



## **R95C-8UI Process Data AOI Guide, v4**

### **October 13<sup>th</sup>, 2023**

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 from a Banner R95C-8UI device via an IO-Link Master to an Allen-Bradley PLC.

#### **Components**

Banner\_R95C8UI\_PD\_v4\_AOI.L5X

#### **UDTs Packaged with the AOI**

Banner\_R95C\_8UI\_PDIO\_v4

Banner\_R95C\_8UI\_PDI1\_v4

Banner\_R95C\_8UI\_PDIO\_v4

Banner\_R95C\_8UI\_Sensor\_Profile\_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 ..... 1

2. Configuring the IO-Link Master ..... 3

3. Configuring the AOI ..... 4

4. Using the AOI ..... 8

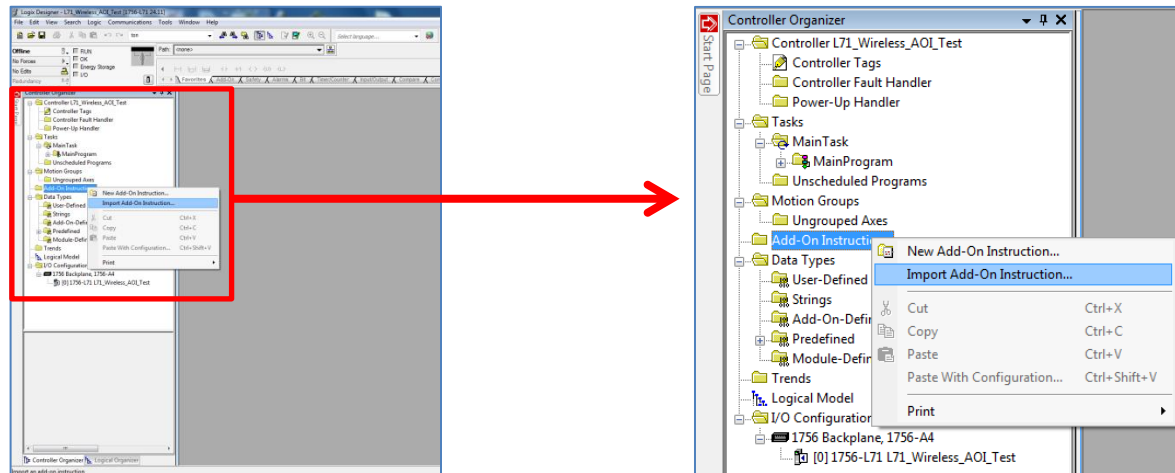
Appendix A R95C-8UI Process Data ..... 9

