



P2M2HBVW MODUFLEX  
POWERLINK  
FUNCTION BLOCK FOR  
B&R X20 PLC  
QUICK START GUIDE

## CONTENTS

PREFACE .....	3
1. P2M powerlink device Configuration procedure On automation studio .....	4
2. Importing Function Block .....	10
3. Process Data Function Block Instruction.....	12
3.1. Add the function block “P2M2HBVW_PD_FB” instance into the project .....	12
3.1. Using the instruction, Example of code “P2M2HBVW_BNR_X20_R0 .....	13
3.1.1. Solenoid Valves Control (Cyclic).....	14
3.1.2. Writing System Commands (Cyclic) .....	15
3.1.3. Monitoring Status Bits (Cyclic) .....	15
4. Parameter Data Example Code .....	16
4.1. Add the function block “P2M2HBVW_PRM_FB” instance into the project .....	16
4.2. Parameter Code Instruction.....	17
4.2.1. Reading Parameters.....	17
4.2.2. Writing Parameters.....	19
APPENDIX.....	22

## PREFACE

This Quick Start Guide (QSG) is designed to help integrate Parker Hannifin's P2M Powerlink valve manifold into a B&R 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 Powerlink device.

The "P2M2HBVW\_PD\_FB" / "P2M2HBVW\_PRM\_FB" FB's facilitates communication and handling of process / parameters data between PLC and the Parker coupler Powerlink slave device.

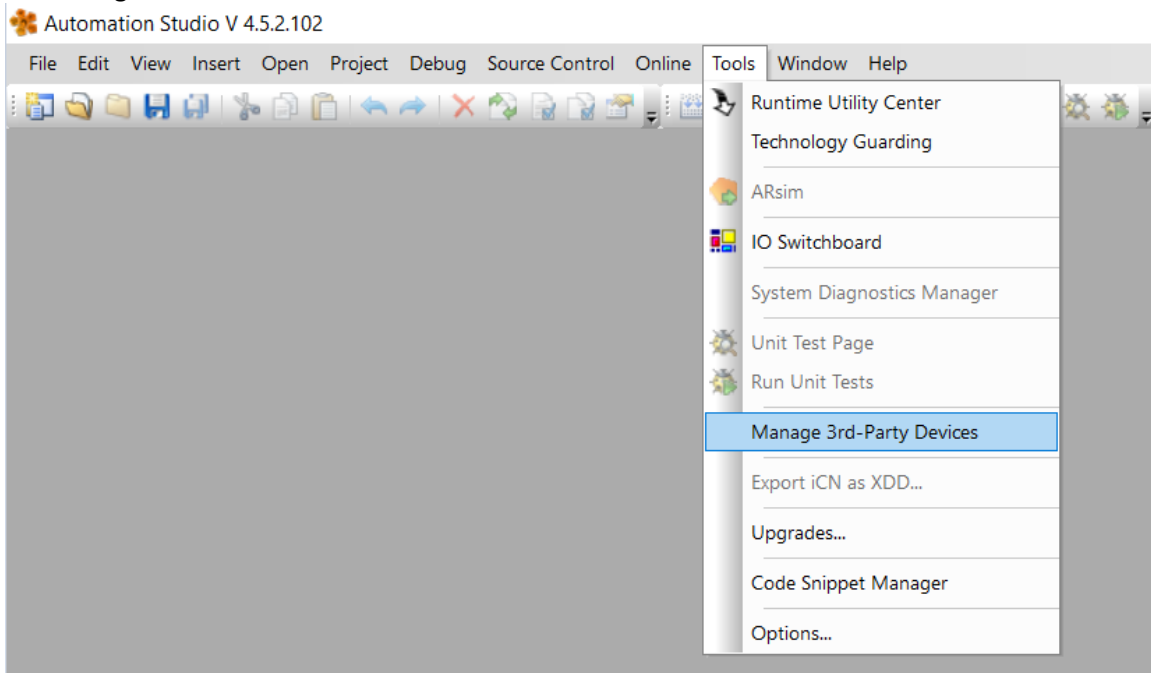
You can download resources such as the XDD configuration file, this QSG, the "P2M2HBVW\_BNR\_X20\_R0 Lib" Function Block's library, a sample Automation Studio program "P2M2HBVW\_BNR\_X20\_R0" and the full P2M manual here:

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

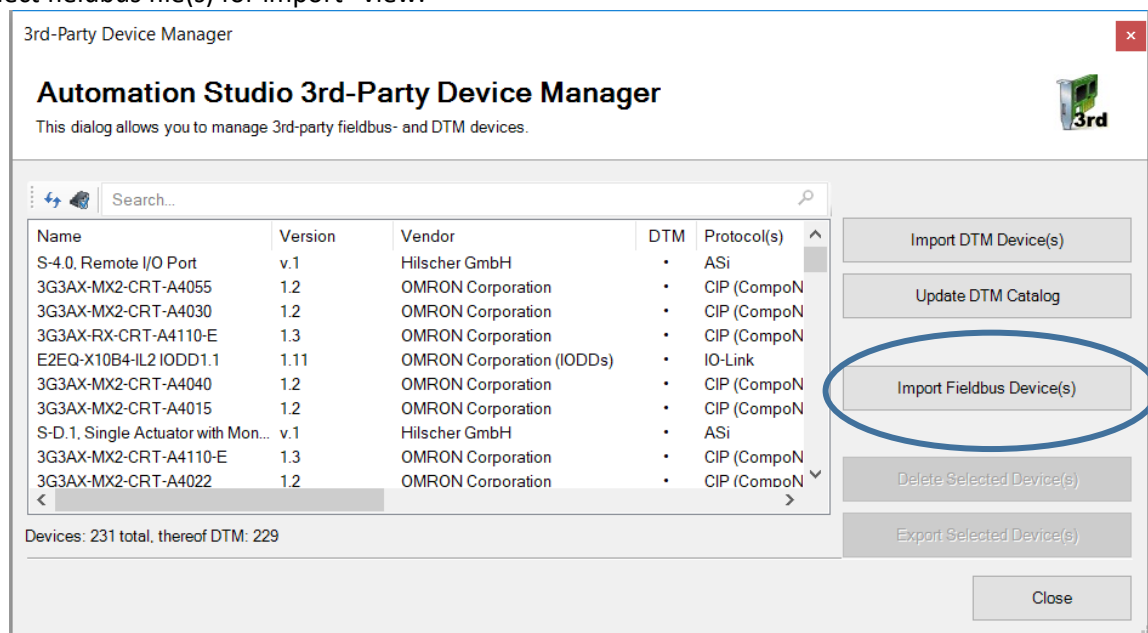
## 1. P2M POWERLINK DEVICE CONFIGURATION PROCEDURE ON AUTOMATION STUDIO

### a. P2M File description installation

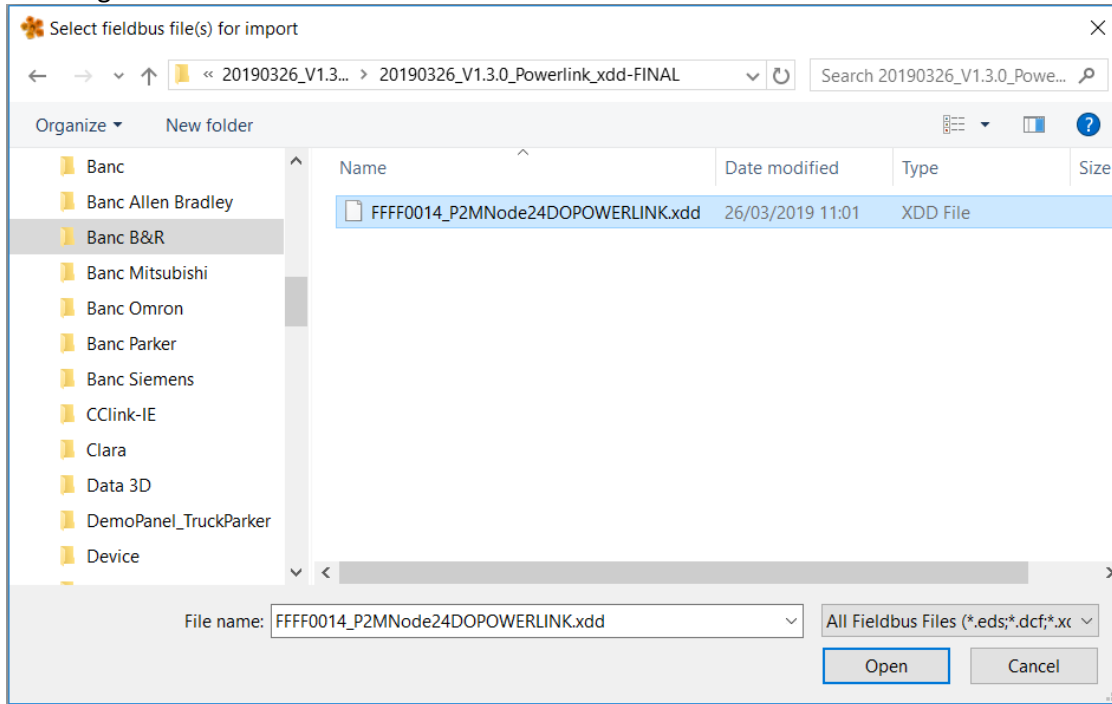
Click on “Tools > Manage 3<sup>rd</sup>-Party Devices” in the automation studio menu bar to open the 3<sup>rd</sup>-Party Device Manager.



In the “3<sup>rd</sup>-Party Device Manager” interface right click on “Import Fieldbus device(s)” to pop up the “Select fieldbus file(s) for import” view.



In the “Select fieldbus file(s) for import” view, move to the folder where P2M xdd file download from parker website is stored, select the xdd file then click “open” to install the device into automation studio device catalog.

