Mitsubishi Motion AOI Solutions

MR-J4-TM AOI User Manual 3.002

Version 3.000 or later

R-216-SSGTSG-005-*

This document applies to the Mitsubishi Electric Corporation product components and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

MITSUBISHI ELECTRIC CORPORATION PROVIDES THIS DOCUMENT "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

This document could contain technical inaccuracies or typographical errors. Changes are made periodically to the information herein. Mitsubishi Electric Corporation may make improvements and changes at any time to the product(s) and/or program(s) described in this document.

Table of Content

1. AOIS VERSION 3.000 OR NEWER 5

2. RSLOGIX VERSION 5

3. MANAGED SWITCHES 6

4. BEST PRACTICES 6

5. DRIVE COMMUNICATION AOIS 7

MCOM_MITSUBISHI – Mitsubishi Drive Communication 7

6. CONFIGURATION AOIS 8

MMCFG – Mitsubishi Motion Axis Configuration 8

7. STATE AOIS 12

MMSO - Mitsubishi Motion Axis Servo On 12
MMSF - Mitsubishi Motion Axis Servo Off 12
MMASD - Mitsubishi Motion Axis Shutdown 13
MMASR - Mitsubishi Motion Axis Shutdown Reset 13
MMAFR - Mitsubishi Motion Axis Fault Reset 13

8. MOTION AOIS 14

MMAS - Mitsubishi Motion Axis Stop 14
MMAJ - Mitsubishi Motion Axis Jog 14
MMAH – Mitsubishi Motion Axis Home 16
MMRP – Mitsubishi Motion Redefine Position 17
MMAM – Mitsubishi Motion Axis Move 18
MMAT - Mitsubishi Motion Axis Torque 21
MMRP – Mitsubishi Motion Redefine Position 21

9. EVENT AOIS 22

MMAW - Mitsubishi Motion Arm Watch & MMDW - Mitsubishi Motion Disarm Watch 22
MMAR - Mitsubishi Motion Arm Registration & MMDR - Mitsubishi Motion Disarm Registration 24

10. ADDITIONAL AOIS 27

MMReadP - Mitsubishi Motion Read Parameter or CANopen Object 27
MMWriteR - Mitsubishi Motion Write RAM Parameter 27
MMWriteE - Mitsubishi Motion Write EEPROM Parameters 28
MMRE - Mitsubishi Motion Read Error 29

11. GEARING AOI 30

MMAG - Mitsubishi Motion Axis Gearing 30

12. SETTING IP ADDRESS 31

13. ROCKWELL PLC CONFIGURATION 33

14. IMPORTING AOIS 36

15. APPENDICES 39

Description of User Defined Data Structure: 39
MITSUBISHI_SERVO 39
Error Codes 41

Cross Reference of CANOpen Object vs J4 Parameters 43
LED & Servo States 44
Servo Digital I/O interface Connections 45
Master/Slave Interface Connections for Gearing 46
Write Loop 1 47
Write Loop 2 48
Initialize 50
Explicit MSG 51

16. MAJOR REVISIONS 53

17. REFERENCES 53

1. AOIs version 3.000 or Newer

Previous AOI version 2.000 only supported units in Pulses. New AOIs version 3.000 or later support different engineering unit as shown in the table below.

User unit setting	0: mm	1: inch	2: deg	3:PLS
Position command	mm [-9999.99 to 9999.99]	Inch [-999.999 to 999.999]	Deg [-360.000 to 360.000]	Pls
Speed command	mm/sec	inch/sec	Deg/sec	Pls
Acceleration &Deceleration	mm/sec^2	inch/sec^2	Deg/sec^2	Pls/sec^2

This document will describe the settings of user units as well as detailed explanation of the all AOIs.

To utilize the new version of AOIs, the user must follow the important points below:

1) The user must use the EDS file when creating a project and install the EDS file before importing the AOI.
2) The user must use the new AOI [MCOM_MITSUBISHI] for each servo drive (unique instances) to establish communication between controller and servo drive and needs to execute it every PLC scan.
3) The user doesn't need to use the COP (read from <servo>:I and write to <servo>:O) instruction in the new project. This instruction is included in the MCOM_MITSUBISHI AOI.
4) The user must execute the AOI [MMCFG] at least once to load the servo parameters after changing the gear ratio or user units.
5) The user must set the servo parameter PT01 =x2xx for user unit setting in degree and other unit setting set the servo parameter PT01=x3xx by using MR configurator or using AOIs..
6) A project created with version 2.000 may not mix AOIs. User must modify the project with new AOIs including data types as well as the position, speed, acceleration and deceleration value in user unit.

2. RSLogix Version

These AOIs need version 20.03 or newer.

The AOIs version 3.000 or newer require the EDS files.

The AOI [MCOM_MITSUBISHI] for each servo drive (unique instances) to establish communication between controller and servo drive and needs to execute it every PLC scan.

These AOIs is used for individual axis and do not use Motion Groups.

3. Managed Switches

From Rockwell Literature "Ethernet Design Considerations" Publication ENET-RM002C-EN-P - May 2013

As a general rule for unmanaged switches, make sure of the following:

- Your application does not contain I/O traffic or
- Your application has I/O control and the following is true:
  - The network is not directly connected to the IT network
  - All nodes on the network are Rockwell Automation devices
  - There is no potential to overload a device with traffic

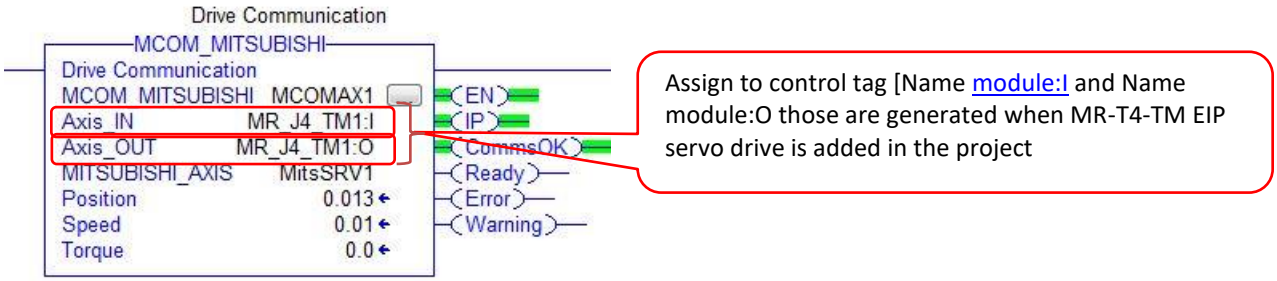
4. Best Practices

- Set an RPI between 1.0 ms and 50.0 ms. These AOIs are not embedded instructions. They rely on the RPI to maintain a viable servo update rate. Maximum RPI can be set 100 ms. Anything longer and the AOIs may not work properly.
- Each AOI must have a unique instance. E.g. if you have multiple MMAM AOIs, each one must have its own instance, do not share instance names.
- All AOIs use the same Mitsubishi_Servo UDT for that axis. E.g. if you have 2 axes, you should only have 2 Servo UDTs, one for each axis. (An Allen Bradley UDT = Mitsubishi SDT). A single AOI uses only one servo UDT.
- If communication is lost and regained, some AOIs need to be rebooted (e.g. MMAG re-programs registers for the gearing function during first scan). Either reboot the PLC or go from Run mode to Program mode and back or Initialize the AOIs.
- NOTE: Servo firmware versions before and including B1 do not recognize Merge in the MMAM.
- Do not set a loop continuously reading parameters using MMReadP. If you want to monitor servo settings such as parameters, inputs and outputs - use Explicit messaging (see Appendix). Explicit reading of any servo error thru MMRE can be triggered by Misubishi_Servo.Control bit3 (servo error) or bit7 (servo warning).
- Do not execute MMCFG , MMRE, MMWriteR, MMreaP and MMWRiteE at the same time.

5. Drive Communication AOIs

MCOM_MITSUBISHI – Mitsubishi Drive Communication

A MCOM_MITSUBISHI is used to communicate with servo drive. The purpose of this AOI is to issue commands from the controller and receive data from servo drive. This block must execute every scan of the controller cycle.



Variable Name (Type)	Description
Axis_IN (Module-Defined Data: IN: MR-J4-TM)	The data type is generated when MR-J4-TM is added in the project
Axis_Out (Module-Defined Data: Out: MR-J4-TM)	The data type is generated when MR-J4-TM is added in the project
MITSUBISHI_AXIS (User Defined UDT: MITSUBISHI_SERVO)	Create a user data type [MITSUBISHI_SERVO] that is used for identifying in the user program
Position (REAL)	Display actual servo position in user unit
Speed (REAL)	Display actual speed in user unit/Sec
Torque(REAL)	Display actual servo torque in % of rated torque
CommsOK (BOOL)	True indicates the communication between controller and drive is OK
Ready(BOOL)	True indicates the servo amplifier ready states[Servo enabled]
Error(BOOL)	True indicates the servo experience an Error
Warning(BOOL)	True indicates the servo experience a warning

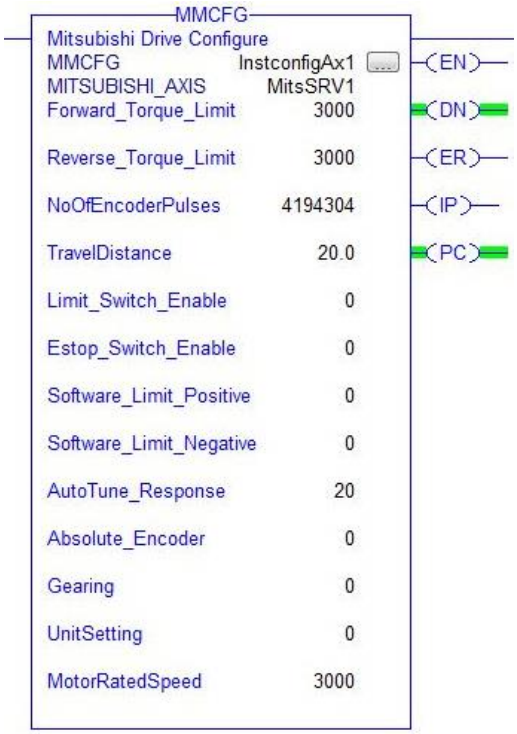
6. Configuration AOIs

- **True/False or Rising Edge** means the AOI can start with either a true rung or a one-shot. Once started, the logic will continue whether the rung is true or false.
- **Rising Edge** means it will execute on the rising edge and then lock out. There is no continuing logic when the rung is false
- **Latched True** means the AOI only executes when the rung is true.

MMCFG – Mitsubishi Motion Axis Configuration

- **MMCFG** is True/False or Rising Edge
- An embedded **MMWriteE** is automatically executed.
  Note: This can take up to 10 seconds - it stores all the parameters from RAM to EEPROM. Do not turn off power to recycle power for at least 10 seconds
- You must cycle power to the servo amplifier after executing **MMCFG** AOIs

Variable Name (Type)	Description
MITSUBISHI_AXIS (User Defined UDT: MITSUBISHI_SERVO)	Use the UDT that is used in MCOM_MITSUBISHI AOIs
Forward_Torque_Limit Reverse_Torque_Limit DINT	Setting -1 = maintains existing value Set the rate torque in % (x0.1) from 0 to 10000
NoOfEncoderPulses DINT	The number of motor encoder pulse that is equivalent to the travel distance setting (***Note 1***) in user unit -1 = maintains existing value
TravelDistance REAL	Travel distance setting value (in user unit mm/inch/Deg/pls) that is equivalent distance of the number of motor encoder pulse (***Note 1***) -1 = maintains existing value
Limit Switch Enable DINT	-1 = maintains existing value 0 = disabled, no limit switches (0C00) 1 = Positive limit switch active 2 = Negative limit switch Active 3 = Both limit switches active
Estop Switch Enable DINT	-1 = maintains existing value 0 = disabled, no Estop switch (2100) 1 = Estop switch active, without deceleration stop(0000) 2 = Estop switch active, with deceleration stop (2000) 3 = No Estop switch, without deceleration stop (0100)



Variable Name (Type)	Description
Software Limit position[DINT]	Maintains existing values if Positive value and Negative value are equal
Software Limit Negative[DINT]	Maintains existing values if Positive value and Negative value are equal
AutoTune Response	-1 = maintains existing value and Setting Range is 1 to 40
Absolute Encoder	-1 = maintains existing value 0 = no battery backup (incremental system, position will be lost after power down) 1 = battery backup(absolute system)
Gearing	(***slave servo only***) – -1 = maintains existing value 0= no gearing 1= Gearing active
UnitSetting(DINT)	Setting value : 0: mm; 1: inch; 2: degree; 3:PLS; 4 : mm [linear motor
MotorRatedSpeed	Set motor rated speed in r/min or mm/sec to calculate acceleration and deceleration in the engineering unit. set the value according to motor specification

Note 1: User unit setting in degree type, set NoOfEncoderPulses to motor side shaft revolutions and TravelDistance to number of load side shaft revolutions. That setting will result of 360 degree movement of load side shaft. Make sure both values are integers number and below 16384. Also set servo parameter PT01 =x2xx.

The following section shows examples of electronic gear ratio setting.

Example1: Ball screw user unit setting in mm

Machine specifications:
Ball screw lead = 20 mm
Encoder resolution = 4194304 pulse/rev
Servomotor: HG-KR series.
Therefore, set

NoOfEncoderPulses = 4194304
TravelDistance = 20.0
Unitsetting= 0 and
MotorRatedSpeed=3000.

Must set the servo parameter PT01 =x**3**xx

Example2: Ball screw user unit setting in inch

Machine specifications:

Ball screw lead: = 0.5 [inch]

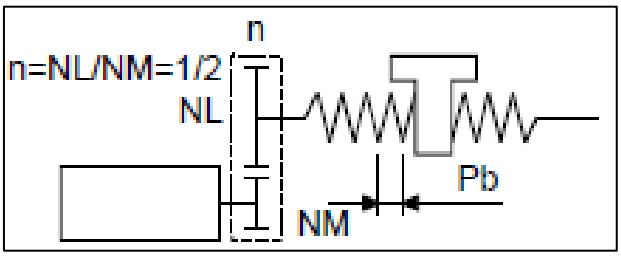
Reduction ratio: n = 1/2

Servo motor resolution = 4194304 [pulse/rev]

Servo motor: **HG-JR[1500r/min]** series

NoOfEncoderPulses = 4194304
TravelDistance = 0.25
Unitsetting= 1 and
MotorRatedSpeed=**1500**.

Must set the servo parameter PT01 =x**3**xx



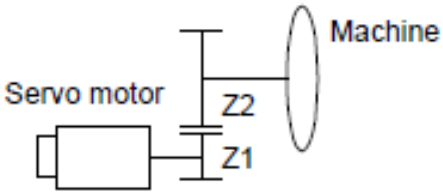
Example 3: A system user unit setting in Degree:

Machine specifications:
Reduction ratio: = 11/25

Encoder resolution = 4194304 pulse/rev
Servo motor: HG-SR[2000r/min] series

Therefore, setting example
NoOfEncoderPulses = 25
TravelDistance = 11,
Unitsetting=2 and
MotorRatedSpeed=2000.

Must set the servo parameter PT01 =x**2**xx



Pt (Servo motor resolution): 4194304 pulses/rev
Z1: Number of gear teeth on servo motor side
Z2: Number of gear teeth on machine side
Z1: Z2 = 11:25

One revolution of the load side shaft movement will represent 360.00 degrees.

