3 Chapter 3 EtherNet/IP

Introduction to EtherNet/IP

Today, with the introduction of EtherNet/IP (Industrial Protocol), a user can collect, configure, and control using one protocol. EtherNet/IP is a network communication standard capable of handling large amounts of data at speeds of 10 Mbps or 100 Mbps, and at up to 1,500 bytes per packet. The specification uses an open protocol at the application layer.

EtherNet/IP makes use of the standard off-the-shelf Ethernet chip sets and the currently installed physical media (hardware connections) and incorporates what is known today as the Common Industrial Protocol (CIP); an open protocol at the application layer fully managed by the Open DeviceNet Vendors Association (ODVA, http://www.odva.org). CIP is the critical component providing the ability to collect, configure, and control utilizing both implicit messaging (real-time I/O messaging), and explicit messaging (information/configuration messaging), with full support for peer-to-peer and multi-master configurations.

PM Connectivity over EtherNet/IP

To establish communications with the PLC the EZ-ZONE™ PM controller must be connected to the network, where it will either assume or be given an IP address. There are two ways in which an IP address can be established: Dynamic Host Configuration Protocol (DHCP, where a DHCP server on the network provides an IP address); or a fixed IP address (manually entered). The PM controller's default is set to DHCP. To change the IP addressing method to fixed IP follow the steps below:

1. Push and hold the up and down arrow keys on the front panel for six seconds to go the the Setup Menu.
2. Push the up or down arrow key until [CoM] (Communications Menu) appears in upper display and [SEt] in the lower display.
3. Push the green Advance Key ◉ to enter the Communications Menu [CoM].
4. Push up arrow key to go to the Communications 2 Submenu. The upper display shows [2], and the lower display shows [CoM].
5. Push the Advance Key ◉ until the upper display shows [dhCP] and lower display shows [iP.M].
6. Push the up arrow to change to Fixed Address

Note: Excessive writes to the PM may cause premature EEPROM failure. For more detail see the section entitled "Saving Settings to Nonvolatile Memory".

EtherNet/IP Indicator Lights

The PM has four indicator lights on the top of the controller, all of which are used with EtherNet/IP. The characteristics of the Module Status and Network Status LED's are defined by Open DeviceNet Vendors Association (ODVA), while the Active and Link indicator lights are defined in the Ethernet specification.

Module Status Indicator

Table 3.1

Indicator State	Summary	Requirement
Steady Off	No power	If no power is supplied to the device, the module status indicator shall be steady off.
Steady Green	Device operational	If the device is operating correctly, the module status indicator shall be steady green.
Flashing Green	Standby	If the device has not been configured, the module status indicator shall be flashing green.
Flashing Red	Minor fault	If the device has detected a recoverable minor fault, the module status indicator shall be flashing red. NOTE: An incorrect or inconsistent configuration would be considered a minor fault.
Steady Red	Major fault	If the device has detected a non-recoverable major fault, the module status indicator shall be steady red.
Flashing Green / Red	Self-test	While the device is performing its power up testing, the module status indicator shall be flashing green / red.

Network Status Indicator

Table 3.2

Steady Off	Not powered, no IP address	If the device does not have an IP address (or is powered off), the network status indicator shall be steady off.
Flashing Green	No connections	If the device has no established connections, but has obtained an IP address, the network status indicator shall be flashing green.
Steady Green	Connected	If the device has at least one established connection (even to the Message Router), the network status indicator shall be steady green.
Flashing Red	Connection timeout	If one or more of the connections in which this device is the target has timed out, the network status indicator shall be flashing red. This shall be left only if all timed out connections are reestablished or if the device is reset.
Steady Red	Duplicate IP	If the device has detected that its IP address is already in use, the network status indicator shall be steady red.
Flashing Green /Red	Self-test	While the device is performing its power up testing, the network status indicator shall be flashing green / red.

Link Status Indicator

Table 3.3

Steady Off	Not powered, unknown link speed	If the device cannot determine link speed or power is off, the network status indicator shall be steady off.
Red	Link speed = 10 Mbit	If the device is communicating at 10 Mbit, the link LED will be red..
Green	Link speed = 100 Mbit	If the device is communicating at 100 Mbit, the link LED will be green.

