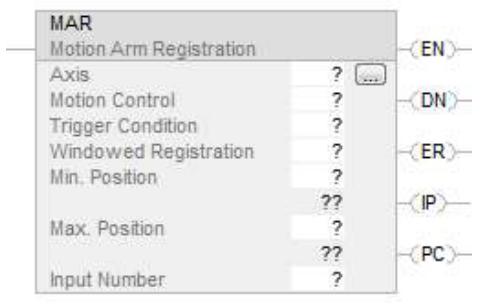
Motion Arm Registration (MAR)

This information applies to the CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, and GuardLogix 5580 controllers. Controller differences are noted where applicable.

Use the Motion Arm Registration (MAR) instruction to arm registration event checking for the specified axis. When the instruction is called, a registration event is armed based on the selected Registration Input and the specified Trigger Condition. When the specified Registration Input transition satisfies the Trigger Condition, the axis computes the axis position at the moment the event occurred based on hardware latched encoder count data and stores it in the associated Registration Position variable in the axis data structure. Also, the instruction's Event (PC) bit is simultaneously set, as well as the Registration Event Status bit in the axis data structure. If Windowed Registration is selected, only registration events whose computed registration position falls within the Max and Min Position window are accepted. If the Registration Position falls outside this window the registration event checking is automatically rearmed.

Available Languages

Ladder Diagram



Function Block

This instruction is not available in function block.

Structured Text

MAR(Axis,MotionControl, TriggerCondition, WindowedRegistration, MinimumPosition, MaximumPosition, InputNumber);

Operands

Ladder Diagram

Operand	Type CompactLogix 5370, Compact GuardLogix 5370, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480	Type ControlLogix 5570, GuardLogix 5570, ControlLogix 5580, and GuardLogix 5580 controllers	Format	Description

Axis	AXIS_CIP_DRIVE	AXIS_CIP_DRIVE AXIS_SERVO AXIS_SERVO_DRIVE AXIS_GENERIC_DRIVE AXIS_GENERIC **Tip:** AXIS_GENERIC is supported by the ControlLogix 5570 and the GuardLogix 5570 controllers only.	Tag	Name of the axis to perform operation on
Motion Control	MOTION_INSTRUCTION	MOTION_INSTRUCTION	Tag	Structure used to access instruction status parameters.
Trigger Condition	BOOLEAN	BOOLEAN	Immediate	Defines the Registration Input transition that defines the registration event. Select either: 0 = trigger on positive edge 1 = trigger on negative edge.
Windowed Registration	BOOLEAN	BOOLEAN	Immediate	Enable (1) if registration is to be Windowed, that is, that the computed Registration Position must fall within the specified Min and Max Position limits to be accepted as a valid registration event. Select either: 0 = disabled 1 = enabled.

Minimum Position	REAL	REAL	Immediate or Tag	Used when Windowed Registration is enabled. Registration Position must be greater than Min. Position limit before registration event is accepted.
Maximum Position	REAL	REAL	Immediate or Tag	Used when Windowed Registration is enabled. Registration Position must be less than Max. Position limit before registration event is accepted.
Input Number	UINT32	UINT32	1 or 2	Specifies the Registration Input to select. 1 = Registration 1 Position 2 = Registration 2 Position.

See *Structured Text Syntax* for more information on the syntax of expressions within structured text.

For the operands that require you to select from available options, enter your selection as:

This Operand	**Has These Options Which You**	
	Enter as Text	**Or Enter as a Number**
TriggerCondition	positive_edge negative_edge	0 1
WindowedRegistration	disabled enabled	0 1

MOTION_INSTRUCTION Structure

Mnemonic	**Description**

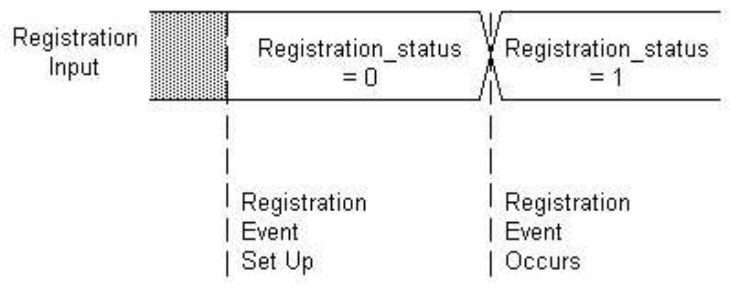
.EN (Enable) Bit 31	It is set to true when the rung makes a false-to-true transition and remains set until the servo message transaction is completed and the rung goes false.
.DN (Done) Bit 29	It is set to true when the axis registration event checking has been successfully armed.
.ER (Error) Bit 28	It is set to to true to indicate that the instruction detected an error, such as if you specified an unconfigured axis.
.IP (In Process) Bit 26	It is set to true on positive rung transition and cleared to false after the registration event has occurred, or has been superseded by another Motion Arm Reg command, or terminated by a Motion Disarm Reg command.
.PC (Process Complete) Bit 27	It is set to true when a registration event occurs.

Description

The MAR instruction sets up a registration event to store the actual positions of the specified physical axis on the specified edge of the selected dedicated high speed Registration input for that axis.

