EtherCAT Motion Control

User Manual

Suzhou VEICHI Electric Technology Co.Ltd

08/2023

Catalouge

1. Overview

EtherCAT is a field-ready, ultra-high-speed I/O network with short communication refresh cycles, low synchronization jitter, low hardware cost, and several other feature. It uses a standard Ethernet physical layer and a conventional Ethernet card, and the media can be twisted pair or fiber.

VC5 supports standard EtherCAT interface (1 RJ45 interface), Supports up to 32 bus servo axes, 4 local pulse axes, 2 probes per axis, up to 72 EtherCAT slaves, linear topology, EtherCAT bus cycles can be set to 250us minimum.

EtherCAT Interface Specifications

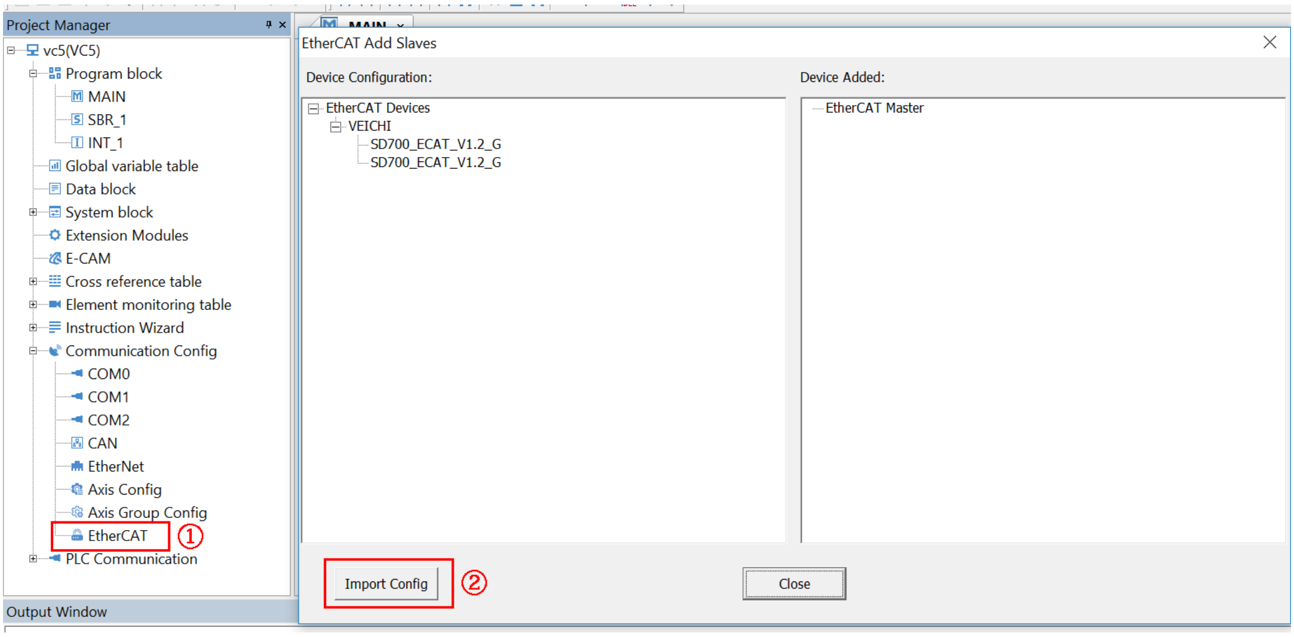
Items	Specification
Transmission speed	100Mbps：100BASE-TX
Modulation	Baseband
Topology	Wire, Daisy chain
Transmission medium	Category 5 or higher twisted pair or shielded twisted pair with aluminum foil and braided mesh
Transmission distance	Distance between nodes: 100m or less
Number of connections	72

2. Master Station Configuration

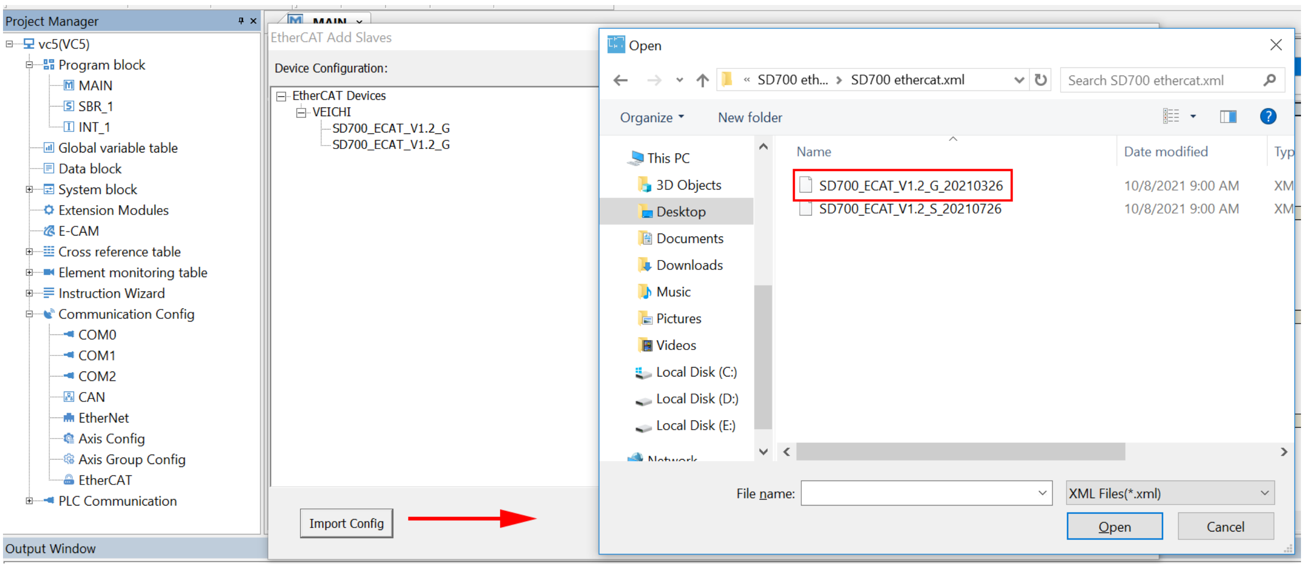
2.1 Importing Device XML

Importing device XML means importing the ETG (EtherCAT Technical Committee) compliant device description files (with the suffix ".XML") into the programming software Auto Studio, which parses the files to generate EtherCAT devices that can be added or deleted by the user. The common EtherCAT slave devices of VEICHI are integrated inside the programming software Auto Studio and do not need to be installed separately. If you need to use a third party EtherCAT device, you must first install the description file for that device. Take the example of importing the VEICHI drive SD700.

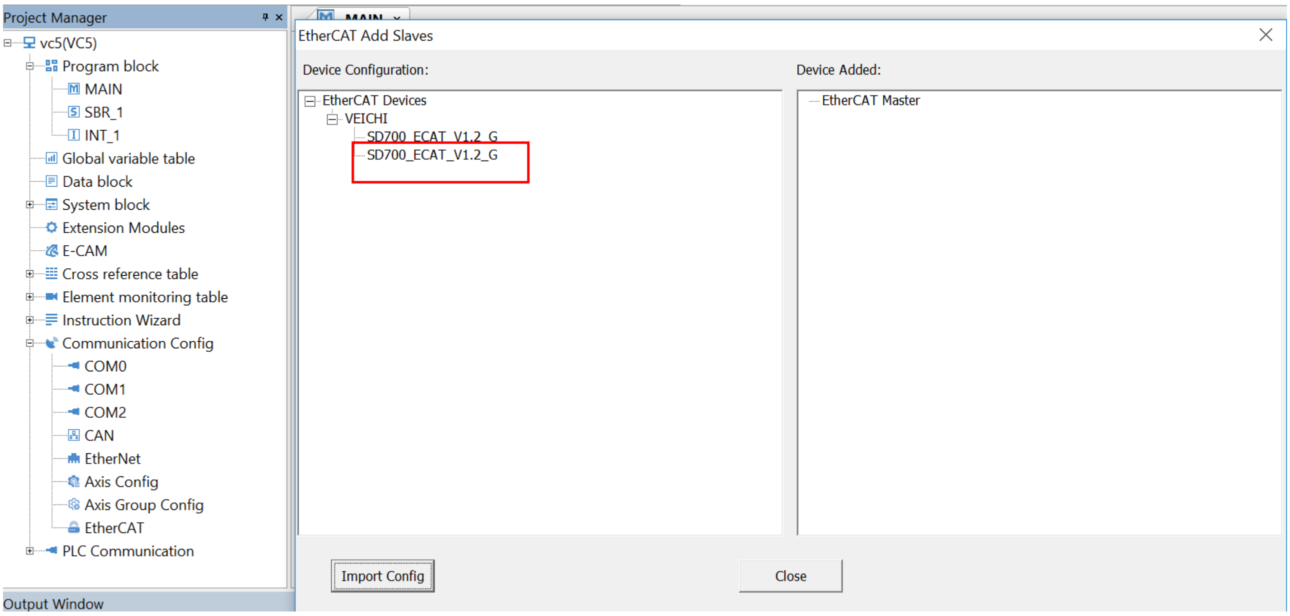
1. Create a new VC5 project, select the EtherCAT option, right-click and select, Add Device, select ImportConfig.



2.In the pop-up dialog box, select the XML file to be added and import it；

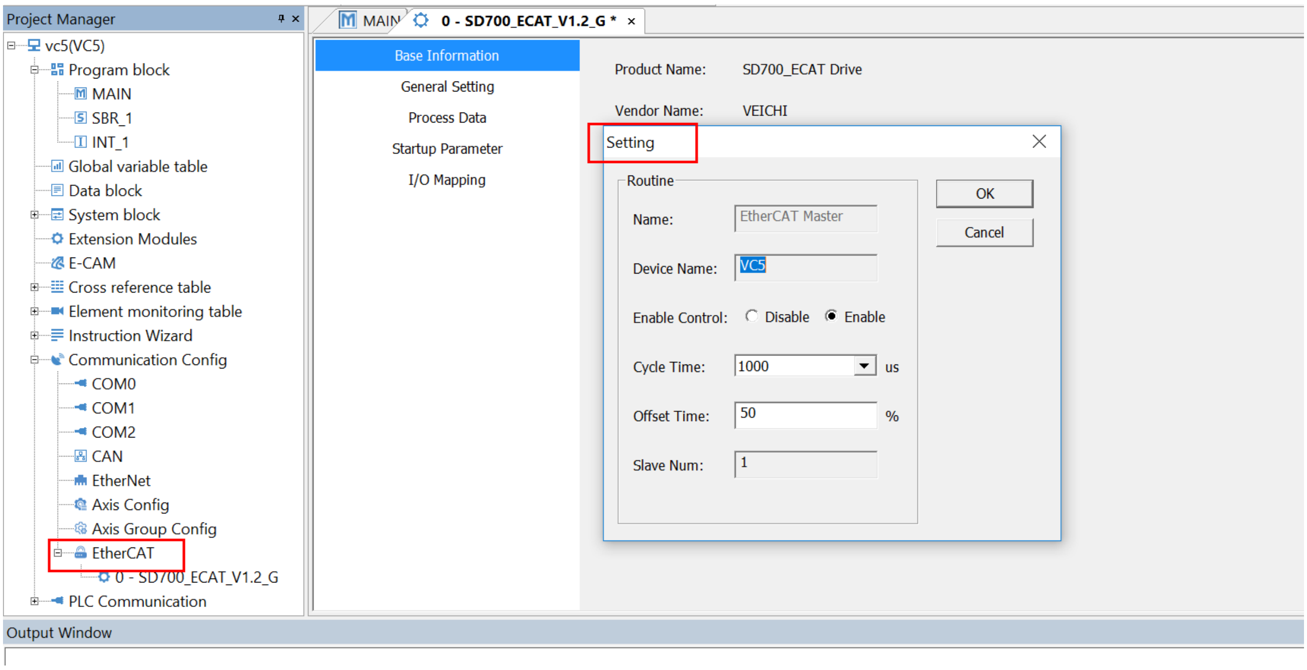


3. If you need to import more than one device, you can repeat step 2. After the file is successfully imported, you can see the newly imported devices under the EtherCAT device list, and then choose to close the window.



2.2 Master Settings

Select the EtherCAT option and click right mouse button to open the basic settings interface, where you can set the EtherCAT master enable or disable and the cycle time and synchronization offset.



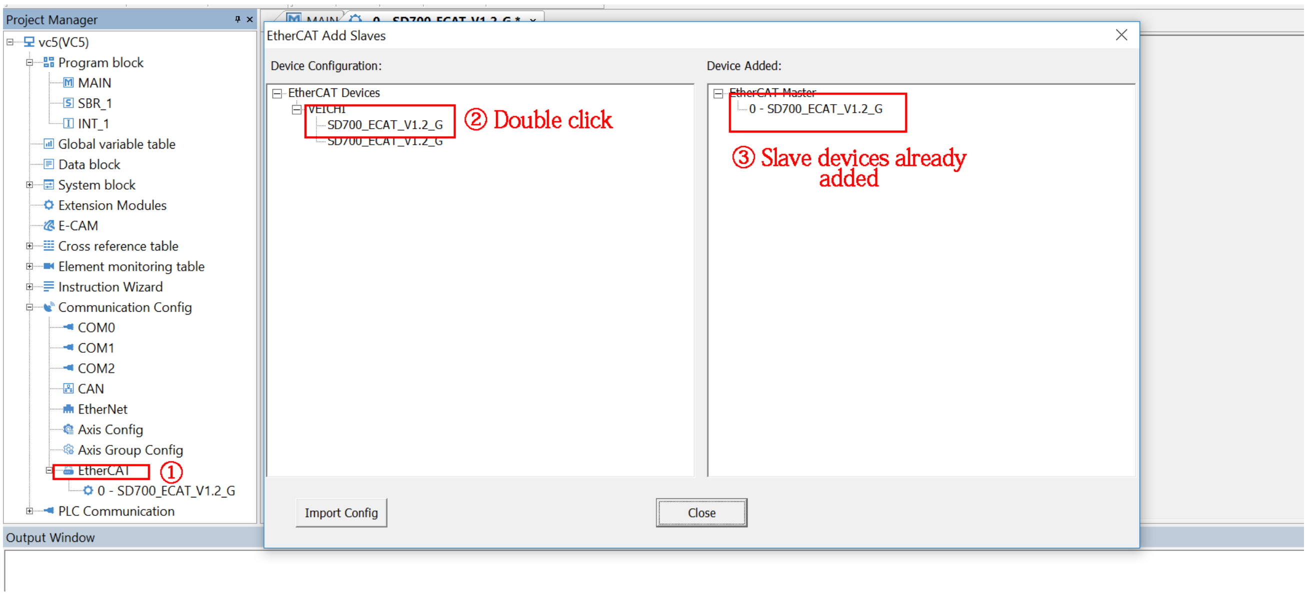
Enable control: Disable means to disable the EtherCAT master, to disable all masters and slaves under the bus, the bus servo axis associated with the slave will no longer run, enable is the opposite of it, the default is disable.

Cycle time: EtherCAT data frame sending interval and cycle time of EtherCAT tasks.

Synchronous offset: Percentage relative offset of the EtherCAT task with respect to the Sync0 interrupt of the slave.

2.3 Adding Slave Devices

After the master is set up, select the EtherCAT option and right click on Add Device, take the example of adding drive SD700：

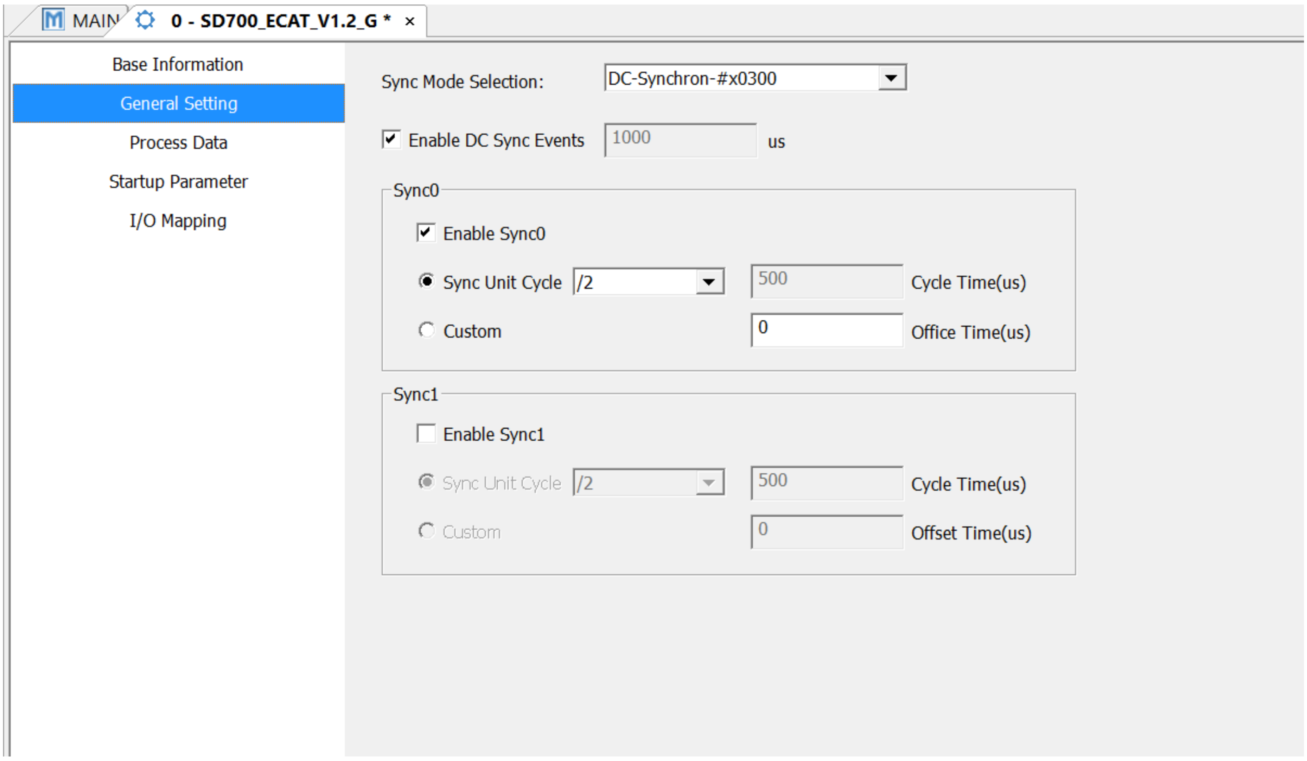


After the slave has been successfully added, you can see the added EtherCAT slave devices under the EtherCAT option.

3. Slave Configuration

3.1 General settings

This section is used to set the synchronous operation mode of the slave with the following interface:



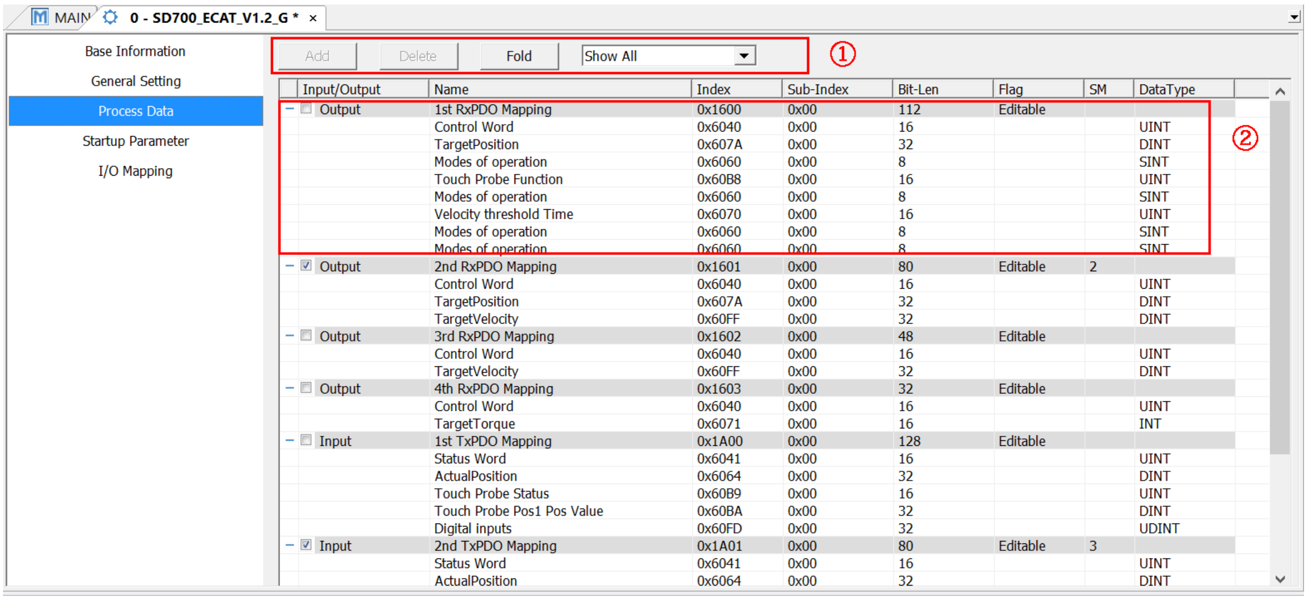
Sync mode selection：Operating mode synchronized to distributed clock (DC-Synchron)

In this mode, the slave's Sync interrupt can be configured and enabled by default for DC sync events, with the Sync0 interrupt enabled and the Sync1 interrupt not enabled.

3.2 Process data manipulation

The process data interface is used for editing PDO:

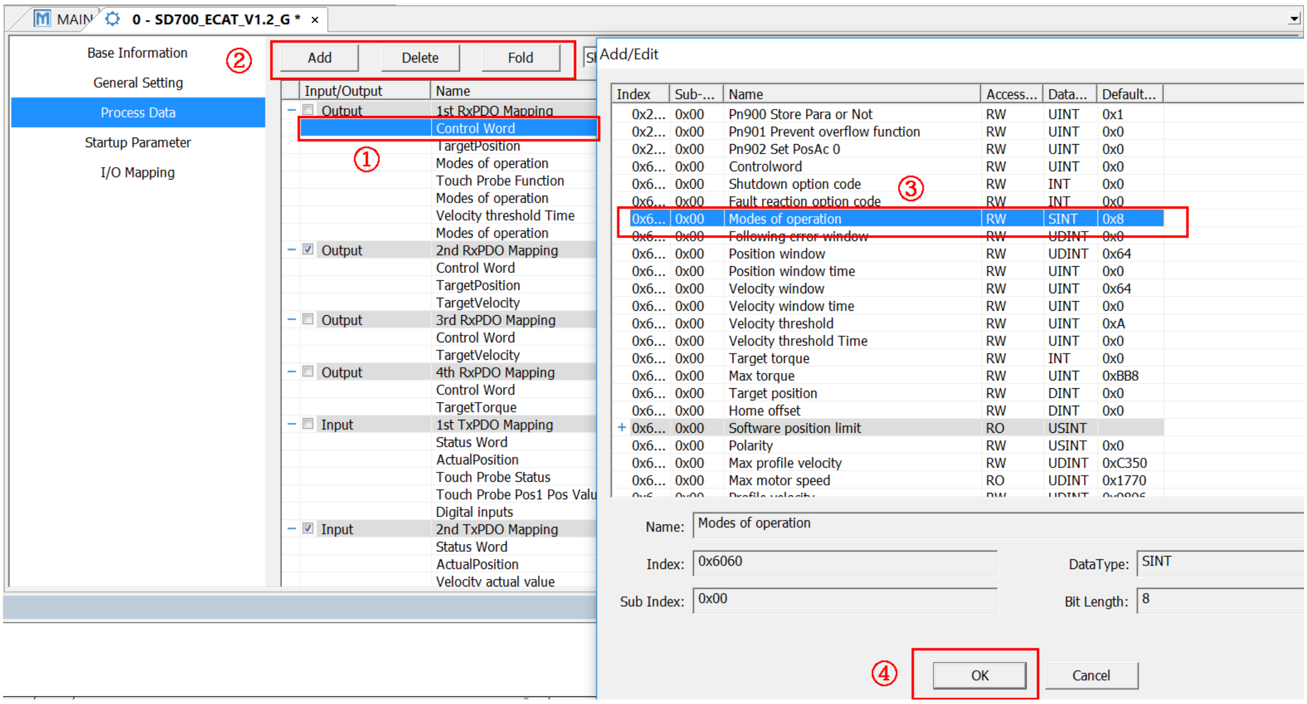
The interface is displayed as follows.



1. — PDO edit button

2. — PDO display area

PDO is divided into output group PDO and input group PDO according to data flow direction. The output group PDO represents the process data sent by the EtherCAT master to the EtherCAT slaves， for example, control word 0x6040， the input group PDO represents the process data sent by the EtherCAT slave to the master. Each slave may have multiple sets of PDO or a single set of PDO, where as shown in the figure above， the first set of input PDO and the first set of output PDO can be edited to add, edit, delete, etc. Take the example of adding a new PDO :



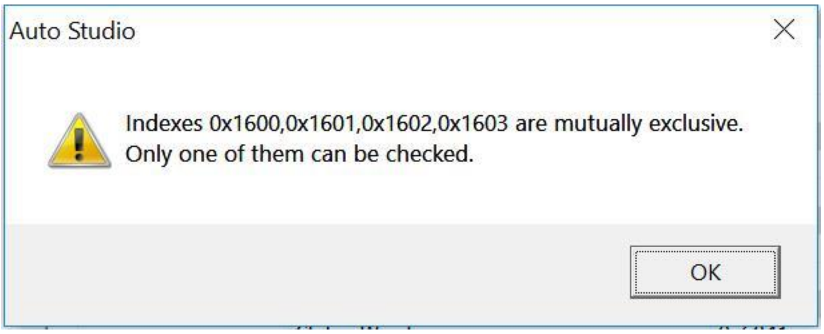
1、— Select a PDO in the first group

2、— Click to add

3、—Sclect 0x6060

4、—Click OK

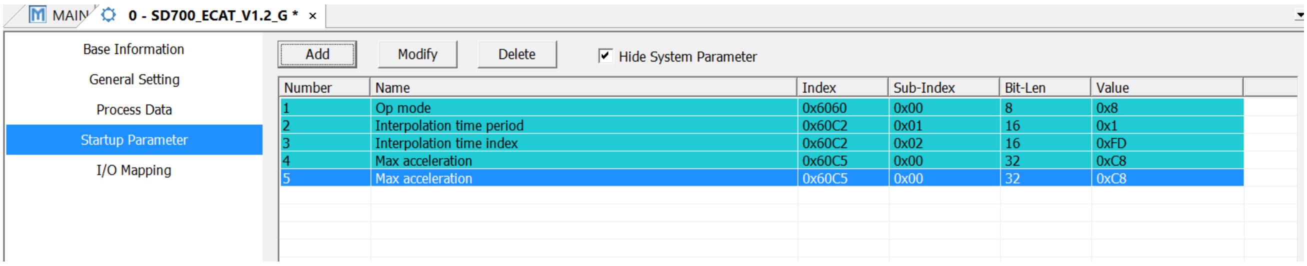
When a slave has multiple groups of PDO's, the groups of PDO's may have mutually exclusive relationships with each other. Such as the SD700. only one group can be selected at a time. This mutually exclusive relationship varies with the slave, and some devices can have multiple groups checked.



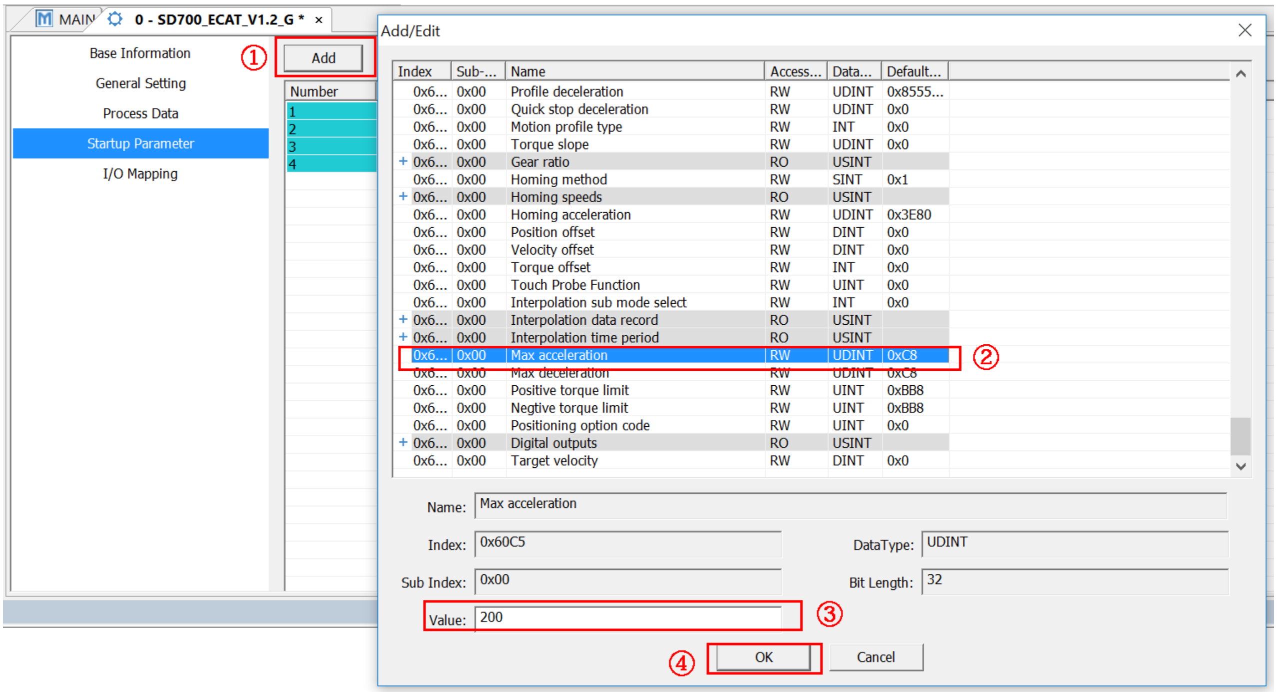
The master downloads the PDO configuration relationships to the EtherCAT slaves in the form of start-up parameters via PDO assignment and PDO mapping.

3.3 Start-up parameters

Start-up parameters mean that the PDO configuration information of the slave, the factory setting parameters and some parameters specified by the protocol (e.g. 402 protocol) are written to the slave by writing SDO when the slave is in the PreOP state. Take SD700 as an example:



In this interface users can add startup parameters as needed, for example, add the object dictionary 0x60C5 and change the value to 300.



1、— Click to add

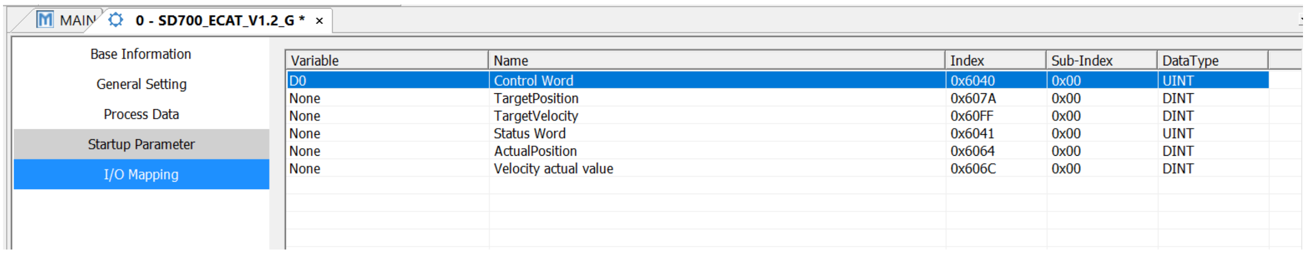
2、—Sclect 0x60C5

3、— Change the value to 300

4、—Click OK

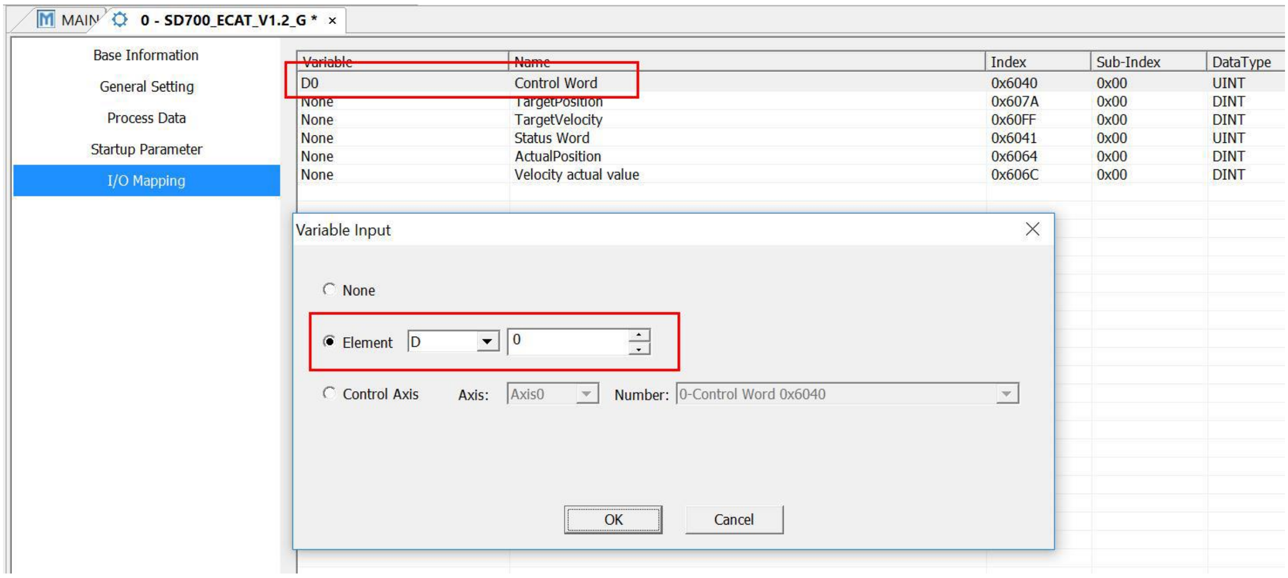
3.4 I/O mapping function

The EtherCAT slave modules can only be controlled by operating variables if the PDO data is connected to the PLC internal variables. The I/O function mapping interface is shown in the following figure:

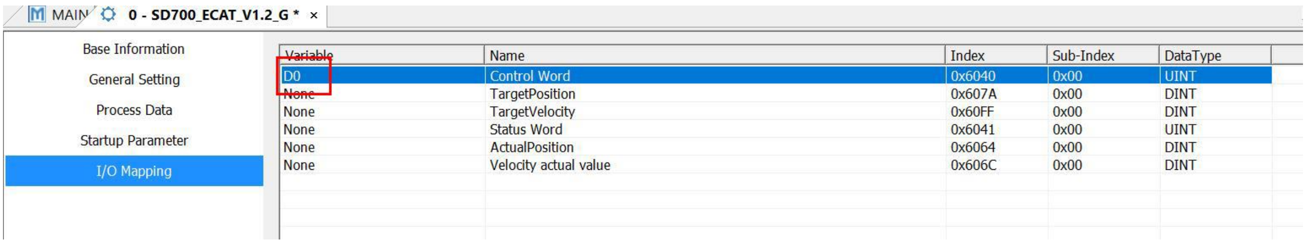


Associated Variables

1. Double-click the variable with the left mouse button to bring up the variable input setting interface, which can set the associated variables.



2. The interface after the association is completed is shown below.



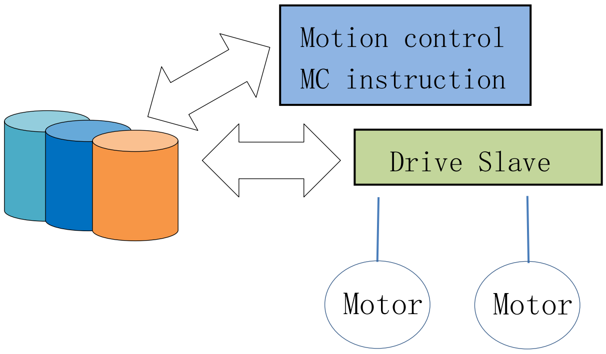
4. Motion Control

4.1 The Introduction of Motion Control Axes

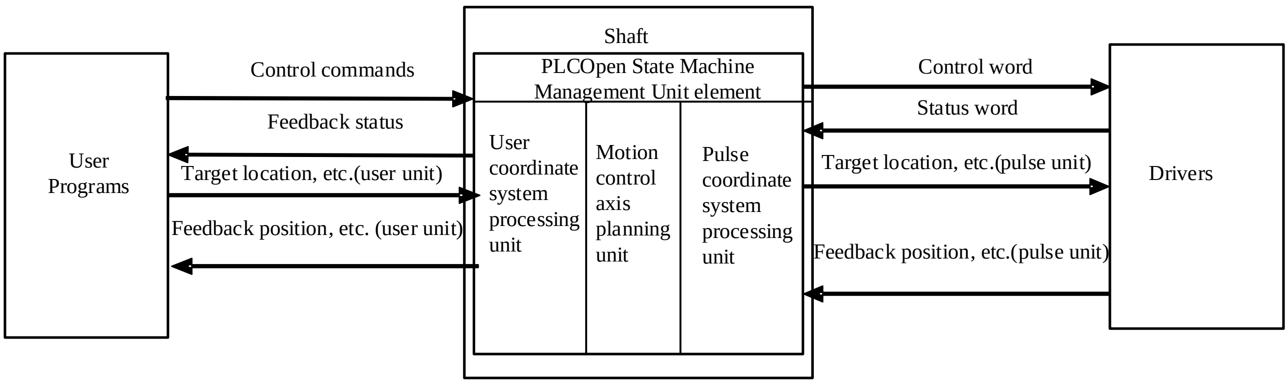
4.1.1 Overview

Basic composition and control logic

In VC5 motion control system, the object of motion control is called axis. The axes are the bridge between the drive and the PLC commands. The motion control axes of the VC5 are used to control EtherCAT bus drives compliant with the 402 protocol, as well as the local high-speed pulse outputs and high-speed pulse inputs of the VC5.



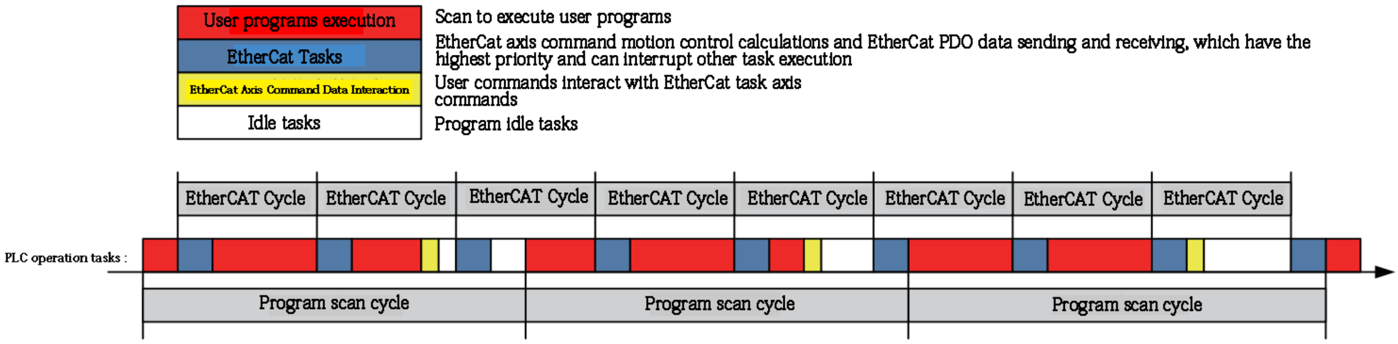
Inside the PLC, the basic composition and processing logic of the axes are as follows：



Command Type

Classification	Command
Control category	Such as MC_Power, enable control command
Sports category	Such as MC_Jog, point motion command
Status category	Such as MC_ReadStatus, the read axis status command

The PLC first parses these above commands into control commands, target positions (user units) and other parameters to the axis structure， after receiving these commands, the axis structure is converted into a command form conforming to the Cia402 protocol specification through coordinate system conversion, PLCOpen state machine management, and motion planning unit processing， A dictionary of related objects that control the servo, such as the enable of the servo via 0x6040 and the motion of the drive via 0x607a. The driver feeds the relevant status to the axis structure via the Cia402 protocol, e.g. the current status via 0x6041 and the current position via 0x6064. The structure receives these states and feeds them to the relevant commands through the internal control logic. The axis control commands are parsed in PLC tasks, the PLCOpen state machine and path planning are executed in the EtherCAT task, and the timing of the two tasks is scheduled as follows.



Axis Type

Axis Type	Description
Bus servo axis	Axes controlled with EtherCAT slave servo drives. If the dummy axis mode is not enabled, this axis is used to assign it to the actual servo drive to be used. The bus servo axis supports several basic modes of control: torque, point, speed, and home return.
Local pulse axis	Axes controlled by pulse drivers using local high-speed IO control. VC5 allows setting 4 local pulse axes, Y0/Y1, Y2/Y3, Y4/Y5, Y6/Y7. Each pulse output channel can be optionally set to pulse + direction or CW/CCW. A maximum of two probe terminals per pulse output channel. The local pulse axis supports point, speed, and home return control in several basic modes, and does not support torque mode.
Bus encoder axis	Reserved
Local encoder axis	High-speed counter 0~7

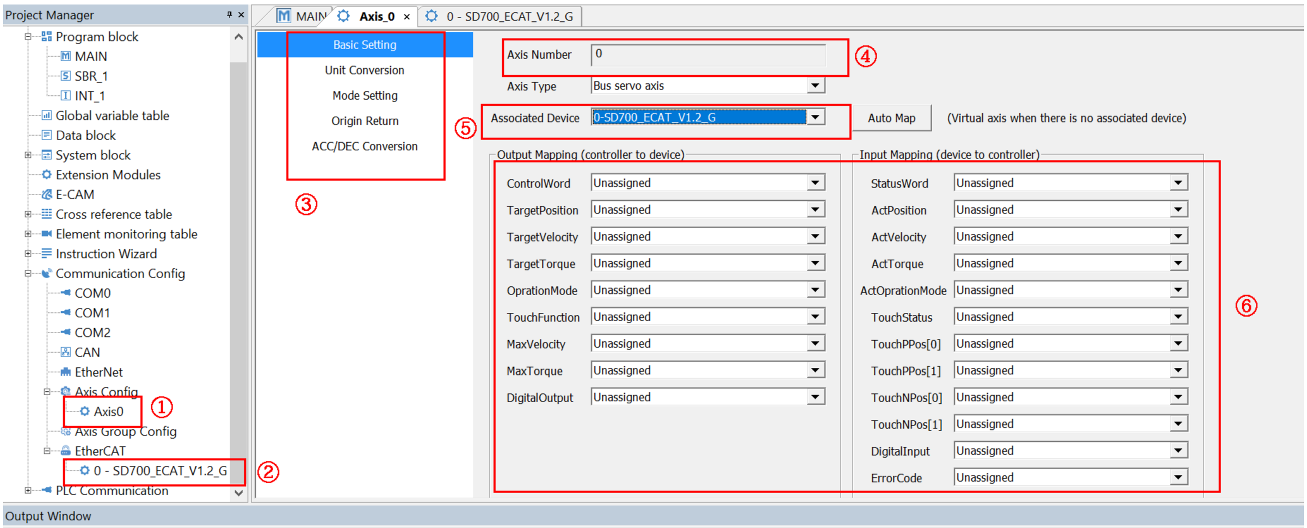
To completely describe the properties of the axes, monitor their status, and control their motion, the axes are divided into three parts:

Axis Structure	Function Description.
Shaft configuration parameters	Used to configure individual parameters of the axis, such as gear ratio, home return type, encoder mode, etc.
Axis system variables	Monitoring of axis operating status and abnormal information, such as current position, axis fault codes, etc.
Axis control commands	In the user program, the MC motion control commands are used to execute axis motion control.

	The axis control commands are divided into control category (such as MC_Power), motion category (such as MC_Jog) and status category (MC_ ReadStatus).

Shaft configuration parameters

In the project, the axis configuration screen is as follows.



