# **PMEUCM VALVES**

**Installation Manual** 

This manual covers the PMEUCM VALVES application and installation procedures. This application allows redundant rings of Modbus valve actuators to be controlled by an M580 PAC. Actuators from Limitorque, AUMU, EIM, and ROTORK may be used on the same ring network with a single interface to the PAC.

Effective: May 7, 2018



Niobrara Research & Development Corporation P.O. Box 3418 Joplin, MO 64803 USA

Telephone: (800) 235-6723 or +1 (417) 624-8918 Facsimile: +1 (417) 624-8920 www.niobrara.com All trademarks and registered trademarks are the property of their respective owners. Subject to change without notice.

© Niobrara Research & Development Corporation 2018. All Rights Reserved.

### Contents

1	Introduction	7
2	SE PME DTM Library	9
	Installation	9
3	PMEUCM SETUP.EXE	.15
4	PMEUCM TCPOPEN SETUP.EXE	.17
5	Adding the PMEUCM to Unity Pro	.19
-	Adding the PMEUCM to Unity Hardware Catalog Manager	20
6	NRD PTK DTM UTIL	.25
	Open the NRD DTM Tool	25
	Installing a new file	26
7	System Operation	.31
	Full Frame Data Arrays	31
	Outbound Operations	32
	Inbound Operations	32
	TCPOPEN DTM Variables	32
	Freshness	33
	Inputs	33
	UCM_Runtime_Status	34
	UCM_Halt_Line_Number	36
	UCM_Error	36
	UCM_Status	37
	SI_Remote_IP	38
	SI_Remote_Port	40
	SI_Local_Port	40
	SI_Status	41
	SI_Out_Handshake_W0	42
	SI_In_Handshake_W0	42
	SI_Number_W0	43
	SI_Length_W0	43
	SI_More_W0	43
	SI_Data_W0	43
	Outputs	45
	UCM_Command	45
	SO_Remote_IP	46

SO Remote Port	47
SO Local Port	48
SO Command	48
SO Out Handshake W0	
SO In Handshake W0	50
SO Number W0	50
SO Length W0	51
SO More W0	51
SO_Data_W0	51
TCPOPEN DFBs	53
TCPOPEN_Inbound	53
PLC State Machines	57
Socket Control and Timing	58
SYSUPTIME	58
8 QLOAD the TCPOPEN UCM Application	61
QLOAD the TCPOPEN Application	61
UCM BOOT firmware too old	64
UCM OS too old	64
9 Unity Pro Operations	67
New Project	67
DTM Hardware Catalog Update	73
Link the DTM to the PMEUCM Hardware	80
DTM Configuration	
E1 and E2 Ethernet Port	
UCM OS Settings	85
S1 and S2 Serial Port	86
Applying and Installing Changes to the DTM	
Import FBD Code	90
Import Modbus/TCP server ST section	95
Build All or Build Changes	96
Transfer Project to PLC	97
PLC Set Address	97
PLC Connect	99
Transfer Project to PLC	99
10 Front Panel Operation	103
LED Panel	103
PTK Board Controlled Lights	103
UCM OS Controlled Lights	104
USER Controlled Lights	
LCD and Joystick Operation	106
Fault Indication	
Normal Operation	108
Backlight.	
Menus	108

	Menu Screen	
	Sockets Screens	
	Config Menu	
	Stats Menu	
	System Menu	
11	Debug Web Server	
	Home Page	
	Socket Status Page	
12	Debug Telnet Server	
13	Modbus/TCP Server Example	
14	Modbus/TCP Server+Client Example2	131

### **1** Introduction

The Niobrara PMEUCM is a programmable communication card for the Schneider Electric x80 PAC platform. It is capable of running a custom application for performing communication translations between serial and/or Ethernet protocols for the Modicon M580 Automation platform.

This document covers an application provides a Modbus RTU serial ring network for controlling Limitorque, EIM, AUMU, and ROTORK valve actuators. Both serial ports of the PMEUCM are connected to a closed ring of actuators and the PMEUCM can determine if the ring is complete and which valves are accessible from either port of the PMEUCM. This network topology allows for the breaking of the network in one location while still providing control of all of the actuators.

This application may be used in either the PMEUCM0302 with external isolated RS-232<>RS-485 converters for each serial port, or the PMEUCM0312 with its built-in isolated 2-wire RS-485 serial ports. The application works the same in both models of the PMEUCM.

The DTM provided with this application requires a maximum of 120 bytes of DIO I/O which means that up to 17 PMEUCMs may be installed on a normal '40 series M580 CPU with 2048 bytes of DIO. The PMEUCM may be mounted in the local CPU rack of a stand-alone M580 ('20 or '40 series). The PMEUCM may also be mounted in Ethernet Remote Racks of '40 series CPUs. This application may be used in Ethernet Remote Racks of Hot Standby (HSBY) CPUs as well.

Each PMEUCM supports up to 254 Modbus actuators on its copper ring network.



The example in Figure 1.1 shows three Modbus valves connected to the two serial ports of a PMEUCM0312 forming a ring network. The M580 CPU may open or close any of the valves. The ring network allows for full control of the remote valves in the event of a broken cable between any two valves or the PMEUCM.

## 2 SE PME DTM Library

The latest version of Schneider Electric's PME DTM Library must be installed before attempting to use the PMEUCM. The PMEUCM requires many newly added features to the PME DTM.

The latest version is available at Niobrara's web site:

http://www.niobrara.com/programs/PME\_DTM\_Library\_SETUP\_1\_0\_30.EXE



### Installation

NOTE: Unity Pro must be closed before installing the PME\_DTM\_Library. NOTE: Unity Pro V11.1 and V12.0 are shipped with an older version of the

PME\_DTM\_Library already installed. The installed version must be updated to allow the PMEUCM0302 to operate properly.

NOTE: Unity V13 includes the proper DTM Library. It is not necessary to update the library in V13.

NOTE: Installing a new version of the PME\_DTM\_LIBRARY will remove all currently installed DDXML files from the repository.

Use the File>Restore feature of the NRDPTKDDXMLUTIL program to recover the previously installed DTMs.

NOTE: This example shows version 1.0.30. The actual file downloaded may not be this version but the procedure is the same.

After downloading the PME\_DTM\_Library\_SETUP\_1\_0\_30.EXE file, run it to begin the installation.

PME_DTM_Library_SETUP Setup: Installation Fo	-		×
This will install the Schneider PME DTM Library on your directory	our com	nputer. Cho	ose a
- Destination Folder			
C: Wiobrara		Browse	
Space required: 29.0MB Space available: 688.5GB	1		
Cancel Nullsoft Install System v2,46		Inst	all

The self extractor will install the setup files into the <u>c:Niobrara</u> folder and automatically start the S-E installation wizard.

· 문 DTM Library SET	IIP Setury Completed
Completed	
Show <u>d</u> etails	
Cancel Nullsoft 1	Install System v2,46 < Back Gose
Schneider Electric PME DT	M Library - 1.0.30
	Welcome to the Installation Wizard for Schneider
,	Welcome to the Installation Wizard for Schneider Electric PME DTM Library
	Welcome to the Installation Wizard for Schneider Electric PME DTM Library
,	Welcome to the Installation Wizard for Schneider Electric PME DTM Library The Installation Wizard will allow you to modify, repair, or remove Schneider Electric PME DTM Library. To continue, click Next.
	Welcome to the Installation Wizard for Schneider Electric PME DTM Library The Installation Wizard will allow you to modify, repair, or remove Schneider Electric PME DTM Library. To continue, click Next.
	Welcome to the Installation Wizard for Schneider Electric PME DTM Library The Installation Wizard will allow you to modify, repair, or remove Schneider Electric PME DTM Library. To continue, click Next.
5	Welcome to the Installation Wizard for Schneider Electric PME DTM Library The Installation Wizard will allow you to modify, repair, or remove Schneider Electric PME DTM Library. To continue, click Next.
5	Welcome to the Installation Wizard for Schneider Electric PME DTM Library The Installation Wizard will allow you to modify, repair, or remove Schneider Electric PME DTM Library. To continue, dick Next.
Schneider	Welcome to the Installation Wizard for Schneider Electric PME DTM Library The Installation Wizard will allow you to modify, repair, or remove Schneider Electric PME DTM Library. To continue, dick Next.
Schneider Electric	Welcome to the Installation Wizard for Schneider Electric PME DTM Library The Installation Wizard will allow you to modify, repair, or remove Schneider Electric PME DTM Library. To continue, dick Next.
Scheider	Welcome to the Installation Wizard for Schneider Electric PME DTM Library         The Installation Wizard will allow you to modify, repair, or remove Schneider Electric PME DTM Library. To continue, click Next.

If you have a previous version of the PME DTM Libraray installed, you may be prompted to Repair or Remove the previous installation. Select Repair.

🔀 Schneider Ele	ctric PME DTM Library - 1.0.30	×
<b>Program Maintenance</b> Modify, repair, or remove the program.		Schneider Electric
Repair	Repair installation errors in the program. This option fixes corrupt files, shortcuts, and registry entries.	s missing or
○ <u>R</u> emove	Remove Schneider Electric PME DTM Library from your co	mputer.
	< <u>B</u> ack <u>N</u> ext >	Cancel

When finished a screen like the following should be displayed.



After selecting "Finish", the S-E Wizard will close. Now close the PME\_DTM\_Library\_Setup window to complete the setup.

PME_DTM_Library_SETUP Setup: Completed	—	
Completed		
Show <u>d</u> etails		
Cancel Nullsoft Install System v2.46	< <u>B</u> ack	ie -

If you have not installed the PMEUCM\_SETUP.EXE program and added the PMEUCM0302 (or PMEUCM0312) to the Unity Hardware Catalog, proceed immediately to the next chapter before starting Unity Pro.

## 3 PMEUCM\_SETUP.EXE

The latest version of Niobrara's PMEUCM\_SETUP must be installed before attempting to use the PMEUCM. This setup installs many utilities needed to configure the PMEUCM. The user may access this file at:

http://www.niobrara.com/html/pmeucm\_cut.html



Download and run PMEUCM\_SETUP.EXE. A box will appear prompting the user to choose a directory in which to install. The default is <u>C:\Niobrara</u>, as shown below.

PMEUCM_SETUP Setup: Installation Folder -	
This will install important PMEUCM-related files on your directory	computer. Choose a
Destination Folder	Browse
Space required: 9.4MB	
Space available: 688.5GB	
Cancel Nullsoft Install System v3.01	<u>I</u> nstall

PMEUCM_SETUP Setup: Completed	—		$\times$
Completed			
Show <u>d</u> etails			
Cancel Nullsoft Install System v3.01	< <u>B</u> ack	Clos	æ

## 4 PMEUCM\_VALVES\_SETUP.EXE

The latest version of Niobrara's PMEUCM\_VALVES\_SETUP must be installed. This setup installs many files needed to configure the PMEUCM for the VALVES application. The user may access this file at:

http://www.niobrara.com/html/apps/pmeucm/VALVES.html



PMEUCM VALVES Manual

Download and run PMEUCM\_VALVES\_SETUP.EXE. A box will appear prompting the user to choose a directory in which to install. The default is <u>C:\Niobrara</u>\Apps\PMEUCM\VALVES, as shown below.

PMEUCM_TCPOPEN_SETUP Setup: Installation	—		×
This will install important TCPOPEN-related files on y a directory	your ca	omputer. Ch	oose
Destination Folder		Browse	
Space required: 2.5MB			
Space available: 671.3GB			
Cancel Nullsoft Install System v2.46		<u>I</u> nsta	

## 5 Adding the PMEUCM to Unity Pro

Unity Pro versions 8.1 and higher provide a method for adding third party modules to their hardware catalog. Niobrara provides the necessary .cpx files as part of PMEUCM\_SETUP.EXE. The user may access this file at:

http://www.niobrara.com/html/pmeucm\_cut.html



### Adding the PMEUCM to Unity Hardware Catalog Manager

**NOTE**: Unity Pro must be not be running to access the Hardware Catalog Manager.

After the setup program is finished, start the Hardware Catalog Manager, located at Start>All Programs>Schneider Electric. In the File menu, click on Import User Devices, as shown below.

Hardware Catalog Manager	-		×
<u>Eile Edit View Service H</u> elp			
Import User Devices Ctrl+I			
Export User Devices Ctrl+E	E	Build Catal	og
Exit	41.	-1 k / 100	
€ Safety	ADC	IT MODIFIC	ations
Ensors     Third party products		Close	
Build Import/Export Log /			
Add one or several devices from archive file			

Choose the folder where PMEUCM\_SETUP.EXE installed the .cpx file. This is normally the 'c:\Niobrara\PMEUCM\DTM\' folder.

Inside the DTM folder is a file for Unity V11, V11.1, V12, and V13:

 $M580\_PME\_UCM\_0302\_for\_V11\_and\_V12.cpx$ 

Also present is a cpx file for Unity V12 after installing the Unity Hot Fix V12HF\_M580\_S908\_CCOTF:

M580\_PME\_UCM\_0302\_after\_V12HF\_M580\_S908\_CCOTF.cpx

There is only one cpx file for the PMEUCM0312 and it works with Unity V11-13:

M580\_PME\_UCM\_0312.cpx

PMEUCM VALVES Manual

Choose the appropriate file for the installed version of Unity Pro, then click 'Open'.



🔳 Import User De	vices		×
User devices			
PME_UCM			
Select All			
	UK	Uancel	

The Hardware Catalog Manager will show a dialog box displaying its progress.

Import User Dev	vices		×
User devices			
PME_UCM			
Select All			
	UK	Cancel	
		C. I. O.	

Hardware Catalog Manager	- 🗆 🗙		
File Edit View Service Help			
CANopen drop  CANopen drop  Constributed I/Os  Con	Build Catalog Abort Modifications <b>Close</b>		
Building Please wait when building the catalog			
Catalog building started at 14:12:12. Initializing database for update Process database Finalize database Build / Import/Export / Log / For Help, press F1			

When it is finished, it will appear as below.

+ Hardware Catalog Manager	- 🗆 🗙
<u>File Edit View Service H</u> elp	
CANopen drop	Build Catalog
Motor control Safety Sensors	Abort Modifications
Third party products	Close
port of catalog archive file C:\Niobrara\PMEUCM\DTM\M580_PME porting device PMEUCM0202 atalog archive import done	E_UCM_0302_for_V11
<     Huild A Import/Export / Log /	>

Close the Hardware Catalog Manager, and start Unity Pro. The PMEUCM0302 can now be chosen from the Hardware Catalog under the "Third party products" section.

🛺 PLC bu			
Bus:	0 BME P58 1020 02.10	×	
0	CPS 0 2000 1020 New Device		×
	Topological Address:	0.2	OK Cancel
∥ ⊥	Part Number	Description	Help
	Modicon M580 local drop		
	Analog		
	Communication		
	Counting		
	Discrete		
	Motion		
	Third party products		
	PME SWT 0100	Partner Module Ethemet System Weighing Transmitter 1 channel	
	PME OCH DELL	Partner Module Ethemet UCM	
	PME UCM 0302	User Programmable Module for customer Serial and Ethemet networks	

## 6 NRD PTK DTM UTIL

The NRD DTM Tool is installed by the PMEUCM\_SETUP.EXE program. The user may access this file at: http://www.niobrara.com/programs/PMEUCM\_SETUP.EXE,

#### **Open the NRD DTM Tool**

The next step is to open the Niobrara DTM Tool. Select Programs > Niobrara > PMEUCM > DTM > DTM Utility.

NOTE: The DTM Tool must be at least revision 22FEB2018 to operate with the PMEUCM0302 or PMEUCM0312.

Elle Help         Image: PME SWT 0100         NRDDDXMLUTIL.EXE         Rev. 22FEB2018         Niobrara R&D Corp.         Joplin, M0 USA         Technical Support at +1 417-624-8918         www.niobrara.com         This program is used to convert comma separated formated files into the DDXML file used by Schneider-Electric Unity Pro for use with the PMEUCM0302 module.         The DTM tree on the left is taken from the PME DDXML_Reposito in Unity Pro.         It is necessary to 'Update' the DTM section of the Hardware Cata in Unity Pro after any modification.	INRD PTK DTM Too Key. 22FEB2018	– 🗆 X
PME SWT 0100       NRDDDXMLUTIL.EXE         Rev. 22FEB2018       Niobrara R&D Corp.         Joplin, MO USA       Technical Support at +1 417-624-8918         www.niobrara.com       This program is used to convert comma separated formated files into the DDXML file used by Schneider-Electric Unity Pro for use with the PMEUCM0302 module.         The DTM tree on the left is taken from the PME DDXML_Reposito in Unity Pro.         It is necessary to 'Update' the DTM section of the Hardware Cata in Unity Pro after any modification.	<u>F</u> ile <u>H</u> elp	
		NRDDDXMLUTIL.EXE         Rev. 22FEB2018         Niobrara R&D Corp.         Joplin, MO USA         Technical Support at +1 417-624-8918         www.niobrara.com         This program is used to convert comma separated formated files into the DDXML file used by Schneider-Electric Unity Pro for use with the PMEUCM0302 module.         The DTM tree on the left is taken from the PME DDXML_Reposito in Unity Pro.         It is necessary to 'Update' the DTM section of the Hardware Cata in Unity Pro after any modification.
< >		< >

The tree on the left of the screen shows the PTK DTMs installed in Unity Pro. In this case, it shows the PME SWT 0100 Weighing Module from SCAIME.

### Installing a new file

Select File > "Install new .txt..." and then browse to the location where the txt file is located and select the file to install:

"c:\Niobrara\apps\PMEUCM\VALVES\PME UCM 0302 VALVES v1 01.txt"

**NOTE**: The file version number in this example is 1.14 but may be different for a newer version of the setup file. If multiple versions of the .txt file are present, it is normally advised to choose the highest version numbered file.

		NRD PTK DTM Tool Rev. 18JAN2017		- 🗆
	File	Help		
Install new. bd Remove from Repository Restore after DTM upgrade Rebuild .bt from Unity Pro .XSY Exit		PME SW1 0100	NRDDDXMLUTIL.EXE Rev. 18JAN2017 Niobrara R&D Corp. Joplin, MO USA Technical Support at +1 417-624-8918 www.niobrara.com This program is used to convert comma sep into the DDXML file used by Schneider-Elect with the PMEUCM0302 module. The DTM tree on the left is taken from the PM It is necessary to 'Update' the DTM section	arated form ric Unity Pr ME DDXML, of the Hard

NR&D DDXML Utility		×
$\leftarrow \rightarrow \cdot \uparrow$	« Niobrara > apps > PMEUCM > VALVES	Search VALVES
Organize 🔻 Ne	v folder	III 🔹 🕶 🚺 💡
Documents	^ Name	Date modified Type Size
👆 Downloads	PME UCM 03X2_VALVES_v1_01.txt	4/1/2018 8:54 PM Text Document 12 KB
👌 Music		
Pictures		
Videos		
骗 Windows (C:)	¥	
	File name: PME UCM 03X2_VALVES_v1_01.txt	✓ Text Files, (*.TXT) ✓
		<u>O</u> pen ▼ Cancel

After selecting "Open", the main screen should now change to show a new entry in the tree.

Status information is displayed on the right side of the screen. If there is an error during the compile, the error description and source code line number will be

PMEUCM VALVES Manual

#### displayed.



Clicking on the + will expand the VALVES structure to show an overview of the PLC Inputs and Outputs.

