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.

- 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

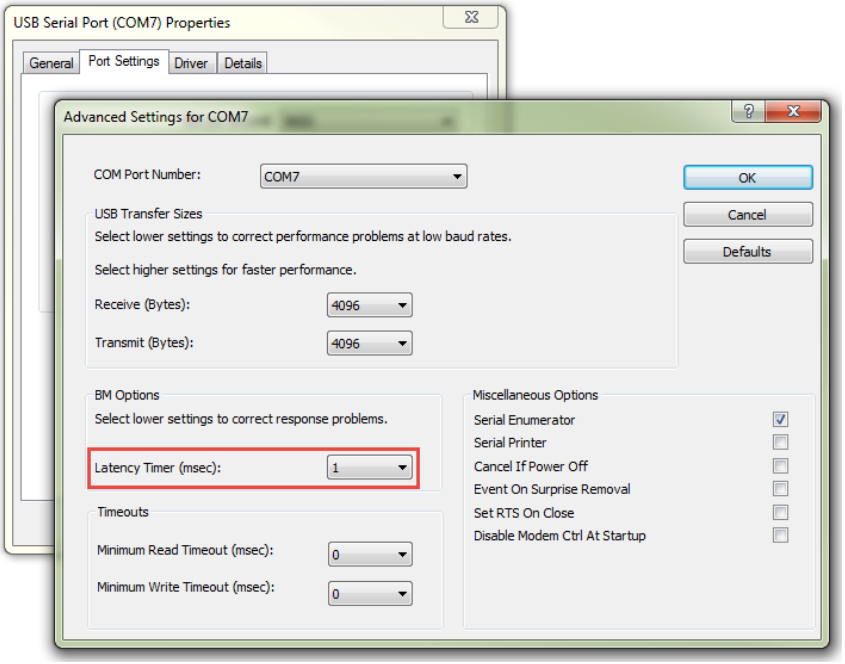
Basically, you need to program the PLC to send data to the controller and tell the PLC what data is expected from the controller. The PLC will use a generic I/O structure for this definition. The PLC will be programmed for the requested packet interval time, the size of the structure for inputs/outputs 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 with a managed switch supporting IGMP snooping. The controller supports 10/100Mbps in half-duplex. 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.

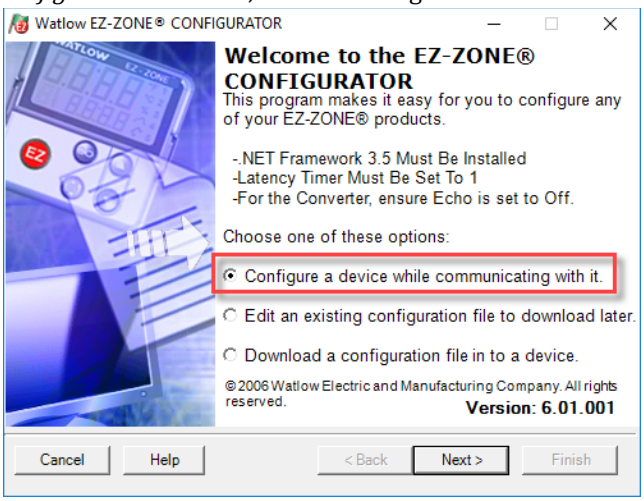
If you need to change the default Implicit Assemblies, you are required to use an Explicit message instruction from the PLC. To establish explicit communications between Master and Slave devices, configuration steps need to be executed within the PLC as well as within the EZ-ZONE PM using either the 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.

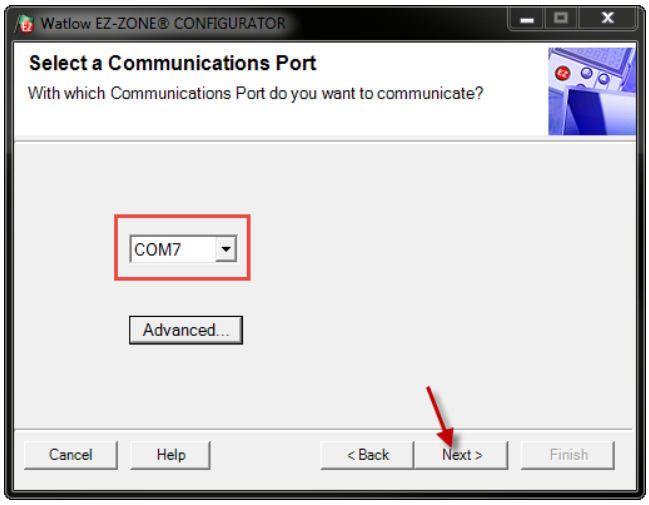
1) *Connect the PC* to the EZ-ZONE PM on terminals CD, CE and CF of slot C using an EIA-485 serial port. Typically the converter is a USB to EIA-485 device such as B&B Electronics 485USBTB-2W. Install Windows driver included with converter.
2) *Check latency timer* in serial driver of PC for 1mS setting. 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.



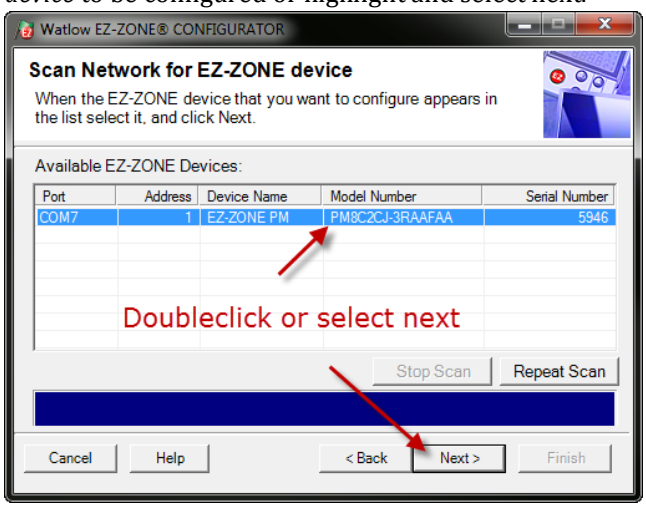
3) *Install EZ-ZONE Configurator* software on to PC. The program is located on the Watlow website under Download Center, Software and Demos category.
4) *Execute EZ-ZONE Configurator* software, choose 'Configure a device…'



5) Locate serial port and click next.

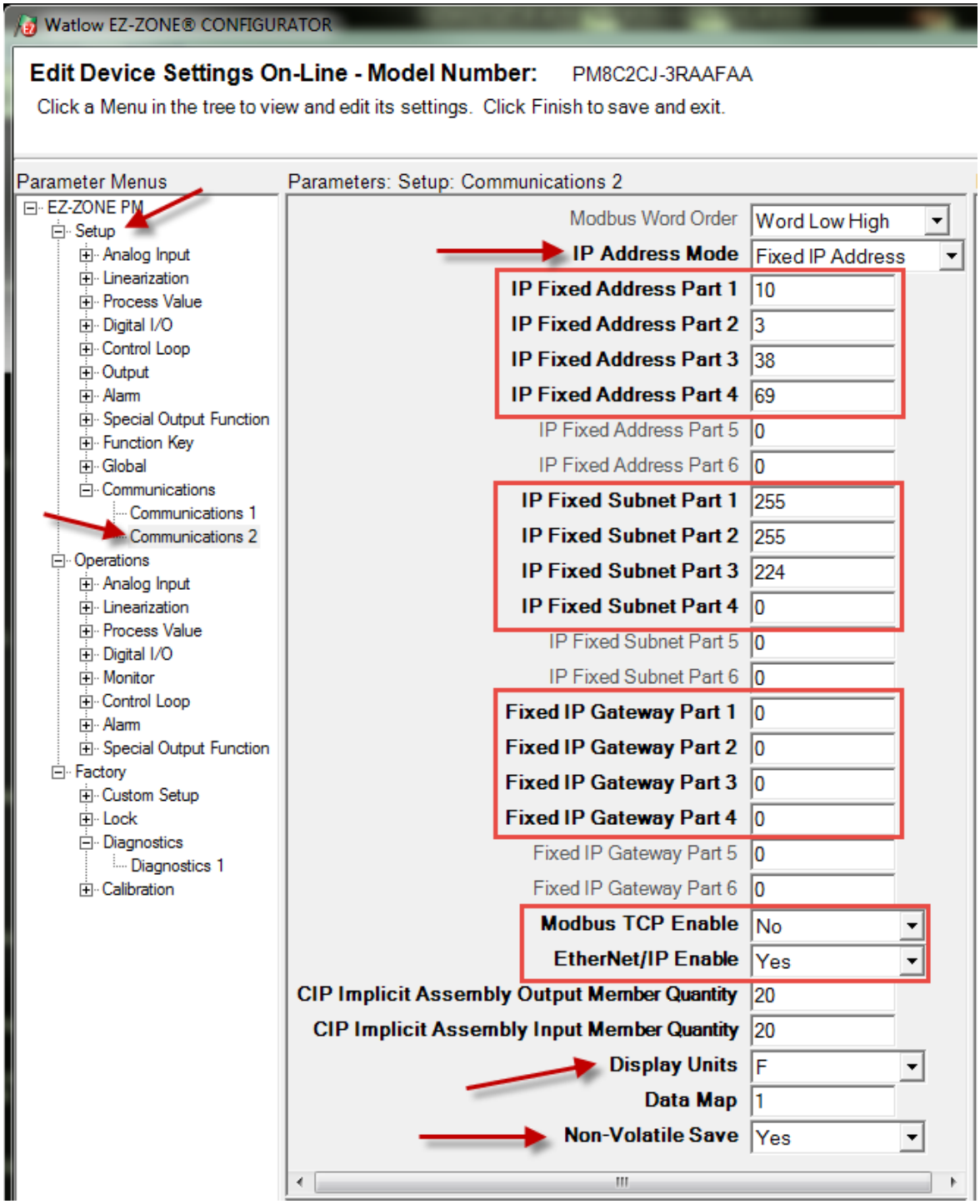


6) *Doubleclick on PM device* to be configured or highlight and select next.



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.



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, Attritbute	PM Data Type	Parameter	Parameter Class, Instance, Attritbute	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

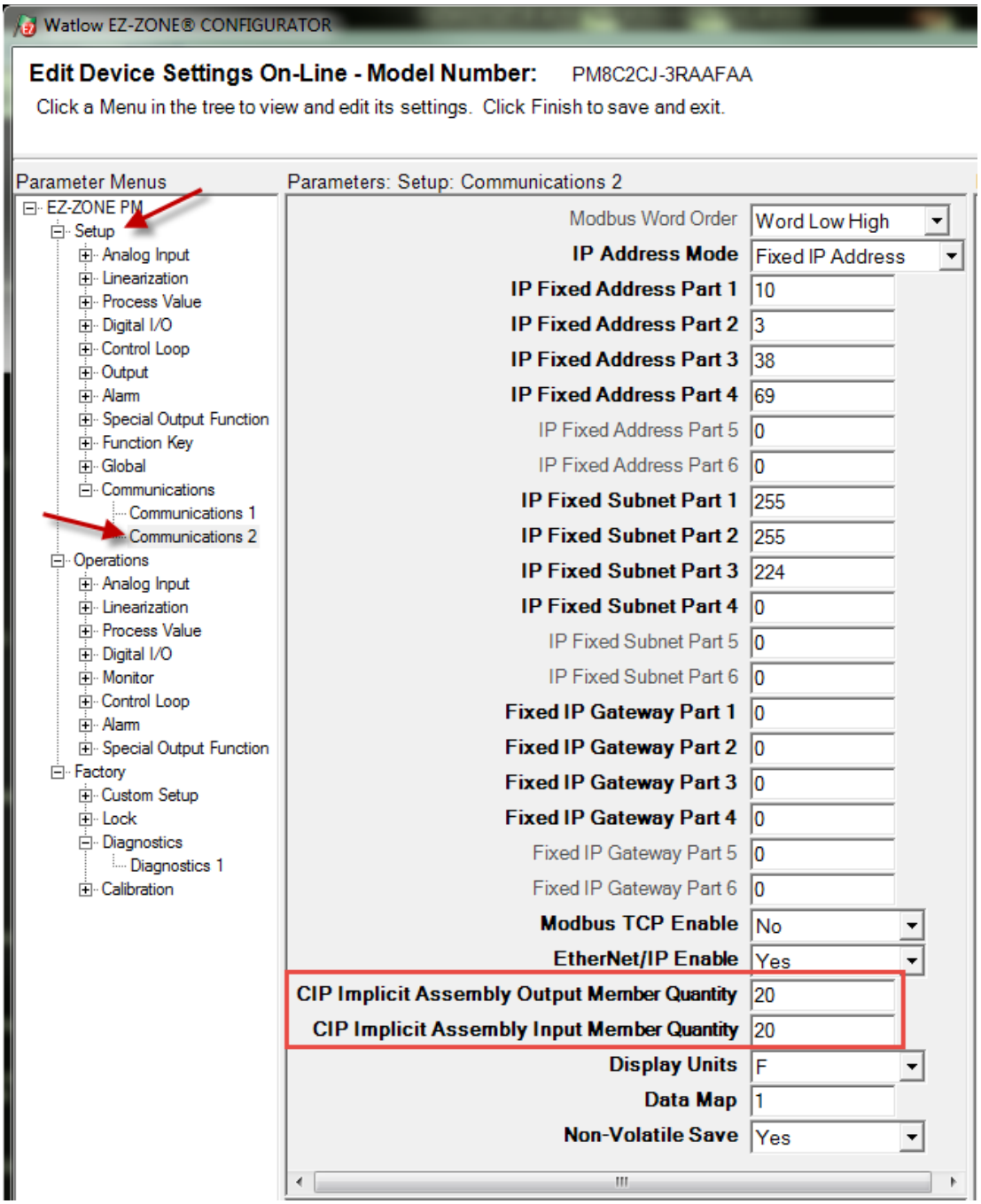
15) *Identify the parameters in the controller to be read* by the PLC. The PLC references these as inputs. The controller references these as CIP Implicit Assembly Output Member Quantity (T to O). Count the number identified.

