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

Table of Content

EZ-ZONE® RM & 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 Originator also called Master from/to the Target or Slave device, i.e., implicitly (real-time I/O messaging) and explicitly (information/configuration messaging). For your reference, the EZ-ZONE® RM is always the Target whereas the PLC is the Originator on the network. In this document, we will abbreviate this as O2T and T2O to signify the direction of data flow.

All EZ-ZONE RM assembly members (inputs and outputs) are 32-bits in length. In addition to the implicit members defined, the RMA controller will return one 32-bit status word in the T2O assembly.

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. The PLC will open port 44818 or 0xAF12 to make a connection.

Implicit Communications (real-time I/O messaging)

Implicit messaging is real-time I/O messaging. This method places different demands on the system and due to time critical 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). This protocol contains the data alone without requiring a response from the target device. All data that is passed implicitly is defined in the configuration or start up process. Since this method of communications contains the predefined data alone, it has low overhead and is therefore able to deliver the time-critical requirements of control. An analog input may change often and rapidly and therefore would be a good tag selection for being in the implicit table.

Explicit Communications (exception based messaging)

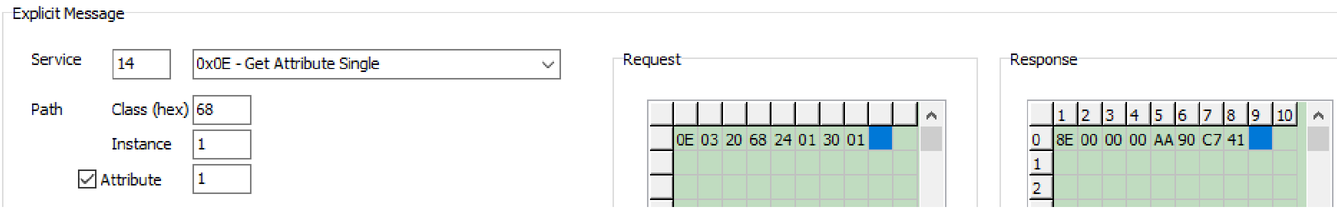
Explicit messaging is executed on demand and can vary in size. Every message must be individually configured to execute a specific Message Type, e.g., CIP Generic and a specific Service Type, e.g., Get Attribute Single or Set Attribute Single. Each device will interpret the message, act upon the task 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, configuration or non-time sensitive data. Instructing a control loop to initiate an Autotune is rarely done and therefore should be done explicitly. Some tags may need to be modified by a keypad in addition to the PLC access. The parameters are read/written on an exception based concept; in other words, read/write only when needed.

Only Get Attribute Single and Set Attribute Single are supported in the EZ-ZONE products.

An explicit message contains numbers in following order:
1) Service Code, (0x0E for Get Attribute Single or 0x10 for Set Attribute Single).
2) Quantity of bytes specified 0x03
3) Segment 0x20 for 8-bit class. See EPATH in ODVA Specification documentation.
4) Class in byte format (0x01 to 0xFF)
5) Segment 0x24 for 8-bit Instance. See EPATH in ODVA Specification documentation.
6) Instance in byte format (0x01 to 0xFF)
7) Segment 0x30 for 8-bit attribute. See EPATH in ODVA Specification documentation.
8) Attribute in byte format (0x01 to 0xFF)

The controller response is contained in the last four bytes in Little Endian format.

Class 0x68, Instance 0x01, Attribute 0x01 was requested to be read which is analog input 1 of an RMH module. The result returned 0x41C790AA = 24.94564 °C using Little Endian format.



EtherNet/IP Summary

By using both forms of communication, EtherNet/IP can prioritize time-critical I/O communications (implicit) over non-critical messages (explicit) while allowing for both to occur simultaneously. Watlow EtherNet/IP equipped devices supports both Implicit and Explicit communications. Users should determine the tags that will be transferred and segregate the tags between implicit and explicit requirements.

Tags that are explicit may be changed from two locations depending on logic. If the same tag were part of the implicit assembly, the value changed via keypad would be over written ever several hundred milliseconds by the PLC. Some users are surprised that a value keeps resetting after changing from the keypad yet no longer exhibits that action when the Ethernet cable is disconnected.

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

EZ-ZONE RMA Legacy Gateway Theory of Operation

The RMA module is a specific EZ-ZONE RM module type that connects the attached combination of RMC, RMH, RML, RMS, and RME module types to a PLC network and acts as a translation device. In this document, we are discussing the RMAx-A3xx-AAxx model number utilizing the EtherNet/IP protocol. Another document describes the same module using the Modbus TCP protocol therefore will not be discussed here. The character x in the shown model number are placeholders for optional digits.

To commission Ethernet with the RMA the user must configure the Ethernet port settings. The following parameters are available to be configured.

Modbus Word Order – LowHigh *or* HighLow
IP Address Mode – Fixed *or* DHCP
IP Fixed Address Part 1 to 4 – 0 *to* 255
IP Fixed Subnet Part 1 to 4 – 0 *to* 255
Fixed IP Gateway Part 1 to 4 – 0 *to* 255
Modbus TCP Enable – Yes *or* No
EtherNet/IP Enable – Yes *or* No
Display Units – F *or* C
Non-Volatile Save – Yes *or* No

Watlow suggest setting IP Address Mode to Fixed, IP Fixed Address and Subnet to match PLC logical network, Modbus TCP Enable = No, EtherNet/IP Enable = Yes, Display Units to desired and Non-Volatile = Yes. Power must be removed and restored to RMA after IP address detail is configured before values are utilized.

Not counting the RMA module, an EZ-ZONE RM system may consist of 1 to 16 modules in any combination. Watlow suggest the address listed on each module be set via the orange button for a unique address starting at 1 up to the maximum modules present. The RMA module should remain with the default address 17 shown as the letter 'J'. The other modules will include numbers 1 to 9, then A, b, C, d, E, F, and H. There can only be one RMA module to translate the combination of modules connected in the system.

Each addressed module 1 to 16 is represented by a virtual gateway internal to the RMA. The user programs the Local Remote Gateways for each of the modules present on the network. Example: Each Local Remote Gateway associates to a module 1 to 1, 2 to 2, 3 to 3 etc. Each parameter listed in the Local Remote Gateway determines the behavior of data relevant to the associated module.

The following are the parameters available to set for each Local Remote Gateway instance.
Device Enable – Yes *or* No
Device Status – On *or* Off
Modbus Address Offset – 0 *to* 65,535
CIP Instance Offset – 0 *to* 255
CIP Implicit Assembly Output Member Quantity – 0 *to* 40
CIP Implicit Assembly Input Member Quantity – 0 *to* 40

The device status changes to 'Yes' for each device enabled Local Remote Gateway instance provided the associated module is connected to the RMA. The device status word received in the PLC changes value based on those that display 'Yes' indicating it presence in the system. See RMA Device Status defined later in this document.

As a device gateway, the RMA performs three functions.

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

1) Coordinates a connection and display status LEDs of the connection. The PLC will request a connection to the module at the selected IP address when power is first applied. The connection establishes the quantity of data to be exchanged implicitly in each direction.
2) Routes explicit messages to correct module and return response to PLC. CIP Instance Offset is used to route explicit messages from the PLC to the appropriate module. Specifying an offset provides control over which message instruction from the PLC affects a module on the RM rack. See Explicit MSG Instruction Data Flow diagram later in this document.
3) Performs implicit data transfers. For each Local Remote Gateway enabled the RMA module will sequentially gather the CIP Implicit Assembly Output Member Quantity specified from the other modules, add a device status word at top of list and forward to the PLC's input table referenced as T2O. Note that the T2O is always specified as one more member than selected in the RMA due to the additional status word. There is one status word per system so is not included in the Local Remote Gateway settings.

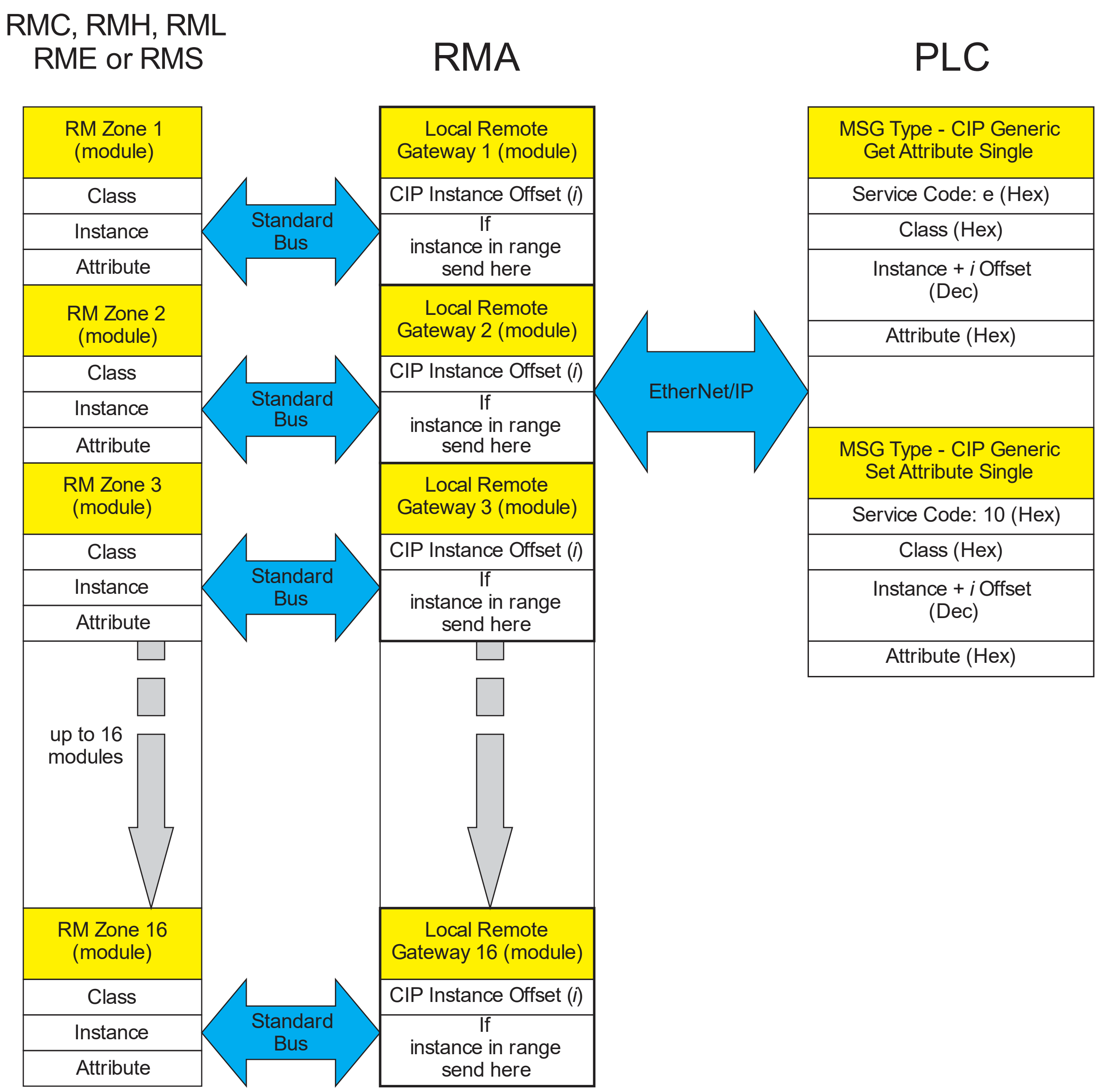
   The PLC sends its implicit output data table to the input table of the RMA. These go into CIP Implicit Assembly Input Member Quantity table referenced as O2T.

   The rate of data exchange occurs based on the PLC requested packet interval setting (RPI). If a Local Remote Gateway is disabled, that module is ignored. As an example, the application may have 5 modules but only modules 1 and 3 have data of interest to the PLC. In this situation Device Enabled is set for Local Remote Gateway 1 and 3. The RMA gets the quantity of implicit outputs, add a device status word and returns the input table of the PLC. See Implicit Data Flow diagrams later in this document.

Data table only contains data and no identifiers of the data. It is imperative that the quantity of inputs/outputs expected to be sent/received be of the correct size and known during the setup process. Failure to match both originator and target results in connection error messages on PLC or incorrect data being identified. This is the number one mistake new users make on commissioning RM products with PLCs.

Use message type CIP Generic Get Attribute Single to read tags and Set Attribute Single to write tags. Use CIP Instance Offset and add to instance read/write instruction to route to appropriate RM module. Use CIP Instance Offset in each Local Remote Gateway setting to determine which PLC instance + offset will apply.

