



P2M2HBVL OR P2HL IO-LINK
ADD-ON INSTRUCTION FOR
ROCKWELL PLC WITH BALLUFF
BNI006A OR BNI008M
ETHERNET/IP IO-LINK MASTER
QUICK START GUIDE

PREFACE

This Quick Start Guide (QSG) is designed to help integrate Parker Hannifin's P2M or P2H IO-Link valve manifold into an Allen-Bradley (AB) PLC environment utilizing the Balluff BNI006A or BNI008M EIP IO-Link Master module. The QSG assumes that you are already using the Balluff AOI for its master module and that it is communicating to the AB PLC via an Ethernet-IP network. You can find this AOI and instructions on how to implement it here:

http://usa.balluff.com/AOI/distribution/AOI/AOI_BNI006A_50_31_040.zip

The QSG is agnostic to IO Link Device Classification, such that it shall function the same whether you are controlling an A-Class or B-Class P2M / P2H Module. The guide will walk the user through obtaining the necessary files, importing/configuring the AOI, and initiating parameter reads and writes from/to the P2M / P2H IO-Link device.

The "P2M2HBVL_P2HL_AB_BNI006A_PRM_Rx" AOI facilitates the call-up of the acyclic service data.

The "P2M2HBVL_P2HL_AB_BNI006A_PD_Rx" AOI facilitates communication and handling of process data between PLC and the IO-Link slave device.

You can download resources such as the IODD configuration file, this QSG, a sample RSLogix5000 file and the full user manual here:

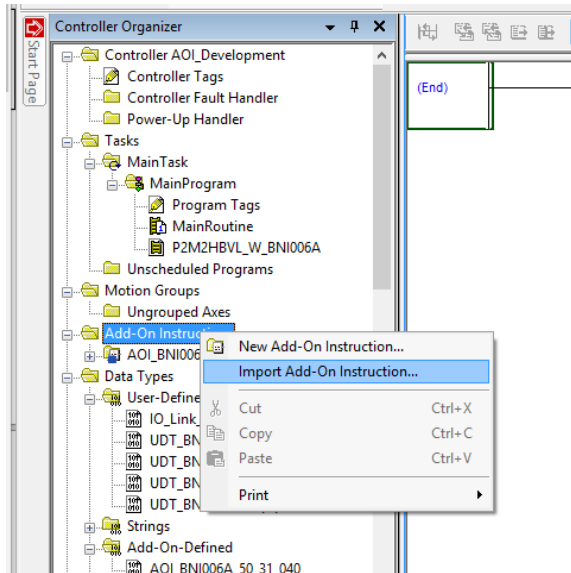
www.parker.com/PDN/io-link

Process Data Add-On Instruction

The “P2M2HBVL_P2HL_AB_BNI006A_PD_Rx” AOI simplifies the usage of Parker P2M and P2H IO-Link devices with Allen-Bradley CompactLogix, ControlLogix and GuardLogix PLCs when connected, via Ethernet/IP, to a Balluff BNI006A or BNI008M IO-Link Master. Data is mapped to user-friendly control and diagnostic tags on the PLC side.

IMPORTING THE INSTRUCTION

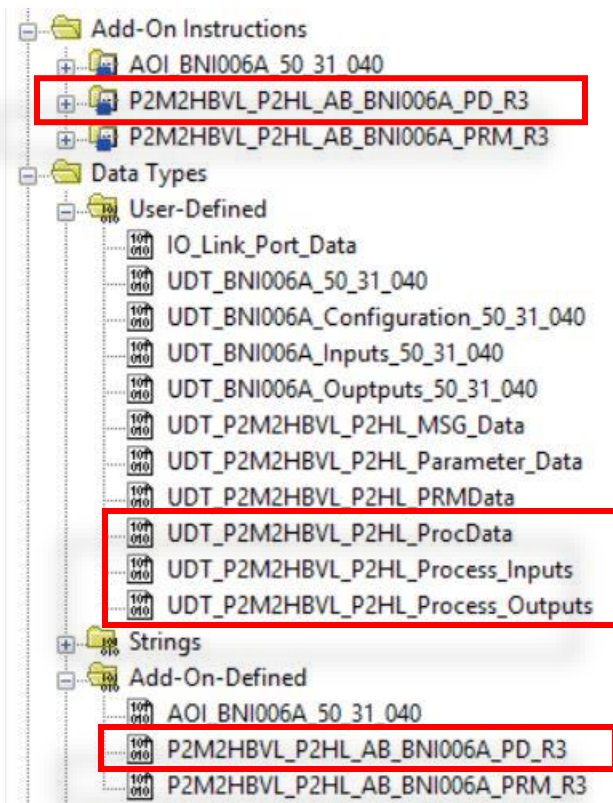
1. Right click Add-On Instruction in Controller Organizer and select “Import Add-On Instruction...”



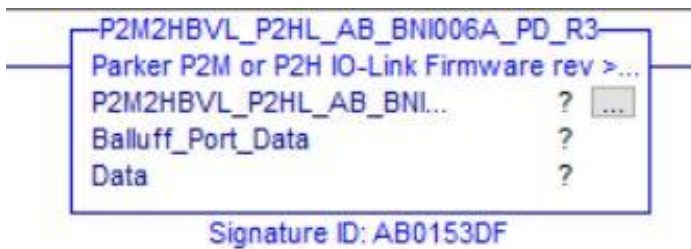
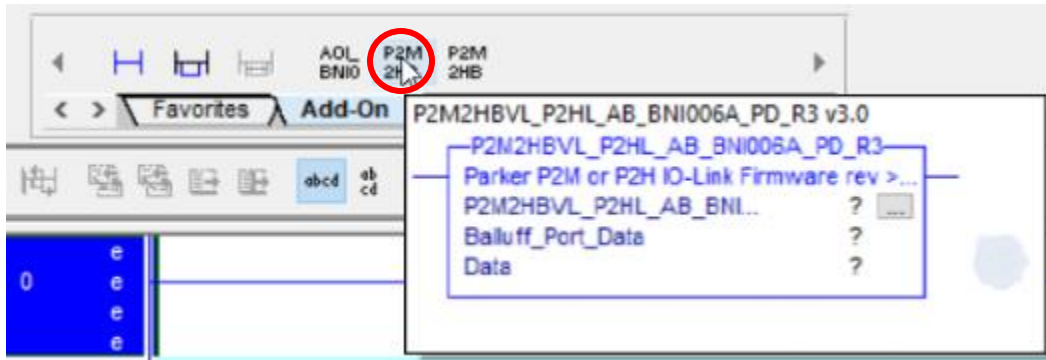
2. Select the “P2M2HBVL_P2HL_AB_BNI006A_PD_Rx” where _Rx is the revision of AOI.

