



P2M2HBVL OR P2HL IO-LINK  
ADD-ON INSTRUCTION FOR ROCKWELL  
PLC WITH AB 1732E-8IOLM12R  
ETHERNET/IP IO-LINK MASTER  
QUICK START GUIDE

TABLE OF CONTENTS

PREFACE .....4

Process Data Add-On Instruction Set Up .....5

    USING THE INSTRUCTION ..... 10

Parameter Data Add-On Instruction Set Up .....12

    CONFIGURING THE PATH..... 17

    SETTING UP WRITE MESSAGE..... 17

MESSAGE ERROR HANDLING .....20

APPENDIX .....21

    Process Data Structures .....21

    Parameter Data Structures.....23

Figure 1 1732E AOP data structure with P2H on channel 0 using IODD file .....	5
Figure 2 Setting up IOL Master using IODD or Generic Profile's.....	6
Figure 3 How to import AOI.....	7
Figure 4 List of process data aoI's and data types after import.....	8
Figure 5 Select AOI to add to rung .....	9
Figure 6 Data Placement for Process data function block .....	9
Figure 7 P2M/P2H UDT tag.....	10
Figure 8 Fully configured AOI.....	10
Figure 9 Example Toggling Solenoid Valves (Cyclic) .....	11
Figure 10 Monitoring Status Bits (Cyclic) .....	11
Figure 11 How to import AOI.....	12
Figure 12 LIST OF PROCESS DATA AOI'S AND DATA TYPES AFTER IMPORT .....	13
Figure 13 Select AOI to add to rung.....	14
Figure 14 Creating AOI instance tag .....	14
Figure 15 Completed Function block example .....	15
Figure 16 Read Message Configuration .....	15
Figure 17 Read message trigger and Read Data structure.....	16
Figure 18 Configure Path for Message Block.....	16
Figure 19 Write message configuration.....	17
Figure 20 Write data tags .....	18
Figure 21 Clear Cycle counts.....	18
Figure 22 Write High/low Aux Voltage limit.....	19
Figure 23 Reset Cycle Counts and System Command Example .....	19
Figure 24 Message Error handling.....	20
Figure 25 Process Data Structures.....	21
Figure 26 Process Data Structure Inputs.....	21
Figure 27 Process Data Structure Outputs.....	22
Figure 28 Parameter Data Structures.....	23
Figure 29 Parameter Data structure.....	23
Figure 30 Parameter Data Message Data.....	24
Figure 31 Parameter Data Message Error.....	24

## 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 Allen Bradley 1732E-8IOLM12R EIP IO-Link Master module. The QSG assumes that you are already using the Rockwell Add On Profile (AOP) v 3.01.60 for the 1732-E8IOLM12R master module and that it is communicating to the AB PLC via an Ethernet-IP network. An older AOP can be used but some feature of the module are not available however the function blocks should still function.

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\_1732E8IOLM12R\_PRM\_Rx" AOI facilitates the call-up of the acyclic service data.

The "P2M2HBVL\_P2HL\_AB\_1732E8IOLM12R\_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:

[http://solutions.parker.com/PDN\\_softwarefiles](http://solutions.parker.com/PDN_softwarefiles)

## PROCESS DATA ADD-ON INSTRUCTION SET UP

The “P2M2HBVL\_P2HL\_AB\_1732E8IOLM12R\_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 an 1732E-8IOL IO-Link Master and using the Add On Profile provided by AllenBradley.

To set up the 1732E-8IOL please follow the instructions provided by Rockwell. This instruction set covers how to set up P2M/P2H Add On Instructions after the 1732E-8IOL has been already set up. The image below shows the AOP data structure along with a P2H module assigned to channel 0 using the **P2H IODD file** (which is contained in the Parker download package). Note you can use a Generic Device Profile if you do not want to use the IODD, but you should specify **1 Input Byte and 3 Output bytes** for IO-Link data size at a minimum for this function block to work properly.

**FIGURE 1 1732E AOP DATA STRUCTURE WITH P2H ON CHANNEL 0 USING IODD FILE**

+	AB_1734_8IOL.C	{...}	{...}		AB:1732_8IOL1.C:0
-	AB_1734_8IOL.I	{...}	{...}		AB:1732_8IOLC784D7E1:I:0
+	AB_1734_8IOL.I.Fault	2#0000_0000_0000_0000_00...		Binary	DINT
+	AB_1734_8IOL.I.Status	{...}	{...}		AB:1732_8IOL_Struct_Status:I:0
+	AB_1734_8IOL.I.Ch0DiagEvent	{...}	{...}		AB:1732_8IOL_Struct_Event:I:0
+	AB_1734_8IOL.I.Ch1DiagEvent	{...}	{...}		AB:1732_8IOL_Struct_Event:I:0
+	AB_1734_8IOL.I.Ch2DiagEvent	{...}	{...}		AB:1732_8IOL_Struct_Event:I:0
+	AB_1734_8IOL.I.Ch3DiagEvent	{...}	{...}		AB:1732_8IOL_Struct_Event:I:0
+	AB_1734_8IOL.I.Ch4DiagEvent	{...}	{...}		AB:1732_8IOL_Struct_Event:I:0
+	AB_1734_8IOL.I.Ch5DiagEvent	{...}	{...}		AB:1732_8IOL_Struct_Event:I:0
+	AB_1734_8IOL.I.Ch6DiagEvent	{...}	{...}		AB:1732_8IOL_Struct_Event:I:0
+	AB_1734_8IOL.I.Ch7DiagEvent	{...}	{...}		AB:1732_8IOL_Struct_Event:I:0
-	AB_1734_8IOL.I.Ch0Data	{...}	{...}	Binary	SINT[1]
-	AB_1734_8IOL.I.Ch0Data[0]	2#0000_0000		Binary	SINT
	AB_1734_8IOL.I.Ch0Data[0].0	0		Decimal	BOOL
	AB_1734_8IOL.I.Ch0Data[0].1	0		Decimal	BOOL
	AB_1734_8IOL.I.Ch0Data[0].2	0		Decimal	BOOL
	AB_1734_8IOL.I.Ch0Data[0].3	0		Decimal	BOOL
	AB_1734_8IOL.I.Ch0Data[0].4	0		Decimal	BOOL
	AB_1734_8IOL.I.Ch0Data[0].5	0		Decimal	BOOL
	AB_1734_8IOL.I.Ch0Data[0].6	0		Decimal	BOOL
	AB_1734_8IOL.I.Ch0Data[0].7	0		Decimal	BOOL
-	AB_1734_8IOL.O	{...}	{...}		AB:1732_8IOLF7833396:O:0
-	AB_1734_8IOL.O.Ch0Data	{...}	{...}	Binary	SINT[3]
+	AB_1734_8IOL.O.Ch0Data[0]	2#0000_0001		Binary	SINT
+	AB_1734_8IOL.O.Ch0Data[1]	2#0000_0001		Binary	SINT
+	AB_1734_8IOL.O.Ch0Data[2]	2#0000_0001		Binary	SINT

Example of AOP with Channel 0 configured with Parker P2H Module. Assigning the P2H using the IODD to Channel 0 automatically sets the data size.

