



## EZ-ZONE® Configurator Software

# EZ-ZONE® Configurator Software Provides Easier Set-up for EZ-ZONE Products

EZ-ZONE® configurator software allows Watlow® EZ-ZONE products to be configured in one simple process. Its interface is flexible and easier to read than the basic remote user interface (RUI). It operates without requiring purchase of communications options as it uses the standard bus communications protocol that is included with all EZ-ZONE products. The EZ-ZONE Configurator software is available as a free download at [www.watlow.com](http://www.watlow.com).

## Features and Benefits

### Communicates with EZ-ZONE products via standard bus protocol

- Works regardless of which communications option is purchased or even when no communications option is purchased

### Detects EZ-ZONE devices and reads up configuration

- Allows easy access to any setting

### Presents pages and menus as they are in the controller's display, RUI and manuals

- Enables the user to easily locate what they are looking for

### Wizard-style editor with menu explorer

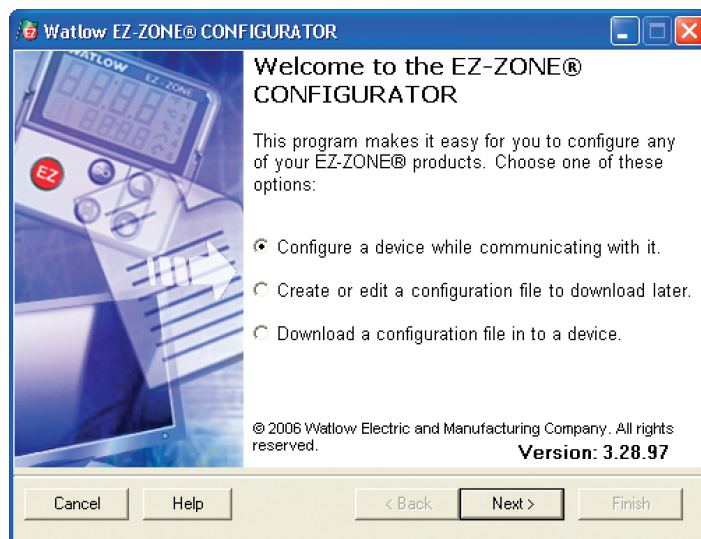
- Allows for easy examination of each menu
- Enables the user to skip directly to the parameters they want to work with

### On-screen parameter help

- Reduces configuration errors
- Helps the user take full advantage of available features

### Copy parameter settings

- Decreases the time to configure especially for multi-loop controllers



## Features and Benefits (con't)

### Save configuration files on the computer with all the information required to set up a controller

- Preserves the settings for archiving, recovery or to simplify setting up another EZ-ZONE product
- Enables files to be e-mailed or made available to users on a network or via the internet to aid them with set up

### View or modify configuration files saved during on-line editing sessions

- Allows users to get a jump on setting up EZ-ZONE products
- Aids in supporting remote users

### Downloads saved configuration files

- Simplifies configuration of EZ-ZONE products

### Flexible and smart compatibility checking

- Ensures configuration files are only loaded into devices that are similar enough to the original that the settings make sense



*Powered by Possibility*

To be automatically connected to the nearest North American Technical Sales Office:

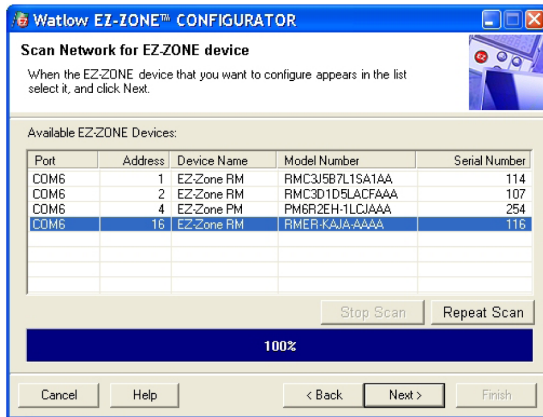
**1-800-WATLOW2 • [www.watlow.com](http://www.watlow.com)  
[inquiry@watlow.com](mailto:inquiry@watlow.com)**

### International Technical Sales Offices:

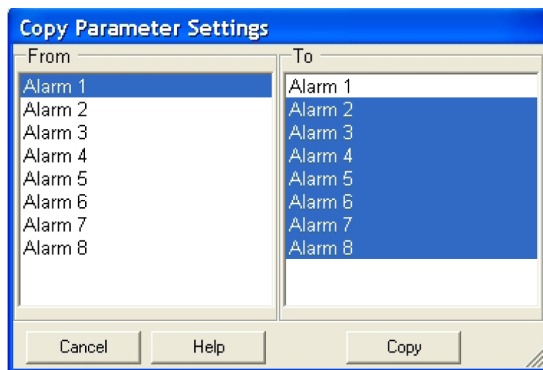
Austria	+43 6244 20129 0	India	+91 40 6661 2700
China	+86 21 3532 8532	Italy	+39 02 458 8841
France	+33 1 41 32 79 70	Japan	+81 3 3518 6630
Germany	+49 7253 9400 0	Korea	+82 2 2169 2600

Mexico	+52 442 256 2200
Singapore	+65 6773 9488
Spain	+34 91 675 1292
Taiwan	+886 7 288 5168
UK	+44 115 964 0777

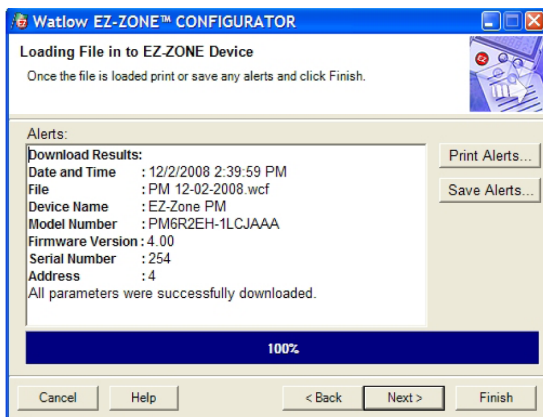
## Illustrated Features



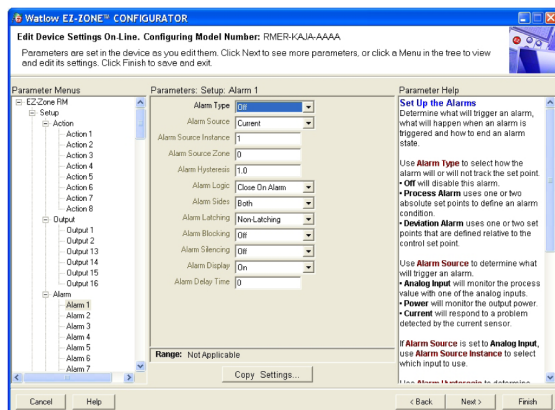
Detects EZ-ZONE devices connected to the computer's communications ports.



Copy feature speeds up configuration.



Confirms that parameter downloads were successful and reports exceptions.



Menu explorer allows users to skip directly to desired parameter or browse each setting.

## Compatibility

EZ-ZONE Configurator is compatible with all versions of EZ-ZONE products, but can be used to download configuration files only to products meeting the requirements listed below.

Product	Minimum Firmware Version
EZ-ZONE RUI	3.0
EZ-ZONE ST	4.0*
EZ-ZONE PM	7.0
EZ-ZONE PM Express	1.0
EZ-ZONE RM Control Module	1.0
EZ-ZONE RM Expansion Module	1.0
EZ-ZONE RM Access Module	1.0
EZ-ZONE RM High-Density Control Module	5.0
EZ-ZONE RM High-Density Limit Module	5.0
EZ-ZONE RM High-Density Scanner Module	5.0

\* Configuration files may be downloaded to EZ-ZONE ST controllers originally purchased with revision 4.0 or later only.

## System Requirements

### Minimum Requirements

- 485 Communications Port: USB port and USB-to-485 converter, or serial COM port (232) and 232-to-485 converter
- Microprocessor: Pentium® IV or equivalent
- Memory: 128 MB RAM (256 MB recommended)
- Disk Space: 140 MB (100 MB if Microsoft.NET Framework is already installed.)
- Video: 800 x 600 (1024 x 768 or higher recommended)

### Operating System Recommended

- Windows® 10
- Windows® 8.1
- Windows® 7

Watlow® and EZ-ZONE® are registered trademarks of Watlow Electric Manufacturing Company.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation.

Pentium® is a registered trademark of Intel Corporation.

# **EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic**

## **Example AB CompactLogix or ControlLogix PLC**

### **EtherNet/IP Fundamentals**

EtherNet/IP is built on the Common Industrial Protocol (CIP) at a foundational level. When communicating using CIP there are two ways to communicate to/from the Master and Slave devices, i.e., implicitly (real-time I/O messaging) and explicitly (information/configuration messaging). For your reference, the EZ-ZONE® PM is always the Slave whereas the PLC is the Master on the network.

#### **Implicit Communications - Defined**

Implicit messaging is real-time I/O messaging. It places different demands on the system due to the time critical nature of this form of communications the protocol must be able to support multi-casting while also ensuring that the time to execute the task is as fast as possible. To do this effectively, EtherNet/IP incorporates a protocol called User Datagram Protocol/Internet Protocol (UDP). Basically, this protocol contains the data alone without requiring a response from the Slave device. All data that is passed implicitly is defined in the configuration or start up process. Because this method of communications contains the predefined data alone, it is considered to be low overhead and is therefore able to deliver the time-critical requirements for control.

By using both forms of communication EtherNet/IP can prioritize time-critical I/O communications over non-critical messages while allowing for both to occur simultaneously. Watlow EtherNet/IP equipped devices supports both Explicit and Implicit communications.

#### **Explicit Communications - Defined**

Explicit messaging is executed on demand and can vary in size. Every message must be individually configured to execute a specific Message Type, e.g., CIP Generic and a specific Service Type, e.g., Get Attribute Single. Each device will interpret the message, act upon the task and then generate a response. This message type encapsulates information about the protocol itself as well as the instructions that need to be carried out in a TCP/IP packet. When a message is sent using TCP/IP it requires a response from the device. As stated above, this type of message is generally reserved for diagnostics and configuration.

### **Getting Started**

In this document, EZ-ZONE PM, PM, controller, and slave all mean the same thing. This document will not cover basic configuration of the EZ-ZONE PM. That information is covered in the User's Guide which can be found on the Watlow website - <http://www.watlow.com/literature/manuals.cfm>

#### **Understanding the Application Requirements**

- Will there be a need to infrequently read or write parameters between the Master and Slave? Explicit communications can be executed with minimal effort to accomplish this task.
- When using implicit communications determine what data (EZ-ZONE parameters) will be transferred implicitly (inputs and outputs) between the Master and Slave ensuring that the maximum number of 20 inputs and 20 outputs members is not exceeded.
- Compare your requirements of implicit data to the default assembly in the product. If what you need is already there, you may not want to change the assembly.
- In this documentation, the input assembly is referred to as the Originator to Target (O to T) and the output assembly is referred to as the Target to Originator (T to O). The Originator is the Master (usually a PLC) and the Target is the Slave. All EZ-ZONE PM assembly members (inputs and outputs) are 32-bits in length. In addition to the implicit members defined, the controller will return one 32-bit status word in the T to O.

# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

- The Requested Packet Interval (RPI) setting in the PLC determines how quickly the assembly information (I/O) is to be refreshed. When communicating implicitly, the Master (PLC) controls the cyclic timing (I/O updates) via the RPI setting. The RPI setting should be set at 250 mS or more.