When an MAR instruction is executed, the RegEventStatus bit is set to 0 (FALSE) and the selected Registration input for the specified axis is monitored until a Registration input transition of the selected type (the registration event) occurs. When the registration event occurs, the RegEventStatus bit for the axis is set to 1 (TRUE) and the Actual Position of the axis is stored in the Registration Position variable corresponding to the registration input (for example, Registration 1 Position 1 or Registration 2 Position).

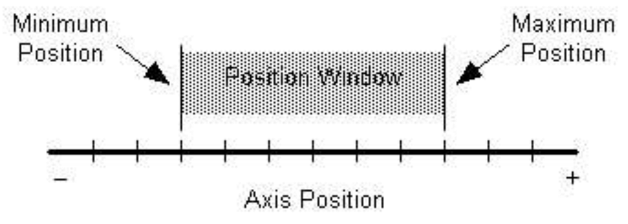
Registration



Multiple registration events may be active at any time for a given axis, but only one may be active per registration input. Each event is monitored independently and may be checked using the appropriate RegEventStatus bit.

Windowed Registration

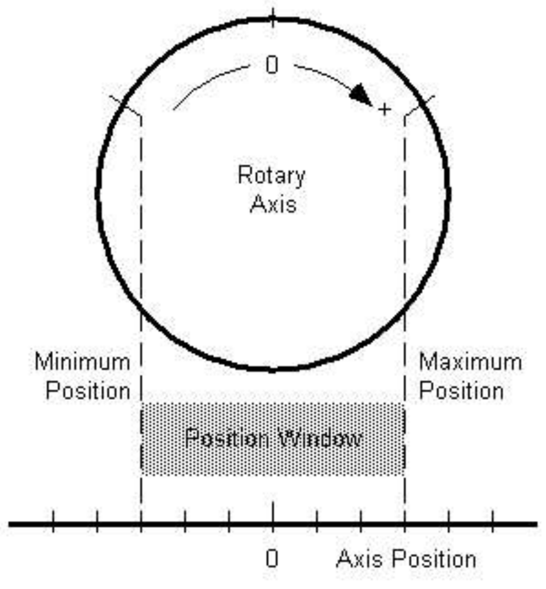
When the Windowed Reg checkbox is checked, the selected trip state only results in a registration event if it occurs when the axis is within the window defined by the minimum and maximum positions as shown below.



Enter values or tag variables for the desired absolute positions that define the position window within which the selected trip state of the Registration input is valid. Windowed registration is useful in providing a mechanism to ignore spurious or random transitions of the registration sensor, thus improving the noise immunity of high-speed registration inputs.

For linear axes, the values can be positive, negative, or a combination. However, the Minimum Position value must be less than the Maximum Position value for the registration event to occur. For rotary axes, both values must be less than the unwind value set in the motion controller's machine setup menu. The Minimum Position value can be greater than the Maximum Position value for registration windows that cross the unwind point of the axis, as shown below.

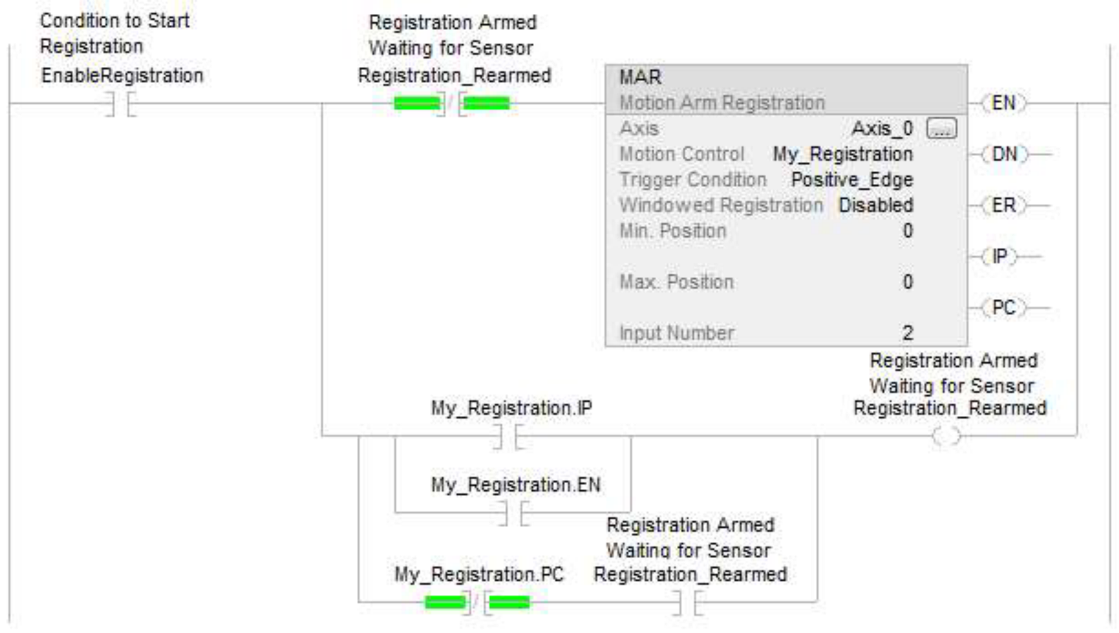
Position Window for Rotary Axis



Rearming an MAR Instruction

If your application requires rapid and continuous detection of a registration sensor, we recommend that you use the following logic:

Ladder Logic for Continuous Registration Detection


Condition to Start Registration
EnableRegistration
Registration Armed Waiting for Sensor
Registration_Rearmed
MAR
Motion Arm Registration
Axis Axis_0
Motion Control My_Registration
Trigger Condition Positive_Edge
Windowed Registration Disabled
Min. Position 0
Max. Position 0
Input Number 2
EN
DN
ER
IP
PC
Registration Armed Waiting for Sensor
Registration_Rearmed
My_Registration.IP
My_Registration.EN
My_Registration.PC
Registration Armed Waiting for Sensor
Registration_Rearmed

To rearm the MAR instruction, the rung must change from false to true. The rate at which this logic functions depends on the following:

- program scan time
- motion task coarse update rate

Important:	In large I/O connections, force values can slow down the rate at which the controller processes repetitive motion registration.

To successfully execute a MAR instruction, the targeted axis must be configured as either a Servo or Feedback Only axis. Otherwise, the instruction errs.

Important:	The instruction execution may take multiple scans to execute because it requires multiple coarse updates to complete the request. The Done (.DN) bit is not set immediately, but only after the request is completed.

In this transitional instruction, the relay ladder, toggle the Rung-condition-in from cleared to set each time the instruction should execute.

Affects Math Status Flags

No

Major/Minor Faults

None specific to this instruction. See *Common Attributes* for operand-related faults.

Execution

Ladder Diagram

Condition/State	**Action Taken**
Prescan	The .EN, .DN, .ER, and .IP bits are cleared to false.
Rung-condition-in is false	The .EN bit is cleared to false if either the .DN or .ER bit is true.
Rung-condition-in is true	The .EN bit is set to true and the instruction executes.
Postscan	N/A

Structured Text

Condition/State	**Action Taken**
Prescan	See Prescan in the Ladder Diagram table.
Normal execution	See Rung-condition-in is false, followed by rung is true in the Ladder Diagram table.
Postscan	See Postscan in the Ladder Diagram table.

Error Codes

See *Motion Error Codes (.ERR)* for Motion Instructions.

Extended Error Codes

Extended Error Codes provide additional instruction specific information for the Error Codes that are generic to many instructions. The following Extended Error codes help to pinpoint the problem when the MAR instruction receives a Servo Message Failure (12) error message. See *Motion Error Codes (.ERR)* for Motion Instructions.

Associated Error Code (decimal)	**Extended Error Code (decimal)**	**Meaning**
SERVO_MESSAGE_FAILURE (12)	No Response (2)	Not enough memory resources to complete request. (SERCOS)
SERVO_MESSAGE_FAILURE (12)	Invalid value (3)	Registration input provided is out of range.
SERVO_MESSAGE_FAILURE (12)	Device in wrong state (16).	Redefine Position, Home, and Registration 2 are mutually exclusive. (SERCOS)

Extended Error codes for the Parameter Out of Range (13) error code work a little differently. Rather than having a standard enumeration, the number that appears for the Extended Error code refers to the number of the operand as they are listed in the faceplate from top to bottom with the first operand being counted as zero. Therefore for the MAR instruction, an extended error code of 4 would refer to the Min Position value. You would then have to check your value with the accepted range of values for the instruction.

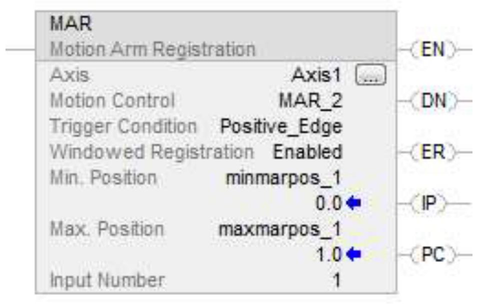
Status Bits

MAR Changes to Status Bits

Bit Name	**State**	**Meaning**
RegEvent1ArmedStatus RegEvent2ArmedStatus	TRUE	The axis is looking for a registration event.
RegEvent1Status RegEvent2Status	FALSE	The previous registration event is cleared.

Example

Ladder Diagram



```
MAR
Motion Arm Registration                    -(EN)-
Axis                      Axis1  ...
Motion Control            MAR_2            -(DN)-
Trigger Condition   Positive_Edge
Windowed Registration     Enabled          -(ER)-
Min. Position         minmarpos_1
                              0.0 ←        -(IP)-
Max. Position         maxmarpos_1
                              1.0 ←        -(PC)-
Input Number                    1
```

Structured Text

MAR(Axis1, MAR_2, Positive_Edge, enabled, minmarpos_1, maxmarpos_1, 1);

See also

[Motion Event Instructions]

[Motion Error Codes (.ERR)]

[Structured Text Syntax]

[Common Attributes]

[MAR Flow Chart]

How are we doing?