

[Instruction Set](#) > [Multi-Axis Coordinated Motion Instructions](#) > Motion Coordinated Linear Move (MCLM)

Motion Coordinated Linear Move (MCLM)

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.

Use the MCLM instruction to start a single or multi-dimensional linear coordinated move for the specified axes within a Cartesian coordinate system. You can define the new position as absolute or incremental.

Important: Tags used for the motion control attribute of instructions should only be used once. Re-use of the motion control tag in other instructions can cause unintended operation. This may result in damage to equipment or personal injury.

Important: Risk of Velocity and/or End Position Overshoot

If you change move parameters dynamically by any method, that is by changing move dynamics (MCD or MCCD) or by starting a new instruction before the last one has completed, be aware of the risk of velocity and/or end position overshoot.

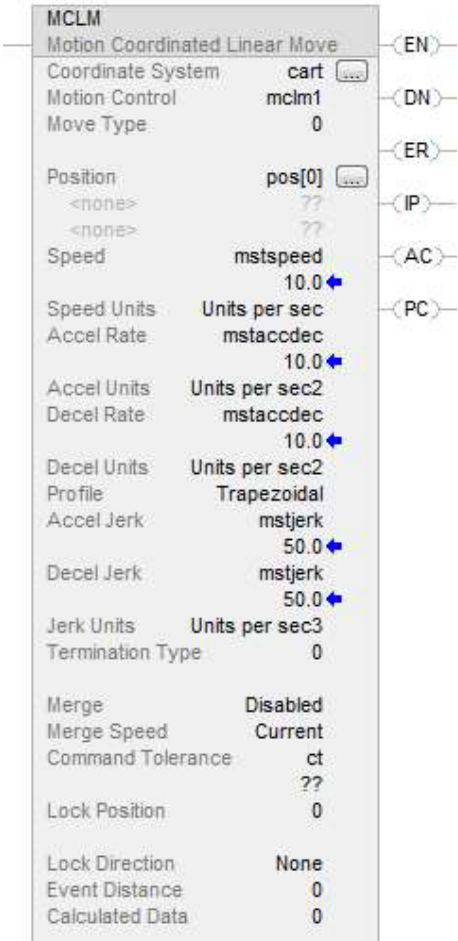
A Trapezoidal velocity profile can overshoot if maximum deceleration is decreased while the move is decelerating or is close to the deceleration point.

An S-curve velocity profile can overshoot if:

- maximum deceleration is decreased while the move is decelerating or close to the deceleration point; or
- maximum acceleration jerk is decreased and the axis is accelerating. Keep in mind, however, that jerk can be changed indirectly if it is specified in % of time.

Available Languages

Ladder Diagram



Search

- Quick Start Steps
- Logix Designer
- Module Information
- Instruction Set
 - Logix 5000 Controllers Instruction and Application Considerations
 - Logix Designer Application Instruction Set
 - Interpret the Attribute Tables
 - Array Concepts
- CIP Axis Attributes
- Module Configuration
 - Attributes
 - Bit Addressing
 - Common Attributes
 - Data Conversions
 - Elementary data types
 - LINT data types
 - Floating Point Values
 - Immediate values
 - Index Through Arrays
 - Math Status Flags
 - Motion Error Codes (.ERR)
 - Structures
- Equipment Sequence instructions
- Equipment Phase Instructions
- Alarm Instructions
- Advanced Math Instructions
- Array (File)/Misc Instructions
- Array (File)/Shift Instructions
- ASCII Conversion Instructions
- ASCII Serial Port Instructions
- ASCII String Instructions
- Bit Instructions
- Compare Instructions
- Debug Instructions
- Drives Instructions
- Drive Safety Instructions
- For/Break Instructions
- Filter Instructions
- Function Block Attributes
- Structured Text Attributes
- Compute/Math Instructions
- Move/Logical Instructions
- Input/Output Instructions
- License Instructions
- Math Conversion Instructions
- Metal Form Instructions
- Motion Configuration



Function Block

This instruction is not available in function block.

Structured Text

MCLM (CoordinateSystem, MotionControl, MoveType, Position, Speed, Speedunits, Accelrate, Accelunits, Decelrate, Decelunits, Profile, Acceljerk, Deceljerk, Jerkunits, TerminationType, Merge, Mergespeed, CommandTolerance,LockPosition, LockDirection, EventDistance, CalculatedData);

Operands

There are data conversion rules for mixed data types within an instruction. See Data Conversion.

Ladder Diagram and Structured Text

The Motion Coordinated Linear Move (MCLM) instruction performs a linear move using up to three (3) axes statically coupled as primary axes in a Cartesian coordinate system. You specify whether to use an absolute or incremental target position, the desired speed, maximum acceleration, maximum deceleration, acceleration jerk, deceleration jerk, and the units of each. The actual speed is a function of the programmed units of the speed (Units per sec, or % of Maximum, as configured for the coordinate system), and the combination of primary axes that are commanded to move. Each axis is commanded to move at a speed that allows all axes to reach the programmed endpoint (target position) at the same time.

