MAC MI/O 67 Fieldbus Manifold

EtherNET/IP – AOIs

Allen Bradley PLC Software Manual

Manual Version Control

Version	Date	Author	Change Description
5.0	02/15/2020	c.elston	Added New Analog Modules Analog Voltage (MIO67A-AN-01-02) Analog Current (MIO67A-AN-02-02)
4.1	03/07/2019	c.elston	Revised clerical errors and corrections. Added updated pin-out references. Update graphics.
4.0	8/17/2018	c.elston	Revised clerical errors and corrections. Added MIO 67 Digital I/O and Power Plus AOIs
3.0	05/14/2018	c.elston	Added MIO 67 Config, Analog Voltage, Analog Current AOI. Updates to Solenoid Control AOI with more alarming and diagnostics.
2.0	01/03/2018	c.elston	Added MIO 67 Solenoid Control AOI
1.0	10/20/2017	c.elston	Initial Release

AOI Block Version Control

Name	Version	Date	Author	Change Description
MIO_67_Solenoid_Control	2.2	04/11/2018	c.elston	Added fault logic
MIO_67_Solenoid_Control	2.1	01/05/2017	c.elston	Added false enable logic
MIO_67_Solenoid_Control	2.0	12/29/2017	c.elston	Fixed open circuit logic, update desc.
MIO_67_Solenoid_Control	1.0	12/12/2017	c.elston	Initial Release
MIO_67_Config_Modules	3.0	01/25/2020	c.elston	Added Type 12-13 New Analog Modules
MIO_67_Config_Modules	2.0	08/22/2018	c.elston	Updated selections from 0-5 to 0-11 types
MIO_67_Config_Modules	1.0	12/29/2017	c.elston	Initial Release
MIO_67_Analog_Voltage	2.0	01/25/2020	c.elston	Added Configurable (MIO67A-AN-01-02)
MIO_67_Analog_Voltage	1.0	04/28/2018	c.elston	Initial Release
MIO_67_Analog_Current	2.0	01/25/2020	c.elston	Added Configurable (MIO67A-AN-02-02)
MIO_67_Analog_Current	1.0	04/28/2018	c.elston	Initial Release
MIO_67_Power_Plus	1.0	08/25/2018	c.elston	Initial Release
MIO_67_Digital_IO	1.0	08/25/2018	c.elston	Initial Release

TABLE OF CONTENTS

1 INTRODUCTION

This manual is provided as a guide for using MAC MI/O 67 EtherNET/IP Fieldbus Manifold with RS Logix 5000 software by Allen Bradley. This manual provides detailed configuration instructions to configure Ethernet/IP to RS Logix 5000 software projects.

1.1 Definitions, Acronyms, and Abbreviations

Terms	Definition
EDS	Electronic Data Sheet. A text file that contains configuration data for specific device types. The EDS for a device is provided by the device vendor and is required for compliance with ODVA standards
AOP	Add-On Profile. Logix Designer component that can be separately installed and used for configuring one or more modules.
Simple EDS AOP	EDS Add-On Profile. The Add-On Profile that uses EDS content to construct the profile to support various module types.
Profile	A subsystem of Logix Designer. Each supported module type has an associated profile. The profile provides information needed to establish topology and module-defined data types, as well as graphical user interface for configuration (in some cases)
AOI	Add-On Instruction. A vendor specific function block that is imported into RS Logix 5000 programming software to control the vendor supplied hardware.

1.2 Supported PLC Controllers

At this time only, Allen Bradley CompactLogix and ControlLogix PLC CPUs that use RS Logix 5000 software are supported. Sample projects can be downloaded from MAC website.

IMPORTANT NOTICE:
Allen Bradley Micrologix, Micro800, SLC500 or PLC5 PLC CPUs are **NOT supported** using RS Logix 500 software.

1.3 PLCs with Version 19 or earlier firmware

Configuration of EtherNET/IP will be done with a Generic EtherNET/IP Block.
Please see section 4.0 below for more information. Contact manufacturer before using version 19 PLC with MIO-67.

1.4 PLCs with Version 20 or newer firmware

Configuration of EtherNET/IP will be done with MAC provided Simple EDS AOP (Add-on Profile) file.
Please see section 5.0 below for more information.

1.5 Add-On Instructions

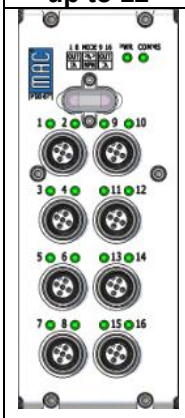
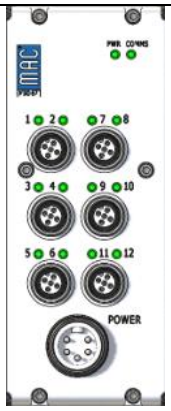
AOI Name	AOI Brief Description
MIO_67_Solenoid_Control	Controls up to 32 solenoids per valve station.
MIO_67_Config_Modules	Used one time, to help configure the valve station with add-on modules.
MIO_67_Analog_Current	Allows access to the Analog Current add-on module.
MIO_67_Analog_Voltage	Allows access to the Analog Voltage add-on module.
MIO_67_Power_Plus	Controls the Power Plus add-on module.
MIO_67_Digital_IO	Allows access to the Digital IO add-on module.

1.6 Supported MAC Valve Types

Bullet Series	42 Series	46 Series
82 Series	92 Series	93 Series

For other valve types, please consult the factory

1.7 Typical Physical Layout of MI/O 67

Add-On Module #4 up to 12	Add-On Module #3	Add-On Module #2	Add-On Module #1	Communication Module	Valves
					

2 SYSTEM SUMMARY

The MAC Valves MI/O 67 is a device that exchanges the Ethernet/IP protocol. Ethernet/IP is an open protocol, based on the CIP (Control Information Protocol). ControlNet International (CI) originated Ethernet/IP in 1999. This new protocol was accomplished by modifying CIP, so as to provide a solid industrial protocol. Later, Ethernet/IP was also adopted by Open DeviceNet Vendor Association (ODVA), and Industrial Ethernet Association (IEA). An Ethernet/IP network is an open network consisting of one or more master devices and multiple slave devices. Because it is an open network, the system will consist of products from a wide variety of vendors. It is important to note that Ethernet/IP, and the commonly known Ethernet, are two slightly different protocols.

The master (a PC or PLC with its network scanner) and slave devices are connected via a standard D-coded M12 connector on an Ethernet cable. The valve and electronics +24VDC will have to be supplied to the MI/O-67 manifold via additional cables.

This system is also DLR compatible. To utilize this mode, both EtherNet I/P ports on the MI/O-67 must be used plus a DLR capable master which acts as the ring supervisor, must be in the system and enabled. Refer to the master's manual for DLR topology and set up information. There are also several white papers on the web which better describe Device Layer Ring (DLR).

The MAC MI/O-67 Fieldbus Manifold is a slave device within the Ethernet/IP network. Thus, it will respond to all of the commands associated with the network like any other node of its type.

Each manifold occupies a single node on the network. The output portion consumes 210 bytes and produces 210 bytes. The configuration occupies 190 bytes. The system is highly configurable and can have a large variety of Digital Input/Output, Analog Input/Output (voltage or current), and Valve Driver modules. It all starts with the Communications Module and a valve stack.

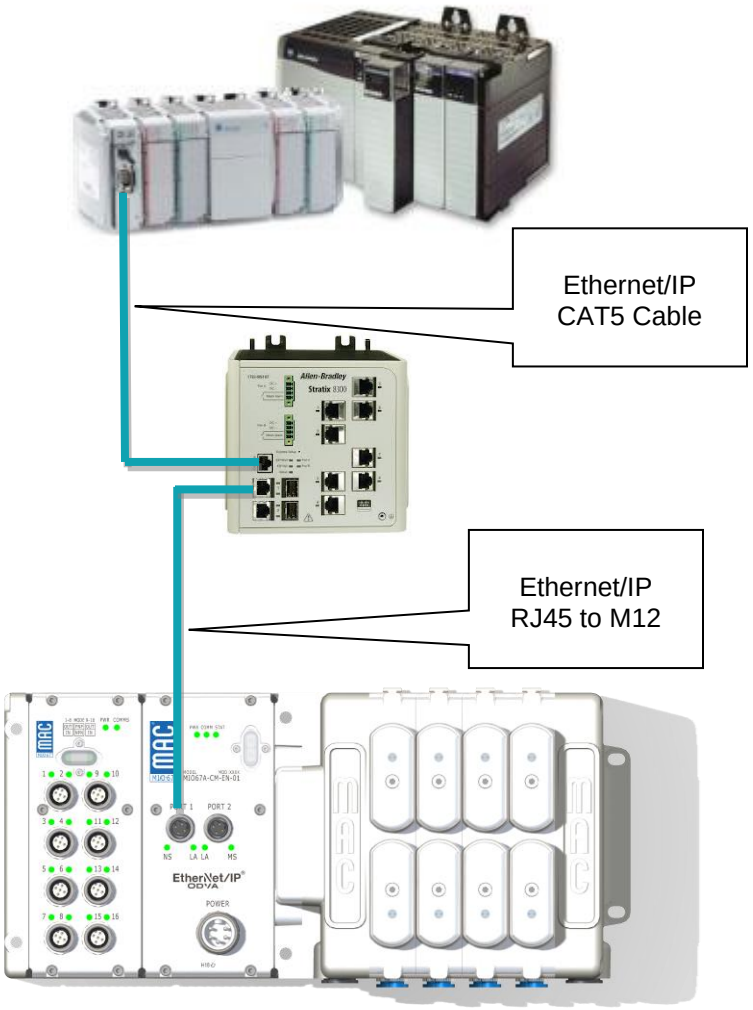
The main communications module is called the Comms Module. Its functions is to provide front-end interfacing to the EtherNet I/P line, operate 32 valve drivers for the stack valves, route power for the stack valves and electronics, and control the CAN bus backplane which interfaces the functional modules.

The functional modules are Analog Modules (voltage and current), Digital I/O, and a Power Plus Module.

The stack will come fully assembled. However, if a need arises to add or subtract modules, turn off all power and air prior to changing the module configuration.

The valve stack can operate up to 32 solenoids in any combination of double and single solenoid valves. It is set up for 24VDC valves.

2.1 Typical Connection Diagram



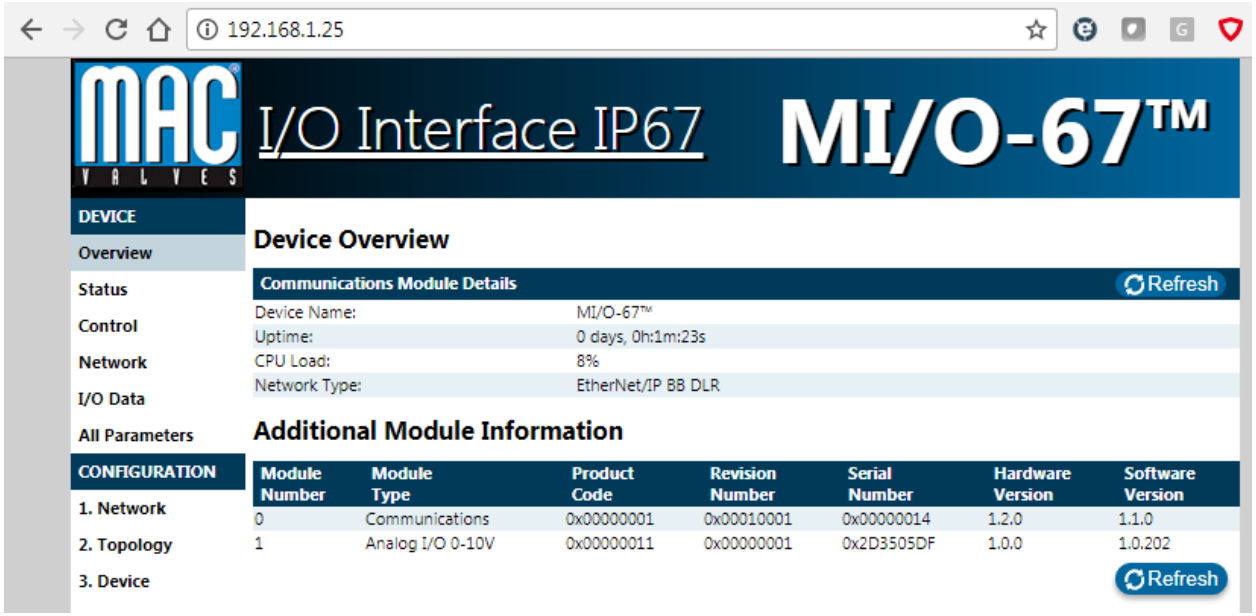
3 CUSTOMIZE THE IP ADDRESS

The factory default IP address:

192.168.1.25

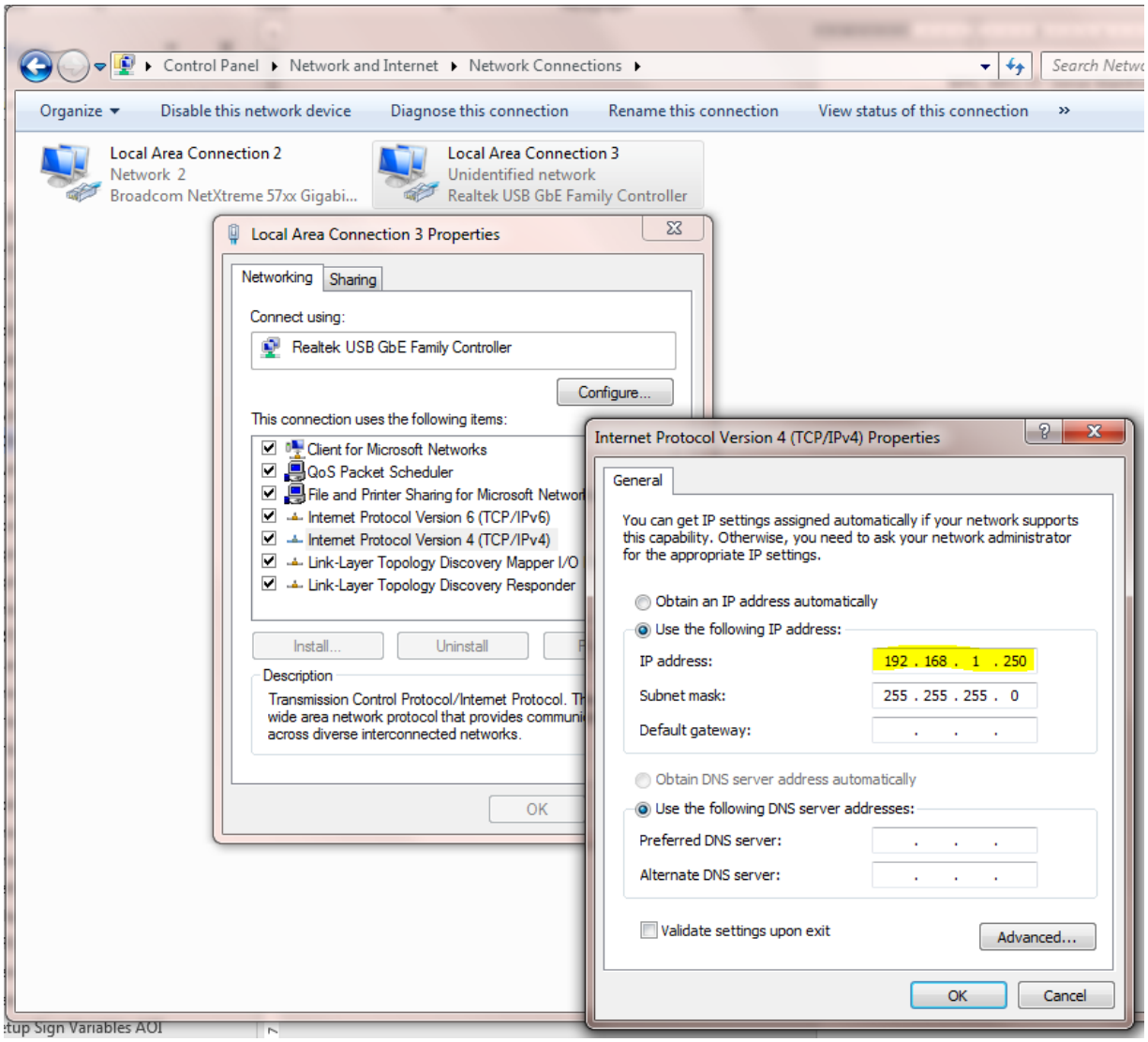
After applying power to the fieldbus manifold, and connecting the appropriate ethernet CAT5 cables, use a web browser to connect to the MI/O 67 using the default IP address provided. Enter the default IP address in the browser URL.

Note: If the IP address is not known, use the “IP Config Tool” downloaded from MAC.



Note: Computer IP address must be set to the same network subnet of 192.168.1.xxx to establish a connection to the MI/O 67 fieldbus manifold. Typically, the computer network adapter is set to a fixed IP address to make this connection.

In this example below, the computer IP address is set to 192.168.1.250 while the MAC MI/O 67 fieldbus manifold IP address is 192.168.1.25.



Once connected to the MI/O 67 fieldbus manifold via the web browser:

Select “Network” under the Configuration Menu.

Change the network settings

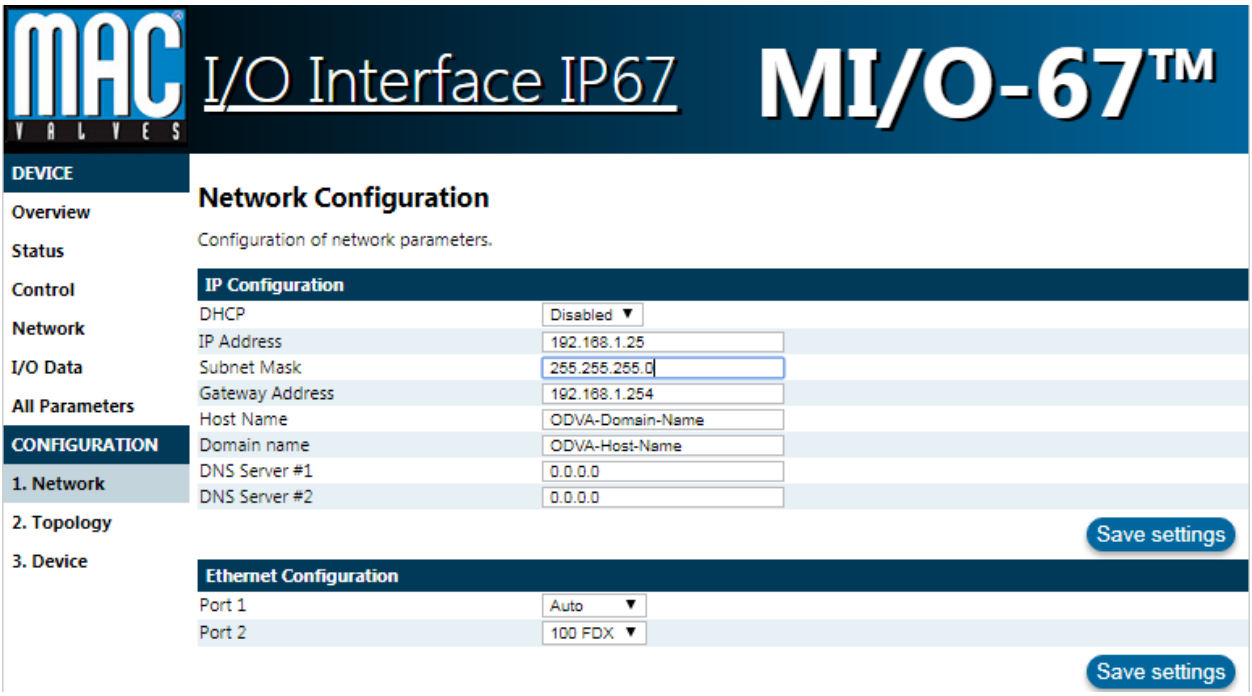
IP Address
SubnetMask
Gateway Address
Host Name
Domain Name
DNS Server #1
DNS Server #2

Settings to match the industrial machine network adress scheme. Typically DHCP is left disabled for EtherNET/IP slave devices to connect to EtherNET/IP master devices such as PLCs.

Click “Save Settings”

Power Cycle REQUIRED

Before the MI/O 67 Fieldbus Manifold network changes will take effect



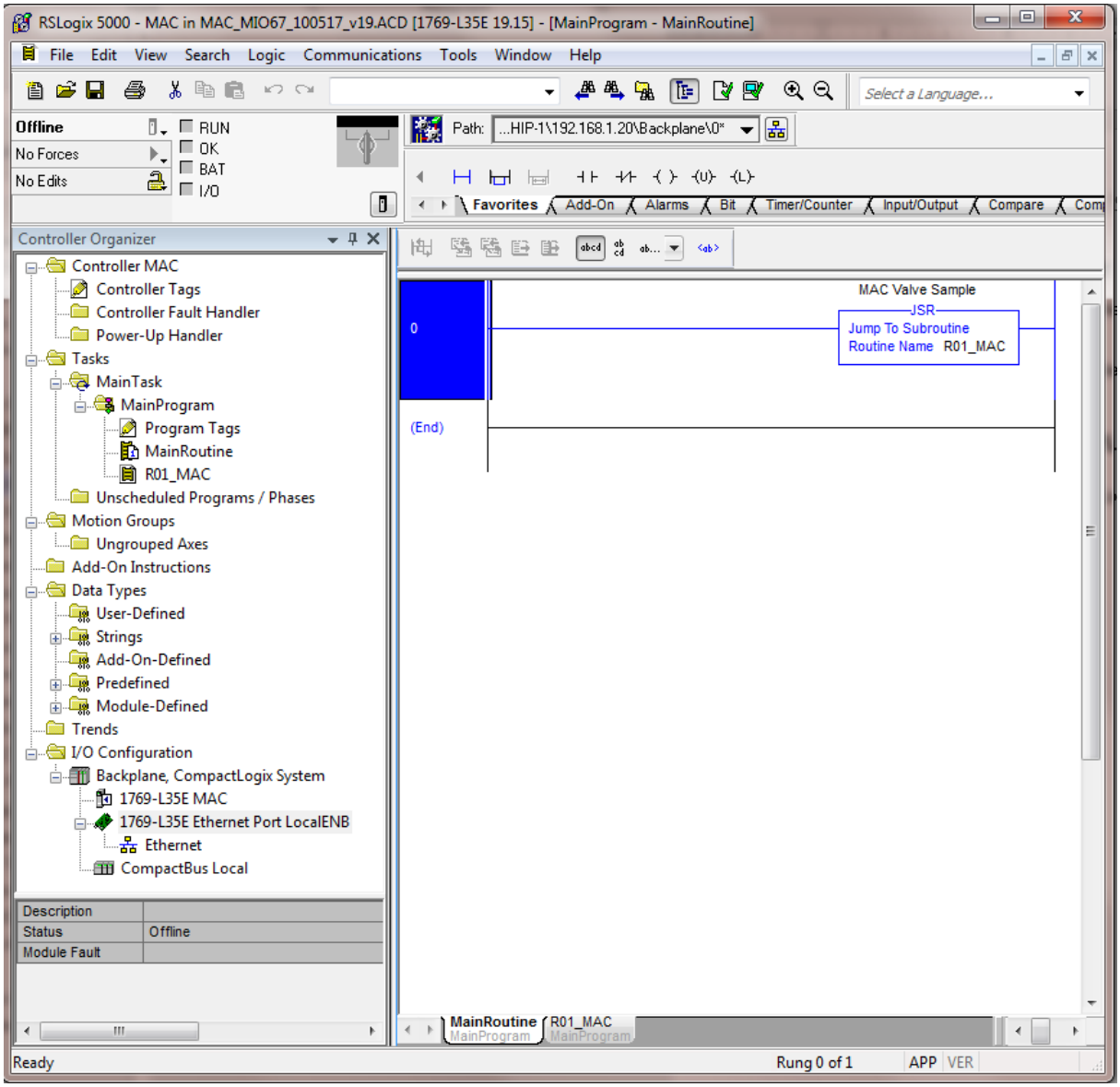
4 PLCs WITH VERSION 19 or EARLIER FIRMWARE

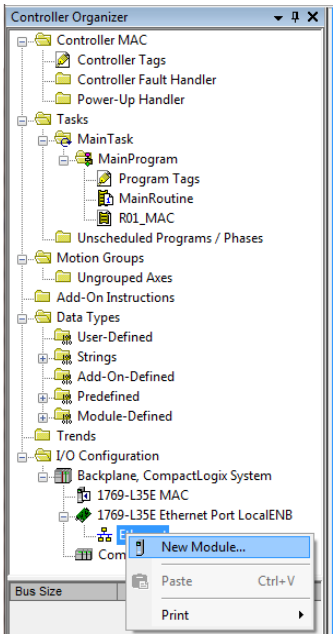
PLCs with firmware version 19 or earlier do not support Allen Bradley's Simple EDS AOP (Add-On Profile) installer.

EtherNET/IP slave configuration will be done manually with a **Generic EtherNET/IP block.**

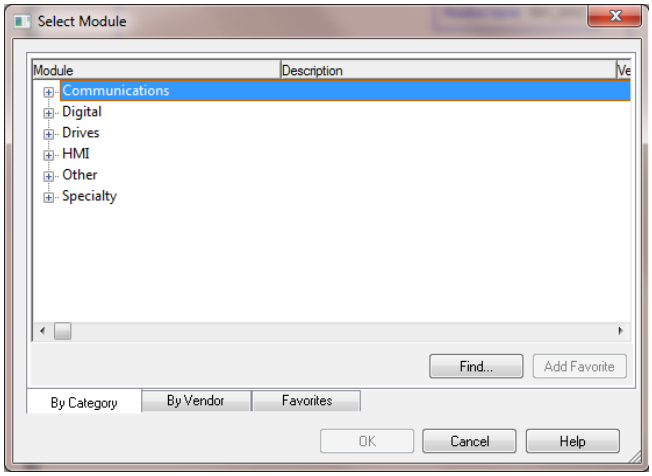
4.1 Configure the MI/O 67 to the PLC using Generic Ethernet Module

Open an existing or new RS Logix 5000 project file, OFFLINE.