1、— Motion control axes

2、— EtherCAT bus drivers

3、— List of axis configuration and monitoring options

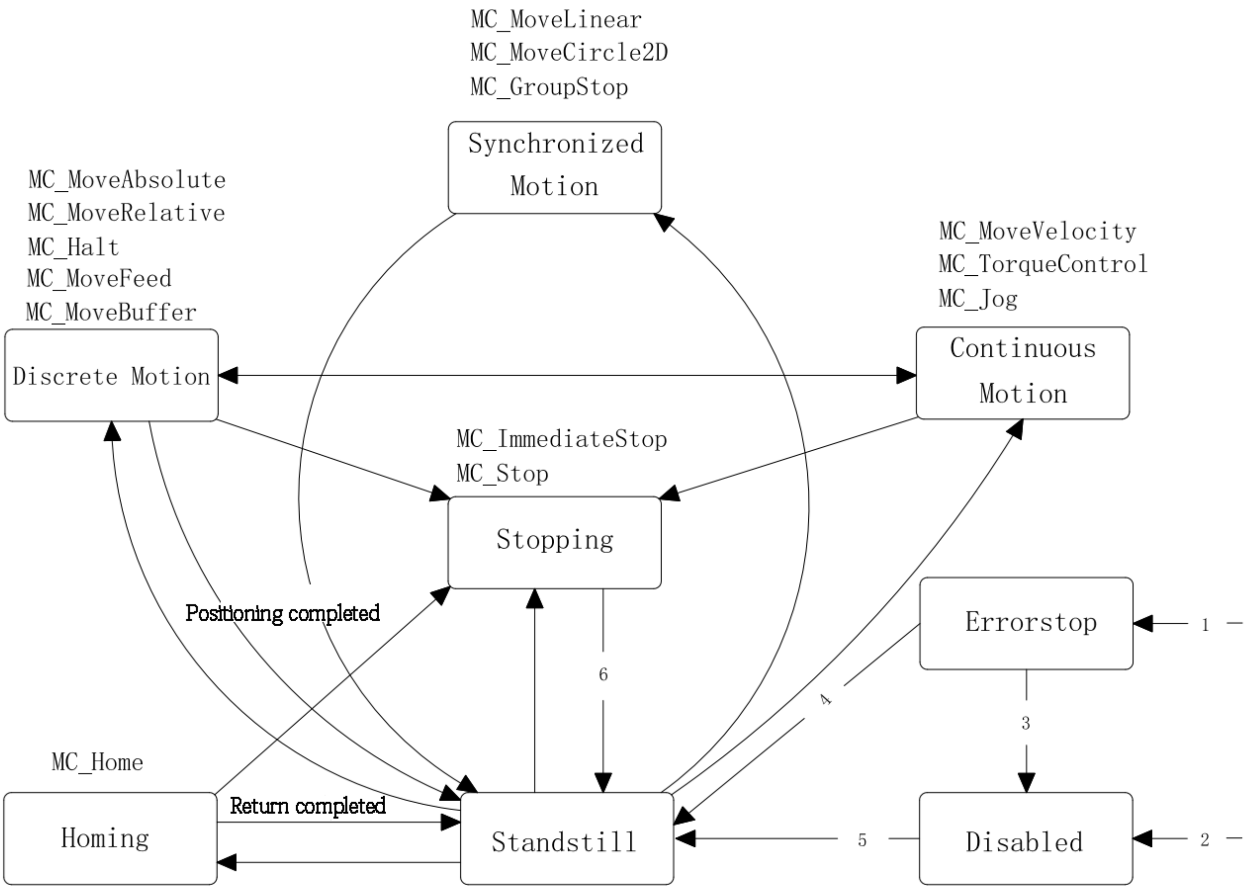
4、— Axis number (It's the unique access ID of the axis)

5、— Associated entity drivers

6、— Detailed parameter settings

4.1.2 PLCOpen state machine

VC5 the state and motion of the axes are managed based on the PLCOpen state machine, and different functions are accomplished in each different state. The state transition diagram is as follows:



The detailed description is below：

Status	Function description
Disable	Not enabled
Error stop	Fault shutdown status
Standstill	Enable state
Homing	Origin regression state
Stopping	Stop status
Discrete Motion	Discrete motion
Continuous Motion	Continuous grain running state
Synchronized Motio	Synchronize the health status

State migration conditions：

Transform action	Conversion condition
1	Enters this state as soon as the fault detection logic of the shaft detects a fault
2	When the shaft is faultless and MC_Power.Enable=FALSE
3	When the call MC_Reset reset axis fails and MC_Power.Status=FASLE
4	When the call MC_Reset reset axis fails and MC_Power.Status=TRUE
5	When MC_Power.Enable=TRUE and MC_Power.Status=TRUE
6	MC_Stop (MC_ImmediateStop). Done=TRUE MC_Stop (MC_ImmediateStop). When Execute=FALSE

4.1.3 The units of the axis

Two units are used in the shaft structure, the user unit and the pulse unit.

The units of the axis	Functional description
User Units	The unit of measurement used on the command side in millimeters, centimeters, angles, etc. is called the user unit, usually expressed in Unit. The user coordinate system is divided into a linear coordinate system and a rotating coordinate system according to the different operating conditions. A linear coordinate system typically contains a zero point, with forward motion when the target position increases and the target position decrements as reverse motion. Linear coordinate systems can set positive and negative soft limits. The rotation coordinate system consists of a zero point and a rotation period in which the target position increases clockwise and the target position decreases to counterclockwise.
Pulse units	The unit used on the driver side is measured in the number of pulses, usually expressed in pluse. The driver usually contains two parameters: pulse zero and The number of pulses of the encoder when the motor turns one rotation The process of finding and setting the zero point of a servo motor pulse is called origin regression. The number of pulses in one revolution of the servo motor is often used to calculate the distance traveled by the work piece.

4.1.4 Axis configuration parameters

Classification	Content	Bus servo axis	Local pulse axis
Basic setting	Axis number	√	√
	Shaft type	√	√
	Automatic mapping	√	×

	PDO	√	×
	Reverse	√	√
Unit conversion	Command pulse number of Motor/Encoder rotation circle	√	√
	Working stroke of Motor/Encoder rotation circle	√	√
	Wheel ratio molecule	√	√
	Denominator of wheel ratio	√	√
Mode/Parameter setting	Encoder mode	√	√
	Linear/Rotation mode setting	√	√
	Software limit	√	√
	Software error response	√	√
	Following error	√	√
	Shaft speed setting	√	√
	Torque limit	×	×
	Probe setup	×	√
	Output setting	×	√
	Hardware limit section	√	√
	No ErrorStop status after magnetic limit	√	√
Origin return setting	Origin signal	√	√
	Positive limit	√	√
	Stop	√	√
	Z signal	√	√
	Origin return direction	√	√
	Origin input detection direction	√	√
	Origin return list	√	√
	Origin return speed	√	√
	Approach speed of origin return	√	√
	Origin return acceleration	√	√
	Origin return timeout	√	√
	Limit terminal setting	×	√

	Positive terminal setting	×	√
	Origin signal setting	×	√

4.1.5 Axis system variables

Axis system variables	Description
_SYS_MC_AXIS	Data structure of the motion control axes
_SYS_ECAT_Master	Ethercat master status information
_SYS_ECAT_SLAVE	EtherCAT slave status information

4.1.5.1 _SYS_MC_AXIS Information about motion control axes

64 axes visible on the software (shown as axis 0)

Name	Date type	R/W type	Description	Operating parameter groups
_McAxis[0]	_sMCAXIS _INFO[64]	—	—	—
_McAxis[0].dPulsesPreCycle	DINT	WR	Number of pulses in one revolution of motor/encoder	1
_McAxis[0].fDistancePreCycle	REAL	WR	Movement of the table in one rotation	
_McAxis[0].dNumerator	DINT	WR	Gear ratio molecule	
_McAxis[0].dDenorminator	DINT	WR	Gear ratio denominator	
_McAxis[0].bDirection	BOOL	WR	Direction	
_McAxis[0].bSoftLimitEnable	BOOL	WR	Software limited position enable	2
_McAxis[0].fPLimit	REAL	WR	Positive limit values in linear mode	
_McAxis[0].fNLimit	REAL	WR	Negative limit values in linear mode	
_McAxis[0].iLineRotateMode	INT	WR	Linear/rotary mode selection 0: Linear mode 1: Rotary mode	
_McAxis[0].fRotation	REAL	WR	Rotation period in rotary mode	
_McAxis[0].EncodeMode	INT	WR	Encoder mode (valid for bus servo axes) 1: Absolute mode 0: Incremental mode	3
_McAxis[0].iHomeMethod	INT	WR	Home return mode	4
_McAxis[0].fHomeVelocity	REAL	WR	Home return speed	
_McAxis[0].fHomeApproachVelocity	REAL	WR	Home return approach speed	

_McAxis[0].fHomeAcceleration	REAL	WR	Home return acceleration	
_McAxis[0].iHomeTimeOut	INT	WR	Home return timeout	
_McAxis[0].bPLimitTerminalPolarity	BOOL	WR	Positive limit input polarity (local pulse axis valid)	5
_McAxis[0].bNLimitTerminalPolarity	BOOL	WR	Negative limit input polarity (local pulse axis valid)	
_McAxis[0].bHomeTerminaPolarity	BOOL	WR	Home input polarity (local pulse axis valid)	
_McAxis[0].iPLimitType	INT	WR	Positive limit position input type: 0: X, 1: M, 2: S (local pulse axis valid)	
_McAxis[0].iPLimitID	INT	WR	Positive limit position input number: X0~7 / M0~M range / S0~S range (valid for local pulse axis)	
_McAxis[0].iNLimitType	INT	WR	Negative limit position input type: 0: X, 1: M, 2: S (local pulse axis valid)	
_McAxis[0].iNLimitID	INT	WR	Negative limit position input number (valid for local pulse axis)	
_McAxis[0].iHomeInType	INT	WR	Home position input type: 0: X, 1: M, 2: S (local pulse axis valid)	
_McAxis[0].iHomeInID	INT	WR	Home position input number (valid for local pulse axis)	
_McAxis[0].iEncoderInType	INT	WR	Local encoder input type (Local encoder valid)	
_McAxis[0].iEncoderRstInEn	INT	WR	Local encoder reset input valid (Local encoder valid)	
_McAxis[0].iEncoderRstInID	INT	WR	Local encoder reset input selection (Local encoder valid)	
_McAxis[0].iEncoderEnInEn	INT	WR	Local encoder enable input active (Local encoder active)	
_McAxis[0].iEncoderEnInID	INT	WR	Local encoder enable input selection (Local encoder active)	
_McAxis[0].iEncoderPreSetInEn	INT	WR	Local encoder preset input active (Local encoder active)	
_McAxis[0].iEncoderPreSetInID	INT	WR	Local encoder preset input selection (Local encoder valid)	
_McAxis[0].iPluseMethod	INT	WR	Pulse output mode (local pulse axis active)	7
_McAxis[0].bTouchProbeEn0	BOOL	WR	Probe 0 input enable (local pulse axis valid)	
_McAxis[0].iTouchProbeID0	INT	WR	Probe 0 input number: 0~7 X0~7 (Local pulse axis valid)	
_McAxis[0].bTouchProbeEn1	BOOL	WR	Probe 1 input enable (local pulse axis valid)	

_McAxis[0].iTouchProbeID1	INT	WR	Probe 1 input number: 0~7 X0~7 (Local pulse axis valid)	
_McAxis[0].bCmpEnable	BOOL	WR	Compare output enable (local pulse axis valid)	8
_McAxis[0].iCmpOutID	INT	WR	Comparison output port number: 0~7: Y0~7 (Local pulse axis valid)	
_McAxis[0].iCmpOutUnit	INT	WR	Compare output units (valid for local pulse axis)	
_McAxis[0].iCmpOutWidth	INT	WR	Compare output width (valid for local pulse axis)	
_McAxis[0].fErrorStopDeceleration	REAL	WR	Axis fault deceleration	9
_McAxis[0].fFollowErrorWindow	REAL	WR	Follow error threshold	
_McAxis[0].fMaxVelocity	REAL	WR	Maximum velocity	
_McAxis[0].fMaxJerkVelocity	REAL	WR	Maximum acceleration	
_McAxis[0].fMaxAcc	REAL	WR	Maximum acceleration	
_McAxis[0].iMaxTorque	INT	WR	Maximum torque	
_McAxis[0].dConfigReserved[16]	DINT	WR	Reserved	
_McAxis[0].iMapped	INT	RO	Axis parameter mapping flags	
_McAxis[0].iType	INT	RO	Axis type	
_McAxis[0].iSlave	INT	RO	Axis mapping number	
_McAxis[0].iVirtualAxis	INT	RO	Virtual Axis Flag	
_McAxis[0].iEnableStatus	INT	RO	Axis enable status	
_McAxis[0].iAlmStatus	INT	RO	Axis alarm status	
_McAxis[0].iAxisOprationStatus	INT	RO	Axis control mode status	
_McAxis[0].iCheckDoingStatus	INT	RO	Axis current motion status 0: stopped, 1: running	
_McAxis[0].iInterpNum	INT	RO	Interpolation channel number	
_McAxis[0].iInterpBitType	INT	RO	Control mode bit empty flag bit0:P_Task bit1:S_Task. bit2:F_Task bit3:cam	
_McAxis[0].dCommandPulse	DINT	RO	Current output command position	
_McAxis[0].dDeltaCommand	DINT	RO	Distance between the encoder feedback position and the command position	
_McAxis[0].dEncoderPos	DINT	RO	Encoder feedback position	
_McAxis[0].dStatusReserved[16]	DINT	RO	Reserved	

4.1.5.2 _SYS_ECAT_MASTER Master status information

Name	Date type	R/W type	Description
_ECATMaster.bMasterEnableState	BOOL	RO	Master enable status

_ECATMaster.bLinkState	BOOL	RO	Master connection status
_ECATMaster.dCycleTime	DINT	RO	Master synchronization period
_ECATMaster.dTaskExeTime	DINT	RO	Master task execution time
_ECATMaster.iMasterState	INT	RO	Master bus status
_ECATMaster.iSlaveNumber	INT	RO	Number of master-connected slaves
_ECATMaster.iDcSlaveNumber	INT	RO	Number of master-connected slaves supporting dc
_ECATMaster.iLossPackeCounter	INT	RO	Master packet loss accumulation counter
_ECATMaster.dPdoInLength	DINT	RO	Master PDO input length
_ECATMaster.dPdoOutLength	DINT	RO	Master PDO output length
_ECATMaster.iCycleJitter	INT	RO	Master sync cycle jitter
_ECATMaster.dReserved[32]	DINT	RO	Reserved

4.1.5.3 _SYS_ECAT_SLAVE Slave status information

72 axes visible on the software (shown as axis 0)

Name	Date type	R/W type	Description
_ECATSlave	_sECTSLV_INFO[72]		EtherCAT slave operating status
_ECATSlave[0]	sECTSLV_ INFO		
_ECATSlave[0].iState	INT	RO	Current bus status of the slave
_ECATSlave[0].iALstatescode	INT	RO	AL status code
_ECATSlave[0].iConfigaddr	INT	RO	Configuration address
_ECATSlave[0].iAliasaddr	INT	RO	Slave Alias
_ECATSlave[0].dEep_man	DINT	RO	Slave vendor ID
_ECATSlave[0].dEep_id	DINT	RO	Slave device ID
_ECATSlave[0].iItype	INT	RO	Interface type
_ECATSlave[0].iDtype	INT	RO	Device type
_ECATSlave[0].iObits	INT	RO	Number of PDO output bits

_ECATSlave[0].iObytes	INT	RO	Number of PDO output bytes
_ECATSlave[0].iOstartbit	INT	RO	PDO Output Start Bits
_ECATSlave[0].iIbits	INT	RO	Number of PDO input bits
_ECATSlave[0].iIbytes	INT	RO	Number of PDO input bytes
_ECATSlave[0].iIstartbit	INT	RO	PDO input start bits
_ECATSlave[0].iHasdc	INT	RO	Support DC
_ECATSlave[0].iPtype	INT	RO	PHY interface type
_ECATSlave[0].iTopology	INT	RO	Topology
_ECATSlave[0].iActiveports	INT	RO	Active port
_ECATSlave[0].iParent	INT	RO	Slave number of parent
_ECATSlave[0].iParentport	INT	RO	Parent connection number
_ECATSlave[0].iEntryport	INT	RO	Entry port
_ECATSlave[0].iSlotConfig	INT	RO	Slot active configuration
_ECATSlave[0].iRdSlotsNum	INT	RO	Number of slots to read
_ECATSlave[0].dRdSlotsIds[32]	DINT	RO	Slot ID list
_ECATSlave[0].dReserved[32]	DINT	RO	Reserved

4.1.6 List of axis commands

VC5 the list of supported single-axis control commands is as follows：

Directives	Functional description
MC_Power	Enable control commands
MC_Reset	Reset fault command
MC_ReadStatus	Read axis status commands
MC_ReadAxisError	Read axis error commands
MC_ReadDigitalInput	Read digital input commands
MC_Move	Positioning commands
MC_ReadPosition	Read position commands
MC_ReadVeloccity	Read speed commands
MC_ReadActualTorque	Read the actual torque commands
MC_SetPosition	Set the location directive
MC_TouchProbe	Probe commands
MC_MoveRelative	Relative positioning commands
MC_MoveAbsolute	Absolute positioning commands
MC_MoveVelocity	Speed command
MC_Jog	Jog motion command
MC_RunStatus	Read the axis operating status commands
MC_Home	Origin regression command
MC_Stop	Stop command
MC_Halt	Pause the command
MC_MoveSuperImposed	Displacement overlay command
MC_ FollowVelocity	Motion speed overlay command

MC_MoveFeed	Interrupt the fixed-length command
MC_MoveBuffer	Multi-segment position directives
MC_Linear	Straight-line interpolation commands
MC_Circle_CW	Clockwise circular arc interpolation commands
MC_Circle_CCW	Counterclockwise circular arc interpolation commands
MC_SyncTorqueControl	Synchronous torque control command
MC_MoveVelocityCSV	Speed command basing on adjustable CSV pulse width
MC_SyncMoveVelocity	Synchronous speed control command basing on adjustable CSV pulse width
MC_MoveThreeDimensionalCircular	3D circular arc interpolation command
MC_MoveSpiral	Helical interpolation command

The list of axis group control commands supported by VC5 is as follows:

Command	
MC_MoveLinear	Linear interpolation
MC_MoveCircle	Circular arc interpolation
MC_GroupStop	Stop axis group running
MC_GroupPause	Pause axis group running
MC_AddAxisToGroup	Add an axis to an axis group
MC_RemoveAxisFromGroup	Remove from axis group
MC_PathAdd	Path added
MC_PathMov	Path Movement
MC_SetForwardlookingPara	Forwardlooking parameter set command

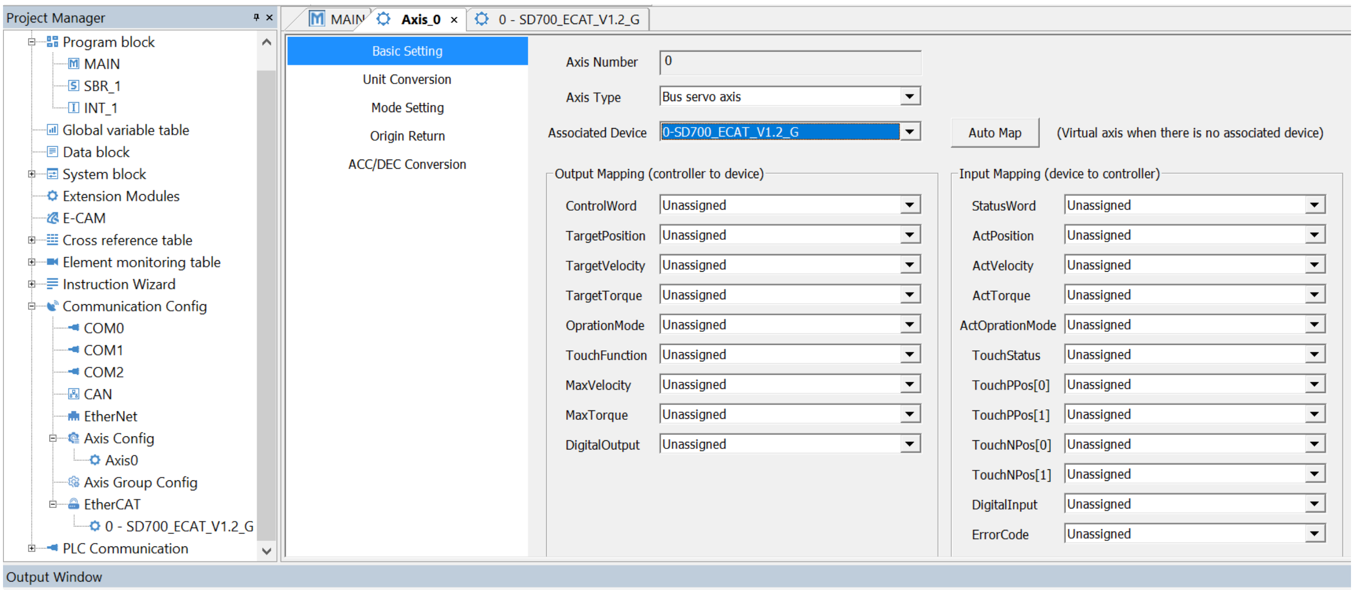
4.2 Motion Control Axis Configuration

4.2.1 Bus servo axis and local pulse axis comparison

Project	This pulse output	EtherCAT Bus driver
The shaft types are different	Local pulse axis	Bus servo axis
The output device is different	You need to set up the local IO terminals, and each of the two forms a group, respectively Y0/ Y1、Y2/Y3、Y4/Y5 and Y6/Y7	PDO needs to be configured and mapped to the loop variable of the axis
Pulse output form	Pulse + direction, CW/CCW and AB phase are supported, and you need to select in the "Mode Settings" -> "Pulse Output Settings" field	No setup is required
Probe function	Two probes are supported, and one of the X0-X7 can be selected for each probe terminal. This function need to select in the "Mode Settings" -> "Pulse Output Settings" field	Probe terminals need to be configured
The origin returns to the location	Support the return method other than the Z signal specified in the 402 protocol. The limit signal and origin signal of the local pulse output axis can be selected from the origin return setting interface.	Support the 1-35 zero-back mode setting specified in the 402 protocol, and the limit and origin signals need to be set on the driver side.

4.2.2 Basic settings

The basic setup interface for axes is used to set the type of axis, select functions such as entity drive devices, etc. The basic settings interface is shown in the following figure：



Axis number: Each axis is assigned a separate number, the range is 0-71, can not be modified manually. The axis number can be accessed as an input parameter to the MC command and is unique.

Axis Type: The options for the shaft type are Bus servo axis, Local Pulse Axis, Bus Encoder Axis, and Local Encoder Axis.

Associated devices: Valid only in bus servo axis and local pulse axis modes. If it is a bus servo axis, it is used to select the EtherCAT servo drive; In the case of a local pulse axis, it is used to select a local high-speed output terminal. VC5 has four sets of high-speed output terminals Y0/Y1, Y2/Y3, Y4/Y5, and Y6/Y7 to choose from.

Cyclic variables: Valid only in bus encoder axis and bus servo axis modes. EtherCAT slaves communicate periodically based on PDO, and the axes are connected to the object dictionary of the EtherCAT slave via cyclic variables. When automatic mapping is selected, the mapping process is automatically assigned and cannot be configured manually.

Bus servo axis cycle variables:

The list of variables is as follows, see Standard 402 Protocol for details on the meaning of the object dictionary.

A list of variables

Loop variables	Object dictionary	Function
Controlword	0x6040	Control words
TargetPosition	0x607A	The target location value
TargetVelocity	0x60FF	The target speed value
Set torque	0x6071	The target torque
Modes of operation	0x6060	Control mode, the setting range is 6: Origin regression mode 8: Periodic Synchronous Position Mode (CSP) 10: Periodic synchronous torque mode (CST)
Touch probe function	0x60b8	Probe control word
Physical outputs	0x60fe:1	Digital output
StatusWord	0x6041	Status word
ActPosition	0x6064	The actual location value
ActVelocity	0x606C	Actual speed feedback value
Torque actual value	0x6077	The actual moment value
Modes of operation display	0x6061	The current control mode
Digital inputs	0x60fd	Digital input terminal status, functions as follows：

		Bit2: Origin switch Bit1: Forward limit switch Bit0: Reverse limit switch
Touch Probe Status	0x60b9	Probe status
Touch Probe Pos1 Pos Value	0x60ba	Probe 1 ascends to the position along the edge
Touch Probe Pos1 Neg Value	0x60bb	Probe 1 descends along the position
Touch Probe Pos2 Pos Value	0x60bc	Probe 2 ascends along the position
Touch probe 2 falling edge	0x60bd	Probe 2 descends along the position
Error code	0x603f	Drive fault code

The parameters that need to be set for unit conversion

Relevant parameters

The parameter name	Function
Number of pulses for one revolution of motor/encoder	Depending on the encoder resolution, set the number of pulses for the motor to turn 1 revolution
The amount of movement of the table rotated in one turn	The amount of movement of the work piece 1 turn on the side
Gear ratio numerator	Gear ratio numerator
Gear ratio denominator	Gear ratio denominator

The bus driver (local pulse axis) uses pulse units when controlling the motor, and the motion control command side uses common units of measurement such as millimeters, degrees, inches, etc., which we call user units (Units). The two units are converted to and from within the configuration parameter axis.

The formula for calculating from user units to pulse units is as follows：

$$\text{Number of pulses (pulse)} = \frac{\text{Motor/encoder number of pulses for one revolution [DINT] * Gear ratio numerator [DINT]}}{\text{Motor / encoder movement for one revolution [REAL] * Gear ratio denominator [DINT]}} * \text{Moving distance [Uint]}$$

The association of the unit conversion setting with the command unit, in the user unit:

The location unit is Unit,

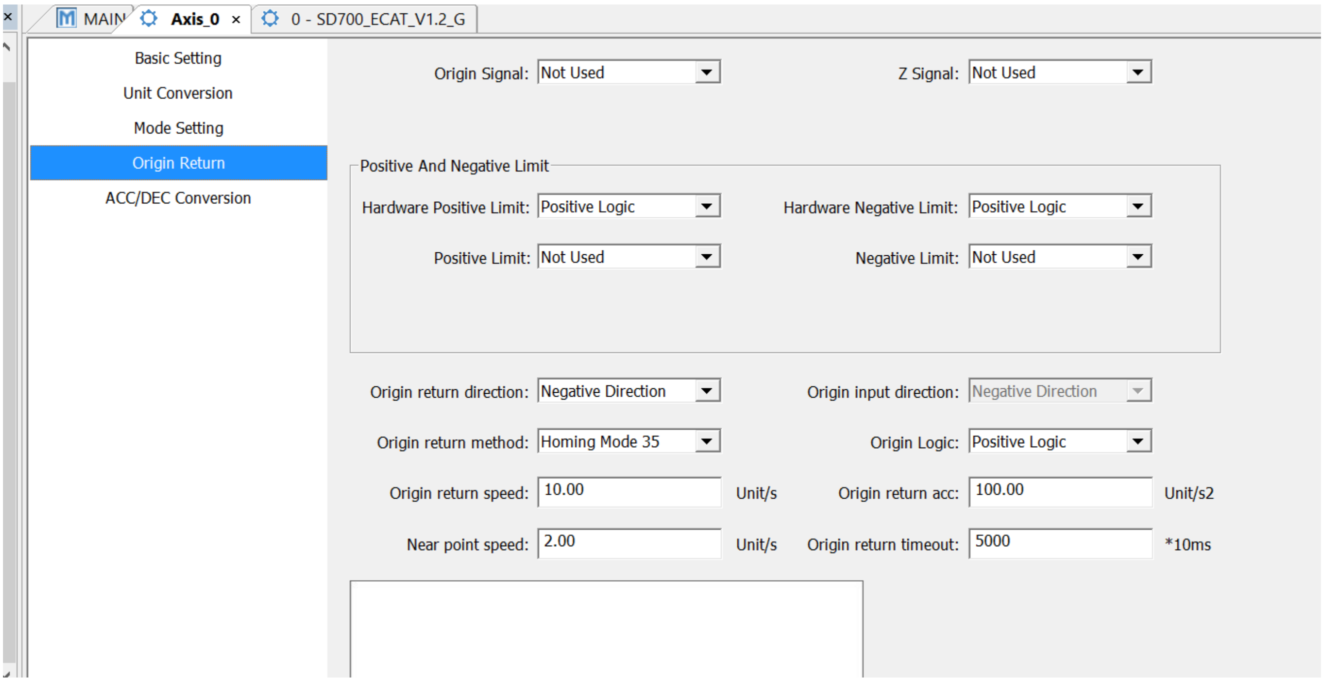
Speed units are Unit/s,

The acceleration unit is Unit/s².

4.2.3 Mode settings

4.2.3.1 Configuration interface

The basic settings interface is as follows, the selected axis type is different, the visible parameter list is also different, subject to the actual display of the software.



4.2.3.2 Encoder mode

The encoder mode is only valid in bus servo axis mode and is used with incremental encoder type servo drives and absolute encoder servers, please select according to the type of servo drive actually used. The PLC is handled as follows:

Incremental mode

The PLC side does not take into the account to increase in the number of laps caused by the overflow of the servo drive encoder 32-bit counter. PLC does not save the current position of the encoder during power failure and restart, and after re-powering up, the current position of the shaft is calculated only according to the position of the single turn fed by the servo drive.

Absolute mode

The PLC side takes into the account to increase in the number of laps caused by the overflow of the servo drive encoder 32-bit counter. PLC saves the current position of the encoder feedback when the power is disconnected and restarted, and after the power is re-energized, the encoder position of the shaft and the position of the servo drive feedback saved inside the PLC are read inside the PLC during the initialization process, and the position of the feedback of the servo drive is calculated the current absolute position of the outlet.

4.2.3.3 Mode settings

Depending on the actual operating conditions, the motion control axis can be set to linear mode and ring mode.

Linear mode

Linear mode is typically used in devices with a range of mechanical motion in the X-Y linear coordinate system.

Linear mode usually contains a zero point.

An increase in the feedback position during motion indicates forward motion, and vice versa.

It is allowed to set forward software limits and reverse software limits, and when the software limit is enabled, the axle can only move within the limit range.

Absolute positioning mode: when the target position is greater than the starting position, the distance between the target position and the starting position is moved forward; When the target location < the starting position, the distance from the starting position to the target location is moved in reverse.

Relative positioning mode: when the target displacement is greater than 0, the distance of the forward movement of the target displacement, when the target displacement is less than 0, the distance of the reverse motion "target displacement".

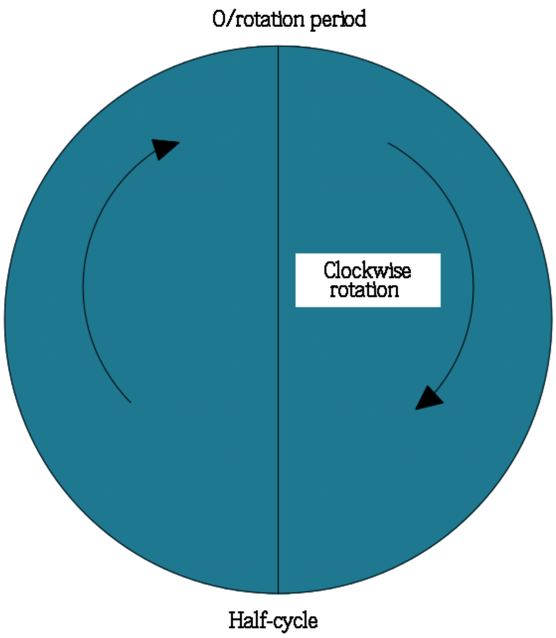
Speed class command processing method in linear mode: the target speed is greater than 0 for forward motion, and the target speed is less than 0 for reverse motion.

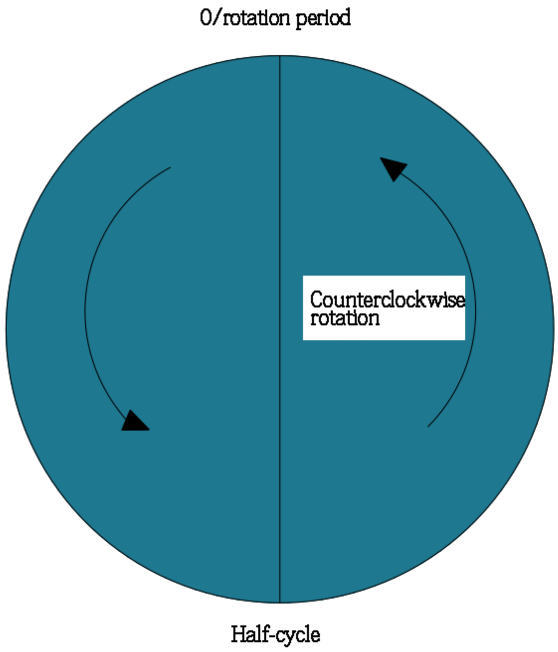
Ring mode

The ring mode is a mode in the form of a change counter that repeats infinite counts over a set range. Usually used in turntables or reels, etc.

Torus mode usually consists of a zero point and a rotation period. The feedback position range of the ring counter is 0< = the feedback position < rotation period.

In ring mode, it is considered clockwise if the feedback position increases, and counterclockwise if the feedback position decreases.

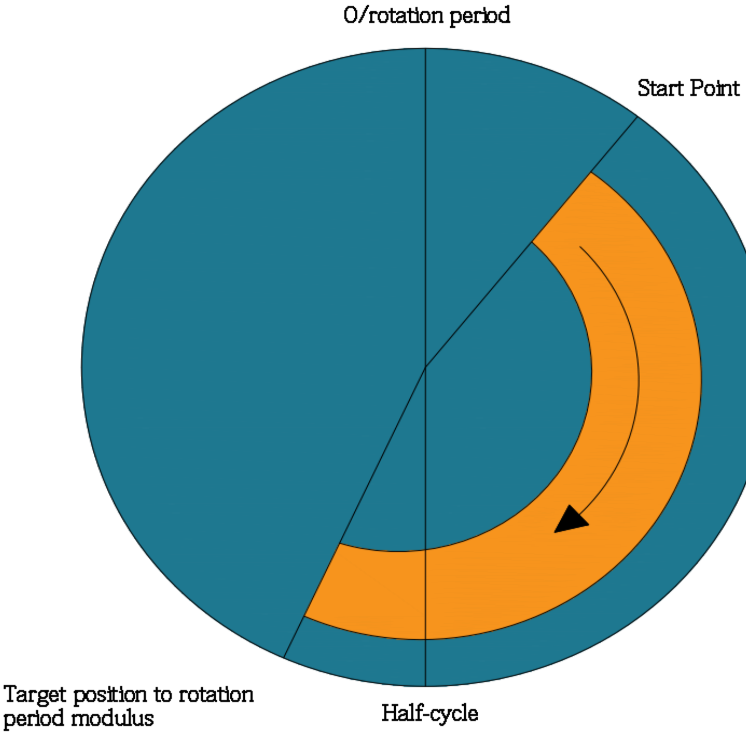




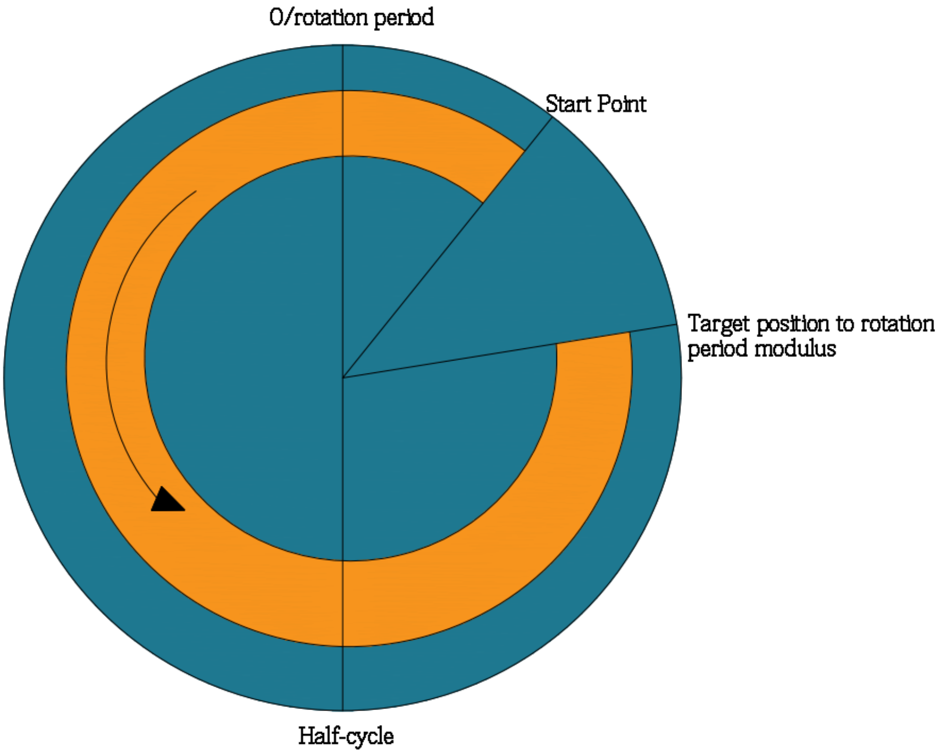
There are no software limits in ring mode.

Relative positioning processing mode: the distance of the target displacement is moved clockwise if the target displacement is greater than 0, and the distance of the target displacement | the target displacement is moved counterclockwise |.

Absolute positioning processing: forward: first take the modulus of the rotation period of the target position, and then move the axis clockwise from the starting position to the target position.



Reverse: First take the modulus of the rotation period from the target position, and then move the axis counterclockwise from the starting position to the target position.



Shortest distance: First take the target position to the rotation period to obtain the target position 1, and then calculate the displacement from the starting point clockwise to the target position, if the displacement is less than or equal to half a period then move clockwise, otherwise use counterclockwise movement to the target position.

Current direction: Moves to the target position in the direction of the most recent movement of the axis, and forward to the target position if it is the first time on power. Speed command processing method in ring mode: the target speed is greater than 0 and the target speed is less than 0 when it moves counterclockwise.

4.2.3.4 Abnormal deceleration

During the operation of the shaft, if the logic failure of the motion command itself causes the shaft to switch to the errorstop state, the shaft will do T-type deceleration according to the deceleration set by the shaft fault deceleration, and the shaft will not enter the errorstop state until the deceleration reaches 0.

4.2.3.5 Follow error

During the execution of the positioning and speed commands, the servo drive actually works in CSP (Periodic Synchronous Position) mode, and the planning of the position curve is done on the PLC side. The PLC sends the target position to the servo drive through the 0x607A, the servo drive the servo motor movement, the position of the motor encoder is fed back to the PLC through the 0x6064, and the difference between the 0x607A and the 0x6064 is generated due to the servo drive and the motor itself. This difference is converted into user units and is called a follow error. Set the maximum follow error in VC5. If the absolute value of the following error of the axis exceeds the maximum following error, the axis reports that the following error is too large and enters the error state.

4.2.3.6 Software limits

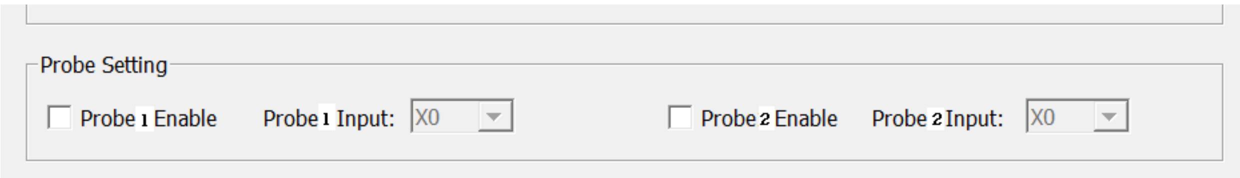
If the soft limit is effective, during the operation of the axis, the absolute position of the axis will be detected at all times from the current speed and then decelerated to 0 in accordance with the set limit., and if the absolute position of the shaft exceeds the limit range, the shaft will execute the soft limit deceleration algorithm and interrupt the positioning or speed commands currently being executed. The soft limit in origin regression and torque mode is invalid.

4.2.3.7 Axis speed settings

VC5It allows you to set three parameters: maximum speed, maximum acceleration, and maximum adding acceleration. When the target speed, acceleration, deceleration and other parameters in the positioning command or speed command exceed the speed limit value, the relevant command reports a fault and the axis enters the Errorstop state. In the bus servo axis, the maximum speed is also converted into pulse units by the unit conversion function and written to the object dictionary 0x607f of the servo drive via the start-up parameters.

4.2.3.8 Probe settings

The local pulse axis enables the probe terminals via probe settings.
In the local pulse axis, each shaft can be configured with up to two probe terminals. The probe terminal source can be selected X0-X7.
When the probe terminals are enabled, the local pulse axis can use probe commands and interrupt fixed-length commands.



4.2.3.9 Pulse output settings

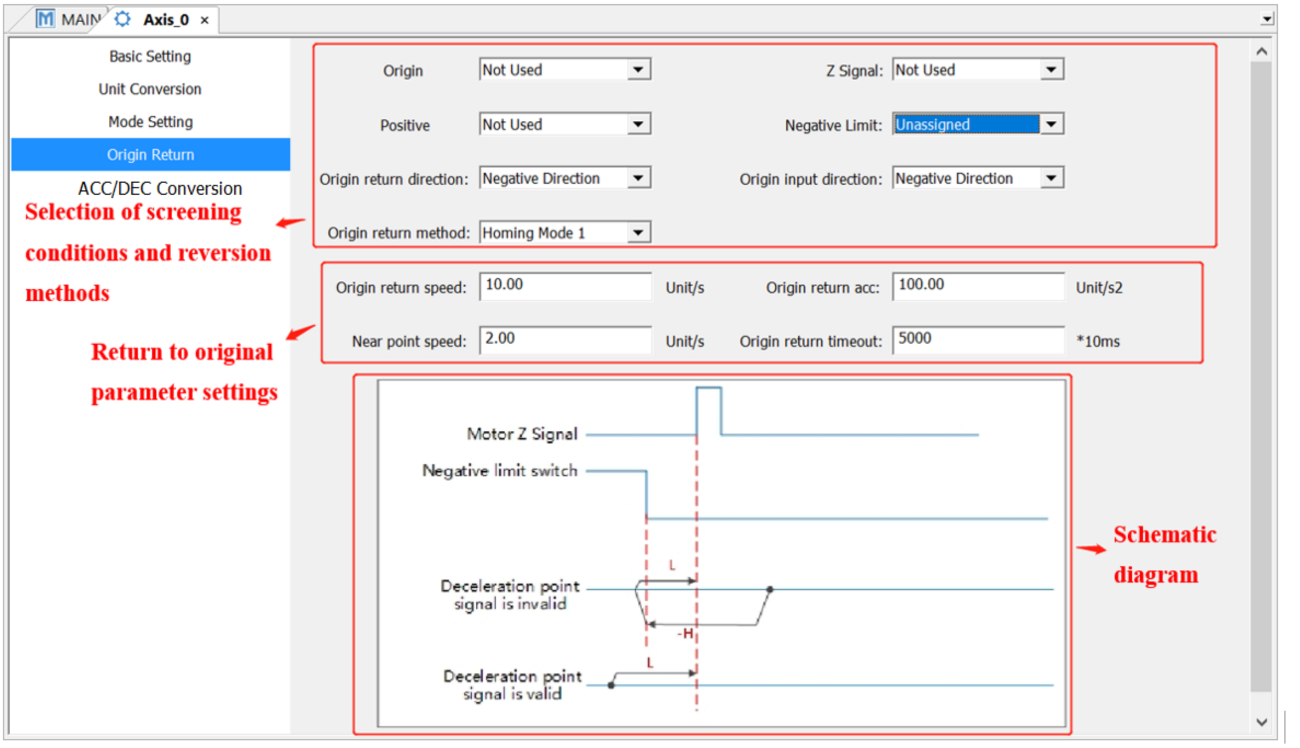
The local pulse axis allows Y0/Y1, Y2/Y3, Y4/Y5, and Y6/Y7 to be set as 4 local pulse axes.
The local pulse axis allows the output of pulses in pulse+direction, CW/CCW or AB phase format.
For channels that have been set to pulse axis, when the pulse + direction is selected, Y0, Y2, Y4, Y6 are the pulse terminals, and Y1, Y3, Y5, and Y7 are the direction terminals. When CW/CCW is selected, Y0, Y2, Y4, and Y6 are CW pulse terminals, and Y1, Y3, Y5, and Y7 are CCW terminals.