Operand	Type	Format	Description
Coordinate System	COORDINATE_SYSTEM	Tag	Coordinated group of axes.
Motion Control	MOTION_INSTRUCTION	Tag	Structure used to access instruction status parameters.
Move Type	SINT, INT, or DINT	Immediate or Tag	Select the Move Type: 0 = Absolute 1 = Incremental
Position	REAL	Array tag []	[coordinate units]
Speed	SINT, INT, DINT, or REAL	Immediate or Tag	[coordinate units]
Speed Units	SINT, INT, or DINT	Immediate	0 = Units per Sec 1 = % of Maximum 4 = Units per MasterUnit
Accel Rate	SINT, INT, DINT, or REAL	Immediate or Tag	[coordinate units]
Accel Units	SINT, INT, or DINT	Immediate	0 = Units per Sec ² 1 = % of Maximum 4 = Units per MasterUnit ²

- ✓ [Motion Configuration Instructions](#)
- ▷ [Motion Event Instructions](#)
- ▷ [Motion Group Instructions](#)
- ▷ [Motion Move Instructions](#)
- ▷ [Motion State Instructions](#)
- ▲ [Multi-Axis Coordinated Motion Instructions](#)

[Master Driven Coordinated Control \(MDCC\)](#)
[Motion Calculate Transform Position \(MCTP\)](#)
[Motion Coordinated Change Dynamics \(MCCD\)](#)
[Motion Coordinated Circular Move \(MCCM\)](#)
[Motion Coordinated Transform with Orientation \(MCTO\)](#)
[Motion Coordinated Path Move \(MCPM\)](#)
[Motion Calculate Transform Position with Orientation \(MCTPO\)](#)
[Motion Coordinated Linear Move \(MCLM\)](#)
[Motion Coordinated Shutdown \(MCSD\)](#)
[Motion Coordinated Shutdown Reset \(MCSR\)](#)
[Motion Coordinated Stop \(MCS\)](#)
[Motion Coordinated Transform \(MCT\)](#)
[Speed, acceleration, deceleration, and jerk enumerations for coordinated motion](#)
[Status Bits for Motion Instructions \(MCLM, MCCM\) when MDCC Is Active](#)
[Change between master driven and time driven modes for Coordinated Motion instructions](#)
[Choose a Termination Type](#)
[Common Action Table for Slave Coordinate System and Master Axis](#)
[Input and Output Parameters Structure for Coordinate System Motion Instructions](#)
[Returned Calculated Data Parameter for Coordinated System Motion Instruction](#)

Decel Rate	SINT, INT, DINT, or REAL	Immediate or Tag	[coordinate units]
Decel Units	SINT, INT, or DINT	Immediate	0 = Units per Sec ² 1 = % of Maximum 4 = Units per MasterUnit ²
Profile	SINT, INT, or DINT	Immediate	0 = Trapezoidal 1 = S-curve
Accel Jerk	SINT, INT, DINT, or REAL	Immediate or Tag	You must always enter values for the Accel and Decel Jerk operands. This instruction only uses the values if the Profile operand is configured as S-curve. Enter the jerk rates in these Jerk Units. 0 = Units per sec ³ 1 = % of Maximum 2 = % of Time 4 = Units per MasterUnit ³ 6 = % of Time-Master Driven Use these values to get started. Accel Jerk = 100 (% of Time) Decel Jerk = 100 (% of Time) Jerk Units = 2
Decel Jerk	SINT, INT, DINT, or REAL	Immediate or Tag	
Jerk Units	SINT, INT, or DINT	Immediate	
Termination Type	SINT, INT, or DINT	Immediate or Tag	0 = Actual Tolerance 1 = No Settle 2 = Command Tolerance 3 = No Decel 4 = Follow Contour Velocity Constrained 5 = Follow Contour Velocity Unconstrained 6 = Command Tolerance Programmed See Choose a Termination Type in the Related Topics section below.
Merge	SINT, INT, or DINT	Immediate	0 = Disabled 1 = Coordinated Motion 2 = All Motion
Merge Speed	SINT, INT, or DINT	Immediate	0 = Programmed

- ▷ [Logical and Move Instructions](#)
- ▷ [Program Control Instructions](#)
- ▷ [Sequencer Instructions](#)
- ▷ [Special Instructions](#)
- ▷ [Timer and Counter Instructions](#)
- ▷ [Trigonometric Instructions](#)
- ▷ [Process Control Instructions](#)
- ▷ [Select/Limit Instructions](#)
- ▷ [Sequential Function Chart \(SFC\) Instructions](#)
- ▷ [Statistical Instructions](#)
- ▷ [Safety Instructions](#)
- ▷ [Studio 5000 Logix Designer Glossary](#)

			1 = Current
Command Tolerance	REAL	Immediate Real or Tag	<p>The position on a coordinated move where blending should start. This parameter is used in place of Command Tolerance in the Coordinate System if Termination Type 6 is used.</p> <p>Tip: Termination type 2 is identical to Termination Type 6 except the Command Tolerance value from the coordinate system is used and this parameter is ignored.</p>
Lock Position	REAL	Tag	<p>Position on the Master Axis where a Slave should start following the master after the move has been initiated on the Slave Axis.</p> <p>See the Structure section below for more information.</p>
Lock Direction	UINT32	Immediate	<p>Specifies the conditions when the Lock Position should be used.</p> <p>See the Structure section below for more information.</p>
Event Distance	REAL ARRAY or 0	Array Tag	<p>The position(s) on a move measured from the end of the move.</p> <p>See the Structure section below for more information.</p>
Calculated Data	REAL ARRAY or 0	Array Tag	<p>Master Distance(s) (or time) needed from the beginning of the move to the Event Distance point.</p> <p>See the Structure section below for more information.</p>

When you enter enumerations for the operand value in structured text, multiple word enumerations must be entered without spaces. For example: when entering Decel Units the value should be entered as unitspersec² rather than Units per Sec² as displayed in the ladder logic.