Example3: Linear Motor in mm

Machine specifications:
Encoder Motor resolution = 1 µm
Therefore, set
NoOfEncoderPulses = 1000
TravelDistance = 1.0 [mm]
Unitsetting= 4 and
MotorRatedSpeed=2000 [mm/sec, set the value according to motor specification]

Must set the servo parameter
PT01 =x**3**xx
PL02 =1
PL03=1

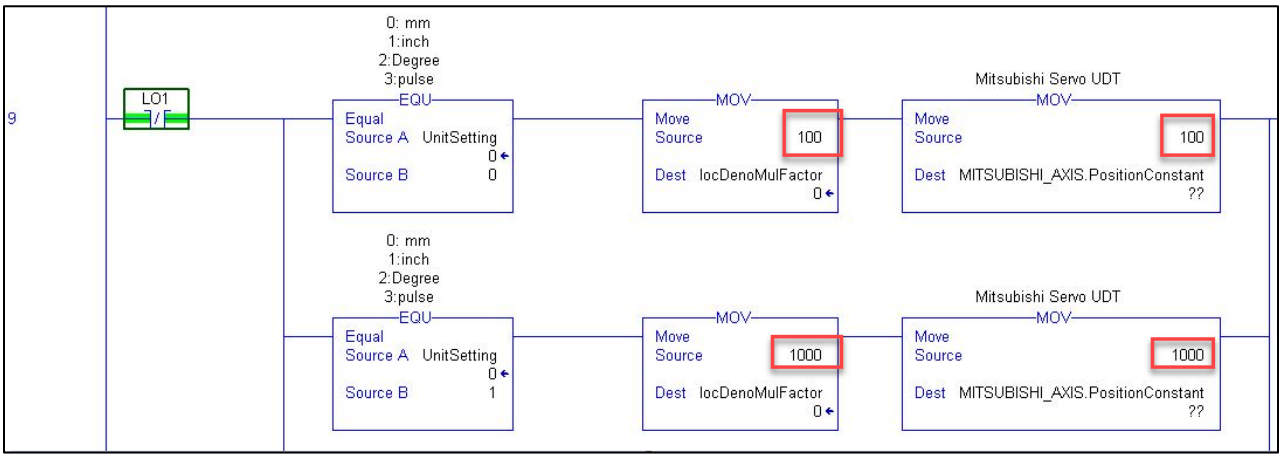
Set the number of digits after decimal point for command position:

By default, the command position (when in inches) is set the 3 digit after decimal point and for mm 2 digit after decimal point. That's means

For user unit mm: Position command setting range -9999.99 to 9999.99 mm.

Inch: Position command setting range -999.999 to 999.999 inches.

You can change this setting value by modifying the MMCFG at the rung 9. The following screen shows the default setting value.

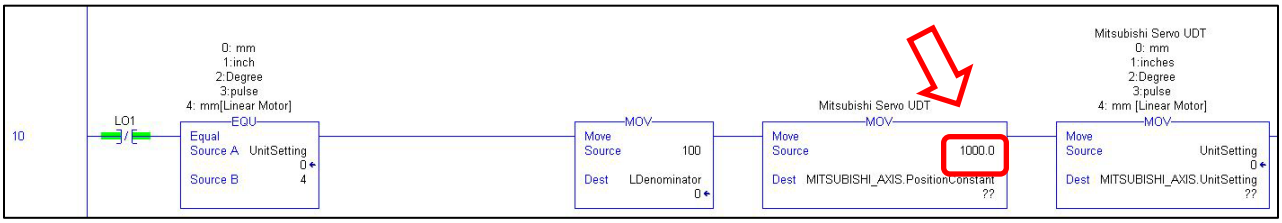


The following example shows other than the default setting:

For user unit mm: LocDenoMulFactor to 1000 and MITSUBISHI_AXIS.PositionConstan to 1000, in this case the position command range will be -999.999 to 999.999 mm. & " mm"& " mm"

For user unit inch: LocDenoMulFactor to 10000 and MITSUBISHI_AXIS.PositionConstan to 10000, in this case the position command range will be -99.9999 to 99.9999 inches.

In case of linear motor the position command range is -999.999 to 999.999 mm, if the application required the position command range more than that, please change the value MITSUBISHI_AXIS.PositionConstan to 100 at rung 10. That modification would allow the command position range -9999.99 to 9999.99 mm.



7. State AOIs

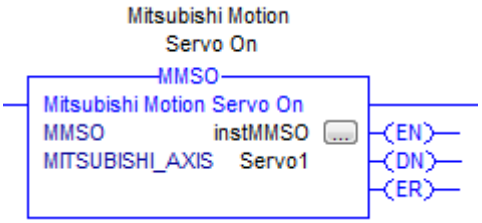
- **True/False or Rising Edge** means the AOI can start with either a true rung or a one-time. Once started, the logic will continue whether the rung is true or false.
- **Rising Edge** means it will execute on the rising edge and then lock out. There is no continuing logic when the rung is false. With a latched bit, the rung will have to go false, then true to re-trigger the AOI.
- **Latched True** means the AOI only executes when the rung is true.

MMSO - Mitsubishi Motion Axis Servo On

Use the Motion Servo On (MMSO) instruction to enable the servo amplifier

DN on MMSO and MMSF shows state of servo based on servo enabled state. If servo is enabled, MMSO.DN is on at all times.

- MMSO is **Rising Edge**. It enables the drive when the rung makes a false-to-true transition



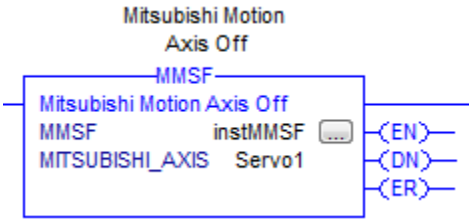
See Servo States

MMSF - Mitsubishi Motion Axis Servo Off

Use the Motion Servo OFF (MMSF) instruction to deactivate the drive output for the specified axis.

DN on MMSO and MMSF shows state of servo based on servo enabled. If servo is not enabled, MMSF.DN is on at all times.

- MMSO is **Rising Edge**. It disables the drive when the rung makes a false-to-true transition

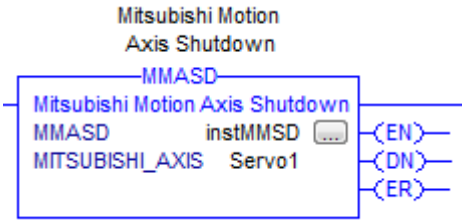


MMASD - Mitsubishi Motion Axis Shutdown

Use the Motion Axis Shutdown (MMASD) instruction to force a specified axis into the Shutdown state. The Shutdown state of an axis is the condition where the drive output is disabled. The axis remains in the Shutdown state until MMASR –Mitsubishi Motion Axis Shutdown Reset is executed.

- MMASD is **Rising Edge**. It disables the drive when the rung makes a false-to-true transition

See Servo States

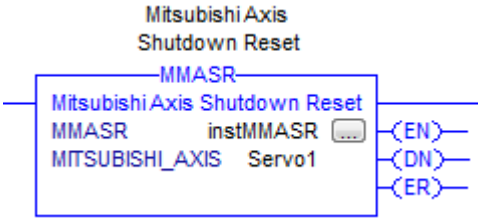


MMASR - Mitsubishi Motion Axis Shutdown Reset

Use the Motion Axis Shutdown (MMASR) instruction to transition an axis from an existing Shutdown state to an Axis Ready state. All faults associated with the specified axis are automatically cleared.

- MMASR is **Rising Edge.** It clears the motion axis shutdown state when the rung makes a false-to-true transition

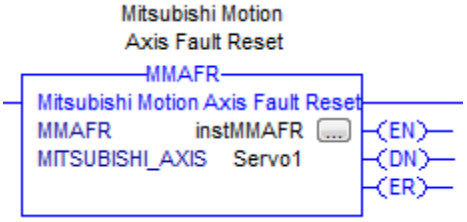
See Servo States



MMAFR - Mitsubishi Motion Axis Fault Reset

Use the Motion Axis Fault Reset (MMAFR) instruction to clear all motion Errors and waring for an axis. This is the only method for clearing axis motion errors.

- MMAFR is **Rising Edge.** It reset the error when the rung makes a false-to-true transition



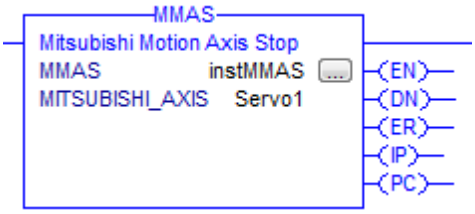
8. Motion AOIs

- **True/False or Rising Edge** means the AOI can start with either a latched true rung or a one-shot. Once started, the logic will continue whether the rung is true or false.
- **Latched True** means the AOI only executes when the rung is true.

MMAS - Mitsubishi Motion Axis Stop

Use the Motion Axis Stop (MAMS) instruction to stop a specific motion process on an axis or to stop the axis completely.

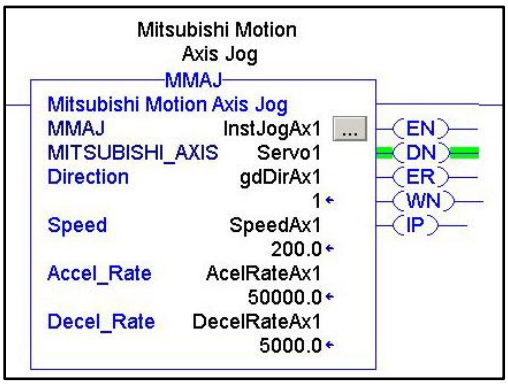
- **MMAS** (Motion Axis Stop) is **Latched True**. Once it is set, the axis cannot move.
- If you use it with a one-shot, it will stop motion but any motion command can start the servo again.
- When latched and active, the servo does not move and ignores all motion commands,
- If you try to move with **MMAS** active - Error = 1017.
- **MMAS** stops motion and unconditionally clears the target buffer.
    - If you want to be able to resume after stopping an MMAM move, use absolute moves.
- If you trigger a stop when servo is not enabled, you do not get an alarm.



MMAJ - Mitsubishi Motion Axis Jog

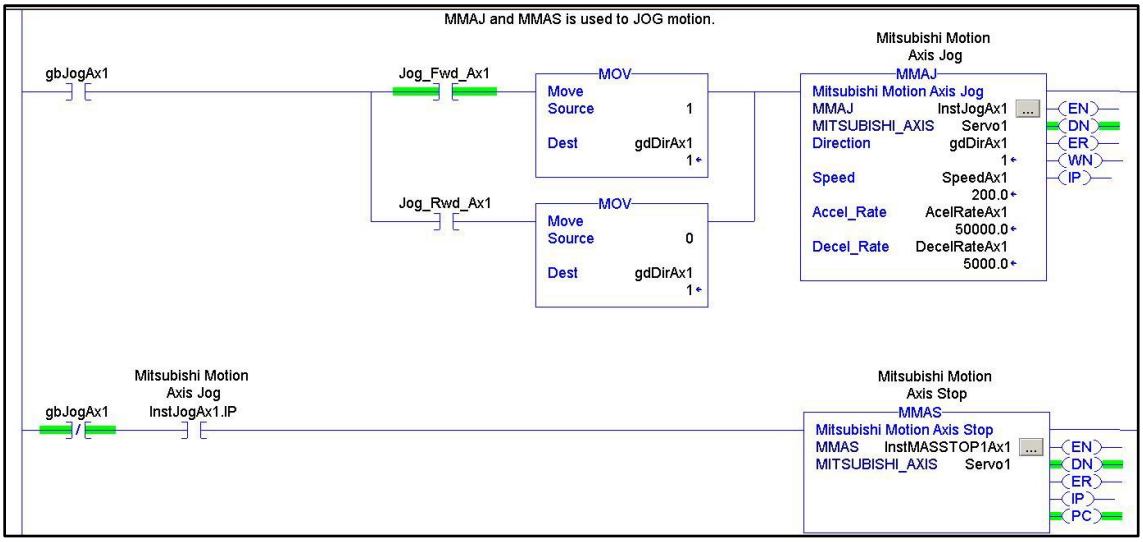
Use the Motion Axis Jog (MMAJ) instruction to move an axis at a constant speed until execute MMAS AOI.

- MMAJ (Motion Axis Jog) is **True/False or Rising Edge**.
- Direction is 0 or 1.
- Speed is always positive value (cannot be set a negative value)
- Multiple Moves, during a jog, a second jog command will merge immediately and will not change the outputs (DN, IP) of the first AOI instant.
- To stop, use **MMAS** Mitsubishi Motion Axis Stop.



Variable Name (Type)	Description
Direction (DINT)	0 : Forward Direction 1: Reverse Direction
Speed(REAL)	Speed to move the axis in user unit/sec.[Set the speed value between 0 to MITSUBISHI _SERVO.MaxVelocityLimit]
Accel_Rate (REAL)	Acceleration rate of the axis in user unit/Sec^2 [Set the acceleration rate between MITSUBISHI_SERVO.MinAccelerationRate and MITSUBISHI_ SERVO. MaxAccelerationRate]
Decel_Rate (REAL)	Deceleration rate of the axis in user unit/Sec^2 [Set the deceleration rate between MITSUBISHI_SERVO.MinAccelerationRate and MITSUBISHI_SERVO. MaxAccelerationRate]

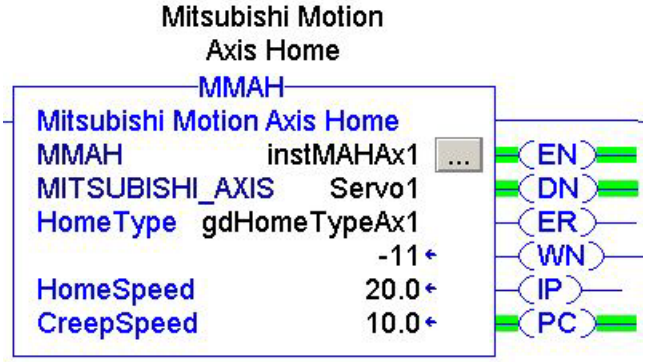
Note: once started, Jog will run continuously regardless of the rung state. If you want to control jog through a pushbutton, program the MMAJ to execute with a one-shot or as long as the button is pressed. Then to stop, program a rung with an MMAS triggered by when you let go of the button.



MMAH – Mitsubishi Motion Axis Home

Use the Motion Axis Home (MMAH) instruction to home an axis.

- MMAH - Homing is **True/False or Rising Edge**
- Direction is based on Home type
- Command and actual position become zero



It has the **MMWriteR** AOI embedded inside for Parameter Mapping – HomeType, HomeSpeed and CreepSpeed.

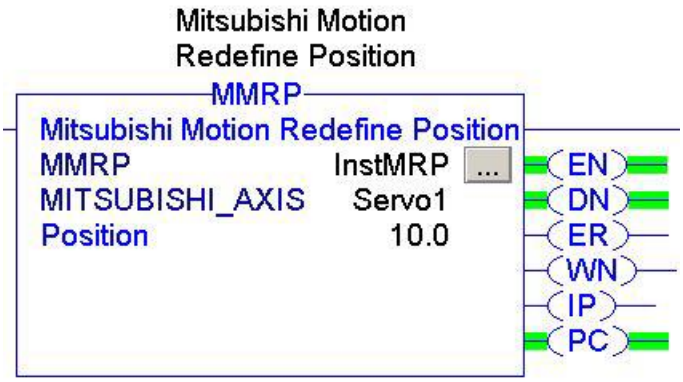
Variable Name (Type)	Description
HomeType [DINT]	See details in user manual
HomeSpeed[REAL]	Speed to move the axis in user unit/sec.[Set speed value between 0 to MITSUBISHI _SERVO.MaxVelocityLimit]
CreepSpeed[REAL]	Speed to move the axis in user unit/sec.[Set speed value between 0 to MITSUBISHI _SERVO.MaxVelocityLimit]

For detailed description of all homing commands, MR-J4-TM SERVO AMPLIFIER INSTRUCTION MANUAL (EtherNet/IP) SH030226

MMRP – Mitsubishi Motion Redefine Position

Use the MMRP instruction to change the axis Position.