FIGURE 2 SETTING UP IOL MASTER USING IODD OR GENERIC PROFILE'S

**IO-Link**

1732E-8IOL12MR/A

- Ch 0 - IO-Link
- Ch 1 - IO-Link
- Ch 2 - IO-Link
- Ch 3 - IO-Link
- Ch 4 - IO-Link
- Ch 5 - IO-Link
- Ch 6 - IO-Link
- Ch 7 - IO-Link

DANGER: Parameter changes by external source are shown only after Refresh. External changes could be overwritten without notice.

Status: Offline

Channel	Mode	Vendor	Device	Application Specific Name	Electronic Keying	Process Data Input	Data Storage
0	IO-Link	Parker Hannifin	P2H Network No...	P2H1966	Exact Match		Disabled
1	IO-Link						
2	IO-Link						
3	IO-Link						
4	IO-Link						
5	IO-Link						
6	IO-Link						
7	IO-Link						

Change...

Refresh

OK Cancel Apply Help

**Change Channel Configuration**

Channel	Mode	Vendor	Device	Application Specific Name	Electronic Keying	Process Data Input	Data Storage	Change Device
0	IO-Link	Parker Hannifin	P2H Network Node - ...	P2H1966	Exact M...		Disabled	...
1	IO-Link							...
2	IO-Link							...
3	IO-Link							...
4	IO-Link							...
5	IO-Link							...
6	IO-Link							...
7	IO-Link							...

Discover Devices... Path: DESKTOP-M

changes by external are shown only after External changes are overwritten without notice.

Status: Offline

**Select IO-Link Device**

- IO-Link
  - Allen-Bradley
    - Color Registration Mark Sensor - 45CRM
    - Inductive Proximity Sensor - 871C
    - Inductive Proximity Sensor - 871FM
    - Inductive Proximity Sensor - 871TM
    - Measurement Sensor - 45LMS
    - Photoelectric Sensor - 42AF
    - Photoelectric Sensor - 42EF
    - Photoelectric Sensor - 42JT
    - Pressure Sensor - 836P
    - Temperature Sensor - 837T
  - Parker Hannifin
    - Parker Moduflex IO-Link
      - P2M2HBVL12400A (Class A) | P2M2HBVL12400B (Class B)
      - V1.1
        - P2H Network Node - IO Link
        - Parker Moduflex IO Link
  - Generic

Register IODD... Create Cancel

**Select IO-Link Device**

- IO-Link
  - Allen-Bradley
    - Color Registration Mark Sensor - 45CRM
    - Inductive Proximity Sensor - 871C
    - Inductive Proximity Sensor - 871FM
    - Inductive Proximity Sensor - 871TM
    - Measurement Sensor - 45LMS
    - Photoelectric Sensor - 42AF
    - Photoelectric Sensor - 42EF
    - Photoelectric Sensor - 42JT
    - Pressure Sensor - 836P
    - Temperature Sensor - 837T
  - Parker Hannifin
    - Parker Moduflex IO-Link
      - P2M2HBVL12400A (Class A) | P2M2HBVL12400B (Class B)
      - V1.1
        - P2H Network Node - IO Link
        - Parker Moduflex IO Link
  - Generic

Register IODD... Create Cancel

**Edit Generic Device Properties**

Channel: 1

Electronic Keying: Disabled

Vendor ID:

Device ID:

