Motion Coordinated Circular Move (MCCM)

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 MCCM instruction to initiate a two or three-dimensional circular coordinated move for the specified axes within a Cartesian coordinate system. New position is defined as either an absolute or incremental position and done at the desired speed. The actual speed of the MCCM is a function of the mode of the move (commanded speed or percent of maximum speed). The speed of the move is based on the time it takes to complete the circular move using the programmed axes. Each axis is commanded to move at a speed that allows for all axes to reach the endpoint (target position) at the same time.

The dimension of the circle is defined by the number of axes contained within the coordinate system. For example, if you have a coordinate system that contained three axes with an MCCM instruction that has motion in only two dimensions, the resultant move is still considered a three-dimensional arc or circle.

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.

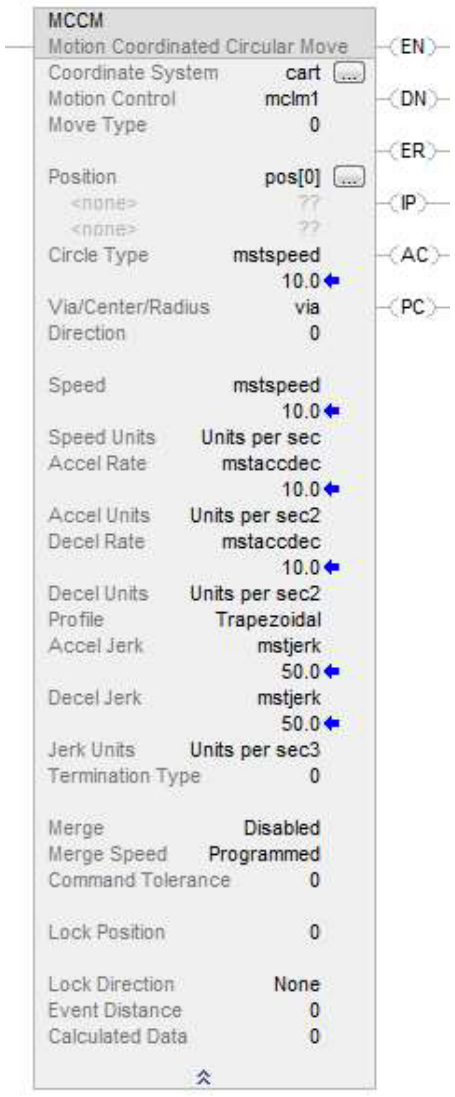
An S-curve velocity profile can overshoot if:

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



Function Block

This instruction is not available in function block.

Structured Text

MCCM (CoordinateSystem, MotionControl, MoveType, Position, CircleType, Via/Center/Radius, Direction, 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

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]

Circle Type	SINT, INT, or DINT	Immediate or Tag	0 = Via 1 = Center 2 = Radius 3 = Center Incremental
Via/Center/Radius	REAL	array tag[] (via/center) Immediate or tag (radius)	[coordinate units]
Direction	SINT, INT, or DINT	Immediate or Tag	**2D** **3D** 0 = CW Shortest 1 = CW Longest 2 = CW Full Shortest Full 3 = CCW Full Longest Full
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 $MasterUnit^2$
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

- 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

Decel Jerk	SINT, INT, DINT, or REAL	Immediate or Tag	instruction only uses the values if the Profile is configured as S-curve. Accel Jerk is the acceleration jerk rate for the coordinate system. Decel Jerk is the deceleration jerk rate for the coordinate system. 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
Jerk Units	SINT, INT, or DINT	Immediate or Tag	
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 1 = Current

Command Tolerance	REAL	Immediate Real or Tag	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. **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.
Lock Position	REAL	Immediate or Tag	Position on the Master Axis where a Slave should start following the master after the move has been initiated on the Slave Axis. See the Structure section below for more information.
Lock Direction	UINT32	Immediate Real or Tag	Specifies the conditions when the Lock Position should be used. See the Structure section below for more information.
Event Distance	REAL ARRAY or 0	Array	The position(s) on a move measured from the end of the move. See the Structure section below for more information.
Calculated Data	REAL ARRAY or 0	Array	Master Distance(s) (or time) needed from the beginning of the move to the Event Distance point. See the Structure section below for more information.

