

[Instruction Set](#) > [Motion Move Instructions](#) > Master Driven Axis Control (MDAC)

# Master Driven Axis Control (MDAC)

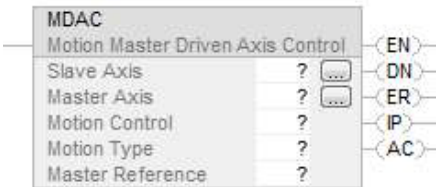
This information applies to the CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, and GuardLogix 5580 controllers. Controller differences are noted where applicable.

The Master Driven Speed Control (MDSC) function provides the ability to synchronize one or more motion axes to a common Master Axis. The MDSC uses the Master Driven Axis Control (MDAC) instruction, which assigns a Master:Slave relationship for single axes.

Using the MDSC function provides synchronization of axes without the use of CAM profiles. The MDSC functions may also be used with Time Driven CAM profiles (MATC) to make them act like position driven profiles synchronized with a Master Axis.

## Available Languages

### Ladder Diagram



### Function Block

This instruction is not available in function block.

### Structured Text

MDAC (SlaveAxis, MasterAxis, MotionControl, MotionType, MasterReference);

## Operands

### Ladder Diagram

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

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 Instructions](#)

▷ [Motion Event Instructions](#)

Slave Axis	AXIS_CIP_DRIVE  AXIS_VIRTUAL	AXIS_SERVO  AXIS_SERVO_DRIVE  AXIS_GENERIC_DRIVE  AXIS_CIP_DRIVE  AXIS_VIRTUALAX	Tag	<p>The Single Axis being controlled by the Master Axis when the motion planner is in Master Driven mode. Ellipsis launches Axis properties dialog.</p> <p>The MDAC link is broken when the following instructions are executed:</p> <p>On the Slave Axis: MAS (All), MCS (All), MGS, MASD, MCSD, MGSD, a mode change. Note that MAS (anything other than All) and MCS (coordinated) do NOT break the MDAC 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 MDAC link.</p> <p>A mode change (Rem Run to Rem Prog or Rem Prog to Rem Run) or an axis fault also breaks the MDAC link.</p>
------------	------------------------------------	--	-----	---

Master Axis	AXIS_CONSUMED	AXIS_CONSUMED	Tag	Any configured
-------------	---------------	---------------	-----	----------------

- ▷ [Motion Group Instructions](#)
- ◀ [Motion Move Instructions](#)
- [Master Driven Axis Control \(MDAC\)](#)
- [Motion Axis Gear \(MAG\)](#)
- [MAG Flow Chart \(True\)](#)
- [Motion Axis Home \(MAH\)](#)
- [MAH Flow Chart \(True\)](#)
- [Motion Axis Jog \(MAJ\)](#)
- [Motion Axis Move \(MAM\)](#)
- [Motion Axis Position Cam \(MAPC\)](#)
- [MAPC Flow Chart \(True\)](#)
- [Motion Axis Stop \(MAS\)](#)
- [Motion Axis Time Cam \(MATC\)](#)
- [MATC Flow Chart \(True\)](#)
- [Motion Calculate Cam Profile \(MCCP\)](#)
- [Motion Calculate Slave Values \(MCSV\)](#)
- [Motion Change Dynamics \(MCD\)](#)
- [MCD Flow Chart \(True\)](#)
- [Motion Redefine Position \(MRP\)](#)
- [MRP Flow Chart \(True\)](#)
- [Speed, Acceleration, Deceleration, and Jerk Enumerations](#)
- [Status Bits for Motion Instructions \(MAM, MATC, MAJ\) When MDAC Is Active](#)
- [Time Based Planning](#)
- [Change between Master Driven and Time Driven](#)
- [Modes for Single Axis Motion instructions](#)
- [Common Action Table for Slave and Master Axis](#)
- ▷ [Motion State Instructions](#)
- ▷ [Multi-Axis Coordinated Motion Instructions](#)
- ▷ [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](#)

	AXIS_CIP_DRIVE  AXIS_VIRTUAL  <b>Tip:</b> AXIS_CONSUMED is supported by Compact GuardLogix 5580, CompactLogix 5380, and CompactLogix 5480 controllers only.	AXIS_SERVO  AXIS_SERVO_DRIVE  AXIS_GENERIC_DRIVE  AXIS_CIP_DRIVE  AXIS_VIRTUAL		Configured Single Axis that the Slave Axis follows. The Master Axis can be any axis that has been configured.  Ellipsis launches Axis properties dialog.
Motion Control	MOTION_INSTRUCTION	MOTION_INSTRUCTION	Tag	Control tag for the instruction.
Motion Type	UNIT32	UNIT32	Immediate or Tag	Specifies the move type (MAM, MAJ, MATC, or MAM - Master Offset Move for a Position Cam) executing on the Slave Axis that will be controlled by the Master Axis when a single axis motion instruction is programmed in Master Driven Mode. The enumerations for Motion Type are:  All = 0  Move = 1  Jog = 2  Time Cam = 3  Master Offset Move = 4
Master Reference	UNIT32	UNIT32	Immediate or Tag	Selects the Master Axis position source as either Actual Position (0) or Command Position (1).

