VMP EtherNet/IP™ Guide

Abstract

This document is to be used by controls engineers and other technical personal responsible for communications between programmable logic controllers (PLC's) and the variable metering pump (VMP) produced by Zaxis Inc. The document assumes the reader is well acquainted with EtherNet/IP™ as defined the ODVA association. The reader should also have a good working knowledge of TCP/IP and UDP/IP protocols.

Last updated
18-Aug-20

Table of Contents

VMP EtherNet/IP™ Guide 0
Introduction 1
EtherNet/IP™ for the VMP overview 2
VMP Configuration for Ethernet/IP™ 3
PLC Configuration for Ethernet/IP™ 3
Configuration with EDS file 4
Generic Ethernet Module configuration 7
VMP Tag Groups 12
Configuration Tags 12
PLC Output Tags 12
Input Tags 13
Pump Setup/Program parameters 15
Setup Parameters 15
Reading and Writing Setup Parameters 16
Program Parameters 19
TCP/IP Network Configuration 22
Trouble Shooting 26

Introduction

Zaxis's VMP is an automated variable metering pump used for fluid dispensing and metering. The VMP uses internal ceramics that are designed for millions of maintenance free cycles with minimal to zero downtime. The valve-less ceramic internals provide syringe-like precision with drift free accuracy. Eliminating the inaccuracy associated with tubing and diaphragm pumps and built a robust system that is precise and dependable.

Communications with the VMP can be accomplished in different ways, RS232/485, TCP/IP and EtherNet/IP™. This document will describe how to setup EtherNet/IP™ using an Allen-Bradley 1769 CompactLogix Controller. Other types of PLC can be used if compliant to the EtherNet/IP™ specifications as defined the ODVA association.

EtherNet/IP™ for the VMP overview

The "IP" in "EtherNet/IP" refers to "Industrial Protocol" and should not be confused with Ethernet or the TCP/IP and UDP/IP protocols. Although EtherNet/IP™ uses these technologies and protocols to accomplish the desired communications between the VMP and PLC's.

The VMP software consists of a number of different modules, as for EtherNet/IP™ the software follows the recommended implementation as defined in the ODVA specifications. See *The CIP Networks LIBRARY Volume 1 and 2.* The VMP EtherNet/IP™ software levels are: Application, Session, Transport, Network, Data Link, Physical as show below in Table 1.

Application	VMP EtherNet/IP
Session	Explicit/Implicit Messaging (CIP)
Transport	TCP/UDP
Network	Internet Protocol (IP)
Data Link	Ethernet
Physical	Peer-to-peer, multicast, unicast

Table 1 - VMP EtherNet/IP™ Adaptation of CIP

The Application and Session layers (as shown in top two rows) are customized for the VMP, while the lower layers (as shown in bottom four rows) follow the IEEE 802.3 standards with little or no customization. Thus, allowing existing EtherNet switches and/or networks to communicate with the VMP.

The VMP supports both DHCP and static Ethernet configurations. Factory default use the following static network configuration:

I/P Address	192.168.1.130
Network Mask	255.255.255.0
Gateway	192.168.1.1

Table 2 - Factory Default Network Configuration

These defaults can be changed using the VMP TSi (Touch Screen interface) or PLC MSG command as described below. It is recommended that a static configuration be used to ensure devices that communicate to the VMP can rely on the same I/P addresses for each VMP. However, it is possible to use DHCP if the DHCP server and/or Name server are able to resolve a VMP hostname to a static I/P address.

Because EtherNet/IP™ relays on TCP/UDP it is necessary that the VMP and any device wishing to communicate with the VMP over Ethernet have the correct network configuration. The network *ping* command can be used to diagnose network problems etc.

EtherNet/IP™ uses two forms of messaging:

- **Unconnected messaging** is used to: 1) establish connections, 2) low-priority messages, and 3) infrequent messages. Unconnected messages are handled by the Unconnected Message Manager, e.g. the UCMM software module.
- **Connected messaging** allocates resources that are dedicated to a particular purpose, such as real-time I/O data transfers. Connection resources are reserved and configured by the UCMM (Unconnected Message Manager) and then allocated to the desired VMP resource.

Connections are created by a Connection Originator (or Originator for short) and the device that responds to the connection request, called the Connection Target (or just Target).

EtherNet/IP™ specifics two types of message connections:

- **Explicit message connections** are point-to-point relationships. These would be considered general purpose connections and are used to access general information about the VMP. E.g. revision number, serial number, etc. Explicit messages almost always use TCP/IP connections to communicate.
- **Implicit (I/O data) connections** are created to communicate application-specific I/O data at regular intervals. Implicit messages use UDP/IP resources to communicate.

The VMP uses Explicit messages to create a connection with the other devices (e.g. a PLC's etc.) and when requested creates an implicit connection to periodically return the current status of the VMP. E.g. busy, motor enabled, current program etc.

For further information and documentation on message types etc. see OVDA publication PUB00138_EtherNetIP_Ethernet_Technology.pdf.

VMP Configuration for Ethernet/IP™

Other than configuring the correct Ethernet (TCP/IP) parameters on the VMP, no other configuration is required for Ethernet/IP™ communications. Taking note of the VMPs' I/P Address, netmask and gateway, (see Table 2 - Factory Default Network Configuration) are required to configure the devices that communicate with the VMP.

PLC Configuration for Ethernet/IP™

There are two ways to configure an Allen Bradly PLC to communicate with a VMP.

1. Download the EDS (Electronic Data Sheet) file from the Zaxis, Inc. web site (http://zaxisinc.com/service/). Then follow the steps described below in the Configuration with EDS file section below.
2. Add a generic EtherNet I/P module to your project and configure the module using the steps described below. See the Generic Ethernet Module configuration below.

Configuration with EDS file

Once the EDS file, and any other accompanying Logix user defined types have been downloaded from the Zaxis, Inc. web site the EDS file should be registered using the Rockwell Automation's EDS Wizard. A number of resources can be found on the Internet showing the proper procedure to register an EDS file.

Once the VMP EDS file has been registered the configuration can be done as follows:

1. Start Logix Designer and Ensure the Communications are offline.
2. Create a new VMP Ethernet module by right clicking on the **Ethernet** menu item displayed on the **Controller Organizer** window (Figure 5) and select **New Module…** which should display the **Select Module Type** window (Figure 6).

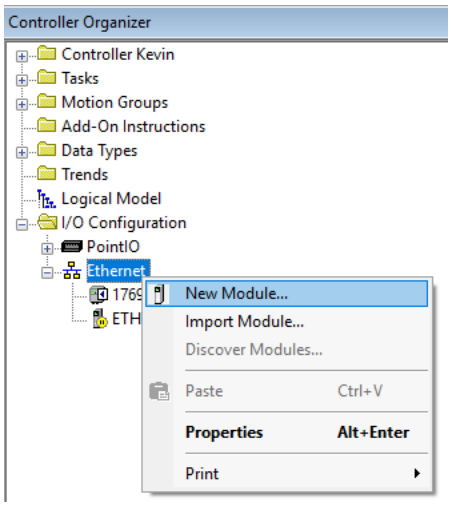


Figure 1 – Create New Ethernet-Module

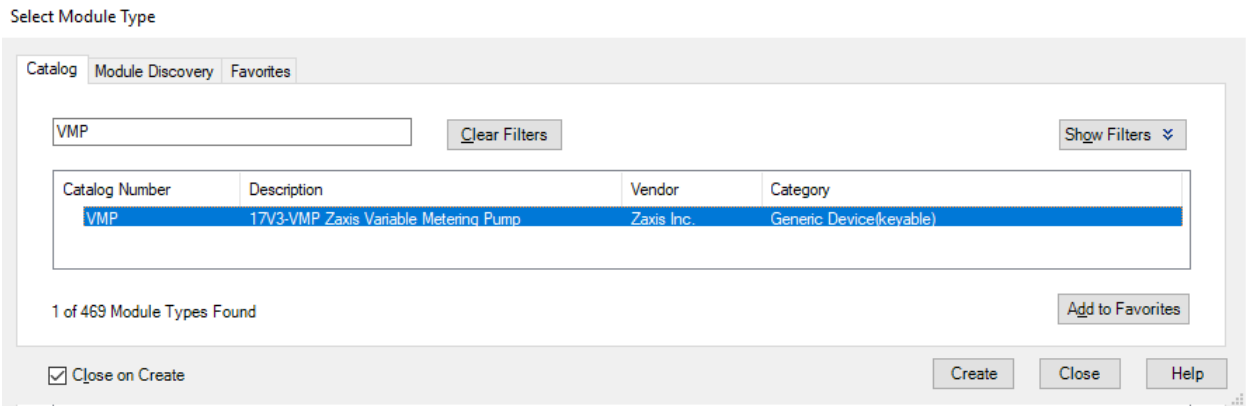
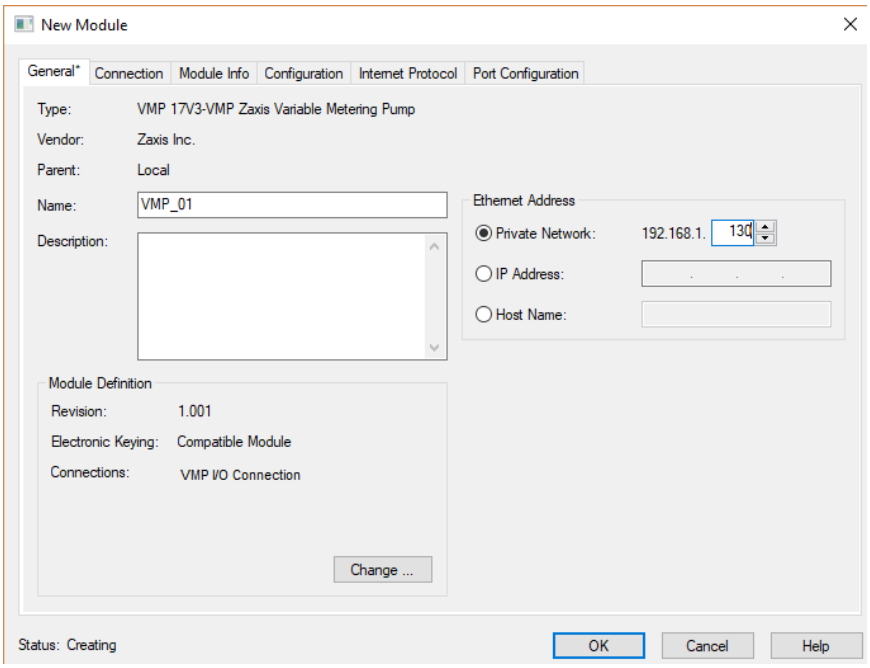