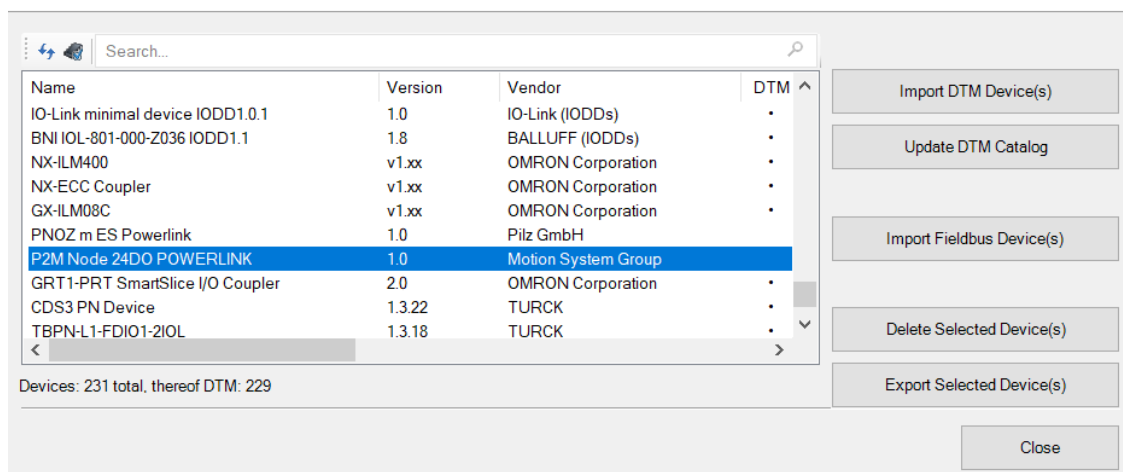


View the just installed slave in the automation studio device catalog, then click “close” to leave the “3<sup>rd</sup>-Party Device Manager.

3rd-Party Device Manager

### Automation Studio 3rd-Party Device Manager

This dialog allows you to manage 3rd-party fieldbus- and DTM devices.

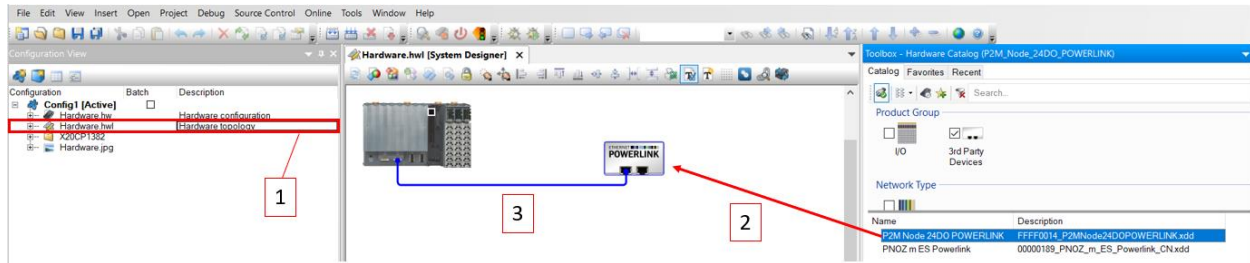


Now the device can be used into an automation studio project.

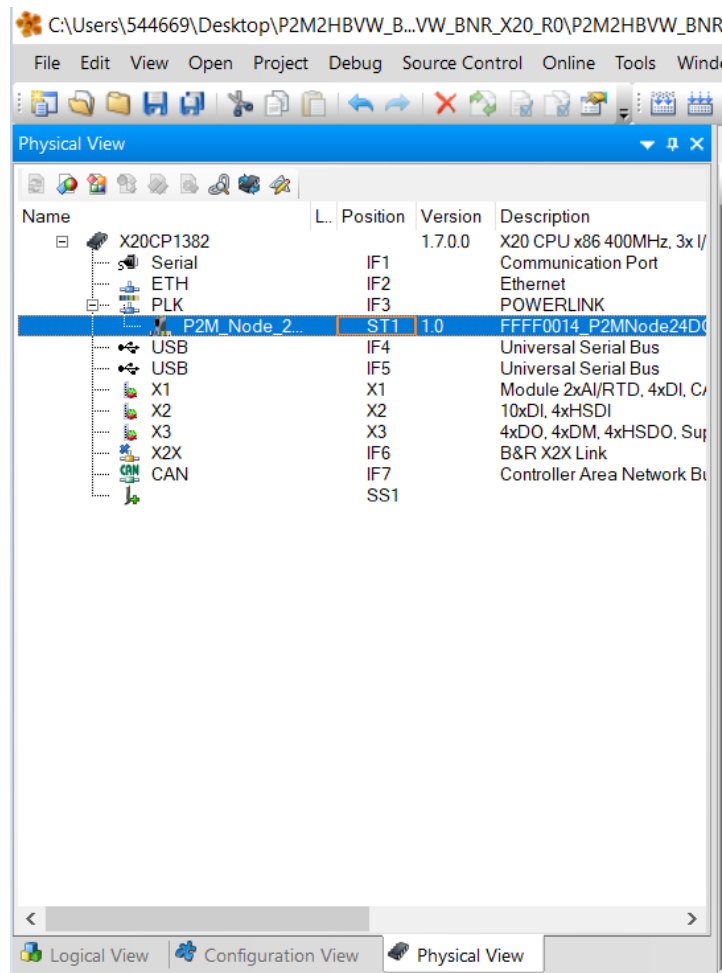
b. Add P2M device to your created project

QSG assumes that you are already using the Powerlink network interface with a B&R configured PLC into your project.

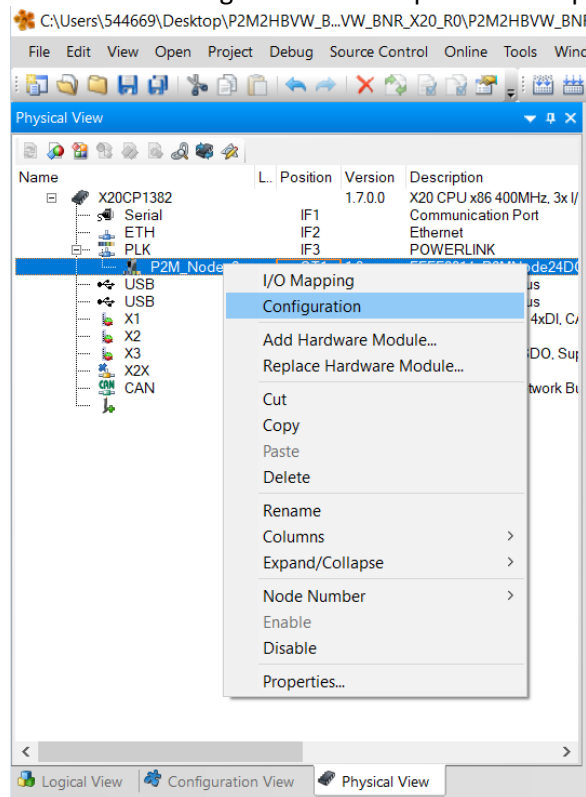
In the “Configuration View” (1) double click “Hardware.hwl” to open the Hardware.hwl[System designer] view. (2) Drag and drop the P2M Powerlink device from Toolbox – Object Catalog to Hardware.hwl[System designer], (3) connect the Powerlink port of the P2M device to the Powerlink port of the PLC.



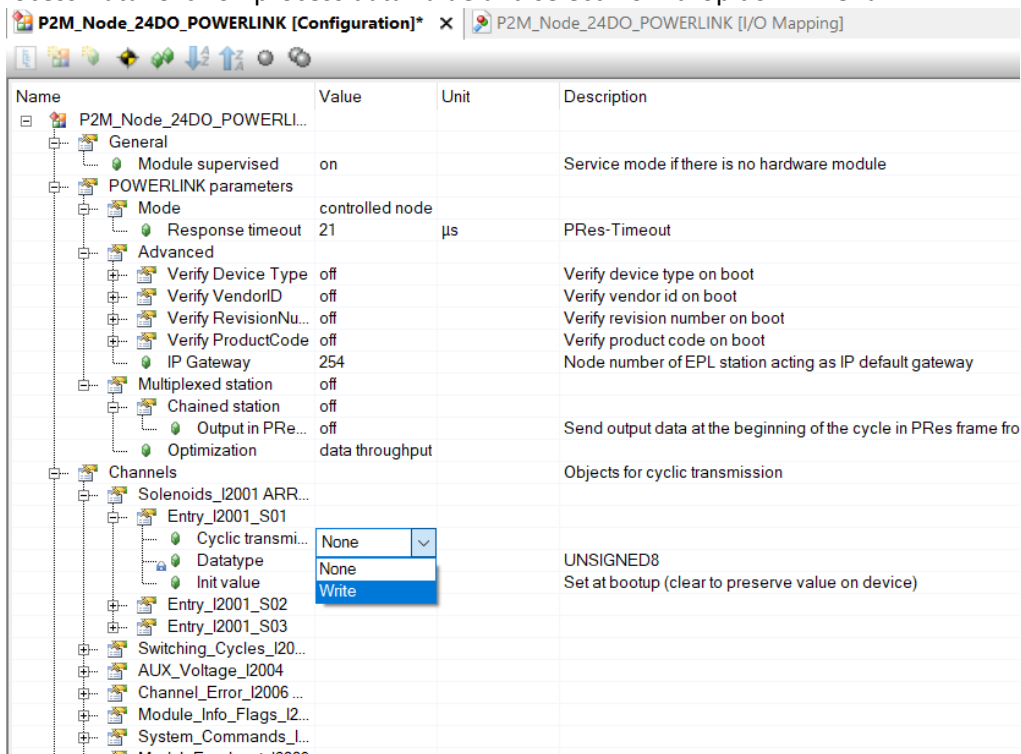
In the “Physical View” double click the “STx” to change the node address to match the address set on the physical device.