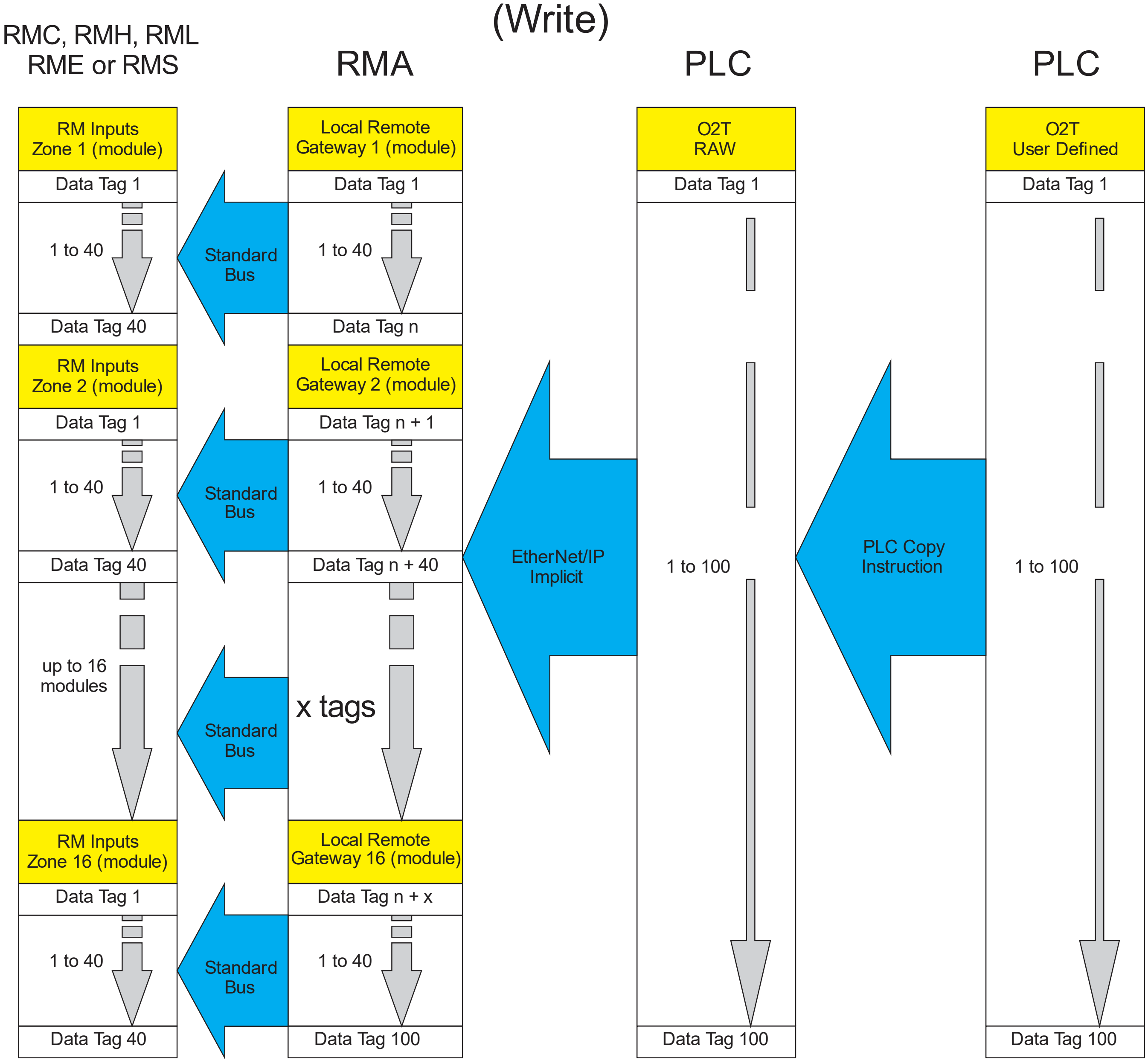
Explicit MSG Instruction Data Flow



i = CIP Instance Offset

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

Implicit O2T Data Flow (Write)



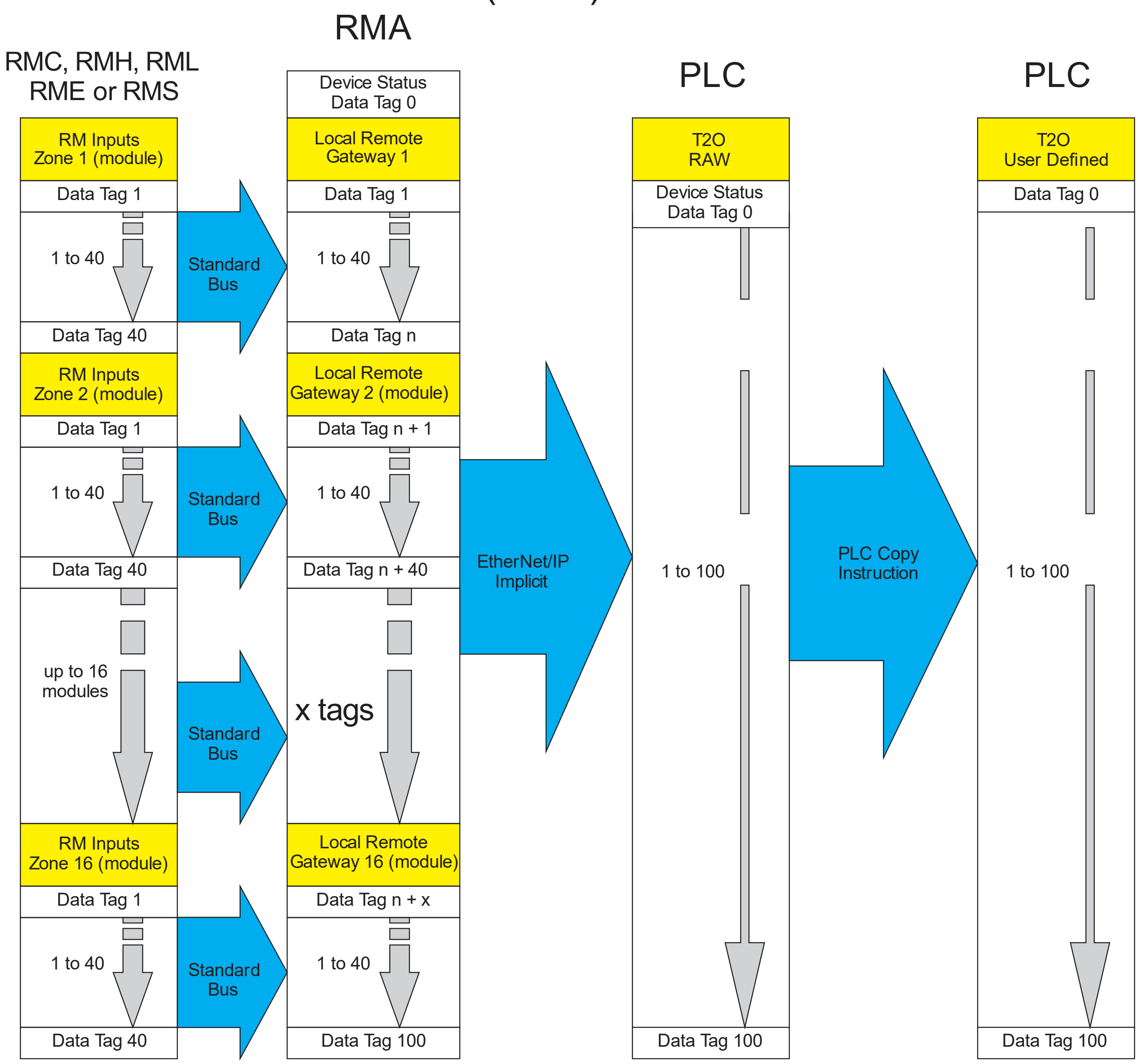
If enabled, 1 to 40 tags for each RM module, max of 100 members

n = number in range of 1 to 40
x = number of previous tags

O2T = Originator (PLC) to Target (RM)

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

Implicit
T2O Data Flow
(Read)



If enabled, 1 to 40 tags
for each RM module,
max of 100 members

n = number in range of 1 to 40
x = number of previous tags

T2O = Target (RM) to Originator (PLC)

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

Hardware Requirements

Connection from the PLC to the RMA module is via EtherNet/IP. It is highly recommended that the path from the PLC to the RMA is through a managed Ethernet Switch supporting IGMP Snooping. EtherNet/IP typically contains significant broadcast traffic and this feature will segregate the broadcast within groups. If too much broadcast traffic is processed by the RMA, resources may be overwhelmed and communication may be disrupted. The RMA supports half-duplex traffic at 10/100MHZ which is auto negotiated. Should the switch fail to properly negotiate the duplex or speed correctly, you may need to configure the switch to fixed values. Another advantage of a managed switch is the ability to capture data on a mirrored port. Wireshark (a free program available on Internet) may be used to capture the traffic between the PLC and RM to a PC via the mirrored port for diagnostics when needed.

Configuration of the attached RM modules and RMA for IP Addressing is only accomplished via the EIA-485 serial connection on slot C of any one module. Therefore, you will need a USB to EIA-485 converter so the PC may configure parameters to match your application. A suggested converter is Watlow part number 0847-0326-0000. Set the latency timer to 1mS in the PC under device manager. See RMA user's guide for detail on EIA-485 wiring. Then use EZ-ZONE Configurator available from the Watlow web site for system configuration. Configuration of the parameters are not available via Ethernet. Settings changes to IP addressing require a power cycle of the RMA to take effect.

Getting Started

Before real data can be obtained in the PLC data tables, the RM modules must be correctly configured and wired. Valid sensors must be attached to read sensor data. The PLC is only aware of the one RM system and does not have information programmed about individual modules. The RMA is sending a combined view of the RM system to the PLC. See previous diagrams for how implicit data is combined into one large table in the RMA and passed to the PLC.

This document will not cover basic configuration of the EZ-ZONE RM. That information is covered in each of the appropriate User's Guides which can be found on the Watlow website - http://www.watlow.com/literature/manuals.cfm

Understanding the Application Requirements

- Identify the desired tags (data points) that need to be exchanged between the PLC and the RM system. It is beneficial to create a written table in Excel or similar detailing the I/O tags for each module. Once all tables are created, one master table for inputs and one for outputs will be created by the combined individual tables. Tables will be created for RM data inputs (O2T) separately from RMA data outputs (T2O).
- Will there be a need to infrequently read or write parameters between the originator and target? Explicit communications can be executed with minimal effort to accomplish this task using a message instruction. Use message instructions (explicit messages) for non-time sensitive parameters and parameters that require change on exception only.
- When using implicit communications determine what data (EZ-ZONE parameters) will be transferred implicitly (inputs and outputs) between the originator and target ensuring that the maximum number of 40 inputs/40 outputs per module with a total of 100 inputs/100 outputs per RMA are not exceeded. These numbers do not include the device status word.
- Compare your requirements of implicit data to the default assembly in the product. Change the implicit table to include the desired tags in the desired order. Do not include O2T tags for unused parameters as the PLC will send uninitialized values to the RM system.

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

Sequence of Tasks

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 RM input and output assembly for each module connected to the RMA. This is then programmed into the controller. The data can be represented in degrees Fahrenheit or Celsius independent of the Remote User Interface (RUI) display when using the controller for temperature. When you need to change the default Implicit Assemblies, you are required to use an Explicit message instruction from the PLC or a utility provided by Watlow. To establish explicit communications between Originator and Target devices, configuration steps need to be executed within the PLC using RSLogix5000 as well as within the EZ-ZONE RM using EZ-ZONE Configurator software connected to the EIA-485 port of the controller. After the configuration requirements have been met, PLC programming examples will follow.

First let's review the sequence of tasks to be accomplished in the EZ-ZONE RM controller. The keypad method using a Remote User Interface (RUI) will not be detailed in this application note.

1) Connect all modules side-to-side on DIN rail. Place RMA module last or first on assembled rack.
2) Wire analog inputs and outputs plus additional I/O if used to RM system.
3) Apply 24V to RM system.
4) Set unique address to each module starting at 1 and increment towards 16. Keep RMA at address 17 or letter 'J'.
5) Connect a PC to the EZ-ZONE RM on terminals -CD, +CE and CF of slot C using an USB to EIA-485 serial port. If there is more than one RM module, use any set of the slot C terminals as these are all connected via a backplane. One excellent choice for a serial converter is a B&B Electronics 485USBTB-2W sold under Watlow part number 0847-0326-0000 for PC connection.
6) Install appropriate Windows driver included with converter or available from vendor's web site.
7) Check latency timer in serial driver of PC for 1mS setting. Not all serial 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.
8) Install EZ-ZONE Configurator software on to PC. The program is located on the Watlow website under Download Center, Software and Demos category.
9) Execute EZ-ZONE Configurator software, choose 'Configure a device…', click Next.
10) Select communications port assigned to converter. Click Next.
11) Select RMA model number form list. Double click or highlight and select Next.
12) Select and configure Communications 2 settings.
13) Select and configure Local Remote Gateway settings.
14) Configure any additional settings to match the application such as data logging, real time clock…
15) Select Finish and save file if desired.
16) Start configurator again and repeat process for other modules programming all behavior of sensors, control loops, logic and alarms as desired.
17) Select Finish and save files each time for all modules.
18) If no additional settings are required, close configurator and remove power from RM system.
19) Connect Ethernet cable between managed Ethernet switch with IGMP snooping enabled and RMA.
20) Apply 24V to RM system.
21) You are now ready to configure implicit assemblies in PLC and RMA.
22) When communication is no longer needed using the EIA-485 connection, disconnect wires CD, CE and CF. Do not leave an unpowered converter connected even for future use. If a connection is desired outside panel, use a converter that is always powered rather than port powered. Failure to follow this recommendation may result in communication errors on Ethernet.