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

CIP Implicit Assembly Target (PM) to Originator (Master)					
Assembly Members	PM Assembly Class, Instance, Attritbute	PM Data Type	Parameter	Parameter Class, Instance, Attritbute	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.

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 ideentified 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.

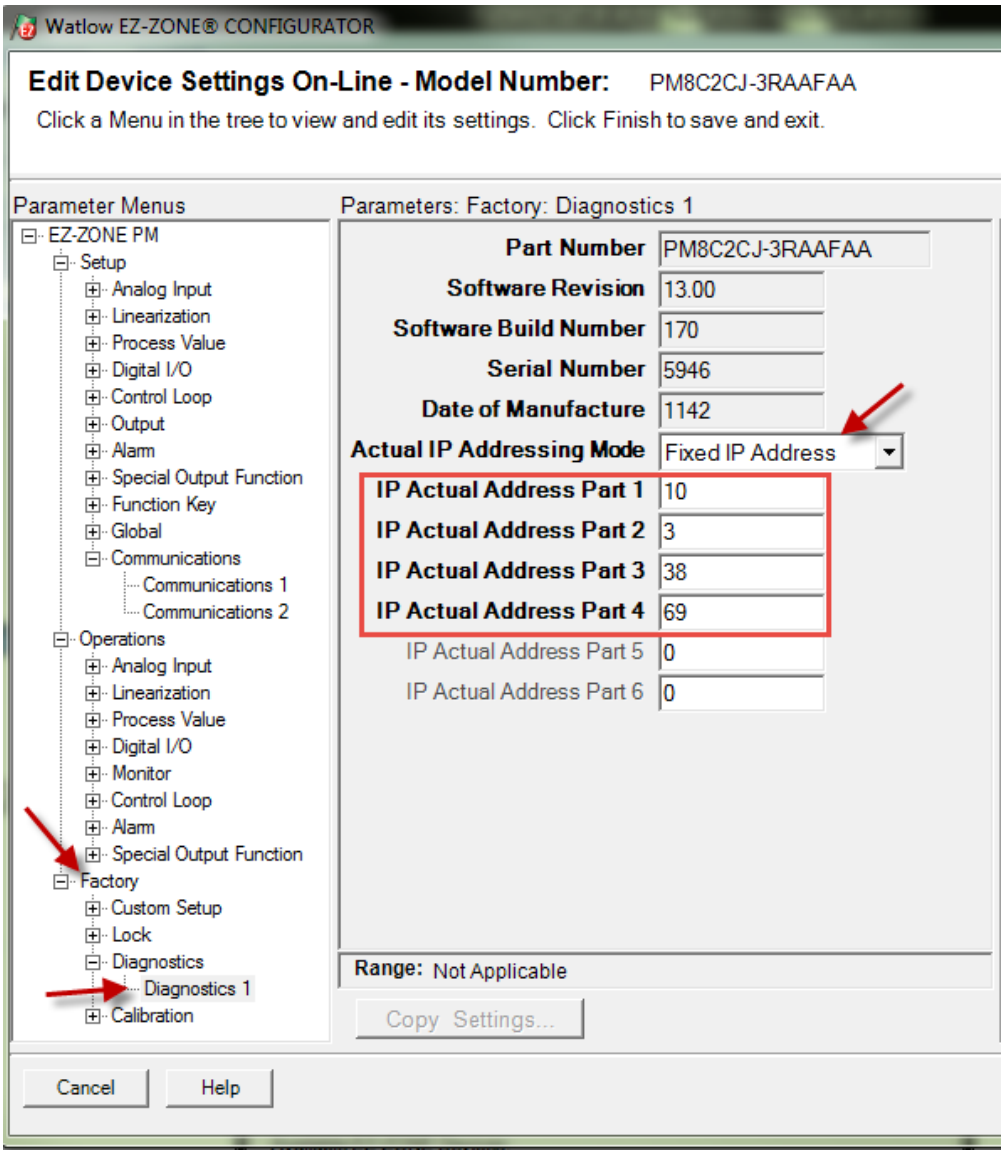


17) *Remove power from the controller.* You must cycle power on the EZ-ZONE PM controller for a new Fixed IP address to take effect.

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

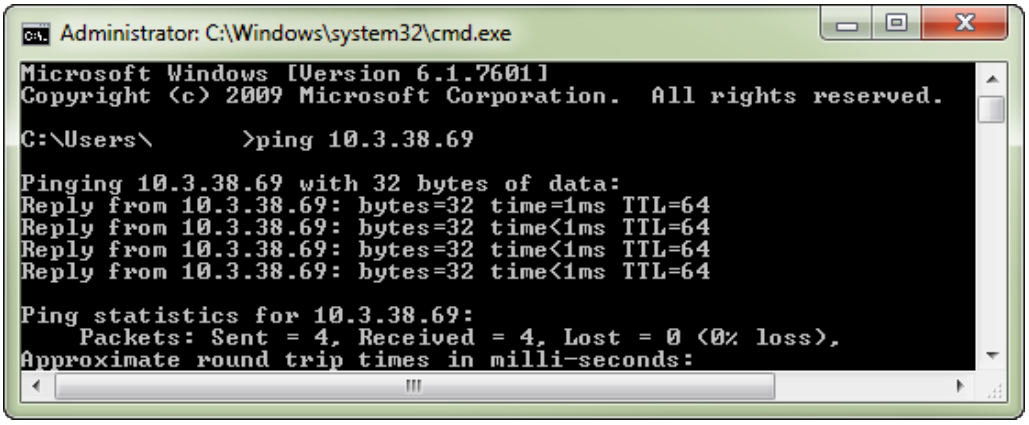
what action. That is not related to setting up PLC communications so we will not cover that in this document.

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.



EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic Example AB CompactLogix or ControlLogix PLC

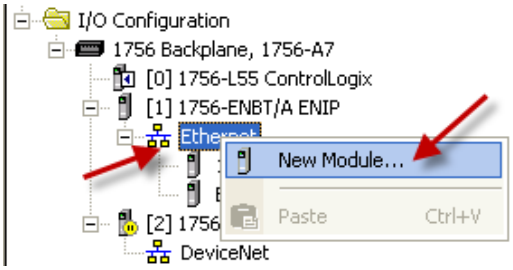
You can use your PC to check the IP address of the PM controller by using a DOS Command prompt and ping the address for response. The PC must be on the same logical IP address as the controller. This screen capture shows four responses of a device at 10.3.38.69 so we know the IP address is correct (if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



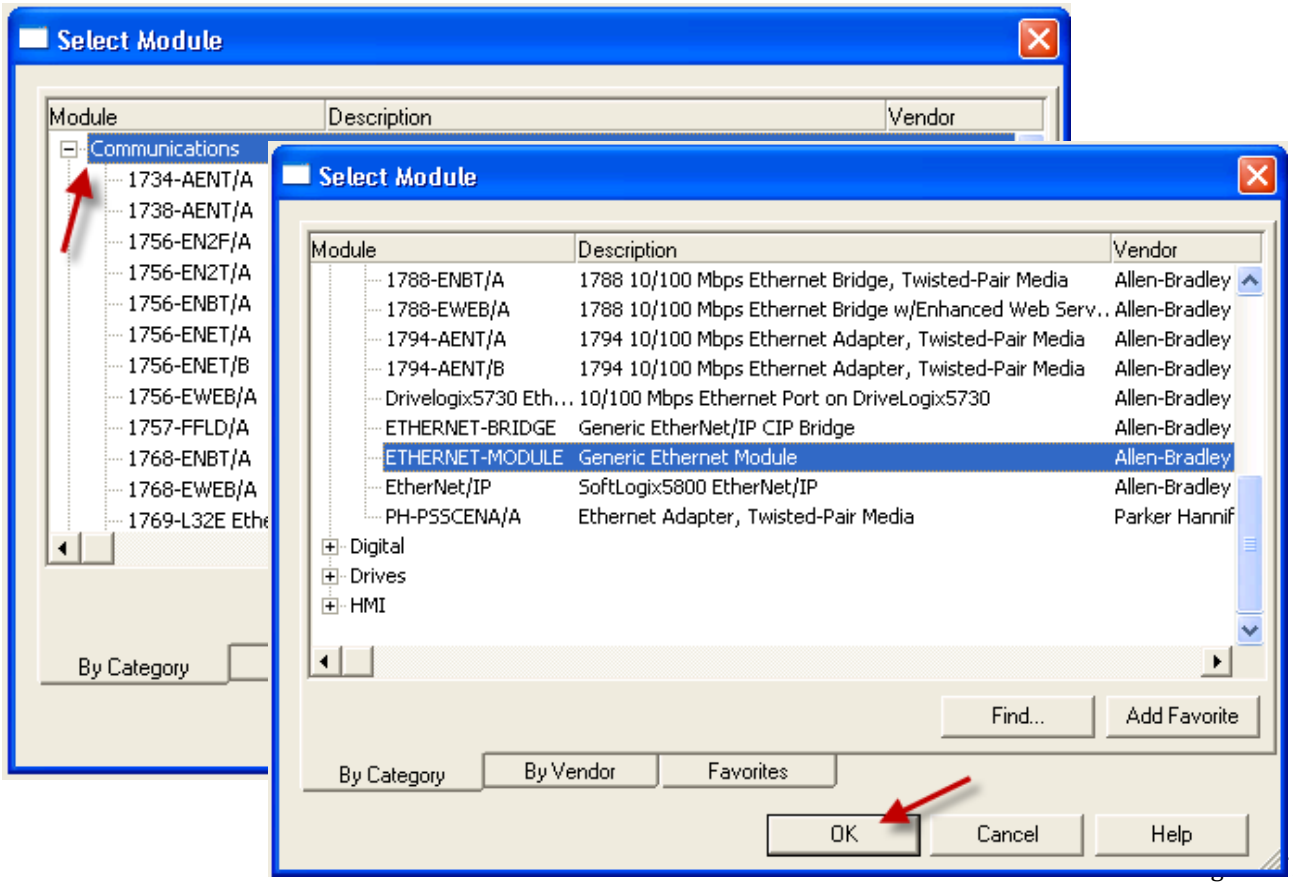
EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic Example AB CompactLogix or ControlLogix PLC

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

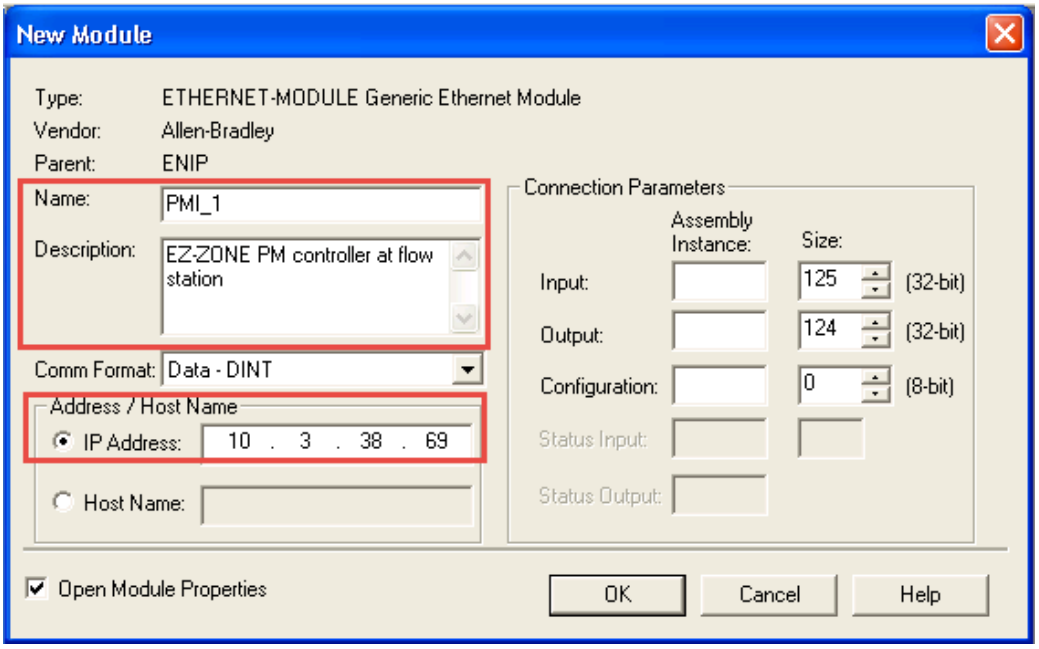
1) *Create a list of PLC outputs like the default* show in the users' guide if you plan to change the O to T assembly members. Previously you identified the controller parameters to be written into from the PLC. The PLC references these as outputs. The controller references these as CIP Implicit Assembly 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.



