



P2M2HBVE MODUFLEX  
ETHERNET/IP  
FUNCTION BLOCK FOR  
OMRON PLC  
QUICK START GUIDE

## CONTENTS

PREFACE .....	3
1. P2M Ethernet/Ip device configuration procedure On sysmac .....	4
2. Process Data Function Block Instruction.....	7
2.1. Importing the function block .....	7
2.2. Call function block “P2M2HBVE_FB” into the project .....	8
2.3. Using the instruction, sample code .....	8
2.3.1. Solenoid Valves Control (Cyclic).....	9
2.3.2. Writing System Commands (Cyclic) .....	9
2.3.3. Monitoring Status Bits (Cyclic) .....	10
3. Parameter Data Example Code .....	11
3.2. Importing sample program .....	11
3.3. Parameter Code Instruction.....	11
3.3.1. Config request path.....	11
3.3.2. Reading Parameters.....	12
3.3.3. Writing Parameters.....	15
APPENDIX.....	17

## PREFACE

This Quick Start Guide (QSG) is designed to help integrate Parker Hannifin's P2M EtherNet/IP valve manifold into an Omron PLC environment.

The guide will walk the user through obtaining the necessary files, importing/configuring the FB, and initiating parameter reads and writes from/to the P2M EtherNet/IP device.

The "P2M2HBVE\_FB" FB facilitates communication and handling of process data between PLC and the Parker coupler EtherNet/IP slave device.

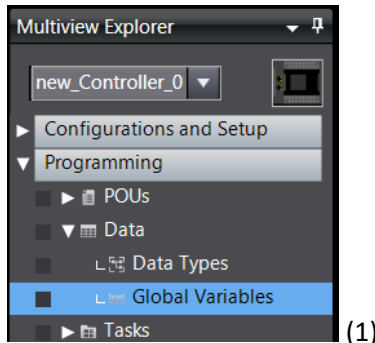
You can download resources such as the EDS configuration file, this QSG, the "P2M2HBVE\_PD" Function Block, a sample SYSMAC program "P2M2HBVE\_OMRON\_NX\_RO" and the full P2M manual here:

[http://www.parker.com/pdn/P2M\\_IE](http://www.parker.com/pdn/P2M_IE)

## 1. P2M ETHERNET/IP DEVICE CONFIGURATION PROCEDURE ON SYSMAC

a. Create process input and output data variable to connect to the target P2M.

In the (1) Multiview Explorer>> Programming>> Data>> Global Variables (2) edit Input and output variables.



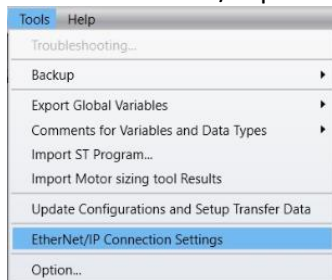
(1)

Name	Data Type	Initial Value	AT	Retain	Constant	Network Publish
P2M_EIP_Input	ARRAY[0..6] OF BYTE			<input type="checkbox"/>	<input type="checkbox"/>	Input
P2M_EIP_Output	ARRAY[0..3] OF BYTE			<input type="checkbox"/>	<input type="checkbox"/>	Output

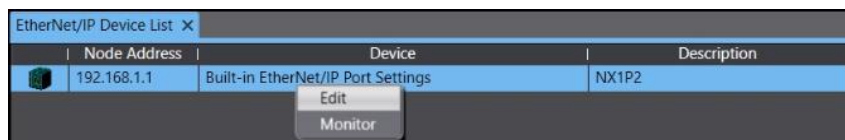
(2)

b. P2M File description installation

In the menu bar of Sysmac Studio interface (1) click on “Tools”>EtherNet/IP Connection Settings, (2) right click on PLC EtherNet/IP node then Edit Or double click the PLC EtherNet/IP node to open the “Built-in EtherNet/IP port settings connection settings”.



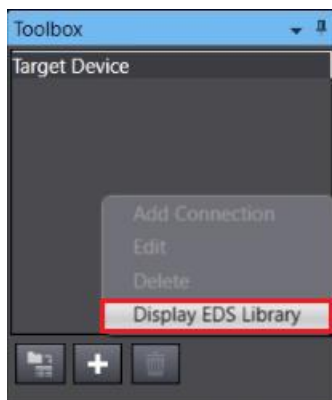
(1)



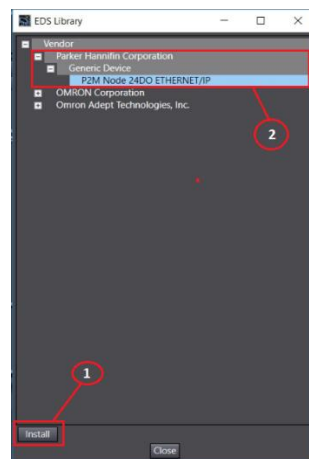
(2)

Install the P2M EDS file: right click in the Toolbox Tab and click on “Display EDS Library” to open the EDS Library (3).

Click on install and the select P2M EDS file in the folder where it is stored and click on open to view your device in the EDS Library, close the “EDS Library Interface” (3).



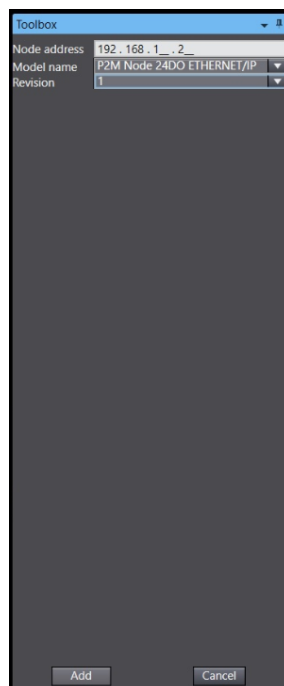
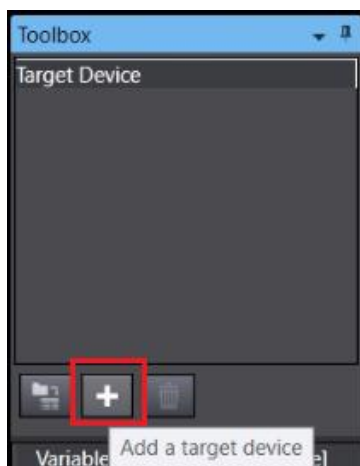
(3)