4.2.3.10 Hardware limits

Among the system variables of the axis are bphlimit (hardware positive limit) and bnhlimit (hardware negative limit) variables that represent the state of hardware limit, and when the hardware limit logic is set as positive logic, they correspond to the values of bit1 and bit0 0x60fd by the object dictionary, respectively. When set to negative logic, they are inverse to the values of bit1 and bit0 in the 0x60fd, respectively.
The logic of setting the hardware limit is only reflected in the above variables and has no effect on the limit shutdown treatment when the servo touches the limit.

4.2.4 The origin returns setting

VC5 supports the 1-35 return mode supported by the Cia402 protocol.
The origin return setting interface is as follows:



The parameters in the configuration interface are as follows:

Set the parameters	Description
Origin signal	Lets you choose whether to use the origin signal When Unassigned is selected, it is not used as a forced filter. When you choose not to use, the return mode that must use the origin signal is removed. When Use is selected, the original mode that does not support the origin signal is removed.
Z signal	Used to select whether to use the motor z signal When Unassigned is selected, it is not used as a forced filter. When you choose not to use it, the return mode that must use the z signal is removed. When you choose to use it, the original mode that does not support the z signal is removed.
Positive limit	Lets you choose whether to use a hardware right limit signal When Unassigned is selected, it is not used as a forced filter. When you choose not to use, the return mode that must use the positive limit signal is removed. When Use is selected, the return mode that does not support negative limit signals is removed.
Negative limits	Used to select whether to use a hardware left limit signal s When Unassigned is selected, it is not used as a forced filter. When you choose not to use, the return mode that must use the negative limit signal is removed.

	When Use is selected, the return mode that does not support negative limit signals is removed.
The origin returns in the direction	Lets you set the direction of motion at the beginning of the origin regression Forward: The direction of motion when the limit (origin) signal input is invalid is positive, otherwise it is reversed Negative: The direction of motion when the limit (origin) signal input is invalid is negative, otherwise it is reversed.
The origin input direction	The direction of motion when the origin signal is reached Forward: Touches the origin signal edge during forward motion to stop. Negative: The edge of the signal that touches the origin during negative motion stops.
Origin logic	Positive logic (high level) Negative logic (low level)
How the origin is returned	When setting the range 1-35, write the object dictionary 0x6098 as a startup parameter.
The origin return speed	The origin returns the speed, converts the user unit to pulse unit, and then writes the object dictionary 0x6099 subindex No. 1 in the form of a startup parameter.
The origin returns the acceleration	The origin returns the acceleration, converts user units to pulse units, and writes them to the object dictionary 0x609A as startup parameters.
The origin returns to the perimeter speed	The origin returns the speed, converts the user unit to pulse unit, and then writes the object dictionary 0x6099 subindex No. 2 in the form of a startup parameter.
The origin returns a timeout period	Unit 10ms

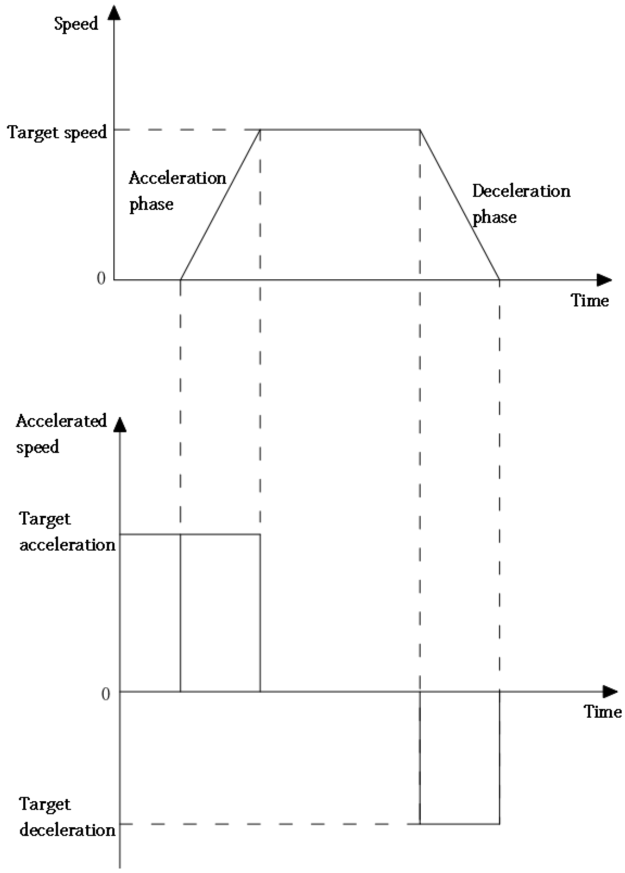
In actual use, the origin regression method is defined by several parameters such as origin signal, positive limit, negative limit, Z signal, origin return direction, origin input direction, and then the desired mode is selected through the origin return mode option.

4.2.5 The curve type

VC5Two speed curves are supported: T-type acceleration and deceleration and 5-segment S curve acceleration and deceleration, which are determined by the SpeedMode parameter in the command. In addition, when the shaft hits the limit or otherwise needs to perform a fault deceleration shutdown to make the shaft enter the Errorstop state, it is also performed according to the T-curve.

T-type velocity curve:

In the command, when SpeedMode = 0, the shaft does T-type acceleration and deceleration. In the T-type velocity curve, the axis is curved according to the target position, target speed, target acceleration, and target deceleration. In the acceleration and deceleration process, the actual acceleration and deceleration are fixed, and the positioning curve is shown in the following figure：



● Target position: The position that the index axis will eventually reach in the absolute positioning command, and the unit is Unit.

● Target speed: The maximum speed that can be achieved during the operation of the axis, in units of Unit/s (user units per second).

● Target acceleration: The amount of change in the speed of the shaft per second when it is accelerated, in Unit/t2.

● Target deceleration: The amount of change in the speed of the shaft per second when it is decelerating, in Unit/t2.

● Acceleration phase, assuming that the initial velocity of the shaft is Vs, the target velocity is Vt, and the target acceleration is Acc, then the time of the acceleration phase：

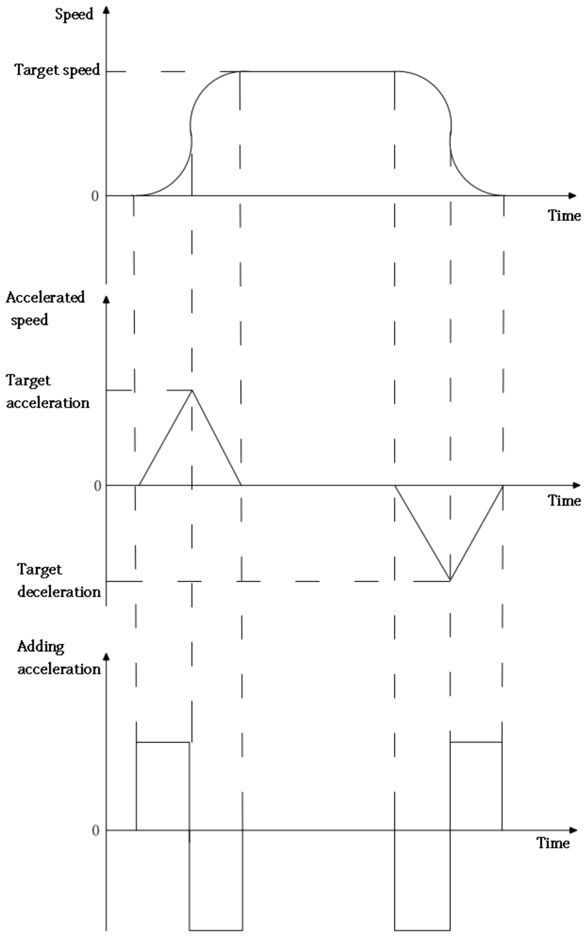
Tacc = (Vt - Vs)/Acc

● Deceleration phase, assuming that the initial speed of the shaft is Vs, the target speed is Ve, and the target deceleration is Dec, then the time of the deceleration phase:

Tdec = (Vs - Ve) / Dec

S-shaped velocity curve

When SpeedMode = 1, the shaft does S-type acceleration and deceleration. In the 5-segment S curve, the axis is curved according to the target position, target speed, target acceleration, and target deceleration, where the target acceleration and target deceleration refer to the maximum value reached during the acceleration and deceleration process. The positioning **Curve is shown in the following figure：**



In the 5-segment S-type speed curve, according to the state of acceleration, acceleration reduction, uniformity, acceleration and deceleration, deceleration 5 stages, there must be no uniform acceleration and uniform deceleration stage, in the acceleration, acceleration and deceleration and other variable acceleration stage in the actual Jerk is H5u internal calculation, the user can not set.

● Target position: The position that the index axis will eventually reach in the absolute positioning command, and the unit is Unit.

● Target speed: The maximum speed that can be achieved during the operation of the axis, in units of Unit/s (user units per second).

● Target acceleration: The maximum amount of change in the speed of the shaft per second when it is running variable acceleration, in Unit/t2. The acceleration of the velocity curve at the moment when the velocity changes from the acceleration stage to the deceleration (t2) must be the target acceleration.

● Target deceleration: The maximum change in the speed of the shaft per second when it is running to decelerate, the unit is Unit/t2, and the deceleration of the speed curve from the deceleration stage to the deceleration of this moment (t5) must be the target deceleration.

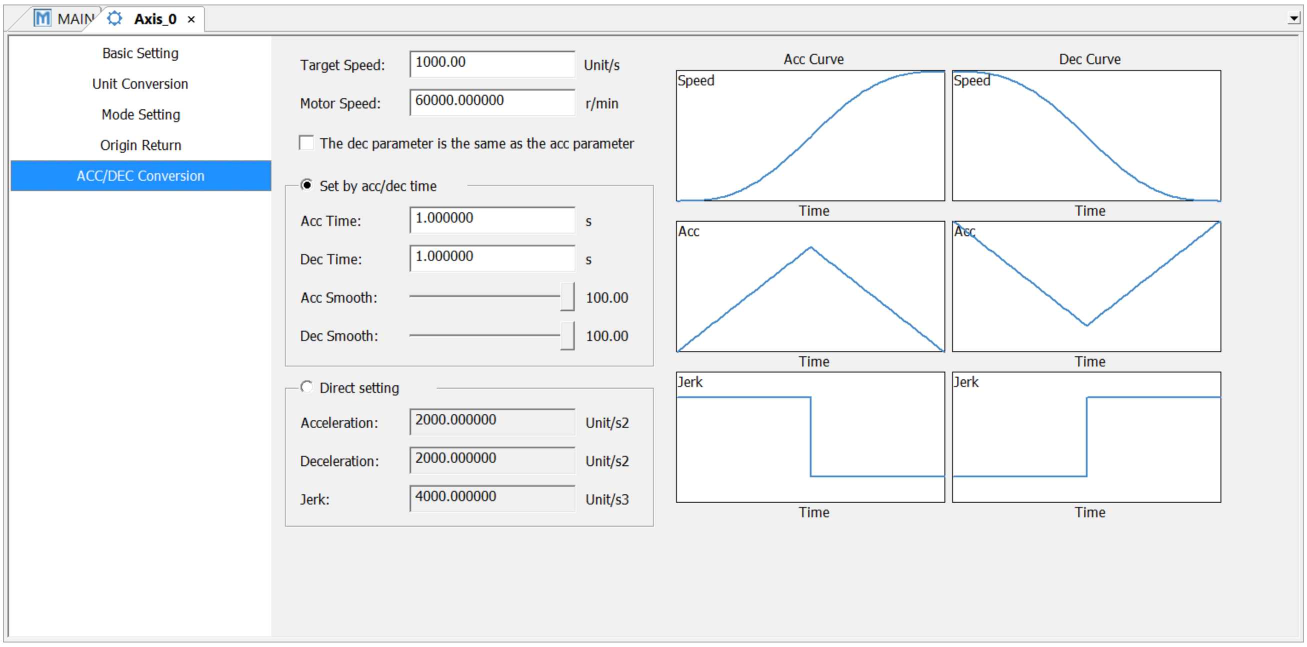
● Acceleration phase, assuming that the initial velocity of the shaft is Vs, the target velocity is Vt, and the target acceleration is Acc, then the time of the acceleration phase: Tacc = 2* (Vt - Vs)/Acc.

● Deceleration phase, assuming that the initial speed of the shaft is Vs, the target speed is Ve, and the target deceleration is Dec: Tdec = 2* (Vs - Ve) / Dec.

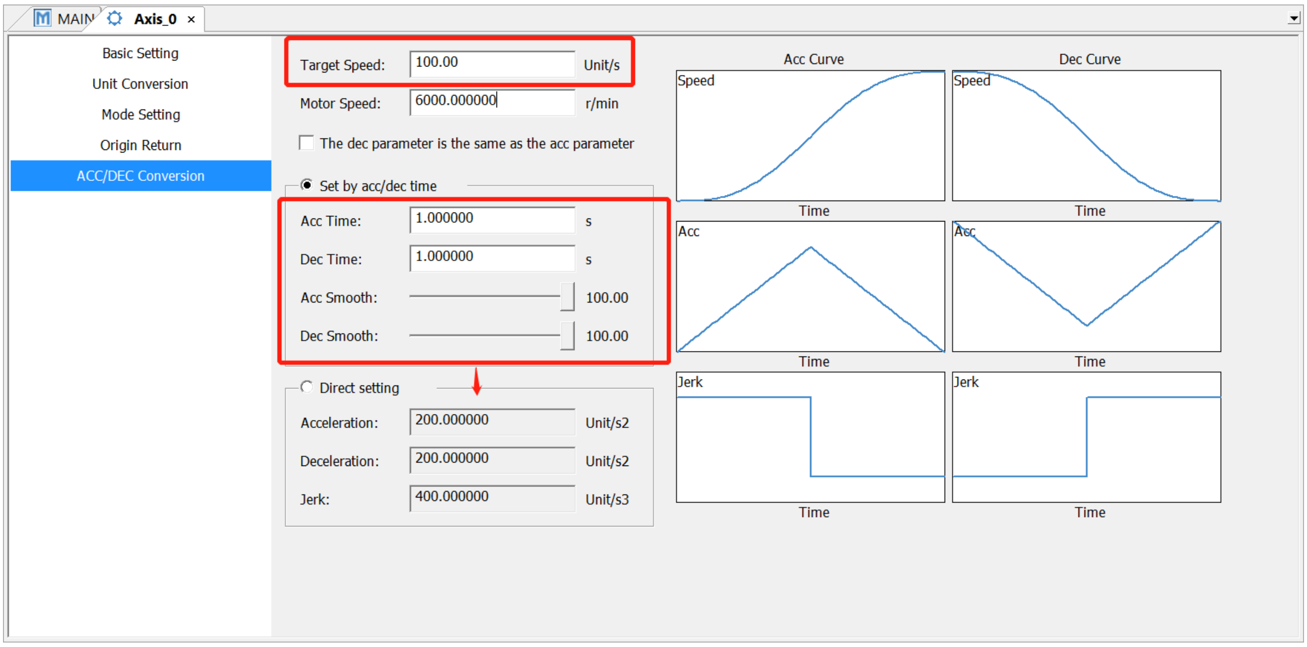
4.2.6 Acceleration and deceleration time conversion

In the 5-segment S-type velocity curve, the acceleration, deceleration, and acceleration rate of the curve can be calculated according to the acceleration time.

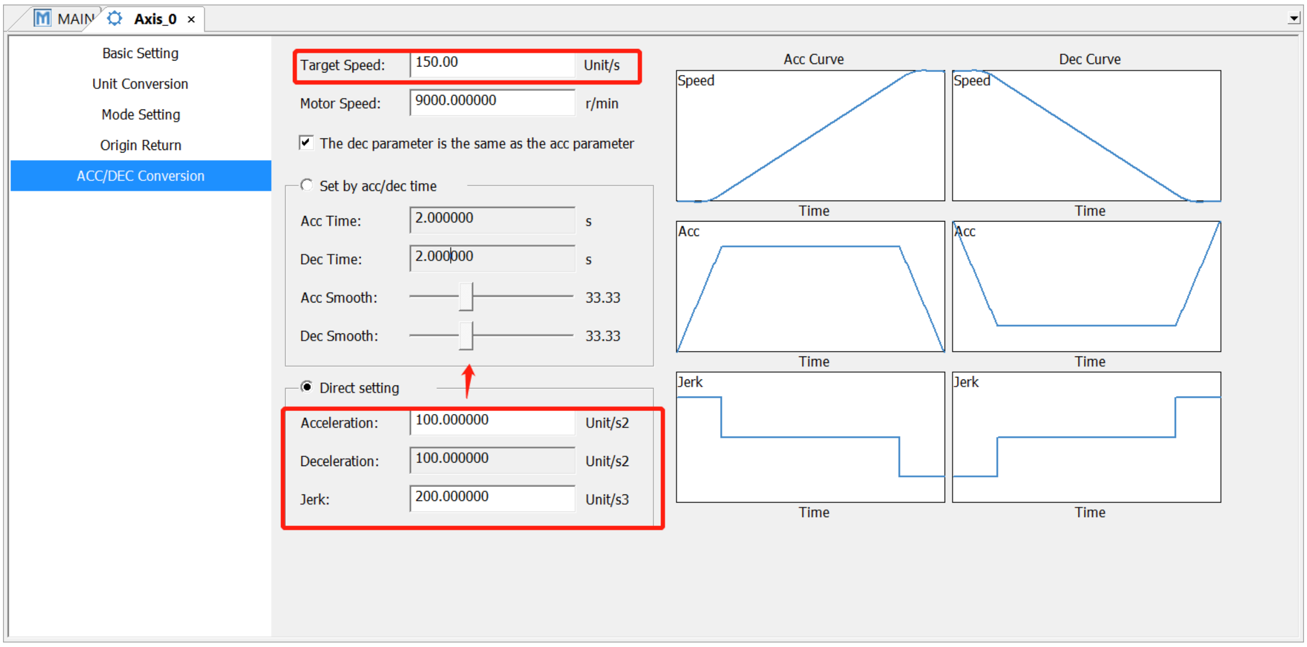
The default display target speed is 1000Unit/s, the motor speed is 60000r/min, and the actual target speed is filled in when using.



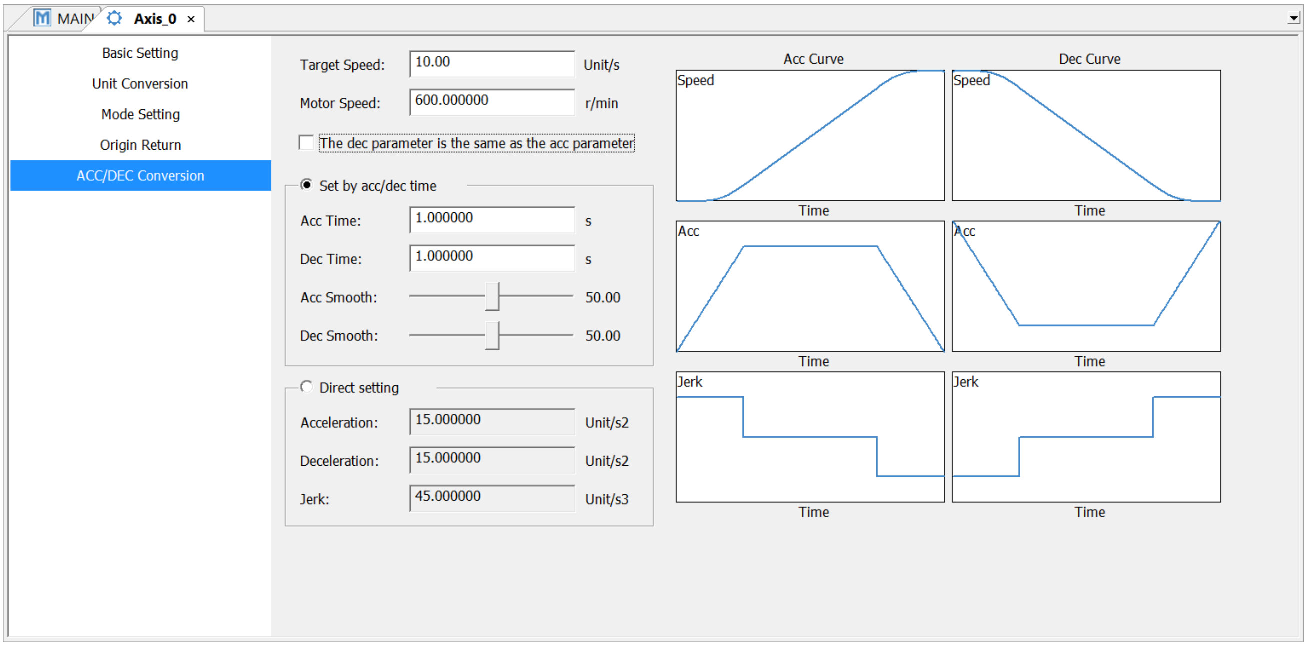
If the target speed is 100Unit/s and the acceleration/deceleration time is 1s, the acceleration/deceleration and acceleration change rate will be calculated automatically.



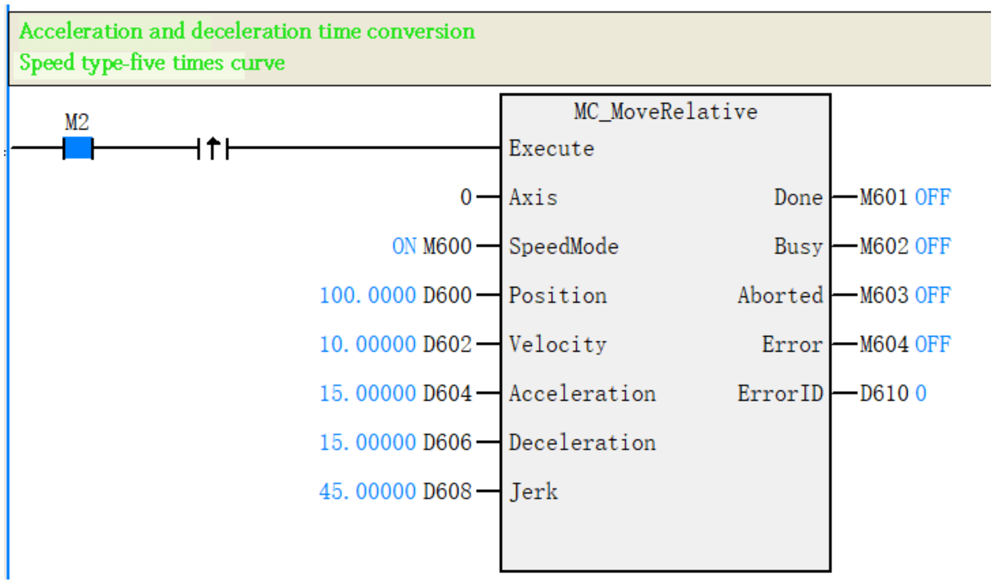
You can also directly set the target speed, acceleration/deceleration and acceleration change rate, and automatically calculate the acceleration/deceleration time.



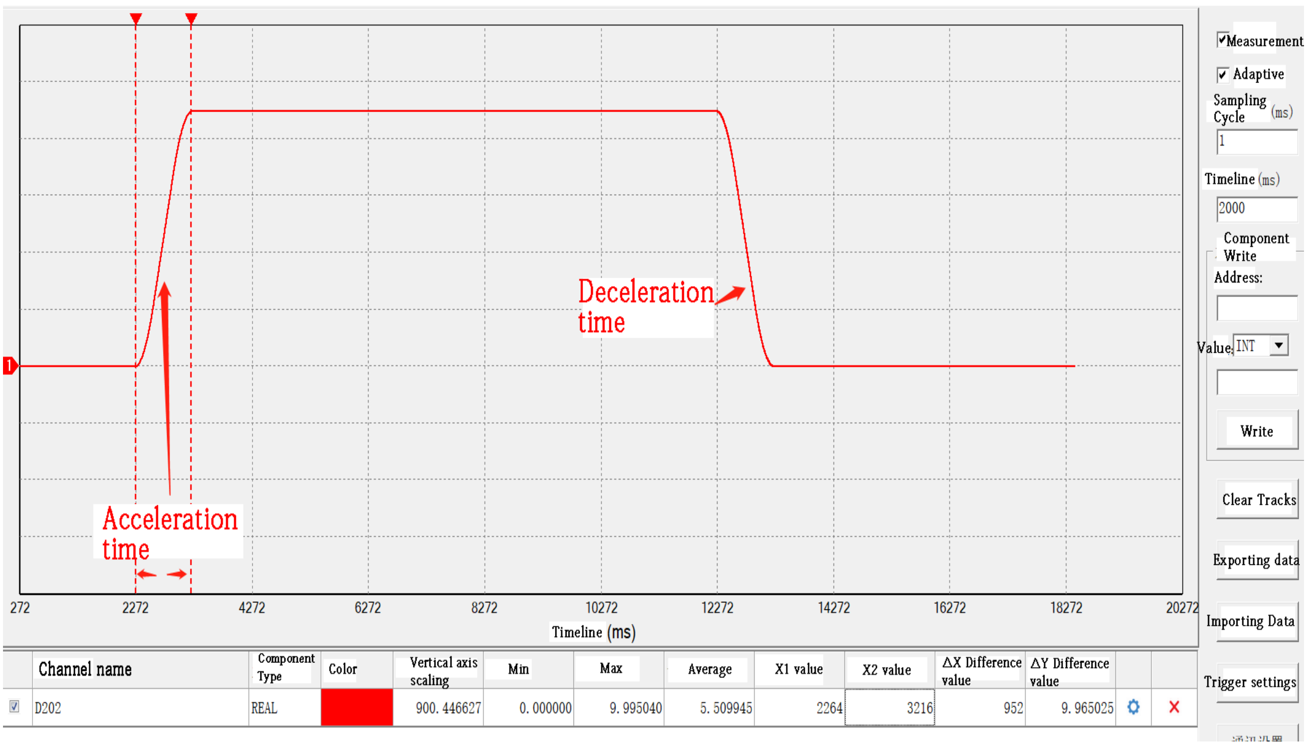
Example: For a relative motion, the acceleration/deceleration time is expected to be 1s, and the acceleration/deceleration and change rate are calculated as 15, 15 and 45 respectively according to the table.



Program monitoring:



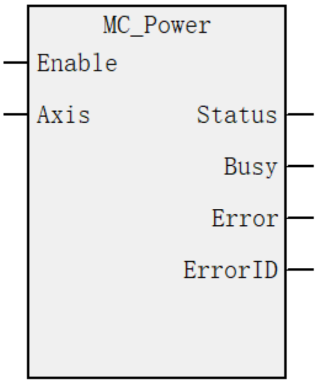
The curve monitors the speed curve, and its acceleration and deceleration time can be observed, as shown in the following figure:



4.3. MC Axis Control Instruction

4.3.1 MC_Power (Enable control command)

Drawing Block:



Command list format

Input/Output	Name	Data type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis number	Enable Valid and invalid
OUT	Status	BOOL	M/S	TRUE,FALSE	Axis enable flag	
OUT	Busy	BOOL	M/S	TRUE,FALSE	In-Run flag	
OUT	Error	BOOL	M/S	TRUE,FALSE	Error flag	
OUT	Error ID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

MC_Power Commands apply to EtherCAT bus axes and local pulse axes to set the enable state of the shaft at high levels.

Axis number setting	The ascending edge command on the Enable latches the axis specified by Axis. If the axis number is used to access the axis, when Enable=TRUE is modified, the command is error-reported, and the originally controlled axis is enabled. If the axis number is used to access the axis, modify the Axis when Enable=FALSE, if the axis corresponding to the Axis is originally in the enabled state, and this command is the last Power command executed in a PLC scan cycle, the axis corresponding to the Axis will be enabled becomes unrunnable.
Functional description	After setting Enable to True, the axis enters the enabled state and the Status signal of the command is valid. The PLCOpen state machine of the axis enters the StandStill state from the disabled state. After activation, motion commands such as MC_MoveRelative can be executed. After setting Enable to False, you can unenable the axis and interrupt the execution of motion commands such as MC_MoveAbsolute. After the enabled state is lifted, the axis does not accept action commands and cannot be controlled. However, non-motion commands such as MC_Power,

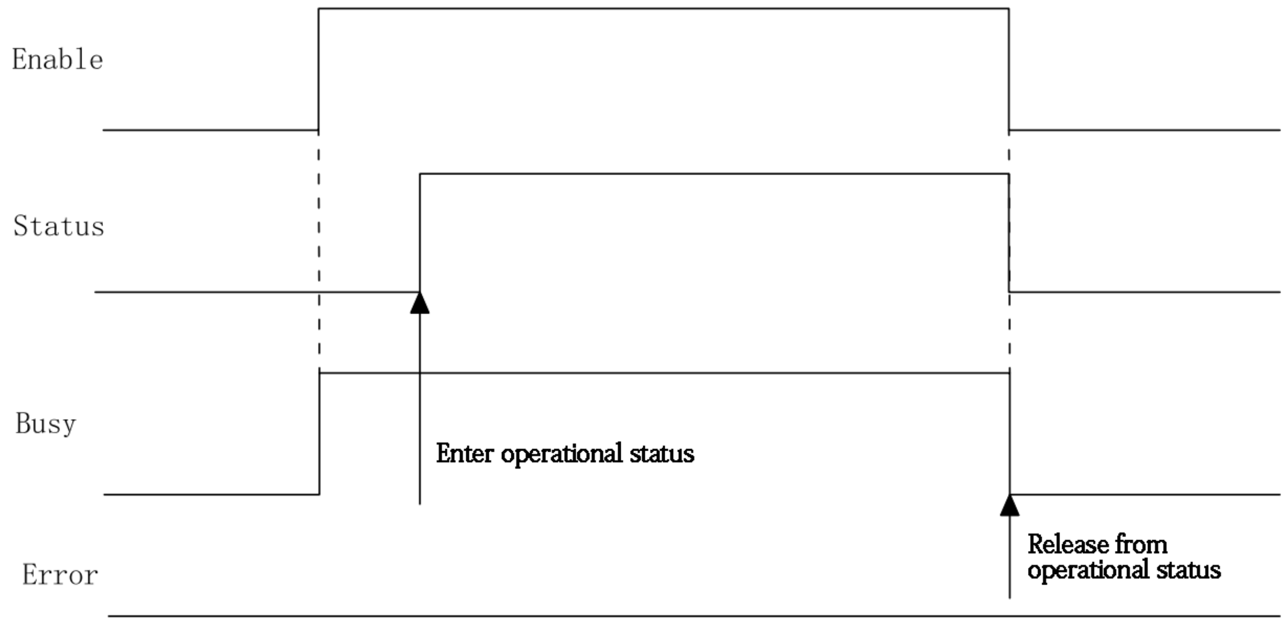
	MC_Reset, MC_ SetPosition can be executed. When the axis enters the errorstop state due to failure, the MC_Power cannot switch the axis to the standstill state after the MC_Reset is re-enabled, and the command must be called first to recover the failure of the axis.
Multiple calls	When using multiple MC_Power command, multiple calls are not allowed for the same axis number.

Error report

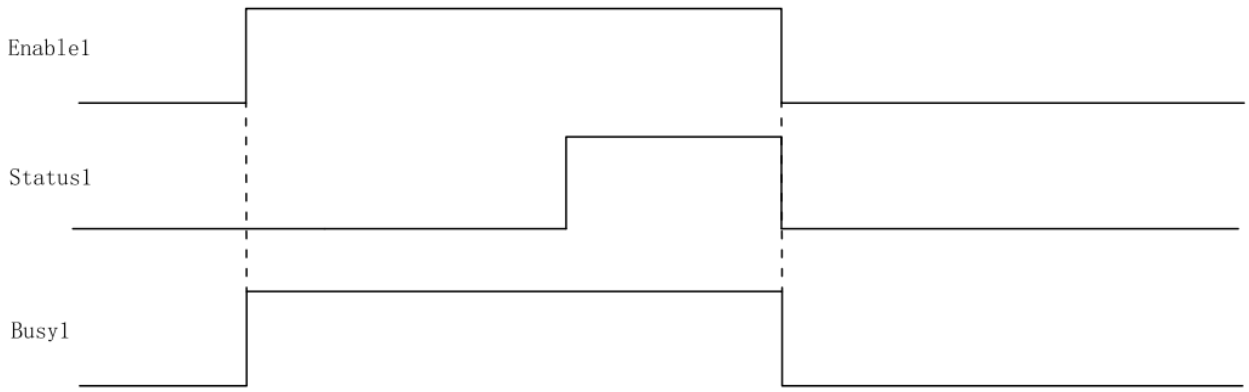
When the axis number does not exist, the command reports an error
When the axis type is wrong, the command reports an error
When the axis initialization fails, an error is reported
An error is reported when the four parameters of control word, status word, target position, and current position are not configured in the PDO of the axis

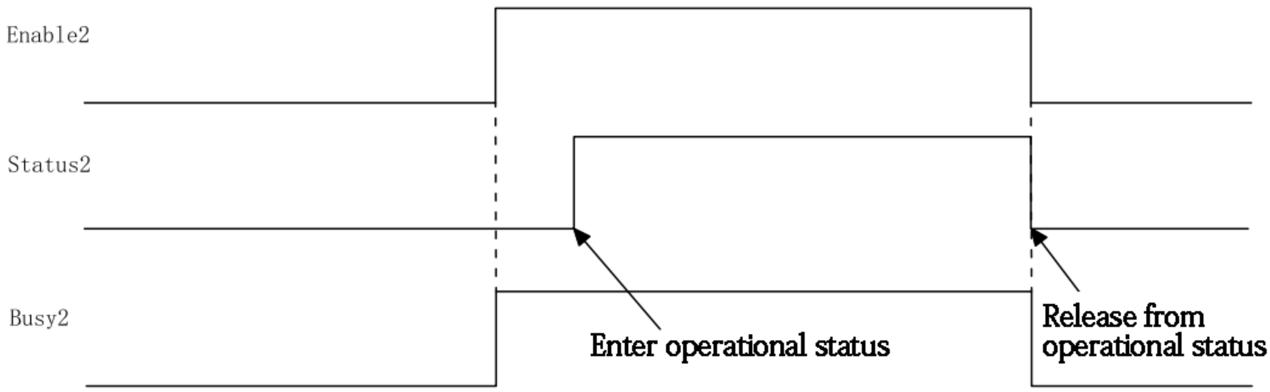
Timing diagram

● When a MC_Power command is used, the axis is enabled normally.



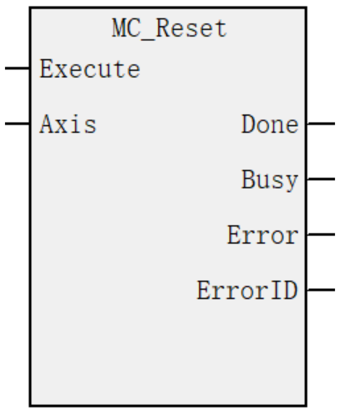
● When two MC_Power commands are used, the axis is normally enabled.





4.3.2 MC_Reset (Reset fault command)

Drawing Block:



Command list format

Input/Output	Name	Data type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	constant/D/R/W	0~71	Axis name/Number	Execute Rising Edge
OUT	Done	BOOL	M/S	TRUE,FALSE	Reset the completion flag	

OUT	Busy	BOOL	M/S	TRUE,FALSE	In-Run flag	
OUT	Error	BOOL	M/S	TRUE,FALSE	Error flag	
OUT	Error ID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

MC_Reset The commands apply to both EtherCAT bus shafts and local axes for resetting shaft faults, rising edge triggering.

On the rising edge of the Execute signal valid, the command attempts to reset the fault of the axis, if the reset is successful, the Done output is valid, otherwise the Error signal is valid, and ErrorID will give the reason for the reset failure
After a successful reset, the PLCOpen state machine of the axis enters the StandStill state if the drive is in the enabled state, or the Disabled state if the drive is not enabled

Interrupt

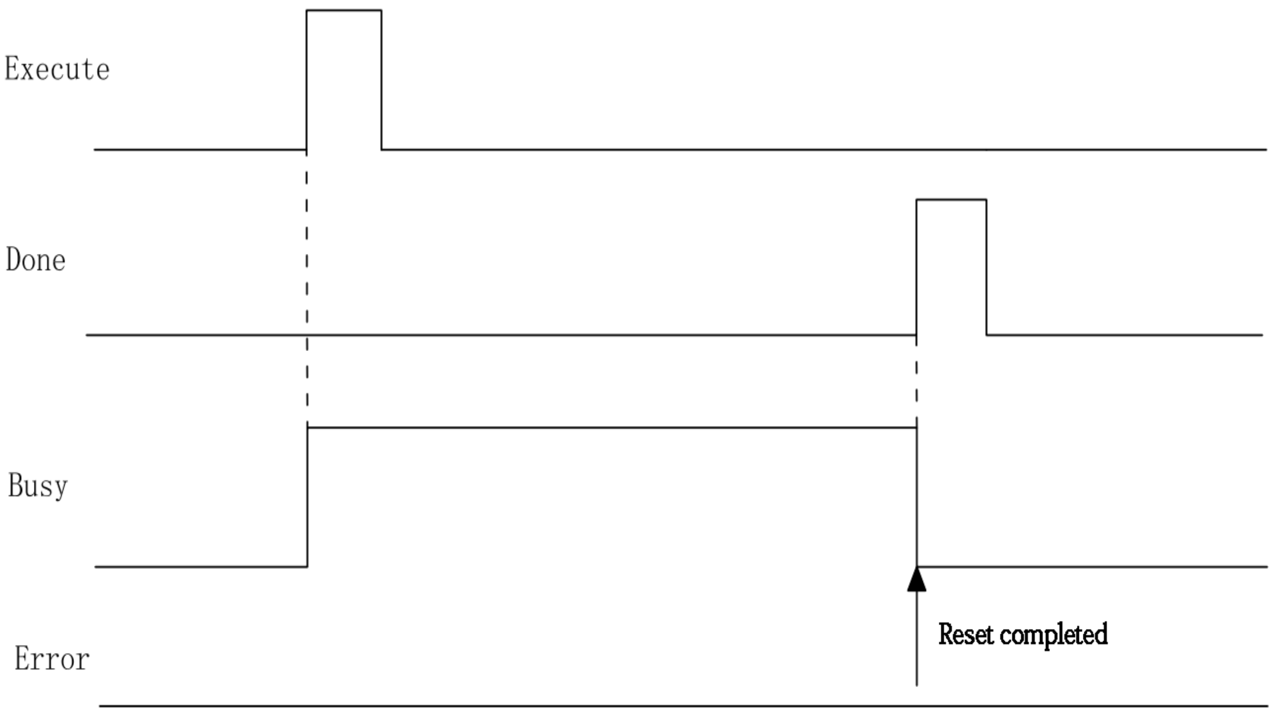
This command does not interrupt the output signal and cannot be interrupted during execution
If there are two reset commands in a scan cycle, as long as one reset command is valid, regardless of the sequence, the program will start to execute the reset logic, if the reset is successful, the triggered command Done signal output is valid

Error report

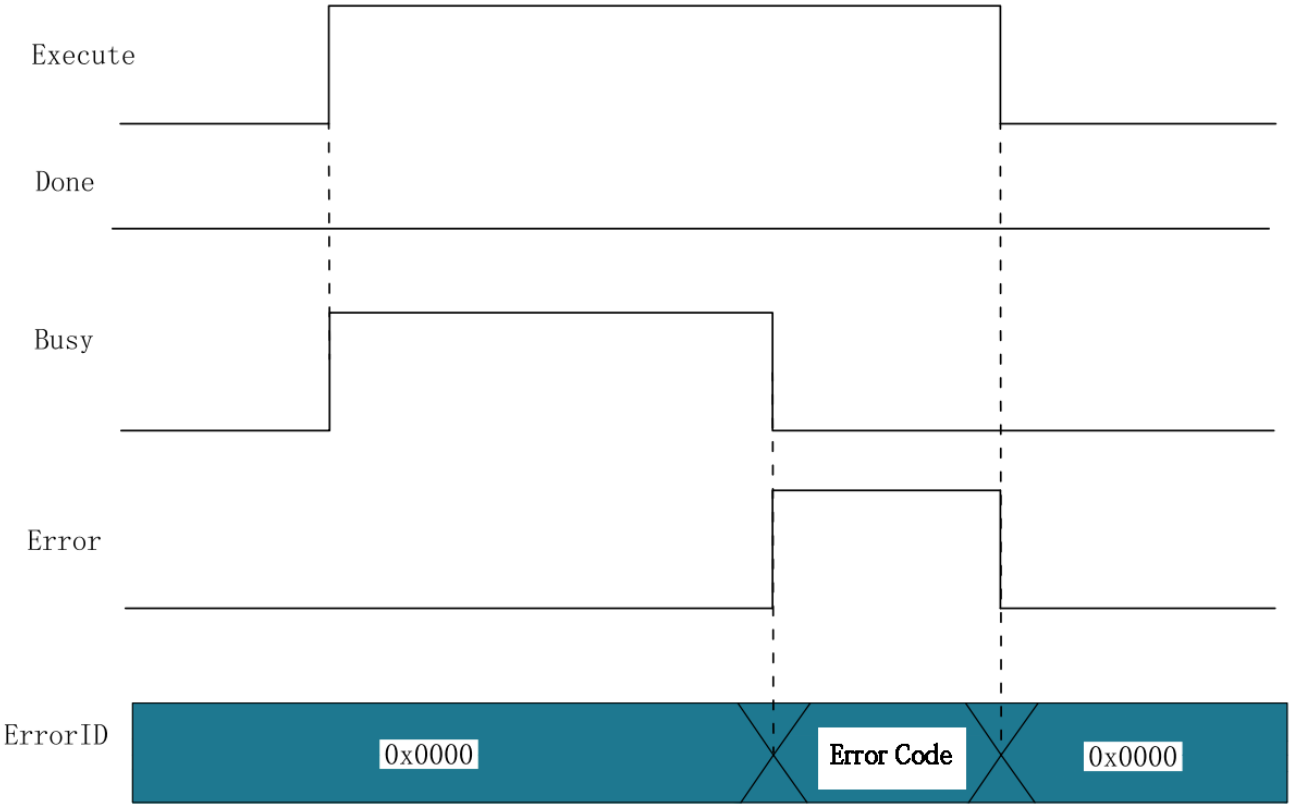
When the axis number does not exist, the command reports an error
When the axis type is wrong, the command reports an error
When the axis initialization fails, an error is reported
When this command is called without a fault, the command reports an error
An error is reported when the shaft itself cannot be reset

Timing diagram

● A fault that invokes MC_Reset command when an axis fails and successfully resets the axis.

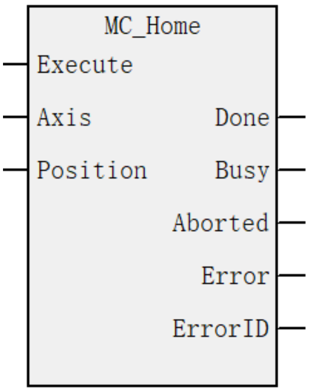


● When a non-resettable fault occurs with the drive.