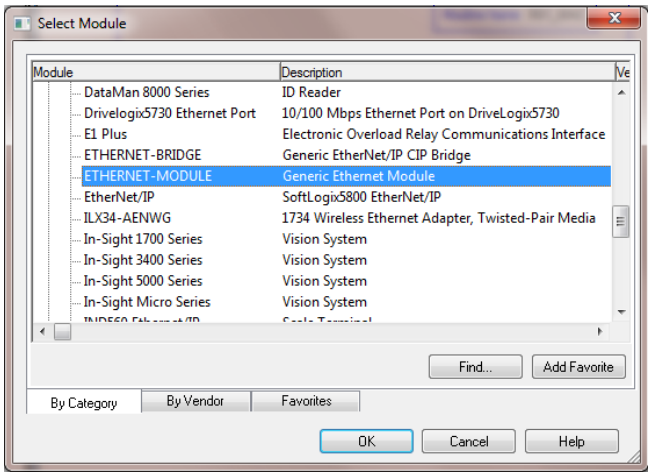




Right click either the PLC CPU, or Ethernet ICON under the PLC CPU.
Choose “New Module”



Choose “Communications”



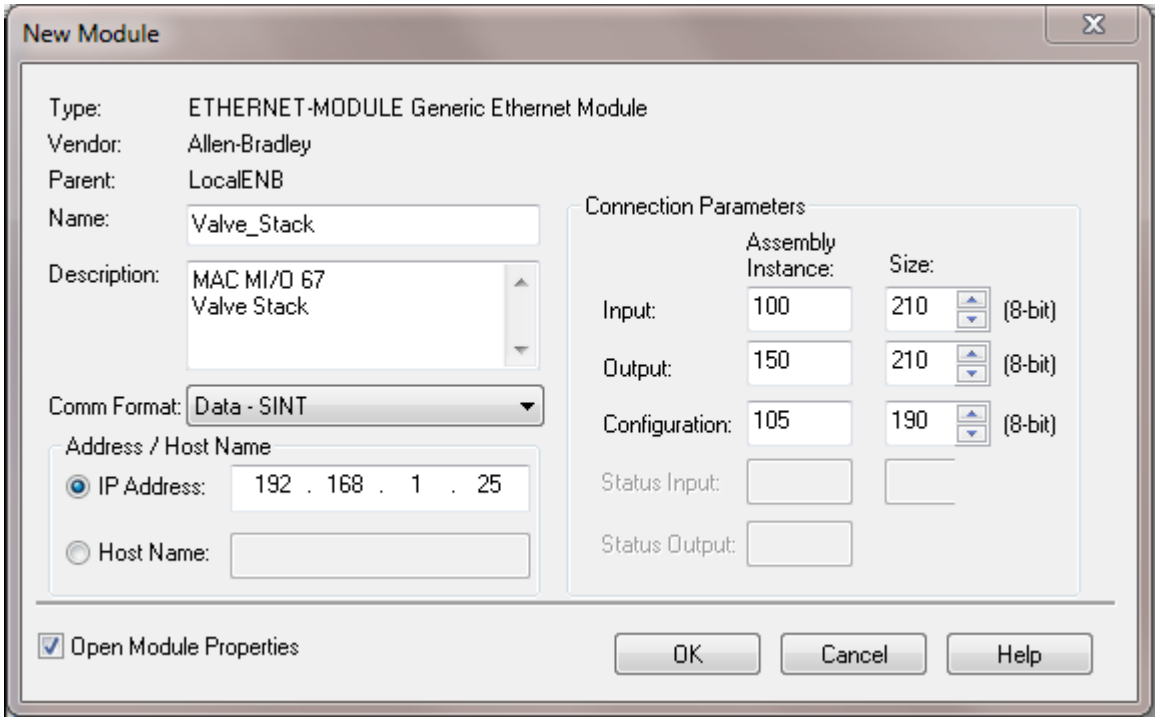
Scroll down and select ETHERNET-MODULE “Generic Ethernet Module”
Choose “Ok”

Enter the following information into the New Module dialog:

Name: This can be any name that will become the PLC Tag Name
Description: Describe the valve manifold
Comm Format: Choose **DATA-SINT**
Address/Host Name: Enter the IP Address Assigned to this Valve Manifold

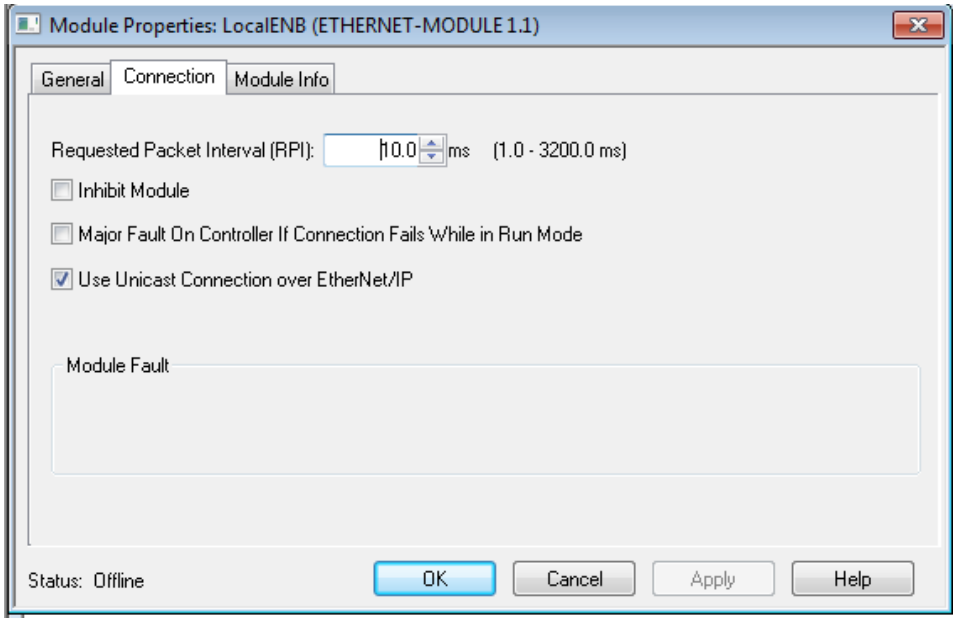
Connection Input: Assembly Instance 100 with 210 bytes
Connection Output: Assembly Instance 150 with 210 bytes
Connection Configuration: Assembly Instance 105 with 190 bytes

Choose “OK”

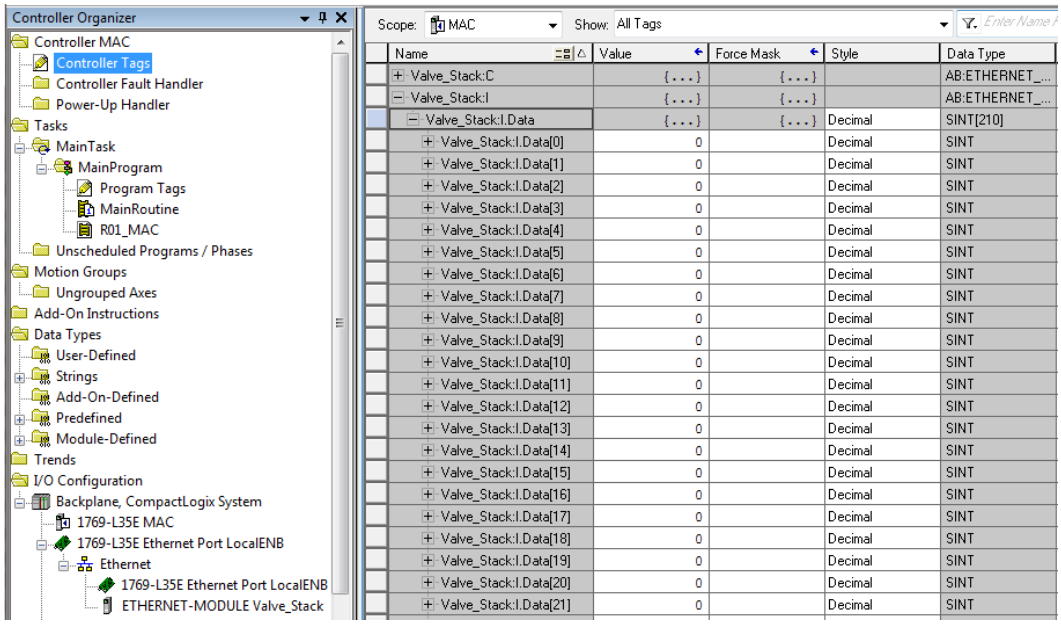


Under the Connection Tab

Set the RPI rate, 10.00 ms is recommended as a default
Choose your optional configuration
Use Unicast Connection over EtherNET/IP is set by default

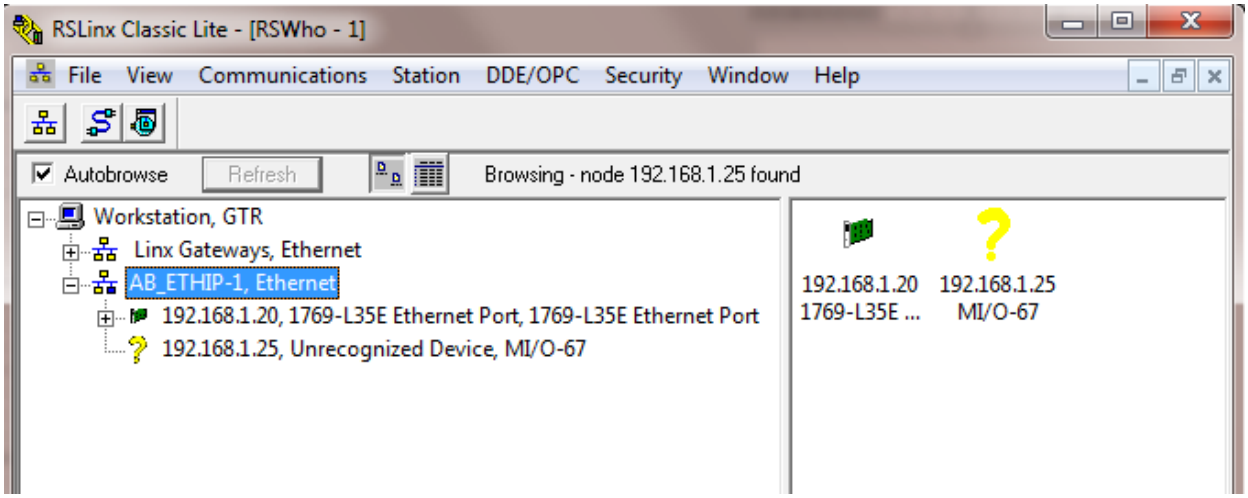


Verify PLC Tags where created under Controller Tags



4.2 Configure the MI/O 67 in RS Linx

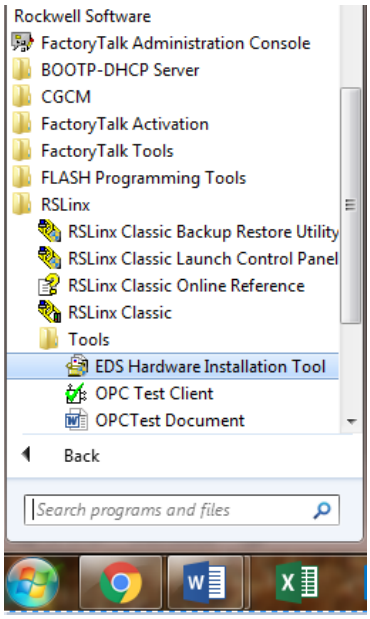
Optionally register the MI/O 67 device on the EtherNET/IP network.



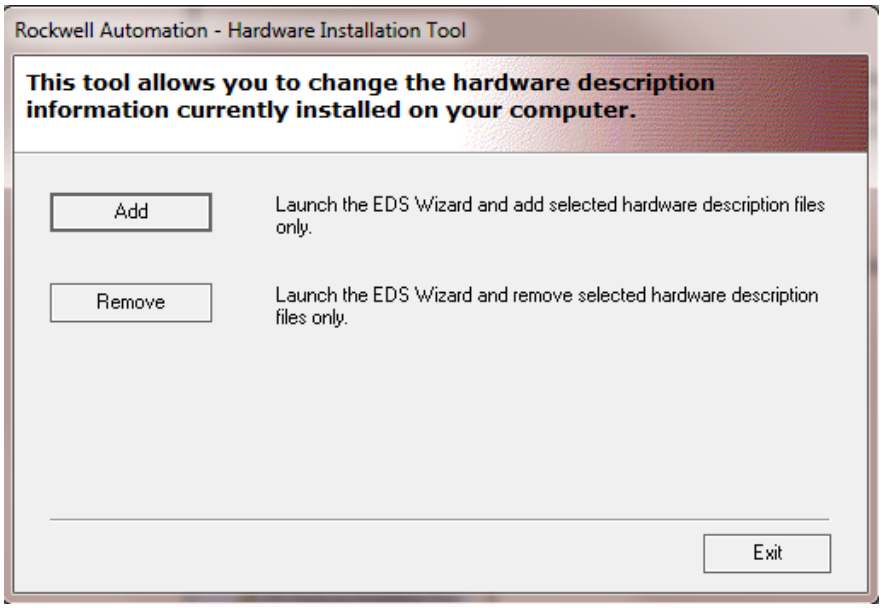
For version 19 or earlier PLCs, the EDS file maybe be used **optionally** to register the MI/O 67 on the EtherNET/IP network using the EDS Hardware Installation Tool provided by Rockwell Software. This step is **not required** to configure the MI/O 67 to an Allen Bradley PLC, however will identify the fieldbus manifold communication module correctly in the RS Linx network browser. This step will correct the "Unrecognized Device" in RS Linx.

This tool is typically found here:

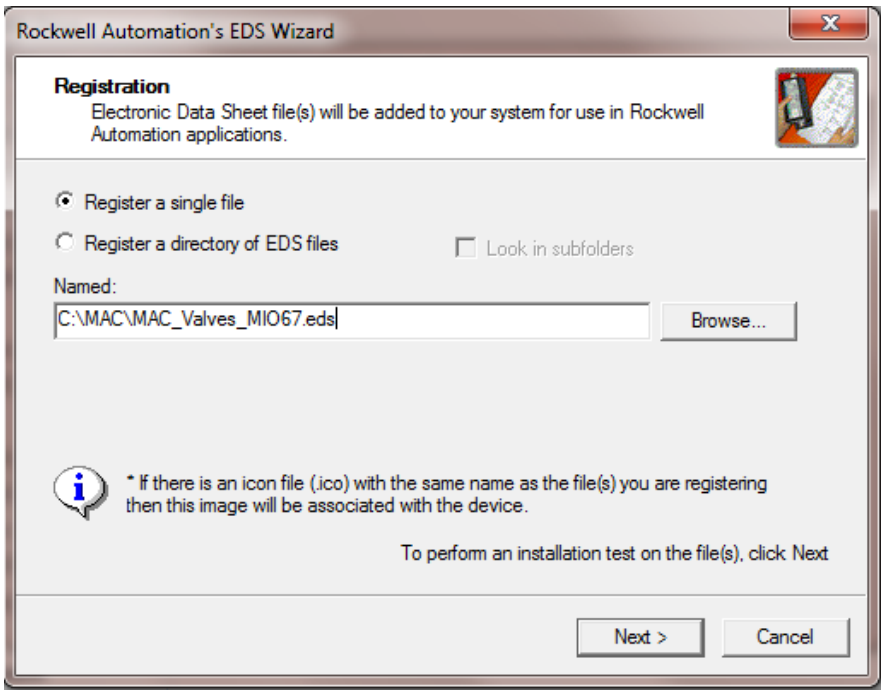
Rockwell Software > RSLinx > Tools > EDS Hardware Installation Tool



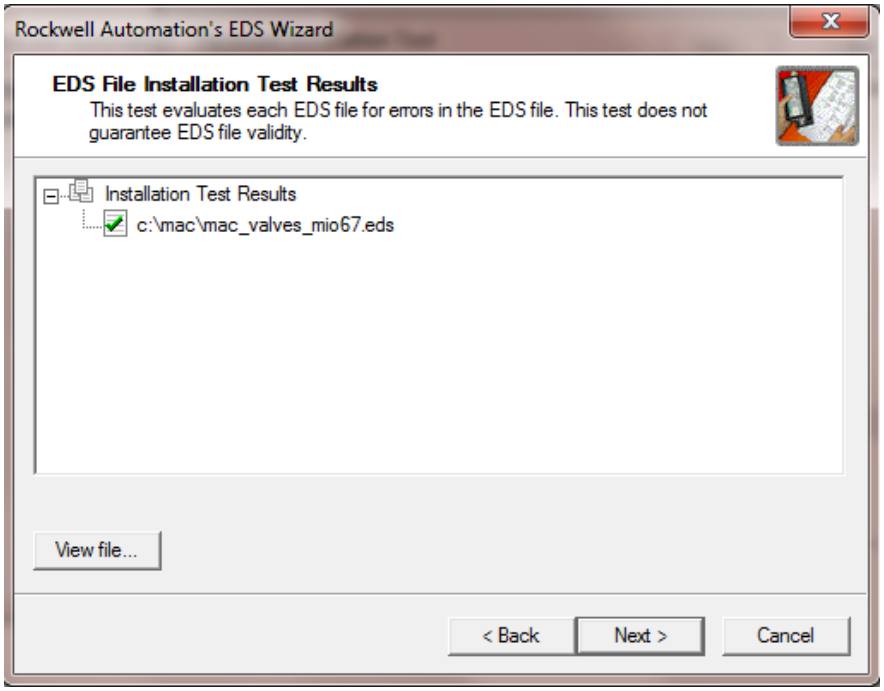
Start the EDS Hardware Installation Tool
Choose “Add”



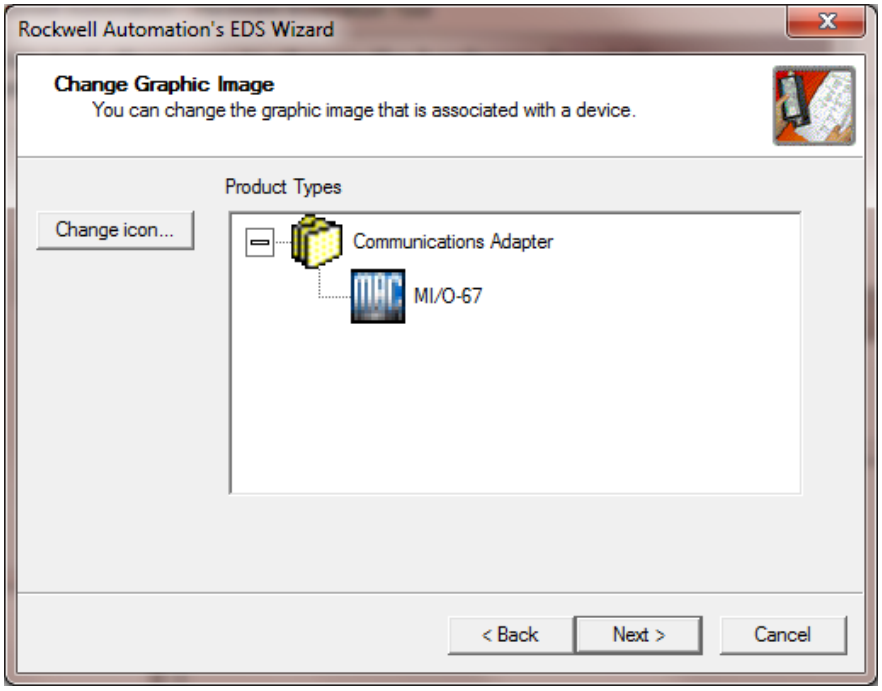
Choose “Register a single file”
Browse to the MAC MI/O 67 EDS file provided.
Choose “Next >”



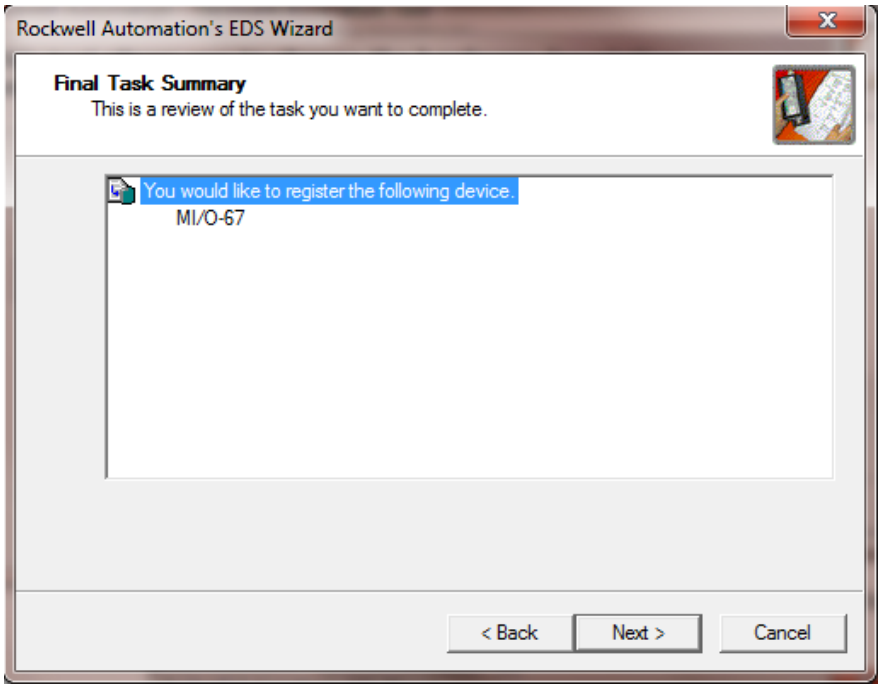
Confirm file location
Confirm correct EDS file
Choose “Next >”

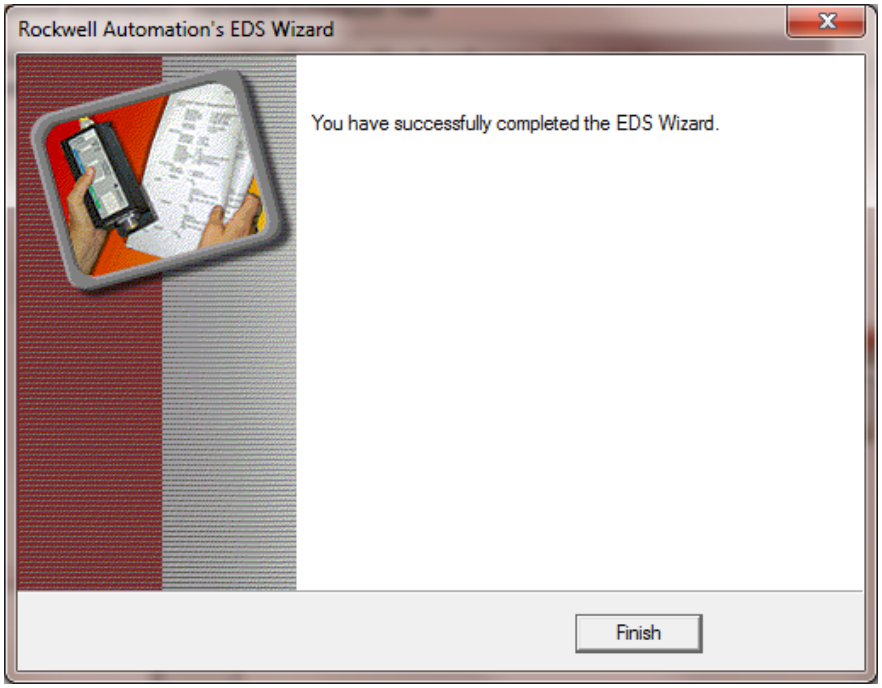


Choose a new ICON or use the supplied embedded ICON file provided by MAC
Choose “Next >”

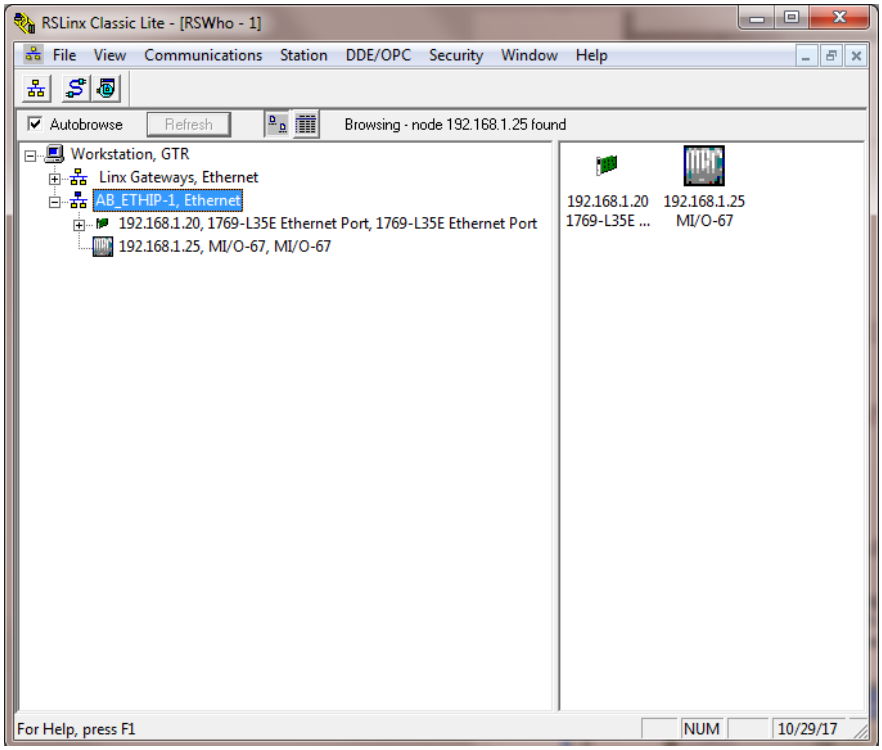


Verify Final Step
Choose “Next >”





Verify RS Linx now identifies MI/O 67 Fieldbus Manifold on EtherNET/IP network



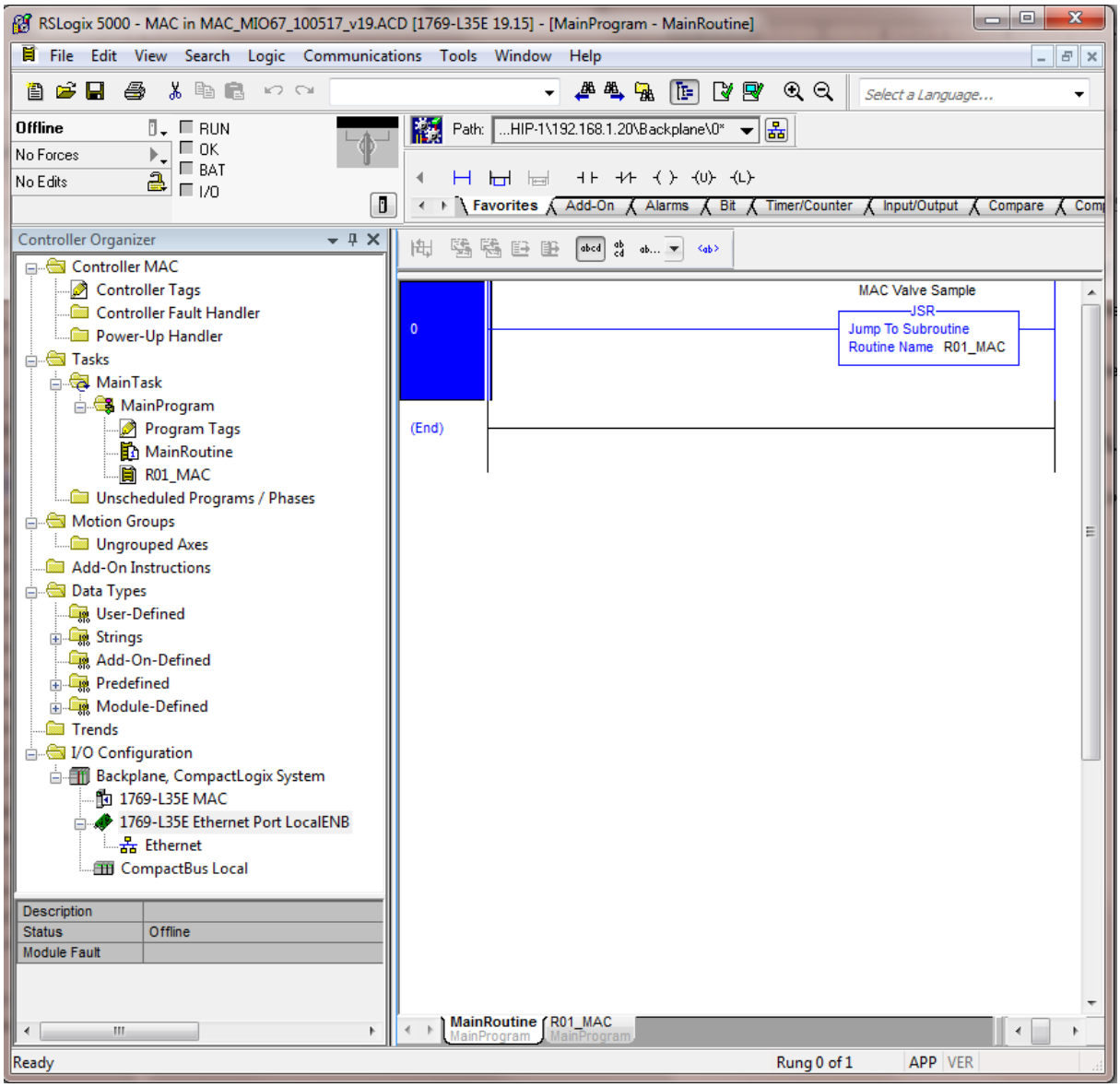
5 PLCs WITH VERISON 20 or NEWER FIRMWARE

Configuration of EtherNET/IP will be done with MAC provided Simple EDS AOP (Add-on Profile).