Define Tags for Implicit Assembly

Create a document of listed tags to be transferred from the O2T for each RM module in the system. Create a second document of listed tags to be transferred from the T2O for each RM module in the system. Using an Excel spreadsheet is an ideal method to capture these requirements. Tip: Use tag names that identify the module and instance of the desired tag. As an example; the PLC should send the control mode for control loop of an RMC module. If there are four control loops, then record tag 1 as control mode, module 1, instance 1 and so on. Follow with the next tags to be included. Group all tags for each module together. You cannot enter all control modes for all modules followed by all set points for all modules.

Below is an example of three RMC modules (addresses) each having specific tags.

O2T Tags

Tag#	Addr.	Inst.	Tag Name	Data Type	Parameter Name	Description	Parameter ID	Inst.
1	1	1	CM_M1_I1	UINT	Control Mode Loop 1	Change loop mode	8001	1
2	1	2	CM_M1_I2	UINT	Control Mode Loop 2	Change loop mode	8001	2
3	1	3	CM_M1_I3	UINT	Control Mode Loop 3	Change loop mode	8001	3
4	1	4	CM_M1_I4	UINT	Control Mode Loop 4	Change loop mode	8001	4
5	1	1	SP_M1_I1	REAL	Control-Loop Set Point 1	Change set point	7001	1
6	1	2	SP_M1_I2	REAL	Control-Loop Set Point 2	Change set point	7001	2
7	1	3	SP_M1_I3	REAL	Control-Loop Set Point 3	Change set point	7001	3
8	1	4	SP_M1_I4	REAL	Control-Loop Set Point 4	Change set point	7001	4
9	2	1	CM_M2_I1	UINT	Control Mode Loop 1	Change loop mode	8001	1
10	2	2	CM_M2_I2	UINT	Control Mode Loop 2	Change loop mode	8001	2
11	2	3	CM_M2_I3	UINT	Control Mode Loop 3	Change loop mode	8001	3
12	2	4	CM_M2_I4	UINT	Control Mode Loop 4	Change loop mode	8001	4
13	2	1	SP_M2_I1	REAL	Control-Loop Set Point 1	Change set point	7001	1
14	2	2	SP_M2_I2	REAL	Control-Loop Set Point 2	Change set point	7001	2
15	2	3	SP_M2_I3	REAL	Control-Loop Set Point 3	Change set point	7001	3
16	2	4	SP_M2_I4	REAL	Control-Loop Set Point 4	Change set point	7001	4
17	3	1	CM_M3_I1	UINT	Control Mode Loop 1	Change loop mode	8001	1
18	3	2	CM_M3_I2	UINT	Control Mode Loop 2	Change loop mode	8001	2
19	3	3	CM_M3_I3	UINT	Control Mode Loop 3	Change loop mode	8001	3
20	3	4	CM_M3_I4	UINT	Control Mode Loop 4	Change loop mode	8001	4
21	3	1	SP_M3_I1	REAL	Control-Loop Set Point 1	Change set point	7001	1
22	3	2	SP_M3_I2	REAL	Control-Loop Set Point 2	Change set point	7001	2
23	3	3	SP_M3_I3	REAL	Control-Loop Set Point 3	Change set point	7001	3
24	3	4	SP_M3_I4	REAL	Control-Loop Set Point 4	Change set point	7001	4

Notice in this example there is a total of 24 tags in the system for O2T. This number will be used later in programming the PLC's output size. Write down this quantity for later programming.

There are 8 tags for each module or address utilized. This number will be programmed into the Local Remote Gateway for each instance under CIP Implicit Assembly Input Member Quantity. You program this setting in the RMA setup page. There is one parameter for each Local Remote Gateway 1 to 16. Write down the quantity for each module having desired tags for later programming.

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

In this example, PV_M1_I1 are abbreviations for PV = Process Value, M = Module Address, and I = Instance. That same concept was applied to AE = Analog Input Error and HP = Heat Power.

T2O Tags

Tag#	Addr.	Inst.	Tag Name	Data Type	Parameter Name	Description	Parameter ID	Inst.
1	1	1	PV_M1_I1	REAL	Filtered Process Value 1	Read control loop process value	8031	1
2	1	2	PV_M1_I2	REAL	Filtered Process Value 2	Read control loop process value	8031	2
3	1	3	PV_M1_I3	REAL	Filtered Process Value 3	Read control loop process value	8031	3
4	1	4	PV_M1_I4	REAL	Filtered Process Value 4	Read control loop process value	8031	4
5	1	1	AE_M1_I1	UINT	Analog Input Error 1	Read analog input error status	4002	1
6	1	2	AE_M1_I2	UINT	Analog Input Error 2	Read analog input error status	4002	2
7	1	3	AE_M1_I3	UINT	Analog Input Error 3	Read analog input error status	4002	3
8	1	4	AE_M1_I4	UINT	Analog Input Error 4	Read analog input error status	4002	4
9	1	1	HP_M1_I1	REAL	Heat Power Loop 1	Read PID requested heat power	8011	1
10	1	2	HP_M1_I2	REAL	Heat Power Loop 2	Read PID requested heat power	8011	2
11	1	3	HP_M1_I3	REAL	Heat Power Loop 3	Read PID requested heat power	8011	3
12	1	4	HP_M1_I4	REAL	Heat Power Loop 4	Read PID requested heat power	8011	4
13	2	1	PV_M2_I1	UINT	Filtered Process Value 1	Read control loop process value	8031	1
14	2	2	PV_M2_I2	UINT	Filtered Process Value 2	Read control loop process value	8031	2
15	2	3	PV_M2_I3	UINT	Filtered Process Value 3	Read control loop process value	8031	3
16	2	4	PV_M2_I4	UINT	Filtered Process Value 4	Read control loop process value	8031	4
17	2	1	AE_M2_I1	UINT	Analog Input Error 1	Read analog input error status	4002	1
18	2	2	AE_M2_I2	UINT	Analog Input Error 2	Read analog input error status	4002	2
19	2	3	AE_M2_I3	UINT	Analog Input Error 3	Read analog input error status	4002	3
20	2	4	AE_M2_I4	UINT	Analog Input Error 4	Read analog input error status	4002	4
21	2	1	HP_M2_I1	REAL	Heat Power Loop 1	Read PID requested heat power	8011	1
22	2	2	HP_M2_I2	REAL	Heat Power Loop 2	Read PID requested heat power	8011	2
23	2	3	HP_M2_I3	REAL	Heat Power Loop 3	Read PID requested heat power	8011	3
24	2	4	HP_M2_I4	REAL	Heat Power Loop 4	Read PID requested heat power	8011	4
25	3	1	PV_M3_I1	REAL	Filtered Process Value 1	Read control loop process value	8031	1
26	3	2	PV_M3_I2	REAL	Filtered Process Value 2	Read control loop process value	8031	2
27	3	3	PV_M3_I3	REAL	Filtered Process Value 3	Read control loop process value	8031	3
28	3	4	PV_M3_I4	REAL	Filtered Process Value 4	Read control loop process value	8031	4
29	3	1	AE_M3_I1	UINT	Analog Input Error 1	Read analog input error status	4002	1
30	3	2	AE_M3_I2	UINT	Analog Input Error 2	Read analog input error status	4002	2
31	3	3	AE_M3_I3	UINT	Analog Input Error 3	Read analog input error status	4002	3
32	3	4	AE_M3_I4	UINT	Analog Input Error 4	Read analog input error status	4002	4
33	3	1	HP_M3_I1	REAL	Heat Power Loop 1	Read PID requested heat power	8011	1
34	3	2	HP_M3_I2	REAL	Heat Power Loop 2	Read PID requested heat power	8011	2
35	3	3	HP_M3_I3	REAL	Heat Power Loop 3	Read PID requested heat power	8011	3
36	3	4	HP_M3_I4	REAL	Heat Power Loop 4	Read PID requested heat power	8011	4

Notice in this example there is a total of 36 tags in the system for T2O. The RMA will add a device status word to make 37 tags total sent to the PLC. This number will be used later in programming the PLC's input size. Write down this quantity for later programming.

We have three module each having 12 tags defined. There is no requirement that the O2T or T2O assembly for each module have the same quantity of tags. This number will be programmed into the Local Remote Gateway for each instance under CIP Implicit Assembly Output Member Quantity. You program this setting in the RMA setup page. There is one parameter in each Local Remote Gateway 1 to 16. We are only enabling and entering Local Remote Gateways 1 to 3 in this example. Disable remaining Local Remote Gateways as these are not used. Write down these quantities for later programming.

*Both tables above are just examples of the definition process. Do not assume that these tags will be populated, the listed tags apply nor these are the correct quantity in your application. It is your job to define the desired tags for the hardware on hand and the application. Next program the implicit tags as desired, tally the quantity used and program the RMA and PLC to match your choices.

You may wish to obtain the program Assembly Programming Application from the Watlow web site https://www.watlow.com/-/media/documents/software-and-demos/ezzone-assembly-programmer.ashx . The purpose of the program is to program the implicit tags within each RM module via standard bus. Standard bus communication is over the EIA-485 connection on the RM. Alternatively you can program the implicit tags in the RM using explicit message instructions via EtherNet/IP. The Assembly Programmer Application download contains a spreadsheet that may be used to document your implicit settings. It includes a readme file called *Setting the Implicit Table using Assembly Programmer Application.pdf* that has additional instructions on setting the implicit tags of the RM modules. Note that the O2T is referenced as instance 1 while T2O is referenced as instance 2. The Excel sheet might look something like this –

