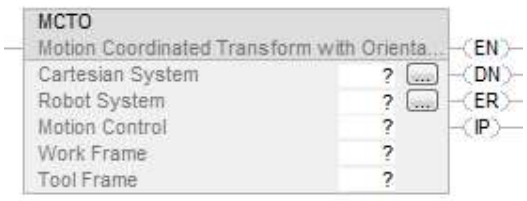
Motion Coordinated Transform with Orientation (MCTO)

This information applies to the Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, and GuardLogix 5580 controllers.

Use the MCTO instruction to establish a bidirectional transform that is set up between a Cartesian and a robot systems with coordinates that are joint axes of a robot. The XYZ translation coordinates and the RxRyRz orientation coordinates in the fixed angle convention define the Cartesian coordinates. The geometrical configurations of the robots typically have joint axes that are not orthogonal. The coordinate system type, such as Delta, specifies the geometrical configurations.

Available Languages

Ladder Diagram



Function Block

This instruction is not available in function block.

Structured Text

MCTO(CartesianSystem, RobotSystem, MotionControl, WorkFrame, ToolFrame);

Operands

Important: Do not use the same tag name for more than one instruction in the same program. Do not write to any instruction output tag under any circumstances.

ATTENTION: If instruction operands are changed while in Run mode, the pending edits must be accepted and the controller mode cycled from Program to Run for the changes to take effect.

Configuration

The following table provides the operands used to configure the instruction. These operands cannot be changed at runtime.

Operand	Data Type	Format	Description
Cartesian System	COORDINATE_SYSTEM	tag	Cartesian coordinate system used to program the moves.
Robot System	COORDINATE_SYSTEM	tag	Non-Cartesian coordinate system that controls the actual equipment.
Motion Control	MOTION_INSTRUCTION	tag	Control tag for the instruction.

Work Frame	POSITION_DATA	immediate tag	Work Frame offsets are the offsets used to locate the user Work frame of the Robot relative to the origin of the Robot base frame. These offsets consist of an XYZ and RxRyRz value. This allows the programs to be written in the user work space or world frame and transformed to the Robot base frame. To rotate the target position around the X, Y, or Z axis or offset the target position along the X, Y, or Z axis of the Robot base coordinate system, enter the degrees of rotation into the Rx, Ry, and Rz tag members in units of degrees of rotation, and enter the offset distances into the X, Y, and Z tag members in coordination units. Set the ID member to a value greater than or equal to zero. To maintain the work frame at the robot base frame, leave structure values at zero, or set the operand tag value to zero.
Tool Frame	POSITION_DATA	immediate tag	Tool Center Point (TCP) offsets are the offsets used to locate the tool center relative to the center of the End of Arm. These offsets consist of an XYZ and RxRyRz value. To have the target position account for an attached tool having translation and/or orientation offsets, enter the tool offset distances into the X, Y, and Z tag members in coordination units. Enter the degrees of tool rotation into the Rx, Ry, and Rz tag members in units of degrees of rotation. Set the ID member to a value greater than or equal to zero. To have the target position reflect only the point at the end-of-arm, leave the structure values at zero, or set the operand tag value to zero.

For further information of configuring coordinate systems refer to the Motion Coordinate System User Manual publication MOTION-UM002.

Important: Do not write to any instruction output tag under any circumstances.

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	Same as Rung-condition-in is false.
Rung-condition-in is false	The .EN, .DN, and .ER are cleared to false.
Rung-condition-in is true and .EN bit is false	The .EN bit is set to true and the instruction executes.
Rung-condition-in is true and .EN bit is true	N/A
Postscan	Same as Rung-condition-in is false.