- MMRP - is **True/False or Rising Edge**
- Command and actual position become defined Position [absolute]
- The axis position cannot be redefined on-the-fly [motion is take place]



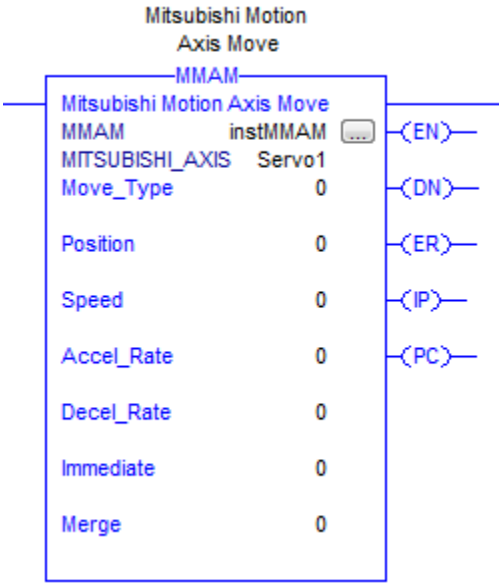
MMAM – Mitsubishi Motion Axis Move

Use the Motion Axis Move (MMAM) instruction to move an axis to a specified position.

- MMAM (Motion Axis Move) is **True/False or Rising Edge**.
- Multiple Moves using **Merge** and **Immediate**
- Position command range is based unit setting. The default setting is given below

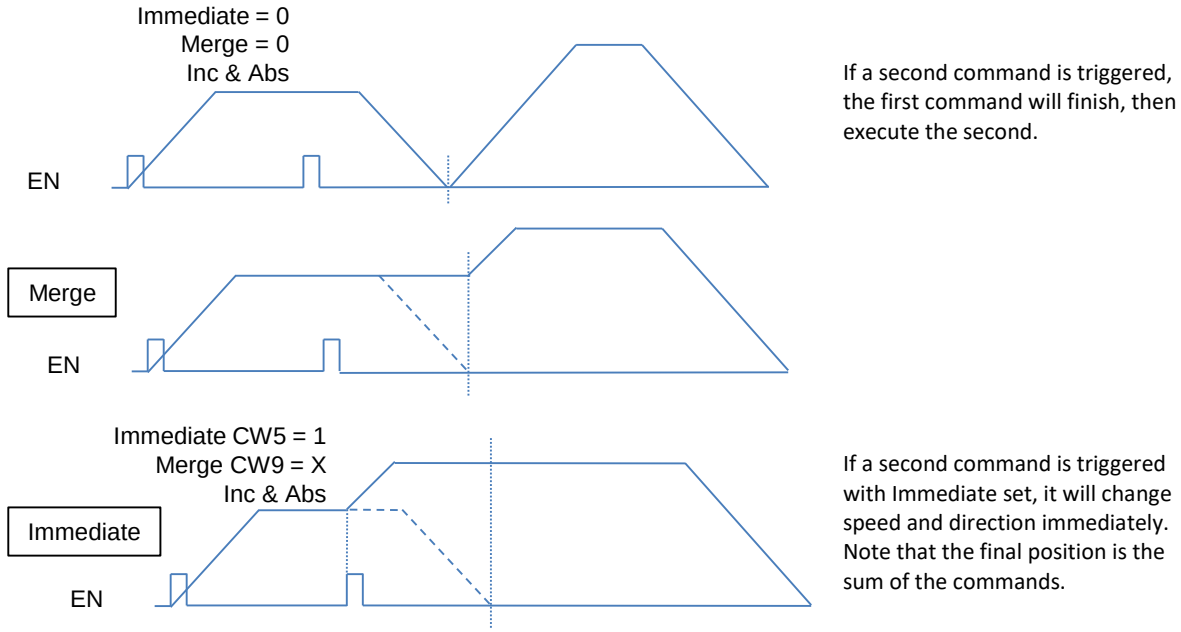
  Pulse :-999999 to 999999 pulses
  Mm: -9999.99 to 9999.99 mms
  Inch: -999.999 to 999.999 inches
  Degree: -360.000 to 360.000 degree

Note: you may change the setting range see details MMCFG AOIs

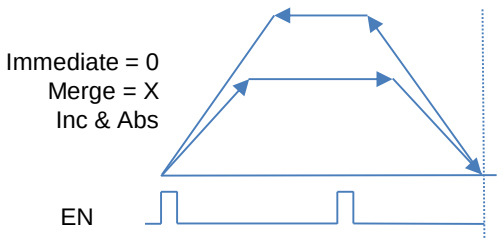


Variable Name (Type)	Description
MoveType(DINT)	0→ Absolute and 1→Incremental move [Incremental mode doesn't support in Degree mode]
Position[Real)	Absolute system→ Position to move and Incremental system → Distance to move (polarity determines direction)
Speed(REAL)	Speed to move the axis in user unit/sec.[Set speed value between 0 to MITSUBISHI _SERVO.MaxVelocityLimit]
Accel_Rate (REAL)	Acceleration rate of the axis in user unit/Sec2 [Set acceleration rate between MITSUBISHI_SERVO.MinAccelerationRate and MITSUBISHI_ SERVO. MaxAccelerationRate]
Decel_Rate (REAL)	Deceleration rate of the axis in user unit/Sec2 [Set deceleration rate between MITSUBISHI_SERVO.MinAccelerationRate and MITSUBISHI_SERVO. MaxAccelerationRate]
Immediate (BOOL)	See details in the figure below
Merge (BOOL)	See details in the figure below

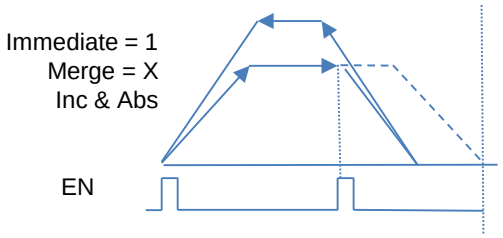
You can merge/blend a second move command using Immediate and Merge inputs. However, note the distance of the second command is always added to the distance of the first command.



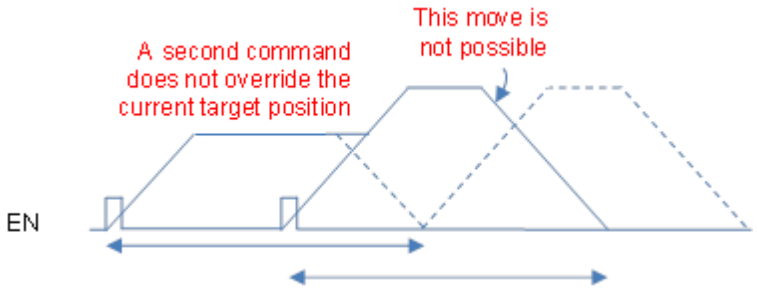
Second move is same distance but negative



If a second command is triggered before the first is complete, the first command will finish, then execute the second.



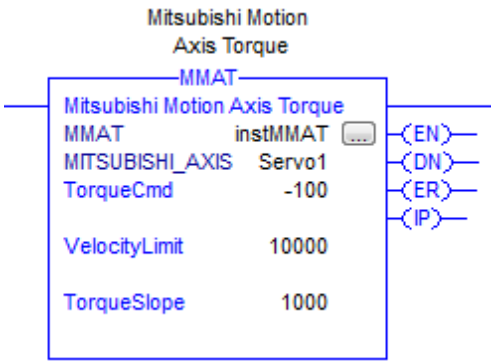
If a second command is triggered with Immediate set before the first is complete, it will change speed and direction immediately. Note that the final position is the sum of the commands.



MMAT - Mitsubishi Motion Axis Torque

Use the Motion Axis Torque (MMAT) instruction to move an axis at a constant torque until you tell it to stop.

- MMAT (Motion Axis Torque) is **True/False or Rising Edge**.
- **Velocitylimit** cannot be negative and is in unit/sec
- Direction is determined by TorqueCmd polarity.
- **TorqueCmd** is in units of % of Rated Torque (100 => 100% of Rated Torque)
- **TorqueSlope** is Rated Torque/10/sec
- To stop, use MMAS Mitsubishi Motion Axis Stop.
- Note that stopping torque move with MMAS will leave the servo with 0 torque and will not hold position.



MMRP – Mitsubishi Motion Redefine Position

Use the Motion redefine position to set change the command or actual position of an axis. The value specified by Position is used to update the Actual or Command position of Axis.

- MMRP - **True/False or Rising Edge**
- Position - The value to use to change the axis position to or offset to current

Note: 1) MMRP is not available in the degree mode axis.

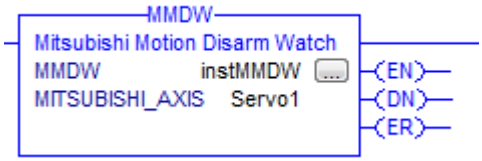
2) Offset value is store in the PLC. PLC reset and download the project may delete the redefine position. Please execute the MMRP after reset or download project.

9. Event AOIs

- **True/False or Rising Edge** means the AOI can start with either a true rung or a one-shot. Once started, the logic will continue whether the rung is true or false.
- **Rising Edge** means it will execute on the rising edge and then lock out. There is no continuing logic when the rung is false
- **Latched True** means the AOI only executes when the rung is true.

MMAW - Mitsubishi Motion Arm Watch & MMDW - Mitsubishi Motion Disarm Watch

- **MMAW** is True/False or Rising Edge
- **MMDW** is Rising Edge



- TriggerCondition –
  - 0 = Forward - the servo module looks for the actual position to change from less than the watch position to greater than the watch position.
  - 1 = Reverse – the servo module looks for the actual position to change from greater than the watch position to less than the watch position.
- **Position** that triggers PC output
- **PC** - will go high if servo moves in correct direction from outside to inside the position.

If **TriggerCondition** is 0 and actual position is greater than **Position**, **PC** is set. It is reset by another rising edge trigger to MMAW or MMDW.

If **TriggerCondition** is 1 and actual position is less than **Position**, **PC** is set. It is reset by another rising edge trigger to MMAW or MMDW.

In the MITSUBISHI_SERVO UDT:

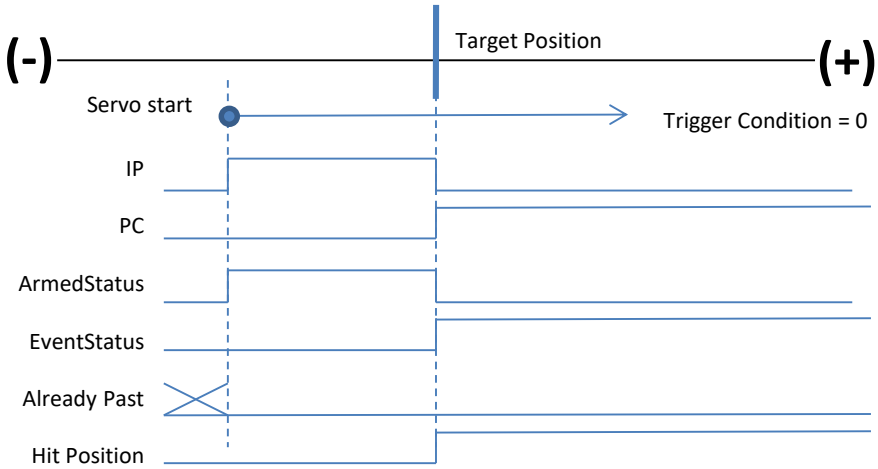
- **WatchEventArmedStatus** is a Boolean set if the Watch logic is active. Reset when event occurs or MMDW.
- **WatchEventStatus** is a Boolean flag set when event occurs. It is reset by another rising edge trigger to MMAW or MMDW.

1. If **TriggerCondition** is 0 and actual position reaches greater than **Position**, **PC** is set. It is reset by another rising edge trigger to MMAW or MMDW.
2. If **TriggerCondition** is 1 and actual position reaches less than **Position**, **PC** is set. It is reset by another rising edge trigger to MMAW or MMDW.

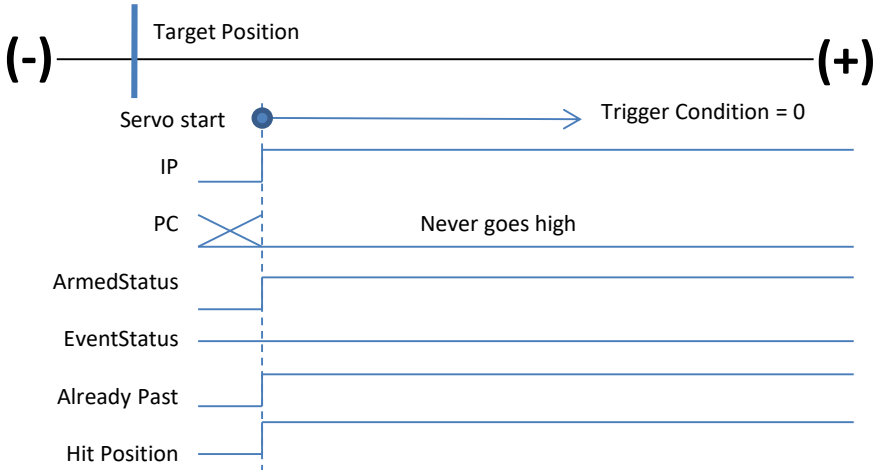
3. If the AOI is executed with servo ALREADY past the programmed postion (heading away from the programmed position), it will set the MMAW instance dot operator - <instMMAW>.**AlreadyPast** and **PC** will never be set.

To reset the watch you have to re-trigger MMAW. MMDW only turns off the watch and clears the outputs of the MMAW, it does not re-trigger anything.

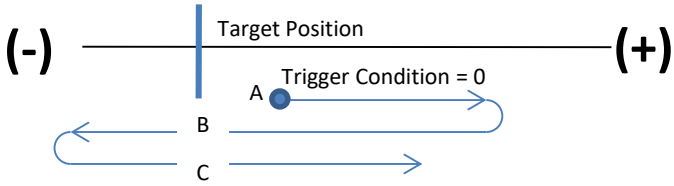
Example 1

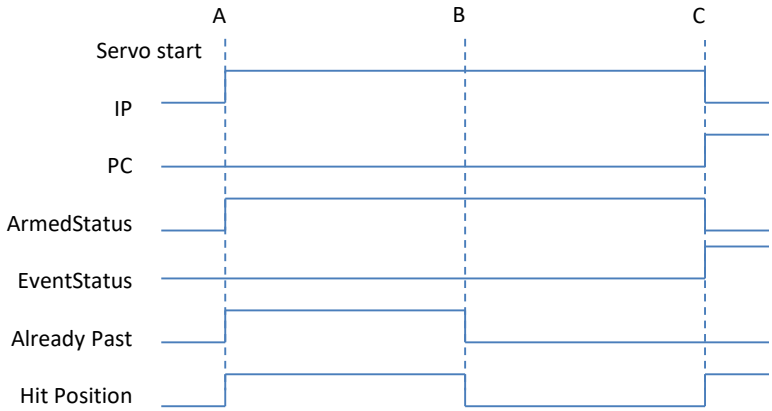


Example 2



Example 3





MMAR - Mitsubishi Motion Arm Registration & MMDR - Mitsubishi Motion Disarm Registration

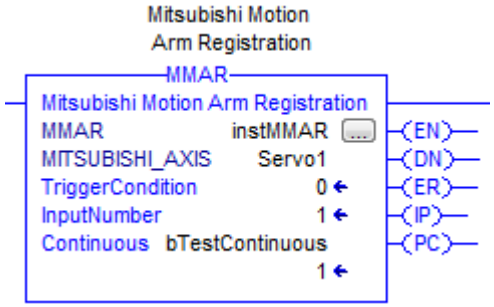
- MMAR is **True/False or Rising Edge**
- MMDR is **Rising Edge**
- If you use non continuous (= 0) you need to Retrigger the rung to reset the PC outputs.
- If you change any setting, you need to re-initialize the AOI. <MMAR.Initialize>

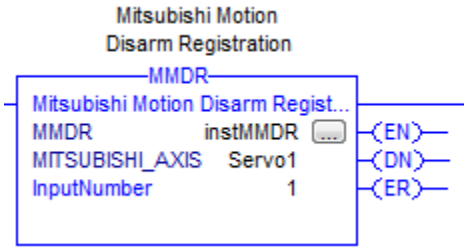
Cycle the PLC at least once or trigger <MMAR instance>.Initialize after adding MMAR to the program or changing any setting. MMAR reprograms an internal register for Registration functions.