### Configuration for Data Exchange

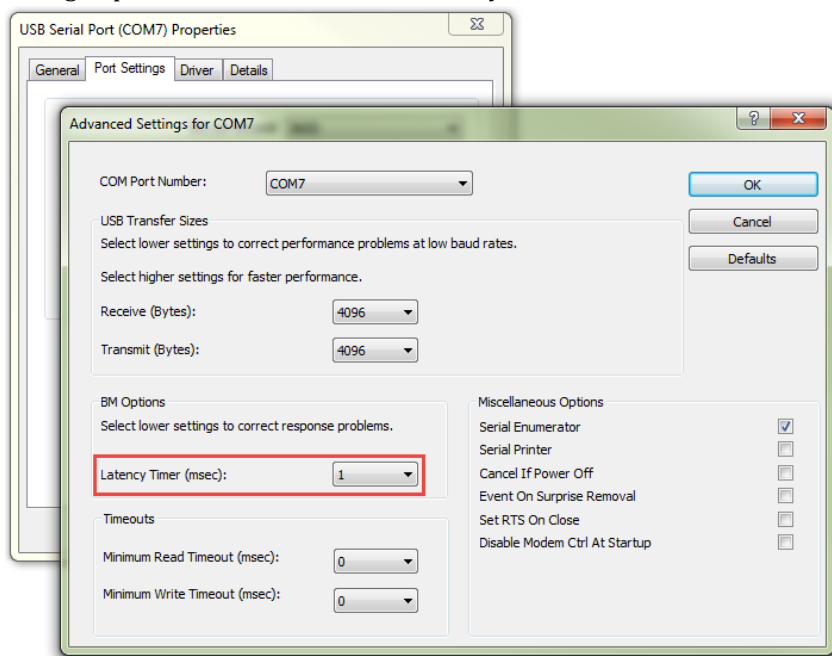
Basically, you need to program the PLC to send data to the controller and tell the PLC what data is expected from the controller. The PLC will use a generic I/O structure for this definition. The PLC will be programmed for the requested packet interval time, the size of the structure for inputs/outputs and the IP address of the controller.

The EZ-ZONE controller Ethernet port will be programmed with an IP address and subnet mask. We suggest you use a fixed IP address. You will define the size of the input and output assembly. This is then programmed into the controller. The data can be represented in degrees Fahrenheit or Celsius independent of the LED display when using the controller for temperature.

If you need to change the default Implicit Assemblies, you are required to use an Explicit message instruction from the PLC. To establish explicit communications between Master and Slave devices, configuration steps need to be executed within the PLC as well as within the EZ-ZONE PM using either the keypad or EZ-ZONE Configurator software connected to the EIA-485 port of the controller. After the configuration requirements have been met, programming examples will follow.

First let's review the sequence of tasks to be accomplished in the EZ-ZONE PM controller. The keypad method will not be detailed in this application note.

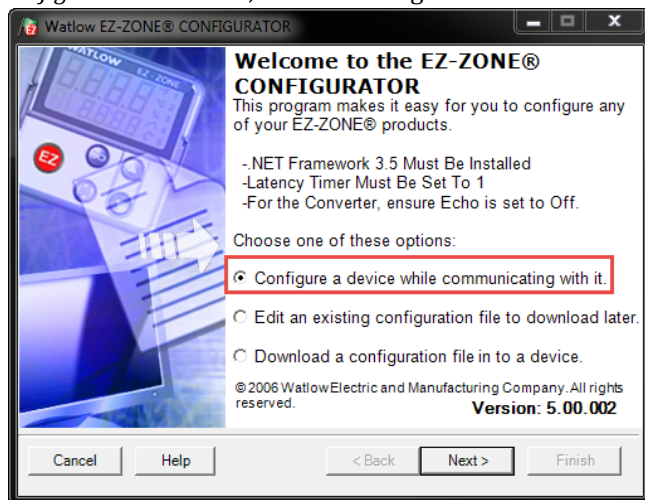
- 1) *Connect the PC* to the EZ-ZONE PM on terminals CD, CE and CF of slot C using an EIA-485 serial port. Typically the converter is a USB to EIA-485 device such as B&B Electronics 485USBTB-2W. Install Windows driver included with converter.
- 2) *Check latency timer* in serial driver of PC for 1mS setting. Not all drivers have this setting but if available, change to 1mS. Located in Device Manager, Ports, Properties of specific com port used. Then Port Settings, Advanced... button. This setting improves communications reliability.



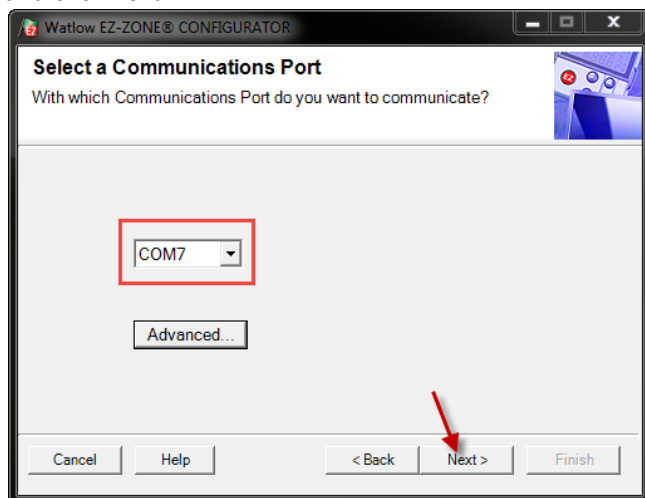
# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

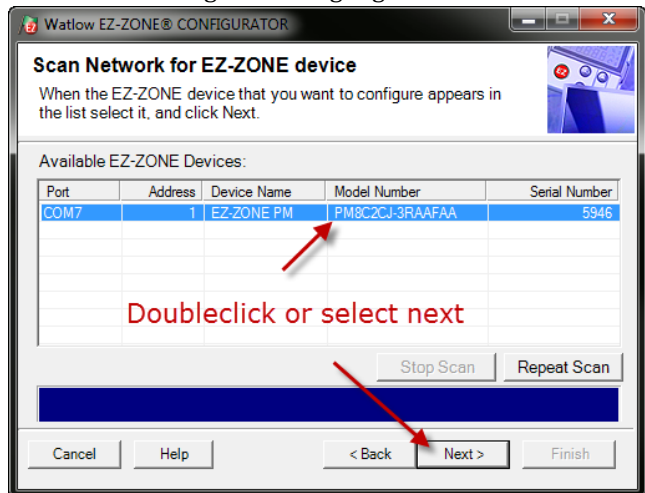
- 3) Install EZ-ZONE Configurator software on to PC. The program is located on the Watlow website under Download Center, Software and Demos category.
- 4) Execute EZ-ZONE Configurator software, choose 'Configure a device...'



- 5) Locate serial port and click next.



- 6) Doubleclick on PM device to be configured or highlight and select next.





# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

- 7) Enter Setup Page, Communications 2 Menu.
- 8) Identify the controller's IP address to be utilized.
- 9) Choose Fixed IP Address compatible with PLC network.
- 10) Enter IP Fixed Address Part 1 to 4, Subnet Part 1 to 4 and Gateway Part 1 to 4. Typically the Fixed IP Gateway Part 1 – 4 is set to 0.
- 11) Set Modbus TCP Enable to 'No' and EtherNet/IP Enable to 'Yes'.
- 12) Set Display units for communications. This is independent of the units on the LED display.
- 13) Choose if PLC writes are saved to EEPROM. Excessive writes will eventually wear out the EEPROM. Whenever the PLC write value changes the PM parameter, the value is committed to EEPROM every 3-second if Non-Volatile Save is set to Yes.

**Watlow EZ-ZONE® CONFIGURATOR**

**Edit Device Settings On-Line - Model Number:** PM8C2CJ-3RAAFAA

Click a Menu in the tree to view and edit its settings. Click Finish to save and exit.

**Parameter Menus**

- [-] EZ-ZONE PM
  - [-] Setup
    - [-] Analog Input
    - [-] Linearization
    - [-] Process Value
    - [-] Digital I/O
    - [-] Control Loop
    - [-] Output
    - [-] Alarm
    - [-] Special Output Function
    - [-] Function Key
    - [-] Global
    - [-] Communications
      - [-] Communications 1
      - [-] Communications 2
  - [-] Operations
    - [-] Analog Input
    - [-] Linearization
    - [-] Process Value
    - [-] Digital I/O
    - [-] Monitor
    - [-] Control Loop
    - [-] Alarm
    - [-] Special Output Function
  - [-] Factory
    - [-] Custom Setup
    - [-] Lock
    - [-] Diagnostics
      - [-] Diagnostics 1
    - [-] Calibration

**Parameters: Setup: Communications 2**

Modbus Word Order: Word Low High

**IP Address Mode:** Fixed IP Address

IP Fixed Address Part 1	10
IP Fixed Address Part 2	3
IP Fixed Address Part 3	38
IP Fixed Address Part 4	69
IP Fixed Address Part 5	0
IP Fixed Address Part 6	0

IP Fixed Subnet Part 1	255
IP Fixed Subnet Part 2	255
IP Fixed Subnet Part 3	224
IP Fixed Subnet Part 4	0
IP Fixed Subnet Part 5	0
IP Fixed Subnet Part 6	0

Fixed IP Gateway Part 1	0
Fixed IP Gateway Part 2	0
Fixed IP Gateway Part 3	0
Fixed IP Gateway Part 4	0
Fixed IP Gateway Part 5	0
Fixed IP Gateway Part 6	0

**Modbus TCP Enable:** No

**EtherNet/IP Enable:** Yes

**CIP Implicit Assembly Output Member Quantity:** 20

**CIP Implicit Assembly Input Member Quantity:** 20

**Display Units:** F

**Data Map:** 1

**Non-Volatile Save:** Yes

# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

- 14) *Identify the parameters in the controller to be written from the PLC.* The PLC references these as outputs. The controller references these as CIP Implicit Assembly Input Member Quantity (O to T). Count the number identified. See the PM users' guide for any parameter having a CIP register for choices. Also check the default implicit assembly structure listed in the Appendix of the PM users' guide. (partially shown here)

### CIP Implicit Assembly Structures

#### CIP Implicit O to T (Originator to Target) Assembly Structure

CIP Implicit Assembly Originator (Master) to Target (PM)					
Assembly Members	PM Assembly Class, Instance, Attribute	PM Data Type	Parameter	Parameter Class, Instance, Attribute	PLC Data Type
1	0x77, 0x01, 0x01	DINT	Loop Control Mode	0x97, 0x01, 0x01	DINT
2	0x77, 0x01, 0x02	DINT	Closed Loop Set Point	0x6B, 0x01, 0x01	REAL
3	0x77, 0x01, 0x03	DINT	Open Loop Set Point	0x6B, 0x01, 0x02	REAL
4	0x77, 0x01, 0x04	DINT	Alarm 1 - Alarm High Set Point	0x6D, 0x01, 0x01	REAL
5	0x77, 0x01, 0x05	DINT	Alarm 1 - Alarm Low Set Point	0x6D, 0x01, 0x02	REAL

- 15) *Identify the parameters in the controller to be read by the PLC.* The PLC references these as inputs. The controller references these as CIP Implicit Assembly Output Member Quantity (T to O). Count the number identified.

#### CIP Implicit T to O (Target to Originator) Assembly Structure

CIP Implicit Assembly Target (PM) to Originator (Master)					
Assembly Members	PM Assembly Class, Instance, Attribute	PM Data Type	Parameter	Parameter Class, Instance, Attribute	PLC Data Type
1	Cannot be changed	Binary	Device Status	none	DINT
2	0x77, 0x02, 0x01	DINT	Analog Input 1, Analog Input Value	0x68, 0x01, 0x01	REAL
3	0x77, 0x02, 0x02	DINT	Analog Input 1, Input Error	0x68, 0x01, 0x02	REAL
4	0x77, 0x02, 0x03	DINT	Analog Input 2, Analog Input Value	0x68, 0x02, 0x01	REAL
5	0x77, 0x02, 0x04	DINT	Analog Input 2, Input Error	0x68, 0x02, 0x02	REAL

For now, let's assume you will use the whole list as is and accept all 20 input and output assembly members as defined. If you wanted only the first 5 assembly members, then you can change the size to match. You can also change the arrangement of parameters in the list or redefine those assembly members. We will cover changing an assembly member later in the document. Whatever your choices are, they must be sequential and unique in this list.

# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

- 16) Enter the identified number from above step in the appropriate location. The PLC Inputs are written from the PM CIP Implicit Assembly Outputs (T to O). The PM CIP Implicit Assembly Inputs (O to T) are written from the PLC Outputs. In the previous step you should have identified the parameters to be implicitly transferred and have a count to enter here. Do not count the status word as one of the PM outputs in this number to be entered here. Since we assumed we will use the default assembly list, this screen capture shows 20 outputs and 20 inputs as viewed by the PM controller.

**Watlow EZ-ZONE® CONFIGURATOR**

**Edit Device Settings On-Line - Model Number:** PM8C2CJ-3RAAFAA

Click a Menu in the tree to view and edit its settings. Click Finish to save and exit.