INRD PTK DTM Tool Rev. 22FEB2018	-	×
<u>File</u> <u>H</u> elp		
□□□□         PAKE SVM 70100           □□□         PAKE UCM 0322_VALVES_V1_01           □□□         □□           □□         □□           □□         □           □□         □           □□         □           □□         □           □□         □           □□         □           □□         □           □□         □           □□         □           □□         □           □□         □           □□         □           □□         □           □         □           □         □           □         □           □         □           □         □           □         □           □         □           □         □           □         □           □         □           □         □           □         □           □         □           □         □           □         □           □         □           □         □           □         □ <td>NRDDDXMLUTIL.EXE Rev. 22FEB2018 Niobrara R&amp;D Corp. Joplin, MO USA Technical Support at +1 417-624-8918 www.niobrara.com</td> <td>Â</td>	NRDDDXMLUTIL.EXE Rev. 22FEB2018 Niobrara R&D Corp. Joplin, MO USA Technical Support at +1 417-624-8918 www.niobrara.com	Â
	<	→

The version number here is shown to be 01.01. This version number must agree with the VALVES application running in the PMEUCM.

Expanding the tree further reveals the Ethernet/IP Assemblies configured for this application.

The total byte counts for PLC Inputs and Outputs are 120 and 87 bytes respectively.

The Ethernet/IP RPI is defaulted to 15mS updates.

Expanding the Inputs reveals the variable passed from the UCM to the M580 CPU.

File       Help         Image: Contract of the system of the s
Last_Good_Slave_S1, birte, 1245, 2      Last_Good_Slave, S2, BYTE, 1244, 3      Last_Good_Slave, BYTE, 1245, 4      S1_Polling_Slave, BYTE, 1246, 5      Lot_Landshake_Echo, BYTE, 1247, 6      Lin_Handshake, BYTE, 1248, 7      Lin_Slave, ARRAY[07] of BYTE, 1249, 8      Lin_Slave, ARRAY[07] of WORD, 1257, 16      Lin_Position, ARRAY[07] of INT, 1265, 32      Output Byte Count = 61 (87)      Last_Good_Slave_S2, BYTE, 102, Instance = x271C, File = 58201, RPI = 15, Byte Count = 61 (87)

Expanding the Outputs shows the variables from the CPU to the UCM.



## 7 System Operation

The DTM for the VALVES application defines the variables used for transferring data between the M580 CPU and the PMEUCM. Some of the variables are passed on every transaction between the CPU and the UCM (UCM\_Command, UCM\_Status, Last\_Good\_Slave\_S1, etc.) The actual data that needs to be sent or received on for all of the possible valves is too large to be sent each DIO transaction. A window scheme is used for this data and a DFB is provided to automatically transfer the actuator data to/from the UCM through the data window. The DFB uses a global structure variable to provide the setpoint, position, and status for all possible valves on a given UCM.

### Valve DDT

A Device Data Type variable structure called 'Valve' is used in the PLC for the data for each actuator in the field.

Name	Туре	✓ Comment ✓
🖃 🗇 Valve	<struct></struct>	
- Senable	BOOL	
🔶 Status	WORD	
	BOOL	
FullOpen	BOOL	
I FullClose	BOOL	
Stopped	BOOL	
🛶 Opening	BOOL	
- 🕒 Closing	BOOL	
Jammed	BOOL	
NotIn Remote	BOOL	
CombinedFault	BOOL	
I OverTemp	BOOL	
	BOOL	
NetFaultB	BOOL	
- Solar Sola	BOOL	
🕒 CloseTorqueFault	BOOL	
ValveOpManually	BOOL	
PhaseFault	BOOL	
Osition	INT	0=Closed, 100=Open
🔶 Setpoint	INT	19200=Closed, 19250 = 50%, 19300=Open
V_Type	BYTE	1=Limitorque, 2=EIM, 3=Rotork, 4=Auma
Setpoin Tto Send	BOOL	
<ul> <li>OldSetpoint</li> </ul>	INT	
<b>b</b>		

- Enable Bool. Read/Write. Set this to TRUE on valves that are in service.
- Status Word. Read-Only. Bit-map status of the valve. Same for all valve types.
  - Online Bool. Read-Only. 1=Online, 0=Offline
  - FullOpen Bool. Read-Only. 1=Valve at full open limit. 0=Not at full open.
  - FullClose Bool. Read-Only. 1=Valve at full close limit. 0=Not full close.

- Stopped Bool. Read-Only. 1=Valve not moving and not full open or full close. 0=Valve moving or full open or full close.
- Opening Bool. Read-Only. 1=Valve moving in direction of full open. 0=Valve not moving in direction of full open. May be closing or stopped.
- Closing Bool. Read-Only. 1=Valve moving in direction of full close. 0=Valve not moving in direction of full close. May be opening or stopped.
- Jammed Bool. Read-Only. 1=Valve is jammed (unexpected stop).
   0=Valve is not jammed.
- NotInRemote Bool. Read-Only. 1=Valve is not in Remote Control.
   0=Valve in remote control mode.
- CombinedFault Bool. Read-Only. 1=Some fault condition occurring. This fault is an 'or' of several fault conditions. 0=No faults.
- OverTemp Bool. Read-Only. 1=Valve motor is faulted with Overtemperature. 0=Not overtemperature.
- NetFaultA Bool. Read-Only. 1=Valve serial port 'A' has timed-out (not received a messages within 1 minute). 0=serial communication on port 'A' is good.
- NetFaultB Bool. Read-Only. 1=Valve serial port 'B' has timed-out (not received a messages within 1 minute). 0=serial communication on port 'B' is good.
- OpenTorqueFault Bool. Read-Only. 1=Valve faulted while attempting to open.
- CloseTorqueFault Bool. Read-Only. 1=Valve faulted while attempting to close.
- ValveOpManually Bool. Read-Only. 1=Valve position has been altered manually.
- PhaseFault Bool. Read-Only. 1=Electrical phase problem occurred.
- Position Integer. Read-Only. Percent open Position of the valve 0-100. 0=Full close. 100=Full Open.
- Setpoint Integer. Read/Write. Psudo-bitmap control and position setting.
  - $\circ$  0 = Ignored. No action.
  - $\circ$  256 = Full Open. Valve to move to full open position.
  - $\circ$  512 = Stop. Valve to stop immediately.
  - $\circ$  768 = Full Close. Valve to move to full close position.
  - 1280 = Start ESD. Valve to begin Emergency Shutdown Procedure.

Valve will change to the default ESD position depending upon configuration.

- $\circ$  1536 = Stop ESD. Valve to end Emergency Shutdown Procedure. Action depends upon valve configuration.
- 19200 19300 = Move to position 0-100 where 19200 = Full Close and 19300 = Full Open. 19250 = 50% open.
- V\_Type Integer. Read/Write. Determines the type of valve for a given slave.
  - $\circ$  0=Disabled
  - 1=Limitorque
  - 2=EIM
  - 3=Rotork
  - 4=Auma
- SetpointToSend Bool. Read/Write. Used by the Actuator DFB to queue up a new setpoint to transfer to the UCM.
- OldSetpoint Integer. Read/Write. Used by the Valves DFB to queue new setpoint for transfer to the UCM.

### Valve\_Data DDT

A Device Data Type called 'Valve\_Data' is used by each UCM in the PLC to store all of the status/setupoints for each possible valve. This structure is simply an 'array[1-255] of Valve'.

Name	-	Туре	Comment •
	<b>b</b>	1	
	Valve_Data	ARRAY[1255] OF Valve	
Ē	- 🗐 Valve_Data[1]	Valve	
	- 🔶 Enable	BOOL	
	🔶 Status	WORD	
	- 🔶 Online	BOOL	
	- 🐤 FullOpen	BOOL	
	- 🔶 FullClose	BOOL	
	- Stopped	BOOL	
	Opening	BOOL	
	Closing	BOOL	
	🔶 Jammed	BOOL	
	NotIn Remote	BOOL	
	CombinedFault	BOOL	
	OverTemp	BOOL	
	🔶 NetFaultA	BOOL	
	NetFaultB	BOOL	
	Open Torque Fault	BOOL	
	CloseTorqueFault	BOOL	
	ValveOp Manually	BOOL	
	PhaseFault	BOOL	
	- Osition	INT	0=Closed, 100=Open
	- Setpoint	INT	19200=Closed, 19250 = 50%, 19300=Open
	V_Type	BYTE	1=Limitorque, 2=EIM, 3=Rotork, 4=Auma
	Setpoin Tto Send	BOOL	
	OldSetpoint	INT	
÷	- 🗐 Valve_Data[2]	Valve	
÷	🗐 Valve_Data[3]	Valve	
÷	🗐 Valve_Data[4]	Valve	
	A Value Data (C)	Mahaa	

### Valve\_Map DDT

The final Device Data Type is called 'Valve\_Map' which is used by the Valves DFB to inform the UCM of the configured 'types' of actuators (Limitore, EIM, etc.) It is simply an array of bytes and not normally accessed by the user code.

🖃 🕘 Valve_Map	ARRAY[031] OF BYTE
🔶 Valve_Map[0]	BYTE
🔶 Valve_Map[1]	BYTE
🔶 Valve_Map[2]	BYTE
Valve_Map[3]	BYTE
🔶 Valve_Map[4]	BYTE
Valve_Map[5]	BYTE
🔶 Valve_Map[6]	BYTE
🔶 Valve_Map[7]	BYTE
🔶 Valve_Map[8]	BYTE
A Value Mael01	DVTC

### **Outbound Operations**

In this example, assume that socket 3 is already connected to a remote server (or client).

The M580 needs to send a message on socket 3 of 110 bytes. The following sequence would be followed:

- 1. The PLC code would wait until UCM1\_Data\_OUT\_Length[3] = 0. If this length is non-zero then the last transmitted frame is not finished being moved to the UCM. Wait for this value to become zero before loading new data into UCM1\_Data\_OUT[3].
- 2. Copy the 110 bytes that need to be transmitted into

UCM1\_Data\_OUT[3][0] through UCM1\_Data\_OUT[3][109].

- 3. Set the new length value UCM1\_Data\_OUT\_Length[3] := 110.
- 4. The DFB VALVES\_Outbound watches for UCM1\_Data\_OUT\_Length to be > 0 and will then transfer the new data to the UCM across the backplane. When the data is completely moved to the UCM, the DTM will zero UCM1\_Data\_OUT\_Length[3] so the process may start again.

### **Inbound Operations**

As in the above example, Socket 3 is already connected and 55 bytes will be received by the UCM from the remote end of the connection.

- 1. The PLC code will wait until UCM1\_Data\_IN\_Length[3] > 0. The DFB VALVES\_Inbound will set the value to be greater than zero when the complete frame data is ready to be used by the PLC program. In this case, the length value will be set to 55 bytes.
- 2. The PLC code will then parse the incoming message as contained in UCM1\_Data\_IN[3][0] through UCM1\_Data\_IN[3][54].

PMEUCM VALVES Manual

3. When the PLC code is finished pulling the data out of UCM1\_Data\_IN[3] then it will set UCM1\_Data\_IN\_Length[3] := 0.

NOTE: Inbound and Outbound length values must be within the range of 1-1452.

### VALVES DTM Variables

The VALVES DTM provides the variables used in the Unity program to monitor and control the operation of the VALVES PMEUCM application.

NOTE: Many of these variables are used by the Inbound and Outbound DFBs and should not be altered elsewhere in the PLC application.

Name 👻	Value	Туре 👻	Comment
		T_UCM1	
Freshness	1	BOOL	Global Freshness
Freshness_1	1	BOOL	Freshness of Object
🗈 🗇 Inputs		T_UCM1_IN	Input Variables
		T_UCM1_OUT	Output Variables

#### Freshness

A read-only BOOL variable named 'Freshness' is provided to report the status of the data connection between the DIO Master (M580 CPU or eNOC) and the PMEUCM across the Ethernet backplane. Freshness becomes TRUE when all of the Ethernet/IP connections between the M580 and the PTK board are active and passing data.

It is good policy to use the state of 'Freshness' to control the logic relating to the VALVES application in the M580.

if PME\_UCM\_0302\_VALVES.Freshness then (\* process the UCM data \*) end\_if;

There will be at least one additional 'Freshness\_1' BOOL variables present which will provide the status of each Ethernet/IP connection to the UCM. The standard PME UCM 0302\_VALVES DTM only uses a single Ethernet/IP connection so it will only show Freshness\_1.

The 'Freshness' variable will only be TRUE if all of the individual 'Freshness\_1' and 'Freshness\_2' variables are also true.

PME_UCM_0202_TCPOPENw1		T_PME_UCM_0202_TCPO	
🐤 Freshness	1	BOOL	Global Freshness
🔶 Freshness_1	1	BOOL	Freshness of Object
🐤 Freshness_2	1	BOOL	Freshness of Object
🗄 🗐 Inputs		T_PME_UCM_0202_TCPO	Input Variables
🗄 🖅 🗾 Outputs		T_PME_UCM_0202_TCPO	Output Variables
<b>b</b>			

#### Inputs

The PLC Input data is read-only and includes a block of 'overhead' data required by the PME Generic DTM provided by Schneider-Electric. This data may is normally ignored in the M580 project.

NOTE: The Device\_Name does give the current version of the running VALVES application '09MAY2017'.

Name 🔹	Value	Туре 💌	Comment
	1	T_UCM1	1
	1	BOOL	Global Freshness
	1	BOOL	Freshness of Object
E. Inputs		T_UCM1_IN	Input Variables
DEVICE_NAME	'TCPOpen 09MAY201'	string[64]	
DEVICE_STATUS	3	UINT	
	1	BOOL	
DEVICE_HEALTH	1	BOOL	
SPI_ERROR	0	BOOL	
INIT_PARAM_CORRUP	0	BOOL	
	0	BOOL	
CONTROL_FW_MISSIN	0	BOOL	
CONTROL_FW_DOWNL	0	BOOL	
🔶 Free0	0	BYTE	Unused Variable
Free 1	0	BYTE	Unused Variable
ETH_STATUS	129	BYTE	
PORT1_LINK	1	BOOL	
CCOTF_IN_PROGRESS	0	BOOL	
	0	BOOL	
GLOBAL_STATUS	1	BOOL	
SERVICE_STATUS	16	BYTE	
SNTP_SERVICE	0	BOOL	
SNMP_SERVICE	0	BOOL	
FDR_SERVICE_B1	1	BOOL	
FDR_SERVICE_B2	0	BOOL	
FDR_SERVICE_B3	0	BOOL	
FDR_SERVICE_B4	0	BOOL	
ETH_PORT1_INFO	1	BYTE	
	1	BOOL	
ETH_PORT1_FUNCTIO,	0	BOOL	
🔶 Free2	0	BYTE	Unused Variable
UCM_Runtime_Status	16#C000	WORD	.15=Run; LSB=Runtime
	0	UINT	UCM Runtime Error Hal
UCM_Error	0	INT	0=No Error; 1=Wrong D
UCM_Status	2#0000_0000_0001_0001	WORD	.0=E1 LinkOK; .1=Dup
SI_Remote_IP		ARRAY[015] OF DWORD	Socket IP Address
SI_Remote_Port		ARRAY[015] OF UINT	TCP or UDP Remote P
		ARRAY[015] OF UINT	TCP or UDP Local Port
🕀 📕 SI_Status		ARRAY[015] OF BYTE	.0=Connect; .1=1/2 clo
SI_Out_Handshake_W0	3841	INT	Outbound handshake
SI_In_Handshake_W0	3842	INT	Inbound handshake
Image: SI_Number_W0		ARRAY[07] OF BYTE	Socket Number for Dat

#### PMEUCM VALVES Manual

The actual UCM provided data starts with UCM\_Runtime\_Status.

### UCM\_Runtime\_Status

WORD – The UCM\_Runtime\_Status provides an indication that the VALVES application is running properly in the UCM. The bits of this word may be monitored. This value is best viewed in hexadecimal.

Bits	Meaning	Notes
Bit 15	1=Running 0=Halted	Normally bit 15 is ON.
Bit 14	1=Module Configured and Running Normal 0=Module not fully configured	Normally bit 14 is ON.
0-13	Last Halt Error condition	See table below

Code	Description	
C0xx	Application Running, if xx nonzero, xx=last halting error (in hex)	
0	Terminated by clearing all thread run bits	
1	STOP statement executed	
2	Illegal instruction exception	
3	Division by Zero	
4	Out of heap space for ON CHANGE	
5	Out of heap space for ON RECEIVE	
6	Unsupported run-time call, likely compiler/firmware mismatch	
7	Parameter or array index out of range	
8	Downloaded code corrupt, CRC Error	
9	CPU Address exception	
10	Stack Underflow	
11	TCP Error -1, likely compiler/firmware mismatch	
12	TCP Error -2, contact Niobrara	
13	TCP Error -3, not enough sockets or buffers, See register 66. Also IP address or gateway not initialized	
14	Hardware not authorized to run user code	

Free2	0	BYTE
UCM_Runtime_Status	16#C000	WORD
UCM_Halt_Line_Number	0	UINT

In the above screenshot, the UCM\_Runtime\_Status = 16#C000 which is the normal value.

### UCM\_Halt\_Line\_Number

UINT – The Halt Line Number value provides the source code line number where the most recent runtime halting error has occurred. This value should be zero during normal operation. Contact Niobrara Technical Support if this value is non-zero.

### UCM\_Error

INT – The UCM\_Error value provides an indication that the VALVES application has an issue with configuration.

Value	Meaning	Notes
0	No Configuration Errors	
1	PLC Modbus/TCP Connection Error	M580 May have TCP Port 502 Security Enabled.
2,3	Reserved	
4	Bad DTM Filename	
4	Bad DLL Version	
5	Bad DTM File Version	
8	Bad Assembly Size	
9	Reserved	
10	Bad E1 IP Address	Reverts to 10.10.10.10
11	Bad E1 Subnet Mask	Reverts to 255.0.0.0
12	Bad E1 Default Gateway	Reverts to 0.0.0.0
13	Bad E2 IP Address	Reverts to 10.10.10.11
14	Bad E2 Subnet Mask	Reverts to 255.0.0.0
15	Bad E2 Default Gateway	Reverts to 0.0.0.0
16,17	Reserved	
18	PLC in STOP	
19	Reserved	
20	No Link on UCM Backplane	
21	Duplicate IP on BP	
22-26	Reserved	
27	Watchdog Expired	
28	Bad DIO Ch Count	
29	Bad DIO Output Count	
30	Bad DIO Input Count	
31	HSBY FDR Do NOT Match	

### UCM\_Status

WORD – The UCM\_Status provides a bit-mapped indication of conditions in the module. Presently, the first 6 bits provide indication of the three Ethernet ports.
Bit	Meaning	Notes
0 (lsb)	1 = E1  Link OK 0 = E1 Link off	
1	1 = E1 duplicate IP Address 0 = E1 not in duplicate IP	LCD displays offending MAC
2	1 = E2 Link OK 0 = E2 Link off	
3	1 = E2 duplicate IP Address 0 = E2 not in duplicate IP	LCD displays offending MAC
4	1 = BP Link OK 0 = BP Link Off	
5	1 = BP duplicate IP Address 0 = BP not in duplicate IP	LCD displays offending MAC
6-15	Reserved	

UCM_Error	0	INT
UCM_Status	2#0000_0000_0001_1101	WORD
		ARRAY[015

In this screenshot, the following bits are TRUE:

- Bit 0 E1 Link is ON
- Bit 2 E2 Link is ON
- Bit 3 E2 is in Duplicate IP Address
- Bit 4 Backplane Link is ON

## SI\_Remote\_IP

ARRAY[0..15] of DWORD – This 16 variables show the IP Address of the remote end of a connection for each socket compressed into a DWORD. Each 8 bit BYTE of the DWORD is the octet of the IP Address.