Please download and obtain the **MAC_Valves_MIO67.eds** file.

5.1 Configure the MI/O 67 to the PLC using Simple EDS AOP (Add-on Profile) file

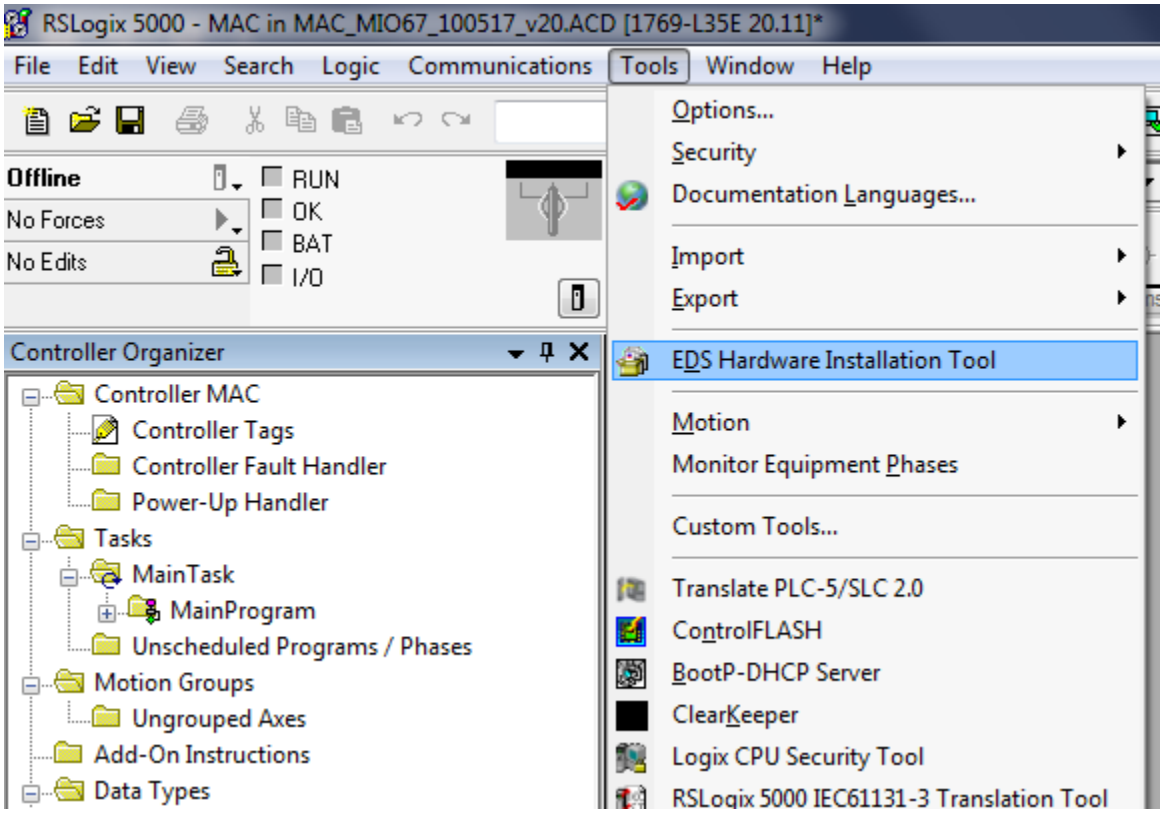
Open an existing or new RS Logix 5000 project file, OFFLINE.



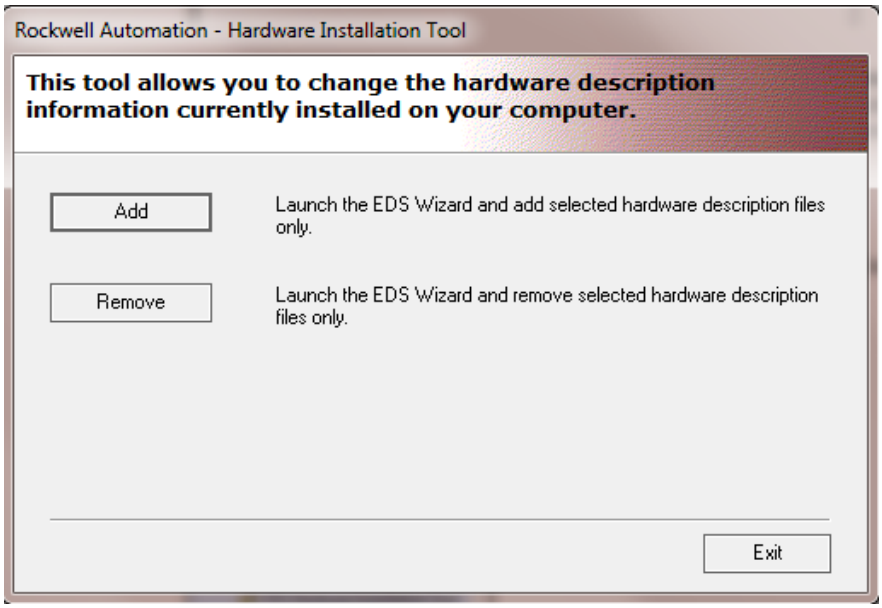
For version 20 or later firmware PLCs, the EDS file will be installed using the EDS Hardware Installation Tool Wizard built into the RS Logix 5000 software to register the MI/O 67 on the EtherNET/IP network.

This tool is typically found here: (inside the RSLogix 5000 software)

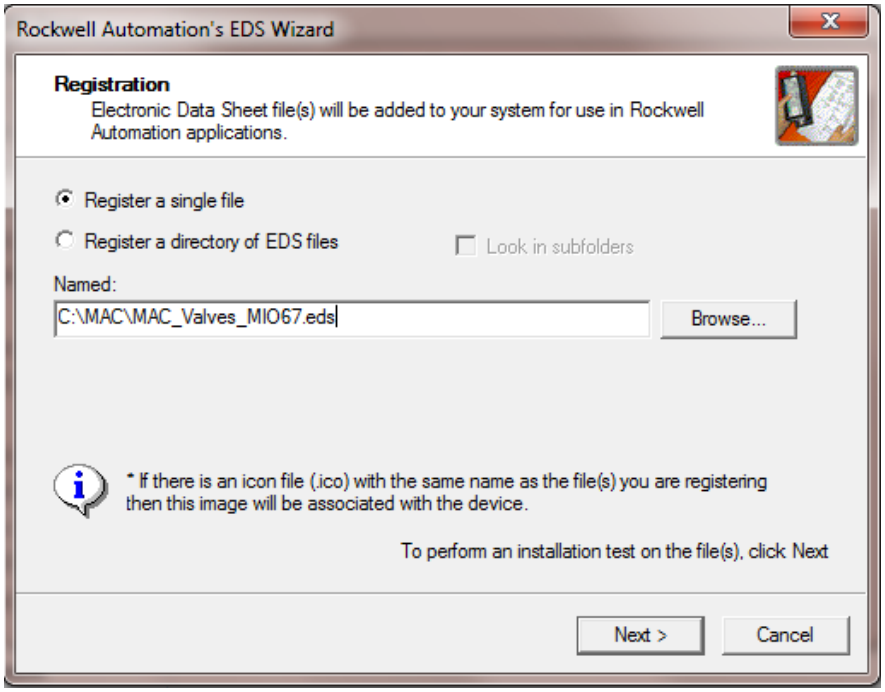
Tools -> EDS Hardware Installation Tool



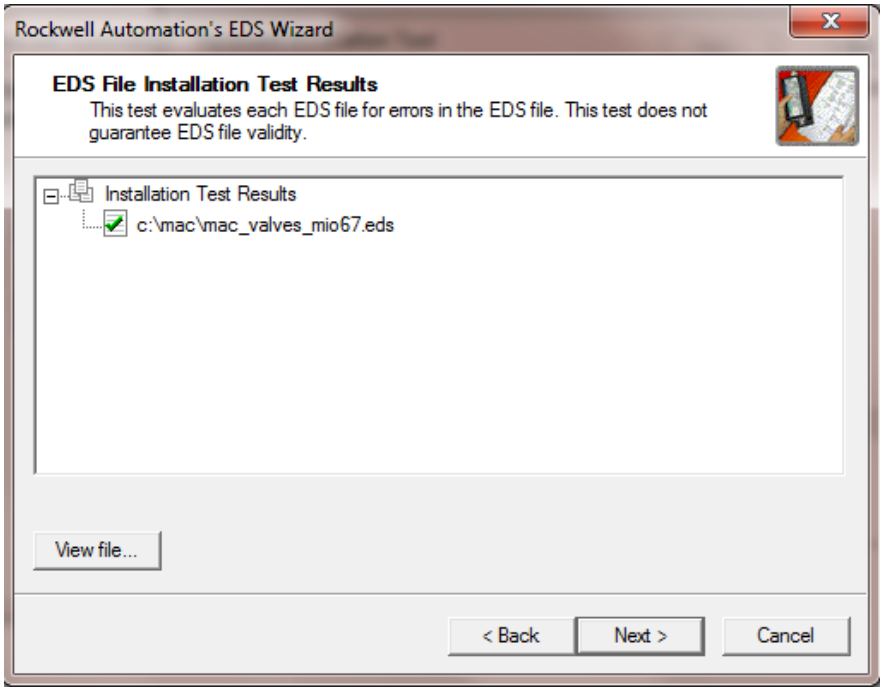
Start the EDS Hardware Installation Tool
Choose “Add”



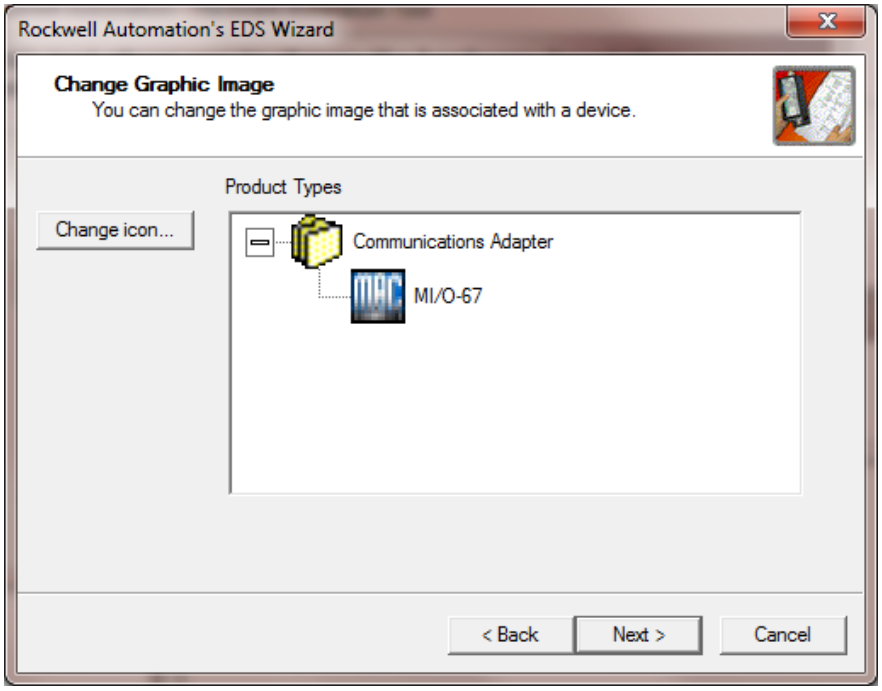
Choose “Register a single file”
Browse to the MAC MI/O 67 EDS file provided.
Choose “Next >”



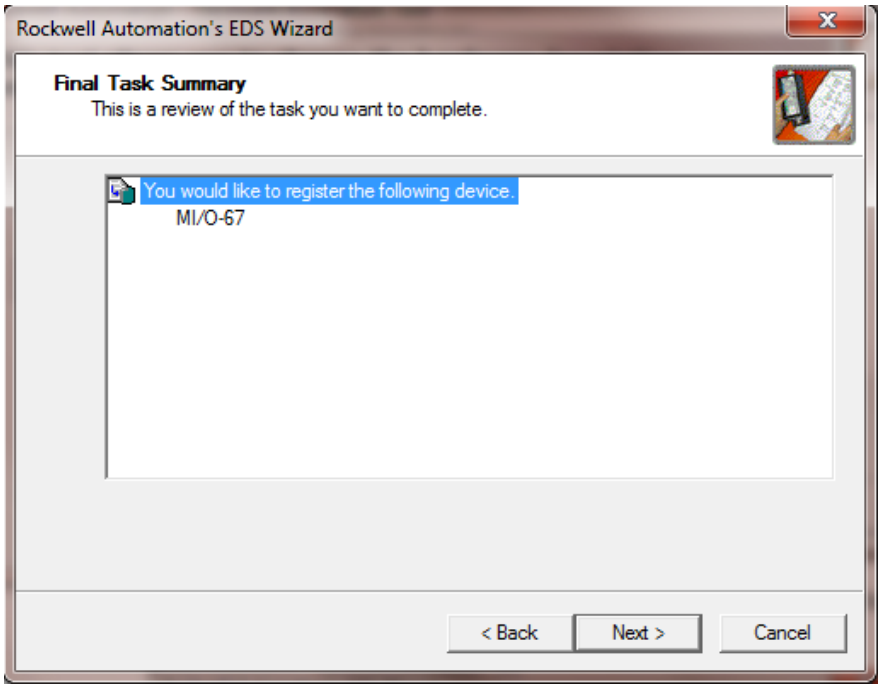
Confirm file location
Confirm correct EDS file
Choose “Next >”

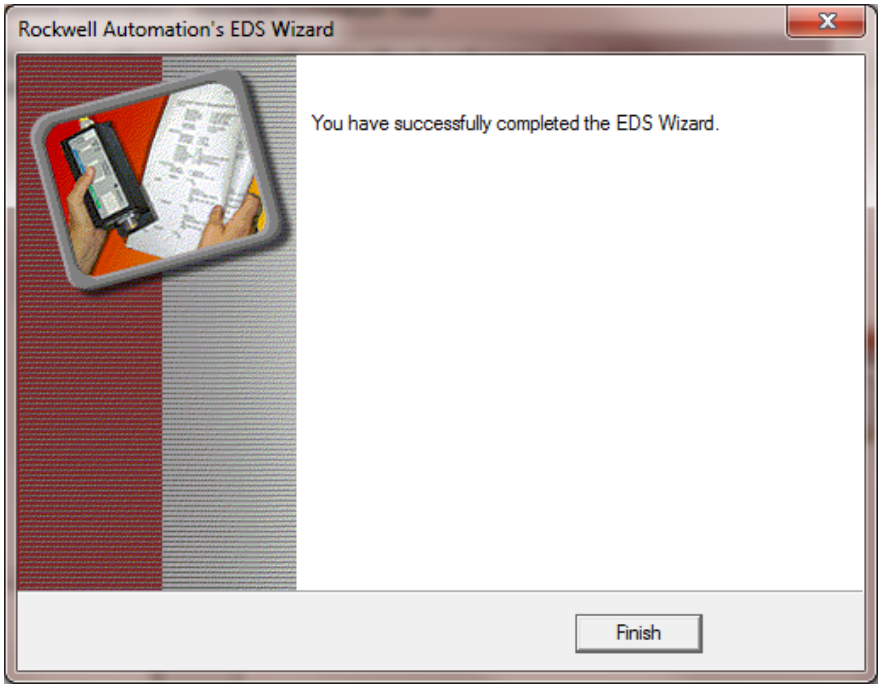


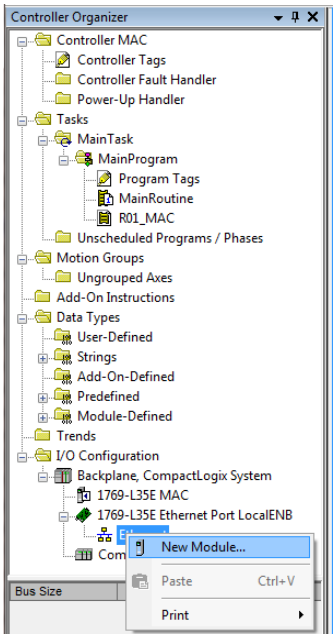
Choose a new ICON or use the supplied embedded ICON file provided by MAC
Choose “Next >”



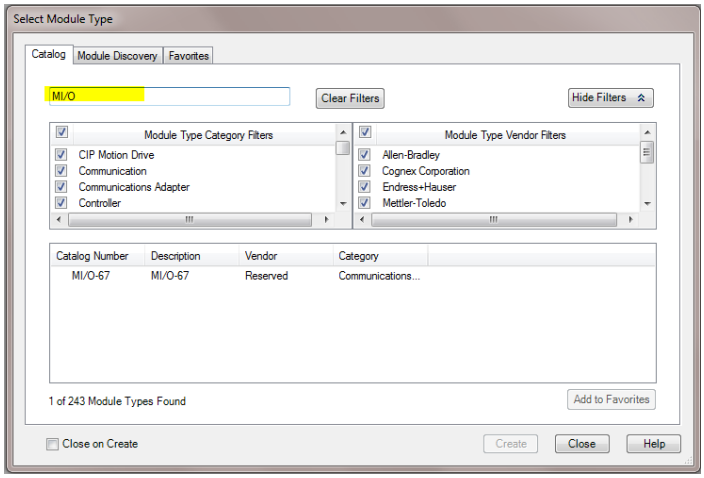
Verify Final Step
Choose “Next >”



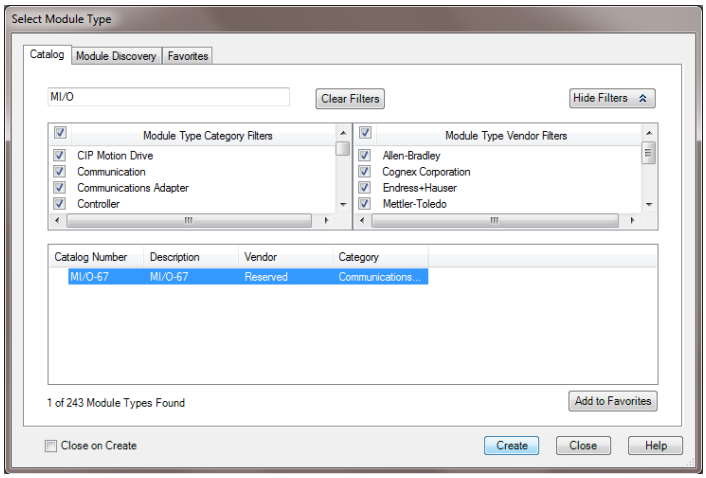




Right click either the PLC CPU, or Ethernet ICON under the PLC CPU.
Choose “New Module”



Type “MI/O” in the search box.



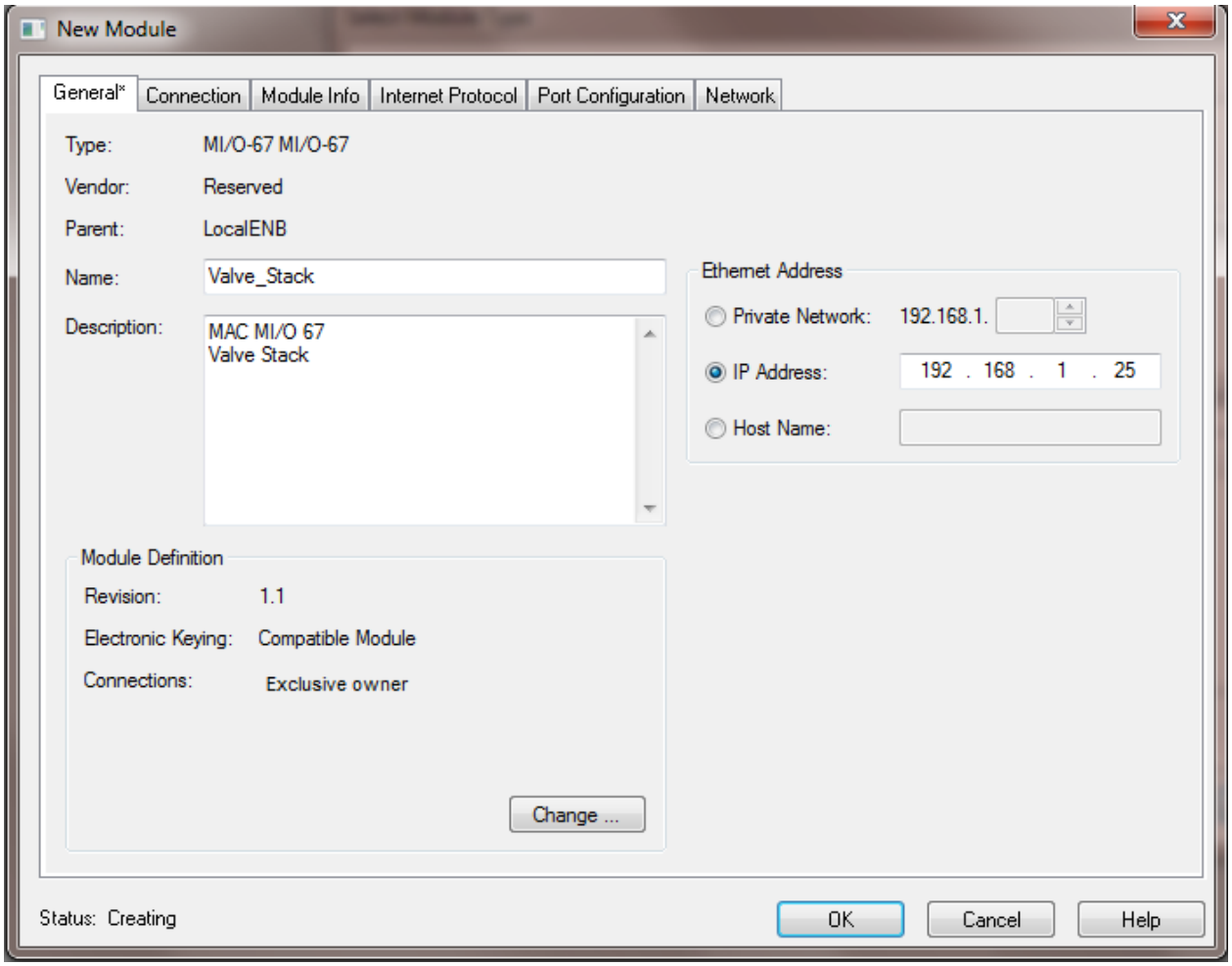
Choose MI/O-67 then click Create

Enter the following information into the New Module dialog:

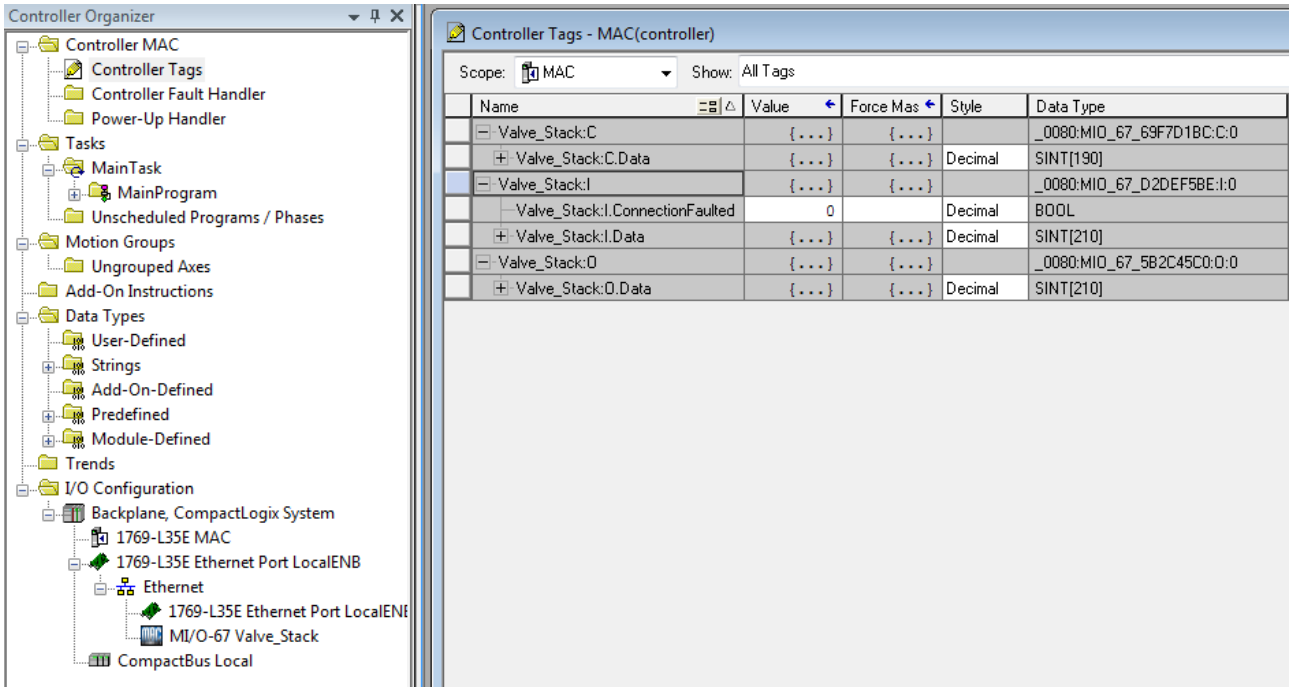
Name: This can be any name that will become the PLC Tag Name
Description: Describe the valve manifold
IP Address: Enter the IP Address Assigned to this Valve Manifold

Choose “OK”

It is recommended to leave the rest of the settings default at this time including 10 msec RPI rate and SINT format under the Change option.