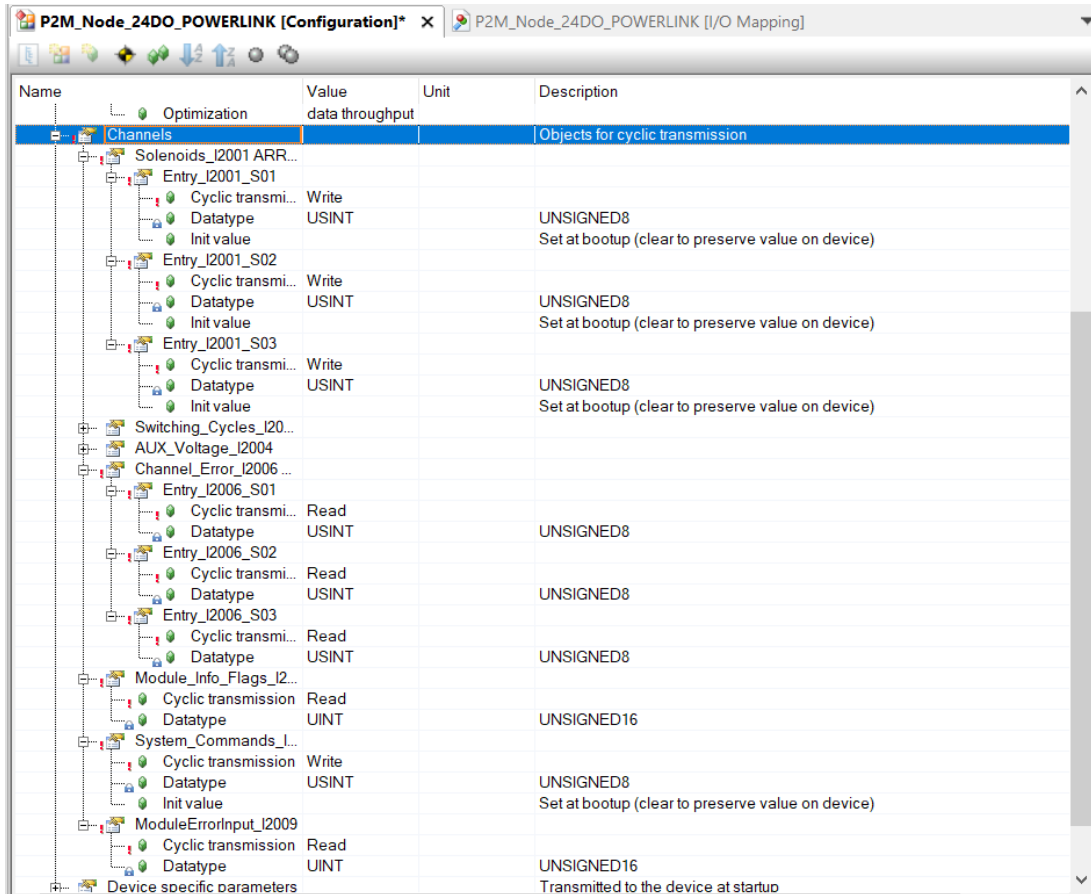
Right click the P2M device and click on “configuration” to set process data properties of the P2M



In configuration interface of the device click on channel and set the None/read/write properties of the viewing Process Data. Click on process data value and select from drop down menu.

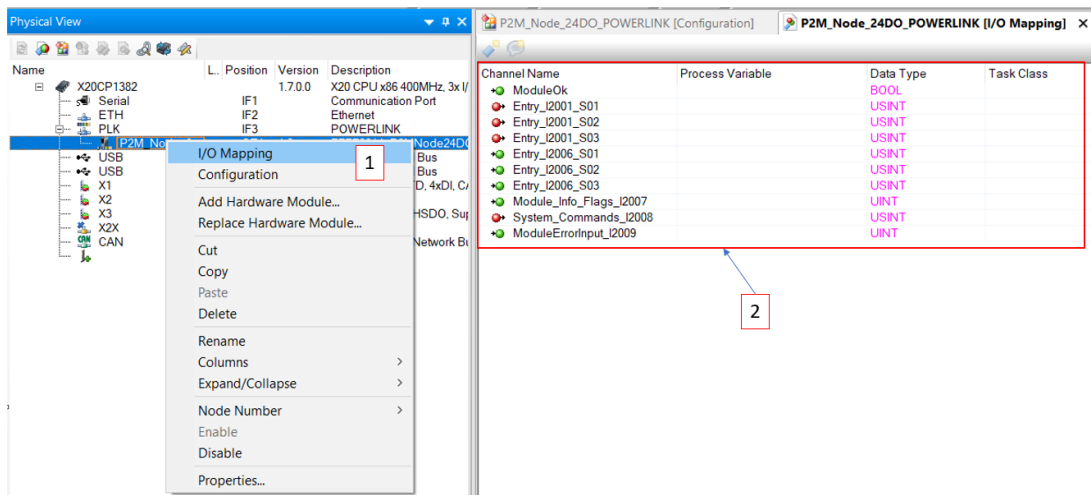


View below the list of the process data with the modified value, each modified value from default value has a red “exclamation” mark in front.



Name	Value	Unit	Description
Optimization	data throughput		
Channels			Objects for cyclic transmission
Solenoids_I2001_ARR...			
Entry_I2001_S01			
Cyclic transmi...	Write		
Datatype	USINT		UNSIGNED8
Init value			Set at bootup (clear to preserve value on device)
Entry_I2001_S02			
Cyclic transmi...	Write		
Datatype	USINT		UNSIGNED8
Init value			Set at bootup (clear to preserve value on device)
Entry_I2001_S03			
Cyclic transmi...	Write		
Datatype	USINT		UNSIGNED8
Init value			Set at bootup (clear to preserve value on device)
Switching_Cycles_I20...			
AUX_Voltage_I2004			
Channel_Error_I2006 ...			
Entry_I2006_S01			
Cyclic transmi...	Read		
Datatype	USINT		UNSIGNED8
Entry_I2006_S02			
Cyclic transmi...	Read		
Datatype	USINT		UNSIGNED8
Entry_I2006_S03			
Cyclic transmi...	Read		
Datatype	USINT		UNSIGNED8
Module_Info_Flags_I2...			
Cyclic transmission	Read		
Datatype	UINT		UNSIGNED16
System_Commands_I...			
Cyclic transmission	Write		
Datatype	USINT		UNSIGNED8
Init value			Set at bootup (clear to preserve value on device)
ModuleErrorInput_I2009			
Cyclic transmission	Read		
Datatype	UINT		UNSIGNED16
Device specific parameters			Transmitted to the device at startup

The list of the Process Data can be viewed in the I/O mapping by right clicking on the device in the “Physical View”.

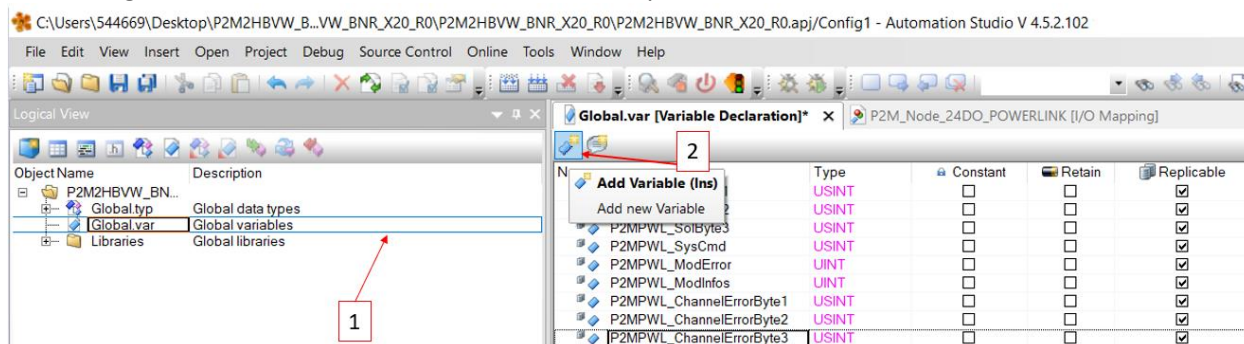


Channel Name	Process Variable	Data Type	Task Class
ModuleOk		BOOL	
Entry_I2001_S01		USINT	
Entry_I2001_S02		USINT	
Entry_I2001_S03		USINT	
Entry_I2006_S01		USINT	
Entry_I2006_S02		USINT	
Entry_I2006_S03		USINT	
Module_Info_Flags_I2007		UINT	
System_Commands_I2008		USINT	
ModuleErrorInput_I2009		UINT	

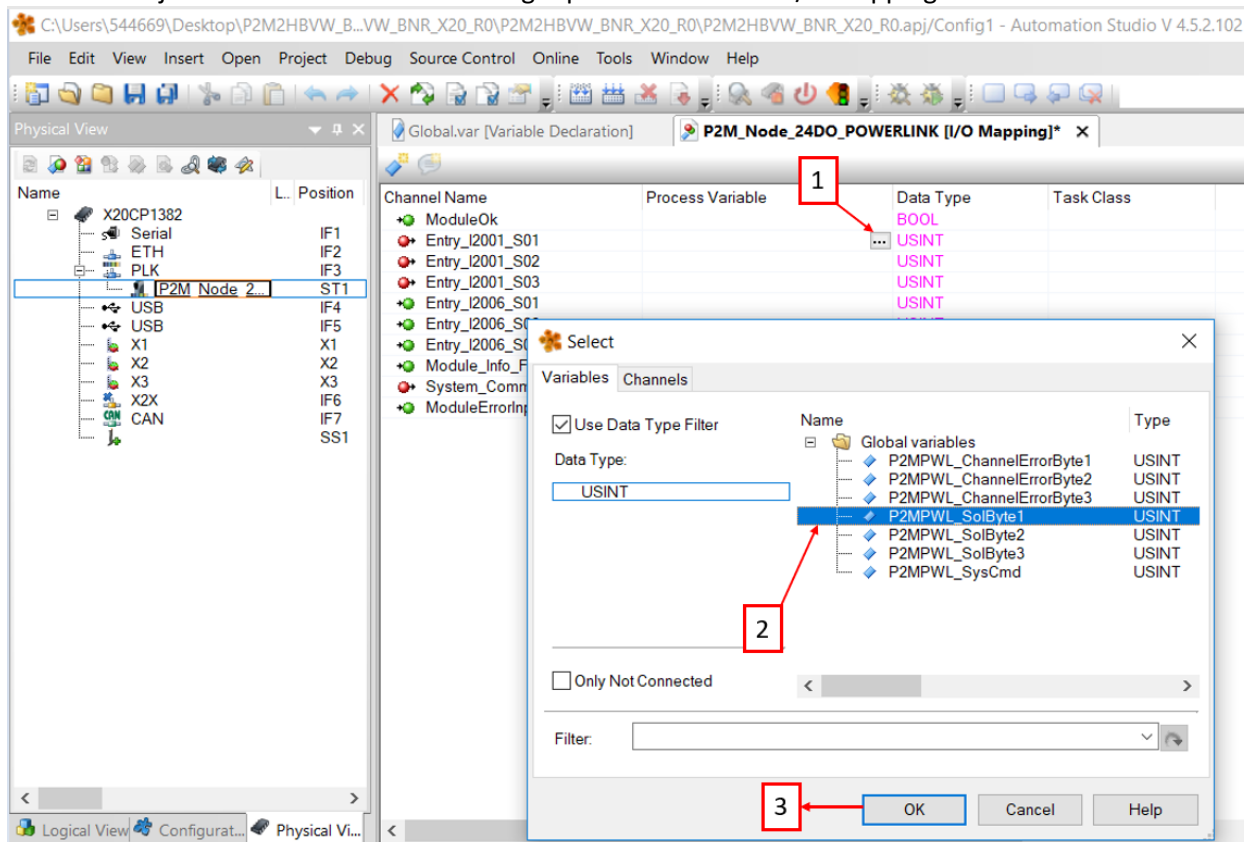


- c. I/O Map: Create the Global process data variable and connect them to the P2M target I/O mapping.