P2M2HBVL_P2HL_AB_BNI006A_PD_R3.L5X 2/8/2019 3:07 PM Logix Designer X... 57 KB

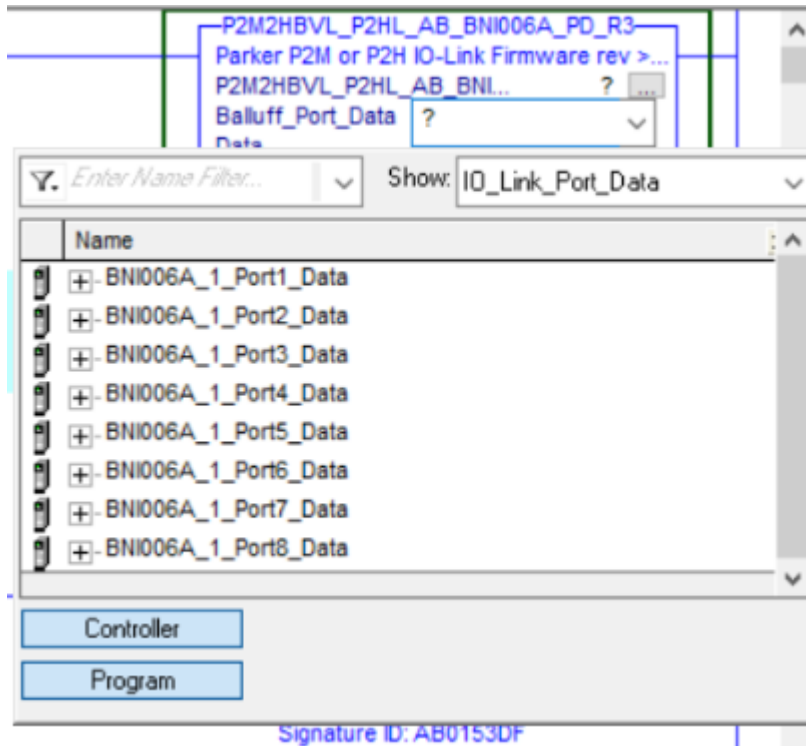
Choose OK on Import Configuration Window and you should then see the new AOI instance along with User-Defined and Add-On Defined data types created in the controller organizer.



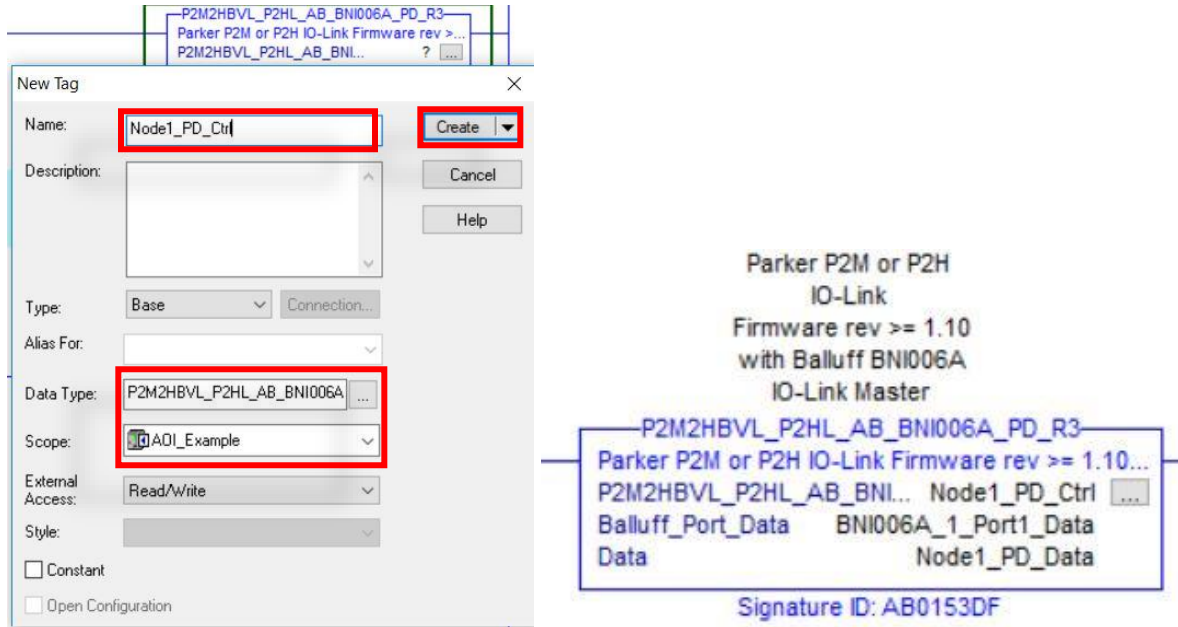
3. Add instance of instruction to an empty rung of ladder by clicking on the AB_P2M under the Add-On tab in the top toolbar. The instruction will drop onto the selected rung.



4. Point the Balluff_Port_Data field to the port that the P2M / P2H is connected to on the BNI006A/BNI008M. If you choose the incorrect port the AOI will NOT function and undesired results are likely. IO-Link is sensitive to the port which it has been assigned to communicate on.



