



R95C MDR Process Data AOI Guide, v5
March 2nd, 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 from a Banner R95C MDR device via an IO-Link Master to an Allen-Bradley PLC.

Components

Banner_R95CMDR_PD_v5_AOI.L5X

UDTs Packaged with the AOI

Banner_R95CMDR_PDIO_v5

Banner_R95CMDR_PDI_v5

Banner_R95CMDR_PDO_v5

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.

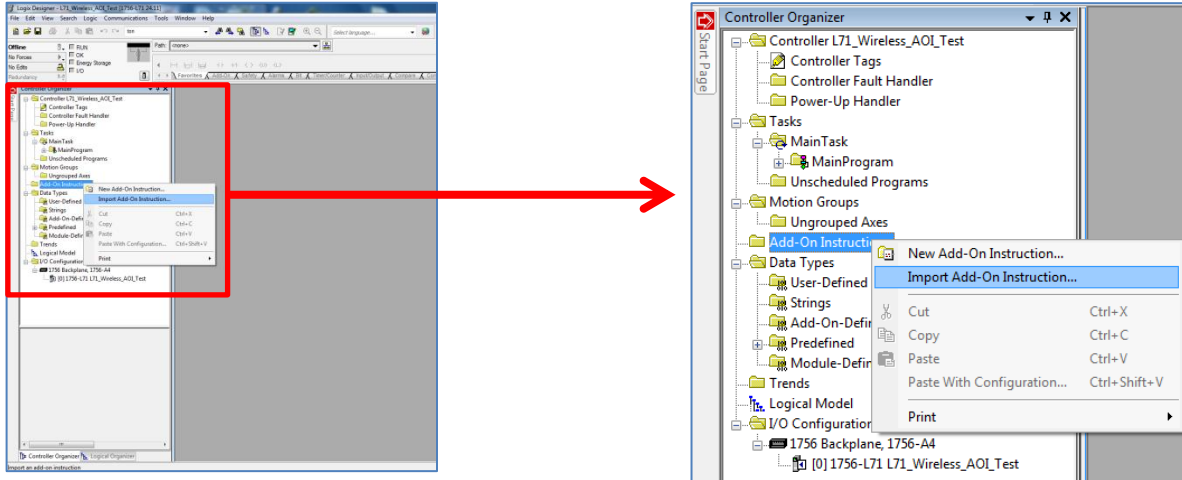
Contents

1. Installation Process	1
2. Configuring the IO-Link Master	2
3. Configuring the AOI.....	3
4. Using the AOI.....	5
Appendix A R90C-B21 R95C-8B21 Process Data	6
Appendix B IO-Link Master Cheat Sheet	8

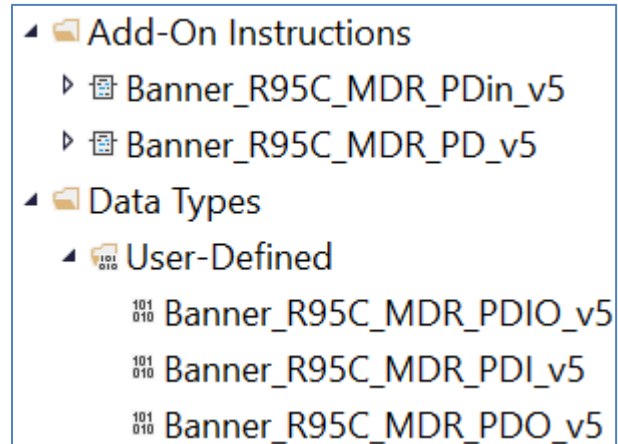
1. Installation Process

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_R95C_MDR_PD_v5_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 shown to the right.
6. AOI Installation into the Logix Designer software is complete.



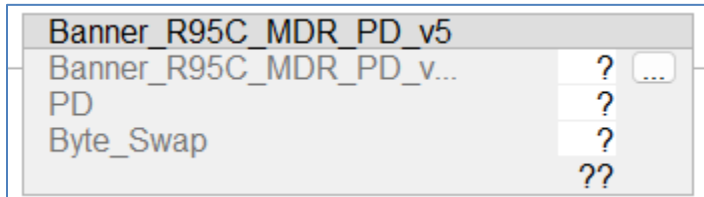
2. Configuring the IO-Link Master

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_R95C_MDR_PD_v5” 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_R95C_MDR_PD_v5”. Click New Tag. Name the new tag. This example uses the name “MDR_IOLM1_01_PD_Status”. The example naming convention accounts for this being a R95C MDR 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_R95C_MDR_PD_v5”. 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 “PD” in the AOI. Click on “New Tag”. Give the tag a name. This example uses the name “MDR_IOLM1_01_PD”. Notice that the Data Type is “Banner_R95C_MDR_PDIO_v5”. Click Create.