In the “Logical View” double click on “Global.var” to open the interface, then create the user variable.

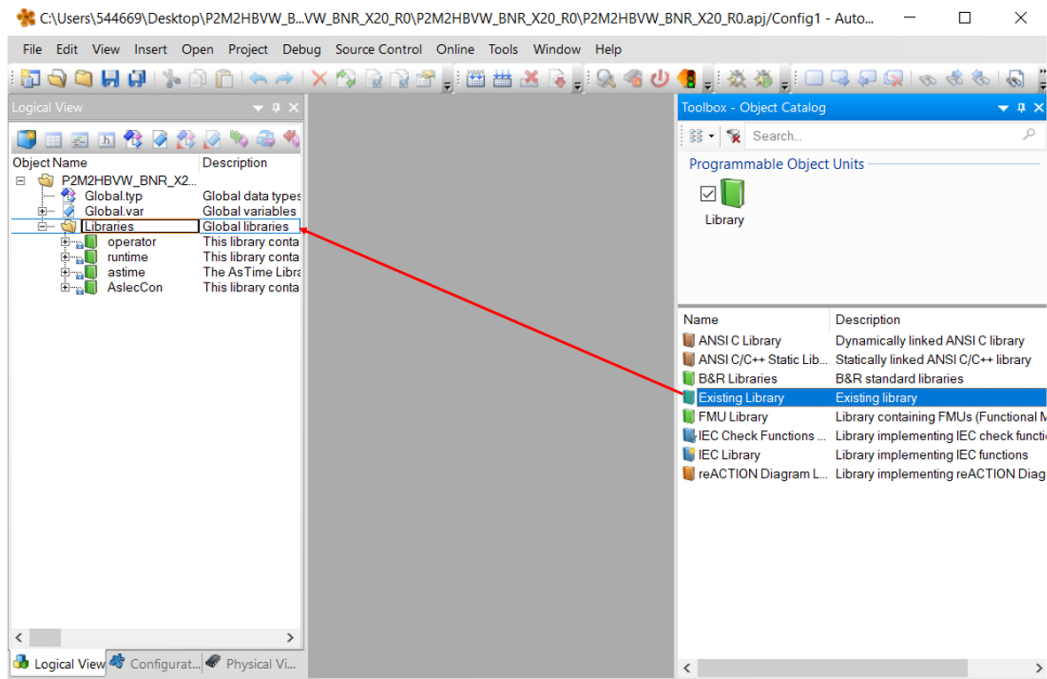


Connect the just created variable to the target process data from I/O mapping view

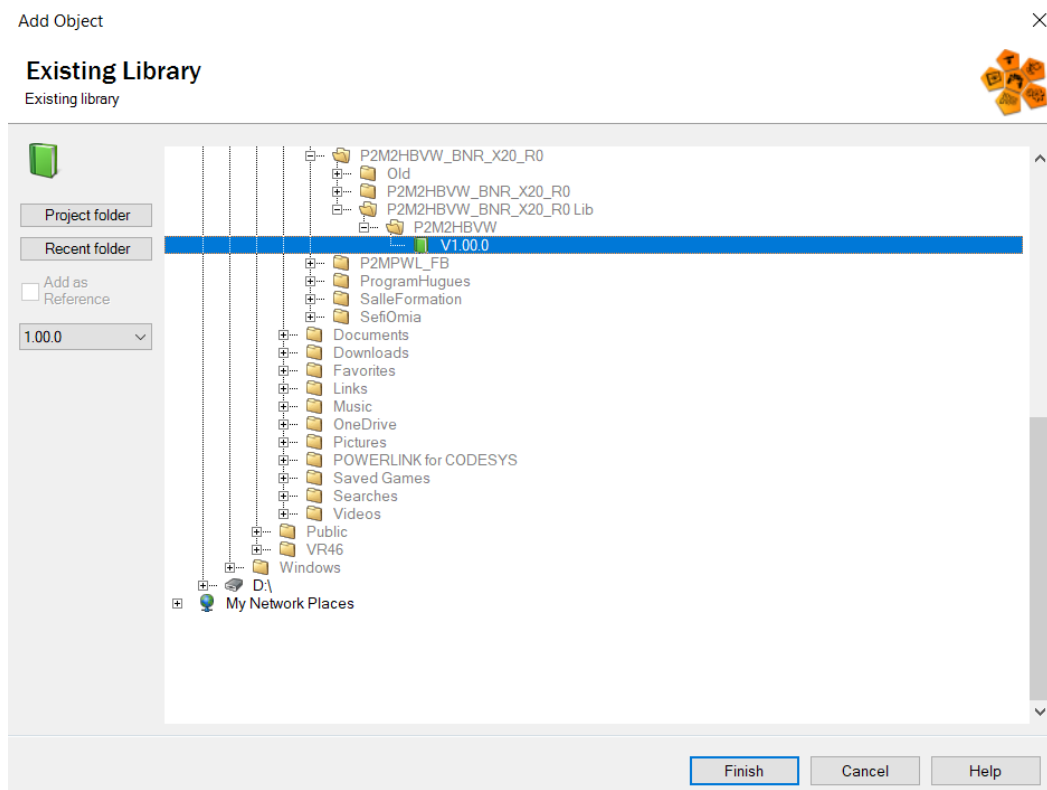


## 2. IMPORTING FUNCTION BLOCK

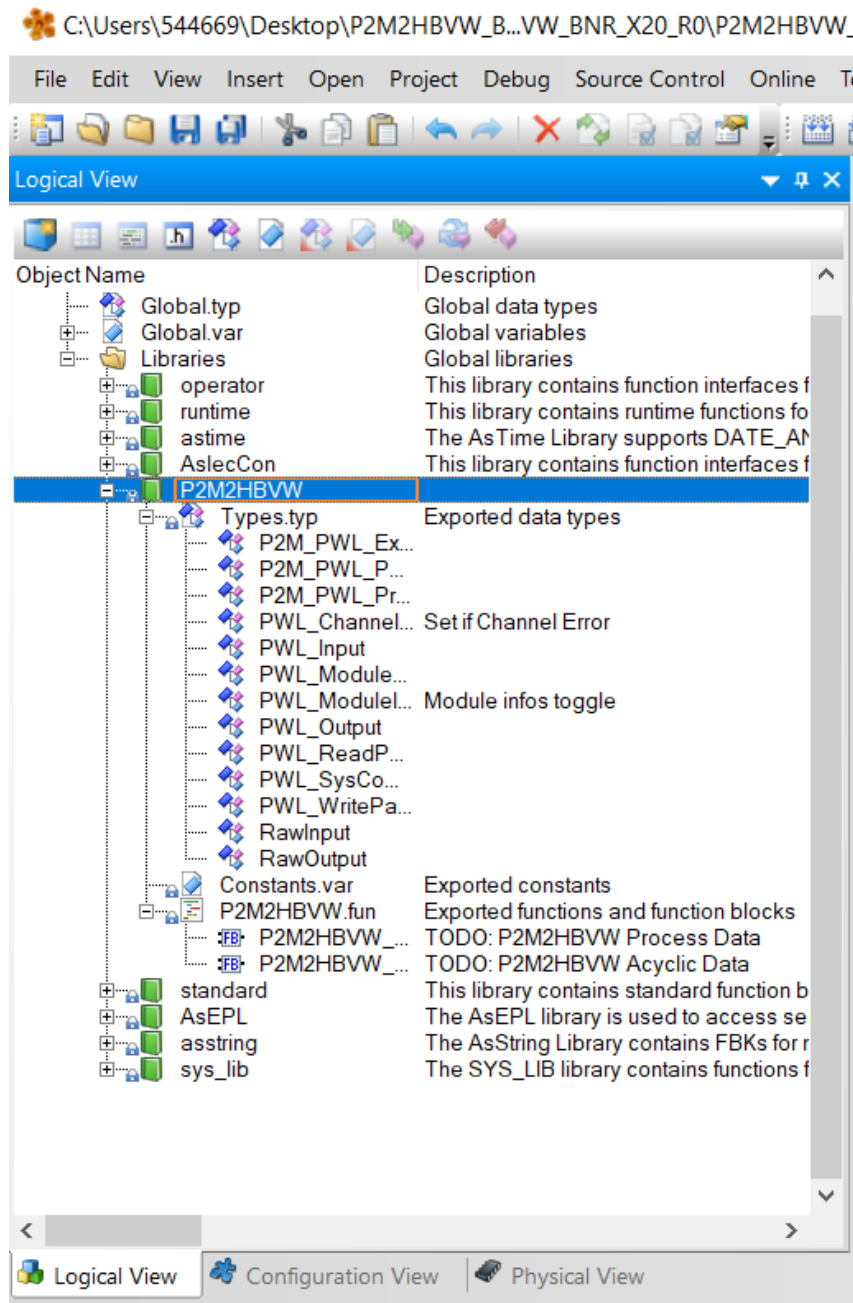
In the “Logical View” drag and drop “Existing Library” from “Toolbox – Object Catalog” to “Libraries”.



In the “Add Object” pop up interface, moves to the folder where the download library from website is stored select the library and click on “finish”.