4.3.3 MC_Home (Return to origin command)

Drawing block:



Command list format

Input/Output	Name	Data type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/Number	Execute rising edge
IN	Position	REAL	Constant/D/R/W	Positive/Negative/0	Origin offset	Execute rising edge
OUT	Done	BOOL	M/S	TRUE,FALSE	Return to	

					origin completion flag	
OUT	Busy	BOOL	M/S	TRUE,FALSE	Running flag	
OUT	Aborted	BOOL	M/S	TRUE,FALSE	Execution interrupt	
OUT	Error	BOOL	M/S	TRUE,FALSE	Error flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

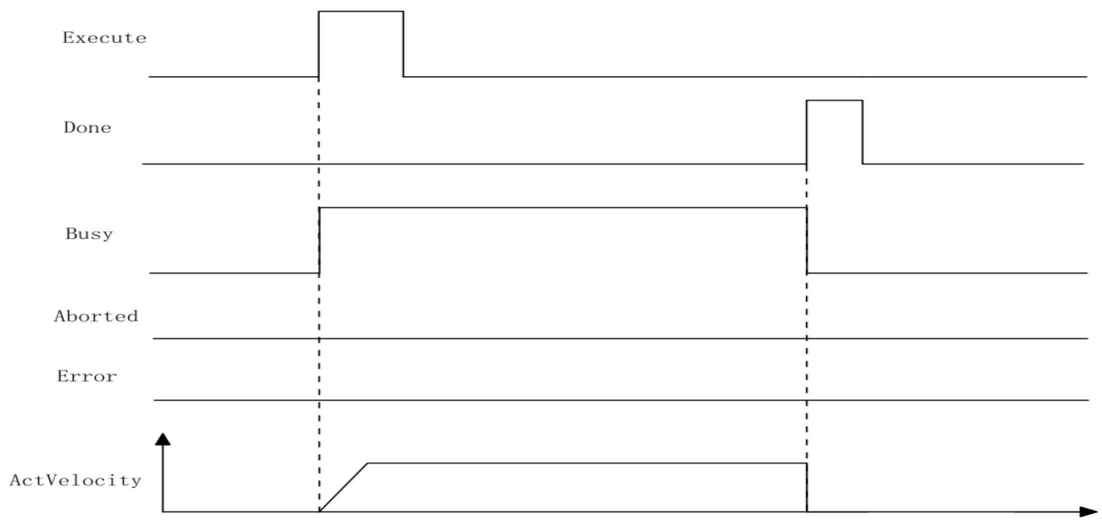
Description of functions and commands

This command is used to implement the origin regression function of the EtherCAT bus axis and the local pulse axis, and the command is valid on the rising Edge.

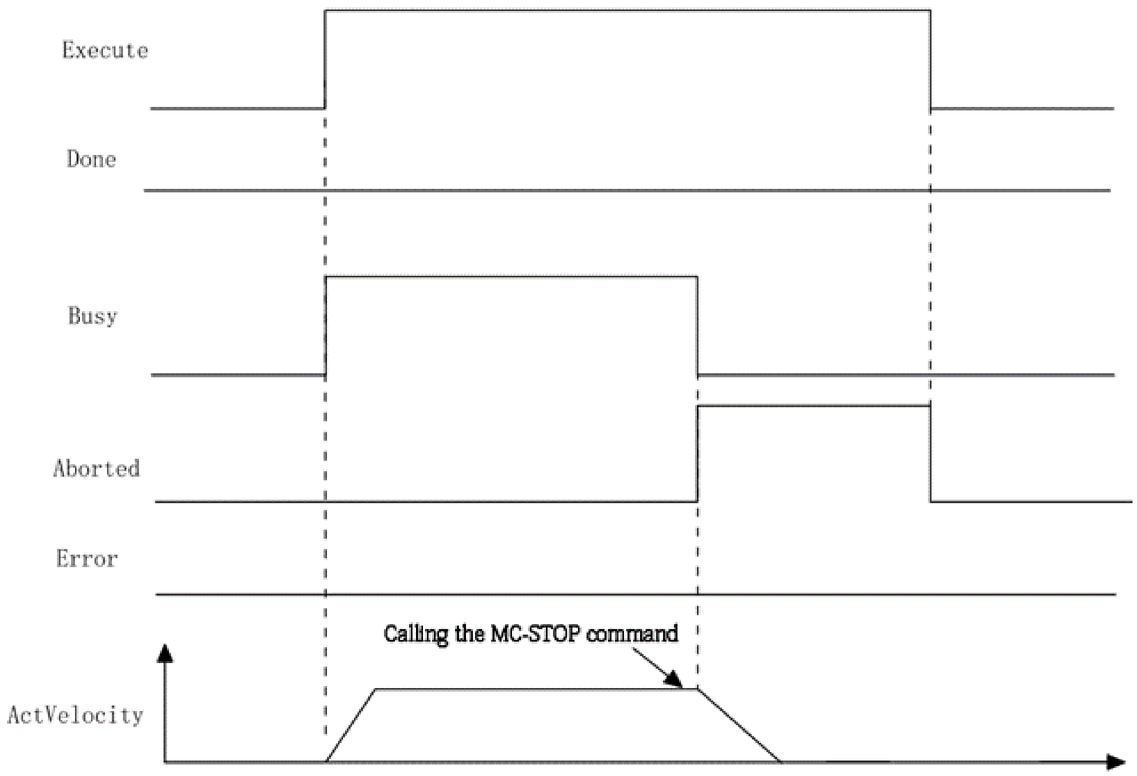
The setting of the axis number	Latches on the rising edge of the Execute input; During execution, modification to other axis numbers to report errors. After execution is completed and energy flow is valid, modify to other axis number to report an error. After execution is completed and energy flow is invalid, modify to other axis number is invalid and wait for rise to be valid.
Functional description	This command can only be invoked by using MC_Power directive to switch the axis to the enabled state; On the rising edge of the command, the function block latches the Position input parameter, the axis is in the Homing state and does the origin regression movement; Position is used to set the origin offset; Calling this command in virtual axis mode will return to zero in absolute mode No. 35 in the 402 protocol.
Multiple calls	The back-to-origin command does not allow repeated calls, when invoking a MC_Home command so that the axis is in the homing state after calling other MC_Home commands, the invoked command after the call is wrong.

Timing diagram

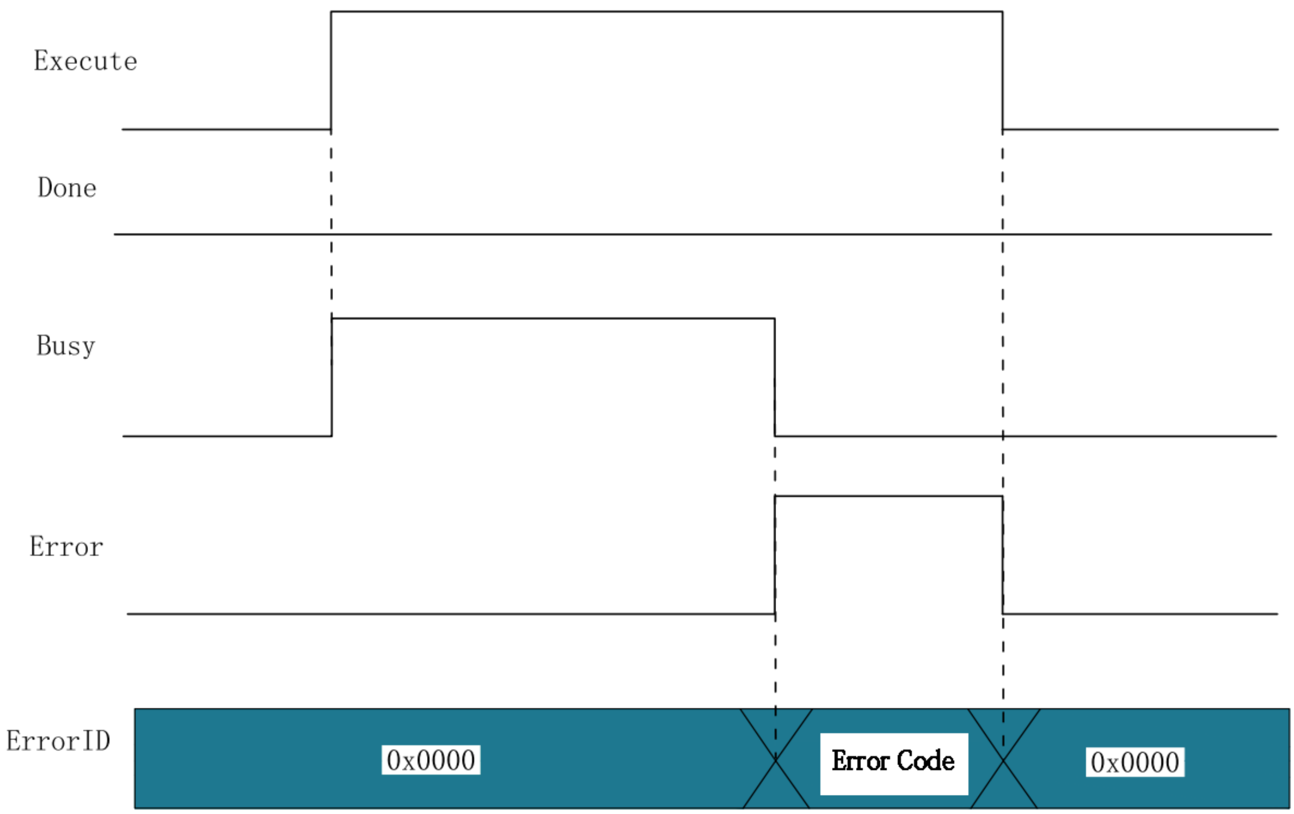
● Origin regression is initiated, and the drive performs the origin regression action normally.



● During origin regression MC_Stop command is called to interrupt the origin regression action.

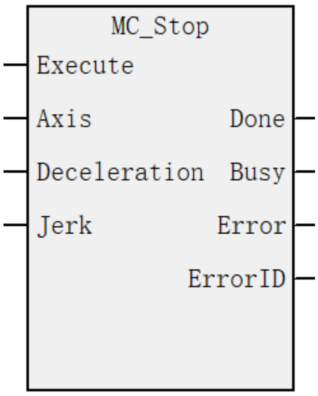


● The drive failed during origin regression.



4.3.4 MC_Stop (Stop command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable（soft components)	Rang	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/Number	Execute Rising edge
IN	Deceleration	REAL	Constant/D/R/W	positive number	Decelerate	Execute Rising edge
IN	Jerk	REAL	Constant/D/R/W	positive number	Speed step	Execute Rising edge
OUT	Done	BOOL	M/S	TRUE,FALSE	Stop completion flag	
OUT	Busy	BOOL	M/S	TRUE,FALSE	In-Run flag	
OUT	Error	BOOL	M/S	TRUE,FALSE	Error flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	ErrorID	

Description of functions and commands

This command is used for the EtherCAT bus axis or the local pulse axis to implement the stop function, and the command is valid on the rising edge.

The setting of the axis number	Latches on the rising edge of the Execute input; During execution, modification to other axis numbers to report errors. After execution is completed and energy flow is valid, modify to other axis number to report an error. After execution is completed and energy flow is invalid, modify to other axis number is invalid and wait for rise to be valid.
Functional description	This directive can only be invoked if the axis is switched to the enabled state using MC_Power command. On the rising edge of the Execute input, the function block latches input parameters such as Deceleration, and the axis is in the Stopping state and does a deceleration movement. After the deceleration is complete, the Do signal is valid and remains in the Stopping state for the duration of Execute=TRUE. When Execute=FALSE and Done=TRUE, the axis switches from the

	Stopping state to the Standstill state. The axis is in different running states when the command is called, and the stopping method is also different. 1. When the axis is executing a positioning command or when it is running continuously, the parameter CurveType is used to set the type of speed curve. SpeedMode = 0 indicates a T-shaped curve, in which case the speed of the axis will be slowed down according to the value set by Deceleration; SpeedMode = 1 represents a 5-segment S curve, in which case Deceleration represents the maximum deceleration achieved by the shaft during deceleration. 2. When the shaft is in the origin regression state, this command triggers the Halt flag of the driver's control word, and the drive performs deceleration according to the preset parameters, in which case Decelestial is invalid.

Restart command

The same stop command is executed repeatedly, and the same stop command is retriggered during the deceleration and shutdown according to the deceleration at the time of the last trigger.

Multiple calls

MC_Stop command cannot be called multiple times, other stop commands are called during the validity of a stop command, and other stop commands report failure.

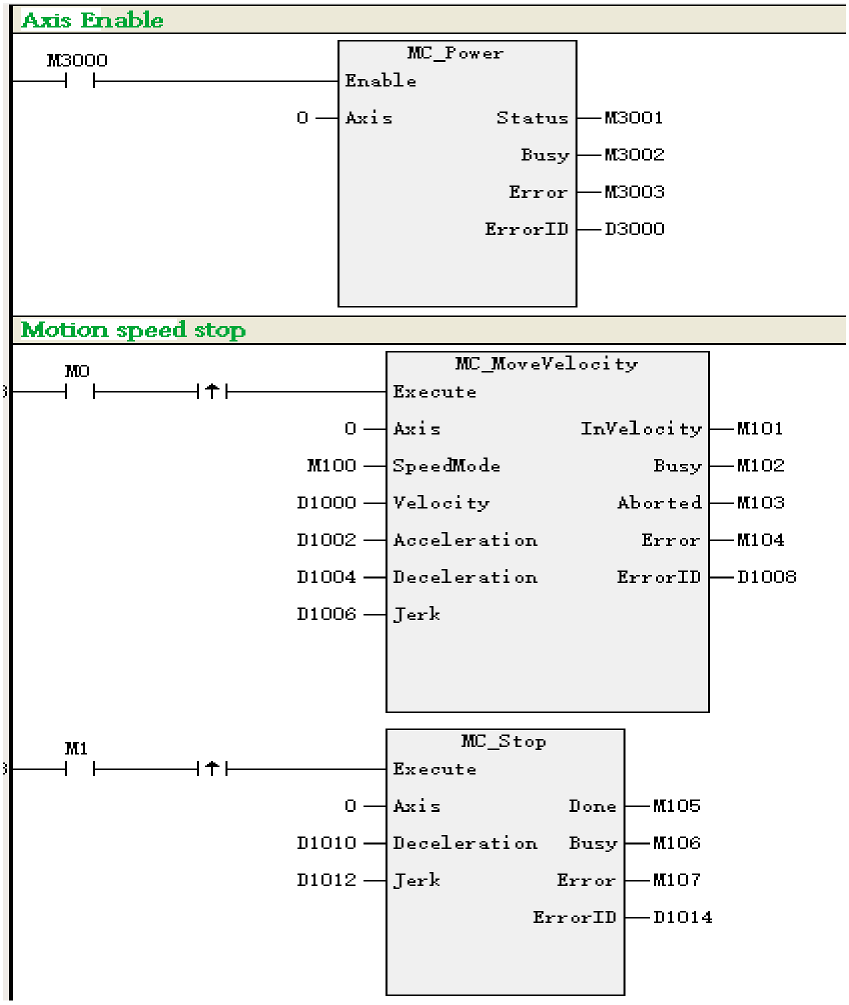
Interrupt

When the command is valid, the axis is in the Stopping state, other motion commands cannot interrupt the command, and when the command is invalid, the axis is switched from the Stopping state to the Standstill state, and other motion control commands can run.

Program examples

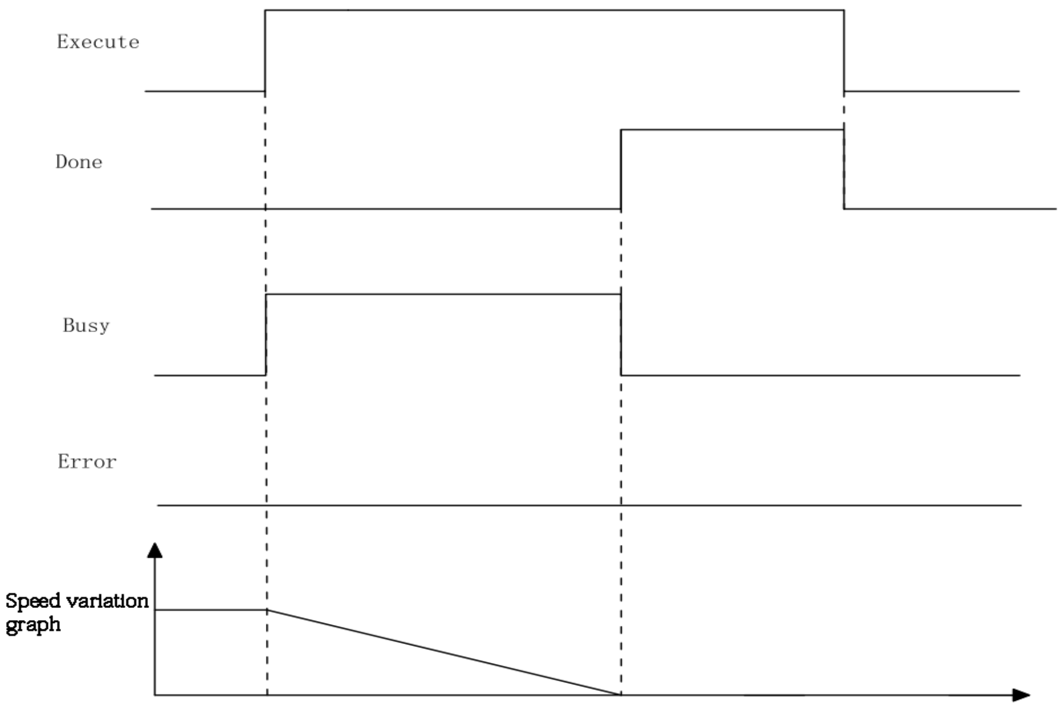
The stop function is implemented by the following routine.

1. Set M3000 to ON and enable the axis Axis_0.
2. Set M0 to ON Rear Axle Axis_0 run at the set speed.
3. Set M1 to ON and stop running Axis_0 the rear axle.

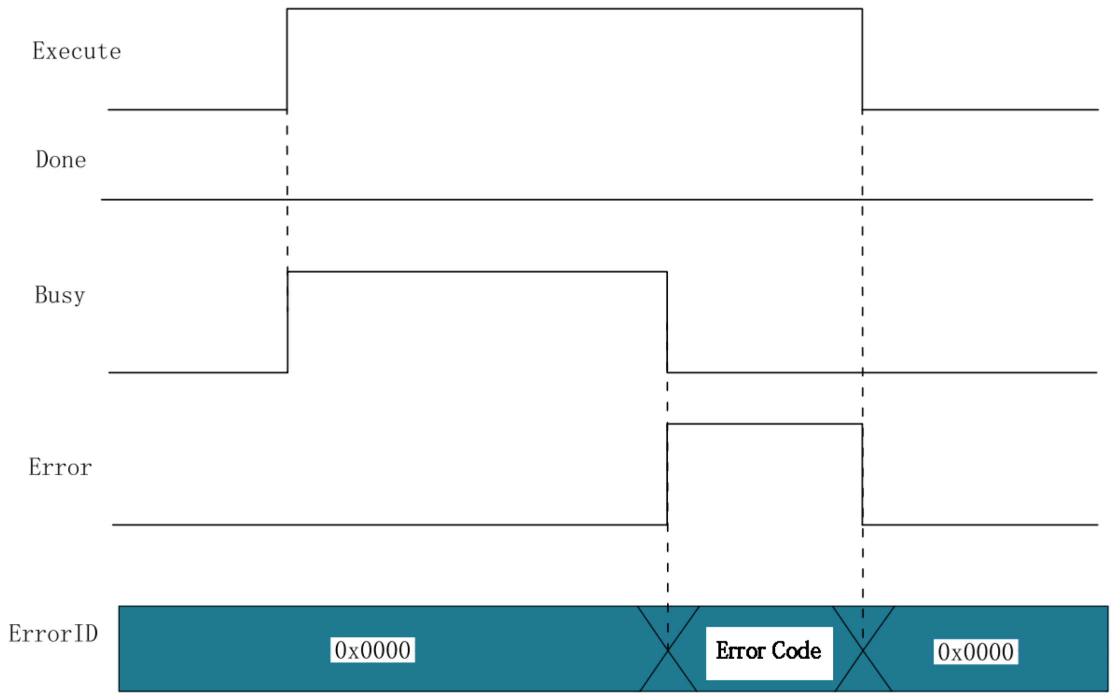


Timing diagram

● Call the MC_MoveVelocity directive before calling this directive.

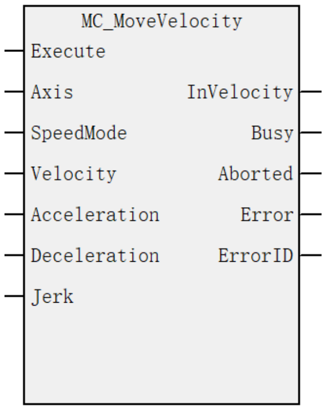


● The drive failed while the command was running.



4.3.5 MC_MoveVelocity (Speed command)

Drawing block:



Command list format

Input/Output	Name	Data type	Applicable (soft components)	Rang	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis number	Execute Valid
IN	SpeedMode	BOOL	M/S	TRUE, FALSE	Speed Type: Type 0T, Type 1S	Execute Rising edge
IN	Velocity	REAL	Constant/D/R/	Positive	The target	Execute Rising

			W	Number	speed	edge
IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	Execute Rising edge
IN	Deceleration	REAL	Constant/D/R/W	Positive Number	Decelerate	Execute Rising edge
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed step	Execute Rising edge
OUT	InVelocity	BOOL	M/S	TRUE, FALSE	Reach the target speed flag	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interrupts	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	ErrorID	

Description of functions and commands

This command is used to control the EtherCAT bus axis or the local pulse axis for absolute positioning with active rising edges.

The setting of the axis number	Axis is latched along the rising edge of the Execute input； During execution, modification to other axis numbers to report errors. After execution is completed and energy flow is valid, modify to other axis number to report an error. After execution is completed and energy flow is invalid, modify to other axis number is invalid and wait for rise to be valid..
Functional description	This directive can only be invoked if the axis is switched to the enabled state using MC_Power command. On the rising edge of the Execute input, the command latches the input parameters on the left side of Velocity, Acceleration, etc., the trigger axis runs at the speed set by Velocity, and switches the PLCOpen state machine of the axis to the ContinuousMotion state. SpeedMode is used to set the type of speed curve. SpeedMode = 0 indicates a T-shaped curve in which the speed of the axis will accelerate or decelerate according to the values set by Acceleration and Deceleration. SpeedMode = 1 represents a 5-segment S curve, in which case Acceleration and Deprovide represent the maximum acceleration and minimum deceleration achieved by the shaft during acceleration and deceleration. After calling this command, if you want to stop the movement of the axis, you can call the MC_Stop command.

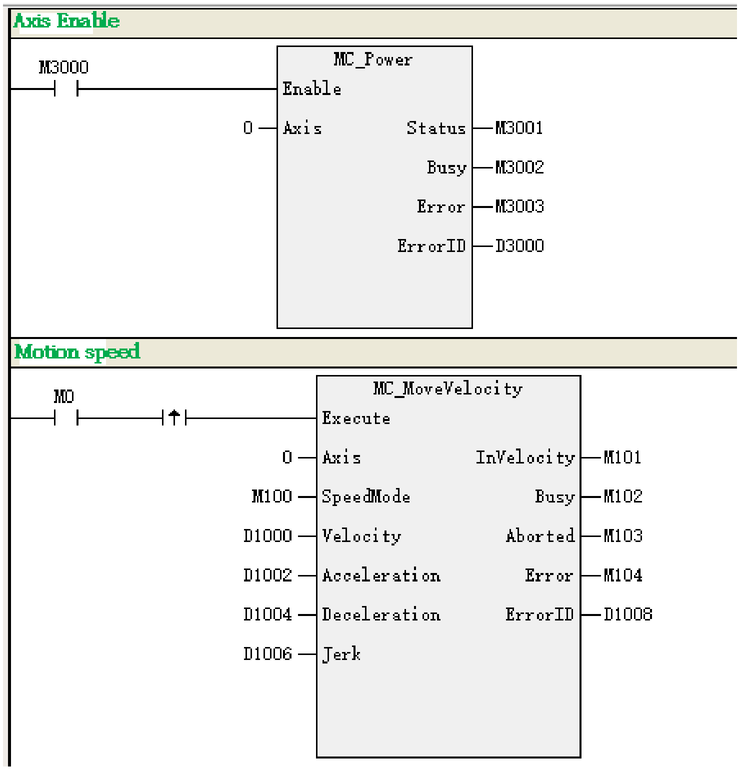
Interrupt

When the command is valid, other motion commands cannot interrupt the instruction, and other motion control commands need to execute MC_Stop command first..

Program examples

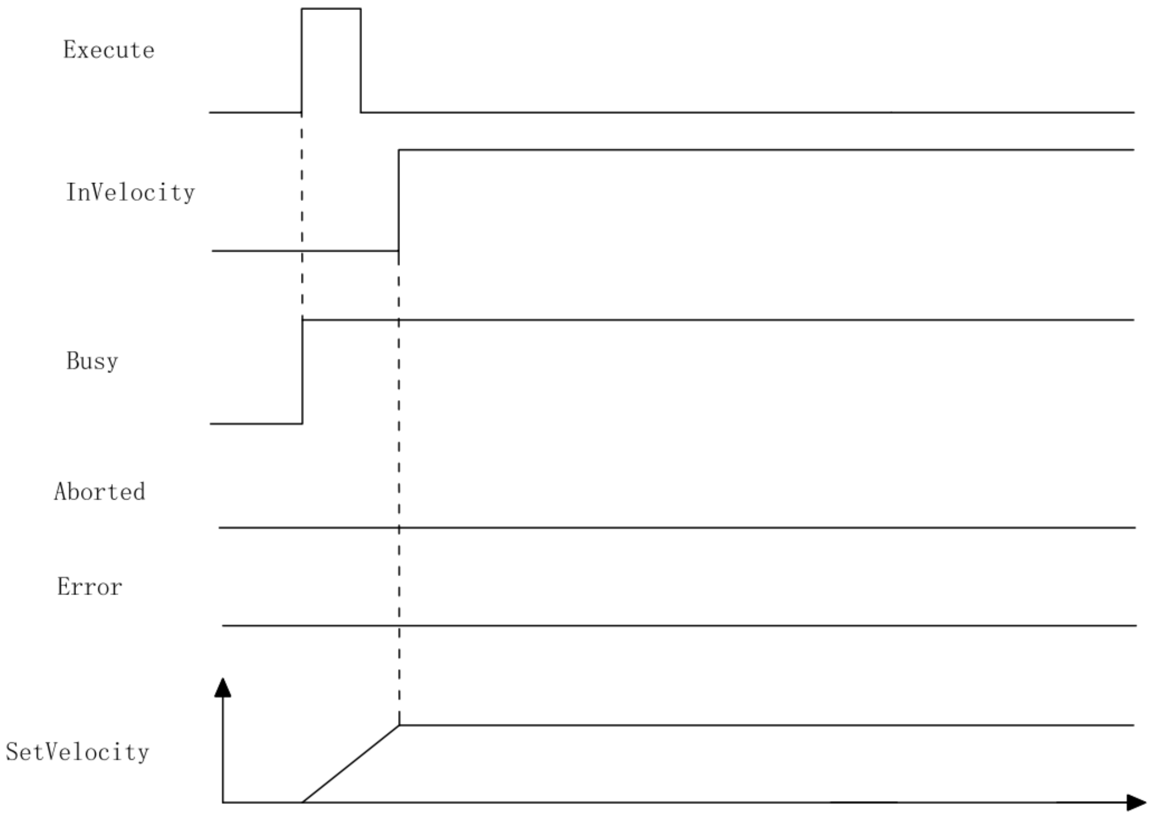
The speed movement function is implemented by the following routines.

1. Set M3000 to ON and enable the axis Axis_0.
2. Set M0 to ON Rear Axle Axis_0 run at the set speed.

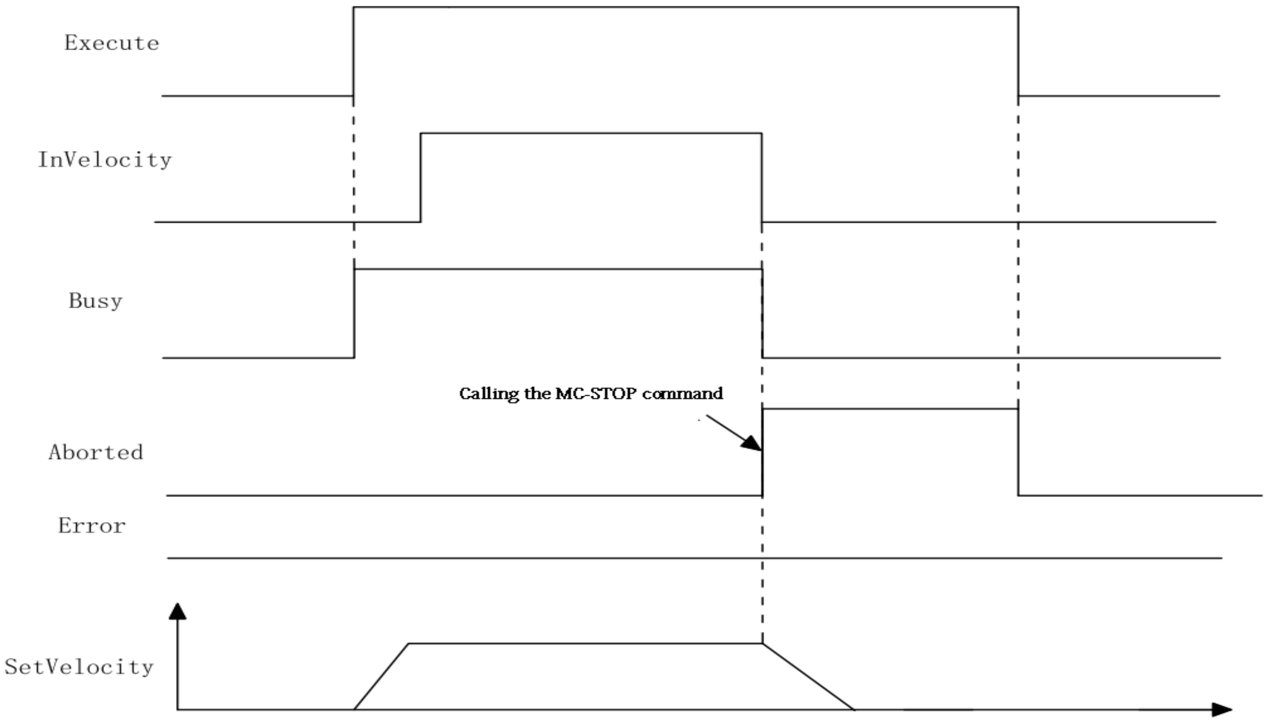


Timing Diagram

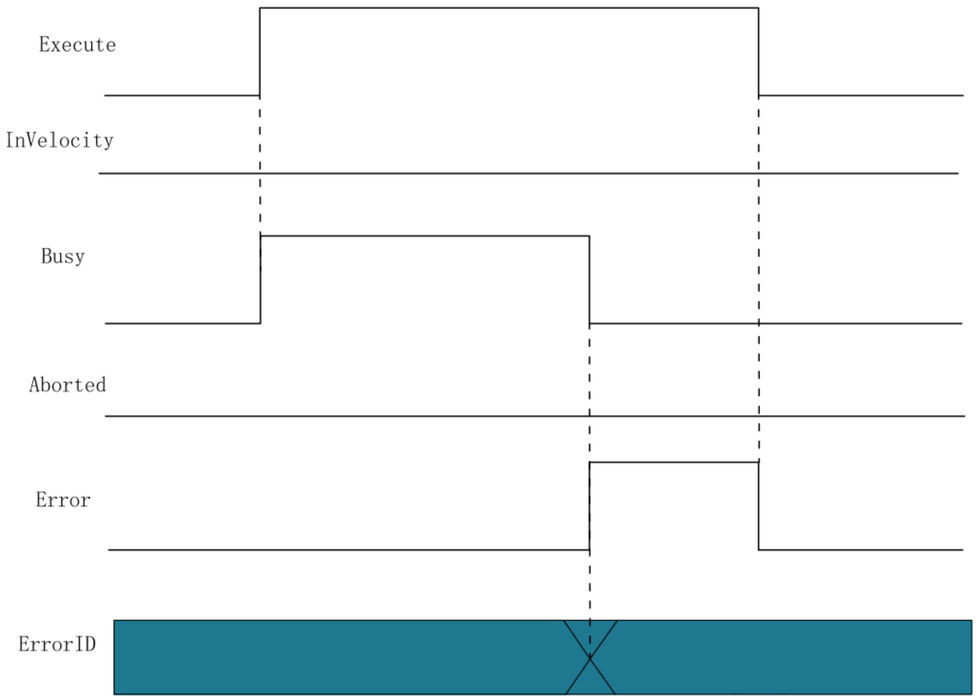
● When the axis is in the StandStill state, call this command to do continuous motion under the T-curve.



● When the shaft is interrupted by Mc_Stop command during motion.

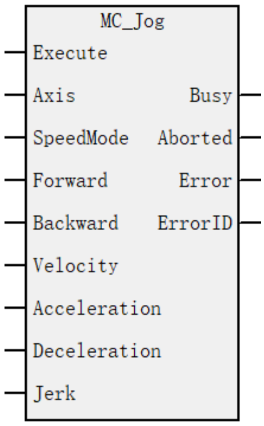


● When the shaft fails during acceleration, the drive fails.



4.3.6 MC_Jog (Jog motion command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Rang	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/number	Execute Valid
IN	SpeedMode	BOOL	M/S	TRUE ,FALSE	Speed type: 0T type, 1S type	Execute Valid
IN	Forward	BOOL	M/S	TRUE, FALSE	Forward movement	Execute Valid
IN	Backward	BOOL	M/S	TRUE, FALSE	Reverse movement	Execute Valid
IN	Velocity	REAL	Constant/D/R/W	Positive Number	The target speed	Execute Valid
IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	Execute Valid
IN	Deceleration	REAL	Constant/D/R/W	Positive Number	Decelerate	Execute Valid
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed step	Execute Valid
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interrupts	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

This command is used to implement the jogging function of the EtherCAT bus axis or the local pulse axis, and the level is active.

The setting of the axis number	Upward Axis latches on the rising edge of the Enable input. If Axis is set to the axis number, modify Axis during Enable=TRUE, and the originally controlled axis enters the ErrorStop state; If Axis is set to the axis number, modifying Axis during Enable=FALSE is invalid.

Functional description	This directive can only be invoked if the axis is switched to the enabled state using MC_Power command. 1. On the rising edge of the command, the function block latches the input parameters such as Velocity, Acceleration, Deceleration, SpeedMode, etc., and switches the state machine of the axis to Continuous Motion mode and enters the jog state. 2. When Enable=TRUE, if you call MC_Stop, MC_MoveRelative or the like, the jog command will be interrupted, and the Aborted output of the jog execution is valid. 3. When JogForward is active, the axis moves forward at the Velocity speed, and when JogBackward is active, the shaft moves backwards at the Velocity speed. When JogForward and JogBackward are valid at the same time， the axis stops moving, the command reports a failure, and the axis stops moving, but it does not enter the ErrorStop state. 4. When Enable=TRUE, the axis runs in one direction and encounters a limit, the command reports a fault, stops the movement of the shaft but does not enter the ErrorStop state. After retriggering the Jog command, the axis can be moved in the opposite direction. 5. SpeedMode is used to set the type of speed curve. SpeedMode = 0 indicates a T-shaped curve in which the speed of the axis will accelerate or decelerate according to the values set by Acceleration and Deceleration. SpeedMode = 1 represents a 5-segment S curve, in which case Acceleration and Degraphic represent the maximum acceleration and minimum deceleration achieved by the shaft during acceleration and deceleration.

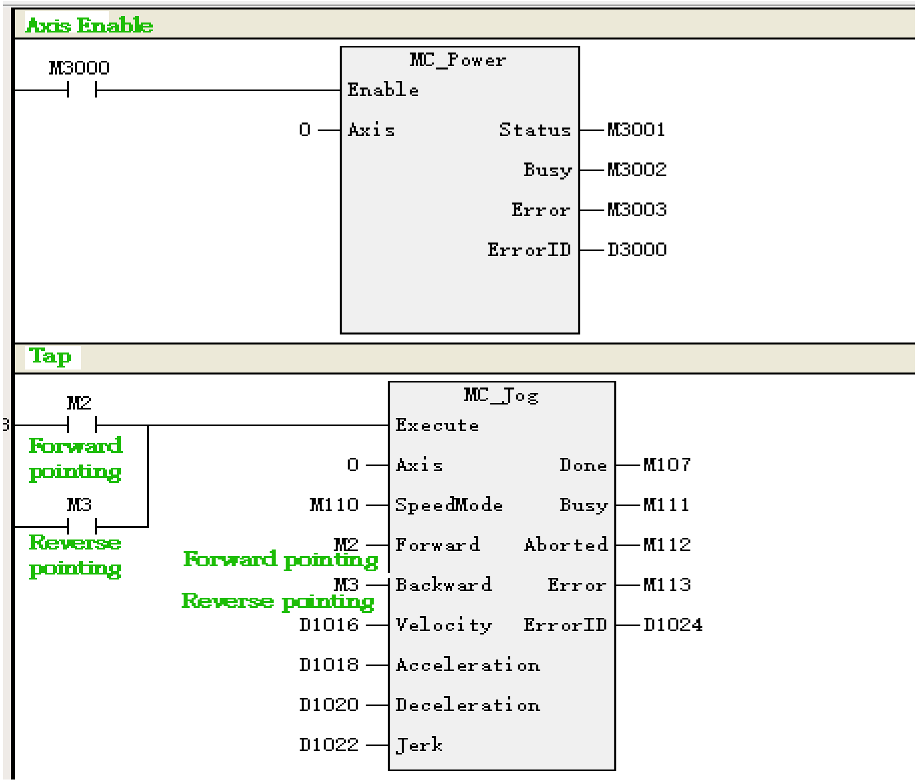
Interrupt

When the command is valid, other motion commands cannot interrupt the instruction, and other motion control commands need to execute MC_Stop command first..

Program examples

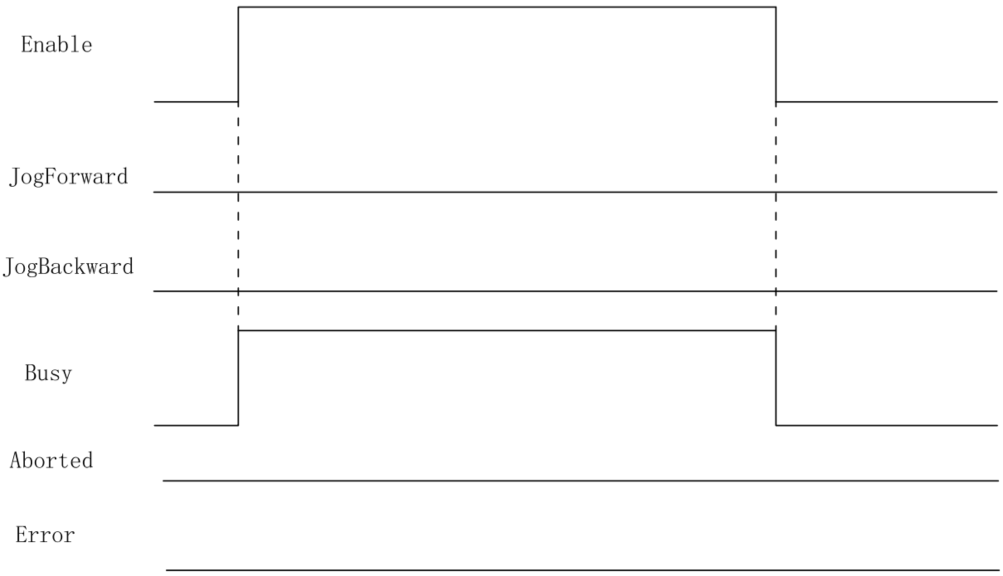
The jog function is implemented by the following routine.

1. Set M3000 to ON and enable the axis Axis_0.
2 .Set M100 to ON, the point-activated energy flow is active.
3. Set M2 to ON Rear Axle Axis_0 run forward at 5Unit/s.
4. Set M2 to OFF and stop running Axis_0 the rear axle.
5. Set M3 to ON Rear Axle Axis_0 run in reverse at 5Unit/s.
6. Set M3 to OFF and stop running Axis_0 the rear axle.

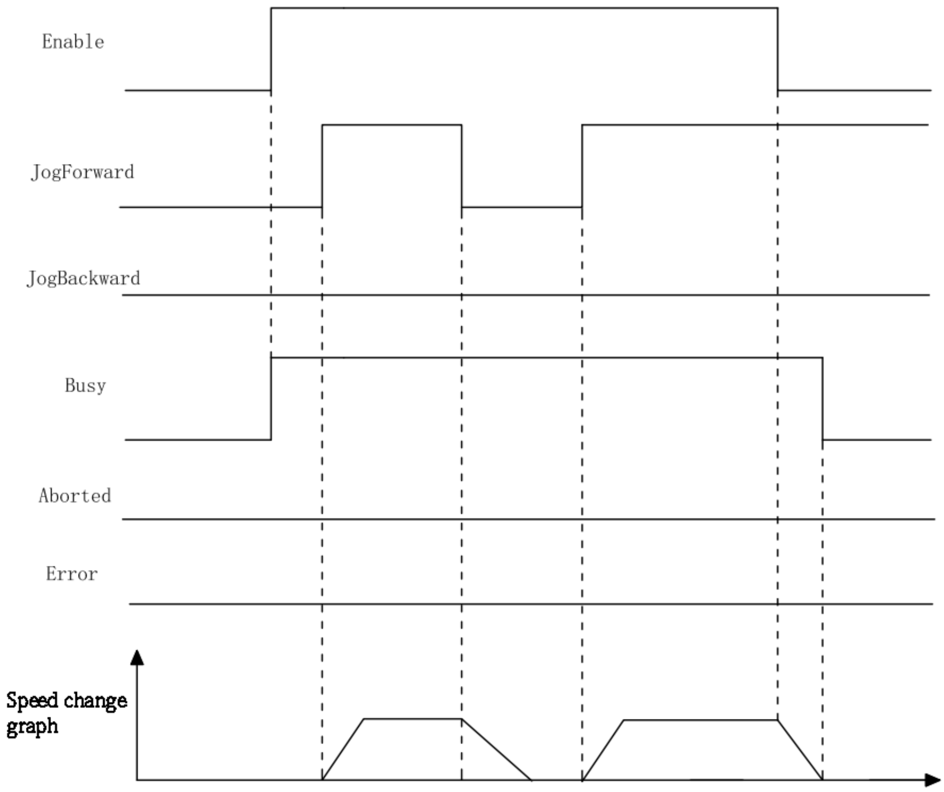


Timing Diagram

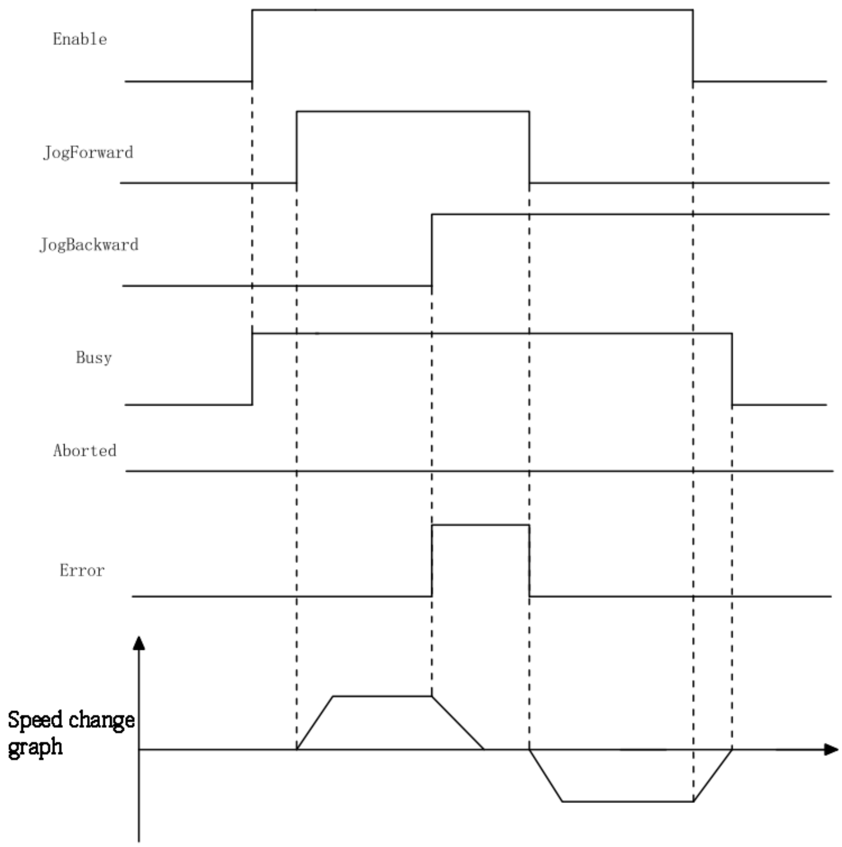
● When only the Enable input is valid, the axis enters the ContinousMotion state and the Busy output is valid.