Structured Text

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

When entering enumerations for the operand value in structured text, multiple word enumerations must be entered without spaces. For example: when entering Decel Units the value is entered as unitspersec^2 rather than Units per Sec^2 as displayed in the ladder logic.

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 Control	No enumeration	Tag
Move Type	No enumeration	0 (Absolute) 1 (Incremental)
Position	No enumeration	Array Tag
Circle Type	No enumeration	Tag 0 = Via 1 = Center 2 = Radius 3 = Center Incremental
Via/Center/Radius	No enumeration	Array Tag (via/center) Immediate or tag (radius)
Direction	No enumeration	**2D** **3D** 0=CW Shortest 1=CW Longest 2=CW Full Shortest Full
Speed	No enumeration	Immediate or Tag
Speed Units	unitspersec %ofmaximum unitspermasterunit	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 Units	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 unitspermasterunit3 %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 system 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 the axes configured as primary axes (up to 3) are included in speed calculations. Only primary axes participate in the actual circular move.

Motion Control

The following control bits are affected by the MCCM 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 resets when the rung transitions from false to true. It sets when the coordinated move fails to initiate successfully. It can also be set with the Done bit when a queued instruction encounters a runtime error.
.IP (In Process) Bit 26	The In Process bit sets when the coordinated move is successfully initiated. It resets when there is a succeeding move and the coordinated move reaches the new position, or when there is no succeeding move and the coordinated move reaches the termination type specifications, or when the coordinated move is superseded by another MCCM or MCLM instruction with a Merge Type of Coordinated Move or when terminated by an MCS or an MCSD 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 01	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 02	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.

Move Type

The Move Type operand determines the method used by the position array to indicate the path of the coordinated move and the method the via/center/radius parameter uses to indicate the via and center circle positions. The options are: Absolute or Incremental.

- Absolute - the coordinate system moves to the specified Position at the defined Speed, using the Accel and Decel Rates as designated by their respective operands, along a circular path.
  When an axis is configured for rotary operation, absolute moves are handled in the same manner as with linear axes. When the axis position exceeds the Unwind parameter, an error is generated.
- The sign of the specified position array is interpreted by the controller as the direction for the move. Negative position values instruct the interpolator to move the rotary axis in a negative direction to obtain the desired absolute position. A positive value indicates that positive motion is desired to reach the target position. To move to the unwind position in the negative direction a negative unwind position value must be used as 0 and -0 are treated as 0. When the position is greater than the unwind value, an error is generated. The axis can move through the unwind position but never incrementally more than one unwind value.
- 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, along a circular path.

The specified distance is interpreted by the interpolator and can be positive or negative. Negative position values instruct the interpolator to move the rotary axis in a negative direction, while positive values indicate positive motion is desired to reach the target position.

Position

The Position operand is a one dimensional array whose dimension is at least equivalent to the number of axes specified in the coordinate system. It is the position array that defines the new absolute or incremental position.

Circle Type

The Circle Type operand specifies how the array labeled via/center/radius is interpreted. The options are: Via, Circle, Radius, Center Incremental.

- Via indicates that the via/center/radius array members specify a via point between the start and end points.
- Center indicates that the via/center/radius array members contain the circle center.
- Radius indicates that the first via/center/radius array member contains the radius. Other members are ignored. Radius is valid only in two-dimensional coordinate systems.
- Center Incremental indicates that the via/center/radius array members define a position that always incrementally defines the center of the circle regardless of Move Type operand. Sign of the incremental value is measured from the start point to the center.

Via/Center/Radius

Depending on the selected Move Type and Circle Type, the via/center/radius position parameter defines the absolute or incremental value of a position along the circle, the center of the circle, or the radius of the circle as defined in the following table. If the Circle Type is via or center, the via/center/radius position parameter is a one-dimensional array,

whose dimension is defined to be at least the equivalent of the number of axes specified in the coordinate system. If the Circle type is radius, the via/center/radius position parameter is a single value.