When the Touch Probe input is asserted, the actual position at that moment is captured and stored in Registration registers.

The Touch Probe inputs are in CN3 pins 1 and 10.

The captured positions are in the AOI instance – x.**A_TouchProbeValue** and x.**B_TouchProbeValue** in user unit**.**





Input	Range	Description
Trigger Condition	= 0 trigger on positive edge = 1 trigger on negative edge	
InputNumber	0, 1 or 2	Which probe to use. 0 = disable that channel
Continuous	= 0 Once captured, AOI must be toggled to re-arm = 1 AOI will automatically re-arm after every probe input assertion.	If AOI is not **Continuous**, PC (Process Complete) will come on with the first probe input assertion. PC will stay on and AOI will not re-arm until AOI enable is toggled. If AOI is set **Continuous**, PC is a one-shot for each probe input assertion.

RegEventArmedStatus in the MITSUBISHI_SERVO UDT is set when the AOI is enabled in Logic. It is reset when…

- After a capture if not in Continuous mode.
- With MMDR
- With an error.

RegEventStatus in SDT goes low when the AOI is enabled. It goes high when a capture occurs. It is reset when…

- RegEventArmedStatus is toggled

Output	Continuous	Description
A_PC B_PC	= 0 Latched	**Latched** when position is captured. Stays set until the AOI is toggled or initialized
	= 1 Continuous update	**Pulsed** output for 1 scan only when position is captured.

There are two probe inputs 1 & 2. These are independent of the two Channels A & B. You assign the inputs to any channel.

Example 1

Channel A is Touch Probe input 2, falling edge and continuous.

Channel B is Touch Probe input 2, rising edge and continuous.

Example 2

Channel A is Touch Probe input 1, rising edge and continuous.

Channel B is Touch Probe input 2, rising edge and continuous.

Example 3

Channel A is Touch Probe disabled (= 0).

Channel B is Touch Probe input 2, rising edge and latched.

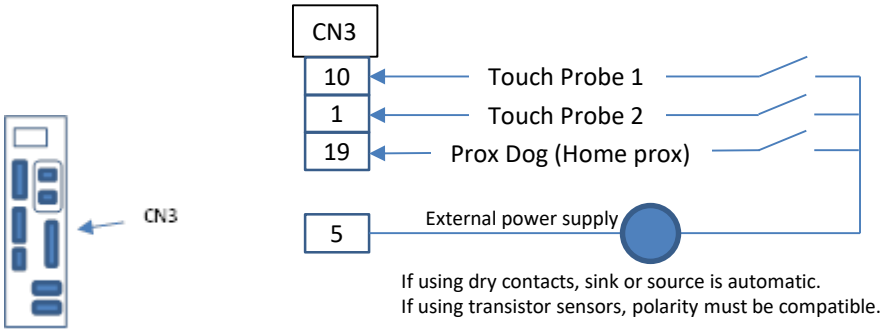
Example 4

Channel A is Touch Probe input 1, rising edge and ***latched***.

Channel B is Touch Probe input 1, falling edge and ***continuous*** <<< NOT ALLOWED.

You cannot have both latched and continuous on the same input.

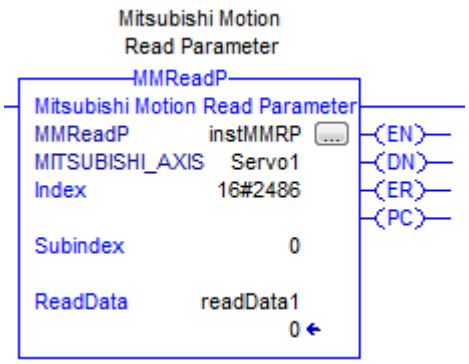
The **MMAR** AOI will not have window registration. The registration function in the servo works all the time with no limitations. Rather than embed windowing code inside the AOI, it is left to the user to write their own windowing logic if desired.



10. Additional AOIs

- **True/False or Rising Edge** means the AOI can start with either a true rung or a one-shot. Once started, the logic will continue whether the rung is true or false.
- **Rising Edge** means it will execute on the rising edge and then lock out. There is no continuing logic when the rung is false
- **Latched True** means the AOI only executes when the rung is true.

MMReadP - Mitsubishi Motion Read Parameter or CANopen Object



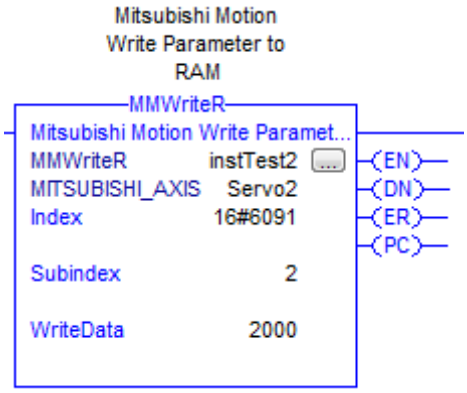
- MMReadP is **True/False or Rising Edge**
- This is a single read AOI that uses CANOpen objects.
- To convert J4 parameter addresses to CANOpen see Cross Reference of CANOpen Parameters

E.g. Using the mapping Cross Reference, Creep Speed PT06 converts to 16#2486h sub-index 0

MMWriteR - Mitsubishi Motion Write RAM Parameter

- MMWriteR is **True/False or Rising Edge**

This is an auxiliary AOI for writing parameters or CANOPN object with the J4 Servo. MMWriteR writes to the RAM memory. When you cycle power, these values are not retained.To permanently load into memory EEPROM, see MMWriteE.



This is a single write AOI that uses CANOpen addresses.This example writes to CANOpen register 6091h sub index 2 which is electronic gear denominator (scaling).

To convert J4 parameter addresses to CANOpen see Cross Reference of CANOpen Parameters.

Using the mapping Cross Reference PA07 converts to 16#2007h sub-index 0.

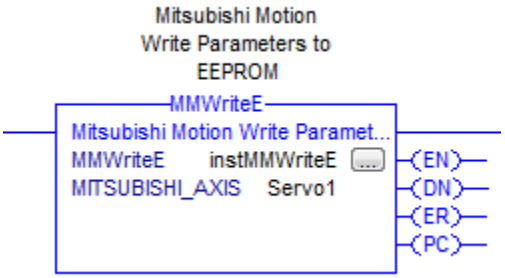
Note: You can program a loop to do multiple writes. See Write Loop examples in Tables & Lists

MMWriteE - Mitsubishi Motion Write EEPROM Parameters

- MMWriteE is **True/False or Rising Edge**
- **You do not enter parameters in this AOI. It will save what is already in RAM at the time of** execution.

MMWriteE writes all the Parameters settings to EEPROM.

MMWriteR only writes to RAM – when the servo reboots, the values in the EEPROM load to servo parameters



Note: This can take up to 10 seconds - it stores all the parameters from RAM to EEPROM. Do not turn off power to recycle power for at least 10 seconds

MMRE - Mitsubishi Motion Read Error

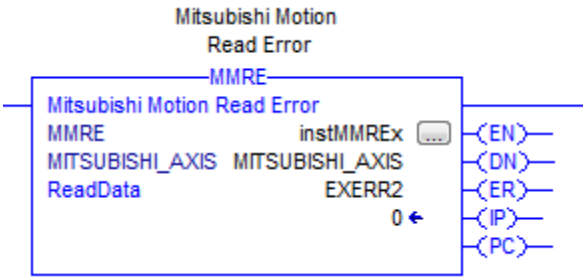
- **MMRE** is Rising Edge

MMRE (Motion Read Error) - Is a stripped down AOI designed to be embedded in other AOIs (MMAH, MMAM, MMAJ, MMAT, etc). This AOI reads the servo error using Parameter Mapping. It can be used as a standalone but it does not have LO state logic, only Parameter Mapping state logic. It does not have any error checking on its own.

Do not set up a loop reading this continuously. You can trigger a stand alone MMRE by monitoring Misubishi_Servo. Control bit3 (servo error) or bit7 (servo warning)

Most other AOIs that have MMRE embedded with a local instance tag of MMRE in their own parameter tags.

In the event of a Servo Error, MMRE will output the servo error into EXERR and the parent AOI will move Error Code 1020 into ERROR.



The value of the alarm is returned that description of the values is as follows.

Bit	Description
0 to 15	Alarm detail No.
16 to 31	Alarm No.

In example the value "00160003h" will be returned If [AL. 16.3] occurs.

Details description of the Alarm code refer to the MELSERVO-J4 Servo amplifier INSTRUCTION MANUAL (TROUBLE SHOOTING) SH(NA)030109

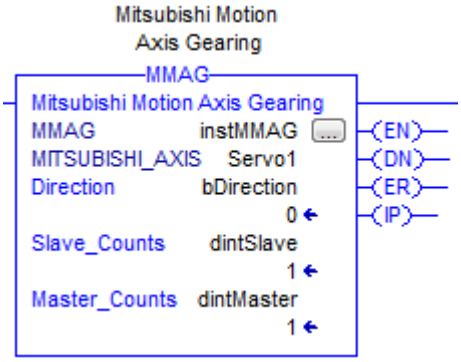
11. Gearing AOI

MMAG - Mitsubishi Motion Axis Gearing

The slave axis will follow the master axis. The master encoder counts are ported to the slave.

- *All MMAG commands are for the slave (geared) axis.* You do not program anything for the master axis which can be any servo axis or encoder with differential line driver A/B/Z output.
- **MMAG** is **Latched True**. When active, the slave will follow. When not active the slave drops out.
- **Direction** = 0 slave moves in same direction. **Direction** = 1 slave moves in opposite direction.
- Slave Counts/Master Counts

Cycle the PLC at least once after adding MMAG to the program. MMAG reprograms an internal register during first scan for Gearing functions



To enable Gearing function, you must set the 1 to Gearing input of MMCFG AOIs.

For cabling between Master and Slave, see Appendices.

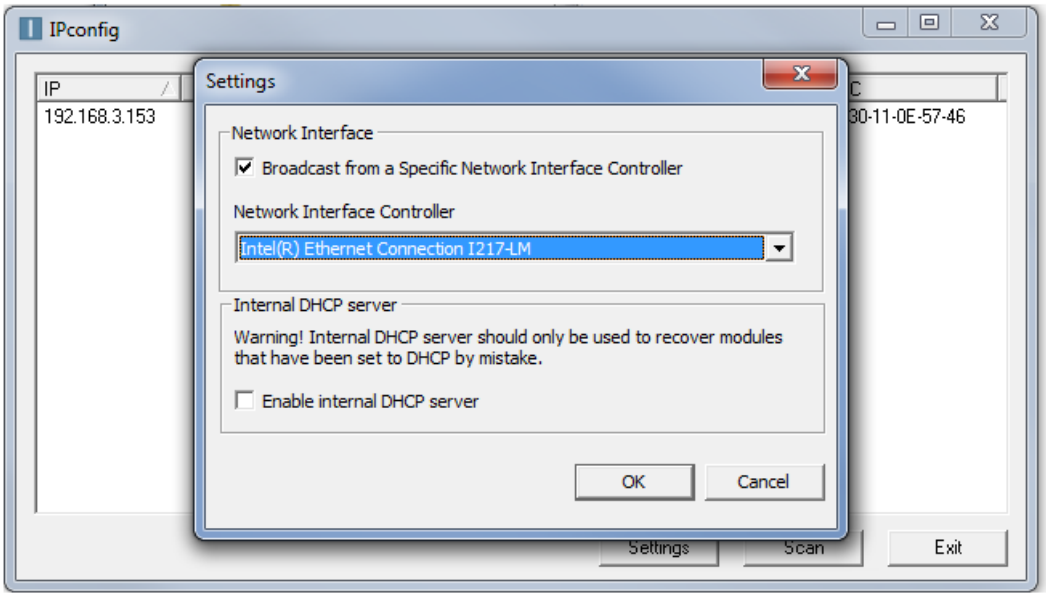
Remember, only the slave gets the parameters. If you try to enable gearing on the any axes you will get error 70.3 on the axes if the gearing cable is not connected.

The Servo instance will be in Gearmode when MMAG AOI is true (EN = 1). If you drop the PLC out of run mode while gearing (e.g. go to program mode), then back to run mode, the Servo itself will drop out of gearing but the AOI will still be in Gearmode. You will have to toggle the AOI to reset the servo functionality. Also, if you try to jog or move on the slave axis after toggling run mode, you will get an error that you are still in Gearmode until you turn off the MMAG rung.

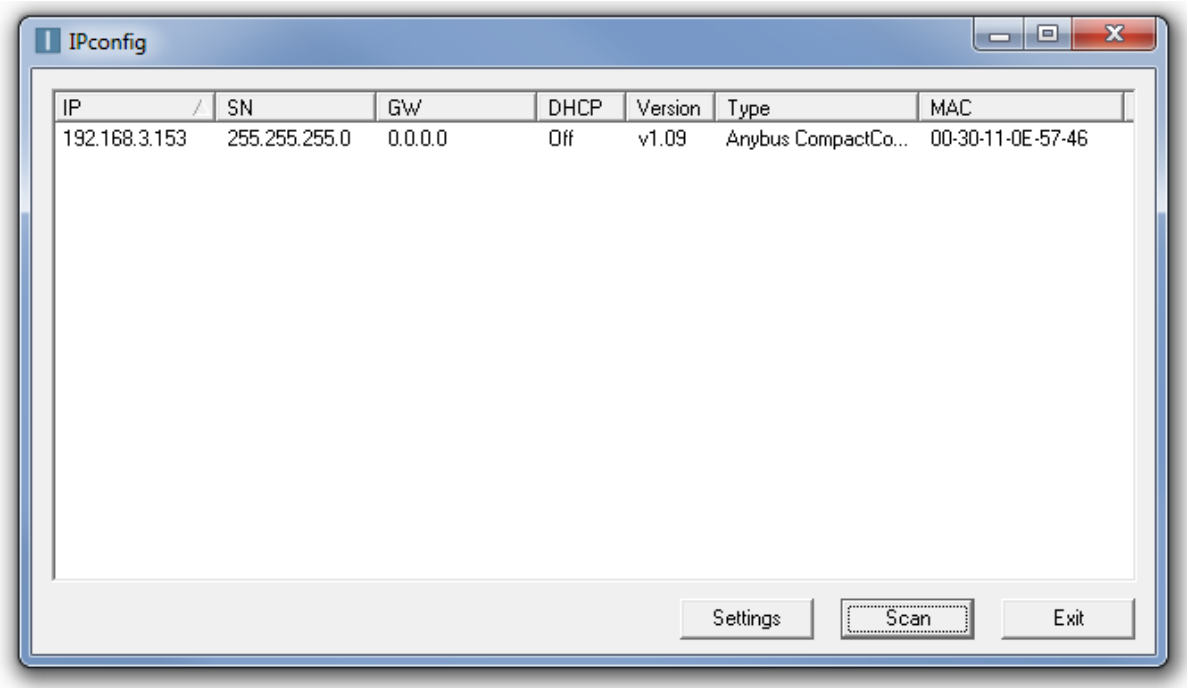
12. Setting IP Address

To change the IP address of the servo amplifier Anybus module, use IPConfig tools from HMS Anybus