Below is the view of the just imported library

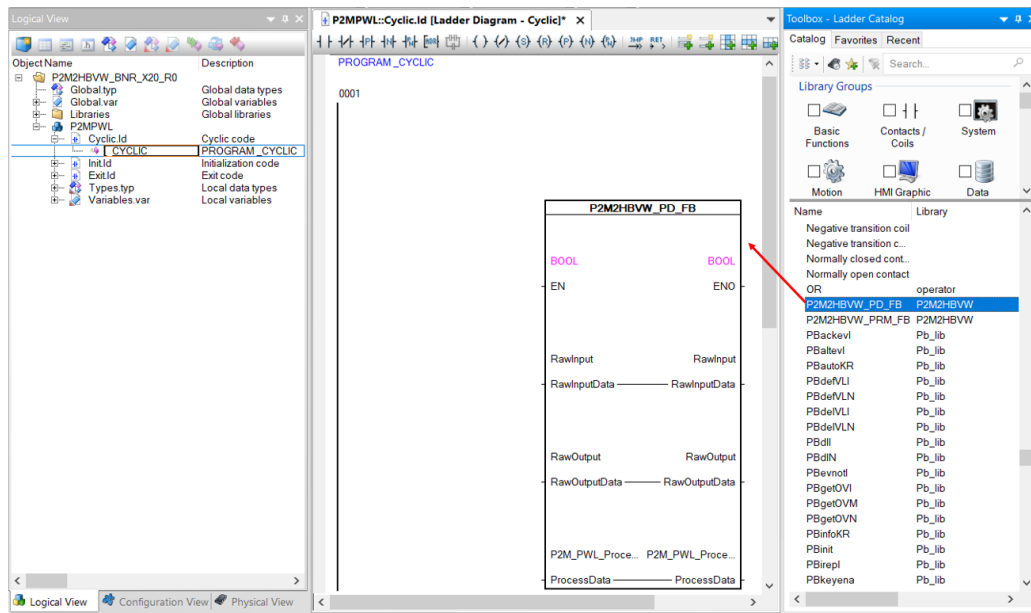


### 3. PROCESS DATA FUNCTION BLOCK INSTRUCTION

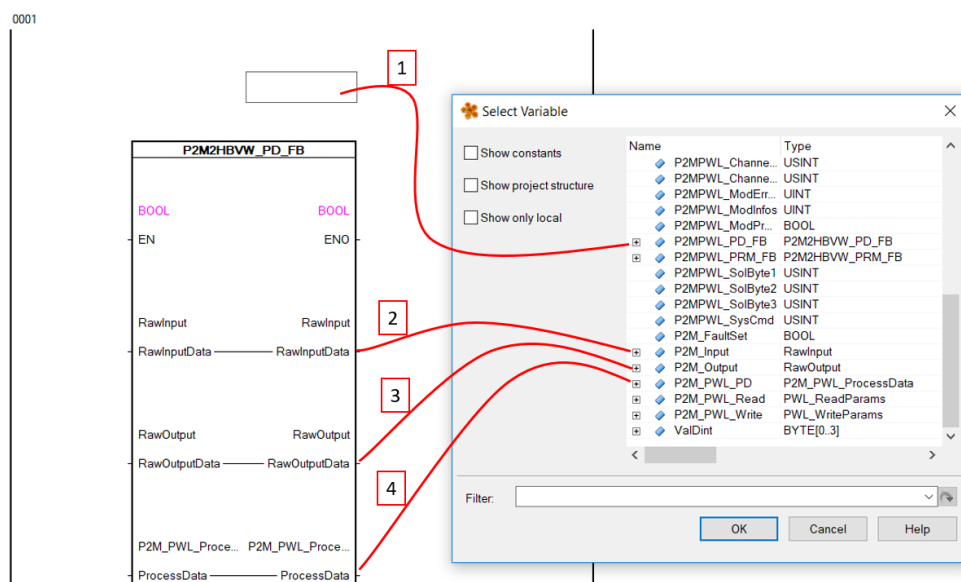
The “P2M2HBVW\_PD\_FB” Function Block simplifies the usage of Parker P2M2HBVW Powerlink devices with B&R PLCs when connected via Powerlink. Data is mapped to user-friendly control and diagnostic tags on the PLC side.

#### 3.1. Add the function block “P2M2HBVW\_PD\_FB” instance into the project

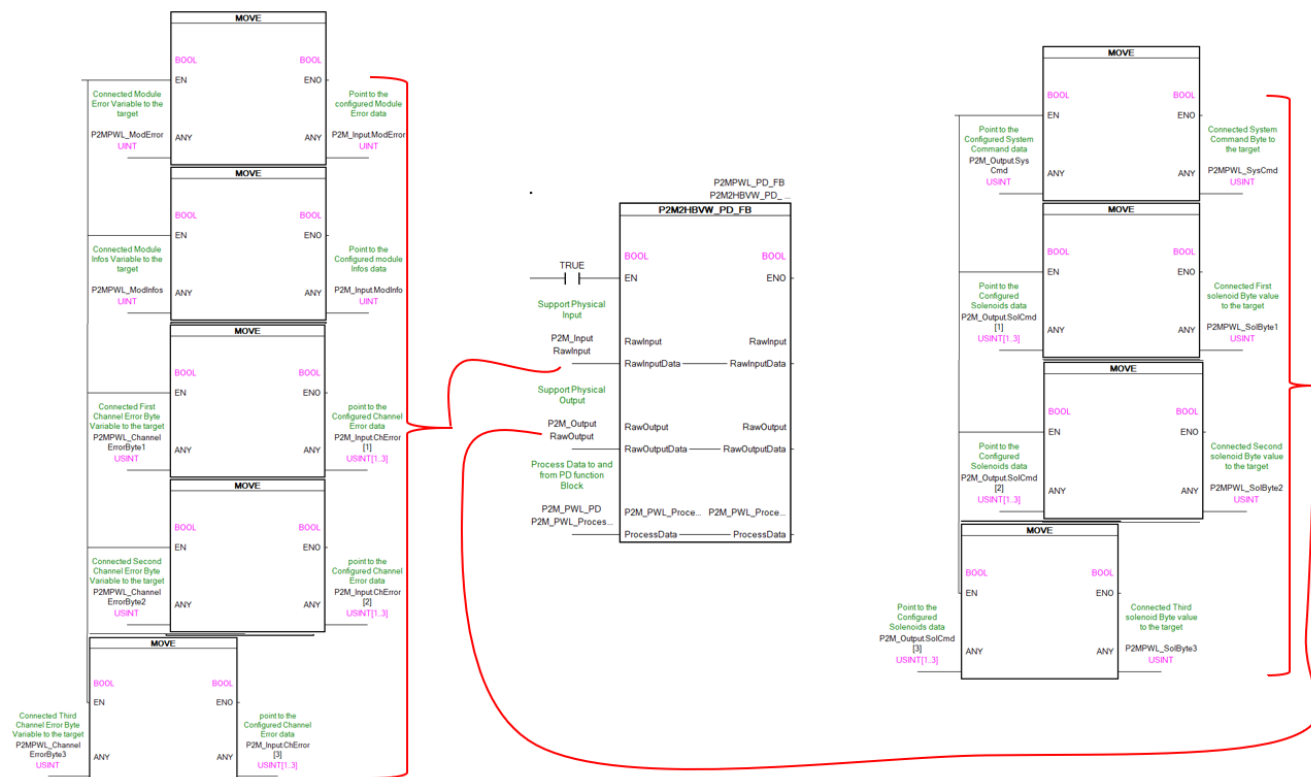
Drag and drop the function block “P2M2HBVW\_PD\_FB” from Toolbox – Ladder Catalog tab to an empty ladder network of the program cyclic interface.



Assign a variable to the FB (1), Double click the rectangle above the function block and select the corresponding variable from the list of the created, select the corresponding variable for the rest of the function tags [RawInputData (2), the RawOutputData (3) and the ProcessData (4)].



The RawInput and the RawOutput should be pointed to the global variable connected to the target input and output. See Appendix for structure breakdown of the “Data” variable.

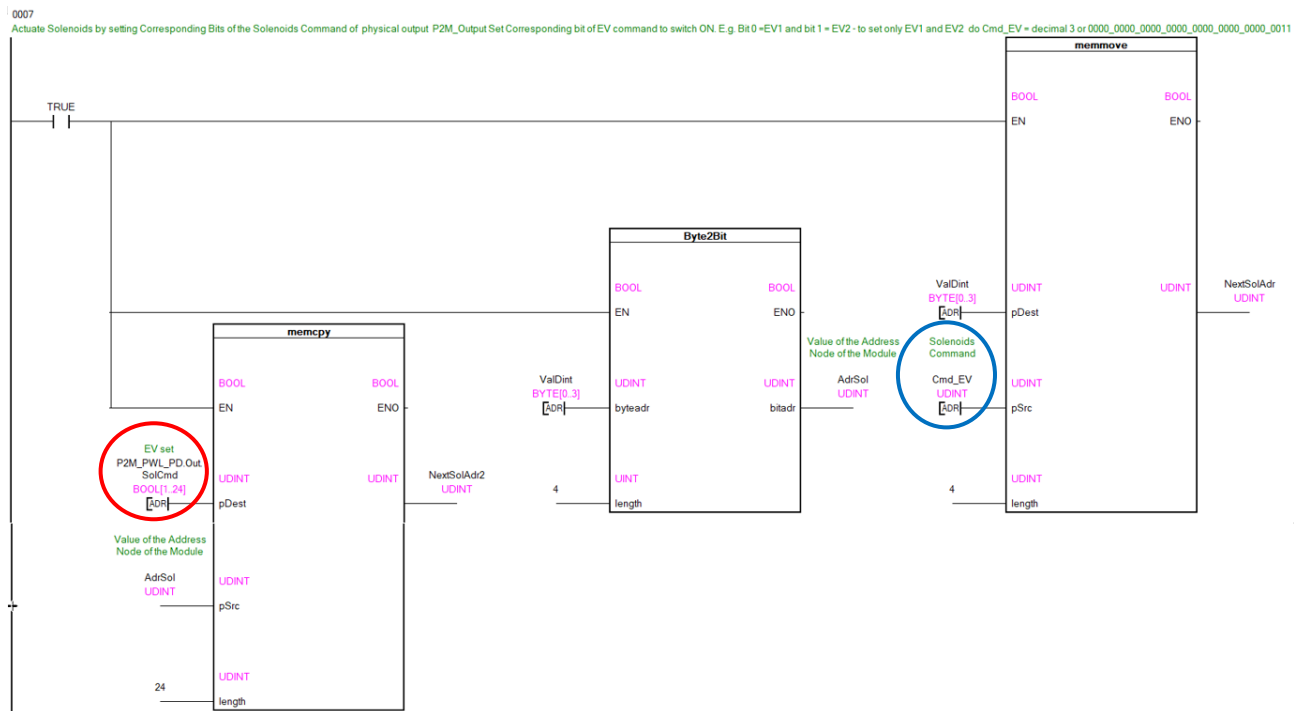


### 3.1. Using the instruction, Example of code “P2M2HBVW\_BNR\_X20\_R0

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\_PWL\_PD.In.xxx and P2M\_PWL\_PD.Out.xxx are live tags with real data just by calling the FB. See appendix for all data points available. See ladder logic examples below.

