



P2M2HBVT MODUFLEX  
ETHERCAT  
FUNCTION BLOCK FOR  
OMRON PLC  
QUICK START GUIDE

## CONTENTS

PREFACE .....	3
1. P2M EtherCAT device Configuration procedure On sysmac.....	4
2. Importing Function Block.....	8
3. Process Data Function Block Instruction.....	9
3.1. Add the function block “P2M2HBVT_PD” instance into the project .....	9
3.1. Using the instruction, Example of code .....	10
3.1.1. Solenoid Valves Control (Cyclic).....	11
3.1.2. Writing System Commands (Cyclic) .....	11
3.1.3. Monitoring Status Bits (Cyclic) .....	12
4. Parameter Data Example Code .....	13
4.1. Add the function block “P2M2HBVT_PRM” instance into the project .....	13
4.2. Parameter Code Instruction.....	14
4.2.1. Reading Parameters.....	14
4.2.2. Writing Parameters.....	16
APPENDIX.....	18

## PREFACE

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

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

The "P2M2HBVT\_PD" / "P2M2HBVT\_PRM" FB's facilitates communication and handling of process / parameters data between PLC and the Parker coupler EtherCAT slave device.

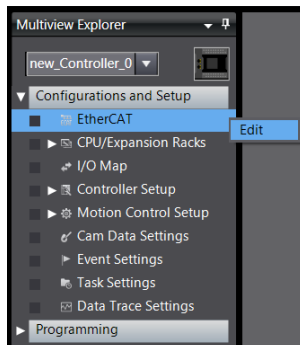
You can download resources such as the ESI configuration file, this QSG, the "P2M2HBVT\_PD" and "P2M2HBVT\_PRM" Function Block's, a sample SYSMAC program "P2M2HBVT\_OMRON\_NX\_R0" and the full P2M manual here:

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

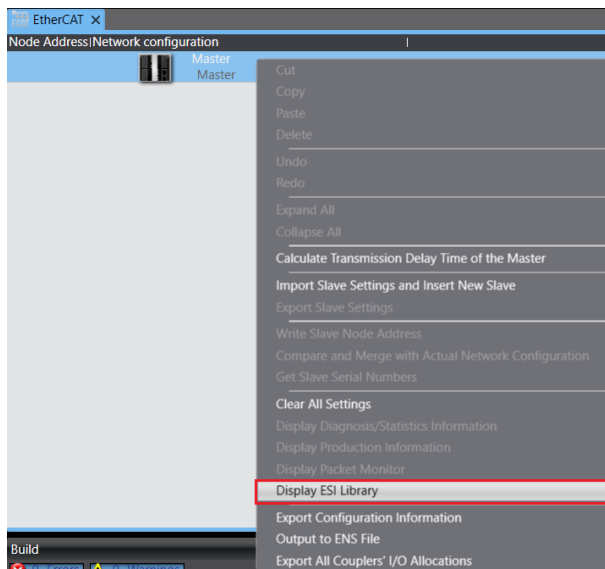
## 1. P2M ETHERCAT DEVICE CONFIGURATION PROCEDURE ON SYSMAC

### a. P2M File description installation

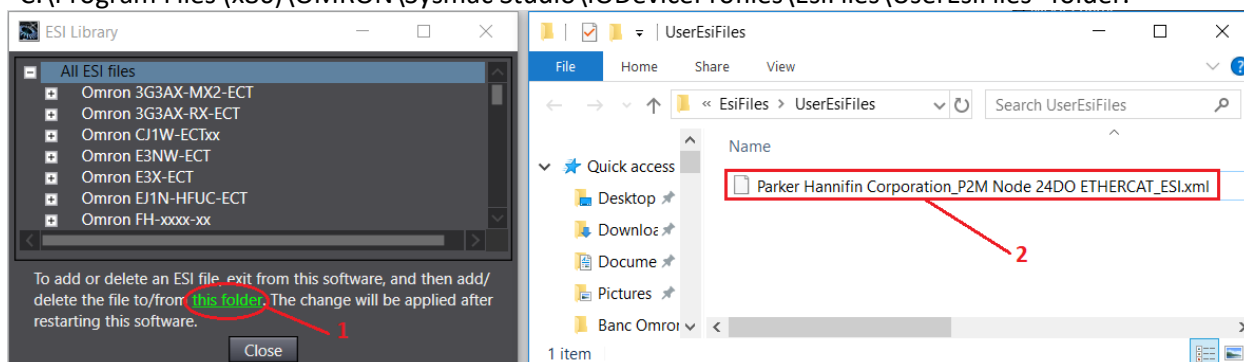
In the multiView Explorer of your created project click on “Configurations and Setup” and double click “EtherCAT” or Right click on “EtherCAT”>> Edit to open the “Node Address|Network configuration” view.



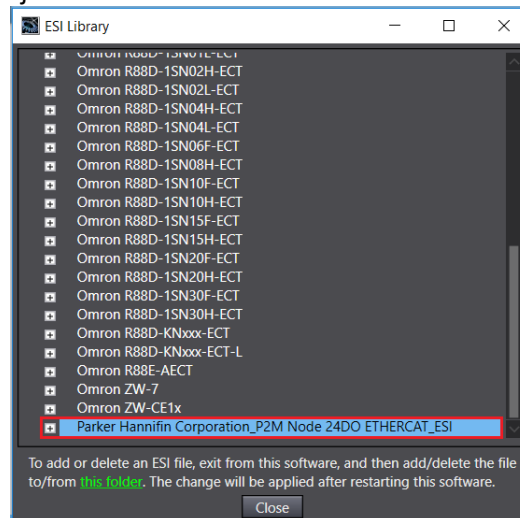
In the “Node Address|Network configuration” interface right click “Master” and select “display ESI Library”.



In the “ESI Library” Tab click on “this folder” to add P2M ESI file. Drag and drop the P2M ESI file from your stored folder to the opened “C:\Program Files (x86)\OMRON\Sysmac Studio\IODeviceProfiles\EsiFiles\UserEsiFiles” folder.