(4)

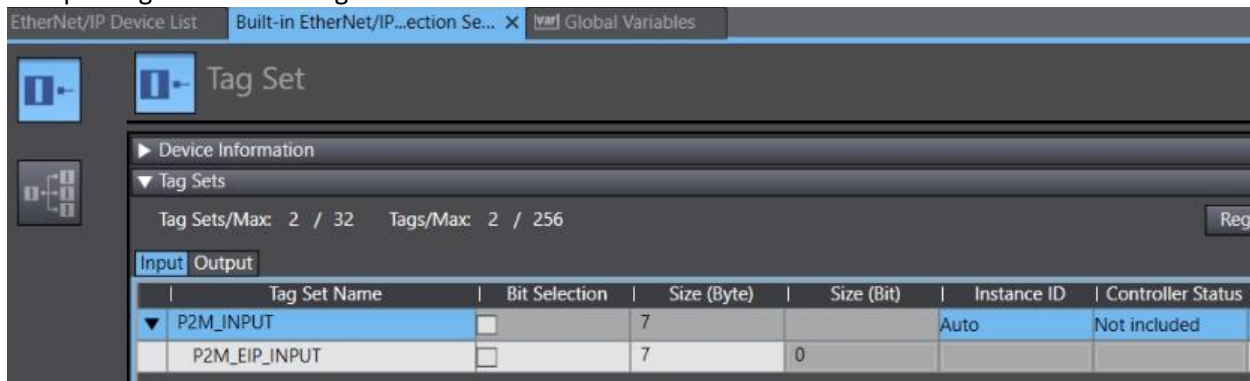
c. Add P2M device to the created project

In the toolbox Tab click on (+) “Add a target device”, then select the device and assign an IP address and click “Add”.

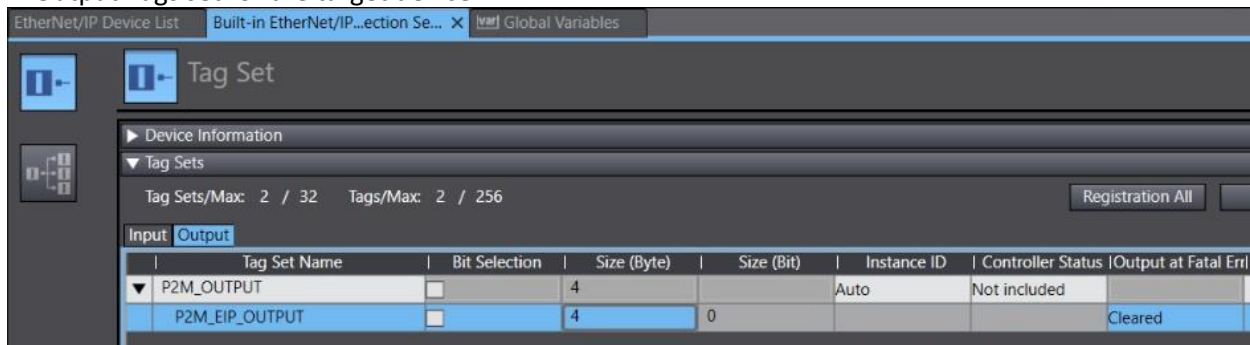


d. Settings procedure

Input Tags set for the target device



Output Tags set for the target device



Connect the settings tags to the P2M Target device

EtherNet/IP Device List   Built-in EtherNet/IP...ection Se... X   Global Variables

**Connection**

▼ Connection  
Connections/Max: 2 / 32

Target Device	Connection	Connection I/O	Input/Output	Target Variable	Size [Byte]	Originator Variable	Size [Byte]	Connection Type	RPI [ms]	Timeout V
192.168.1.2 P2M Node 24DO ETHER	default_001	Exclusive owner	Input	100	7	P2M_INPUT	7	Multi-cast con	50.0	RPI x 4
			Output	150	4	P2M_OUTPUT	4	Point to Point c		

1   2   3   4   5

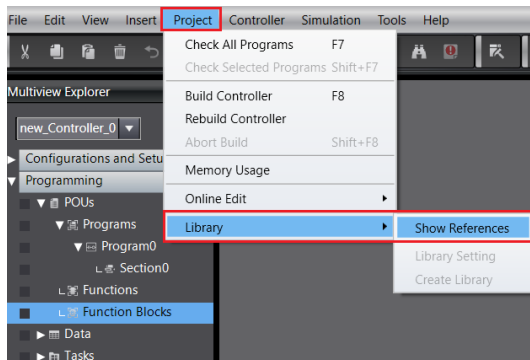
+

## 2. PROCESS DATA FUNCTION BLOCK INSTRUCTION

The “P2M2HBVE\_FB” Function Block simplifies the usage of Parker P2M EtherNet/IP devices with Omron NX and NJ PLCs when connected via EtherNet/IP. Data is mapped to user-friendly control and diagnostic tags on the PLC side.

### 2.1. Importing the function block

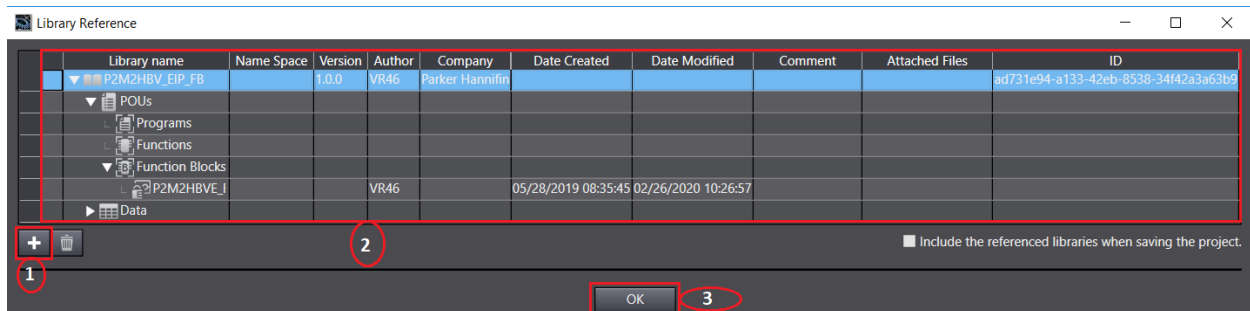
In the menu bar click on “Project>> Library>> Show References”



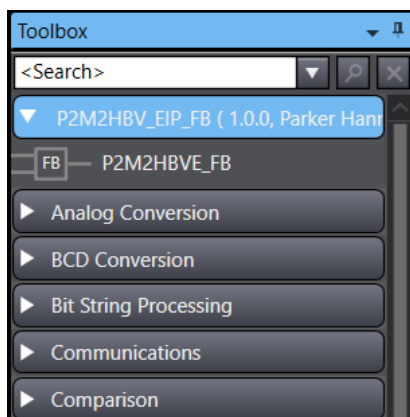
In the Reference Library Pop Up interface click (+), then move to the folder where the file Library



then click open to add the function block into the Library Reference.

