

[Instruction Set](#) > [Multi-Axis Coordinated Motion Instructions](#) > Master Driven Coordinated Control (MDCC)

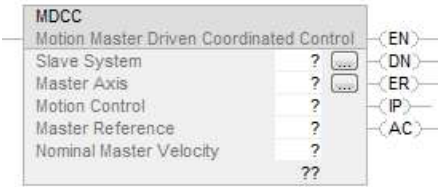
# Master Driven Coordinated Control (MDCC)

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.

The Master Driven Coordinated Control (MDCC) instruction defines a Master:Slave relationship between a Master Axis and a Slave Coordinate System.

## Available Languages

### Ladder Diagram



### Function Block

This instruction is not available in function block.

### Structured Text

MDCC (SlaveCoordinateSystem, MasterAxis, MotionControl, MasterReference, NominalMasterVelocity);

## Operands

### Ladder Diagram and Structured Text

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

Search

- [Quick Start Steps](#)
- [Logix Designer](#)
- [Module Information](#)
- ▴ [Instruction Set](#)

[Logix 5000 Controllers Instruction 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 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 Motion](#)

|                         |  |   |               |  |
|-------------------------|--|---|---------------|--|
| Slave System            | COORDINATE_SYSTEM  | COORDINATE_SYSTEM   | tag           | <p>The Coordinate System that the Master Axis controls when the motion planner is in Master Driven mode. Ellipsis launches the Coordinate System properties dialog.</p> <p>Upon verification, you receive a verification error if the Slave is a non-Cartesian Coordinate System or if the Master Axis is in the Slave Coordinate System.</p> <p>The MDCC link is broken when the following instructions are executed:</p> <p>On any axis in the Slave Coordinate System or the Slave Coordinate System: MAS (All), MCS (All), MGS, MASD, MCSD, MGSD, a mode change. Note that MAS (anything other than All) and MCS do NOT break the MDCC link.</p> <p>The Shutdown instructions (MGSD, MASD, MCSD) never go IP.</p> <p>On the Master Axis: MASD, MCSD, and MGSD. Note that MAS and MCS for any Stop Type, including All, do NOT break the MDCC link.</p> <p>A mode change (Rem Run to Rem Prog or Rem Prog to Rem Run) or an axis fault also breaks the MDCC link.</p> |
| Master Axis             | AXIS_CONSUMED<br>AXIS_CIP_DRIVE<br>AXIS_VIRTUAL<br><b>Tip:</b> AXIS_CONSUMED is supported by Compact GuardLogix 5580, CompactLogix 5380, and CompactLogix 5480 controllers only. | AXIS_CONSUMED<br>AXIS_SERVO<br>AXIS_SERVO_DRIVE<br>AXIS_GENERIC<br>AXIS_GENERIC_DRIVE<br>AXIS_CIP_DRIVE<br>AXIS_VIRTUAL | tag           | Any configured Single Axis that the Slave Coordinate System follows. The Master Axis can be any axis that has been configured.   |
| Motion Control          | MOTION_INSTRUCTION   | MOTION_INSTRUCTION  | tag           | Control tag for the instruction.   |
| Master Reference        | UNIT32   | UNIT32  | immediate tag | Selects the Master position source as either Actual Position (0) or Command Position (1).  |
| Nominal Master Velocity | REAL   | REAL  | immediate tag | Used as an additional conversion constant for master driven MCPM moves programmed in percentage of maximum.  |

## Master Reference

The Master Reference for an MDCC instruction selects the Master Axis position source.

The enumerations for Master Reference Axis are:

Actual – Slave motion is generated from the actual (current) position of the Master Axis as measured by its encoder or other feedback device.

Command – Slave motion is generated from the command (desired) position of the Master Axis. Because there is no Command Position for a Feedback Only Axis, if you select either Actual or Command for Master Reference, the Actual Position of the Master Axis is used. The Actual and Command Position are always the same for this case. No error is generated. Because there is no Actual Position for a Virtual axis, if you select either Actual or Command for Master Reference, the Command Position is used. No error will be generated.

An error is generated if a MDCC instruction is executed that changes the Master Reference of a Slave Coordinate

[Motion Coordinated Circular Move \(MCCM\)](#)

[Motion Coordinated Transform with Orientation \(MCTO\)](#)

[Motion Coordinated Path Move \(MCPM\)](#)

[Motion Calculate Transform Position with Orientation \(MCTP\)](#)

[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 instruction](#)

[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](#)

- [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](#)

An error is generated if a MDCC instruction is executed that changes the Master Reference or a Slave Coordinate System that is in motion. The new MDCC instruction will error and the original one will remain active.

## Nominal Master Velocity

If master driven coordinated move is programmed to run at a speed that equals 100% of maximum and master axis runs at its Nominal Master Velocity, then the coordinated move will run at the configured maximum speed of the coordinate system.