For example, if the remote IP Address for socket 1 is 192.168.0.111 then

 $SI_Remote_{IP[1]} = 3232235631(dec) = 16\#C0A8_006F(hex)$ 

Looking at the value in hexadecimal reveals the IP Address:

C0 = 192A8 = 168 00 = 06F = 111

A simple method for moving this data to BYTE variables is to use the 'Shift Right' (SHR) ST command. For example, the program includes an array of bytes RemIP[0..3]. The following code would move the Remote IP address of socket 3

PMEUCM VALVES Manual

into this array.

RemIP[0] := SHR(UCM1.Inputs.SI\_Remote\_IP[3],24); RemIP[1] := SHR(UCM1.Inputs.SI\_Remote\_IP[3],16); RemIP[2] := SHR(UCM1.Inputs.SI\_Remote\_IP[3],8);

RemIP[3] := SHR(UCM1.Inputs.SI\_Remote\_IP[3],0);

The first SRH shifts the DWORD 24 bits and loads the BYTE variable. The next lines shift 16 bits and 8 bits. The last line doesn't shift at all and could just be a straight assignment.

UCM_Status	2#0000_0000_0001_0101	WORD
SI_Remote_IP		ARRAY[015] OF DWORD
SI_Remote_IP[0]	0	DWORD
SI_Remote_IP[1]	0	DWORD
SI_Remote_IP[2]	2886729928	DWORD
SI_Remote_IP[3]	0	DWORD
SI_Remote_IP[4]	2886729739	DWORD
SI_Remote_IP[5]	0	DWORD
SI_Remote_IP[6]	0	DWORD
SI_Remote_IP[7]	0	DWORD
SI_Remote_IP[8]	2886729738	DWORD
SI_Remote_IP[9]	0	DWORD
SI_Remote_IP[10]	0	DWORD
SI_Remote_IP[11]	0	DWORD
SI_Remote_IP[12]	0	DWORD
SI_Remote_IP[13]	0	DWORD
SI_Remote_IP[14]	0	DWORD
SI_Remote_IP[15]	0	DWORD
SI Remote Port		ARRAYIO 151 OF LIINT

The above screenshot shows three active socket connections on sockets 2, 4, and 8.

- SI\_Remote\_IP[2] = 2886729928 (dec) = AC1000C8 (hex)
- SI\_Remote\_IP[4] = 2886729739 (dec) = AC10000B (hex)
- SI\_Remote\_IP[8] = 2886729838 (dec) = AC10000A (hex)

Splitting this data by bytes shows the remote IP Address: 172.16.0.200 for socket 2.

- AC (hex) = 172 (decimal)
- 10 (hex) = 16 (decimal)
- 00 (hex) = 0 (decimal)
- C8 (hex) = 200 (decimal)

Socket 4 has a remote IP Address of 172.16.0.11.

• AC (hex) = 172 (decimal)

- 10 (hex) = 16 (decimal)
- 00 (hex) = 0 (decimal)
- 0B (hex) = 11 (decimal)

Socket 8 has a remote IP Address of 172.16.0.10.

- AC (hex) = 172 (decimal)
- 10 (hex) = 16 (decimal)
- 00 (hex) = 0 (decimal)
- 0A(hex) = 10 (decimal)

### SI\_Remote\_Port

ARRAY[0..15] of UINT – These 16 variables show the TCP (or UDP) port number of the remote end of a connection for each socket.

	-	
🖃 📕 SI_Remote_Port		ARRAY[015] OF UINT
SI_Remote_Port[0]	0	UINT
SI_Remote_Port[1]	0	UINT
SI_Remote_Port[2]	61633	UINT
SI_Remote_Port[3]	0	UINT
SI_Remote_Port[4]	2816	UINT
SI_Remote_Port[5]	0	UINT
SI_Remote_Port[6]	0	UINT
SI_Remote_Port[7]	0	UINT
SI_Remote_Port[8]	502	UINT
SI_Remote_Port[9]	0	UINT
SI_Remote_Port[10]	0	UINT
SI_Remote_Port[11]	0	UINT
SI_Remote_Port[12]	0	UINT
SI_Remote_Port[13]	0	UINT
SI_Remote_Port[14]	0	UINT
SI_Remote_Port[15]	0	UINT
		ADDAMO AD OF LINT

The screen shot above shows socket 2 connected to remote port 61633 while socket 4 is connected to remote port 2816 and socket 8 on port 502.

### SI\_Local\_Port

ARRAY[0..15] of UINT – These 16 variables show the TCP (or UDP) port number of the local end of a connection for each socket.

m 🖬 orTuculoreTucu		Anna de la companya d
🚊 🛯 SI_Local_Port		ARRAY[015] OF UINT
SI_Local_Port[0]	0	UINT
SI_Local_Port[1]	0	UINT
SI_Local_Port[2]	502	UINT
SI_Local_Port[3]	0	UINT
SI_Local_Port[4]	502	UINT
SI_Local_Port[5]	0	UINT
	0	UINT
SI_Local_Port[7]	0	UINT
SI_Local_Port[8]	2816	UINT
SI_Local_Port[9]	0	UINT
SI_Local_Port[10]	0	UINT
SI_Local_Port[11]	0	UINT
SI_Local_Port[12]	0	UINT
SI_Local_Port[13]	0	UINT
SI_Local_Port[14]	0	UINT
SI_Local_Port[15]	0	UINT

The screen shot below shows sockets 0 and 4 connected to local port 502 and socket 8 connected to port 2816.

## SI\_Status

ARRAY[0..15] of  $BYTE - The SI_Status$  provides a bit-mapped indication of condition of each socket.

Bit	Meaning	Notes
0	1 = Connection ACTIVE 0 = No connection	
1	$1 = \frac{1}{2} \text{ closed}$ 0 = normal	<sup>1</sup> / <sub>2</sub> closed means a FIN was received from the remote end but the socket has not been formally closed.
2	1 = Server Listening (or Client trying to connect) 0 = Not listening or trying	
3-7	Reserved	

	SI_Status		ARRAY[015] OF BYTE
	SI_Status[0]	4	BYTE
	SI_Status[1]	4	BYTE
	SI_Status[2]	1	BYTE
	SI_Status[3]	4	BYTE
	SI_Status[4]	1	BYTE
	SI_Status[5]	4	BYTE
	SI_Status[6]	4	BYTE
(	SI_Status[7]	4	BYTE
	SI_Status[8]	1	BYTE
	SI_Status[9]	0	BYTE
	SI_Status[10]	0	BYTE
	SI_Status[11]	0	BYTE
	SI_Status[12]	0	BYTE
	SI_Status[13]	0	BYTE
	SI_Status[14]	0	BYTE
	SI_Status[15]	0	BYTE
	01.0 - 11 - 11 - 14/0	0500	INIT

The screenshot above shows sockets 2, 4, and 8 connected (value = 1) while sockets 0, 1, 3, 5, 6, and 7 are listening (value = 4). Sockets 9-15 are not enabled.

### SI\_Out\_Handshake\_W0

**NOTE**: This value is used by the VALVES\_Outbound DFB and should not be used elsewhere in the PLC program.

INT – This is the feedback handshake value from the UCM to acknowledge the reception of an SO\_Data\_W0 block. The UCM echos the value of the SO\_Out\_Handshake\_W0 to the SI\_Out\_Handshake\_W0 after parsing the data block. VALVES\_Outbound DFB code watches this value and when the outbound and inbound handshake values are equal, the outbound window may be used for the next transfer of data.

### SI\_In\_Handshake\_W0

**NOTE**: This value is used by the VALVES\_Inbound DFB and should not be used elsewhere in the PLC program.

INT – This is the handshake value from the UCM to indicate new data is present in the SI\_Data\_W0 block. The VALVES\_Inbound\_DFB compares this value with the SO\_In\_Handshake\_W0 value and if they are different then it can process the new data block. After retrieving the data block, the PLC echos the value of the SI\_In\_Handshake\_W0 to the SO\_In\_Handshake\_W0 and the UCM seeing that the Out value matches the In value is allowed to place new data into the In block.

### SI\_Number\_W0

**NOTE**: This value is used by the VALVES\_Inbound DFB and should not be used elsewhere in the PLC program.

ARRAY[0..7] of BYTE – This array holds the socket number for each of the possible 8 blocks of inbound data in the SI\_Data\_W0 window. Valid numbers are 0 through 15. The UCM places a 255 value in unused block location. The VALVES\_Inbound DFB code will look at this array to know where to place the incoming socket data.

### SI\_Length\_W0

**NOTE**: This value is used by the VALVES\_Inbound DFB and should not be used elsewhere in the PLC program.

ARRAY[0..7] of INT – This array holds the length for each of the possible 8 blocks of inbound data in the SI\_Data\_W0 window. Valid lengths are 1 through 1134. The UCM places a 0 value in unused block location. The VALVES\_Inbound DFB code will look at this array to know how much of the data is to be placed in the appropriate array.

### SI\_More\_W0

**NOTE**: This value is used by the VALVES\_Inbound DFB and should not be used elsewhere in the PLC program.

BYTE – This byte is a bit-map of the 8 possible data blocks in the SI\_Data\_W0 window. If a bit in this byte is TRUE then the data for that particular socket is too large to transfer in the current SI\_Data\_W0. 'More' data is queued to be sent to the PLC in the next handshake transfer.

## SI\_Data\_W0

**NOTE**: This value is used by the VALVES\_Inbound DFB and should not be used elsewhere in the PLC program.

ARRAY[0..1133] OF BYTE – This is the data block for window W0 from the UCM to the PLC. Valid data is placed starting at element [0] of this block up to element [1133]. The data in this block may be from 1 to 8 different sockets.

The UCM loads a value of 252 (decimal) 16#FC (hex) into unused bytes in the window.

Name -	Value	Туре
SI_Out_Handshake_W0	-1282	INT
SI In Handshake W0	-1318	INT
SI Number W0	1	ARRAYID 71 OF BYTE
SI Number W0(0)	2	BYTE
SI_Number_W0[1]	0	BYTE
CI Number_W0[1]	255	DITE
SI_Number_W0[2]	200	BTIE
SI_Number_W0[3]	255	BAIE
SI_Number_W0[4]	255	BYTE
SI_Number_W0[5]	255	BYTE
SI Number W0[6]	255	BYTE
SI Number W0[7]	255	BYTE
CL Leasth W/D	200	ADDAVID TLOF INT
SI_Length_vv0	10	ARRATE
SI_Length_WU[U]	12	INI
SI_Length_W0[1]	33	INT
<ul> <li>SI_Length_W0[2]</li> </ul>	0	INT
SI_Length_W0[3]	0	INT
SI Length W0[4]	0	INT
SI Length W0[5]	0	INT
SI Length W0[6]	0	INT
SI_Length_W0[0]	0	INT
SI_Length_WU[/]	0	INT
SI_More_W0	0	BYIE
SI_Data_W0	1.02	ARRAY[01133] OF BYTE
SI_Data_W0[0]	28	BYTE
SI_Data_W0[1]	180	BYTE
SI Data W0[2]	0	BYTE
	0	BYTE
	0	DYTE
SI_Data_W0[4]	0	DITE
SI_Data_wu[5]	6	BTIE
SI_Data_W0[6]	2	BYIE
SI_Data_W0[7]	3	BYTE
SI_Data_W0[8]	0	BYTE
SI_Data_W0[9]	0	BYTE
SI Data W0[10]	0	BYTE
SI Data W0[11]	20	BYTE
SI Data W0[12]	12	BYTE
CI Data W0[12]	100	OVTE
SI_Data_W0[15]	lou	DITE
SI_Data_WU[14]	0	BTIE
SI_Data_W0[15]	0	BYTE
SI_Data_W0[16]	0	BYTE
<ul> <li>SI_Data_W0[17]</li> </ul>	27	BYTE
SI_Data_W0[18]	3	BYTE
SI Data W0[19]	3	BYTE
SI Data W0[20]	24	BYTE
SI Data W0[21]	0	BYTE
	0	DVTE
	24	DITE
SI_Data_WU[23]	24	BTIE
SI_Data_W0[24]	32	BYTE
SI_Data_W0[25]	0	BYTE
SI_Data_W0[26]	0	BYTE
- SI Data W0[27]	0	BYTE
SI Data W0[28]	0	BYTE
	A	BYTE
	60	OVTE
SI_Data_W0[30]	60	BITE
SI_Data_W0[31]	0	BYIE
SI_Data_W0[32]	0	BYTE
SI_Data_W0[33]	0	BYTE
SI_Data_W0[34]	0	BYTE
SI Data W0[35]	0	BYTE
SI Data W0[36]	0	BYTE
	0	BYTE
	0	DYTE
J_Jata_Wu[38]	0	DITE
SI_Data_W0[39]	0	BYIE
SI_Data_W0[40]	0	BYTE
SI_Data_W0[41]	0	BYTE
SI_Data_W0[42]	0	BYTE
- SI Data W0[43]	0	BYTE
SI Data W0(44)	0	BYTE
SI Deta W0[45]	252	BYTE
	262	eration 4
JI Data WUI401	6.76	DITE

The above screenshot shows data from sockets 2 and 8 placed into the SI\_Data\_W0 window. The references for socket 8 are highlighted.

The data from socket 2 has a length of 12 bytes. The data for socket 2 is located in the first 12 bytes of the SI\_Data\_W0 array (SI\_Data\_W0[0]..[11]).

The data from socket 8 has a length of 33 bytes. The data for this socket is in the next 33 bytes after the data from socket 2 (SI\_Data\_W0[12]...[44]).

Note: SI\_More\_W0 = 0 which tells the DFB that this is the complete socket frame data for all entries in the list.

### **Outputs**

#### UCM\_Command

WORD – This is a bit-mapped command to temporally override operation of the VALVES application. These settings are normally controlled by the DTM

Bits	Meaning	Notes
0	1 = Force on OS MBTCP Server 0 = Use DTM Setting	Uses TCP port 503 when ON.
1	1 = Force on Web Server 0 = Use DTM Setting	Uses TCP port 81 when ON.
2	1 = Force on Telnet Server 0 = Use DTM Setting	Uses TCP port 24 when ON.
3	1 = Turn on Red LED behind Screen 0 = Turn off LED behind screen	
4	<ul><li>1 = Force a single read of Primary</li><li>PRN file from FDR server.</li><li>0 = Normal</li></ul>	Rising edge triggers a single read operation. Bit must be zeroed before triggered again.
5	1 = Force a single read of Secondary PRN file from HSBY Secondary FDR server. 0 = Normal	Rising edge triggers a single read operation. Bit must be zeroed before triggered again.
6	1 = Force a single write of Primary PRN file into Secondary HSBY FDR server.	Rising edge triggers a single write. Bit must be zeroed before triggered again. Write only occurs if Primary PRM is valid and secondary PRM has zero length or miss-matched checksum.

In the screenshot below, the DTM is configured to disable all three features. The PLC may set the three bits in the UCM\_Command word to override the DTM settings and enable the OS Modbus/TCP server (port 503), Debug Web Server (port 81), and Debug Telnet Server (port 24).

01.09				
ity Ethernet Configuration Application Configu	ration Pr	rocess Data		
Parameter Name		Current Value	Default Value	Unit
	1	255.0.0.0	255.0.0.0	
Default Gateway	0	0.0.0.0	0.0.0.0	
🖓 UCM OS Settings				
	[	Disabled	503	
	[	Disabled	81	
- Debug TELNET Server Port	[	Disabled	24	
	3	1452	1452	
	1	10	10	Seconds
Location Name (10 characters Max)				
🖓 S1 Serial Port				
Port 1 Mode	L	UCM OS RTU Slave	UCM OS RTU Slave	
Port 1 Baud Rate	9	9600	9600	BPS

## SO\_Remote\_IP

ARRAY[0..15] of DWORD – These 16 variables declare the target IP Address for a client connection on a socket compressed into a DWORD. Each 8 bit BYTE of the DWORD is the octet of the IP Address.

For example, if the remote IP Address for socket 3 is 206.223.51.16 then

SO\_Remote\_IP[1] = 3470734096(dec) = 16#CEDF\_3310(hex)

Looking at the value in hexadecimal reveals the IP Address:

CE (hex) = 206 (decimal)

DF (hex) = 223 (decimal)

33 (hex) = 51 (decimal)

10 (hex) = 16 (decimal)

A simple method for moving this data to the DWORD variable is to use the 'Shift

Left' (SHL) ST command. The following code would move the address 206.223.51.16 into the SO\_Remote\_IP of socket 12.

#### PME\_UCM\_0302\_VALVES.Outputs.SO\_Remote\_IP[12] := DINT\_TO\_DWORD(SHL(206,24)+SHL(223,16)+SHL(51,8)+SHL(16,0));

The first SRL shifts the 206 value 24 bits into the MSB of the DWORD. The next lines shift 16 bits and 8 bits. The last line doesn't shift at all and could just be a straight assignment. The DINT\_TO\_DWORD conversion is required because Unity treats the addition of the SHL of constants as a DINT.

	v	WOND
🚊 🛛 📕 SO_Remote_IP		ARRAY[015] OF DWORD
SO_Remote_IP[0]	0	DWORD
SO_Remote_IP[1]	0	DWORD
SO_Remote_IP[2]	0	DWORD
SO_Remote_IP[3]	0	DWORD
SO_Remote_IP[4]	0	DWORD
SO_Remote_IP[5]	0	DWORD
SO_Remote_IP[6]	0	DWORD
SO_Remote_IP[7]	0	DWORD
SO_Remote_IP[8]	2886729738	DWORD
SO_Remote_IP[9]	0	DWORD
SO_Remote_IP[10]	0	DWORD
SO_Remote_IP[11]	0	DWORD
SO_Remote_IP[12]	0	DWORD
SO_Remote_IP[13]	0	DWORD
SO_Remote_IP[14]	0	DWORD
SO_Remote_IP[15]	0	DWORD
SO_Remote_Port		ARRAY[015] OF UINT

The above screenshot shows socket 8 (as a client) connecting to remote IP Address 2886729738 (decimal) = AC10000A (hex) which indicates the remote IP Address of 172.16.0.10.

- AC (hex) = 172 (decimal)
- 10 (hex) = 16 (decimal)
- 00 (hex) = 0 (decimal)
- 0A(hex) = 10(decimal)

### SO\_Remote\_Port

ARRAY[0..15] of UINT – These 16 variables declare the target TCP (or UDP) port number for a client connection on a socket. This value is ignored for Server connections.

	-	
🖃 📕 SO_Remote_Port		ARRAY[015] OF UINT
SO_Remote_Port[0]	0	UINT
SO_Remote_Port[1]	0	UINT
SO_Remote_Port[2]	0	UINT
SO_Remote_Port[3]	0	UINT
SO_Remote_Port[4]	0	UINT
SO_Remote_Port[5]	0	UINT
SO_Remote_Port[6]	0	UINT
SO_Remote_Port[7]	0	UINT
SO_Remote_Port[8]	502	UINT
SO_Remote_Port[9]	0	UINT
SO_Remote_Port[10]	0	UINT
SO_Remote_Port[11]	0	UINT
SO_Remote_Port[12]	0	UINT
SO_Remote_Port[13]	0	UINT
SO_Remote_Port[14]	0	UINT
SO_Remote_Port[15]	0	UINT
📅 📕 CO Loopt Part		ADDAVIA 151 OF LINT

The above screenshot shows socket 8 targeting port 502.