Move Type	Cycle Type	Behavior
Absolute	Via	The via/center/radius position array defines a position along the circle. For a non-full circle case, the Position parameter array defines the endpoint of the arc. For a full circle case, the Position parameter array defines any second point along the circle except the endpoint.
Incremental	Via	The sum of the via/center/radius position array and the old position defines the position along the circle. For a non-full circle case, the sum of the Position parameter array and the old position defines the endpoint of the arc. For a full circle case, the sum of the Position parameter array and the old position defines any second point along the circle except the endpoint.
Absolute	Center	The via/center/radius position array defines the center of the circle. For a non-full circle case, the Position parameter array defines the endpoint of the arc. For a full circle case, the Position parameter array defines any second point along the circle except the endpoint.
Incremental	Center	The sum of the via/center/radius position array and the old position defines the center of the circle. For a non-full circle case, the sum of the Position parameter array and the old position defines the endpoint of the arc. For a full circle case, the sum of the Position parameter array and the old position defines any second point along the circle except the endpoint.
Absolute or Incremental	Radius	The via/center/radius position single value defines the arc radius. The sign of the value is used to determine the center point to distinguish between the two possible arcs. A positive value indicates a center point that generates an arc less than 180 degrees. A negative value indicates a center point that generates an arc greater than 180 degrees. This Circle Type is only valid for two-dimensional circles. The position parameter array follows the Move Type to define the endpoint of the arc.
Absolute	Center Incremental	The sum of the via/center/radius position array and the old position defines the center position of the circle. For a non-full circle case, the Position parameter array defines the endpoint of the arc. For a full circle case, the Position parameter array defines any second point along the circle except the endpoint.
Incremental	Center Incremental	The sum of the via/center/radius position array and the old position defines the center position of the circle. For a non-full circle case, the sum of the Position parameter array and the old position defines the endpoint of the arc. For a full circle case, the sum of the Position parameter array and the old position defines any second point along the circle except the endpoint.

Direction

The Direction operand defines the rotational direction of a 2D circular move as either clockwise or counterclockwise according to the right-hand screw rule. For a 3D circular move the direction is either Shortest or Longest. In both 2D and 3D it can also indicate if the

circular move is to be a full circle.

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 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 an S-curve velocity profile.

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.

If you want to convert engineering units to % of Time, use these equations.

For Accel Jerk:

$$j_a\,[\mathrm{EU/s^3}] = \frac{{a_{max}}^2\,[\mathrm{EU/s^2}]}{v_{max}\,[\mathrm{EU/s}]}\left(\frac{200}{j_a\,[\%\ \mathrm{of\ time}]} - 1\right)$$

For Decel Jerk:

$$j_a\,[\mathrm{EU/s^3}] = \frac{{a_{max}}^2\,[\mathrm{EU/s^2}]}{v_{max}\,[\mathrm{EU/s}]}\left(\frac{200}{j_a\,[\%\ \mathrm{of\ time}]} - 1\right)$$

If you want to convert % of Time to engineering units, use these equations.

For Accel Jerk:

$$j_a\,[\%\ \mathrm{of\ time}] = \frac{2}{1 + \frac{j_a\,[\mathrm{EU/s^3}]\ v_{max}\,[\mathrm{EU/s}]}{a_{max}^2\,[\mathrm{EU/s^2}]}}\,100$$

For Decel Jerk:

$$j_d\,[\%\ \mathrm{of\ time}] = \frac{2}{1 + \frac{j_d\,[\mathrm{EU/s^3}]\ v_{max}\,[\mathrm{EU/s}]}{d_{max}^2\,[\mathrm{EU/s^2}]}}\,100$$

Important Consideration

If you program tangent circles with different Jerk rates (Decel Jerk of first circle and Accel Jerk of the second circle), then you might get a slight velocity discontinuity at the intersection of the two circles. The size of the discontinuity depends on magnitude of the Jerk difference. In other words, the smaller the Jerk difference, the smaller the velocity glitch. Therefore, we recommend that you do not program Jerk rates on tangent circles.