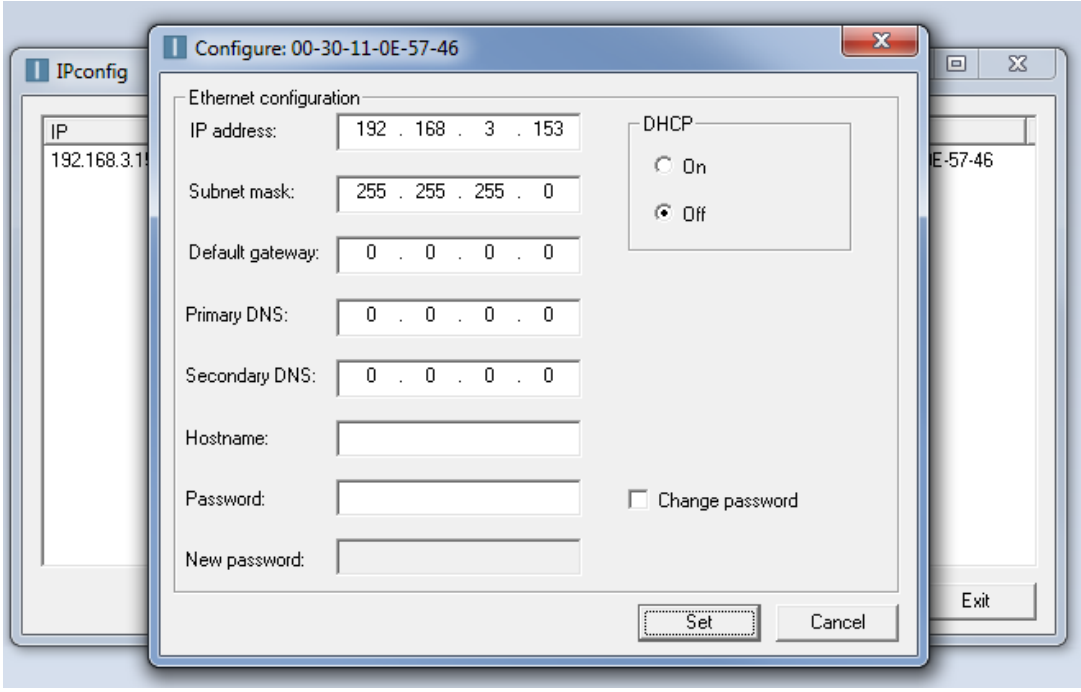
1) Launch the **Ipconfig** tools
2) Click on settings and make sure you are not wireless.



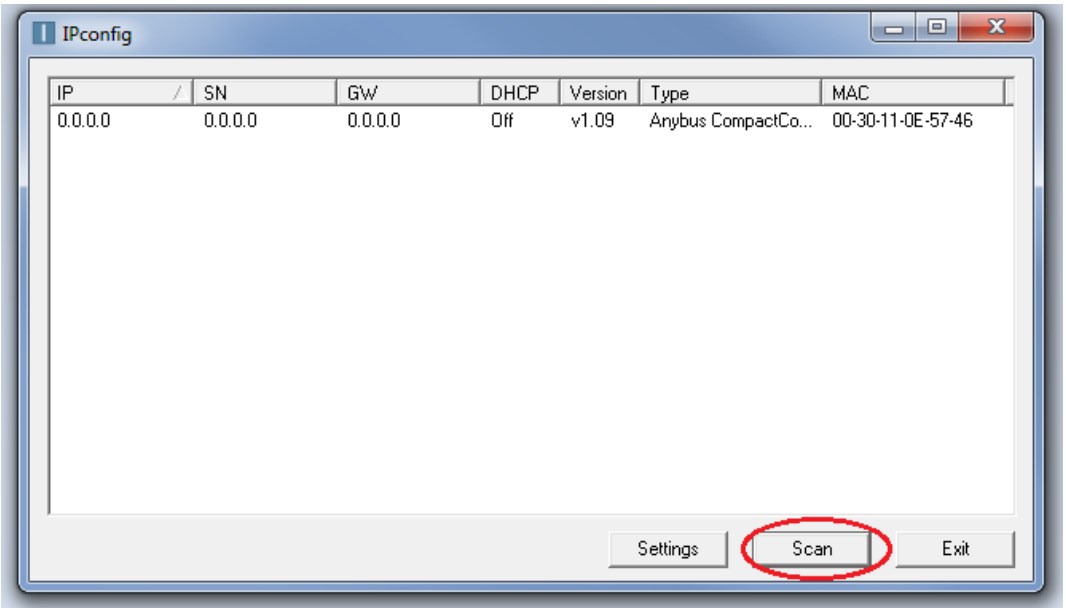
3) Click on **Scan** to find Anybus module. It will find all Anybus modules regardless of octet value.



4) To change setting, first double click on the entry to get to settings and Change the IP address here.



5) If you change anything, entry will go blank, click on **Scan** again to verify change



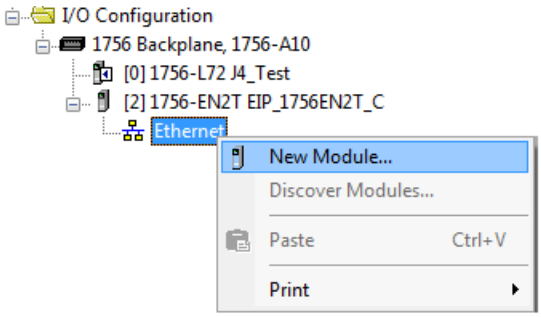
13. Rockwell PLC configuration

The configuration steps for a CompactLogix project are described this section. These steps are used to communicate with a servo amplifier. It is assumed that the user has basic knowledge in using RSLogix5000 software to perform the basic configuration steps.

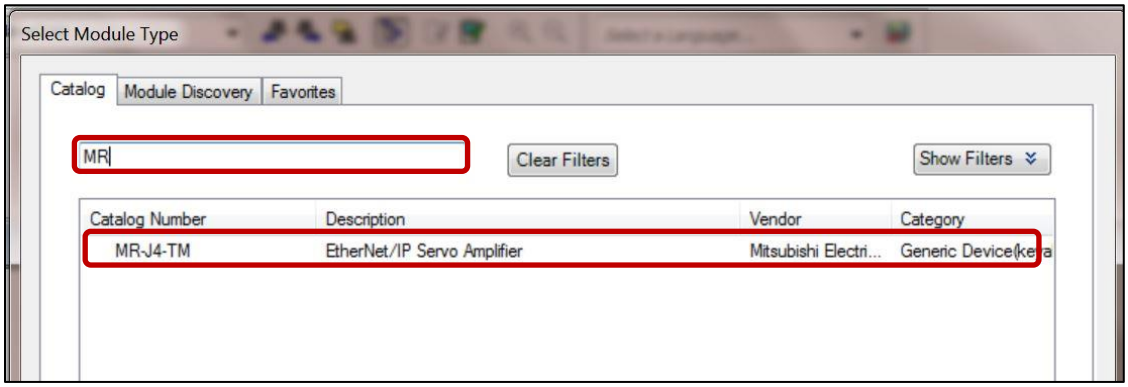
1) Install the EDS file

2) Create a new project in the RSLogix5000 using the proper revision level as the ControlLogix controller. In this example, the revision level is 20.

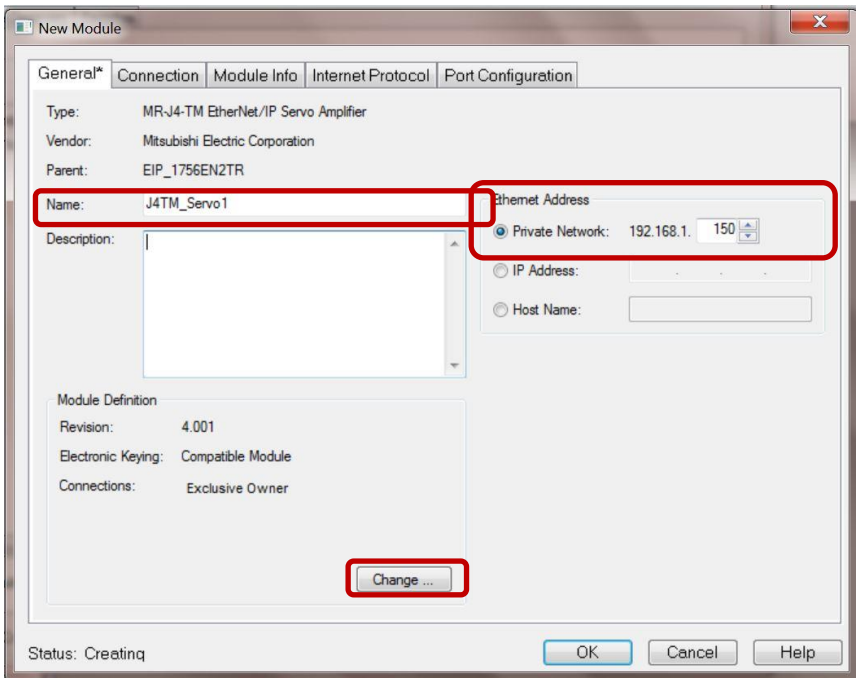
3) Under I/O configuration right click on the 1756 EN2T and Ethernet icon and and choose “New Module…”



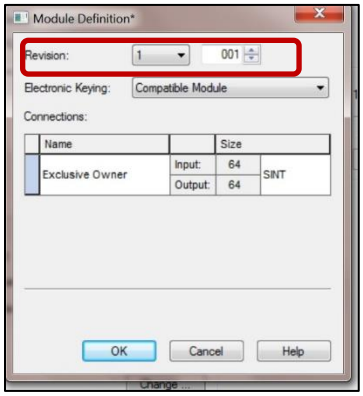
4) In the “Select Module” pop-up window, enter the “MR” in the Catalog tab and then select "MR-J4-TM" and double click on it.



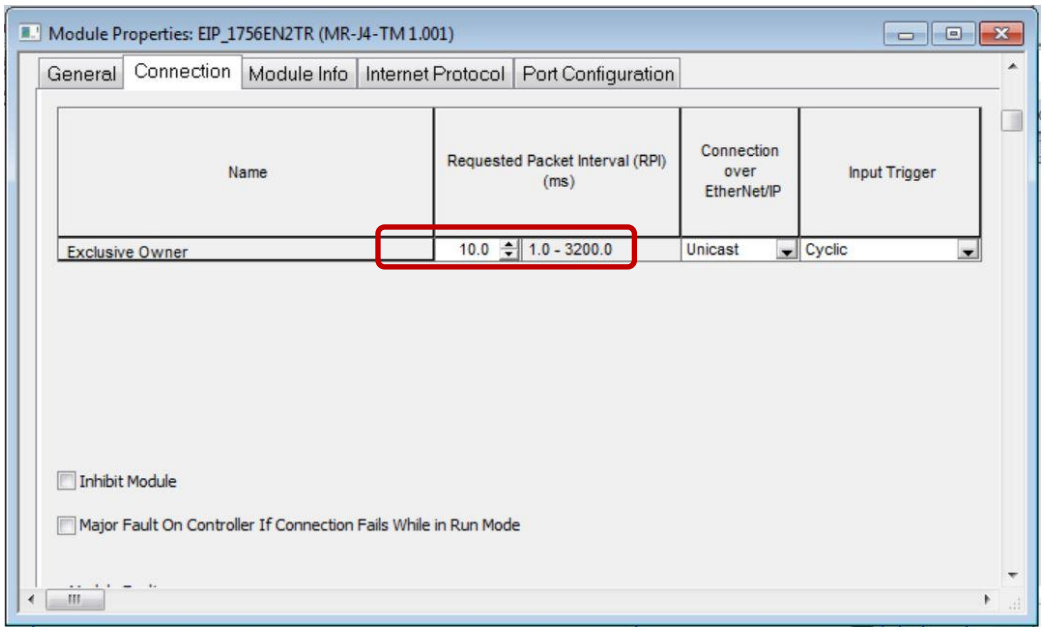
5) Enter the proper Name and IP Address and then click the **Change** button to select right revision number of MR-J4-TM drive.



6) Choose revision of MR-J4-TM amplifier or Choose the [**Disable Keying**] on the Electronic Keying option if you don't know revision number of MR-J4-TM amplifier . in this example shows the MR-J4-TM amplifier which revision number is 1.1.

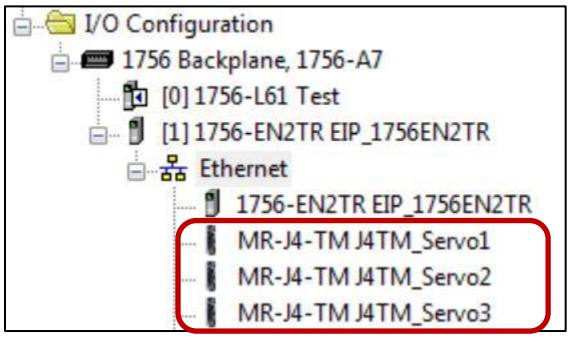


7) Click OK and then Click on connection tab to set the RPI interval between 1.0 – 100.0 ms



8) Then Click OK button. Now MR-j4-TM amplifier is added in the project.

In this example shows three MR-J4-TM is added under I/O configuration.



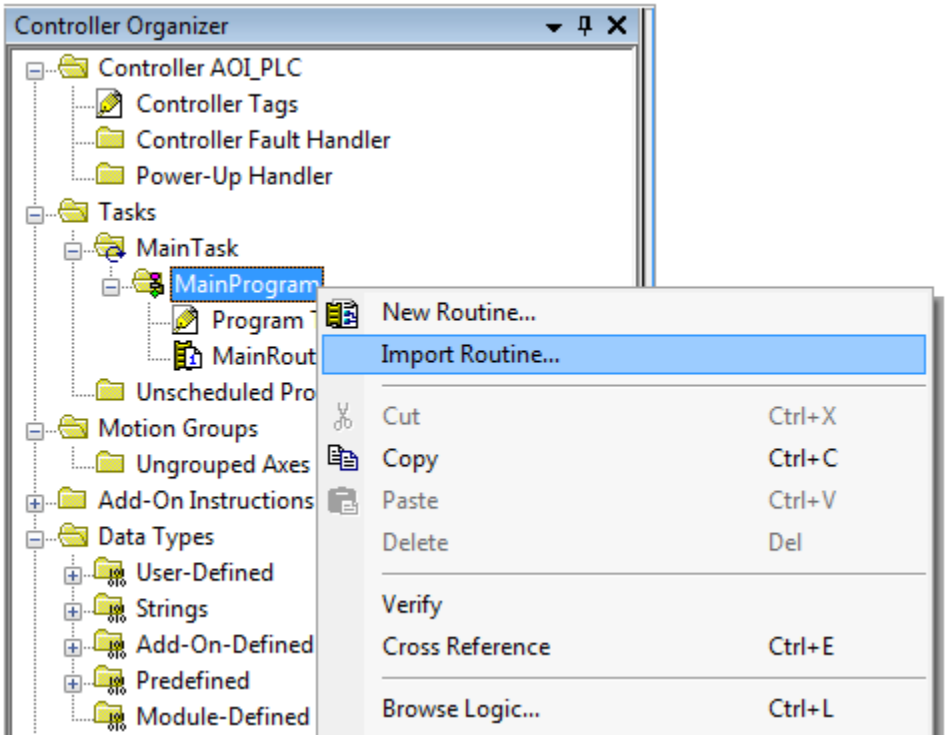
14. Importing AOIs

There are 2 ways to import the AOIs:

- Import all of them at once using AOIImportRoutine.L5X
- Import them individually from the AOI ZIP file.