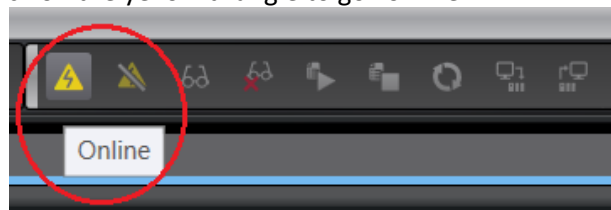


Close the opened library folder. Close and open Sysmac studio, follow the instruction above to open “ESI Library” Tab and view the just installed slave.



b. Add P2M device to the created project

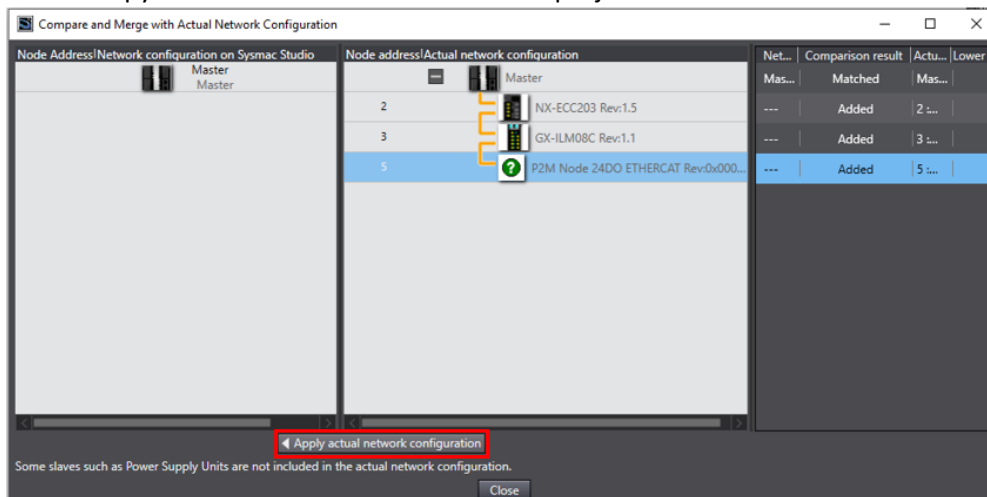
In the tool menu Tab click on the yellow triangle to go “online”



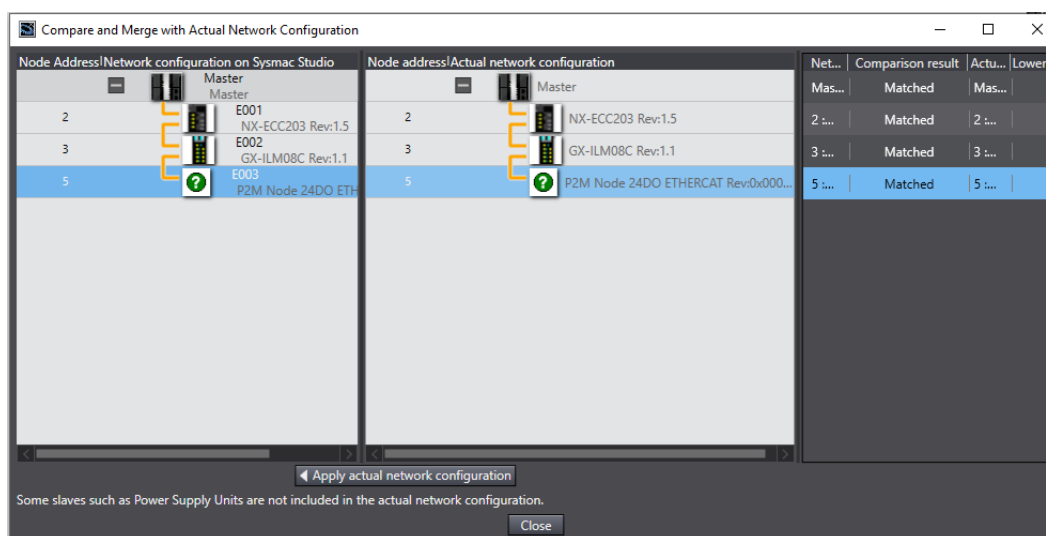
In EtherCAT “Node Address|Network configuration” interface right click on “Master” in the drop-down menu click on **“Compare and Merge with Actual Network Configuration”**



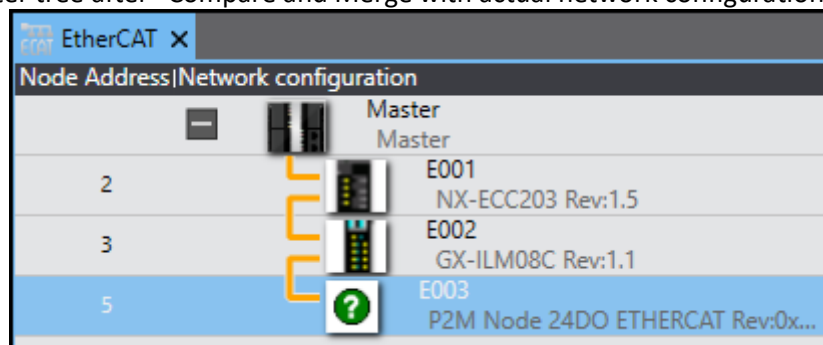
In the “Compare and Merge with Actual Network Configuration” tab click on **“Apply actual network configuration”** to copy the online device into the offline project.



View of the “Apply actual network configuration” function result, click on close to turn back in the “Node Address|Network configuration” interface.

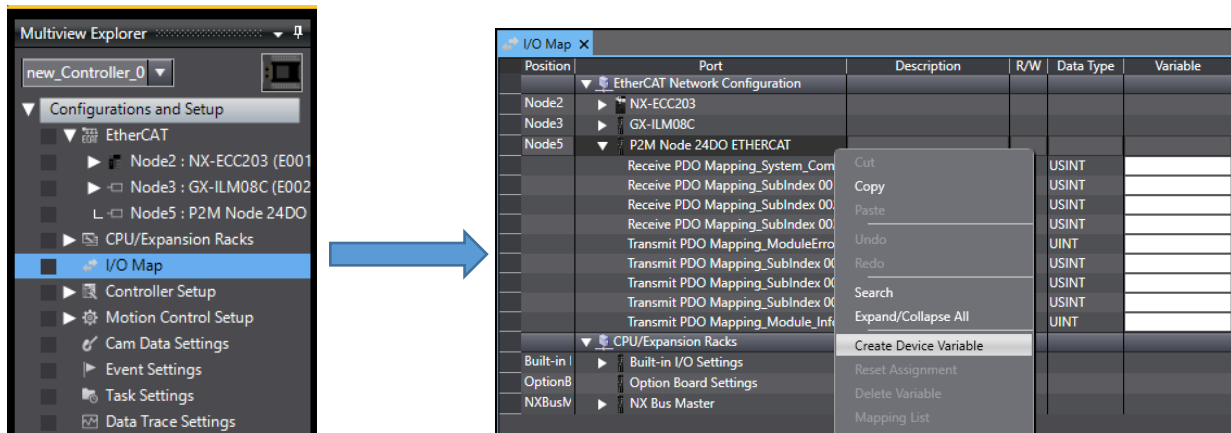


View of the Master tree after “Compare and Merge with actual network configuration”



### c. I/O Map

Open I/O Map to create input and output variable connected to the device, double click I/O map in the Configurations and setup drop-down menu. Right click P2M etherCAT coupler and click on “Create Device Variables”.