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 MCCM 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:

CUP = Coarse Update Period

MAS = Master Axis Speed

- If a Command Tolerance value of 100% is used for the first move in the sequence then:

SafetyMargin1 = CUP * MAS

else

SafetyMargin1 = CUP * MAS * .02

- For all other moves in the blending sequence between first and last:

SafetyMargin2 = CUP * MAS * .02 * number of blending moves between 1st and last

- If a Command Tolerance value of 100% is used for the last in the sequence then:

SafetyMargin3 = CUP * MAS

else

SafetyMargin3 = CUP * MAS * .02

- Final SafetyMargin = SafetyMargin1 + SafetyMargin2 + SafetyMargin3

Once a sequence is programmed and verified, it will repeat.

Click the link Choose a Termination Type in the Related Topics section below for information.

Merge

The merge defines whether or not to turn the motion of all specified axes into a pure coordinated move. The options are: 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 result in superimposed motion on the affected axes. An error is flagged if a second instruction is initiated in the same coordinate system or in another coordinate system containing any axes in common with the coordinate system that is active.
- Coordinated Motion - Any currently executing coordinated motion instructions involving the same specified coordinate system are terminated, and the active motion is blended into the current move at the speed defined in the merge speed parameter. Any pending coordinated motion instructions in the specified coordinate system are canceled. Any currently executing system single axis motion instructions involving any axes defined in the specified coordinate system are not affected by the activation of this instruction, and 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 canceled.

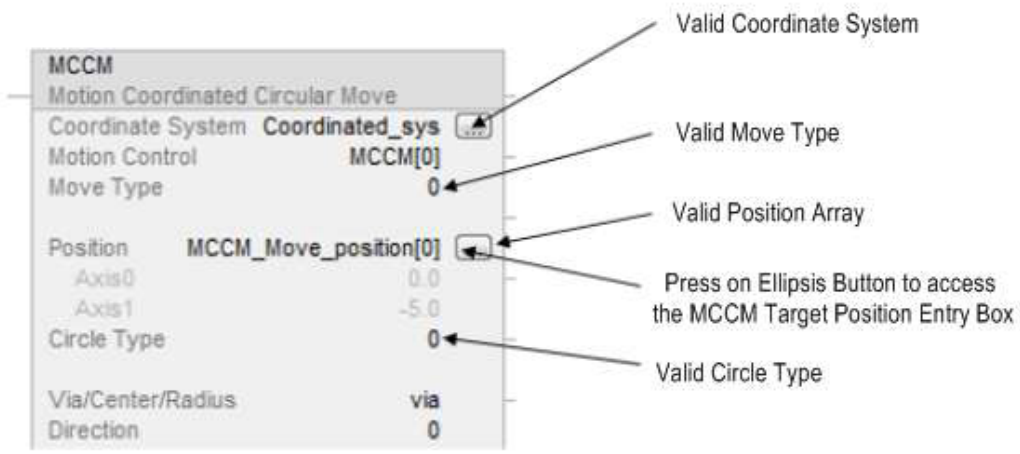
Merge Speed

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. Current speed is the vector sum of all motion (for example, jogs, MAM's, and geared motion) for all axes defined in the current coordinate system.

MCCM Target Position Entry Dialog Box

The MCCM Target Position Entry Dialog box is accessed by pressing the ellipsis button to the right of the position operand of the ladder instruction faceplate. The Target Position Entry box can only be accessed if the coordinate system for the instruction has been named, has a valid tag name for the Position operand that contains enough elements to accommodate the number of axes, selected a valid Move Type and a valid Circle Type. If these criteria have not been satisfied, an error message is displayed on the status bar.

MCCM Ladder Valid Values for Accessing Target Position Entry Box



Press the ellipsis and the **Target Position Entry Position** tab dialog box displays.