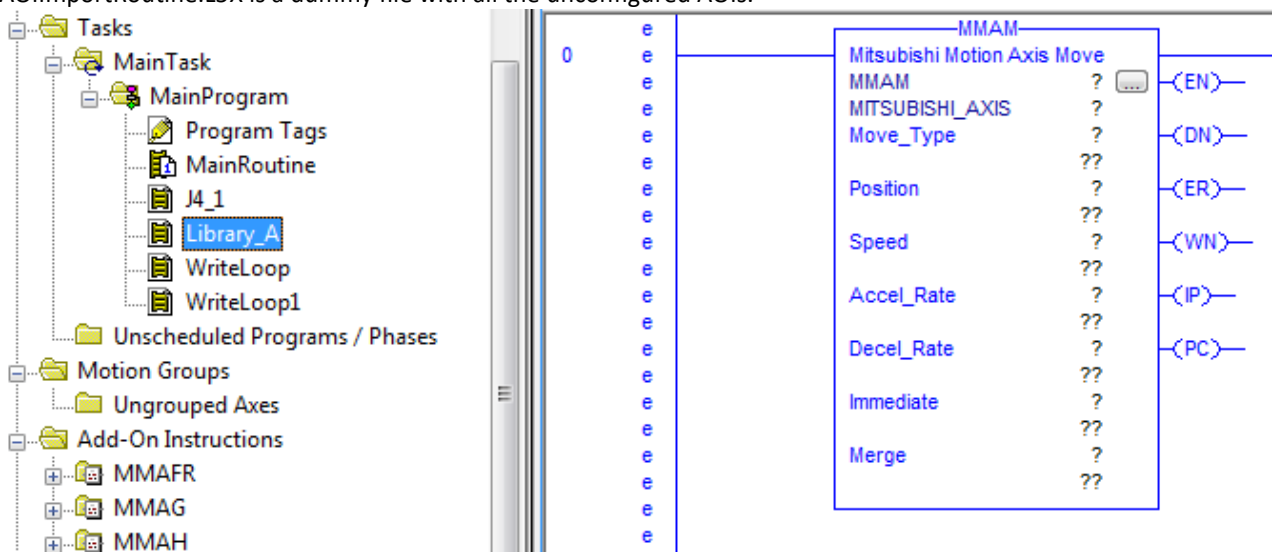
AOIImportRoutine.L5X

This is a dummy routine that has all the AOIs on several rungs. The AOIs are not configured; they have no instances and no values. When you import this library, it loads the AOIs in the tree and also loads all the support files you will need.

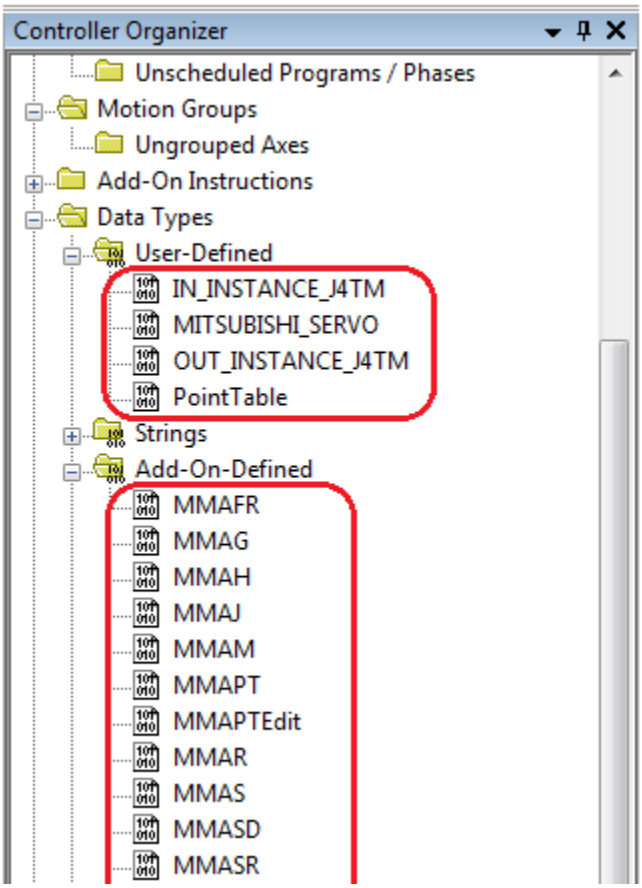


Import Routine, NOT the Import Task.

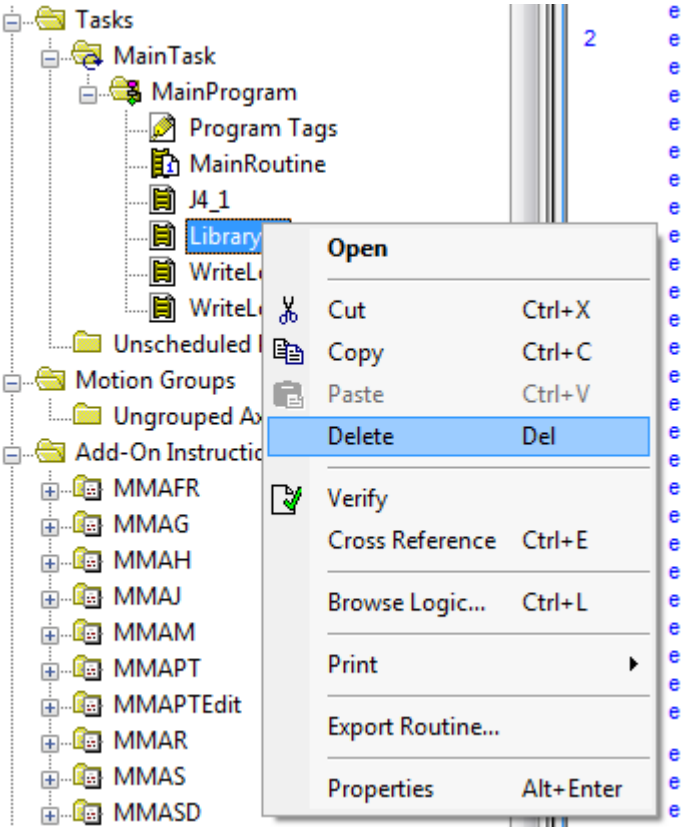
AOIImportRoutine.L5X is a dummy file with all the unconfigured AOIs.



When it is imported, it brings all the support files as well as the AOIs.



Once imported, you can delete Library subroutine. The Add-On Instructions will remain.



15. Appendices

Description of User Defined Data Structure:

MITSUBISHI_SERVO

No.	Name	Data Type	Description
1	Control	OUT_INSTANCE_J4TM	SDT of Output Instance [Servo command from PLC]
2	Status	IN_INSTANCE_J4TM	SDT of Input Instance[Status from Servo Drive]
3	Initialize	BOOL	To clear the internal state flags and the outputs of AOI (IP, ER, PC, etc)
4	ErrorReset	BOOL	To clear the error and reset of all AOIs
5	WatchEventStatus	BOOL	set when the event occurs
6	WatchEventArmedStatus	BOOL	Watch function is active
7	ShutdownStatus	BOOL	Servo is off state
8	MoveMode	BOOL	MMAM AOI is Active
9	JogMode	BOOL	MMAJ AOI is active
10	TorqueMode	BOOL	MMAT (torque mode) AOI is active
11	GearingMode	BOOL	Gearing is active
12	Homing	BOOL	MMAH AOI is active
13	Homed	BOOL	Drive Homed status
14	Direction	BOOL	Moving direction
15	RegEventStatus1	BOOL	Become true when a capture occurs (1st Input)
16	RegEventArmedStatus1	BOOL	Become true when the MMAR AOI is enabled in Logic(1st Input)
17	RegEventStatus2	BOOL	Become true when a capture occurs (2nd Input)
18	RegEventArmedStatus2	BOOL	Become true when the MMAR AOI is enabled in Logic (2nd Input)
19	Registration1PositiveEdgePosition	REAL	Mark detection capture Value in user unit
20	Registration1NegativeEdgePosition	REAL	Mark detection capture Value in user unit
21	Registration2PositiveEdgePosition	REAL	Mark detection capture Value in user unit
22	Registration2NegativeEdgePosition	REAL	Mark detection capture Value in user unit
23	CommBusy	BOOL	Reserve for future use
24	StopMove	BOOL	MMAS AOI is active
25	ResumeMove	BOOL	Reserve for future use
26	Mapping	BOOL	Reserve for future use
27	PositionConstant	REAL	Internal used for position command conversion
28	AccConstant	REAL	Internal used for Acceleration and deceleration command conversion
29	VelocityConstant	REAL	Internal used for Speed command conversion
30	GearNumerator	DINT	Electronic gear ratio setting value (Numerator [PA06])

No.	Name	Data Type	Description
31	GearDenominator	DINT	Electronic gear ratio setting value (Denominator [PA07])
32	MotorRatedSpeed	DINT	Motor rated speed (setting value: 3000,2000,1500 or 1000 rpm)
33	ActPosition	REAL	Actual position of servo drive (monitoring)in user unit
34	ActSpeed	REAL	Actual Velocity (monitoring value) in user unit/sec
35	UnitSetting	DINT	User unit setting value (0: mm 1:inches 2:Degree 3:pulse)
36	ServoReady	BOOL	Servo Ready Status signal
37	ServoError	BOOL	Servo experience Error
38	ServoWarning	BOOL	Servo experience warning
39	ConnectionFaulted	BOOL	Servo drive not responding
40	EncoderResolution	DINT	setting value (4194304 Pls/rev for Rotary encoder)
41	MaxAccelerationRate	REAL	The Maximum acceleration and deceleration limit value (user unit/Sec^2)
42	MinAccelerationRate	REAL	The Minimum acceleration and deceleration limit value (user unit/Sec^2)
43	PosCmdPositionLimit	REAL	Max position command setting allowable range (user unit)
44	NegCmdPositionLimit	REAL	Min position command setting allowable range(user unit)
45	MaxVelocityLimit	REAL	Maximum speed command setting allow(user unit)
46	ActTorque	REAL	Monitoring % of rated Torque of servo motor

Error Codes

Error codes have Extended Errors EXERR to further define the fault.

Error Code	EXERR	Description
1013	0	Invalid Axis
	1	Speed out of range (MMAM) -999999 <=> 999999
	2	Invalid Move Type – Absolute/Incremental (MMAM, MMAPT)
	3	Invalid Position
	4	Invalid Speed out of range (MMAJ, MMAM, MMAH)
	5	Invalid Acceleration rate (MMAJ, MMAM)
	6	Invalid Deceleration rate (MMAJ, MMAM)
	7	Invalid Home Type (MMAH)
	8	Invalid Torque Slope
	9	Invalid Torque Limit (MMAT)
	10	Invalid Speed = 0 (MMAM, MMAJ)
	11	Invalid Home Speed (MMAH)
	12	Invalid Home Creep Speed (MMAH)
	13	Invalid Parameter Group Invalid (MMWP, MMRP)
	14	Invalid Speed cannot be negative (MMAT)
	15	Invalid Parameter Invalid (MMWP) E.g. < 0
	16	Invalid input number (MMAR)
	17	Invalid AutoTune Response (MMCFG)
	18	Invalid Scaling Numerator (MMCFG)
	19	Invalid Scaling Denominator (MMCFG)
	20	Invalid Limit Switch Enable (MMCFG)
	21	Invalid Estop Switch Enable (MMCFG)
	22	
	23	
	24	Invalid Forward Torque Limit (MMCFG)
	25	Invalid Reverse Torque Limit (MMCFG)
	26	A single probe cannot be both continuous and latched (MMAR)
	29	Triggered MMAR while MMDR is active
	30	Invalid Gearing value - must be -1,0,1 (MMCFG)
	31	Invalid Absolute Encoder value - must be -1,0,1 – (MMCFG)
	33	Invalid Unit setting [MMCFG] or Incremental move doesn't support in Degree mode[MMAM]

Error Code	EXERR	Description
1014	AOI ID[1]	Not Homed (MMAM, MMAPT, MMAW)
1015	<error code from servo>	Servo in Warning condition SW.7
1016	AOI ID[1]	Cannot change modes unless servo is stopped/done e.g. attempt MMAM if in Jog (EXERR = 1) e.g. attempt MMAJ if Homing (EXERR = 4) e.g. attempt MMAH if in Gearing (EXERR = 6)
1017	AOI ID[1]	MMAS is on, cannot try to move anything.
1018	AOI ID[1]	Cannot trigger gearing if in JogMode, TorqueMode or Homing. EXERR is what mode you are in
1020	<error code from servo>	Servo in Error condition SW.3
1021	AOI ID[1]	Servo is disabled and a move instruction was attempted – servo may be shutdown, not enabled
1022	<Which parameter>	EXERR value 16#10 (16) = Illegal code number. • Check Parameter InstructionCode2 E.g. You wrote "6083" instead of "16#6083", Subindex out of range
1023	<Which parameter>	16#20 (32) = Unreachable parameter. Check WriteData –
1024	<Which parameter>	16#30 (48) = Out of range Address out of range or value out of range E.g. For writing data I out 16#6064. I should have put 16#6064_0000
1025	<index line>	An invalid entry in the Point Table UDT. EXERR gives the index number
1026	AOI ID[1]	CommBusy Watchdog – CommBusy on too long, no communication with J4
1027	0	Gearing flags not responding. Servo may have lost reprogramed Gearing registers. Intialize or recycle power to PLC.
1028	AOI ID[1]	Attempted motion while AOI is in Initialize state
1050	1	Gearing dropped out (2D16.14 went low)
1060	AOI ID	Wrong unit

[1] In the Mitsubishi_Servo UDT, the following are placed in the extended register EXERR to indicate which AOI instruction set this error

(e.g. 1017, 1021) or which mode/AOI is active (e.g. 1016)

1 = MMAJ - Jog
2 = MMAM - Move/Position
3 = MMAT - Torque
4 = MMAH - Home
5 = MMAPT – Point Table
6 = MMAG – Gearing
7 = MMAW - Watch
8 = MMCFG – Configure Servo
9 = MMAR – Registration
10 = MMAFR – Fault Reset

Cross Reference of CANOpen Object vs J4 Parameters

You can access the servo parameters two ways. There is the original CANOpen address and there is a mapping table so that J4 parameters have an equivalent CANOpen address.

J4	CANOpen
PA_ _	2001h to 2020h
PB_ _	2081h to 20C0h
PC_ _	2101h to 2150h
PD_ _	2181h to 21B0h
PE_ _	2201h to 2240h
PF_ _	2281h to 22C0h
PL_ _	2401h to 2430h
PT_ _	2481h to 24D0h
PN_ _	2581h to 25A0h

Note:
J4 parameters are decimal
CANOpen parameters are hexadecimal

PA14 <> 2014h
PA14 = 200Eh

Note: Parameters starts at 1, not 0.
E.g. there is no PA00.

For example the CANOpen Home object is 6099h. Sub-index 1 is the homing speed and sub-index 2 is the creep speed.

Index	Sub	Name	Data Type	Access	PDO Mapping
6099h	0	Homing speeds	UNSIGNED8	ro	Impossible
	1	Speed during search for switch	UNSIGNED32	rw	Possible
	2	Speed during search for zero			

In the J4 parameters PT05 is the homing speed and PT06 is the creep speed.

Positioning control Selected Items Write

No.	Abbr.	Name	Units	Setting range
PT01	**CTY	For manufacturer setting		0200-0300
PT02	*TOP1	For manufacturer setting		0000-0001
PT03	*FTY	For manufacturer setting		0000-0300
PT04	*ZTY	For manufacturer setting		0000-0300
PT05	ZRF	Home position return speed	r/min	0.00-167772.15
PT06	CRF	Creep speed	r/min	0.00-167772.15

Using the table above, PT05 is 2485h and PT06 is 2486h. ***With the mapping table, all sub-indexes are 0***.