Calculate actual master axis velocity in percentage of the maximum as a percentage of actual master velocity with respect to nominal master velocity. For example, nominal master vel =50 ips therefore actual vel = 40 ips = (40/50) \* 100 = 80%.

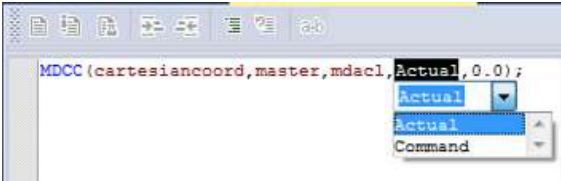
Calculate actual slave velocity in a percentage of the maximum as a product of programmed slave velocity in the percentage of the maximum and actual master axis velocity percentage of the maximum. For example, programmed slave velocity = 50% of the maximum, master velocity = 80% of the maximum, final slave velocity = 80 \* 50 = 40% of the maximum.

Convert final slave velocity in a percentage of maximum to Units/sec as usual. For example, final velocity = 40%, coordinate system maximum is speed 200 Units/sec, final velocity limit is 200 \* (40/100) = 80 Units/sec.

## Structured Text

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

Note that you have the option to browse for enumerations in the Structured Text Editor as shown below.



## Outputs

| Mnemonic                      | Description   |
|-------------------------------|---|
| .EN<br>(Enable) Bit<br>31     | The enable bit is set when the rung transitions from false-to-true and stays set until the rung goes false.   |
| .DN (Done)<br>Bit 29          | The done bit is set when the coordinate MDCC instruction is successfully initiated.   |
| .ER (Error)<br>Bit 28         | The error bit is set when there is an invalid combination of parameters in the MDCC instruction.  |
| .IP (In<br>Process)<br>Bit 26 | The in process bit is set when the MDCC instruction is activated and reset by an instruction (for example, the MCSD instruction).   |
| .AC (Active)<br>Bit 23        | The active bit is set when an MCLM or MCCM is activated (that is, when the AC bit of the MCLM or MCCM instruction is set) on a Coordinate System that is selected as the Slave Coordinate System of the MDCC instruction. |

## Description

MDCC 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.

The Motion Master Driven Coordinate Control instruction (MDCC) is used by the Master Driven Speed Control (MDSC) to synchronize one or more motion axes or Coordinate System to a common Master Axis when the Slave System is a Coordinate System.

The MDCC instruction defines the relationship between the external Master Axis and the Slave Coordinate System for the MCLM and the MCCM Instructions. When an MDCC is executed (goes IP), the specified Slave Coordinate System in the MDCC instruction is logically geared with the designated Master Axis. After motion on the Master Axis is initiated, all the axes in the Coordinate System specified as the Slave Coordinate System follow the Master Axis motion at the programmed dynamics of the programmed instruction.

There are no changes in any active motion when a new MDCC instruction is activated. Activating an MDCC instruction just puts the parameters programmed in the MDCC instruction into a pending state. The parameters in the pending MDCC instruction are changed if you execute a succeeding MDCC instruction before a new MCLM or

MCCM instruction is activated. The MDCC becomes active (AC bit is set) only when all programmed motion is complete.

MCLM instruction is activated. The MDCC becomes active (AC bit is set) only when all queued motion is complete and the motion queue is empty.

All motion in the queue keeps using the same Master Axis even if there is a pending MDCC with a different master. The values in the pending MDCC instruction are only used when the next MCLM or MCCM instruction is activated on the Slave Coordinate System when the queue is empty, or a MCLM or MCCM is executed (goes IP) with a Merge type of All or Coordinate. (Note that this is because the merge empties the queue.)

## Motion Direct Command and the MDCC Instruction

To obtain Motion Direct support for the MDCC instruction, you must first program an MDCC in one of the supporting program languages before you execute an MCLM or MCCM in Master driven Mode.

## Affects Math Status Flags

No

## Major/Minor Faults

None specific to this instruction. See Index Through Arrays for array-indexing 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 and rung-condition-in 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.

## Verification Errors

An invalid or No Master Axis will cause new errors to be generated when verified by the programming software. The following conditions may cause this error:

- The Master Axis is a member of the Slave Coordinate System.
- The Master Axis or the Slave Coordinate System is not configured.
- The Master Axis or an axis in the Slave Coordinate System is inhibited.
- A redefine position is in progress.
- Home of the Master axis or an axis in the Slave Coordinate System is in progress.

## Example

### Ladder Diagram



## Structured Text

```
MDCC(myMdccSlaveSystem,myMdccMasterAxis,myMdccMotionControl,Command,myMdccNominalMasterVelocity);
```

# See also

[Structured Text Syntax](#)

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

[Index Through Arrays](#)

Copyright © 2019 Rockwell Automation Technologies, Inc. All Rights Reserved.

[How are we doing?](#)