Figure 2 - Select Zaxis VMP Module

3. Click the Create button to create the VMP module, which should display the following window.



4. Assign and I/P address and name to the newly created module.
5. Clicking on the **Change …** button will allow the viewing of the data type for the Input and Output messages to/from the VMP and the current revision of the module.

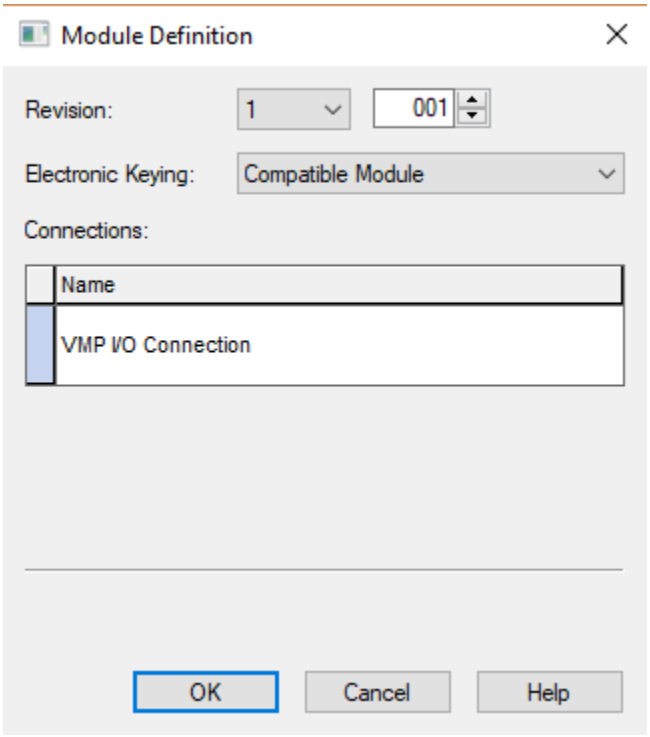
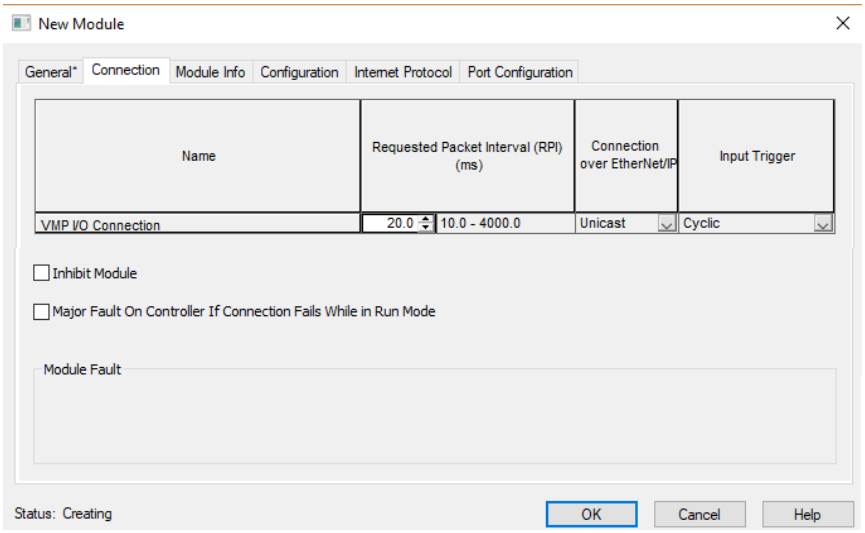
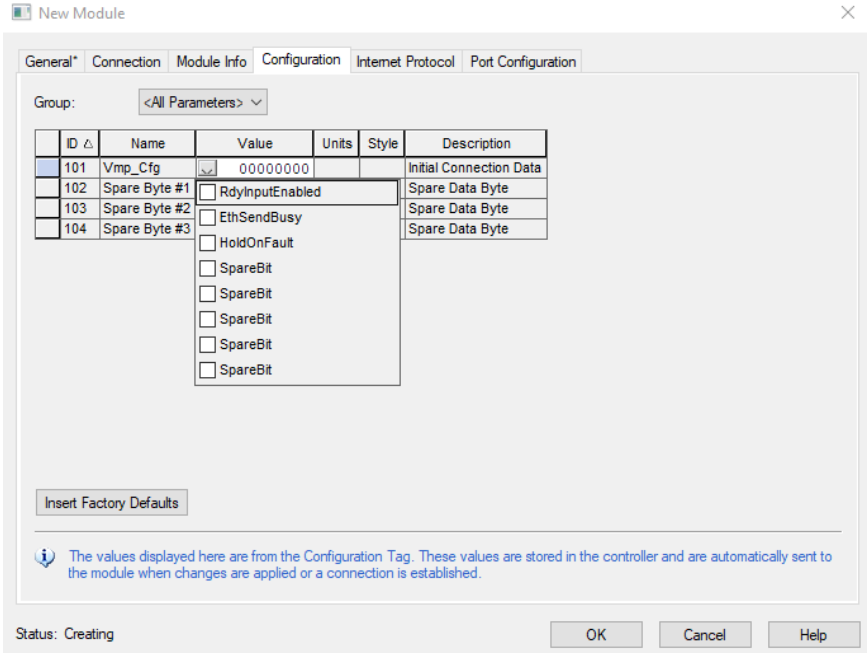


Figure 3 - Assign DINT for Data type

6. The Connection tab allows the RPI value to be changed, the default value of 20.0 milliseconds is the recommended value.



7. The Configuration tab allows the changing of the configuration parameters to be modified. The Configuration parameters are sent to the VMP whenever the PLC creates an I/O connection to the VMP. The purpose of these bits is described below.



8. Clicking the **OK** button will finish the creation of the module, causing the newly created item to appear, as shown below in Figure 4 below.

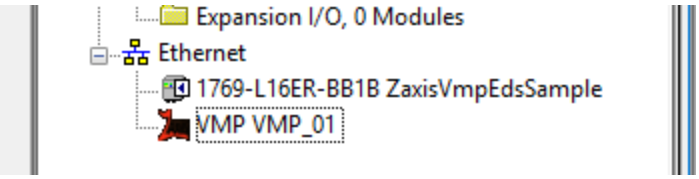


Figure 4 - Zaxis VMP Ethernet I/P Module

Other configuration items can be completed by double clicking on the newly created module and setting the desired parameters etc.

Generic Ethernet Module configuration

Configuring the Allen-Bradley 1769 CompactLogix Controller for communications with the VMP via EtherNet/IP™ consists of creating a new *Generic Ethernet Module*. Use the following steps.

9. Start Logix Designer and Ensure the Communications are offline.
10. Create a new ETHERNET-MODLE by right clicking on the **Ethernet** menu item displayed on the **Controller Organizer** window (Figure 5) and select **New Module…** which should display the **Select Module Type** window (Figure 6).

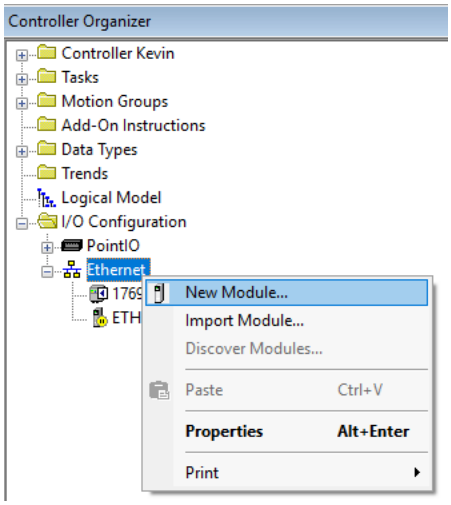


Figure 5 - Create New Ethernet-Module

11. With the **Select Module Type** window use the filter text box to find a **Generic Ethernet Module** as show below. Then click the **Create** button.