Appendix B IO-Link Master Cheat Sheet ..... 10

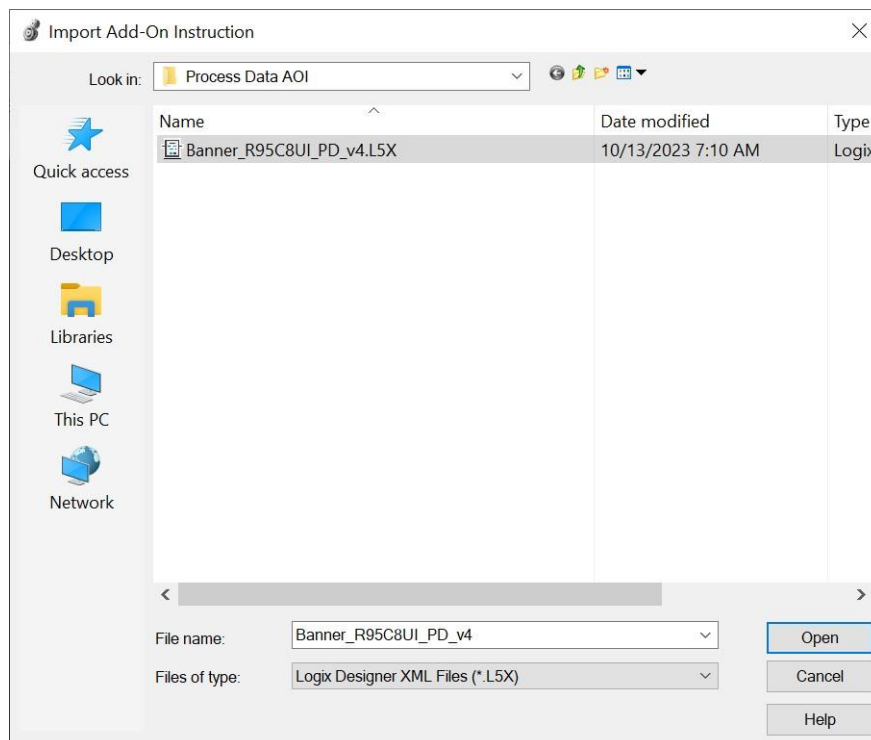
## 1. Installation Process

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

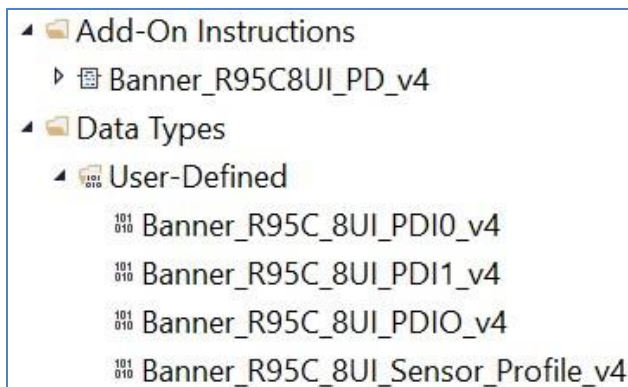
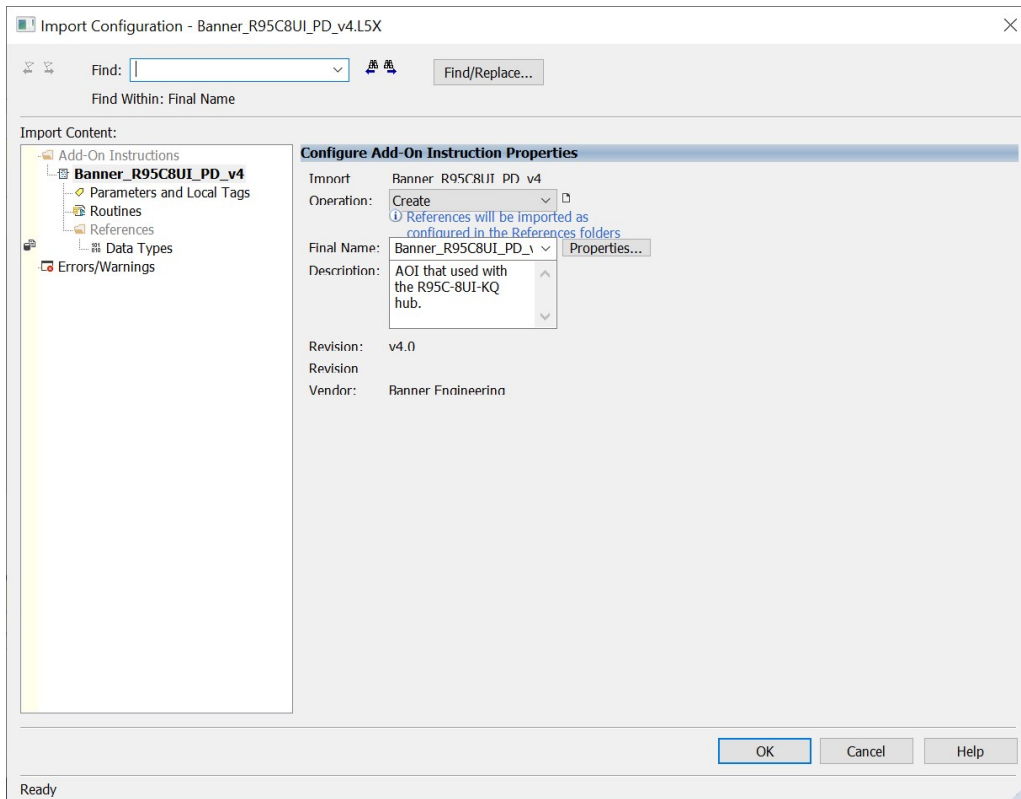
1. Open a project.
2. In the Controller Organizer window, right-click on the Add-On Instruction folder. Select the Import Add-On Instruction option.



3. Navigate to the correct file location and select the AOI to be installed. In this example the "Banner\_R95C8UI\_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.