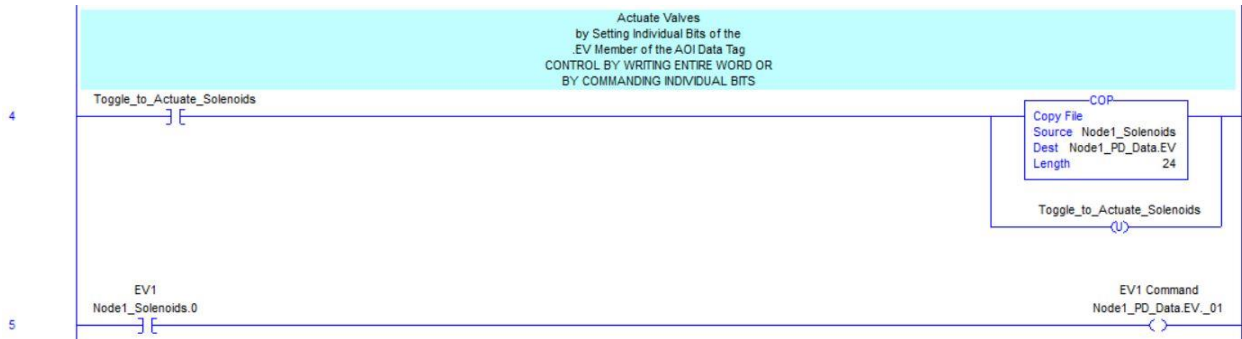
- Assign an instance name for the AOI and create other tags necessary for operation. Right click on the question marks and select “New Tag”. Note that the name must be unique for each tag and each instance of the P2M / P2H AOI. The scope and data type fields will auto-populate with the correct values, so these should not need to be changed. All fields are required. See Appendix for structure breakdown of the “Data” variable.



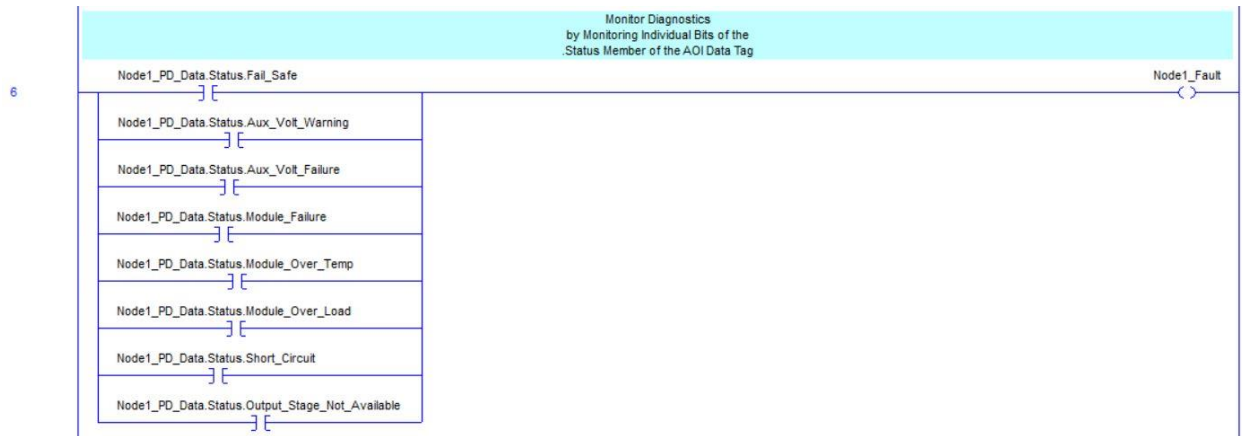
USING THE INSTRUCTION

It is important to note the difference between cyclic and acyclic data. Process Data (cyclic) is updated without a request; whereas Parameter Data (acyclic) requires the program to toggle a bit to pull (or push) data. Cyclic data includes input status and valve output control. This means that Node1_PD_Data.Status.xxx and Node1_PD_Data.EV.##_ are live tags (containing real data) that exist simply because the AOI instruction was used. See appendix for all data points available. See ladder logic examples below:

a. Toggling Solenoid Valves (Cyclic)



b. Monitoring Status Bits (Cyclic)

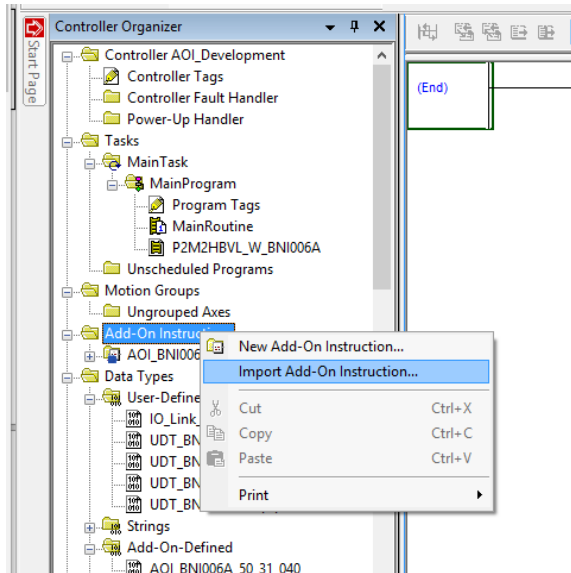


Parameter Data Add-On Instruction

The “P2M2HBVL_P2HL_AB_BNI006A_PRM_Rx” AOI simplifies the usage of Parker P2M and P2H IO-Link devices with Allen-Bradley CompactLogix, ControlLogix and GuardLogix PLCs when connected, via Ethernet/IP, to a Balluff BNI006A or BNI008M IO-Link Master. The AOI facilitates the reading and writing of parameter data between the PLC and the Parker P2M or P2H IO-Link device.