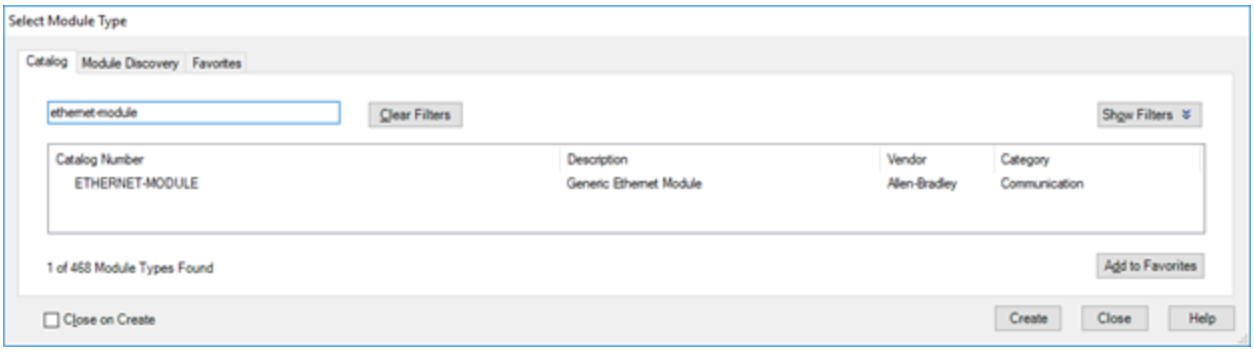


Figure 6 - Select Generic Ethernet Module

12. After clicking on the **Create** button the **New Module** information can be filled in.
    a. Enter a valid name
    b. Enter a description
    c. Ensure the **Comm Format** is set to **Data - DINT**
    d. Enter the **IP address** of the VMP (default I/P is: 192.168.1.130).
    e. Enter **1** for the **Input** Assembly Instance with a Size of **8**.
    f. Enter **2** for the **Output** Assembly Instance with a Size of **8**.
    g. Enter **4** for the **Configuration** with a Size of **4** (or 0 if Configuration data is not desired).

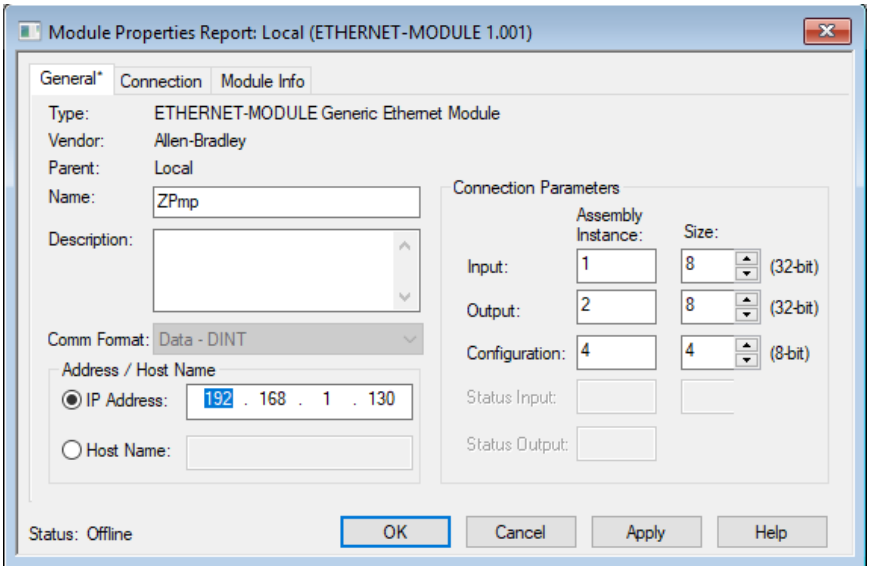


Figure 7 - New Ethernet module properties

The **Configuration** size is optional, it can be 0 (zero) or 4. When zero no initial configuration will be done when communications to the VMP is established. When set to 4 a block of 4 bytes will be sent to the VMP, which are used to do initial configuration. See Configuration Tags below.

13. Clicking the **OK** button creates the new module as well as the necessary controller tags used to interface with the VMP and displays the **Connection** tab of the newly created module as seen in

Figure 8 below. Set the **Requested Packet Interval (RPI)** to 20 e.g. 20 milliseconds second, or as close to the number of milliseconds required for the VMP to complete one cycle or loop. If this value is too low the VMP becomes less responsive, if too high the idle time of the VMP becomes excessive. Th **RPI** should be adjusted after a significant number of test runs have been completed and a more accurate value can be determined.

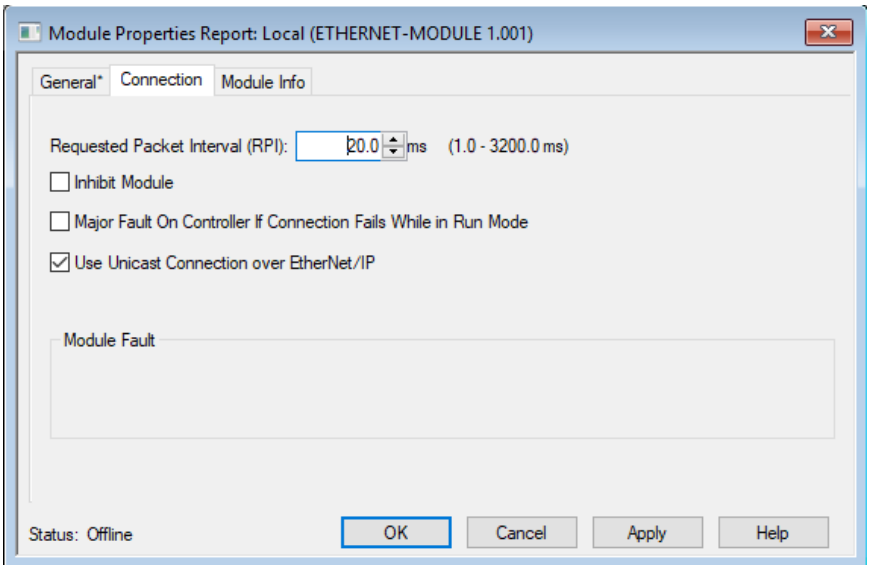


Figure 8 - Setting the RPI

14. Click the **Apply** button to save the properties of the newly created module.

15. When the module is online, clicking on the **Module Info** tab should request basic information from the VMP as shown below in Figure 9. Clicking on the **Module Info** also tests the ethernet connection parameters to ensure they are correct. If an error is returned after clicking on the Module Info, double check the IP address of the VMP and the IP address entered in Step 12 above to ensure they are the same.

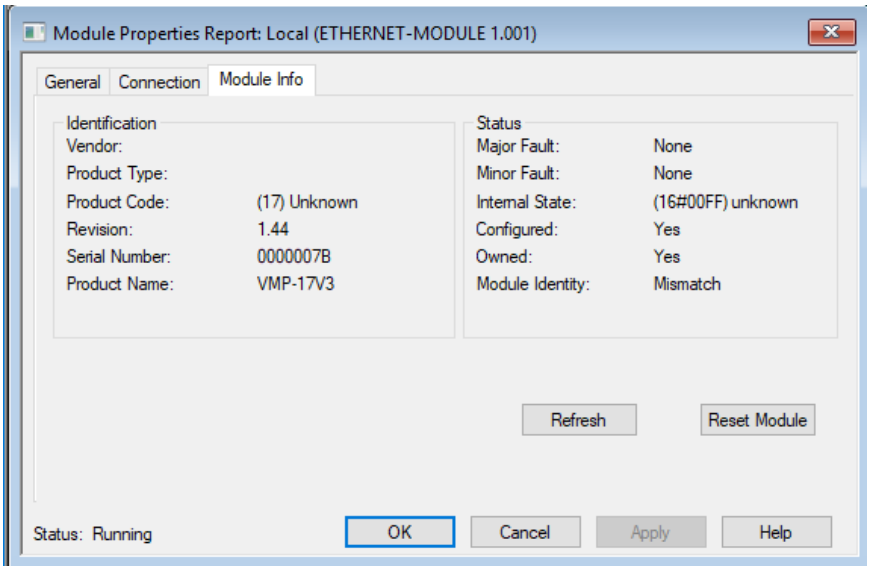


Figure 9 - Module Information

The Vendor field will be blank until the Zaxis vendor Id has been published by OVDA and accepted and implemented by Rockwell Automation Technologies, Inc. The serial number is given as a hex number, but when converted to decimal, should match the leading digits of the serial number as seen on the TSi screen. The alpha-numeric characters after the **VMP-** in the **Product Name:** should match the trailing alpha-numeric characters of the serial number as seen on the TSi screen.