Download the example code from parker website and change the controller and set the addresses to match your physical devices.

### 3.1.1. Solenoid Valves Control (Cyclic)



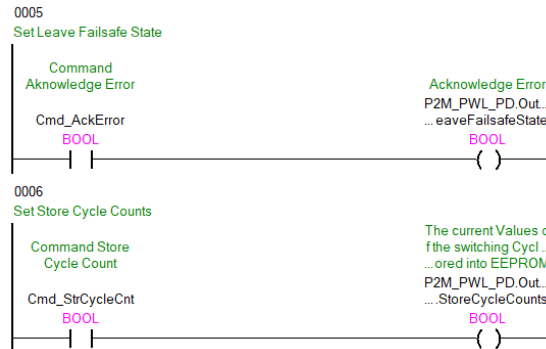
**Value correlation to Solenoid Address**

Cmd_EV = 1	P2M_PWL_PD.Out.SolCmd[1]=1
Cmd_EV = 2	P2M_PWL_PD.Out.SolCmd[2]=1
Cmd_EV = 4	P2M_PWL_PD.Out.SolCmd[3]=1
Cmd_EV = 8	P2M_PWL_PD.Out.SolCmd[4]=1
Cmd_EV = 16	P2M_PWL_PD.Out.SolCmd[5]=1
.	
.	
.	
Cmd_EV = 8 388 608	P2M_PWL_PD.Out.SolCmd[24]=1

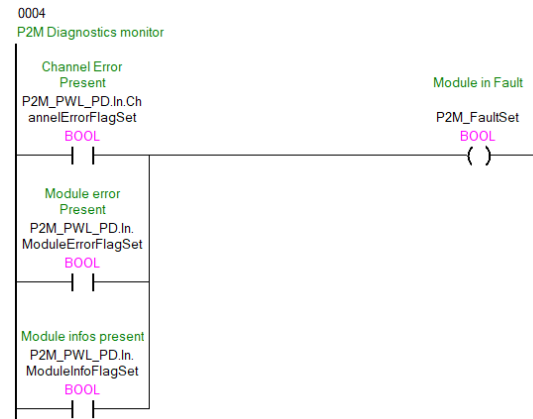
### 3.1.2. Writing System Commands (Cyclic)

Setting “P2M\_PWL\_PD.Out.SysCmd.LeaveFailsafeState” output element is equivalent to writing 0x01 to the first byte of output process data.

Setting “P2M\_PWL\_PD.Out.SysCmd.StoreCycleCounts” output element is equivalent to writing 0x02 to the first byte of output process data.



### 3.1.3. Monitoring Status Bits (Cyclic)



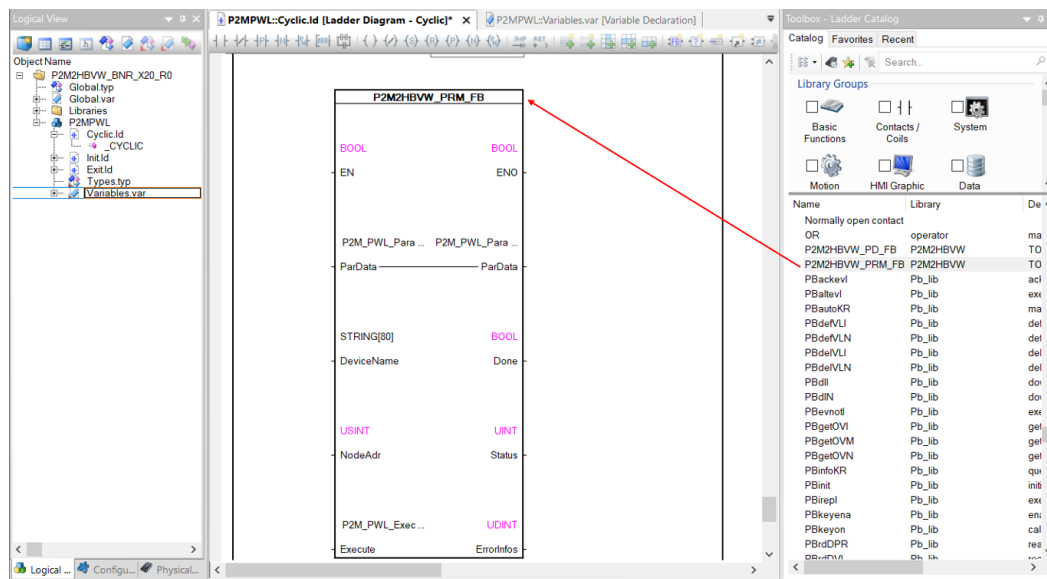
If any of the status bits are set, further information can be attained by evaluating specific diagnostic bits.

Channel Error	Module Error	Module info Flags
P2M_PWL_PD.In.ChannelError.CH[1]	P2M_PWL_PD.In.ModuleError.Ack_Required	P2M_PWL_PD.In.ModuleInfos.WatchdogValveMicro
P2M_PWL_PD.In.ChannelError.CH[2]	P2M_PWL_PD.In.ModuleError.AuxVoltageWarning	P2M_PWL_PD.In.ModuleInfos.EEPROM_Error
P2M_PWL_PD.In.ChannelError.CH[3]	P2M_PWL_PD.In.ModuleError.AuxVoltageFailure	P2M_PWL_PD.In.ModuleInfos.WatchdogComMicro
P2M_PWL_PD.In.ChannelError.CH[4]	P2M_PWL_PD.In.ModuleError.TemperatureWarning	P2M_PWL_PD.In.ModuleInfos.HeartbeatNotToggling
P2M_PWL_PD.In.ChannelError.CH[5]	P2M_PWL_PD.In.ModuleError.OutputChannelError	P2M_PWL_PD.In.ModuleInfos.HeartbeatState
P2M_PWL_PD.In.ChannelError.CH[6]	P2M_PWL_PD.In.ModuleError.ModuleError	
P2M_PWL_PD.In.ChannelError.CH[7]	P2M_PWL_PD.In.ModuleError.OutputStageNotAvailable	
P2M_PWL_PD.In.ChannelError.CH[8]		
P2M_PWL_PD.In.ChannelError.CH[9]		
P2M_PWL_PD.In.ChannelError.CH[10]		
P2M_PWL_PD.In.ChannelError.CH[11]		
P2M_PWL_PD.In.ChannelError.CH[12]		
P2M_PWL_PD.In.ChannelError.CH[13]		
P2M_PWL_PD.In.ChannelError.CH[14]		
P2M_PWL_PD.In.ChannelError.CH[15]		
P2M_PWL_PD.In.ChannelError.CH[16]		
P2M_PWL_PD.In.ChannelError.CH[17]		
P2M_PWL_PD.In.ChannelError.CH[18]		
P2M_PWL_PD.In.ChannelError.CH[19]		
P2M_PWL_PD.In.ChannelError.CH[20]		
P2M_PWL_PD.In.ChannelError.CH[21]		
P2M_PWL_PD.In.ChannelError.CH[22]		
P2M_PWL_PD.In.ChannelError.CH[23]		
P2M_PWL_PD.In.ChannelError.CH[24]		

## 4. PARAMETER DATA EXAMPLE CODE

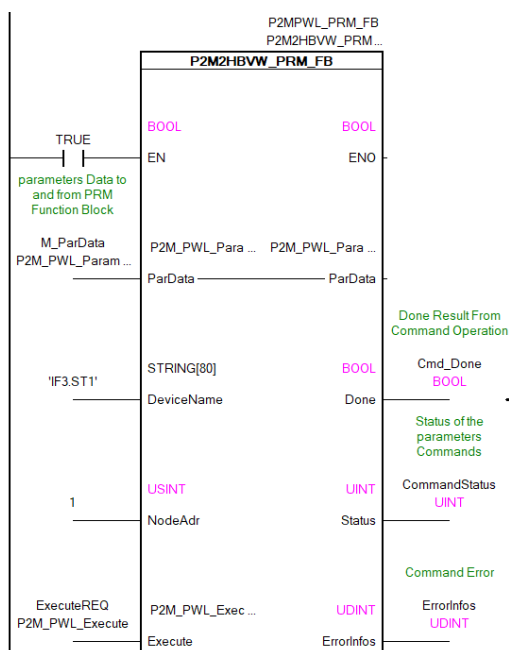
### 4.1. Add the function block “P2M2HBVW\_PRM\_FB” instance into the project

Drag and drop the function block “P2M2HBVW\_PRM\_FB” from Toolbox – Ladder Catalog tab to an empty ladder network of the program cyclic interface.



Assign a variable to the FB, Double click the rectangle above the function block and select the corresponding variable from the list of the created, select the corresponding variable for the rest of the function tags.

Read DeviceName and NodeAdr from the “Physical View” Tab and assign to the parameter function block.