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

Use the entries in this table as a guide when entering structured text operands.

This Operand	Has These Options Which You	
	Enter as Text	Or as
Coordinate System	No enumeration	Tag
Motion	No enumeration	Tag

Control		Tag
Move Type	No enumeration	0 (Absolute) 1 (Incremental)
Position	No enumeration	Array Tag
Speed	No enumeration	Immediate or Tag
Speed Units	unitspersec %ofmaximum unitspermasterunits	0 1 4
Accel Rate	No enumeration	Immediate or Tag
Accel Units	unitspersec ² %ofmaximum unitspermasterunit ²	0 1 4
Decel Rate	No enumeration	Immediate or Tag
Decel Unit	unitspersec ² %ofmaximum unitspermasterunit ²	0 1 4
Profile	trapezoidal s-curve	0 1
Accel Jerk	No enumeration	Immediate or tag You must always enter a value for the Accel and Decel Jerk operands. This instruction only uses the values if the Profile is configured as S-curve. Use these values to get started. Accel Jerk = 100 (% of Time) Decel Jerk = 100 (% of Time) Jerk Units = 2
Decel Jerk	No enumeration	
Jerk Units	unitspersec ³ %ofmaximum %oftime unitspermasternit3 %oftimemasterdriven	0 1 2 (use this value to get started) 4 6

Termination Type	No enumeration	0 = Actual Tolerance 1 = No Settle 2 = Command Tolerance 3 = No Decel 4 = Follow Contour Velocity Constrained 5 = Follow Contour Velocity Unconstrained 6 = Command Tolerance Programmed See Choose a Termination Type in the Related Topics section below.
Merge	disabled coordinatedmotion allmotion	0 1 2
Merge Speed	programmed current	0 1
Command Tolerance	No enumeration	Immediate or Tag
Lock Position	No enumeration	Immediate, Real, or Tag
Lock Direction	None immediateforwardonly immediatereverseonly positionforward positionreverse	0 1 2 3 4
Event Distance	No enumeration	Array
Calculated Data	No enumeration	Array

Coordinate System

The Coordinate System operand specifies the set of motion axes that define the dimensions of a Cartesian coordinate system. For this release the coordinate system supports up to three (3) primary axes. Only those axes configured as primary axes are included in the coordinate velocity calculations.

Motion Control

The following control bits are affected by the MCLM instruction.

Mnemonic	Description
.EN (Enable) Bit 31	The Enable bit is set when the rung transitions from false to true and resets when the rung goes from true to false.

--	--

.DN (Done) Bit 29	The Done bit sets when the coordinated instruction has been verified and queued successfully. Because it's set at the time it's queued it may appear as set when a runtime error is encountered during the verify operation after it comes out of the queue. It resets when the rung transitions from false to true.
.ER (Error) Bit 28	The Error bit is reset when the rung transitions from false to true. It is set when the coordinated move has not successfully initiated. It is also set with the Done Bit when a queued instruction encounters a runtime error.
.IP (In Process) Bit 26	The In Process bit is set when the coordinated move is successfully initiated. It is reset when there is no succeeding move and the coordinated move reaches the new position, or when there is a succeeding move and the coordinated move reaches the specifications of the termination type, or when the coordinated move is superseded by another MCLM or MCCM instruction with a merge type of Coordinated Move, or when terminated by an MCS instruction.
.AC (Active) Bit 23	When you have a coordinated move instruction queued, the Active bit lets you know which instruction is controlling the motion. It sets when the coordinated move becomes active. It is reset when the Process Complete bit is set or when the instruction is stopped.
.PC (Process Complete) Bit 27	The Process Complete bit is reset when the rung transitions from false to true. It is set when there is no succeeding move and the coordinated move reaches the new position, or when there is a succeeding move and the coordinated move reaches the specified termination type.
.ACCEL (Acceleration Bit) Bit 00	The Acceleration bit sets while the coordinated move is in the acceleration phase. It resets while the coordinated move is in the constant velocity or deceleration phase, or when coordinated motion concludes.
.DECEL (Deceleration Bit) Bit 01	The Deceleration bit sets while the coordinated move is in the deceleration phase. It resets while the coordinated move is in the constant velocity or acceleration phase, or when coordinated motion concludes.
.CalculatedDataAvailable Bit 02	The CalculatedDataAvailable bit is set in response to an Event Distance request in an instruction faceplate parameter for these instructions.