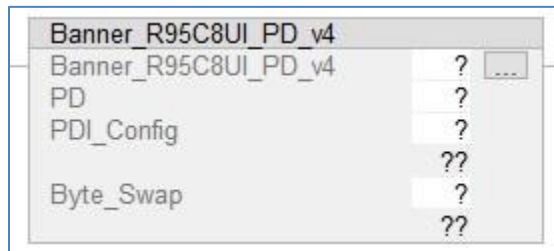
## **2. Configuring the IO-Link Master**

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 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.

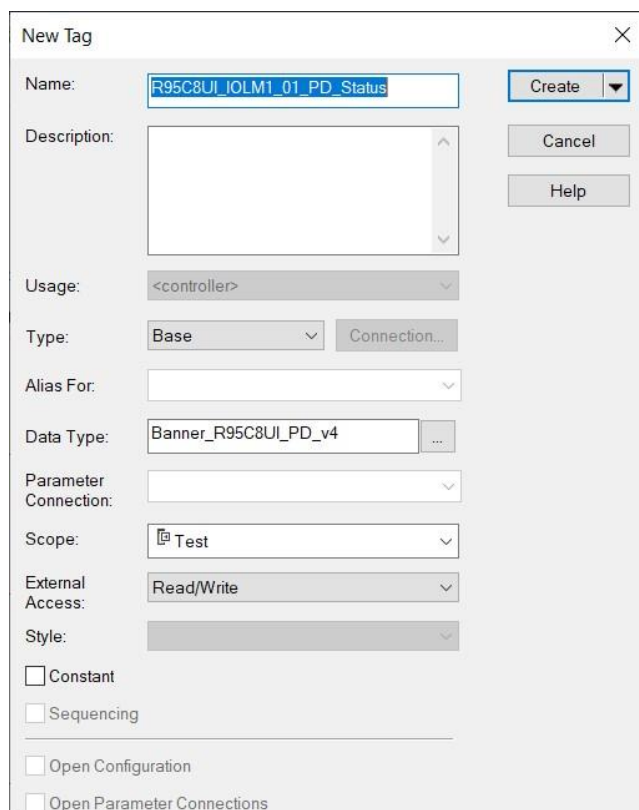
### 3. Configuring the AOI

1. Add the “Banner\_R95C8UI\_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\_R95C8UI\_PD\_v4”. Click New Tag. Name the new tag. This example uses the name “R95C8UI\_IOLM1\_01\_PD\_Status”. The example naming convention accounts for this being a R95C-8B22 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\_R95C8UI\_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.



New Tag

Name: R95C8UI\_IOLM1\_01\_PD\_Status

Description:

Usage: <controller>

Type: Base

Alias For:

Data Type: Banner\_R95C8UI\_PD\_v4

Parameter Connection:

Scope: Test

External Access: Read/Write

Style: Base

☐ Constant

☐ Sequencing

☐ Open Configuration

☐ Open Parameter Connections

Create

Cancel

Help

- Now we will right-click on the question mark on the line labeled “PD” in the AOI. Click on “New Tag”. Give the tag a name. This example uses the name “R95C8UI\_IOLM1\_01\_PD”. Notice that the Data Type is “Banner\_R95C8UI\_PD\_v4”. Click Create.

This array will handle the displaying of the parsed Process Data for the R95C-8UI.

New Tag

Name: R95C8UI\_IOLM1\_01\_PD

Description:

Usage: <controller>

Type: Base Connection...

Alias For:

Data Type: Banner\_R95C\_8UI\_PDIO\_v4

Parameter Connection:

Scope: Test

External Access: Read/Write

Style:

☐ Constant

☐ Sequencing

☐ Open Configuration

☐ Open Parameter Connections

Create Cancel Help

- Next the PDI\_Config value needs to be set. This controls which of the two possible process data options the AOI should use to parse the data. The PDI\_Config is set in the parameter data. The first option is to set the value manually. The second option is to use the tag from the parameter data AOI and link it to this location. This can only be done if the parameter data AOI is used in the PLC. Example of what both approaches look like are shown below.

Banner_R95C8UI_PD_v4	
Banner_R95C8UI_PD_v4	R95C8UI_IOLM1_01_PD_Status ...
PD	R95C8UI_IOLM1_01_PD
PDI_Config	0
Byte_Swap	0

Banner_R95C8UI_PD_v4	
Banner_R95C8UI_PD_v4	R95C8UI_IOLM1_01_PD_Status ...
PD	R95C8UI_IOLM1_01_PD
PDI_Config	R95C8UI_IOLM1_01.Write.VSC.PDI_Config 0 ←
Byte_Swap	0

- The next line in the AOI is a setting to account for byte swapping. In the case of the R95C-8UI, the Process Data In 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.