**Parameter Menus**

- EZ-ZONE PM
  - Setup
    - Analog Input
    - Linearization
    - Process Value
    - Digital I/O
    - Control Loop
    - Output
    - Alarm
    - Special Output Function
    - Function Key
    - Global
    - Communications
      - Communications 1
      - Communications 2
  - Operations
    - Analog Input
    - Linearization
    - Process Value
    - Digital I/O
    - Monitor
    - Control Loop
    - Alarm
    - Special Output Function
  - Factory
    - Custom Setup
    - Lock
    - Diagnostics
      - Diagnostics 1
    - Calibration

**Parameters: Setup: Communications 2**

Modbus Word Order: Word Low High

IP Address Mode: Fixed IP Address

IP Fixed Address Part 1: 10

IP Fixed Address Part 2: 3

IP Fixed Address Part 3: 38

IP Fixed Address Part 4: 69

IP Fixed Address Part 5: 0

IP Fixed Address Part 6: 0

IP Fixed Subnet Part 1: 255

IP Fixed Subnet Part 2: 255

IP Fixed Subnet Part 3: 224

IP Fixed Subnet Part 4: 0

IP Fixed Subnet Part 5: 0

IP Fixed Subnet Part 6: 0

Fixed IP Gateway Part 1: 0

Fixed IP Gateway Part 2: 0

Fixed IP Gateway Part 3: 0

Fixed IP Gateway Part 4: 0

Fixed IP Gateway Part 5: 0

Fixed IP Gateway Part 6: 0

Modbus TCP Enable: No

EtherNet/IP Enable: Yes

**CIP Implicit Assembly Output Member Quantity: 20**

**CIP Implicit Assembly Input Member Quantity: 20**

Display Units: F

Data Map: 1

Non-Volatile Save: Yes

- 17) Remove power from the controller. You must cycle power on the EZ-ZONE PM controller for a new Fixed IP address to take effect.
- 18) That completes the communications configuration in the PM controller. You will need to configure other parameters for the application such as sensor type, how the control loop functions and which outputs perform



# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

what action. That is not related to setting up PLC communications so we will not cover that in this document.

You can validate the IP Actual Addressing Mode and Address used by the PM controller in the Factory Page, Diagnostics Menu. This is handy; especially when DHCP is used instead of Fixed IP addressing.

**Watlow EZ-ZONE® CONFIGURATOR**

**Edit Device Settings On-Line - Model Number:** PM8C2CJ-3RAAF AA

Click a Menu in the tree to view and edit its settings. Click Finish to save and exit.

**Parameter Menus**

- EZ-ZONE PM
  - Setup
    - Analog Input
    - Linearization
    - Process Value
    - Digital I/O
    - Control Loop
    - Output
    - Alarm
    - Special Output Function
    - Function Key
    - Global
  - Communications
    - Communications 1
    - Communications 2
  - Operations
    - Analog Input
    - Linearization
    - Process Value
    - Digital I/O
    - Monitor
    - Control Loop
    - Alarm
    - Special Output Function
  - Factory
    - Custom Setup
    - Lock
    - Diagnostics
      - Diagnostics 1
    - Calibration

**Parameters: Factory: Diagnostics 1**

<b>Part Number</b>	PM8C2CJ-3RAAF AA
<b>Software Revision</b>	13.00
<b>Software Build Number</b>	170
<b>Serial Number</b>	5946
<b>Date of Manufacture</b>	1142
<b>Actual IP Addressing Mode</b>	Fixed IP Address
<b>IP Actual Address Part 1</b>	10
<b>IP Actual Address Part 2</b>	3
<b>IP Actual Address Part 3</b>	38
<b>IP Actual Address Part 4</b>	69
IP Actual Address Part 5	0
IP Actual Address Part 6	0

**Range:** Not Applicable

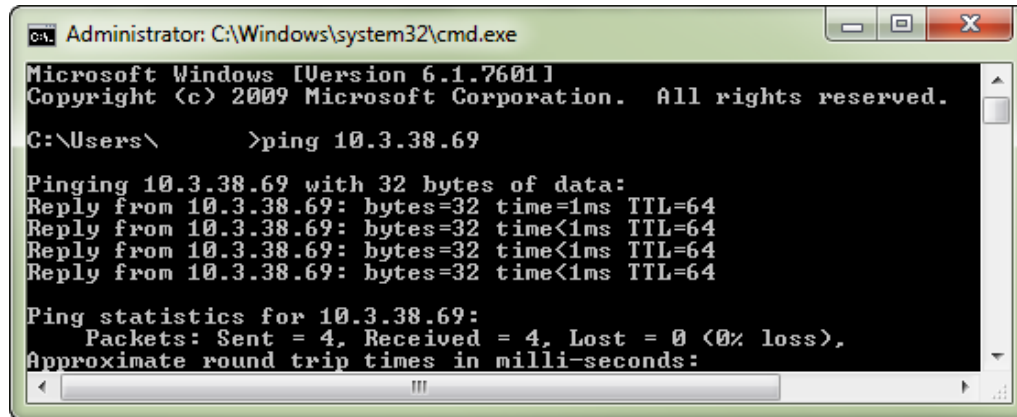
Copy Settings...

Cancel Help

## EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

### Example AB CompactLogix or ControlLogix PLC

You can use your PC to check the IP address of the PM controller by using a DOS Command prompt and ping the address for response. The PC must be on the same logical IP address as the controller. This screen capture shows four responses of a device at 10.3.38.69 so we know the IP address is correct (if using this in the controller) and that the wiring is working.



```
C:\Users\ >Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\ >ping 10.3.38.69

Pinging 10.3.38.69 with 32 bytes of data:
Reply from 10.3.38.69: bytes=32 time=1ms TTL=64
Reply from 10.3.38.69: bytes=32 time<1ms TTL=64
Reply from 10.3.38.69: bytes=32 time<1ms TTL=64
Reply from 10.3.38.69: bytes=32 time<1ms TTL=64

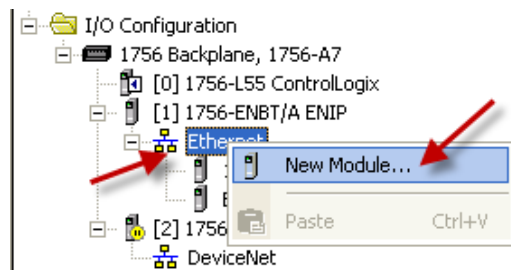
Ping statistics for 10.3.38.69:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
```

# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

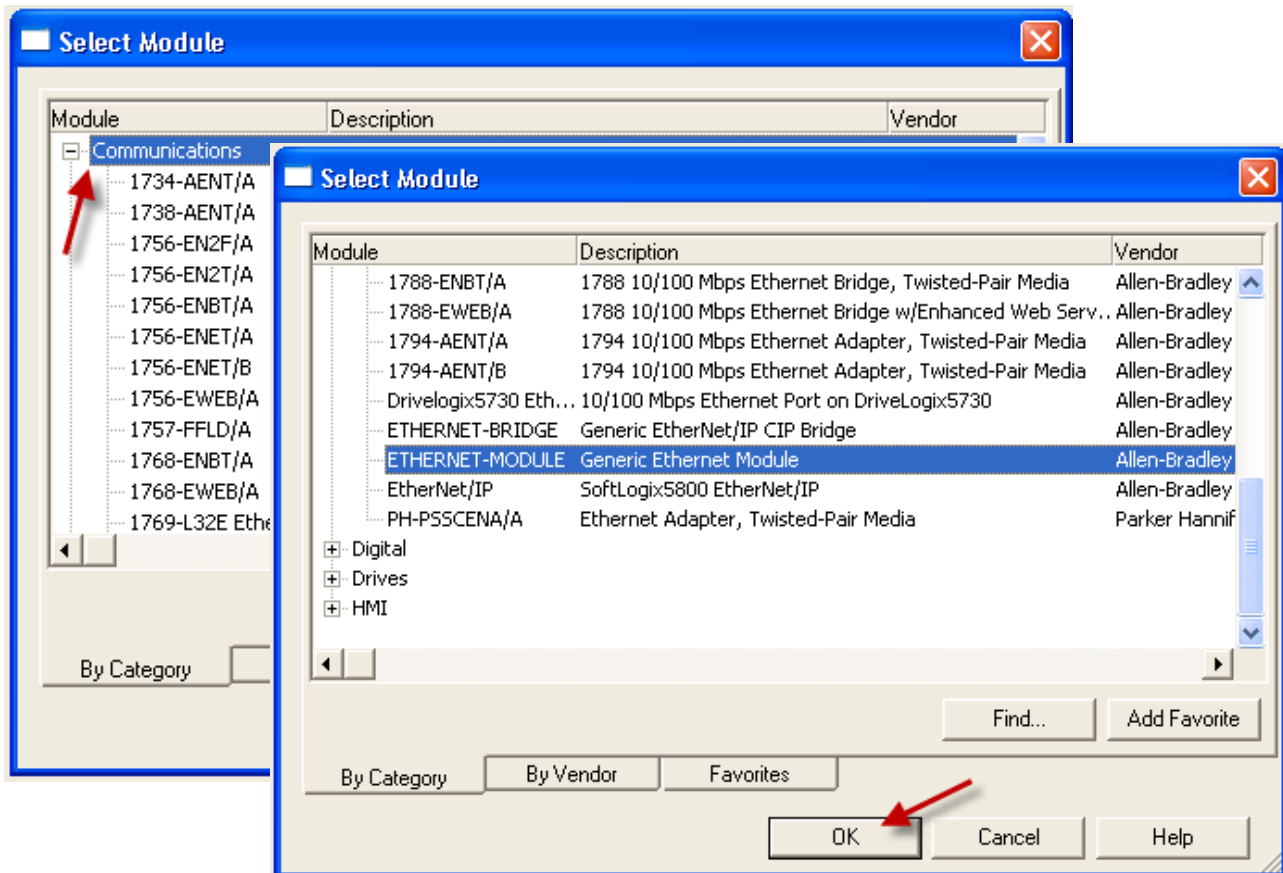
## Example AB CompactLogix or ControlLogix PLC

Now let's review the sequence of tasks to be accomplished in the PLC.

- 1) *Create a list of PLC outputs like the default show in the users' guide if you plan to change the O to T assembly members.* Previously you identified the controller parameters to be written into from the PLC. The PLC references these as outputs. The controller references these as CIP Implicit Assembly Input Member Quantity (O to T). See the PM Integrated Users' Guide for any parameter having a CIP register for other choices. Count the assembly member number required. This is the same thing we did earlier but repeated here in case you skipped to this section.
- 2) *Create a list of PLC inputs like the default show in the users' guide if you plan to change the T to O assembly members.* The PLC references these as inputs. The controller references these as CIP Implicit Assembly Output Member Quantity (T to O). Count the assembly member number required. See the PM users' guide for any parameter having a CIP register for other choices. This is the same thing we did earlier but repeated here in case you skipped to this section.
- 3) For now, let's assume you will use the whole list as is and accept all 20 inputs with the assembly members as defined. Had you only wanted the first 5 assembly members, then you can change the size to match in the PM controller and PLC. You can also change the arrangement of parameters in the list or redefine those assembly members. We will cover changing an assembly member later in the document.
- 4) *Open RSLogix5000 and add a Generic Module to PLC project. Right click on Ethernet and select New Module.*



- 5) *Expand Communications and scroll to ETHERNET-MODULE with description of Generic Ethernet Module. Select this option and click OK.*



# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

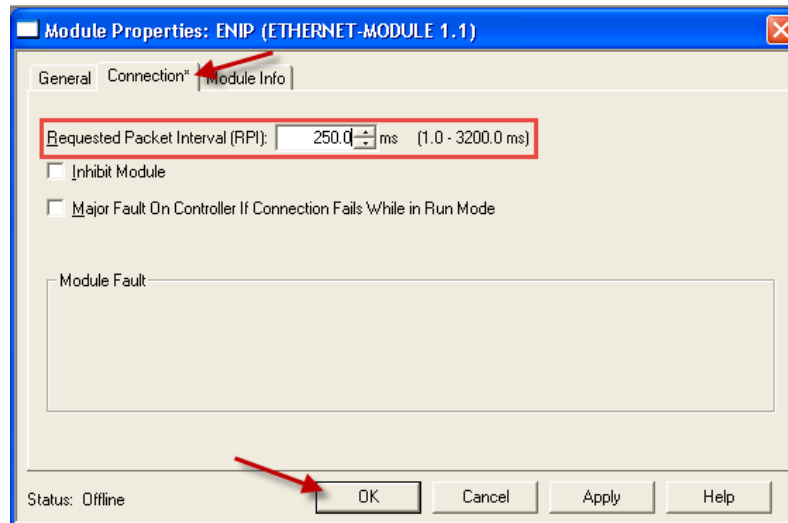
- 6) Enter a descriptive name, add description to identify product on network and enter IP Address of PM Controller. I used PMI\_1 for PM Integrated controller with ZONE 1 on display.