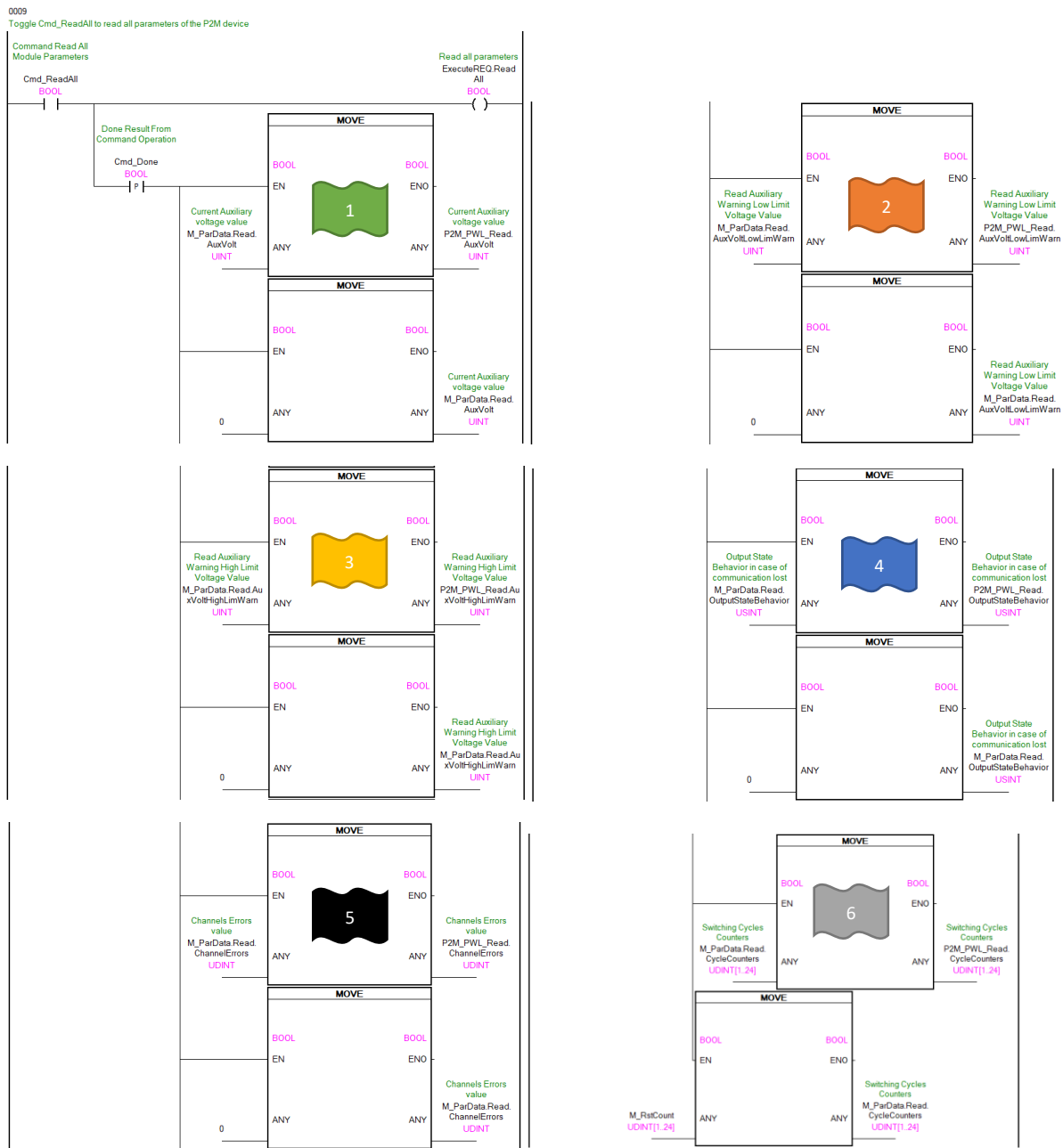


## 4.2. Parameter Code Instruction

**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.

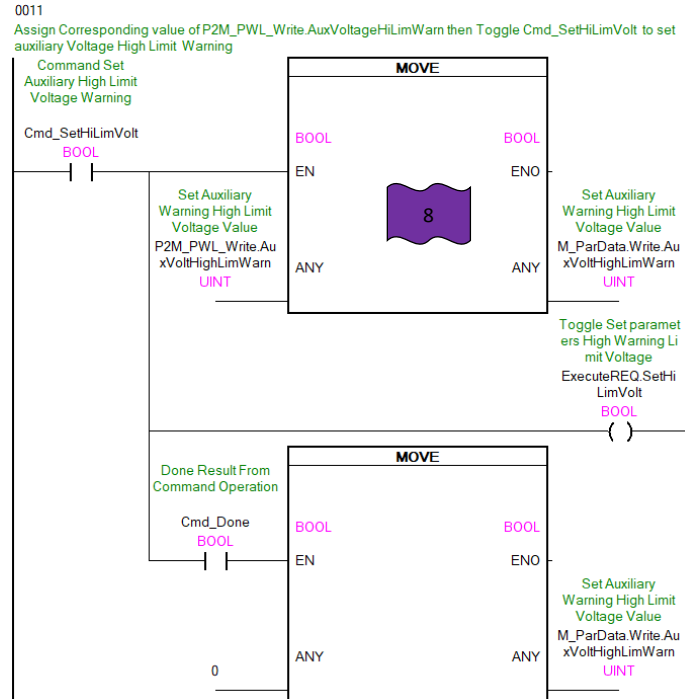
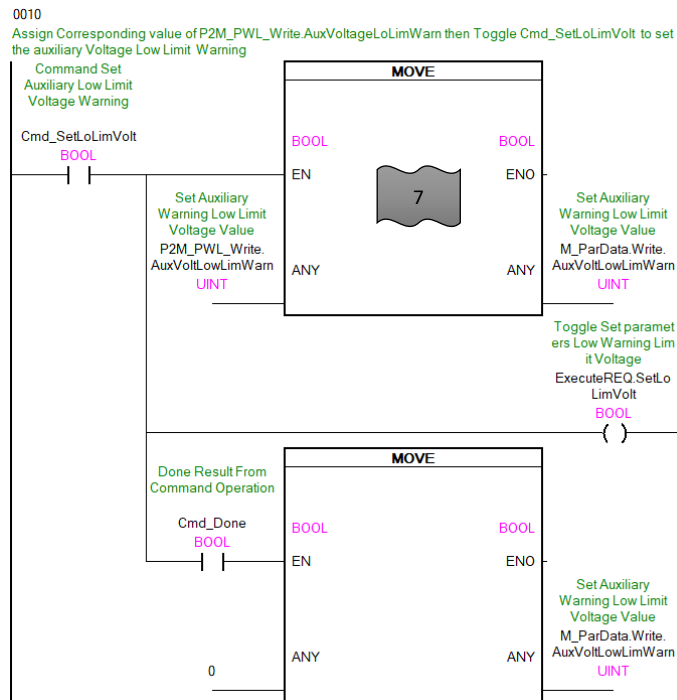
see below for logic example for each parameter. You may rename the Function Block Control tags to match your tag naming convention.



	Description	Index	Destination Element
1	Read Auxiliary Voltage	2004	P2M_PWL_Read.AuxVolt
2	Read Auxiliary Voltage Low Limit Warning	200B	P2M_PWL_Read.AuxVoltLowLimWarn
3	Read Auxiliary Voltage High Limit Warning	200C	P2M_PWL_Read.AuxVoltHighLimWarn
4	Read Output State Behavior	200D	P2M_PWL_Read.OutputStateBehavior
5	Read Channel Errors	2006	P2M_PWL_Read.ChannelErrors
6	Read Cycle Counters	2002	P2M_PWL_Read.CycleCounters

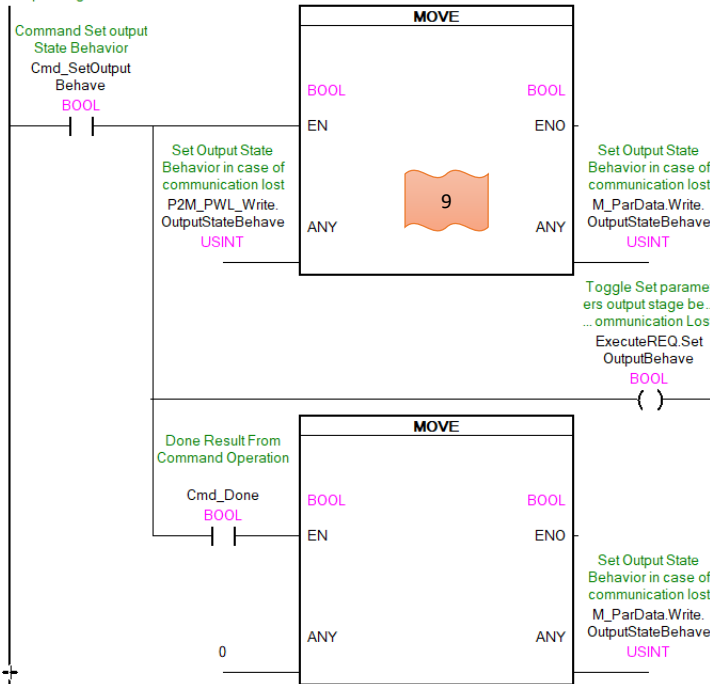
### 4.2.2. Writing Parameters.

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



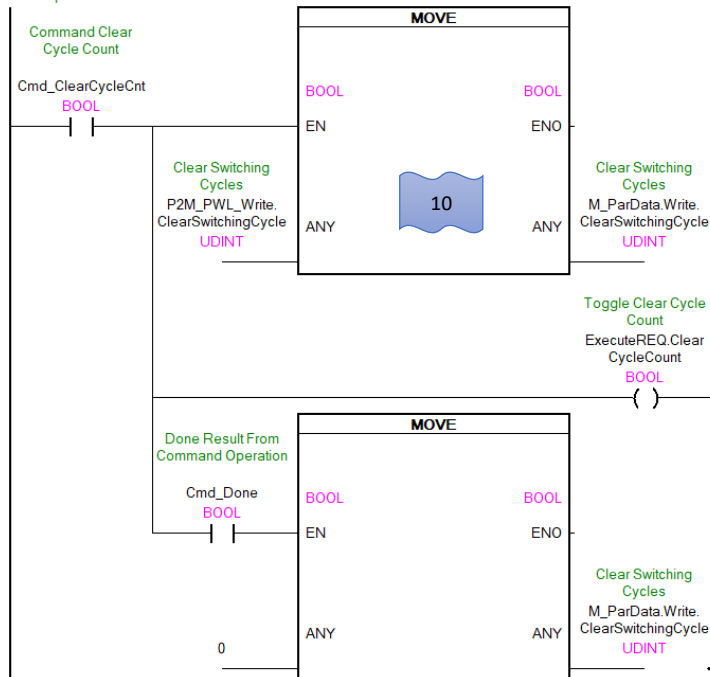
0012

Assign Corresponding value of P2M\_PWL\_Write.OutputStateBehave then toggle Cmd\_SetOutputBehave to set Output Stage behavior After Communication lost.