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

	A	B	C	D	E	F	G	H	I	J	K
1		Originator [PLC] to Target [EZ-ZONE] - Instance 1									
2		Pointers of Data AB PLC Output Assembly = 100, Set Ai.nb of RUI/Gateway								Value Referenced by Pointer	
3	Assembly Row (element)	Parameter ID (contains table pointer)	Parameter ID Write Value (data pointer)	Parameter Instance Wrtie Value (data pointer)	CIP - Explicit write Class, Inst, Attritbute (table pointer)	CIP - Write Class, Inst, Attritbute (data pointer)	Parameter Name and Function (description)	Data Type (pointer)		Parameter ID (contains value)	Data Type (data value)
4	1	19001	1001	255	0x77, 0x01, 0x01	0x65, 0xFF, 0x01	not assigned	DINT		20001	
5	2	19002	1001	255	0x77, 0x01, 0x02	0x65, 0xFF, 0x01	not assigned	DINT		20002	
6	3	19003	1001	255	0x77, 0x01, 0x03	0x65, 0xFF, 0x01	not assigned	DINT		20003	
7	4	19004	1001	255	0x77, 0x01, 0x04	0x65, 0xFF, 0x01	not assigned	DINT		20004	
8	5	19005	1001	255	0x77, 0x01, 0x05	0x65, 0xFF, 0x01	not assigned	DINT		20005	
9	6	19006	1001	255	0x77, 0x01, 0x06	0x65, 0xFF, 0x01	not assigned	DINT		20006	
10	7	19007	1001	255	0x77, 0x01, 0x07	0x65, 0xFF, 0x01	not assigned	DINT		20007	
11	8	19008	1001	255	0x77, 0x01, 0x08	0x65, 0xFF, 0x01	not assigned	DINT		20008	
12	9	19009	1001	255	0x77, 0x01, 0x09	0x65, 0xFF, 0x01	not assigned	DINT		20009	
13	10	19010	1001	255	0x77, 0x01, 0x0A	0x65, 0xFF, 0x01	not assigned	DINT		20010	
14	11	19011	1001	255	0x77, 0x01, 0x0B	0x65, 0xFF, 0x01	not assigned	DINT		20011	
15	12	19012	1001	255	0x77, 0x01, 0x0C	0x65, 0xFF, 0x01	not assigned	DINT		20012	
16	13	19013	1001	255	0x77, 0x01, 0x0D	0x65, 0xFF, 0x01	not assigned	DINT		20013	
17	14	19014	1001	255	0x77, 0x01, 0x0E	0x65, 0xFF, 0x01	not assigned	DINT		20014	
18	15	19015	1001	255	0x77, 0x01, 0x0F	0x65, 0xFF, 0x01	not assigned	DINT		20015	
19	16	19016	1001	255	0x77, 0x01, 0x10	0x65, 0xFF, 0x01	not assigned	DINT		20016	
20	17	19017	1001	255	0x77, 0x01, 0x11	0x65, 0xFF, 0x01	not assigned	DINT		20017	
21	18	19018	1001	255	0x77, 0x01, 0x12	0x65, 0xFF, 0x01	not assigned	DINT		20018	
22	19	19019	1001	255	0x77, 0x01, 0x13	0x65, 0xFF, 0x01	not assigned	DINT		20019	
23	20	19020	1001	255	0x77, 0x01, 0x14	0x65, 0xFF, 0x01	not assigned	DINT		20020	
24	21	19020	1001	255	0x77, 0x01, 0x15	0x65, 0xFF, 0x01	not assigned	DINT		20021	
25	22	19020	1001	255	0x77, 0x01, 0x16	0x65, 0xFF, 0x01	not assigned	DINT		20022	
26	23	19020	1001	255	0x77, 0x01, 0x17	0x65, 0xFF, 0x01	not assigned	DINT		20023	
27	24	19020	1001	255	0x77, 0x01, 0x18	0x65, 0xFF, 0x01	not assigned	DINT		20024	
28	25	19020	1001	255	0x77, 0x01, 0x19	0x65, 0xFF, 0x01	not assigned	DINT		20025	
29	26	19020	1001	255	0x77, 0x01, 0x1A	0x65, 0xFF, 0x01	not assigned	DINT		20026	
30	27	19020	1001	255	0x77, 0x01, 0x1B	0x65, 0xFF, 0x01	not assigned	DINT		20027	
31	28	19020	1001	255	0x77, 0x01, 0x1C	0x65, 0xFF, 0x01	not assigned	DINT		20028	
32	29	19020	1001	255	0x77, 0x01, 0x1D	0x65, 0xFF, 0x01	not assigned	DINT		20029	
33	30	19020	1001	255	0x77, 0x01, 0x1E	0x65, 0xFF, 0x01	not assigned	DINT		20030	
34	31	19020	1001	255	0x77, 0x01, 0x1F	0x65, 0xFF, 0x01	not assigned	DINT		20031	
35	32	19020	1001	255	0x77, 0x01, 0x20	0x65, 0xFF, 0x01	not assigned	DINT		20032	
36	33	19020	1001	255	0x77, 0x01, 0x21	0x65, 0xFF, 0x01	not assigned	DINT		20033	
37	34	19020	1001	255	0x77, 0x01, 0x22	0x65, 0xFF, 0x01	not assigned	DINT		20034	
38	35	19020	1001	255	0x77, 0x01, 0x23	0x65, 0xFF, 0x01	not assigned	DINT		20035	
39	36	19020	1001	255	0x77, 0x01, 0x24	0x65, 0xFF, 0x01	not assigned	DINT		20036	
40	37	19020	1001	255	0x77, 0x01, 0x25	0x65, 0xFF, 0x01	not assigned	DINT		20037	
41	38	19020	1001	255	0x77, 0x01, 0x26	0x65, 0xFF, 0x01	not assigned	DINT		20038	
42	39	19020	1001	255	0x77, 0x01, 0x27	0x65, 0xFF, 0x01	not assigned	DINT		20039	
43	40	19020	1001	255	0x77, 0x01, 0x28	0x65, 0xFF, 0x01	not assigned	DINT		20040	

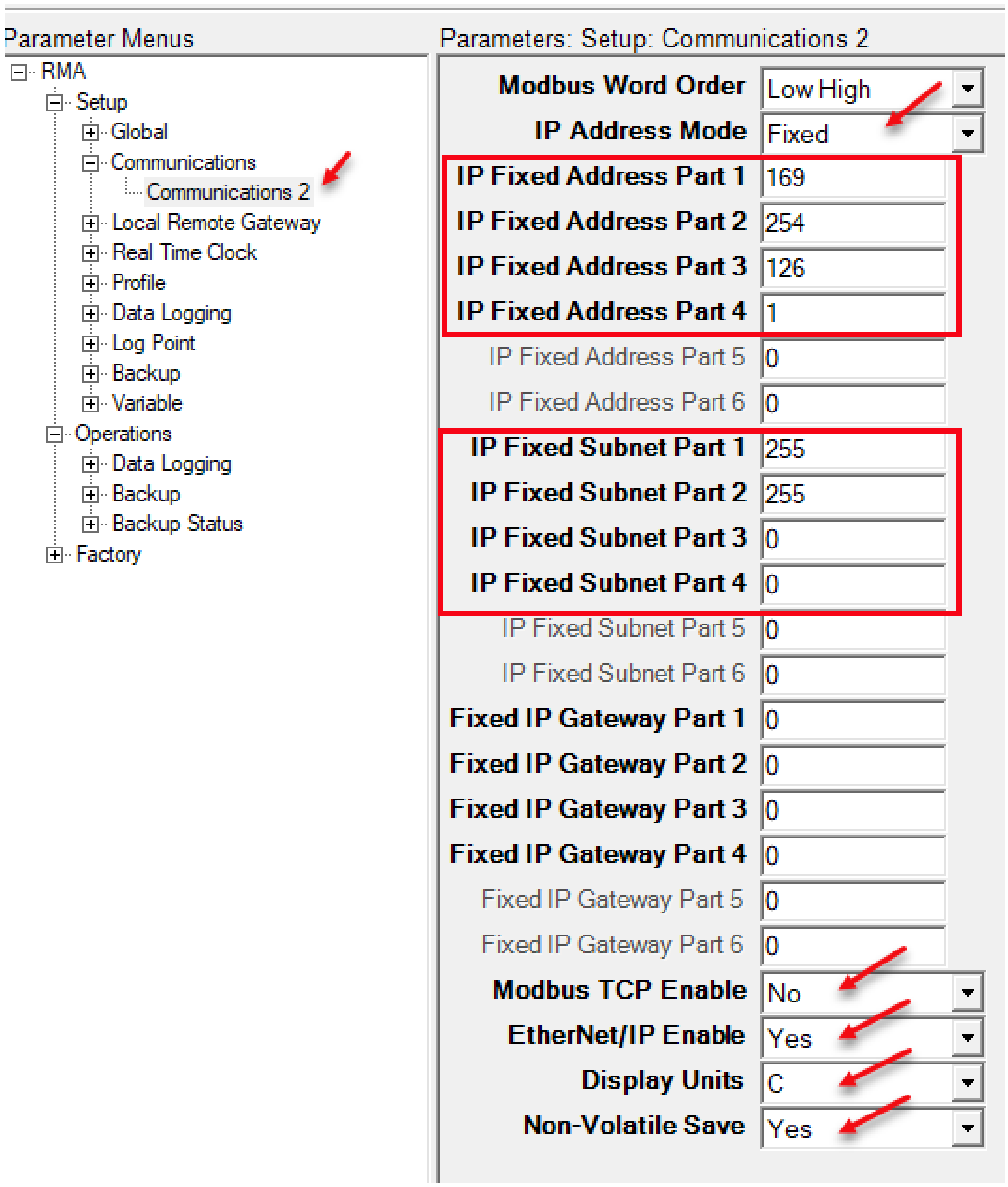
RMH Assembly Hex / RMH Assembly Dec / RMS Assembly Hex / RMS Assembly Dec / RML Assembly Hex / RML Assembly Dec / PN

Configuration for Data Exchange

RMA

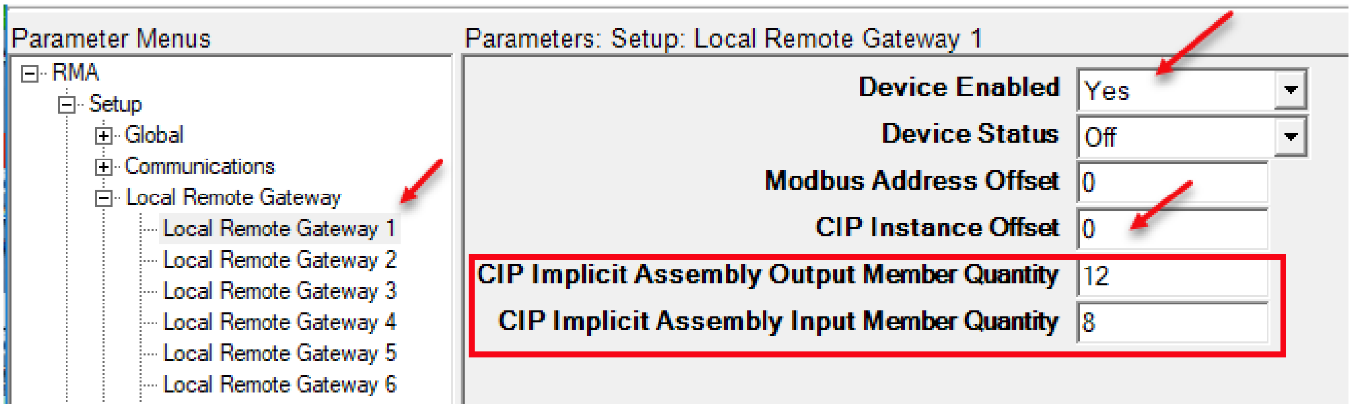
Configure Communications (port 2)

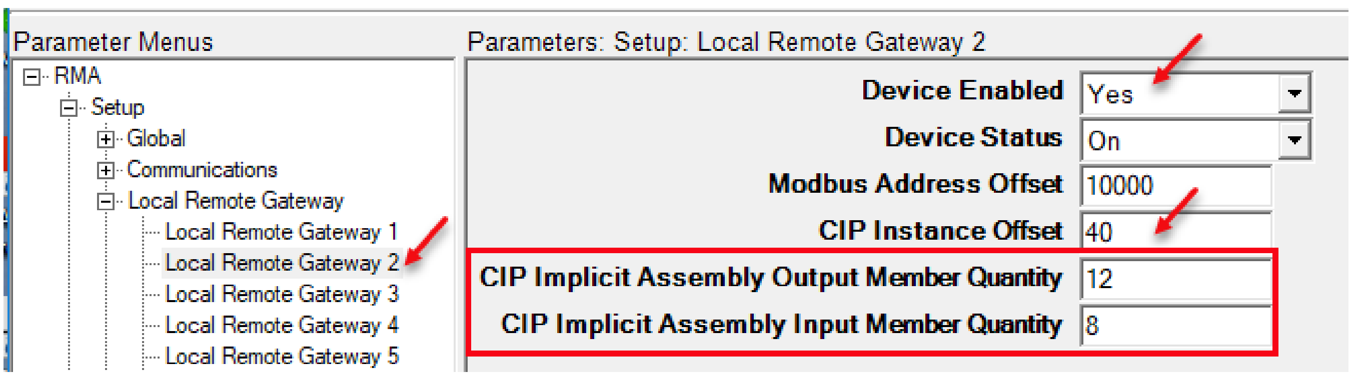
Using EZ-ZONE Configurator on a PC connected to the RMA via EIA-485, set the following to match your system. Display Units only affects the data over EtherNet/IP. In the Setup page, Global menu the display units affect Remote Terminal Unit option if present independently.

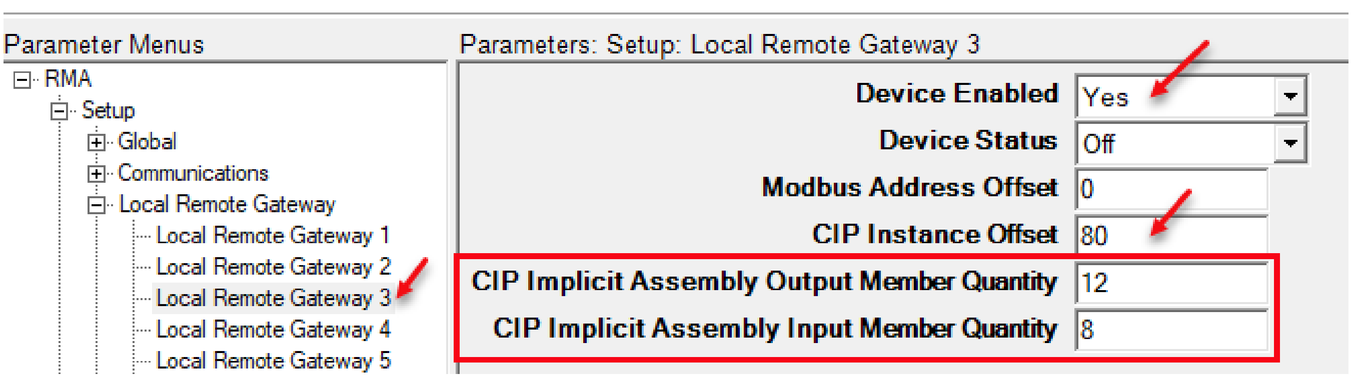


Configure Local Remote Gateways

Using EZ-ZONE Configurator on a PC connected to the RMA via EIA-485, set the following to match our example thus far. For PLC inputs, the RM outputs 12 + 12 + 12 + 1 = 37 tags. For PLC outputs, the RM inputs 8 + 8 + 8 = 24. Using CIP Instance Offset in multiples of 40 means we can explicitly write to instances no higher than 40 per module. The maximum instances per system is 255. Therefore, choose this value wisely based on explicit messaging requirements. In our example we used a total of 100 instance offsets; 0 - 39, 40 - 79, and 80 – 99.