Structured Text

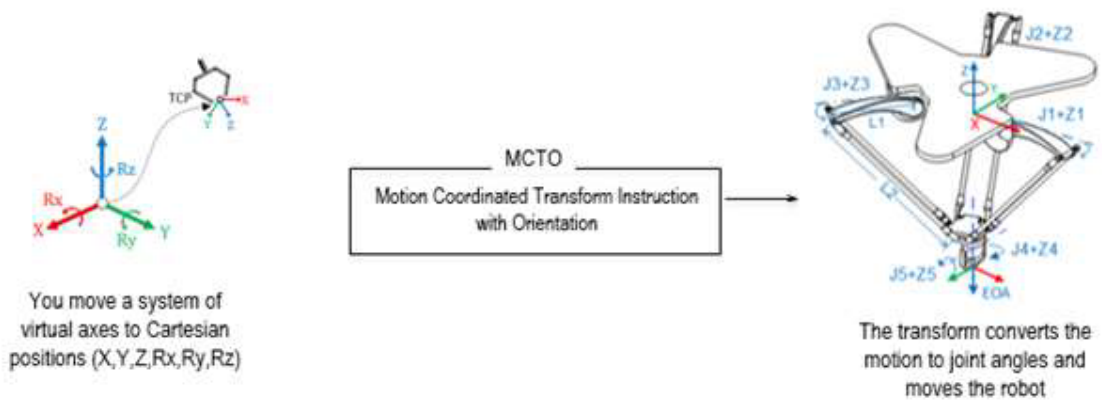
Condition/State	Action Taken
Prescan	See Prescan in the Ladder Diagram table.
Normal execution	See Rung-condition-in is false followed by Rung-condition-in is true in the Ladder Diagram table.
Postscan	See Postscan in the Ladder Diagram table.

Data Flow of MCTO instruction between two coordinate systems

Establish the reference frame for the joint coordinate system to ensure the transformations work properly. Refer to the applicable geometry configuration topic for further information on establishing the reference frame for a robot coordinate system.

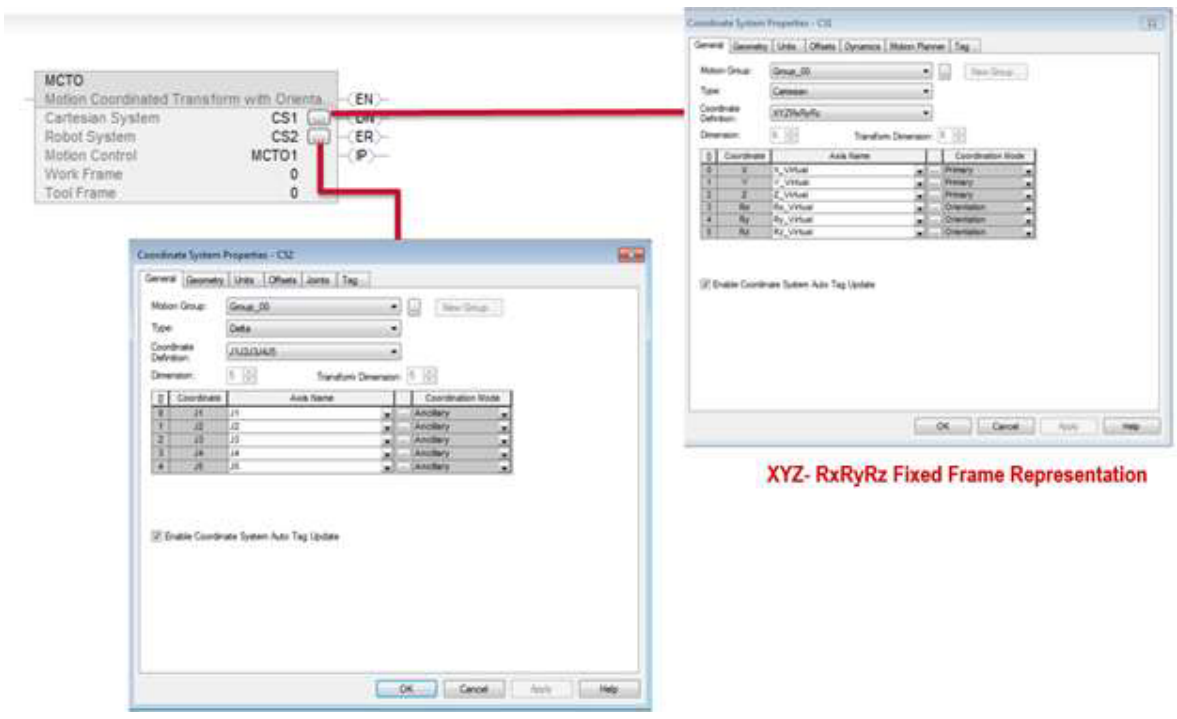
Once the robot reference frame is established, the robot can be moved to any desired position in joint space when transform is not enabled. When MCTO is initiated for the first time, a forward transform is executed first to set the corresponding Cartesian coordinate positions. Once the MCTO instruction is active, a bidirectional transform link is established, so that if the Cartesian coordinate is commanded to move to a target position in the Cartesian space, the robot moves to the Cartesian target coordinates along a linear path. Similarly, if the Robot is commanded to move to a joint coordinate position, the Robot moves to the joint target position along a non-Cartesian path. While MCTO instruction is active, the system maintains the coordinate system related data for Cartesian and Robot coordinate systems.