Verify PLC Tags where created under Controller Tags

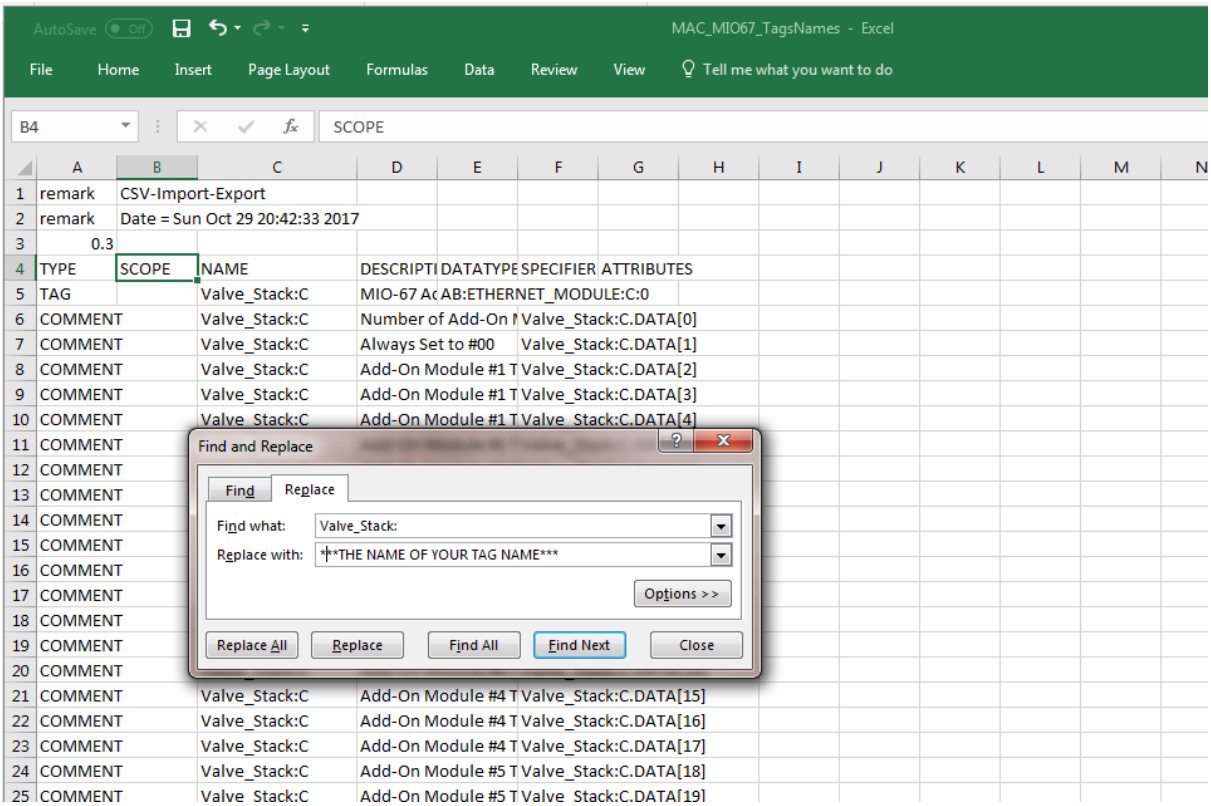


6 Importing PLC Tag Comments and Descriptions

Import the Excel CSV Sheet provided. This CSV sheet is setup to support **SINT format**.

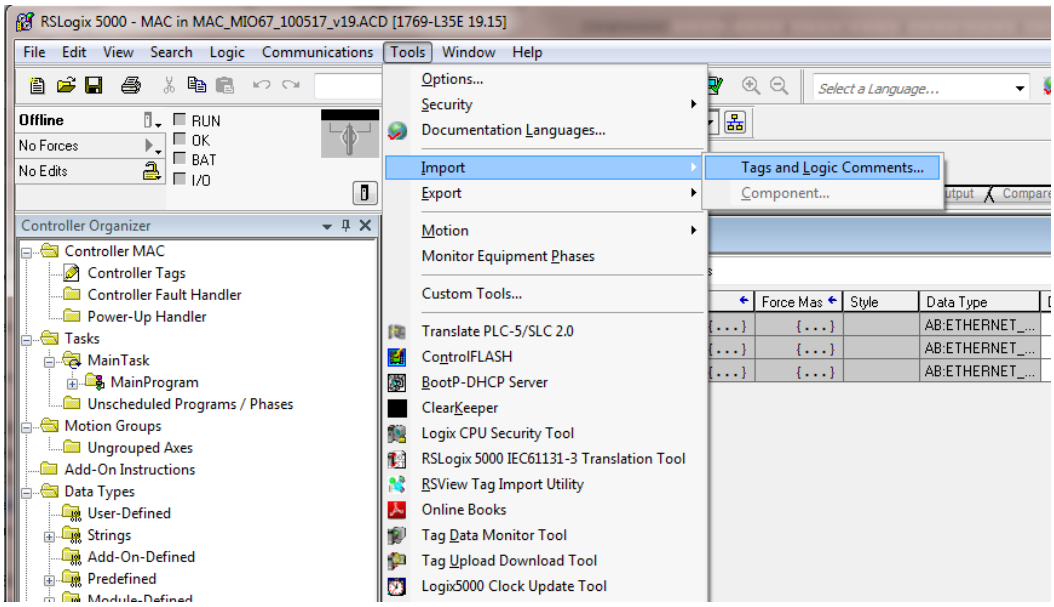
MAC_MIO67-TagsNames.CSV

Open the CSV file in Excel.
Find and Replace “Valve_Stack:” to match your project tag name.
HINT: Don’t forget the “:” at the end of your tag name.
Save the CSV file.

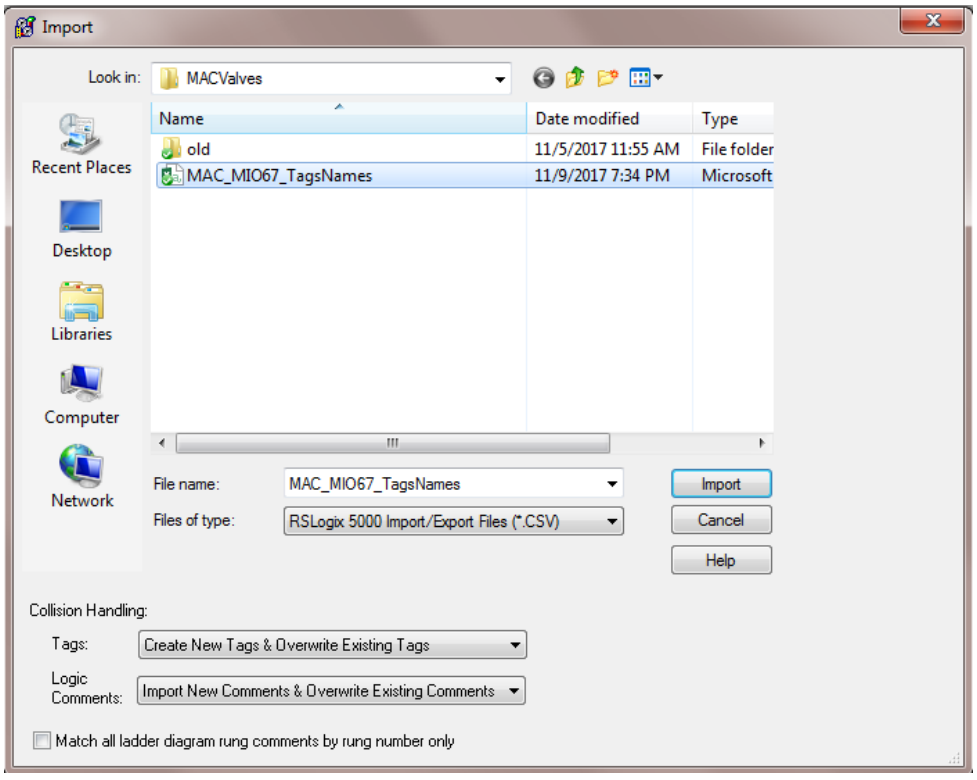


	A	B	C	D	E	F	G	H
1	remark	CSV-Import-Export						
2	remark	Date = Sun Oct 29 20:42:33 2017						
3	0.3							
4	TYPE	SCOPE	NAME	DESCRIPTI	DATATYPE	SPECIFIER	ATTRIBUTES	
5	TAG		Valve_Stack:C	MIO-67 Ac	AB:ETHERNET_MODULE:C:0			
6	COMMENT		Valve_Stack:C	Number of Add-On M		Valve_Stack:C.DATA[0]		
7	COMMENT		Valve_Stack:C	Always Set to #00		Valve_Stack:C.DATA[1]		
8	COMMENT		Valve_Stack:C	Add-On Module #1 T		Valve_Stack:C.DATA[2]		
9	COMMENT		Valve_Stack:C	Add-On Module #1 T		Valve_Stack:C.DATA[3]		
10	COMMENT		Valve_Stack:C	Add-On Module #1 T		Valve_Stack:C.DATA[4]		
11	COMMENT							
12	COMMENT							
13	COMMENT							
14	COMMENT							
15	COMMENT							
16	COMMENT							
17	COMMENT							
18	COMMENT							
19	COMMENT							
20	COMMENT							
21	COMMENT		Valve_Stack:C	Add-On Module #4 T		Valve_Stack:C.DATA[15]		
22	COMMENT		Valve_Stack:C	Add-On Module #4 T		Valve_Stack:C.DATA[16]		
23	COMMENT		Valve_Stack:C	Add-On Module #4 T		Valve_Stack:C.DATA[17]		
24	COMMENT		Valve_Stack:C	Add-On Module #5 T		Valve_Stack:C.DATA[18]		
25	COMMENT		Valve_Stack:C	Add-On Module #5 T		Valve_Stack:C.DATA[19]		

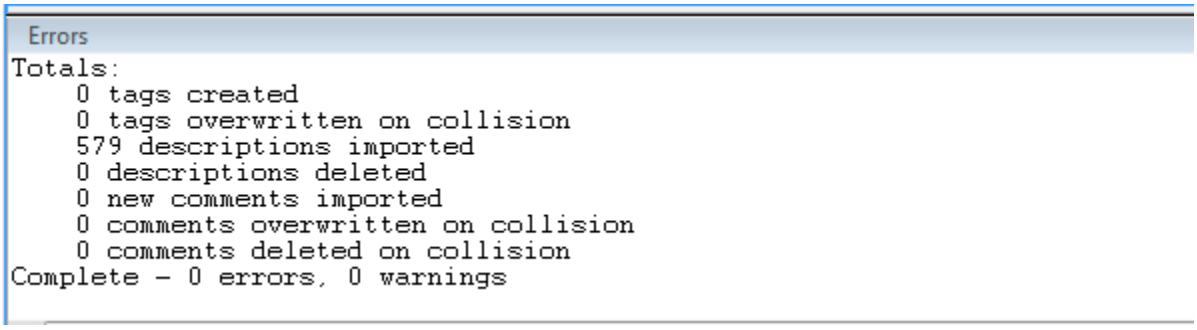
Open RS Logix 5000 Project OFFLINE
Choose Tools > Import > Tags and Logic Comments….



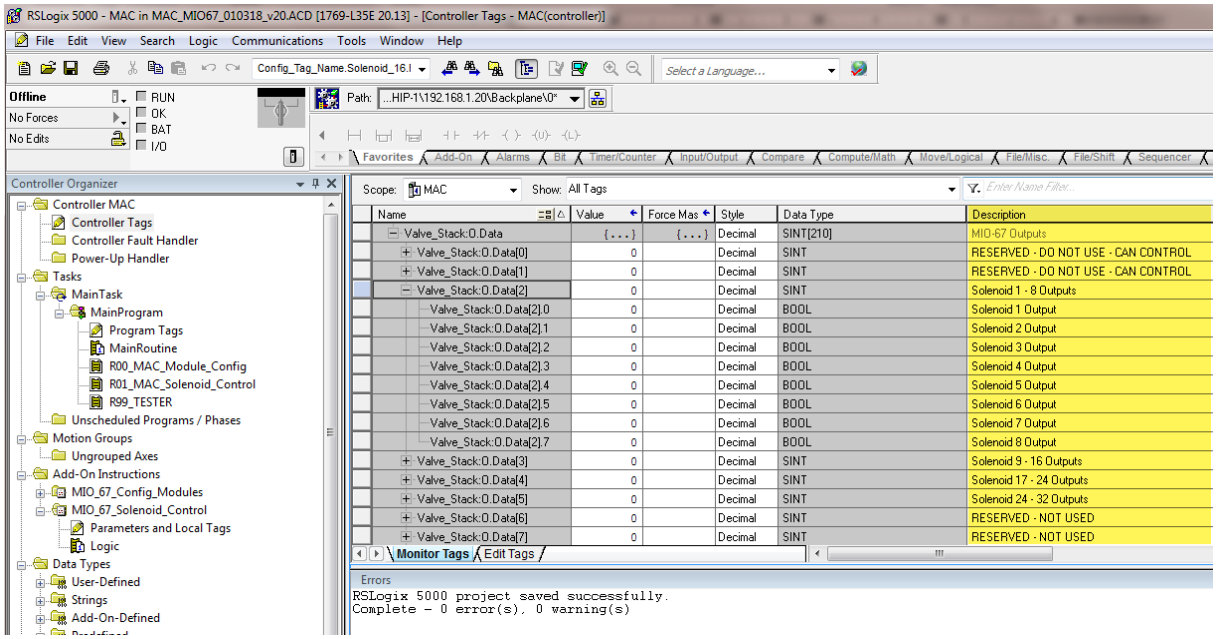
Pick the Edited CSV file
Choose Import



Check for Import Errors



Open Controller Tags
Verify Tags Imported by Expanding Tags
If Tags are properly imported, you should see the “Descriptions” populated, denoted in Yellow.



7 Verify Basic Valve Operation

Download PLC program and go online with PLC. If the stack has additional modules installed, the modules will need configured before going online with the PLC. See section 8 for configuration instructions.

Check to make sure “I/O OK” LED is steady green. Blinking I/O OK LED indicates that the PLC is not communicating via Ethernet/IP to the one or more Slave Devices.

Expand TagName:O.Data[2] and toggle a couple of valves to a “1” state to verify valve stack and valve operations.

Name	Value	Force Mas	Style	Data Type	Description
Valve_Stack:O.Data	{...}	{...}	Decimal	SINT[210]	MIO-67 Outputs
Valve_Stack:O.Data[0]	0		Decimal	SINT	RESERVED - DO NOT USE - CAN CONTROL
Valve_Stack:O.Data[1]	0		Decimal	SINT	RESERVED - DO NOT USE - CAN CONTROL
Valve_Stack:O.Data[2]	5		Decimal	SINT	Solenoid 1 - 8 Outputs
Valve_Stack:O.Data[2].0	1		Decimal	BOOL	Solenoid 1 Output
Valve_Stack:O.Data[2].1	0		Decimal	BOOL	Solenoid 2 Output
Valve_Stack:O.Data[2].2	1		Decimal	BOOL	Solenoid 3 Output
Valve_Stack:O.Data[2].3	0		Decimal	BOOL	Solenoid 4 Output
Valve_Stack:O.Data[2].4	0		Decimal	BOOL	Solenoid 5 Output
Valve_Stack:O.Data[2].5	0		Decimal	BOOL	Solenoid 6 Output
Valve_Stack:O.Data[2].6	0		Decimal	BOOL	Solenoid 7 Output
Valve_Stack:O.Data[2].7	0		Decimal	BOOL	Solenoid 8 Output
Valve_Stack:O.Data[3]	0		Decimal	SINT	Solenoid 9 - 16 Outputs
Valve_Stack:O.Data[4]	0		Decimal	SINT	Solenoid 17 - 24 Outputs
Valve_Stack:O.Data[5]	0		Decimal	SINT	Solenoid 24 - 32 Outputs
Valve_Stack:O.Data[6]	0		Decimal	SINT	RESERVED - NOT USED
Valve_Stack:O.Data[7]	0		Decimal	SINT	RESERVED - NOT USED

7.1 Understanding Soleniod Bit Mapping

There are three types of solenoids typically offered by MAC. Depending how the customer orders the valve manifold will determine which solenoid outputs are mapped in the PLC output registers.

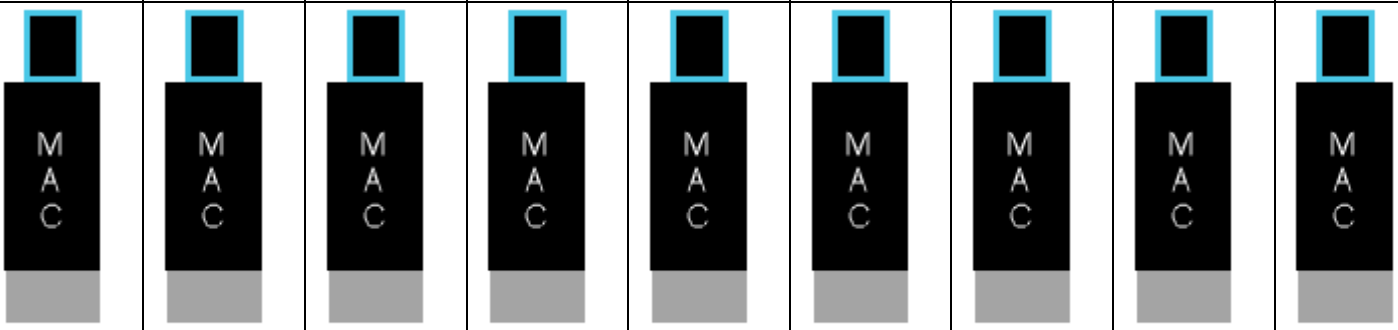
Type 1: Single Solenoid

Type 2: Single Solenoid wired as a double

Type 3: Double Solenoid

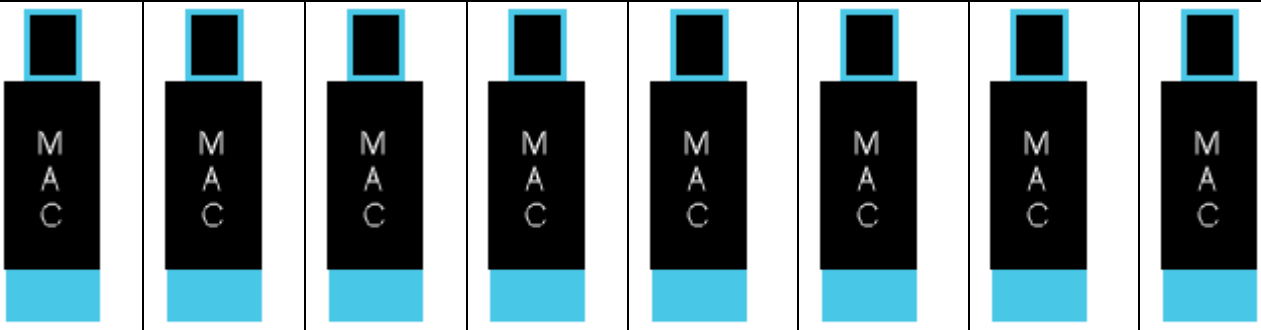
7.1.1 Type 1: Single Solenoid

Single solenoids use solenoid outputs in sequence in relation to the PLC output mapping. For example, if a valve manifold is configured such as below where all the valves are single solenoid, note the solenoid control output bits.

Solenoid 1	**Solenoid 2**	**Solenoid 3**	**Solenoid 4**	**Solenoid 5**	**Solenoid 6**	**Solenoid 7**	**Solenoid 8**	**Solenoid 9**
(EIP) EtherNET/IP Address Below								
O.Data[2].0	O.Data[2].1	O.Data[2].2	O.Data[2].3	O.Data[2].4	O.Data[2].5	O.Data[2].6	O.Data[2].7	O.Data[3].0
AOI Tag Name Below (BOOL) Method and (DINT) Method (see Section 10 in this manual)								
Solenoid_1.Output	Solenoid_2.Output	Solenoid_3.Output	Solenoid_4.Output	Solenoid_5.Output	Solenoid_6.Output	Solenoid_7.Output	Solenoid_8.Output	Solenoid_9.Output
Solenoid_Output_All.0	Solenoid_Output_All.1	Solenoid_Output_All.2	Solenoid_Output_All.3	Solenoid_Output_All.4	Solenoid_Output_All.5	Solenoid_Output_All.6	Solenoid_Output_All.7	Solenoid_Output_All.8
								

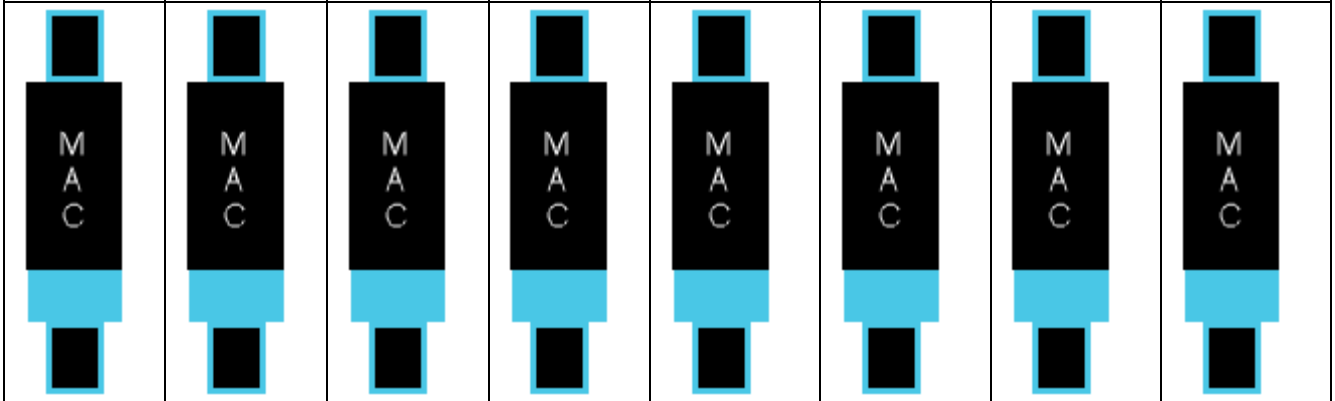
7.1.2 Type 2: Single Solenoid (wired as a double)

Single solenoids wired as double contain internal wiring that would be used as a double valve in the future someday. For example, if the customer wanted to replace a single solenoid with a double solenoid, this would be possible to do this. To control these types of solenoids, the PLC logic for the un-used solenoid is skipped. Solenoid 2, 4, 6, 8, 10, 12, 14, 16, 18 are un-used in the PLC logic.

Solenoid 1	**Solenoid 3**	**Solenoid 5**	**Solenoid 7**	**Solenoid 9**	**Solenoid 11**	**Solenoid 13**	**Solenoid 15**
(EIP) EtherNET/IP Address Below							
O.Data[2].0	O.Data[2].2	O.Data[2].4	O.Data[2].6	O.Data[3].0	O.Data[3].2	O.Data[3].4	O.Data[3].6
AOI Tag Name Below (BOOL) Method and (DINT) Method (see Section 10 in this manual)							
Solenoid_1.Output	Solenoid_3.Output	Solenoid_5.Output	Solenoid_7.Output	Solenoid_9.Output	Solenoid_11.Output	Solenoid_13.Output	Solenoid_15.Output
Solenoid_Output_All.0	Solenoid_Output_All.2	Solenoid_Output_All.4	Solenoid_Output_All.6	Solenoid_Output_All.8	Solenoid_Output_All.10	Solenoid_Output_All.12	Solenoid_Output_All.14
							
Solenoid 2 (unused)	**Solenoid 4 (unused)**	**Solenoid 6 (unused)**	**Solenoid 8 (unused)**	**Solenoid 10 (unused)**	**Solenoid 12 (unused)**	**Solenoid 14 (unused)**	**Solenoid 16 (unused)**
(EIP) EtherNET/IP Address Below							
O.Data[2].1	O.Data[2].3	O.Data[2].5	O.Data[2].7	O.Data[3].1	O.Data[3].3	O.Data[3].5	O.Data[3].7
AOI Tag Name Below (BOOL) Method and (DINT) Method (see Section 10 in this manual)							
Solenoid_2.Output	Solenoid_4.Output	Solenoid_6.Output	Solenoid_8.Output	Solenoid_10.Output	Solenoid_12.Output	Solenoid_14.Output	Solenoid_16.Output
Solenoid_Output_All.1	Solenoid_Output_All.3	Solenoid_Output_All.5	Solenoid_Output_All.7	Solenoid_Output_All.9	Solenoid_Output_All.11	Solenoid_Output_All.13	Solenoid_Output_All.15

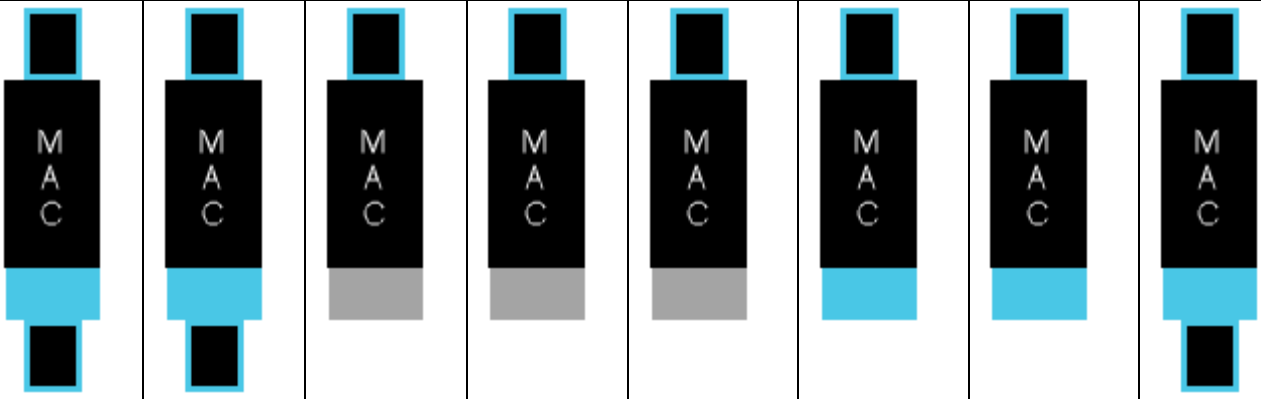
7.1.3 Type 3. Double Solenoid

Double solenoids use solenoid outputs in sequence in relation to the PLC output mapping. Different from singles, the sequence is “A” solenoid first then “B”, then the next valve. For example, if a valve manifold is configured such as below where all the valves are double solenoid, note the solenoid control output bits.