.TrackingMaster Bit 03	<p>The TrackingMaster bit is set when the acceleration is complete in MDSC Mode. This means that the Slave axis is synchronized to the Master Axis. It is reset when any of the following occurs on the slave axis:</p> <p>When the slave axis starts either to accelerate or decelerate for any reason, such as an MCD or a MAS being issued etc.</p> <p>The Slave Axis is linked to another Master axis. For this case the bit is set again in the new instruction status word when the Slave Axis starts tracking the new Master axis again.</p> <p>The slave axis is stopped. The Tracking Master is cleared as soon as the stop is initiated on the slave axis.</p> <p>This bit is never set when Lock Direction== NONE.</p> <p>Note that the Tracking Master bit on the slave is not affected by any operation (i.e. stop, MCD etc on the) master axis.</p> <p>The Tracking Master bit will always be reset in Time Driven Mode.</p>
------------------------	--

Move Type

The Move Type operand specifies the method used to indicate the coordinated move path. The Move Type can be either Absolute or Incremental.

- Absolute - the axes move via a linear path to the position defined by the position array at the Speed, Accel Rate and Decel Rate as specified by the operands.

When the axis is configured for rotary operation, an Absolute Move type behaves in the same manner as for a linear axis. When the axis position exceeds the Unwind parameter, it is unwound. In this way, axis position is never greater than the Unwind value nor less than zero.

The sign of the specified position is interpreted by the interpolator and can be either positive or negative. Negative position values instruct the interpolator to move the rotary axis in a negative direction to obtain the desired absolute position, while positive values indicate that positive motion is desired to reach the target position. When the position value is greater than the unwind value, an error is generated. The axis never moves through more than one unwind cycle before stopping at an absolute position.

- Incremental - the coordinate system moves the distance as defined by the position array at the specified Speed, using the Accel and Decel rates determined by the respective operands, via a linear path.

The specified distance is interpreted by the interpolator and can be positive or negative. Negative position values instruct the interpolator to move the axis in a negative direction, while positive values indicate positive motion is desired to reach the target position. Motion greater than one unwind cycle is allowed in Incremental mode.

Position

A one dimensional array, whose dimension is defined to be at least the equivalent of the number of axes specified in the coordinate system. The Position array defines either the new absolute or incremental position.

Speed

The Speed operand defines the maximum vector speed along the path of the coordinated move.

Speed Units

The Speed Units operand defines the units applied to the Speed operand either directly in coordinate units of the specified coordinate system or as a percentage of the maximum values defined in the coordinate system.

Accel Rate

The Accel Rate operand defines the maximum acceleration along the path of the coordinated move.

Accel Units

The Accel Units operand defines the units applied to the Accel Rate operand either directly in coordinate units of the specified coordinate system or as a percentage of the maximum values defined in the coordinate system.

Decel Rate

The Decel Rate operand defines the maximum deceleration along the path of the coordinated move.

Decel Units

The Decel Units operand defines the units applied to the Decel Rate operand either directly in coordinate units of the specified coordinate system or as a percentage of the maximum values defined in the coordinate system.

Profile

The Profile operand determines whether the coordinated move uses a trapezoidal or S-curve velocity profile.

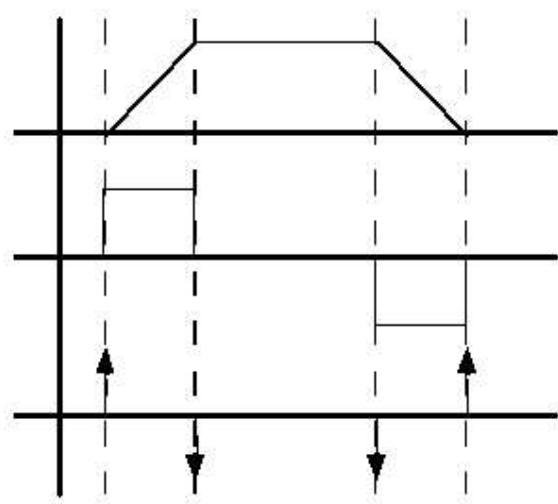
The ControlLogix motion controller provides trapezoidal (linear acceleration and deceleration), and S-curve (controlled jerk) velocity profiles. A guide to the effects of these motion profiles on various application requirements is given below.

Velocity Profile Effects

Profile	ACC/DEC	Motor	Priority of control			
			Highest to Lowest			
Trapezoidal	Fastest	Worse	Acc/Dec	Velocity	Position	
S-Curve	2X Slower	Best	Jerk	Acc/Dec	Velocity	Position

Trapezoidal

The trapezoidal velocity profile is the most commonly used profile since it provides the most flexibility in programming subsequent motion and the fastest acceleration and deceleration times. The maximum change in velocity is specified by acceleration and deceleration. Since jerk is not a factor for trapezoidal profiles, it is considered infinite and is shown as series of vertical lines in the following graph.