Target Position Entry Dialog Box Fields

Feature	Description
Axis Name	This column has the names of each axis in the coordinate system named in the ladder faceplate. These names are not editable.
Target Position/Target Increment	The values in this column are numeric. They show the endpoint or incremental departure of the move depending on the active Move Type. The column heading indicates which is displayed.
Actual Position	This column contains the current actual position of the axes in the coordinate system. These values update dynamically when on-line and the Coordinate System Auto Tag Update is enabled.
Via Position/Via Increment Center Position/Center Increment Radius	Depending on the Circle Type selected, this column contains the Via point position or increment, the Center Position or increment.
Set Targets = Actuals	This button is enabled when the Move Type is Absolute and is used to copy the value from the Actual Position fields to the Target Position fields.
Set Vias = Actuals	This button is only active if the Move Type is Absolute. It is used to copy the values from the Actual Position fields to the Vias Fields.

The Move Type and Circle Type selected govern the appearance of this dialog box. The following table illustrates how the screen is affected by the combinations of Move Type and Circle Type selected.

Target Position Entry Dialog Box Changes

Move Type	Circle Type	Behavior
Absolute	Via	Target column is entitled Target Position. Via column is entitled Via Position. Set Targets = Actuals button is active. Set Vias = Actuals button is active.

Incremental	Via	Target column is entitled Target Increment. Via Column is entitled Via Increment. Set Targets = Actuals button is inactive (appears dimmed). Set Vias = Actuals button is inactive (appears dimmed).
Absolute	Center	Target column is entitled Target Position. Center column is entitled Center Position. Set Targets = Actuals button is active. Set Vias = Actuals button is active.
Incremental	Center	Target column is entitled Target Increment. Center Column is entitled Center Increment. Set Targets = Actuals button is inactive (appears dimmed). Set Vias = Actuals button is inactive (appears dimmed).
Absolute	Radius	Target column is entitled Target Position. Radius column is entitled Radius. Set Targets = Actuals button is active. Set Vias = Actuals button is inactive (appears dimmed).
Incremental	Radius	Target column is entitled Target Increment. Radius Column is entitled Radius. Set Targets = Actuals button is inactive (appears dimmed). Set Vias = Actuals button is inactive (appears dimmed).
Absolute	Center Incremental	Target column is entitled Target Position. Center Incremental column is entitled Center Incremental. Set Targets = Actuals button is active. Set Vias = Actuals button is inactive (appears dimmed).
Incremental	Center Incremental	Target column is entitled Target Increment. Center Incremental column is entitled Center Incremental. Set Targets = Actuals button is inactive (appears dimmed). Set Vias = Actuals button is inactive (appears dimmed).

MCCM 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 MCCM 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 zero too.

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.

For the MCCM 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)	0	Coordinate System	Number of primary axes is not 2 or 3.
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	Circle Type	Circle Type is either less than 0 or greater than 4.
Parameter Out Of Range (13)	5	Via/Center/Radius	The size of the Via/Center array is not large enough to provide positions for all of the axes in the defining via/center point.
Parameter Out Of Range (13)	6	Direction	Direction is either less than 0 or greater than 3.
Parameter Out of Range (13)	7	Speed	Speed is less than 0.
Parameter Out of Range (13)	9	Accel Rate	Accel Rate is less than or equal to 0.
Parameter Out of Range (13)	11	Decal Rate	Decel Rate is less than or equal to 0.
Parameter Out of Range (13)	14	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.

MCCM 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
- Coordinate Motion

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

Axis Status Bits

Bit Name	Meaning
CoordinatedMotionStatus	Sets when the MCCM instruction executes and is cleared when the instruction completes.