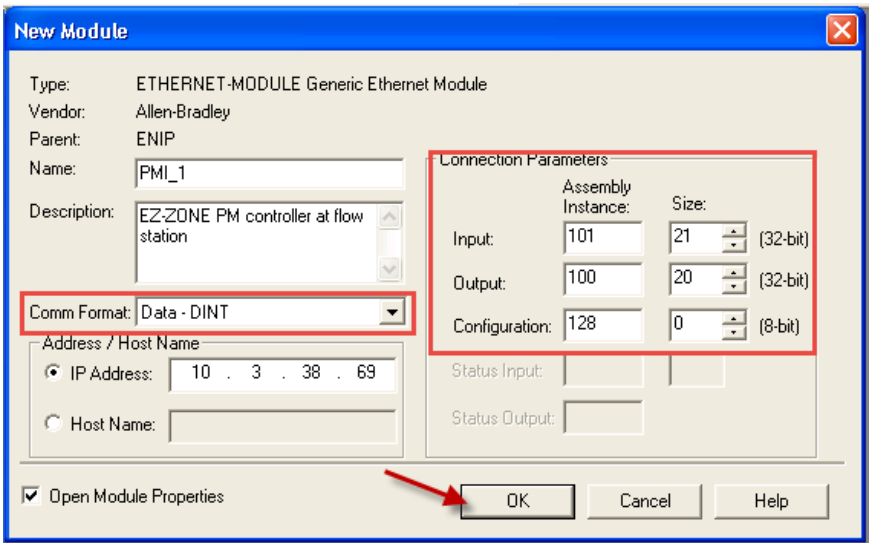
5) *Expand Communications* and scroll to ETHERNET-MODULE with description of Generic Ethernet Module. Select this option and click OK.



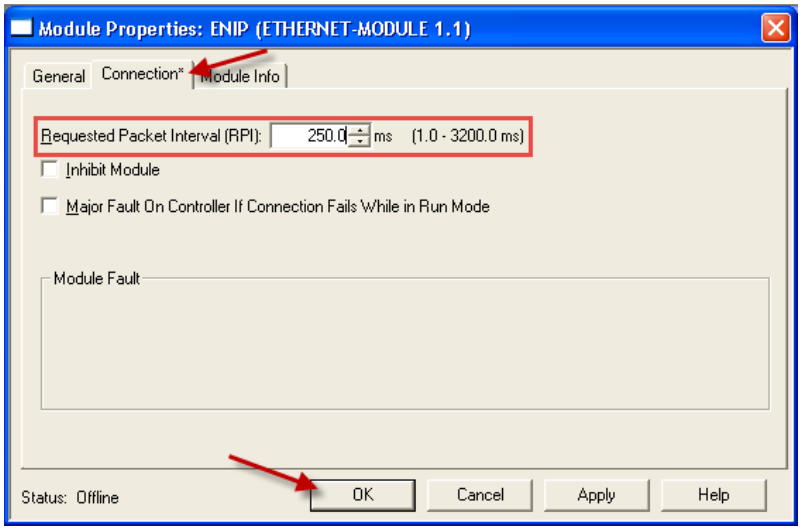
6) *Enter a descriptive name, add description to identify product on network and enter IP Address of PM Controller.* I used PMI _1 for PM Integrated controller with ZONE 1 on display.



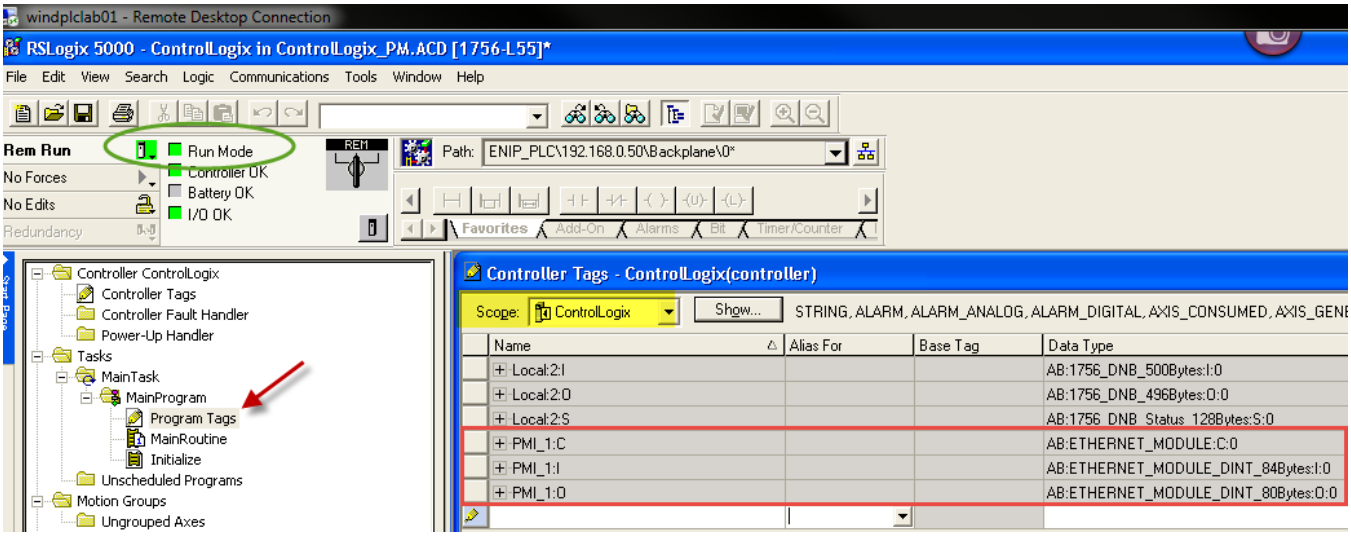
7) *Define the I/O Implicit Assembly*; enter Input Assembly Instance 101, Output Assembly Instance 100, and Configuration 128 with a size of 0. Set the Input Size to the number of parameters to be read from the EZ-ZONE PM controller by the PLC plus one 32-bit value representing the status of the PM controller. 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.



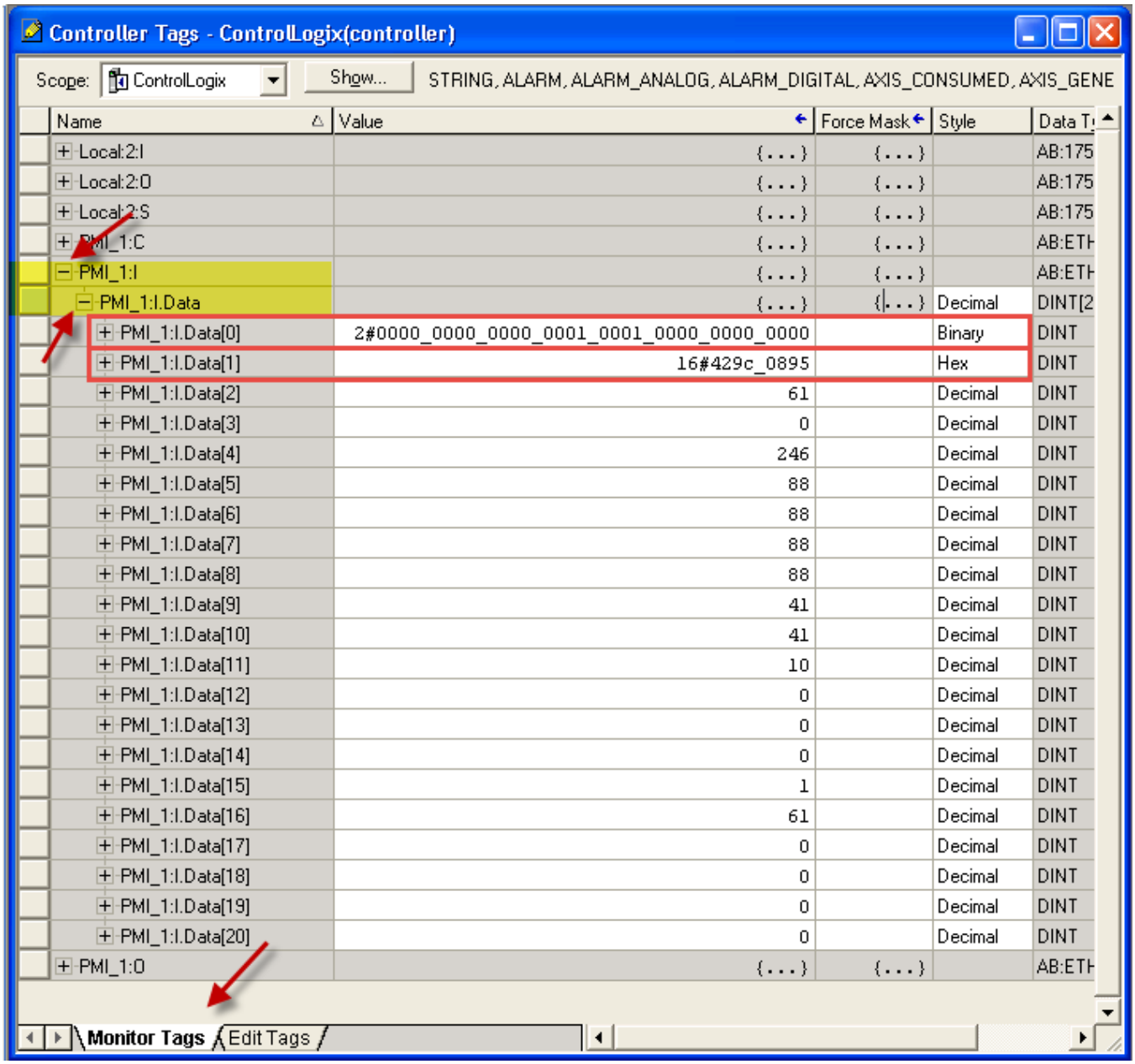
8) *Enter Requested Packet Interval (RPI) setting* in Module properties, connection tab. Enter a value equal to or greater than 250.0 mS. Use the (RPI) setting to minimize network traffic if concerned. This setting determines the speed for data transaction of implicit messaging; larger values mean less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