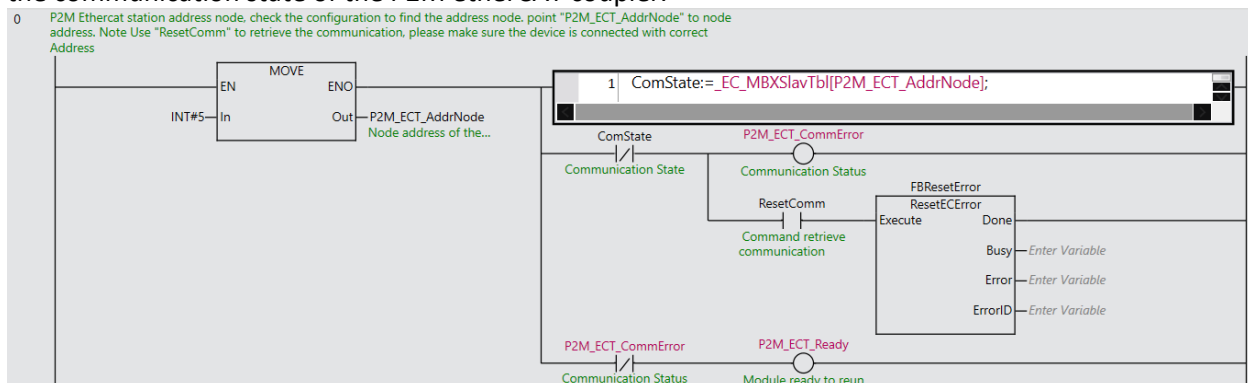


View of newly created variables.

Position	Port	Description	R/W	Data Type	Variable	Variable Comment	Variable Type
EtherCAT Network Configuration							
Node2	NX-ECC203						
Node3	GX-ILM08C						
Node5	P2M Node 24DO ETHERCAT						
	Receive PDO Mapping_System_Comm		W	USINT	E003_Receive_PDO_Mapping_System_Commands_2008_00		Global Variables
	Receive PDO Mapping_SubIndex 001		W	USINT	E003_Receive_PDO_Mapping_SubIndex_001_2001_01		Global Variables
	Receive PDO Mapping_SubIndex 002		W	USINT	E003_Receive_PDO_Mapping_SubIndex_002_2001_02		Global Variables
	Receive PDO Mapping_SubIndex 003		W	USINT	E003_Receive_PDO_Mapping_SubIndex_003_2001_03		Global Variables
	Transmit PDO Mapping_ModuleError		R	UINT	E003_Transmit_PDO_Mapping_ModuleErrorInput_2009_00		Global Variables
	Transmit PDO Mapping_SubIndex 001		R	USINT	E003_Transmit_PDO_Mapping_SubIndex_001_2006_01		Global Variables
	Transmit PDO Mapping_SubIndex 002		R	USINT	E003_Transmit_PDO_Mapping_SubIndex_002_2006_02		Global Variables
	Transmit PDO Mapping_SubIndex 003		R	USINT	E003_Transmit_PDO_Mapping_SubIndex_003_2006_03		Global Variables
	Transmit PDO Mapping_Module_Info		R	UINT	E003_Transmit_PDO_Mapping_Module_Info_Flags_2007_00		Global Variables
CPU/Expansion Racks							
Built-in	Built-in I/O Settings						
Option8	Option Board Settings						
NXBusV	NX Bus Master						

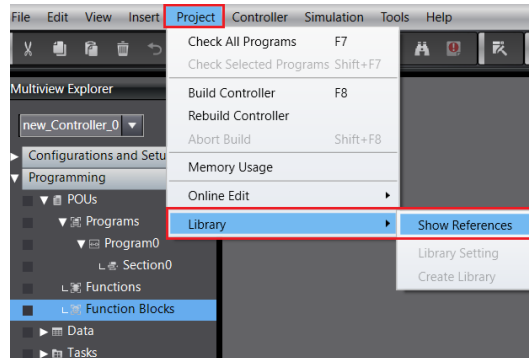
### d. Communication state read

Use the message communication enabled slave table “\_EC\_MBXSlaVtBl” from Sysmac library to view the communication state of the P2M etherCAT coupler.

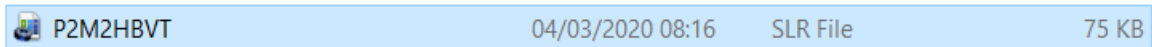


## 2. IMPORTING FUNCTION BLOCK

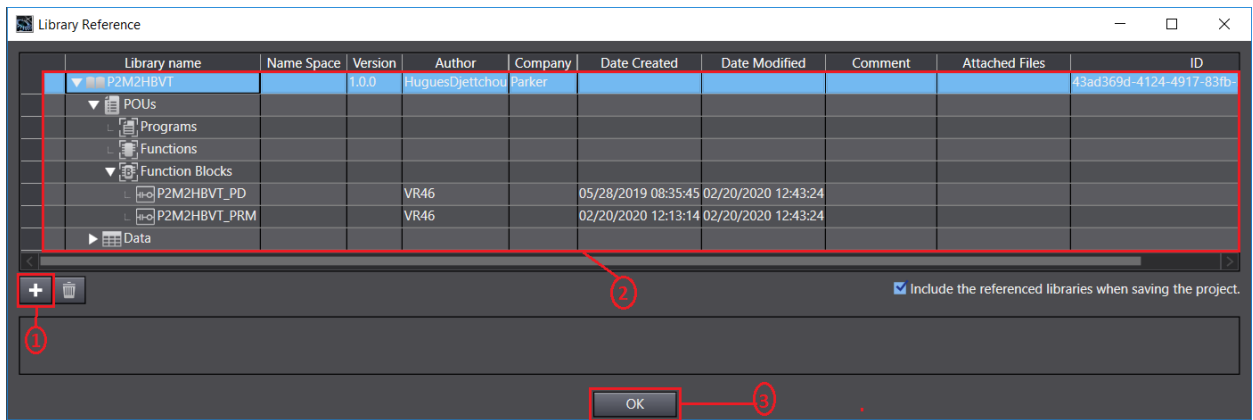
In the menu bar click on “Project>> Library>> Show References”



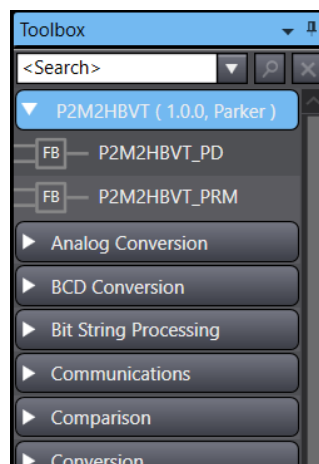
In the Reference Library Tab interface click (+), move to the folder where the file Library