### SO\_Local\_Port

ARRAY[0..15] of UINT – These 16 variables declare the local TCP (or UDP) port number for a server or client connection on a socket. A value of zero for a client connection allows the UCM OS to choose an appropriate ephemeral port number. The screenshot below shows sockets 0-7 listening on port 502 and sockets 8-15 are not configured.

🕮 🔲 SU_Remote_Port		AKKAT[U., 15] OF UINT	ICF or UDF Remote Port Number; Ignred during Server Operation
🚊 🛛 📘 SO_Local_Port		ARRAY[015] OF UINT	TCP or UDP Local Port Number; 0=Automatic for Client Operation
SO_Local_Port[0]	502	UINT	
SO_Local_Port[1]	502	UINT	
SO_Local_Port[2]	502	UINT	
SO_Local_Port[3]	502	UINT	
SO_Local_Port[4]	502	UINT	
SO_Local_Port[5]	502	UINT	
SO_Local_Port[6]	502	UINT	
SO_Local_Port[7]	502	UINT	
SO_Local_Port[8]	0	UINT	
SO_Local_Port[9]	0	UINT	
SO_Local_Port[10]	0	UINT	
SO_Local_Port[11]	0	UINT	
SO_Local_Port[12]	0	UINT	
SO_Local_Port[13]	0	UINT	
SO_Local_Port[14]	0	UINT	
SO_Local_Port[15]	0	UINT	
🗄 🛛 📕 SO_Command		ARRAY[015] OF BYTE	.0=Connect; .1=Reset; .2=Port2; .3=Client; .4=UDP; .5=No Nagle;
🗄 🛛 📕 SO_Socket_State		ARRAY[015] OF BYTE	Socket State for display on front LCD

## SO\_Command

ARRAY[0..15] of BYTE – These 16 bit-mapped variables control the sockets in the UCM.

Bit	Meaning	Notes
0	1 = Connect 0 = Close	Clearing bit sends RST on open connections.
1	1 = Reserved for Serial 0 = Use Ethernet	Current version ignores this bit. Leave this bit FALSE
2	1 = Use Port 2 0 = Use Port 1	
3	1 = Client 0 = Server	
4	1 = UDP $0 = TCP$	

🚊 🛛 📕 SO_Command		ARRAY[015] OF BYTE
SO_Command[0]	1	BYTE
SO_Command[1]	1	BYTE
SO_Command[2]	1	BYTE
SO_Command[3]	1	BYTE
SO_Command[4]	1	BYTE
SO_Command[5]	1	BYTE
SO_Command[6]	1	BYTE
SO_Command[7]	1	BYTE
SO_Command[8]	13	BYTE
SO_Command[9]	0	BYTE
SO_Command[10]	0	BYTE
SO_Command[11]	0	BYTE
SO_Command[12]	0	BYTE
SO_Command[13]	0	BYTE
SO_Command[14]	0	BYTE
SO_Command[15]	0	BYTE
All astedabard to O2	27265	INT

The above screenshot shows sockets 0 through 7 commanded to connect as a TCP server on E1.

- Bit 0 = 1 (Connect)
- Bit 1 = 0 (Ethernet)
- Bit 2 = 0 (E1)
- Bit 3 = 0 (Server)
- Bit 4 = 0 (TCP)

Socket 8 has a value of 13 (decimal) = 1101 (binary) which indicates a TCP client on E2.

• Bit 0 = 1 (Connect)

- Bit 1 = 0 (Ethernet)
- Bit 2 = 1 (E2)
- Bit 3 = 1 (Client)
- Bit 4 = 0 (TCP)

Further SO\_Command Examples:

To set socket 3 to be a TCP Server on E1 using window W0 with Nagle ON use the following commands to set the individual command bits:

UCM1.Outputs.SO\_Command.0 = TRUE; {Connect} UCM1.Outputs.SO\_Command.1 = FALSE; {Enet} UCM1.Outputs.SO\_Command.2 = FALSE; {E1} UCM1.Outputs.SO\_Command.3 = FALSE; {Server} UCM1.Outputs.SO\_Command.4 = FALSE; {TCP}

### SO\_Out\_Handshake\_W0

**NOTE**: This value is used by the VALVES\_Outbound DFB and should not be used elsewhere in the PLC program.

INT – This is the feedback handshake value from the PLC to indicate to the UCM that new data is included in the DIO transfer. The UCM echos the value of the SO\_Out\_Handshake\_W0 to the SI\_Out\_Handshake\_W0 after parsing the data block.

### SO\_In\_Handshake\_W0

**NOTE**: This value is used by the VALVES\_Inbound DFB and should not be used elsewhere in the PLC program.

INT – This is the handshake value from the PLC to indicate new data has been parsed from SI\_Data\_W0 block. The VALVES\_Inbound\_DFB copies the SI\_IN\_Handshake\_W0 value to this variable to indicate that the new data has been received.

### SO\_Number\_W0

**NOTE**: This value is used by the VALVES\_Outbound DFB and should not be used elsewhere in the PLC program.

ARRAY[0..7] of BYTE – This array holds the socket number for each of the possible 8 blocks of outbound data in the SO\_Data\_W0 window. Valid numbers are 0 through 15. The DFB places a 255 value in unused block location. The

UCM will look at this array to know where to transmit the outbound socket data.

#### SO\_Length\_W0

**NOTE**: This value is used by the VALVES\_Outbound DFB and should not be used elsewhere in the PLC program.

ARRAY[0..7] of INT – This array holds the length for each of the possible 8 blocks of inbound data in the SO\_Data\_W0 window. Valid lengths are 1 through 1180. The UCM places a 0 value in unused block location. The VALVES\_Inbound DFB code will look at this array to know how much of the data is to be placed in the appropriate array.

#### SO\_More\_W0

**NOTE**: This value is used by the VALVES\_Outbound DFB and should not be used elsewhere in the PLC program.

BYTE – This byte is a bit-map of the 8 possible data blocks in the SO\_Data\_W0 window. If a bit in this byte is TRUE then the data for that particular socket is too large to transfer in the current SO\_Data\_W0. 'More' data is queued to be sent to the PLC in the next handshake transfer.

#### SO\_Data\_W0

**NOTE**: This value is used by the VALVES\_Outbound DFB and should not be used elsewhere in the PLC program.

ARRAY[0..1179] OF BYTE – This is the data block for window W0 from the UCM to the PLC. Valid data is placed starting at element [0] of this block up to element [1179]. The data in this block may be from 1 to 8 different sockets.

Note: The DFB loads the value 253 (decimal) into unused bytes below the actual data.

Name	•	Value	Туре
- • SO_Ou	t_Handshake_W0	19733	INT
- • S0_h	Handshake_W0	19696	INT
E- SO_Nu	mber_W0		ARRAY[07] OF BYTE
- I SO	_Number_W0[0]	2	BYTE
- ISO - ISO	_Number_W0[1]	8	BYTE
- 🔶 SO	_Number_W0[2]	255	BYTE
- 🔶 SO	_Number_W0[3]	255	BYTE
90	Number W0[4]	255	BYTE
- 🔶 SO	Number W0[5]	255	BYTE
S0	Number W0[6]	255	BYTE
- • SO	Number W0[7]	255	BYTE
B- \$0 1e	anth W0		ARRAYID 7LOF INT
00_08	Leagth W000	49	INT
	Longh Worki	12	INT
50	Length W0[7]	0	INT
	Length_WU[2]	0	INT
- 50	Length_Wu[3]	0	INT
- • SU	Length_WU[4]	0	INT
- 50	_Length_W0[5]	0	INT
- • SO	Length_W0[6]	0	INT
- 🐤 SO	_Length_W0[7]	0	INT
- SO_Ma	ore_W0	0	BYTE
😑 📕 SO_Da	ta_W0		ARRAY[01179] OF BYTE
- <b>S</b>	Data W0[0]	69	BYTE
- • SO	Data W0[1]	128	BYTE
- SO	Data W0(2)	0	BYTE
- 00	Data W0(3)	0	BYTE
	Data W0(4)	0	BYTE
	Data W0151	43	BYTE
50	_Data_W0[0]	2	DVTC
- 90	_Lata_wo[6]	6	DIL
S0	_uata_wu[/]	3	BTIE
- <b>S</b> O	_Data_W0[8]	40	BYTE
- 🐤 SO	_Data_W0[9]	26	BYTE
- <b>\$</b> 50	_Data_W0[10]	71	BYTE
- <b>S</b>	_Data_W0[11]	0	BYTE
- 🔶 SO	_Data_W0[12]	9	BYTE
50	Data_W0[13]	139	BYTE
- <b>S</b>	Data W0[14]	64	BYTE
	Data W0[15]	0	BYTE
S0	Data W0[16]	0	BYTE
00	Data_W0[17]	16	DVTC
30	Data Working	10	DITE
	Data W0[10]	210	BITE
- 50	_Data_wo[19]	0	BYTE
- SO	_Data_W0[20]	0	BYTE
- 🍤 SO	_Data_W0[21]	0	BYTE
- <b>S</b>	_Data_W0[22]	0	BYTE
- 🔶 SO	_Data_W0[23]	0	BYTE
- 🔶 SO	_Data_W0[24]	0	BYTE
S0	Data W0(25)	0	BYTE
- 🔶 SO	Data W0[26]	0	BYTE
	Data W0(27)	0	BYTE
- SO	Data W0[28]	0	BYTE
	Data W0[20]	0	BYTE
	Data W0(20)	0	BITE
- 50	_Data_wu[30]	0	BTIE
- 🌖 SO	_Data_W0[31]	0	BYTE
- • SO	_Data_W0[32]	0	BYTE
- 🔷 SO	_Data_W0[33]	0	BYTE
- 🔷 50	_Data_W0[34]	0	BYTE
- 🕒 SO	Data_W0[35]	0	BYTE
- I SO	Data W0[36]	0	BYTE
	Data W0(37)	0	BYTE
- 90	Data W0(38)	0	BYTE
- 50	Data W01291	0	RYTE
50	Data W0(40)	0	DVTE
	_uata_wu[eu]	0	DITE
- 90	Data_w0[41]	0	BTIE
- <b>S</b> O	_Data_w0[42]	0	BITE
S0	_Data_W0[43]	0	BAIF
- 🔶 SO	_Data_W0[44]	0	BYTE
- 🔷 SO	_Data_W0[45]	0	BYTE
- 🗢 SO	_Data_W0[46]	0	BYTE
- I SO	_Data_W0[47]	98	BYTE
	Data W0[48]	217	BYTE
Q1	Data W0(49)	50	BYTE
	Data W01501	144	RYTE
	Data W0/E11	0	BYTE
50	_uata_wu[51]	0	DITE
- <b>S</b> O	_uata_W0[52]	0	BTIE
- • SO	_Data_W0[53]	0	BALE
- 🔶 SO	_Data_W0[54]	6	BYTE
- 🔶 SO	_Data_W0[55]	2	BYTE
- 🔶 SO	Data_W0[56]	3	BYTE
- <b>S</b>	Data W0(57)	0	BYTE
	Data W01581	0	BYTE
00	Data W0(59)	0	BYTE
- 50	Data Witten	11	BYTE
50	Data MIDIC11	262	DVTE
- 50	_Data_w0[61]	200	DITE
- SO	LARA WUBZI	233	I DI LE

PMEUCM VALVES Manual

7 System Operation 54

The above screenshot shows output data for both sockets 2 and 8. The information for socket 8 is highlighted for clarity.

Socket 2 data has a length of 49 bytes so the actual frame data is located in SO\_Data\_W0[0]...[48].

Socket 8 data has a length of 12 bytes which corresponds to SO\_Data[49]...[60].

## VALVES DFBs

Two DFBs are provided to automatically control the socket data to/from the UCM. They both use the DTM structure for the PME UCM 0302\_VALVES and the UCM1\_Data\_IN (or OUT) array and the UCM1\_Data\_IN\_Length (or \_OUT) array.

These two DFBs will automatically be installed if both of these section files are imported into a Unity Pro project:

```
c:\Niobrara\apps\PMEUCM\VALVES\VALVES_example1.xbd
```

or with these two:

```
c:\label{eq:linear} c:\label{eq:linear} c:\label{eq:linear} on the set of t
```

and

c:\Niobrara\apps\PMEUCM\VALVES\VALVES\_data\_out.xbd

These DFBs may be installed manually into a Unity Pro project by using these files:

c:\Niobrara\apps\PMEUCM\VALVES\VALVES\_inbound.xdb

and

```
c:\label{eq:loss_outbound.xdb} c:\label{eq:loss_outbound.xdb
```

## VALVES\_Inbound

NOTE: For the following discussion, it is assumed that the sockets in question are connected to remote servers (or clients). TCP and UDP operate in the same manner.



The above figure shows the overview of the process of an Ethernet frame of length X sent to the PMEUCM and the same data arriving in the UCM1\_Data\_IN array for the socket with the length X.

The VALVES UCM application works with the DTM and VALVES\_Inbound DFB to successfully transfer the full Ethernet frame data to the correct array of bytes in the M580.

The incoming Ethernet frame from the remote device may have a length of 1 to 1452 bytes. (The UCM's Window size sets this limit of 1452 bytes.)

The VALVES DTM provides an an inbound (to PLC) data window that is used to transfer the incoming socket data to the PLC. M580 DTM byte count restrictions limit this window to a maximum of 1134 bytes for the inbound DIO. The PMEUCM application and the VALVES\_Inbound DFB work together to break long messages into multiple DIO transactions and then re-assemble the original frame data for the PLC to use.

The VALVES\_Inbound DFB loads the complete frame data into the UCM1\_Data\_IN array of bytes for the particular socket and sets the UCM1\_Data\_IN\_Length value for that socket when the entire frame data is present. The PLC code simply watches for

UCM1\_Data\_IN\_Length[socket#] > 0

to know that new data has arrived and needs to be processed. When the PLC code is finished with the socket data, it sets

UCM1\_Data\_IN\_Length[socket#] := 0

to signal VALVES\_Inbound that it can post new data as it arrives.



Most of the time, incoming Ethernet frames are smaller than 1134 bytes. Additionally, data from several sockets may come in to the UCM at the same time and get queued up for the DIO transfer to the M580. The UCM and the DFB work together to pack as many frame bytes into the 1134 byte window to optimize the DIO operation.

Up to 8 'blocks' of socket data may be packed into the window. For each 'block' there is a byte to indicate the socket number and an INT to indicate the length of the block. The amount of room in the last block may not be large enough to hold the queued data for the socket so the SI\_More.[socketnumber] bit is set to tell the DFB that it needs to wait for the rest of the socket data before setting the UCM1\_Data\_IN\_Length to the proper value.

The window will be filled until one of the following occurs:

- The data window is filled (1180 bytes PLC>UCM or 1134 bytes UCM > PLC)
- All 8 blocks are used (even if the window total byte count is < window size)</li>
- Or, there is no more queued data to transfer.

For ease of example: let's assume the window is only 1000 bytes and Socket 1 needs to send 150 bytes while Socket 5 sends 500 and socket 9 sends 300. There are 50 unused bytes in the window but that is all of the data to be transferred.

Block	Socket Number	Length	Window Bytes	More Flag	Socket Bytes
0	1	150	Bytes 0149	0	0149
1	5	500	Bytes 150649	0	0499
2	9	300	Bytes 650949	0	0299
3	255	0		0	
4	255	0		0	
5	255	0		0	

6	255	0	0
7	255	0	0

Now, suppose that the above example also includes data from socket 10 of 100 bytes and socket 11 of 250 bytes that also needed to be sent at the same time. The rest of this window (until windowsize is matched) will be filled and the More flag will be set to tell the other side to buffer this data as more is coming (so wait for the next handshake exchange).

Block	Socket Number	Length	Window Bytes	More Flag	Block Bytes
0	1	150	Bytes 0149	0	0149
1	5	500	Bytes 150649	0	0499
2	9	300	Bytes 650949	0	0299
3	10	50	Bytes 950999	1	049
4	255	0		0	
5	255	0		0	
6	255	0		0	
7	255	0		0	

The next transaction would look like this:

Block	Socket Number	Length	Window Bytes	More Flag	Block Bytes
0	10	50	Bytes 049	0	5099
1	11	250	Bytes 50299	0	0249
2	255	0		0	
3	255	0		0	
4	255	0		0	
5	255	0		0	
6	255	0		0	
7	255	0		0	

When the other side sees the 'More' flag = 0, it can then send the entire 100 bytes out socket 10.

If a socket has more data bytes than the window size, it may take the entire window for this transaction and the first part of the next transaction window.

When one side (CPU or UCM) has data it needs to send to the other, that side waits for the directional handshake variables to become equal which means that the window is available. The side then claims the window, sets the block socket number(s), block socket length(s), copies the block data into the window, sets the more flag if needed, and then increments the handshake. When the other end sees the handshake become unequal, it looks at the socket number and then the code that handles that socket copies the data from the window and then echoes the handshake value to free up the window.

## **PLC State Machines**

The programming languages of the M580 do not allow for multi-threaded operation where a given operation may wait for a long period of time before receiving a response. Therefore, a state machine is a useful method of program flow control. The ST examples included with the VALVES files use some form of state machine.

The mbustcp\_server1.xst Modbus Server example uses a machine with five states.

- State 0 Forces the socket to be closed by setting the command byte to zero. The state is then advanced to 10.
- State 10 The code stays in this state until the socket status reported by the UCM also returns to zero indicating that the socket is closed. The state is then advanced to 20.
- State 20 Sets up the configuration of the sockets in use (port E1 or E2, TCP, remote TCP port number) and commands the UCM to listen on the socket. The state is then advanced to 30.
- State 30 The socket is waiting for a connection from a remote client. As soon as a connection is established, the state is advanced to 100.
- State 100 The socket is connected and waiting for a Modbus/TCP • message from the client. When the UCM1 Data IN Length[UCM socketnumber] > 0 then a new message has been received from the client. The inbound Modbus query is parsed and а reply is built and told to transmit bv setting UCM1 Data OUT Length[UCM socketnumber] to the new length.

There are other ways of moving through this state machine. If the socket is marked as not connected, the machine will revert to state 0. There are timers

running to detect if the socket is idle too long causing it to revert to state 0 if needed.

# Socket Control and Timing

It is advised that the PLC program take active control of how long a socket is allowed to attempt to connect or disconnect. Timers are needed for deciding how long to wait for a reply to a query. These timers do not need to be highly accurate or precise. Most Ethernet events can deal with several seconds of leaway.

The example ST segments included in the VALVES setup use a very simple timer techinque that is easy to implement with a minimum of code.

## SYSUPTIME

The VALVES\_Inbound DFB includes a public UDINT called SYSUPTIME as used as a free-running upcounter that increments once per second from the time the PLC starts running the program.

```
PLC_Seconds := BCD_TO_INT(%SW50/256);
if PLC_Seconds <> Last_PLC_Seconds then
    Last_PLC_Seconds := PLC_Seconds;
    SYSUPTIME := SYSUPTIME + 1;
end_if;
```

System word %SW50 provides the PLC's Real Time Clock (RTC) count of seconds in BCD with the seconds in the MSB of %SW50. The first line of code converts this value into an INT and places the seconds in the variable PLC\_Seconds. The remainder of this code increments the UDINT variable SYSUPTIME once per second.

This SYSUPTIME may then be used to set a future reference time by adding a number of seconds to this value and store the result in another UDINT variable.