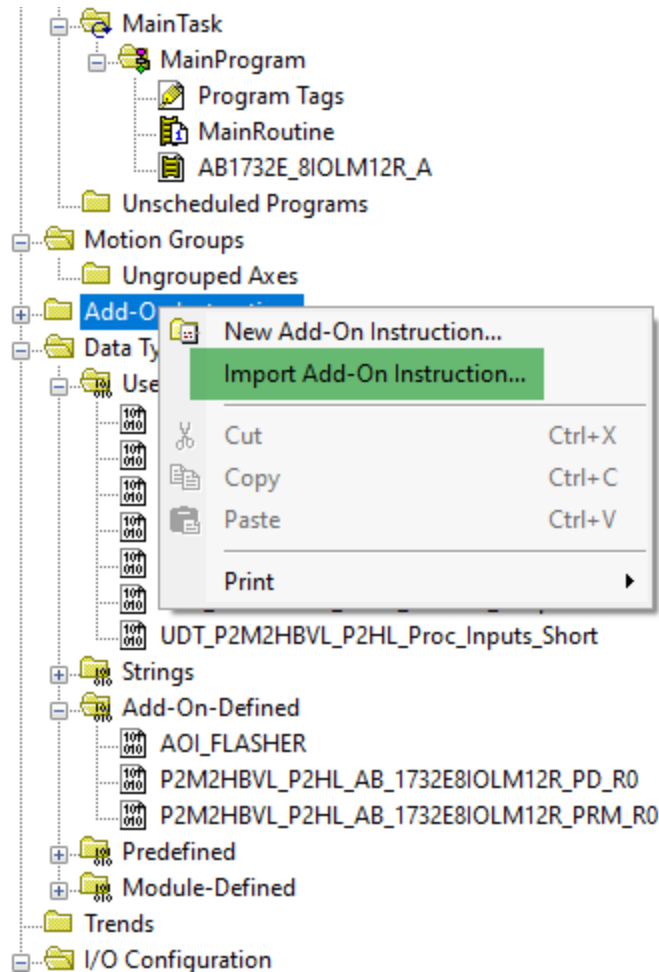
Input Length: 1 bytes

Output Length: 3 bytes

Register IODD... Create Cancel

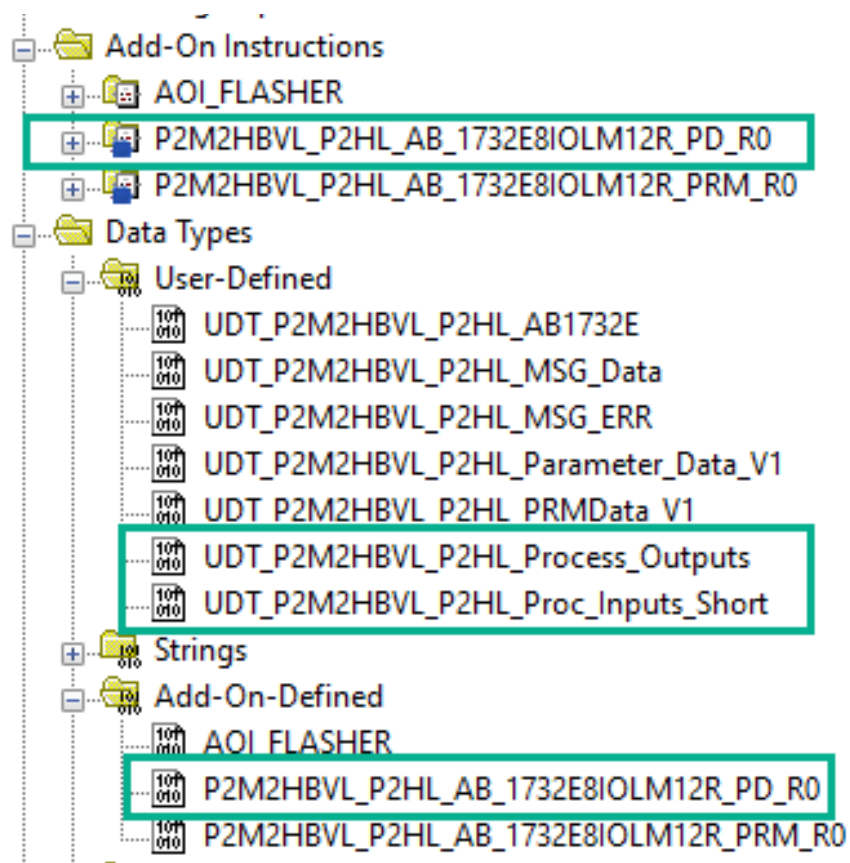
- 1) Go to Ethernet devices and select properties of the 8IOLM12R then select IO-Link.
- 2) Select the IO-Link Channel to configure
- 3) Select change
- 4) Select change device
- 5) If you choose to import the IODD file you see it listed
- 6) Otherwise you can choose generic
- 7) Input the data sizes 1 byte Input and 3 bytes Output lengths.

FIGURE 3 HOW TO IMPORT AOI



1. Right click Add-On Instruction in Controller Organizer and select “Import Add-On Instruction...”
2. Select the “P2M2HBVL\_P2HL\_AB\_1732E8IOLM12R\_PD\_Rx” where \_Rx is the revision of AOI.
3. 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.

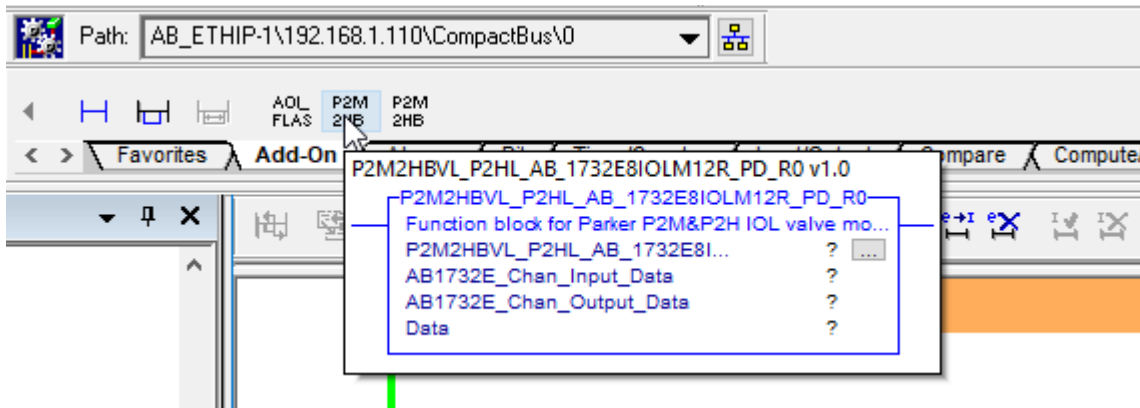
#### FIGURE 4 LIST OF PROCESS DATA AOI'S AND DATA TYPES AFTER IMPORT





4. Add instance of AOI instruction to an empty rung of ladder by clicking on the P2M under the Add-On tab in the top toolbar. The instruction will drop onto the selected rung.

FIGURE 5 SELECT AOI TO ADD TO RUNG



5. Once the channel is configured for the P2M/P2H IO-link node the data will look like Figure 6. Place the data tags according to input data and output data in the AOI block.

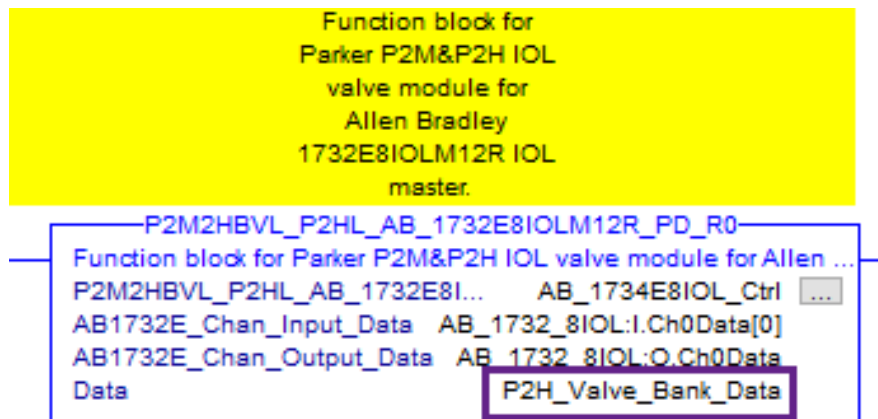
FIGURE 6 DATA PLACEMENT FOR PROCESS DATA FUNCTION BLOCK

When the Channel is configured the data size will show up like below. 1 Input byte and 3 Output bytes for P2M/P2H IO-Link node.

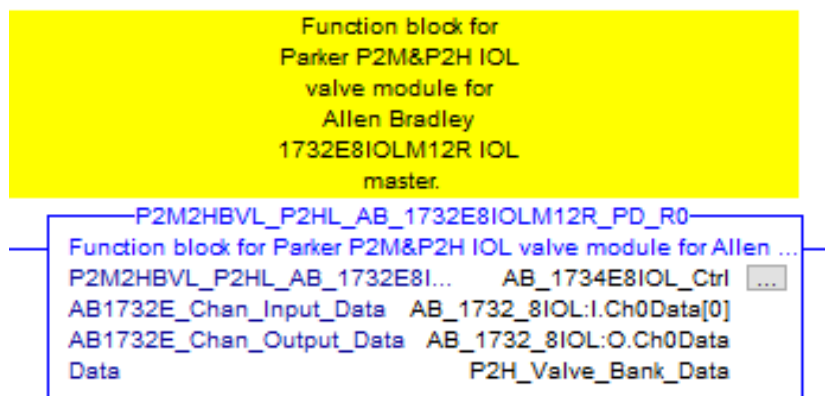
AB_1734_8IOL-I.Ch7DiagEvent	{...}	{...}		AB:1732_8IOL_Struct_Event:I:0
AB_1734_8IOL-I.Ch0Data	{...}	{...}	Binary	SINT[1]
AB_1734_8IOL-I.Ch0Data[0]	2#1000_0000		Binary	SINT
AB_1734_8IOL-U	{...}	{...}		AB:1732_8IOLF7833396:0:0
AB_1734_8IOL-O.Ch0Data	{...}	{...}	Binary	SINT[3]
AB_1734_8IOL-O.Ch0Data[0]	2#0000_0001		Binary	SINT
AB_1734_8IOL-O.Ch0Data[1]	2#0000_0001		Binary	SINT
AB_1734_8IOL-O.Ch0Data[2]	2#0000_0001		Binary	SINT
AB_1734E IOL ctrl	{...}	{...}		P2M2HBVL_P2HL_AB_1732E8IOLM12R