The remaining Local Remote Gateways 4 to 16 should have Device Enabled = No. Device Status changes on its own when the RMA detects their presence. You do not need to change the status. The PLC device status has bits that indicate which modules are equal to On.

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

Program Tags into each RM module

There are two methods to change the tags inside the RM modules.

The first and easiest may be to use the software tool available on the Watlow web site at https://www.watlow.com/-/media/documents/software-and-demos/ezzone-assembly-programmer.ashx. Read the file "Setting Implicit Message Assembly.pdf" in EZ-ZONE included in downloaded zip file.
The second method detailed next requires using a message instruction in the PLC to write the Class, Instance and Attribute value to each location.

The O2T assembly instance 1 defines the tags the PLC will write to the RM modules.
The T2O assembly instance 2 defines the tags the PLC will read from the RM modules.

When modifying the implicit assemblies of *multiple* RM modules on the network, the CIP Instance Offset must be used. Several RM module types do not have anything programmed as default and therefore require the tags be defined.

Note: All numbers that are preceded by 0x are in hexadecimal format; numbers without the prefix of 0x are in decimal format. To change the 3rd Assembly Row at *Zone address 2* from what is shown above to Control Mode loop 3, first find the appropriate CIP address in the RMC User's Guide. It is 0x97, 3, 0x01. Then add the CIP Offset to both the tag to be written and the location written. In this example, the PLC will write to 0x77, 41, 0x03.

These are partial tables of our example listing the first four tags in each.

RMC Zone address 1 - O2T Assembly Instance 1						
Assembly Row (element)	CIP - Explicit write Class, Inst, Attribute (table pointer)	CIP - Write Class, Inst, Attribute (data pointer)	Parameter Name and Function (description)	Data Type (pointer)		Data Type (data value)
1	0x77, 1, 0x01	0x97, 1, 0x01	Control Loop 1, Control Mode	DINT		REAL
2	0x77, 1, 0x02	0x97, 2, 0x01	Control Loop 2, Control Mode	DINT		REAL
3	0x77, 1, 0x03	0x97, 3, 0x01	Control Loop 3, Control Mode	DINT		REAL
4	0x77, 1, 0x04	0x97, 4, 0x01	Control Loop 4, Control Mode	DINT		REAL

RMC Zone address 2 - O2T Assembly Instance 1						
Assembly Row (element)	CIP - Explicit write Class, Inst, Attribute (table pointer)	CIP - Write Class, Inst, Attribute (data pointer)	Parameter Name and Function (description)	Data Type (pointer)		Data Type (data value)
1	0x77, 41, 0x01	0x97, 41, 0x01	Control Loop 1, Control Mode	DINT		REAL
2	0x77, 41, 0x02	0x97, 42, 0x01	Control Loop 2, Control Mode	DINT		REAL
3	0x77, 41, 0x03	0x97, 43, 0x01	Control Loop 3, Control Mode	DINT		REAL
4	0x77, 41, 0x04	0x97, 44, 0x01	Control Loop 4, Control Mode	DINT		REAL

RMC Zone address 3 - O2T Assembly Instance 1

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

Assembly Row (element)	CIP - Explicit write Class, Inst, Attribute (table pointer)	CIP - Write Class, Inst, Attribute (data pointer)	Parameter Name and Function (description)	Data Type (pointer)		Data Type (data value)
1	0x77, 81, 0x01	0x97, 81, 0x01	Control Loop 1, Control Mode	DINT		REAL
2	0x77, 81, 0x02	0x97, 82, 0x01	Control Loop 2, Control Mode	DINT		REAL
3	0x77, 81, 0x03	0x97, 83, 0x01	Control Loop 3, Control Mode	DINT		REAL
4	0x77, 81, 0x04	0x97, 84, 0x01	Control Loop 4, Control Mode	DINT		REAL

RMC Zone address 1 - T2O Assembly Instance 2

Assembly Row (element)	CIP - Explicit write Class, Inst, Attribute (table pointer)	CIP - Write Class, Inst, Attribute (data pointer)	Parameter Name and Function (description)	Data Type (pointer)		Data Type (data value)
1	0x77, 2, 0x01	0x68, 1, 0x01	Analog Input 1 Value	DINT		REAL
2	0x77, 2, 0x02	0x68, 2, 0x01	Analog Input 2 Value	DINT		REAL
3	0x77, 2, 0x03	0x68, 3, 0x01	Analog Input 3 Value	DINT		REAL
4	0x77, 2, 0x04	0x68, 4, 0x01	Analog Input 4 Value	DINT		REAL

RMC Zone address 2 - T2O Assembly Instance 2

Assembly Row (element)	CIP - Explicit write Class, Inst, Attribute (table pointer)	CIP - Write Class, Inst, Attribute (data pointer)	Parameter Name and Function (description)	Data Type (pointer)		Data Type (data value)
1	0x77, 42, 0x01	0x68, 41, 0x01	Analog Input 1 Value	DINT		REAL
2	0x77, 42, 0x02	0x68, 42, 0x01	Analog Input 2 Value	DINT		REAL
3	0x77, 42, 0x03	0x68, 43, 0x01	Analog Input 3 Value	DINT		REAL
4	0x77, 42, 0x04	0x68, 44, 0x01	Analog Input 4 Value	DINT		REAL

RMC Zone address 3 - T2O Assembly Instance 2

Assembly Row (element)	CIP - Explicit write Class, Inst, Attribute (table pointer)	CIP - Write Class, Inst, Attribute (data pointer)	Parameter Name and Function (description)	Data Type (pointer)		Data Type (data value)
1	0x77, 82, 0x01	0x68, 81, 0x01	Analog Input 1 Value	DINT		REAL
2	0x77, 82, 0x02	0x68, 82, 0x01	Analog Input 2 Value	DINT		REAL
3	0x77, 82, 0x03	0x68, 83, 0x01	Analog Input 3 Value	DINT		REAL
4	0x77, 82, 0x04	0x68, 84, 0x01	Analog Input 4 Value	DINT		REAL

The explicit message instruction configuration now becomes a Set Attribute Single operation while a specific tag must be created which contains the new parameter address pointer (New_Assembly_Data) to be written to the RMC assembly. The message configuration would change as shown below for module 2 in this example.

The Source Element was created as a 3-dimensional array using the SINT data type because the Class, Instance and Attribute objects are 8-bits in length.

- New_Assembly_Data	{. . .}	Decimal	SINT[3]
+ New_Assembly_Data[0]	16#97	Hex	SINT
+ New_Assembly_Data[1]	43	Decimal	SINT
+ New_Assembly_Data[2]	16#01	Hex	SINT

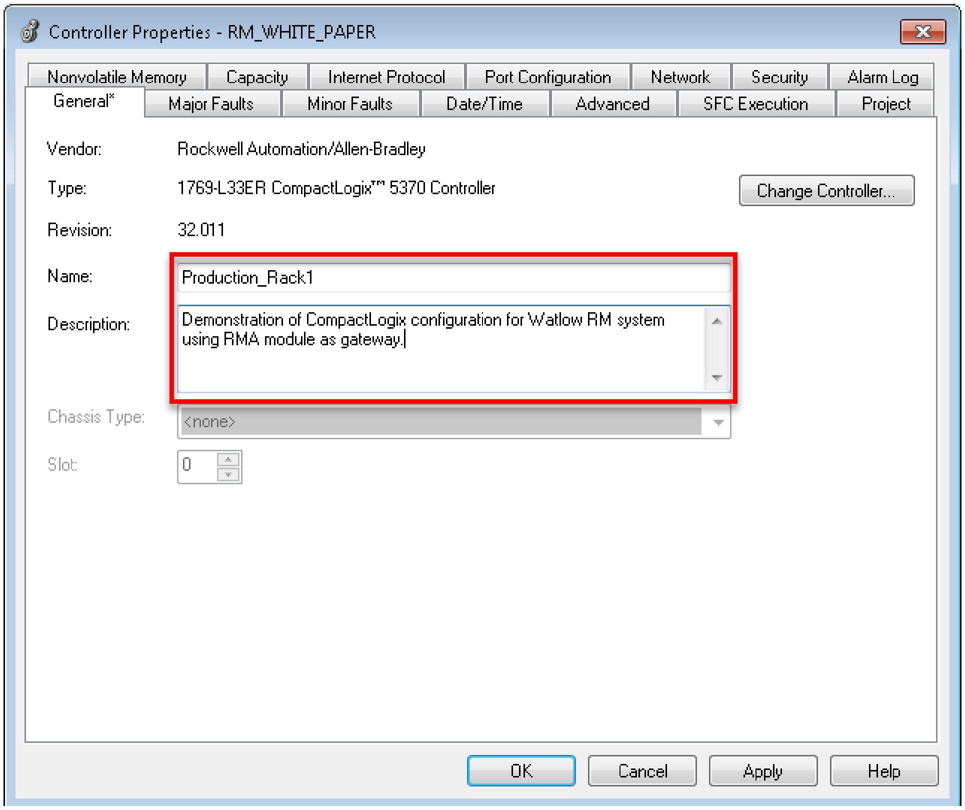
Notice in the graphic above that the instance for the 3rd assembly location as well as the parameter to be written have the listed offset (40) added to each. Offset was 0 for first module, 40 for second module and 80 for third module. The same offset applies when writing the pointer. These are the values entered in the RMA under local remote gateway for CIP Offsets. The graphic below shows changing the 3-assembly member with the pointer for control mode, control loop 4 at address 3.

Cyclic data should now be exchanged between the RM and PLC. The raw data in the PLC will display values in DINT format. The User Defined type should convert the raw data to integer and real numbers as defined.

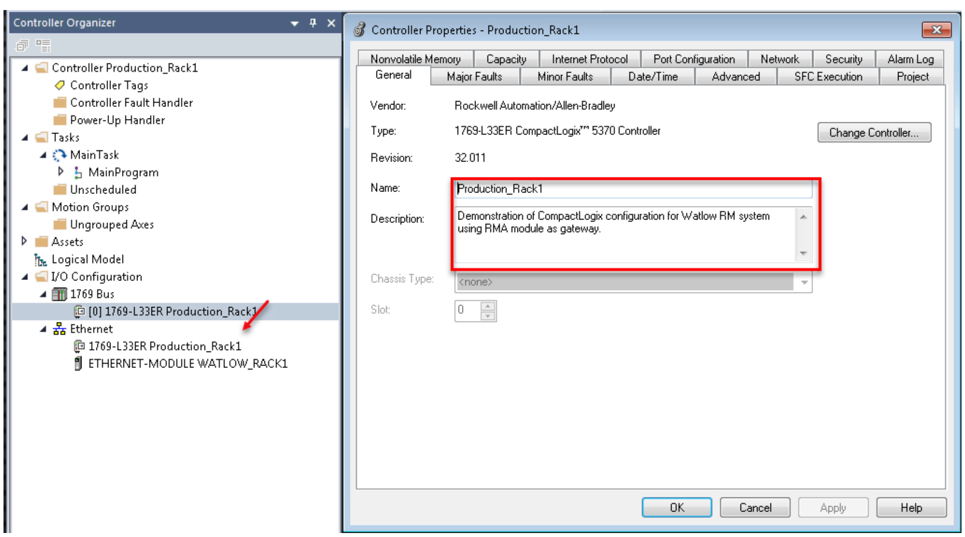
PLC

Configure Generic Ethernet Module

The steps to program a typical AB CompactLogix 5370 controller, 1769-L33ER PLC in general are listed below. You’re model of PLC and firmware revision will affect the screens displayed. The items highlighted in red call attention to input. Do not simply enter what is shown as these are example screens.

