WLF_LC25C_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 “WLF_IOLM1_01_PD”. Notice that the Data Type is “Banner_WLF_LC25C_PDIO_v4”. Click Create.

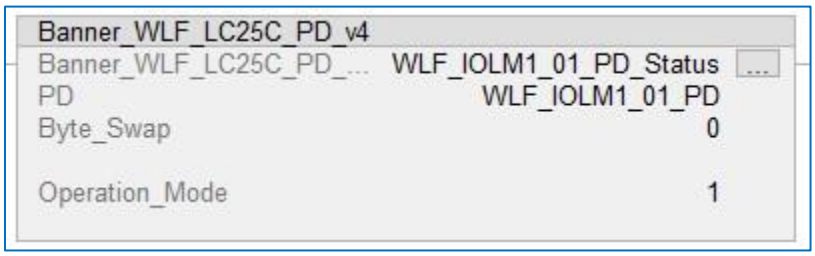
   This array will handle the displaying of the parsed Process Data Out for the WLF device.

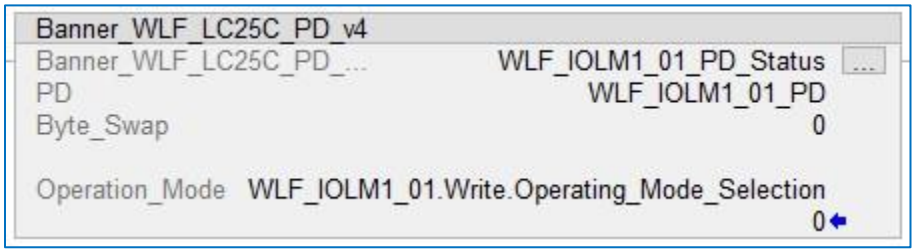


4. The next line in the AOI is a setting to account for byte swapping. In the case of the WLF, the Process Data Out is 32 bytes long. IO-Link Masters may read each pair of bytes in either order, so this AOI must be ready to perform a byte swap. Enter a “0” or a “1” to toggle this setting. See Appendix B for more information.

5. The line labeled “Operation Mode” allows the AOI to know which of seven possible Process Data Out definitions is currently in use. The choices for this setting are “0” (Segment mode), “1” (Run mode), “2” (Level mode), “3” (Dim and Blend mode), “4” (Gauge mode), “5” (LED mode), and “6” (Demo mode). The default setting is “1”. This AOI needs to know which mode selection has been made in the device.

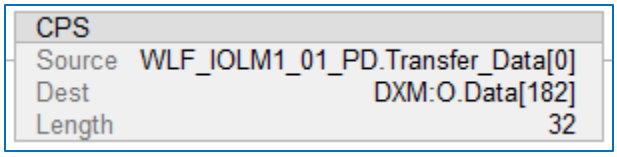
   There are two ways to achieve this goal. We can simply type in the correct number as a constant, or we can link this WLF Process Data AOI to the WLF Parameter Data AOI. See Appendix A for more information about WLS Pro Process Data.



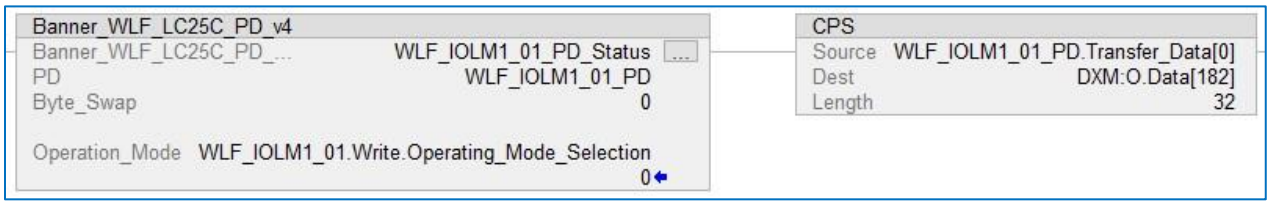


   NOTE: if you type in the incorrect number (i.e. it does not match the device’s current configuration) you will get incorrectly displayed Process Data Out information.