- 7) Define the I/O Implicit Assembly; enter Input Assembly Instance 101, Output Assembly Instance 100, and Configuration 128 with a size of 0. Set the Input Size to the number of parameters to be read from the EZ-ZONE PM controller by the PLC plus one 32-bit value representing the status of the PM controller. This screen capture shows Comm Format as Data - DINT (32-bit). If we use the default of 20 read members (T to O), add 1 for a status word so the Input Size is 21. If we use the default of 20 write members (O to T), the Output Size is 20. When set for Data - INT (16-bit) or Data - SINT (8-bit), change the size appropriately. The PLC Input Size must be set to a minimum of 1 (32-bit) and Output Size of 0. That would mean no implicit messaging will occur other than a status word. You only do this if you intend on using explicit messages exclusively. The smallest EZ-ZONE PM CIP Implicit Assembly Input/Output Member Quantity size is 0. **The most common mistake is entering incorrect connection parameters here.** Remember to choose the sizes based on need from 0 to 20 members in the PM controller. You can use the requested packet interval (RPI) setting to minimize traffic if concerned.

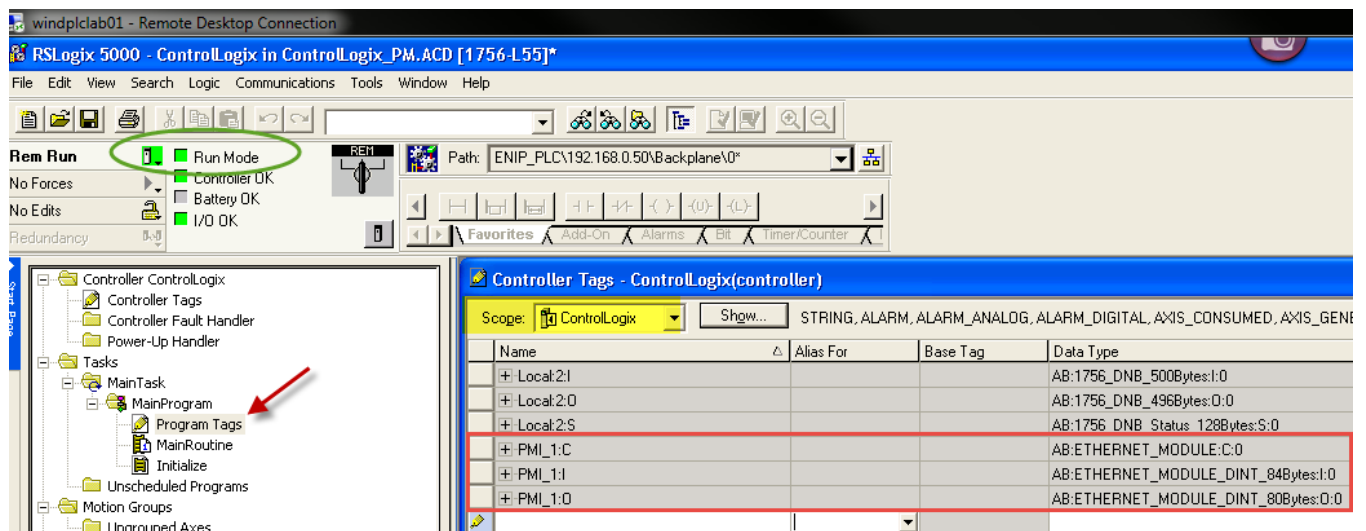
# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

- 8) Enter Requested Packet Interval (RPI) setting in Module properties, connection tab. Enter a value equal to or greater than 250.0 ms. Use the (RPI) setting to minimize network traffic if concerned. This setting determines the speed for data transaction of implicit messaging; larger values mean less frequent transactions.



- 9) Connect the PLC to the PM controller via EtherNet/IP using an industrial EtherNet/IP rated switch.
- 10) Load project to PLC to test communications.
- 11) Click on Program Tags, Scope: ControlLogix while in Run Mode. Note the newly created tags are there.





# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

- 12) View Monitor Tags tab. Expand the Input Data (PMI\_1:I in my project) to see actual raw data being displayed. This is the Target to Originator data. Change the Style for the first Assembly Member to Binary. Change the Style for the second Assembly Member to Hex. We used the default PM controller assembly in our example so the first Assembly Member is the status word shown here. It will always be the same. The second Assembly Member is the Analog Input 1, Analog Input Value shown as 16#429c\_0895. Since this value is a 32-bit floating point (real data type) shown in raw format, the number is large and constantly changing. If we convert this to a real number, it equals 78.01676 which happen to be the temperature of 78 being displayed on the PM controller in degrees F. Recall that the display units of the communications are independent of the display units for the LED. Be sure the communications and LED units are the same if comparisons are to be made.

Controller Tags - ControlLogix(controller)

Scope: ControlLogix Show... STRING, ALARM, ALARM\_ANALOG, ALARM\_DIGITAL, AXIS\_CONSUMED, AXIS\_GENE

Name	Value	Force Mask	Style	Data Type
Local:2:I	{...}	{...}		AB:175
Local:2:O	{...}	{...}		AB:175
Local:2:S	{...}	{...}		AB:175
PMI_1:C	{...}	{...}		AB:ETH
PMI_1:I	{...}	{...}		AB:ETH
PMI_1:I.Data	{...}	{...}	Decimal	DINT[2
PMI_1:I.Data[0]	2#0000_0000_0000_0001_0001_0000_0000_0000		Binary	DINT
PMI_1:I.Data[1]	16#429c_0895		Hex	DINT
PMI_1:I.Data[2]	61		Decimal	DINT
PMI_1:I.Data[3]	0		Decimal	DINT
PMI_1:I.Data[4]	246		Decimal	DINT
PMI_1:I.Data[5]	88		Decimal	DINT
PMI_1:I.Data[6]	88		Decimal	DINT
PMI_1:I.Data[7]	88		Decimal	DINT
PMI_1:I.Data[8]	88		Decimal	DINT
PMI_1:I.Data[9]	41		Decimal	DINT
PMI_1:I.Data[10]	41		Decimal	DINT
PMI_1:I.Data[11]	10		Decimal	DINT
PMI_1:I.Data[12]	0		Decimal	DINT
PMI_1:I.Data[13]	0		Decimal	DINT
PMI_1:I.Data[14]	0		Decimal	DINT
PMI_1:I.Data[15]	1		Decimal	DINT
PMI_1:I.Data[16]	61		Decimal	DINT
PMI_1:I.Data[17]	0		Decimal	DINT
PMI_1:I.Data[18]	0		Decimal	DINT
PMI_1:I.Data[19]	0		Decimal	DINT
PMI_1:I.Data[20]	0		Decimal	DINT
PMI_1:O	{...}	{...}		AB:ETH

Monitor Tags Edit Tags

# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

- 13) View Monitor Tags tab. Expand the Output Data (PMI\_1:0 in my project) to test the Originator to Target assembly. Recall that the default assembly has the first Assembly Member defined as Loop Control Mode. According to the PM Integrated Users' guide, the values for Control Mode are 10=Auto (Closed Loop), 54=Manual (Open Loop) or 62=Off (no control). The register is of data type DINT.

**Controller Tags - ControlLogix(controller)**

Scope: ControlLogix Show... STRING, ALARM, ALARM\_ANALOG, ALARM\_DIGITAL, AXIS\_CONSUMED, AXIS\_GENERI

Name	Value	Force Mask	Style	Data Type
Local:2:I	{...}	{...}		AB:1756
Local:2:O	{...}	{...}		AB:1756
Local:2:S	{...}	{...}		AB:1756
PMI_1:C	{...}	{...}		AB:ETHE
PMI_1:I	{...}	{...}		AB:ETHE
PMI_1:I.Data	{...}	{...}	Decimal	DINT[21]
PMI_1:O	{...}	{...}		AB:ETHE
PMI_1:O.Data	{...}	{...}	Decimal	DINT[20]
PMI_1:O.Data[0]	0		Decimal	DINT
PMI_1:O.Data[1]	0		Decimal	DINT
PMI_1:O.Data[2]	0		Decimal	DINT

**CIP Implicit 0 to T (Originator to Target) Assembly Structure**

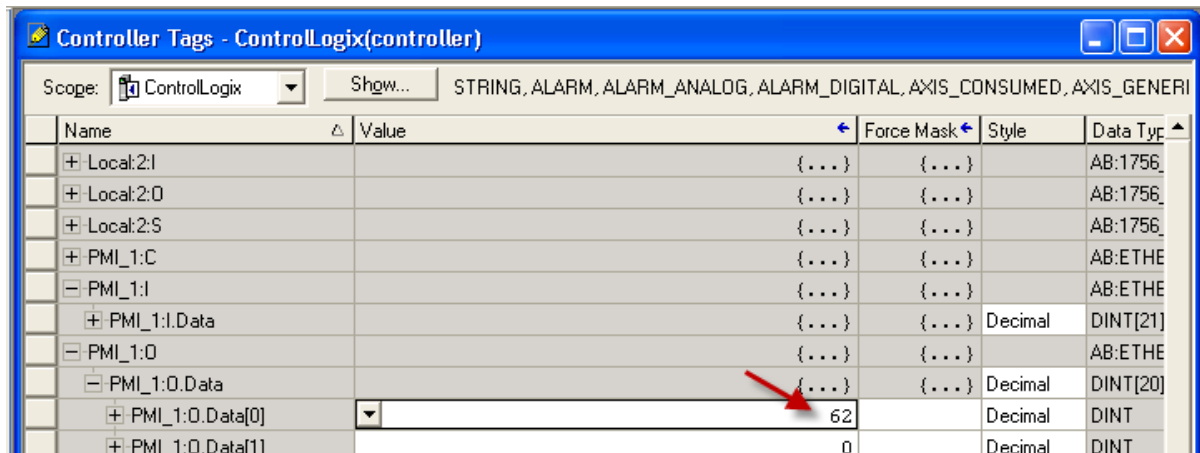
CIP Implicit Assembly Originator (Master) to Target (PM)					
Assembly Members	PM Assembly Class, Instance, Attribute	PM Data Type	Parameter	Parameter Class, Instance, Attribute	PLC Data Type
1	0x77, 0x01, 0x01	DINT	Loop Control Mode	0x97, 0x01, 0x01	DINT
PMI_1:O.Data[10]		0		Decimal	DINT
PMI_1:O.Data[11]		0		Decimal	DINT
PMI_1:O.Data[12]		0		Decimal	DINT
PMI_1:O.Data[13]		0		Decimal	DINT
PMI_1:O.Data[14]		0		Decimal	DINT
PMI_1:O.Data[15]		0		Decimal	DINT
PMI_1:O.Data[16]		0		Decimal	DINT
PMI_1:O.Data[17]		0		Decimal	DINT
PMI_1:O.Data[18]		0		Decimal	DINT
PMI_1:O.Data[19]		0		Decimal	DINT

Monitor Tags / Edit Tags

# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

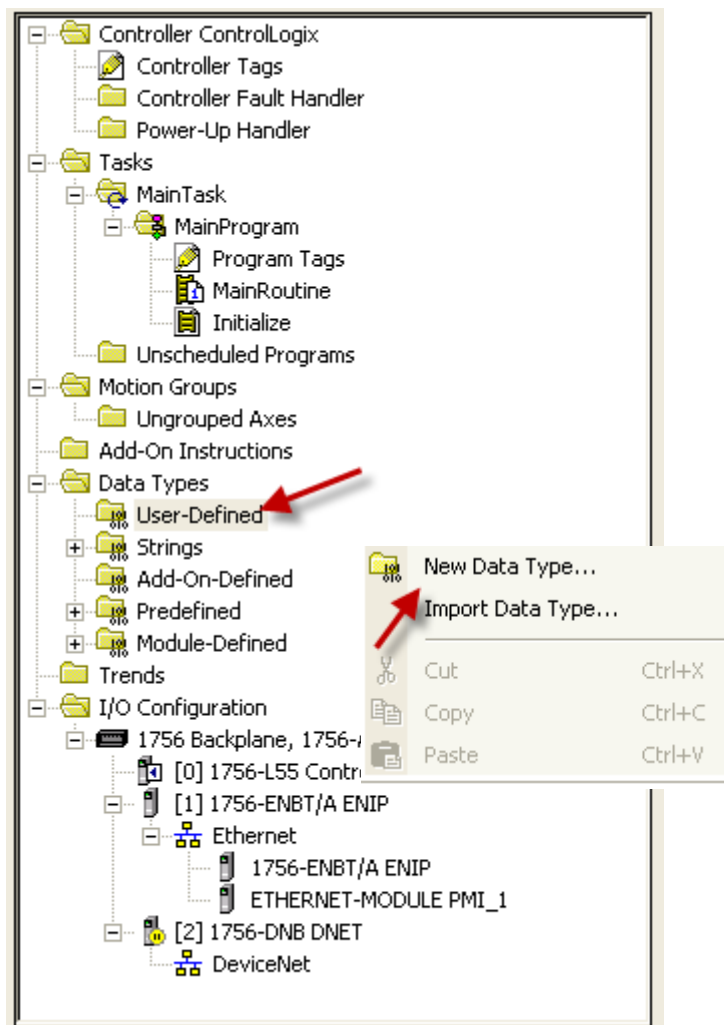
## Example AB CompactLogix or ControlLogix PLC

- 14) Double click in the Value box and change the value to 62 to place the PM control loop to Off (no control). Validate the PM controller displays the word **oFF** where the set point is on the green LED.