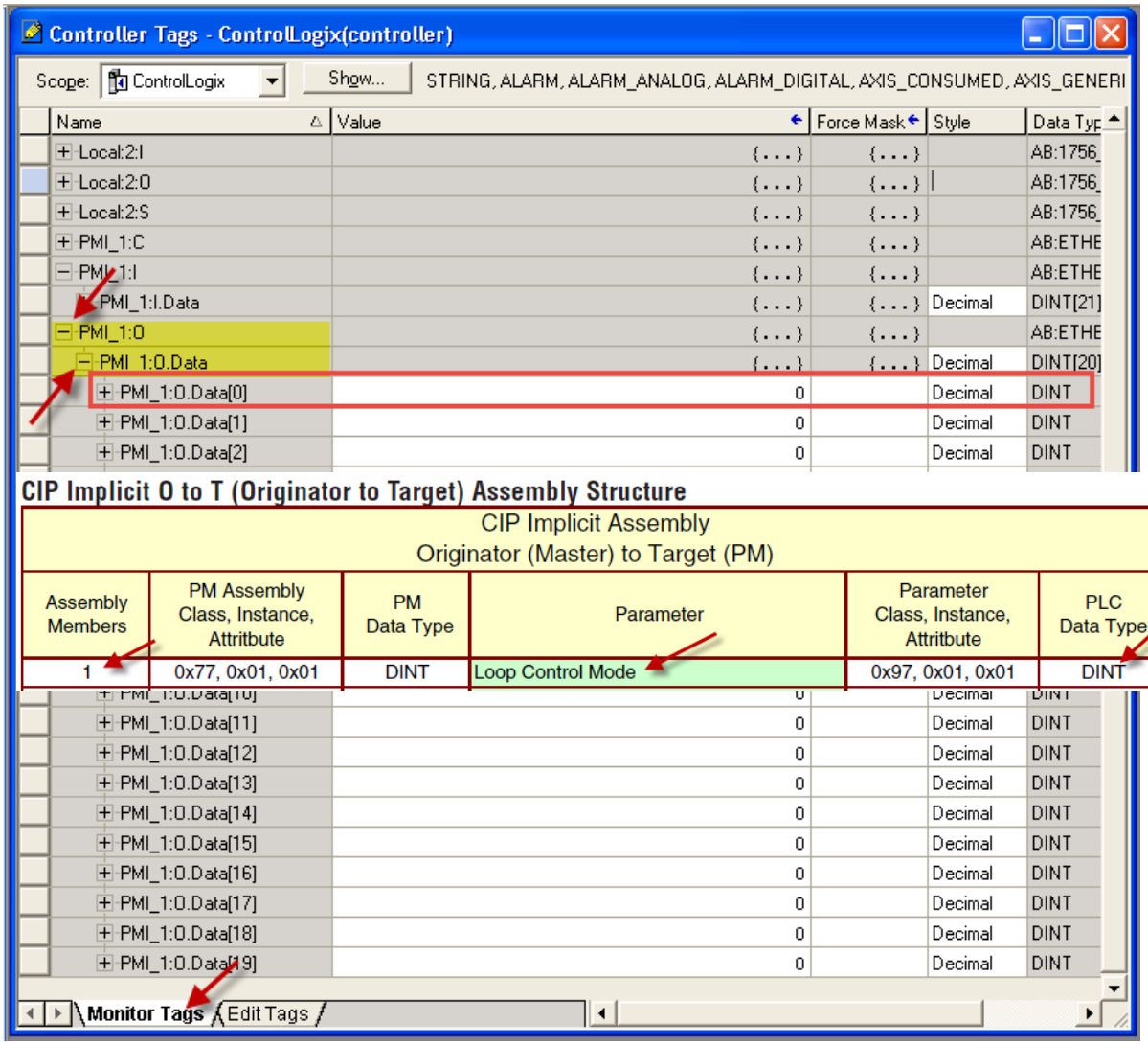


12) *View Monitor Tags tab. Expand the Input Data (PMI_1:**I** in my project)* to see actual raw data being displayed. This is the Target to Originator data. Change the Style for the first Assembly Member to Binary. Change the Style for the second Assembly Member to Hex. We used the default PM controller 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.



Name	Value	Force Mask	Style	Data T
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

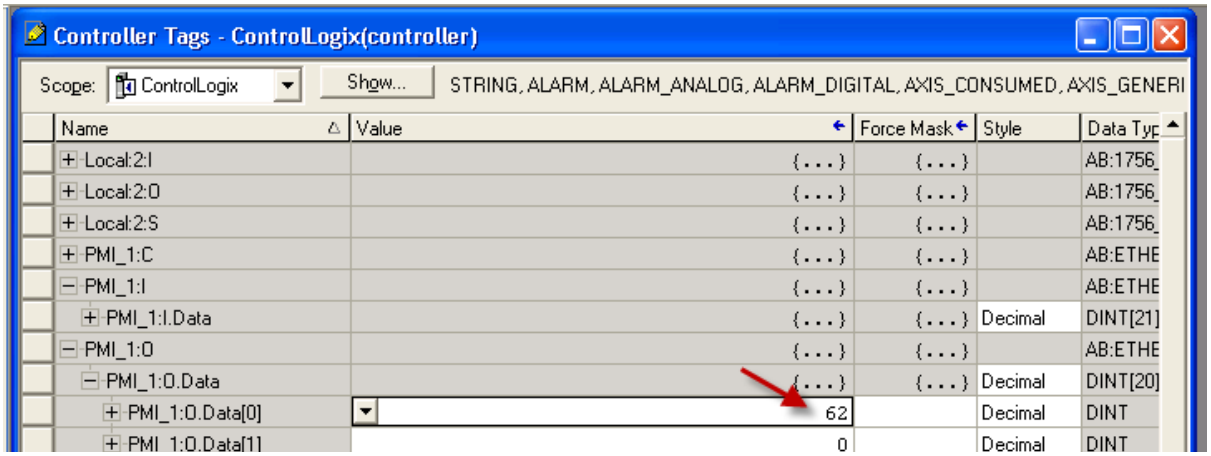
13) *View Monitor Tags tab. Expand the Output Data (PMI_1:**O** in my project)* to test the Originator to Target assembly. Recall that the default assembly has the first Assembly Member defined as Loop Control Mode. According to the PM Integrated Users' guide, the values for Control Mode are 10=Auto (Closed Loop), 54=Manual (Open Loop) or 62=Off (no control). The register is of data type DINT.



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

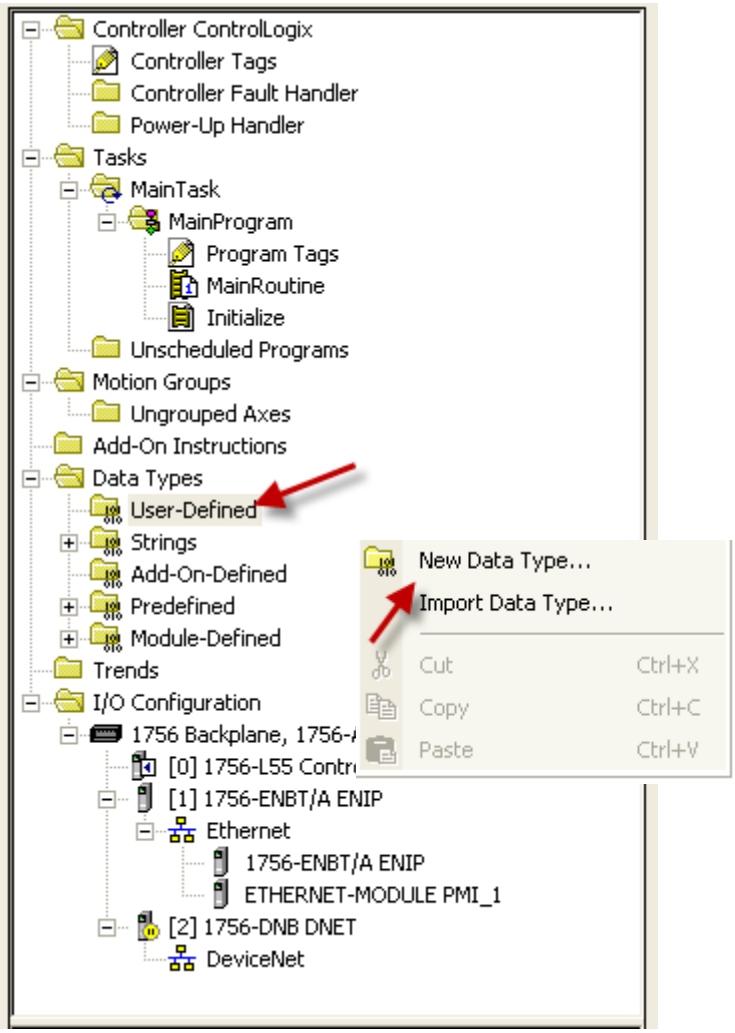
CIP Implicit Assembly Originator (Master) to Target (PM)					
Assembly Members	PM Assembly Class, Instance, Attritbute	PM Data Type	Parameter	Parameter Class, Instance, Attritbute	PLC Data Type
1	0x77, 0x01, 0x01	DINT	Loop Control Mode	0x97, 0x01, 0x01	DINT

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.

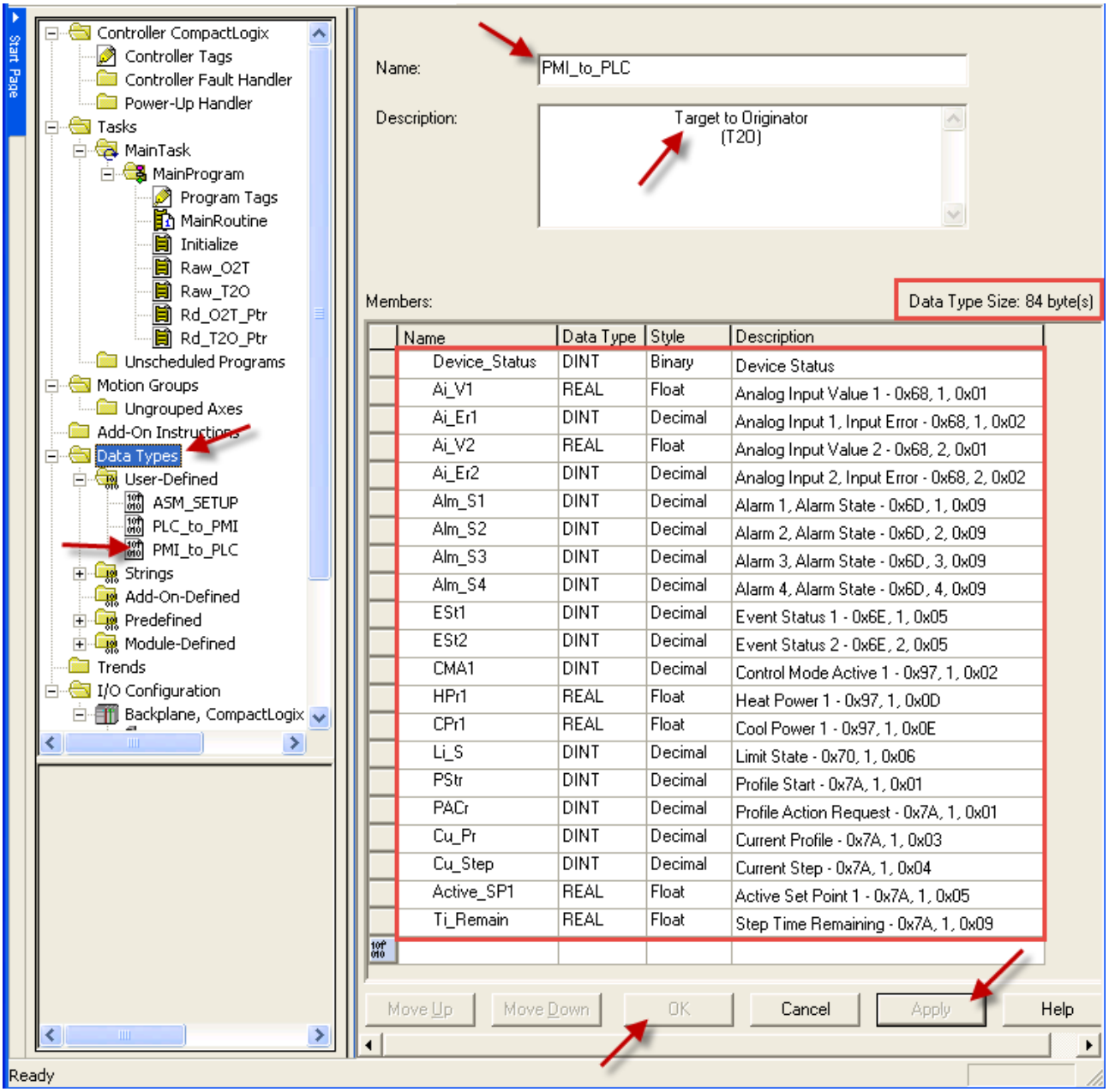


We now have a functioning system where implicit messaging is occurring. The next step is to create a structure to convert the raw data to identifiable tags and readable values.