For example, the following ST code would add 10 seconds to the current SYSUPTIME.

```
FutureTime := VALVES Inbound 0.SYSUPTIME + 10;
```

Later, a simple compare will determine if the 'timer' is expired:

if SYSUPTIME >= FutureTime then

(\* the timer has expired \*)

end\_if;

This type of timer is handy because it works with all sockets and the test for

PMEUCM VALVES Manual

'expired' is very simple.

PMEUCM VALVES Manual

7 System Operation 62

# 8 **QLOAD the VALVES UCM Application**

The standard PMEUCM is shipped from the factory with the VALVES application preloaded. QLOAD is used update the original version with new versions.

The QLOAD is installed by the PMEUCM\_SETUP.EXE program. The user may access this file at:

http://www.niobrara.com/programs/PMEUCM\_SETUP.EXE,

## **QLOAD** the VALVES Application

The QLOAD utility is used to load applications into the PMEUCM. Start QLOAD by Start > Programs > Niobrara > QLOAD. The first time QLOAD is started, it should look something like this:

🚯 QUCM File Downloader - 06Jun 🗕 🗆 🗙
<u>File Advanced Configure</u>
Load File Browse
Modbus Serial Modbus TCP Program Info Module Info
▼ 9600 ▼ Baud
255 Modbus Drop C 7 Bits
Application 1 C Application 2     S Bits
Even Verity ASCII
Set Defaults
Query Start Download Cancel

Click on the Browse button and select the VALVES file.

NOTE: There may be multiple versions of the VALVES file. These versions will have filenames of the form: PMEUCM0302\_VALVES\_xxxzYYzz.qcc where xxxx is the Year,

YY is the Month, and zz is the Day.

For example, the VALVES application of version 09MAY2017 would have this filename:

C:\Niobrara\apps\PMEUCM\VALVES\PMEUCM0302\_VALVES\_20170509.qcc

🍓 File to Transfer to	QUCM					×
← → • ↑ 📘	« TI10	685100A (C:) > Niobrara > apps > PME	UCM > TCPOPEN	✓ Ö Search TCF	OPEN	Q
Organize 🔻 🛛 Ne	w folder					?
ineDrive 🍊	^	Name	Date modified	Туре	Size	
💻 This PC	а.	10 PMEUCM0302_TCPOPEN_20170509.qd	c 5/9/2017 8:36 PM	Niobrara Compile	. 461 KB	
Desktop						
Documents						
Downloads	File <u>n</u> an	ne: PMEUCM0302_TCPOPEN_20170509.qcd	c		oject Cance	~

Now select the ModbusTCP tab.

🎨 QUCM File Downloader - 06Jun 🗕 🗆 🗙
<u>File Advanced Configure H</u> elp
Load File C:\Niobrara\apps\PMEUCM\TCPOPE - Browse
10.10.10.10 503 TCP Port
255 Modbus Drop
Application 1
Set Defaults
Query Start Download Cancel

Make sure that the IP Address is set to match the PMEUCM E1 port of 10.10.10.10, the TCP Port is set to 503, Modbus Drop is 255, and Application 1 radio button is set.



Connect the Ethernet port of the computer to E1 on the PMEUCM with a standard CAT5/6 cable.

Set the Ethernet port of the computer to be on the same 10.10.10.x subnet as the PMEUCM.

Press "Start Download" to begin the loading of the program into the PMEUCM.

PMEUCM0302_TCPOPEN_20170509.qcc - Downloading File
Downloading liobrara\apps\PMEUCM\TCPOPEN\PMEUCM0302_TCPOPEN_20170509
11979 words transmitted.
Cancel

When the download is finished, the program should automatically start and the screen should look something like this:

T	CPOPEN
ן קר	LC Wait Slot:5 NE UCM Ø3
B	0.0.
P	0.0
E	10.10.
1	10.10
E	10.10.
2	10.11

This screen shows that the UCM is located in Rack Slot 5 and is waiting on the M580 PLC to inform it of the Rack Name. Once the UCM has the name of its rack, it is allowed to perform DHCP to obtain the IP Address for the backplane.

PMEUCM VALVES Manual

In this screen shot, the UCM's backplane is set to an IP Address of 0.0.0.0 and is effectively disabled.

The UCM Ethernet port E1 is at the factory default IP Address of 10.10.10.10.

UCM E2 is at 10.10.10.11.

## UCM BOOT firmware too old

It is possible that the screen shows that the UCM BOOT code is not current and must be updated. Please download the latest version of the PUCM\_SETUP.EXE file from <u>http://www.niobrara.com/programs/PMEUCM\_SETUP.EXE</u> and follow instructions in Chapter 6, Loading new firmware over Ethernet.

TCPOpen 28MAR2016
BOOT Min: 25FEB2016
Actual: 23FEB2016
Must Load New BOOT
Press KEY

## UCM OS too old

It is possible that the screen shows that the UCM OS is not current and must be updated. Please download the latest version of the PUCM\_SETUP.EXE file from <u>http://www.niobrara.com/programs/PMEUCM\_SETUP.EXE</u> and follow instructions in Chapter 6, Loading new firmware over Ethernet.

TCPOpen 28MAR2016
OS Min: 27FEB2016
Actual: 23FEB2016
Must Load New OS
Press KEY

PMEUCM VALVES Manual

# 9 Unity Pro Operations

**NOTICE**: The newest version of the PTK\_DTM\_Library must be installed before attempting to use the PMEUCM. See Chapter 2.

## **New Project**

This example starts with a new project in Unity Pro XL V11.0.

- The PME UCM 0302 will be installed in the CPU rack slot 6.
- The M580 P581020 is the chosen CPU.
- Most of the IP Addresses will be left at their default settings.
  - The CPU will be at the default IP Address of 192.168.10.1
  - The PME UCM backplane will be at 192.168.10.3
  - The PME UCM E1 and E2 ports will be set to 172.16.0.10 and 172.16.0.11

The BME P58 1020 CPU is chosen, along with a BME XBP 0800 eight slot Ethernet backplane.

New Project			×
Show all versions			ОК
PLC	Min.OS Version	Description	Cancel
Modicon M340			
Modicon M580			<u>H</u> elp
BME H58 2040	02.10	CPU 580-2 ETH HSBY remote and distributed IO	
BME H58 4040	02.10	CPU 580-4 ETH HSBY remote and distributed IO	
BME H58 6040	02.10	CPU 580-6 ETH HSBY remote and distributed IO	
BME P58 1020	02.10	CPU 580-1 ETH distributed IO	
BME P58 2020	02.10	CPU 580-2 ETH distributed IO	
BME P58 2040	02.10	CPU 580-2 ETH remote and distributed IO	
BME P58 3020	02.10	CPU 580-3 ETH distributed IO	
BME P58 3040	02.10	CPU 580-3 ETH remote and distributed IO	
BME P58 4020	02.10	CPU 580-4 ETH distributed IO	
BME P58 4040	02.10	CPU 580-4 ETH remote and distributed IO	
BME P58 5040	02.10	CPU 580-5 ETH remote and distributed IO	
BME P58 6040	02.10	CPU 580-6 ETH remote and distributed IO	
Momentum Unity			
Premium			
🗄 🔤 Quantum			
Rack	Description		^
Modicon M580 local drop			-
B. Rack			_
BME XBP 0400	4 SLOTS ETHERNE	T BACKPLANE	
BME XBP 0602	6 SLOTS REDUNDA	NT ETHERNET BACKPLANE	
BME XBP 0800	8 SLOTS ETHERNE	T BACKPLANE	
BME XBP 1002	10 SLOTS REDUND	ANT ETHERNET BACKPLANE	¥
Project Settings			
Settings File:	<default settings=""></default>		

#### Double-Click on the the 'PLC Bus' to see the CPU rack view:



After selecting the "PLC Bus" in the Structural View Tree, double-click on the Ethernet ports of the CPU to open the configuration submodule.

The following services must be enabled in the DIO Master:

- FTP
- TFTP
- DHCP/BOOTP
- EIP

The simple method to enable these services is to select the 'Unlock Securty' button.

0.0 : EIO : CommHeadDIOL2											•
DIO Communicator Head											
CommHeadDIOL2	C Security	D IPConfig	1 RSTP	SNMP	NTP	Switch	0 GoS	Service	ePort	Advanced Se	ettings
	Global policy Services		Enforce Sec	urity			$\boldsymbol{<}$	Unlock Secur	ity		
	F	TP: Disa	abled	~	DH	CP / BOOTP :	Disab	led	~		
		IP: Disa	abled	~		SIMMP :	Disab	led	~		
	HI	IP : Disa	abled	~		CIP :	Disab	led	~		
	Access Contro		/								
	Subnet	IP Addre	ss	Subnet ma	isk 🗐	FTP   TFTP	HTTP	Port502	EIP SI	IMP A	
	Yes 🗸	192.168.10	0.1	255.255.0	.0						
	No 🗸										
	No 🔍										
	No 🗸										
	No 🗸										
	No									- ·	
Function:											
010											

After unlocking the security, click the check box in the tool bar to accept the change.

🚳 Unity Pro XL : <no name="">*</no>		- 🗆 X
<u>File Edit View Services Tools Build Let D</u> ebu	Window Help	
	Ø 🔲 👌 🕸 🛗 🔂 🖥 🔂 🖬 🖓 א א	## ##   🛄 🖳   🏁 🦉 🖉 🛄 🕵    🚰 🖽    🦹 🕅
Project Browser		
B Structural view	BIO DIO CommHeadRIODIOT6L2	
Project ^	RIU DIO Communicator Head	
E PLC bus	CommHeadRIODIO16L2	P 10 NTP 10 ServicePort
2: EIO Bus	Global policy	^
Derived Data Types	Enforce Sequeity	Linlock Security
Variables & FB instances		
Derived Variables	Services	DHCD / ROOTD +
U Device DDT Variables	FIF: Enabled V	
Elementary FB Instances	TFTP : Enabled V	SNMP : Enabled ~
Hardware Catalog	HTTP : Enabled ~	EIP : Enabled ~
- Modicon M580 local drop		
⊞ ··· Analog ⊞ ··· Communication	Access Control	
Counting     Discrete	Disabled V	_
B Motion		
B⊡ Hack B⊡ Supply		
Third party products     Weighing	Function:	
	K K K K K K K K K K K K K K K K K K K	> ×
	° ,	
I I I EIO Bus A RIO Bus CANopen A PLC b		
X		
Build Changes Import/export	User errors ) FDT log event ) Search/Replace /	
Ready	HMI R/W mode OFFLINE US	B:SYS NOT BUILT Q
	1 1	

Now close the submodule.

For this example, the UCM will be located in slot 6 of the local rack. After right clicking on slot 6, a select "New Device".



The PME UCM 0302 is located in the 'Third Party products' section. Select the UCM and click 'OK'.

Topological Address:		
	0.6	OK Cancel
Pat Number     Description             Modicon M580 local drop             Communication             Counting             Discrete             Motion             Third party products             PME SWT 0100             PME UCM 0302             User Programmable Module for customer Serial and Ethemet networks		Help

PMEUCM VALVES Manual

9 Unity Pro Operations 73

🖵 PLC bus						
Bus:	0	BME P58 1020	02.10	$\sim$		
	CPS 2000				×BEE	

The UCM will now appear in the rack.

The PLC rack window may now be closed.

## DTM Hardware Catalog Update

The next step is to force an update of the DTM Catalog. The DTM Catalog is accessed through Tools > Hardware Catalog.

Unity Pro XL : <no name="">*</no>											⊐ ×
File Edit View Services Tool:	s Build PLC De	ebug Window ⊢	lelp								
✓ Project Browser Alt+1	o or 🗹 🦊	) Ci, jo   🖩   🎍	5 🕸 🖱 🖽	👼 🖥 🗖 🖶	PH PH #4		✓ 28 26	RUN 🖳 🔛	er 🗇 📶 🖣	6880	8 N8
Hardware Catalog Alt+2											
In Types Library Browser Alt+3											
Operator Screen Library Alt+4	_										
Search / Replace Alt+5											
Diagnostic Viewer Alt+6											
PLC Screen Alt+7											
Variable Window Alt+8	ř										
Data Editor Alt+9											
DTM Browser Alt+Shift+1	105										
Bookmarks Alt+Shift+2											
Trending tool											
Convert Partially											
Network Inspector	ices										
Ethernet Network Manager	1										
E Types Library Manager	-										
<u>C</u> ustomize											
Options											
Project Settings											
Animation Tables 0 Operator Screens B ■ Documentation											
<u></u>											
×											
Build Change	s / Import/export	A User errors		A Search/Rep	lace /						
Opens the Hardware Catalog editor			н	MI R/W mode OFF	LINE	USB:S	ivs		N	OT BUILT	Q

The Hardware Catalog Window should appear and look something like this:

Hardware Catalog					x
All devices		Device	Туре		^
Device types		140NOC77100 (from	Device		
i vendors		140NOC77101 (from	Device		
🕂 Groups		140NOC78000 (from	Device		
i Protocols	١ <u>ق</u>	140NOC78100 (from	Device		
	١ <u>ق</u>	Advanced Generic E	Device		
	١ <u>ق</u>	ALTIVAR61 Revision	Device		
		ALTIVAR61 Revision	Device		
		ALTIVAR71 Revision	Device		
		ALTIVAR71 Revision	Device		
		ALTIVAR71 Revision	Device		
	1	BME AHI 0812	Gateway		
	19	DME ALIO 0412	C-+		× .
	<u>  &lt;</u>			2	
Update					
EIO Bus A RIO Bus A C	ANo	pen 👌 PLC bus 👌 DTM a	atalog		

Click on the "DTM catalog" tab at the bottom.

Then Click on the "Update" button.

A message box should pop up asking if it is ok to update the catalog. Select "Yes".

Note: This box opens every time the Update button is clicked.



#### A progress window pops open.

Updating Dtm catalog	
25%	
GenericDDXMLDTM.DTMCc	ire.1
Cancel	

After the catalog update is complete, the new Niobrara DTM device should be listed in the hardware catalog. Also, the "User Errors" display should show "Information: The update of the Dtm catalog is finished"

Hardware Catalog							×
Vendors		Device	Туре	Vendor	Version	Poiss	
Advanced Micro Cont as	3	PME UCM 0302_TCPOPEN	Device	Niobrara	01.14	2017-05-09	
AGM Electronics, Inc.							
HMS Industrial Showrks							
Niobrara							
Bockwell Automation							
Rockwell Automation - R							
Rockwell Automation/En							
< >>							_
Update							
EIO Bus \ RIO Bus \ CA	Nop	en $\lambda$ PLC bus $\lambda$ DTM catalog	1				

Now, Open the DTM browser by selecting Tools > DTM Browser.

#### PMEUCM VALVES Manual
🔯 Unity Pro XL : <no name="">*</no>								
File Edit View Services Tool	s Build PLC De	bug Window Help						
✓ Project Browser Alt+1	o or 🖂 🏓	९ 🖉 🔲 🚠 🚳 🗄	i 🔠 🔚 🔚 🖬 🐻 🕯	1 24 #4	- 126 26 🔳	🛚 🖳   RUM STOP 🗇	11 🛼 🧏 🗆 🗆	8 N8
Hardware Catalog Alt+2	-							
Jo Types Library Browser Alt+3								
Operator Screen Library Alt+4								
Search / Replace Alt+5								
Diagnostic Viewer Alt+6								
PLC Screen Alt+7								
Variable Window Alt+8	ř							
Data Editor Alt+9	_							
DTM Browser Alt+Shift+1	es							
Bookmarks Alt+Shift+2								
Trending tool								
Convert Partially								
Network Inspector	nces							
Ethernet Network Manager	1							
🖶 Types Library Manager								
<u>C</u> ustomize								
Options								
Project Settings								
, Animation Tables								
Information: The Update of Information: The Upda	the Dtm catalog the Dtm catalog	is finished is finished <b>\User errors</b> \FDT log	event X Search/Repla	ice /				
Opens DTM Browser			HMI R/W mode OFFL	INE US	B:SYS		NOT BUILT	Q

The DTM Browser will open and show a tree with the CPU at 192.168.10.1.

Right click on the CPU and select "Add".



A window will pop up showing all of the installed DTMs. Scroll down until you reach the PME UCM 0302 VALVES device by Niobrara.

Add						×
	Device	Туре	Vendor	Version	Date	^
	Anybus Communicator - Sl	Device	HMS Industrial N	1.60		
	Anybus Communicator - Sl	Device	HMS Industrial N	2.3		
	Anybus Communicator CA	Device	HMS Industrial N	1.3		
	Anybus-C EtherNet/IP (fro	Device	HMS Industrial N	1.27		
	Anybus-S EtherNet/IP Re	Device	HMS Industrial N	1.26		
	Anybus-S EtherNet/IP Re	Device	HMS Industrial N	2.1		
	PME UCM 0302_TCPOPEN	Device	Niobrara	01.14	2017-05-09	
	1305 AC Drive Revision 6	Device	Rockwell Autom	6.1		_
	1305 AC Drive Revision 7	Device	Rockwell Autom	7.1		
	1336 IMPACT Drive Revis	Device	Rockwell Autom	1.1		_
	1336 IMPACT Drive Revis	Device	Rockwell Autom	2.1		
	1336 IMPACT Drive Revis	Device	Rockwell Autom	3.1		_
	1336 IMPACT Drive Revis	Device	Rockwell Autom	4.1		
	1336 PLUS Drive Revisio	Device	Rockwell Autom	1.1		
	1336 PLUS Drive Revisio	Device	Rockwell Autom	2.1		
	1336 PLUS Drive Revisio	Device	Rockwell Autom	3.1		
	1336 PLUS Drive Revisio	Device	Rockwell Autom	4.1		_
	1336 PLUS Drive Revisio	Device	Rockwell Autom	5.1		
	1336 PLUS II Drive Revisi	Device	Rockwell Autom	10.1		¥
	Add DTM				Close	

Notice that it has the version 01.14 which matches the SW version in the txt file.

Press Enter or "Add DTM" to load the DTM for the PMEUCM. A window will pop up with information about the DTM.

At this point the "Alias name" may be modified. The example ST and DFB code used later requires this "Alias name" to be set to 'UCM1'.



So change the name to 'UCM1'.

Properties	of device	e				×
General	Device in	nformation	DTM information	Protocol in	formation	
- DTM r	name man	agement —				
Name	e:	UCM1				
			OK		Cancel	Help

Pressing "OK" will add the DTM device to the DTM Browser.

Add new device.
25% - UCM1

The PMEUCM is now added to the tree below the CPU.



# Link the DTM to the PMEUCM Hardware

It is time to actually associate the DTM instance with the actual PMEUCM device. This is done inside the DTM Browser window.

Right click on the CPU and select Open.



NOTE: The "Source IP Address" is a pull-down listing of all of the IP Addresses of the Unity Pro PC. Make sure to select an address that is on the same subnet as the M580 PLC. In this case the IP Address of 192.168.10.200 is selected since the PLC is at 192.168.10.1.

NOTE: This list only shows the active IP Addresses for the Unity PC. The user must have an active Ethernet connection to proceed.