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



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



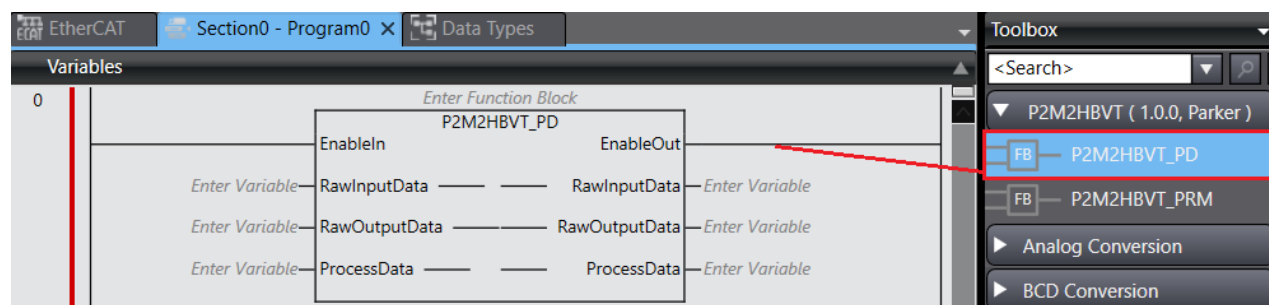


### 3. PROCESS DATA FUNCTION BLOCK INSTRUCTION

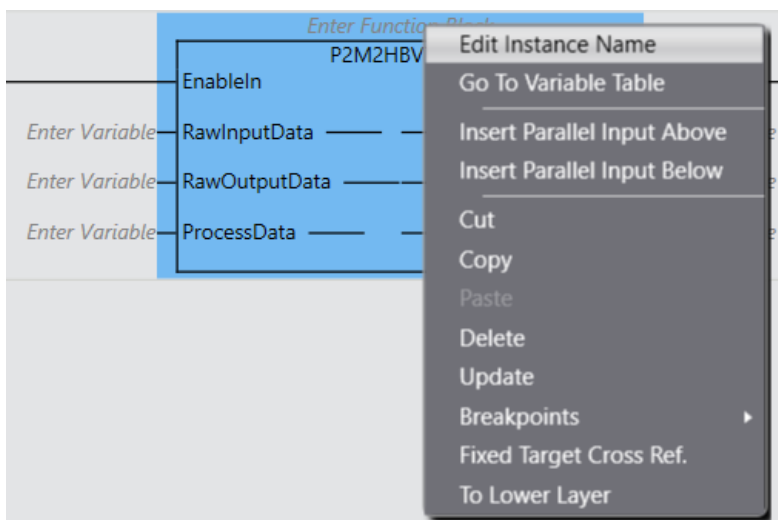
The “P2M2HBVT\_PD” Function Block simplifies the usage of Parker EtherCAT coupler P2M devices with Omron NX and NJ PLCs when connected via EtherCAT. Data is mapped to user-friendly control and diagnostic tags on the PLC side.

#### 3.1. Add the function block “P2M2HBVT\_PD” instance into the project

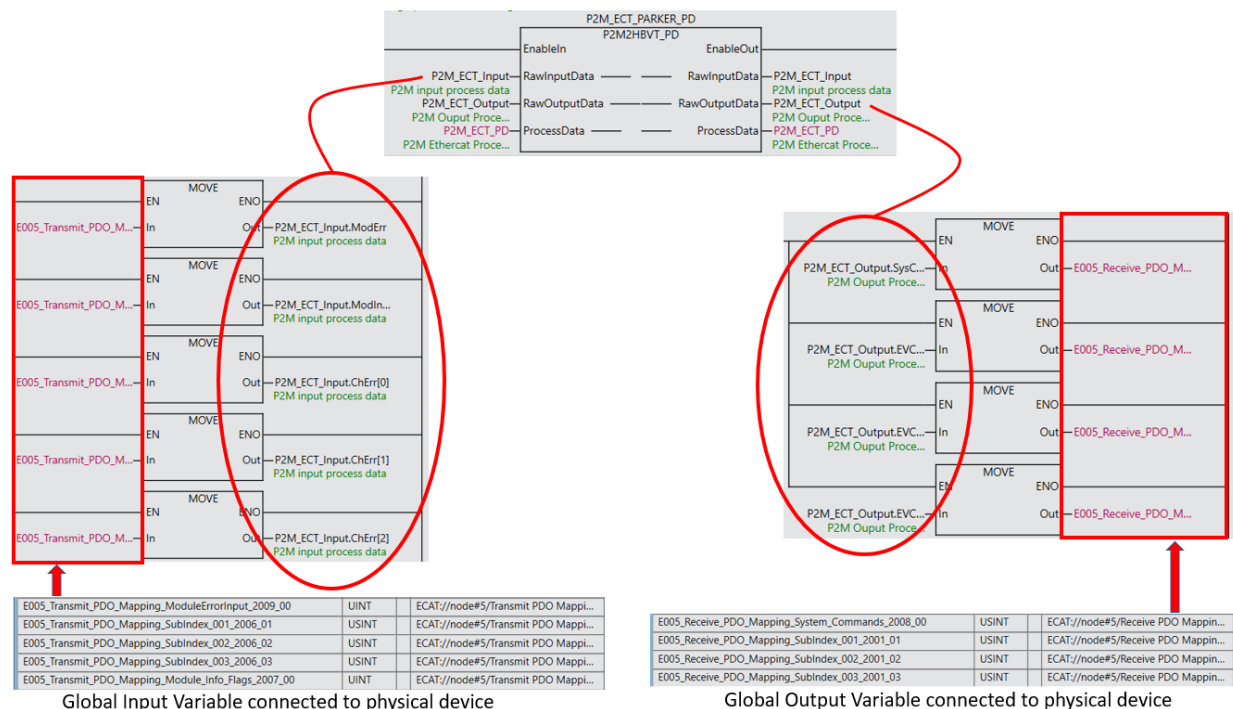
Drag and drop the function block “P2M2HBVT\_PD” from toolbox tab to an empty ladder rung of the program section interface.



Assign an instance name for the FB and create other tags necessary for operation. Right click on “Enter Function Block” marks and select “Edit Instance Name” or “Edit” for the variable when right clicking on “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.