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



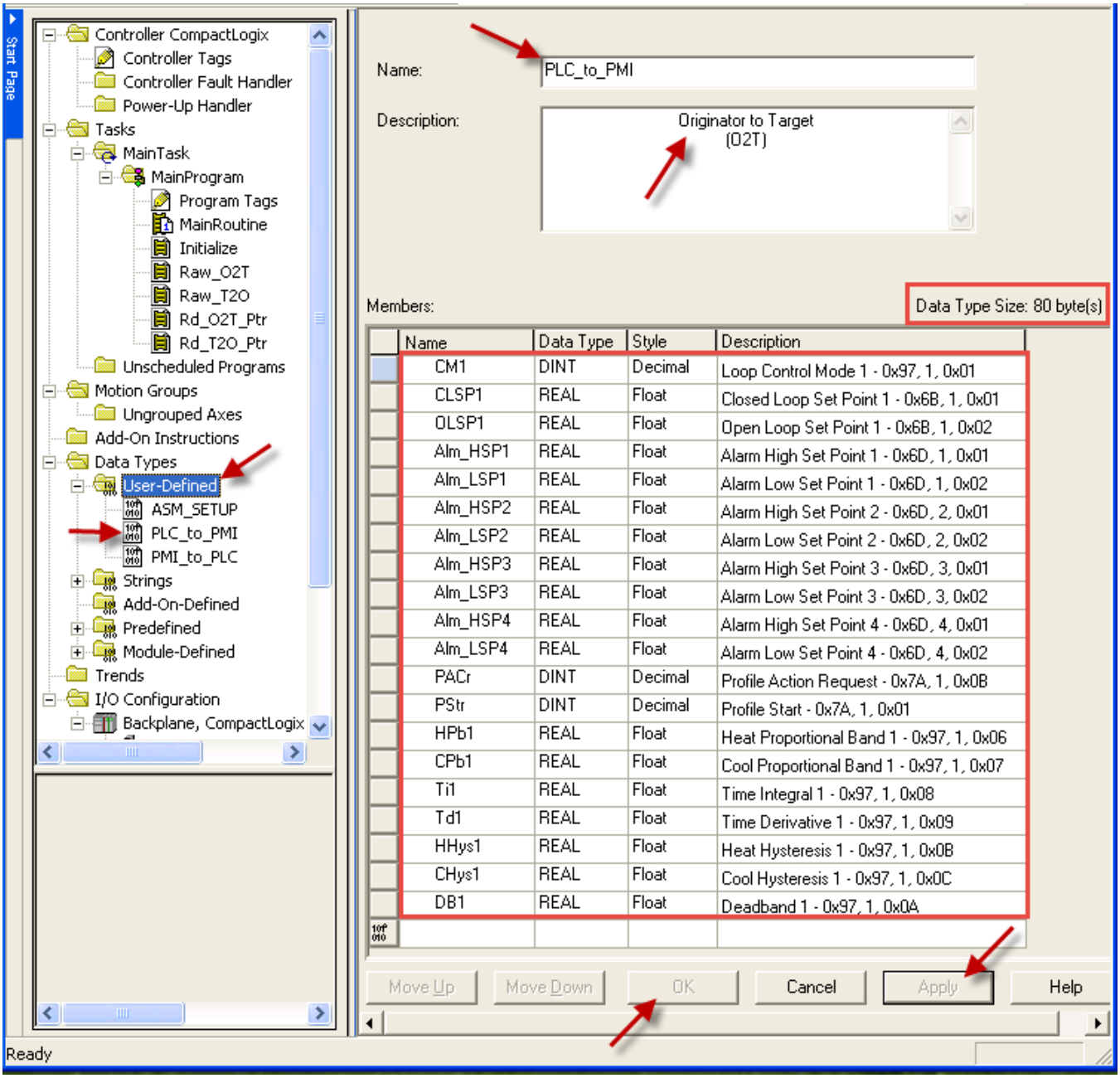
16) *Define User Data Type* to match output assembly of PM controller. Provide a name to help identify the data structure. Include a description for further understanding. For each Member, enter in a short name, data type for the parameter being entered and a description. Click Apply and the screen will 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 T2O. 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.



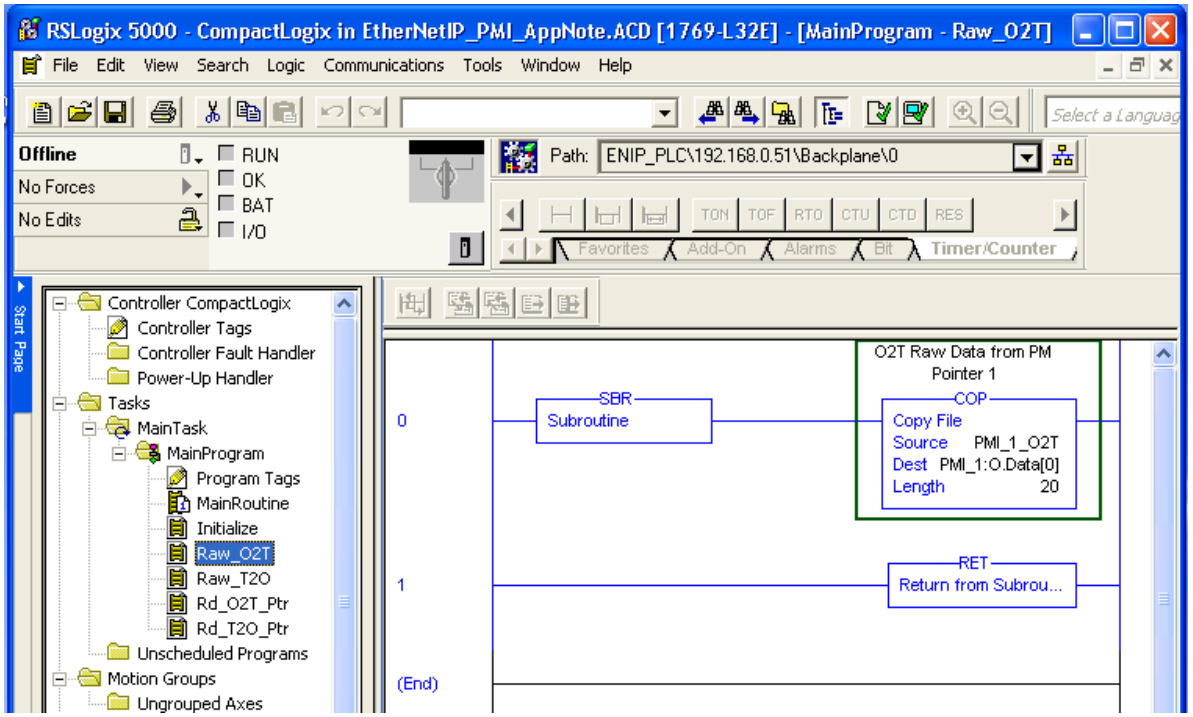
Ai_V2

EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic Example AB CompactLogix or ControlLogix PLC

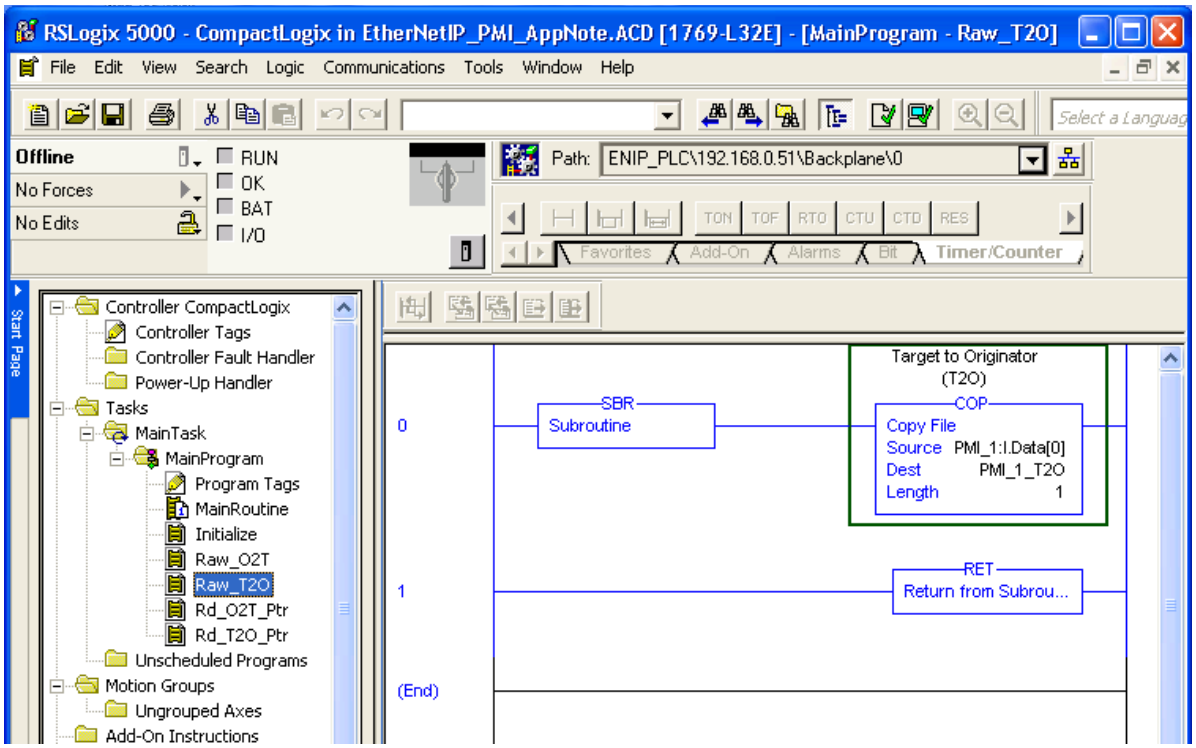
17) *Define User Data Type* to match input assembly of PM controller. Provide a name to help identify the data structure. Include a description for further understanding. For each Member, enter in a short name, data type for the parameter being entered and a description. Click Apply and the screen will 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.



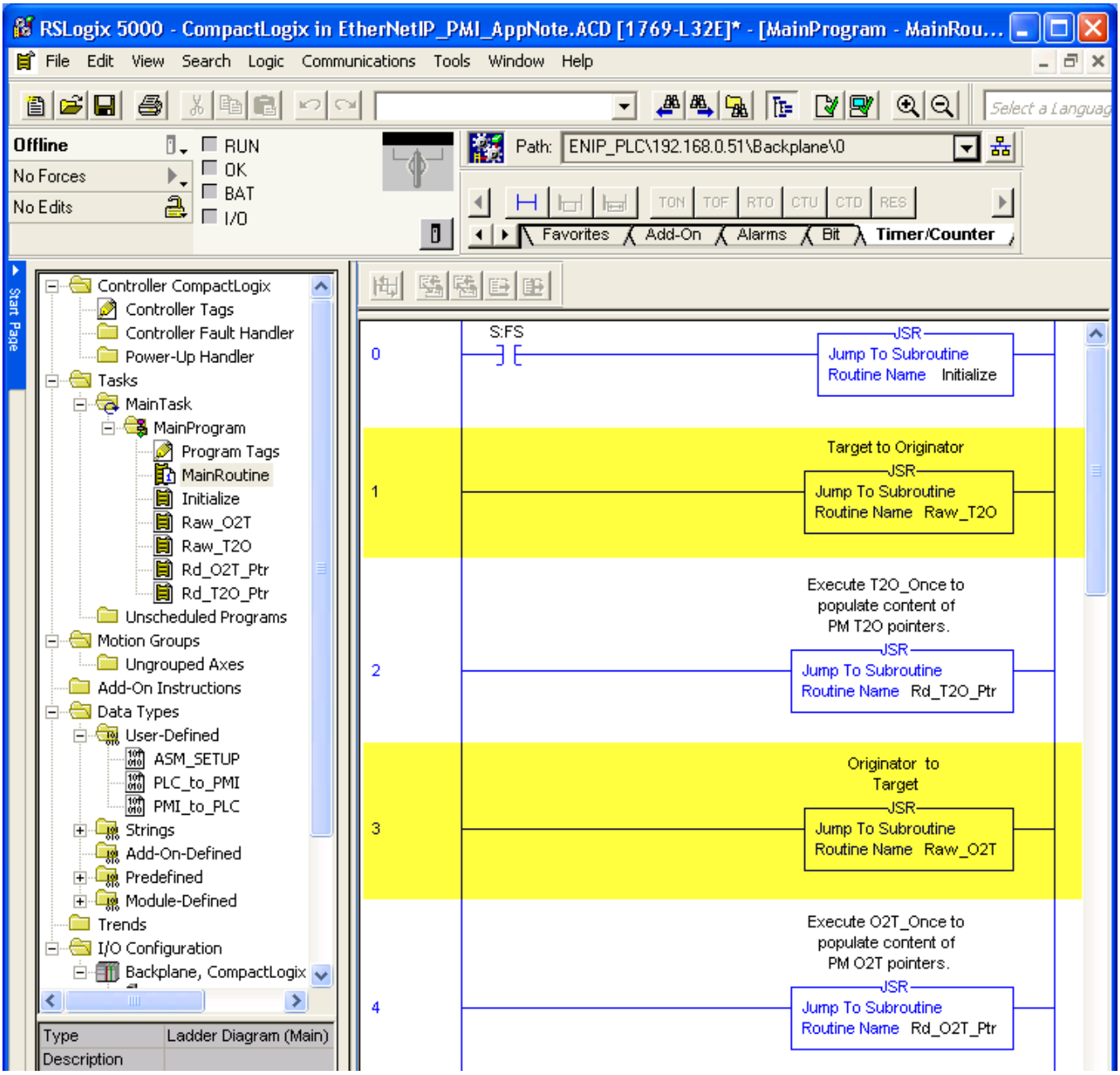
18) *Program ladder logic to copy data between user data type (O2T) and implicit assembly originator.* Add a Subroutine with copy block. This instruction will copy from the user structure PMI_1_O2T into PMI_1:O.Data(0). Be sure the Length (20 in my example) matches the number of members being copied as defined earlier.