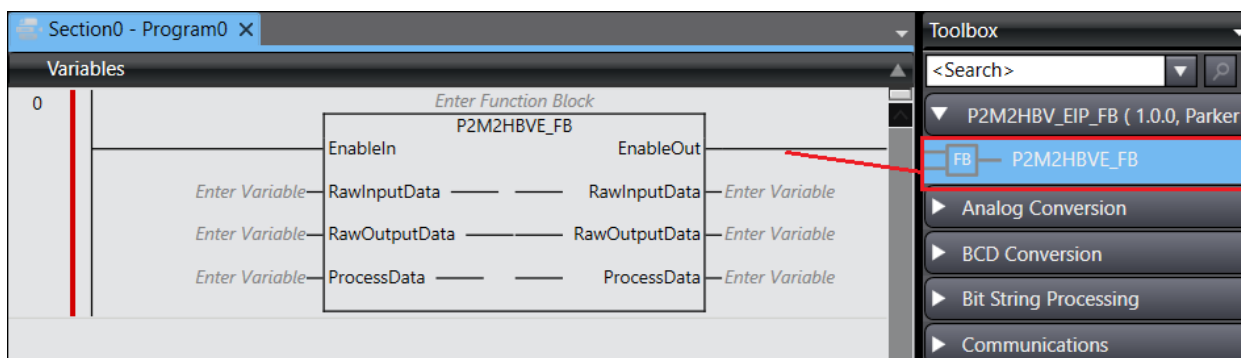


Click OK on Library reference and you should then see the new FB instance in the toolbox tab of the programming object.



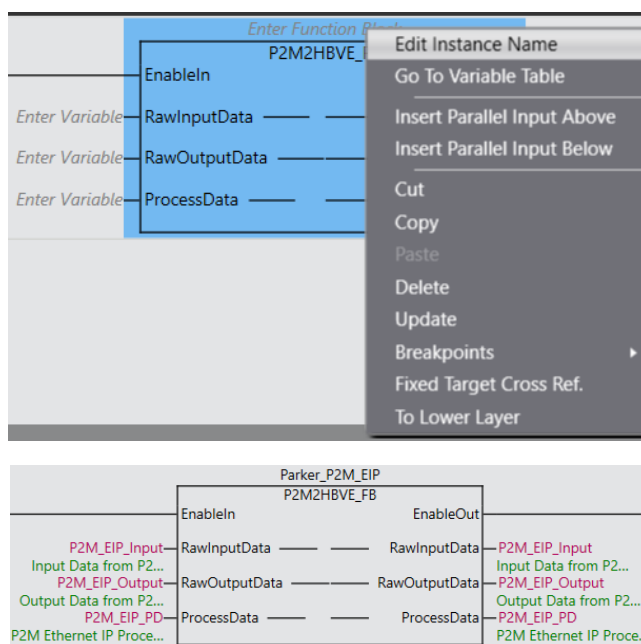
## 2.2. Call function block “P2M2HBVE\_FB” into the project

Drag and drop the function block “P2M2HBVE\_FB” from toolbox tab to the open program interface to add the instance of instruction to an empty rung of ladder.



Assign an instance name for the FB and create other tags necessary for operation. Right click on the “Enter Function Block” marks and select “Edit Instance Name” or “Edit” for the variable by clicking “Enter Variable”. Note that the name must be unique for each tag and each instance of the function block. The edited name instance or variable will appear with the correct data type in the program local variable.

The RawInputData and the RawOutputData should be pointed to the global input and output variable connected to the P2M device target. See Appendix for structure breakdown of the “Data” variable.

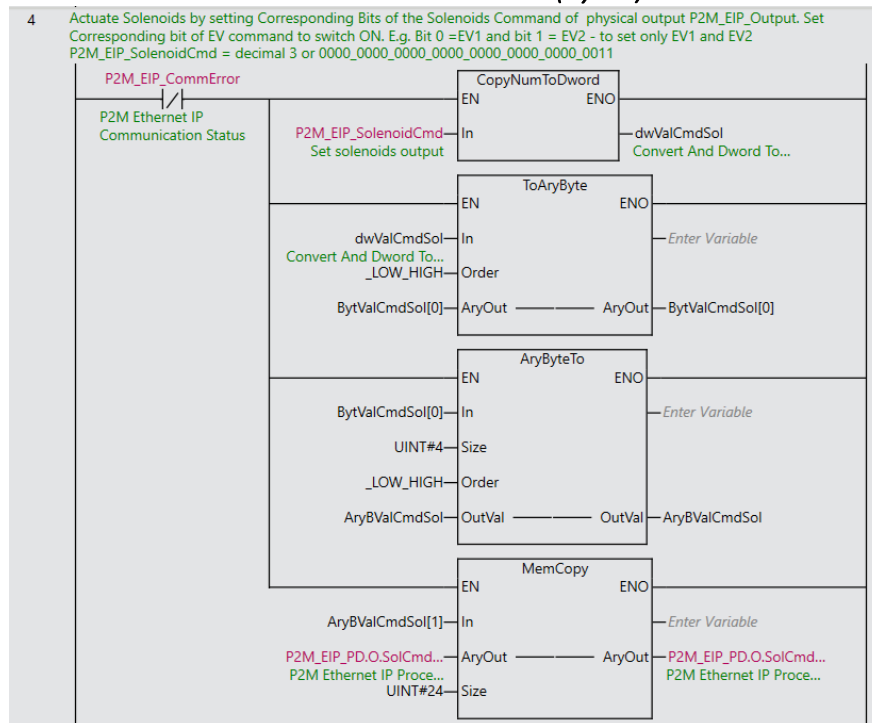


## 2.3. Using the instruction, sample code

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 data. Cyclic data includes input module status, valve output control and system command. This means that P2M\_EIP\_PD.I.xxx and P2M\_EIP\_PD.O.xxx are live tags with real data just by calling the FB. See appendix for all data points available. See ladder logic examples below:



### 2.3.1. Solenoid Valves Control (Cyclic)

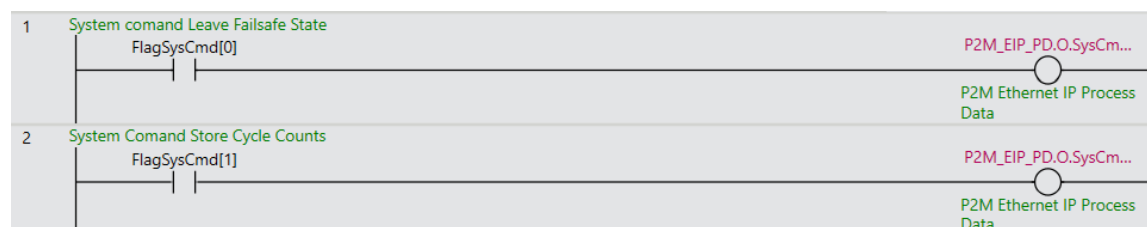


Value correlation to Solenoid Address	
P2M_EIP_SolenoidCmd = 1	"P2M_EIP_PD".O.SolCmd.EV[1]=1
P2M_EIP_SolenoidCmd = 2	"P2M_EIP_PD".O.SolCmd.EV[2]=1
P2M_EIP_SolenoidCmd = 4	"P2M_EIP_PD".O.SolCmd.EV[3]=1
P2M_EIP_SolenoidCmd = 8	"P2M_EIP_PD".O.SolCmd.EV[4]=1
P2M_EIP_SolenoidCmd = 16	"P2M_EIP_PD".O.SolCmd.EV[5]=1
.	
.	
.	
P2M_EIP_SolenoidCmd = 8 388 608	"P2M_EIP_PD".O.SolCmd.EV[24]=1

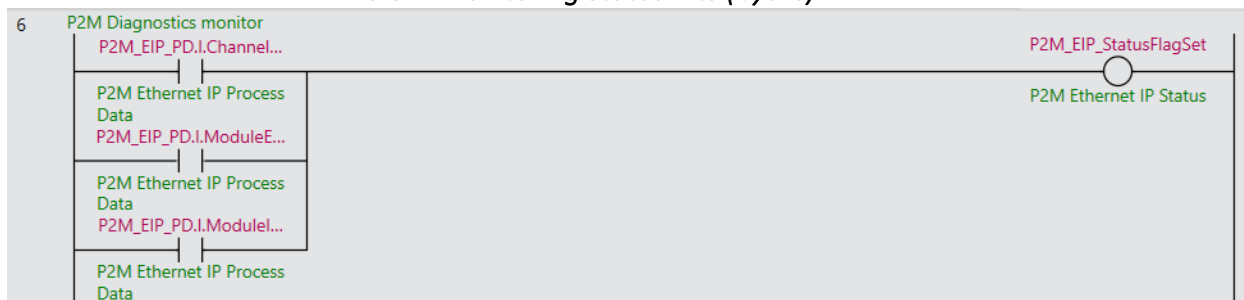
### 2.3.2. Writing System Commands (Cyclic)

Setting "SysCmd.LeaveFailsafeState" output element is equivalent to writing 0x01 to the first byte of output process data.

Setting "SysCmd.StoreCycleCounts" output element is equivalent to writing 0x02 to the first byte of output process data.



### 2.3.3. Monitoring Status Bits (Cyclic)



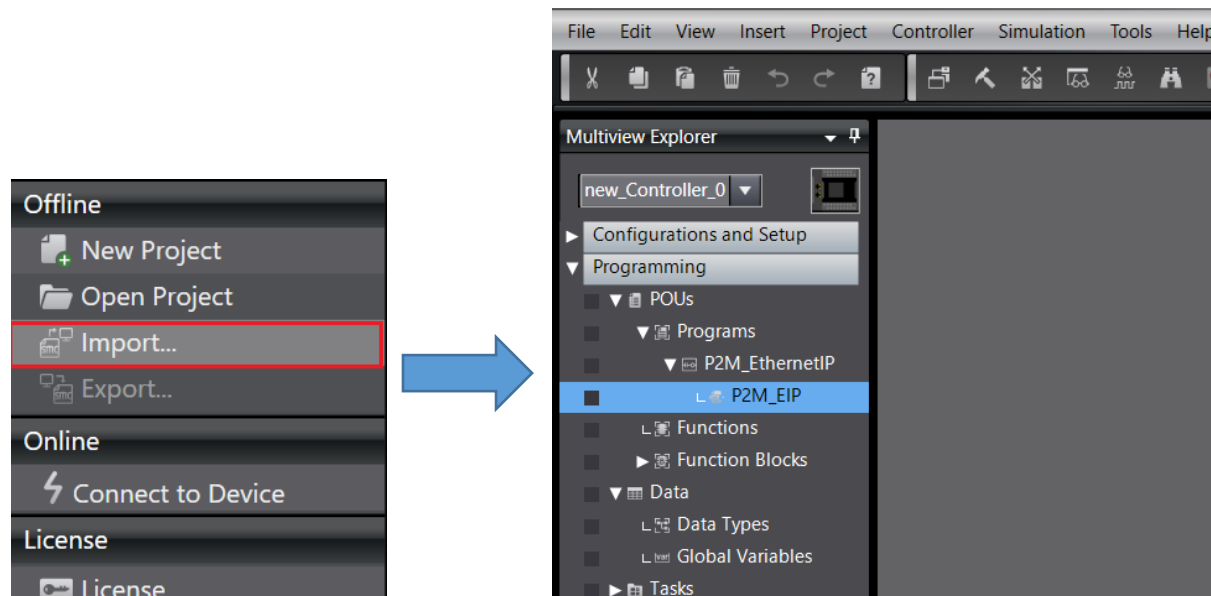
If any of the status bits are set, further information can be attained by evaluating specific diagnostic bits.