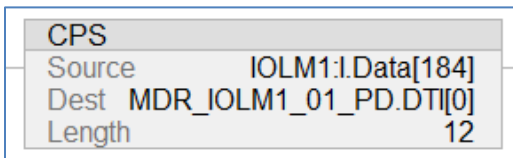
This array will handle the displaying of the parsed Process Data for the R95C MDR.

4. The next line in the AOI is a setting to account for byte swapping. In the case of the R95C MDR, the Process Data In is 4 bytes long and the Process Data Out is 4 bytes. 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.

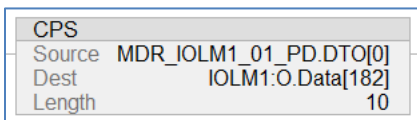
NOTE: If the IO-Link Master you are using requires byte swapping be set to “1”.

- The final step required before we download and run the R95C MDR Process Data AOI involves adding two File Synchronous Copy (CPS) instructions. These instructions allow the AOI to read from and write to the raw Process Data values found in the register tags of the IO-Link Master.

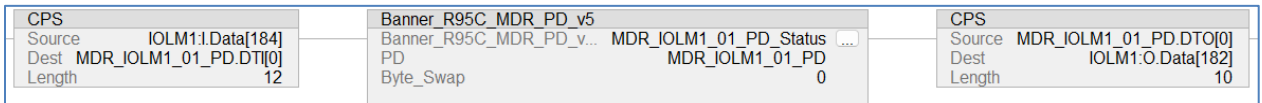
Add a CPS instruction before the AOI on the ladder rung that looks like the one seen below. Refer to Appendix B for which byte to start with in the “Source” area. In this case, the IO-Link Master in question has the raw Process Data In values for a device connected to port 2 starting at byte 184. For the “Destination”, we will enter the “DTI[0]” location, as seen below. Finally, the length will be 12 bytes to allow for byte swapping if necessary.



The second CPS instruction sends the Process Data Output to the correct location for the IO-Link Master. It requires a size of 10 bytes. The Source will be the DTO[0] and is linked to the output location as shown in Appendix B.

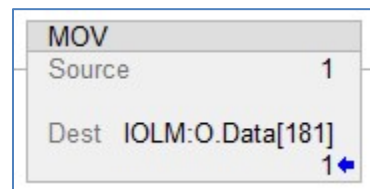


Here is what the entire rung looks like when completed.



If a Banner IO-Link Master is being used, set up a Move block. Send a 1 to the Activate Outputs array value (see table for 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_R95C_MDR_PD_v5” AOI is now ready for use.

4. Using the AOI

The “Banner_R95C_MDR_PD_v5” Add-On Instruction has created a group of tags representing the Process Data In and Out, 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 “MDR_IOLM1_01_PD”. The tag array, seen below, shows what data the UDT has. The “PDI” tag stores all of the input information for the device. Expand the “PDI” tag.

<ul style="list-style-type: none"> ▾ MDR_IOLM1_01_PD ▸ MDR_IOLM1_01_PD.PDI ▸ MDR_IOLM1_01_PD.PDO

“PDI” shows all of the input data for the MDR. Port Channel 1, Port Channel 2, and Analog Value are all arrays. The Channels are an array of 32 booleans. Only 1 through 4 are used for those arrays as there are only 4 channels on the block.

▸ MDR_IOLM1_01_PD.PDI.MDR_Short_Circuit	{...}	0=Inactive, 1=Active. Only 1 through 4 used.
▸ MDR_IOLM1_01_PD.PDI.MDR_Overcurrent	{...}	0=Current Below Limit, 1=Current Above Limit. Only 1 through 4 used.
MDR_IOLM1_01_PD.PDI.MDR_OverVoltage	0	0=Voltage within Range, 1=Voltage above Range
MDR_IOLM1_01_PD.PDI.MDR_UnderVoltage	0	0=Voltage within Range, 1=Voltage below Range
MDR_IOLM1_01_PD.PDI.Block_Temperature	0	0=Temp below limit, 1=Temp above limit
MDR_IOLM1_01_PD.PDI.Aux_Power	0	0=No Aux Power, 1=Aux Power Present
▸ MDR_IOLM1_01_PD.PDI.Port_Channel_1	{...}	0=Off, 1=On. Only 1 through 4 used.
▸ MDR_IOLM1_01_PD.PDI.Port_Channel_2	{...}	0=Off, 1=On. Only 1 through 4 used.
▸ MDR_IOLM1_01_PD.PDI.Analog_Value	{...}	

The “PDO” shows all of the output data for the MDR. All of the data are in arrays. Expand the arrays to see the actual data. Note that Motor_Current_Fault_Reset, Channel_1_Control, and Channel_2_Control only use 1 through 4 in the arrays.