BMEP58_ECPU_EXT - fdtConfiguration			
Communication BME P58 1020		Scl	Electric
<ul> <li>Channel Properties TCP/IP</li> <li>Services</li> <li>Address Server</li> <li>EtherNet/IP Local Slaves</li> <li>Local Slave 1</li> <li>Items</li> <li>Local Slave 2</li> <li>Items</li> <li>Local Slave 3</li> <li>Items</li> <li>Device List</li> <li>[] [S14] UCM1 <eip: 192.168.10.3=""></eip:></li> <li>[] Primary Read/Write Data Connection</li> <li>Items</li> <li>Logging</li> </ul>	Source Address: Source IP Address: Sub-Network Mask: EtherNet/IP Network Detection: Begin detection range address: End detection range address: Modbus Network Detection: Begin detection range address: End detection range address:	206.223.51.238 10.4.19.200 10.0.10.200 10.99.9.238 10.99.738 40.22.64.131 90.0.0.2 127.0.0.1 172.16.0.200 192.168.1.200 192.168.1.200 192.168.13.200 192.168.13.200 192.168.13.200 192.168.13.200 192.168.13.200 192.168.99.200 206.223.51.238 206.223.51.254	
< >> Help IDE Disconnected ① Data set		OK Cancel	Apply

Now click on the PME\_UCM\_0302\_... entry in the list on the left.

Select the "Address Setting" Tab.

The "Identifier" must be modified to define the exact Rack and Slot occupied by the PMEUCM.

In this example, the PMEUCM is located in the CPU rack, Slot 6. Therefore, the Identifier must be set for "Mx80\_06\_PMEUCM03".

NOTE: If the PMEUCM is located in a remote rack, the YYY value is the thumbwheel (rotary switches) setting of the eCRA, not necessarily the logical rack number.

So, if the PMEUCM is in remote rack 1, slot 6, the Identifier would normally be C001\_06\_PMEUCM03.

After setting the Identifier, click "Apply" to accept the settings and close the window.

Communication BME P58 1020		
Channel Properties  TCP/IP  Services  Address Server  Local Slave 1  Items Local Slave 2  Items	Propert & Address Setting IP Configuration IP Address: Subnet Mask: Gateway:	192 . 168 . 10 . 3         255 . 255 . 0 . 0         192 . 168 . 10 . 1
Local Slave 3     Leave     Device List     [514] UCM1 <eip: 192.168.10.3="">     Primary Read/Write Data Conce     Items     Logging</eip:>	ddress Server IDHCP for this device: Identified by: Identifier:	The format MUST be :- Mx80_XX_PMEUCM03 or MS80_XX_PMEUCM03 or MS80_XX_PMEUCM03 or CYYY_XX_PMEUCM03 or Enabled Where, XX = Slot number(range 01-11) YYY = CRA rack no.(range 000-159) Mx80_06_PMEUCM03
< Help	>	OK Cancel Apply

# **DTM Configuration**

The PMEUCM must be configured through fields in the DTM screen.

Right click on the PMEUCM entry in the DTM Browser Tree and select "Open"



The DTM screen for the PMEUCM will open. Select the "Application Configuration" tab.

PMEUCM0302						
Module						
01.14						
tity Ethernet Configuration Application Config	guration Process Data					
0					C	5
Parameter Name	Current Value	Default Value	Unit	Minimum Value	Maximum Value	^
Ethernet Ports						
E1 Ethernet Port						
IP Address	10.10.10	10.10.10.10				
Subnet Mask	255.0.0.0	255.0.0.0				
Default Gateway	0.0.0.0	0.0.0.0				
E2 Ethernet Port						
IP Address	10.10.11	10.10.10.11				
	255.0.0.0	255.0.0.0				
Default Gateway	0.0.0.0	0.0.0.0				_
- The Serial Ports						_
- The S1 Serial Port						_
- Port 1 Mode	UCM OS RTU Slave	UCM OS RTU Slave				_
— Port 1 Baud Rate	9600	9600	BPS			<b>v</b>

This tab shows the configuration settings for E1 and E2 ports, global OS settings, and S1 and S2 serial ports.

### E1 and E2 Ethernet Port

Edit the strings for IP Address, Subnet Mask and Default Gateway for both E1 and E2. For this example, E1 will be set to 172.16.0.10 with a subnet mask of 255.255.255.0 while E2 is set to 172.16.0.11 and also with a subnet mask of 255.255.255.0.

	PMEUCM0302					
	Module					
1000	01.14					
	01.14					
dentity	Ethernet Configuration	Application Configu	ration	Process Data		
		0				
Para	meter Name			Current Value	Default Value	Unit
무 문	B Ethernet Ports					
E	🗄 🚪 E1 Ethernet Port					
			1	172.16.0.10	10.10.10.10	
			1	255.255.255.0	255.0.0.0	
	Default Gatew	ay		0.0.0.0	0.0.0.0	
E	E2 Ethernet Port					
	IP Address		1	172.16.0.11	10.10.10.11	
			1	255.255.255.0	255.0.0.0	
	Default Gatew			0.0.0.0	0.0.0.0	
<b>p</b> -2	B Serial Ports					
E	🚦 🖁 S1 Serial Port					
	- Port 1 Mode			UCM OS RTU Slave	UCM OS RTU Slave	
						-

### **UCM OS Settings**

- OS Modbus/TCP Server Port The PMEUCM Operating System has its own Modbus/TCP server used (used by QLOAD). Normally this is set to use TCP Port 503 to avoid situations where the M580 is serving on the standard Modbus/TCP port 502. Valid settings are:
  - Disabled
  - · 502
  - 503 (Default)
- Debug Web Server Port The VALVES application has a built-in web server to assist in debugging an M580 application. Normally this value is set to use TCP port 81 to avoid situations where the M580 is serving on the standard port 80. Valid settings are:
  - Disabled

- · 80
- 81 (default)

• Debug TELNET Server Port - The VALVES application has a built-in TELNET server to assist in debugging an M580 application. Normally this value is set to use TCP port 24 to avoid situations where the M580 is serving on the standard port 23. Valid settings are:

- Disabled
- · 23
- 24 (default)

OS Max TCP Segment Size – The UCM OS can have its maximum segment size adjusted for use in VPN applications. Valid settings are:

- 1452 (default)
- 750 (VPN)
- OS TCP Keep Alive Time The number of seconds an idle socket waits before sending a Keep Alive to the remote end of the connection. Normally this value is set to 10 seconds to ensure unused sockets are closed quickly. Valid settings are:
  - 10 (default)
  - ° 30

٠

•

- · 60
- Location Name This is a 10 character (max) text string to name the UCM. This location name is shown on the front screen and web page.

#### S1 and S2 Serial Port

**NOTE**: The current release of VALVES does not provide support for serial port operation. These configuration values are in the DTM for future use and are ignored by the module.

Parameter Name	Current Value	Default Value	Unit
- 😓 S1 Serial Port			
Port 1 Mode	UCM OS RTU Slave	UCM OS RTU Slave	
Port 1 Baud Rate	9600	9600	BPS
Port 1 Parity	EVEN	EVEN	
Port 1 Data Bits	8	8	
Port 1 Stop Bits	1	1	
	x0a	x0a	
Terminator based on time between	Disabled	Disabled	mS
🖻 😓 S2 Serial Port			
Port 2 Mode	UCM OS RTU Slave	UCM OS RTU Slave	
Port 2 Baud Rate	9600	9600	Baud
Port 2 Parity	EVEN	EVEN	
<			

- Mode The serial ports may be set to several different operating modes.
  - UCM OS RTU Slave (default) This mode allows the operating system to control the serial port and respond to Modbus RTU messages.
  - Modbus RTU Framing This mode configures the port to be either a Modbus RTU Master or Slave. The port is treated like VALVES mode but the message data is prepended with a word of length of the RTU frame. The serial port automatically calculates and adds the CRC16 checksum on transmitted messages. It also automatically calculates and verifies and removes the CRC16 checksum on received messages. The port also automatically terminates received messages based on 3.5 character times of intercharacter timeout.
  - VALVES This mode allows the serial port to be treated like a UDP socket. The terminating characters and Termination based on intercharacter timeout are used to determine the end of a frame.
- Baud Rate bit rate for the serial port
  - · 1200
  - 2400
  - · 4800
  - 9600 (default)
  - · 19200
  - · 38400
- Parity
  - NONE
  - ODD

- EVEN (default)
- Data Bits
  - 7
  - 8 (default)
- Stop Bits
  - 1 (default)
  - 2
- Terminating Characters This is a list of hexadecimal values preceded by a lower case 'x' that is used by the port to determine the end of a packet when the port is set to VALVES mode. Multiple values may be added by placing a comma between fields. Some Examples:
  - x0a Line Feed (default)
  - $\circ$  x0d Carriage Return
  - $\circ x03 ETX$
  - empty termination by specific characters is disabled
- Terminator based on time between characters This value determines how long the UCM will wait on characters (without a termination character) to be received before sending the message to the M580. Allowed values are:
  - Disabled (default)
  - 10 mS
  - 50 mS
  - 100 mS
  - 200 mS

### Applying and Installing Changes to the DTM

Any time one of these settings is adjusted, the following procedure must be followed:

- 1. After finishing the adjustments to the settings, click "OK" or "Apply" and then "Cancel" to close the DTM window.
- 2. Do a "Build", "Build Changes" or "Rebuild all Project".



3. Transfer this new prm file to the FDR server. Right click on the PME\_UCM entry in the DTM tree and select "Device Menu" > "Additional Functions" > "Transfer to FDR Server". This action causes the PME Generic DTM dll to build a new prm file and send it to the FDR server.



- 4. The easiest thing to do now is to reboot the PMEUCM card to get it to read the PRM file from the FDR server.
  - 1. From the UCM front panel screen, select:
    - 1. 'Menu'
    - 2. 'System'

- 3. 'Reboot' and the module should reboot.
- 2. Or, cycle power on the rack containing the PMEUCM.

After the module boots and establishes a connection with the M580 CPU across the backplane, the new settings will be applied to the application.

If the IP Address for either E1 or E2 has been changed or if any of the OS IP parameters (OS port, TCP keep alive, etc.) have changed, the PME will make these changes, save them to EEPROM, and reboot once.

### Import FBD Code

The VALVES application requires two DFBs to be installed in the Unity Project. The easiest method is to import a pair of Function Block Diagram sections into the project. The two sections are included in the PMEUCM VALVES Setup.

Open the Project Browser and right click on "Sections" and select "Import..."



Browse to the "C:\Niobara\apps\PMEUCM\VALVES\" folder. There should be a list of .xst files.

Look <u>i</u> r	n: 🔄 TCPOPEN		✓	
<u>_</u>	Name	^	Date modified	Туре
	📄 mbust	cp_client1.xst	5/9/2017 11:11 AM	XST File
Quick access	🗋 mbust	cp_client2.xst	5/9/2017 9:24 PM	XST File
	📄 mbust	cp_server1.xst	5/9/2017 11:11 AM	XST File
	🗹 🗋 tcpope	en_data_in.xbd	5/9/2017 11:10 AM	XBD File
Desktop	📄 tcpope	en_data_in2.xbd	5/9/2017 9:24 PM	XBD File
-	tcpope	en_data_out.xbd	5/9/2017 11:11 AM	XBD File
	📄 tcpope	en_data_out2.xbd	5/9/2017 9:24 PM	XBD File
Libraries				
This PC				
Network	<			
NELWOIK	File <u>n</u> ame:	tcpopen_data_in.xbd	~	Import
	Files of type:	All Languages (*.XLD;*.XBD;*	XIL;*XST;*XSF;*X5 ~	Cancel
- <i></i>				
Uptions				
With wizer	d			

Select 'VALVES\_data\_in.xbd' and press 'Import'.

Unity Pro should prompt with a message box:

Туре	Name	New Name	Кеер	Replace	Rename
Duplicate DDT	T_UCM1	T_UCM1_0	×		
Duplicate DDT	T_UCM1_IN	T_UCM1_IN_0	×		
Duplicate DDT	T_UCM1_OUT	T_UCM1_OUT_0	X		
The variable exists	UCM1	UCM1_0	X		

Select 'Keep All' and then 'Ok'.

A new FBD section will be added:



Once again, right click on 'Sections' and select 'Import'.

riotion Communication Ethernet Network		
Sectors	New Section	
Events	Import	
Timer Events	Create builded Activation Conditions Table CTRL+T	
Animation Tables	Add User Directory Add Hyperlink	
_	Zoom in Zoom out	
	Expand all	
	Collapse all	
¥[{MAST <task>} :</task>	Object created: section TCPOPEN_Data_IN	

This time, select 'VALVES\_data\_out.xbd'.

Look in	TCPOPEN	~	🎯 🌶 📂 🛄 🗸	
-1	Name	^	Date modified	Туре
	mbust	p client1.xst	5/9/2017 11:11 AM	XST File
Quick access	mbust	cp client2.xst	5/9/2017 9:24 PM	XST File
	mbust	cp_server1.xst	5/9/2017 11:11 AM	XST File
	tcpope	n_data_in.xbd	5/9/2017 11:10 AM	XBD File
Desktop	tcpope	n_data_in2.xbd	5/9/2017 9:24 PM	XBD File
-	🗹 📄 tcpope	n_data_out.xbd	5/9/2017 11:11 AM	XBD File
<b>•</b>	📄 tcpope	n_data_out2.xbd	5/9/2017 9:24 PM	XBD File
Libraries				
Lange Contract This PC				
Network	<			
Network	File <u>n</u> ame:	tcpopen_data_out xbd	~	Import
			VCT + VCE + VC	Cancel
	Files of type:			

Again, select 'Keep All' followed by 'Ok'.

nport Trouble Report	:				>
Туре	Name	New Name	Кеер	Replace	Rename
Duplicate DDT	UCM_Full_Frame	UCM_Full_Frame_0	×		
Duplicate DDT	T_PME_UCM_03	T_PME_UCM_03	×		
Duplicate DDT	T_PME_UCM_03	T_PME_UCM_03	X		
Duplicate DDT	T_PME_UCM_03	T_PME_UCM_03	X		
Duplicate DDT	T_UCM1	T_UCM1_0	X		
Duplicate DDT	T_UCM1_IN	T_UCM1_IN_0	×		
Duplicate DDT	T_UCM1_OUT	T_UCM1_OUT_0	X	•	
The variable exists	UCM1	UCM1_0	×	•	•
	<u>0</u> k	. <u>C</u> ancel	Kee	p All 🛛 F	Rep! ce All

Now there should be two FBD sections in the project.



PMEUCM VALVES Manual

### Import Modbus/TCP server ST section

Now, import some code that will actually do something with sockets! Right click on the 'Sections' and select 'Import'.

Ethernet Network		
Sector:	New Section	
	Import	
Events	Create builded Activation Conditions Table	CTRL+T
I/O Events	Add User Directory	
Animation Tables	Add Hyperlink	
Documentation	Zoom in	
	Zoom out	
End IMPOBT source ( 0, er	Expand all	=
	Collapse all	

Now select 'mbustcp\_server1.xst' and click 'Import'.

🎯 Import				×
Look in:	TCPOPEN	~	G 🤌 📂 🛄 -	
Quick access Desktop	Name mbustcp mbustcp mbustcp tcpopen tcpopen tcpopen tcpopen	^ _client1.xst _client2.xst _server1.xst _data_in2.xbd _data_in2.xbd _data_out.xbd _data_out2.xbd	Date modified 5/9/2017 11:11 AM 5/9/2017 9:24 PM 5/9/2017 11:11 AM 5/9/2017 11:10 AM 5/9/2017 9:24 PM 5/9/2017 11:11 AM 5/9/2017 9:24 PM	Type XST File XST File XST File XBD File XBD File XBD File XBD File
Libraries	<			>
Network Options	File <u>n</u> ame: Files of <u>type</u> :	mbustcp_server1.xst All Languages (* XLD;* XBD;* XIL;*.	✓ XST;*XSF;*X! ✓	Import Cancel
With wizard.				

Again, select 'Keep All' and then 'Ok'.

Туре	Name	New Name	Кеер	Replace	Rename
Duplicate DDT	UCM_Full_Frame	UCM_Full_Frame_0	×		
Duplicate DDT	T_PME_UCM_03	T_PME_UCM_03	×		
Duplicate DDT	T_PME_UCM_03	T_PME_UCM_03	X		
Duplicate DDT	T_PME_UCM_03	T_PME_UCM_03	×		
Duplicate DDT	T_UCM1	T_UCM1_0	×		
Duplicate DDT	T_UCM1_IN	T_UCM1_IN_0	×	-	
Duplicate DDT	T_UCM1_OUT	T_UCM1_OUT_0	X	-	
Duplicate DFB	TCPOPEN_Inbou	TCPOPEN_Inbou	×		
The variable exists	UCM1_Data_IN	UCM1_Data_IN_0	×		
The variable exists	UCM1_Data_OUT	UCM1_Data_OU	×		
The variable exists	UCM1_Data_OU	UCM1_Data_OU	×		
The FB instance al	TCPOPEN_Inbou	TCPOPEN_Inbou	×		
The variable exists	UCM1_Data_IN	UCM1_Data_IN	×		
The variable exists	UCM1	UCM1_0	×		•

There should now be a new section called 'MbusTCP\_Server1' installed.



Notice that the MbusTCP\_Server1 section was added to the bottom of the list. Most importantly, it is below the VALVES\_Data\_OUT section. The program will function but it will always take one extra PLC scan to actually transmit the Modbus/TCP replies. Performance will be improved if the MbusTCP\_Server1 section is moved up the list to be between the VALVES\_Data\_IN and VALVES\_Data\_OUT.

Simply left-click the MbusTCP\_Server1 item and drag it up between the VALVES sections.



## **Build All or Build Changes**

After importing the wanted segments, it is time to do a 'Build All' or 'Build

Changes' for the Project.

Select "Build > Rebuild All Project" or "Build" > "Build Changes"



# Transfer Project to PLC

After a successful Build, it is time to transfer the project to the M580. This may be done through USB or over Ethernet. Since an Ethernet port is required to transfer the DTM PRM file to the FDR server, the Ethernet connection will be shown.

Connect the Ethernet port of the PC to the Service Port of the M580.

### **PLC Set Address**

Select PLC > Set Address and choose TCPIP for the Media and set the Address of the M580 (192.168.10.1).

	Set Address	? ×
<ul> <li>✓ PLC</li> <li>Address</li> <li>192.168.10.1</li> <li>✓ &lt;a href="mailto:ma&lt;/td&gt;<td>Simulator Address 127.0.0.1</td><td>Bandwidth</td></li></ul>	Simulator Address 127.0.0.1	Bandwidth
Media TCPIP V Communication Parameters	Media TCPIP V Communication Parameters	OK Cancel
Speed rate auto-adaptation at the end of do	wnload	<u>H</u> elp

It is usually a good idea to try the "Test Connection" button to make sure that the PC can connect with the M580.

UnityXL	×
1	Successfully connected to the currently selected target.
	ОК

If successful, press the OK button to close the 'Set Address' window.

### **PLC Connect**

Now select PLC > Connect to open a connection to the M580 CPU.



The bottom display will change from 'OFFLINE' to 'ONLINE'. It should also show 'DIFFERENT' to indicate that the PLC is not the same as Unity Pro.

rent 入 Search/Replace	1
HMI R/W mode OFFLINE	TCPIP:192.168.10.1
HMI R/W mode DIFFERENT	RUN UPLOAD INFO OK TCPIP:192.168.10.1

### **Transfer Project to PLC**

After connecting, transfer the project to the PLC.