Coordinate System Status Bits

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

CommandPosToleranceStatus	Sets for all termination types whenever the distance to the programmed endpoint is less than the configured coordinate system's Command Tolerance value and remains set after the instruction completes. It is reset when a new instruction is started. The CommandPosToleranceStatus (CS_CMD_POS_TOL_STS) status bit in the Coordinate System is set as follows: TT0, TT1, TT4, TT5 - Bit is set when the distance to the endpoint is less than the Command Tolerance value. The bit is cleared when the first move is complete. TT2, TT6 - Bit is set when the distance to the endpoint is less than the Command Tolerance value. 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. TT3 - Bit is set when the distance to the endpoint is less than the Command Tolerance value (like TT2 and TT6). 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.
StoppingStatus	The Stopping Status bit is cleared when the MCCM instruction executes.
MoveStatus	Sets when MCCM begins axis motion. Clears on the .PC bit of the last motion instruction or a motion instruction executes which causes a stop.
MoveTransitionStatus	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.
MovePendingQueueFullStatus	Sets when the instruction queue is full. It clears when the queue has room to hold another new coordinated move instruction.

CoorMotionLockStatus	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. For the enumerations Immediate Forward Only and Immediate Reverse Only, the bit is set immediately when the MCLM or MCCM is initiated. 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. 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.

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

Circular Programming Reference Guide

Circle Type	**Used in 2D/3D/Both**	**Validation Errors**	**Direction - 2D**	**Direction - 3D**	**Comment**
Radius	2D	Error 25; Illegal Instruction Error 45 Endpoint = Startpoint Error 49; R too small (|R| < .001) or R too short to span programmed points.	CW/CCW as viewed from the + perpendicular to the circular plane.	N/A	A + radius forces arc length to be <= 180° (Shortest arc). A - radius forces arc length to be => 180° (Longest arc). Full Circles can be programmed. For full circles: set Position to be any point on circle except Startpoint and use one of the Full direction types.

Center Point	Both	Error 44; Collinearity (3D only) Error 45; Endpoint = Startpoint (3D only) Error 46; Start/End radius mismatch (|R1 - R2| > .15 * R1).	CW/CCW as viewed from the + perpendicular to the circular plane.	Shortest/Longest arc. In Full circles, placement of endpoint defines shortest/longest paths referred to by direction parameter.	Full Circles can be programmed. In 2D only, Endpoint = Startpoint is legal. Therefore, full circles may be generated: By setting Endpoint = Startpoint, in which case, all direction types produce full circles. By setting Endpoint not = Startpoint and using Full direction type. For 3D Full Circles: set Position to be any point on the circle except Startpoint, and use one of the Full direction types. Position defines both arc and Shortest direction types.
Via Point	Both	Error 44; Collinearity Error 45; Endpoint = Startpoint	Via point always determines direction.	Via point always determines direction. Direction operand is only used to determine if circle is partial or full.	Full Circles can be programmed. For full circles: set Position to be any point on circle except Startpoint and use one of the Full direction types.

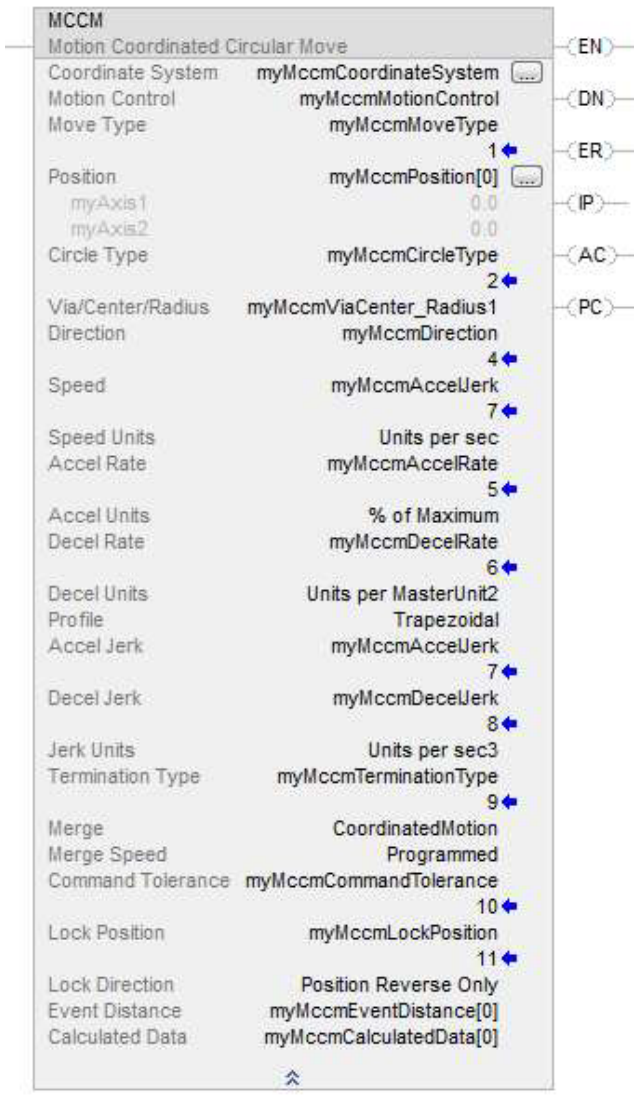
Master Driven Speed Control (MDSC) and Motion Direct Command Support