This diagram illustrates the transformation from a Cartesian coordinate system(X,Y,Z,Rx,Ry,Rz) to a Delta robot geometry with 5 axes (J1J2J3J4J5). The Rx, Ry, and Rz represent orientations around X,Y,Z axes in a positive direction.



Configure an MCTO instruction

The following illustration shows how to configure an MCTO instruction, with the Cartesian coordinate system as the source and the Delta 5 axis geometry as the target. Configure the source and target coordinate systems in the coordinate system dialog box.



Data flow during forward transform when MCTO is active

A forward transform executes when a move executes on joint coordinates of a robot system with an enabled MCTO instruction. During forward transform, the current joint angle positions compute the corresponding Cartesian coordinate positions based on geometry.

Input Data			Computed Output	
CS2:DATA	SOURCE	Active instruction → MCTO	CS1:DATA	Destination
Joint Positions (J1,J2,J3,J4,J5)	Machine Real Coordinate System		Cartesian Positions (X,Y,Z,Rx,Ry,Rz)	Machine Virtual Coordinate System
Link Lengths (L1,L2)	Coordinate System dialog			
Zero Angle Orientations(Z1,Z2,Z3,Z4,Z5)	Coordinate System dialog			
Base and Effector Plate Dimensions (Rb,Re)	Coordinate System dialog			
Swing Arm Offsets (D3, A3, D4, A4, D5)	Coordinate System dialog			
Coupling Direction	Coordinate System dialog			
Coupling Ratio	Coordinate System dialog			
Work Frame Offsets(Array[6])	Instruction Faceplate			
Tool Frame Offsets(Array[6])	Instruction Faceplate			

Data flow during inverse transform when MCTO is active

When MCTO is active, during inverse transform, the instruction uses the current cartesian positions and geometry configuration to compute the corresponding joint angle positions. MCTO returns an Error 61 when moving both Cartesian and Joint coordinate systems simultaneously.

Input			Computed Output	
CS1:DATA	SOURCE	Active instruction → MCTO	CS2:DATA	Destination
Cartesian Position(X,Y,Z,Rx,Ry,Rz)	Machine Virtual Coordinate System		Joint Positions (J1,J2,J3,J4,J5)	Machine Real Coordinate System
Link Lengths (L1,L2)	Coordinate System dialog			
Zero Angle Orientations(Z1,Z2,Z3,Z4,Z5)	Coordinate System dialog			
Base and Effector Plate Dimensions (Rb,Re)	Coordinate System dialog			
Swing Arm Offsets (D3, A3, D4, A4, D5)	Coordinate System dialog			
Coupling Direction	Coordinate System dialog			
Coupling Ratio	Coordinate System dialog			
Work Frame Offsets(Array[6])	Instruction Faceplate			
Tool Frame Offsets(Array[6])	Instruction Faceplate			

Fault codes

For the Motion Coordinated Transform with Orientation (MCTO) instruction, an error will occur at runtime if invalid operand values are provided.