IMPORTING THE INSTRUCTION

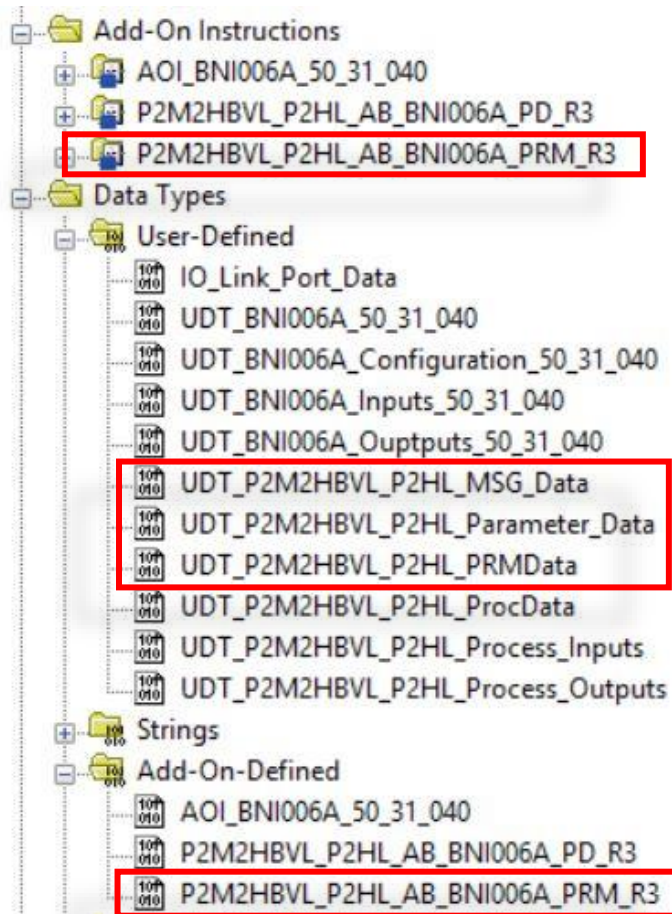
1. Right click Add-On Instruction in Controller Organizer and select “Import Add-On Instruction...”



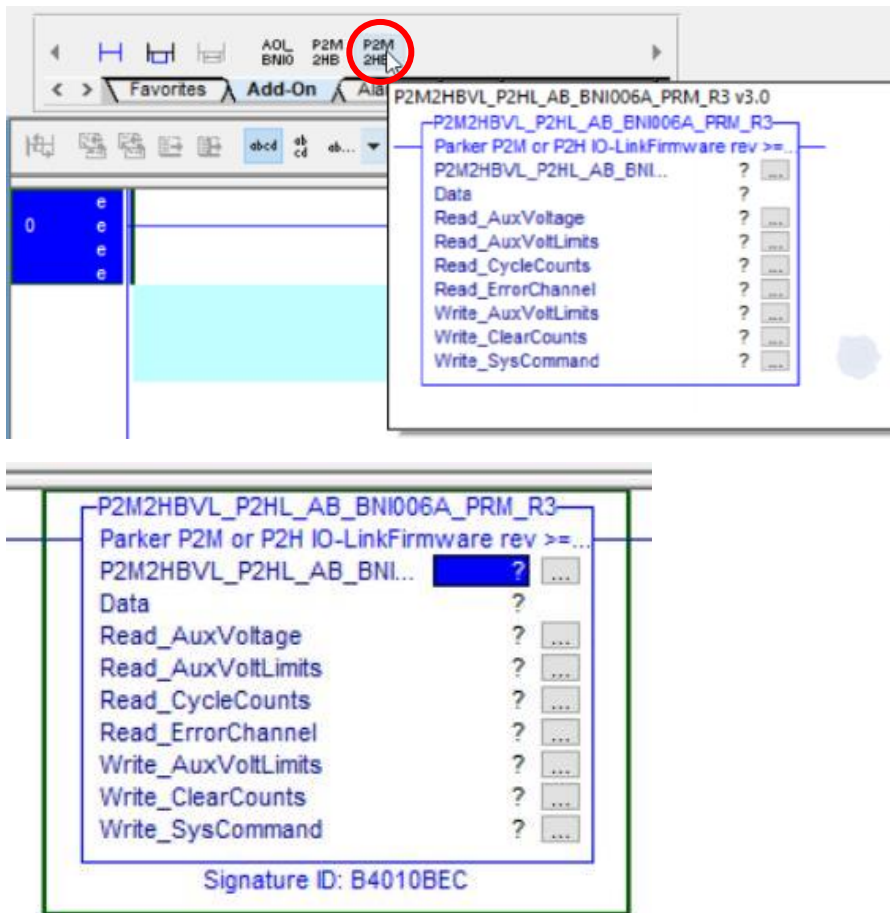
2. Select the “P2M2HBVL_P2HL_AB_BNI006A_PRM_Rx” where _Rx is the revision of AOI.

P2M2HBVL_P2HL_AB_BNI006A_PRM_R3.L5X 2/8/2019 3:07 PM Logix Designer X... 55 KB

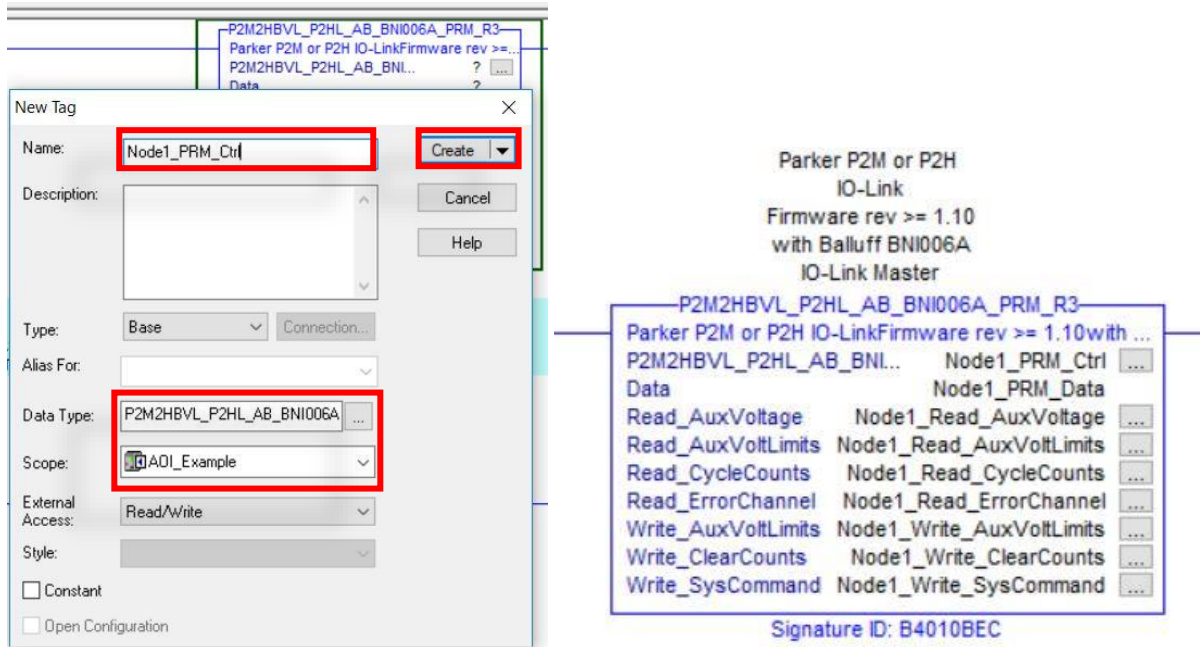
Choose OK on Import Configuration Window and you should then see the new AOI instance along with User-Defined and Add-On Defined data types created in the controller organizer.