FIGURE 7 P2M/P2H UDT TAG



6. The “Data” variable when created will assign the UDT that the raw data will be reassigned too.

FIGURE 8 FULLY CONFIGURED AOI



## 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 P2H\_Valve\_Bank\_Data.xxx and P2H\_Valve\_Bank\_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:

FIGURE 9 EXAMPLE TOGGLING SOLENOID VALVES (CYCLIC)

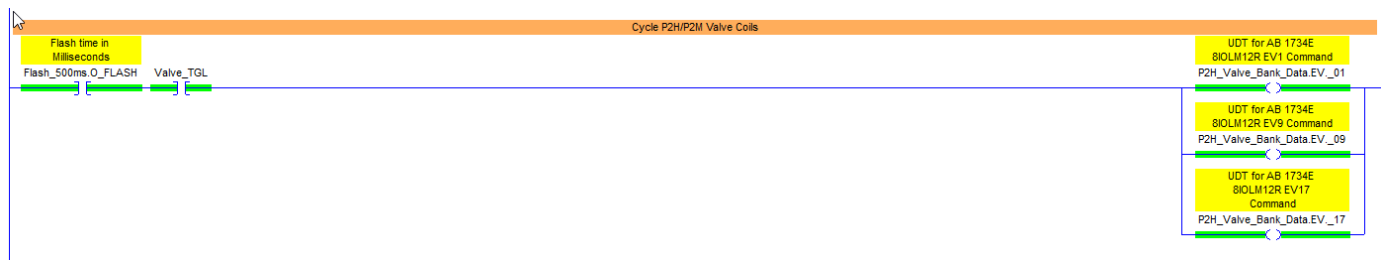
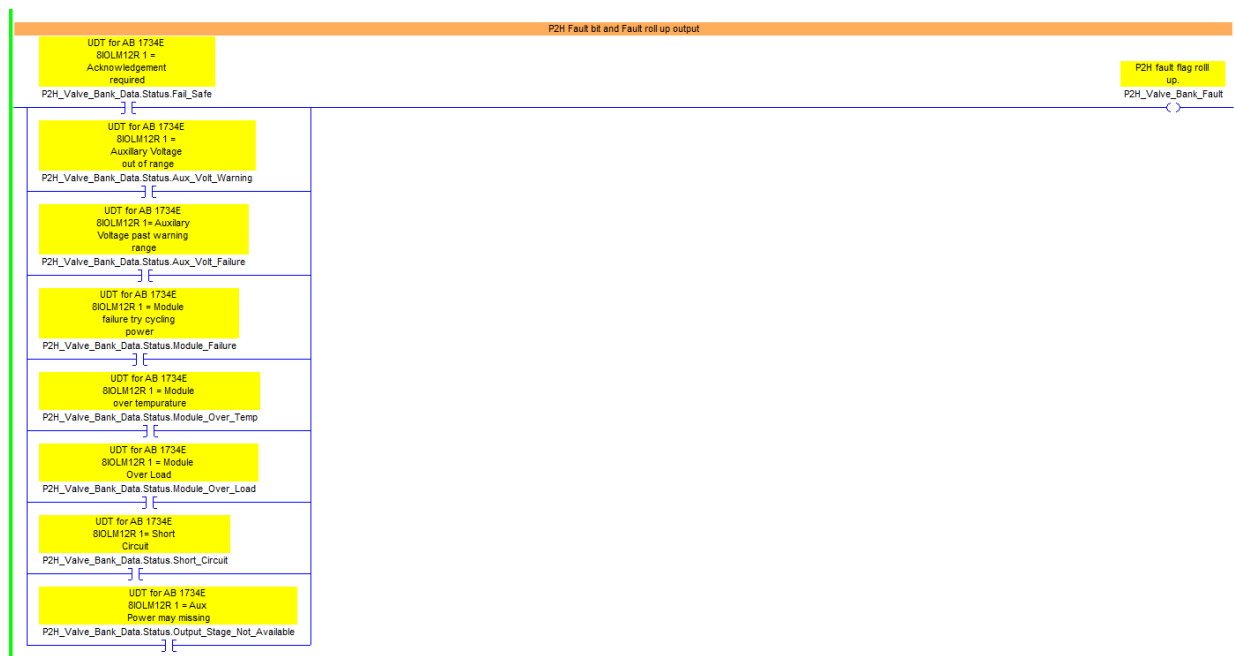


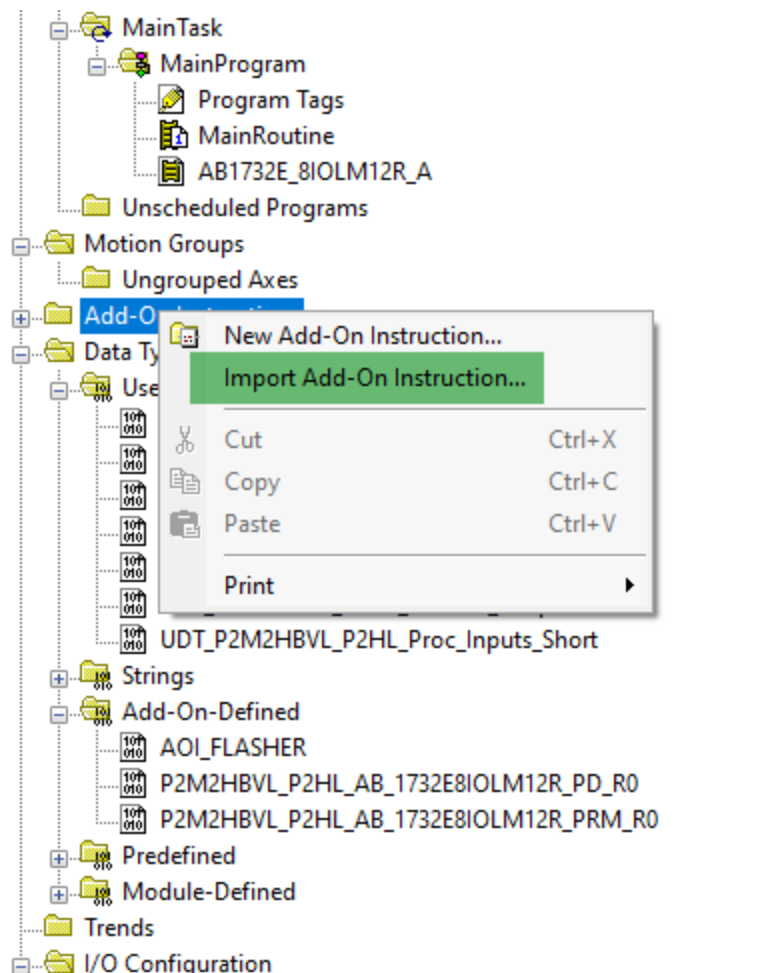
FIGURE 10 MONITORING STATUS BITS (CYCLIC)