16. Open the **Controller Tags** to display the newly created tags for the ZPmp module as shown below.

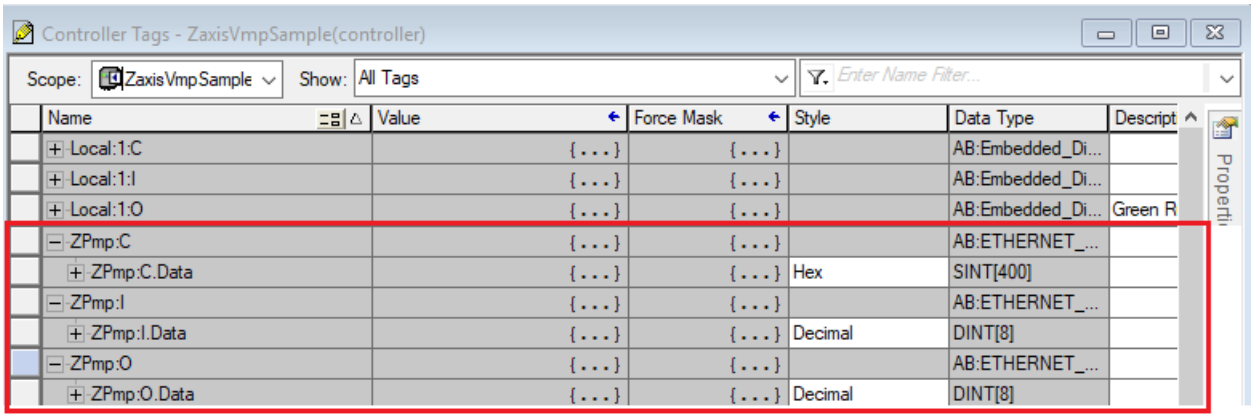


Figure 10 - VMP controller tags

17. As can be seen in Figure 10 three new tag groups have been created which correspond to the Assembly Instances seen in Figure 7. ZPmp:C corresponds to the Configuration assembly

instance, if the size of the Configuration instance is zero the ZPmp:C tag group will not exist. Tag group ZPmp:I correspond to the Input instance and ZPmp:O to the Output instance.

VMP Tag Groups

The tag groups for the VMP will always have the Input and Output tags, the optional Configuration tags will only be available when a size other than zero is placed in the **Size** field of the Configuration Instance. The following tables: Table 3, Table 4 and Table 5, represent the structure of the different tag groups.

As with other tag groups, the controls engineer may create User Defined data type (UDT's) that reference the VMP tags as desired.

Configuration Tags

The configuration tags are used to initialize selected parameters within the VMP. These tags are not required and will not be sent to the VMP if the size of the Configuration instance is set to zero. If the Configuration size is non-zero, it must be set to the size indicated in step 12.g above, or undesirable results will occur.

Configuration tags are sent to the VMP when the communications channel is opened, e.g. when the Ethernet module comes online.

Name	Data Type	Valid Values	Description
ReadyInterlock	BOOL	O or 1	The Ready Input bit is enabled.
EthernetRunReady	BOOL	0 or 1	Causes status messages to be sent on Ethernet port 23 when opened.
HoldOnFault	BOOL	0 or 1	Do not allow the pump to run until the fault bit has been cleared.
SpareBits	BOOL	0	Bits 3 – 15 are spare bits
DecPlaces	SINT	0 to 5	Number of implied decimal points for VolumeDint
Spare	SINT[2]	0	Spare bytes for future use.

Table 3 – Ethernet-Module Configuration Tags

The TSi (touch screen interface) has input fields with the same or similar names to the configuration tags, more detailed information as to the purpose of these tags may be found in the VMP User's Manual. These configuration tags preform the same functions as the TSi fields.

It is possible to set, or write, a subset of tags. When an invalid value is given, such as a -1, the given tag parameter will be ignored. For example, if a -1 is given for the *Program* the current program will not be changed.

PLC Output Tags

Output tags are sent by the PLC to the VMP every **Request Packet Interval (RPI)** milliseconds as configured above, see Figure 8 - Setting the RPI. Once the VMP scans i.e. process, the output tags from the PLC, the input tags are sent from the VMP to the PLC.

Name	Data Type	Valid Values	Description
OkToRun	BOOL	0 or 1	OK to run program. Same as the discrete I/P Ready In bit.
Start	BOOL	0 or 1	Start running the current program. Triggers on leading edge.
Abort	BOOL	0 or 1	Abort the currently running program.
Direction	BOOL	0 or 1	0 = The pump head is turning clockwise, 1 = counter clockwise.
ContinuousRun	BOOL	0 or 1	Run the motor in continuous run mode.
SetMaxShotSize	BOOL	0 or 1	Adjust the pump to deliver the maximum shot size.
SetShotSize	BOOL	0 or 1	Adjust the pump to deliver the indicated volume size.
IgnorePLC	BOOL	0 or 1	Ignore all commands but the Abort command from the PLC.
IncFineAdj	BOOL	0 or 1	Increment the fine adjustment
DecFineAdj	BOOL	0 or 1	Decrement the fine adjustment
SpareBits	BOOL	0	Spare bits.
Program	DINT	0 to 49	Current selected program.
Revolutions	DINT	1 to 9999	Number of revolutions for this program.
Volume	REAL	0 to 2000	Volume for each shot.
Rpm	DINT	60 to 200	Revolutions of the pump head in one minute.
SpareDint1	DINT	0	Spare data
SpareDint2	DINT	0	Spare data
FineAdj	INT	0 to 500	Value to increment/decrement fine adjustment. Default = 1
SpareInt1	INT	0	Spare data

Table 4 - Output Tags

Again, the tag names are similar or the same as the TSi field names, or buttons, e.g. Start In, Abort Run and Continuous Run are buttons on the TSi.

The OkToRun bit preforms the same function as the Ready In bit as discussed in the VMP User's Manual. The Start bit must transition from 0 to 1 before the current program will start.

The FineAdj parameter is used to set the number of fine adjustment steps used to fine tune the volume. A value of zero is the same as one.

Input Tags

Input tags are sent to the PLC from the VMP after the Output tags have been scanned, or processed, by the VMP. Input tags provide the current state of the VMP to the PLC. The following table list the tags returned from the VMP.

Name	Data Type	Valid Values	Description
Ready	BOOL	0 or 1	The VMP is ready to run the current program.
Running	BOOL	0 or 1	The pump is currently running or is busy.
Abort	BOOL	0 or 1	The pump is currently in the Abort state, e.g. not running.

Direction	BOOL	0 or 1	0 = The pump head is turning clockwise, 1 = counter clockwise.
ContinuousRun	BOOL	0 or 1	The motor is in continuous run mode
MotorRunning	BOOL	0 or 1	One of the two motors are running.
AtMaxShot	BOOL	0 or 1	The pump will deliver the max size shot available.
CmdAck	BOOL	0 or 1	Last PLC command is acknowledged by the VMP
BadVolume	BOOL	0 or 1	Invalid volume was requested. E.g. greater than Max volume.
BitsSpare	BOOL	0	Spare and or Reserved bits. Bits 9 thru 28
FaultDispenseMotor	BOOL	0 or 1	Dispense motor is in a fault state.
FaultAdjustMotor	BOOL	0 or 1	Adjust motor is in a fault state.
FaultAdjustMotor	BOOL	0 or 1	The adjust motor position sensor is in a fault state.
Program	DINT	0 to 49	Current selected program.
Revolutions	DINT	0 to 9999	Current value of revolutions.
Volume	REAL	0 to 9.99999	Current volume or shot size the pump is delivering.
Rpm	DINT	0 to 9.99999	Revolutions of the pump head in one minute.
VolumeDint	DINT	0 to 999999	Volume as DINT, with implied decimal places. See DecPlaces in Configuration Tags.
SpareDint2	DINT	0	For future use
FineAdj	INT	0 to 9999	Current fine adjustment value
SpareInt1	INT	0	Spare Int

Table 5 - Input Tags

The Ready bit preforms the same functions as the Ready input signals as described in the VMP User's Manual, e.g. the pump is ready for work.

The CmdAck bit is set to 1 when the pump trying to complete a command from the PLC and set to 0 when the command has been completed. That is, if any of the following bits are active (Running, SetMaxShotSize, IncFineAdj, DecFineAdj, SetShotSize the CmdAck bit is high. Thus, allowing the PLC to know when the command has been completed and the pump is ready for the next command that would cause the motor to run.

The VolumeDint is the volume with an implied decimal place (e.g. volume * 1000). The number of implied digits can be between 0 and 5. See the DecPlaces field in Configuration Tags.

Pump Setup/Program parameters

The VMP has two different types of setup parameters: 1) parameters that remain the same regardless of which program is running and 2) parameters for each individual program. This document will refer to these different types as setup paraments and program parameters respectively.

Regardless of the type of parameter (setup or program) both are saved in flash memory and are read into RAM (random access memory) each time the pump is powered up and/or reset. The flash memory doesn't require a battery and is therefore more reliable for longer periods of time.

This also implies that these parameters do not (and should not) be written to the VMP each time a program (loop) is run. Doing so causes unnecessary overhead and slows the VMP's response times causing timeout problems.

Setup Parameters

Setup parameters are those parameters that remain the same regardless of which program (or loop) is being executed. Setup parameters are typically set once on initial setup of the VMP and very seldom changed. The following table describes the Setup parameters.