3. Add instance of instruction to an empty rung of ladder by clicking on the AB_P2M under the Add-On tab in the top toolbar. The instruction will drop onto the selected rung.

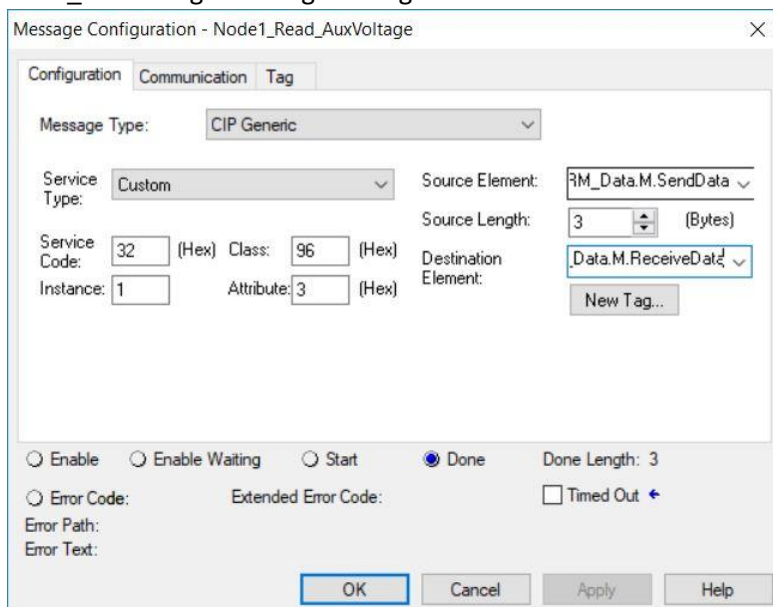


- Assign an instance name for the AOI and create other tags necessary for operation. To do this, right click on the question marks and select “New Tag”. Note that the name must be unique for each tag and each instance of the P2M / P2H AOI. The scope and data type fields will auto-populate with the correct values, so these should not need to be changed. Click the ellipsis button next to each message tag to provide configuration information. Configuration of all tags is required. See Appendix for structure breakdown of the “Data” variable.



Note: for the following configuration windows, the Instance field represents the port (1-8) on the Balluff IO-Link Master to which the P2M / P2H is connected. Attribute field shall be 3 for read operations and 2 for write operations. All other fields can be entered directly as shown below.

Read_AuxVoltage Message Configuration:



Read_AuxVoltLimits Message Configuration:

Message Configuration - Node1_Read_AuxVoltLimits

Configuration Communication Tag

Message Type: CIP Generic

Service Type: Custom Source Element: RM_Data.M.SendData

Service Code: 32 (Hex) Class: 96 (Hex) Source Length: 3 (Bytes)

Instance: 1 Attribute: 3 (Hex) Destination Element: Data.M.ReceiveData

New Tag...

☐ Enable ☐ Enable Waiting ☐ Start ☒ Done Done Length: 5

☐ Error Code: Extended Error Code: ☐ Timed Out

Error Path:
Error Text:

OK Cancel Apply Help

Read_CycleCounts Message Configuration:

Message Configuration - Node1_Read_CycleCounts

Configuration Communication Tag

Message Type: CIP Generic

Service Type: Custom Source Element: RM_Data.M.SendData

Service Code: 32 (Hex) Class: 96 (Hex) Source Length: 3 (Bytes)

Instance: 1 Attribute: 3 (Hex) Destination Element: Data.M.ReceiveData

New Tag...

☐ Enable ☐ Enable Waiting ☐ Start ☒ Done Done Length: 97

☐ Error Code: Extended Error Code: ☐ Timed Out

Error Path:
Error Text:

OK Cancel Apply Help

Read_ErrorChannel Message Configuration:

Message Configuration - Node1_Read_ErrorChannel

Configuration Communication Tag

Message Type: CIP Generic

Service Type: Custom

Source Element: RM_Data.M.SendData

Source Length: 3 (Bytes)

Service Code: 32 (Hex) Class: 96 (Hex)

Instance: 1 Attribute: 3 (Hex)

Destination Element: Data.M.ReceiveData

New Tag...

☐ Enable ☐ Enable Waiting ☐ Start ☒ Done Done Length: 5

☐ Error Code: Extended Error Code: ☐ Timed Out

Error Path:

Error Text:

OK Cancel Apply Help

Write_AuxVoltLimits Message Configuration:

Message Configuration - Node1_Write_AuxVoltLimits

Configuration Communication Tag

Message Type: CIP Generic

Service Type: Custom

Source Element: _Data.M.SendData[0]

Source Length: 7 (Bytes)

Service Code: 32 (Hex) Class: 96 (Hex)

Instance: 1 Attribute: 2 (Hex)

Destination Element: _Data.M.SendData[7]

New Tag...

☐ Enable ☐ Enable Waiting ☐ Start ☒ Done Done Length: 1

☐ Error Code: Extended Error Code: ☐ Timed Out

Error Path:

Error Text:

OK Cancel Apply Help

Write_ClearCounts Message Configuration:

Message Configuration - Node1_Write_ClearCounts

Configuration Communication Tag

Message Type: CIP Generic

Service Type: Custom

Service Code: 32 (Hex) Class: 96 (Hex) Instance: 1 Attribute: 2 (Hex)

Source Element: I_Data.M.SendData[0]

Source Length: 7 (Bytes)

Destination Element: I_Data.M.SendData[7]

New Tag...