Extended Error codes

Extended Error codes help to further define the error message given by the instruction. Their meaning is dependent upon the Error Code with which they are associated.

Error Code	EX_ERROR Code	Description
13	3	Value Out Of Range (base angle) Any orientation angle > 360 or < -360
13	3	Value Out Of Range (base ID) ID equals -1
13	4	Value Out Of Range (tool angle) Any orientation angle > 360 or < -360
13	4	Value Out Of Range (tool ID) ID equals -1
61	1	Connection Conflict: Transform Motion Group Error. Cartesian or Robot coordinate system is ungrouped, or associated to a different motion group from the other.
61	2	Connection Conflict: Transform Duplicate Systems Error. Operand 0 and Operand 1 are of the same coordinate system type.
61	3	Connection Conflict: Transform Source Dimension Error. Transform dimension of the Cartesian coordinate system is < 2.
61	4	Connection Conflict: Transform Target Dimension Error. Robot coordinate system Transform Dimension is equal to zero.
61	5	Attempting to use the CS as multiple sources - FAN-OUT error.
61	6	Attempting to use the CS as multiple targets - FAN-IN error.
61	9	Connection Conflict Transform Axes Overlap Error An axis is a member of both the Cartesian and Robot systems.

61	10	Axis or axes are in motion with the below exception: Gearing or camming motion is in progress on the Cartesian CS axes. The Cartesian CS slave axes cannot have a master axis from the Robot CS.
61	12	Connection Conflict Transform Invalid Link Length Link length values for any robot geometry must be > 0.0 units.
61	13	Axis is shut down
61	14	Axis is inhibited
61	15	Connection Conflict Transform Invalid Delta Configuration Link Length1 must not be equal to LinkLength2 End Effector Offset1 Re must not be negative For Delta J1J2J6 and Delta J1J2J3J6 End Effector Offset3(D3) must not be negative (Link length 1 + Rb - Re) must be less than link length 2 (Link length 1 + Rb – Re) must be positive or greater than zero
67	1	Invalid Transform Position Invalid Rx Orientation Invalid Rx orientation value at EOA transformations. For 4 axis geometries only Rx = 180 deg position is allowed. For all other positions, it will generate error.
67	2	Invalid Transform Position Invalid Ry Orientation Invalid Ry orientation value at EOA transformations. For 4 axis geometries, after removing work frame and tool frame, there should not be any Ry orientation values at EOA
67	3	Invalid Transform Position Invalid Rz Orientation

147	3-5	Invalid Orientation Scaling Constant Extended error code 3 : Rx Extended error code 4 : Ry Extended error code 5 : Rz Orientation axis has scaling constant > MAX_K_CONSTANT_FOR_ORIENTATION_AXIS. or has scaling constant which is not Integer or has Conversion ratio between Coordination Units and Position Units other than 1:1.
148	3-5	MCPM Orientation Offset Not Rx orientation offset not valid. Extended error code 3 : Rx orientation offset not valid. Extended error code 4 : Ry orientation offset not valid. Extended error code 5 : Rz orientation offset not valid. If robot Geometry is (MO_CD_J1J2J6 or MO_CD_J1J2J3J6) and: • workframe offset Rx isn't 0 or • workframe offset Ry isn't 0 or • toolframe offset Rx isn't 0 or • toolframe offset Ry isn't 0 Or if robot Geometry is MO_CD_J1J2J3J4J5 and: • workframe offset Rx isn't 0 or • workframe offset Ry isn't 0 or • toolframe offset Rx isn't 0 or • toolframe offset Rz isn't 0
149	3-5	Orientation Axis Not Virtual Extended error code 3 : Rx axis must be virtual if transforms are enabled. Extended error code 4 : Ry axis must be virtual if transforms are enabled. Extended error code 5 : Rz axis must be virtual if transforms are enabled.
151	4	Joint Angle Beyond Limits This indicates the error condition when the Joint 4 in a 5 axis Delta goes beyond turns counter range limit (45899.99<J4<-45900) Ext Error 4 : Joint J4 Beyond Limit
151	5	Joint Angle Beyond Limits This indicates error condition when the Joint 5 in a 5 axis Delta goes beyond +/-179, +179 > J5 <-179 Ext Error 5 : Joint J5 Beyond Limit