The Transfer Project to PLC window should look something like this:

Unity Pro XL : <no na<="" p=""></no>	me>*		- 0
<u>File Eart View Servic</u> sconn <u>e</u> ct t <u>A</u> ddress	Ctrl+K	peoug window Help	₩[*] \$2 \$2 \$2 \$2 \$2 \$2 \$2 \$2 \$2 \$2 \$2 \$2 \$2
andard Mode nulation Mode			
mpare	CHU		
insfer Project to PLC insfer Project from PLC insfer Project from Primary t	Ctrl+Shift+L o StandBy PLC		
ve Data from PLC to Eile store <u>D</u> ata from File to PLC			
fety/Maintenance op it	Ctrl+Shift+M Ctrl+R		
odate Upload Information			
odate Init <u>V</u> alues with Current odate Local Init Values with Pl	Values. LC Init Values.		
roject <u>B</u> ackup	•		
emory Consumption ate Ram Vie <u>w</u> er			
Variables 6     Element     Derive     Derive     Element     Derive     Derive     Element     Derive     Element     Communic     Element     Element     Element	& FB instances tary Variables DDT Variables DDT Variables ved Variables tary FB Instances I FB Instances ation letwork		
× Linking Process succeede	d :0 Error(s) , 0 Wa	ning(s)	
Build	Changes / Import/e	ont À Useremons À FDTlogevent À Search/Replace	7
Transfers the current PC p	roject into PLC	HMI R/W mode DIFFERENT	RUN UPLOAD INFO OK TCPIP:192.168.10.1 MEM BUILT

Transfer Projec	t to PLC			×
PC Project		Overwritten F	PLC Project	
Name: Version: Last Build:	Project 0.0.1 1/19/2016 2:12:38 AM	Name: Version: Last Build:	Project 0.0.17 1/17/2016 11:48:56 PM	
PLC Rur	n after Transfer		Cancel	
	nansiei		Cancer	

It is usually convenient to check the PLC Run after Transfer box.

Transfer Proje	ct to PLC		×
PC Project		Overwritten PLC Project	
Name: Version: Last Build:	Project 0.0.1 5/12/2017 11:35:14 AM	Name:ProjectVersion:0.0.1Last Build:5/11/2017 10:21:55 PM	
PLC Ru	n after Transfer Transfer	Cancel	

If the PLC is in RUN, you will be prompted to Stop the M580.

WARINING: Stopping a running PLC may result in injury or death. Make sure that you understand the consequences of halting a running program.



The transfer should look like this:

Project transfer	
	71% - Packet 86/120 is successfully <mark>sent.</mark>
	Cancel

The Run confirmation screen will be shown if the "Run after Transfer" was selected.



Selecting "OK" will start the PLC.

The UCM should make some beeping sounds as the PLC transitions from STOP > RUN and RUN>STOP. It also beeps when booted, and when the setup is complete and the application is properly communicating with the M580 DIO.

PMEUCM VALVES Manual

9 Unity Pro Operations 103

#### PMEUCM VALVES Manual

# **10 Front Panel Operation**

# LED Panel

Most of the LED indicators on the top panel are controlled by the user application with a few LEDs controlled by the UCM operating system.



### **Top Panel Lights**

The meaning of these lights is described in the following table.

Label	Color	Description
RUN	Green	ON – The VALVES application is running. NOTE: This is NOT an indication of the run/halt state of the PLC.
ERR	Red	ON – There is a configuration issue. The front panel LCD will contain more information about the error condition.
PWR	Green	ON - The PMEUCM has proper 24Vdc power from the Ethernet backplane.
User 3	Green	The DTM configuration from the Primary CPU is correct.
User 4	Amber	The DTM configuration is not verified.
User 5	Green	The PRM file in the HSBY Secondary FDR server is being checked.
User 6	Amber	The PRM file in the HSBY Secondary FDR server is not verified.
BP Active	Green	DIO Operation with the Primary CPU is active.

PLC Run	Green	M580 CPU is in RUN and DIO is active.
E1 Link	Green	Ethernet port E1 has active LINK.
E2 Link	Green	Ethernet port E2 has active LINK.

Additionally, there are two RED lights behind the LCD. One red light will only be on when the backlight is OFF to help visually indicate a problem.

The second red light is controlled by UCM\_COMMAND.3 from the PLC code.

# LCD and Joystick Operation

The front panel LCD provides status information about the PMEUCM and user interaction with the setup and operation of the card/application.

The information displayed on the "splash" screen varies depending on the



configuration and state of the module.

### **Fault Indication**

If the application is in a fault condition, the screen will show the fault in an inverted text box. The PTK IP Address is shown as BP (backplane). The E1 and E2 IP Addresses are also shown.

Many fault display screens change between multiple views showing the found condition and the required condition.

Fault	Condition	Solution
Duplicate IP Address	E1 or E2 in Conflict with another device	Change IP Address
PTK FW Too Old	PTK FW needs updating	Update PTK with Unity Loader
Waiting on PTK slot unknown BP = 0.0.0.0	Unit just booted	Wait for PTK to establish comms with M580
Waiting on PTK slot unknown BP = 10.10.x.y	Unit up for a while, BP still at factory IP Address	Check rack addressing in PLC
Waiting on PTK slot known BP = 192.168.0.x	PTK unable to load prm file	Wait, Check "Identifier" DTM setting, or "Transfer to FDR server"
Wrong DTM File	DTM File name must be 'PME UCM 0302_VALVES'	Install correct DTM in DTM browser
Wrong DTM Version	DTM version must match application requirement	Update installed DTM with DTM Utility
Wrong DTM Byte Count	DTM configuration is wrong	Contact Niobrara Tech Support
Wrong Util Version	Nrdptkddxmlutil.exe too old	Install new version and update DTM
Wrong DLL Version	Generic PME DTM too old	Install new Generic PME DTM to Unity

### **Normal Operation**

The screen shows an overview of the 16 possible socket connections.



### Backlight

The backlight time is controlled by user code. In this case there is a timer that keeps the backlight on for 180000 mS (3 minutes) when there is no activity of the joystick. At the end of this timer, the UCM code changes the screen back to the splash screen.

#### Menus

Moving the joystick will cause the application to show various menus to access status or setup screens. Move the highlighted cursor around with the joystick. Typically a right press will act as "Enter" while a left press will act as "Escape". Sometimes a push in "Enter" is needed (Factory Default for example).

### Menu Screen



Moving the joystick to the right while on the splash screen shows the main menu screen. Move the joystick up and down to highlight the items and to the right or push in for Enter to select. Move left to exit.

#### **Sockets Screens**



These data screens allow quick viewing of the online/offline status of the 16 possible sockets. The top part of the screen shows four of the sockets with the left column the entry number (0-15) and the right column the status of the connection.

Possible states are:

- "Conted" = Connected
- "Listng" = Server Listening
- "Trying" = Client trying to connect
- "Idle" = Idle, not active

The bottom portion of the screen shows:

- Target UCM Ethernet port E1 or E2
- TCP or UDP
- local TCP port number for servers or the remote port for clients
- IP Address of the remote device
- State condition of the PLC program

### **Config Menu**

The Config menu shows the current IP Address of the PTK card and allows the modification of the UCM's E1 and E2 IP Address.

NOTE: The UCM's E1 and E2 may only be changed if the backplane interface is inactive.

### Stats Menu

Statistical counters are provided for a variety of fields.



### System Menu

The system items allow the user to reset the application to factory default settings or exit to the UCM operating system.



Press Enter means to push in on the joystick.

# 11 Debug Web Server

A simple web server is included in the VALVES application. The TCP port number that the web server listens on is controlled by the setting in the DTM editor. Possible values are:

- 0 = Disabled
- 80 = Standard TCP port for web servers
- 81 = Default (in case the VALVES PLC code is running a web server)

	PMEUCM0202 Module 01.09					
dentity Ethe	ernet Configuration	Application Configura	ation	Process Data		
		Q				
Paramete	r Name			Current Value	Default Value	
	Default Gateway			0.0.0.0	0.0.0.0	
Б-За E	P R E2 Ethernet Port					
- IP Address			192.168.30.12	10.10.10.11		
	Subnet Mask			255.255.255.0	255.0.0.0	
Default Gateway				192.168.30.1 0.0.0.0		
中品 U	ICM OS Settings					
OS Modbus/TCP Server Port (QLOAD)				503	503	
				81	81	
	Debug TELNET Server Port			24	24	
OS Max TCP Segment Size				1452	1452	
	OS TCP Keep Alive	Time		10	10	
	Location Name (10	characters Max)		Test Lab		

Note: Remember to do a 'Build Changes' and then 'Transfer to FDR server' after

PMEUCM VALVES Manual

modifications to the DTM. The UCM will need to be rebooted after the update.

# Home Page

· → C' fi	10.10.10.1	2:81						1	🔂 🐵 📭	
lome				TCP	OpenTe	st Lab	•			
onriguration					PLC in RUN					
tatistics					Sockets					
lelp	Number	Type	TCP/UDP	UCM Port	Local IP	Local Port	Remote IP	Remote Port	Status	
	0	Server	TCP	E1	10.10.10.12	502	10.10.10.10	2333	Connected	
	1	Server	TCP	E1	10.10.10.12	502	0.0.0.0	0	<u>Listening</u>	
	2	Server	TCP	E1	10.10.10.12	502	0.0.0.0	0	<u>Listening</u>	
	3	Server	TCP	E1	10.10.10.12	502	0.0.0.0	0	<u>Listening</u>	
	4	Server	TCP	E2	192.168.30.12	502	192.168.30.10	1169	Connected	
	5	Server	TCP	E2	192.168.30.12	502	0.0.0.0	0	<u>Listening</u>	
	6	Server	TCP	E2	192.168.30.12	502	0.0.0.0	0	<u>Listening</u>	
	7	Server	TCP	E2	192.168.30.12	502	0.0.0.0	0	<u>Listening</u>	
	8	Client	UDP	E2	192.168.30.12	0	192.168.30.10	502	Connected	
	9	Client	TCP	E1	10.10.10.12	0	10.10.10.10	502	Connected	
	10	Not Configured	N/A	E1	N/A	N/A	N/A	N/A	<u>Idle</u>	
	11	Not Configured	N/A	E1	N/A	N/A	N/A	N/A	<u>Idle</u>	
	12	Not Configured	N/A	E1	N/A	N/A	N/A	N/A	Idle	
	13	Not Configured	N/A	E1	N/A	N/A	N/A	N/A	<u>Idle</u>	
	14	Not Configured	N/A	E1	N/A	N/A	N/A	N/A	<u>Idle</u>	
	15	Not Configured	N/A	E1	N/A	N/A	N/A	N/A	<u>Idle</u>	
				E1 I	Address = 10.	10.10.12				

The Home page shows an overview of the 16 sockets. Clicking on a 'Status' link moves to a detail page for that socket.

# Socket Status Page

Each socket includes a page that shows the last 20 'window' messages for both PLC Inbound and Outbound.
TCPOpenTest	Lab	×								Scott		C		×
← → C ⋔	i 🗋 10.1	0.10.12:8	1/SockPag	ge/00/						ŝ	ABP	Į.		Ξ
<u>Home</u> Configuration					TC	POpen	Fest La	ab						-
Statistics						Socket Nur	nber 0							
<u>Help</u>		Sockets												
		Numb	er Type	TCP/UDP	UCM Port	Local IP	Local Port	Remote IP	Remote Port	Statu	IS			
		0	Server	TCP	E2	192.168.30.12	502	10.10.10.10	2333	Connec	ted			
	Last In Window Data													
	Number	Length	Handshal	ke Time	Stamp			Data (h	ex)					
	6	12	44882	03/18/20	080 3.586	2A 41 00 00 00 06 02 03 00 46 00 0A								
	7	33	44878	03/18/20 13:44:28	080 3.002	2A 40 00 00 00 00 00 00 00 00 00 00 00 00	1B 02 10 00 0 00 00	C8 00 0A 14	00 00 00 00 00	00 00 00	00 (	00 00	00	
	8	12	44874	03/18/20 13:44:27	)80 7.465	2A 3F 00 00 00	06 02 03 00 4	46 00 0A						
	9	33	44869	03/18/20 13:44:20	080 5.791	2A 3E 00 00 00 00 00 00 00 00 00	1B 02 10 00 0 00 00	C8 00 0A 14	00 00 00 00 00	00 00 0	0 00	00 00	00	
	10	12	44865	03/18/20 13:44:20	080 5.268	2A 3D 00 00 00	06 02 03 00	46 00 0A						
	11	33	44861	03/18/20 13:44:25	)80 5.744	2A 3C 00 00 00 00 00 00 00 00 00	1B 02 10 00 0 00 00	C8 00 0A 14	00 00 00 00 00	00 00 0	0 00	00 00	00	
	12	12	44857	03/18/20 13:44:25	080 5.074	2A 3B 00 00 00	06 02 03 00 -	46 00 0A						
	13	33	44852	03/18/20 13:44:24	080 4.388	2A 3A 00 00 00 00 00 00 00 00 00	1B 02 10 00 0 00 00	C8 00 0A 14	00 00 00 00 00	00 00 0	0 00	00 00	00	
	14	12	44846	03/18/20 13:44:23	080 3.585	2A 39 00 00 00 0	06 02 03 00 4	46 00 0A						

The data is shown in hexadecimal. The timestamp is based on the UCM's RTC.

### 12 Debug Telnet Server

A simple telnet server is included in the VALVES application. The TCP port number that the telnet server listens on is controlled by the setting in the DTM editor. Possible values are:

- 0 = Disabled
- 23 = Standard TCP port for telnet servers
- 24 = Default (in case the VALVES PLC code is running a telnet server)

	PMEUCM0202				
1	Module				
1000	01.09				
Identity E	thernet Configuration	Application Configura	ation	Process Data	
		0			
Parame	eter Name			Current Value	Default Valu
	- Default Gateway			0.0.0.0	0.0.0.0
<b>□</b> -묾	E2 Ethernet Port				
	IP Address			192.168.30.12	10.10.10.11
	- Subnet Mask			255.255.255.0	255.0.0.0
	Default Gateway			192.168.30.1	0.0.0.0
中品	UCM OS Settings				
	- OS Modbus/TCP Se	erver Port (QLOAD)		503	503
	Debug Web Server	Port		81	81
	Debug TELNET Ser	ver Port		24	24
	OS Max TCP Segme	ent Size		1452	1452
	OS TCP Keep Alive	Time		10	10
	- Location Name (10	characters Max)		Test Lab	
노모	S1 Serial Dort				
- Descri	ntion				

Note: Remember to do a 'Build Changes' and then 'Transfer to FDR server' after

modifications to the DTM. The UCM will need to be rebooted after the update.

The Microsoft telnet client is not always installed on PCs. Most systems may enable the telnet client by:

Control Panel > Programs and Features > Turn Windows features on and off.

To use the standard Microsoft telnet client, simply open a command prompt and enter: >telnet 10.10.10.12 24

where 10.10.10.12 is the target IP Address and 24 is the target TCP port number.

Con Telnet 10.10.10.12		- C	x
<pre>\$[1] Paused. Press space bar to continue \$[8] is connected. \$[4] is connected. \$[8] is connected. \$[8] is connected. \$[9] is connected.</pre>			^
0 Keystroke=0			
S[0] Resumed. Press space bar to pause s[0] is connected. s[4] is connected. s[8] is connected. s[9] is connected.			
RX(0):2E 94 00 00 00 1B 02 10 00 C8 00 0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00		
TX(0):2E 94 00 00 00 06 02 10 00 C8 00 0A			
RX(0):2E 95 00 00 00 06 02 03 00 46 00 0A			
TX(0):2E 95 00 00 00 17 02 03 14 00 00 00 00 00 00 00 00 00 00 00 00 00			
RX<0>:2E 96 00 00 00 1B 02 10 00 C8 00 0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00		
TX(0):2E 96 00 00 00 06 02 10 00 C8 00 0A			
RX{0>:2E 97 00 00 00 06 02 03 00 46 00 0A			
TX(8):2E 97 00 00 00 17 02 03 14 00 00 00 00 00 00 00 00 00 00 00 00 00			
RX(8):2E 98 00 00 00 1E 02 10 00 C8 00 0H 14 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00		
1/10/-22 70 00 00 00 00 02 10 00 00 00 00			
TX(A):2E 99 AA AA AA 17 A2 A3 14 AA			
RX(0):22 9A 00 00 00 1B 02 10 00 C8 00 0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00		
TX<0>:2E 9A 00 00 00 06 02 10 00 C8 00 0A			
RX(0):2E 9B 00 00 00 06 02 03 00 46 00 0A			
TX<0>:2E 9B 00 00 00 17 02 03 14 00 00 00 00 00 00 00 00 00 00 00 00 00			
RX(0):2E 9C 00 00 00 1B 02 10 00 C8 00 0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00		
TX(0):2E 9C 00 00 00 06 02 10 00 C8 00 0A			
RX(0):2E 9D 00 00 00 06 02 03 00 46 00 0A			
TX(0):2E 9D 00 00 00 17 02 03 14 00 00 00 00 00 00 00 00 00 00 00 00 00			
RX(0):2E 9E 00 00 00 1B 02 10 00 C8 00 0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00		
TX(0):2E 9E 00 00 00 06 02 10 00 C8 00 0A			
Keystroke =			
Paused. Press space bar to continue s[0] is connected. s[4] is connected. s[8] is connected. s[9] is connected.			
ζ.			> .:

The following keystrokes adjust the operation of the telnet server:

- Space Bar pauses/starts the display motion. It also shows a list of the connected sockets.
- 0 through 9 enables/disables sockets 0 through 9 inclusive
- A through F enables/disables sockets 10 through 15 inclusive. May be either upper or lower case.

- P- forces a read of the prm file from the PTK board.

# 13 Modbus/TCP Server Example



This example uses the ST segment VALVES\_mbtcp\_server\_w0.xst code to allow external Modbus/TCP clients (zapreg32) to read+write holding registers in a virtual slave inside the M580 CPU.

Build a new project with the CPU and the PMEUCM in the local rack, slot 3.



#### Unlock the Security in the CPU.

0.0 : EIO : CommHeadRIODIG	O16L2											
RIO DIO Communicator Head												
		1	1	1 mer		Low		1				
Channel 0	C Security	🕕 IPConfig	7 0 RSTP	C SNMP			ServicePa	nt				
	- Global policy						1					
			Enforce Sec	urity					Unlock Sec	urity		
	Services										_	
		FTP : D	Disabled	$\sim$	D	HCP / BO	DOTP :	Disa	oled		$\sim$	
	-	IFTP : D	Disabled	$\sim$		5	SNMP :	Disa	oled		$\sim$	
		TTP : D	Visabled	~			EIP :	Disa	bled		~	
	Access Cont	rol										
	Access Cont Enabled	rol	~									
	Access Cont Enabled Subnet	rol IP Add	v Iress	Subnet ma	ısk [	FTP	TFTP	HTTP	Port502	EIP	SNMP	
	Access Cont Enabled Subnet Yes	rol IP Add 192.168	V Iress 8.10.1	Subnet ma 255.255.0	15k	FTP	TFTP	HTTP	Port502	EIP	SNMP	î
	Access Cont Enabled Yes V No V	IP Add 192.168	Iress           8.10.1	Subnet m. 255.255.(	ısk	FTP	TFTP	HTTP	Port502	EIP	SNMP	
	Access Cont Enabled Yes V No V No V	IP Add 192.165	V Iress 8.10.1	Subnet ma 255.255.(	15 <b>k</b>				Port502		SNMP	· ·
	Access Cont Enabled Yes X No X No X No X	IP Add 192.165	s.10.1	Subnet m: 255.255.(	<u>ISK [</u> J.O				Port502		SNMP	
	Access Cont Enabled Yes X No X No X No X	IP Add 192.165	ress 8.10.1	Subnet m. 255.255.(	•sk   .0				Port502			
unction:	Access Cont Enabled Yes X No X No X No X No X No X	IP Add 192.16	V Iress	Subnet m. 255.255.(	15k				Port502			

### Add the VALVES DTM.

DTM Browser
Host PC Host PC Hos

Set the "Identifier" to match the Rack+Slot number of the PMEUCM.