Channel Error	Module Error	Module info Flags
"P2M_EIP_PD".I.ChannelError.CH[1]	"P2M_EIP_PD".I.ModuleError.AckRequired	"P2M_EIP_PD".I.ModuleInfoFlags.WatchdogValveMicro
"P2M_EIP_PD".I.ChannelError.CH[2]	"P2M_EIP_PD".I.ModuleError.AuxVoltageWarning	"P2M_EIP_PD".I.ModuleInfoFlags.EEPROM_Error
"P2M_EIP_PD".I.ChannelError.CH[3]	"P2M_EIP_PD".I.ModuleError.AuxVoltageError	"P2M_EIP_PD".I.ModuleInfoFlags.WatchdogComMicro
"P2M_EIP_PD".I.ChannelError.CH[4]	"P2M_EIP_PD".I.ModuleError.TemperatureWarning	"P2M_EIP_PD".I.ModuleInfoFlags.HeartbeatNotToggling
"P2M_EIP_PD".I.ChannelError.CH[5]	"P2M_EIP_PD".I.ModuleError.OutputDriverChannelError	"P2M_EIP_PD".I.ModuleInfoFlags.HeartbeatState
"P2M_EIP_PD".I.ChannelError.CH[6]	"P2M_EIP_PD".I.ModuleError.ModuleError	
"P2M_EIP_PD".I.ChannelError.CH[7]	"P2M_EIP_PD".I.ModuleError.OutputStageNotAvailable	
"P2M_EIP_PD".I.ChannelError.CH[8]		
"P2M_EIP_PD".I.ChannelError.CH[9]		
"P2M_EIP_PD".I.ChannelError.CH[10]		
"P2M_EIP_PD".I.ChannelError.CH[11]		
"P2M_EIP_PD".I.ChannelError.CH[12]		
"P2M_EIP_PD".I.ChannelError.CH[13]		
"P2M_EIP_PD".I.ChannelError.CH[14]		
"P2M_EIP_PD".I.ChannelError.CH[15]		
"P2M_EIP_PD".I.ChannelError.CH[16]		
"P2M_EIP_PD".I.ChannelError.CH[17]		
"P2M_EIP_PD".I.ChannelError.CH[18]		
"P2M_EIP_PD".I.ChannelError.CH[19]		
"P2M_EIP_PD".I.ChannelError.CH[20]		
"P2M_EIP_PD".I.ChannelError.CH[21]		
"P2M_EIP_PD".I.ChannelError.CH[22]		
"P2M_EIP_PD".I.ChannelError.CH[23]		
"P2M_EIP_PD".I.ChannelError.CH[24]		

### 3. PARAMETER DATA EXAMPLE CODE

#### 3.2. Importing sample program

Launch "Sysmac studio" then click on Import, open the folder where "P2M2HBVE\_OMRON\_NX\_R0 QSG" sample code is stored, select the file .smc2 and click open