▸ MDR_IOLM1_01_PD.PDO.Motor_Current_Fault_Reset	{...}	0=No Action, 1=Reset. Only 1 through 4 used.
▸ MDR_IOLM1_01_PD.PDO.Port_Channel_1_Control	{...}	0=Off, 1=On. Only 1 through 4 used.
▸ MDR_IOLM1_01_PD.PDO.Port_Channel_2_Control	{...}	0=Off, 1=On. Only 1 through 4 used.
▸ MDR_IOLM1_01_PD.PDO.Analog_Output	{...}	

Appendix A R95C MDR Process Data

The R95C MDR has 11 bytes of Process Data In and 10 bytes of Process Data Out. Only part of the data is shown below. See the IODD file for the full mapping.

ProcessDataIn "Process Data Input" id=PI_PDIn									
bit length: 88									
data type: 88-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	83	Boolean	false = Inactive, true = Active					MDR 4 Short Circuit	MDR 4 Short Circuit
2	82	Boolean	false = Current below limit, true = Current above limit					MDR 4 Overcurrent	MDR 4 Overcurrent
3	81	Boolean	false = Inactive, true = Active					MDR 3 Short Circuit	MDR 3 Short Circuit
4	80	Boolean	false = Current below limit, true = Current above limit					MDR 3 Overcurrent	MDR 3 Overcurrent
5	79	Boolean	false = Inactive, true = Active					MDR 2 Short Circuit	MDR 2 Short Circuit
6	78	Boolean	false = Current below limit, true = Current above limit					MDR 2 Overcurrent	MDR 2 Overcurrent
7	77	Boolean	false = Inactive, true = Active					MDR 1 Short Circuit	MDR 1 Short Circuit
8	76	Boolean	false = Current below limit, true = Current above limit					MDR 1 Overcurrent	MDR 1 Overcurrent
9	75	Boolean	false = Voltage within Range, true = Voltage above Range					MDR OverVoltage	MDR OverVoltage
10	74	Boolean	false = Voltage within Range, true = Voltage below Range					MDR UnderVoltage	MDR UnderVoltage
11	73	Boolean	false = Temp below limit, true = Temp above limit					Block Temperature	Block Temperature
12	72	Boolean	false = No Aux Power, true = Aux Power Present					Aux Power	Aux Power

ProcessDataOut "Process Data Out" id=PO_PDout

bit length: 80

data type: 80-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	75	Boolean	false = No action, true = Reset					Motor 4 Current Fault Reset	Motor 4 Current Fault Reset
2	74	Boolean	false = No action, true = Reset					Motor 3 Current Fault Reset	Motor 3 Current Fault Reset
3	73	Boolean	false = No action, true = Reset					Motor 2 Current Fault Reset	Motor 2 Current Fault Reset
4	72	Boolean	false = No action, true = Reset					Motor 1 Current Fault Reset	Motor 1 Current Fault Reset
5	71	Boolean	false = Inactive, true = Active					Port 4 Pin2-Channel 2 Control	Port 4 Pin2-Channel 2 Control
6	70	Boolean	false = Inactive, true = Active					Port 4 Pin4-Channel 1 Control	Port 4 Pin4-Channel 1 Control
7	69	Boolean	false = Inactive, true = Active					Port 3 Pin2-Channel 2 Control	Port 3 Pin2-Channel 2 Control
8	68	Boolean	false = Inactive, true = Active					Port 3 Pin4-Channel 1 Control	Port 3 Pin4-Channel 1 Control
9	67	Boolean	false = Inactive, true = Active					Port 2 Pin2-Channel 2 Control	Port 2 Pin2-Channel 2 Control
10	66	Boolean	false = Inactive, true = Active					Port 2 Pin4-Channel 1 Control	Port 2 Pin4-Channel 1 Control
11	65	Boolean	false = Inactive, true = Active					Port 1 Pin2-Channel 2 Control	Port 1 Pin2-Channel 2 Control
12	64	Boolean	false = Inactive, true = Active					Port 1 Pin4-Channel 1 Control	Port 1 Pin4-Channel 1 Control
13	48	16-bit Integer						Port 4 Pin 5 - Analog Output Control (mV)	Port 4 Pin 5 - Analog Output Control (mV)
14	32	16-bit Integer						Port 3 Pin 5 - Analog Output Control (mV)	Port 3 Pin 5 - Analog Output Control (mV)
15	16	16-bit Integer						Port 2 Pin 5 - Analog Output Control (mV)	Port 2 Pin 5 - Analog Output Control (mV)
16	0	16-bit Integer						Port 1 Pin 5 - Analog Output Control (mV)	Port 1 Pin 5 - Analog Output Control (mV)

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 it 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