Solenoid 1 “A”	**Solenoid 3 “A”**	**Solenoid 5 “A”**	**Solenoid 7 “A”**	**Solenoid 9 “A”**	**Solenoid 11 “A”**	**Solenoid 13 “A”**	**Solenoid 15 “A”**
(EIP) EtherNET/IP Address Below							
O.Data[2].0	O.Data[2].2	O.Data[2].4	O.Data[2].6	O.Data[3].0	O.Data[3].2	O.Data[3].4	O.Data[3].6
AOI Tag Name Below (BOOL) Method and (DINT) Method (see Section 10 in this manual)							
Solenoid_1.Output	Solenoid_3.Output	Solenoid_5.Output	Solenoid_7.Output	Solenoid_9.Output	Solenoid_11.Output	Solenoid_13.Output	Solenoid_15.Output
Solenoid_Output_All.0	Solenoid_Output_All.2	Solenoid_Output_All.4	Solenoid_Output_All.6	Solenoid_Output_All.8	Solenoid_Output_All.10	Solenoid_Output_All.12	Solenoid_Output_All.14
							
Solenoid 2 “B”	**Solenoid 4 “B”**	**Solenoid 6 “B”**	**Solenoid 8 “B”**	**Solenoid 10 “B”**	**Solenoid 12 “B”**	**Solenoid 14 “B”**	**Solenoid 16 “B”**
(EIP) EtherNET/IP Address Below							
O.Data[2].1	O.Data[2].3	O.Data[2].5	O.Data[2].7	O.Data[3].1	O.Data[3].3	O.Data[3].5	O.Data[3].7
AOI Tag Name Below (BOOL) Method and (DINT) Method (see Section 10 in this manual)							
Solenoid_2.Output	Solenoid_4.Output	Solenoid_6.Output	Solenoid_8.Output	Solenoid_10.Output	Solenoid_12.Output	Solenoid_14.Output	Solenoid_16.Output
Solenoid_Output_All.1	Solenoid_Output_All.3	Solenoid_Output_All.5	Solenoid_Output_All.7	Solenoid_Output_All.9	Solenoid_Output_All.11	Solenoid_Output_All.13	Solenoid_Output_All.15

7.1.4 Mixture Example

Below we show a mixture of all three types

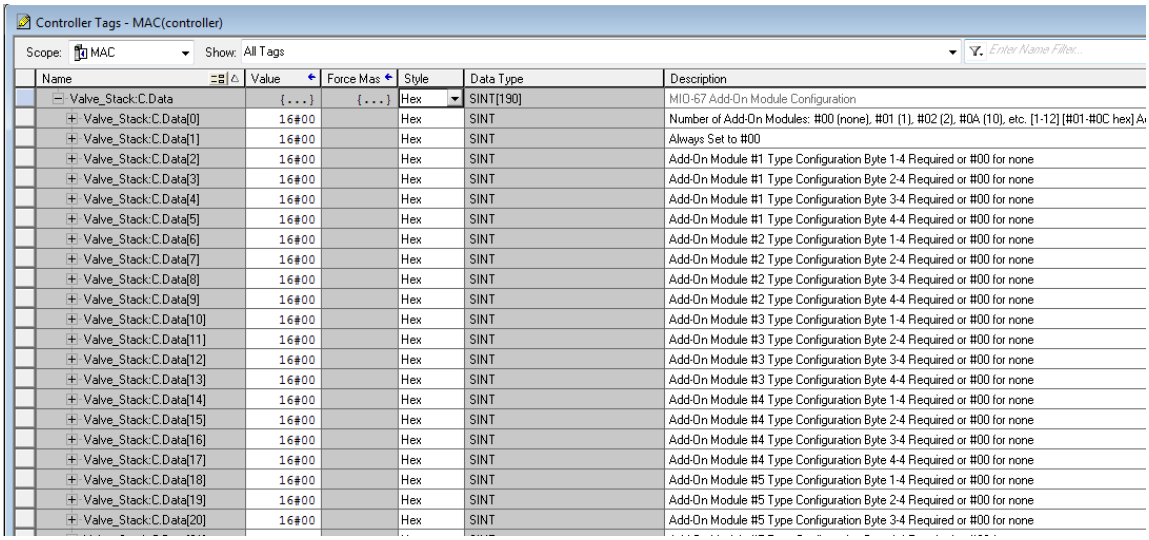
Double	Double	Single	Single	Single	Single wired Double	Single wired Double	Double
Solenoid 1 “A”	**Solenoid 3 “A”**	**Solenoid 5 “A”**	**Solenoid 6 “A”**	**Solenoid 7 “A”**	**Solenoid 8 “A”**	**Solenoid 10 “A”**	**Solenoid 12 “A”**
(EIP) EtherNET/IP Address Below							
O.Data[2].0	O.Data[2].2	O.Data[2].4	O.Data[2].5	O.Data[2].6	O.Data[2].7	O.Data[3].1	O.Data[3].3
AOI Tag Name Below (BOOL) Method and (DINT) Method (see Section 10 in this manual)							
Solenoid_1.Output	Solenoid_3.Output	Solenoid_5.Output	Solenoid_6.Output	Solenoid_7.Output	Solenoid_8.Output	Solenoid_10.Output	Solenoid_12.Output
Solenoid_Output_All.0	Solenoid_Output_All.2	Solenoid_Output_All.4	Solenoid_Output_All.5	Solenoid_Output_All.6	Solenoid_Output_All.7	Solenoid_Output_All.9	Solenoid_Output_All.11
							
Solenoid 2 “B”	**Solenoid 4 “B”**				**Solenoid 9 “B” (unused)**	**Solenoid 11 “B” (unused)**	**Solenoid 13 “B”**
(EIP) EtherNET/IP Address Below							
O.Data[2].1	O.Data[2].3				O.Data[3].0	O.Data[3].2	O.Data[3].4
AOI Tag Name Below (BOOL) Method and (DINT) Method (see Section 10 in this manual)							
Solenoid_2.Output	Solenoid_4.Output				Solenoid_9.Output	Solenoid_11.Output	Solenoid_13.Output
Solenoid_Output_All.1	Solenoid_Output_All.3				Solenoid_Output_All.8	Solenoid_Output_All.10	Solenoid_Output_All.12

8 Configuring Expansion Modules

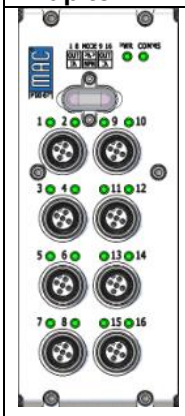
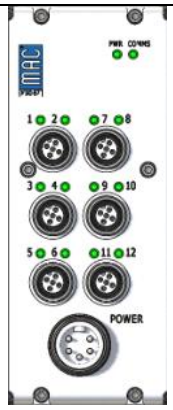
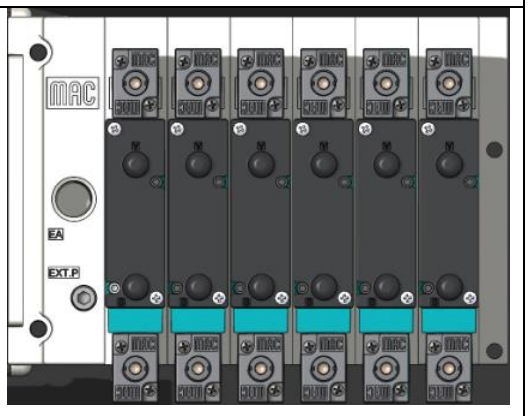
Expansion modules will need to be configured in the PLC before they are active and enabled under the TagName:C.Data tag. Refer to the **Control Manual for MAC Ethernet I/P MI/O-67 Manifold Document: UI-173** for further instructions how to configure expansion modules.

Use the included Configuration AOI to make this step a breeze. See below

Change the column “Style” to HEX.
Set the number of modules installed in TagName:C.Data[0] Byte in HEX.
Set the codes from LEFT to RIGHT for the type of expansion modules added to the communication modules. The first module LEFT of the communication module is Add-On Module #1. Enter the 4 byte code in HEX for each module used in the system up to 12 modules can be added.



Name	Value	Force Mas	Style	Data Type	Description
Valve_Stack:C.Data	{...}	{...}	Hex	SINT[190]	MIO-67 Add-On Module Configuration
Valve_Stack:C.Data[0]	16#00		Hex	SINT	Number of Add-On Modules: #00 (none), #01 (1), #02 (2), #0A (10), etc. [1-12] [#01-#0C hex] A
Valve_Stack:C.Data[1]	16#00		Hex	SINT	Always Set to #00
Valve_Stack:C.Data[2]	16#00		Hex	SINT	Add-On Module #1 Type Configuration Byte 1-4 Required or #00 for none
Valve_Stack:C.Data[3]	16#00		Hex	SINT	Add-On Module #1 Type Configuration Byte 2-4 Required or #00 for none
Valve_Stack:C.Data[4]	16#00		Hex	SINT	Add-On Module #1 Type Configuration Byte 3-4 Required or #00 for none
Valve_Stack:C.Data[5]	16#00		Hex	SINT	Add-On Module #1 Type Configuration Byte 4-4 Required or #00 for none
Valve_Stack:C.Data[6]	16#00		Hex	SINT	Add-On Module #2 Type Configuration Byte 1-4 Required or #00 for none
Valve_Stack:C.Data[7]	16#00		Hex	SINT	Add-On Module #2 Type Configuration Byte 2-4 Required or #00 for none
Valve_Stack:C.Data[8]	16#00		Hex	SINT	Add-On Module #2 Type Configuration Byte 3-4 Required or #00 for none
Valve_Stack:C.Data[9]	16#00		Hex	SINT	Add-On Module #2 Type Configuration Byte 4-4 Required or #00 for none
Valve_Stack:C.Data[10]	16#00		Hex	SINT	Add-On Module #3 Type Configuration Byte 1-4 Required or #00 for none
Valve_Stack:C.Data[11]	16#00		Hex	SINT	Add-On Module #3 Type Configuration Byte 2-4 Required or #00 for none
Valve_Stack:C.Data[12]	16#00		Hex	SINT	Add-On Module #3 Type Configuration Byte 3-4 Required or #00 for none
Valve_Stack:C.Data[13]	16#00		Hex	SINT	Add-On Module #3 Type Configuration Byte 4-4 Required or #00 for none
Valve_Stack:C.Data[14]	16#00		Hex	SINT	Add-On Module #4 Type Configuration Byte 1-4 Required or #00 for none
Valve_Stack:C.Data[15]	16#00		Hex	SINT	Add-On Module #4 Type Configuration Byte 2-4 Required or #00 for none
Valve_Stack:C.Data[16]	16#00		Hex	SINT	Add-On Module #4 Type Configuration Byte 3-4 Required or #00 for none
Valve_Stack:C.Data[17]	16#00		Hex	SINT	Add-On Module #4 Type Configuration Byte 4-4 Required or #00 for none
Valve_Stack:C.Data[18]	16#00		Hex	SINT	Add-On Module #5 Type Configuration Byte 1-4 Required or #00 for none
Valve_Stack:C.Data[19]	16#00		Hex	SINT	Add-On Module #5 Type Configuration Byte 2-4 Required or #00 for none
Valve_Stack:C.Data[20]	16#00		Hex	SINT	Add-On Module #5 Type Configuration Byte 3-4 Required or #00 for none

Add-On Module #4 up to 12	Add-On Module #3	Add-On Module #2	Add-On Module #1	Communication Module	Valves
					

8.1 Analog Module (Voltage)

Code: 0x91,0x01,0x0c,0x81

8.2 Analog Module (Current)

Code: 0x91,0x01,0x0c,0x82

8.3 Power Plus

Code: 0x91,0x01,0x03,0x84

8.4 Digital I/O

Dip Switch 000 Input/NPN:Input/NPN
Code: 0x91,0x01,0x03,0x70

Dip Switch 001 Input /NPN:Output:
Code: 0x91,0x01,0x03,0x71

Dip Switch 010 Input /PNP: Input /PNP
Code: 0x91,0x01,0x03,0x72

Dip Switch 011 Input /PNP:Output
Code: 0x91,0x01,0x03,0x73

Dip Switch 100 Output: Input /NPN
Code: 0x91,0x01,0x03,0x74

Dip Switch 101 Output:Output
Code: 0x91,0x01,0x03,0x75

Dip Switch 110 Output: Input /PNP
Code: 0x91,0x01,0x03,0x76

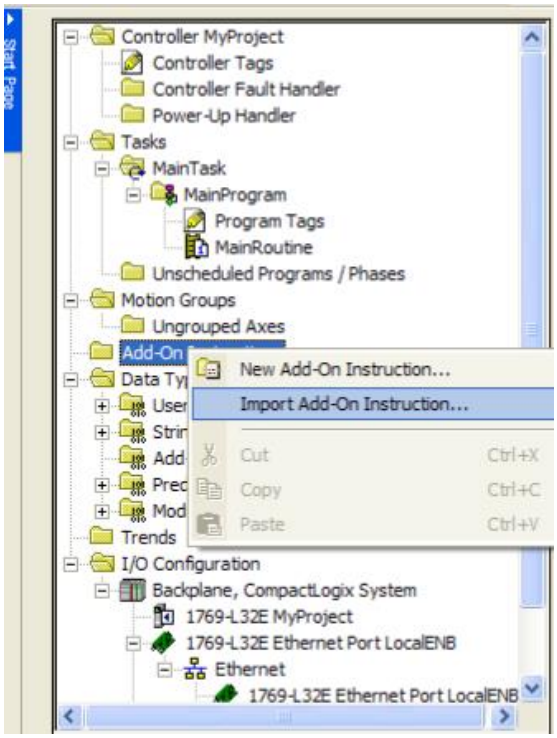
Dip Switch 111 Output:Output
Code: 0x91,0x01,0x03,0x77

9 Importing Add-on Instructions

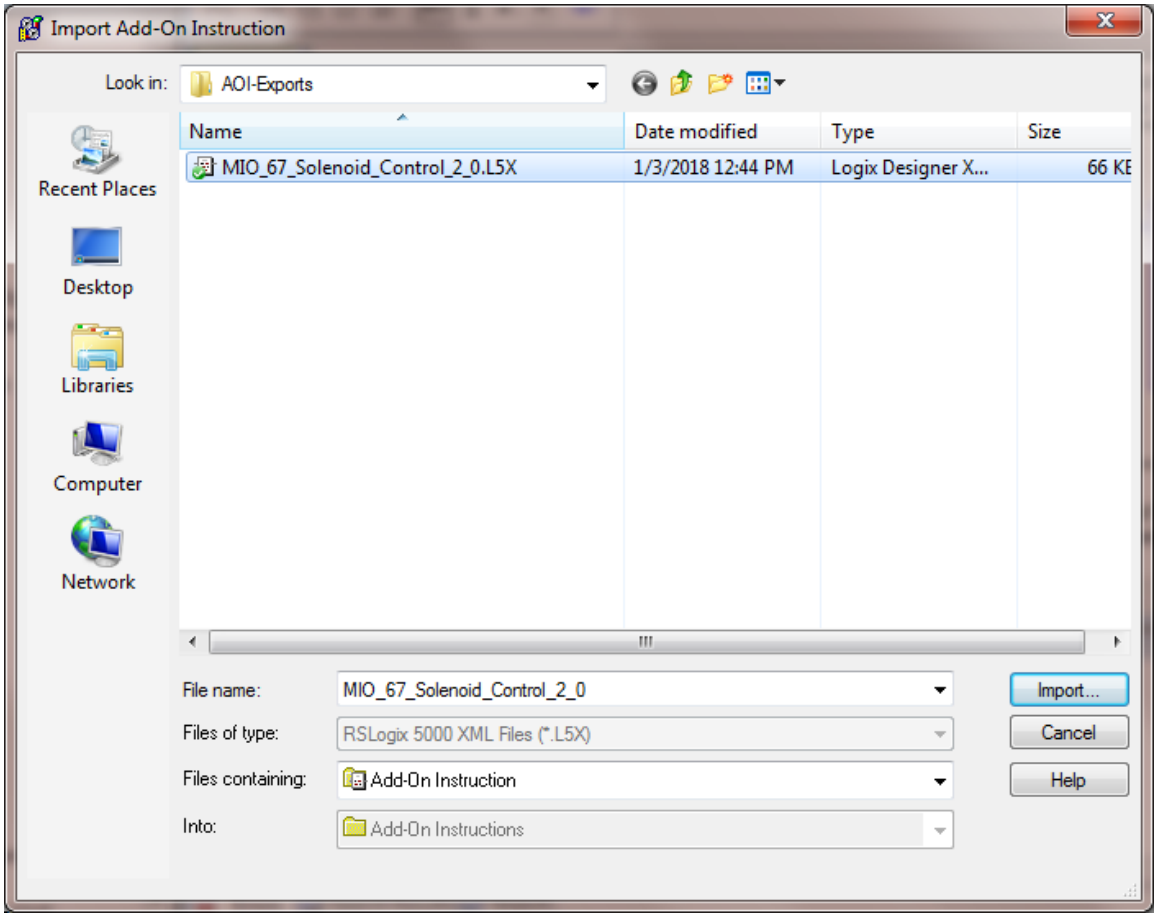
AOIs provided for the MIO 67 are used to ease integration and configuration. AOI must be imported into a project first.

Firmware version 16.0 and GREATER is required to support AOIs. If your PLC firmware is older than version 16.0, AOIs will not be supported in the PLC.

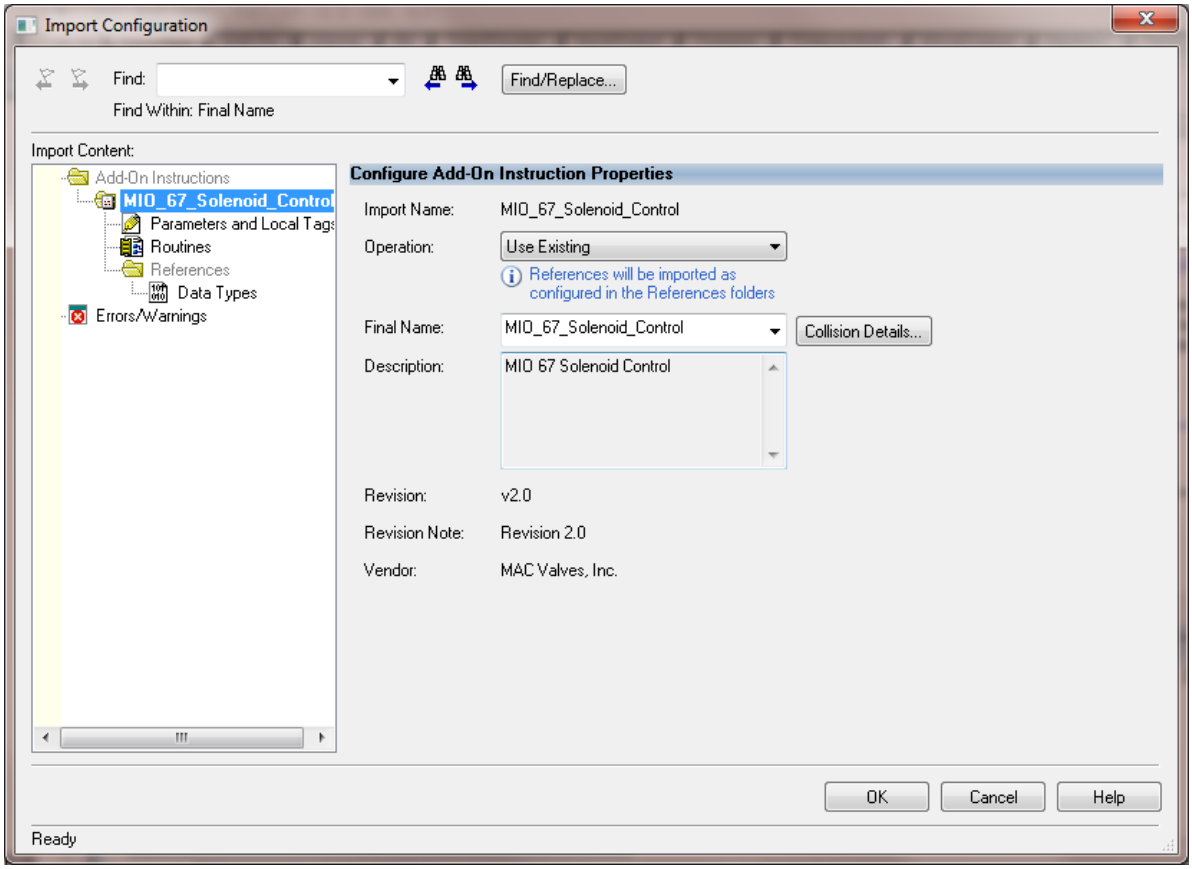
To import an AOI, right click on the Add-On Instruction folder and select “Import Add-On Instruction”



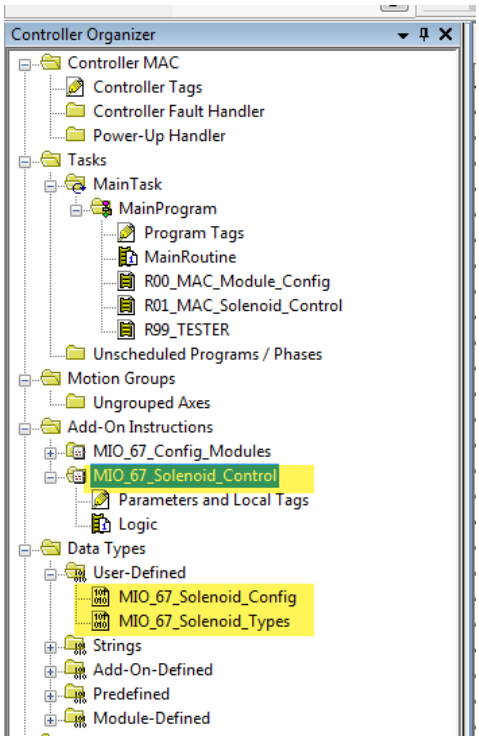
Browse to the AOI-Exports folder from the ZIP file.
Choose the AOIs you would like to import.
In this case, the MIO 67 Solenoid Control AOI is selected to be imported.



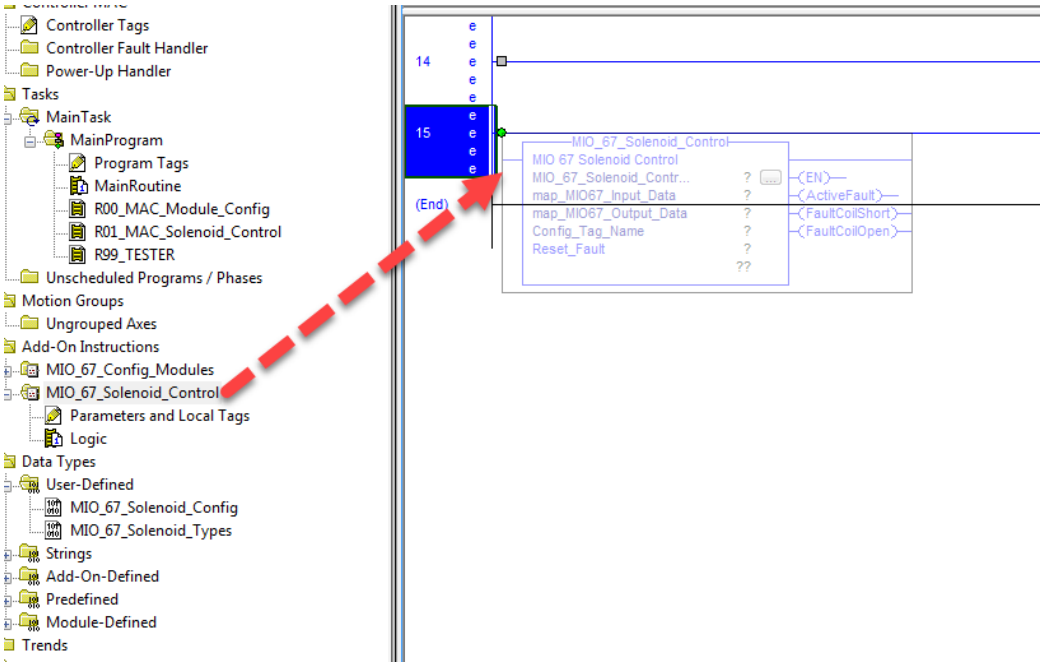
An import confirmation screen will appear. Confirm the AOI version and select OK.



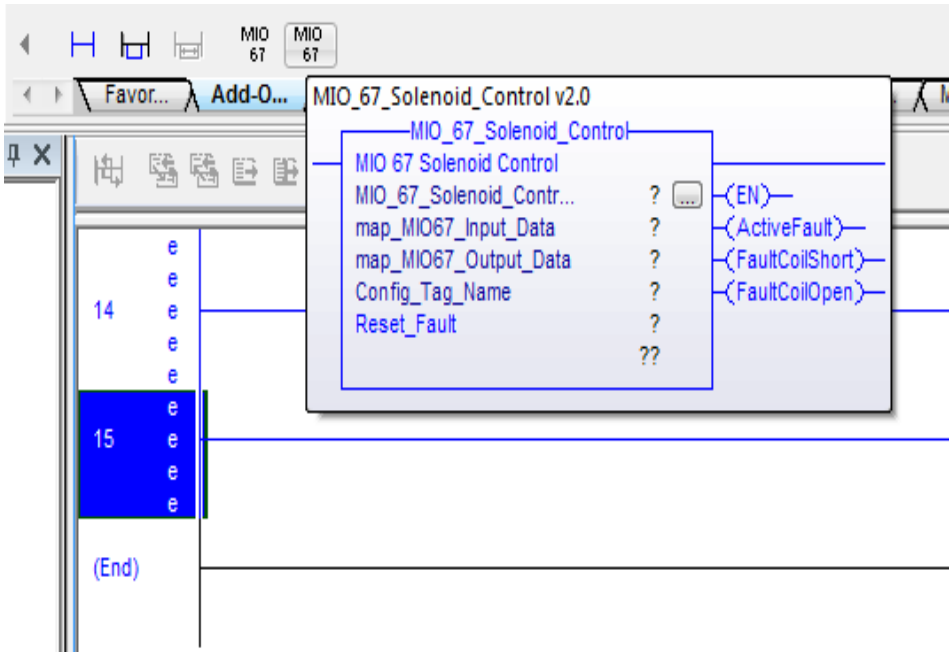
Some AOIs will also import their own UDT (User Defined Types). In this case, the Solenoid Control also imported MIO_67_Solenoid_Config and MIO_67_Solenoid_Types.

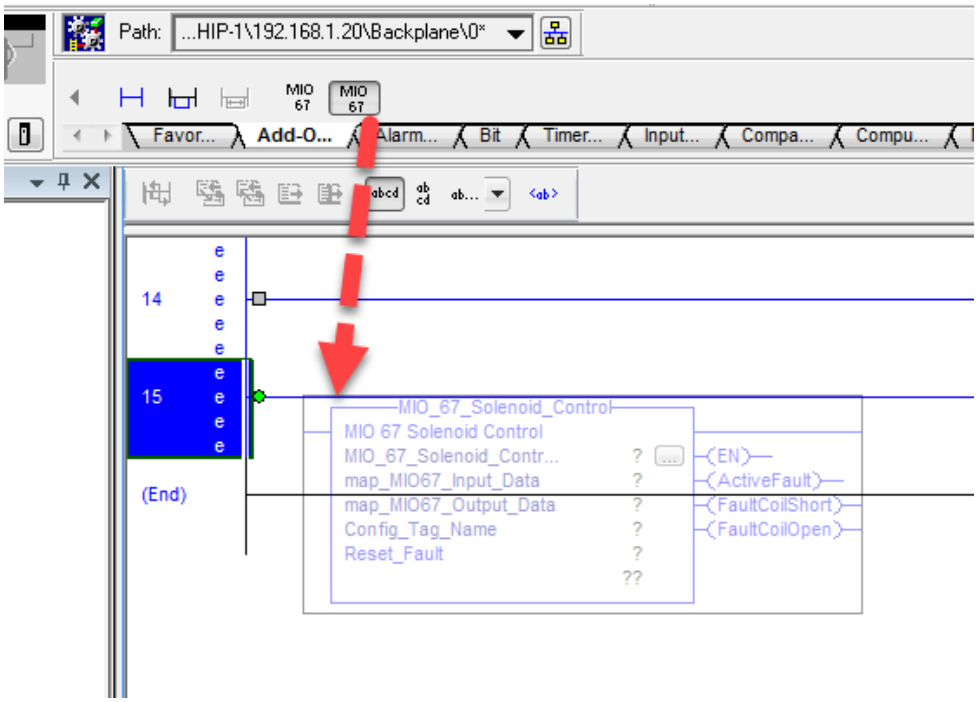


AOIs can be added to your project by dragging and dropping the folder to a rung.