Activity Status Indicator

Table 3.4

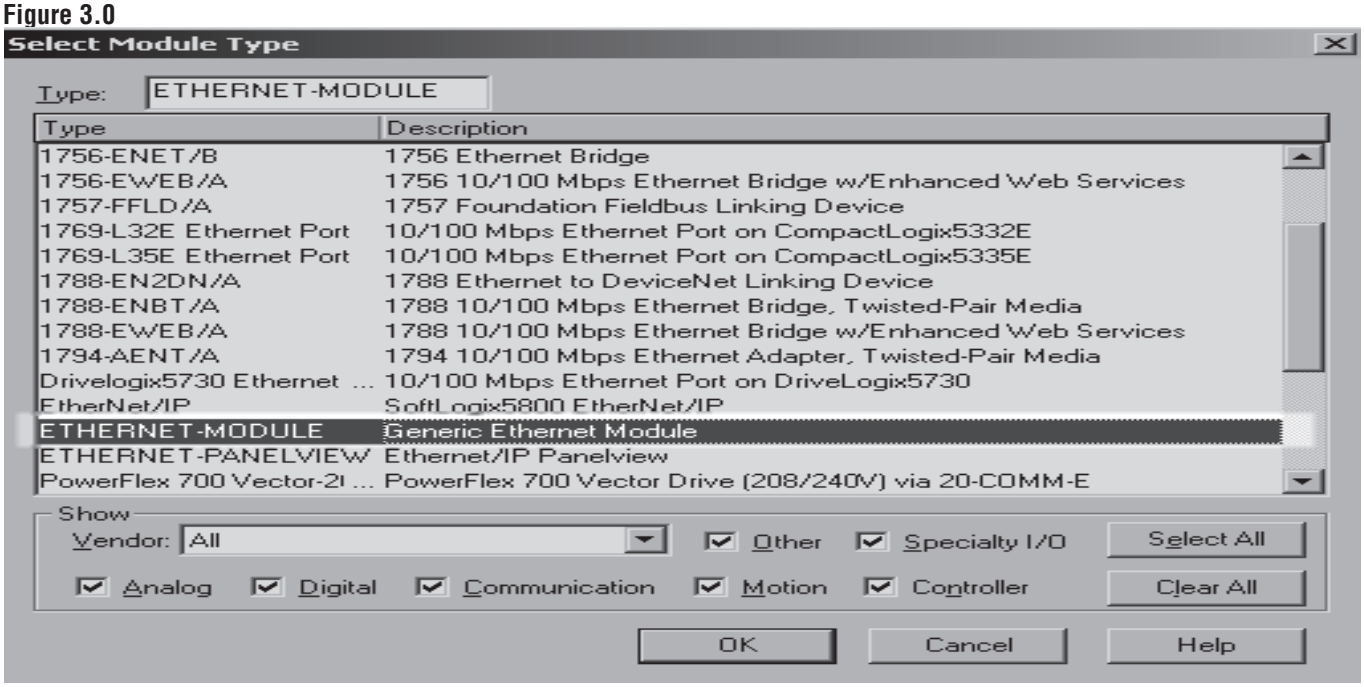
Flashing Green	Detects activity	If the MAC detects activity, the LED will be flashing green.
Red	Link speed = 10Mbit	If the MAC detects a collision, the LED will be red.

I/O Configuration using an Allen-Bradley Logix Family Processor

The setup steps may vary depending on the controller. The specific control used in the examples is a CompactLogix 1769-L32E. Follow the steps below to add and configure the PM as a generic Ethernet module.

1. In the I/O configuration, right click on the Ethernet Port, (in this case: 1769-L32E Ethernet Port LocalENB) and add a new module.