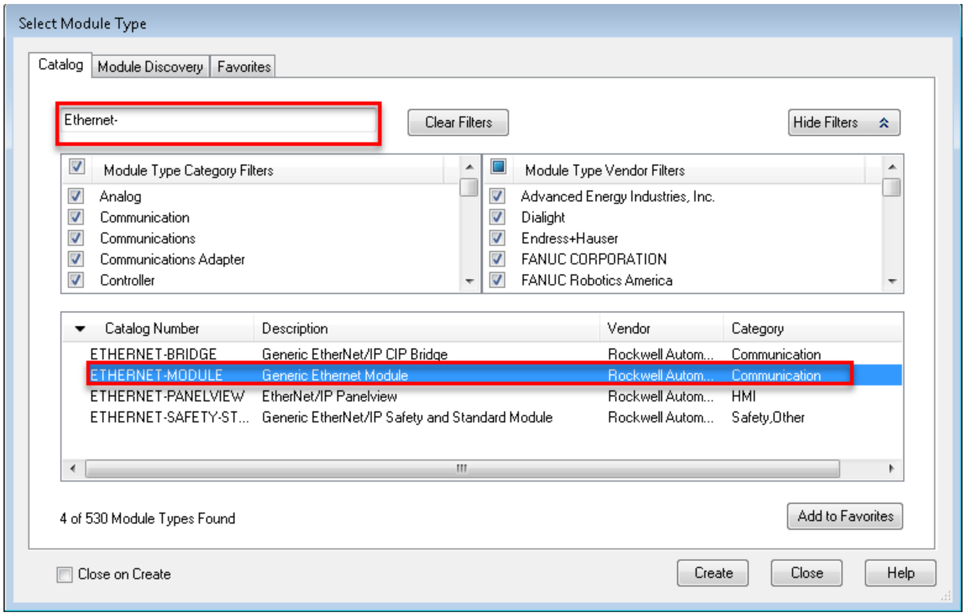


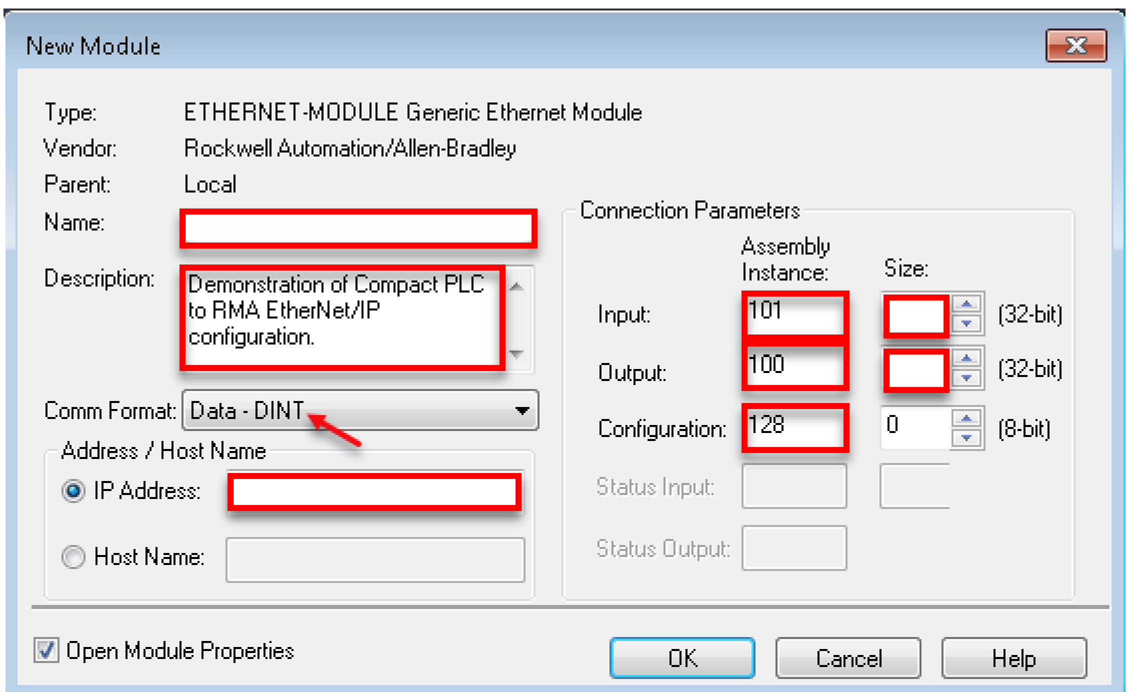
Controller Properties - RM_WHITE_PAPER
Nonvolatile Memory
Capacity
Internet Protocol
Port Configuration
Network
Security
Alarm Log
General*
Major Faults
Minor Faults
Date/Time
Advanced
SFC Execution
Project
Vendor:
Rockwell Automation/Allen-Bradley
Type:
1769-L33ER CompactLogix™ 5370 Controller
Change Controller...
Revision:
32.011
Name:
Production_Rack1
Description:
Demonstration of CompactLogix configuration for Watlow RM system using RMA module as gateway.
Chassis Type:
<none>
Slot:
0
OK
Cancel
Apply
Help



Controller Organizer
Controller Production_Rack1
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
MainTask
MainProgram
Unscheduled
Motion Groups
Ungrouped Axes
Assets
Logical Model
I/O Configuration
1769 Bus
[0] 1769-L33ER Production_Rack1
Ethernet
1769-L33ER Production_Rack1
ETHERNET-MODULE WATLOW_RACK1
Controller Properties - Production_Rack1
Nonvolatile Memory
Capacity
Internet Protocol
Port Configuration
Network
Security
Alarm Log
General
Major Faults
Minor Faults
Date/Time
Advanced
SFC Execution
Project
Vendor:
Rockwell Automation/Allen-Bradley
Type:
1769-L33ER CompactLogix™ 5370 Controller
Change Controller...
Revision:
32.011
Name:
Production_Rack1
Description:
Demonstration of CompactLogix configuration for Watlow RM system using RMA module as gateway.
Chassis Type:
<none>
Slot:
0
OK
Cancel
Apply
Help

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





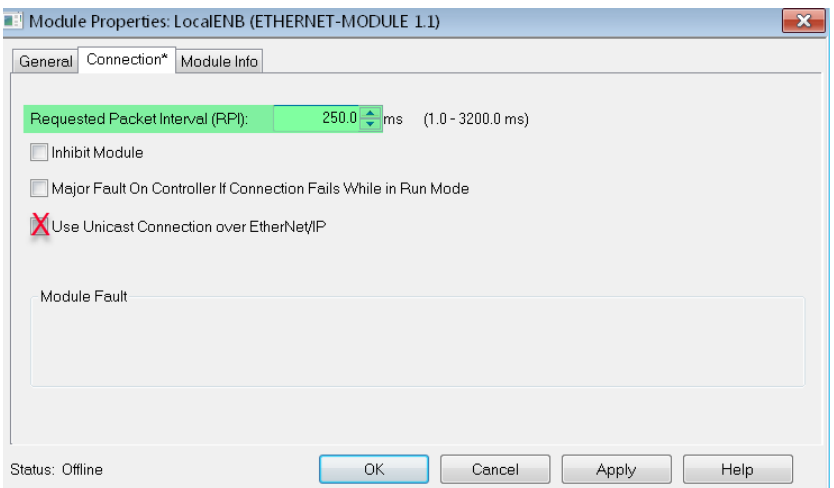
1) Using RSLogix5000 software, add an additional I/O Ethernet module. Select Ethernet-Module Generic Ethernet Module.
2) Define new module properties;
Name: Given name becomes part of controller tags to be used in program. This example will use RACK_1
Description: Allows detail about the purpose of the module.

Comm Format: Defines how data is to be treated within PLC.
Note: All EZ-ZONE assembly members are 32-bits in length.
In our example, you recorded the number of tags earlier, if 36 PLC input and 24 PLC output (32-bit) members are in use, the appropriate Comm Formats would be:
DINT (32-bit): *Inputs = 37, Output = 24, Configuration = 0

* The RMA output assembly within EZ-ZONE modules have one dedicated 32-bit status member that is always present. The PLC input assembly size will be n+1 where n = the size of the combined (all modules) output assemblies. Each enabled Local Remote Gateway in the RMA list the dedicated assembly size on a module-by-module basis.

IP Address: Enter assigned RMA IP Address. Ex. 192.168.1.100
Assembly Instance Input (101): Size defines number of members to be included in the PLC Input Implicit Assembly which comes from the sum “CIP Implicit Assembly Output Member Quantity” and status word.
Assembly Instance Output (100): Size defines number of members to be included in the PLC Output Implicit Assembly which goes into “CIP Implicit Assembly Input Member Quantity.
Configuration (128): Enter zero.

3) The PLC will be programmed for the requested packet interval time. 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 greater.
4) If PLC supports Unicast Connection, select the option.



5) Create two data type structures of User-Defined, one each for T2O and O2T. In this example, the names used are T2O and O2T. The provided names will be appended to the tags by RSLogix 5000 software.

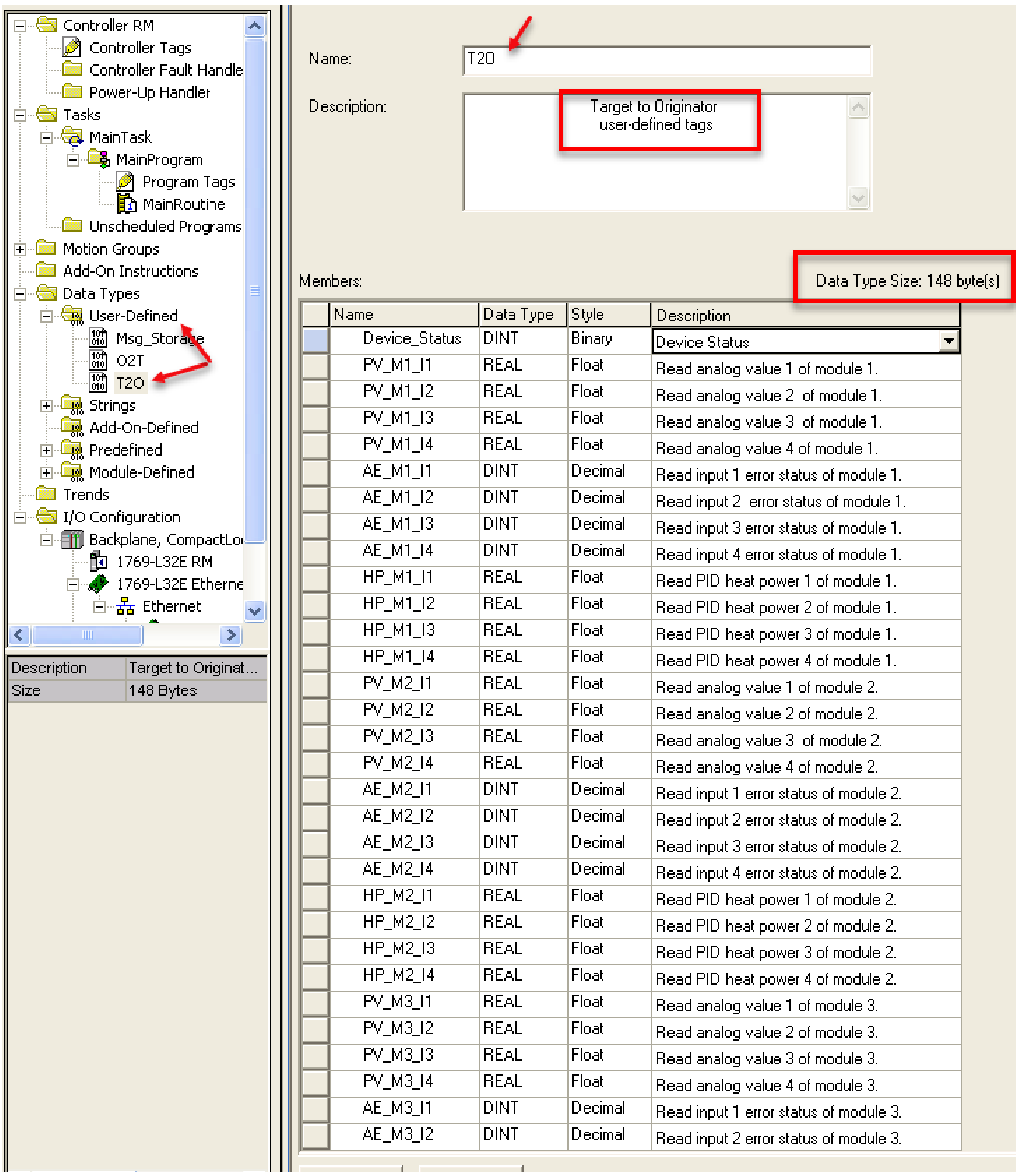
Create User Defined Structure

Watlow suggests a User-Defined Data Type be created for both implicit assemblies which will reflect the appropriate data format. This will simplify the programming when transferring I/O data between Originator (PLC) and Target (RMC module) devices.

Shown in next pages, a User-Defined data type was created (T2O) using the tags previously defined. Provide a name and description for the user-defined structure being created. Then enter each tag by Name, Data Type and Description. The Style is populated based on data type. The Data Type Size increments in byte count to track the size of the structure. Remember that the T2O are read only parameters. Ensure that tags are only entered once and are valid for your hardware/application.

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

Note that the screen capture does not show all 37 tags entered as they are off screen. The total tag count uses 148 bytes of data transfer. There are 36 tags for the three RMC modules plus the device status word for the system giving a total of 37 tags. Each tag is 32-bits or 4 bytes wide. (37 tags * 4 bytes = 148 bytes)



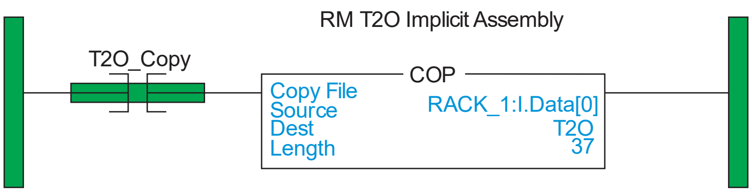
EZ-ZONE® RM & EtherNet/IP™ Configuration & Ladder Logic Example AB CompactLogix or ControlLogix PLC

The corresponding T2O controller tags are created as shown below.