S-curve

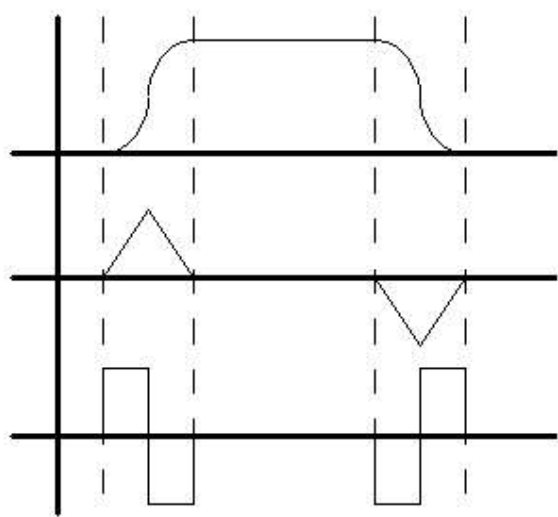
S-curve velocity profiles are most often used when the stress on the mechanical system and load needs to be minimized. The S-curve profile, however, sacrifices acceleration and deceleration time compared to the trapezoidal. The maximum rate at which velocity can accelerate or decelerate is further limited by jerk.

Coordinate motion acceleration and deceleration jerk rate calculations are performed when these instructions are started.

- MAJ
 - MAM
 - MAS
 - MCD
- MCS
 - MCCD
 - MCCM
 - MCLM

The calculated Jerk Rate produces triangular acceleration and deceleration profiles, as shown in the following diagram.

S-curve Accel/Decel Time



For an S-curve move, the Jerk rate is determined based on the programmed velocity, acceleration, and deceleration values, not on the length of the move. Logix Designer software attempts to keep the Jerk rate constant when blending moves that have the same acceleration and deceleration values, even though the move may not be long enough to reach the programmed velocity (velocity-limited move).

If an S-curve move is configured as	Then increasing the velocity
Not velocity-limited	Decreases the execution time of the move
Velocity-limited	Increases the execution time of the move

For S-curve moves that are programmed with a zero speed, the Jerk Rate is determined by the rate of speed programmed for the previous instruction with a non-zero speed.

See the MCCD instruction for more details about the impact changes made by an MCCD instruction.

Accel Jerk

Accel Jerk defines the maximum acceleration jerk for the programmed move. For more information on calculating Accel Jerk, refer to the Jerk Units section below.

Decel Jerk

Decel Jerk defines the maximum deceleration jerk for the programmed move. For more information on calculating Decel Jerk, refer to the Jerk Units section below.

Jerk Units

The jerk units define the units that are applied to the values entered in the Accel Jerk and Decel Jerk operands. The values are entered directly in the position units of the specified coordinate system or as a percentage. When configured using % of Maximum, the jerk is applied as a percentage of the Maximum Acceleration Jerk and Maximum Deceleration Jerk operands specified in the coordinate system attributes. When configured using % of Time, the value is a percentage based on the Speed, Accel Rate, and Decel Rate specified in the instruction.

Termination Type

For Master Driven Speed Control (MDSC), when all sequential instructions run in the same mode (Master Driven Mode or Time Driven Mode), then all termination types are supported. If the termination type switches in the coordinated motion queue, errors may generate depending on the sequence of motion types.

The following is only applicable if a move on a slave Coordinate System uses a Blending Termination Type (Termination Types 2, 3, or 6) and is programmed in MDSC mode.

If you use the Calculated Data returned in the last MCLM instruction of a motion sequence to program the length that the master axis has to move for the motion sequence to go PC, then there is the possibility that you will have to add a small safety margin to the Calculated Data. If you do not add this margin, there is a chance that the motion of the master axes completes before the entire motion sequence programmed on the Slave Coordinate System finishes. If this occurs, the last motion instruction on the Slave Coordinate System remains active and does not go PC. The value of the small safety margin is dependent on the Command Tolerance used for the first and last move in the motion sequence as follows:

Tip: Safety margin calculations use the following abbreviations:

CUP = Coarse Update Period

MAS = Master Axis Speed

1. If a Command Tolerance value of 100% is used for the first move in the sequence then:
 $\text{SafetyMargin1} = \text{CUP} * \text{MAS}$

else

 $\text{SafetyMargin1} = \text{CUP} * \text{MAS} * .02$
2. For all other moves in the blending sequence between first and last:
 $\text{SafetyMargin2} = \text{CUP} * \text{MAS} * .02 * \text{number of blending moves between 1st and last}$
3. If a Command Tolerance value of 100% is used for the last in the sequence then:
 $\text{SafetyMargin3} = \text{CUP} * \text{MAS}$

else

$$\text{SafetyMargin3} = \text{CUP} * \text{MAS} * .02$$

4. Final SafetyMargin = SafetyMargin1 + SafetyMargin2 + SafetyMargin3
Once a sequence is programmed and verified, it will repeat.

Refer to the See also section for more information.

Merge

The Merge operand determines whether or not to turn the motion of all specified axes into a pure coordinated move. The Merge options include: Merge Disabled, Coordinated Motion, or All Motion.