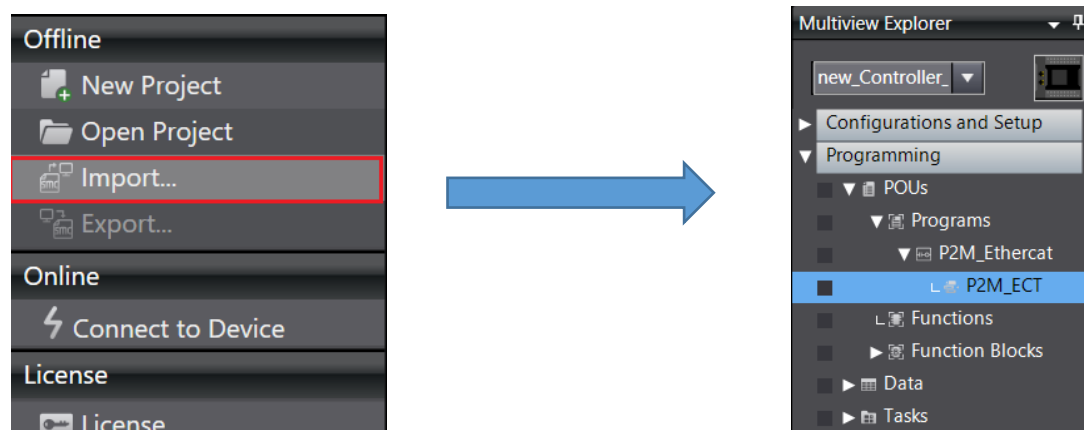
The RawInputData and the RawOutputData should be pointed to the physical input and output variable connected to the P2M device target. See Appendix for structure breakdown of the “Data” variable.



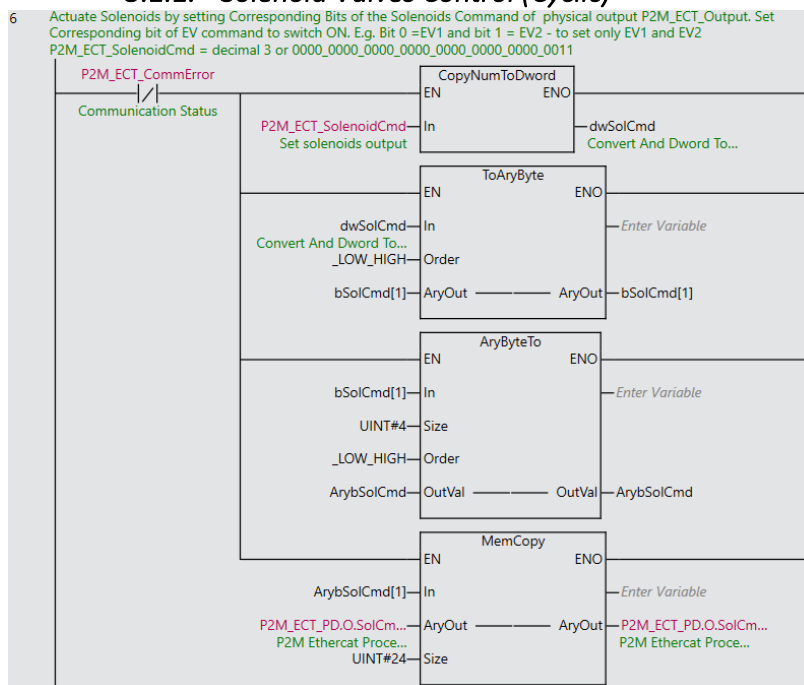
### 3.1. Using the instruction, Example of code

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\_ECT\_PD.I.xxx and P2M\_ECT\_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.

Launch “Sysmac studio” then click on Import, open the folder where the sample code “P2M2HBVT\_OMRON\_NX\_R0” is stored, select the file .smc2 and click open



### 3.1.1. Solenoid Valves Control (Cyclic)



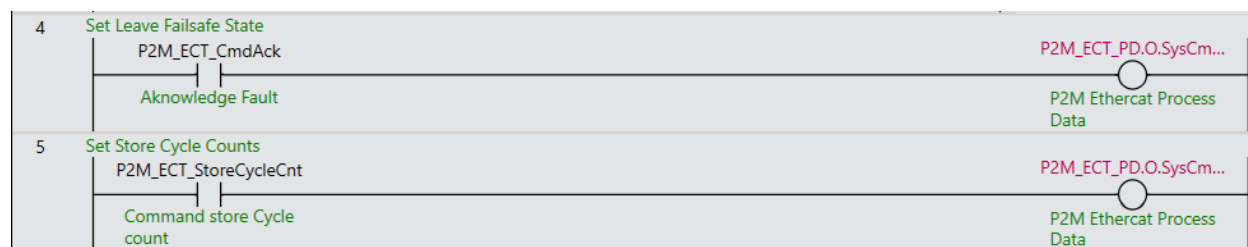
**Value correlation to Solenoid Address**

P2M_ECT_SolenoidCmd = 1	"P2M_ECT_PD".O.SolCmd.EV[1]=1
P2M_ECT_SolenoidCmd = 2	"P2M_ECT_PD".O.SolCmd.EV[2]=1
P2M_ECT_SolenoidCmd = 4	"P2M_ECT_PD".O.SolCmd.EV[3]=1
P2M_ECT_SolenoidCmd = 8	"P2M_ECT_PD".O.SolCmd.EV[4]=1
P2M_ECT_SolenoidCmd = 16	"P2M_ECT_PD".O.SolCmd.EV[5]=1
.	
.	
.	
P2M_ECT_SolenoidCmd = 8 388 608	"P2M_ECT_PD".O.SolCmd.EV[24]=1

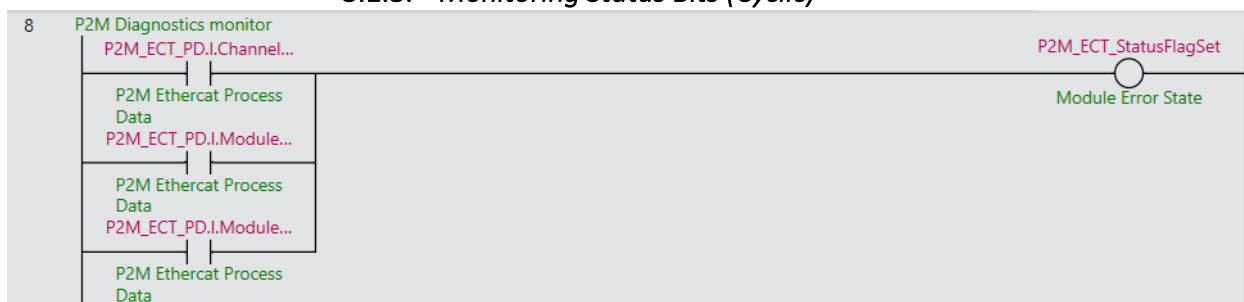
### 3.1.2. Writing System Commands (Cyclic)

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.