Figure 3.0

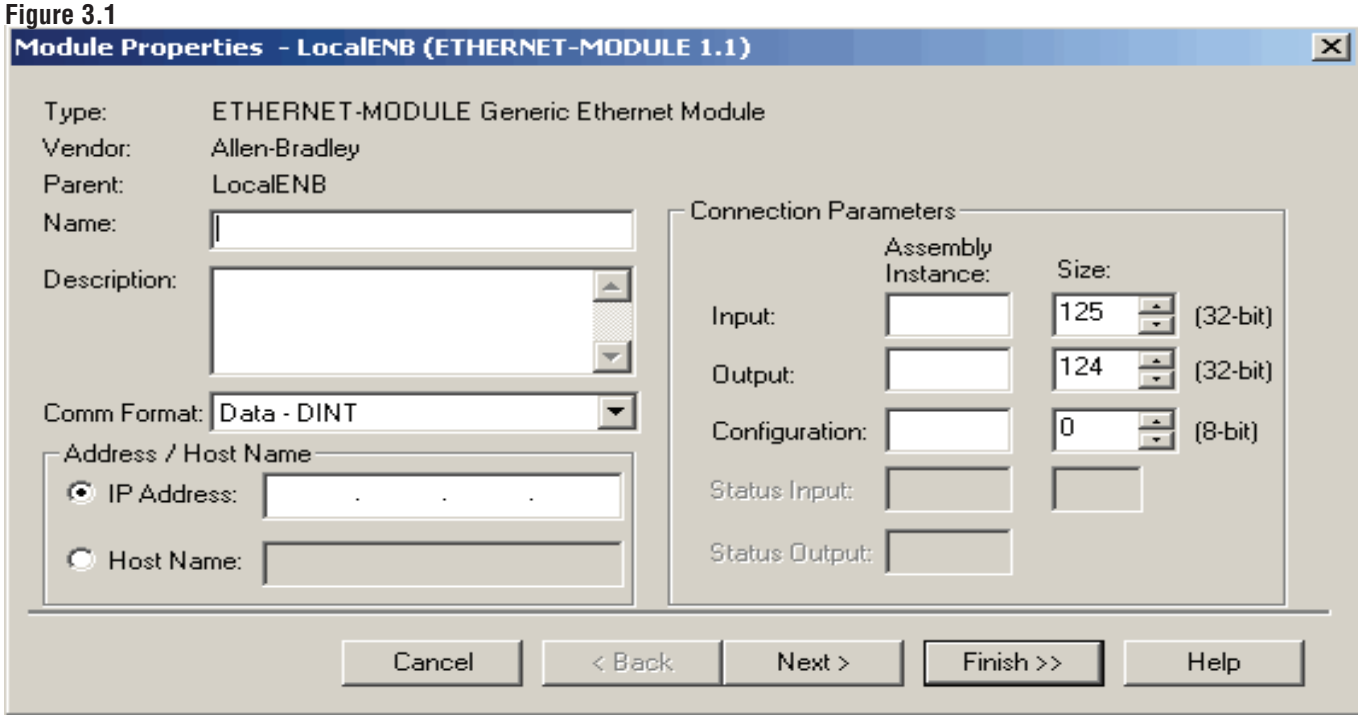


2. Select “Generic Ethernet Module” and click OK.

Configuring PM Properties using the RSLogix 5000

After clicking OK, the following screen will appear. You must complete all the fields in this screen except the description field.

Figure 3.1



Name

This field, will automatically be used as the controller name and will be used in the program when referencing PM inputs or outputs.

Description

No entry required.

Comm Format

As can be seen in the PM I/O assemblies below, the PM data formats depend on the tag name being written to or read from.

As can be seen in the chart below, the data types used by the PM vary. Although multiple "Comm Formats" can be configured, for ease in configuration and programming it is suggested that it be configured as INT. Configuration examples will follow.

IP Address

Enter here, the DHCP or fixed IP address previously acquired.

Assembly Instance

Input, PM to CompactLogix

This field identifies the Target to Originator (T → O) input assembly 0x65 (101 decimal).

Output, CompactLogix to PM

This field identifies Originator to Target (O → T) output assembly 0x64 (100 decimal).

Configuration

The PM does not use the configuration instance 0x80 (128 decimal), however it still needs to be entered here.

Assembly Size

The assembly size is dependent upon the "Comm Format."

T →O INT: 42 or DINT: 21

O → T INT: 40 or DINT: 20

The size for the configuration instance, although not used, will always be set to 0.

Table 3.5

Supported Attribute Data Types			
CIP	**PM**	**Access**	**Size (Bytes)**
USINT	UByte	RW	1
SINT	Byte	RW	1
UINT	UWord	RW	2
INT	Word	RW	2
UDINT	ULong	RW	4
DINT	Long	RW	4
REAL	Float	RW	4

Target to Originator (T to O) - Default Assembly

Table 3.6

Attribute Name	EIP Class ID	EIP Instance ID	EIP Attribute ID	Data Type
Analog Input 1 Process Value	104	1	1	REAL
Analog Input 1 Eror Status	104	1	2	DINT
Analog Input 2 Process Value	104	2	1	REAL
Analog Input 2 Error Status	104	2	2	DINT
Alarm 1 State	109	1	9	DINT
Alarm 2 State	109	2	9	DINT
Alarm 3 State	109	3	9	DINT
Alarm 4 State	109	4	9	DINT
Digital Input 5 Status	110	1	5	DINT
Digital Input 6 Status	110	2	5	DINT
Control Mode Active	151	1	2	DINT
Heat Power	151	1	13	REAL
Cool Power	151	1	14	REAL
Limit State	112	1	6	DINT
Profile Start	122	1	1	DINT
Profile Action Request	122	1	11	DINT
Active File	122	1	3	DINT
Active Step	122	1	4	DINT
Active Set Point	122	1	5	REAL
Step Time Remaining	122	1	9	REAL

In using the input assembly define the following sizes based on the configured “Comm Format” in RSLogix5000.