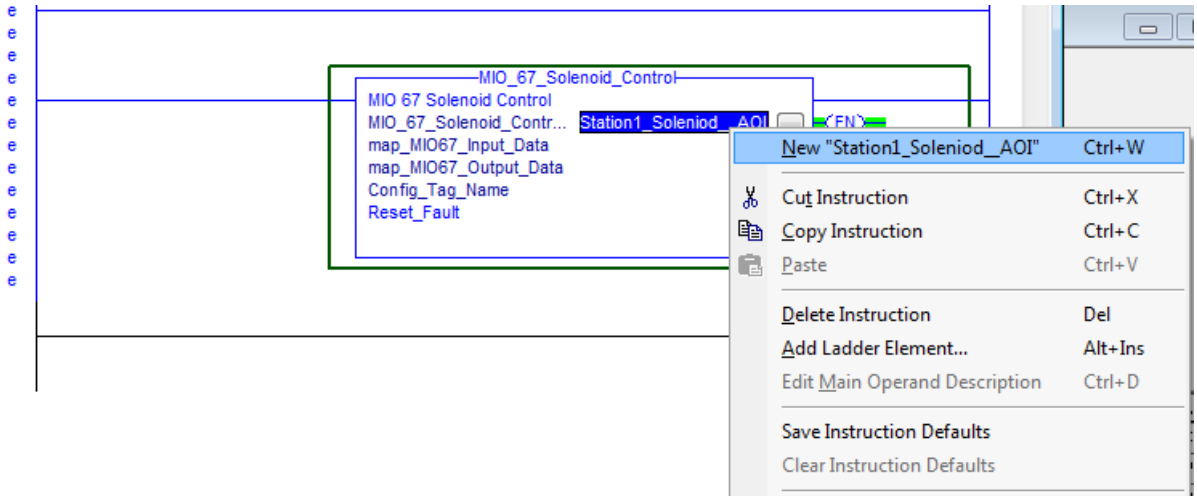
Or by selecting the Add-On toolbar and dragging and dropping from there.





Once AOIs are live on a rung, you may begin assigning tag names.

It is recommended to create a tag name in the AOI block, then right click on the new tag, and choose “New “YOUR TAG NAME”. This will insure that this tag name is properly created with the AOI tag types.

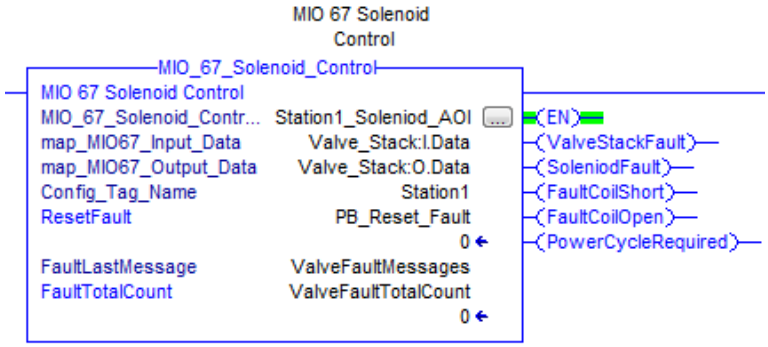


10 USING AOI INSTRUCTIONS IN THE PROJECT

10.1 MIO_67_Solenoid_Control v2.2

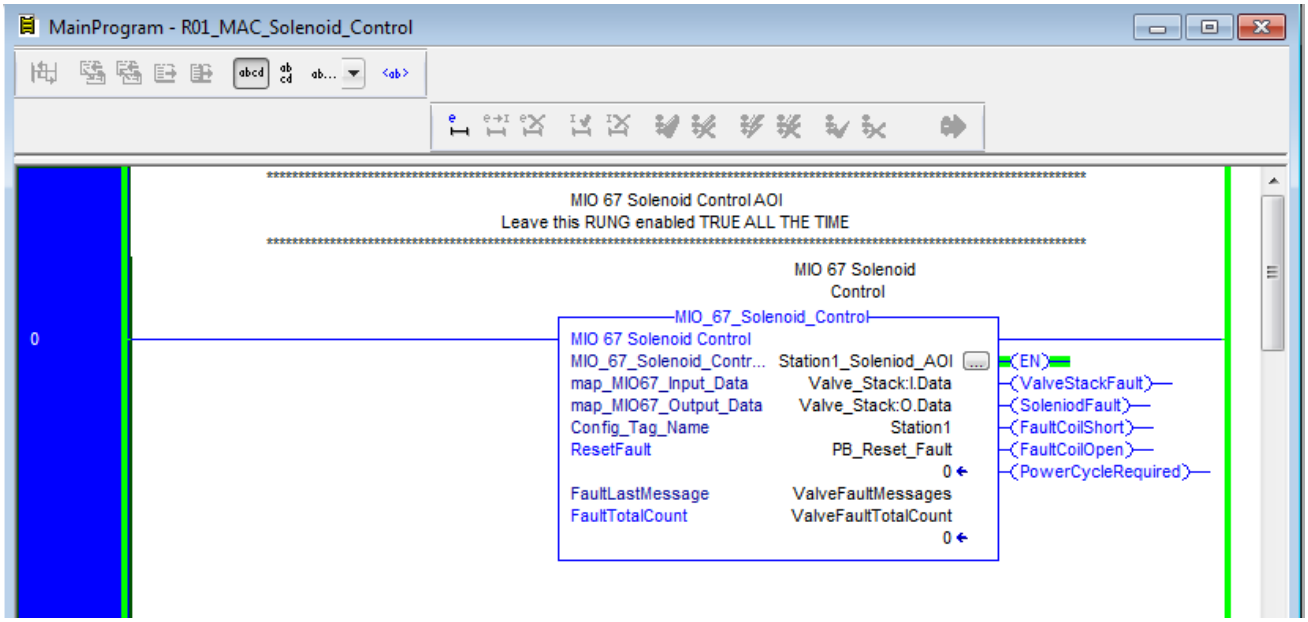
The purpose of this AOI is to provide BOOL or DINT mapping to the 32 solenoid outputs per MIO 67 manifold stations. The AOI also helps check built in fault diagnostics. If any of the solenoids where to fault, the ActiveFault status bit will be a latch with a logical “1” and the matching fault status will also latch. Once a fault is recovered, toggling the Reset_Fault will reset the ActiveFault status bit.

NOTICE: **FaultCoilOpen** faults need to be configured in the MAC webtool or refer to UI-173 for more information. They are not active unless they are enabled.

AOI Sample Tag Names filled out **(VERSION 2.2 AOI shown)**	Parameter Type	Description Examples
	MIO_67_Solenoid_Control AOI Tag Name	User Given
	map_MIO67_Input_data EIP:I.DATA	User Given
	map_MIO67_Output_data EIP:O.DATA	User Given
	Config_Tag_Name Config Tag Name	User Given
	ResetFault BOOL	Instruction Interface Reset Fault Bit 0 = normal 1 = reset fault
	FaultLastMessage STRING	Fault Message Default: No Faults
	FaultTotalCount DINT	Fault Total Active Count Default: 0

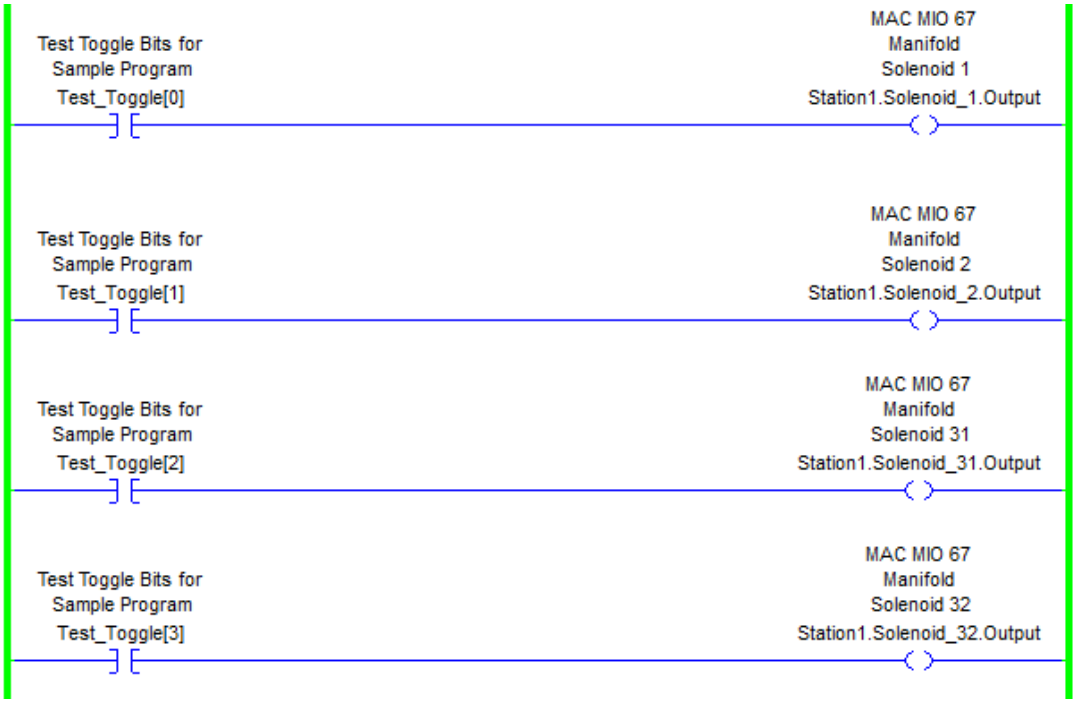
Ladder Logic Example:

Leave the AOI enabled true all the time.



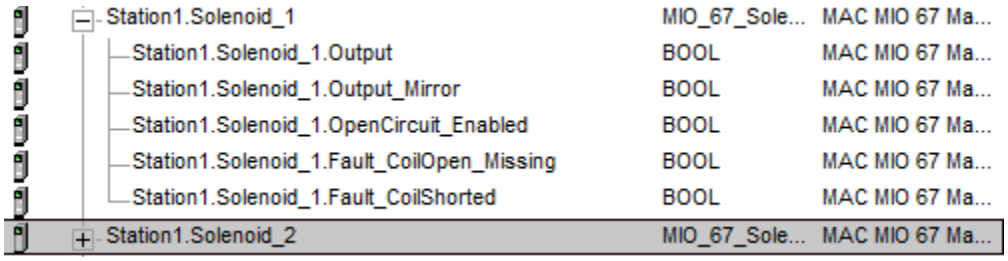
10.1.1 Example Controlling Solenoids via BOOL tag names.

Solenoid outputs are easy to control using the Config_Tag_Name + Solenoid_xx.Output naming convention. Here is an example of controlling Solenoid 1 and 2, 31 and 32.



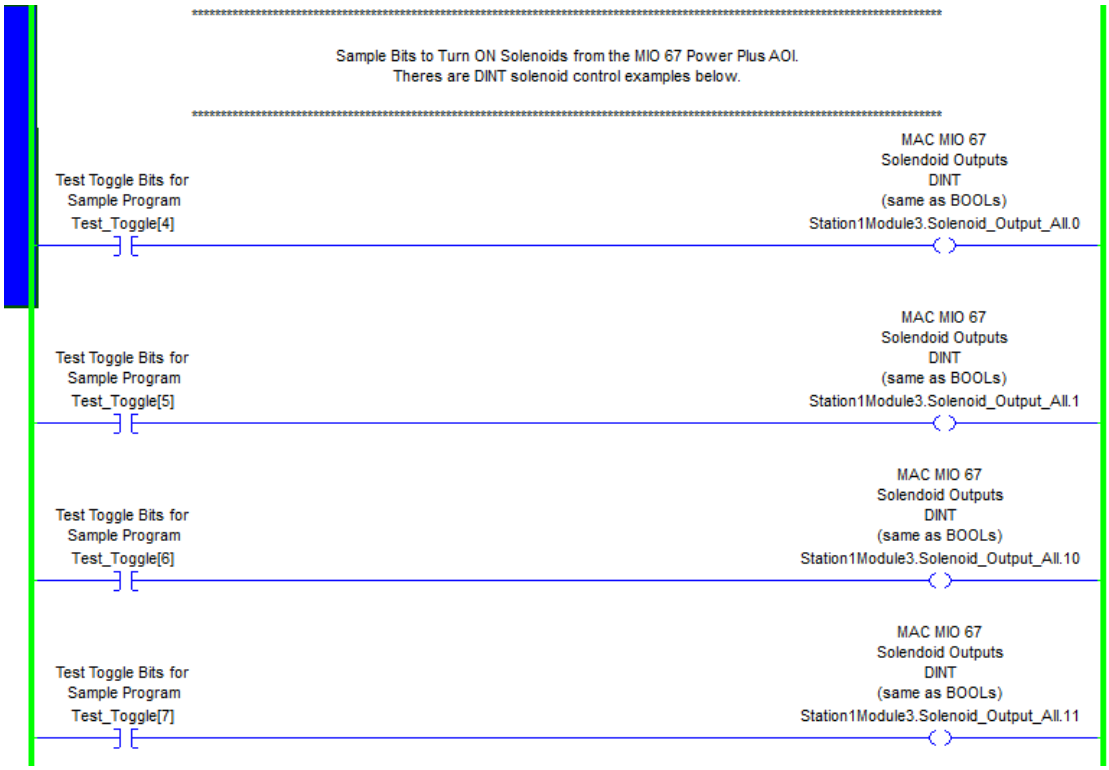
10.1.2 Other Solenoid BOOL Bits

Each Solenoid has different types of information that is accessible about the solenoid, such as output control, output mirror status, whether the open circuit detection is enabled or not, and two kinds of faults. Coil open / missing or Coil shorted.



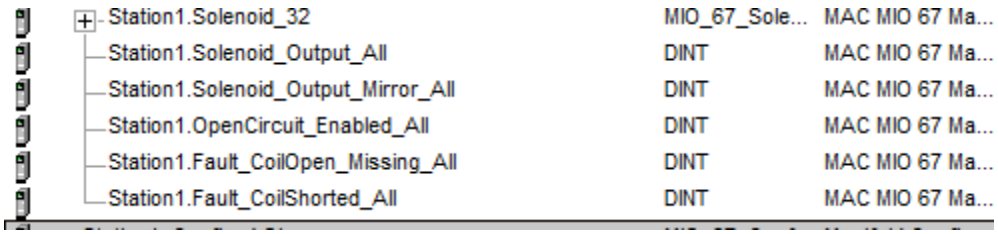
10.1.3 Example Controlling Solenoids via DINT tag names.

Solenoid outputs are easy to control using the Config_Tag_Name + Solenoid_Output_ALL.xx naming convention. Here is an example of controlling Solenoid 1 and 2, 31 and 32.



10.1.4 Other Solenoid DINTs

Other DINT registers are accessible about the solenoids, such as output control, output mirror status, whether the open circuit detection is enabled or not, and two kinds of faults. Coil open / missing or Coil shorted.

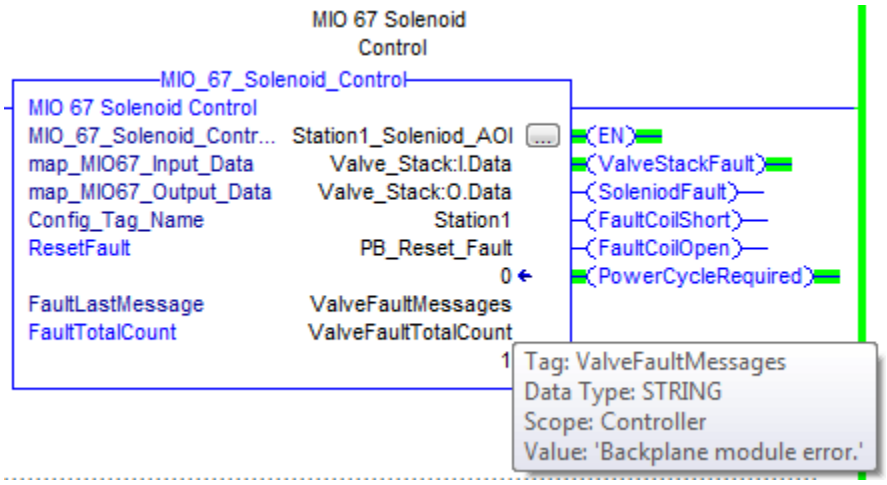


10.1.5 Valve Stack Faults

Valve stack faults where added as BIT FLAGs. They are accessible via the Config_Tag_Name + Fault_Valve_xxxxx naming convention. These bits correlate to the front panel message and fault counts.

Station1.Solenoid_30	{...}	{...}		udt_MIO_67_Solenoid_Types	MAC MIO 67 Manifold Solenoid 30
Station1.Solenoid_31	{...}	{...}		udt_MIO_67_Solenoid_Types	MAC MIO 67 Manifold Solenoid 31
Station1.Solenoid_32	{...}	{...}		udt_MIO_67_Solenoid_Types	MAC MIO 67 Manifold Solenoid 32
Station1.Solenoid_Output_All	0		Decimal	DINT	MAC MIO 67 Manifold Solendoid Outputs DINT (same as BOOLs)
Station1.Solenoid_Output_Mirror_All	0		Decimal	DINT	MAC MIO 67 Manifold Solenoid Output Mirror Statuses DINT (same as BOOLs)
Station1.OpenCircuit_Enabled_All	0		Decimal	DINT	MAC MIO 67 Manifold OpenCircuit Enabled Solenoid Statuses DINT (same as BO
Station1.Fault_CoilOpen_Missing_All	0		Decimal	DINT	MAC MIO 67 Manifold Fault CoilOpen or Missing Solenoid Statuses DINT (same a
Station1.Fault_CoilShorted_All	0		Decimal	DINT	MAC MIO 67 Manifold Fault CoilShorted Solenoid Statuses DINT (same as BOOL
Station1.Fault_Valve_Slave_NOP	0		Decimal	BOOL	MAC MIO 67 Manifold Fault Valve - One or more slaves are not operational
Station1.Fault_Valve_Module_OUT_NOP	0		Decimal	BOOL	MAC MIO 67 Manifold Fault Valve - One or more modules are not operational on th
Station1.Fault_Valve_Module_IN_NOP	0		Decimal	BOOL	MAC MIO 67 Manifold Fault Valve - One or more modules are not operational on th
Station1.Fault_Valve_Invaild_ID	0		Decimal	BOOL	MAC MIO 67 Manifold Fault Valve - Invalid module ID configuration
Station1.Fault_Valve_Config_InProgress	0		Decimal	BOOL	MAC MIO 67 Manifold Fault Valve - Configuration in-progress "Status"
Station1.Fault_Valve_Backplane_Budget	0		Decimal	BOOL	MAC MIO 67 Manifold Fault Valve - Backplane power budget exceeded
Station1.Fault_Valve_Backplane_Error	1		Decimal	BOOL	MAC MIO 67 Manifold Fault Valve - Backplane module error
Station1.Fault_Valve_Memory_Corrupt	0		Decimal	BOOL	MAC MIO 67 Manifold Fault Valve - Parameters store in memory are corrupt
Station1.Fault_Valve_CAN_Passive_Error	0		Decimal	BOOL	MAC MIO 67 Manifold Fault Valve - CAN controller in passive error state
Station1.Fault_Valve_CAN_Recieve_Error	0		Decimal	BOOL	MAC MIO 67 Manifold Fault Valve - CAN receive overrun error
Station1.Fault_Valve_CAN_Transmit_Error	0		Decimal	BOOL	MAC MIO 67 Manifold Fault Valve - CAN transmit overrun error
Station1.Fault_Valve_CAN_Bus_Off	0		Decimal	BOOL	MAC MIO 67 Manifold Fault Valve - CAN BUS-off error
Station1.Fault_Valve_External_IO	0		Decimal	BOOL	MAC MIO 67 Manifold Fault Valve - External I/O error

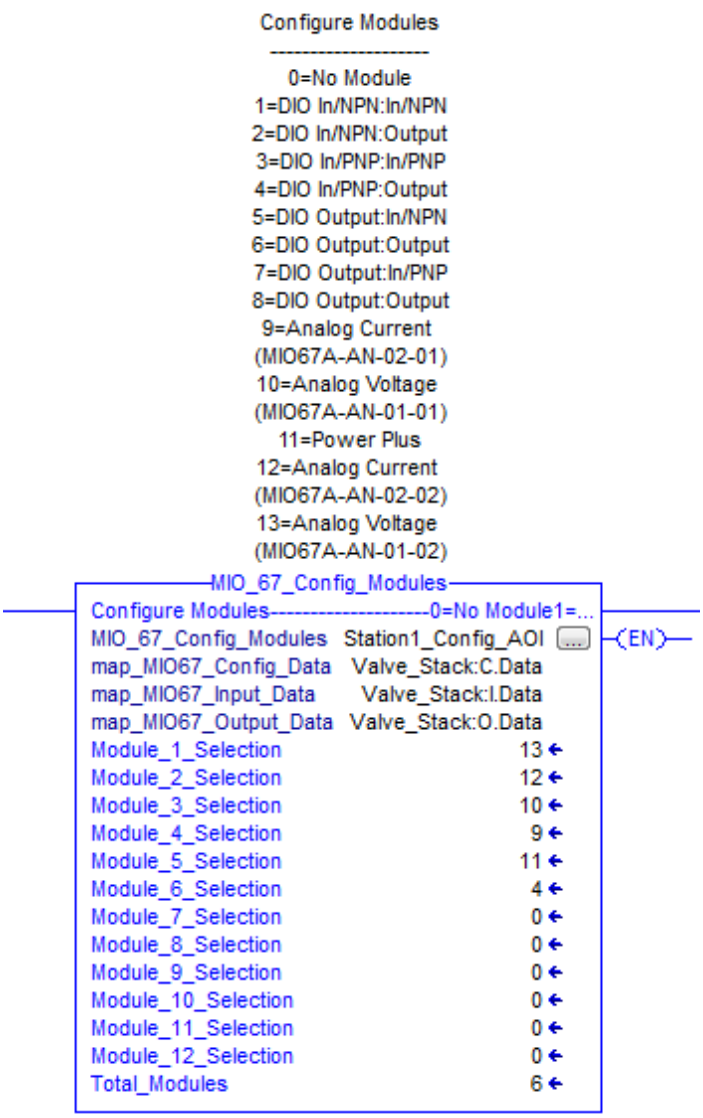
Example: The “ValveStackFault” bit is set. Hovering your mouse over the ValveFaultMessages ONLY shows you the last message in STRING “Text”. The ValveFaultTotalCount is = 1. We can see the Station1.Fault_Valve_Backplane_Error is set. Consult MAC technical support when errors occur.



Station1.Fault_Valve_Module_IN_NOP	0		Decimal	BOOL
Station1.Fault_Valve_Invaild_ID	0		Decimal	BOOL
Station1.Fault_Valve_Config_InProgress	0		Decimal	BOOL
Station1.Fault_Valve_Backplane_Budget	0		Decimal	BOOL
Station1.Fault_Valve_Backplane_Error	1		Decimal	BOOL
Station1.Fault_Valve_Memory_Corrupt	0		Decimal	BOOL
Station1.Fault_Valve_CAN_Passive_Error	0		Decimal	BOOL

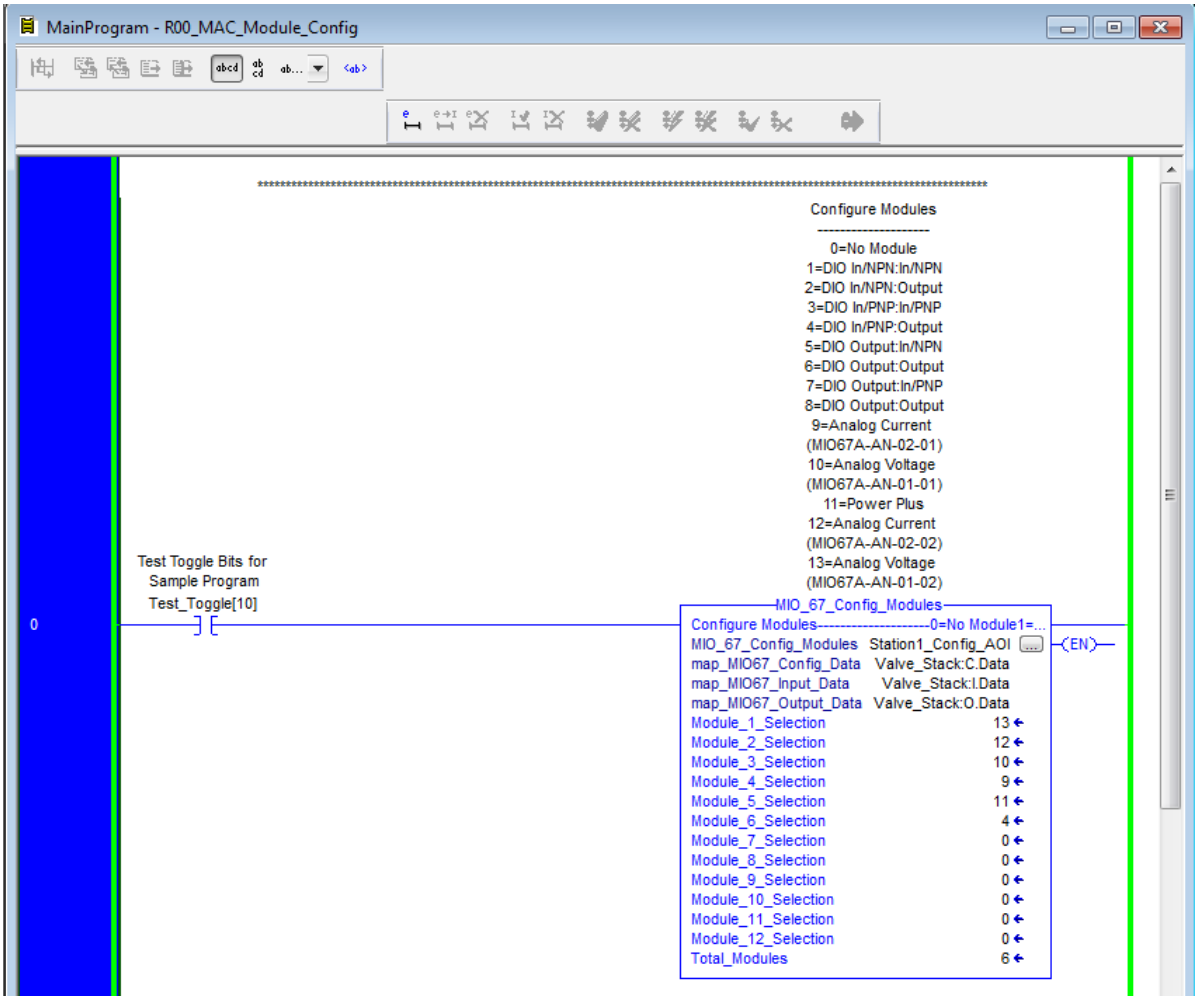
10.2 MIO_67_Config_Modules v3.0

The purpose of this AOI is to correctly write the HEX CODES that are required in the correct memory mapping locations when using “Add-On” modules to the valve stack. The HEX CODE vs memory location is described in U-173 MAC manual. This AOI will make it easy to setup the HEX codes. Add-On modules are number 0 – 13 by type. Enter the type of module in which slot it is physically located.