We now have a functioning system where implicit messaging is occurring. The next step is to create a structure to convert the raw data to identifiable tags and readable values.

- 15) Right click on User-Defined below Data Types. Select New Data Type...



# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

- 16) Define User Data Type to match output assembly of PM controller. Provide a name to help identify the data structure. Include a description for further understanding. For each Member, enter in a short name, data type for the parameter being entered and a description. Click Apply and the screen will update the Data Type Size. It must match the previous configuration for assembly size. It is shown in bytes instead of word length here (32-bits = 4 bytes).

In this screen capture I labeled the default assembly members for the T20. I included the Class, Instance and Attribute number for each element in the description to cross reference to the PM Integrated Users' Guide to ensure I know which items are being used; Class is in Hex, Instance in Decimal and Attribute in Hex format. If you change the default assembly, you will change the Name, Data Type and Description here to match. The Data Type defines the Style and this must match the data type listed in the PM Integrated User's Guide for the appropriate parameter. REAL and Float mean the same thing and DINT and unsigned integer are the same. The PM Integrated Users' Guide will identify parameter Data Type as float or unsigned integers.

Name: PMI\_to\_PLC

Description: Target to Originator (T20)

Members: Data Type Size: 84 byte(s)

Name	Data Type	Style	Description
Device_Status	DINT	Binary	Device Status
Ai_V1	REAL	Float	Analog Input Value 1 - 0x68, 1, 0x01
Ai_Er1	DINT	Decimal	Analog Input 1, Input Error - 0x68, 1, 0x02
Ai_V2	REAL	Float	Analog Input Value 2 - 0x68, 2, 0x01
Ai_Er2	DINT	Decimal	Analog Input 2, Input Error - 0x68, 2, 0x02
Alm_S1	DINT	Decimal	Alarm 1, Alarm State - 0x6D, 1, 0x09
Alm_S2	DINT	Decimal	Alarm 2, Alarm State - 0x6D, 2, 0x09
Alm_S3	DINT	Decimal	Alarm 3, Alarm State - 0x6D, 3, 0x09
Alm_S4	DINT	Decimal	Alarm 4, Alarm State - 0x6D, 4, 0x09
ES1	DINT	Decimal	Event Status 1 - 0x6E, 1, 0x05
ES2	DINT	Decimal	Event Status 2 - 0x6E, 2, 0x05
CMA1	DINT	Decimal	Control Mode Active 1 - 0x97, 1, 0x02
HPr1	REAL	Float	Heat Power 1 - 0x97, 1, 0x0D
CP1	REAL	Float	Cool Power 1 - 0x97, 1, 0x0E
Li_S	DINT	Decimal	Limit State - 0x70, 1, 0x06
PStr	DINT	Decimal	Profile Start - 0x7A, 1, 0x01
PACr	DINT	Decimal	Profile Action Request - 0x7A, 1, 0x01
Cu_Pr	DINT	Decimal	Current Profile - 0x7A, 1, 0x03
Cu_Step	DINT	Decimal	Current Step - 0x7A, 1, 0x04
Active_SP1	REAL	Float	Active Set Point 1 - 0x7A, 1, 0x05
Ti_Remain	REAL	Float	Step Time Remaining - 0x7A, 1, 0x09

Move Up Move Down OK Cancel Apply Help

Ai\_V2

# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

- 17) Define User Data Type to match input assembly of PM controller. Provide a name to help identify the data structure. Include a description for further understanding. For each Member, enter in a short name, data type for the parameter being entered and a description. Click Apply and the screen will update the Data Type Size. It must match the previous configuration for assembly size. It is shown in bytes instead of word length here (32-bits = 4 bytes).

In this screen capture I labeled the default assembly members for the O2T. I included the Class, Instance and Attribute number for each element in the description to cross reference to the PM Integrated Users' Guide to ensure I know which items are being used; Class is in Hex, Instance in Decimal and Attribute in Hex format. If you change the default assembly, you will change the Name, Data Type and Description here to match. The Data Type defines the Style and this must match the data type listed in the PM Integrated User's Guide for the appropriate parameter. REAL and Float mean the same thing and DINT and unsigned integer are the same. The PM Integrated Users' Guide will identify parameter Data Type as float or unsigned integers.

Name: PLC\_to\_PMI

Description: Originator to Target (O2T)

Members: Data Type Size: 80 byte(s)

Name	Data Type	Style	Description
CM1	DINT	Decimal	Loop Control Mode 1 - 0x97, 1, 0x01
CLSP1	REAL	Float	Closed Loop Set Point 1 - 0x6B, 1, 0x01
OLSP1	REAL	Float	Open Loop Set Point 1 - 0x6B, 1, 0x02
Alm_HSP1	REAL	Float	Alarm High Set Point 1 - 0x6D, 1, 0x01
Alm_LSP1	REAL	Float	Alarm Low Set Point 1 - 0x6D, 1, 0x02
Alm_HSP2	REAL	Float	Alarm High Set Point 2 - 0x6D, 2, 0x01
Alm_LSP2	REAL	Float	Alarm Low Set Point 2 - 0x6D, 2, 0x02
Alm_HSP3	REAL	Float	Alarm High Set Point 3 - 0x6D, 3, 0x01
Alm_LSP3	REAL	Float	Alarm Low Set Point 3 - 0x6D, 3, 0x02
Alm_HSP4	REAL	Float	Alarm High Set Point 4 - 0x6D, 4, 0x01
Alm_LSP4	REAL	Float	Alarm Low Set Point 4 - 0x6D, 4, 0x02
PACr	DINT	Decimal	Profile Action Request - 0x7A, 1, 0x0B
PStr	DINT	Decimal	Profile Start - 0x7A, 1, 0x01
HPb1	REAL	Float	Heat Proportional Band 1 - 0x97, 1, 0x06
CPb1	REAL	Float	Cool Proportional Band 1 - 0x97, 1, 0x07
Ti1	REAL	Float	Time Integral 1 - 0x97, 1, 0x08
Td1	REAL	Float	Time Derivative 1 - 0x97, 1, 0x09
HHys1	REAL	Float	Heat Hysteresis 1 - 0x97, 1, 0x0B
CHys1	REAL	Float	Cool Hysteresis 1 - 0x97, 1, 0x0C
DB1	REAL	Float	Deadband 1 - 0x97, 1, 0x0A

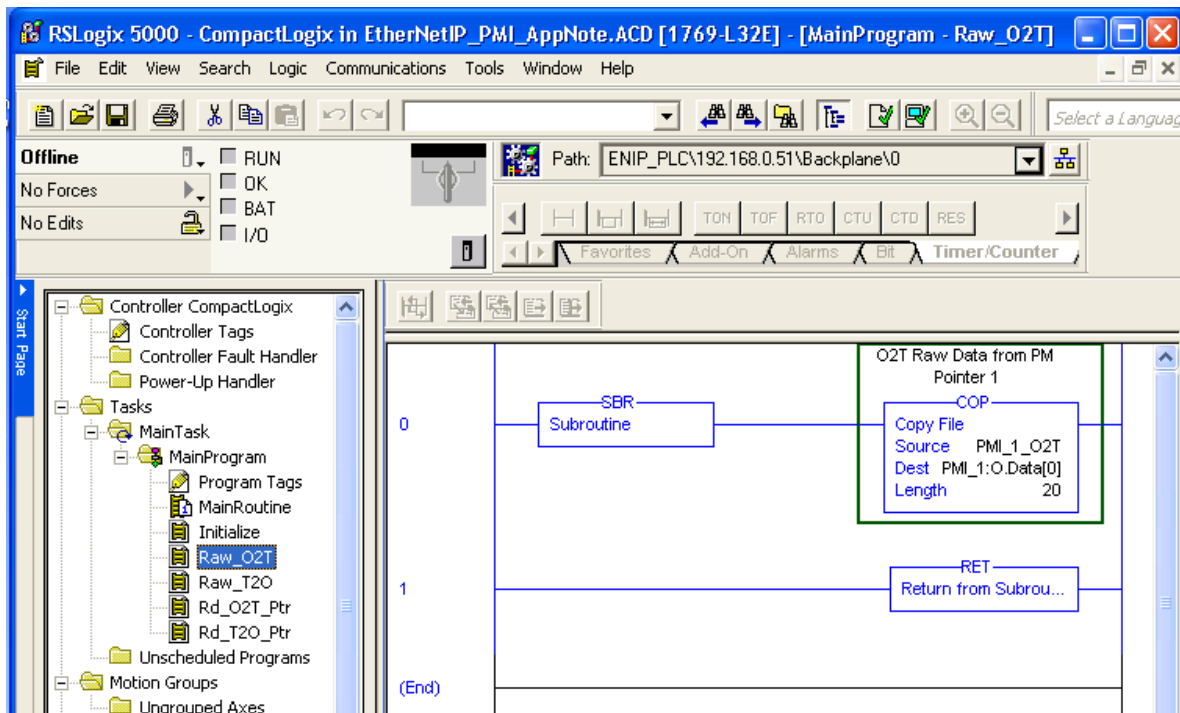
Move Up Move Down OK Cancel Apply Help



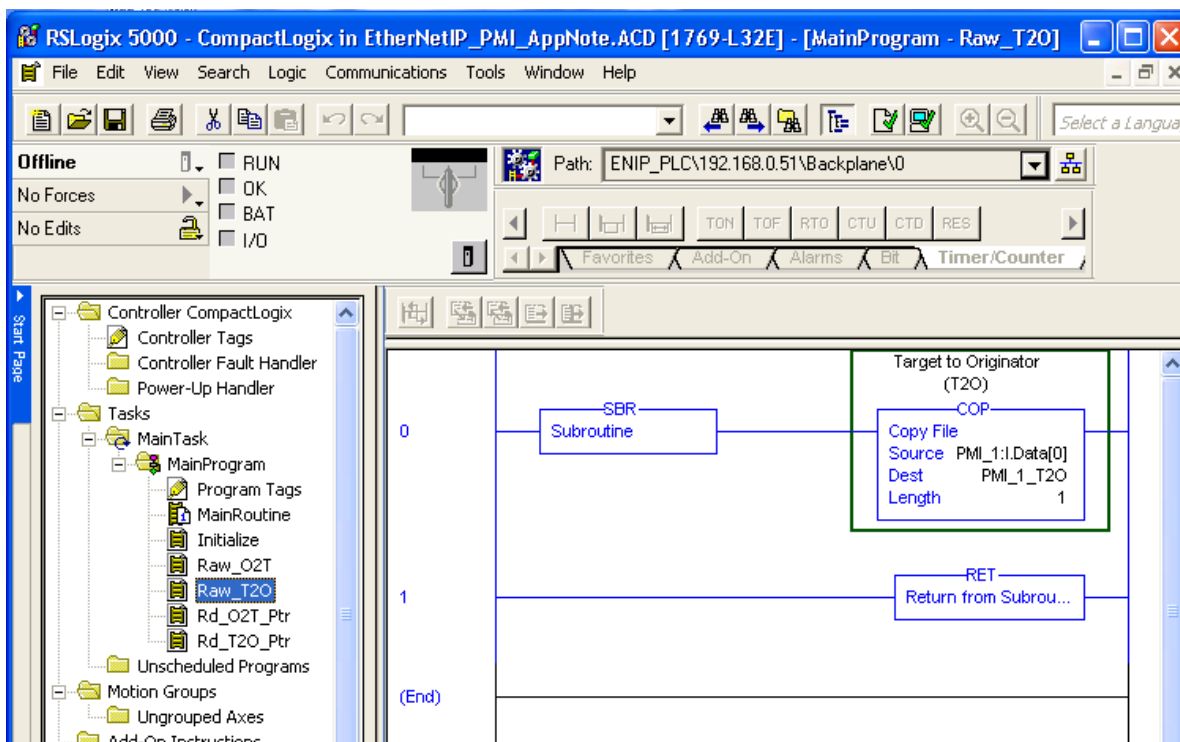
# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

- 18) Program ladder logic to copy data between user data type (O2T) and implicit assembly originator. Add a Subroutine with copy block. This instruction will copy from the user structure PMI\_1\_O2T into PMI\_1:O.Data(0). Be sure the Length (20 in my example) matches the number of members being copied as defined earlier.