DINT: 21

INT: 42

Originator to Target (O to T) - Default Assembly

Table 3.7

Attribute Name	EIP Class ID	EIP Instance ID	EIP Attribute ID	Data Type
Loop Control Mode	151	1	1	DINT
Closed Loop Set Point	107	1	1	REAL
Open Loop Set Point	107	1	2	REAL
Alarm 1 High Set Point	109	1	1	REAL
Alarm 1 Low Set Point	109	1	2	REAL
Alarm 2 High Set Point	109	2	1	REAL
Alarm 2 Low Set Point	109	2	2	REAL
Alarm 3 High Set Point	109	3	1	REAL
Alarm 3 Low Set Point	109	3	2	REAL
Alarm 4 High Set Point	109	4	1	REAL
Alarm 4 Low Set Point	109	4	2	REAL
Profile Action Request	122	1	11	DINT
Profile Start	122	1	1	DINT
Heat Proportional Band	151	1	6	REAL
Cool Proportional Band	151	1	7	REAL
Time Integral	151	1	8	REAL
Time Derivative	151	1	9	REAL
Heat Hysteresis	151	1	11	REAL
Cool Hysteresis	151	1	12	REAL
Deadband	151	1	10	REAL

In using the output assembly define the following sizes based on the configured “Comm Format” in RSLogix5000.

DINT: 20

INT: 40

Note: Excessive writes to the PM may cause premature EEPROM failure. If using the O to T assembly it is recommended that EEPROM writes be disabled (factory default). For more detail see the section entitled "Saving Settings to Nonvolatile Memory".

Communications between ControlLogix & the EZ-ZONE™ PM

Configuring the PM enables both real-time I/O connections (implicit messaging) and non-time critical (explicit messaging) communications. Information will be transferred between the control and the PM using either implicit and or explicit connections. All implicit messages are sent and received cyclically at the rate of the Requested Packet Interval (RPI), where explicit messages are typically initiated via a message instruction in the control program. It is recommended that the RPI be set above 100ms. Generally, explicit messages are used as a tool for configuration. For example, to change the default T-to-O or O-to-T assembly structure in the PM from the factory defaults as defined above, the user would use an explicit message instruction.

Ladder Logic Examples

In the ladder logic examples that follow, please note how the PM and its associated tags were configured.

First, let's take a look at the "Comm Format" briefly discussed earlier with a recommendation to configure it as INT. In this section we will see why.

One of the advantages of using the Logix family of controls is that users can define their own data types. Creating two unique, user-defined data types (T to O and O to T) makes programming the PLC to communicate with the Watlow PM controller very easy. The name given for these data types is up to the user. In this example, the user-defined data types and styles were created to match the default PM O-to-T and T-to-O assemblies.

Notice in Figure 3.4 (PM T to O) that the first location is identified as "Device Status." This does not represent one of the 20 members, and it is required. Currently, if bit 16 is set to 1, as shown in figure 3.2 below (PM to PLC), it indicates valid communications between the Ethernet card and the PM. If set to 0, communications have failed.

Figure 3.2

Figure 3.3 & 3.4

Name: PM_O_to_T

Description: PLC to Watlow PM

Members: Data Type Size: 80 byte(s)

Name	Data Type	Style	Description
Ctrl_Mode	DINT	Decimal	PM Control Mode
CLSP	REAL	Float	CLSP
OLSP	REAL	Float	OLSP
ALm1H_SP	REAL	Float	Alm SP1 High
ALm1L_SP	REAL	Float	Alm SP1 Low
ALm2H_SP	REAL	Float	Alm SP2 High
ALm2L_SP	REAL	Float	Alm SP2 Low
ALm3H_SP	REAL	Float	Alm SP3 High
ALm3L_SP	REAL	Float	Alm SP3 Low
ALm4H_SP	REAL	Float	Alm SP4 High
ALm4L_SP	REAL	Float	Alm SP4 Low
Pro_Act_Req	DINT	Decimal	Profile_Action_Request
Pro_Strt	DINT	Decimal	Profile Start
H_PB	REAL	Float	Heat Proportional Band
C_PB	REAL	Float	Cool Proportional Band
Integral	REAL	Float	Integral
Derivative	REAL	Float	Derivative
On_Off_HHys	REAL	Float	Heat On-Off Hysteresis
On_Off_CHys	REAL	Float	Cool On-Off Hysteresis
DB	REAL	Float	PID DeadBand

Name: PM_T_to_O

Description: Watlow PM - Input to PLC

Members: Data Type Size: 84 byte(s)

Name	Data Type	Style	Description
Device_Stat	DINT	Binary	PM Device Status
PV1	REAL	Float	AI 1 Process Variable
In_Err_Stat1	DINT	Decimal	AI 1 Input Error Status
PV2	REAL	Float	AI 2 Process Variable
In_Err_Stat2	DINT	Decimal	AI 2 Input Error Status
Alm_Stat1	DINT	Decimal	Alarm1 Status
Alm_Stat2	DINT	Decimal	Alarm2 Status
Alm_Stat3	DINT	Decimal	Alarm3 Status
Alm_Stat4	DINT	Decimal	Alarm4 Status
Event1_State	DINT	Decimal	Event 1 Status
Event2_State	DINT	Decimal	Event 1 Status
PM_Ctrl_Mode	DINT	Decimal	PM_Control_Mode
H_Pwr	REAL	Float	Heat Output Power
C_Pwr	REAL	Float	Cool Output Power
Lim_State	DINT	Decimal	Limit State
RB_Strt_Pro	DINT	Decimal	Profile Start Readback
RB_Pro_Act_Req	DINT	Decimal	Profile Action Request Readback
Pro_Curr_File	DINT	Decimal	Profile - Current File
Pro_Curr_Stp	DINT	Decimal	Profile - Current Step
Pro_Prod_SP	REAL	Float	Profile - Produced Set Point
Pro_Rem_ST	REAL	Float	Profile - Remaining Step Time