Name	Data Type	Valid Values	Description
ReadyInterlock	BOOL	0 or 1	This bit must be set before the pump can run.
EthernetRunReady	BOOL	0 or 1	Causes status messages to be sent on Ethernet port 23 when opened.
HoldOnFault	BOOL	0 or 1	Do not allow the pump to run until the fault bit has been cleared.
Bits 3 thru 31	BOOL	0	Spare bits
Baud	DINT	1152	Baud rate for the RS485 bus.
Pump Number	DINT	1 to 128	RS485 bus pump number.
SpareDint0	DINT	0	Spare Data
SpareDInt1	DINT	0	Spare Data
FirmwareRev	REAL	Read Only	Current pump Firmware revision.
Bore	REAL	0.125, 0.1875, 0.25, 0.375, 0.5, 0.5096, 0.6250	Bore size of the piston in inches. Micro pumps must have a bore less than or equal to 0.25 and VMP type pumps must have a bore size equal to 0.25 or greater.
SpareReal0	REAL	0	For future use
SpareReal1	REAL	0	For future use
Serial Number	STRING	Read only	Serial number of the pump.

Table 6 - Configuration Parameters

The above configuration parameters are typically used to create a **User-Defined** data type for easy access as show below in **Error! Reference source not found.**.

Reading and Writing Setup Parameters

Setup parameters can be read and written using the PLC Message (MSG) instruction.

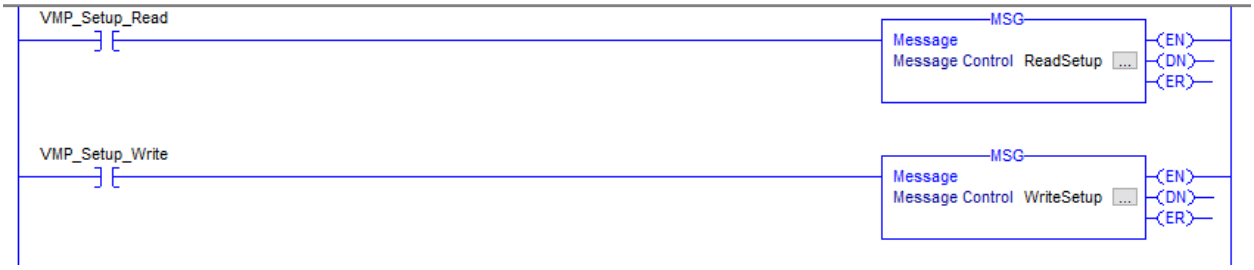


Figure 11 - MSG Tag

Setup parameters are read using the CIP Read All Attributes message and written using the CIP Write All Attributes message. These CIP messages (Read/Write All Attributes) are created on the PLC using the *CIP Generic* **Message Type** with a **Service Code** of 1 (e.g. Read all attributes) or 2 (e.g. Write all attributes), a vendor specific **Class** code of 0x64 (Hex) and an **Instance** of 1. As show below in Figure 12 and Figure 15 below.

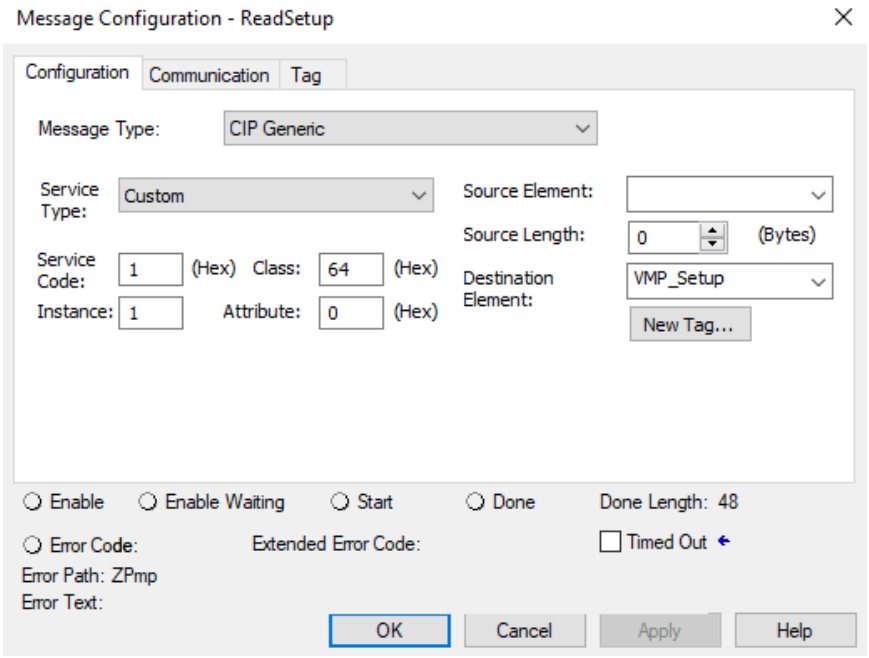


Figure 12 - Read VMP Configuration Parameters MSG

The **Destination Element** should be a *User-Defined* data type which minors the tags described in Table 6 and shown in **Error! Reference source not found.**.

The *Communication* tab references the **Path** of the VMP Ethernet module that was added above in the VMP Configuration for Ethernet/IP™ section, as show below in Figure 13.

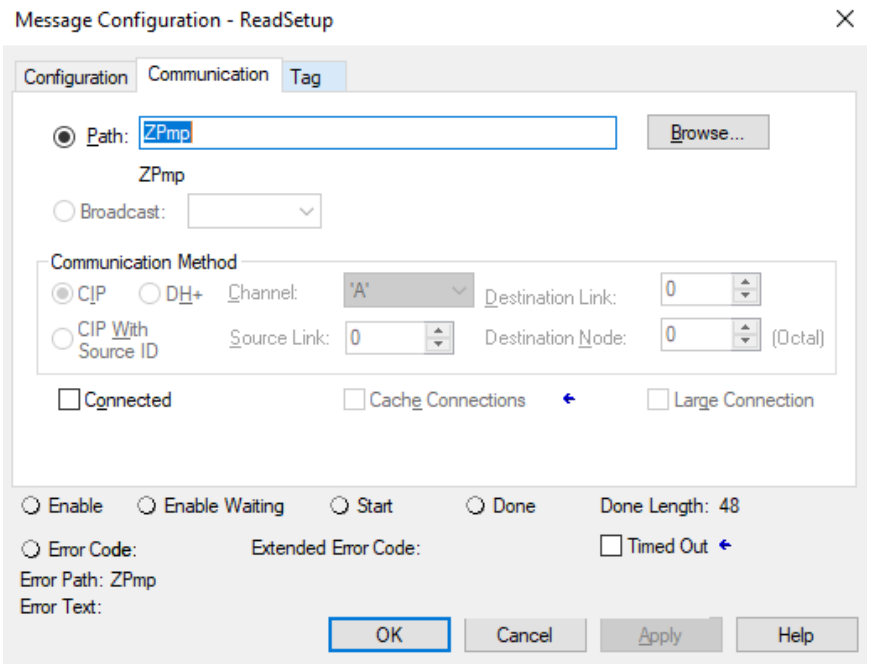


Figure 13 - MSG Communication tab

The MSG Tag can be given any meaningful name as shown below Figure 14 below to complete the configuration.

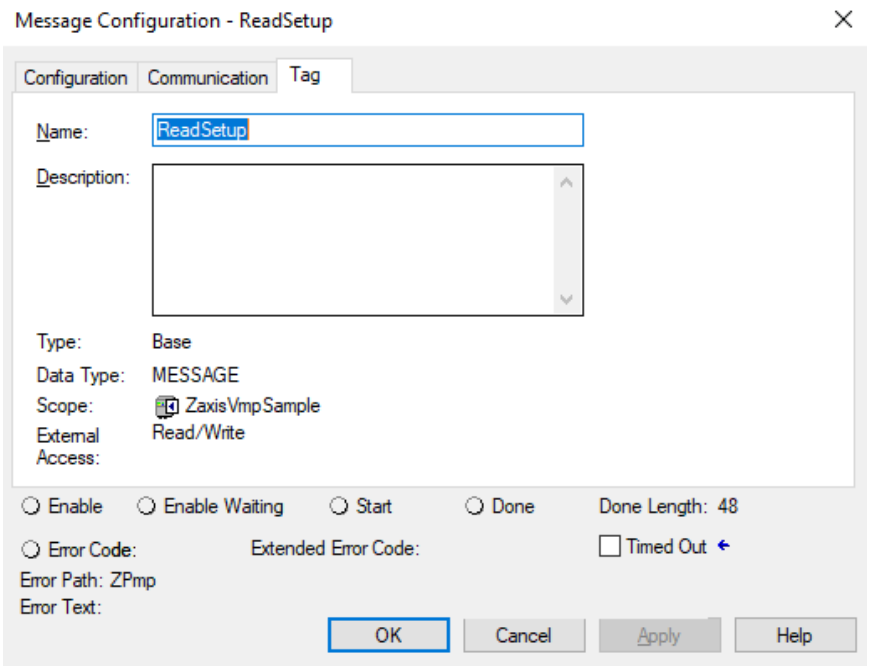


Figure 14 - MSG Tag tab

The only difference between reading and writing setup parameters is the **Service Code**, it changes from 1 to a 2, and the **Source Element** now references the user defined data to be written. The MSG parameters for writing are show in Figure 15 below.

The *Communication* tab and *Tag* tab have the same parameters as reading the setup.