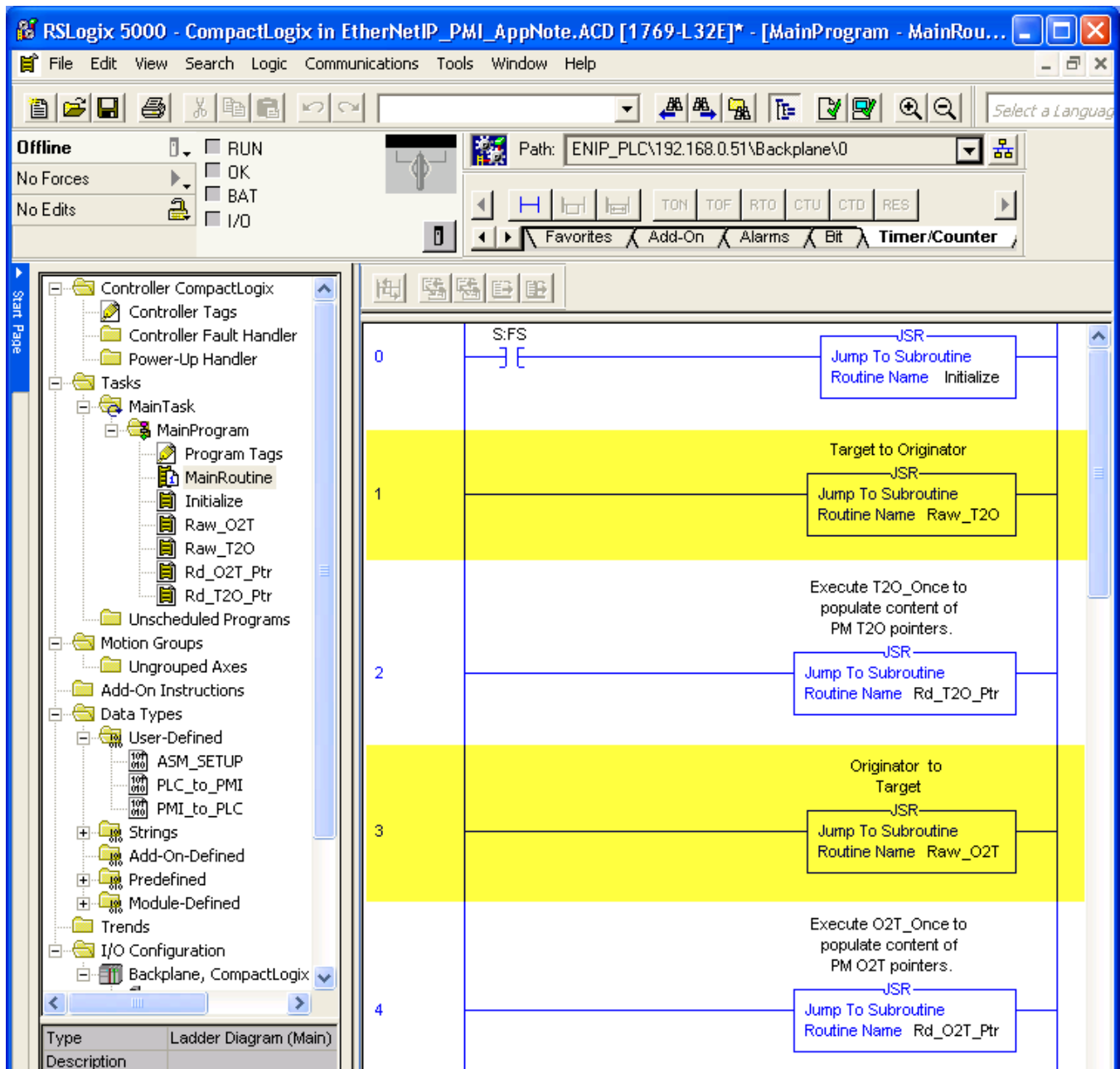
- 19) Program ladder logic to copy data between user data type target and implicit assembly originator (T2O). Add Subroutine with copy block. This instruction will copy into the user data type PMI\_1:I.Data(0) to PMI\_1\_T2O. Be sure the Length is set to 1.



# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

- 20) Insert ladder logic in MainRoutine to call subroutines. The subroutines will now copy between raw data format and the user defined data type. Data will appear correctly converted under Controller Tags, PMI\_1\_O2T and PMI\_1\_T20.



# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

- 21) Enter Run Mode on PLC and test PMI\_1\_O2T user defined data type tags. Here I changed the control mode between Manual=54 and Auto=10. Then I validated that the controller responded. Next I changed the CLSP1 (Closed Loop Set Point 1) to 85.0 and validated the response.

The screenshot shows the RSLogix 5000 software interface. The top menu bar includes File, Edit, View, Search, Logic, Communications, Tools, Window, and Help. The toolbar contains various icons for file operations and logic editing. The main window is divided into three panes:

- Left Pane (Project Explorer):** Shows the project structure. The 'Controller CompactLogix' folder is expanded, showing 'Controller Tags', 'Controller Fault Handler', and 'Power-Up Handler'. The 'Tasks' folder is also expanded, showing 'MainTask' and 'MainProgram'. The 'MainProgram' folder is expanded, showing 'Program Tags', 'MainRoutine', 'Initialize', 'Raw\_O2T', 'Raw\_T2O', 'Rd\_O2T\_Ptr', and 'Rd\_T2O\_Ptr'. The 'Data Types' folder is expanded, showing 'User-Defined', 'ASM\_SETUP', 'PLC\_to\_PMI', and 'PMI\_to\_PLC'. The 'Strings' folder is expanded, showing 'Add-On-Defined', 'Predefined', and 'Module-Defined'. The 'Trends' folder is also expanded.
- Top Right Pane (Controller Information):** Shows the controller path: 'ENIP\_PLC\192.168.0.51\Backplane\0'. It also displays the 'Run Mode' status, which is currently 'Run Mode'.
- Bottom Right Pane (Monitor Tags):** Displays a table of tags for the 'CompactLogix' controller. The table has columns for Name, Value, Force, Style, Data Type, and Description. The tags are listed in the table below.

Name	Value	Force	Style	Data Type	Description
PMI_1:O.Data	{...}	{...}	Decimal	DINT[20]	O2T Raw Dat
PMI_1_O2T	{...}	{...}		PLC_to_PMI	Originator to T
PMI_1_O2T.CM1	10		Decimal	DINT	Originator to T
PMI_1_O2T.CLSP1	85.0		Float	REAL	Originator to T
PMI_1_O2T.OLSP1	0.0		Float	REAL	Originator to T
PMI_1_O2T.Alm_HSP1	300.0		Float	REAL	Originator to T
PMI_1_O2T.Alm_LSP1	0.0		Float	REAL	Originator to T
PMI_1_O2T.Alm_HSP2	10.0		Float	REAL	Originator to T
PMI_1_O2T.Alm_LSP2	-10.0		Float	REAL	Originator to T
PMI_1_O2T.Alm_HSP3	0.0		Float	REAL	Originator to T
PMI_1_O2T.Alm_LSP3	0.0		Float	REAL	Originator to T
PMI_1_O2T.Alm_HSP4	0.0		Float	REAL	Originator to T
PMI_1_O2T.Alm_LSP4	0.0		Float	REAL	Originator to T
PMI_1_O2T.PACr	0		Decimal	DINT	Originator to T
PMI_1_O2T.PStr	0		Decimal	DINT	Originator to T
PMI_1_O2T.HPb1	25.0		Float	REAL	Originator to T
PMI_1_O2T.CPb1	180.0		Float	REAL	Originator to T
PMI_1_O2T.Ti1	0.0		Float	REAL	Originator to T
PMI_1_O2T.Td1	0.0		Float	REAL	Originator to T
PMI_1_O2T.HHys1	3.0		Float	REAL	Originator to T
PMI_1_O2T.CHys1	3.0		Float	REAL	Originator to T
PMI_1_O2T.DB1	0.0		Float	REAL	Originator to T

The 'Monitor Tags' window is active, showing the 'Monitor Tags' tab. The status bar at the bottom indicates 'Ready'.

# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

22) Test PMI\_1\_T20 user defined data type tags. Here we see the fixed Device Status word. The Ai\_V1 (Analog Input Value 1) is shown and is constantly changing. Note the raw value has been converted to 80.46 degrees; a real useable number. Also note Ai\_Er1 (Analog Input 1, Input Error) is shown as 61=None. The HPr1 (Heat Power 1) is shown as 18.25% power.

RSLogix 5000 - CompactLogix in EtherNetIP\_PMI\_AppNote.ACD [1769-L32E]\* - [Controller Tags - CompactLogix(controller)]

File Edit View Search Logic Communications Tools Window Help

Path: ENIP\_PLC\192.168.0.51\Backplane\0

Scope: CompactLogix Show... Show All

Name	Value	Force	Style	Data Type	Description
PMI_1_T20	{...}	{...}		PMI_to_PLC	Target to Orig
PMI_1_T20.Device_Status	2#0000_0000_0000_0001_0001_0000_0000_0000		Binary	DINT	Target to Orig
PMI_1_T20.Ai_V1	80.46485		Float	REAL	Target to Orig
PMI_1_T20.Ai_Er1	61		Decimal	DINT	Target to Orig
PMI_1_T20.Ai_V2	0.0		Float	REAL	Target to Orig
PMI_1_T20.Ai_Er2	246		Decimal	DINT	Target to Orig
PMI_1_T20.Alm_S1	88		Decimal	DINT	Target to Orig
PMI_1_T20.Alm_S2	88		Decimal	DINT	Target to Orig
PMI_1_T20.Alm_S3	88		Decimal	DINT	Target to Orig
PMI_1_T20.Alm_S4	88		Decimal	DINT	Target to Orig
PMI_1_T20.ES1	41		Decimal	DINT	Target to Orig
PMI_1_T20.ES2	41		Decimal	DINT	Target to Orig
PMI_1_T20.CMA1	10		Decimal	DINT	Target to Orig
PMI_1_T20.HPr1	18.256775		Float	REAL	Target to Orig
PMI_1_T20.CP1	0.0		Float	REAL	Target to Orig
PMI_1_T20.Li_S	0		Decimal	DINT	Target to Orig
PMI_1_T20.PStr	1		Decimal	DINT	Target to Orig
PMI_1_T20.PACr	61		Decimal	DINT	Target to Orig
PMI_1_T20.Cu_Pr	0		Decimal	DINT	Target to Orig
PMI_1_T20.Cu_Step	0		Decimal	DINT	Target to Orig
PMI_1_T20.Active_SP1	0.0		Float	REAL	Target to Orig
PMI_1_T20.Ti_Remain	0.0		Float	REAL	Target to Orig

Ready

That completes configuration for the implicit assembly using default values. Note that the previous steps used all 20 input and output members with default values. You can choose to make the input/output list smaller independently of each other. Assume you determine that the PM T20 has what is required in the first three members. Then you only set the T20 assembly size to 4 in RSLogix5000 and to 3 in the PM configuration. Only define the first 4 members in the user defined data type tags. Recall that the first member is the device status word. The rest of the document is devoted to changing or rearranging the members in this list.

# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

### Changing a member in the implicit table using an explicit message.

If you require changing the default implicit assembly to an assembly member not shown or to change the order of the assembly members; you simply create a ladder rung that allows a defined message to be sent. Referring to the implicit assembly located in the EZ-ZONE PM manual, note there are registers to be written that contain a pointer of the data to be located in that row.

## CIP Implicit Assembly Structures

### CIP Implicit O to T (Originator to Target) Assembly Structure

CIP Implicit Assembly Originator (Master) to Target (PM)					
Assembly Members	PM Assembly Class, Instance, Attribute	PM Data Type	Parameter	Parameter Class, Instance, Attribute	PLC Data Type
1	0x77, 0x01, 0x01	DINT	Loop Control Mode	0x97, 0x01, 0x01	DINT
2	0x77, 0x01, 0x02	DINT	Closed Loop Set Point	0x6B, 0x01, 0x01	REAL
3	0x77, 0x01, 0x03	DINT	Open Loop Set Point	0x6B, 0x01, 0x02	REAL
4	0x77, 0x01, 0x04	DINT	Alarm 1 - Alarm High Set Point	0x6D, 0x01, 0x01	REAL

Here we see Assembly Member 3 as a register of the pointer 0x77, 1, 0x03. By default, the pointer in this location is 0x6B, 1, 0x02.

I suggest you first use explicit messaging to read the PM O2T and T2O pointers into a User Defined Data Type structure. I created a User-Defined Data Types called ASM\_SETUP which has three elements (one to hold the class, one to hold the instance and the last one holds the attribute. Pick the Data Type as SINT and the Style as Hex.

The screenshot displays the EZ-ZONE software interface. On the left, the project tree shows the hierarchy: Controller CompactLogix > Tasks > MainTask > Program Tags > MainRoutine > Initialize > Raw\_O2T > Raw\_T2O > Rd\_O2T\_Ptr > Rd\_T2O\_Ptr > Unscheduled Programs > Motion Groups > Ungrouped Axes > Add-On Instructions > Data Types > User-Defined. The 'ASM\_SETUP' data type is highlighted. On the right, the 'ASM\_SETUP' structure is defined with the following members:

Name	Data Type	Style	Description
ClassID	SINT	Hex	
InstanceID	SINT	Hex	
AttributeID	SINT	Hex	

The description field for the structure is: "Structure to hold Class, Instance and Attribute for Implicit configuration".



# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

Then create Controller Tags to dimension of 20 for both O2T\_Members and 20 for T20\_Members. Data Type is ASM\_SETUP[20] which is the structure previously created.

The screenshot shows the 'Controller Tags' window for a CompactLogix controller. The 'Scope' is set to 'CompactLogix'. The 'Show' button is selected. The table lists the following tags:

Name	Alias For	Base Tag	Data Type	Style	Description
O2T_Members			ASM_SETUP[20]		Structure to hold Class, Instance and Attribute for Implicit config
T20_Members			ASM_SETUP[20]		Structure to hold Class, Instance and Attribute for Implicit config

It is not an efficient way to handle the next part but for simplicity, I created 20 of O2T\_MSG\_Ctrl tags to handle some ladder logic for sequencing through the message instructions. Data Type is MESSAGE.

The screenshot shows the 'Controller Tags' window for a CompactLogix controller. The 'Scope' is set to 'CompactLogix'. The 'Show' button is selected. The table lists the following tags:

Name	Alias For	Base Tag	Data Type	Style	Description
O2T_MSG_Ctrl1			MESSAGE		O2T Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl10			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl11			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl12			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl13			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl14			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl15			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl16			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl17			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl18			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl19			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl2			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl20			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl3			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl4			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl5			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl6			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl7			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl8			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem
O2T_MSG_Ctrl9			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Mem

Likewise, I did the same for 20 of T20\_MSG\_Ctrl tags.

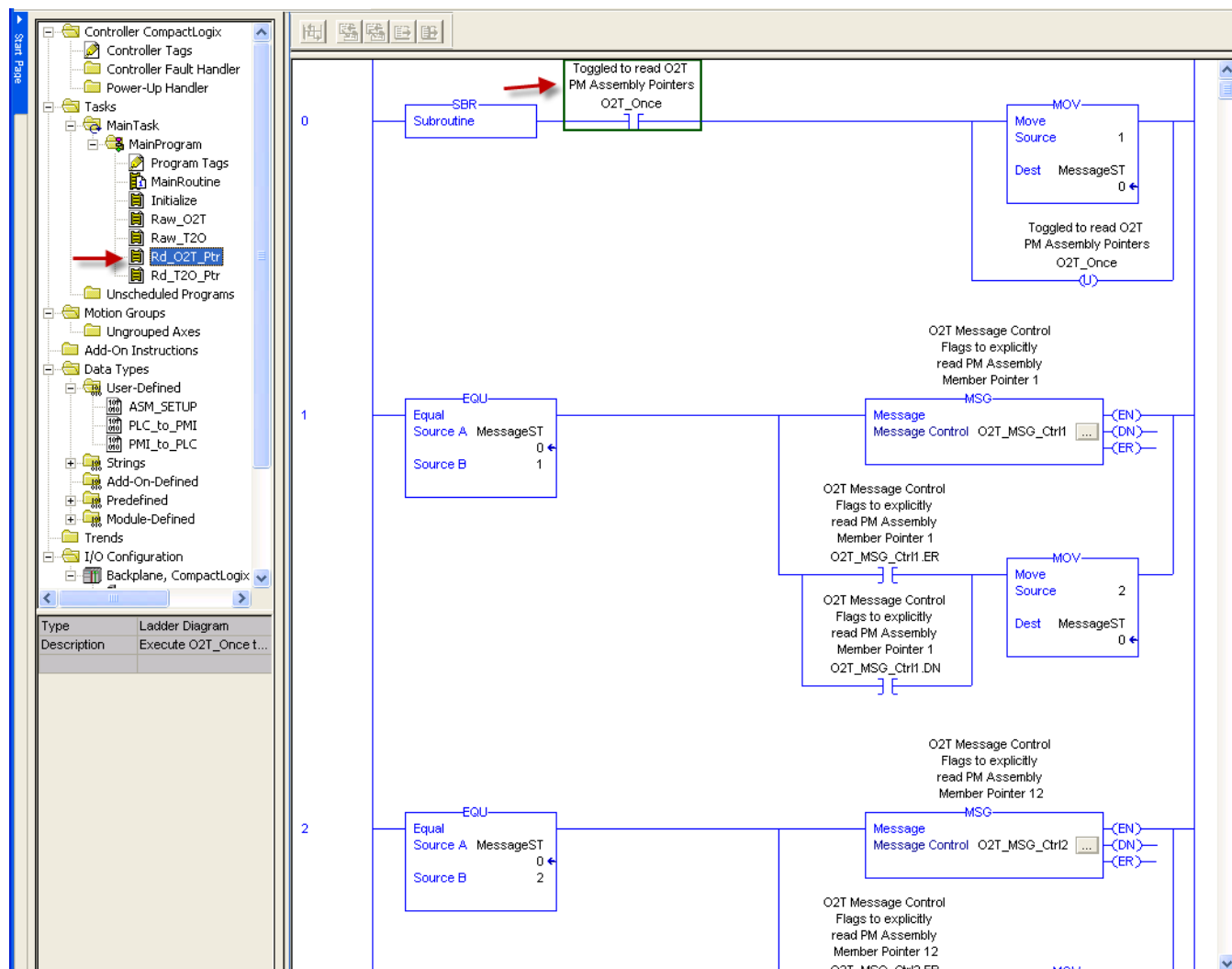
The screenshot shows the 'Controller Tags' window for a CompactLogix controller. The 'Scope' is set to 'CompactLogix'. The 'Show' button is selected. The table lists the following tags:

Name	Alias For	Base Tag	Data Type	Style	Description
T20_MSG_Ctrl1			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl10			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl11			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl12			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl13			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl14			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl15			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl16			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl17			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl18			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl19			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl2			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl20			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl3			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl4			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl5			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl6			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl7			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl8			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P
T20_MSG_Ctrl9			MESSAGE		T20 Message Control Flags to explicitly read PM Assembly Member P

# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

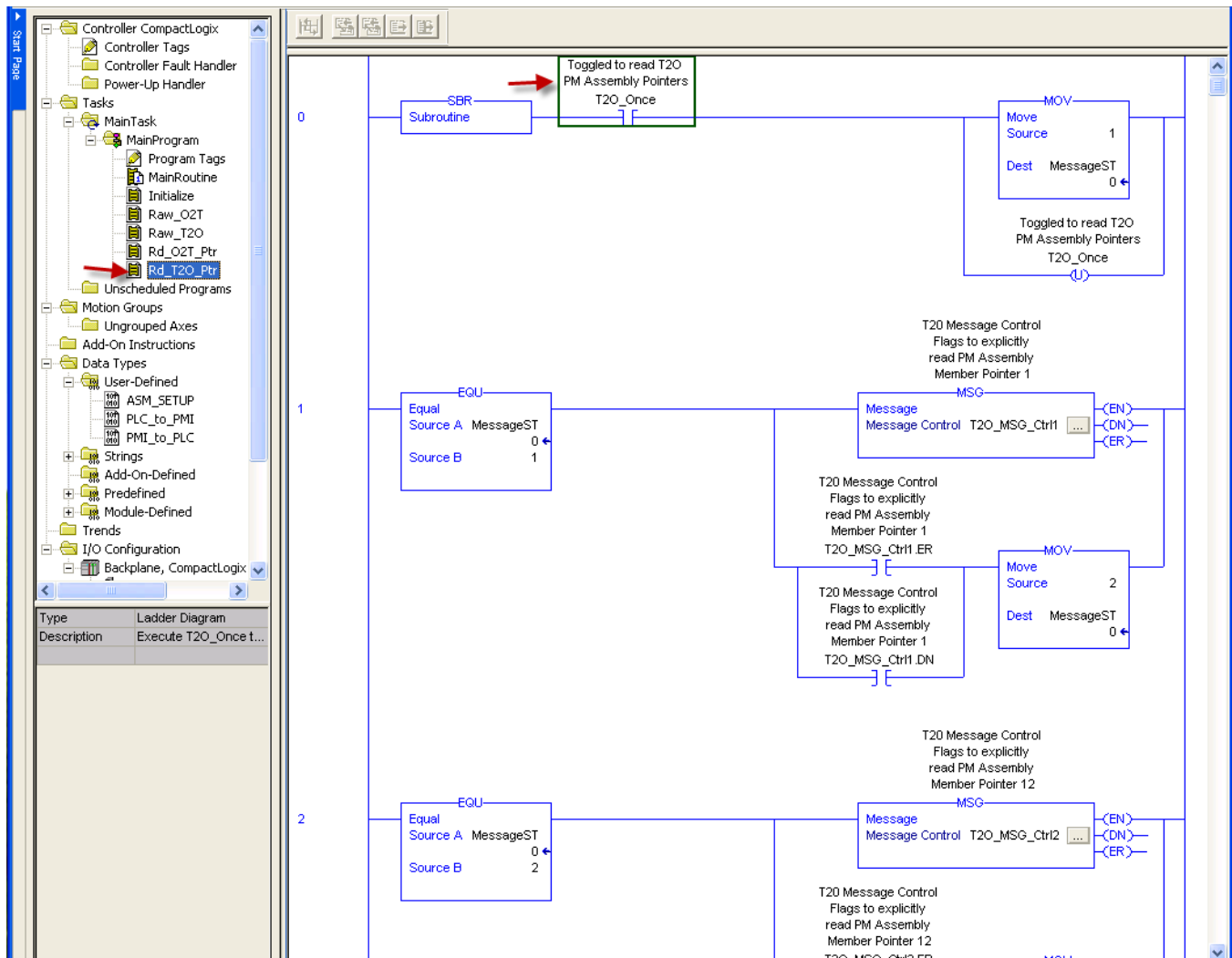
Then I created a subroutine for reading the O2T pointer assembly in ladder logic. I named it Rd\_O2T\_Ptr for Read O2T Pointers. There are 23 rungs; one instruction for each member. Note the contact O2T\_Once; when toggled will run through the process once and transfer the assembly pointers to the ASM\_SETUP structure created earlier.



# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

Then I created a subroutine for reading T20 pointer assembly in ladder logic. I named it Rd\_T20\_Ptr for Read T20 Pointers. There are 23 rungs; one instruction for each member. Note the contact T20\_Once; when toggled will run through the process once and transfer the assembly pointers to the ASM\_SETUP structure created earlier.



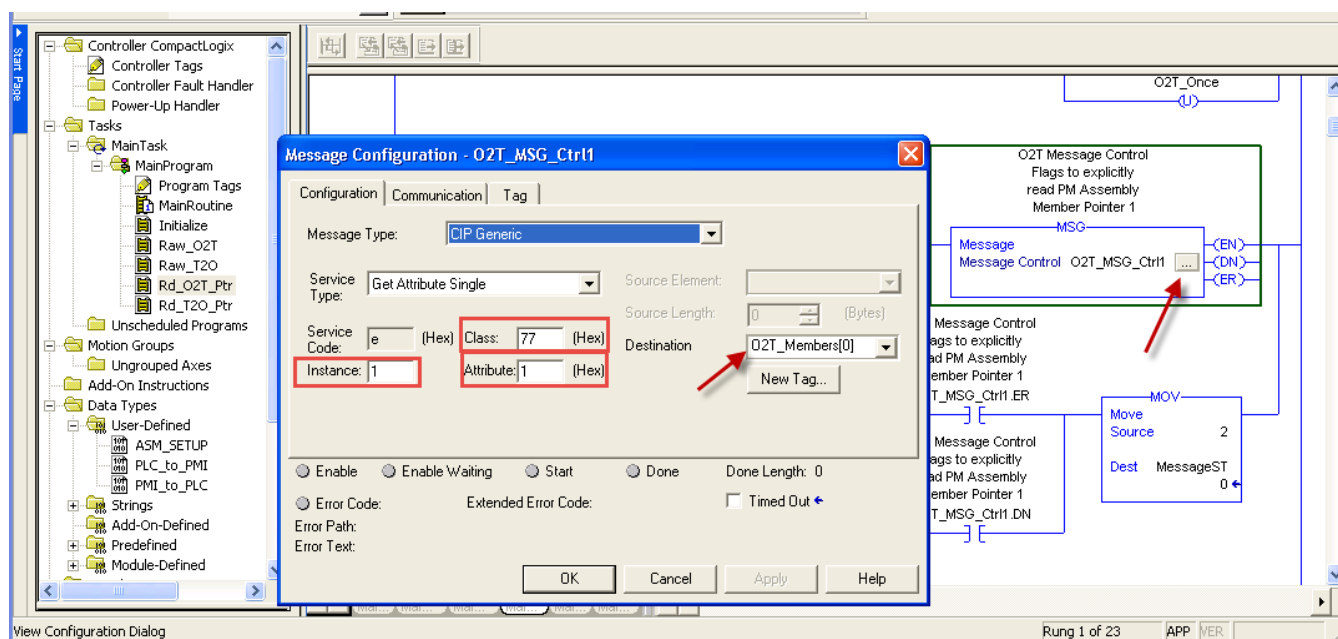
# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

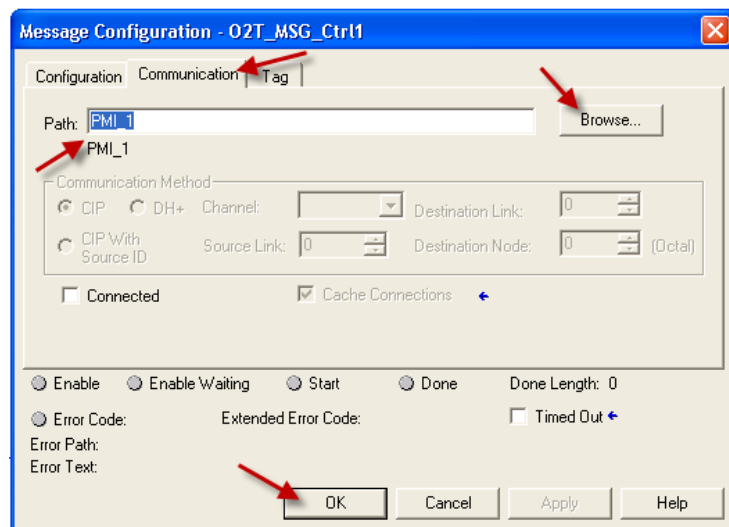
The MSG instruction for each rung defines Message Type as CIP Generic and Service Type Get Attribute Single. You define which assembly member pointer to get and transfer into O2T\_Member[x]. The first message instruction is getting the first assembly member for the Originator to Target (O2T).

### CIP Implicit O to T (Originator to Target) Assembly Structure

CIP Implicit Assembly → Originator (Master) to Target (PM)					
Assembly Members	PM Assembly Class, Instance, Attribute	PM Data Type	Parameter	Parameter Class, Instance, Attribute	PLC Data Type
1	0x77, 0x01, 0x01	DINT	Loop Control Mode	0x97, 0x01, 0x01	DINT
2	0x77, 0x01, 0x02	DINT	Closed Loop Set Point	0x6B, 0x01, 0x01	REAL
3	0x77, 0x01, 0x03	DINT	Open Loop Set Point	0x6B, 0x01, 0x02	REAL



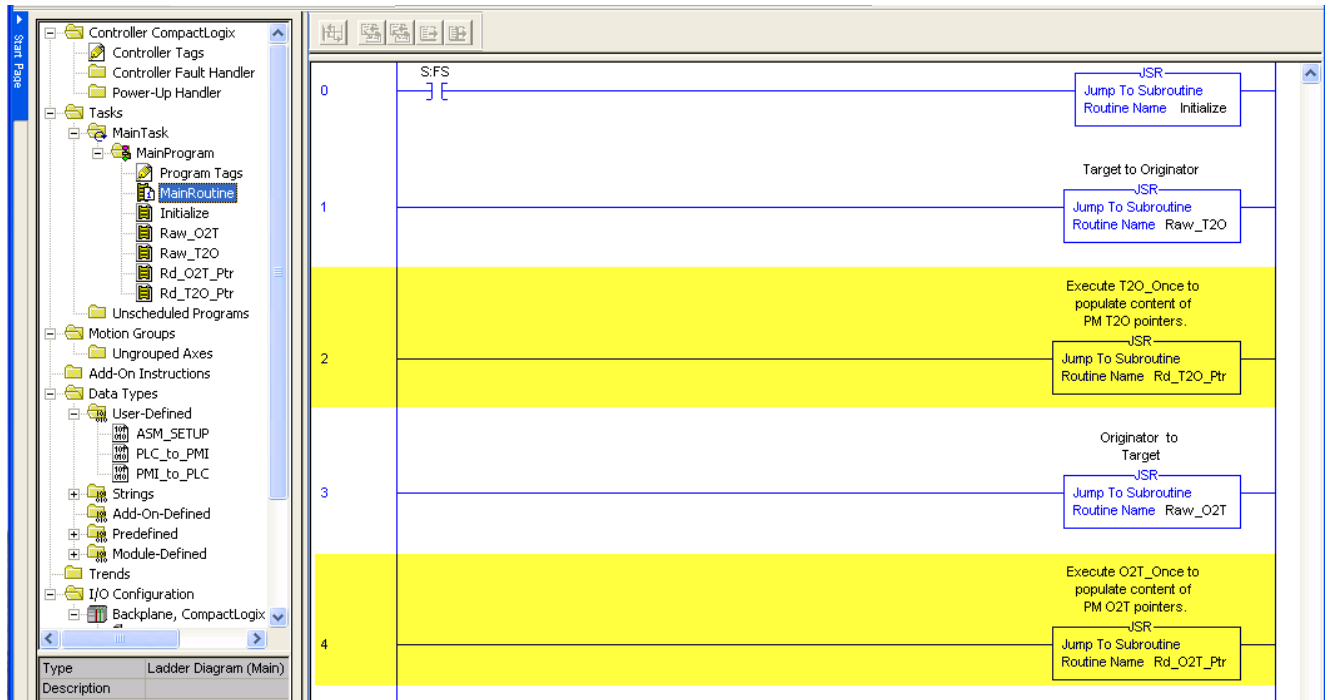
Using the Communication Tab, select the path to the controller.



# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

Don't forget to add code to the MainRoutine to call these subroutines.





# EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic

## Example AB CompactLogix or ControlLogix PLC

Download to PLC, enter Run Mode, trigger the O2T\_Once contact and then view the Controller Tags for O2T\_Members[0] to [19]. Shown here are the first three members. Assembly member 1 has Class 0x97, Instance 0x01, Attribute 0x01 pointer. This is the default parameter pointer of Assembly Member 1. Notice the next two members match as well.

### CIP Implicit O to T (Originator to Target) Assembly Structure

CIP Implicit Assembly Originator (Master) to Target (PM)					
Assembly Members	PM Assembly Class, Instance, Attribute	PM Data Type	Parameter	Parameter Class, Instance, Attribute	PLC Data Type
1	0x77, 0x01, 0x01	DINT	Loop Control Mode	0x97, 0x01, 0x01	DINT
2	0x77, 0x01, 0x02	DINT	Closed Loop Set Point	0x6B, 0x01, 0x01	REAL
3	0x77, 0x01, 0x03	DINT	Open Loop Set Point	0x6B, 0x01, 0x02	REAL

RSLogix 5000 - CompactLogix in EtherNet/IP PMI\_AppNote1.ACD [1769-L32E] - [Controller Tags - CompactLogix(controller)]

File Edit View Search Logic Communications Tools Window Help

Path: ENIP\_PLC\192.168.0.51\Backplane\0

Rem Run: Run Mode, Controller OK, Battery OK, I/O OK

Scope: CompactLogix

Name	Value	Force	Style	Data Type	Description
HardwareID	0		Decimal	DINT	
Local1:1	{...}	{...}		AB:1769_...	
Local1:0	{...}	{...}		AB:1769_...	
Member_1	16#0000_0000		Hex	DINT	
MessageST	0		Decimal	DINT	
O2T_Members	{...}	{...}		ASM_SET...	Structure to hold Class, Instance and Attribute for Implicit
O2T_Members[0]	{...}	{...}		ASM_SET...	Structure - O2T PM Assembly Member 1
O2T_Members[0].ClassID	16#97		Hex	SINT	O2T PM Class ID for Assembly Member 1
O2T_Members[0].InstanceID	16#01		Hex	SINT	O2T PM Instance ID for Assembly Member 1
O2T_Members[0].AttributeID	16#01		Hex	SINT	O2T PM Attribute ID for Assembly Member 1
O2T_Members[1]	{...}	{...}		ASM_SET...	Structure - O2T PM Assembly Member 2
O2T_Members[1].ClassID	16#6b		Hex	SINT	O2T PM Class ID for Assembly Member 2
O2T_Members[1].InstanceID	16#01		Hex	SINT	O2T PM Instance ID for Assembly Member 2
O2T_Members[1].AttributeID	16#01		Hex	SINT	O2T PM Attribute ID for Assembly Member 2
O2T_Members[2]	{...}	{...}		ASM_SET...	Structure - O2T PM Assembly Member 3
O2T_Members[2].ClassID	16#6b		Hex	SINT	O2T PM Class ID for Assembly Member 3
O2T_Members[2].InstanceID	16#01		Hex	SINT	O2T PM Instance ID for Assembly Member 3
O2T_Members[2].AttributeID	16#02		Hex	SINT	O2T PM Attribute ID for Assembly Member 3
O2T_Members[3]	{...}	{...}		ASM_SET...	Structure - O2T PM Assembly Member 4
O2T_Members[3].ClassID	16#6d		Hex	SINT	O2T PM Class ID for Assembly Member 4
O2T_Members[3].InstanceID	16#01		Hex	SINT	O2T PM Instance ID for Assembly Member 4
O2T_Members[3].AttributeID	16#01		Hex	SINT	O2T PM Attribute ID for Assembly Member 4

You can perform the same for the T20\_Once contact and then view the Controller Tags for T20\_Members[0] to [19].

That gets us to the point where we have communications, we validate the assembly configuration and we determined we want to change the order or the pointer for one or more members.

## **EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic**

### **Example AB CompactLogix or ControlLogix PLC**

Suppose you want the Heat Proportional Band in the location. The PM Users' Guide shows the CIP as 0x97, 1, 0x06. Using a message instruction we send 0x97, 1, 0x06 to 0x77, 1, 0x03.

First we need a subroutine to Set Attribute.