6. The final step required before we download and run the WLF 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 Outside. In this example, that is byte 182. The size to be copied is 32 bytes.

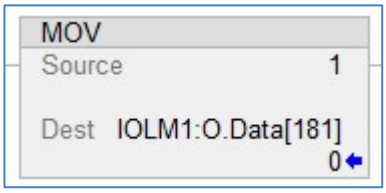


Here is what the entire rung looks like when completed.



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_WLF_PD_v4" AOI is now ready for use.

4. Using the AOI

The “Banner_WLF_PD_v4” Add-On Instruction has created a group of tags representing the WLF 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 “WLF_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 WLF device is in operating mode “1” (Run mode), use the tags found under the “Run_Mode” array, as seen below. If the operating mode is “2” (Level mode), use the corresponding tags in the “Level_Mode” array instead, and so on.

◢ WLF_IOLM1_01_PD	{...}
▷ WLF_IOLM1_01_PD.Segment_Mode	{...}
◢ WLF_IOLM1_01_PD.Run_Mode	{...}
▷ WLF_IOLM1_01_PD.Run_Mode.Animation	0
▷ WLF_IOLM1_01_PD.Run_Mode.Color_1	0
▷ WLF_IOLM1_01_PD.Run_Mode.Color_1_Intensity	0
▷ WLF_IOLM1_01_PD.Run_Mode.Speed	0
▷ WLF_IOLM1_01_PD.Run_Mode.Pulse_Pattern	0
▷ WLF_IOLM1_01_PD.Run_Mode.Color_2	0
▷ WLF_IOLM1_01_PD.Run_Mode.Color_2_Intensity	0
▷ WLF_IOLM1_01_PD.Run_Mode.ScrollBounce	0
▷ WLF_IOLM1_01_PD.Run_Mode.Percent_Width_C1	0
▷ WLF_IOLM1_01_PD.Run_Mode.Direction	0
▷ WLF_IOLM1_01_PD.Level_Mode	0
▷ WLF_IOLM1_01_PD.Dim_Blend_Mode	0
▷ WLF_IOLM1_01_PD.Gauge_Mode_Value	0
▷ WLF_IOLM1_01_PD.LED_Mode	{...}

Controlling the unit in Run Mode requires setting the Animation first. If a value of 1 is entered into Animation, then the light will turn ON with the Color 1 settings. By default, the unit will turn green, since a value of 0 in Color 1 represents green. See the IODD for all the options for the tags listed above.

Appendix A WLF Pro Process Data

The WLF/LC25C have 32 bytes of Process Data Out, mapped into 7 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 the Segment mode (mode 0). This controls the basic on/off/flash/animation state of each segment.

ProcessData id=V_PdT_Segment (condition V_Mode == 0)

ProcessDataOut "Process Data Out Segment Mode" id=V_Pd_OutSegment

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	16-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 1	The state of the segment. Related parameters defined in Segment Parameter Data
2	16	16-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 2	The state of the segment. Related parameters defined in Segment Parameter Data
3	32	16-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 3	The state of the segment. Related parameters defined in Segment Parameter Data
4	48	16-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 4	The state of the segment. Related parameters defined in Segment Parameter Data
5	64	16-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 5	The state of the segment. Related parameters defined in Segment Parameter Data
6	80	16-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 6	The state of the segment. Related parameters defined in Segment Parameter Data
7	96	16-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 7	The state of the segment. Related parameters defined in Segment Parameter Data
8	112	16-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 8	The state of the segment. Related parameters defined in Segment Parameter Data
9	128	16-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 9	The state of the segment. Related parameters defined in Segment Parameter Data
10	144	16-bit UInteger	0 = Off, 1 = On, 2 = Flash, 3 = Animation					Segment 10	The state of the segment. Related parameters defined in Segment Parameter Data

Here is the information for Run mode (mode 1).