▷ [Studio 5000 Logix Designer Glossary](#).

The All for the motion type lets the system know that Master Driven Mode applies to all single axis instructions (that is, MAM, MAJ, MATC, and a MAM (Master Offset Move)). If the Motion Type of the active MDAC is set to anything other than All on a Slave Axis, then executing a MDAC All on the Slave Axis causes a runtime error. For example:

MDAC Move=MAM Master=X Slave=Y - Assigns master and slave axes

MDAC Move=ALL Master=Z Slave=Y - Causes an error because the previous MDAC assigned the MAM to the Y axis. Note that assigning the Slave Y axis to any other Master axis causes the same error because Y is already assigned to a MAM.

To be able to reassign the Master once the All option is used, it is sufficient to execute one of the following for either the Slave Axis or the Master Axis, although you can also choose to execute one of the following for both the Slave Axis and the Master Axis:

- On the Slave Axis: MAS (All), MCS (All), MGS, MASD, MCSD, or MGSD
- On the Master Axis: MASD, MCSD, or MGSD

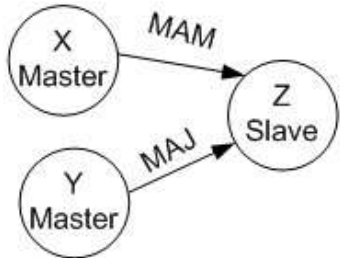
If there is an active MDAC (All) on a Slave, then setting the MDAC instruction to a Motion Type other than All causes a runtime error.

To be able to reassign the Master once anything other than the All option is used, you must execute any of the following:

- on the Slave Axis: MAS (All), MCS (All), MGS, MASD, MCSD, or MGSD
- On the Master Axis: MASD, MCSD, or MGSD

If you assign the same axis to be both a Master and a Slave Axis, an RSLogix 5000 software verification error is generated.

The same Slave axis can be assigned to 2 different Master axes. For example, if X, Y are the Master axes and Z is a Slave axis, then X can be assigned to be the Master axis for MAM instructions for Z and Y is the Master axis for MAJ instructions for Slave Z. Refer to the following illustration.



<p>The Master Axis for an active Slave may be changed by executing a MDAC for the slave with a different Master. The MDAC puts the pending Master in a queue for the Slave. When the next single axis motion instruction is executed (goes IP) in either Replacement or Merge mode, the queued Master Axis becomes the Master for the Slave the next time a single axis motion is started. For example:</p> <p>In this example, the masters are already running. MDAC Move=MAM Master=X Slave=Z</p>	<p>Assigns master and slave axis</p>
<p>MAM X</p>	<p>Start MAM motion on the master 1</p>
<p>MAM Z</p>	<p>Start MAM motion on the slave</p>
<p>MAM Y</p>	<p>Start MAM motion on the Y master 2</p>

MDAC Move=MAM    Master=Y Slave=Z	Y is pending to be the new master
MAM Z	Start new MAM motion on the slave using new master 2

Use different control words on all MDAC instructions. If the same control word is used for the active and pending MDAC, the IP bit of the pending MDAC will not work properly. When separate axis control words are used for the active and the pending MDAC instructions, the active and the pending MDAC will both have their IP bits set. However, only the active MDAC will have its AC bit set. This is standard operation for all instructions. No detection of the same control word used on multiple instructions is made.

## Master Reference

The Master Reference for an MDAC 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 MDAC instruction is executed that changes the Master Reference of a slave axis that is in motion. The new MDAC instruction will error and the original one will remain active.

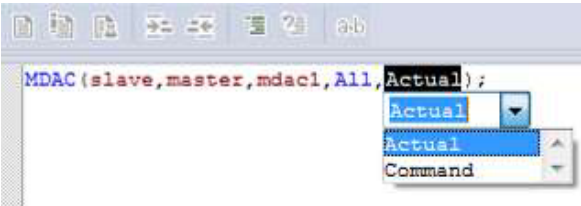
## Structured Text

Operand	Type	Format	Description
Source	SINT INT DINT REAL STRING structure	tag	Initial element to copy  Important: the Source and Destination operands should be the same data type, or unexpected results may occur

Destination	SINT	tag	Initial element to be overwritten by the Source  Important: the Source and Destination operands should be the same data type, or unexpected results may occur
	INT		
	DINT		
	REAL		
	STRING		
	LINT		
	structure		
Length	DINT	immediate or tag	Number of Destination elements to copy

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 in the following.