Now, to use the new data types defined above. Two controller tags where created (see figure 3.5 & 3.6) and when prompted for the data type, the user-defined data types defined above were selected.

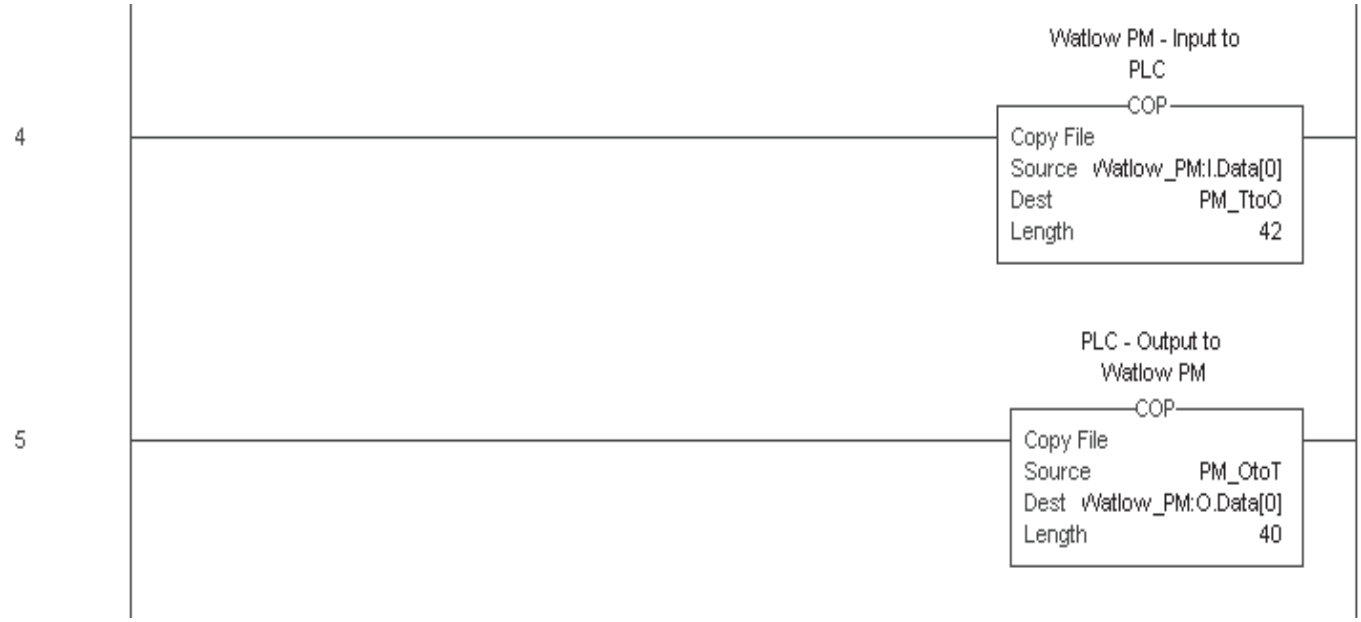
Figure 3.5 & 3.6

PM_TtoO		PM_T_to_O	Watlow PM - Input to PLC
PM_TtoO.Device_Stat	Binary	DINT	Watlow PM - Input to PLC PM Device Status
PM_TtoO.PV1	Float	REAL	Watlow PM - Input to PLC AI 1 Process Variable
PM_TtoO.In_Err_Stat1	Decimal	DINT	Watlow PM - Input to PLC AI 1 Input Error Status
PM_TtoO.PV2	Float	REAL	Watlow PM - Input to PLC AI 2 Process Variable
PM_TtoO.In_Err_Stat2	Decimal	DINT	Watlow PM - Input to PLC AI 2 Input Error Status
PM_TtoO.Alm_Stat1	Decimal	DINT	Watlow PM - Input to PLC Alarm1 Status
PM_TtoO.Alm_Stat2	Decimal	DINT	Watlow PM - Input to PLC Alarm2 Status
PM_TtoO.Alm_Stat3	Decimal	DINT	Watlow PM - Input to PLC Alarm3 Status
PM_TtoO.Alm_Stat4	Decimal	DINT	Watlow PM - Input to PLC Alarm4 Status
PM_TtoO.Event1_State	Decimal	DINT	Watlow PM - Input to PLC Event 1 Status
PM_TtoO.Event2_State	Decimal	DINT	Watlow PM - Input to PLC Event 1 Status
PM_TtoO.PM_Ctrl_Mode	Decimal	DINT	Watlow PM - Input to PLC PM_Control_Mode
PM_TtoO.H_Pwr	Float	REAL	Watlow PM - Input to PLC Heat Output Power
PM_TtoO.C_Pwr	Float	REAL	Watlow PM - Input to PLC Cool Output Power
PM_TtoO.Lim_State	Decimal	DINT	Watlow PM - Input to PLC Limit State
PM_TtoO.RB_Strt_Pro	Decimal	DINT	Watlow PM - Input to PLC Profile Start Readback
PM_TtoO.RB_Pro_Act_Req	Decimal	DINT	Watlow PM - Input to PLC Profile Action Request Readback
PM_TtoO.Pro_Curr_File	Decimal	DINT	Watlow PM - Input to PLC Profile - Current File
PM_TtoO.Pro_Curr_Stp	Decimal	DINT	Watlow PM - Input to PLC Profile - Current Step
PM_TtoO.Pro_Prod_SP	Float	REAL	Watlow PM - Input to PLC Profile - Produced Set Point
PM_TtoO.Pro_Rem_ST	Float	REAL	Watlow PM - Input to PLC Profile - Remaining Step Time

PM_OtoT		PM_O_to_T	PLC to Watlow PM
PM_OtoT.Ctrl_Mode	Decimal	DINT	PLC to Watlow PM PM Control Mode
PM_OtoT.CLSP	Float	REAL	PLC to Watlow PM CLSP
PM_OtoT.OLSP	Float	REAL	PLC to Watlow PM OLSP