## PARAMETER DATA ADD-ON INSTRUCTION SET UP

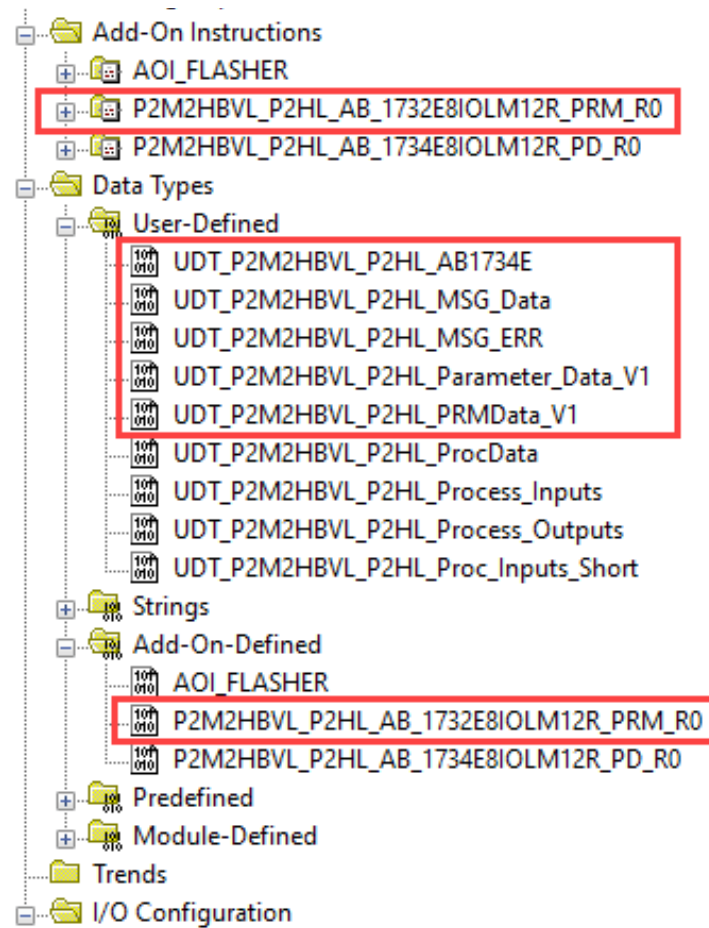
The “P2M2HBVL\_P2HL\_AB\_1732E8IOLM12R\_PRM\_Rx” AOI simplifies the usage of Parker P2M and P2H IO-Link devices with Allen-Bradley CompactLogix, ControlLogix and GuardLogix PLCs when connected, via Ethernet/IP, to a Allen Bradley 1732E8IOLM12R IO-Link Master. The AOI facilitates the reading and writing of parameter data between the PLC and the Parker P2M or P2H IO-Link device.

FIGURE 11 HOW TO IMPORT AOI



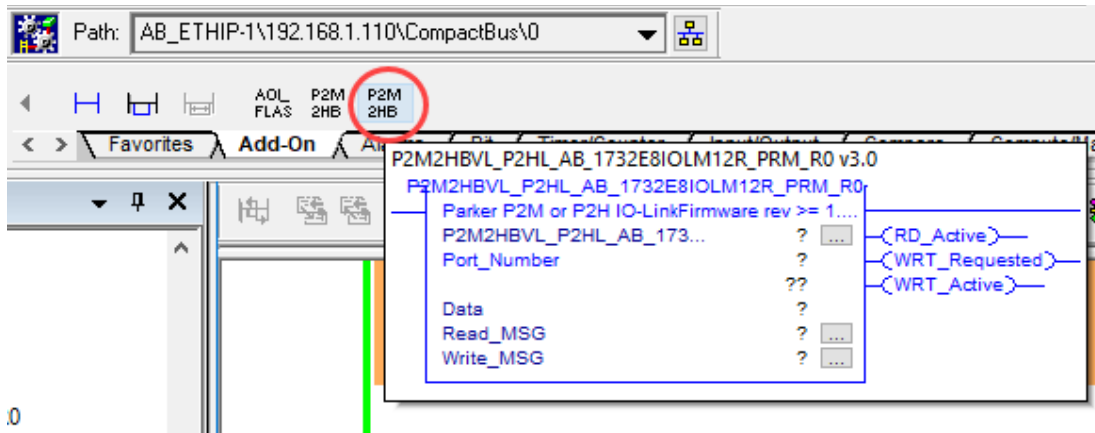
1. Right click Add-On Instruction in Controller Organizer and select “Import Add-On Instruction...”
2. Select the “P2M2HBVL\_P2HL\_AB\_1732E8IOLM12R\_PRM\_Rx” where \_Rx is the revision of AOI.
3. Choose OK on Import Configuration Window and you should then see the new AOI instance along with User-Defined and Add-On Defined data types created in the controller organizer.

FIGURE 12 LIST OF PROCESS DATA AOI'S AND DATA TYPES AFTER IMPORT



1. Add instance of instruction to an empty rung of ladder by clicking on the P2M under the Add-On tab in the top toolbar. The instruction will drop onto the selected rung.

FIGURE 13 SELECT AOI TO ADD TO RUNG



2. 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. Configure the Data input variable next since this is used by the block for other needs. See Appendix for structure breakdown of the "Data" variable.