- Merge Disabled
Any currently executing single axis motion instructions involving any axes defined in the specified coordinate system are not affected by the activation of this instruction, and results in superimposed motion on the affected axes. Also, any coordinated motion instructions involving the same specified coordinate system runs to completion based on its termination type.
- Coordinated Motion
Any currently executing coordinated motion instructions involving the same specified coordinate system are terminated. The active motion is blended into the current move at the speed defined in the merge speed parameter. Any pending coordinated motion instructions are cancelled. Any currently executing system single axis motion instructions involving any axes defined in the specified coordinate system will not be affected by the activation of this instruction, and will result in superimposed motion on the affected axes.
- All Motion
Any currently executing single axis motion instructions involving any axes defined in the specified coordinate system and any currently executing coordinated motion instructions are terminated. The prior motion is merged into the current move at the speed defined in Merge Speed parameter. Any pending coordinated move instructions are cancelled.

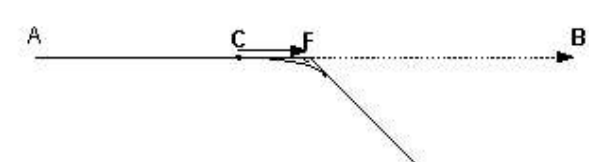
Merge Speed

The Merge Speed operand defines whether the current speed or the programmed speed is used as the maximum speed along the path of the coordinated move when Merge is enabled.

Currently, Coordinated Motion only supports the queueing of one coordinated motion instruction. Therefore the MovePendingStatus bit and the MovePendingQueueFullStatus bit are always the same.

Additional Information On Merging Instructions

A move from point A to point B is initiated as shown in the figure below. When the axis is at point C, a incremental merge to point D is initiated. As a result the current instruction is terminated at point C. The control computes the deceleration distance needed at point C along the vector AB from the current velocity to zero velocity. This distance is shown as vector CF. The imaginary point F is then calculated by adding the vector CF to point C. The resultant merged motion from C to D is shown in the illustration below. The motion follows the curved line starting from C then joins the straight line from F to D. Point D is calculated from the original point of the merge (point C) using the incremental data in the merge instruction. This path is identical as if the original programmed move was made from point A to F then from F to D with a termination type of No Decel.





This example applies to linear merges.

Attempting to merge a circular move can result in programming errors if the resultant path does not define a circle. The circle center in incremental mode is calculated from point C (the point of the merge). However, a circle must exist from point F (the calculated end of the deceleration) to the end of the merged move.

Merging in Incremental Mode

The Merge for coordinated motion operates differently from a merge on an MAM. For the MCLM, any uncompleted motion at the point of the merge is discarded. For example, assume that you have a single axis MCLM programmed in incremental mode from a starting absolute position = 0 and with the programmed incremental distance = 4 units. If a merge occurs at an absolute position of 1 and the merge is another incremental move of 4 units, the move completes at a position = 5.

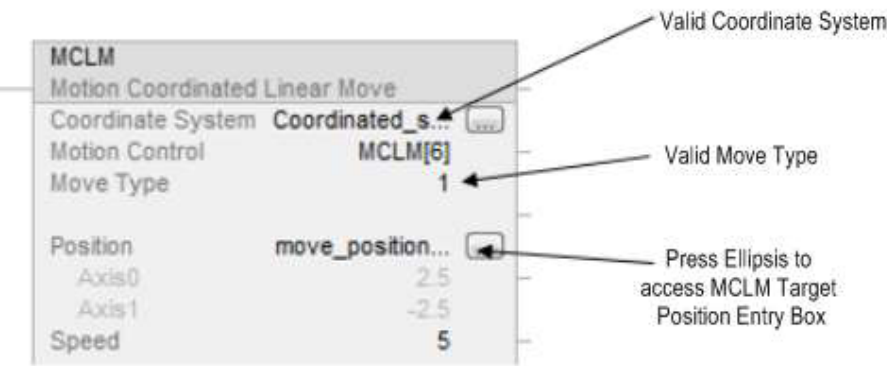
If this example occurs on a MAM programmed in incremental mode, the final position = 8.

MCLM Target Position Entry Dialog

The Target Position Entry Dialog for the MCLM instruction provides an easy format for editing Position. To gain access to the Target Position Entry dialog box you must have inserted the name of the coordinated system into the instruction, you must have a valid tag name entered in the position field with sufficient elements to handle the number of axes, and you must have selected a valid Move Type.

To access the MCLM Instruction Target Position Entry Dialog box, press the ellipsis after the Position line on the instruction faceplate.

MCLM Ladder Valid Values for Accessing Target Position Entry Box



Pressing the ellipsis button at the Position line of the ladder instruction faceplate calls the **Target Position** tab for editing the position values.

The dialog title indicates the Coordinate System and Tag Names for the instruction.

Feature	Description
Axis Name	These fields list the names of each axis contained in the Coordinate System. You cannot alter the axis names in this dialog.
Target Position/Target Increment	This field contains the endpoint or increment of the coordinated move as specified in the instruction faceplate. It is numeric.
Actual Position	These are the current actual positions of the axes in the coordinate system. These positions are updated dynamically when on-line and Coordinate System Auto Tag Update is enabled.

Set Targets = Actuals Button	<p>When in Absolute mode, this button automatically copies the actual position values to the Target Position Column.</p> <p>When in Incremental mode, this button automatically copies the actual position values to the Target Increment Column.</p>
------------------------------	---

The selected Move type governs the appearance and the availability of the Set Targets = Actuals button.