### 3.1.3. Monitoring Status Bits (Cyclic)

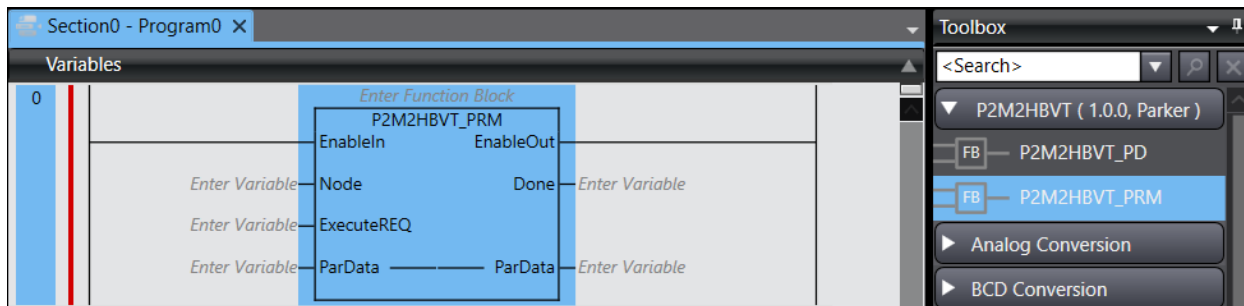


If any of the status bits are set, further information can be attained by evaluating specific diagnostic bits.

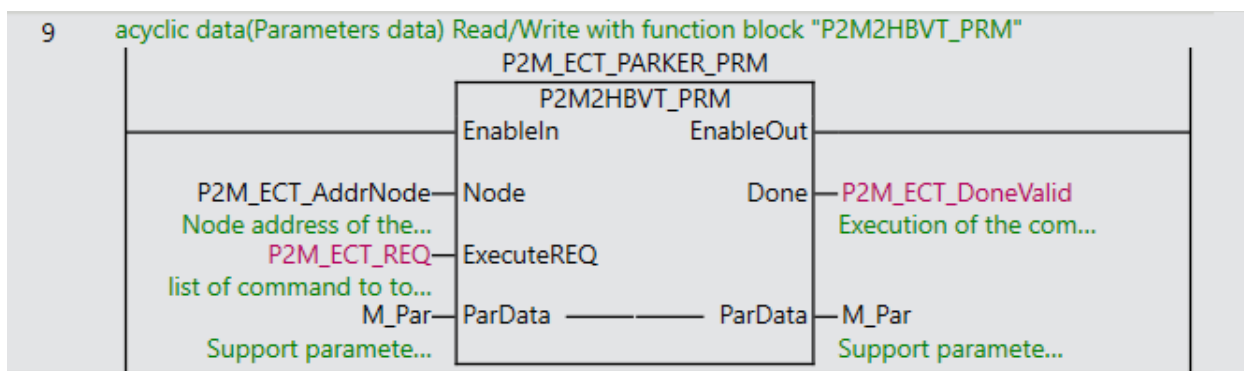
Channel Error	Module Error	Module info Flags
"P2M_ECT_PD".I.ChannelError.CH[1]	"P2M_ECT_PD".I.ModuleError.AckRequired	"P2M_ECT_PD".I.ModuleInfoFlags.WatchdogValveMicro
"P2M_ECT_PD".I.ChannelError.CH[2]	"P2M_ECT_PD".I.ModuleError.AuxVoltageWarning	"P2M_ECT_PD".I.ModuleInfoFlags.EEPROM_Error
"P2M_ECT_PD".I.ChannelError.CH[3]	"P2M_ECT_PD".I.ModuleError.AuxVoltageError	"P2M_ECT_PD".I.ModuleInfoFlags.WatchdogComMicro
"P2M_ECT_PD".I.ChannelError.CH[4]	"P2M_ECT_PD".I.ModuleError.TemperatureWarning	"P2M_ECT_PD".I.ModuleInfoFlags.HeartbeatNotToggling
"P2M_ECT_PD".I.ChannelError.CH[5]	"P2M_ECT_PD".I.ModuleError.OutputDriverChannelError	"P2M_ECT_PD".I.ModuleInfoFlags.HeartbeatState
"P2M_ECT_PD".I.ChannelError.CH[6]	"P2M_ECT_PD".I.ModuleError.ModuleError	
"P2M_ECT_PD".I.ChannelError.CH[7]	"P2M_ECT_PD".I.ModuleError.OutputStageNotAvailable	
"P2M_ECT_PD".I.ChannelError.CH[8]		
"P2M_ECT_PD".I.ChannelError.CH[9]		
"P2M_ECT_PD".I.ChannelError.CH[10]		
"P2M_ECT_PD".I.ChannelError.CH[11]		
"P2M_ECT_PD".I.ChannelError.CH[12]		
"P2M_ECT_PD".I.ChannelError.CH[13]		
"P2M_ECT_PD".I.ChannelError.CH[14]		
"P2M_ECT_PD".I.ChannelError.CH[15]		
"P2M_ECT_PD".I.ChannelError.CH[16]		
"P2M_ECT_PD".I.ChannelError.CH[17]		
"P2M_ECT_PD".I.ChannelError.CH[18]		
"P2M_ECT_PD".I.ChannelError.CH[19]		
"P2M_ECT_PD".I.ChannelError.CH[20]		
"P2M_ECT_PD".I.ChannelError.CH[21]		
"P2M_ECT_PD".I.ChannelError.CH[22]		
"P2M_ECT_PD".I.ChannelError.CH[23]		
"P2M_ECT_PD".I.ChannelError.CH[24]		

## 4. PARAMETER DATA EXAMPLE CODE

4.1. Add the function block “P2M2HBVT\_PRM” instance into the project  
 Drag and drop the function block “P2M2HBVT\_PRM” from toolbox tab to an empty ladder rung of the program section interface.



Assign an instance name for the FB and create other tags necessary for operation. Right click on the “Enter Function Block” and select “Edit Instance Name” or “Edit” for the variable by right 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.