FIGURE 14 CREATING AOI INSTANCE TAG

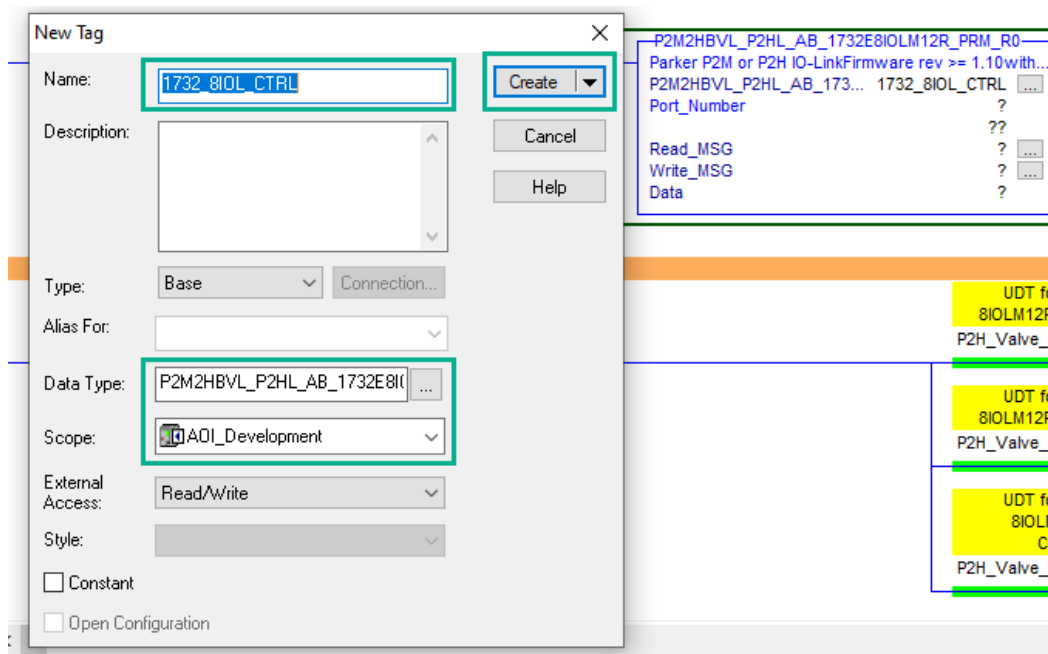
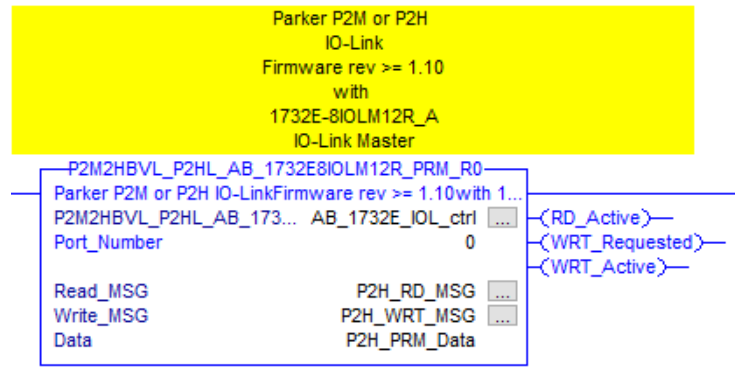


FIGURE 15 COMPLETED FUNCTION BLOCK EXAMPLE

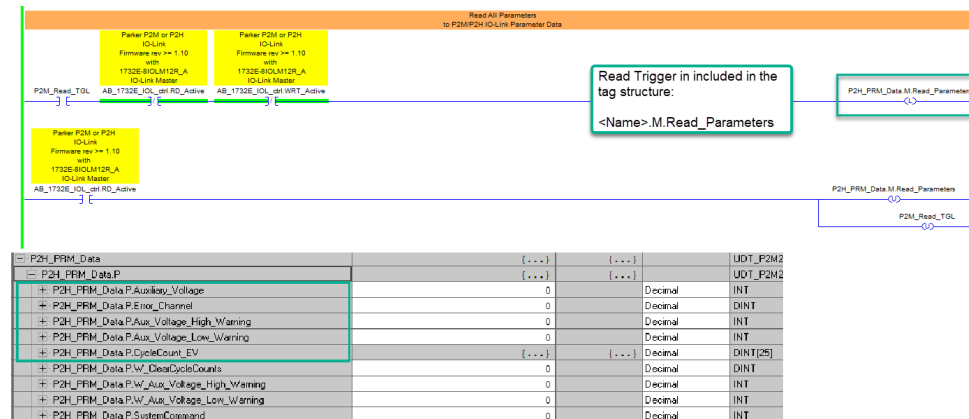


Note: for the following configuration windows:

1. The Service Code = 4d for read operations
2. Class field shall be 3a3. All other fields are controlled by the block
3. Source Element comes for the the Tag created under Data on the block. In this case it would be assigned to: **P2H\_PRM\_Data.M.SendData**
4. Source Element comes for the the Tag created under Data on the block. In this case it would be assigned to: **P2H\_PRM\_Data.M.ReceiveData**
5. Source byte can equal 1 this is controlled by the blocks internal code.

FIGURE 16 READ MESSAGE CONFIGURATION

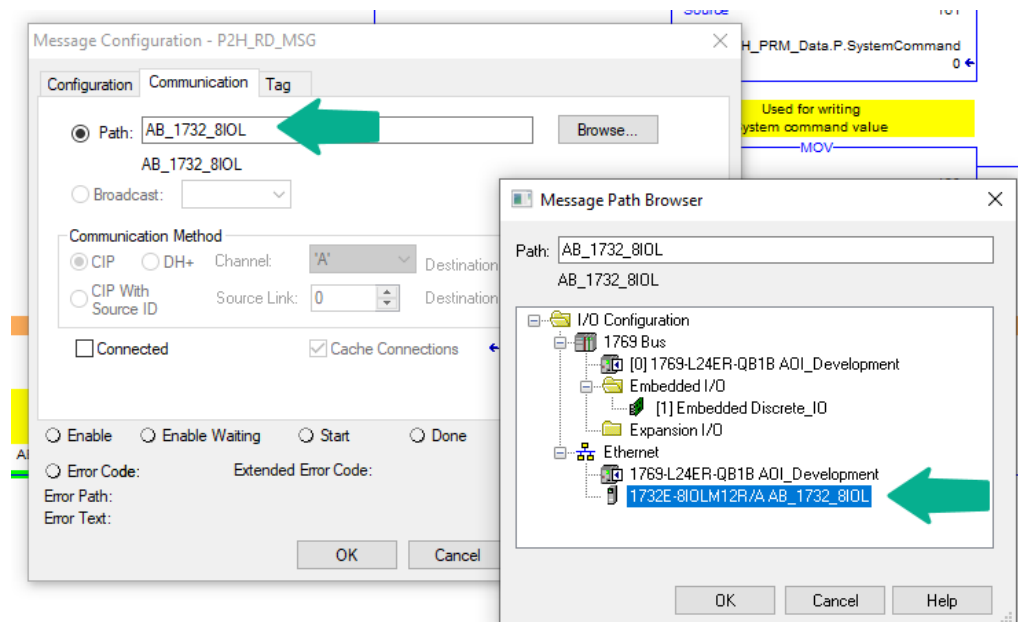
FIGURE 17 READ MESSAGE TRIGGER AND READ DATA STRUCTURE



Once the Read Message variable is created you can trigger a read using the tag in the data structure: <Name>.M.Read\_Parameters the block will read the following:

1. AUX Votage
2. ChannelError
3. AUX Voltage High Warning set point
4. AUX Voltage Low Warning set point
5. CycleCounts for Valves

FIGURE 18 CONFIGURE PATH FOR MESSAGE BLOCK





## CONFIGURING THE PATH

Open the message instruction communication tab (Figure 18). Select browse and select the appropriate 1732E-8IOLM12R in the Ethernet tree. Then Click OK.

## SETTING UP WRITE MESSAGE

The instructions for setting up the write message is the same except:

1. The Service Code = 4e for read operations
2. Class field shall be 3a3. All other fields are controlled by the block
3. Source Element comes for the the Tag created under Data on the block. In this case it would be assigned to: **P2H\_PRM\_Data.M.SendData**
4. Source Element comes for the the Tag created under Data on the block. In this case it would be assigned to: **P2H\_PRM\_Data.M.WriteData**
5. Source byte can equal 1 this is controlled by the blocks internal code.