Name	Value	Force	Style	Data Type	Description
+ RACK_1:C	{...}	{...}		AB:ETHE...	
+ RACK_1:I	{...}	{...}		AB:ETHE...	PLC Input RAW Implicit Data
+ RACK_1:O	{...}	{...}		AB:ETHE...	PLC Output RAW Implicit Data
− T2O	{...}	{...}		T2O	RM to PLC User-defined Structure
+ T2O.Device_Status	2#0000_0000_0000...		Binary	DINT	RM to PLC User-defined Structure Device Status
T2O.PV_M1_I1	0.0		Float	REAL	RM to PLC User-defined Structure Read analog value 1 of module 1.
T2O.PV_M1_I2	0.0		Float	REAL	RM to PLC User-defined Structure Read analog value 2 of module 1.
T2O.PV_M1_I3	0.0		Float	REAL	RM to PLC User-defined Structure Read analog value 3 of module 1.
T2O.PV_M1_I4	0.0		Float	REAL	RM to PLC User-defined Structure Read analog value 4 of module 1.
+ T2O.AE_M1_I1	0		Decimal	DINT	RM to PLC User-defined Structure Read input 1 error status of module 1.
+ T2O.AE_M1_I2	0		Decimal	DINT	RM to PLC User-defined Structure Read input 2 error status of module 1.
+ T2O.AE_M1_I3	0		Decimal	DINT	RM to PLC User-defined Structure Read input 3 error status of module 1.
+ T2O.AE_M1_I4	0		Decimal	DINT	RM to PLC User-defined Structure Read input 4 error status of module 1.
T2O.HP_M1_I1	0.0		Float	REAL	RM to PLC User-defined Structure Read PID heat power 1 of module 1.
T2O.HP_M1_I2	0.0		Float	REAL	RM to PLC User-defined Structure Read PID heat power 2 of module 1.
T2O.HP_M1_I3	0.0		Float	REAL	RM to PLC User-defined Structure Read PID heat power 3 of module 1.
T2O.HP_M1_I4	0.0		Float	REAL	RM to PLC User-defined Structure Read PID heat power 4 of module 1.
T2O.PV_M2_I1	0.0		Float	REAL	RM to PLC User-defined Structure Read analog value 1 of module 2.
T2O.PV_M2_I2	0.0		Float	REAL	RM to PLC User-defined Structure Read analog value 2 of module 2.
T2O.PV_M2_I3	0.0		Float	REAL	RM to PLC User-defined Structure Read analog value 3 of module 2.
T2O.PV_M2_I4	0.0		Float	REAL	RM to PLC User-defined Structure Read analog value 4 of module 2.
+ T2O.AE_M2_I1	0		Decimal	DINT	RM to PLC User-defined Structure Read input 1 error status of module 2.
+ T2O.AE_M2_I2	0		Decimal	DINT	RM to PLC User-defined Structure Read input 2 error status of module 2.
+ T2O.AE_M2_I3	0		Decimal	DINT	RM to PLC User-defined Structure Read input 3 error status of module 2.
+ T2O.AE_M2_I4	0		Decimal	DINT	RM to PLC User-defined Structure Read input 4 error status of module 2.
T2O.HP_M2_I1	0.0		Float	REAL	RM to PLC User-defined Structure Read PID heat power 1 of module 2.
T2O.HP_M2_I2	0.0		Float	REAL	RM to PLC User-defined Structure Read PID heat power 2 of module 2.
T2O.HP_M2_I3	0.0		Float	REAL	RM to PLC User-defined Structure Read PID heat power 3 of module 2.
T2O.HP_M2_I4	0.0		Float	REAL	RM to PLC User-defined Structure Read PID heat power 4 of module 2.
T2O.PV_M3_I1	0.0		Float	REAL	RM to PLC User-defined Structure Read analog value 1 of module 3.
T2O.PV_M3_I2	0.0		Float	REAL	RM to PLC User-defined Structure Read analog value 2 of module 3.
T2O.PV_M3_I3	0.0		Float	REAL	RM to PLC User-defined Structure Read analog value 3 of module 3.
T2O.PV_M3_I4	0.0		Float	REAL	RM to PLC User-defined Structure Read analog value 4 of module 3.
+ T2O.AE_M3_I1	0		Decimal	DINT	RM to PLC User-defined Structure Read input 1 error status of module 3.
+ T2O.AE_M3_I2	0		Decimal	DINT	RM to PLC User-defined Structure Read input 2 error status of module 3.
+ T2O.AE_M3_I3	0		Decimal	DINT	RM to PLC User-defined Structure Read input 3 error status of module 3.
+ T2O.AE_M3_I4	0		Decimal	DINT	RM to PLC User-defined Structure Read input 4 error status of module 3.
T2O.HP_M3_I1	0.0		Float	REAL	RM to PLC User-defined Structure Read PID heat power 1 of module 3.
T2O.HP_M3_I2	0.0		Float	REAL	RM to PLC User-defined Structure Read PID heat power 2 of module 3.
T2O.HP_M3_I3	0.0		Float	REAL	RM to PLC User-defined Structure Read PID heat power 3 of module 3.
T2O.HP_M3_I4	0.0		Float	REAL	RM to PLC User-defined Structure Read PID heat power 4 of module 3.

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

Enter a rung of logic shown to read the Target data into the PLC tag created above. Match the source, destination names and length to what you created.



Recall that when the module was added that there were entries for the PLC input, output and configuration assemblies. The input assembly was defined as having 37 members. The Copy destination tag is one structure so the length is 1. Once the contact (T2O_Copy) is enabled the source data will be copied to the destination as can be seen below.

Now that the data formats correspond to each parameter (member) data type we see values that are more in alignment with expectations.

The first member of T2O is “Device Status”. This member is sourced from the RMA module. Regardless of how many RM modules are connected to the RMA there is only one assembly member referred to as the Device Status. Counting the bits from right to left, bit 15 is shown as being set to a “1”. This bit represents the RMA module being present in the PLC I/O structure. Bits 16 through 31 represent gateway instances (EZ-ZONE modules on Standard Bus). Once the gateway instance is enabled, if there are successful communications to the RMA module the corresponding bit will be set to a one.

PLC Device Status

RM 16	RM 15	RM 14	RM 13	RM 12	RM 11	RM 10	RM 9	RM 8	RM 7	RM 6	RM 5	RM 4	RM 3	RM 2	RM 1	RMA Comm	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Bits 31 to 24								Bits 23 to 16								Bits 15 to 8								Bits 7 to 0							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Above, bits 16, 17 and 18 (RM Standard Bus address 1, 2 and 3) are set to a “1” because RMA has Local Remote Gateways 1, 2 and 3 enabled. In addition, modules 1, 2 and 3 are communicating successfully with the RMA module in this example.

Note: The other bits (16 through 31) being “0” could represent a module problem or may simply mean those instances have not been enabled. If a module is remove such as 2 in this example while still being enabled would cause the bit to clear.

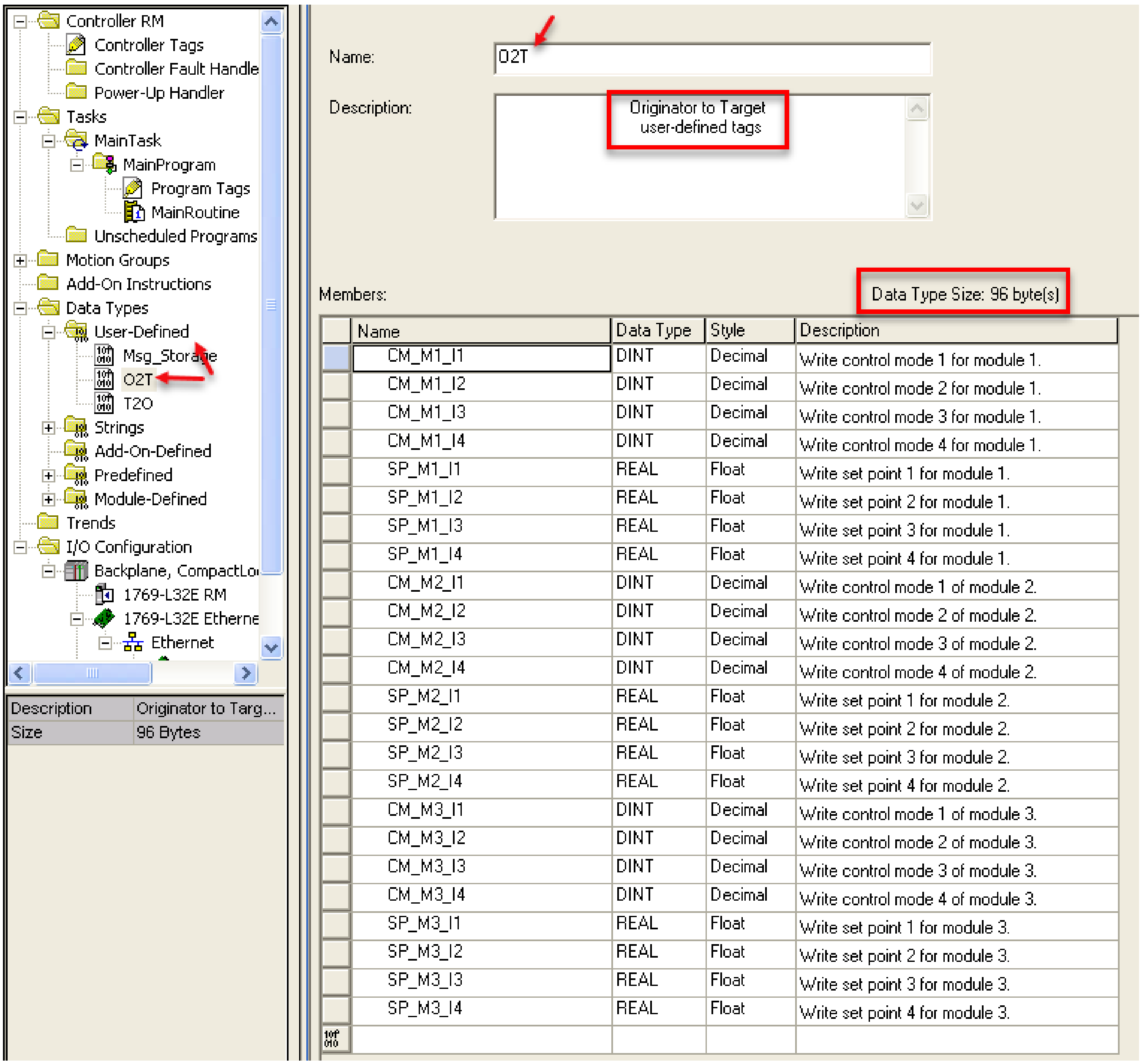
Bits 0 to 14 should not be used to make any decisions as they are not used by the RMA. Bit 15 is always set to 1.

To write data out to the Target, perform the same steps in the creation of the O2T assembly; as can be seen below, the User Defined Data Type is created based on the example. Keep in mind that this assembly represents the output from the PLC or Originator which will be sent to the Target or EZ-ZONE module. The program within the PLC would write values to these tags sent at the RPI rate setting.

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

Referenced below, a user defined data type was created (O2T) using the tags previously defined. Provide a name and description for the user-defined structure being created. Then enter each tag by Name, Data Type and Description. The Style is populated based on data type. The Data Type Size increments in byte count to track the size of the structure. Remember that the O2T are write only parameters. Ensure that a particular tag is only entered once and is valid for your hardware/application.