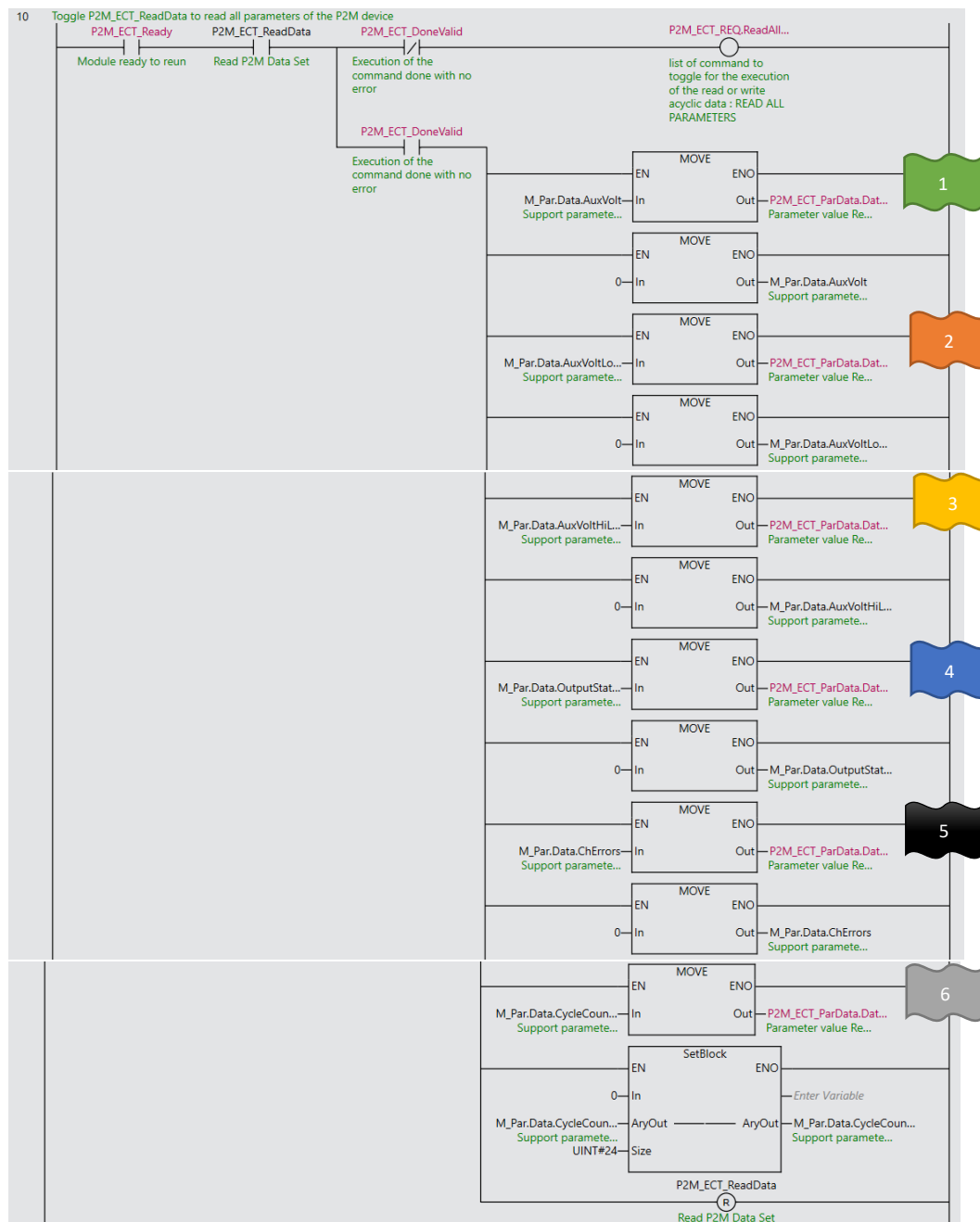
## 4.2. Parameter Code Instruction

(See 3.2. for importing example code)

**Note\*\*:** It is 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 done bit to go high before executing the next request.

### 4.2.1. Reading Parameters.

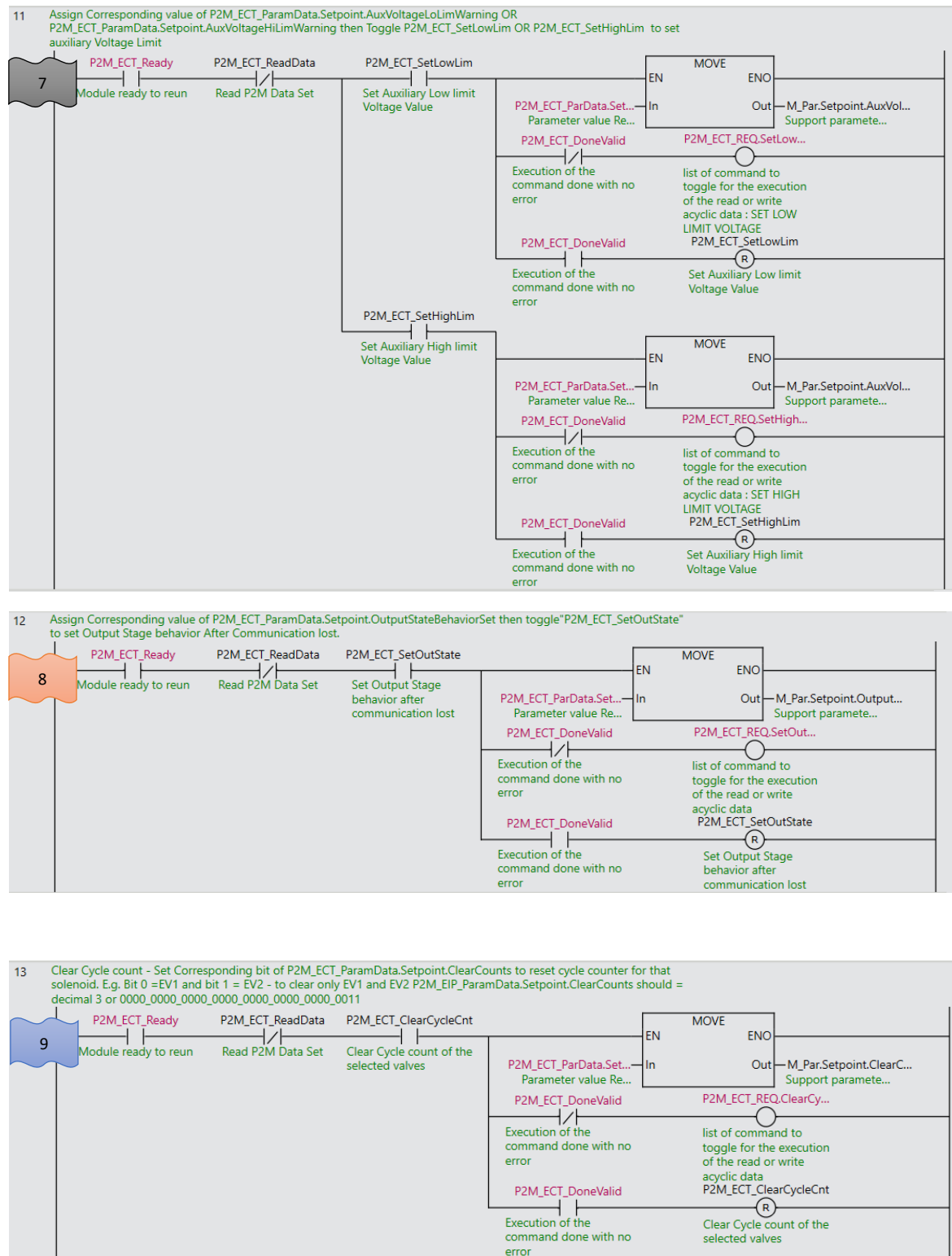
see below for logic example as well as message configuration for each parameter. You may rename the Function Block Control tags to match your tag naming convention.