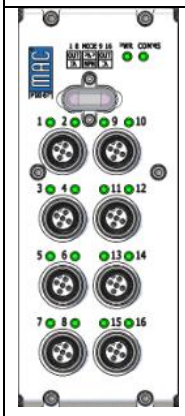
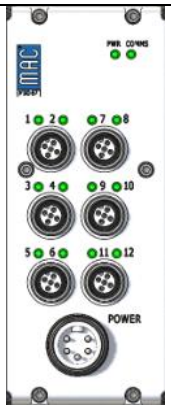
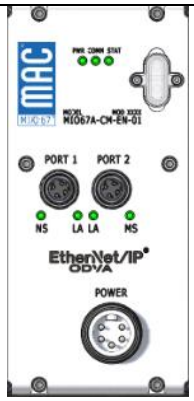
AOI Sample Tag Names filled out **(VERSION 3.0 AOI shown)**	Parameter Type	Description Examples
	MIO_67_Config_Module AOI Tag Name	User Given
	map_MIO67_Config_data EIP:C.DATA	User Given
	map_MIO67_Input_data EIP:I.DATA	User Given
	map_MIO67_Output_data EIP:O.DATA	User Given
	Module_1_Selection Thru Module_12_Selection INT Enter the type number provided either “0-13”	0 = No Module 1 = DIO In/NPN:In/NPN 2 = DIO In/NPN:Output 3 = DIO In/PNP:In/PNP 4 = DIO In/PNP:Output 5 = DIO Output:In/NPN 6 = DIO Output:Output 7 = DIO Output:In/PNP 8 = DIO Output:Output 9 = Analog Current (MIO67A-AN-02-01) 10 = Analog Voltage (MIO67A-AN-01-01) 11 = Power Plus 12 = Analog Current (MIO67A-AN-02-02) 13 = Analog Voltage (MIO67A-AN-01-02)
	Total_Modules DINT	Total Modules Count Configured by Customer Default: Total Counts

Ladder Logic Example and step-by-step process:

1. Create a Test_Toggle bit for this AOI.
2. Type in the module numbers that match the module selection spaces
3. Total_Modules is an output confirmation of how many total add-on blocks are written.
4. Toggle the Test bit TRUE, the Toggle the Test bit FALSE.
5. Leave Config FALSE and only use again to setup a new manifold or change add-on blocks.
6. POWER CYCLE the valve manifold after writing configuration to valve stack.
7. Make sure the Solenoid Control AOI does not show any Valve Faults after configuration is set.

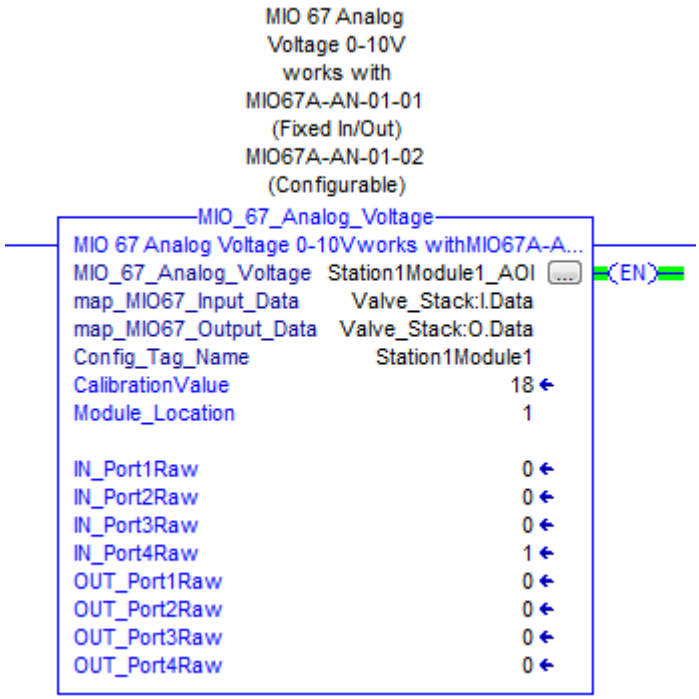


Example above shown

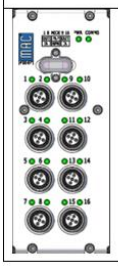
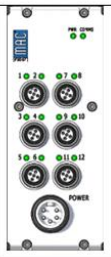
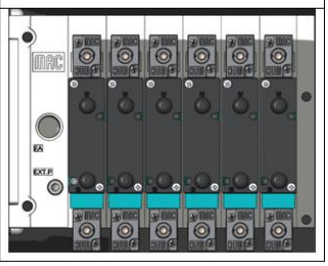
Add-On Module #4 Digital I/O Dip Switches 0 – 0 – 0 In/NPN : In/NPN **Selection “1”**	Add-On Module #3 Power Plus **Selection “11”**	Add-On Module #2 Analog CURRENT (depending on type) **Selection “9” Or Selection “12”**	Add-On Module #1 Analog VOLTAGE (depending on type) **Selection “10” Or Selection “13”**	Communication Module	Valves
					

10.3 MIO_67_Analog_Voltage v2.0

The purpose of this AOI is to get the voltage inputs and outputs from an add-on module. Config tags are created to allow a programmer easy scaling utilizing the SCP (Scale with Parameters) method of using analog controls.

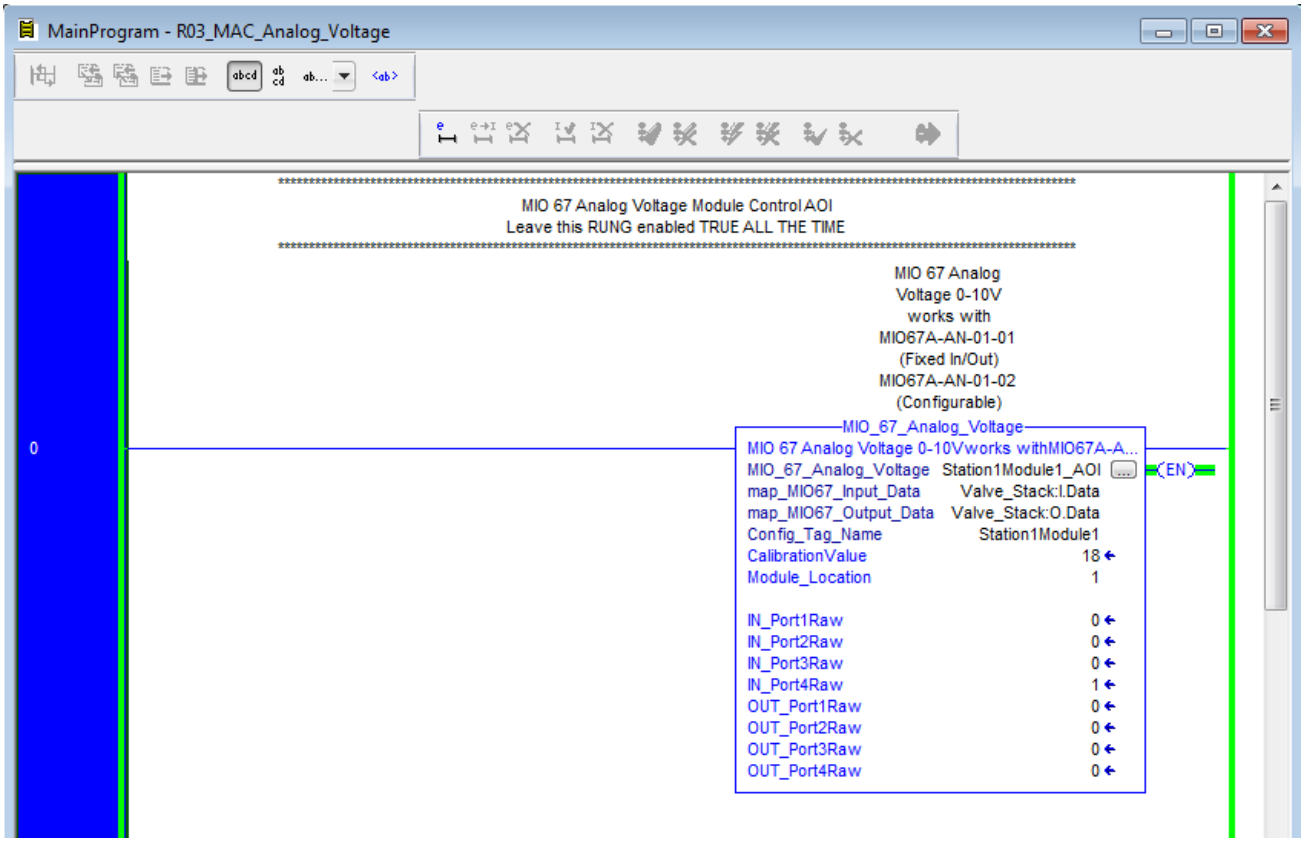
AOI Sample Tag Names filled out **(VERSION 2.0 AOI shown)**	Parameter Type	Description Examples
	MIO_67_Analog_Voltage AOI Tag Name	User Given
	map_MIO67_Input_data EIP:I.DATA	User Given
	map_MIO67_Output_data EIP:O.DATA	User Given
	Config_Tag_Name Config Tag Name	User Given
	CalibrationValue DINT	Raw Value Calibration Used to “tweak” analog to scale value. Default: 0
	Module_Location DINT	Module Location Number 1-12 Default: Declare the module location, see picture below as example
	IN_Port1Raw IN_Port2Raw IN_Port3Raw IN_Port4Raw INT	Input Port Raw Value Use Config_Tag for expanded analog features Range: 0-4096
	OUT_Port1Raw OUT_Port2Raw OUT_Port3Raw OUT_Port4Raw INT	Output Port Raw Value Use Config_Tag for expanded analog features Range: 0-4096

Example: Analog Voltage is located in Module #1 location from EIP Adaptor.

Add-On Module #4 Digital I/O Dip Switches 0 – 0 – 0 In/NPN : In/NPN Selection "1"	Add-On Module #3 Power Plus Selection "11"	Add-On Module #2 Analog CURRENT Selection "9"	Add-On Module #1 Analog VOLTAGE Selection "10"	Communication Module	Valves
					

Ladder Logic Example:

Leave the AOI enabled true all the time.



10.3.1 Example Tag Port_x Structure

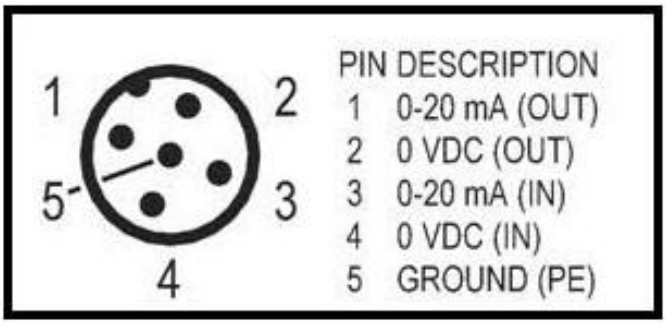
Port access can be written or read from Config_Tag_Name + Portx naming convention.

Station1Module1			udt_MIO_67_Module_Config	MAC MIO 67 Add-On Module	Read/Write
Station1Module1.Port_1			udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 1	Read/Write
Station1Module1.Port_2			udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 2	Read/Write
Station1Module1.Port_3			udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 3	Read/Write
Station1Module1.Port_4			udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 4	Read/Write
Station1Module1.Port_5			udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 5	Read/Write
Station1Module1.Port_6			udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 6	Read/Write
Station1Module1.Port_7			udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 7	Read/Write
Station1Module1.Port_8			udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 8	Read/Write

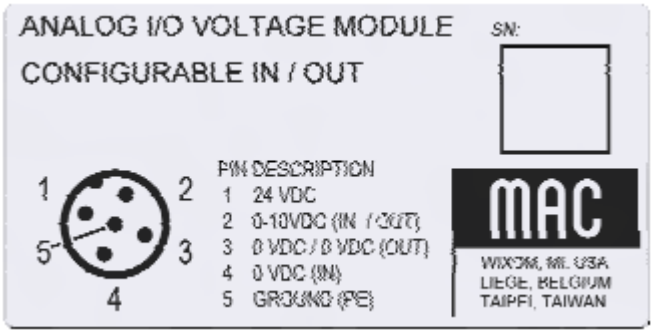
Station1Module1	{...}	{...}		udt_MIO_67_Module_Config	MAC MIO 67 Add-On Module
Station1Module1.Port_1	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 1
Station1Module1.Port_1.IN1	0		Decimal	BOOL	MAC MIO 67 Add-On Module Input #1 I/O Condition
Station1Module1.Port_1.IN2	0		Decimal	BOOL	MAC MIO 67 Add-On Module Input #2 I/O Condition
Station1Module1.Port_1.IN_Raw	137		Decimal	INT	MAC MIO 67 Add-On Module Input Analog Raw Value
Station1Module1.Port_1.IN_Voltage	0.33447266		Float	REAL	MAC MIO 67 Add-On Module Input Analog Voltage Value
Station1Module1.Port_1.IN_Current	0.0		Float	REAL	MAC MIO 67 Add-On Module Input Analog Current Value
Station1Module1.Port_1.IN_Scaled_Min	0.0		Float	REAL	MAC MIO 67 Add-On Module Intput Desired Scaled Min Value
Station1Module1.Port_1.IN_Scaled_Max	360.0		Float	REAL	MAC MIO 67 Add-On Module Intput Desired Scaled Max Value
Station1Module1.Port_1.IN_Scaled_Output	12.128906		Float	REAL	MAC MIO 67 Add-On Module Input User Scaled Output
Station1Module1.Port_1.OUT1	0		Decimal	BOOL	MAC MIO 67 Add-On Module Output #1 I/O Control
Station1Module1.Port_1.OUT2	0		Decimal	BOOL	MAC MIO 67 Add-On Module Output #2 I/O Control
Station1Module1.Port_1.OUT_Raw	0		Decimal	INT	MAC MIO 67 Add-On Module Output Analog Raw Value
Station1Module1.Port_1.OUT_Voltage	0.0		Float	REAL	MAC MIO 67 Add-On Module Output Analog Voltage Value
Station1Module1.Port_1.OUT_Current	0.0		Float	REAL	MAC MIO 67 Add-On Module Output Analog Current Value
Station1Module1.Port_1.OUT_Scaled_Min	0.0		Float	REAL	MAC MIO 67 Add-On Module Output Desired Scaled Min Value
Station1Module1.Port_1.OUT_Scaled_Max	360.0		Float	REAL	MAC MIO 67 Add-On Module Output Desired Scaled Max Value
Station1Module1.Port_1.OUT_Scaled_Output	0.0		Float	REAL	MAC MIO 67 Add-On Module Output User Scaled Output

Port_x.IN1	**NOT USED for Analog Voltage**
Port_x.IN2	**NOT USED for Analog Voltage**
Port_x.IN_Raw	Input Pin 1 Raw INT value from 0-4096 same as front panel values
Port_x.IN_Voltage	Input Pin 1 Scaled value in Voltage 0-10 Volts
Port_x.IN_Current	**NOT USED for Analog Voltage**
Port_x.IN_Scaled_Min	Input scaled value Minimium = 0 Volts
Port_x.IN_Scaled_Max	Input scaled value Maximium = 10 Volts
Port_x.IN_Scaled_Output	Input scaled value actual
Port_x.OUT1	**NOT USED for Analog Voltage**
Port_x.OUT2	**NOT USED for Analog Voltage**
Port_x.OUT _Raw	Output Pin 3 Raw INT value from 0-4096 same as front panel values
Port_x.OUT _Voltage	Output Pin 3 Scaled value in Voltage 0-10 Volts
Port_x.OUT _Current	**NOT USED for Analog Voltage**
Port_x.OUT _Scaled_Min	Output scaled value Minimium = 0 Volts
Port_x.OUT _Scaled_Max	Output scaled value Maximium = 10 Volts
Port_x.OUT_Scaled_Output	Output scaled value actual

10.3.2 Example Analog Voltage Pinout – Fixed Type - MIO67A-AN-01-01

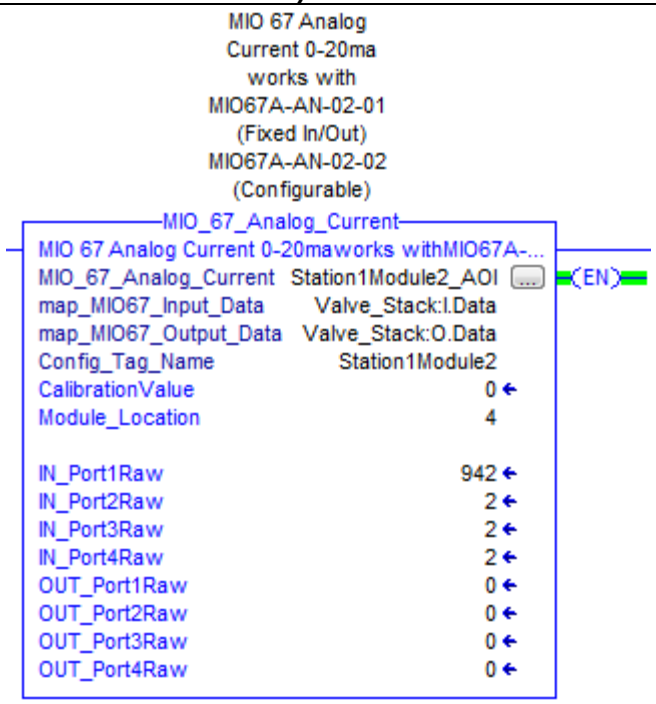


10.3.3 Example Analog Voltage Pinout – Configurable Type - MIO67A-AN-01-02

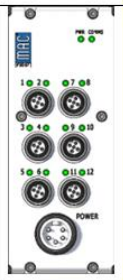
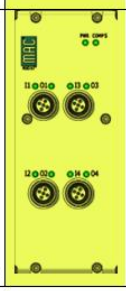
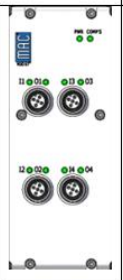
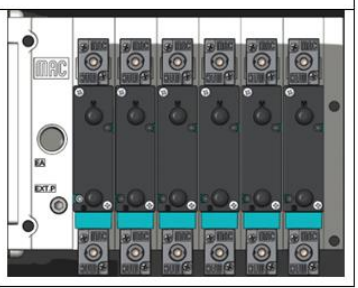


10.4 MIO_67_Analog_Current v2.0

The purpose of this AOI is to get the current inputs and outputs from an add-on module. Config tags are created to allow a programmer easy scaling utilizing the SCP (Scale with Parameters) method of using analog controls.

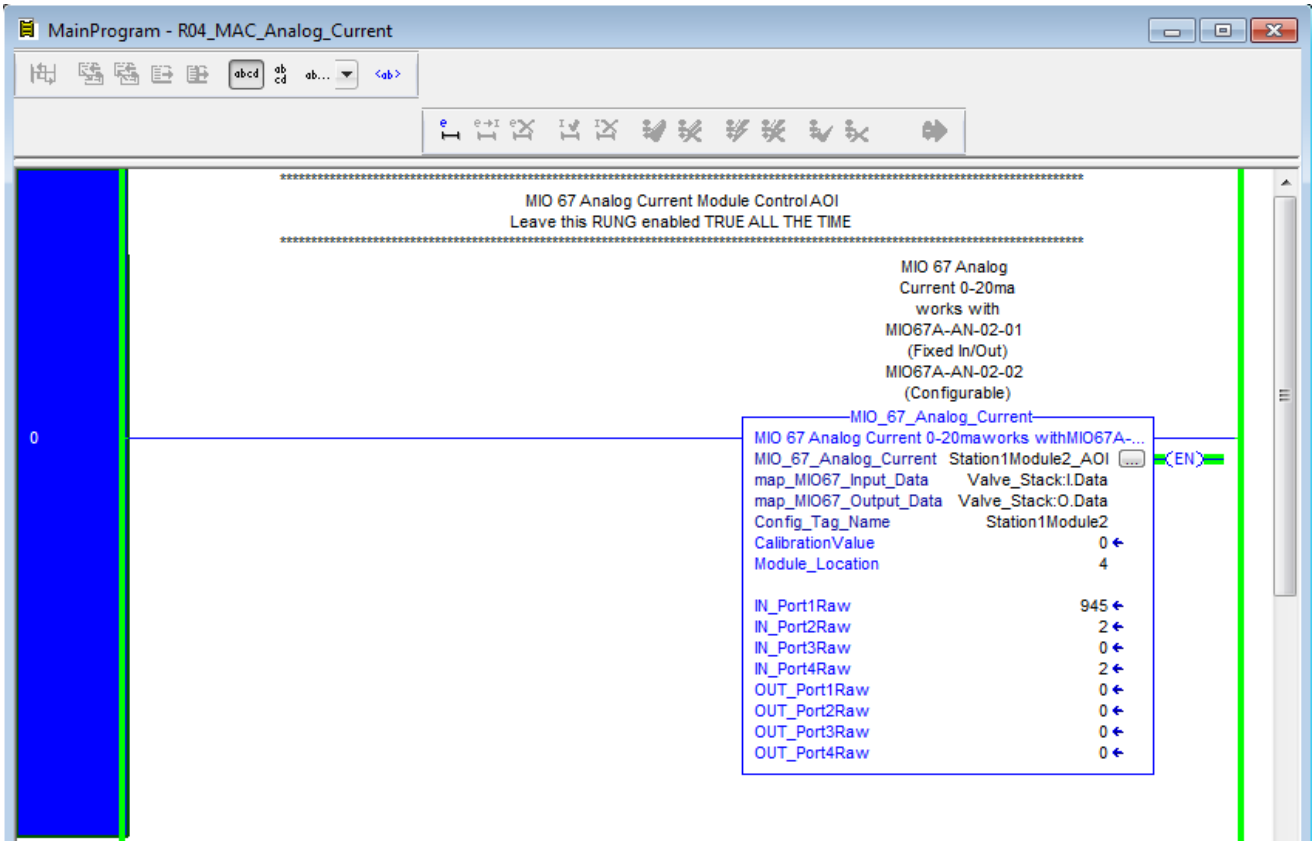
AOI Sample Tag Names filled out **(VERSION 2.0 AOI shown)**	Parameter Type	Description Examples
	MIO_67_Analog_Current AOI Tag Name	User Given
	map_MIO67_Input_data EIP:I.DATA	User Given
	map_MIO67_Output_data EIP:O.DATA	User Given
	Config_Tag_Name Config Tag Name	User Given
	CalibrationValue DINT	Raw Value Calibration Used to “tweak” analog to scale value. Default: 0
	Module_Location DINT	Module Location Number 1-12 Default: Declare the module location, see picture below as example
	IN_Port1Raw IN_Port2Raw IN_Port3Raw IN_Port4Raw INT	Input Port Raw Value Use Config_Tag for expanded analog features Range: 0-4096
	OUT_Port1Raw OUT_Port2Raw OUT_Port3Raw OUT_Port4Raw INT	Output Port Raw Value Use Config_Tag for expanded analog features Range: 0-4096

Example: Analog Current is located in Module #2 location from EIP Adaptor.

Add-On Module #4 Digital I/O Dip Switches 0 – 0 – 0 In/NPN : In/NPN Selection "1"	Add-On Module #3 Power Plus Selection "11"	Add-On Module #2 Analog CURRENT Selection "9"	Add-On Module #1 Analog VOLTAGE Selection "10"	Communication Module	Valves
					

Ladder Logic Example:

Leave the AOI enabled true all the time.



10.4.1 Example Tag Port_x Structure

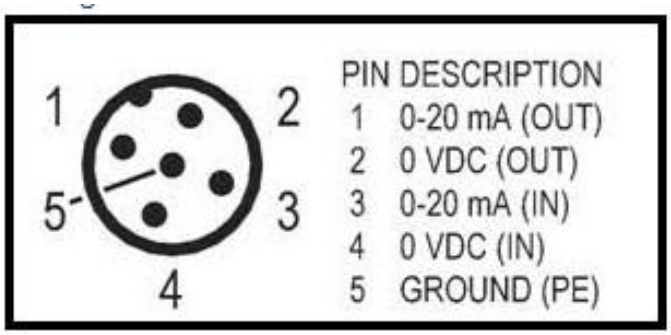
Port access can be written or read from Config_Tag_Name + Portx naming convention.