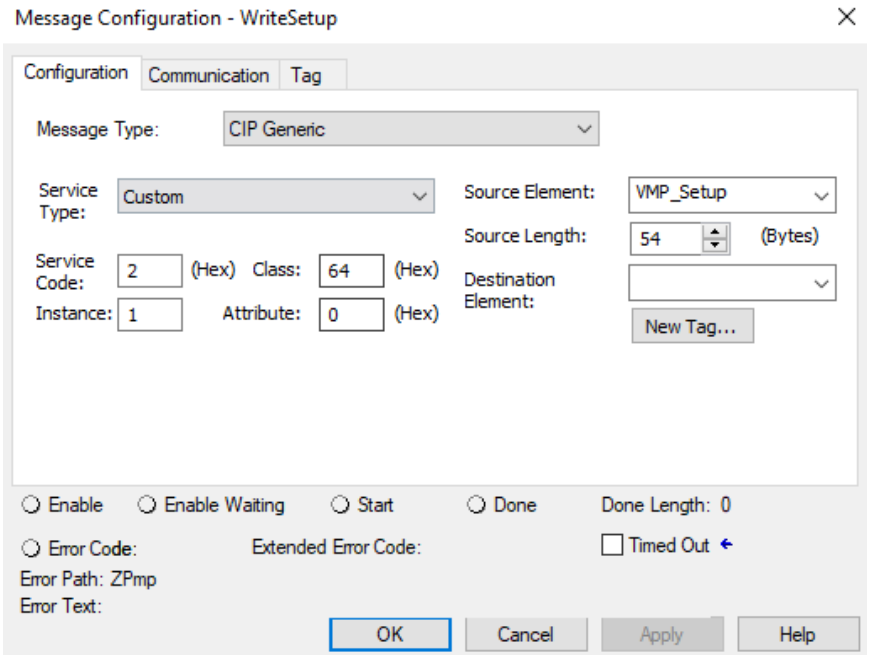


Figure 15 - Write VMP Configuration Parameters MSG

Program Parameters

Program parameters allow the VMP to be configured such that different programs or loops can be run by the same VMP yet perform completely different tasks. Programs can be linked allowing one program to execute after the parent program, with each program having different parameters. Table 7 shows the different program parameters and their meaning.

Name	Data Type	Valid Values	Description
CalDispenseTime	BOOL	0 or 1	Calculate Dispense Time which is returned in AprxDispenseTime.
Direction	BOOL	0 or 1	0 = The pump head should turn clockwise, 1 = counter clockwise.
SetShotSize	BOOL	0 or 1	Set the shot size after writing this program.
ContinuousRun	BOOL	0 or 1	Program should run continuously.
Bits 4 thru 31	BOOL	0	Spare bits.
Program	DINT	0 to 49	Program number to write.
PullBack	DINT	0 to 100	Percentage rotation to reverse the pump after a shot. E.g. suck back.
InitialRPM	DINT	60 to 200	Initial speed of the pump.
RPM	DINT	60 to 200	Speed of the pump after initial RPM.
FinalRPM	DINT	60 to 200	Final speed of the pump after RPM.
FineAdjust	DINT	1, 0, -1	1=Increment, 0=Clear, -1=Decrement fine adjust.
Acceleration	DINT	60 to 200	Set speed of acceleration for initial RPM.
Deceleration	DINT	60 to 200	Set speed of deceleration for final RPM.
LoopCycles	DINT	1 to 999999	Number of times to run the linked program.
LinkedProgram	DINT	-1 to 49	-1=Disabled, 0 to 49 next programs to run.
LinkedTimeDelay	DINT	0 to 9999	Number of seconds to wait before running the linked program.
MotorCurrent	DINT	0 to 100	Percentage current used to drive the motor.
Revolutions	DINT	1 to 9999	Number of revolutions to turn the pump head.
Approx. Time	DINT	Read only	Approximate time in mill-seconds for a dispense time.
Volume	REAL	0 or more	Shot Volume.
Program Name	STRING	Any printable character	Name of the current program.

Table 7 - Program Configuration Parameters

As with the configuration parameters, the program parameters should be used to define a **User-Define** data type as show in Table 8.

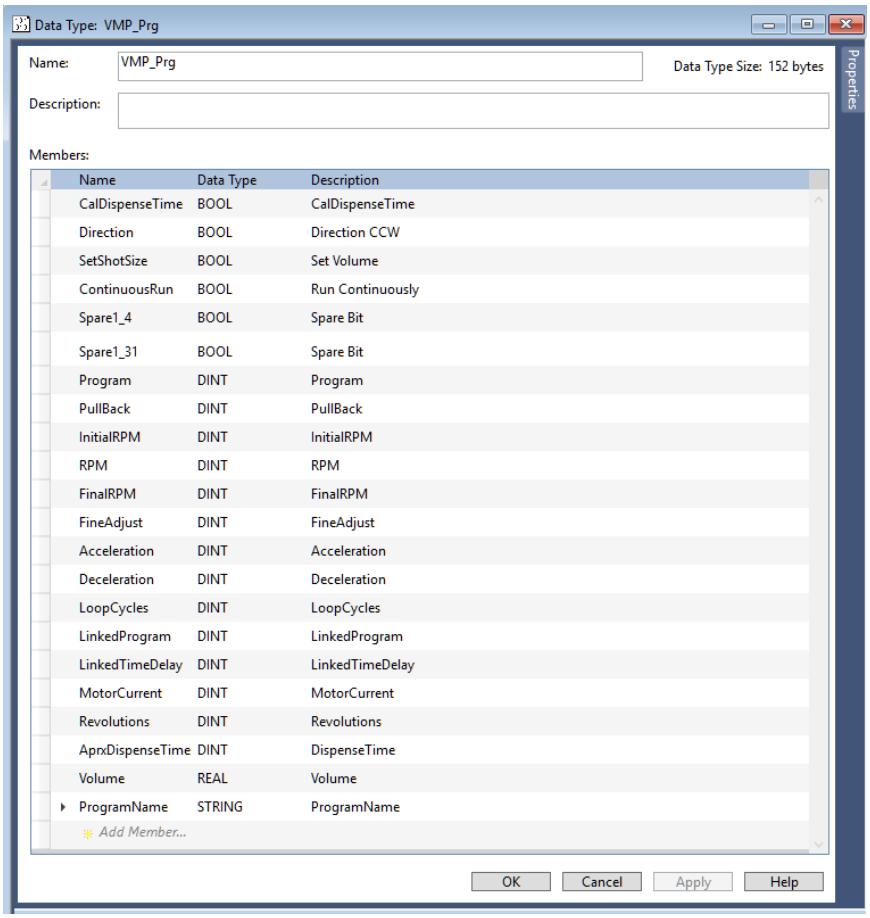


Table 8 - Program Parameters

Program parameters are read using the CIP Read All Attributes message and written using the CIP Write All Attributes message. These CIP messages (Read/Write All Attributes) are configured on the PLC using the *CIP Generic* **Message Type** with a **Service Code** of 1 (e.g. Read all attributes) or 2 (e.g. Write all attributes), a vendor specific **Class** code of 0x65 (Hex) and an **Instance** of 1. As show below in Figure 17, Figure 18 and Figure 18.

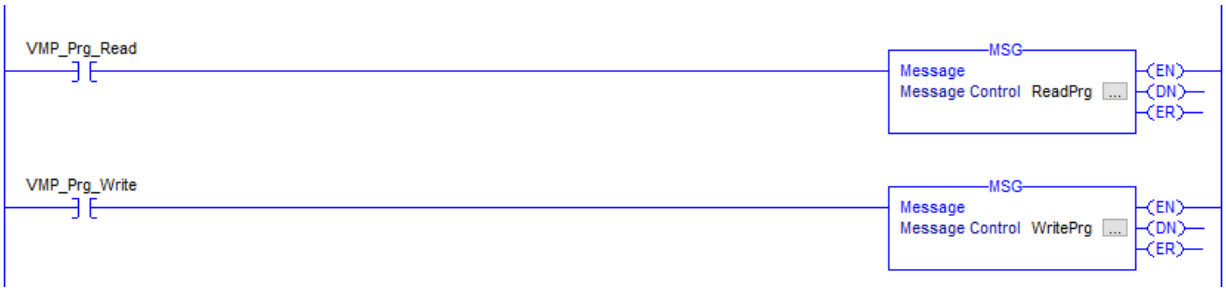


Figure 16 - Reading/Writing program Ladder Logic

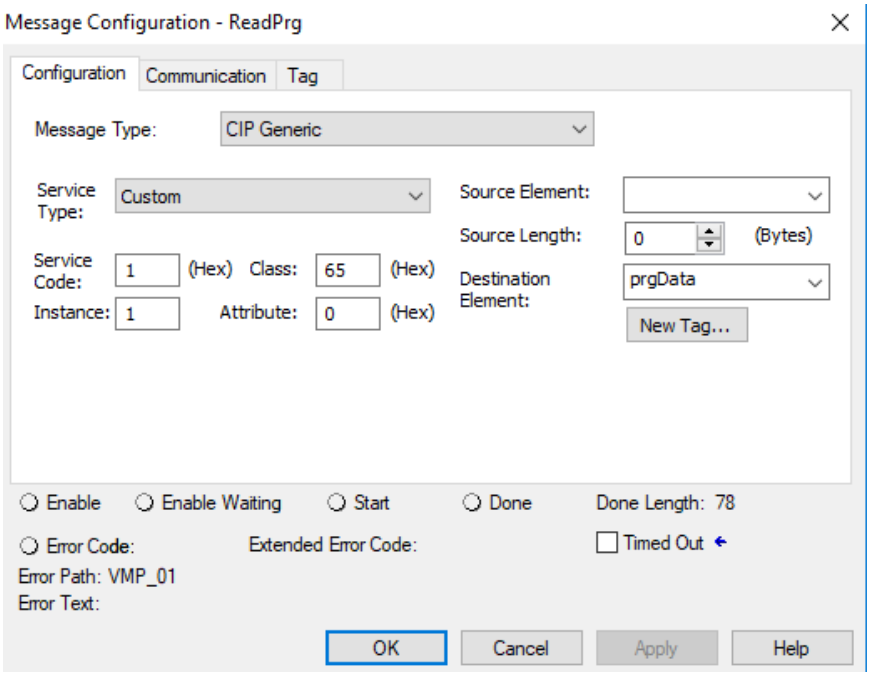


Figure 17 – Read VMP Program Parameters MSG

Writing Program Parameters is the same as reading with the acceptation of the **Service Code** which should be changed to 2 e.g. Write All Attributes, as show in Figure 15 below.