PM_OtoT.ALm1H_SP	Float	REAL	PLC to Watlow PM Alm SP1 High
PM_OtoT.ALm1L_SP	Float	REAL	PLC to Watlow PM Alm SP1 Low
PM_OtoT.ALm2H_SP	Float	REAL	PLC to Watlow PM Alm SP2 High
PM_OtoT.ALm2L_SP	Float	REAL	PLC to Watlow PM Alm SP2 Low
PM_OtoT.ALm3H_SP	Float	REAL	PLC to Watlow PM Alm SP3 High
PM_OtoT.ALm3L_SP	Float	REAL	PLC to Watlow PM Alm SP3 Low
PM_OtoT.ALm4H_SP	Float	REAL	PLC to Watlow PM Alm SP4 High
PM_OtoT.ALm4L_SP	Float	REAL	PLC to Watlow PM Alm SP4 Low
PM_OtoT.Pro_Act_Req	Decimal	DINT	PLC to Watlow PM Profile_Action_Request
PM_OtoT.Pro_Strt	Decimal	DINT	PLC to Watlow PM Profile Start
PM_OtoT.H_PB	Float	REAL	PLC to Watlow PM Heat Proportional Band
PM_OtoT.C_PB	Float	REAL	PLC to Watlow PM Cool Proportional Band
PM_OtoT.Integral	Float	REAL	PLC to Watlow PM Integral
PM_OtoT.Derivative	Float	REAL	PLC to Watlow PM Derivative
PM_OtoT.On_Off_HHys	Float	REAL	PLC to Watlow PM Heat On-Off Hysteresis
PM_OtoT.On_Off_CHys	Float	REAL	PLC to Watlow PM Cool On-Off Hysteresis
PM_OtoT.DB	Float	REAL	PLC to Watlow PM PID DeadBand

You can now use simple logic to create instructions to move implicitly the default assembly structures to and from the PM. Recall that the name given to the I/O module is also used as the I/O tags. Note in the first copy instruction (input from PM to PLC) that the name given to the module appears as the source (Watlow_PM). Likewise, in the second copy instruction (output from PLC to PM) the destination tag reflects the module name. The two copy instructions below represent all that's needed to send and receive data from the PM. The copy instructions will copy source tags to destination tags byte for byte, so no further data conversion is needed.

Figure 3.7



Ladder Logic Example

In the likely event that the user wants to change the default assembly structures, this can be done using an explicit message. First, it is necessary to define the assembly setup. Note in Tables 3.8 and 3.9 that both assemblies (O to T and T to O) are accessed via class 119, where the instance identifies input and output with the attribute identifying the member within the instance.