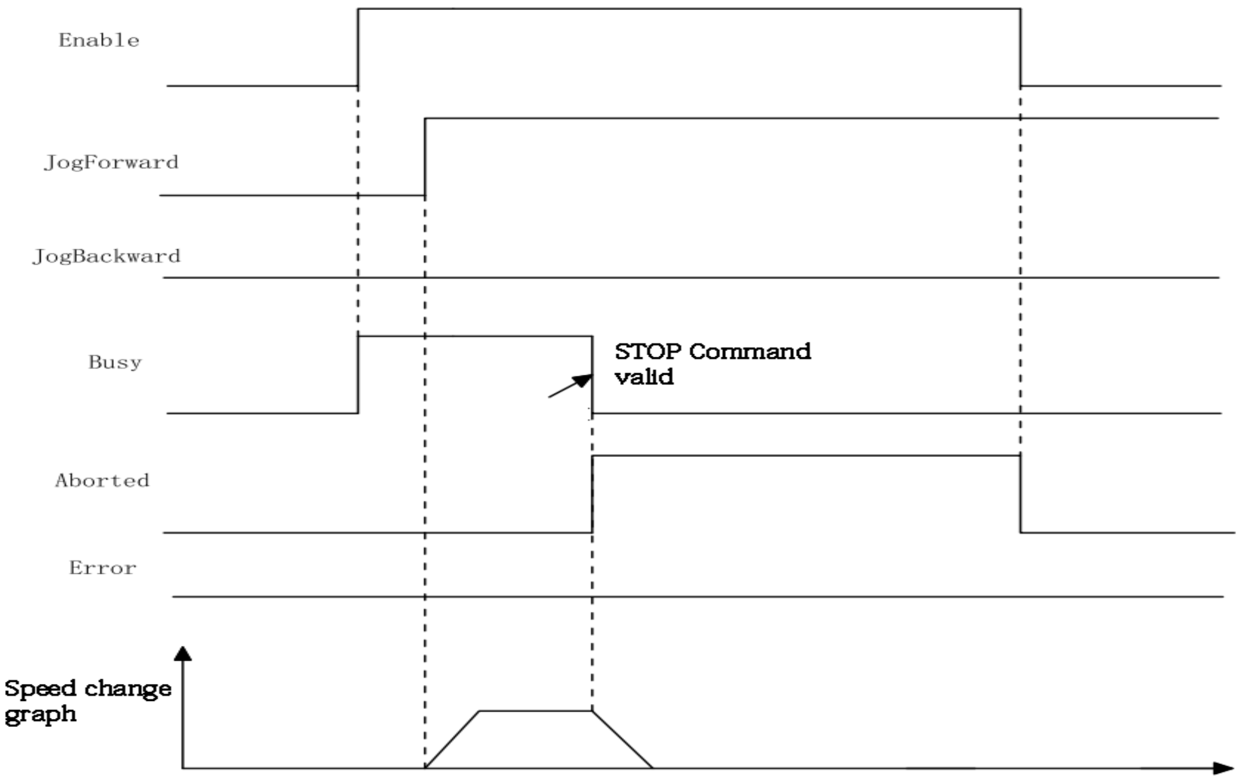
● When the Enable, JogForward input is valid.



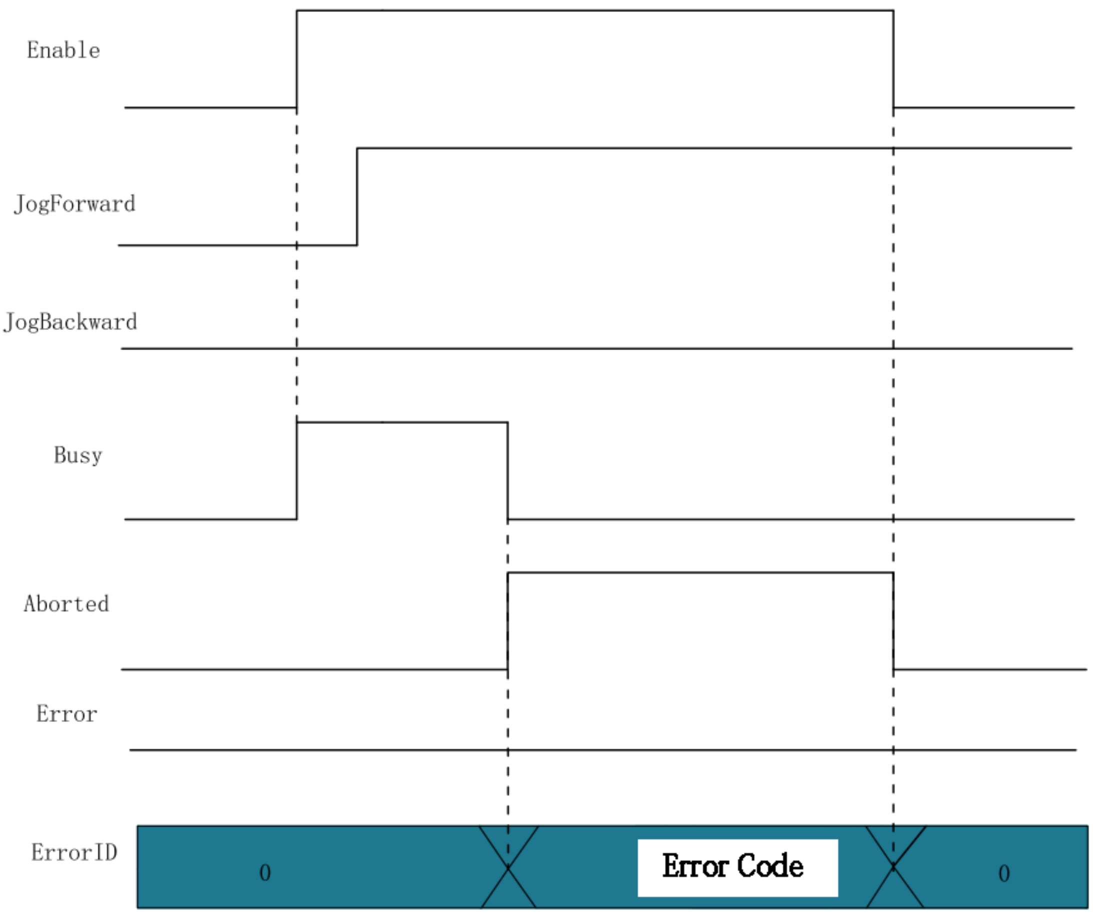
● When the Enable, JogForward inputs are valid, write JogBackward to TRUE.



● When interrupted by the Stop command during execution.

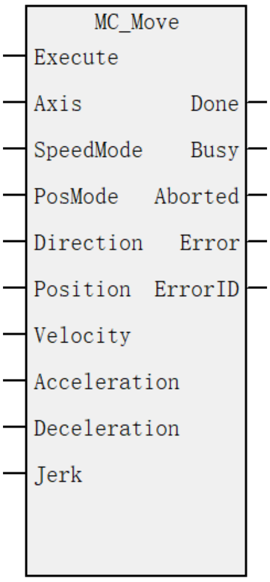


● When the axis is error-corrected.



4.3.7 MC_Move (Positioning command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis number	Execute Rising Edge
IN	SpeedMode	BOOL	M/S	TRUE,FALSE	Speed Type: Type 0T, Type 1S	Execute Rising Edge
IN	PosMode	BOOL	M/S	TRUE,FALSE	Location Type: 0 relative position, 1 Absolute location	Execute Rising Edge
IN	Direction	WORD	Constant/D/R/W	0~3	Positioning direction: (absolute position rotation mode only) 0: Forward (speed greater than 0) 1: Negative (speed less than 0) 2: Shortest distance 3: Current direction	Execute Rising Edge
IN	Position	REAL	Constant/D/R/W	Positive/Negative Number/0	The target location	Execute Rising Edge
IN	Velocity	REAL	Constant/D/R/W	Positive Number	The target speed	Execute Rising Edge

IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	Execute Rising Edge
IN	Deceleration	REAL	Constant/D/R/W	Positive Number	Decelerate	Execute Rising Edge
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed step	Execute Rising Edge
OUT	Done	BOOL	M/S	TRUE,FALSE	Finish stopping	
OUT	Busy	BOOL	M/S	TRUE,FALSE	In-Run flag	
OUT	Aborted	BOOL	M/S	TRUE,FALSE	Execution interrupts	
OUT	Error	BOOL	M/S	TRUE,FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

This command is used to control the EtherCAT bus axis or the local pulse axis to implement the positioning function, effective on the rising edge.

• Absolute or relative value targeting is possible.

• When MoveMode is specified as Absolute Positioning, it is the same as the action of the MC_MoveAbsolute (Absolute Positioning) command. Similarly, when specified as Relative Value Positioning, the action is the same as MC_MoveRelative (Relative Value Positioning) commands.

• The target position and target speed can be modified during the operation of the command.

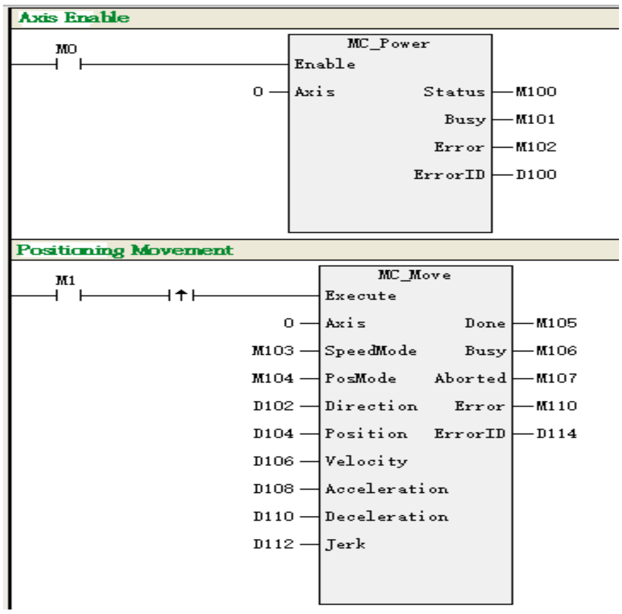
• Direction is not used when positioning command actions as relative values.

Program examples

The jog function is implemented by the following routine.

1. Enable axis Axis_0 after setting M0 to ON.

2. Set M1 to ON rear axle Axis_0 run to the target position at the set target speed, and modify the target position and running speed during operation.

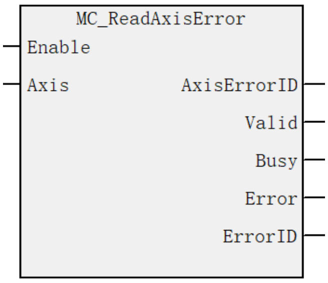


Timing diagram

Omit

4.3.8 MC_ReadAxisError (Read axis error command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Rang	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis number	Enable Vlid
OUT	AxisErrorID	WORD	D/R/W	0X0~0XFFFF	Axis error code: If 0x603F is configured in the PDO, the value of the 0x603f of the EtherCAT bus driver is displayed, otherwise 0 is displayed	
OUT	Valid	BOOL	M/S	TRUE ,FALSE	Valid flags	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

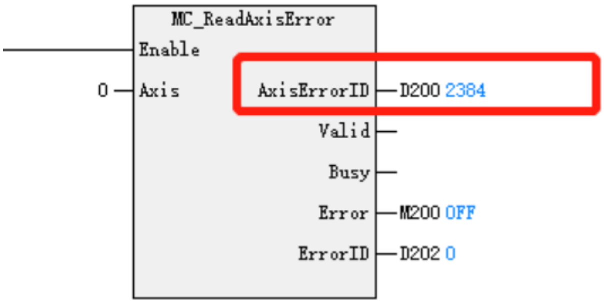
Commands are used to read faults on the EtherCAT bus axis or the local pulse axis.

If 0x603f is configured in the PDO of the bus drive, AxisErrorID is used to display the value of the 0x603F in real time, otherwise 0 is displayed.

If the servo (SD700) reports error code AL.950, the command shows error code 2384 (decimal).

Causes	Treatment measures
Target position 607Ah or actual position 6064h exceeds the limit value of 607Dh	Change the value of 607Dh or 607Ah

As shown in the following figure:



Interrupt

This command does not have a interruption flag and can run multiple commands at the same time.

Error

The command is an error when the axis number is not present;

The command reports an error when the axis initialization fails;

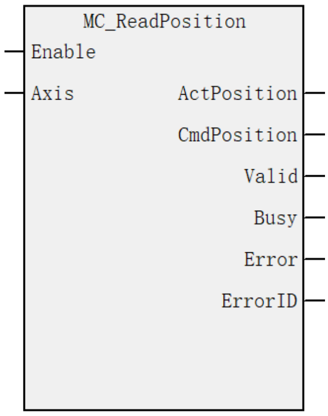
The command reports an error when the shaft type is wrong.

Timing diagram

Omit

4.3.9 MC_ReadPosition (Read position command)

Drawing Block:



Format of Reference List

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis number	Enable Valid
OUT	ActPosition	REAL	D/R/W		The actual position of the axis	
OUT	CmdPosition	REAL	D/R/W		Axis command position	

OUT	Valid	BOOL	M/S	TRUE, FALSE	Valid flags	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

This command is used to read the feedback position of the EtherCAT bus axis or the local pulse axis, and the high level is effective.

When Enable=TRUE, if the PDO in the EtherCAT bus axis is configured with 0x6064, ActPosition displays the feedback position of the axis.

Interrupt

This command does not have a interruption flag and can run multiple commands at the same time.

Error

The command reports an error when the axis number is not present.

The command reports an error when axis initialization fails.

The command reports an error when the shaft type is wrong.

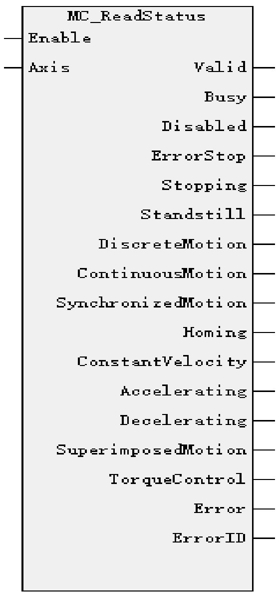
Command error is reported when there is no configuration 0x6064 in the PDO of the EtherCAT bus axis.

Timing diagram

Omit

4.3.10 MC_ReadStatus (Read axis status command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~63	Axis name/axis	Enable

					number	
OUT	Valid	BOOL	M/S	TRUE, FALSE	Valid	
OUT	Busy	BOOL	M/S	TRUE ,FALSE	Busy sign	
OUT	Disabled	BOOL	M/S	TRUE, FALSE	PLCOpen state machine, de-enabled	
OUT	ErrorStop	BOOL	M/S PLCOpen state machine, stop running	TRUE, FALSE	PLCOpen statu machine, fault stop	
OUT	Stopping	BOOL	M/S	TRUE, FALSE	PLCOpen state machine, stop running	
OUT	Standstill	BOOL	M/S	TRUE, FALSE	PLCOpen state machine, enabled and not running	
OUT	DiscreteMotion	BOOL	M/S	TRUE, FALSE	PLCOpen state machine, point running mode	
OUT	ContinuousMotion	BOOL	M/S	TRUE, FALSE	PLCOpen state machine, continuous running mode	
OUT	SynchronizedMotion	BOOL	M/S	TRUE, FALSE	PLCOpen state machine, synchronous running mode	
OUT	Homing	BOOL	M/S	TRUE, FALSE	PLCOpen state machine, homing mode	
OUT	ConstantVelocity	BOOL	M/S	TRUE, FALSE	Axis velocity is 0 Axis is moving at constant speed Invalid in torque mode	
OUT	Accelerating	BOOL	M/S	TRUE, FALSE	The axis is moving with acceleration (the absolute value of the velocity is increasing) Invalid in torque mode	

OUT	Decelerating	BOOL	M/S	TRUE ,FALSE	Axis is doing deceleration motion (absolute value of speed is increasing) Invalid in torque mode	
OUT	SuperimposedMotion	BOOL	M/S	TRUE, FALSE	In superposition or phase shift	
OUT	TorqueControl	BOOL	M/S	TRUE, FALSE	Torque control mode	
OUT	Error	BOOL	M/S	TRUE, FALSE	Wrong sign	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

When Enable=TRUE, this command will read the status and acceleration and deceleration status of the PLCOpen state machine of the shaft.

In torque mode, the parameters ConstantVelocity, Acceleration and Deceleration are always OFF.

The EtherCAT task has a higher priority than the PLC master task. If the state of an axis is present in the EtherCAT task for only one EtherCAT cycle, the state cannot be acquired in the PLC master task.

Interrupt

This command does not have a interruption flag and can run multiple commands at the same time.

Error

The command reports an error when the axis number is not present.

The command reports an error when axis initialization fails.

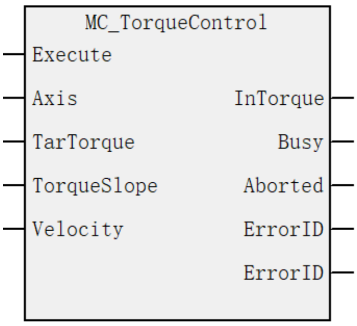
The command reports an error when the shaft type is wrong.

Timing diagram

Omit

4.3.11 MC_TorqueControl (Torque control command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis	Execute

					number	Valid
IN	TarTorque	REAL	Constant/D/R/W	Positive/Negative Number/0	Target torque (in 1% units))	Execute Valid
IN	TorqueSlope	REAL	Constant/D/R/W	Positive number	Torque ramp (in 1% units))	Execute Valid
IN	Velocity	REAL	Constant/D/R/W	Positive number	Maximum speed limit	Execute Valid
OUT	InTorque	BOOL	M/S	TRUE, FALSE	Torque arrival sign: The output is valid when less than 5%.	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interrupts	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

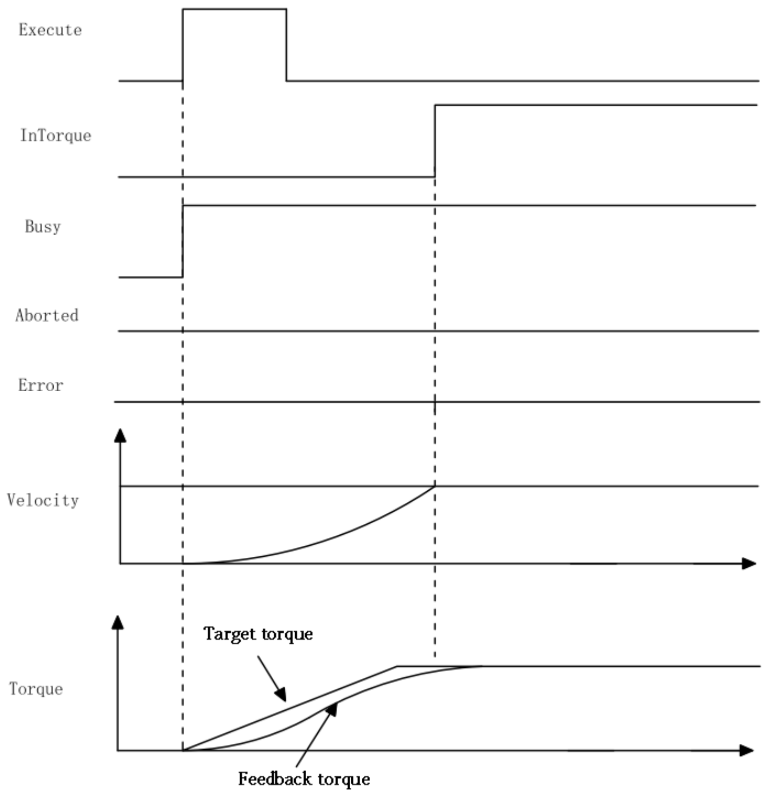
This directive is used to implement the torque control function and is only used for bus lines EtherCAT bus axis. The command rises edge is valid and does not support virtual axis mode.

The setting of the axis number	Axis is latched along the rising edge of the Execute input. During execution, modification to other axis numbers to report errors. After execution is completed and energy flow is valid, modify to other axis number to report an error. After execution is completed and energy flow is invalid, modify to other axis number is invalid and wait for rise to be valid.
Functional description	This directive can only be invoked if the axis is switched to the enabled state using MC_Power command. The torque command requires the drive mapping 0x6040, 0x6041, 0x6060, 0x6061, 0x6071, 0x6077 several PDOs to be used, otherwise the fault is reported. This command adopts the synchronous torque mode of the drive to realize the torque control function. On the rising edge of the command, the function block latches the TarTorque, TorqueSlope and Velocity input parameters, and the axis is in the Continuous Motion state and does the torque movement. ■ TarTorque：The target torque, in 1%, the program decimal place after the decimal point is valid, the direct discard behind, the actual moment of the drive is limited by the maximum positive and negative torque set in the configuration parameters. ■ TorqueSlope: Torque slope, the unit is 1%, the decimal place after the decimal point in the program is valid, and the subsequent direct discard.
Speed control in torque mode	For Veichi Servo Drive, if 0x607f is mapped in the axis parameter, this command limits the maximum speed of the servo motor by 0x607f, and if 0x607f is not mapped, the speed limit is invalid.

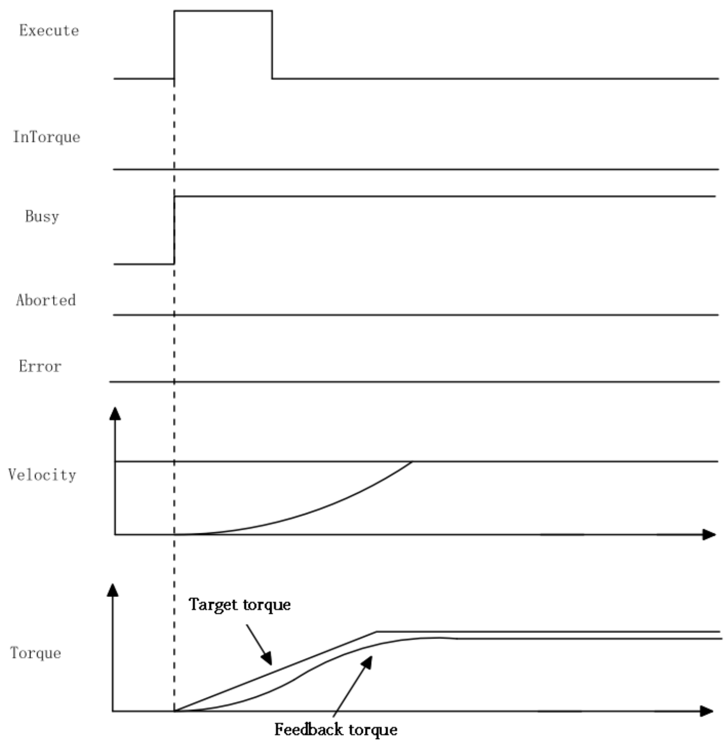
	On the rising edge of Execute, the command converts the Velocity set limit speed into pulse units by writing to 0x607f via PDO. When the torque command is interrupted by another command, the maximum speed of the axis is limited by the "Maximum Speed" item in the configuration screen. For third party drives, Velocity can only be used as a speed limit if the following conditions are met: ■The maximum speed of the servo motor can be limited by 0x607F. ■ 0x607F can be configured into the PDO. ■ The unit of 0x607F is pulse unit, not speed unit.
Stop control in torque mode	The MC_Stop command is called in the torque mode to execute the stop operation. After receiving the Stop command, the drive switches to the synchronous position mode and executes the deceleration according to the deceleration speed set by the Stop command.

Timing diagram

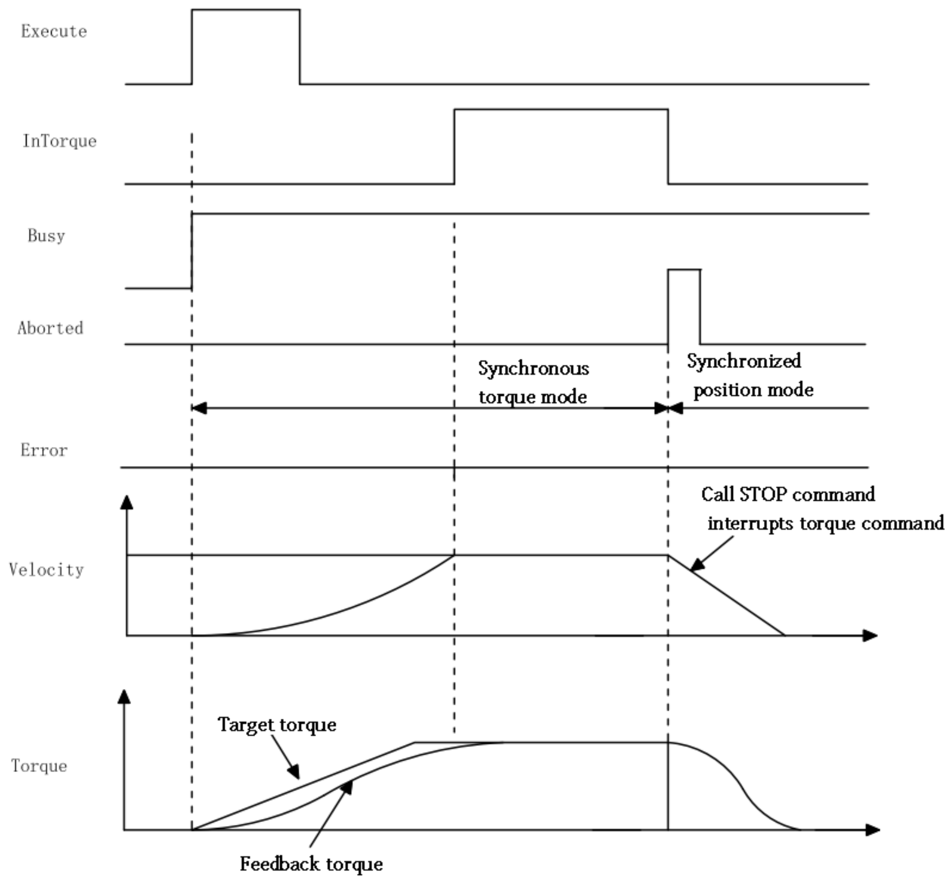
● Trigger the command after setting the target torque, in the case that the actual output torque can reach the target torque.



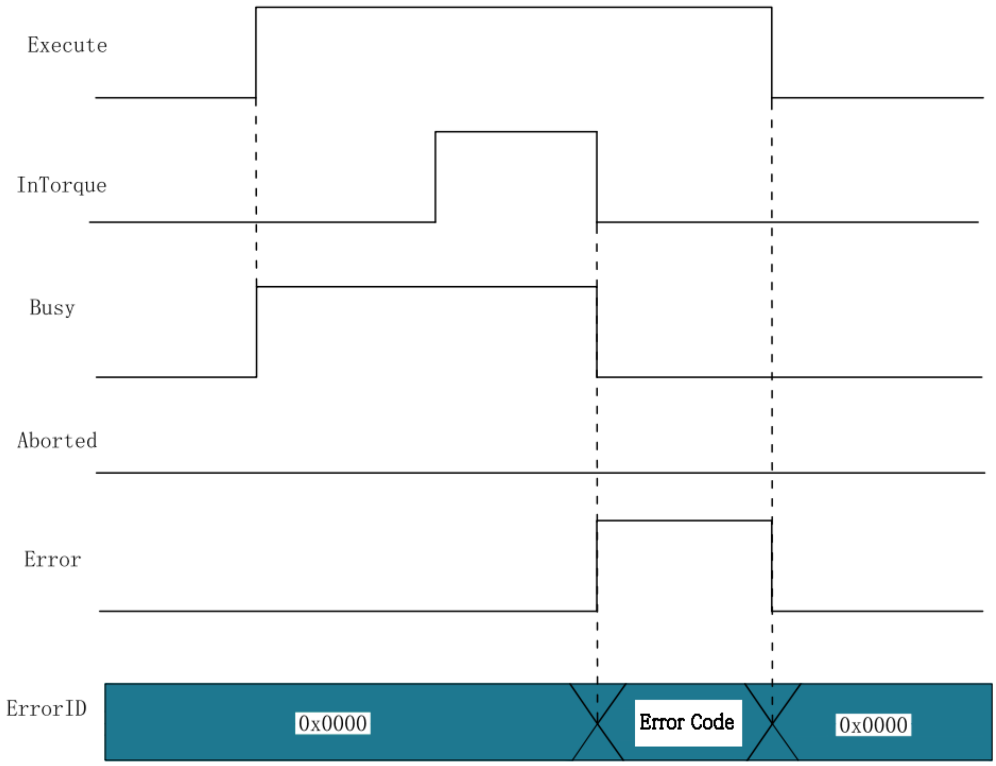
● After setting the target torque, the command is triggered, and finally the actual output torque cannot reach the target torque.



● During the torque operation, the Mc_Stop command interrupts the operation of the command.

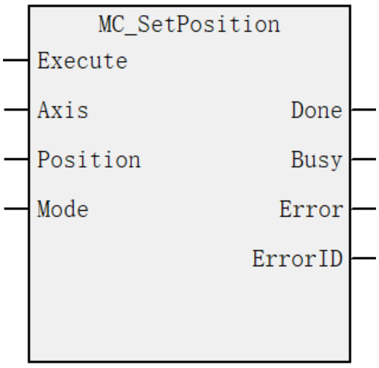


● During the torque operation, the drive is error-reported.



4.3.12 MC_SetPosition (Set position command)

Drawing Block:



Format of Reference List

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis number	Execute Rising Edge
IN	Position	REAL	Constant/D/R/W	Positive/Negative Number/0	location	Execute Rising Edge
IN	Mode	BOOL	M/S	TRUE FALSE	Mode selection: 0: Absolute mode, will Position The value is written	ExecuteRising Edge

					to the current location 1: Relative mode, adding the value of Position on the basis of the current position	
OUT	Done	BOOL	M/S	TRUE, FALSE	Finish flag	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

This command is used to set the current position of the EtherCAT bus axis or the local pulse axis, with the rising edge active.

Only when the PLCOpen state of the axis is in the disabled, standstill, and errorstop states can the current position of the execution setting axis be called, and other status commands report errors.

When Mode=0 (absolute mode), on the rising edge of Execute, the command writes Position to the current position of the axis.

When Mode=1 (relative mode), on the rising edge of Execute, the command adds Position to the current position of the axis.

Interrupt

This command does not support interruption, if there are several commands at the same time in a scan cycle, the command that is valid first will be executed, and if there are other SetPosition commands executed during the validity of the command Busy signal, the other commands will report errors.

Error

The command reports an error when the axis number is not present.

The command reports an error when the shaft type is wrong.

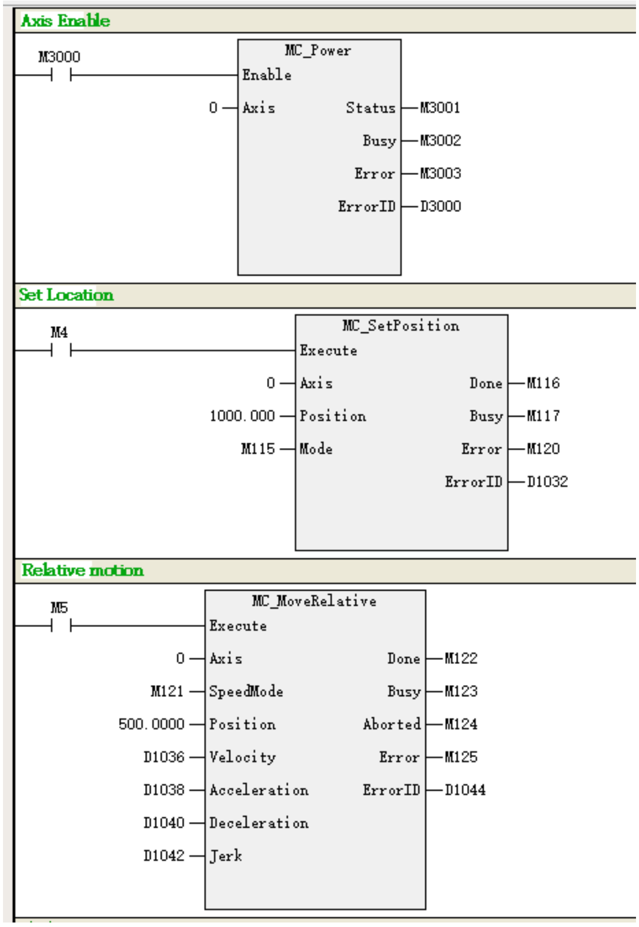
The command reports an error when axis initialization fails.

This command can only be set to take effect when the shaft is down and is in other states.

Program examples

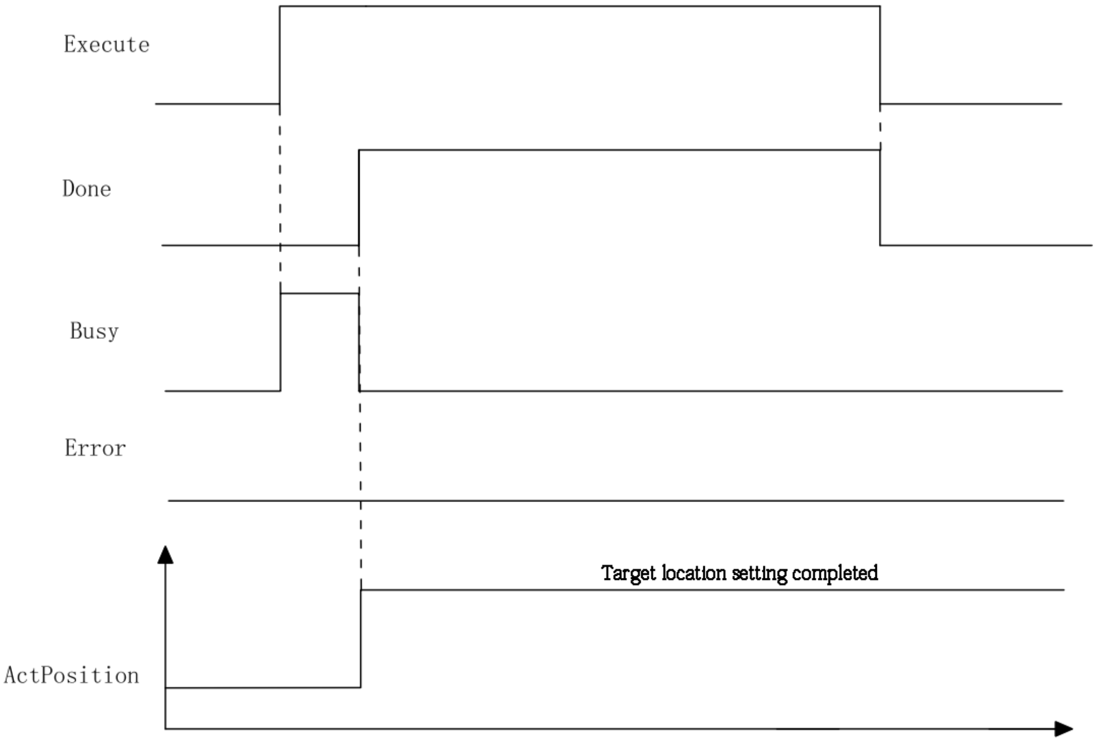
The Set Location function is implemented through the following routines.

1. Set M3000 to ON and enable the axis Axis_0.
2. After setting M4 to ON, the command position of the axis Axis_0 is set to 1000.
3. After setting M5 to ON, the shaft Axis_0 to stop at the set speed to the command position 1500.

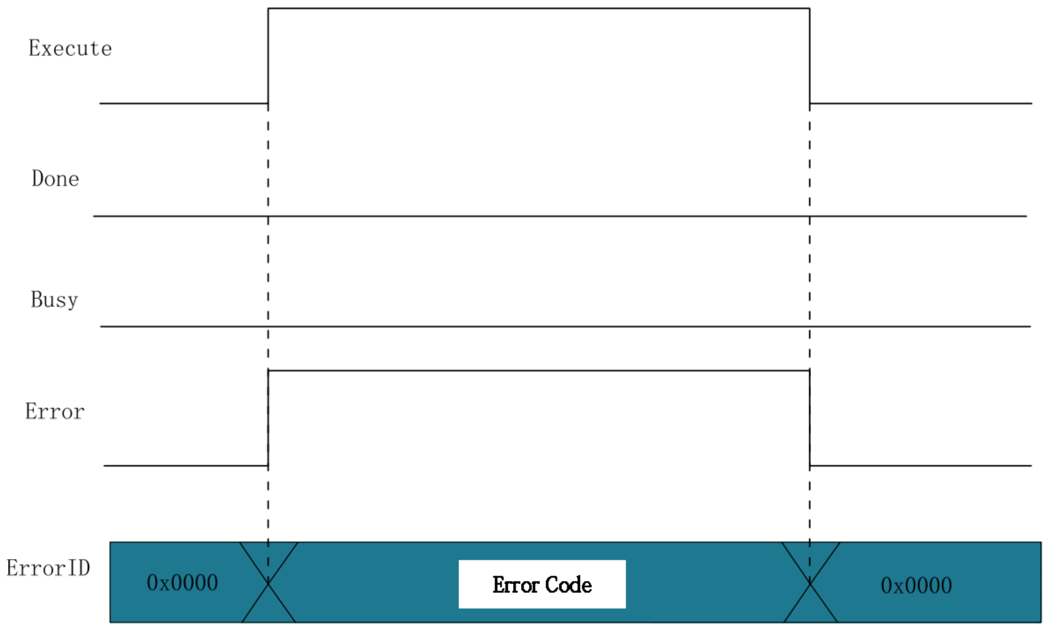


Timing diagram

● The axis executes this command in standstill, relative mode.

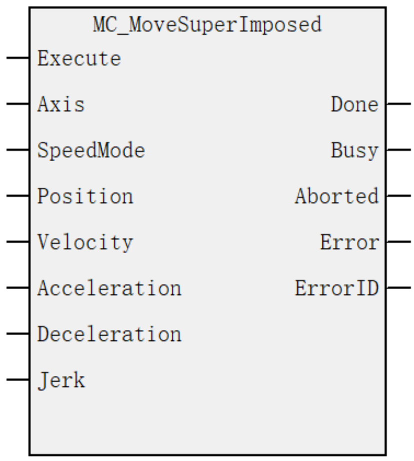


● The axis executes this command while the jog command is valid.



4.3.13 MC_MoveSuperImposed (Displacement overlay command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis number	Execute Rising Edge
IN	SpeedMode	BOOL	M/S	TRUE, FALSE	Speed Type: Type 0T, Type 1S	Execute Rising Edge
IN	Position	REAL	Constant/D/R/W	Positive/Negative Number/0	The target location	Execute Rising

						Edge
IN	Velocity	REAL	Constant/D/R/W	Positive Number	The target speed	Execute Rising Edge
IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	Execute Rising Edge
IN	Deceleration	REAL	Constant/D/R/W	Positive Number	Decelerate	Execute Rising Edge
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed step	Execute Rising Edge
OUT	Done	BOOL	M/S	TRUE, FALSE	Finish stopping	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Aborted	BOOL	M/S	TRUE,FALSE	Execution interrupts	
OUT	Error	BOOL	M/S	TRUE,FALSE	Error flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

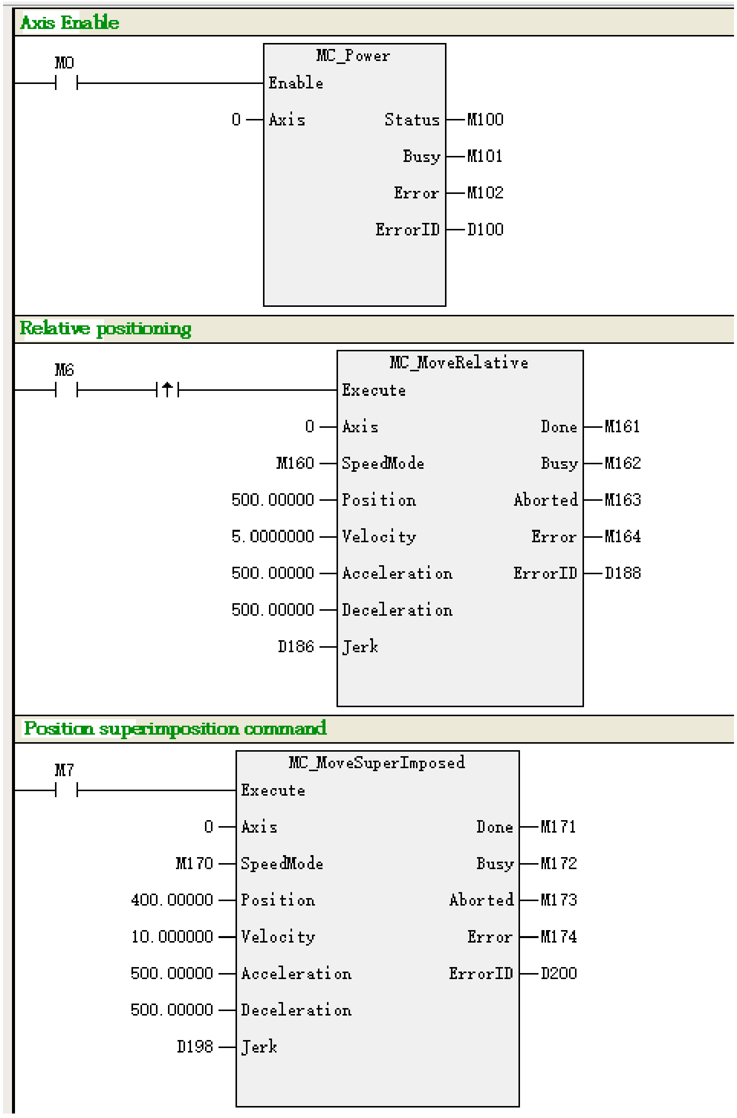
This command is to overlay+ the position and velocity on the original continuous motion displacement position at the same time, this command can be used for discrete_ motion、continuous_ Motion and synchronized_ Motion state.

In StandStill status, the action of this function block is similar to that of MC_ MoveRelative.

Program examples

Taking relative motion as an example, the displacement overlay function is implemented through the following routine.

1. Enable axis Axis_0 after setting M0 to ON.
2. After setting M6 to ON, the axis Axis_0 move relative at a speed of 5Unit/s, and the target position is 500.
3. After setting M7 to ON, the axis Axis_0 move relative at the speed of (5+10)Unit/s, and the target position is superimposed as (500+400).

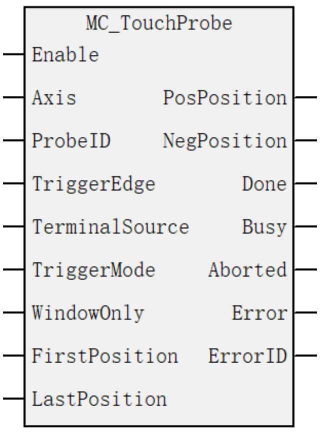


Timing Diagram

Omit

4.3.14 MC_TouchProbe (Probe command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis number	Execute Rising Edge
IN	ProbeID	WORD	Constant/D/R/W	0 - 1	Probe ID：	Execute Rising Edge
IN	TriggerEdge	WORD	Constant/D/R/W	0~2	Edge trigger mode: 0 rising edge 1 falling edge 2 Any edge	Execute Rising Edge
IN	TerminalSource	WORD	Constant/D/R/W	0~1	Probe source (for setting up bus servo drivers only) 0: DI terminal 1: Z pulse	Execute Rising Edge
IN	TriggerMode	WORD	Constant/D/R/W	0~1	Trigger Type: 0: Single trigger 1: Continuous triggering	Execute Rising Edge
IN	WindowOnly	BOOL	M/S	TRUE ,FALSE	Enable the probe window	Execute Rising Edge
IN	FirstPosition	REAL	Constant/D/R/W	Positive/Negative Number/0	The probe window start position	Execute Rising Edge
IN	LastPosition	REAL	Constant/D/R/W	Greater than First Position	The end position of the probe window	Execute Rising Edge
OUT	PosPosition	REAL	D/R/W	Positive/Negative Number/0	The rising edge captures the position	
OUT	NegPosition	REAL	D/R/W	Positive/Negative Number/0	Descending along the capture position	
OUT	Done	BOOL	M/S	TRUE FALSE	Finish stopping	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interrupts	
OUT	Error	BOOL	M/S	TRUE,	Error Flag	

				FALSE		
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

This command is used to implement probe functionality for EtherCAT bus axis or local pulse axis, and is effective at high levels. This command does not support virtual axis mode.