☐ Enable ☐ Enable Waiting ☐ Start ☒ Done Done Length: 1

☐ Error Code: Extended Error Code: ☐ Timed Out

Error Path:
Error Text:

OK Cancel Apply Help

Write_SysCommand Message Configuration:

Message Configuration - Node1_Write_SysCommand

Configuration Communication Tag

Message Type: CIP Generic

Service Type: Custom

Service Code: 32 (Hex) Class: 96 (Hex) Instance: 1 Attribute: 2 (Hex)

Source Element: I_Data.M.SendData[0]

Source Length: 4 (Bytes)

Destination Element: I_Data.M.SendData[4]

New Tag...

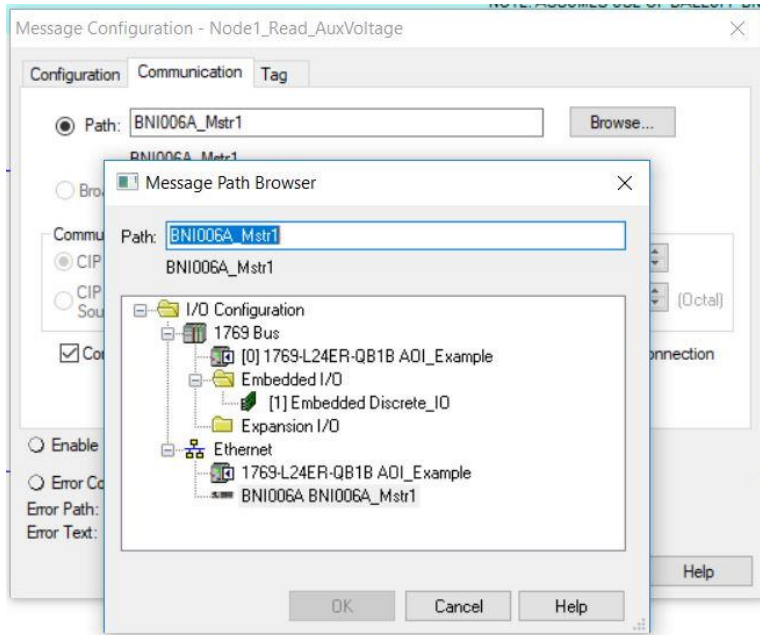
☐ Enable ☐ Enable Waiting ☐ Start ☒ Done Done Length: 1

☐ Error Code: Extended Error Code: ☐ Timed Out

Error Path:
Error Text:

OK Cancel Apply Help

For each message tag, the device path must also be selected within the “Communication” tab. Select the “Browse” button and select the appropriate Balluff IO-Link Master.



USING THE INSTRUCTION

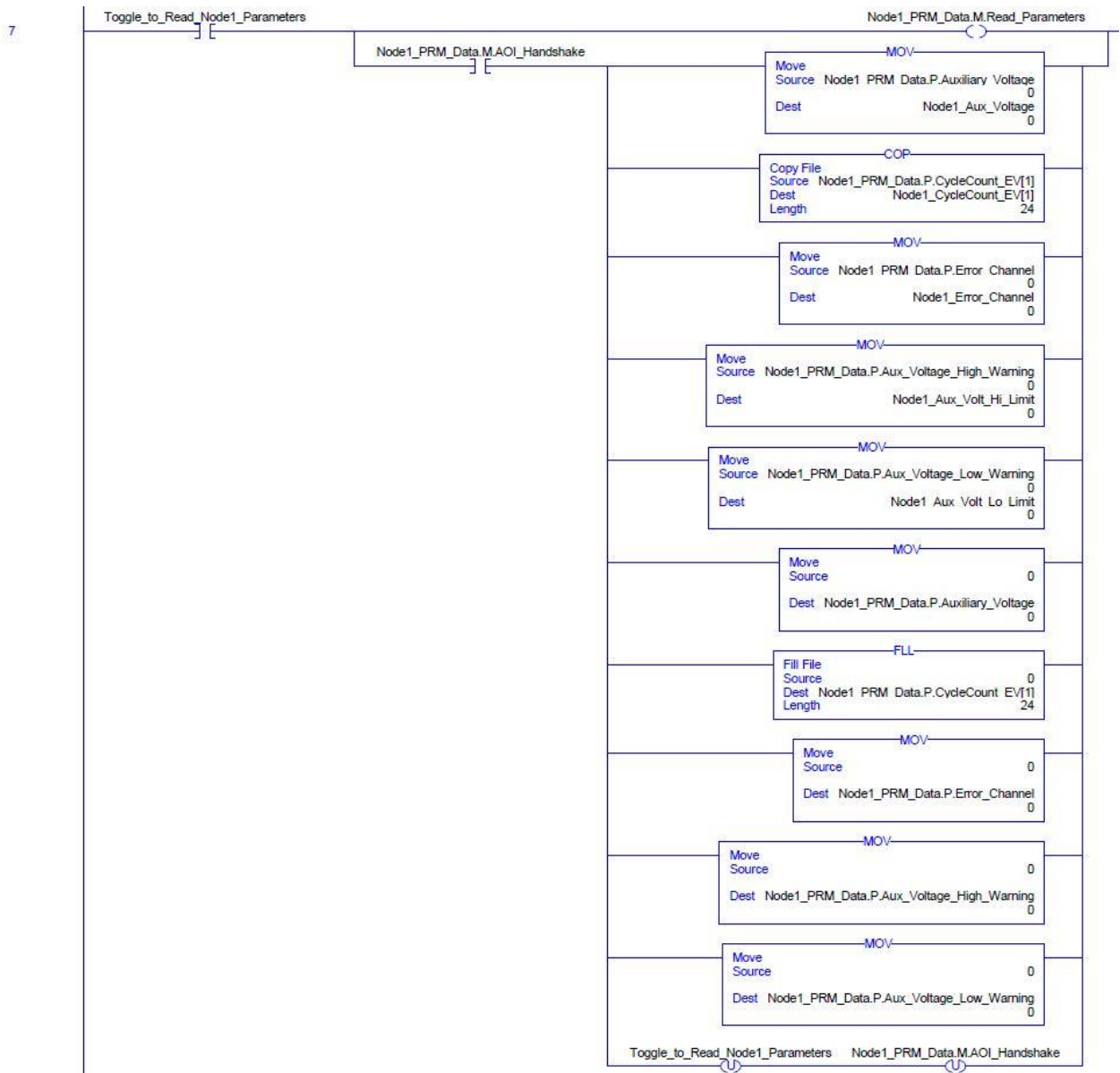
It is important to note the difference between cyclic and acyclic data. Process Data (cyclic) is updated without a request; whereas Parameter Data (acyclic) requires the program to toggle a bit to read or write data contained inside the slave device. See appendix for all data points available. See ladder logic examples below:

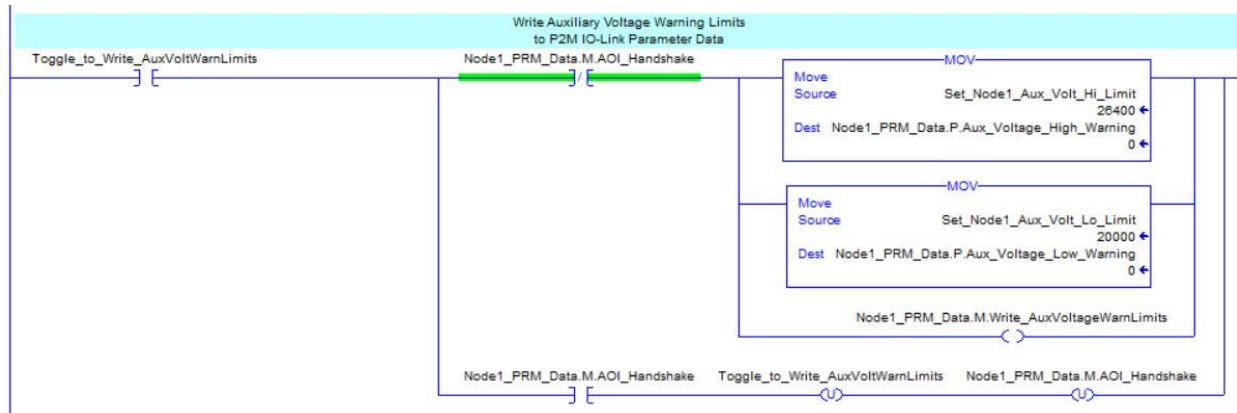
a. Read Parameter Data (Acyclic)

*****When initiating a read OR a write request you must reset the “Node1 PRM Data.M.AOI_Handshake” bit AFTER processing the data*****

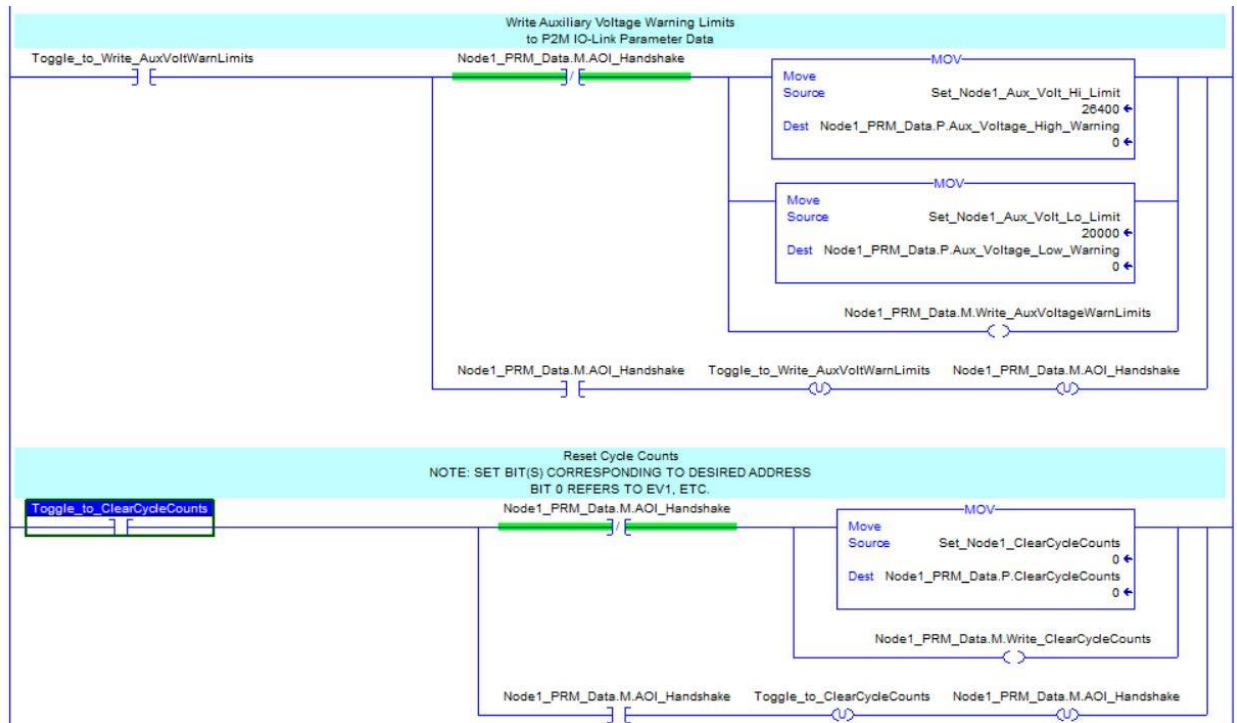
It is also important to not initiate multiple read or write requests at the same time.

Write your logic such that only one of the request bits is turned on at a time and wait for the Handshake bit to go high before executing the next request.



b. Write Parameter Data (Acyclic)

Note: When sending new limit setpoints to the P2M module, the values will not be written unless the high limit is more than one volt greater than the low limit, and greater than zero.