ProcessData id=V_PdT_RunMode (condition V_Mode == 1)

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	8-bit UInteger	0 = Off, 1 = Steady, 2 = Flash, 3 = Two Color Flash, 4 = Two Color Shift, 5 = Ends Steady, 6 = Ends Flash, 7 = Scroll, 8 = Center Scroll, 9 = Bounce, 10 = Center Bounce, 11 = Intensity Sweep, 12 = Two Color Sweep, 13 = Spectrum, 14 = Level Steady, 15 = Level Flash					Animation	The Animation type
2	16	8-bit UInteger	0 = Green, 1 = Red, 2 = Orange, 3 = Amber, 4 = Yellow, 5 = Lime Green, 6 = Spring Green, 7 = Cyan, 8 = SkyBlue, 9 = Blue, 10 = Violet, 11 = Magenta, 12 = Rose, 13 = Daylight White (5000K), 14 = Custom 1, 15 = Custom 2, 16 = Incandescent White (2700K), 17 = Warm White (3000K), 18 = Fluorescent White (4100K), 19 = Neutral White (5700K), 20 = Cool White (6500K)					Color 1	The main color of the Animation. Custom Colors are defined in Parameter data
3	32	8-bit UInteger	0 = High, 1 = Low, 2 = Medium, 3 = Off, 4 = Custom					Color 1 Intensity	The Intensity of Color 1, Custom Intensity defined in Parameter Data
4	48	8-bit UInteger	0 = Medium, 1 = Fast, 2 = Slow, 3 = Custom Flash Rate					Speed	The speed of the Animation
5	64	8-bit UInteger	0 = Normal, 1 = Strobe, 2 = Three Pulse, 3 = SOS, 4 = Random					Pulse Pattern	The pattern of Animation
6	80	8-bit UInteger	0 = Green, 1 = Red, 2 = Orange, 3 = Amber, 4 = Yellow, 5 = Lime Green, 6 = Spring Green, 7 = Cyan, 8 = SkyBlue, 9 = Blue, 10 = Violet, 11 = Magenta, 12 = Rose, 13 = Daylight White (5000K), 14 = Custom 1, 15 = Custom 2, 16 = Incandescent White (2700K), 17 = Warm White (3000K), 18 = Fluorescent White (4100K), 19 = Neutral White (5700K), 20 = Cool White (6500K)					Color 2	The secondary color of the Animation. Only used if Animation has two colors. Custom Colors are defined in Parameter data
7	96	8-bit UInteger	0 = High, 1 = Low, 2 = Medium, 3 = Off, 4 = Custom					Color 2 Intensity	The Intensity of Color 2, Custom Intensity defined in Parameter Data
8	112	8-bit UInteger	0 = Solid, 1 = Tail, 2 = Ripple					Scroll/Bounce Style	The style of scrolling Segment
9	128	8-bit UInteger	1..100 = Percent Width of Color 1					Percent Width of Color 1	The size of scrolling Segment
10	144	8-bit UInteger	0 = Up, 1 = Down					Direction	The direction of Animation

Here is Level mode (mode 2).

ProcessData id=V_PdT_LevelMode (condition V_Mode == 2)

ProcessDataOut "Process Data Out Level Mode" id=V_Pd_OutLevelMode

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	16-bit UInteger						Level Mode Value	Value describing the level of the device, range determined in Level Mode Parameter Data

Here is Dim and Blend mode (mode 3).

ProcessData id=V_PdT_DimAndBlendMode (condition V_Mode == 3)

ProcessDataOut "Process Data Out Dim and Blend Mode" id=TI_PD_Out_DimAndBlendMode

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	16-bit UInteger						Dim and Blend Mode Value	Value describing the amount to blend between colors selected in Dim and Blend Mode Parameter Data

Here is Gauge mode (mode 4).

ProcessData id=V_PdT_GaugeMode (condition V_Mode == 4)

ProcessDataOut "Process Data Out Gauge Mode" id=V_PD_OutGaugeMode

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	16-bit UInteger						Gauge Mode Value	Value describing the position of the main animation

Here is a small portion of the data included in LED mode (mode 5). This mode allows the user full control over every section of three LEDs in the WLF.