The Motion Direct commands are not available in the instruction tree for the MCCM instruction. You must program an MCCM 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

MCCM is not previously executed in an MAM and MAJ in Master Driven Mode.

Example 1

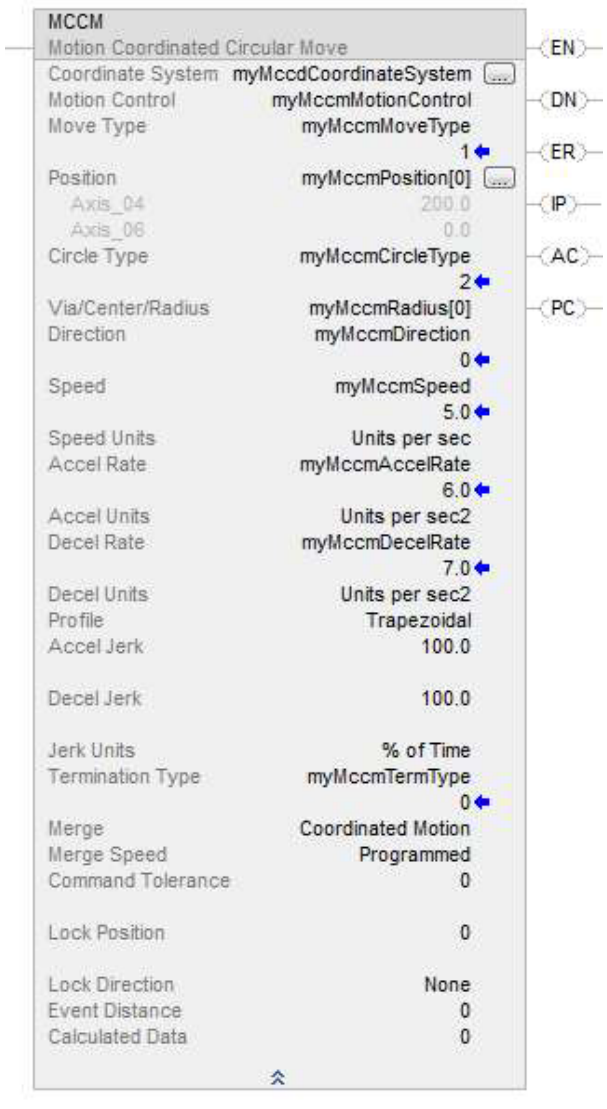
Ladder Diagram



Example 2

Via/Center/Radius parameter as an array type.

Ladder Diagram



Structured Text

MCCM(myMccmCoordinateSystem, myMccmMotionControl, myMccmMoveType, myMccmPosition[0], myMccmCircleType, myMccmRadius[0], myMccmDirection, myMccmSpeed, Unitspersec, myMccmAccelRate, Unitspersec2, myMccmDecelRate, Unitspersec2, Trapezoidal, 100.0, 100.0, %ofTime, myMccmTermType, CoordinatedMotion, Programmed, 0, 0, None, 0, 0);

See also

Structured Text Syntax

Input and Output Parameters Structure for Coordinate System Motion Instructions

Motion Error Codes (.ERR)

Multi-Axis Coordinated Motion Instructions

Common Attributes

How are we doing?