BMEP58_ECPU_EXT Communication BME P58 4040		Schr
Channel Properties TCP/IP Services Address Server EtherNet/IP Local Slaves Local Slave 1 ltems Local Slave 2 ltems Local Slave 3	Properties Address Setting IP Configuration IP Address: Subnet Mask: Gateway:	192 . 168 . 10 . 3         255 . 255 . 0 . 0         192 . 168 . 10 . 1
Items Device List [513] PME_UCM_0202_TCPOPEN < EI Logging	Address Server DHCP for this device: Identified by: Identifier:	Enabled V Device Name V Mx80_03_PMEUCM02

Edit the DTM Application Configuration to set the new IP Address values for E1 and E2. E1 is now 192.168.0.12 while E2 is 10.10.10.13.

PMEUCM0202						
Module						
01.09						_
entity Ethernet Configuration Application Configu	ration	Process Data				
						1
Parameter Name		Current Value	Default Value	Unit	Minimum Value	-
P 🖧 E1 Ethernet Port						
IP Address	1	192.168.0.12	10.10.10.10			
Subnet Mask	1	255.255.0.0	255.0.0.0			
Default Gateway		0.0.0.0	0.0.0.0			
E2 Ethernet Port						
IP Address	1	10.10.10.13	10.10.10.11			
		255.0.0.0	255.0.0.0			
Default Gateway		0.0.00	0.0.00			
UCM OS Settings						
OS Modbus/TCP Server Port (QLOAD)		503	503			
Debug Web Server Port		81	81			
Debug TELNET Server Port		24	24			
OS Max TCP Seament Size		1452	1452			
<					>	
			ОК	Cancel	Apply	

Import the ST sections for the Timer and MBTCP Server W0.



Build the Application and download it into the CPU.

Set the CPU to RUN.

Transfer the prm file to the FDR server by right clicking the UCM tree element in the DTM browser and choosing Device Menu > Additional Features > Transfer to FDR Server.

Reboot the PMEUCM card.

After the card boots and attaches to the M580, the screen should look like this:

TCPOpen								
0	SV	8						
1	SU	9						
2	SD	10						
З	SV	11						
4	SU	12						
5	SV	13						
6	ŝ	14						
7	SU	15						
С	onn	ect	ed					

We see sockets 0 through 7 are listening but not connected. Sockets 8 through 15 are not configured.

Now, open a command prompt and enter the following command:

Zapreg32 is a simple wind32 console application that can be a Modbus/TCP client. The command line above tells it to connect to IP Address 10.10.10.10 using TCP port 502 and poll slave 1 and suppress status register polling.



Press Enter and the screen should look like this:

Com Com	mand Pr	rompt - zapr	eg32 192.16	8.0.12 1 -s				×
Niobrar	a R&D			SY/MAX	Register Viewer		310ct	t2011
REGSTR	HEX	UNSIGN	SIGNED					
110517 1 2 3 4 5 6 7 8 9 10 112 12 14 15 16 17 18		005160 83 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	51GNED 0 83 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		Sy/Max Register View Up and Down arrows to s Page Up and Page Down t Left and Right arrows t 0.9, AF to enter new Up/Down Arrow to build Enter to update without F10 to acknowledge erro Escape to exit.	er o chang o selec value, block v moving r,	registi je by : t moda write, J	er, 10, e,
19 20	0000 0000	0 0	0 0					

This shows the first 20 Holding registers in Slave 1. The data is shown in Hex, Unsigned, and Signed values. Register number 3 should be incrementing. This register is being incremented with each Modbus read by the PLC code.

Note, Zapreg32 is a standard Modbus poller. It uses the older 'Modicon' standard of starting at Holding Register 1. The M580 code is written to use the newer 'Tele' standard of starting at register 0. A look at the ST code in State 20 shows:

Modbus\_Slave\_Data[2] := Modbus\_Slave\_Data[2] + 1;

which explains why Zapreg32 is showing register 3 incrementing.

Back in Unity Pro, open the Variables & FB Instances and right click on the Modbus Slave Data structure and Initialize Animation Table.



We see that there within this Modbus virtual slave, there are 1000 registers and register [2] is incrementing.

lable[l/O ne	twork Editor Skelet	onj				
<u>M</u> odification	Force					
Name		▼ Value	Туре	<b>.</b>	Comment	^
🖃 📕 Modbus	_Slave_Data		ARRA	([0999]		
🔶 Moo	dbus_Slave_Data[0]	0	INT			_
🔶 Moo	dbus_Slave_Data[1]	0	INT			
🔶 Moo	dbus_Slave_Data[2]	12503	INT			
🔶 Moo	dbus_Slave_Data[3]	0	INT			
🔶 Moo	dbus_Slave_Data[4]	0	INT			
🔶 Moo	dbus_Slave_Data[5]	0	INT			
🔶 Moo	dbus_Slave_Data[6]	0	INT			
🔶 Moo	dbus_Slave_Data[7]	0	INT			
🔶 Moo	dbus_Slave_Data[8]	0	INT			
🔶 Moo	dbus_Slave_Data[9]	0	INT			
🔶 Moo	dbus_Slave_Data[10	] 0	INT			
🔶 Moo	dbus_Slave_Data[11	] 0	INT			
🔶 Moo	dbus_Slave_Data[12	] 0	INT			
🔶 Moo	dbus_Slave_Data[13	] 0	INT			
🔶 Moo	dbus_Slave_Data[14	] 0	INT			
🔶 Moo	dbus_Slave_Data[15	] 0	INT			
🔶 Moo	dbus_Slave_Data[16	] 0	INT			
Mor	thus Slave Data[17	1 0	INT			×

Table[I/O network Editor Skeleton]		[	- • ×
Modification Eorce 2	ئو لا [ تو	- Main -	표 위 📘
Name -	Value	Туре 💌	Comment 🔺
🖃 🖳 Modbus_Slave_Data		ARRAY[0999]	
Modbus_Slave_Data[0]	0	INT	
Modbus_Slave_Data[1]	0	INT	
Modbus_Slave_Data[2]	13677	INT	
Modbus_Slave_Data[3]	0	INT	
Modbus_Slave_Data[4]	0	INT	
Modbus_Slave_Data[5]	1234	INT	
Modbus_Slave_Data[6]	0	INT	
Modbus_Slave_Data[7]	0	INT	
Modbus_Slave_Data[8]	0	INT	
Modbus_Slave_Data[9]	0	INT	
Modbus_Slave_Data[	0	INI	
Modbus_Slave_Data[	0	INT	
Modbus_Slave_Data[	0	INT	
Modbus_Slave_Data[	0		
Modbus_Slave_Data[	0		
Modbus_Slave_Data[	0		
	0	INT	
			•

Select "Modification" and change the value of [5] from 0 to 1234.

Now go back to the zapreg32 window and we see that register 6 has the value 1234.

Con Con	nmand Pr	ompt - zapr	eg32 192.16	58.0.12 1 -s				×
Niobrai	•a R&D			SY/MAX	Register Viewer		310ct	;2011
REGSTR	HEX	UNSIGN	SIGNED					
1	0000	Ø	Ø					
2	0000	Ø	Ø		Sy/Max Register View	ver		
3	36C9	14025	14025					
4	0000	Ø	Ø		Up and Down arrows to s	select r	egiste	er,
5	0000	0	00		Page Up and Page Down t	to chang	te pa j	LØ,
6	04D2	1234	1234		Left and Right arrows t	to selec	t mode:	е,
2	ଉଉଉଡ	<u>0</u>	0		09, AF to enter new	v value,		
8	ดดดด	<u>N</u>	<u>N</u>		Up/Down Arrow to build	block w	rite,	
	บบบบ	<u>N</u>	<u>N</u>		Enter to update without	; moving	(a	
10	ଉଉଉଡ	<u></u>	0		F10 to acknowledge erro	or,		
11	ดดดด	្រុ	<u>N</u>					
12	บบบบ	ក	ត		Escape to exit.			
13	ดดดด	ខ	<u></u>					
14	บบบบ	<u>N</u>	<u>N</u>					
15	ิดดดด	ត	<u>N</u>					
16	ดดดด	្រុ	<u>N</u>					
17	ଉଉଉଡ	<u>0</u>	ឲ					
18	0000	0	0					
19	บบบบ	0	0					
20	0000	0	Ø					

Move the cursor in the zapreg window by using the arrow keys. Move the cursor down to register 7 and type in the value 54321 in the unsigned column.

Cit. Com	mand Pr	ompt - zapr	eg32 192.16	i8.0.12 1 -s				×
Niobrar	a R&D			SY/MAX	Register Viewer		310ct	2011
REGSTR	HEX	UNSIGN	SIGNED					
1 23456789 101123144 1123144 115617	0000 0000 3890 0000 04D2 04D2 0000 0000 0000 0000 000	0 14487 0 1234 54321 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 14487 0 1234 -11215 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		Sy/Max Register Vie Up and Down arrows to Page Up and Page Down Left and Right arrows Ø9, AF to enter ne Up/Down Arrow to build Enter to update withou F10 to acknowledge err Escape to exit.	wer select r to chang to selec w value, block w t moving or,	regista je by j t mode wite, (,	er, 10,
18 19 20	0000 0000 0000	0 0 0	0 0 0					

Now go look at the Unity table.

Table[I/O network Editor Skeleton]			- • •
Modification Force 2	- F   F - F	- in 🖉 📕 😕	H 21 🛛 🖁
Name 👻	Value	Туре 👻	Comment 🔺
🖃 🗧 Modbus_Slave_Data		ARRAY[0999]	
Modbus_Slave_Data[0]	0	INT	
Modbus_Slave_Data[1]	0	INT	
Modbus_Slave_Data[2]	14739	INT	
Modbus_Slave_Data[3]	0	INT	
Modbus_Slave_Data[4]	0	INT	
Modbus_Slave_Data[5]	1234	INT	
Modbus_Slave_Data[6]	-11215	INT	
Modbus_Slave_Data[7]	0	INT	
Modbus_Slave_Data[8]	0	INT	
Modbus_Slave_Data[9]	0	INT	
Modbus_Slave_Data[	0	INT	
Modbus Slave Dataí	0	INT	×

The data -11215 is now in register [6] as expected.

Take a look at the front LCD panel. Socket 1 is now filled in to indicate an active connection. (It could have been any of sockets 0, 1, 2, or 3 that where highlighted. It just depends on which one was connected when Zapreg32 started.)



Use the joystick to select "Sockets" and then down arrow to socket 2. This shows connected, E1, TCP, port 502, remote IP 192.168.0.200 and that the server is in State 20 (waiting for Modbus query).



Open a web browser and enter the url of 192.168.0.12:81.

The page shows that socket 2 is connected to the PC. The PC is at IP Address 192.168.0.200.

Clicking on the "Connected" link for socket 2 pulls up the last 20 inbound and outbound messages.

⇒ C fi	🗋 192.	168.0.12:81						<u>ک</u>	💵 🖪 🖸			
<u>ie</u> Equivation				1	ГСРОр	en						
iguration					PLC in RUI	V						
stics	Sockets											
	Number	Туре	TCP/UDP	UCM Port	Local IP	Local Port	Remote IP	Remote Port	Status			
	0	Server	TCP	E1	192.168.0.12	502	0.0.0.0	0	<u>Listening</u>			
	1	Server	TCP	E1	192.168.0.12	502	192.168.0.200	60434	Connected			
	2	Server	TCP	E1	192.168.0.12	502	0.0.0.0	0	<u>Listening</u>			
	3	Server	TCP	E1	192.168.0.12	502	0.0.0.0	0	<b>Listening</b>			
	4	Server	TCP	E2	10.10.10.13	502	0.0.0.0	0	<u>Listening</u>			
	5	Server	TCP	E2	10.10.10.13	502	0.0.0.0	0	Listening			
	6	Server	TCP	E2	10.10.10.13	502	0.0.0.0	0	<b>Listening</b>			
	7	Server	TCP	E2	10.10.10.13	502	0.0.0.0	0	<b>Listening</b>			
	8	Not Configured	N/A	E1	N/A	N/A	N/A	N/A	<u>Idle</u>			
	9	Not Configured	N/A	E1	N/A	N/A	N/A	N/A	<u>Idle</u>			
	10	Not Configured	N/A	E1	N/A	N/A	N/A	N/A	<u>Idle</u>			
	11	Not Configured	N/A	E1	N/A	N/A	N/A	N/A	<u>Idle</u>			
	12	Not Configured	N/A	E1	N/A	N/A	N/A	N/A	<u>Idle</u>			
	13	Not Configured	N/A	E1	N/A	N/A	N/A	N/A	<u>Idle</u>			
	14	Not Configured	N/A	E1	N/A	N/A	N/A	N/A	<u>Idle</u>			
	15	Not Configured	N/A	E1	N/A	N/A	N/A	N/A	<u>Idle</u>			

Click on the "Connected" link for Socket 1.

P TCPOpen	~							Scott	- 🗆	)		
← → C ↑	192.16	8.0.12:8	1/SockPa	age/01/				 22	ABP 📑			
<u>Home</u> <u>Configuration</u>		TCPOpen Socket Number 1										
Statistics												
<u>Help</u>	N. I	T	TODA		Sock	ets						
	Number	Type	TCP/U	F1	192 168 0 12	Local Pol	192 168 0 200	60434	Status			
		Server	ICr		192.106.0.12	502	192.108.0.200	00434	Connected			
			L on -th	Handahal	Last In Win	dow Data	D. (	(herr)				
	N	umber	Length	nandshake	11me Stan	np	Data	(nex)	14			
	/		12	19000	04/20/2016 20:13	8:56.495 7	9 FD 00 00 00 06	01 03 00 00 00	14			
	8		12	18999	04/20/2016 20:13	8:56.370 7	9 FC 00 00 00 00 06	01 03 00 00 00	14			
	9		12	10990	04/20/2016 20:12	8.30.229 7	9 FB 00 00 00 00	01 03 00 00 00	14			
	11	1	12	18997	04/20/2016 20:1	8.55.078 7	9 FA 00 00 00 00	01 03 00 00 00	14			
	12	,	12	18995	04/20/2016 20:1	8-55 863 7	9 F8 00 00 00 06	01 03 00 00 00	14			
	13	3	12	18994	04/20/2016 20:1	8:55 747 7	9 F7 00 00 00 06	01 03 00 00 00	14			
	14	4	12	18993	04/20/2016 20:1	8:55.631 7	9 F6 00 00 00 06	01 03 00 00 00	14			
	15	5	12	18992	04/20/2016 20:1	8:55.497 7	9 F5 00 00 00 06	01 03 00 00 00	14			
	10	5	12	18991	04/20/2016 20:1	8:55.374 7	9 F4 00 00 00 06	01 03 00 00 00	14			
	17	7	12	18990	04/20/2016 20:1	8:55.259 7	9 F3 00 00 00 06	01 03 00 00 00	14			
	18	8	12	18989	04/20/2016 20:1	8:55.143 7	9 F2 00 00 00 06	01 03 00 00 00	14			
	19	9	12	18988	04/20/2016 20:1	8:55.027 7	9 F1 00 00 00 06	01 03 00 00 00	14			
	0		12	18987	04/20/2016 20:1	8:54.901 7	9 F0 00 00 00 06	01 03 00 00 00	14			
	1		12	18986	04/20/2016 20:1	8:54.779 7	9 EF 00 00 00 06	01 03 00 00 00	14			
	2		12	18985	04/20/2016 20:1	8:54.663 7	9 EE 00 00 00 06	01 03 00 00 00	) 14			
	3		12	18984	04/20/2016 20:1	8:54.530 7	9 ED 00 00 00 06	01 03 00 00 00	0.14			
	4		12	18983	04/20/2016 20:1	8:54.411 7	9 EC 00 00 00 06	01 03 00 00 00	) 14			
	5		12	18982	04/20/2016 20:1	8:54.295 7	9 EB 00 00 00 06	01 03 00 00 00	) 14			
	6		12	18981	04/20/2016 20:1	8:54.176 7	9 EA 00 00 00 06	01 03 00 00 00	0.14			
					Lest Out Wi	ndow Data						

This page shows the time-stamped log of the last 20 messages sent to the PLC on Window W0. The hexadecimal data shows the FC 03 message to slave 1 starting at register 0 with a count of 20.

Scrolling down the window shows the reply data returned from the M580.

TCPOpen		×			Scott _ 🗆 🗡
⇒ C f	192.	168.0.12	:81/SockPac	ie/01/	
		4	12 1	0703	04/20/2010 20.18.34.411 73 EC 00 00 00 00 01 03 00 00 01 14
		5	12 1	8982	04/20/2016 20:18:54.295 79 EB 00 00 00 06 01 03 00 00 00 14
		6	12 1	8981	04/20/2016 20:18:54.176 79 EA 00 00 00 06 01 03 00 00 00 14
					Last Out Window Data
	Number	Length	Handshake	Time Stan	np Data (hex)
	8	49	18999	04/20/2016 20:18:56.4	5         79 FC 00 00 00 2B 01 03 28 00 00 00 04 A36 00 00 00 04 D2 D4           31 00 00 00 00 00 00 00 00 00 00 00 00 00
	9	49	18998	04/20/2016 20:18:56.2	5         79 FB 00 00 00 2B 01 03 28 00 00 00 04 A 35 00 00 00 00 4D 2D 4           31 00 00 00 00 00 00 00 00 00 00 00 00 00
	10	49	18997	04/20/2016 20:18:56.1	5 64 79 FA 00 00 00 2B 01 03 28 00 00 00 04 A 34 00 00 00 00 04 D2 D4 31 00 00 00 00 00 00 00 00 00 00 00 00 00
	11	49	18996	04/20/2016	5 41 79 F9 00 00 00 2B 01 03 28 00 00 00 04 A 33 00 00 00 00 4D 2D 4 31 00 00 00 00 00 00 00 00 00 00 00 00 00
	12	49	18995	04/20/2016 20:18:55.9	5 19 79 F8 00 00 00 2B 01 03 28 00 00 00 04 A 32 00 00 00 04 D2 D4 31 00 00 00 00 00 00 00 00 00 00 00 00 00
	13	49	18994	04/20/2016 20:18:55.8	5 00 79 F7 00 00 00 2B 01 03 28 00 00 00 04 A 31 00 00 00 04 D2 D4 31 00 00 00 00 00 00 00 00 00 00 00 00 00
	14	49	18993	04/20/2016 20:18:55.6	5         79 F6 00 00 00 2B 01 03 28 00 00 00 04 A 30 00 00 00 00 04 D2 D4         31 00 00 00 00 00 00 00 00 00 00 00 00 00
	15	49	18992	04/20/2016 20:18:55.5	5 59 79 F5 00 00 00 2B 01 03 28 00 00 00 04 A 2F 00 00 00 04 D2 D4 31 00 00 00 00 00 00 00 00 00 00 00 00 00
	16	49	18991	04/20/2016	5 32 32 31 00 00 00 00 00 00 00 00 00 00 00 00 00
	17	49	18990	04/20/2016	5 15 179 F3 00 00 00 2B 01 03 28 00 00 00 04 A 2D 00 00 00 00 04 D2 D4 31 00 00 00 00 00 00 00 00 00 00 00 00 00
	18	49	18989	04/20/2016 20:18:55.1	5         79 F2 00 00 00 2B 01 03 28 00 00 00 04 A 2C 00 00 00 00 04 D2 D4           31 00 00 00 00 00 00 00 00 00 00 00 00 00

# 14 Modbus/TCP Server+Client Example2