Originator to Target (PLC to PM)

Table 3.8

Attribute Name	CIP Class ID	EIP Instance ID	EIP Attribute ID	Data-Type
OtoT Assembly Setup Instance 1	119	1	1	SINT
OtoT Assembly Setup Instance 2	119	1	2	SINT
OtoT Assembly Setup Instance 3	119	1	3	SINT
OtoT Assembly Setup Instance 4	119	1	4	SINT
OtoT Assembly Setup Instance 5	119	1	5	SINT
OtoT Assembly Setup Instance 6	119	1	6	SINT
OtoT Assembly Setup Instance 7	119	1	7	SINT
OtoT Assembly Setup Instance 8	119	1	8	SINT
OtoT Assembly Setup Instance 9	119	1	9	SINT
OtoT Assembly Setup Instance 10	119	1	10	SINT
OtoT Assembly Setup Instance 11	119	1	11	SINT
OtoT Assembly Setup Instance 12	119	1	12	SINT
OtoT Assembly Setup Instance 13	119	1	13	SINT
OtoT Assembly Setup Instance 14	119	1	14	SINT
OtoT Assembly Setup Instance 15	119	1	15	SINT
OtoT Assembly Setup Instance 16	119	1	16	SINT
OtoT Assembly Setup Instance 17	119	1	17	SINT
OtoT Assembly Setup Instance 18	119	1	18	SINT
OtoT Assembly Setup Instance 19	119	1	19	SINT
OtoT Assembly Setup Instance 20	119	1	20	SINT

Structure of 8-bit Data Type:

{0xCC, 0xII, 0xAA}

Target to Originator (PM to PLC)

Table 3.9

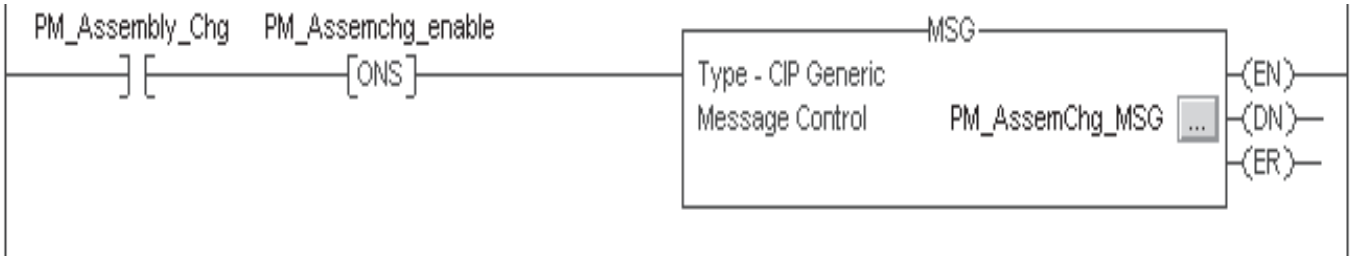
Attribute Name	CIP Class ID	EIP Instance ID	EIP Attribute ID	Data Type
TtoO Assembly Setup Instance 1	119	2	1	SINT
TtoO Assembly Setup Instance 2	119	2	2	SINT
TtoO Assembly Setup Instance 3	119	2	3	SINT
TtoO Assembly Setup Instance 4	119	2	4	SINT
TtoO Assembly Setup Instance 5	119	2	5	SINT
TtoO Assembly Setup Instance 6	119	2	6	SINT
TtoO Assembly Setup Instance 7	119	2	7	SINT
TtoO Assembly Setup Instance 8	119	2	8	SINT
TtoO Assembly Setup Instance 9	119	2	9	SINT
TtoO Assembly Setup Instance 10	119	2	10	SINT
TtoO Assembly Setup Instance 11	119	2	11	SINT
TtoO Assembly Setup Instance 12	119	2	12	SINT
TtoO Assembly Setup Instance 13	119	2	13	SINT
TtoO Assembly Setup Instance 14	119	2	14	SINT
TtoO Assembly Setup Instance 15	119	2	15	SINT
TtoO Assembly Setup Instance 16	119	2	16	SINT
TtoO Assembly Setup Instance 17	119	2	17	SINT
TtoO Assembly Setup Instance 18	119	2	18	SINT
TtoO Assembly Setup Instance19	119	2	19	SINT
TtoO Assembly Setup Instance 20	119	2	20	SINT

Structure of 8-bit Data Type:

{0xCC, 0xII, 0xAA}

For example, the screen captures below explain and illustrate how to change a given member for both the O-to-T and T-to-O assemblies. To change other members within either instance, simply change the instance (1 or 2) and attribute value (1 to 20) in the MSG instruction. For a better understanding of what happens when the instruction is enabled, take a closer look at the message instruction configuration and its associated tags.