In EtherCAT bus axis mode, the driver needs to be configured with probe function (0x60b8), probe status (0x60b9), and latch position (0x60ba/0x60bb/0x60bc/0x60bd).

In local pulse axis mode, a probe source needs to be configured.

● On the rising edge of the command, the command will latch the input parameters on the left side of ProbeID, TriggerEdge, etc., and other status update parameters are invalid.

● At Enable=TRUE, the command blocks latch the current position of the shaft when it detects that the probe input specified by ProbeID is valid and that the probe detection condition is met.

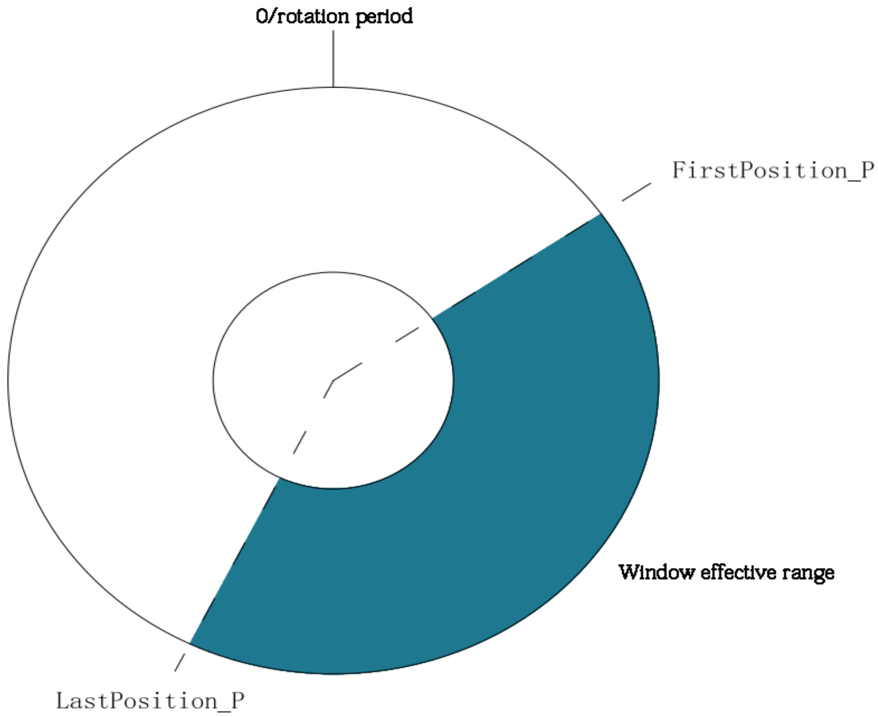
● WindowOnly = FALSE, the window detection function is invalid. As long as the probe input signal is valid, the position of the shaft when the probe signal is valid can be latched.

● WindowOnly = TRUE, the window detection function is effective.

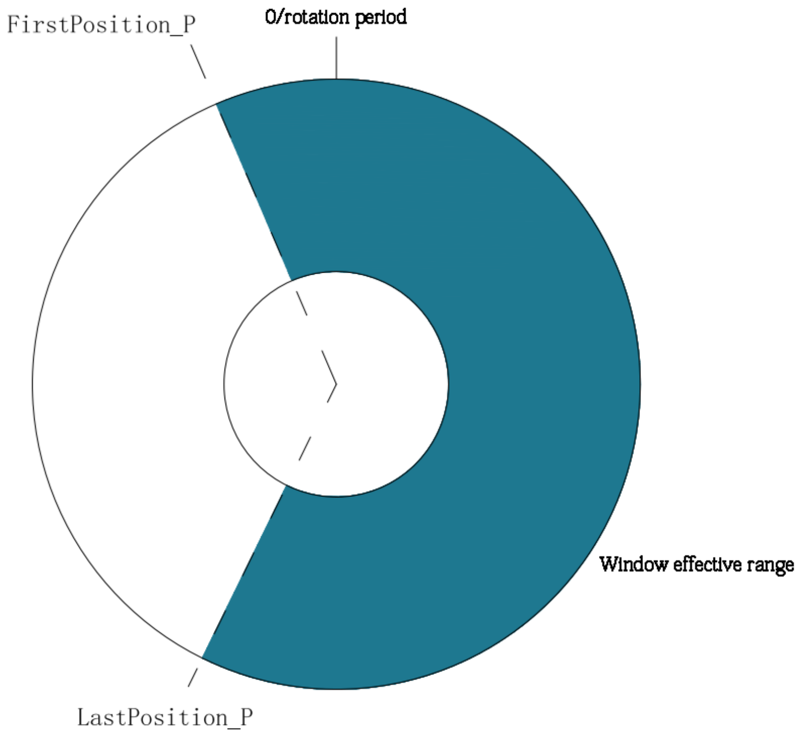
● In linear mode, the command detects the probe signal only if the current position of the axis is inside the interval set by FirstPosition and LastPosition.

● In the ring mode, the command first evaluates the cycle cycle with FirstPosition and LastPosition to obtain the interval position FirstPosition_p and LastPosition_p within a cycle.

■ When FirstPosition_p < LastPosition_p, the effective range of the window is shown in the figure：



■ Then FirstPosition_p > LastPosition_p， the valid range is shown in the figure：



This command can detect the rising and falling edges of the probe signal separately, or simultaneously on the rising and falling edges.

When only the rising edge (falling edge) is detected, the command writes the detected value on the rising edge (falling edge) to the PosPosition (NegPosition), at which point a detection cycle completes to assert the Do signal.

If both the rising and falling edges are detected, the position is written immediately after the command's Enable is valid, when the command detects the rising edge PosPosition，Writing the position to the NegPosition immediately after detecting a falling edge counts as a full detection cycle to output the Done signal, where there is no requirement for the input order of the rising and falling edges.

● For EtherCAT bus drivers, the input TerminalSource of this command can be used to set the terminal type to the DI input terminal or the Z signal of the motor (driver support required). If the driver does not support the Z signal, this command does not report an error.

● This command can be triggered single or continuously. When a single trigger is selected, the output of the done signal is valid to indicate the end of the command execution; When the continuous trigger mode is selected, the Done outputs a valid signal and resets after a PLC scan cycle, and the command automatically begins to detect the new probe input signal.

Bus probes:

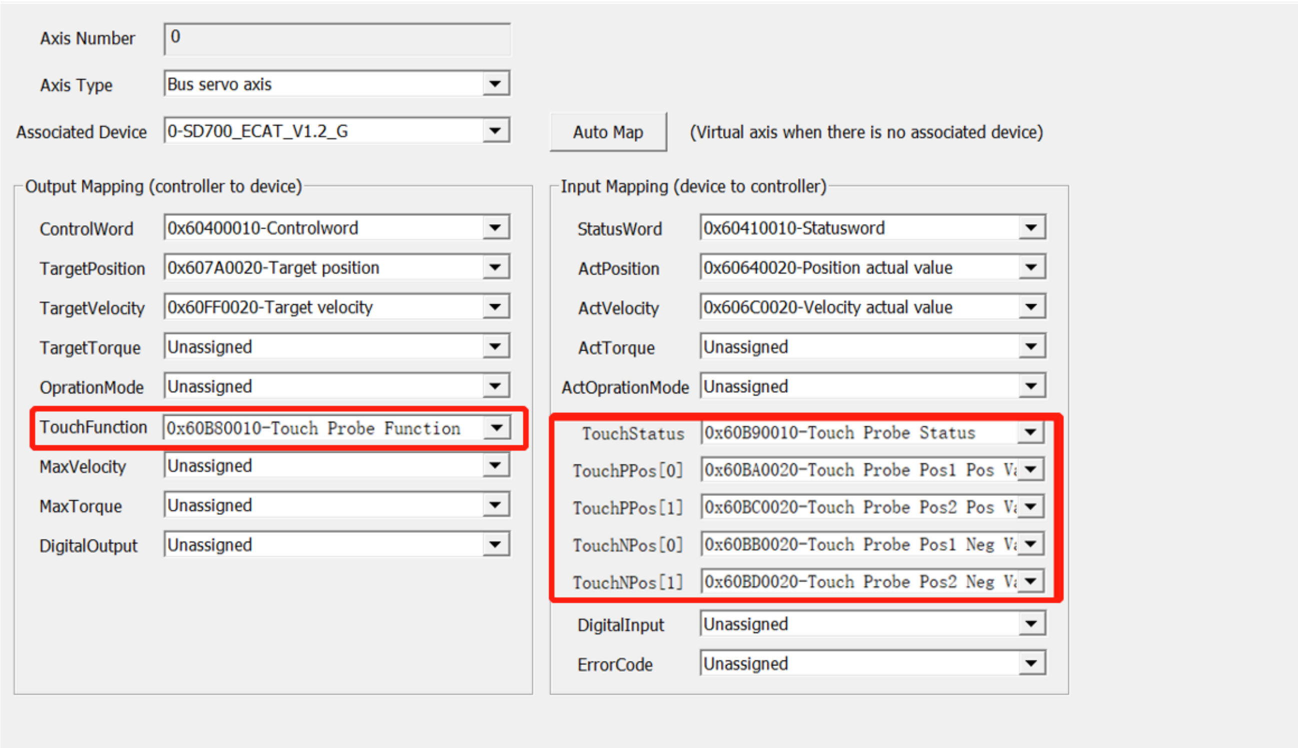
Bus servo drive probe, need to configure the relevant object dictionary and drive DI point function settings, take Flextron SD710 servo drive as an example.

1. Object dictionary related to probes:

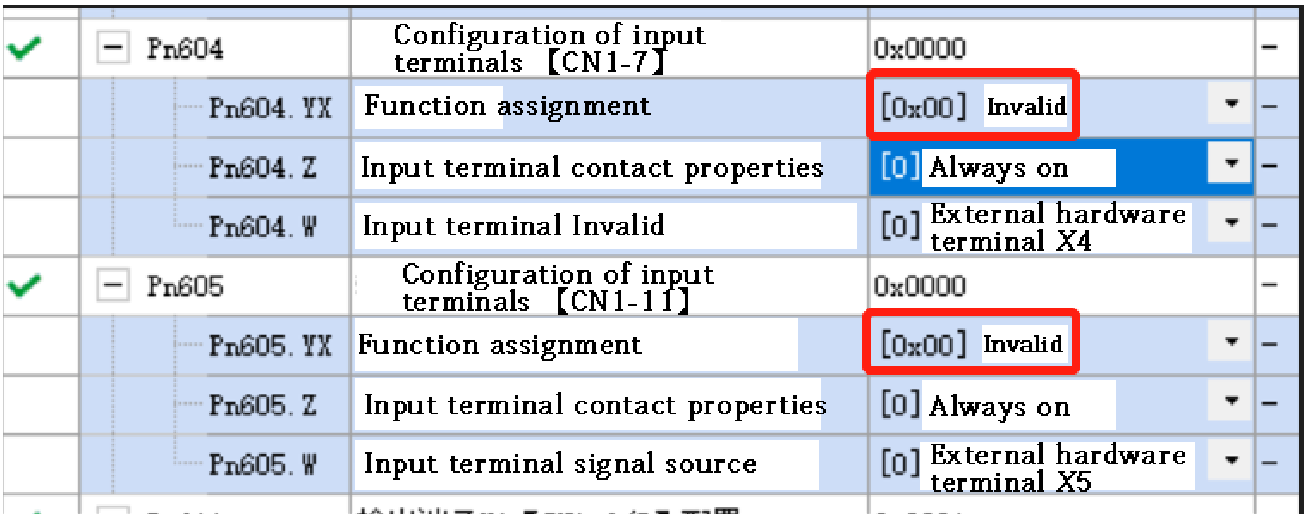
Indexes	Name	Reading and writing	Data type	Unit	Setting range
0x60B8	Probe function	RW	UNIT16	-	0~65535
0x60B9	Probe status	RO	UNIT16	-	-
0x60BA	Probe 1 rising edge latch	RO	INT32	Command unit	-

	position				
0x60BB	Probe 1 falling edge latch position	RO	INT32	Command unit	-
0x60BC	Probe 2 rising edge latch position	RO	INT32	Command unit	-
0x60BD	Probe 2 falling edge latch position	RO	INT32	Command unit	-

2. After adding the PDO, it is automatically mapped to the associated device SD710:



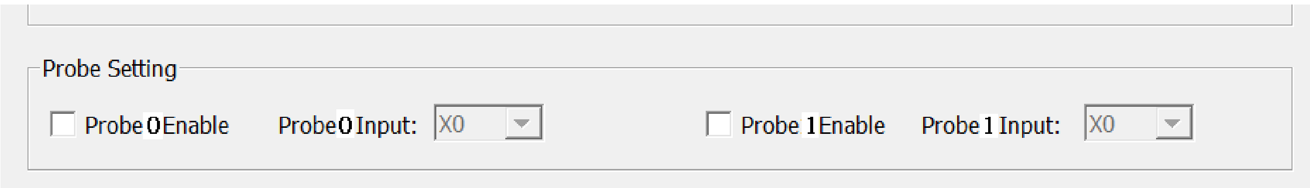
3. DI input terminal probe function setting:



After setting 1, 2, and 3, the bus probe command is available.

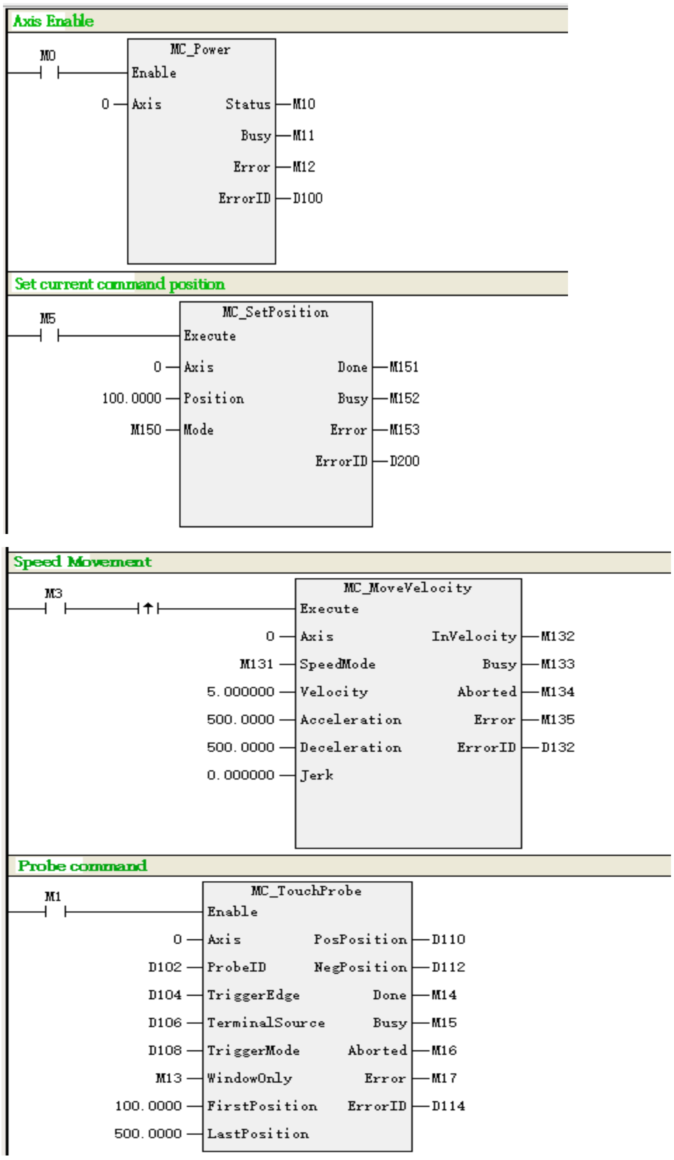
Program examples:

Taking the local pulse axis as an example, the software selects probe 1 to enable and the input selects X0 to implement the probe function through the following routine.



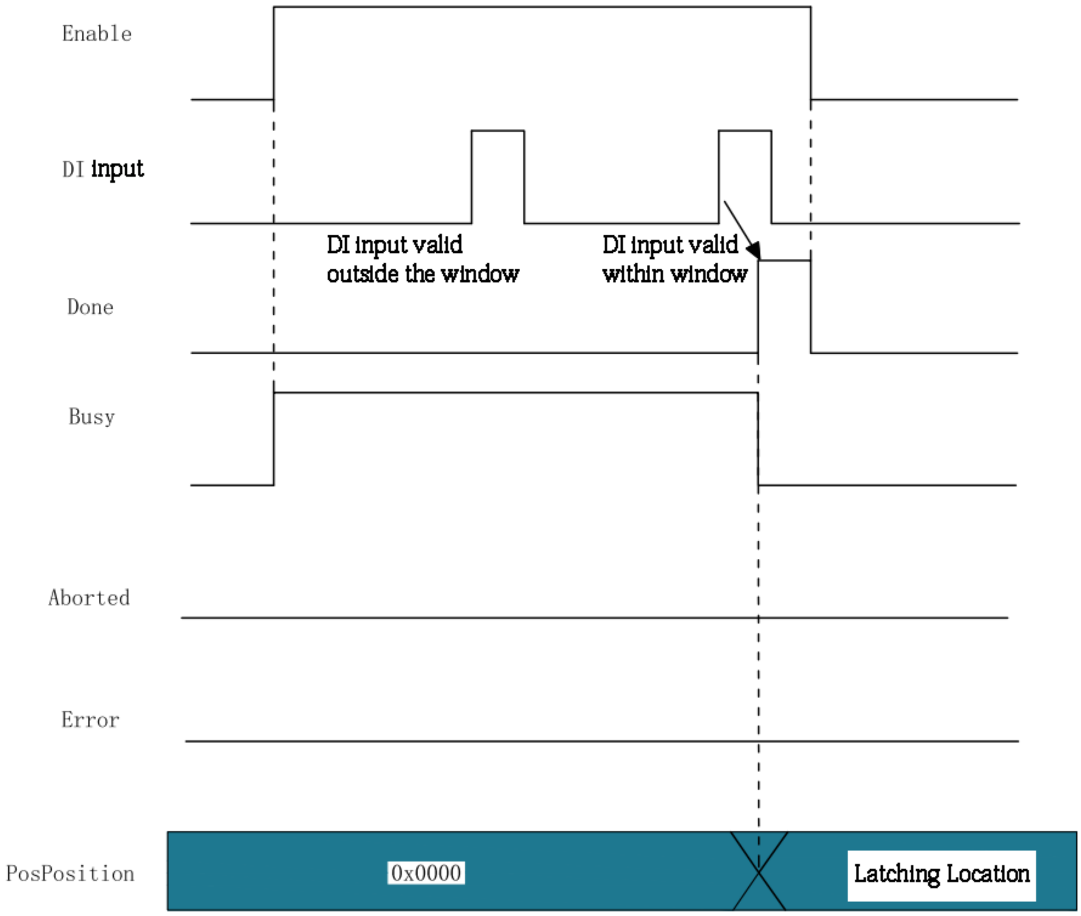
Probe commands: ProbeID=0 (Probe 1), TriggerEdge=2 (Trigger on any edge), TriggerMode=1 (Continuous Trigger), WindowOnly=1 (Probe window enabled).

1. Enable axis Axis_0 after setting M0 to ON.
2. After setting M5 to ON, the command position of the axis Axis_0 is set to 100.
3. After setting M3 to ON, the shaft Axis_0 run at the set speed of 5Unit/s.
4. After setting M3 to ON, in the command position is between 100 and 500, the rising edge of X0 is encountered, and the current position of PosPosition is latched and X0 is encountered on the descending edge, Neg Position latches the current position.

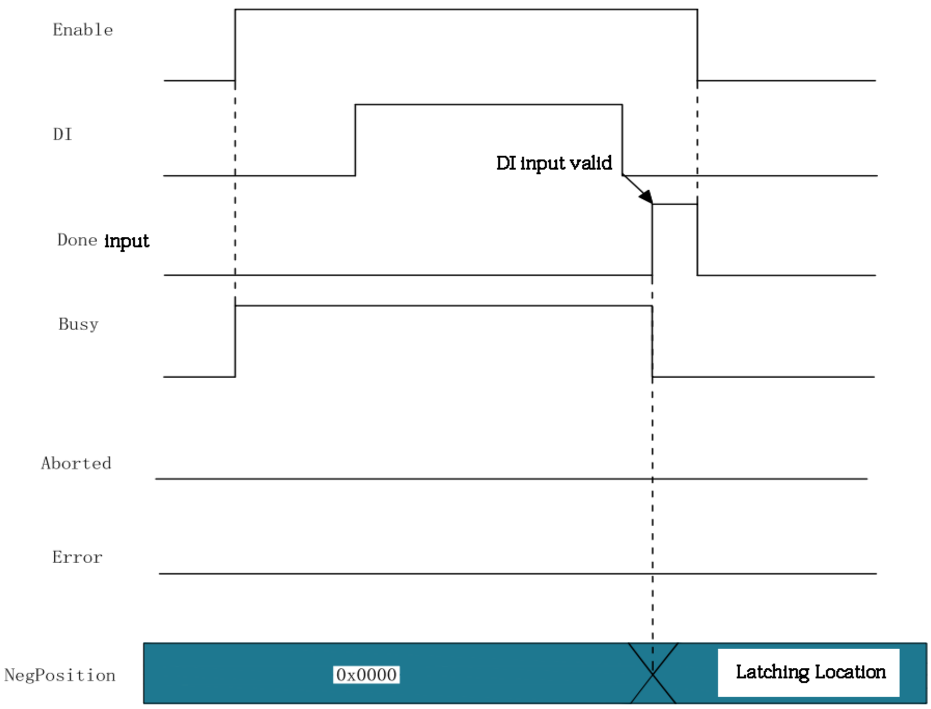


Timing chart:

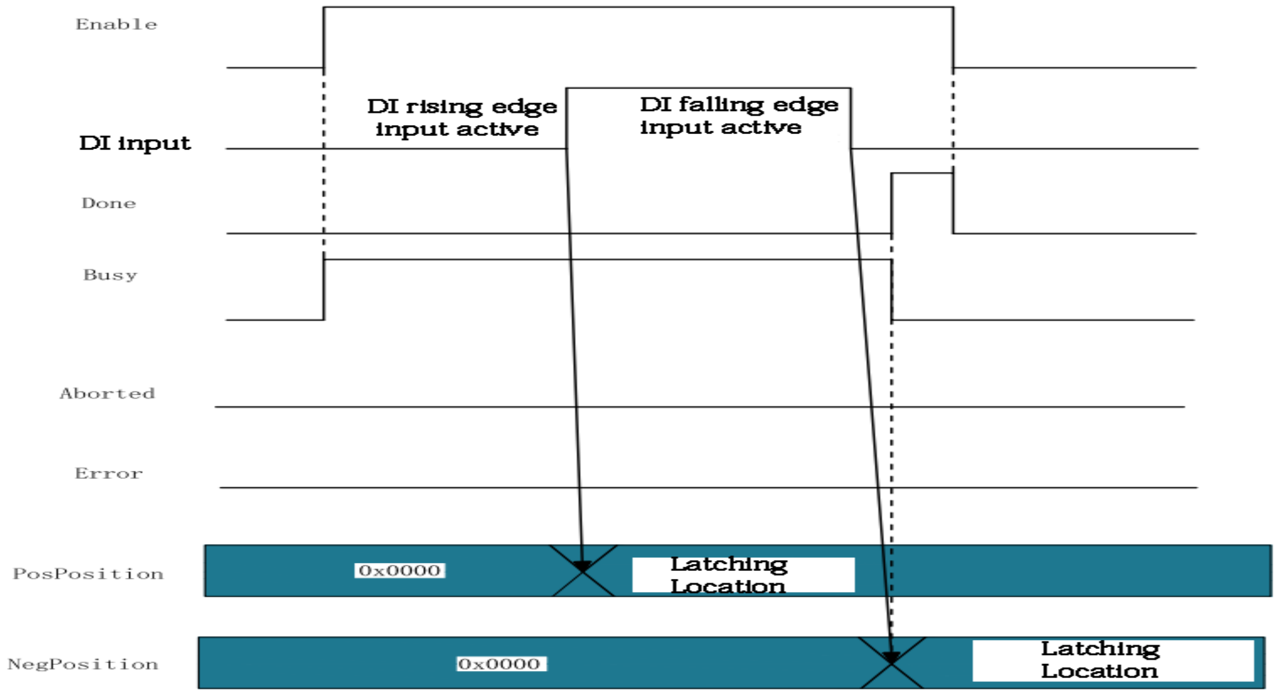
● Probe 1 rising edge active, DI terminal trigger, single trigger, window function effective.



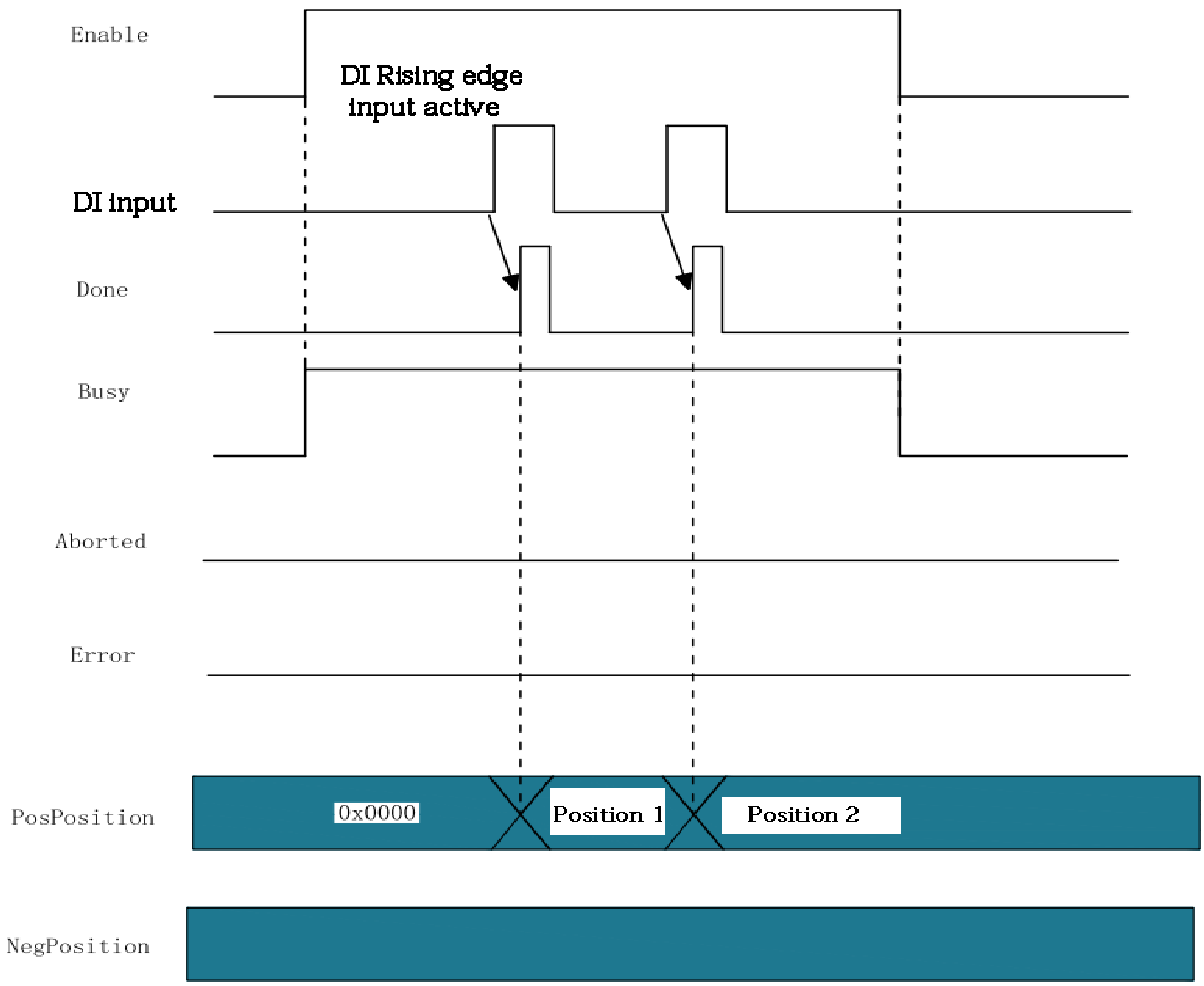
● Probe 1 falling edge active, DI terminal trigger, single trigger, window function invalid.



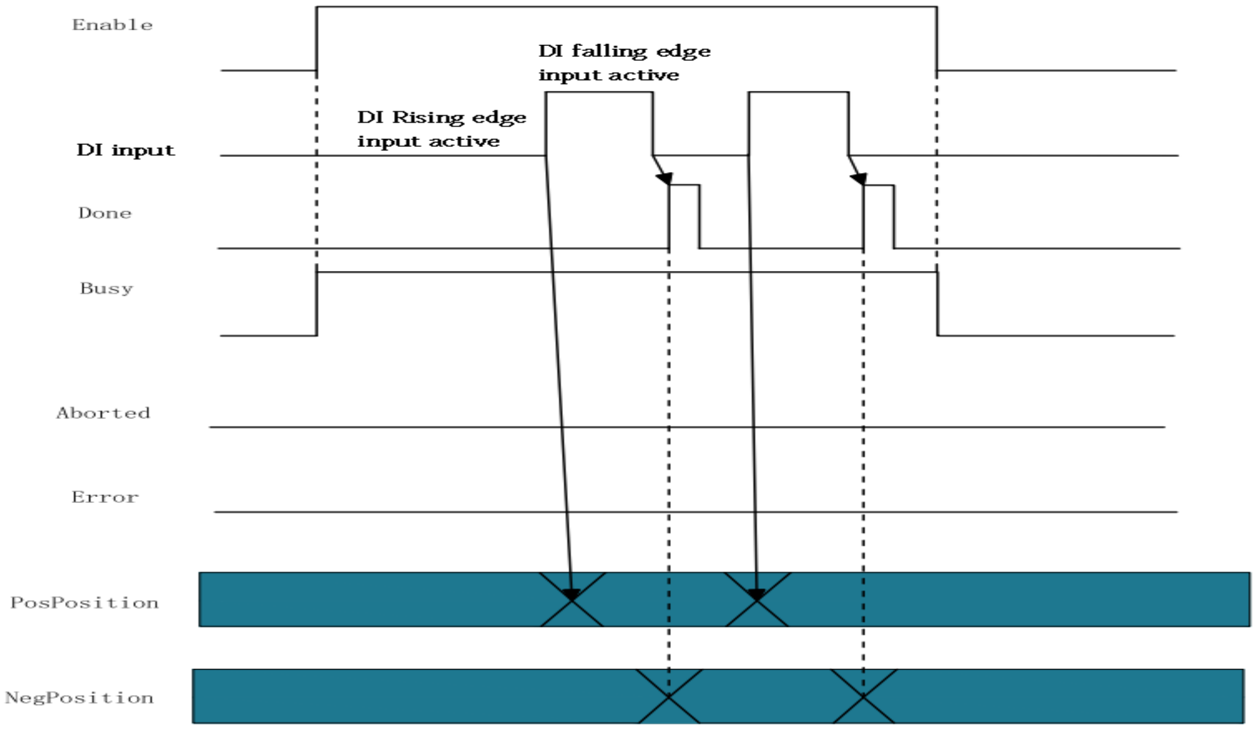
● Probe 1 rising and falling edges active, DI terminal trigger, single trigger, window function invalid.



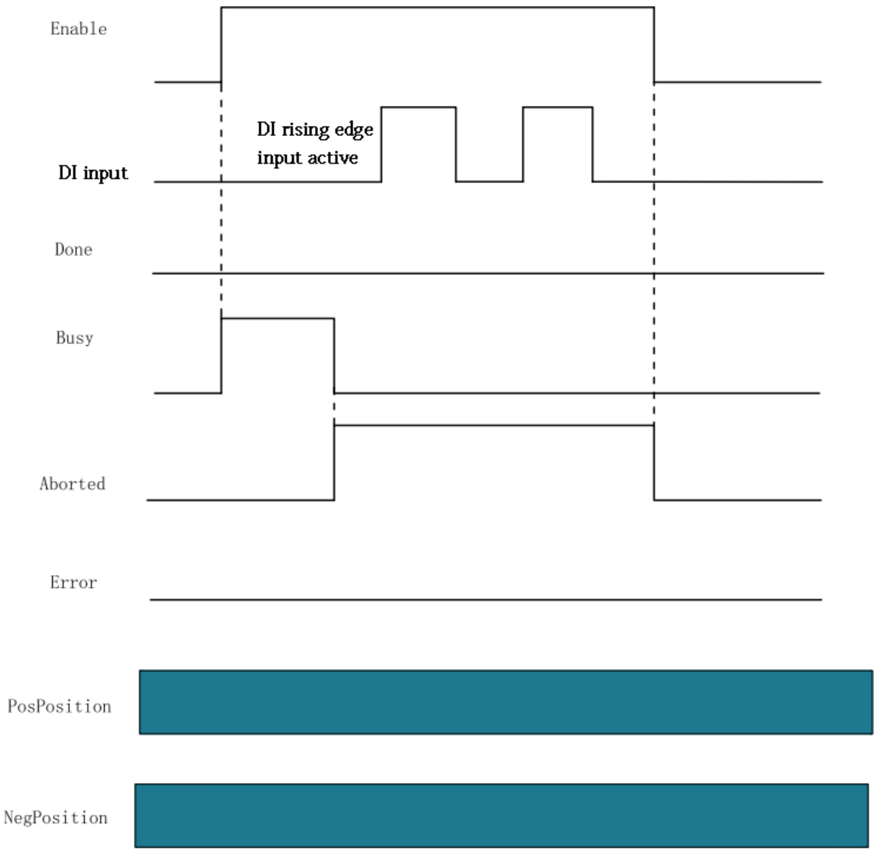
● Probe 1 is active on the rising edge, DI terminal trigger, continuous trigger, window function is invalid.



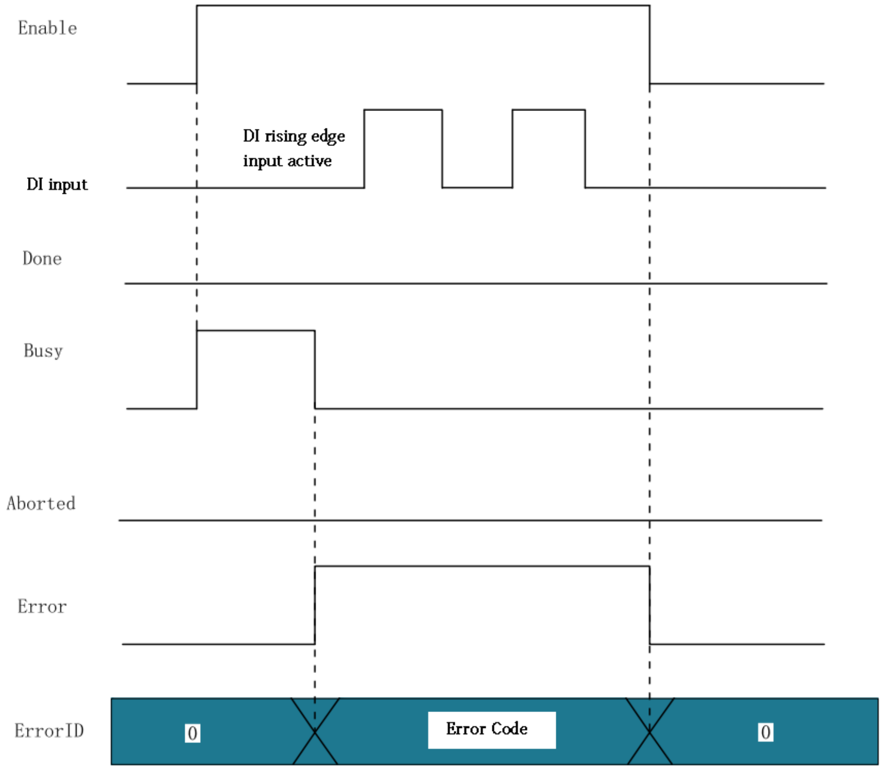
● Probe 1 rising and falling edges active, DI terminal trigger, continuous trigger, when the rising edge and falling edge of the DI terminal are valid after Done produces a valid signal for one cycle, the window function is invalid.



● Probe 1 is interrupted by other probe-related commands, and the window function is invalid.

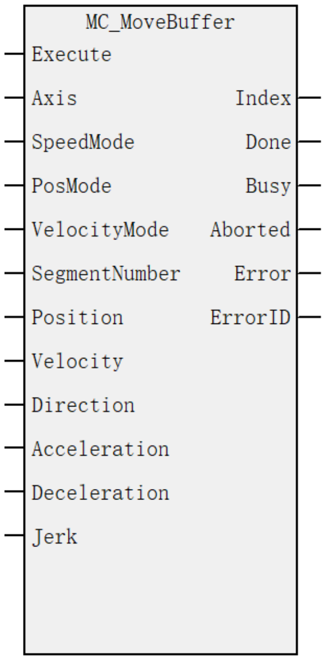


● Probe 1 command failure.



4.3.15 MC_MoveBuffer (Multi-segment position command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable

IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis number	Execute Rising Edge
IN	SpeedMode	BOOL	M/S	TRUE, FALSE	Speed Type: Type 0T, Type 1S	Execute Rising Edge
IN	PosMode	BOOL	M/S	TRUE, FALSE	Location mode: 0 relative, 1 absolute	Execute Rising Edge
IN	Velocity Mode	BOOL	M/S	TRUE, FALSE	Speed switching mode: 0 Switch after deceleration, 1 Maintain the current speed switch	Execute Rising Edge
IN	Segment Number	WORD	Constant/D/R/W	1~32	The number of segments	Execute Rising Edge
IN	Position	REAL	D/R/W	Positive/Negative Number/0	Axis target position array: (1~32 groups)	Execute Rising Edge
IN	Velocity	REAL	D/R/W	Positive Number	Target speed array: (1~32 groups)	Execute Rising Edge
IN	Direction	WORD	D/R/W	0~3	Ring mode absolute positioning direction Start address (1~32 groups) 0: Positive direction (target velocity greater than 0) 1: Negative direction (target speed less than 0) 2: Shortest distance 3: Current direction	Execute Rising Edge
IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	Execute Rising Edge
IN	Deceleration	REAL	Constant/D/R/W	Positive Number	Decelerate	Execute Rising Edge
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed step	Execute Rising Edge
OUT	Index	WORD	D/R/W	0~32	Currently executing segments 1 to 32	
OUT	Done	BOOL	M/S	TRUE, FALSE	Finish stopping	
OUT	Busy	BOOL	M/S	TRUE,	In-Run flag	

				FALSE		
OUT	Aborted	BOOL	M/S	TRUE,FALSE	Execution interrupts	
OUT	Error	BOOL	M/S	TRUE,FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

This command is used to implement multi-segment position positioning functions for bus EtherCAT bus axis or local pulse axis, with active rising edges.

The setting of the axis number	Axis is latched along the rising edge of the Execute input. If Axis is set with the axis number, modifying Axis during Execute=TRUE is invalid. If Axis is set to the axis number, modifying Axis during Execute=FALSE is invalid.
Functional description	This directive can only be invoked if the axis is switched to the enabled state using MC_Power command. The rising edge of the Execute input, the function block latches the Position, Velocity, Direction, SegmentNumber, Acceleration, Deceleration and other input parameters. The axis will be absolutely positioned in buffer mode (PosMode = 0) or relative (PosMode = 1) according to the value set by PosMode. This command implements up to 32 segment positions. ■ Position: target position, array type, maximum 16 levels. Absolute positioning mode is used to set the target absolute position of the axis, and relative positioning mode is used to set the target displacement of the axis. ■ Velocity: target speed, array type, max. 16 levels, for setting the target speed. ■ Direction: The direction of the target in the annular absolute positioning mode. The meaning is MC_MoveAbsolute Direction. ■ SegmentNumber: The number of groups of buffered target positions, target speeds and directions ranges from 1 to 32. If the range is exceeded, an unreasonable parameter fault will be reported. ■ SpeedMode: The type used to set the speed curve. SpeedMode = 0 indicates a T-shaped curve, in which case the speed of the shaft will accelerate or decelerate according to the values set by Acceleration and Deceleration; SpeedMode = 1 represents a 5-segment S-curve, in which case Acceleration and Deceleration represent the maximum acceleration and minimum deceleration achieved by the shaft during acceleration and deceleration. ■ VelocityMode : Speed switching mode, when set to 0, when the shaft reaches the target value of the previous section, the shaft first slows down to 0, and then starts running from 0 speed to the target position of the next section; When set to 1, the axis will run to the target position at the target speed of the previous segment, then switch to the new speed at the acceleration (deceleration) and move towards the new target position.

Program examples

Select 3-segment position, SegmentNumber=3, and implement the multi-segment position function through the following routine.

The speed and position corresponding to the three positions are shown in the figure：

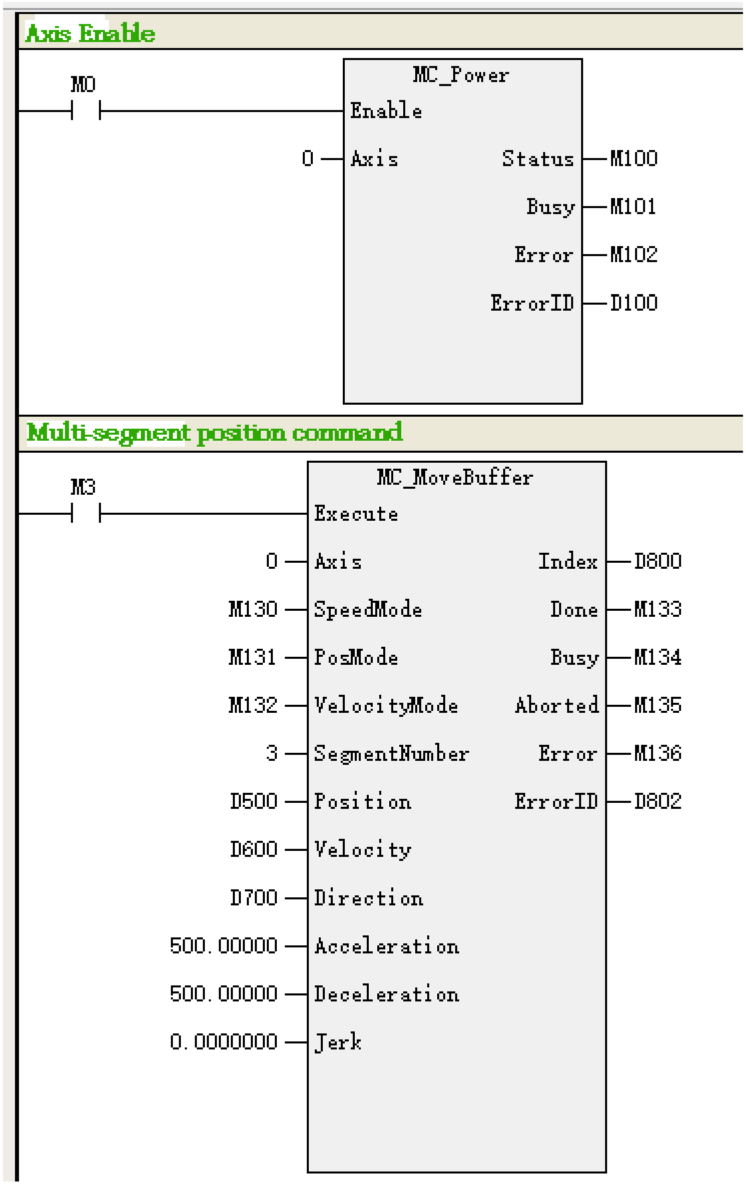
	Element Name	Data Type	Display Format	Current value	New Values	元
1	D600	REAL	Decimal	5. 0000000		
2	D602	REAL		15. 000000	Three-stage speed	
3	D604	REAL		5. 0000000		
4		INT				
5	D500	REAL		100. 00000		
6	D502	REAL		500. 00000	Three-segment position	
7	D504	REAL		800. 00000		

The function block command fills in the registers that set the position and speed of the first segment, and the continuous register address sets the position and speed of the number of consecutive segments.