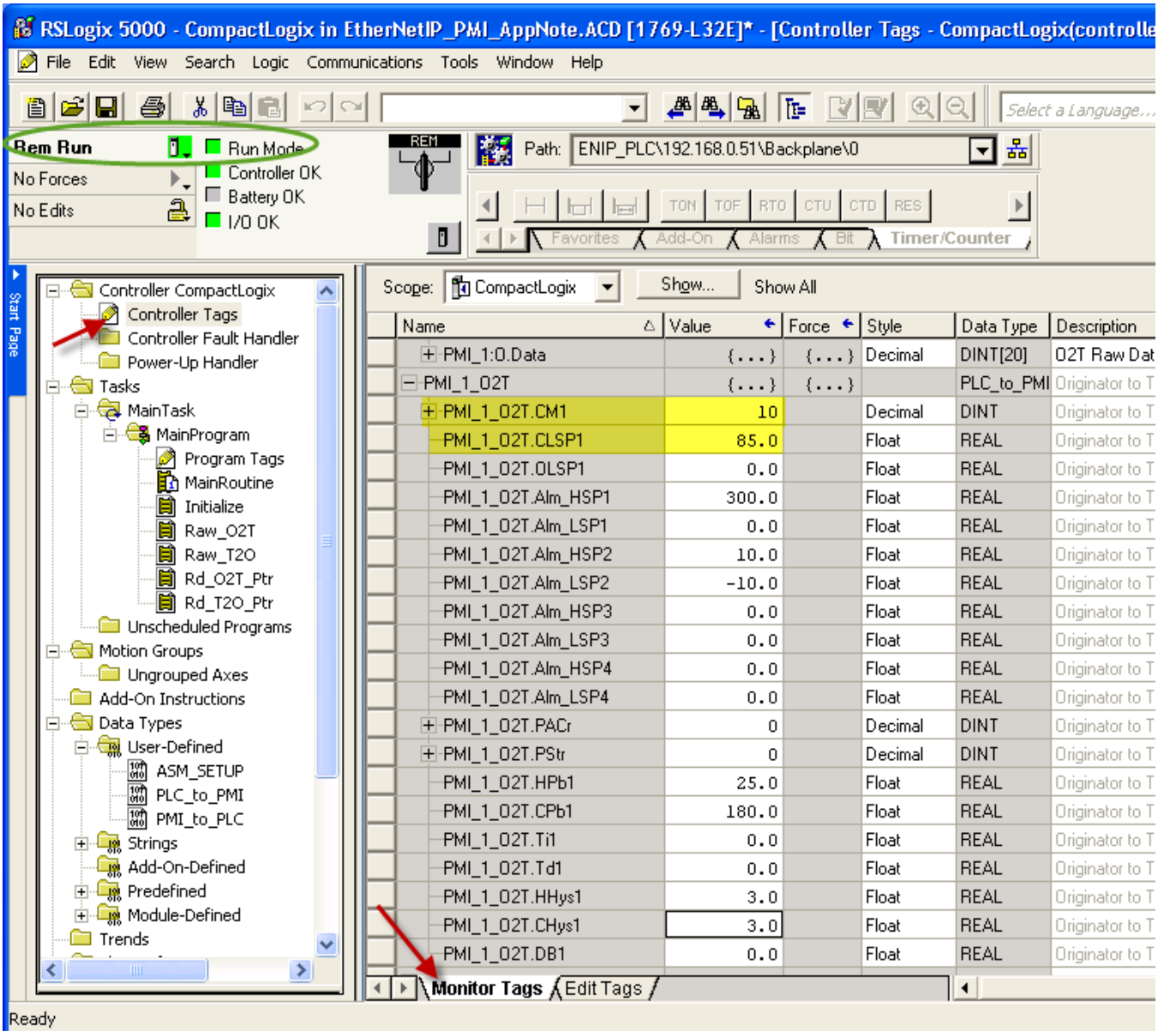
19) *Program ladder logic to copy data between user data type target and implicit assembly originator (T2O).* Add Subroutine with copy block. This instruction will copy into the user data type PMI_1:I.Data(0) to PMI_1_T2O. Be sure the Length is set to 1.



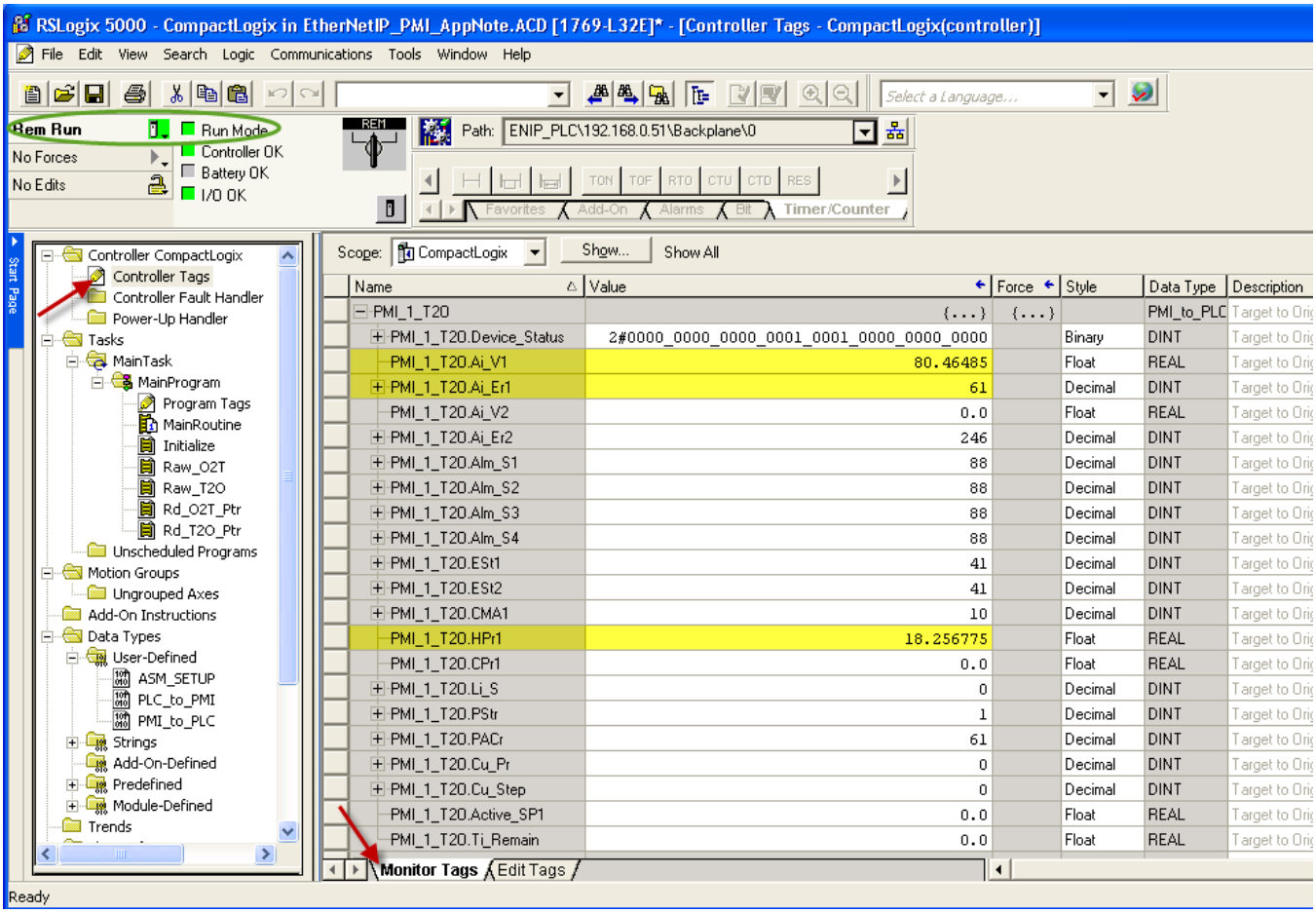
20) *Insert ladder logic in MainRoutine to call subroutines.* The subroutines will now copy between raw data format and the user defined data type. Data will appear correctly converted under Controller Tags, PMI_1_O2T and PMI_1_T2O.



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.



22) *Test PMI_1_T2O 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.



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 T2O has what is required in the first three members. Then you only set the T2O 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.

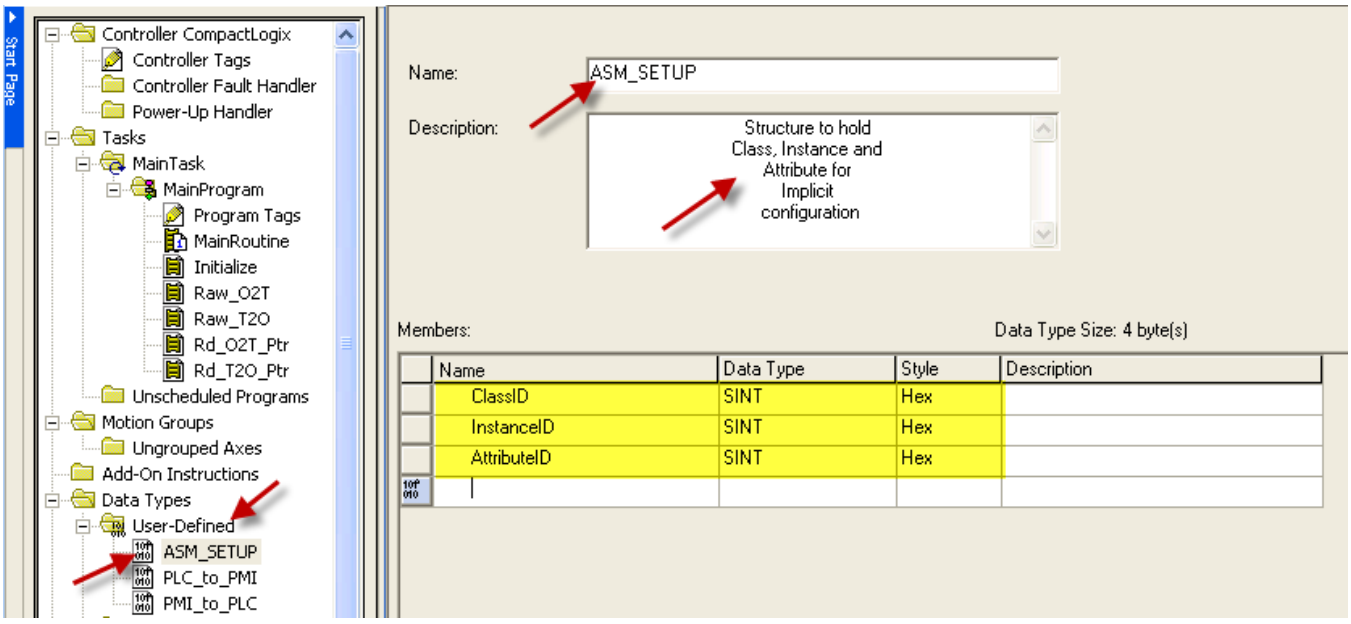
CIP Implicit Assembly Structures

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

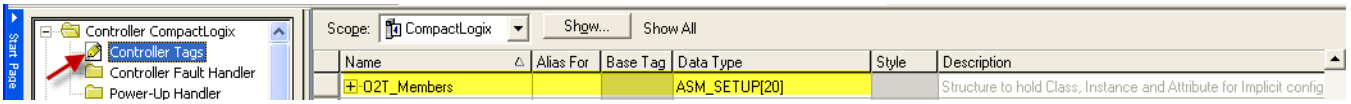
CIP Implicit Assembly Originator (Master) to Target (PM)					
Assembly Members	PM Assembly Class, Instance, Attritbute	PM Data Type	Parameter	Parameter Class, Instance, Attritbute	PLC Data Type
1	0x77, 0x01, 0x01	DINT	Loop Control Mode	0x97, 0x01, 0x01	DINT
2	0x77, 0x01, 0x02	DINT	Closed Loop Set Point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

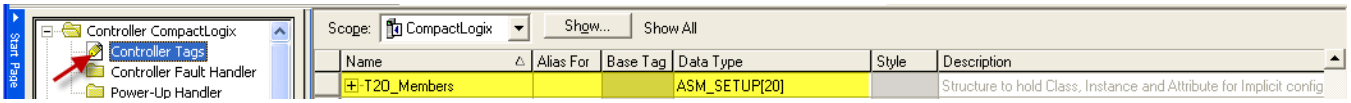
Here we see Assembly Member 3 as a register of the pointer 0x77, 1, 0x03. By default, the pointer in this location is 0x6B, 1, 0x02.