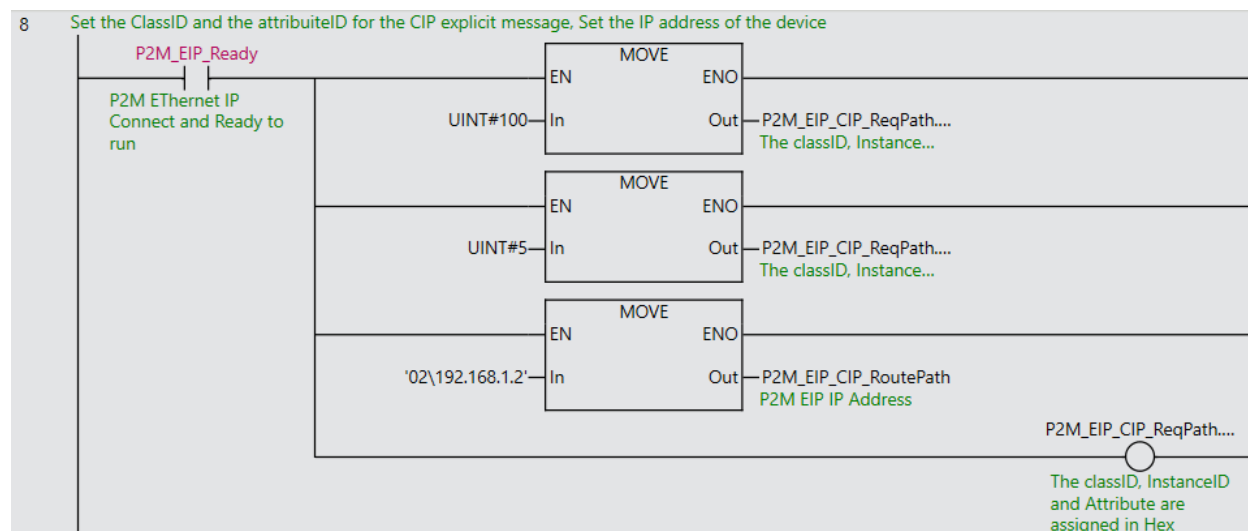


#### 3.3. Parameter Code Instruction

##### 3.3.1. Config request path

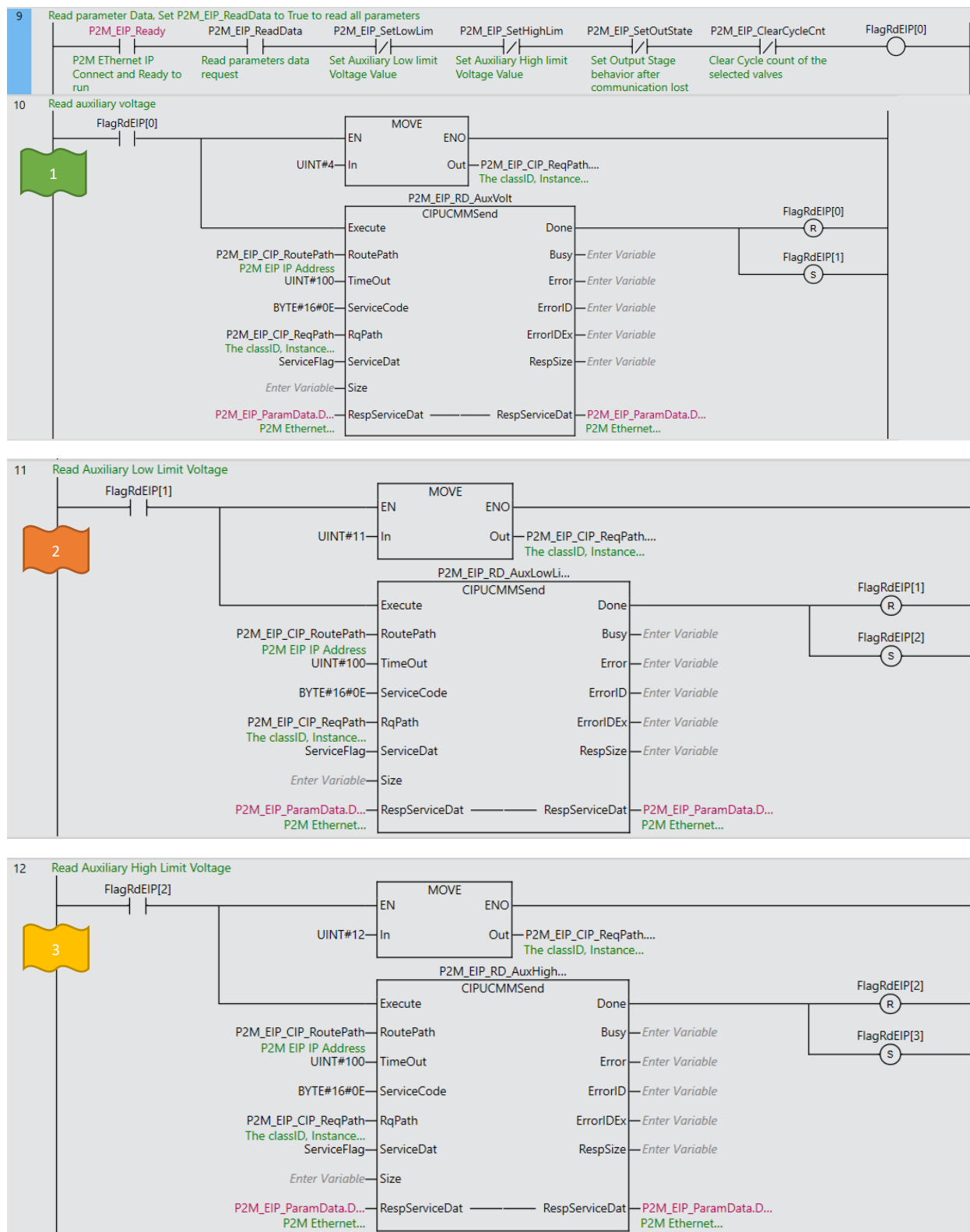
Set the request path. These data are important for the Function Blocks used for the Read/Write