When the Move Type is Absolute, the target column is entitled Target Position and when the Move Type is Incremental, the target column is entitled Target Increment and the Set Targets = Actuals button appears dimmed.

MCLM is a transitional instruction:

- In relay ladder, toggle the Rung-condition-in from false to true each time the instruction should execute.
- In structured text, condition the instruction so that it only executes on a transition. See Structured Text Syntax.

Structure

See *Input and Output Parameters Structure for Single Axis Motion Instructions* for the input and output parameters that are available for the MCLM instruction via the Master Driven Speed Control (MDSC) function. Before any of these parameters is active, you must execute an MDCC instruction and it must be active (IP bit is set).

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.
----------	---

Runtime Error Conditions

You cannot switch from Time Driven Mode to Master Driven Mode if the master speed is zero unless the slave speed is also zero.

Extended Error Codes

Extended Error codes help to further define the error message given for this particular instruction. Their behavior is dependent upon the Error Code with which they are associated.

The Extended Error Codes for Servo Off State (5), Shutdown State (7), Axis Type Not Servo (8), Axis Not Configured (11), Homing In Process Error (16), and Illegal Axis Data type (38) errors all function in the same fashion. A number between 0...n is displayed for the Extended Error Code. This number is the index to the Coordinate System indicating the axis that is in the error condition.

For Error Code Axis Not Configured (11) there is an additional value of -1 which indicates that Coordinate System was unable to setup the axis for coordinate motion. See Motion Error Codes (ERR) for Motion Instructions.

For the MCLM instruction, Error Code 13 - Parameter Out of Range, Extended Errors return a number that indicates the offending parameter as listed on the faceplate in numerical order from top to bottom beginning with zero. For example, 2 indicates the parameter value for Move Type is in error.

Referenced Error Code and Number	Extended Error Numeric Indicator	Instruction Parameter	Description
Parameter Out Of Range (13)	2	Move Type	Move Type is either less than 0 or greater than 1.
Parameter Out Of Range (13)	3	Position	The position array is not large enough to provide positions for all the axes in the coordinate system.
Parameter Out Of Range (13)	4	Speed	Speed is less than 0.
Parameter Out Of Range (13)	6	Accel Rate	Accel Rate is less than or equal to 0.
Parameter Out Of Range (13)	8	Decal Rate	Decel Rate is less than or equal to 0.
Parameter Out Of Range (13)	11	Termination Type	Termination Type is less than 0 or greater than 3.

For the Error Code 54 – Maximum Deceleration Value is Zero, if the Extended Error returns a positive number (0-n) it’s referring to the offending axis in the coordinate system. Go to the Coordinate System Properties General Tab and look under the Brackets ([])column of the Axis Grid to determine which axis has a Maximum Deceleration value of 0. Click on the ellipsis button next to the offending axis to access the Axis Properties screen. Go to the Dynamics tab and make the appropriate change to the Maximum Deceleration Value. If the Extended Error number is -1, this means the Coordinate System has a Maximum Deceleration Value of 0. Go to the Coordinate System Properties Dynamics Tab to correct the Maximum Deceleration value.

MCLM Changes to Status Bits

MCLM Changes to Status Bits

Status bits provide a means for monitoring the progress of the motion instruction. There are three types of Status bits that provide pertinent information.

- Axis Status bits
- Coordinate System Status bits
- Coordinate Motion Status bits

When the MCLM instruction initiates, the status bits undergo the following changes.

Motion Instruction Predefined Data Type Status Bits

See *Status Bits for Motion Instructions* (MCLM, MCCM) when MDCC Is Active

Axis Status Bits

Bit Name	Meaning
CoordinatedMotionStatus	Sets when the instruction starts. Clears when the instruction ends.

Coordinate System Status Bits

Bit Name	Meaning
MotionStatus	Sets when the MCLM instruction is active and the Coordinate System is connected to its associated axes.

Coordinated Motion Status Bits

Bit Name	Meaning
AccelStatus	Sets when vector is accelerating. Clears when a blend is in process or when vector move is decelerating.
DecelStatus	Sets when vector is decelerating. Clears when a blend is in process or when vector move is accelerating.
ActualPosToleranceStatus	Sets for Actual Tolerance termination type only. It sets after the following two conditions are met. 1) Interpolation is complete. 2) The actual distance to the programmed endpoint is less than the configured coordinate system Actual Tolerance value. The bit remains set after an instruction completes. The bit is reset when a new instruction is started.