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.

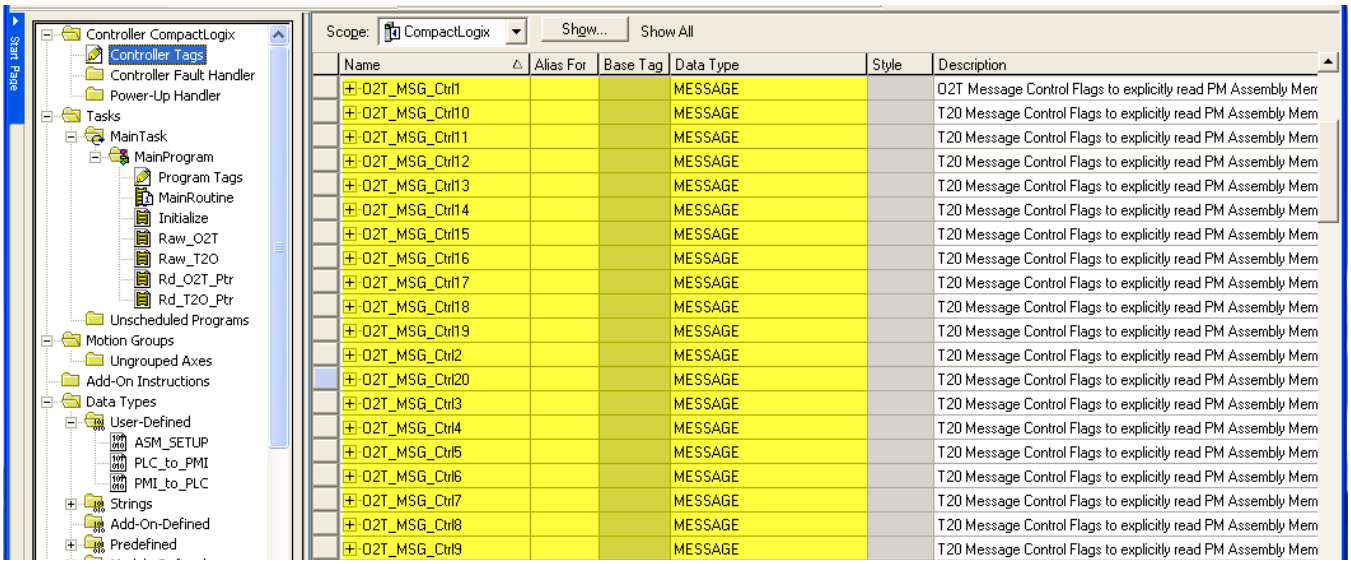


Then create Controller Tags to dimension of 20 for both O2T_Members and 20 for T2O_Members. Data Type is ASM_SETUP[20] which is the structure previously created.

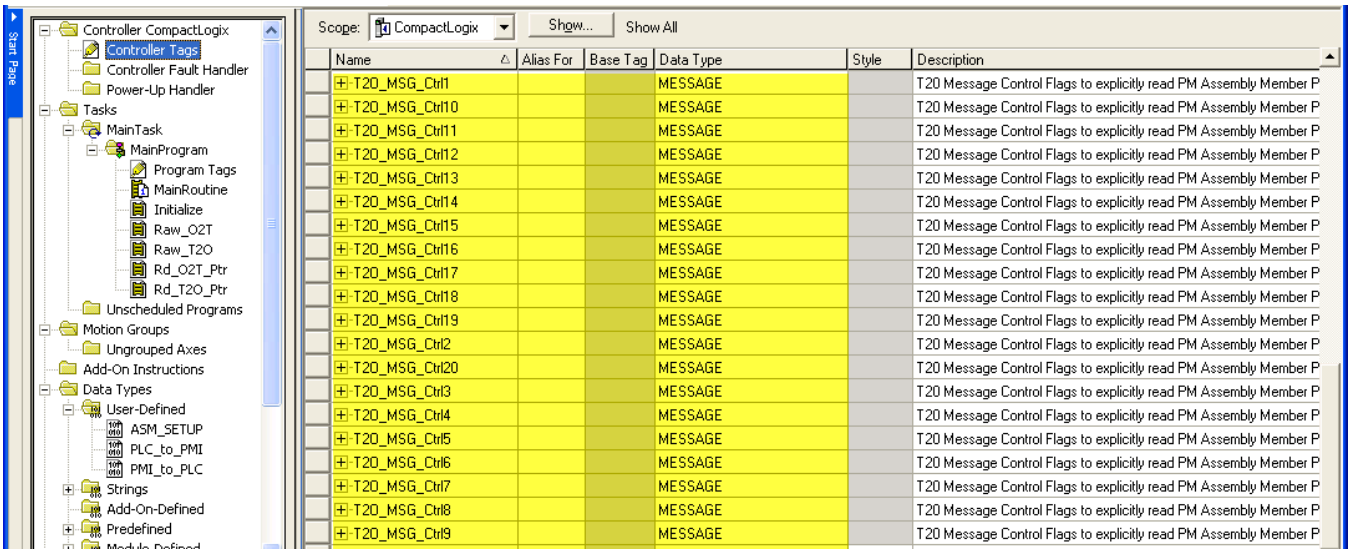




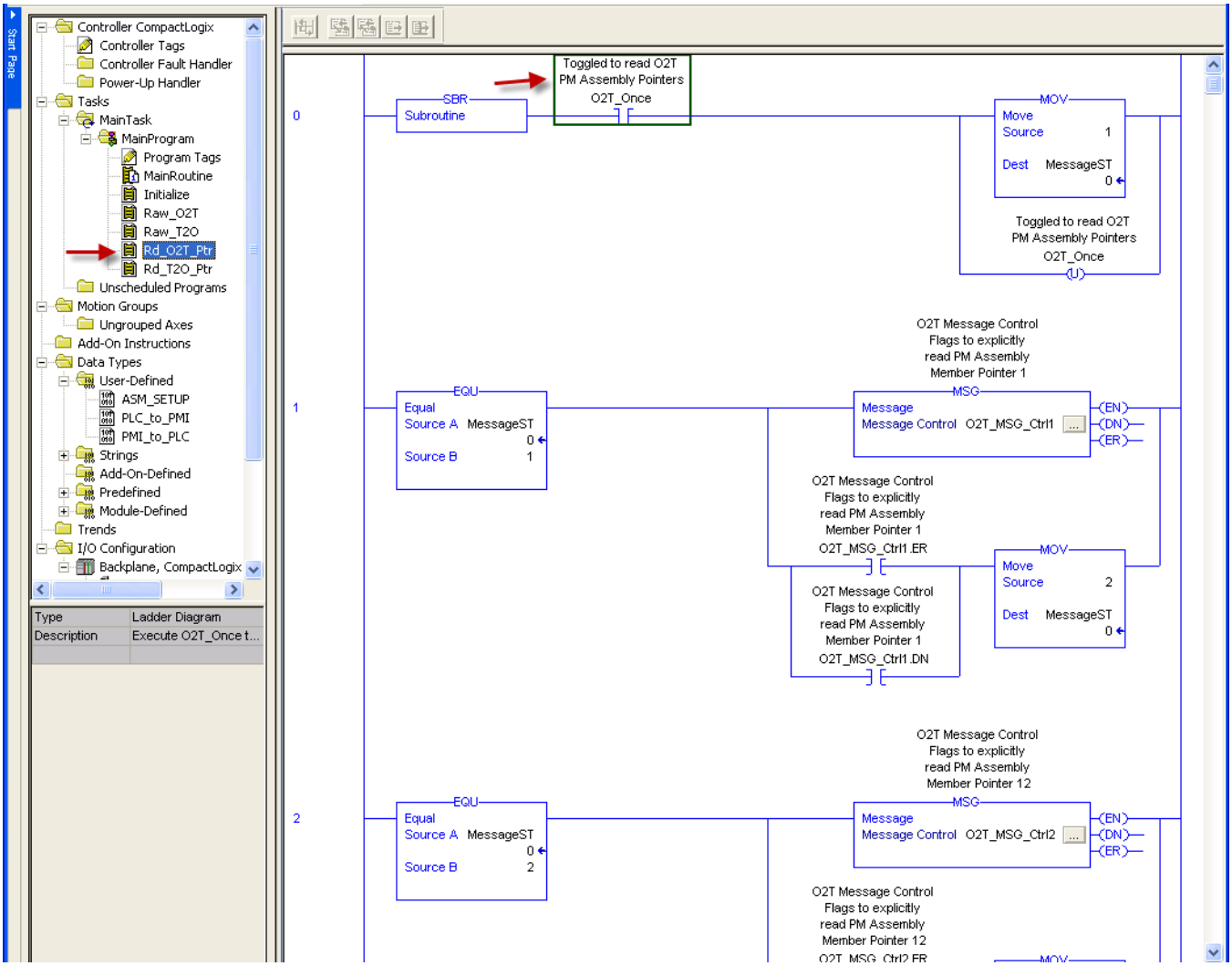
It is not an efficient way to handle the next part but for simplicity, I created 20 of O2T_MSG_Ctrl tags to handle some ladder logic for sequencing through the message instructions. Data Type is MESSAGE.



Likewise, I did the same for 20 of T2O_MSG_Ctrl tags.

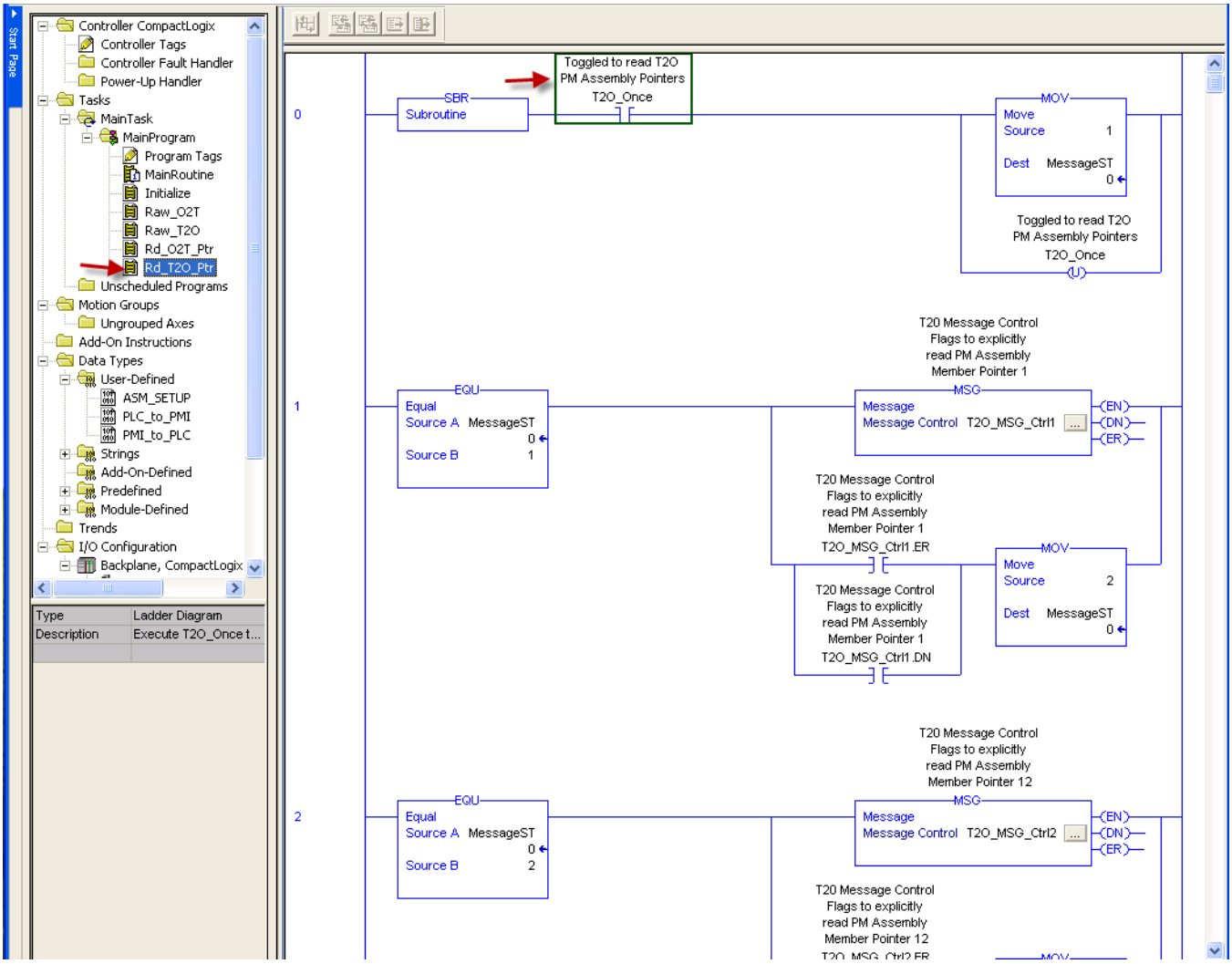


Then I created a subroutine for reading the O2T pointer assembly in ladder logic. I named it Rd_O2T_Ptr for Read O2T Pointers. There are 23 rungs; one instruction for each member. Note the contact O2T_Once; when toggled will run through the process once and transfer the assembly pointers to the ASM_SETUP structure created earlier.



EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic Example AB CompactLogix or ControlLogix PLC

Then I created a subroutine for reading T2O pointer assembly in ladder logic. I named it Rd_T20_Ptr for Read T2O 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.

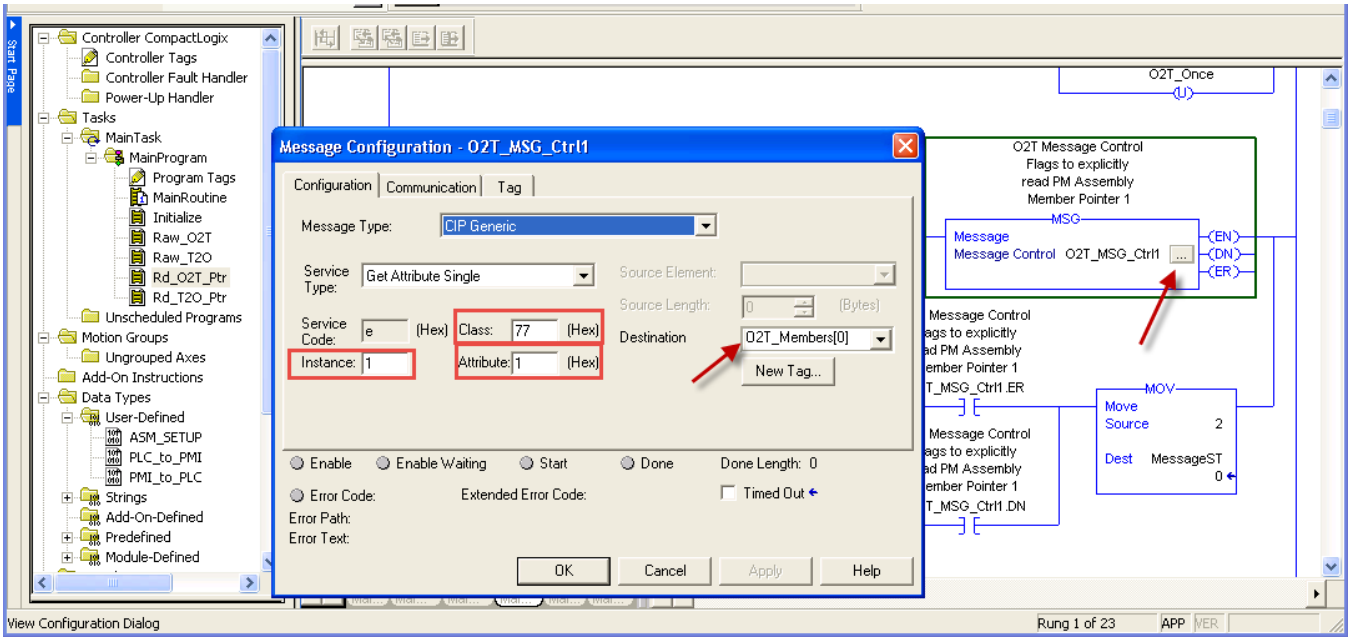


EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic Example AB CompactLogix or ControlLogix PLC

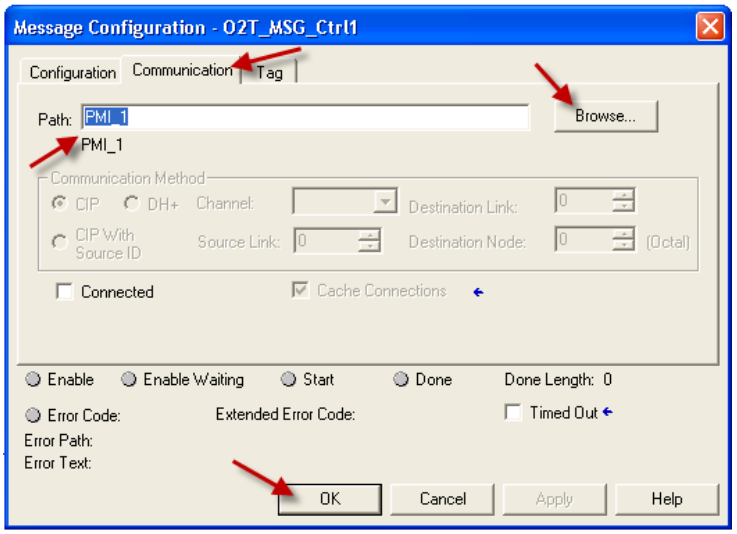
The MSG instruction for each rung defines Message Type as CIP Generic and Service Type Get Attribute Single. You define which assembly member pointer to get and transfer into O2T_Member[x]. The first message instruction is getting the first assembly member for the Originator to Target (O2T).

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

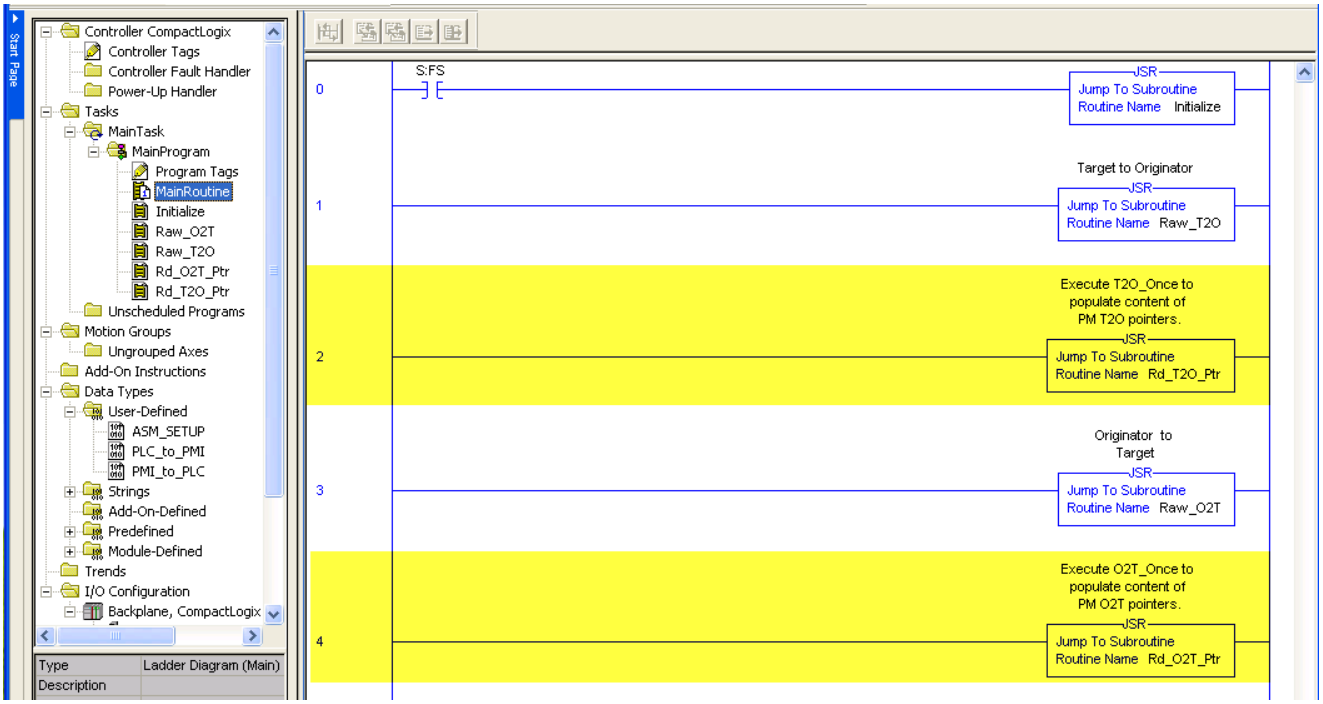


Using the Communication Tab, select the path to the controller.



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.

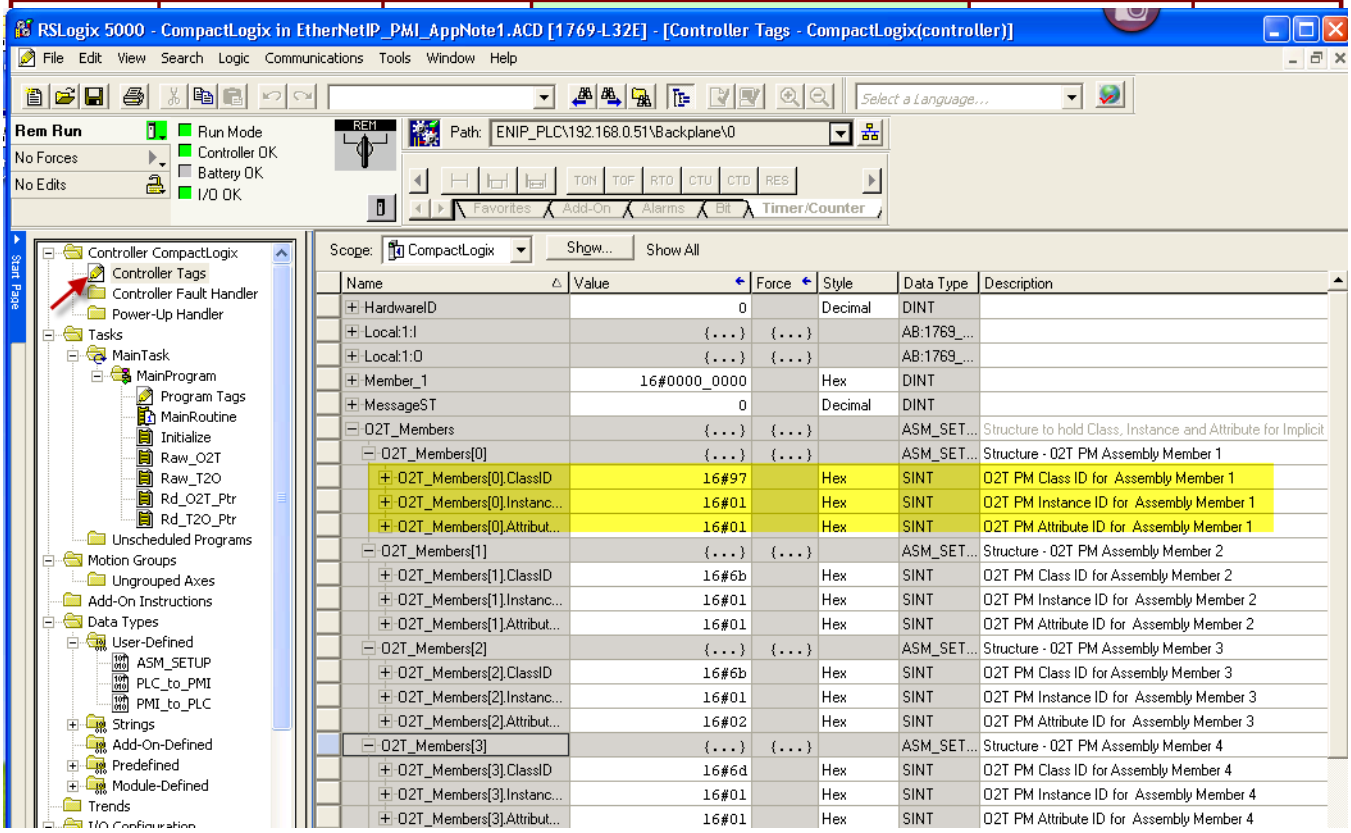


EZ-ZONE® PM & EtherNet/IP™ Configuration & Ladder Logic Example AB CompactLogix or ControlLogix PLC

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, Attritbute	PM Data Type	Parameter	Parameter Class, Instance, Attritbute	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



You can perform the same for the T2O_Once contact and then view the Controller Tags for T2O_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.