APPENDIX

Process Data Structures

User Defined / Add-On Defined Data Structures utilized by AOI “P2M2HBVL_P2HL_AB_BNI006A_PD_Rx”

Name:

Description:

Members: Data Type Size: 16 byte(s)

	Name	Data Type	Style	Description	External Access
☒	Status	UDT_P2M2HBVL_P2HL_Process_Inputs			Read/Write
☒	EV	UDT_P2M2HBVL_P2HL_Process_Outputs			Read/Write

P2M2HBVL or P2HL with BALLUFF BNI006A – AOI QSG

Name:

UDT_P2M2HBVL_P2HL_ProcData

Description:

Members:
Data Type Size: 16 byte(s)

Name	Data Type	Style	Description	External Access
☐ Status	UDT_P2M2HBVL_P2HL_Process_Inputs			Read/Write
☐ Fail_Safe	BOOL	Decimal		Read/Write
☐ Aux_Volt_Warning	BOOL	Decimal		Read/Write
☐ Aux_Volt_Failure	BOOL	Decimal		Read/Write
☐ Module_Failure	BOOL	Decimal		Read/Write
☐ Module_Over_Temp	BOOL	Decimal		Read/Write
☐ Module_Over_Load	BOOL	Decimal		Read/Write
☐ Short_Circuit	BOOL	Decimal		Read/Write
☐ Output_Stage_Not_Available	BOOL	Decimal		Read/Write
☐ Device_OK	BOOL	Decimal		Read/Write
☐ Mismatch_Fault	BOOL	Decimal		Read/Write
☐ Comm_Fault	BOOL	Decimal		Read/Write
☐ Validation_Failed	BOOL	Decimal		Read/Write
☐ Event_1_Error_Code	SINT	Decimal		Read/Write
☐ Event_1_Add_Code_1	SINT	Decimal		Read/Write
☐ Event_1_Add_Code_2	SINT	Decimal		Read/Write
☐ Event_2_Error_Code	SINT	Decimal		Read/Write
☐ Event_2_Add_Code_1	SINT	Decimal		Read/Write
☐ Event_2_Add_Code_2	SINT	Decimal		Read/Write
☐ Event_3_Error_Code	SINT	Decimal		Read/Write
☐ Event_3_Add_Code_1	SINT	Decimal		Read/Write
☐ Event_3_Add_Code_2	SINT	Decimal		Read/Write

Name:

Description:

Members: Data Type Size: 16 byte(s)

Name	Data Type	Style	Description	External Access
EV	UDT_P2M2HBVL_P2HL_Process_Outputs			Read/Write
_01	BOOL	Decimal	EV1 Command	Read/Write
_02	BOOL	Decimal	EV2 Command	Read/Write
_03	BOOL	Decimal	EV3 Command	Read/Write
_04	BOOL	Decimal	EV4 Command	Read/Write
_05	BOOL	Decimal	EV5 Command	Read/Write
_06	BOOL	Decimal	EV6 Command	Read/Write
_07	BOOL	Decimal	EV7 Command	Read/Write
_08	BOOL	Decimal	EV8 Command	Read/Write
_09	BOOL	Decimal	EV9 Command	Read/Write
_10	BOOL	Decimal	EV10 Command	Read/Write
_11	BOOL	Decimal	EV11 Command	Read/Write
_12	BOOL	Decimal	EV12 Command	Read/Write
_13	BOOL	Decimal	EV13 Command	Read/Write
_14	BOOL	Decimal	EV14 Command	Read/Write
_15	BOOL	Decimal	EV15 Command	Read/Write
_16	BOOL	Decimal	EV16 Command	Read/Write
_17	BOOL	Decimal	EV17 Command	Read/Write
_18	BOOL	Decimal	EV18 Command	Read/Write
_19	BOOL	Decimal	EV19 Command	Read/Write
_20	BOOL	Decimal	EV20 Command	Read/Write
_21	BOOL	Decimal	EV21 Command	Read/Write
_22	BOOL	Decimal	EV22 Command	Read/Write
_23	BOOL	Decimal	EV23 Command	Read/Write
_24	BOOL	Decimal	EV24 Command	Read/Write

Parameter Data Structures

User Defined / Add-On Defined Data Structures utilized by AOI

“P2M2HBVL_P2HL_AB_BNI006A_PRM_Rx”

Name:

Description:

Members: Data Type Size: 328 byte(s)

	Name	Data Type	Style	Description	External Access
☒	P	UDT_P2M2HBVL_P2HL_Parameter_Data			Read/Write
☒	M	UDT_P2M2HBVL_P2HL_MSG_Data			Read/Write

Name:

Description:

Members: Data Type Size: 328 byte(s)

	Name	Data Type	Style	Description	External Access
☒	P	UDT_P2M2HBVL_P2HL_Parameter_Data			Read/Write
☐	Auxiliary_Voltage	INT	Decimal		Read/Write
☐	Error_Channel	DINT	Decimal		Read/Write
☐	Aux_Voltage_High_Warning	INT	Decimal		Read/Write
☐	Aux_Voltage_Low_Warning	INT	Decimal		Read/Write
☐	CycleCount_EV	DINT[25]	Decimal		Read/Write
☐	ClearCycleCounts	DINT	Decimal		Read/Write
☐	SystemCommand	INT	Decimal		Read/Write

P2M2HBVL or P2HL with BALLUFF BNI006A – AOI QSG

Name:

UDT_P2M2HBVL_P2HL_PRMDData

Description:

Members:

Data Type Size: 328 byte(s)

	Name	Data Type	Style	Description	External Access
	P	UDT_P2M2HBVL_P2HL_Parameter_Data			Read/Write
	M	UDT_P2M2HBVL_P2HL_MSG_Data			Read/Write
	Read_Parameters	BOOL	Decimal		None
	Write_AuxVoltageWarnLimits	BOOL	Decimal		None
	Write_ClearCycleCounts	BOOL	Decimal		None
	Write_SysCommand	BOOL	Decimal		None
	AOI_Handshake	BOOL	Decimal		None
	Write	BOOL	Decimal		None
	Reset	BOOL	Decimal		None
	Done	BOOL	Decimal		None
	Error	BOOL	Decimal		None
	status	INT	Decimal		None
	ReceiveData	SINT[97]	Decimal		None
	SendData	SINT[97]	Decimal		None
	WriteData	DINT	Decimal		None