The total tag count uses 96 bytes of data transfer. Each tag is 32-bits or 4 bytes wide. (24 tags * 4 bytes = 96 bytes)



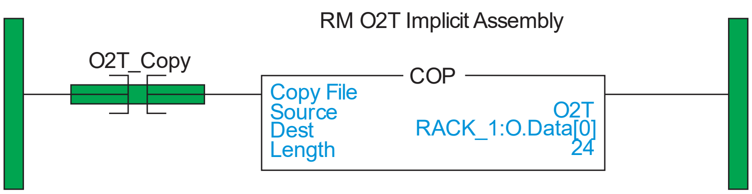
Name	Data Type	Style	Description
CM_M1_I1	DINT	Decimal	Write control mode 1 for module 1.
CM_M1_I2	DINT	Decimal	Write control mode 2 for module 1.
CM_M1_I3	DINT	Decimal	Write control mode 3 for module 1.
CM_M1_I4	DINT	Decimal	Write control mode 4 for module 1.
SP_M1_I1	REAL	Float	Write set point 1 for module 1.
SP_M1_I2	REAL	Float	Write set point 2 for module 1.
SP_M1_I3	REAL	Float	Write set point 3 for module 1.
SP_M1_I4	REAL	Float	Write set point 4 for module 1.
CM_M2_I1	DINT	Decimal	Write control mode 1 of module 2.
CM_M2_I2	DINT	Decimal	Write control mode 2 of module 2.
CM_M2_I3	DINT	Decimal	Write control mode 3 of module 2.
CM_M2_I4	DINT	Decimal	Write control mode 4 of module 2.
SP_M2_I1	REAL	Float	Write set point 1 for module 2.
SP_M2_I2	REAL	Float	Write set point 2 for module 2.
SP_M2_I3	REAL	Float	Write set point 3 for module 2.
SP_M2_I4	REAL	Float	Write set point 4 for module 2.
CM_M3_I1	DINT	Decimal	Write control mode 1 of module 3.
CM_M3_I2	DINT	Decimal	Write control mode 2 of module 3.
CM_M3_I3	DINT	Decimal	Write control mode 3 of module 3.
CM_M3_I4	DINT	Decimal	Write control mode 4 of module 3.
SP_M3_I1	REAL	Float	Write set point 1 for module 3.
SP_M3_I2	REAL	Float	Write set point 2 for module 3.
SP_M3_I3	REAL	Float	Write set point 3 for module 3.
SP_M3_I4	REAL	Float	Write set point 4 for module 3.

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

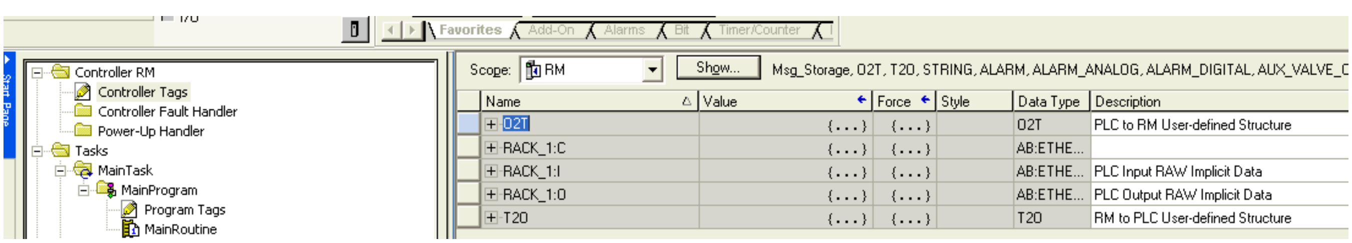
The corresponding O2T controller tag is created as shown below.

Name	Value	Force	Style	Data Type	Description
− O2T	{...}	{...}		O2T	PLC to RM User-defined Structure
+ O2T.CM_M1_I1	0		Decimal	DINT	PLC to RM User-defined Structure Write control mode 1 for module 1.
+ O2T.CM_M1_I2	0		Decimal	DINT	PLC to RM User-defined Structure Write control mode 2 for module 1.
+ O2T.CM_M1_I3	0		Decimal	DINT	PLC to RM User-defined Structure Write control mode 3 for module 1.
+ O2T.CM_M1_I4	0		Decimal	DINT	PLC to RM User-defined Structure Write control mode 4 for module 1.
O2T.SP_M1_I1	0.0		Float	REAL	PLC to RM User-defined Structure Write set point 1 for module 1.
O2T.SP_M1_I2	0.0		Float	REAL	PLC to RM User-defined Structure Write set point 2 for module 1.
O2T.SP_M1_I3	0.0		Float	REAL	PLC to RM User-defined Structure Write set point 3 for module 1.
O2T.SP_M1_I4	0.0		Float	REAL	PLC to RM User-defined Structure Write set point 4 for module 1.
+ O2T.CM_M2_I1	0		Decimal	DINT	PLC to RM User-defined Structure Write control mode 1 of module 2.
+ O2T.CM_M2_I2	0		Decimal	DINT	PLC to RM User-defined Structure Write control mode 2 of module 2.
+ O2T.CM_M2_I3	0		Decimal	DINT	PLC to RM User-defined Structure Write control mode 3 of module 2.
+ O2T.CM_M2_I4	0		Decimal	DINT	PLC to RM User-defined Structure Write control mode 4 of module 2.
O2T.SP_M2_I1	0.0		Float	REAL	PLC to RM User-defined Structure Write set point 1 for module 2.
O2T.SP_M2_I2	0.0		Float	REAL	PLC to RM User-defined Structure Write set point 2 for module 2.
O2T.SP_M2_I3	0.0		Float	REAL	PLC to RM User-defined Structure Write set point 3 for module 2.
O2T.SP_M2_I4	0.0		Float	REAL	PLC to RM User-defined Structure Write set point 4 for module 2.
+ O2T.CM_M3_I1	0		Decimal	DINT	PLC to RM User-defined Structure Write control mode 1 of module 3.
+ O2T.CM_M3_I2	0		Decimal	DINT	PLC to RM User-defined Structure Write control mode 2 of module 3.
+ O2T.CM_M3_I3	0		Decimal	DINT	PLC to RM User-defined Structure Write control mode 3 of module 3.
+ O2T.CM_M3_I4	0		Decimal	DINT	PLC to RM User-defined Structure Write control mode 4 of module 3.
O2T.SP_M3_I1	0.0		Float	REAL	PLC to RM User-defined Structure Write set point 1 for module 3.
O2T.SP_M3_I2	0.0		Float	REAL	PLC to RM User-defined Structure Write set point 2 for module 3.
O2T.SP_M3_I3	0.0		Float	REAL	PLC to RM User-defined Structure Write set point 3 for module 3.
O2T.SP_M3_I4	0.0		Float	REAL	PLC to RM User-defined Structure Write set point 4 for module 3.

Enter the rung of logic shown below to write data from the Originator to the Target.

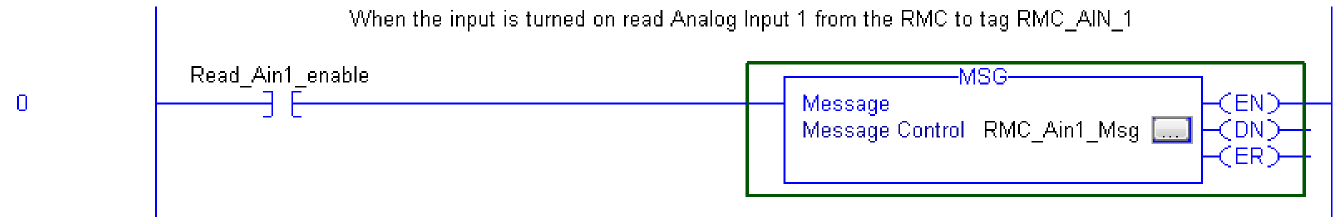


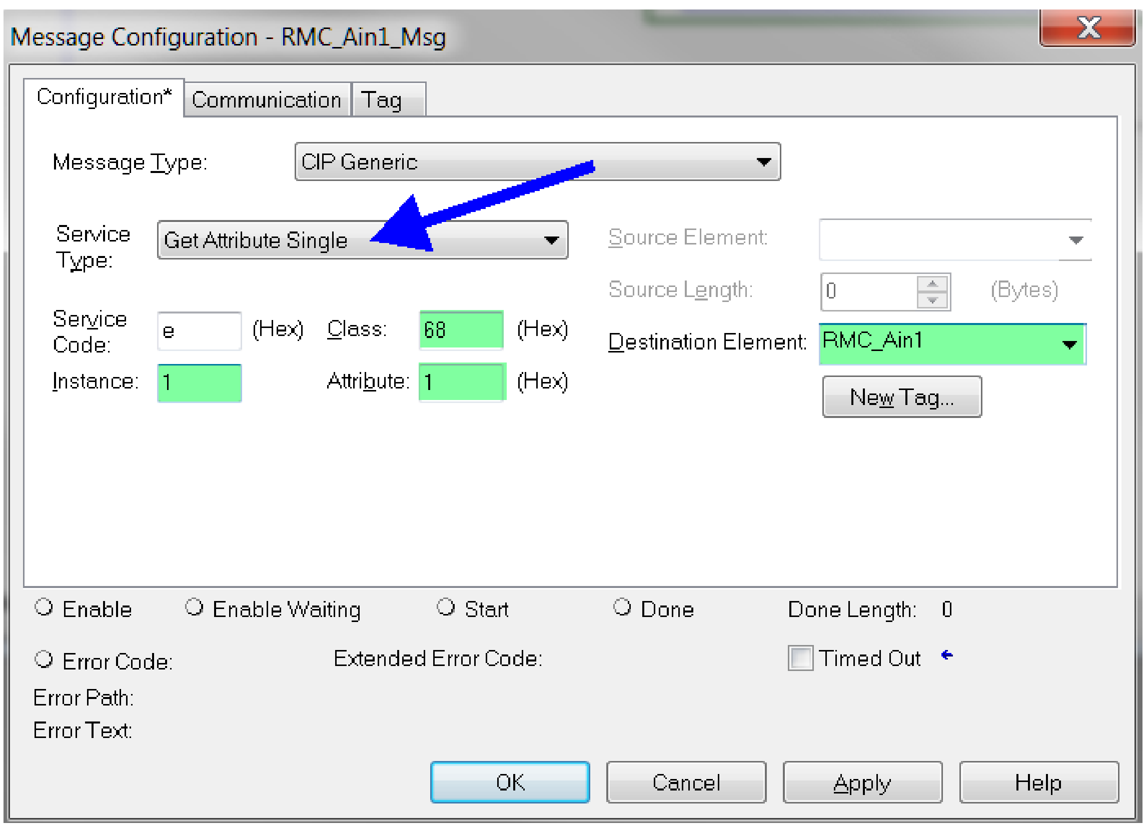
Notice that the source of the copy instruction is now the controller tag created above where the destination is the same name given to the module back in step 2.3.4. In this case, the output assembly was defined as having 24 members as was the destination tag. Therefore, the length is defined as 24. Once the contact (O2T_Copy) is enable the source data will be sent to the destination as can be seen below.



Explicit Programming Examples

The example below will use a very simple and straight forward way to execute an explicit message.
To explicitly read the first Analog input from an RMC module create a rung of logic like that shown below.





- Service Type
This example gets a single parameter (attribute) from the RM, therefore the Service Type is “Get Attribute Single”.

- Class, Instance and Attribute
The CIA for each parameter can be found in the User’s Guide in column labeled CIP. Values in parenthesis () are decimal equivalent of hex number. The Class and Attribute are always entered in hexadecimal where the instance is entered in decimal. The instance is 1 because this is from module 1 and CIP instance offset is 0. In this example to read analog input 1 from module 2, the instance changes from 1 to 41. Do not forget to add the offset.

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

These fields represent the actual CIP address within the RMC for the Analog Input. A CIP instance offset may need to be added to get to the appropriate module.

Display	Parameter Name Description	Range	CIP Class Instance Attribute hex (dec)	Data Type and Access **
Ai oPEr Analog Input Menu				
Ain Ain	*Analog Input (1 to 4)* **Analog Input Value** View the process value.	-1,999.000 to 9,999.000°F or units -1,128.000 to 5,537.000°C	0x68 (104) 1 to 4 1	float R

Troubleshooting

1) Unable to connect from PLC to RM.
   a. Check if link LED is illuminated to check for cable integrity. Red LED = 10Mbit, Green LED = 100Mbit.
   b. Check Ethernet pigtail is wired correctly to RMA.
   c. Check that the Ethernet switch negotiated half-duplex.
   d. Check if IP address and subnet mask are on same logical network.
      i. When using recommended Fixed IP addressing mode, cycle power to RM system for new IP address to be in effect.
      ii. Check if RM may be pinged from PC/PLC if possible.
   e. Check if Module LED is illuminated. Off = No power, Green = Device Operational, Flashing Green = Standby, Flashing Red = Minor Fault, Steady Red = Major Fault.
   f. Check if EtherNet/IP protocol is enabled in RMA.
   g. Check for local remote gateways being enabled for existing modules and disabled for non-existent modules. Within each local remote gateway, “Device Status” should show “On” for modules that exist.
   h. Check CIP Implicit Assembly Input and Output Member Quantity values in RMA to match Output and Input Members of PLC. PLC has one additional input member added to its assembly.
2) Intermittent loss of Ethernet connection.
   a. Check for managed switch having IGMP snooping capability.
   b. Check for duplicate IP address.
   c. Check for intermittent Ethernet cable.
   d. Check for power supply disruptions to RM or Ethernet switch.
3) Wrong tag values being displayed in PLC.
   a. Check for correct tag definition on PLC and in RMA.
   b. Check display units correctly selected for communications. The setting is independent from Remote User Interface display units.
   c. For sensor readings, ensure valid sensors are connected. Check sensor status to determine valid values.
   d. For explicit messages, use CIP offset added to instance of tag to route to correct module.

1241 Bundy Boulevard • P.O. Box 5580 • Winona, Minnesota 55987-5580 USA
Phone: 507-454-5300 • Fax: 507-452-4507 • www.watlow.com • inquiry@watlow.com