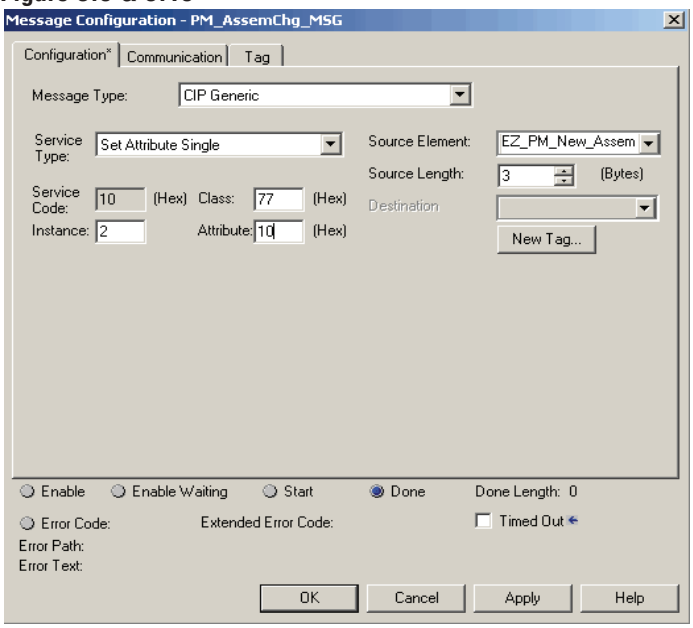
Figure 3.8



In configuring the MSG instruction it is important to use hexidecimal entries for the class, instance and attribute. In the example below (figure 3.9) the 16th location (attribute 10) of the T-to-O assembly structure (instance 1) will be changed. Looking at figure 3.6 above you will see that the 16th member of the T-to-O assembly defaults to "Profile Action Request." Once the configuration is complete, click on the communication tab and define the path to the PM.

When the MSG instruction above is enabled this member will be overwritten, and the new attribute (state of digital output 6) will be defined by the class, instance and attribute contained in the source element (see figure 3.11 below).

Figure 3.9 & 3.10



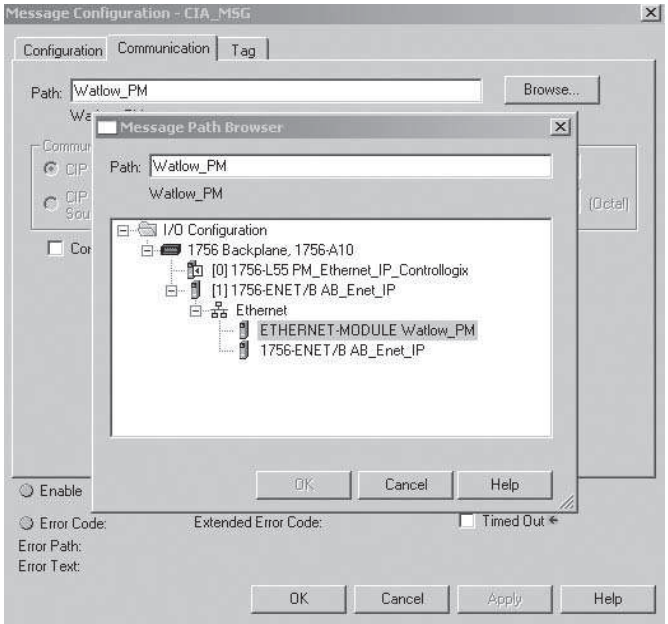


Figure 3.11

EZ_PM_New_Assembly	{...}	Hex	SINT[3]	
EZ_PM_New_Assembly[0]	16#6a	Hex	SINT	Class
EZ_PM_New_Assembly[1]	16#06	Hex	SINT	Instance
EZ_PM_New_Assembly[2]	16#07	Hex	SINT	Attribute

Each member in both the O-to-T and T-to-O assemblies can be changed in this manner. Also, any valid class, instance and attribute not found in the O-to-T and T-to-O assemblies can be read or written to explicitly using a rung of logic similar to the example in figure 3.3h

Saving Settings to Non-volatile Memory

When controller settings are entered from the controller front panel or a remote user interface (RUI) changes are always saved to non-volatile memory (EEPROM). If the controller loses power or is switched off its settings will be restored when power is reapplied.

The EEPROM will wear out after about 1,000,000 writes, which should not be a problem with changes made from the panel or RUI. However if the controller is receiving instructions from a PLC or a computer through a network connection, the EEPROM could, over time, wear out.

By default, settings made through the network are not saved to nonvolatile memory (59). However, every time a setting is changed through the front panel or RUI, all of the controller settings are saved to EEPROM, regardless of the setting of nonvolatile memory save. This parameter can only be changed via the network protocol (i.e., Modbus RTU, Modbus TCP, or EtherNet/IP) and will always be saved to EEPROM.

Non-volatile Save

Modbus Addr: 2494

EtherNet/IP & DeviceNet
Class: 150
Instance: 1
Attribute: 8

Enumeration: yes = 106, no = 59

Note:
Disabling EEPROM writes is available with PM firmware revision 2 and above.