ProcessData id=V_PdT_LedMode (condition V_Mode == 5)

ProcessDataOut "Process Data Out LED Mode" id=V_Pd_OutLedMode

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 = Green, 2 = Red, 3 = Orange, 4 = Amber, 5 = Yellow, 6 = Lime Green, 7 = Spring Green, 8 = Cyan, 9 = SkyBlue, 10 = Blue, 11 = Violet, 12 = Magenta, 13 = Rose, 14 = Daylight White (5000K), 15 = Custom 1					LED 1 Color	LED Mode Color control
2	4	4-bit UInteger	0 = Off, 1 = Green, 2 = Red, 3 = Orange, 4 = Amber, 5 = Yellow, 6 = Lime Green, 7 = Spring Green, 8 = Cyan, 9 = SkyBlue, 10 = Blue, 11 = Violet, 12 = Magenta, 13 = Rose, 14 = Daylight White (5000K), 15 = Custom 1					LED 2 Color	LED Mode Color control
3	8	4-bit UInteger	0 = Off, 1 = Green, 2 = Red, 3 = Orange, 4 = Amber, 5 = Yellow, 6 = Lime Green, 7 = Spring Green, 8 = Cyan, 9 = SkyBlue, 10 = Blue, 11 = Violet, 12 = Magenta, 13 = Rose, 14 = Daylight White (5000K), 15 = Custom 1					LED 3 Color	LED Mode Color control
4	12	4-bit UInteger	0 = Off, 1 = Green, 2 = Red, 3 = Orange, 4 = Amber, 5 = Yellow, 6 = Lime Green, 7 = Spring Green, 8 = Cyan, 9 = SkyBlue, 10 = Blue, 11 = Violet, 12 = Magenta, 13 = Rose, 14 = Daylight White (5000K), 15 = Custom 1					LED 4 Color	LED Mode Color control
5	16	4-bit UInteger	0 = Off, 1 = Green, 2 = Red, 3 = Orange, 4 = Amber, 5 = Yellow, 6 = Lime Green, 7 = Spring Green, 8 = Cyan, 9 = SkyBlue, 10 = Blue, 11 = Violet, 12 = Magenta, 13 = Rose, 14 = Daylight White (5000K), 15 = Custom 1					LED 5 Color	LED Mode Color control
6	20	4-bit UInteger	0 = Off, 1 = Green, 2 = Red, 3 = Orange, 4 = Amber, 5 = Yellow, 6 = Lime Green, 7 = Spring Green, 8 = Cyan, 9 = SkyBlue, 10 = Blue, 11 = Violet, 12 = Magenta, 13 = Rose, 14 = Daylight White (5000K), 15 = Custom 1					LED 6 Color	LED Mode Color control
7	24	4-bit UInteger	0 = Off, 1 = Green, 2 = Red, 3 = Orange, 4 = Amber, 5 = Yellow, 6 = Lime Green, 7 = Spring Green, 8 = Cyan, 9 = SkyBlue, 10 = Blue, 11 = Violet, 12 = Magenta, 13 = Rose, 14 = Daylight White (5000K), 15 = Custom 1					LED 7 Color	LED Mode Color control
8	28	4-bit UInteger	0 = Off, 1 = Green, 2 = Red, 3 = Orange, 4 = Amber, 5 = Yellow, 6 = Lime Green, 7 = Spring Green, 8 = Cyan, 9 = SkyBlue, 10 = Blue, 11 = Violet, 12 = Magenta, 13 = Rose, 14 = Daylight White (5000K), 15 = Custom 1					LED 8 Color	LED Mode Color control
9	32	4-bit UInteger	0 = Off, 1 = Green, 2 = Red, 3 = Orange, 4 = Amber, 5 = Yellow, 6 = Lime Green, 7 = Spring Green, 8 = Cyan, 9 = SkyBlue, 10 = Blue, 11 = Violet, 12 = Magenta, 13 = Rose, 14 = Daylight White (5000K), 15 = Custom 1					LED 9 Color	LED Mode Color control

Note: Only up to LED 9 shown in above image. Depending on length of the WLF there can be up to 40 LED Sections.

Demo mode (mode 6), has no defined Process Data Out.

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*		Comtrol		Balluff		Turck		ifm		Banner	
	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