FIGURE 19 WRITE MESSAGE CONFIGURATION

Message Configuration - P2H\_WRT\_MSG

Configuration Communication Tag

Message Type: CIP Generic

Service Type: Custom

Service Code: 4e (Hex) Class: 3a3 (Hex) Instance: 2 Attribute: 0 (Hex)

Source Element: P2H\_PRM\_Data.M.Se

Source Length: 2 (Bytes)

Destination Element: IM\_Data.M.WriteData

New Tag...

☐ Enable
 ☐ Enable Waiting
 ☐ Start
 ☐ Done
 Done Length: 0

☐ Error Code:
 Extended Error Code:
 ☐ Timed Out

Error Path:

Error Text:

OK Cancel Apply Help

## USING WRITE MESSAGE

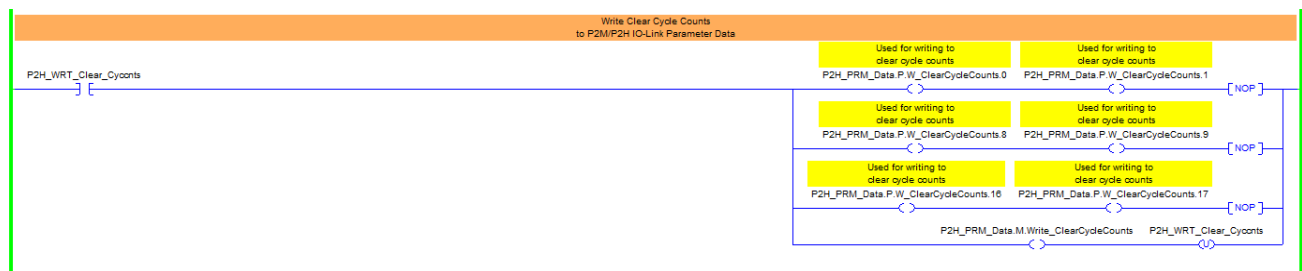
Writing data to the P2H/P2M IO-Link device is handled independently. There is an individual write request for each write parameter. See examples below.

1. Clear Cycle Counts
2. Aux voltage high warning setpoint
3. Aux voltage low warning setpoint

**FIGURE 20 WRITE DATA TAGS**

[-] P2H_PRM_Data	{...}	{...}		UDT_P2M2
[-] P2H_PRM_Data.P	{...}	{...}		UDT_P2M2
+ P2H_PRM_Data.P.Auxiliary_Voltage	0		Decimal	INT
+ P2H_PRM_Data.P.Error_Channel	0		Decimal	DINT
+ P2H_PRM_Data.P.Aux_Voltage_High_Warning	0		Decimal	INT
+ P2H_PRM_Data.P.Aux_Voltage_Low_Warning	0		Decimal	INT
+ P2H_PRM_Data.P.CycleCount_EV	{...}	{...}	Decimal	DINT[25]
+ P2H_PRM_Data.P.W_ClearCycleCounts	0		Decimal	DINT
+ P2H_PRM_Data.P.W_Aux_Voltage_High_Warning	0		Decimal	INT
+ P2H_PRM_Data.P.W_Aux_Voltage_Low_Warning	0		Decimal	INT
+ P2H_PRM_Data.P.SystemCommand	0		Decimal	INT

**FIGURE 21 CLEAR CYCLE COUNTS**

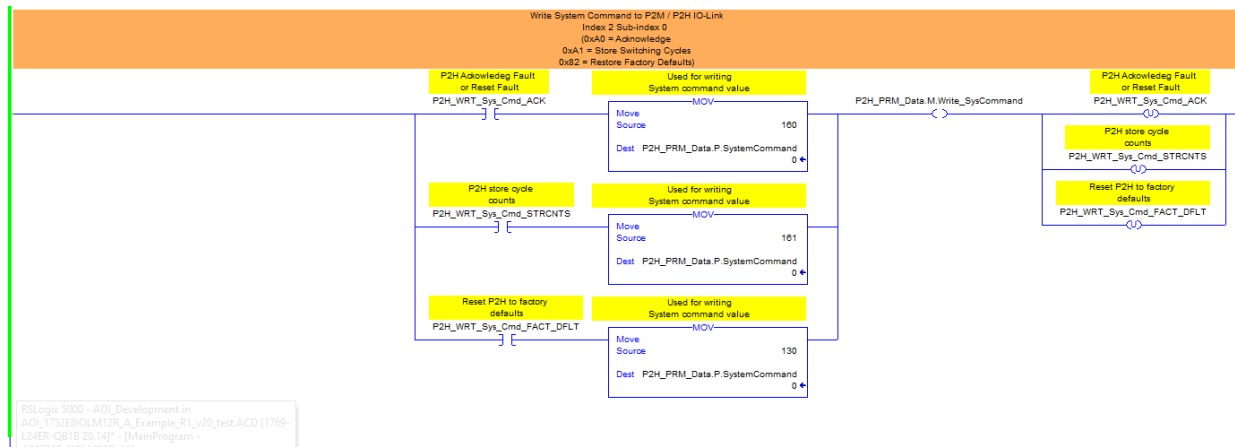


**FIGURE 22 WRITE HIGH/LOW AUX VOLTAGE LIMIT**



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.

**FIGURE 23 RESET CYCLE COUNTS AND SYSTEM COMMAND EXAMPLE**



## MESSAGE ERROR HANDLING

Built in the data structure is message error handling when read or write is triggered if there is an error suspected then the error codes will be displayed here based on what was triggered to read or write.

Please see Allen Bradley message instruction error code list to decode the error.

**FIGURE 24 MESSAGE ERROR HANDLING**

[-] P2H_PRM_Data	{...}	{...}
[+] P2H_PRM_Data.P	{...}	{...}
[+] P2H_PRM_Data.M	{...}	{...}
[-] P2H_PRM_Data.E	{...}	{...}
[+] P2H_PRM_Data.E.Aux_Vtg_Rd_ERR	16#0000	
[+] P2H_PRM_Data.E.Aux_Vtg_Rd_EXTRERR	16#0000	
[+] P2H_PRM_Data.E.Cyc_Cnts_Rd_ERR	16#0000	
[+] P2H_PRM_Data.E.Cyc_Cnts_Rd_EXTRERR	16#0000	
[+] P2H_PRM_Data.E.Chnl_Err_Rd_ERR	16#0000	
[+] P2H_PRM_Data.E.Chnl_Err_Rd_EXTRERR	16#0000	
[+] P2H_PRM_Data.E.Vtg_Wrn_H_L_Rd_ERR	16#0000	
[+] P2H_PRM_Data.E.Vtg_Wrn_H_L_Rd_EXTRERR	16#0000	
[+] P2H_PRM_Data.E.Aux_Vtg_Wrn_Lim_Wr_ERR	16#0000	
[+] P2H_PRM_Data.E.Aux_Vtg_Wrn_Lim_Wr_EXTRERR	16#0000	
[+] P2H_PRM_Data.E.Clr_Cyc_Cnts_Wr_ERR	16#0000	
[+] P2H_PRM_Data.E.Clr_Cyc_Cnts_Wr_EXTRERR	16#0000	
[+] P2H_PRM_Data.E.Sys_Cmd_Wr_ERR	16#0000	
[+] P2H_PRM_Data.E.Sys_Cmd_Wr_EXTRERR	16#0000	