151	6	Joint Angle Beyond Limits This indicates the error condition when the Joint 6 in a 4 axis Delta goes beyond turns counter range limit (45899.99<J6<-45900) Ext Error 6 : Joint J6 Beyond Limit
152	1	Maximum Orientation Speed Exceed for Rx When Orientation axis Rx is commanded to move by an angle greater than or equal to 180 degrees in one coarse update period, this error and extended error will be returned.
152	2	Maximum Orientation Speed Exceed for Ry When Orientation axis Ry is commanded to move by an angle greater than or equal to 180 degrees in one coarse update period, this error and extended error will be returned.
152	3	Maximum Orientation Speed Exceed for Rz When Orientation axis Rz is commanded to move by an angle greater than or equal to 180 degrees in one coarse update period, this error and extended error will be returned.
153	1	Invalid Translation Position MOP Invalid X Translation Translation on X axis is Invalid
153	2	Invalid Translation Position MOP Invalid Y Translation Translation on Y axis is Invalid
153	3	Invalid Translation Position MOP Invalid Z Translation Translation on Z axis is Invalid

Example

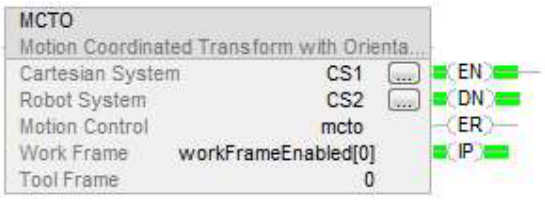
The following examples activate the transformations between Cartesian coordinate system CS1 and robot coordinate system CS2, which is a Delta 5 axis geometry. In the first example, neither work frame nor tool frame are specified, so Cartesian positions are computed at robot end-of-arm (EOA), with respect to the robot base frame.

Ladder Diagram

When MCTO is active, with a move to Cartesian positions X = 7.361, Y = -4.25, Z = -928.18, Rx = 180, Ry = 30 and Rz = -30, the MCTO computes the corresponding Joint angle as shown in the following diagram.

J1.ActualPosition	Contro...	9.999875
J2.ActualPosition	Contro...	9.999875
J3.ActualPosition	Contro...	9.999875
J4.ActualPosition	Contro...	30.0
J5.ActualPosition	Contro...	-30.0

Consider the same MCTO, with work frame offsets enabled for X = 50, Y = 50 and Rz = 50. Now MCTO will compute the Cartesian positions with respect to the new work frame.



When a move is programmed to the same end position, the corresponding joint positions are as shown in the following diagram.

J1.ActualPosition	Contro...	5.511625
J2.ActualPosition	Contro...	8.650625
J3.ActualPosition	Contro...	16.992374
J4.ActualPosition	Contro...	-20.0
J5.ActualPosition	Contro...	-30.0

Consider that the MCTO is programmed with tool frame offsets also enabled for Ry = 50 along with the work frame offsets. MCTO will now compute the Cartesian positions with respect to the end of the new work frame and tool frame.

After programming a move to the same end position the corresponding Joint angle positions are as shown in the following diagram.

J1.ActualPosition	Contro...	8.266875
J2.ActualPosition	Contro...	13.584
J3.ActualPosition	Contro...	22.686874
J4.ActualPosition	Contro...	-20.0
J5.ActualPosition	Contro...	-80.0

For more information on configuring offsets, see Configure Coordinate System Offsets.

Structured Text

MCTO(CS1, CS2, MCTO1, 0, 0);

Tip: For further information on creating geometries with orientation support, see the Motion Coordinate System User Manual publication MOTION-UM002.

See also

Structured Text Syntax

Index Through Arrays

Motion Error Codes (.ERR)

Multi-Axis Coordinated Motion Instructions

Define coordinate system frames

How are we doing?