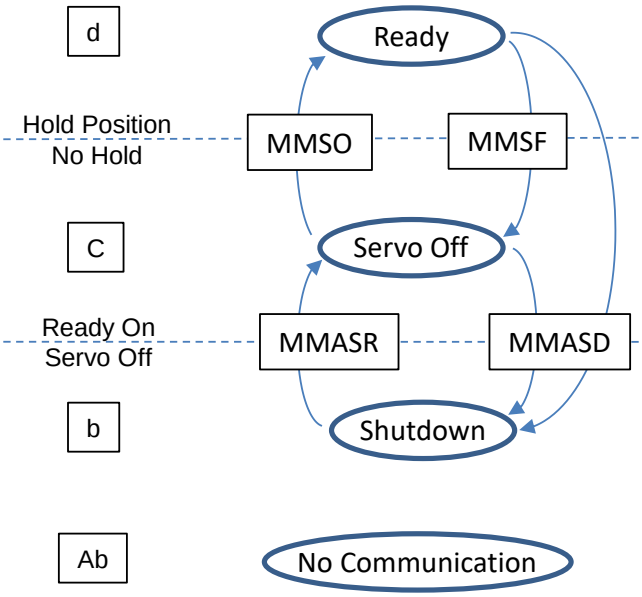
Therefor you can address the Home creep speed 2 ways

- Using index = 6099h and sub-index = 2 or
- Using index = PT06 > 2486h and sub-index = 0.

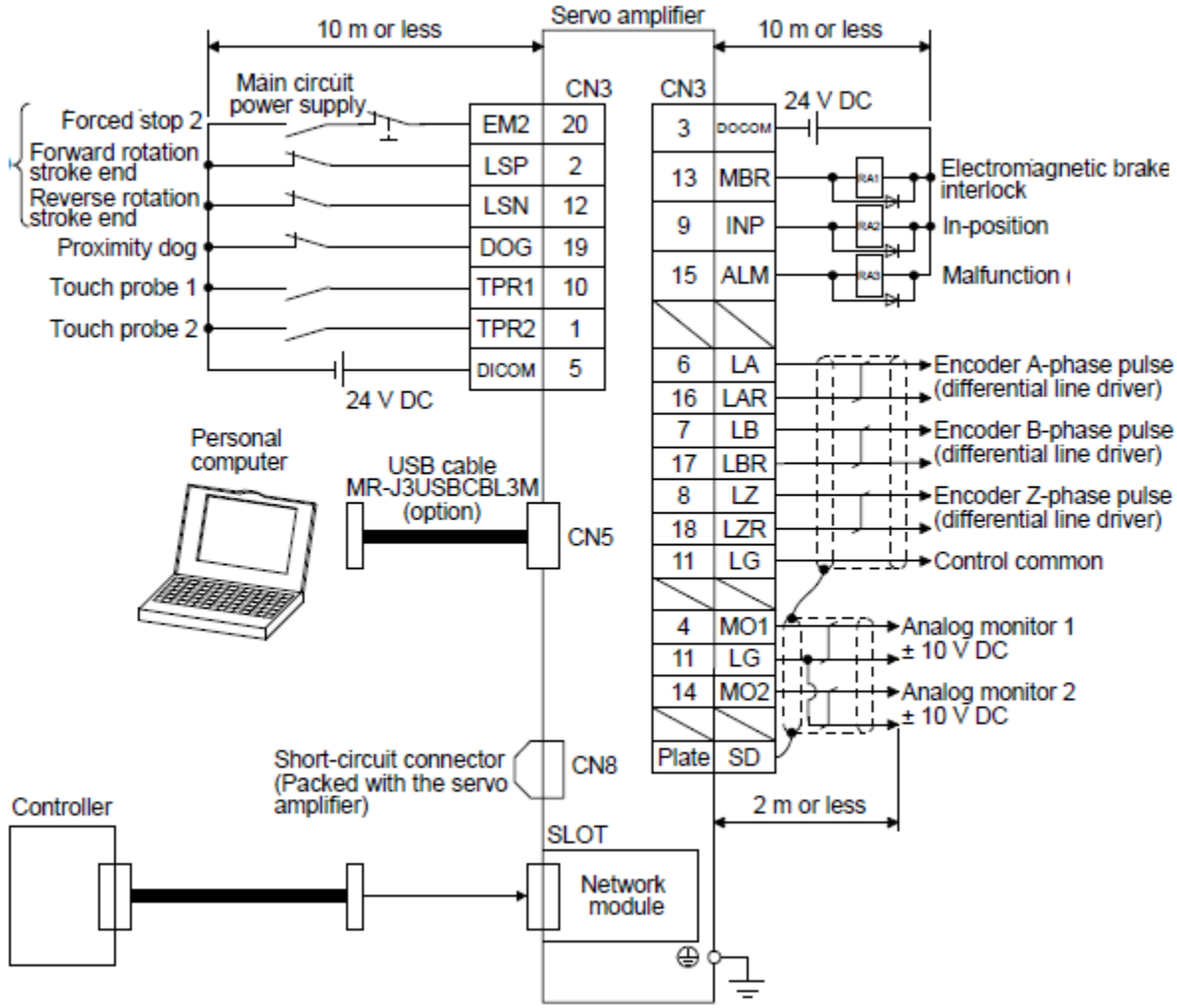
LED & Servo States

d	"d" : Indicates ready-on and servo-on status MMSO Servo On - enables the servo to “d”
C	"C": Indicates ready-on and servo-off status. MMSF Servo Off – • disables the servo to “C” from “d” • If at “C” or “b”, no effect. MMASR Shutdown Reset – resets <servo>.ShutdownStatus flag • If at “b” enables the servo to “C” • If at “C” or “d”, no effect.
b	"b" : Indicates ready-off and servo-off status. MMASD Shutdown - electronically disables the servo to “b”. Must trigger MMASR to reset it internally. Uses <servo>.ShutdownStatus flag.
Ab	No communication between J4 and PLC. Ping both IP addresses for J4 Anybus and PLC Generic Ethernet module. If they ping, then the problem is in the PLC program.

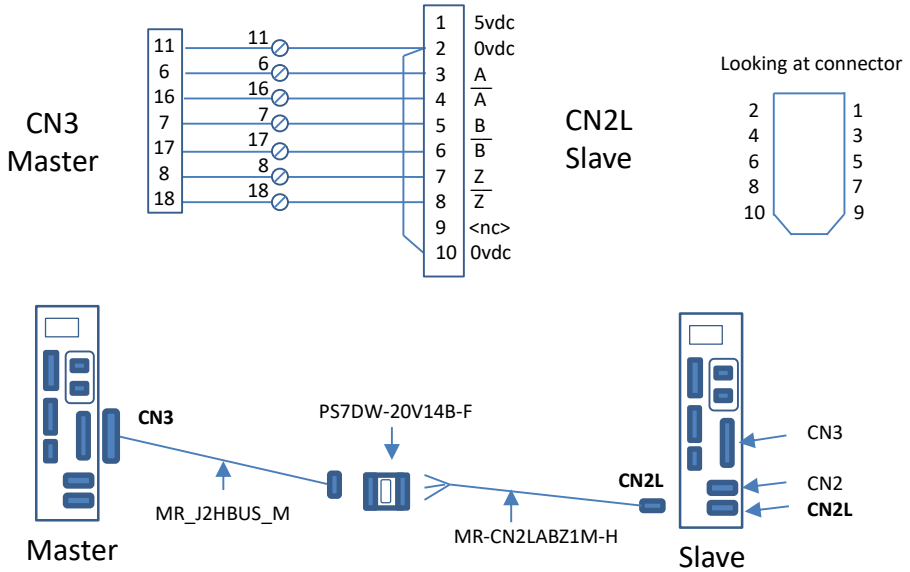
(Power On state based on state at Power Off)



Servo Digital I/O interface Connections



Master/Slave Interface Connections for Gearing



The cable from DIN rail mounted terminal block to CN3

MR-J2HBUS__M	05, 1, 3, 5

e.g.

MR-J2HBUS05M for 0.5 meters

The DIN rail mounted terminal block is PS7DW-20V14B-F

This part number of CN2L connector at one end and flying leads at the other is MR-CN2LABZ1M-H.

Write Loop 1

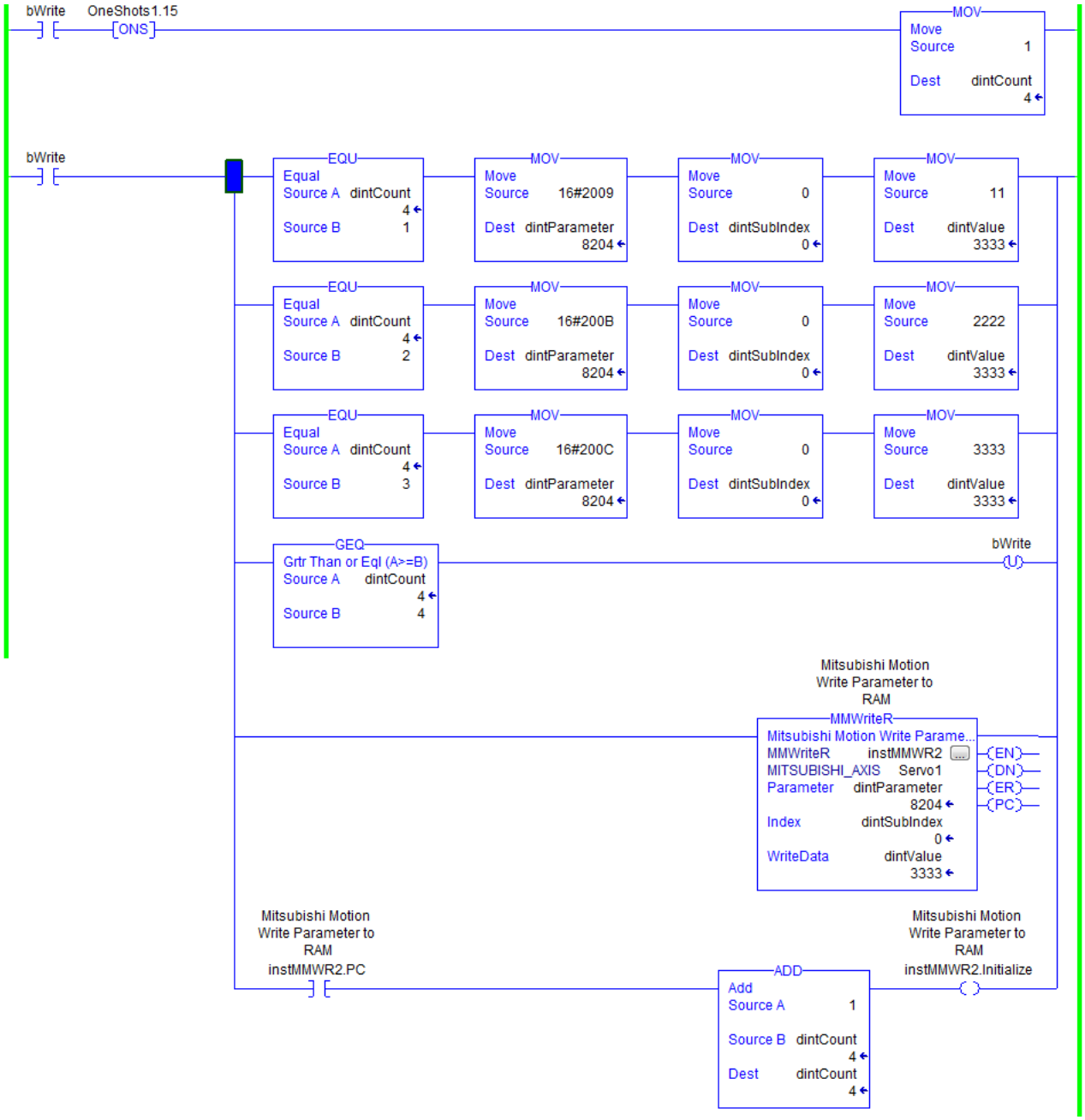
In the following example, 3 parameters are written to the servo:

16#2009 – Auto tuning parameter PA09

16#200A – Forward torque limit PA11

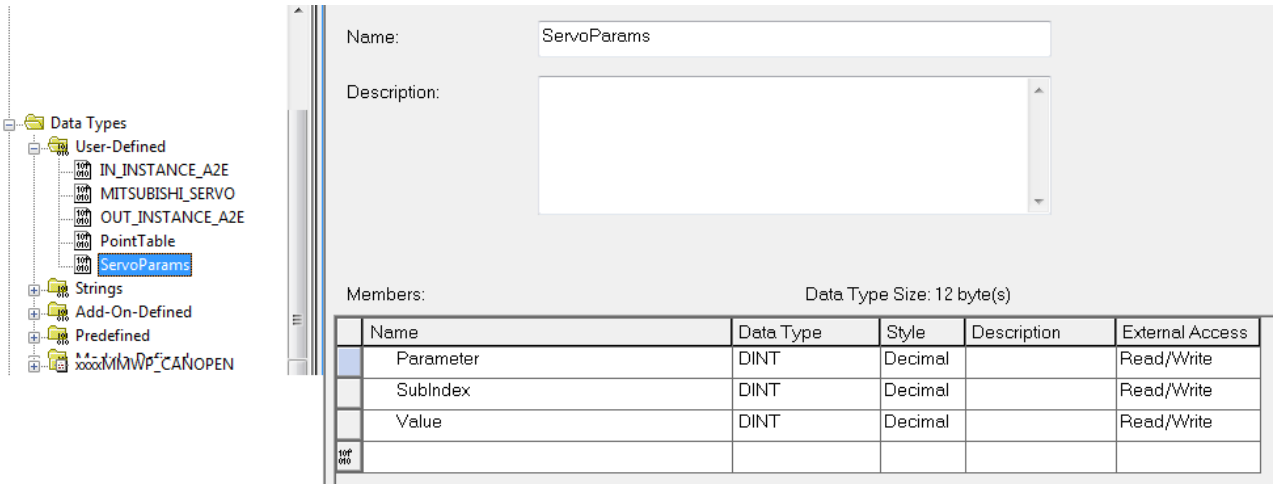
16#200B – Reverse torque limit PA12

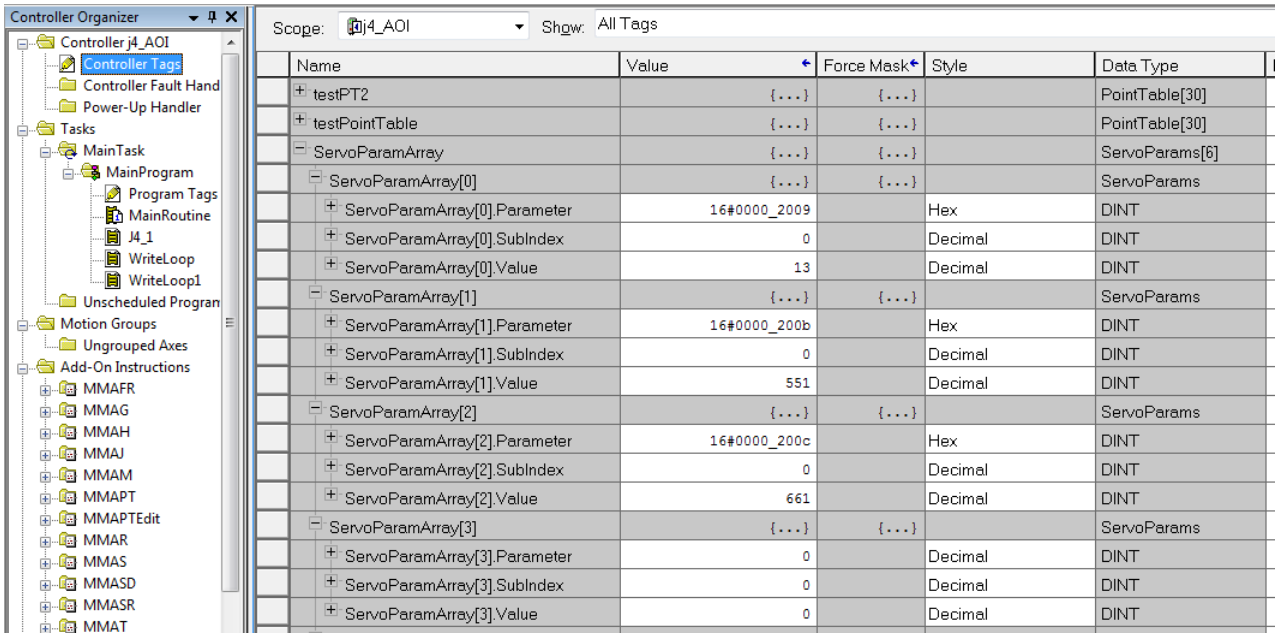
You set **bWrite** high and it will reset itself after the loop is done. You can increase the number of parameters, you simply update the **GEQ** to turn off the loop.