CommandPosToleranceStatus	<p>Sets for all termination types whenever the distance to the programmed endpoint is less than the configured coordinate system Command Tolerance value. The bit remains set after an instruction completes. It resets when a new instruction is started.</p> <p>The CommandPosToleranceStatus (CS_CMD_POS_TOL_STS) status bit in the Coordinate System is set as follows:</p> <p>TT0, TT1, TT4, TT5 - Bit is set when the distance to the endpoint is less than the Command Tolerance value.</p> <p>The bit is cleared when the first move is complete.</p> <p>TT2, TT6 - Bit is set when the distance to the endpoint is less than the Command Tolerance value.</p> <p>The bit is cleared when the blend is started (that is, when the second move is started). Thus, you may not see the bit if the blend is started at the Command Tolerance (CT) point. The blend may have been deferred slightly beyond the CT point if the next move is a short move or for time matching of the acceleration and deceleration of the two adjacent moves.</p> <p>TT3 - Bit is set when the distance to the endpoint is less than the Command Tolerance value (like TT2 and TT6).</p> <p>The bit is cleared when the blend is started. Thus, you may not see the bit if the blend is started at the deceleration point. The blend may have been deferred slightly beyond the deceleration point if the next move is a short move or for time matching of the acceleration and deceleration of the two adjacent moves.</p>
StoppingStatus	<p>The Stopping Status bit is cleared when the MCLM instruction initiates.</p>
MoveStatus	<p>Sets when MCLM begins axis motion. Clears on .PC bit of the last motion instruction or when a motion instruction executes which causes a stop.</p>
MoveTransitionStatus	<p>Sets when No Decel or Command Tolerance termination type is satisfied. When blending collinear moves the bit is not set because the machine is always on path. It clears when a blend completes, the motion of a pending instruction starts, or a motion instruction executes which causes a stop. Indicates not on path.</p>
MovePendingQueueFullStatus	<p>Sets when the instruction queue is full. It clears when the queue has room for a new coordinated motion instruction.</p>

CoordinateSystemLockStatus	<p>Set when an axis lock is requested for an MCLM or MCCM instruction and the axis has crossed the Lock Position. Cleared when an MCLM or MCCM is initiated.</p> <p>For the enumerations Immediate Forward Only and Immediate Reverse Only, the bit is set immediately when the MCLM or MCCM is initiated.</p> <p>When the enumeration is Position Forward Only or Position Reverse Only, the bit is set when the Master Axis crosses the Lock Position in the specified direction. The bit is never set if the enumeration is NONE.</p> <p>The CoordMotionLockStatus bit is cleared when the Master Axis reverses direction and the Slave Axis stops following the Master Axis. The CoordMotionLockStatus bit is set again when the Slave Coordinate System resumes following the Master Axis. The CoordMotionLockStatus bit is also cleared when an MCCS is initiated.</p>
----------------------------	--

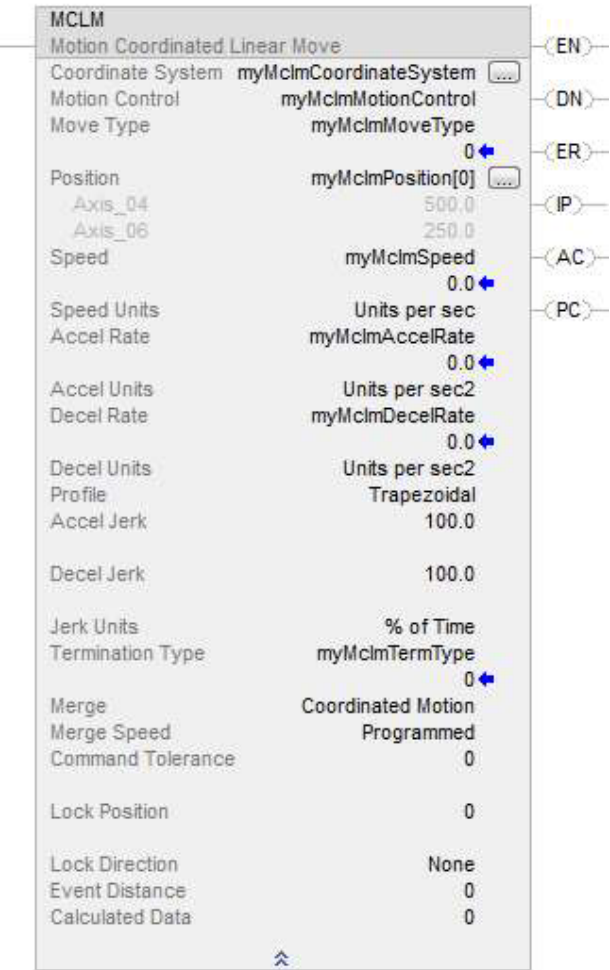
Currently, Coordinated Motion only supports the queueing of one coordinated motion instruction. Therefore the MovePendingStatus bit and the MovePendingQueueFullStatus bit are always the same.

Master Driven Speed Control (MDSC) and Motion Direct Command Support

The Motion Direct commands are not available in the instruction tree for the MCLM instruction. You must program an MCLM in one of the supported programming languages before you execute an MAM or MAJ in Time Driven Mode. A runtime error will occur if an MCLM is not previously executed in an MAM and MAJ in Master Driven Mode.

Examples

Ladder Diagram



Structured Text

MCLM(myMclmCoordinateSystem, myMclmMotionControl, myMclmMoveType, myMclmPosition[0], myMclmSpeed, Unitspersec, myMclmAccelRate, Unitspersec2, myMclmDecelRate, Unitspersec2, Trapezoidal, 100.0, 100.0, %ofTime, myMclmTermType, CoordinatedMotion, Programmed, 0, 0, None, 0, 0);

See also

[Structured Text Syntax](#)

[Motion Error Codes \(.ERR\)](#)

[Multi-Axis Coordinated Motion Instructions](#)

[Common Attributes](#)

[Data Conversions](#)