## MOTION\_INSTRUCTION Bit Leg Definitions for MDAC

Mnemonic	Description
.EN (Enable) Bit 31	The enable bit is set when the rung transitions from false-to-true and stays set until the rung goes false.
.DN (Done) Bit 29	The done bit is set when the Master Driven Axis Control instruction is successfully initiated.
.ER (Error) Bit 28	The error bit is set when there is an invalid combination of parameters in the MDAC instruction.
.IP (In Process) Bit 26	The in process bit is set when the MDAC instruction is activated and reset by an instruction (for example, the MASD instruction).
.AC (Active) Bit 23	The active bit is set when a move (MAJ, MAM or MATC) goes IP in Master Driven mode on the axis that is selected as the Slave Axis of the MDAC instruction. If the Slave Axis is started in Time Driven mode, then the AC bit of the MDAC does not go active. The IP bit of the MDAC instruction does not change at this time.

## Description

The MDAC instruction is used to select a single axis as a Master Axis and a single axis as a Slave Axis. The MDAC instruction connects a Master and Slave Axis for a MAM, MAJ, and MATC instruction. When an MDAC is executed (goes IP), the specified Slave Axis in the MDAC instruction is logically geared with the designated Master Axis. After motion on both the Master and Slave Axes has been initiated, the Slave follows the Master Axis motion using the programmed dynamics of the motion instruction.



There are no changes in any active motion when a new MDAC instruction is activated. Activating a new MDAC instruction puts the parameters programmed in the MDAC into a pending state. The parameters in the pending MDAC instruction are overridden if you execute a succeeding MDAC before a new single axis motion instruction (MAM, MAJ, and MATC) is activated on the slave axis. The most recent values in the pending MDAC instruction are used when a new single axis motion instruction is activated on the Slave Axis.

A MATC with an Execution Schedule of Pending always uses the Instruction mode of the active MATC; Lock Position and Lock Direction are ignored. This is explained further in the following example.

Assume that there are four instructions executed in the following order:

- MDAC1: Master=X, Slave=Y
- MATC1: Slave Y, Immediate
- MDAC2: Master=Z Slave=Y MDAC2 is executed (goes IP) while MATC1 is moving (Master Axis is changed)
- MATC2: Slave=Y, Pending
- MATC2: will use the parameters from MDAC1 (Master and Slave) because MATC1 was moving (active) when MDAC2 was executed. If MATC2 and MDAC2 were executed in the reverse order, you get the same results. This follows the existing paradigm of MAPC instruction where the pending CAM uses the same master as the running CAM.

## Motion Direct Command and the MDAC Instruction

To obtain Motion Direct support for Master Driven Speed Control mode, you must first program an MDAC in one of the supporting program languages before you execute an MAM or MAJ in Time driven Mode.

## Affects Math Status Flags

No

## Major/Minor Faults

None specific to this instruction. See the 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 in the Ladder Diagram table.
Postscan	See Postscan in the Ladder Diagram table.

# Error Codes

See Motion Error Codes (.ERR) for run time errors for motion instructions when MDAC is active.

# 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:

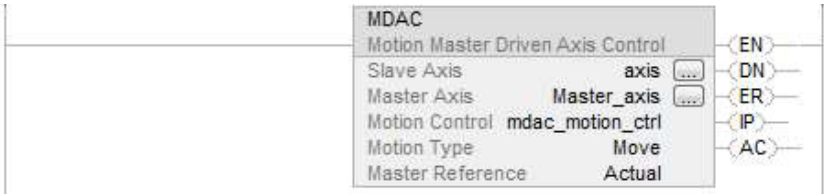
- The Master and Slave Axis are the same.
- Master or Slave Axis is not configured.
- Master or Slave Axis is inhibited.
- Redefine position is in progress.
- Home of an axis is in progress.

MDSC ALL Conflict. The ALL parameter is not allowed while a Slave axis motion generator (for example, the jog motion generator) is already assigned.

# Examples

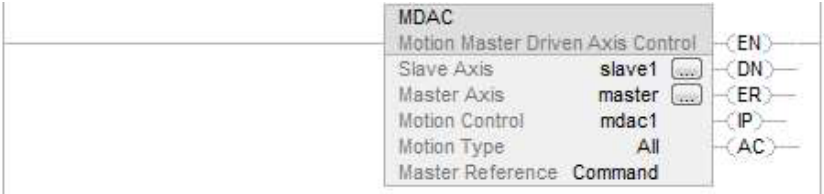
## Example 1

## Ladder Diagram



## Example 2

## Ladder Diagram



# See also

[Common Action Table for Slave and Master Axis](#)

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

[Motion Move Instructions](#)

[Structured Text Syntax](#)

[Common Attributes](#)