1. Enable axis Axis_0 after setting M0 to ON.

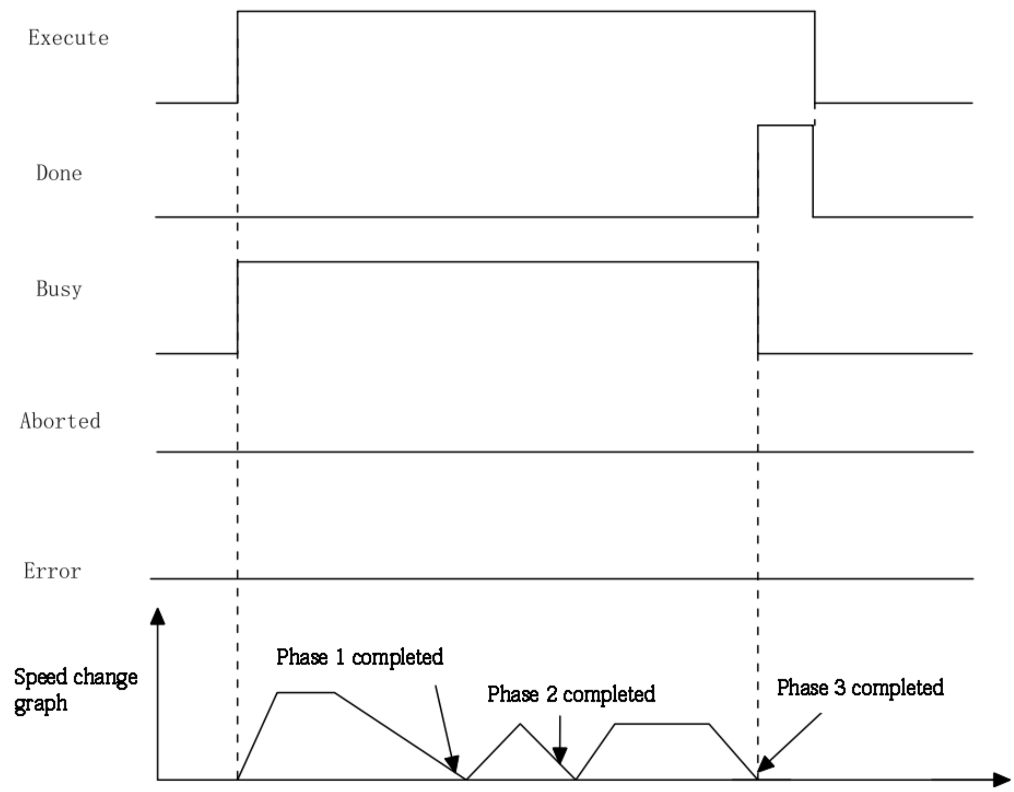
2. After setting M3 to ON, the axis Axis_0 the distance of 100Unit of the first segment running relative to the movement at 5Unit/s, the distance of the second segment running 500Unit at 15Unit/s relative to the movement, and the distance of the third segment running 800Unit at 5Unit/s relative to the movement.

The number of segments in current motion can be monitored via Index.

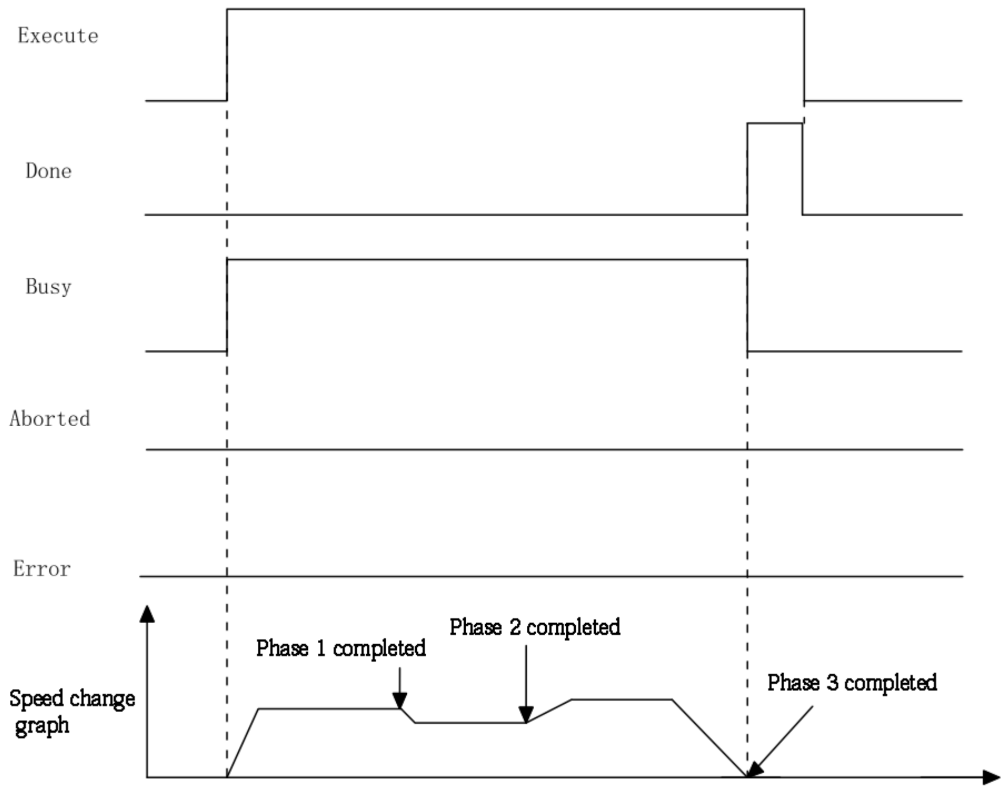


Timing diagram

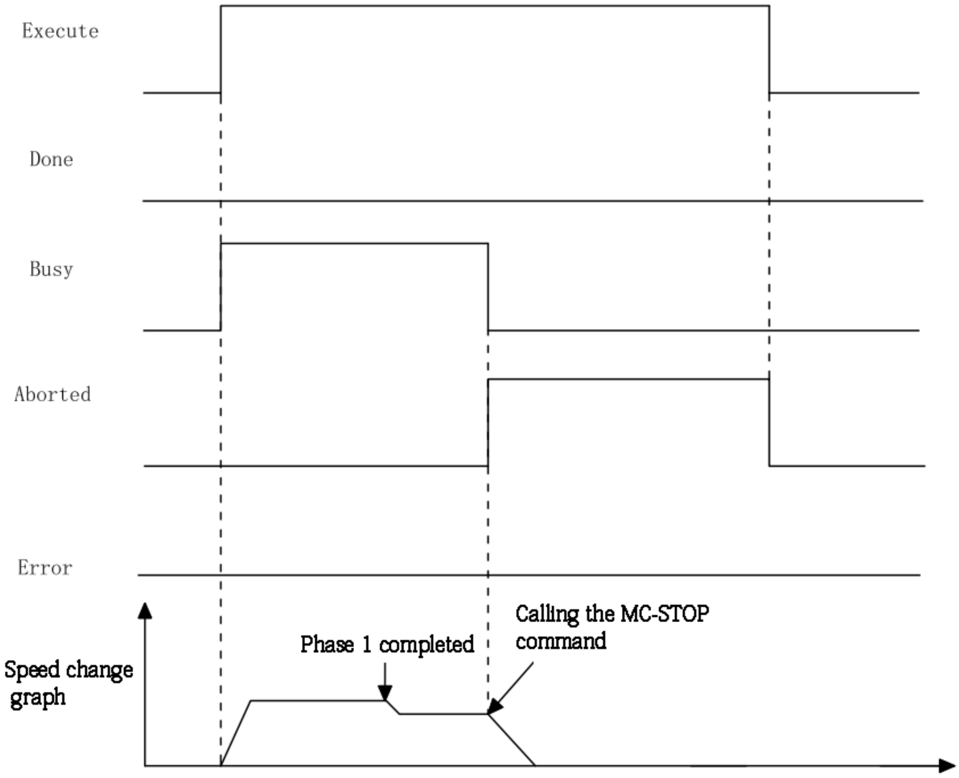
● Set to 3-segment buffering when Velocity Mode = 0.



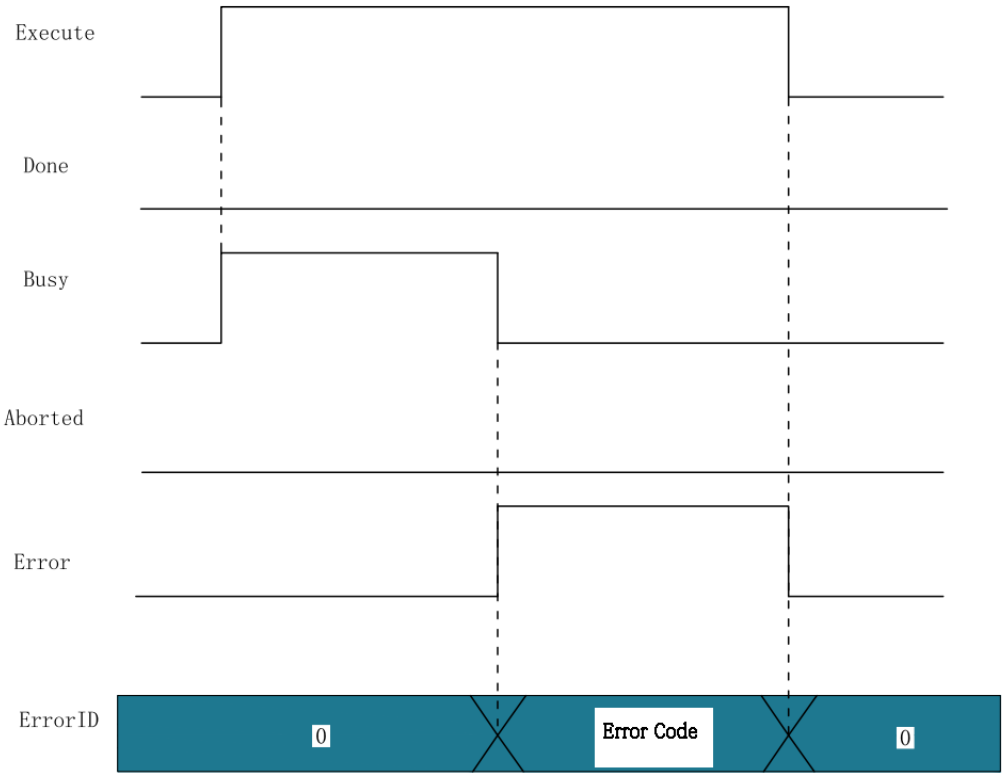
● Set to 3-segment buffering when Velocity Mode = 1.



● Set to 3-segment buffering when interrupted by MC_Stop command during runtime.

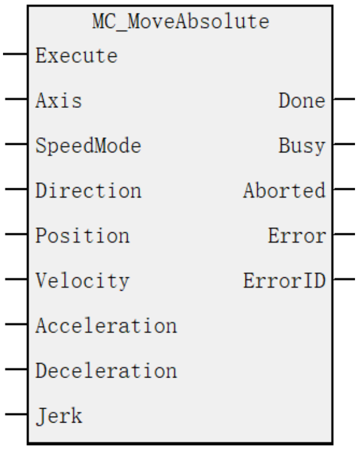


● Set to 3-segment buffering, but when an error is reported during operation.



4.3.16 MC_MoveAbsolute (Absolute positioning command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis Name/Axis Number	Execute Rising Edge
IN	SpeedMode	BOOL	M/S	TRUE,FALSE	Speed Type: Type 0T, Type 1S	Execute Rising Edge
IN	Direction	WORD	Constant/D/R/W	0~3	Positioning direction: (rotation mode only) 0: Forward (velocity greater than 0) 1: Negative (speed less than 0) 2: Shortest distance 3: Current direction	Execute Rising Edge
IN	Position	REAL	Constant/D/R/W	Positive/Negative Number/0	The target location	Execute Rising Edge
IN	Velocity	REAL	Constant/D/R/W	Positive Number	The target speed	Execute Rising Edge
IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	Execute Rising Edge
IN	Deceleration	REAL	Constant/D/R/W	Positive Number	Decelerate	Execute Rising Edge
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed step	Execute Rising Edge
OUT	Done	BOOL	M/S	TRUE,	Finish stopping	

				FALSE		
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Aborted	BOOL	M/S	TRUE ,FALSE	Execution interrupts	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error Flag	
OUT	Error ID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

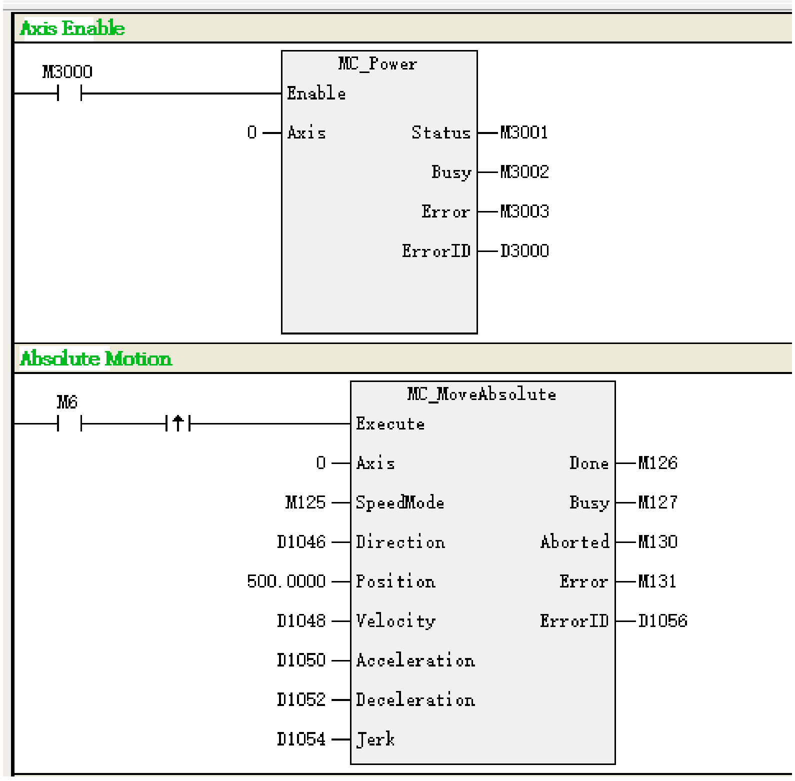
This command is used to control the EtherCAT bus axis or the local pulse axis for absolute positioning with active rising edges.

The setting of the axis number	Axis is latched along the rising edge of the Execute input. If Axis is set with the axis number, modifying Axis during Execute=TRUE is invalid. If Axis is set to the axis number, modifying Axis during Execute=FALSE is invalid.
Functional description	This directive can only be invoked if the axis is switched to the enabled state using MC_Power command. On the rising edge of the Execute input, the command latches the input parameters on the left side of Position, Velocity, etc., triggers the absolute positioning function, and switches the PLCOpen state machine of the axis to the DiscreteMotion state. In linear mode, Position is used to set the target position for pair positioning. If the current position is less than the target position, the axis will move forward and finally reach the position set by Position, and if the current position is greater than the target position, the axis will move in reverse, and finally reach the position set by Position. SpeedMode is used to set the type of speed curve. SpeedMode = 0 indicates a T-shaped curve in which the speed of the axis will accelerate or decelerate according to the values set by Acceleration and Deceleration. SpeedMode = 1 represents the S-segment curve, in which case Acceleration and Deprovide represent the maximum acceleration and minimum deceleration achieved by the shaft during acceleration and deceleration.

Program examples

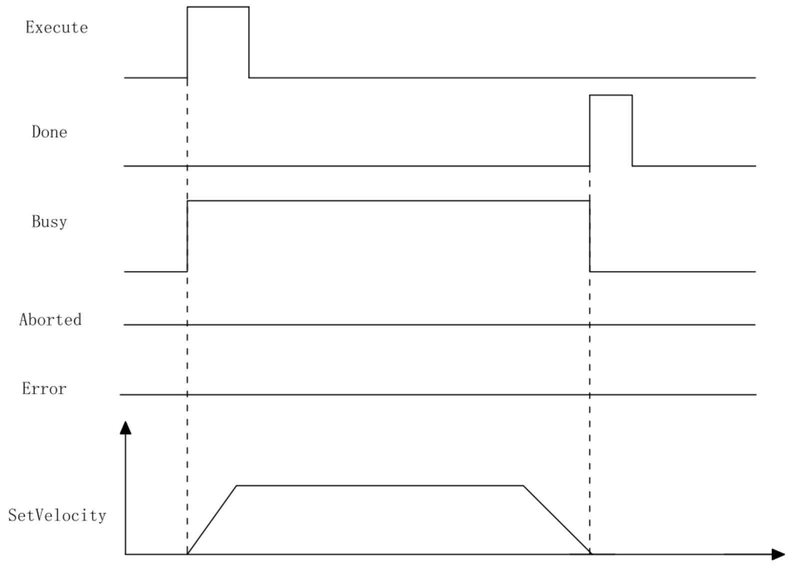
Absolute motion function is achieved by the following routines.

1. Set M3000 to ON and enable the axis Axis_0.
2. After setting M6 to ON, the shaft Axis_0 run at the set speed to a position 500 from the origin.

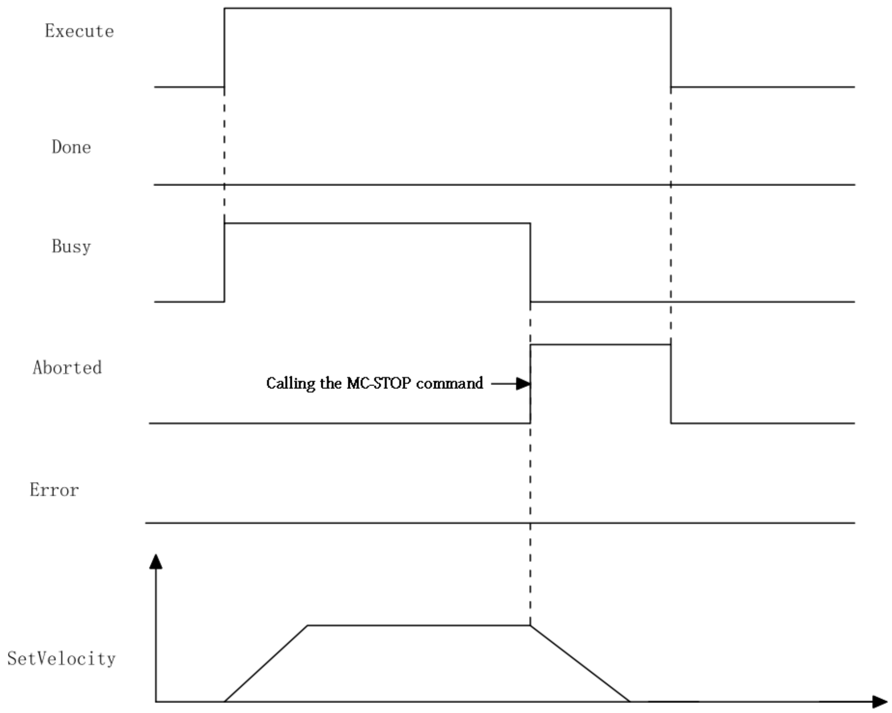


Timing diagram

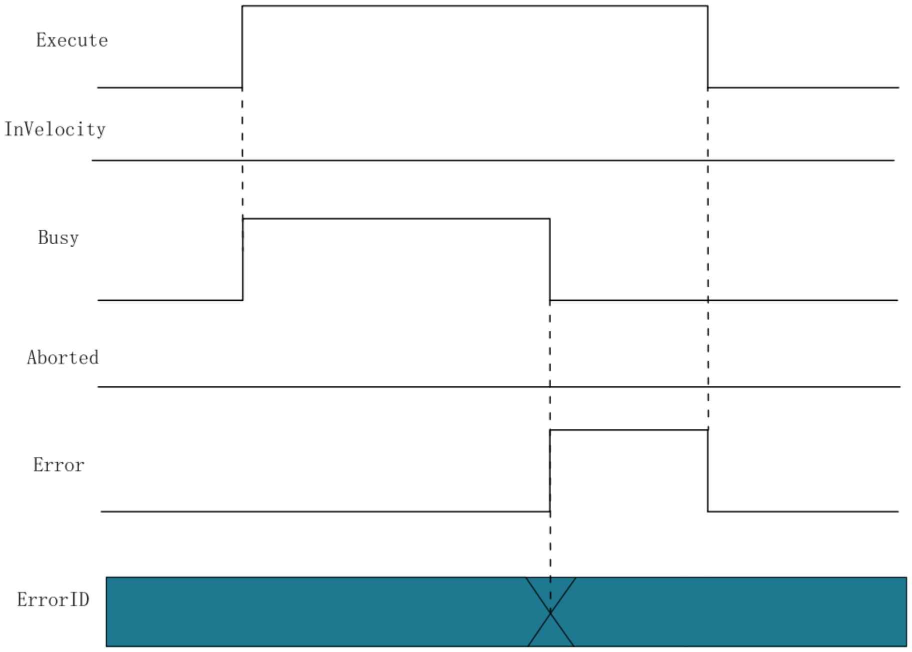
● When the axis is in the StandStill state, call this command to do the absolute positioning motion under the T-curve.



● When the axis is interrupted by Mc_Stop command during absolute positioning motion.

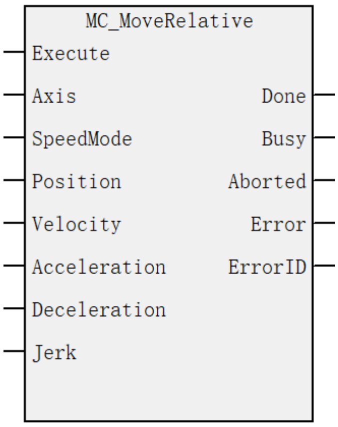


● When the shaft fails during movement, the drive fails.



4.3.17 MC_MoveRelative (Relative positioning command)

Drawing Block:



Command list format

Input/output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis number	Execute Rising Edge
IN	SpeedMode	BOOL	M/S	TRUE,FALSE	Speed Type: Type 0T, Type 1S	Execute Rising Edge
IN	Position	REAL	Constant/D/R/W	Positive/Negative Number/0	The target location	Execute Rising Edge
IN	Velocity	REAL	Constant/D/R/W	Positive Number	The target speed	Execute Valid
IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	Execute Rising Edge
IN	Deceleration	REAL	Constant/D/R/W	Positive Number	Decelerate	Execute Rising Edge
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed step	Execute Rising Edge
OUT	Done	BOOL	M/S	TRUE,FALSE	Finish stopping	
OUT	Busy	BOOL	M/S	TRUE,FALSE	In-Run flag	
OUT	Aborted	BOOL	M/S	TRUE,FALSE	Execution interrupts	
OUT	Error	BOOL	M/S	TRUE,FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

This command is used to control the EtherCAT bus axis or the local pulse axis for relative positioning with the rising edge active.

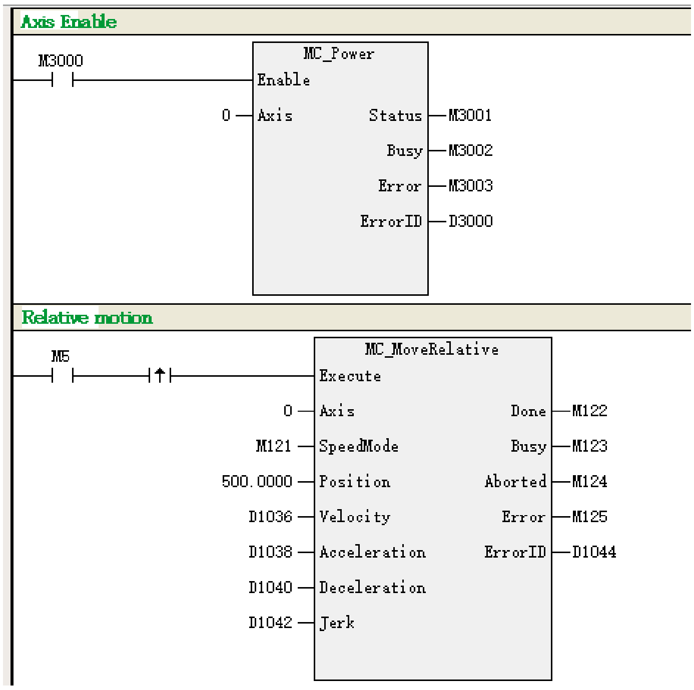
The setting of the axis number	Axis is latched along the rising edge of the Execute input. If Axis is set with the axis number, modifying Axis during Execute=TRUE is invalid. If Axis is set to the axis number, modifying Axis during Execute=FALSE is invalid.

Functional description	This directive can only be invoked if the axis is switched to the enabled state using MC_Power command. On the rising edge of the Execute input, the command latches the input parameters on the left side of Distance, Velocity, etc., triggers the relative positioning function, and switches the PLCOpen state machine of the axis to the DiscreteMotion state. Position is used to set the distance of relative positioning. Whether in linear mode or ring mode, if Position is positive, the axis runs positively at the distance specified by Position, and if Position is negative, the axis runs negatively at the distance specified by Position. SpeedMode is used to set the type of speed curve. SpeedMode = 0 indicates a T-shaped curve in which the speed of the axis will accelerate or decelerate according to the values set by Acceleration and Deceleration. SpeedMode = 1 represents a 5-segment S curve, at which point Acceleration and Deceleration represent the maximum acceleration and minimum deceleration achieved by the shaft during acceleration and deceleration degree.

Program examples

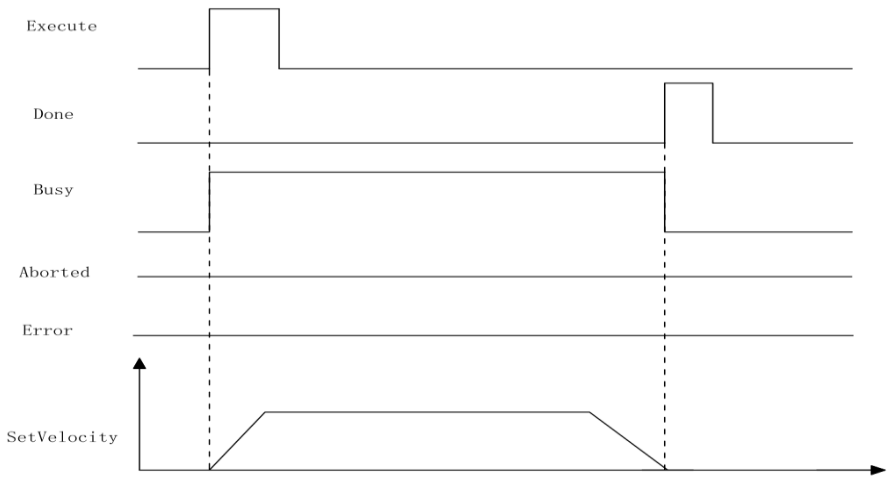
The relative motion function is achieved by the following routines.

1. Set M3000 to ON and enable the axis Axis_0.
2. After setting M5 to ON, the axis Axis_0 run 500 at the set speed on the basis of the current command position.

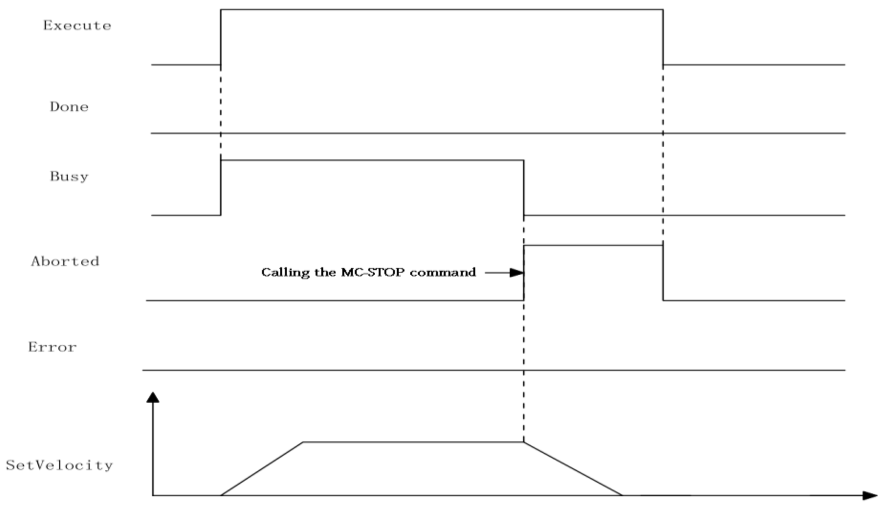


Timing diagram

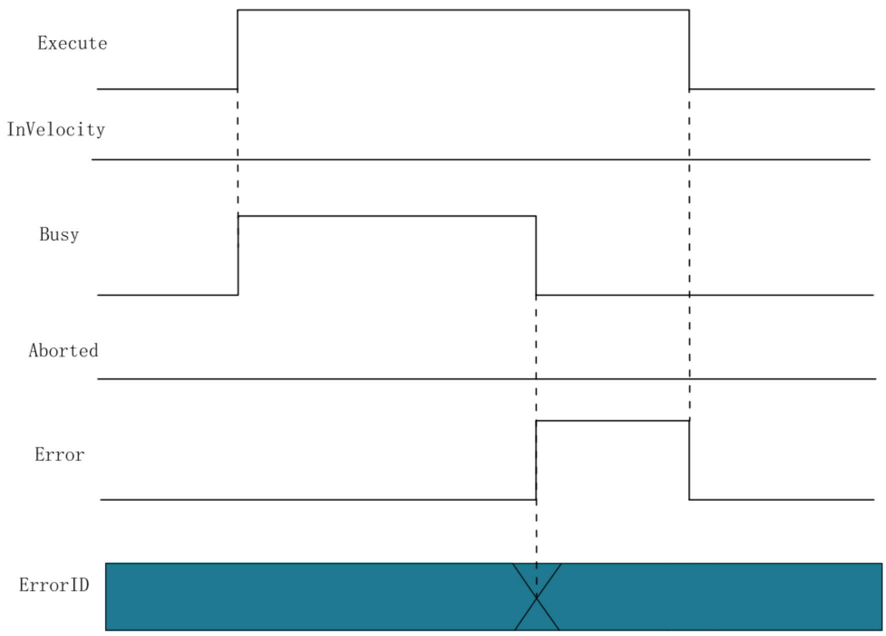
● When the axis is in the StandStill state, call this command to do the relative positioning motion under the T-curve.



● When the axis is interrupted by Mc_Stop command during relative positioning motion.

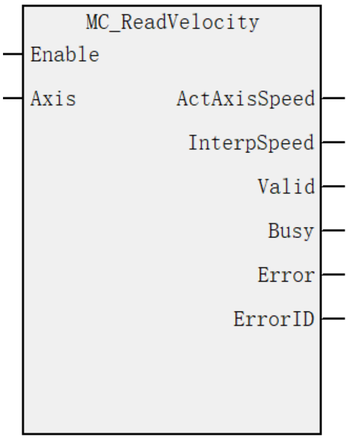


● When the shaft fails during movement, the drive fails.



4.3.18 MC_ReadVelocity (Read speed command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis number	Enable Valid
OUT	ActAxisSpeed	REAL	D/R/W		Shaft speed	
OUT	InterpSpeed	REAL	D/R/W		Interpolation speed: (Uninterpolated time is consistent with shaft speed)	
OUT	Valid	BOOL	M/S	TRUE, FALSE	Valid flags	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

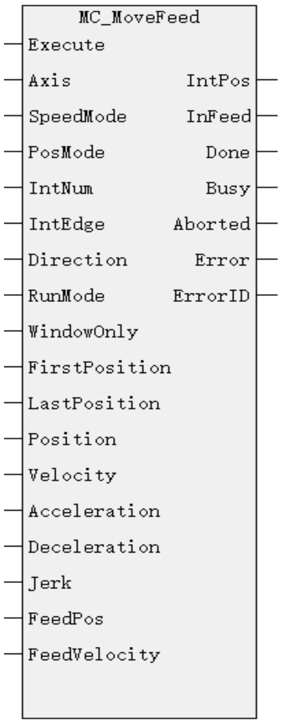
Description of functions and commands

Timing diagram

Omit

4.3.19 MC_MoveFeed (Interrupt fixed-length command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis number	Execute Rising Edge
IN	SpeedMode	BOOL	M/S	TRUE, FALSE	Speed Type: Type 0T, Type 1S	Execute Rising Edge
IN	PosMode	BOOL	M/S	TRUE, FALSE	Location mode: 0 relative, 1 absolute Note: RunMode does not work when it is 1	Execute Rising Edge
IN	IntNum	BOOL	M/S	TRUE, FALSE	Probe Number: Probe 0 or 1	Execute Rising Edge
IN	IntEdge	BOOL	M/S	TRUE, FALSE	Trigger edge: 0 rises, 1 falls	Execute Rising Edge
IN	Direction	WORD	Constant/D/R/W	0~3	Direction of motion in absolute positioning of ring mode 0: positive direction (target	Execute Rising Edge

					velocity is positive) 1: negativeDirection (target velocity is negative) 2: Shortest distance 3: Current direction	
IN	RunMode	BOOL	M/S	TRUE, FALSE	Initial mode 0: Positioning mode 1: Speed mode	Execute Rising Edge
IN	WindowOnly	BOOL	M/S	TRUE, FALSE	Enable interrupt source window 0: Do not enable window detection function 1: Enable window detection function	Execute Rising Edge
IN	FirstPosition	REAL	Constant/D/R/W	Positive/Negative Number/0	Break Source Window Start position	Execute Rising Edge
IN	LastPosition	REAL	Constant/D/R/W	Positive/Negative Number/0	End of the interrupt source window location	Execute Rising Edge
IN	Position	REAL	Constant/D/R/W	Positive/Negative Number/0	Location or distance	Execute Rising Edge
IN	Velocity	REAL	Constant/D/R/W	Positive Number	The target speed	Execute Rising Edge
IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	Execute Rising Edge
IN	Deceleration	REAL	Constant/D/R/W	Positive Number	Decelerate	Execute Rising Edge
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed step	Execute Rising Edge
IN	FeedPos	REAL	Constant/D/R/W	Positive/Negative Number/0	The distance after the interruption	Execute Rising Edge
IN	FeedVelocity	REAL	Constant/D/R/W	Positive Number	Speed after interruption	Execute Rising

						Edge
OUT	IntPos	REAL	D/R/W	Positive/Negative Number/0	The location of the interrupt	
OUT	InFeed	BOOL	M/S	TRUE, FALSE	The interrupt is valid	
OUT	Done	BOOL	M/S	TRUE, FALSE	Finish stopping	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interrupts	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

This command is used to implement the interrupt length function of the bus EtherCAT bus axis or the local pulse axis, the command is valid on the rising edge, and this command does not support virtual axis mode.

The setting of the axis number	Axis is latched along the rising edge of the Execute input. If Axis is set with the axis number, modifying Axis during Execute=TRUE is invalid. If Axis is set to the axis number, modifying Axis during Execute=FALSE is invalid.
Functional description	This directive can only be invoked if the axis is switched to the enabled state using MC_Power command. On the rising edge of the command, the function block latches input parameters such as Position, Velocity, Direction, Acceleration, Deceleration, etc. Before the interrupt arrives, the axis will be absolutely positioned (PosMode = 0) according to the parameters set by Position, Velocity, Direction, Mode, etc. (PosMode = 0), relative positioning (PosMode = 1), after the interrupt source set by IntNum generates an interrupt signal, the axis will make relative motion according to the parameters set by FeedDis and FeedVelo at the position of the interrupt when it arrives. ■ Position: The target location. Used to set the axis when Mode = 0 or Mode = 1 (relative positioning) the target location before the interrupt arrives. ■ Velocity: Target speed, which sets the target speed of the axis before the interrupt arrives. ■ SpeedMode: The type used to set the speed curve. SpeedMode = 0 indicates a T-shaped curve, at which point the speed of the axis will accelerate or decelerate according to the values set by Acceleration and Deceleration; SpeedMode = 1 represents a 5-segment S-curve, in which case Acceleration and Deceleration represent the maximum acceleration and minimum deceleration achieved by the shaft during acceleration and deceleration. ■ Direction: The direction of movement, the meaning of the same MC_MoveAbsolute as Direction. When Mode = 0 (absolute positioning) this parameter is used to set the direction of movement of the axis in toroidal mode. A setting of 0 indicates forward running, a setting of 1 indicates reverse motion, a setting of 2 indicates the shortest path, and a setting of 3 indicates the current direction. ■ PosMode: Interrupts the pre-arrival sport mode. When PosMode=0, the axis will be absolutely positioned before the interrupt arrives; When Mode=1 is 1, the axes will be positioned relative before the interrupt arrives.

	■IntNum: Interrupt source selection, when IntNum = 0, the interrupt source is probe 1, and is valid on the rising edge of probe 1; When IntNum=1, the interrupt source is Probe 2 and is valid on the rising edge of Probe 2. ■ FeedPos: Interrupt the target displacement after arrival. A positive time indicates the distance set by FeedPos to run in the current direction of motion when the interrupt signal arrives, and a negative time indicates the distance set by FeedPos to run in the opposite direction when the interrupt signal arrives. FeedVelocity: Interrupt the target speed after arrival. ■ InFeed: The InFeed output is valid after the interrupt signal arrives.

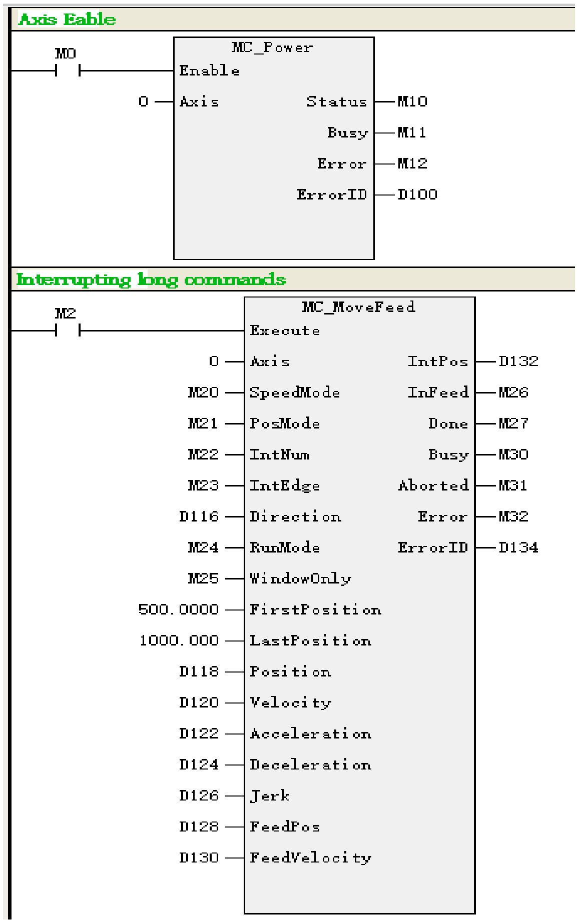
Program examples

The interrupt length function is implemented by using probe 1, terminal X0 rising edge interrupt.

1. Enable axis Axis_0 after setting M0 to ON.

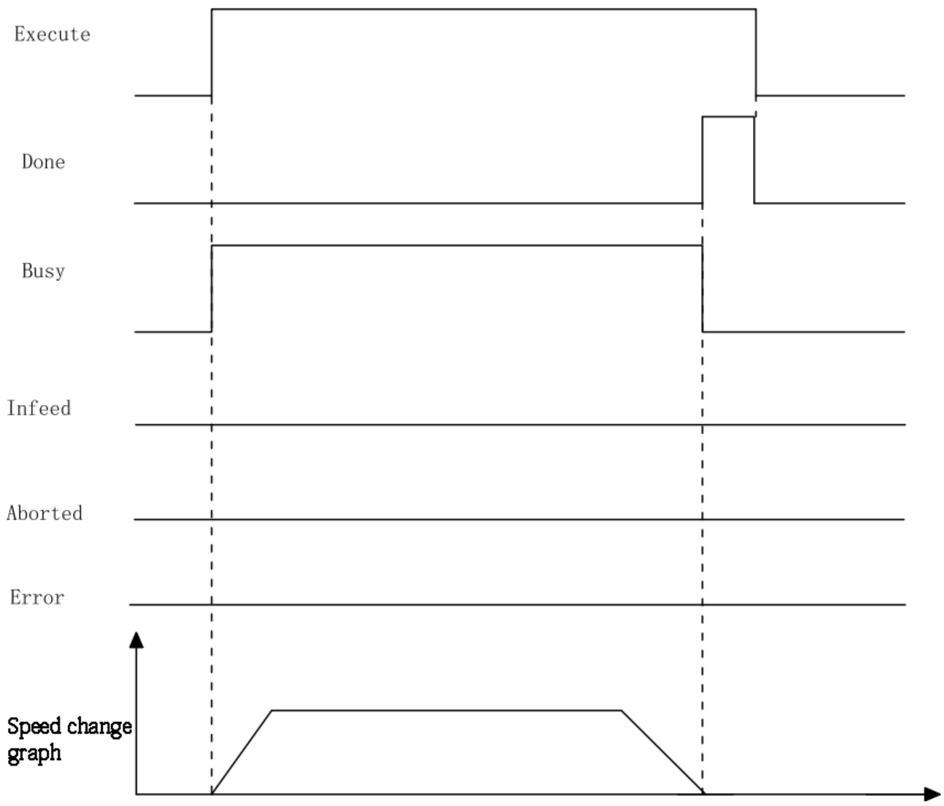
2. After setting M2 to ON, do not enable the interrupt window, position the sport mode, and the axis Axis_0 will make relative motion according to the parameters set by FeedDis and FeedVelo at the position when the interrupt arrives.

3. The rising edge of X0 is detected, the shaft Axis_0 will run the distance after the interruption at the speed after the interrupt, and the IntPos latches the interrupt position.