0013

Clear Cycle count - Set Corresponding bit of P2M\_PWL\_Write.ClearSwitchingCycle to reset cycle counter for that solenoid. E.g. Bit 0 = EV1 and bit 1 = EV2 - to clear only EV1 and EV2 P2M\_PWL\_Write.ClearSwitchingCycle should = decimal 3 or 0000\_0000\_0000\_0000\_0000\_0000\_0011. Toggle Cmd\_ClearCycleCnt to execute the clear operation



	Description	Index	Source Element	Source Length (bytes)
7	<b>Write Auxiliary Voltage Low Limit Warning</b>	200B	P2M_PWL_Write.AuxVoltLowLimWarn	2
8	<b>Write Auxiliary Voltage High Limit Warning</b>	200C	P2M_PWL_Write.AuxVoltHighLimWarn	2
9	<b>Write Output State Behavior</b>	200D	P2M_PWL_Write.OutputStateBehave	1
10	<b>Clear Cycle Counters</b>	2003	P2M_PWL_Write.ClearSwitchingCycle	3

## APPENDIX

### Process Data Structures

User Defined Data Structures utilized by FB “P2M2HBVW\_PD\_FB”

#### ➤ Raw in-out data structure

<b>RawInput</b>					
ModError	UINT		<input checked="" type="checkbox"/>		Point to the configured Module Error data
ChError	USINT[1..3]		<input checked="" type="checkbox"/>		point to the Configured Channel Error data
ModInfo	UINT		<input checked="" type="checkbox"/>		Point to the Configured module Infos data
<b>RawOutput</b>					
SolCmd	USINT[1..3]		<input checked="" type="checkbox"/>		Point to the Configured Solenoids data
SysCmd	USINT		<input checked="" type="checkbox"/>		Point to the Configured System Command data

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

<b>P2M_PWL_ProcessData</b>					
In	PWL_Input		<input checked="" type="checkbox"/>		
Out	PWL_Output		<input checked="" type="checkbox"/>		

##### a) Process input structure (PWL\_Input)

<b>PWL_Input</b>					
ModuleErrorFlagSet	BOOL		<input checked="" type="checkbox"/>		Module error Present
ModuleError	PWL_ModuleError		<input checked="" type="checkbox"/>		
ChannelErrorFlagSet	BOOL		<input checked="" type="checkbox"/>		Channel Error Present
ChannelError	PWL_ChannelError		<input checked="" type="checkbox"/>		
ModuleInfoFlagSet	BOOL		<input checked="" type="checkbox"/>		Module infos present
ModuleInfos	PWL_ModuleInfos		<input checked="" type="checkbox"/>		

##### ModuleError (PWL\_ModuleError)

<b>PWL_ModuleError</b>					
Ack_Required	BOOL		<input checked="" type="checkbox"/>		Set if any major fault active
AuxVoltageWarning	BOOL		<input checked="" type="checkbox"/>		Set if auxiliary Voltage in Warning Range
AuxVoltageFailure	BOOL		<input checked="" type="checkbox"/>		Auxiliary Voltage In Error Range
TemperatureWarning	BOOL		<input checked="" type="checkbox"/>		Set if a temperature increase above warning levels
OutputChannelError	BOOL		<input checked="" type="checkbox"/>		Set if a major fault is detected at the output stage
Module_Error	BOOL		<input checked="" type="checkbox"/>		Set if an internal communication error is active
OutputStageNotAvailable	BOOL		<input checked="" type="checkbox"/>		Set if auxiliary power is missing

##### ChannelError (PWL\_ChannelError)

<b>PWL_ChannelError</b>					
CH	BOOL[1..24]		<input checked="" type="checkbox"/>		Set if Channel Error

##### ModuleInfos (PWL\_ModuleInfos)

<b>PWL_ModuleInfos</b>					
WatchdogValveMicro	BOOL		<input checked="" type="checkbox"/>		Module infos toggle
EEProm_Error	BOOL		<input checked="" type="checkbox"/>		Set If the Watchdog caused the last reset of the output Stage
WatchdogComMicro	BOOL		<input checked="" type="checkbox"/>		Set if any of the (expected) data stored in the EEPROM has been detected
HeartbeatNotToggling	BOOL		<input checked="" type="checkbox"/>		Set if the Watchdog caused the last reset of the communication stage
HeartbeatState	BOOL		<input checked="" type="checkbox"/>		Heartbeat is currently not toggling whilst it should
					Used to troubleshoot the cause of the "heartbeat Not Toggling" error

##### b) Process output structure (PWL\_Output)

<b>PWL_Output</b>					
SysCmd	PWL_SysCommand		<input checked="" type="checkbox"/>		
SolCmd	BOOL[1..24]		<input checked="" type="checkbox"/>		EV set

##### System Commands (PWL\_SysCommand)

<b>PWL_SysCommand</b>					
LeaveFailsafeState	BOOL		<input checked="" type="checkbox"/>		Acknowledge Error
StoreCycleCounts	BOOL		<input checked="" type="checkbox"/>		The current Values of the switching Cycle Counters are stored into EEPROM

## Parameter Data Structures

User Defined Data Structures utilized in the sample code

P2M_PWL_ParamsData					
Read	PWL_ReadParams	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
Write	PWL_WriteParams	<input type="checkbox"/>	<input checked="" type="checkbox"/>		

### a) Read parameters data structure

PWL_ReadParams					
AuxVolt	UINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Current Auxiliary voltage value
AuxVoltLowLimWarn	UINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Read Auxiliary Warning Low Limit Voltage Value
AuxVoltHighLimWarn	UINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Read Auxiliary Warning High Limit Voltage Value
OutputStateBehavior	USINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Output State Behavior in case of communication lost
ChannelErrors	UDINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Channels Errors value
CycleCounters	UDINT[1..24]	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Switching Cycles Counters

### b) Write parameters data structure

PWL_WriteParams					
AuxVoltLowLimWarn	UINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Set Auxiliary Warning Low Limit Voltage Value
AuxVoltHighLimWarn	UINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Set Auxiliary Warning High Limit Voltage Value
OutputStateBehave	USINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Set Output State Behavior in case of communication lost
ClearSwitchingCycle	UDINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Clear Switching Cycles

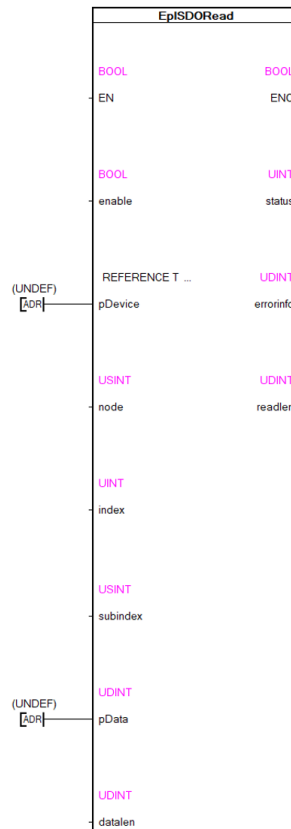
### c) List of the read/write command structure

P2M_PWL_Execute					
ReadAll	BOOL	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Read all parameters
SetLoLimVolt	BOOL	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Toggle Set parameters Low Warning Limit Voltage
SetHiLimVolt	BOOL	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Toggle Set parameters High Warning Limit Voltage
SetOutputBehave	BOOL	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Toggle Set parameters output stage behavior after after Communication Lost
ClearCycleCount	BOOL	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Toggle Clear Cycle Count

Use Automation Studio function blocks “EpISDORed()”, “EpISDOWrite()” to read and write acyclic data.

These function blocks can be found in the Automation Studio library (asEPL libraries) and can be use as an alternative to the parker function block” P2M2HBVW\_PRM\_FB” to read or write a single parameter.

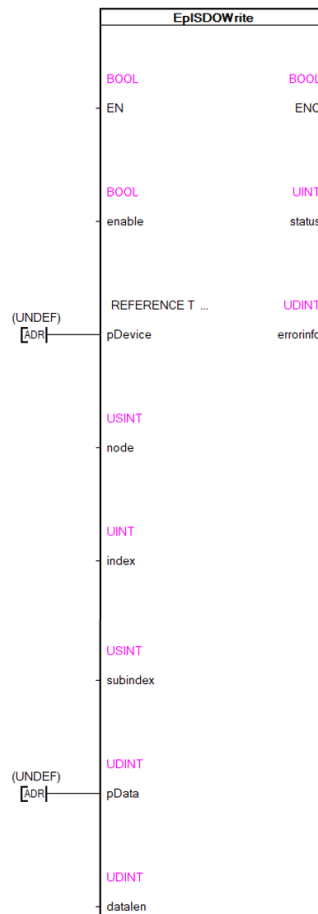
- **EplSDORead()** instruction is used to read a P2M2HBVW's object.



Parameters	pDevice	Node	Index	SubIndex	pData	dataLen
Read Auxiliary Voltage	Device name of the P2M POWERLINK interface (syntax: 'IF3.ST1').	Node Address of your P2M Powerlink coupler	8196	0	target area for the data to be read	2
Read Auxiliary Voltage Low Limit Warning			8203	0		2
Read Auxiliary Voltage High Limit Warning			8204	0		2
Read Output State Behavior			8205	0		1
Read Channel Errors			8198	1 to 3		1
Read Cycle Counters			8194	1 to 24		4



➤ **EplSDOWrite()** instruction is used to Write a P2M2HBVW's object



Parameters	pDevice	Node	Index	SubIndex	pData	datalen
Write Auxiliary Voltage Low Limit Warning	Device name of the P2M POWERLINK interface (syntax: 'IF3.ST1').	Node Address of your P2M Powerlink coupler	8203	0	Source data to be written	2
Write Auxiliary Voltage High Limit Warning			8204	0		2
Write Output State Behavior			8205	0		1
Clear Switching Cycle counters			8195	1 to 3		1