⊞ Station1Module1	{...}	{...}		udt_MIO_67_Module_Config	MAC MIO 67 Add-On Module
⊟ Station1Module2	{...}	{...}		udt_MIO_67_Module_Config	MAC MIO 67 Add-On Module
⊞ Station1Module2.Port_1	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 1
⊞ Station1Module2.Port_2	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 2
⊞ Station1Module2.Port_3	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 3
⊞ Station1Module2.Port_4	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 4
⊞ Station1Module2.Port_5	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 5
⊞ Station1Module2.Port_6	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 6
⊞ Station1Module2.Port_7	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 7
⊞ Station1Module2.Port_8	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 8

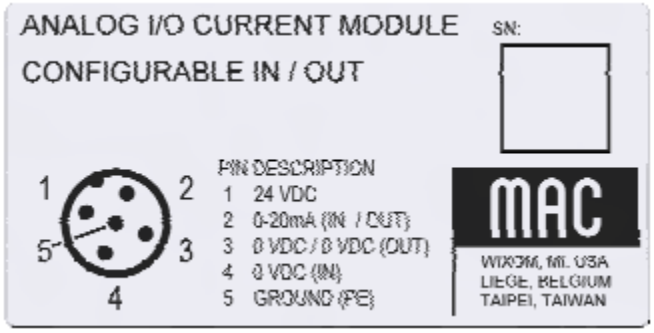
⊟ Station1Module2.Port_1	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 1
Station1Module2.Port_1.IN1	0		Decimal	BOOL	MAC MIO 67 Add-On Module Input #1 I/O Condition
Station1Module2.Port_1.IN2	0		Decimal	BOOL	MAC MIO 67 Add-On Module Input #2 I/O Condition
⊞ Station1Module2.Port_1.IN_Raw	1049		Decimal	INT	MAC MIO 67 Add-On Module Input Analog Raw Value
Station1Module2.Port_1.IN_Voltage	0.0		Float	REAL	MAC MIO 67 Add-On Module Input Analog Voltage Value
Station1Module2.Port_1.IN_Current	6.1506886		Float	REAL	MAC MIO 67 Add-On Module Input Analog Current Value
Station1Module2.Port_1.IN_Scaled_Min	0.0		Float	REAL	MAC MIO 67 Add-On Module Intput Desired Scaled Min Value
Station1Module2.Port_1.IN_Scaled_Max	360.0		Float	REAL	MAC MIO 67 Add-On Module Intput Desired Scaled Max Value
Station1Module2.Port_1.IN_Scaled_Output	110.712395		Float	REAL	MAC MIO 67 Add-On Module Input User Scaled Output
Station1Module2.Port_1.OUT1	0		Decimal	BOOL	MAC MIO 67 Add-On Module Output #1 I/O Control
Station1Module2.Port_1.OUT2	0		Decimal	BOOL	MAC MIO 67 Add-On Module Output #2 I/O Control
⊞ Station1Module2.Port_1.OUT_Raw	0		Decimal	INT	MAC MIO 67 Add-On Module Output Analog Raw Value
Station1Module2.Port_1.OUT_Voltage	0.0		Float	REAL	MAC MIO 67 Add-On Module Output Analog Voltage Value
Station1Module2.Port_1.OUT_Current	0.0		Float	REAL	MAC MIO 67 Add-On Module Output Analog Current Value
Station1Module2.Port_1.OUT_Scaled_Min	0.0		Float	REAL	MAC MIO 67 Add-On Module Output Desired Scaled Min Value
Station1Module2.Port_1.OUT_Scaled_Max	360.0		Float	REAL	MAC MIO 67 Add-On Module Output Desired Scaled Max Value
Station1Module2.Port_1.OUT_Scaled_Output	0.0		Float	REAL	MAC MIO 67 Add-On Module Output User Scaled Output

Port_x.IN1	**NOT USED for Analog Current**
Port_x.IN2	**NOT USED for Analog Current**
Port_x.IN_Raw	Input Pin 1 Raw INT value from 0-4096 same as front panel values
Port_x.IN_Voltage	**NOT USED for Analog Current**
Port_x.IN_Current	Input Pin 1 Scaled value in Current 0-20 mA
Port_x.IN_Scaled_Min	Input scaled value Minimium = 0 mA
Port_x.IN_Scaled_Max	Input scaled value Maximium = 20 mA
Port_x.IN_Scaled_Output	Input scaled value actual
Port_x.OUT1	**NOT USED for Analog Current**
Port_x.OUT2	**NOT USED for Analog Current**
Port_x.OUT _Raw	Output Pin 3 Raw INT value from 0-4096 same as front panel values
Port_x.OUT _Voltage	**NOT USED for Analog Current**
Port_x.OUT _Current	Output Pin 3 Scaled value in Current 0-20 mA
Port_x.OUT _Scaled_Min	Output scaled value Minimium = 0 mA
Port_x.OUT _Scaled_Max	Output scaled value Maximium = 20 mA
Port_x.OUT_Scaled_Output	Output scaled value actual

10.4.2 Example Analog Current Pinout – Fixed Type - MIO67A-AN-02-01



10.4.3 Example Analog Current Pinout – Configurable Type - MIO67A-AN-02-02



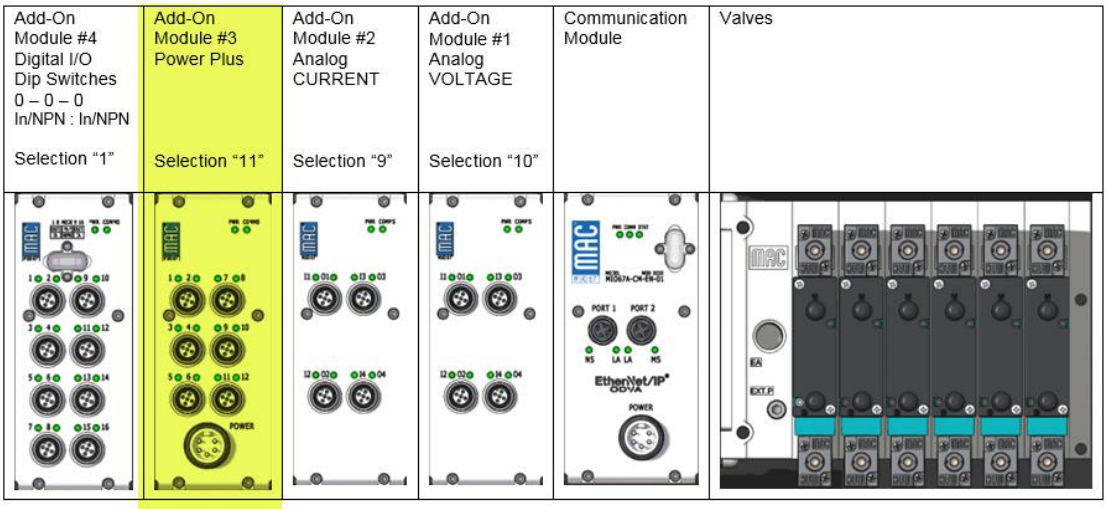
10.5 MIO_67_Power_Plus v1.0

The purpose of this AOI is to provide BOOL or DINT mapping to the 12 solenoid outputs per MIO 67 manifold stations. The AOI also helps check built in fault diagnostics. If any of the solenoids where to fault, the ActiveFault status bit will be a latch with a logical “1” and the matching fault status will also latch. Once a fault is recovered, toggling the Reset_Fault will reset the ActiveFault status bit.

NOTICE: **FaultCoilOpen** faults need to be configured in the MAC webtool or refer to UI-173 for more information. They are not active unless they are enabled. The power plug outputs require an additional power connection to operate.

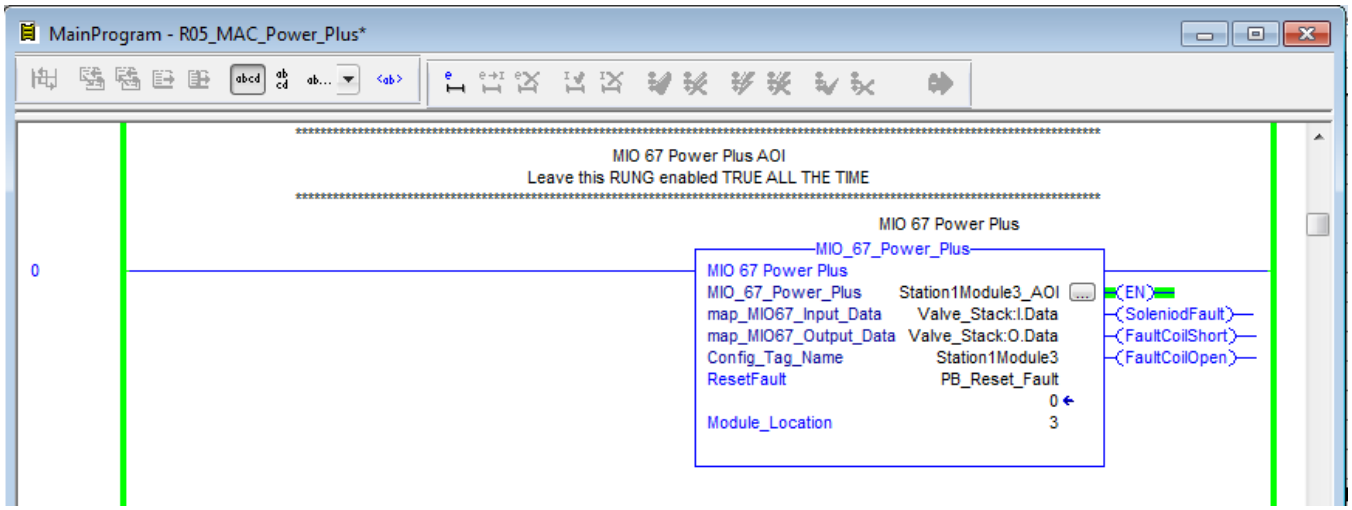
AOI Sample Tag Names filled out **(VERSION 1.0 AOI shown)**	Parameter Type	Description Examples
MIO 67 Power Plus MIO_67_Power_Plus MIO 67 Power Plus MIO_67_Power_Plus Station1Module3_AOI map_MIO67_Input_Data Valve_Stack:I.Data map_MIO67_Output_Data Valve_Stack:O.Data Config_Tag_Name Station1Module3 ResetFault PB_Reset_Fault 0 Module_Location 3 EN SoleniodFault FaultCoilShort FaultCoilOpen	MIO_67_Power_Plus AOI Tag Name	User Given
	map_MIO67_Input_data EIP:I.DATA	User Given
	map_MIO67_Output_data EIP:O.DATA	User Given
	Config_Tag_Name Config Tag Name	User Given
	ResetFault BOOL	Instruction Interface Reset Fault Bit 0 = normal 1 = reset fault
	Module_Location DINT	

Example: Power Plus is located in Module #3 location from EIP Adaptor.



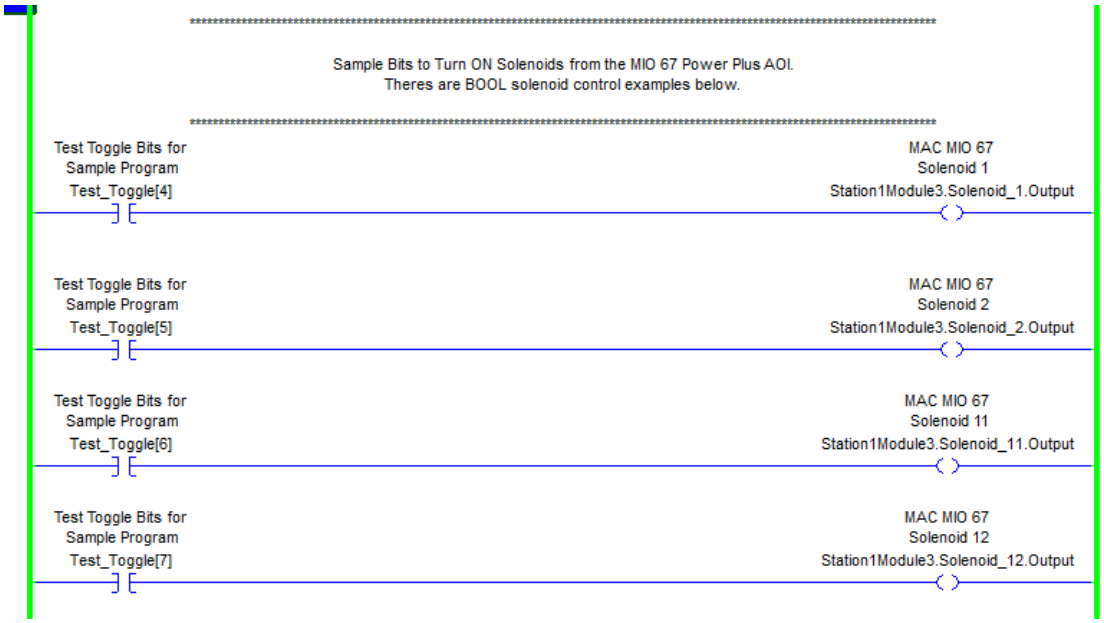
Ladder Logic Example:

Leave the AOI enabled true all the time.



10.5.1 Example Controlling Solenoids via BOOL tag names.

Solenoid outputs are easy to control using the Config_Tag_Name + Solenoid_xx.Output naming convention. Here is an example of controlling Solenoid 1 and 2, 11 and 12.



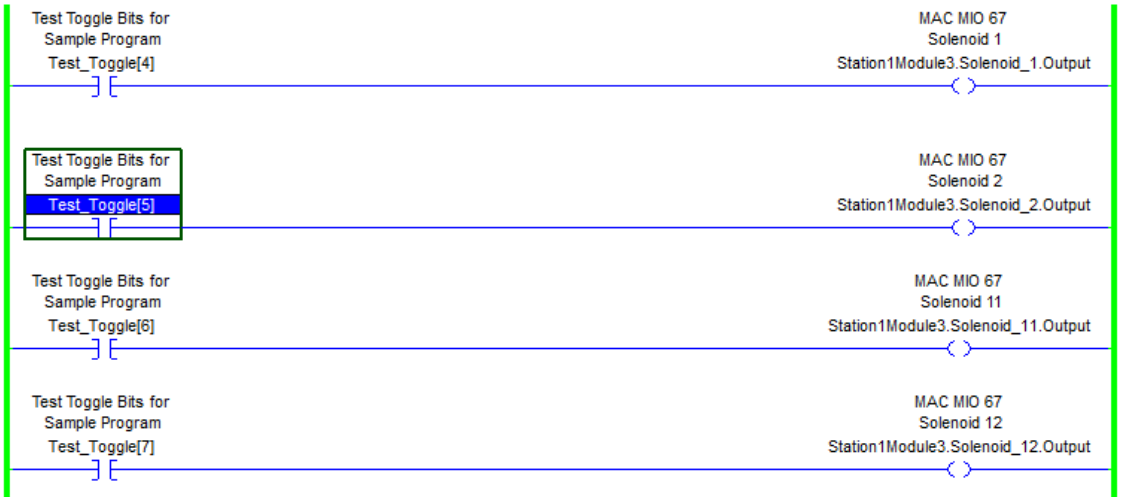
10.5.2 Other Solenoid BOOL Bits

Each Solenoid has different types of information that is accessible about the solenoid, such as output control, output mirror status, whether the open circuit detection is enabled or not, and two kinds of faults. Coil open / missing or Coil shorted.

Station1Module3	{...}	{...}			udt_MIO_67_Solenoid_Config
Station1Module3.Solenoid_1	{...}	{...}			udt_MIO_67_Solenoid_Types
Station1Module3.Solenoid_1.Output	0		Decimal		BOOL
Station1Module3.Solenoid_1.Output_Mirror	0		Decimal		BOOL
Station1Module3.Solenoid_1.OpenCircuit_Enabled	0		Decimal		BOOL
Station1Module3.Solenoid_1.Fault_CoilOpen_Missing	0		Decimal		BOOL
Station1Module3.Solenoid_1.Fault_CoilShorted	0		Decimal		BOOL

10.5.3 Example Controlling Solenoids via DINT tag names.

Solenoid outputs are easy to control using the Config_Tag_Name + Solenoid_Output_ALL.xx naming convention. Here is an example of controlling Solenoid/Output 1 and 2, 11 and 12.

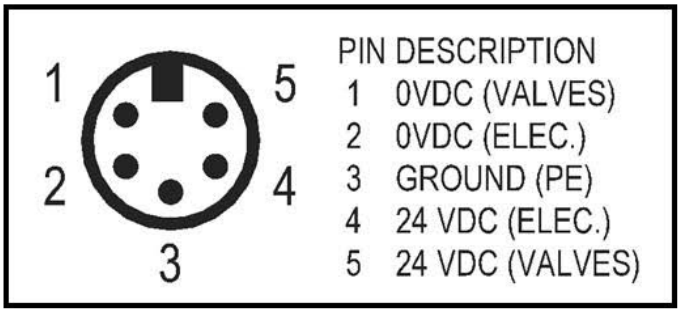


10.5.4 Other Solenoid DINTs

Other DINT registers are accessible about the solenoids, such as output control, output mirror status, whether the open circuit detection is enabled or not, and two kinds of faults. Coil open / missing or Coil shorted.

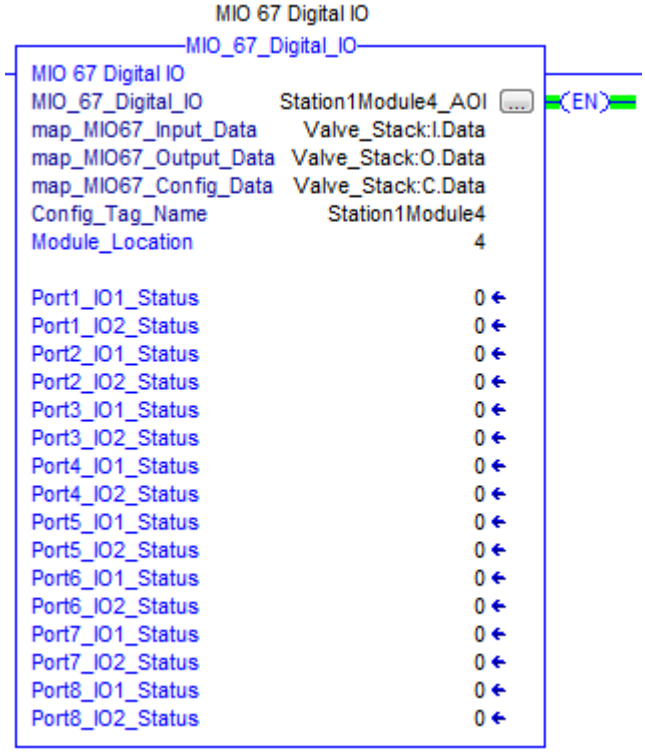
Station1Module3.Solenoid_32	{...}	{...}		udt_MIO_67_Solenoid_Types
Station1Module3.Solenoid_Output_All	0		Decimal	DINT
Station1Module3.Solenoid_Output_Mirror_All	0		Decimal	DINT
Station1Module3.OpenCircuit_Enabled_All	0		Decimal	DINT
Station1Module3.Fault_CoilOpen_Missing_All	0		Decimal	DINT
Station1Module3.Fault_CoilShorted_All	0		Decimal	DINT

10.5.5 Example Power Plus Pinout

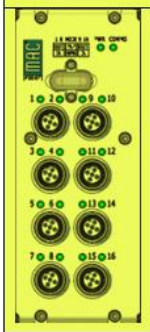
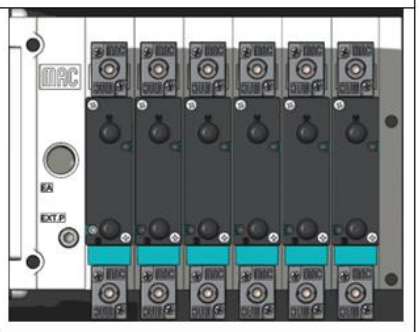


10.6 MIO_67_Digital_IO v1.0

The purpose of this AOI is to get inputs or output data from the add-on module. Digital I/O modules can be configured for INPUT or OUTPUT. Depending on dipswitch selection.

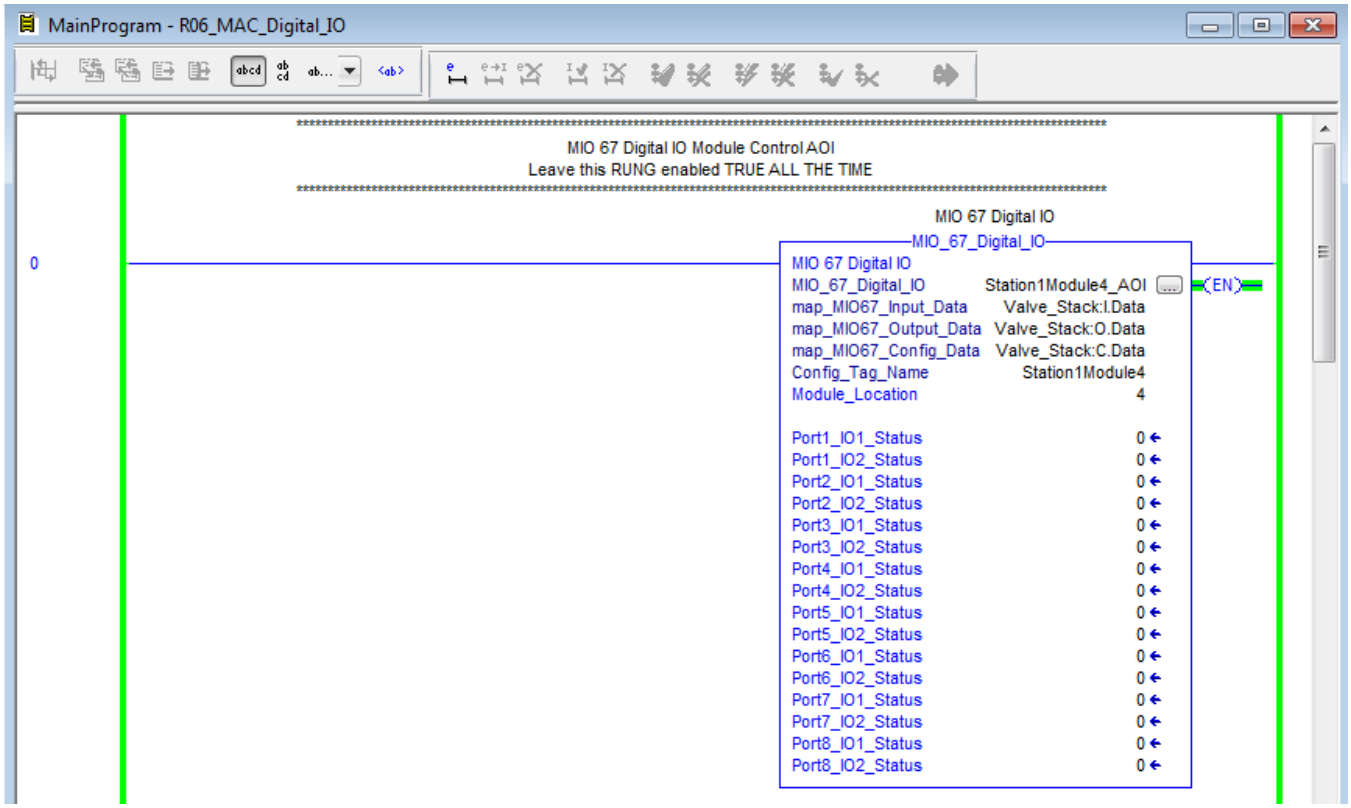
AOI Sample Tag Names filled out **(VERSION 1.0 AOI shown)**	Parameter Type	Description Examples
	MIO_67_Digital_IO AOI Tag Name	User Given
	map_MIO67_Input_data EIP:I.DATA	User Given
	map_MIO67_Output_data EIP:O.DATA	User Given
	map_MIO67_Config_data EIP:C.DATA	User Given
	Config_Tag_Name Config Tag Name	User Given
	Module_Location DINT	Module Location Number 1-12 Default: Declare the module location, see picture below as example
	Port1_IO1_Status Port1_IO2_Status … Port8_IO1_Status Port8_IO2_Status BOOL	Port IO Status Indicator 0 = I/O “Off” 1 = I/O “On”

Example: Digital IO is located in Module #4 location from EIP Adaptor.

Add-On Module #4 Digital I/O Dip Switches 0 – 0 – 0 In/NPN : In/NPN Selection “1”	Add-On Module #3 Power Plus Selection “11”	Add-On Module #2 Analog CURRENT Selection “9”	Add-On Module #1 Analog VOLTAGE Selection “10”	Communication Module	Valves
					

Ladder Logic Example:

Leave the AOI enabled true all the time.



10.6.1 Example Tag Port_x Structure

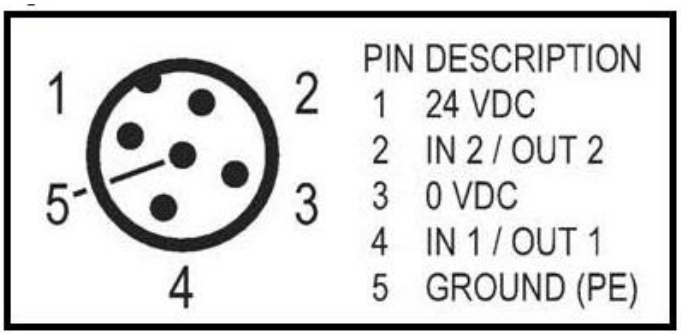
Port access can be written or read from Config_Tag_Name + Portx naming convention.

Station1Module1	{...}	{...}		udt_MIO_67_Module_Config	MAC MIO 67 Add-On Module
Station1Module2	{...}	{...}		udt_MIO_67_Module_Config	MAC MIO 67 Add-On Module
Station1Module4	{...}	{...}		udt_MIO_67_Module_Config	MAC MIO 67 Add-On Module
Station1Module4.Port_1	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 1
Station1Module4.Port_2	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 2
Station1Module4.Port_3	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 3
Station1Module4.Port_4	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 4
Station1Module4.Port_5	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 5
Station1Module4.Port_6	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 6
Station1Module4.Port_7	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 7
Station1Module4.Port_8	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 8

Station1Module4	{...}	{...}		udt_MIO_67_Module_Config	MAC MIO 67 Add-On Module
Station1Module4.Port_1	{...}	{...}		udt_MIO_67_Module_Types	MAC MIO 67 Add-On Module Port 1
Station1Module4.Port_1.IN1	0		Decimal	BOOL	MAC MIO 67 Add-On Module Input #1 I/O Condition
Station1Module4.Port_1.IN2	0		Decimal	BOOL	MAC MIO 67 Add-On Module Input #2 I/O Condition
Station1Module4.Port_1.IN_Raw	0		Decimal	INT	MAC MIO 67 Add-On Module Input Analog Raw Value
Station1Module4.Port_1.IN_Voltage	0.0		Float	REAL	MAC MIO 67 Add-On Module Input Analog Voltage Value
Station1Module4.Port_1.IN_Current	0.0		Float	REAL	MAC MIO 67 Add-On Module Input Analog Current Value
Station1Module4.Port_1.IN_Scaled_Min	0.0		Float	REAL	MAC MIO 67 Add-On Module Intput Desired Scaled Min Value
Station1Module4.Port_1.IN_Scaled_Max	0.0		Float	REAL	MAC MIO 67 Add-On Module Intput Desired Scaled Max Value
Station1Module4.Port_1.IN_Scaled_Output	0.0		Float	REAL	MAC MIO 67 Add-On Module Input User Scaled Output
Station1Module4.Port_1.OUT1	0		Decimal	BOOL	MAC MIO 67 Add-On Module Output #1 I/O Control
Station1Module4.Port_1.OUT2	0		Decimal	BOOL	MAC MIO 67 Add-On Module Output #2 I/O Control
Station1Module4.Port_1.OUT_Raw	0		Decimal	INT	MAC MIO 67 Add-On Module Output Analog Raw Value
Station1Module4.Port_1.OUT_Voltage	0.0		Float	REAL	MAC MIO 67 Add-On Module Output Analog Voltage Value
Station1Module4.Port_1.OUT_Current	0.0		Float	REAL	MAC MIO 67 Add-On Module Output Analog Current Value
Station1Module4.Port_1.OUT_Scaled_Min	0.0		Float	REAL	MAC MIO 67 Add-On Module Output Desired Scaled Min Value
Station1Module4.Port_1.OUT_Scaled_Max	0.0		Float	REAL	MAC MIO 67 Add-On Module Output Desired Scaled Max Value
Station1Module4.Port_1.OUT_Scaled_Output	0.0		Float	REAL	MAC MIO 67 Add-On Module Output User Scaled Output

Port_x.IN1	Input 1 Condition for Digital I/O
Port_x.IN2	Input 1 Condition for Digital I/O
Port_x.IN_Raw	**NOT USED for Digital I/O**
Port_x.IN_Voltage	**NOT USED for Digital I/O**
Port_x.IN_Current	**NOT USED for Digital I/O**
Port_x.IN_Scaled_Min	**NOT USED for Digital I/O**
Port_x.IN_Scaled_Max	**NOT USED for Digital I/O**
Port_x.IN_Scaled_Output	**NOT USED for Digital I/O**
Port_x.OUT1	Output 1 Control for Digital I/O
Port_x.OUT2	Output 2 Control for Digital I/O
Port_x.OUT _Raw	**NOT USED for Digital I/O**
Port_x.OUT _Voltage	**NOT USED for Digital I/O**
Port_x.OUT _Current	**NOT USED for Digital I/O**
Port_x.OUT _Scaled_Min	**NOT USED for Digital I/O**
Port_x.OUT _Scaled_Max	**NOT USED for Digital I/O**
Port_x.OUT_Scaled_Output	**NOT USED for Digital I/O**

10.6.2 Example Digital IO Pinout



10.6.3 Dipswitch Configuration

The module is broken down into two banks of 8 points. The left 4 connectors are considered Bank A and the right 4 connectors are considered Bank B. The dipswitches shown in **Figure 5** will set the bank function of being either input or output connectors for these modules. The pin outs can be found in **Figure 6**.

Each module has sixteen channels on the eight different connectors. The module can be configured for sixteen inputs, sixteen outputs, or eight inputs and eight outputs. For the inputs, along with setting the banks, you can also set whether they are for npn or pnp sensors. The table below shows the dipswitch settings.

Dipswitch			**Bank A**	**Bank B**
A	**B**	**C**		
0	0	0	Input/NPN	Input/NPN
0	0	1	Input/NPN	Output
0	1	0	Input/PNP	Input/PNP
0	1	1	Input/PNP	Output
1	0	0	Output	Input/NPN
1	0	1	Output	Output
1	1	0	Output	Input/PNP
1	1	1	Output	Output