## APPENDIX

### PROCESS DATA STRUCTURES

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

“P2M2HBVL\_P2HL\_AB\_1732E8IOLM12R\_PD\_Rx”

**FIGURE 25 PROCESS DATA STRUCTURES**

Data Type: UDT\_P2M2HBVL\_P2HL\_AB1732E

Name: UDT\_P2M2HBVL\_P2HL\_AB1732E

Description: UDT for AB 1734E  
8IOLM12R

Members: Data Type Size: 8 byte(s)

	Name	Data Type	Style	Description	External Access
<input checked="" type="checkbox"/>	Status	UDT_P2M2HBVL_P2HL_Proc_Inputs_Short			Read/Write
<input checked="" type="checkbox"/>	EV	UDT_P2M2HBVL_P2HL_Process_Outputs			Read/Write
<input type="checkbox"/>					

Buttons: Move Up, Move Down, OK, Cancel, Apply, Help

**FIGURE 26 PROCESS DATA STRUCTURE INPUTS**

Name: UDT\_P2M2HBVL\_P2HL\_Proc\_Inputs\_Short

Description:

Members: Data Type Size: 4 byte(s)

	Name	Data Type	Style	Description	External Access
<input checked="" type="checkbox"/>	Fail_Safe	BOOL	Decimal	1 = Acknowledgement required	Read/Write
<input type="checkbox"/>	Aux_Volt_Warning	BOOL	Decimal	1 = Auxillary Voltage out of range	Read/Write
<input type="checkbox"/>	Aux_Volt_Failure	BOOL	Decimal	1 = Auxillary Voltage past warning range	Read/Write
<input type="checkbox"/>	Module_Failure	BOOL	Decimal	1 = Module failure try cycling power	Read/Write
<input type="checkbox"/>	Module_Over_Temp	BOOL	Decimal	1 = Module over temperature	Read/Write
<input type="checkbox"/>	Module_Over_Load	BOOL	Decimal	1 = Module Over Load	Read/Write
<input type="checkbox"/>	Short_Circuit	BOOL	Decimal	1 = Short Circuit	Read/Write
<input type="checkbox"/>	Output_Stage_Not_Available	BOOL	Decimal	1 = Aux Power may missing	Read/Write
<input type="checkbox"/>					

FIGURE 27 PROCESS DATA STRUCTURE OUTPUTS

Name:

Description:

Members: Data Type Size: 4 byte(s)

Name	Data Type	Style	Description	External Access
_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
<div style="border: 1px solid black; padding: 2px;">10P 010</div>				

## PARAMETER DATA STRUCTURES

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

"P2M2HBVL\_P2HL\_AB\_1732E8IOLM12R\_PRM\_Rx"

FIGURE 28 PARAMETER DATA STRUCTURES

Name:

Description:

Members: Data Type Size: 360 byte(s)

	Name	Data Type	Style	Description	External Access
<input checked="" type="checkbox"/>	P	UDT_P2M2HBVL_P2HL_Parameter_Data_V1			Read/Write
<input checked="" type="checkbox"/>	M	UDT_P2M2HBVL_P2HL_MSG_Data			Read/Write
<input checked="" type="checkbox"/>	E	UDT_P2M2HBVL_P2HL_MSG_ERR			Read/Write
<input type="checkbox"/>					

FIGURE 29 PARAMETER DATA STRUCTURE

Name:

Description:

Members: Data Type Size: 124 byte(s)

	Name	Data Type	Style	Description	External Access
<input checked="" type="checkbox"/>	Auxiliary_Voltage	INT	Decimal	Aux voltage actual value	Read/Write
<input checked="" type="checkbox"/>	Error_Channel	DINT	Decimal	Channel error acutal value	Read/Write
<input checked="" type="checkbox"/>	Aux_Voltage_High_Warning	INT	Decimal	Aux voltage acutal high set point value	Read/Write
<input checked="" type="checkbox"/>	Aux_Voltage_Low_Warning	INT	Decimal	Aux voltage acutal low set point value	Read/Write
<input checked="" type="checkbox"/>	CycleCount_EV	DINT[25]	Decimal	Valve cycle count value	Read/Write
<input checked="" type="checkbox"/>	W_ClearCycleCounts	DINT	Decimal	Used for writing to clear cycle counts	Read/Write
<input checked="" type="checkbox"/>	W_Aux_Voltage_High_Warning	INT	Decimal	Used for writing Aux voltage high warnin	Read/Write
<input checked="" type="checkbox"/>	W_Aux_Voltage_Low_Warning	INT	Decimal	Used for writing Aux voltage low warnin	Read/Write
<input checked="" type="checkbox"/>	SystemCommand	INT	Decimal	Used for writing System command value	Read/Write
<input type="checkbox"/>					

**FIGURE 30 PARAMETER DATA MESSAGE DATA**

Name:

Description:

Members: Data Type Size: 208 byte(s)

Name	Data Type	Style	Description	External Access
Read_Parameters	BOOL	Decimal		None
Write_AuxVoltageWarnLimits	BOOL	Decimal		None
Write_ClearCycleCounts	BOOL	Decimal		None
Write_SysCommand	BOOL	Decimal		None
ADJ_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

**FIGURE 31 PARAMETER DATA MESSAGE ERROR**

Name:

Description:

Members: Data Type Size: 28 byte(s)

Name	Data Type	Style	Description	External Access
Aux_Vtg_Rd_ERR	INT	Hex		Read/Write
Aux_Vtc_Rd_EXTErr	INT	Hex		Read/Write
Cyc_Cnts_Rd_ERR	INT	Hex		Read/Write
Cyc_Cnts_Rd_EXTErr	INT	Hex		Read/Write
Chnl_Err_Rd_ERR	INT	Hex		Read/Write
Chnl_Err_Rd_EXTErr	INT	Hex		Read/Write
Vtg_Wrn_H_L_Rd_ERR	INT	Hex		Read/Write
Vtg_Wrn_H_L_Rd_EXTErr	INT	Hex		Read/Write
Aux_Vtg_Wrn_Lim_Wr_ERR	INT	Hex		Read/Write
Aux_Vtg_Wrn_Lim_Wr_EXTErr	INT	Hex		Read/Write
Clr_Cyc_Cnts_Wr_ERR	INT	Hex		Read/Write
Clr_Cyc_Cnts_Wr_EXTErr	INT	Hex		Read/Write
Sys_Cmd_Wr_ERR	INT	Hex		Read/Write
Sys_Cmd_Wr_EXTErr	INT	Hex		Read/Write