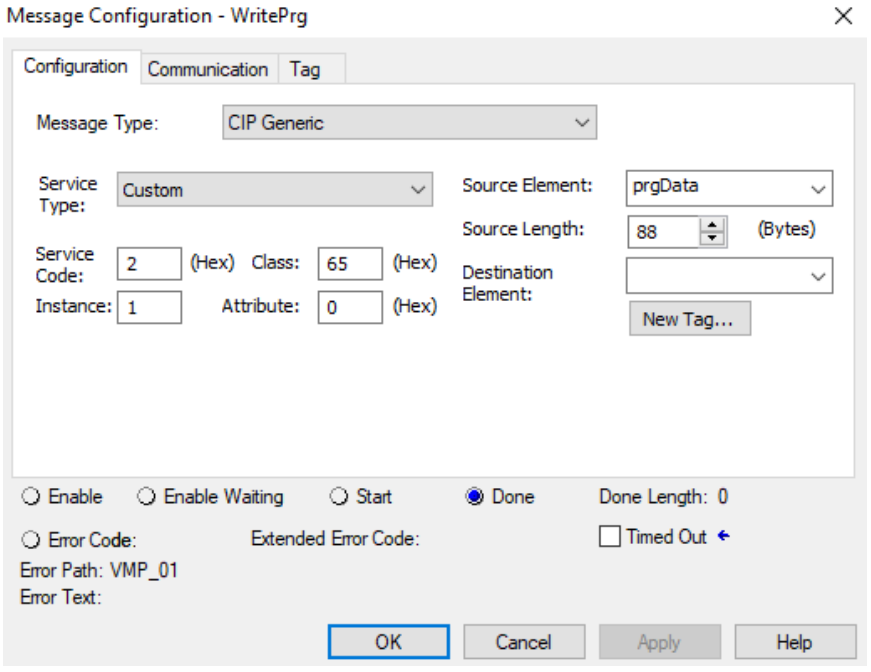


Figure 18 - Write VMP Program Parameters MSG

TCP/IP Network Configuration

The default TCP/IP settings are provided in Table 2 above, however these values can be reconfigured using the predefined CIP TCP/IP object. The object definition is shown in Table 9, and is based on the *TCP/IP Interface Object* in *Volume 2: EtherNet/IP Adaptation of CIP, Chapter 5: Object Library*.

Name	Data Type	Valid Values	Description
Status	DINT	Read only	Interface Status bits. 0x02=Configured from DHCP, 0x04=Hardware configuration.
Configuration Capability	DINT	Read only (0x34)	Bit map of interface capability flags: 0x04=DHCP capable 0x10=Interface configurable 0x20=Hardware configurable
Configuration Control	DINT		Interface control flags: 0x00=Use static configuration 0x02=Use DHCP configuration
Path Size	INT	Read only (3)	Number of 16-bit words in the following padded EPATH.
Path	SINT[4]	Read only	Padded EPATH to physical link object.
IP Address	DINT	0 to 0xFFFFFFFF	Devices I/P address.
Network Address	DINT	0 to 0xFFFFFFFF	Devices network address.
Gateway Address	DINT	0 to 0xFFFFFFFF	Devices gateway I/P address.
Name Server	DINT	0 to 0xFFFFFFFF	Primary name server I/P address.
Name Server 2	DINT	0 to 0xFFFFFFFF	Secondary name server I/P address.
Domain Name	STRING	ASCII Characters	Default domain name.
Host Name	STRING	ASCII Characters	Default host name.
Safety Network Number	SINT[6]	0 to 100	Ignored by VMP.
TTL	SINT	1 to 255	Time-to-Live value for IP Multicast packets. Ignored by VMP.
Alloc Control	SINT	0	Multicast address allocation. Ignored by VMP.
Reserved	SINT	0	Reserved for future use.
Num Mcast	INT	0	Number of multicast addresses to allocate. Ignored by VMP.
Mcast Start Addr	DINT	0	Starting multicast address. Ignored by VMP.
Selected Acd	BOOL	0	Activates use of ACD. Ignored by VMP.
Acd Active	SINT	0	State of ACD activity. Ignored by VMP.
Remote MAC	SINT[6]	0	MAC Address of remote node. Ignored by VMP.
Arp Pdu	SINT[28]	0	ARP PDU. Ignored by VMP.
EtherNetIP Quick Connect	BOOL	0	Quick connect feature. Ignored by VMP.

Encp Inactivity Timeout	INT	0	Inactivity Timeout. Ignored by VMP.

Table 9 - TCP/IP Object definition

The above definition can be used to create a **User-Defined** data type that would look like Figure 19.

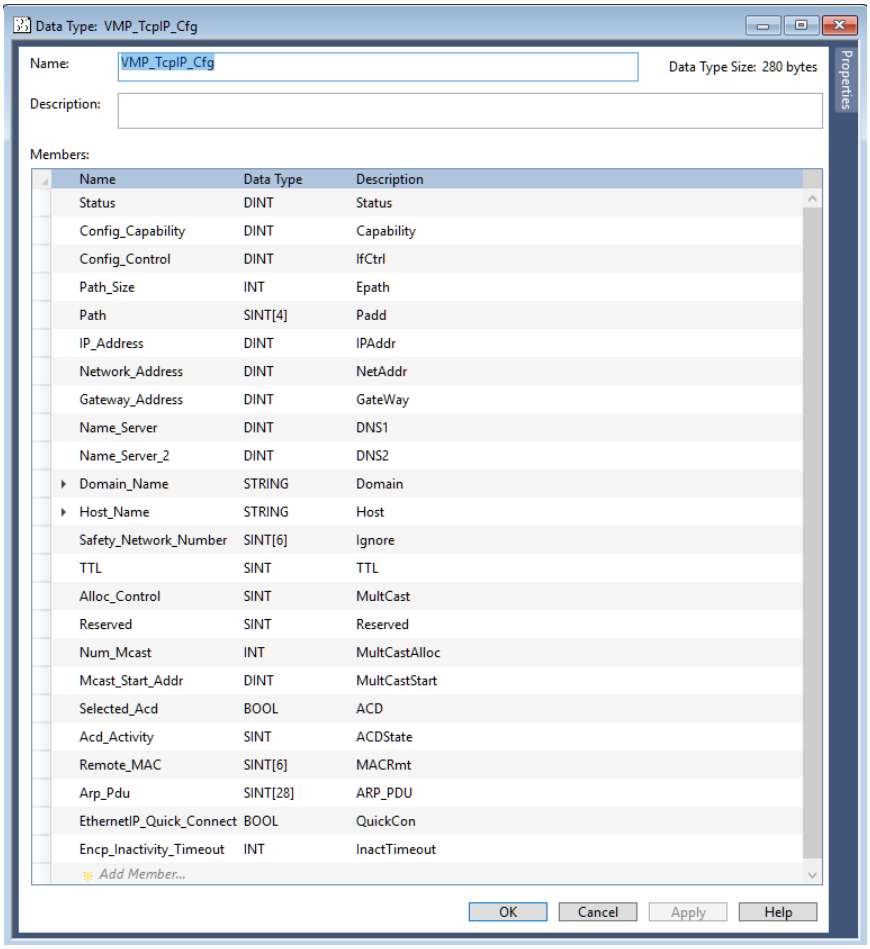


Figure 19 - User define TCP/IP Object

As can be seen from the table above many of the parameters are not required by the VMP. However, this object does allow the reconfiguration of necessary TCP/IP parameters, e.g. I/P address, enable/disable DHCP etc.

TCP/IP parameters are read using the CIP Read All Attributes message and written using the CIP Write All Attributes message. These CIP messages (Read/Write All Attributes) are configured on the PLC using the *CIP Generic* **Message Type** with a **Service Code** of 1 (e.g. Read all attributes) or 2 (e.g. Write all attributes), a CIP **Class** code of 0xF5 (Hex) and an **Instance** of 1. As show in FIGURE 20 and Figure 21.