	Description	Index	Destination Element
1	Read Auxiliary Voltage	2004	P2M_ECT_ParData.Data.Data.AuxVolt
2	Read Auxiliary Voltage Low Limit Warning	200B	P2M_ECT_ParData.Data.AuxVoltLoLim
3	Read Auxiliary Voltage High Limit Warning	200C	P2M_ECT_ParData.Data.AuxVoltHiLim
4	Read Output State Behavior	200D	P2M_ECT_ParData.Data.OutputStateBehavior
5	Read Channel Errors	2006	P2M_ECT_ParData.Data.ChErrors
6	Read Cycle Counters	2002	P2M_ECT_ParData.Data.CycleCounters

### 4.2.2. Writing Parameters.

Writing Auxiliary Voltage Low and High Limits, Output state behavior and clearing cycle counters example.





	Description	Index	Source Element	Source Length (bytes)
7	<b>Write Auxiliary Voltage Low Limit Warning</b>	200B	P2M_ECT_ParData.Setpoint.AuxVoltageLoLimWarning	2
	<b>Write Auxiliary Voltage High Limit Warning</b>	200C	P2M_ECT_ParData.Setpoint.AuxVoltageHiLimWarning	2
8	<b>Write Output State Behavior</b>	200D	P2M_ECT_ParData.Setpoint.OutputStateBehaviorSet	1
9	<b>Clear Cycle Counters</b>	2003	P2M_ECT_ParData.Setpoint.ClearCounts	3

## APPENDIX

### Process Data Structures

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

#### ➤ Raw in-out data structure

▼	RawInData	STRUCT	NJ
	ModErr	UINT	
	ChErr	ARRAY[0..2] OF USINT	
	ModInfos	UInt	

▼	RawOutData	STRUCT	NJ
	SysCmd	USINT	
	EVCmd	Array[0..2] OF USINT	

#### ➤ Device process in-out data structure

▼	P2M2HBV_ProcessData	STRUCT	NJ
	I	P2M_ProcessInputs	
	O	P2M_ProcessOutputs	

#### a) Process input structure

▼	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	

## b) Process output structure

▼	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	

## a) Read parameters data structure

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

## b) Write parameters data structure

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

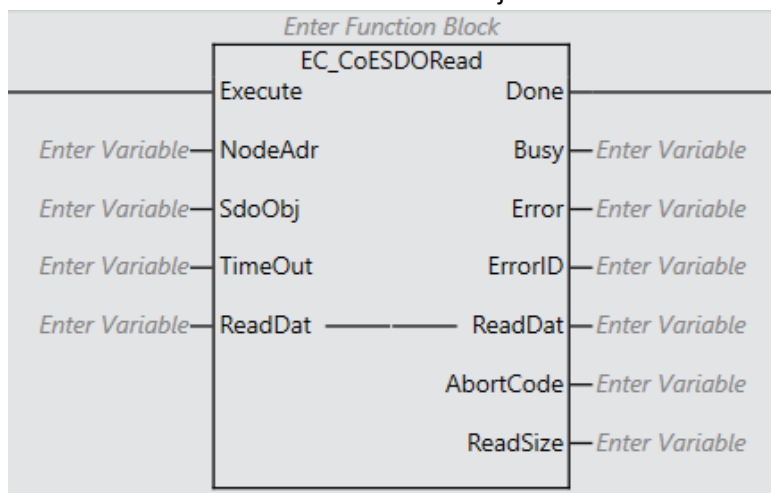
## c) List of command read/write structure

▼ P2M_List_Command	STRUCT	NJ
ReadAllPar	BOOL	
SetLowLimVolt	BOOL	
SetHighLimVolt	BOOL	
SetOutputState	BOOL	
ClearCycleCnt	BOOL	

Use Sysmac function blocks “EC\_CoESDORead”, “EC\_CoESDOWrite” to read and write acyclic data.

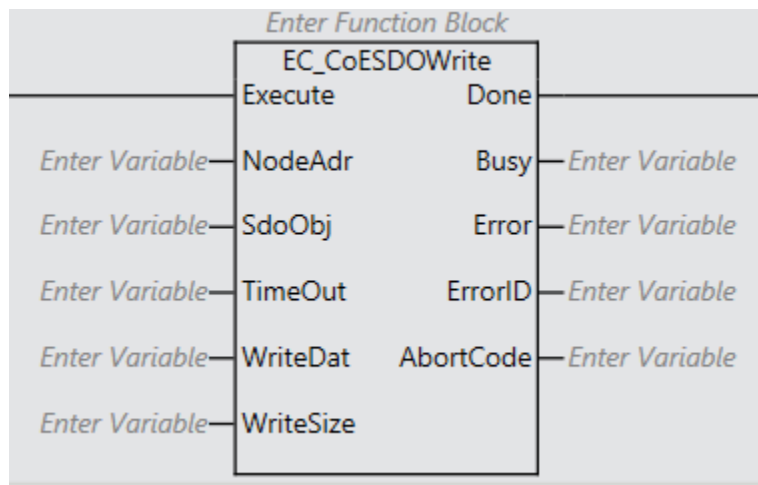
These function blocks can be found in the Sysmac library and can replace parker function block “P2M2HBVT\_PRM”.

- **EC\_CoESDORead** instruction reads a value from a CoE object of the Parker etherCAT coupler



Parameters	NodeAdr	SdoObj.Index	SdoObj.SubIndex	RD_SdoObj.IsCompleteAccess	ReadDat
Read Auxiliary Voltage	Node Address of your P2M EtherCAT coupler	2004	0	FALSE	target area for the data to be read
Read Auxiliary Voltage Low Limit Warning		200B	0	FALSE	
Read Auxiliary Voltage High Limit Warning		200C	0	FALSE	
Read Output State Behavior		200D	0	FALSE	
Read Channel Errors		2006	0	TRUE	
Read Cycle Counters		2002	0	TRUE	

- **EC\_CoESDOWrite** instruction writes a value to a CoE object of the Parker etherCAT coupler



Parameters	NodeAdr	SdoObj.Index	SdoObj.SubIndex	RD_SdoObj.IsCompleteAccess	WriteDat	WriteSize
Write Auxiliary Voltage Low Limit Warning	Node Address of your P2M EtherCAT coupler	200B	0	FALSE	Source data to be written	2
Write Auxiliary Voltage High Limit Warning		200C	0	FALSE		2
Write Output State Behavior		200D	0	FALSE		1
Clear Switching Cycle counters		2003	0	TRUE		5