operation, change the IP address to match the IP of your connected device.

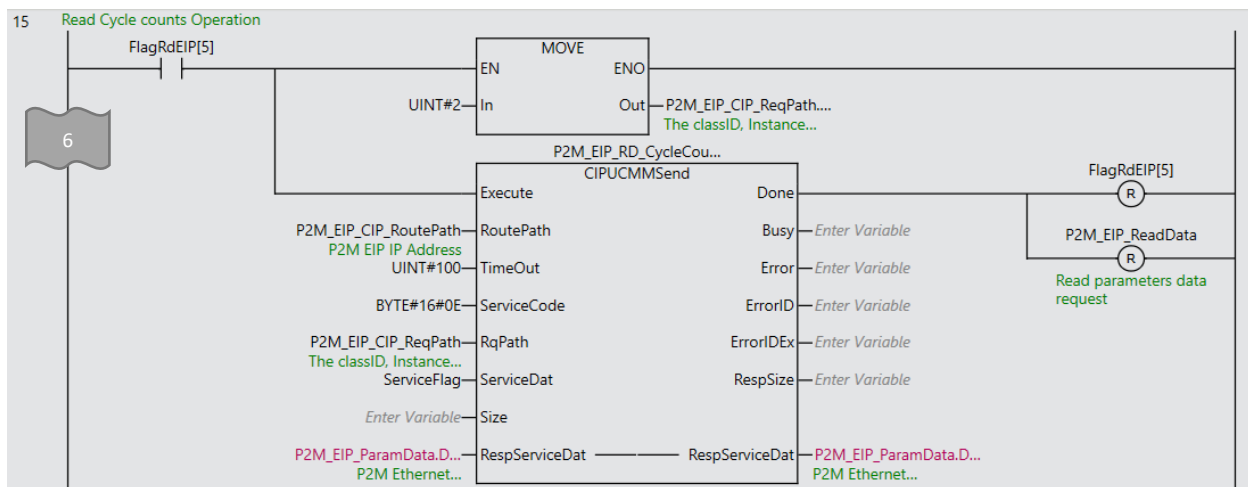
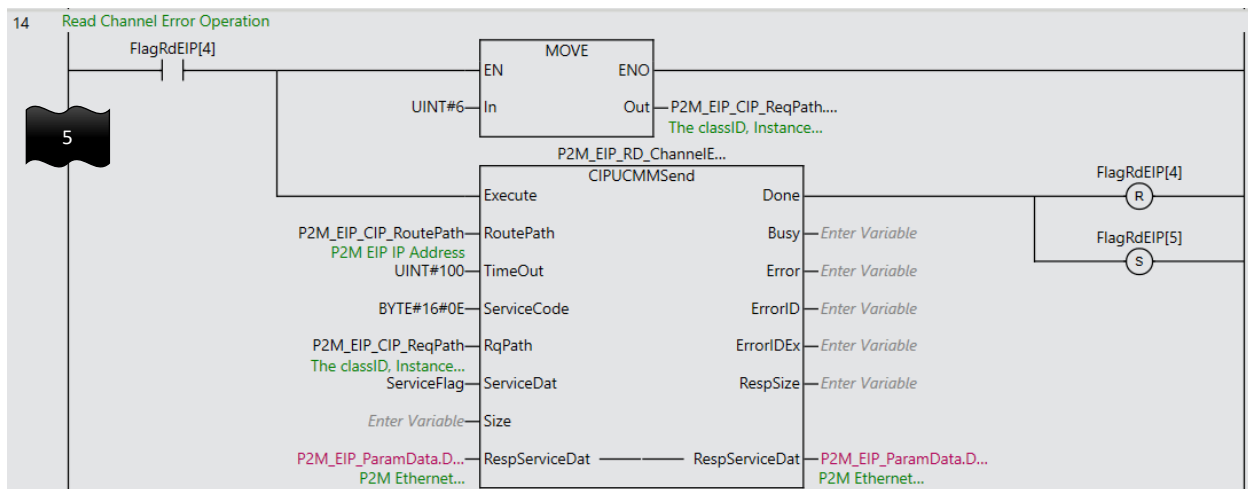
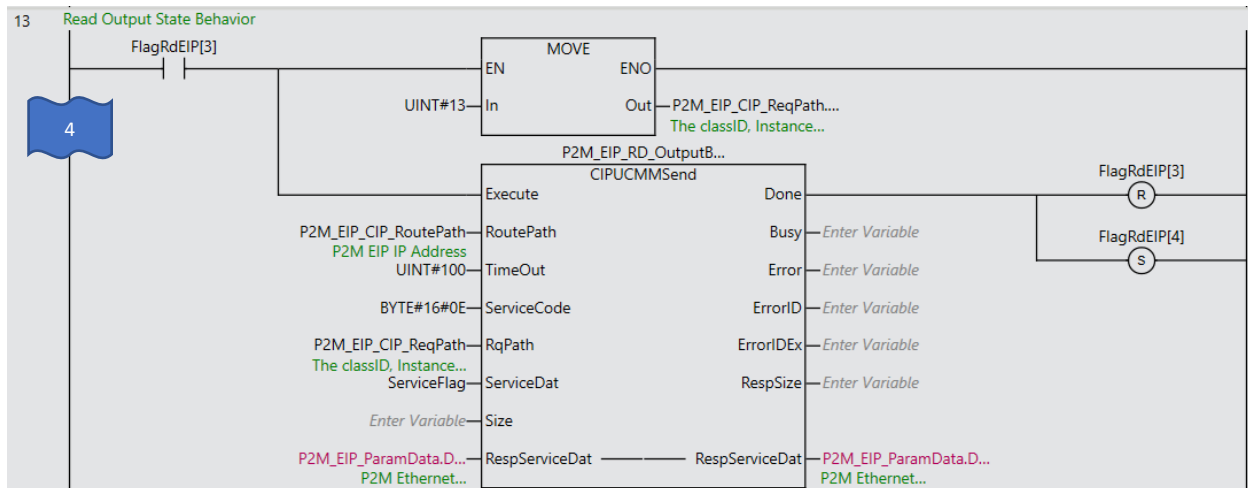
**Note\*\*Class = 100, Attribute = 5**