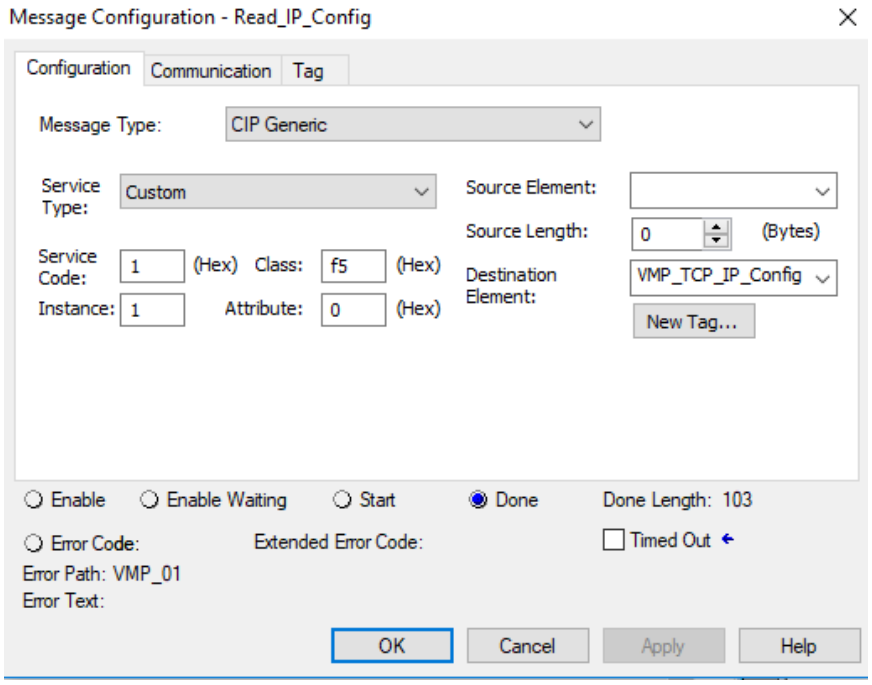


FIGURE 20 - Read TCP/IP Configuration Parameters

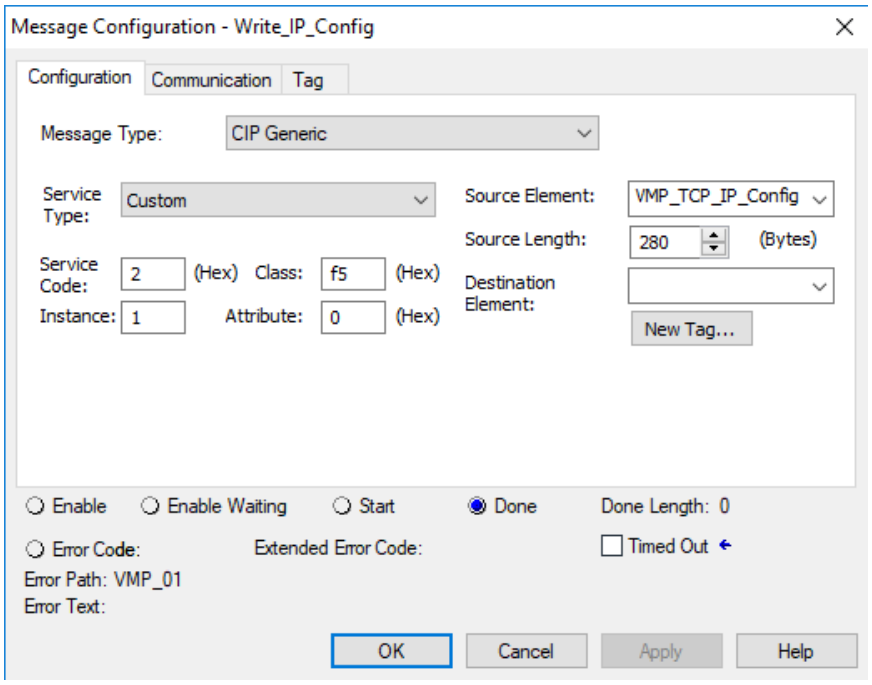


Figure 21 - Write TCP/IP Configuration Parameters

For example, to configure the VMP to use DHCP, simply set the Configuration Capability field to 0x02 and write the TCP/IP object using an MSG instruction.

To set anyone of the I/P addresses (e.g. I/P address, network address, gateway address, etc.) convert each address octet into hex and enter the hex number into the desired field. For example, to set the I/P address of the VMP to 192.168.168.100 enter the hex value 0xC0A8A864 into the IP Address field and write the TCP/IP object to the VMP.

Decimal:	192	168	168	100
Hex:	0xC0	0xA8	0xA8	0x64

Table 10 - I/P Address to Hex conversion example

The VMP default TCP/IP configuration can always be restored by pressing and holding the reset button for less than two seconds. If the button is held for longer than two seconds the VMP will use DHCP to configure TCP/IP.

Trouble Shooting

When problems occur, either in setting up the communications, or communicating with, the VMP, sometimes it is helpful to get more detailed than a simple fault error. The VMP displays error and status messages when a network connection has been established on port 8080. These diagnostic messages may give more detail and/or a better understanding of the problems.

A connection to the VMP diagnostics console can be created using the *Telnet* utility, available on all Unix based system and most Windows system. However, it may need to be turned on under windows. The following figure shows a check mark next to the **Telnet Client**, indicating that it is available for use.

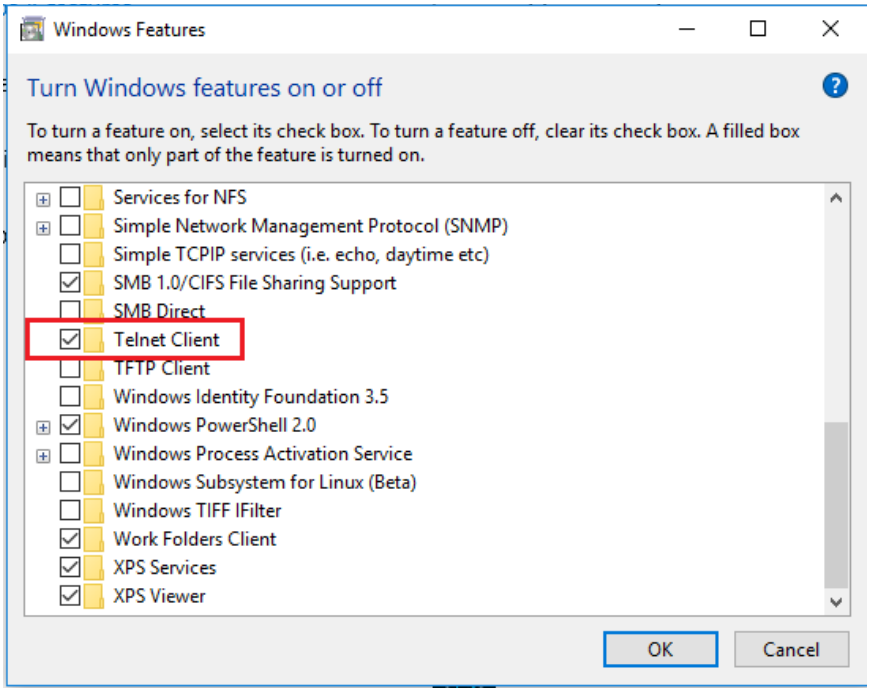


Figure 22 - Windows Telnet Client

With the *Telnet* utility available a connection can, be made using the following command, within a command window (which can be created by "*run cmd*"):

telnet <VMP_IP_address> 8080 e.g. *telnet 192.168.1.130 8080*

Figure 23 shows the output from the diagnostics console. With a line showing a buffer length mismatch, e.g. 34 bytes should be 32 bytes. In this case the error was caused by using a size of 34 for the Configuration Assembly Instance when it should have been 32.

Information can be obtained from the diagnostics console; however, it is recommended that a Zaxis support personal help you in diagnosing and resolving more advanced problems.

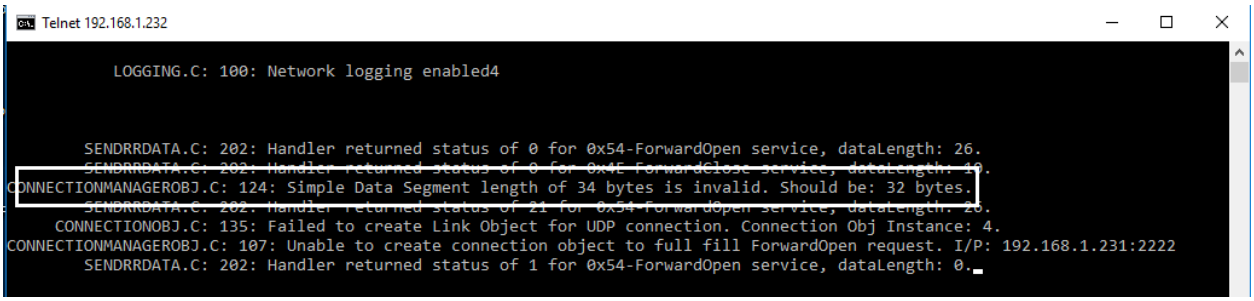


Figure 23 - Diagnostics Console