**NOTE:** If the IO-Link Master shown in this example uses "0" for the byte swapping parameter.



6. The final step required before we download and run the R95C-8UI Process Data AOI involves adding one File Synchronous Copy (CPS) instructions. This instruction allows the AOI to read from raw Process Data values found in the register tags of the IO-Link Master.

Add a CPS instruction before the AOI on the ladder rung that looks like the one seen below. Refer to Appendix B for which byte to start with in the “Source” area. In this case, the IO-Link Master in question has the raw Process Data In values for a device connected to port 1 starting at byte 184. For the “Destination”, we will enter the “DTI[0]” location, as seen below. Finally, the length will be 32 bytes, as that is the size of the R95C-8UI Process Data In.

CPS	
Source	IOLM1:I.Data[184]
Dest	R95C8UI_IOLM1_01_PD.DTI[0]
Length	32

Here is what the entire rung looks like when completed.

CPS	Banner_R95C8UI_PD_v4
Source	Banner_R95C8UI_PD_v4 R95C8UI_IOLM1_01_PD_Status
Dest	PD R95C8UI_IOLM1_01_PD
Length	PDI_Config R95C8UI_IOLM1_01.Write.VSC.PDI_Config
	Byte_Swap 0

The “Banner\_R95C8UI\_PD\_v4” AOI is now ready for use.

## 4. Using the AOI

The “Banner\_R95C8UI\_PD\_v4” Add-On Instruction has created a group of tags representing the Process Data In, broken out into its component parts.

Look in the Controller Tags to find the name you used in Step 4 above. This example used the name “R95CBUI\_IOLM1\_01\_PD”. The tag array, seen below, shows what data the UDT has. The “Set0” and “Set1” tags store all the information for all eight ports of the device. Expand the “Set0” tag. This tag is the default location that will have the data set into it. “Set1” is only used if the “PDI\_Config” parameter is adjusted via the Parameter Data AOI.

▲ R95C8UI_IOLM1_01_PD
▶ R95C8UI_IOLM1_01_PD.Set0
▶ R95C8UI_IOLM1_01_PD.Set1
▶ R95C8UI_IOLM1_01_PD.DTI

“Set0” is broken out into ports 0 through 8. Only ports 1 through 8 are used. The analog data for a port is stored in these locations. Analog values range from 4000 to 20000. The data is divided by 1000 to be the 4 to 20 mA value. Voltage inputs are similar but range from 0 to 10000. Only Ports with analog unit connected to them will have values in the range for the analog unit.

▲ R95C8UI_IOLM1_01_PD.Set0.Value	{...}
▶ R95C8UI_IOLM1_01_PD.Set0.Value[0]	0
▶ R95C8UI_IOLM1_01_PD.Set0.Value[1]	12000
▶ R95C8UI_IOLM1_01_PD.Set0.Value[2]	8100
▶ R95C8UI_IOLM1_01_PD.Set0.Value[3]	6700
▶ R95C8UI_IOLM1_01_PD.Set0.Value[4]	16785
▶ R95C8UI_IOLM1_01_PD.Set0.Value[5]	0
▶ R95C8UI_IOLM1_01_PD.Set0.Value[6]	0
▶ R95C8UI_IOLM1_01_PD.Set0.Value[7]	0
▶ R95C8UI_IOLM1_01_PD.Set0.Value[8]	0

## Appendix A R95C-8UI Process Data

The R95C-8UI has 32 bytes of Process Data In and 2 bytes of Process Data Out.

### ProcessDataIn "Process Data Input" id=PD\_ProcessDataIn

bit length: 256

data type: 256-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	32-bit Integer						Measurement Value 1	The channel 1 measurement device value
5	192	32-bit Integer						Measurement Value 2	The channel 2 measurement device value
9	160	32-bit Integer						Measurement Value 3	The channel 3 measurement device value
13	128	32-bit Integer						Measurement Value 4	The channel 4 measurement device value
17	96	32-bit Integer						Measurement Value 5	The channel 5 measurement device value
21	64	32-bit Integer						Measurement Value 6	The channel 6 measurement device value
25	32	32-bit Integer						Measurement Value 7	The channel 7 measurement device value
29	0	32-bit Integer						Measurement Value 8	The channel 8 measurement device value

## 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