### 3.3.2. Reading Parameters.

see below for logic example as well as FB configuration for each parameter. You may rename the Function Block Control tags to match your tag naming convention. **Note\*\*Service Code = 0E.**

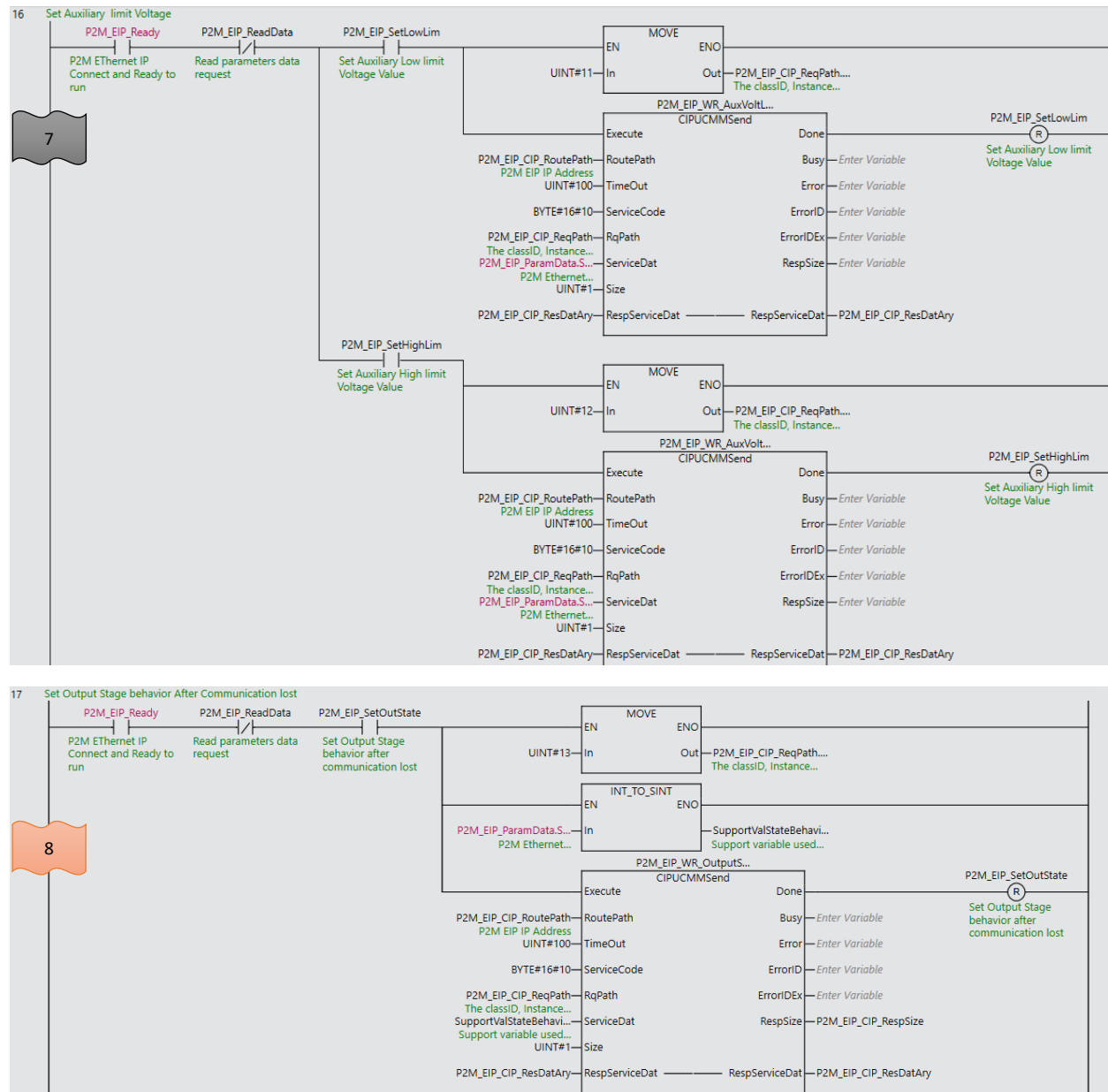


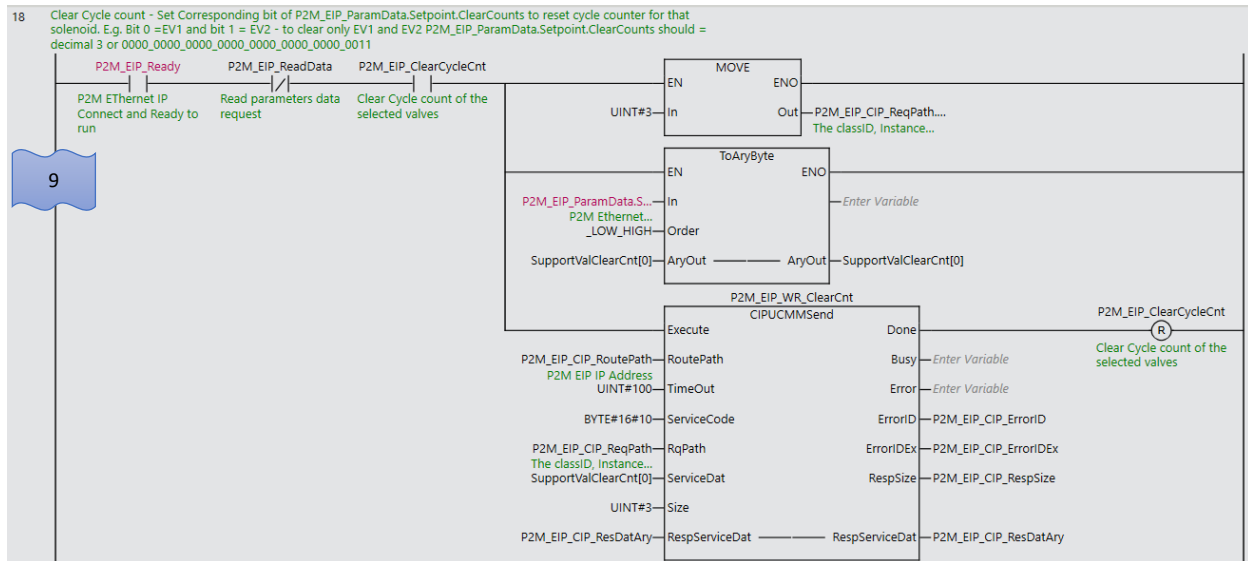


	Description	Instance	Destination Element
1	Read Auxiliary Voltage	4	P2M_EIP_ParamData.Data.Data.AuxVolt
2	Read Auxiliary Voltage Low Limit Warning	11	P2M_EIP_ParamData.Data.AuxVoltLoLim
3	Read Auxiliary Voltage High Limit Warning	12	P2M_EIP_ParamData.Data.AuxVoltHiLim
4	Read Output State Behavior	13	P2M_EIP_ParamData.Data.OutputStateBehavior
5	Read Channel Errors	6	P2M_EIP_ParamData.Data.ChErrors
6	Read Cycle Counters	2	P2M_EIP_ParamData.Data.CycleCounters

### 3.3.3. Writing Parameters.

Writing Auxiliary Voltage Low and High Limits, Output state behavior and clearing cycle counters  
example. **Note\*\***: Service Code = 10





	Description	Instance	Source Element	Source Length (bytes)
7	<b>Write Auxiliary Voltage Low Limit Warning</b>	11	P2M_EIP_ParamData.Setpoint.AuxVoltageLoLimWarning	2
	<b>Write Auxiliary Voltage High Limit Warning</b>	12	P2M_EIP_ParamData.Setpoint.AuxVoltageHiLimWarning	2
8	<b>Write Output State Behavior</b>	13	P2M_EIP_ParamData.Setpoint.OutputStateBehaviorSet	1
9	<b>Clear Cycle Counters</b>	3	P2M_EIP_ParamData.Setpoint.ClearCounts	3



## APPENDIX

### Process Data Structures

User Defined Data Structures utilized by FB “P2M2HBVE\_PD”

▼	P2M2HBV_ProcessData	STRUCT	NJ
	I	P2M_ProcessInputs	
	O	P2M_ProcessOutputs	

▼	P2M_ProcessInputs	STRUCT	NJ
	ChannelError	P2M_ChannelError	
	ChannelErrorFlagSet	BOOL	
	ModuleError	P2M_ModuleError	
	ModuleErrorFlagSet	BOOL	
	ModuleInfoFlags	P2M_ModuleInfosFlags	
	ModuleInfoFlagSet	BOOL	

▼	P2M_ChannelError	STRUCT	NJ
	CH	ARRAY[1..24] OF BOOL	

▼	P2M_ModuleError	STRUCT	NJ
	AckRequired	BOOL	
	AuxVoltageWarning	BOOL	
	AuxVoltageError	BOOL	
	TemperatureWarning	BOOL	
	OutputDriverChannelError	BOOL	
	ModuleError	BOOL	
	OutputStageNotAvailable	BOOL	

▼	P2M_ModuleInfosFlags	STRUCT	NJ
	WatchdogValveMicro	BOOL	
	EEPROM_Error	BOOL	
	WatchdogComMicro	BOOL	
	HeartbeatNotToggling	BOOL	
	HeartbeatState	BOOL	

▼	P2M_SystemCommand	STRUCT	NJ
	LeaveFailsafeState	BOOL	
	StoreCycleCounts	BOOL	

▼	P2M_SolenoidCommand	STRUCT	NJ
	EV	ARRAY[1..24] OF BOOL	

## Parameter Data Structures

User Defined Data Structures utilized in example code

▼	P2M2HBV_Parameters	STRUCT	NJ
	Data	P2M_Data	
	Setpoint	P2M_Setpoint	

▼	P2M_Data	STRUCT	NJ
	AuxVolt	INT	
	AuxVoltLoLim	INT	
	AuxVoltHiLim	INT	
	OutputStateBehavior	INT	
	ChErrors	DINT	
	CycleCounters	ARRAY[1..24] OF DINT	

▼	P2M_Setpoint	STRUCT	NJ
	AuxVoltageLoLimWarning	INT	
	AuxVoltageHiLimWarning	INT	
	OutputStateBehaviorSet	INT	
	ClearCounts	DINT	