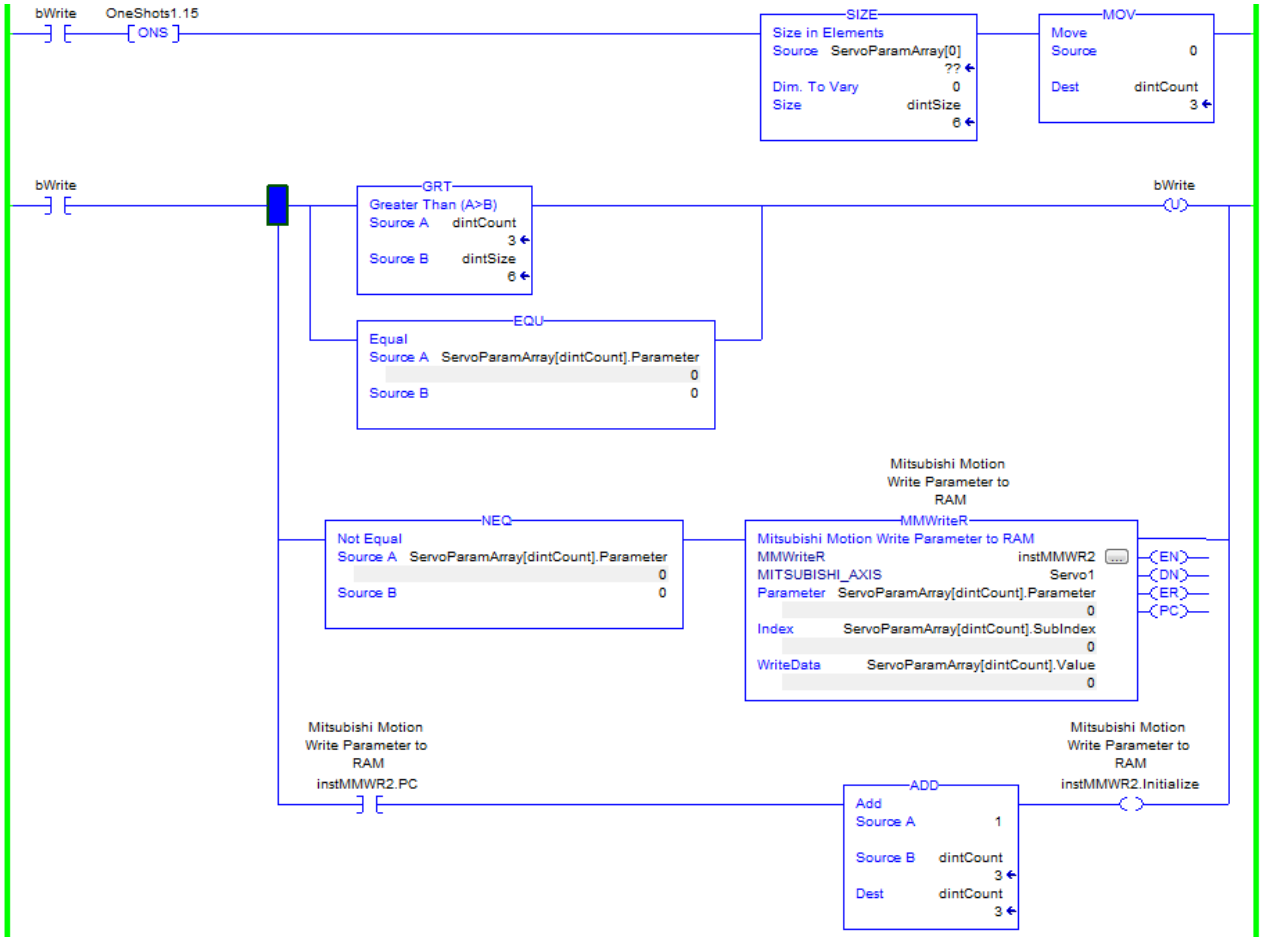


Write Loop 2

For this example, the parameters are stored as an array. The ladder logic sequences thru the array to the end or if it encounters a 0 parameter.





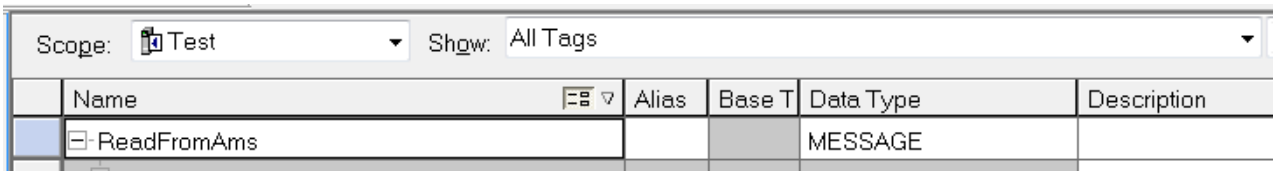

bWrite
OneShots1.15
ONS
SIZE
Size in Elements
Source ServoParamArray[0]
??
Dim. To Vary 0
Size dintSize
6
MOV
Move
Source 0
Dest dintCount
3
bWrite
GRT
Greater Than (A>B)
Source A dintCount
3
Source B dintSize
6
bWrite
U
EQU
Equal
Source A ServoParamArray[dintCount].Parameter
0
Source B 0
Mitsubishi Motion
Write Parameter to
RAM
NEQ
Not Equal
Source A ServoParamArray[dintCount].Parameter
0
Source B 0
MMWriteR
Mitsubishi Motion Write Parameter to RAM
MMWriteR instMMWR2
MITSUBISHI_AXIS Servo1
Parameter ServoParamArray[dintCount].Parameter
0
Index ServoParamArray[dintCount].SubIndex
0
WriteData ServoParamArray[dintCount].Value
0
EN
DN
ER
PC
Mitsubishi Motion
Write Parameter to
RAM
instMMWR2.PC
Mitsubishi Motion
Write Parameter to
RAM
instMMWR2.Initialize
ADD
Add
Source A 1
Source B dintCount
3
Dest dintCount
3

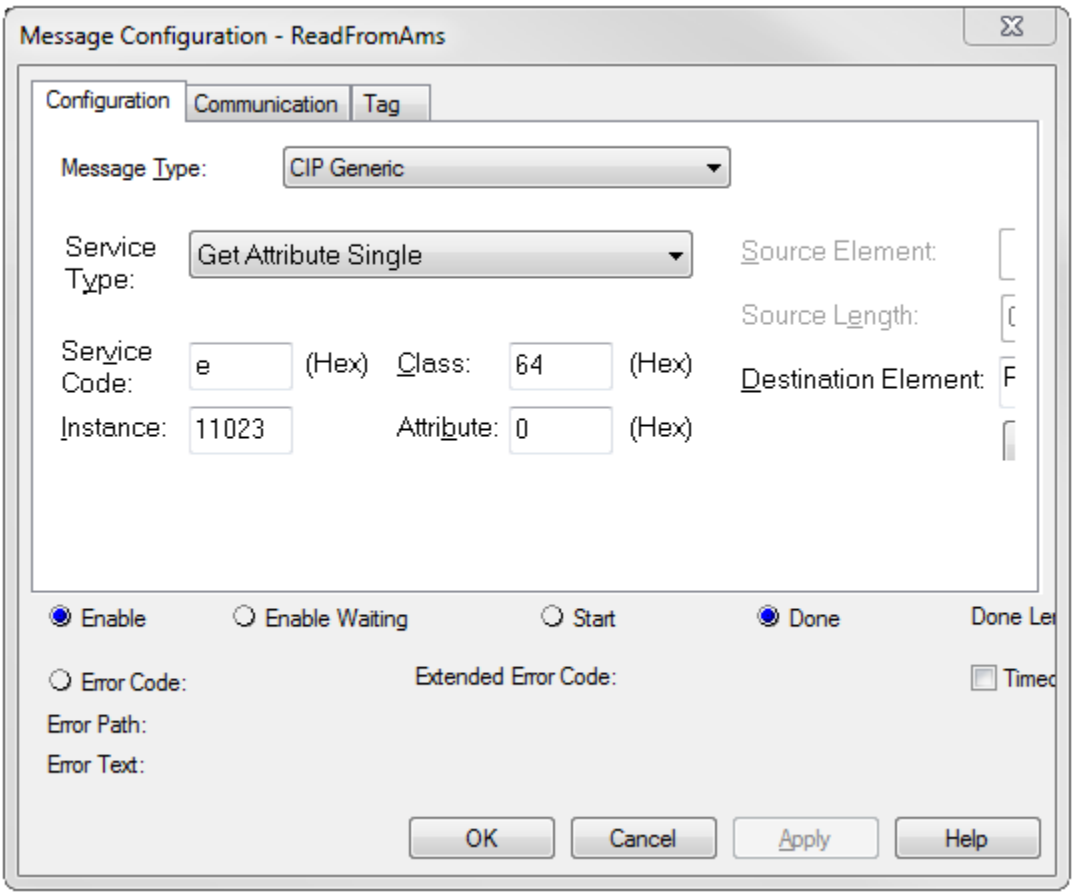
Initialize

Most of the AOIs have an Initialize input that is found under the DOT operator. E.g. instMMAJ.Intialize.

- Clears the internal state flags and the outputs (IP, ER, PC, etc).
- Clears all error messages (except the following)
- With motion AOIs (MMAH, MMAM, MMAJ, MMAT) it will set an error if you have the motion rung energized or if you try to trigger a motion AOI when Initialized is asserted. Error 1028.
- ***Initialize DOES NOT STOP MOTION. It only initializes the AOI state.***
- Once set, it will lock out the AOI until Initialize is reset and the motion rung is NOT active (in the case of motion AOIs MMAH, MMAM, MMAJ, MMAT).

Explicit MSG





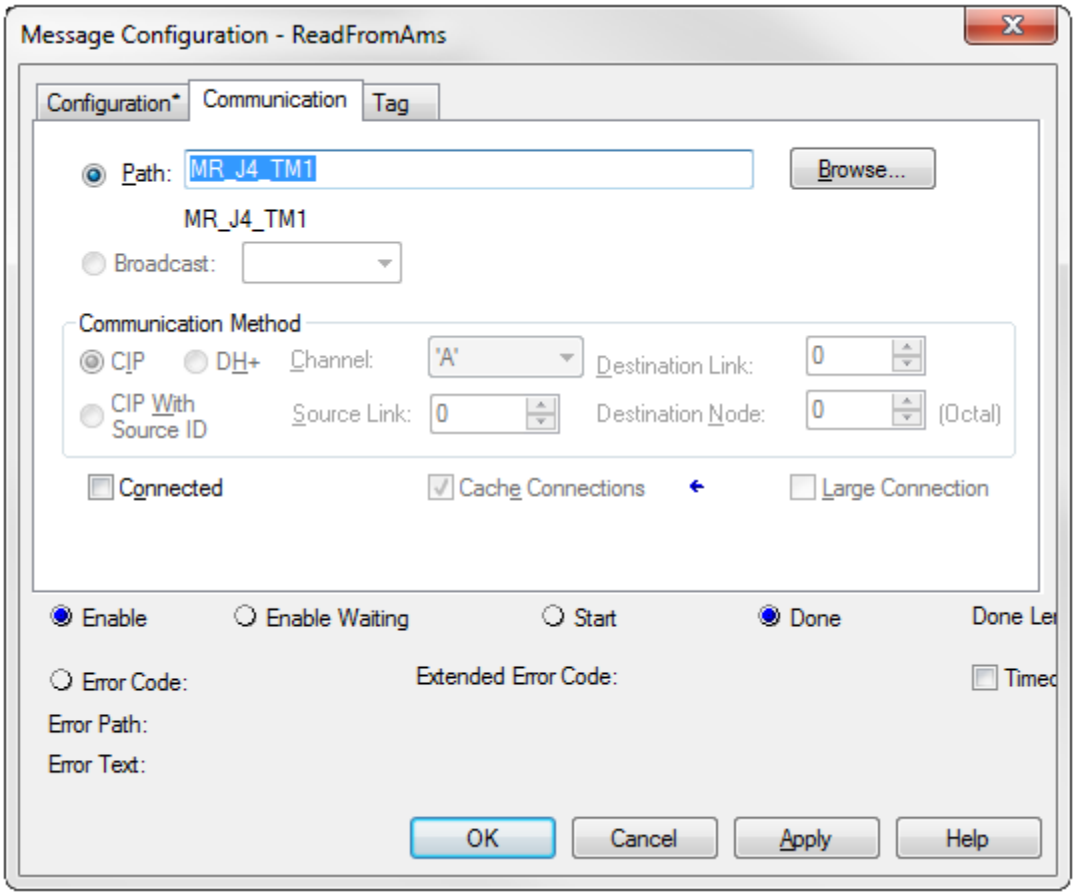
Note the Instance is NOT hex. 11023 = 2B0F Hex

- Message Type = CIP Generic
- Service Type = Get Single Attribute (automatically sets Service Code)
- Service Code (HEX) = "e" - Get Single Attribute
- Class (HEX) = 64 - this is the DS402 Drive Object

(13) Bus voltage (2B0Fh)

Ins ID	Attr ID	Access	Name	Data Type
2B0Fh	0	Get	Bus voltage	UINT

- Instance ID (Decimal) = the object **IN DECIMAL** - 11023 = 2B0F Hex. This is Bus Voltage.
- Attribute ID - the index within the Instance ID - see documentation. Typically 0 for single INT. Some items have indexes 0, 1, 2, …



16. Major Revisions

Date	Version	Description
July 19, 2019	3.000	Document created and Released
June 9, 2021	3.002	Modified

17. References

Company	Document No.	Document Name	Version	Date
Mitsubishi	SH(NA)030226-B	MR-J4-_TM_ SERVO AMPLIFIER INSTRUCTION MANUAL (EtherNet/IP)	B	10/22/2014
	SH(NA)030193ENG-F	Multi-Network Interface AC Servo MELSERVO-J4 Servo amplifier INSTRUCTION MANUAL	F	01/10/2015
	SH(NA)030109	MELSERVO-J4 Servo amplifier INSTRUCTION MANUAL (TROUBLE SHOOTING)	*	12/18/2015
HMS		Driver manual Anybus CompactCom 40 motion driver	1.11	10/9/2014
		EtherNet/IP Extension Anybus CompactCom 40 motion driver	1.03	10/7/2014