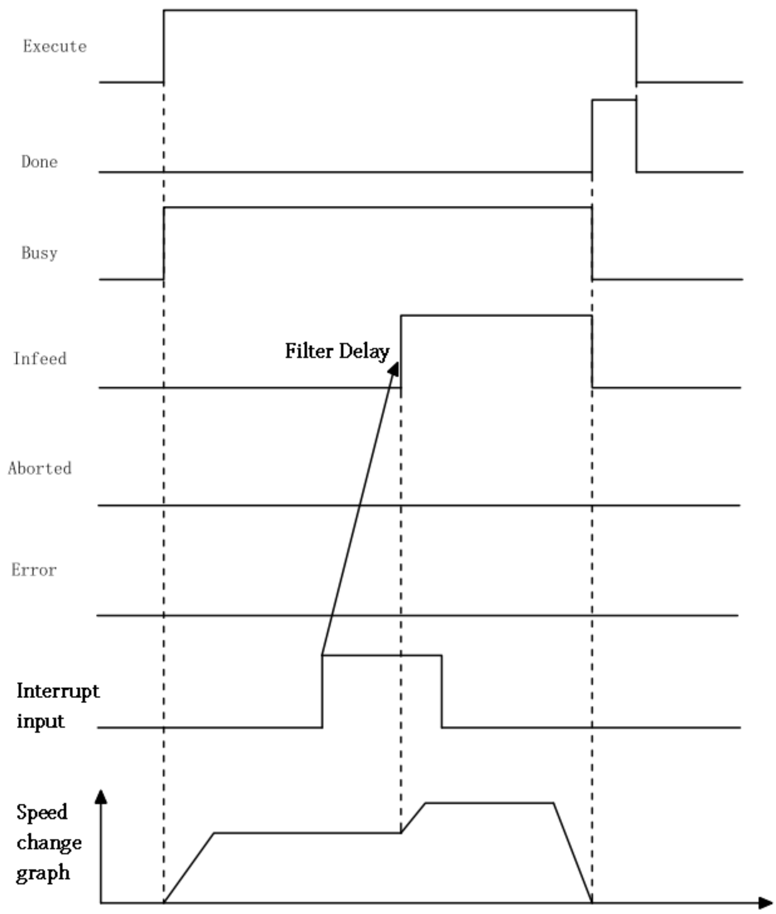


Timing diagram

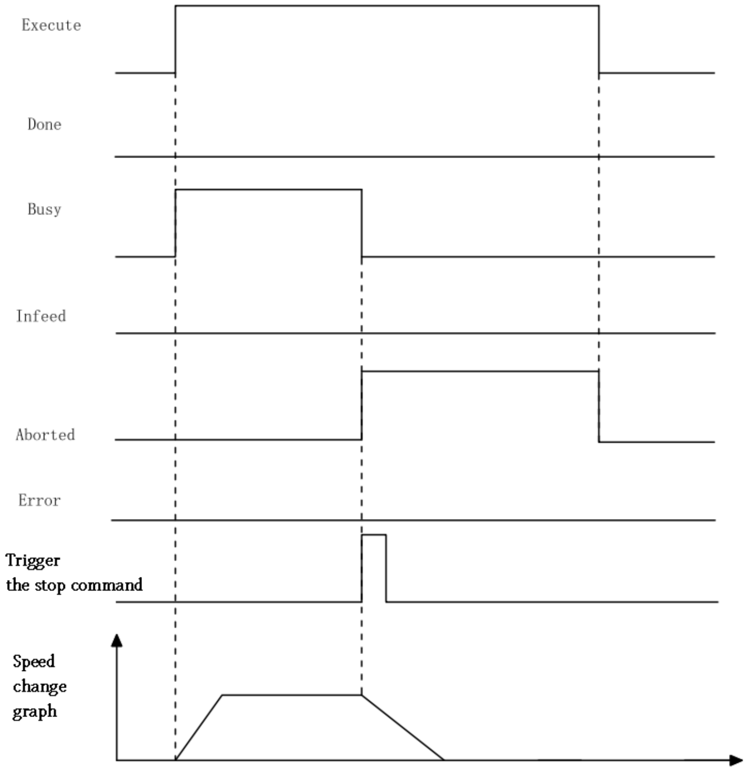
● When the relative positioning, absolute positioning mode is selected, no interrupt signal is triggered and ErrorMode = FALSE.



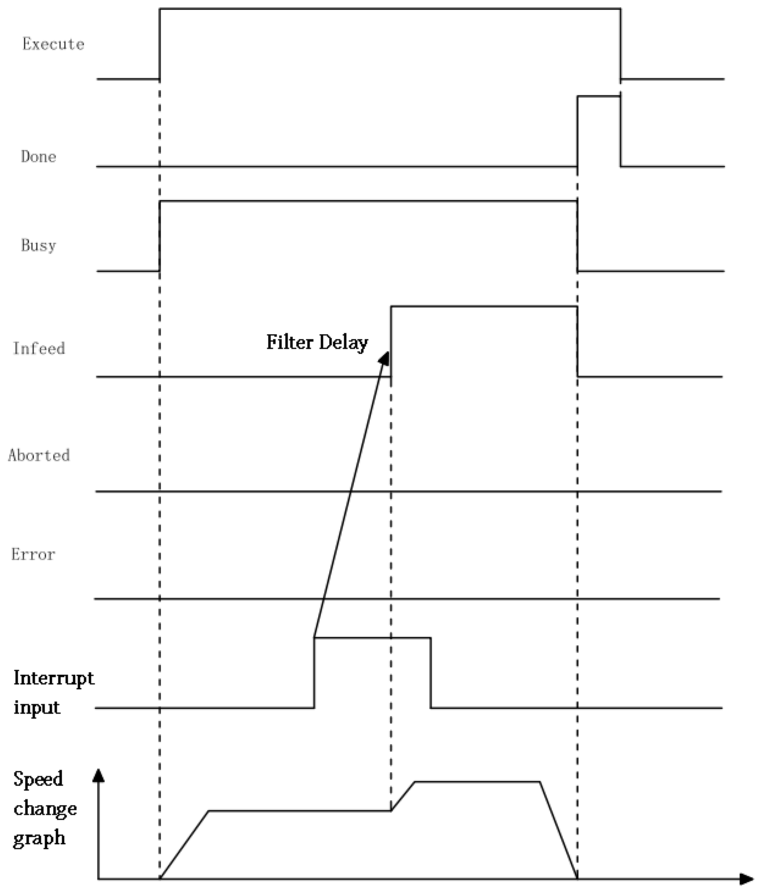
● When the relative positioning, absolute positioning mode is selected, there is an interrupt signal.



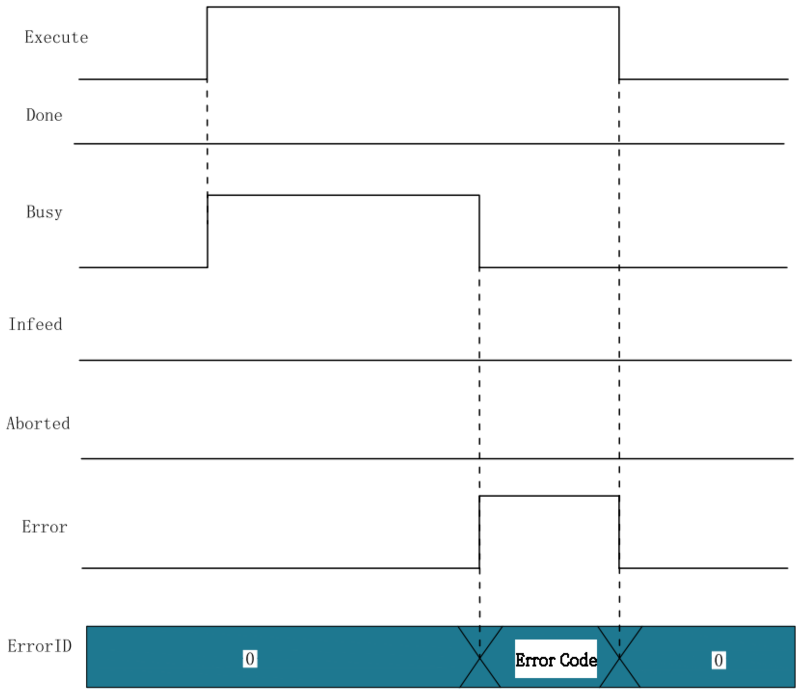
● Select the speed mode, no interrupt is triggered, and after running for a period of time, it is interrupted by MC_Stop command.



● Select the speed mode to trigger the interrupt.

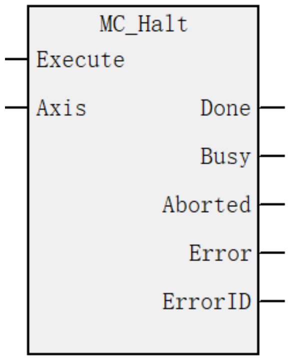


● The command failed while running.



4.3.20 MC_Halt (Pause commmand)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis number	Execute Rising Edge
OUT	Done	BOOL	M/S	TRUE,FALSE	Finish stopping	
OUT	Busy	BOOL	M/S	TRUE,FALSE	In-Run flag	
OUT	Aborted	BOOL	M/S	TRUE,FALSE	Execution interrupts	

OUT	Error	BOOL	M/S	TRUE,FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

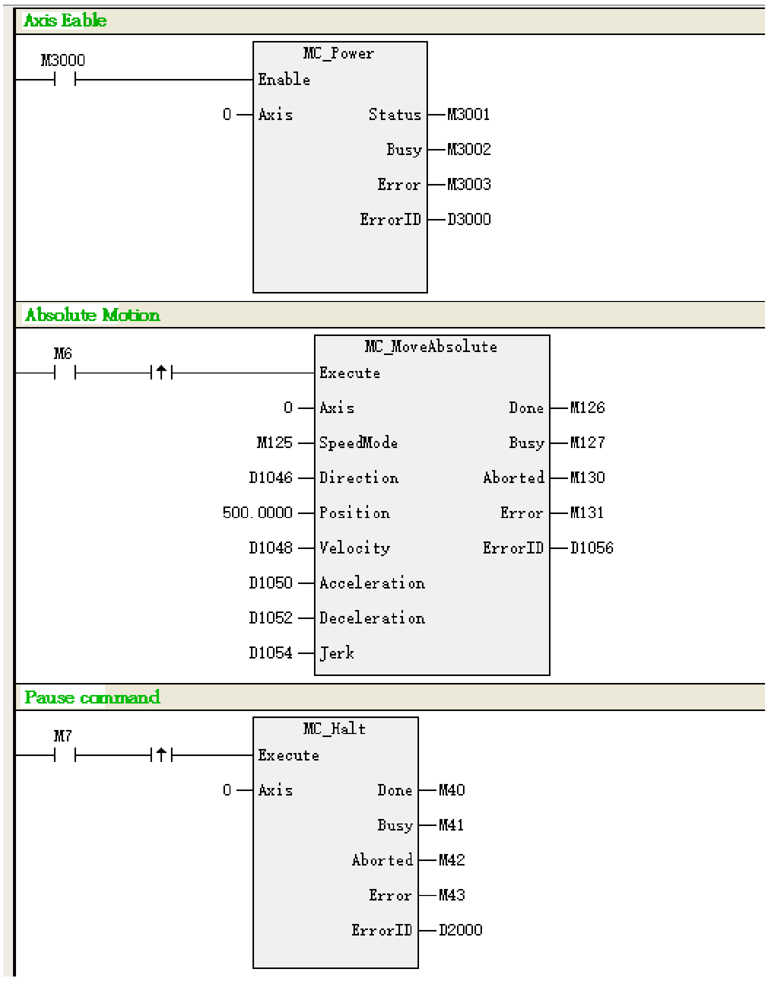
This command is used to implement the pause function of the bus EtherCAT bus axis or the local pulse axis, and the command is valid on the rising edge.

Functional description	This directive can only be invoked if the axis is switched to the enabled state using MC_Power command. The same function as the STOP instruction when pausing for a non-positioning instruction. When pausing for a positioning instruction, a high level enters the temporary stop function. A low level resumes the current axis motion to continue the previous uncompleted motion of positioning.

Program examples

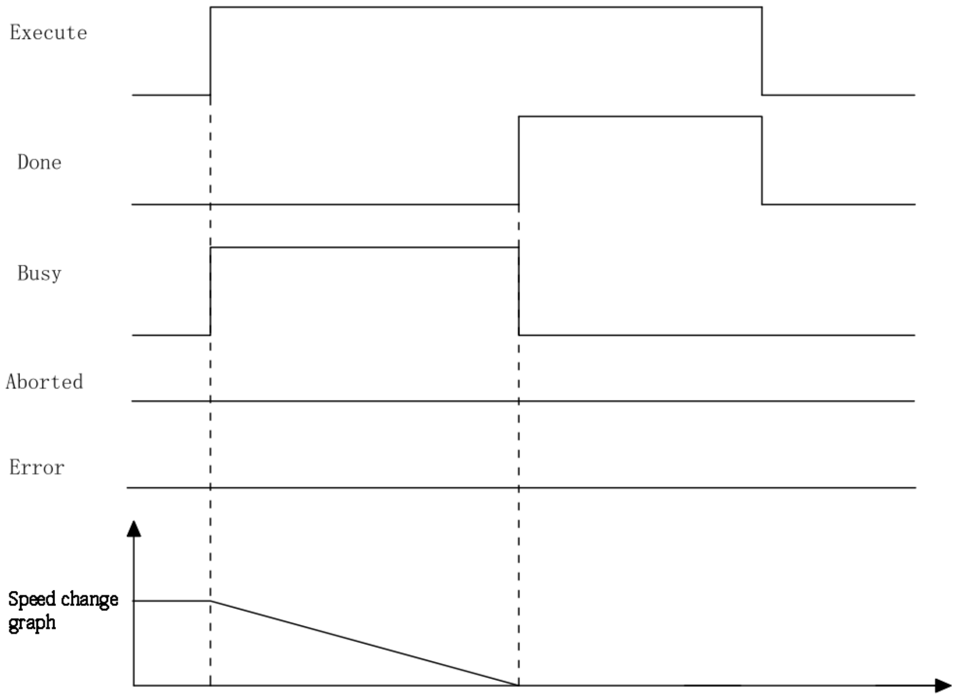
The pause motion function is implemented by using the following routine.

1. Set M3000 to ON and enable the axis Axis_0.
2. Set M6 to ON Rear Axle Axis_0 to set the speed to walk the positioning movement.
3. Set M7 to ON Rear Axle Axis_0 pause to stop positioning motion.
4. Set M7 to OFF Rear Axle Axis_0 continue to walk and position the movement at the set speed.

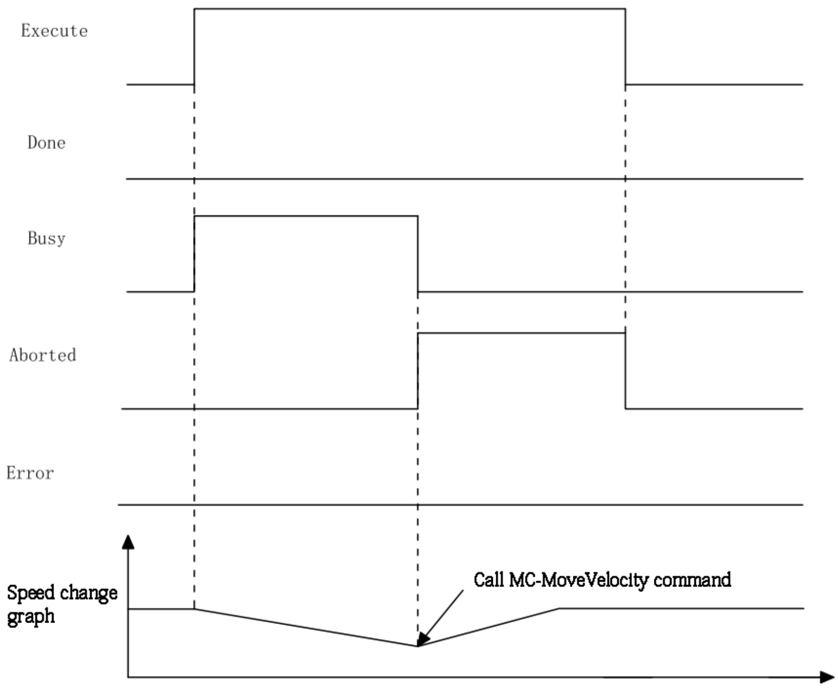


Timing diagram

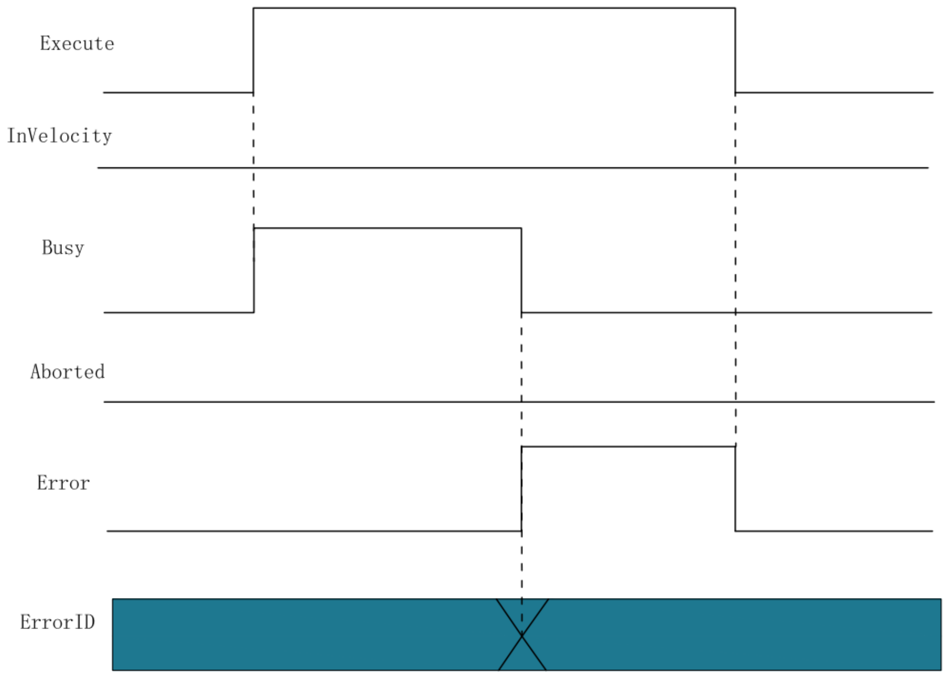
● After the positioning is called, the MC_Halt command is triggered.



● After triggering the MC_Halt command, the recall of the speed command interrupts the execution of the MC_Halt command.

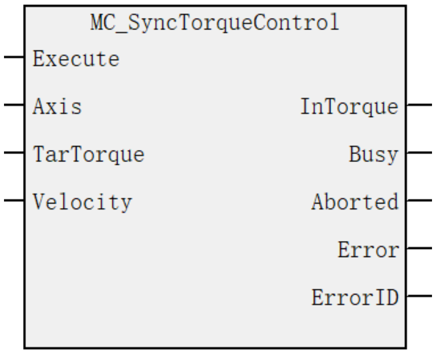


● During the execution of the MC_Halt command, a failure shutdown occurred.



4.3.21 MC_SyncTorqueControl (Synchronous torque control command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis number	Execute Valid
IN	TarTorque	REAL	Constant/D/R/W	Positive/Negative Number/0	Target torque	Execute Valid
IN	Velocity	REAL	Constant/D/R/W	Positive Number	Maximum speed limit	Execute Valid
OUT	InTorque	BOOL	M/S	TRUE, FALSE	Torque arrival sign: The output is valid when less than 5%.	

OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interrupts	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

This command is used to implement the synchronous torque control function and is only used for bus servo axes. The rising edge of the command is valid and the dummy axis mode is not supported.

This command can only be called if the axis is switched to the enable state using the MC_Power command.

The torque command requires drive maps 0x6040,0x6041,0x6060,0x6061,0x6071,0x6077,0x607f for the PDO to work, otherwise it reports a fault.

This command uses the drive synchronous torque mode to implement the torque control function.

When Enable=ON, the function block converts TarTorque, and Velocity from user units to pulse units and sends them to the servo drive on and off, and the axis is in Continuous Motion and does synchronous torque motion.

TarTorque: target torque, the unit is 1%, one decimal after the decimal point in the program is valid, the latter is directly discarded, the actual torque of the drive is subject to configuration

The maximum positive and negative torque limits set in the parameters

Speed control in torque mode For Veichi Servo Drive, if 0x607f is mapped in the axis parameter, this command limits the maximum speed of the servo motor by 0x607f, and if 0x607f is not mapped, the speed limit is invalid.

On the rising edge of Execute, the command converts the Velocity set limit speed into pulse units and writes them to 0x607f via PDO.

When the torque command is interrupted by another command, the maximum speed of the axis is limited by the "Maximum Speed" item in the configuration screen.

For third-party drives, Velocity can only be used as a speed limit if the following conditions are met:

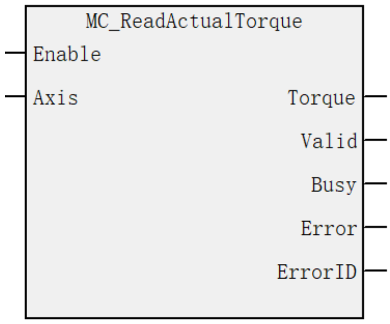
1. The maximum speed of the servo motor can be limited by 0x607F.
2. 0x607F can be configured into PDO
3. 0x607F is in pulse units, not RPM units.

Stop control in torque mode

The MC_Stop command is called in torque mode to perform a stop operation. After receiving the Stop command, the drive switches to synchronous position mode and performs deceleration according to the deceleration rate set by the Stop command.

4.3.22 MC_ ReadActualTorque (Read the actual torque command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis number	Execute Valid
OUT	Torque	REAL	D/R/W	Positive /Negative Number/0	Current torque	
OUT	Valid	BOOL	M/S	TRUE, FALSE	Valid flags	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error Flag	
OUT	Error ID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and directives

This command is based on the feedback torque read out of the EtherCAT bus axis, which is active at a high level. This directive does not support dashed axis mode.

When Enable=ON, if PDO is configured in the EtherCAT bus axis 0x6077 the Valid signal is valid and Torque displays the feedback torque of the axis.

Error

When the axis number does not exist, the command reports an error.

When the axis type is wrong, the command reports an error.

When the axis initialization fails, an error is reported.

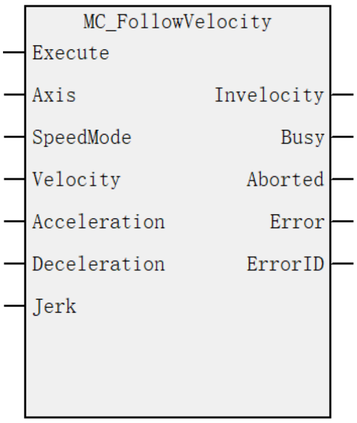
When the EtherCAT bus axis PDO is not configured with 0x6077 command error.

Timing diagrams

Omit

4.3.23 MC_FollowVelocity (Motion speed overlay command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis number	Enable Valid
IN	SpeedMode	BOOL	M/S	TRUE, FALSE	Speed type 0T type, 1S type	Enable Valid
IN	Velocity	REAL	Constant/D/R/W	Positive Number	The target speed	Enable Valid
IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	Enable Valid
IN	Deceleration	REAL	Constant/D/R/W	Positive Number	Decelerate	Enable Valid
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed step	Enable Valid
OUT	Invelocity	BOOL	M/S	TRUE, FALSE	Reach the target speed flag	
OUT	Busy	BOOL	M/S	TRUE FALSE	In-Run flag	
OUT	Aborted	BOOL	M/S	TRUE FALSE	Execution interrupts	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

This command is level valid and overlay speed on the original continuous motion.

This command can be used for discrete_motion, continuous_motion and synchronized_motion states.

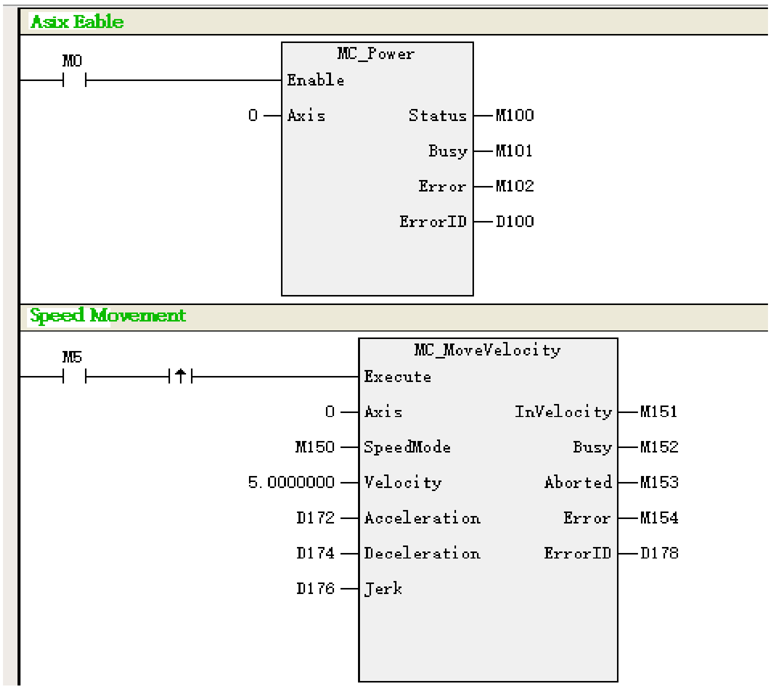
Program examples

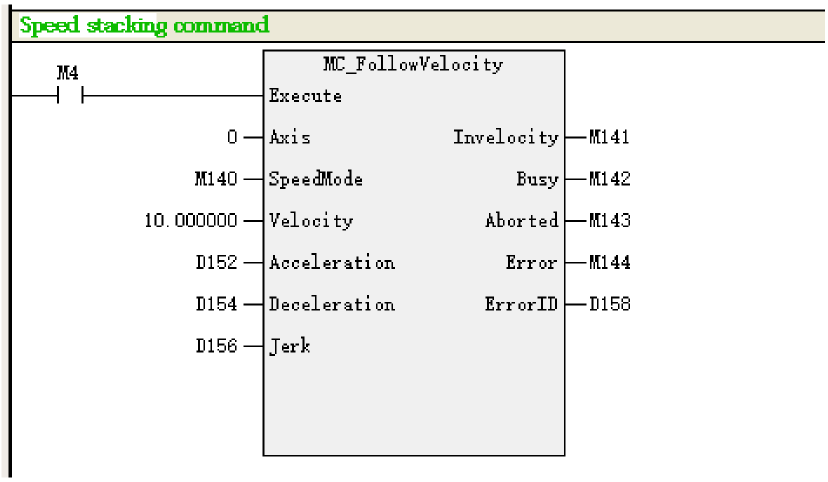
The motion speed overlay function is implemented by the following routine.

1. Enable axis Axis_0 after setting M0 to ON.

2. Set M5 to ON Rear Axle Axis_0 run at 5Unit/s.

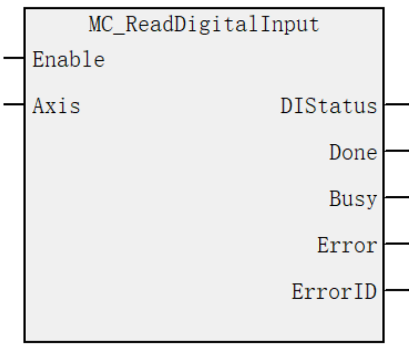
3. Set M4 to ON and superimpose the speed of the rear axle Axis_0 to (5+10) Unit/s operation.





4.3.24 MC_ReadDigitalInput (Read digital input command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~71	Axis name/axis number	Enable Valid
OUT	DIStatus	DWORD	D/R/W		The status of the digital input terminals, in accordance with the standard format of the 402 protocol, is defined as follows: Bit0 - reverse limit signal; Bit1 - forward limit signal Bit2 - origin signal; Bit3~31 - Custom	
OUT	Done	BOOL	M/S	TRUE, FALSE	Finish flag	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

This command applies to EtherCAT bus axes and local pulse shafts, does not support virtual axis mode, and is used to read the terminal status of the digital input terminals of the shaft.

When Enable=TRUE, the Valid signal is valid if the PDO in the requested EtherCAT bus axis is configured with 0x60fd or the left and right limits of the local pulse axis and the origin signal is not all empty.

If it is an EtherCAT bus axis, DIStatus is used to display the digital input 0x60fd of the EtherCAT bus drive at all times, please refer to the corresponding drive manual for the specific definition, such as the 0x60fd definition of the Vintron. SD700 servo drive as shown in the figure：

Object 60FDH: Digital input							
Index	60FDH						
Name	Digital input						
Object Structure	VAR	**Data Type**	Uint 32	**Data Range**	0~0xFFFFFFFF		
Can it be mapped	TPDO	**Accessibility**	RO	**Factory value**	0		
Logic of the current DI terminal of the reaction driver: 0 means invalid, 1 means valid							
Each of them represents the DI signal as follows:							
Bit	31~16	15~5	4	3	2	1	0
Description	Manufacturer customization	Reserved	DO45 level	DO44 Level	Origin switch	Positive overtravel switch	Reverse overtravel switch

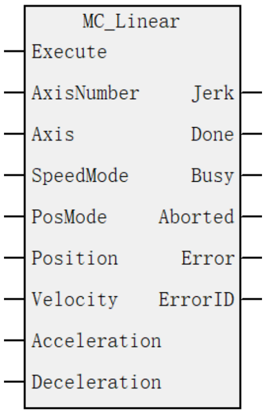
If it is a local pulse axis, it is used to display the input status of the limit and origin signals, otherwise it is displayed as 0.

Timing diagram

Omit

4.3.25 MC_Linear（Linear interpolation command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	AxisNumber	WORD	Constant/D/R/W	2~8	The number of axes	Execute
IN	Axis	WORD	D/R/W	0~71（2~8 group)	An array of axis names/axis numbers	
IN	SpeedMode	BOOL	M/S	TRUE,FALSE（default FALSE)	Speed Type: Type 0T, Type 1S	
IN	PosMode	BOOL	M/S	TRUE,FALSE（default FALSE)	Location mode: 0 relative, 1 absolute	
IN	Position	REAL	D/R/W	Positive/Negative Number/0（2~8group)	An array of target locations	
IN	Velocity	REAL	Constant/D/R/W	Positive Number	The target speed	
IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	
IN	Deceleration	REAL	Constant/D/R/W	Positive Number（default Acceleration)	Decelerate	
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed step	
OUT	Done	BOOL	M/S	TRUE, FALSE	Finish stopping	
OUT	Busy	BOOL	M/S	TRUE,	In-Run flag	

				FALSE		
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interrupts	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and directives

MC_MoveLinear command is used to control the axis linear interpolation function, and the rising edge is active.

This command can only be triggered if the MC_Power command is invoked to switch all axes in the axis group to the StandStill state.

Triggering this command during single-axis motion (e.g. point movement, torque control, return, stop, etc.) is invalid.

After triggering this command, the single-axis PLCOpen state machine is in synchronous motion mode, and the single-axis motion command cannot be interrupted during the movement, and the single-axis PLCOpen state machine enters the StandStill state after the interpolation curve is completed, and the single-axis motion command can be executed at this time.

Parameter description

Velocity indicates the target speed of the interpolator, such as two-axis interpolation, the single-axis running speed can be calculated using the following formula:

$$\text{Single axis running speed} = \frac{\text{Single axis running distance}}{\dfrac{\sqrt{(\text{Axis 1 running distance})^2 + (\text{Axis 2 running distance})^2}}{\text{Synthesis speed}}}$$

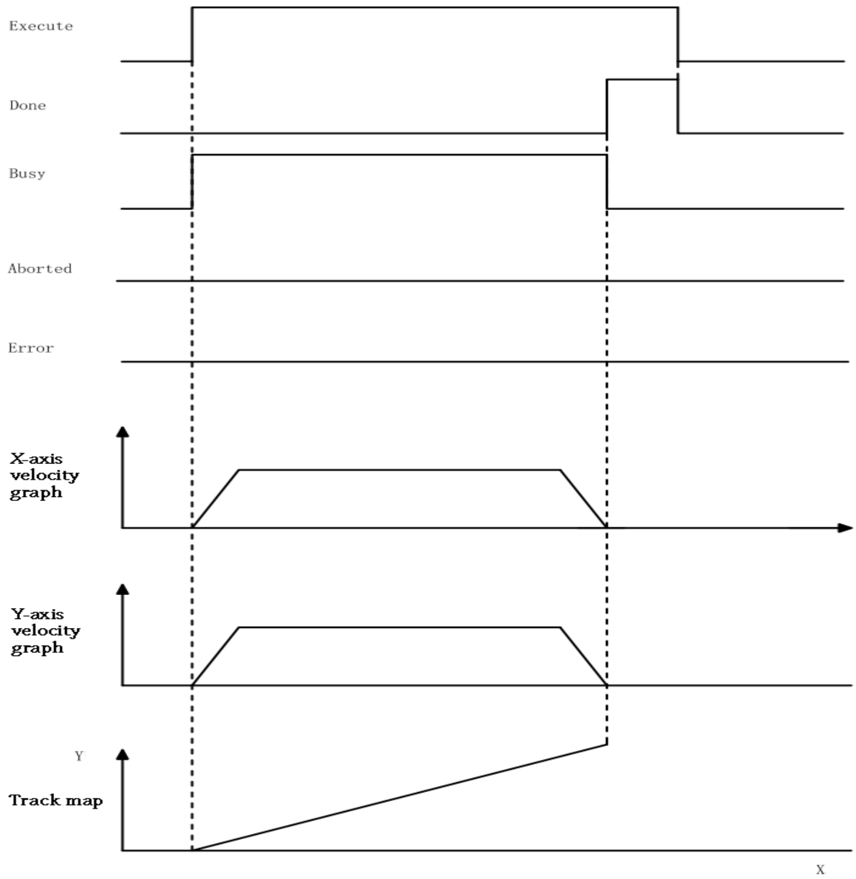
Position is used to set the target position or shift, Position[0] represents the position displacement component of axis 1, Position[1] represents the position displacement component of axis 2, Position[2] represents the position displacement component of axis 3, and Position[3] represents the position displacement component of axis 4.

Axis is used to set the axis number of the interpolated axis, Axis[0] represents the axis number of axis 1, Axis[1] represents the axis number of axis 2, Axis [2] represents the axis number of axis 3, and Axis[3] represents the axis number of axis 4.

A total of 8 axes of linear interpolation are supported.

Timing diagrams

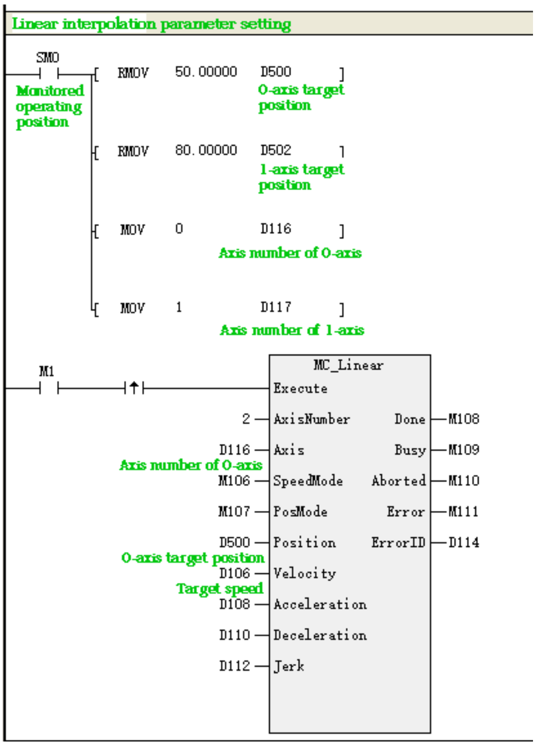
Call a line interpolation command to interpolate in the X axis (Axis0 axis) and Y axis (Axis1 axis).

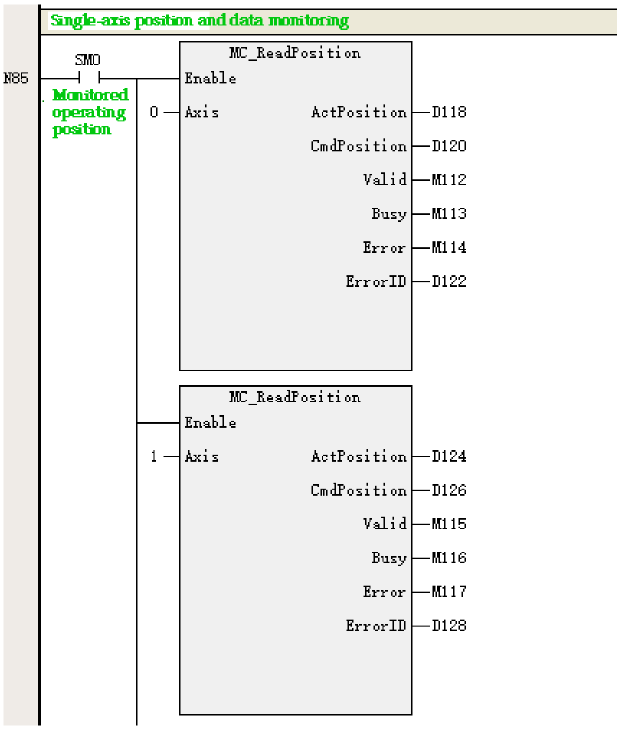


Example of a program:

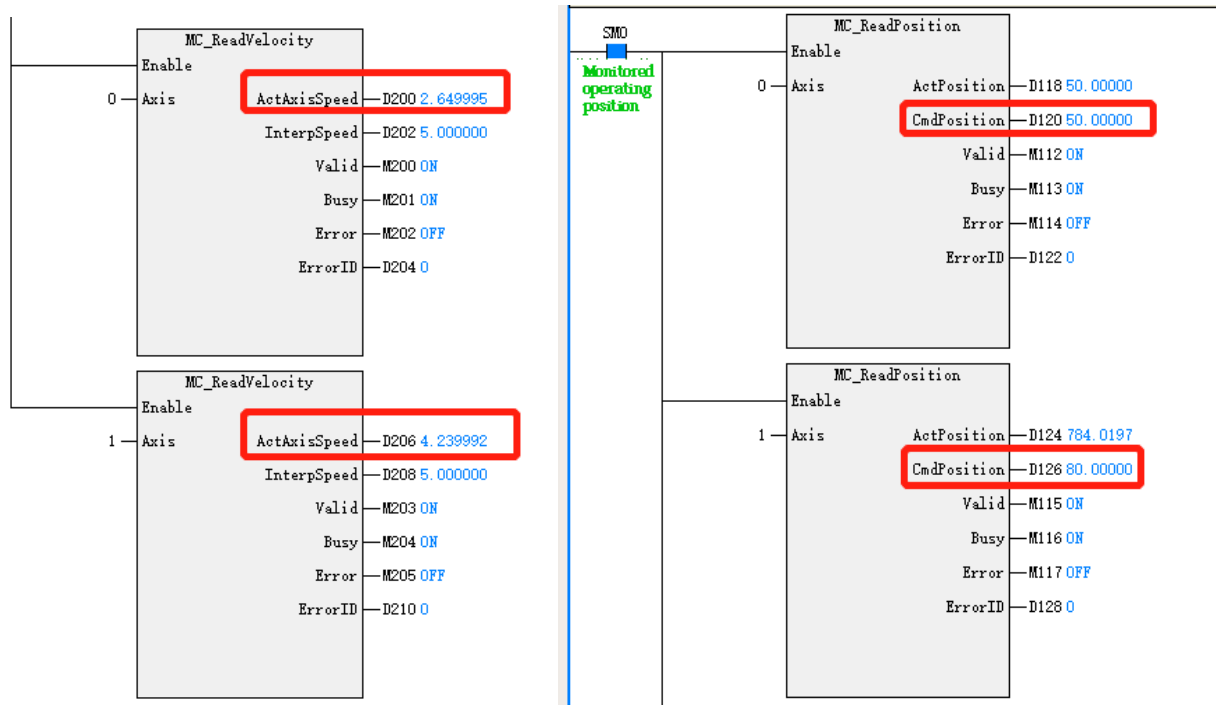
This routine uses relative positioning to position the x-axis and y-axis (50, 80).

Set the target speed to 5Unit/s, you can monitor the single-axis speed and single-axis end position.

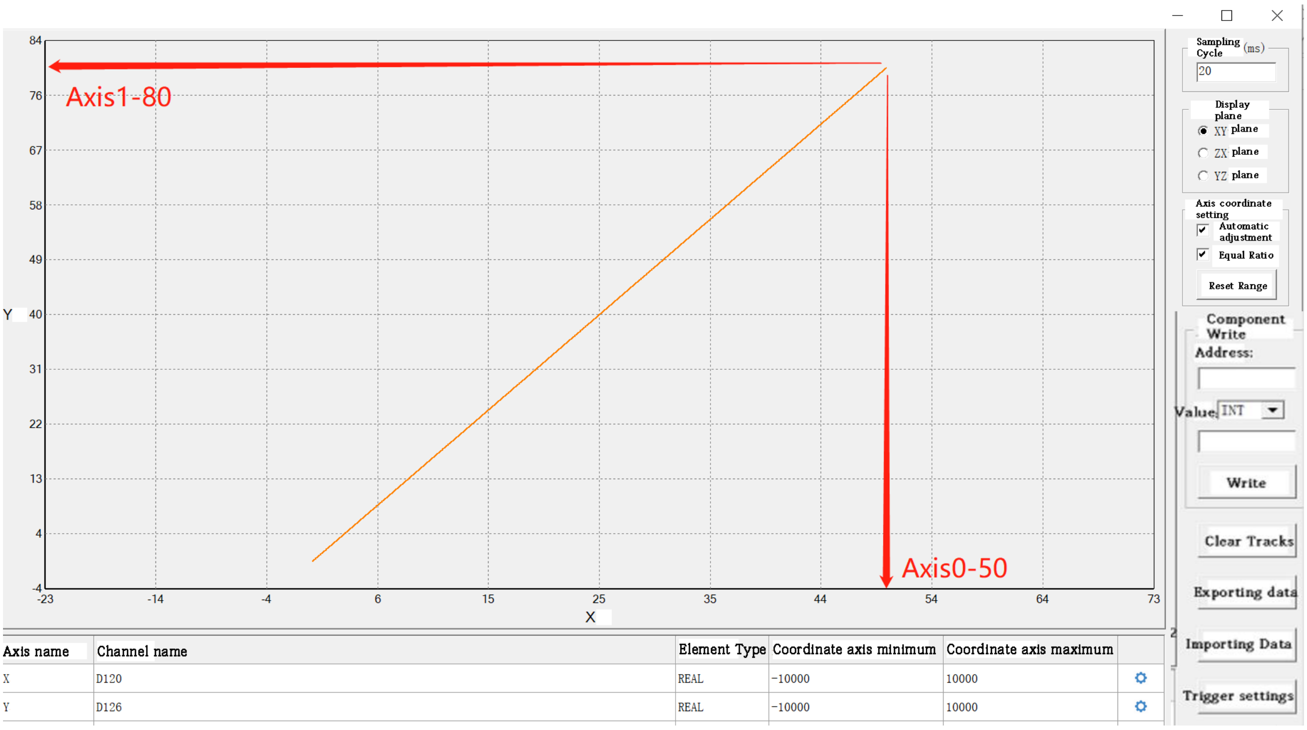




Single-axis speed and single-axis end position monitoring:

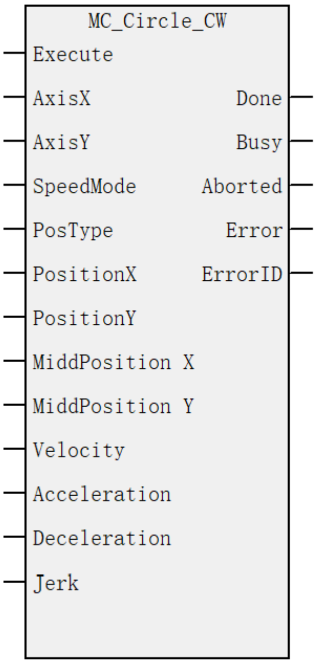


Interpolation curve monitoring:



4.3.26 MC_Circle_CW (Clockwise circular arc interpolation command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	AxisX	WORD	Constant/D/R/W	0~71	X axis name/axis number	Execute
IN	AxisY	WORD	Constant/D/R/W	0~71	Y-axis name/axis number	
IN	SpeedMode	BOOL	M/S	TRUE,FALSE（default FALSE）	Speed Type: Type 0T, Type 1S	
IN	PosType	WORD	Constant/D/R/W	0~3（default0）	Location Type: 0 relative to the center of the circle 1 Absolute center of the circle 2 points on the relative circular arc 3 points on the absolute circular arc	
IN	PositionX	REAL	Constant/D/R/W	Positive/Negative Number/0	The target position of the X axis	
IN	PositionY	REAL	Constant/D/R/W	Positive/Negative Number/0	Y-axis target position	
IN	MiddPosition X	REAL	Constant/D/R/W	Positive/Negative Number/0	X-axis center position/longitude	

					coordinates	
IN	MiddPosition Y	REAL	Constant/D/R/W	Positive/Negative Number/0	Y-axis center position/longitude point coordinates	
IN	Velocity	REAL	Constant/D/R/W	Positive Number	The target speed	
IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	
IN	Deceleration	REAL	Constant/D/R/W	Positive Number （default Acceleration）	Decelerate	
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed step	
OUT	Done	BOOL	M/S	TRUE, FALSE	Finish stopping	
OUT	Busy	BOOL	M/S	TRUE,FALSE	In-Run flag	
OUT	Aborted	BOOL	M/S	TRUE,FALSE	Execution interrupts	
OUT	Error	BOOL	M/S	TRUE,FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

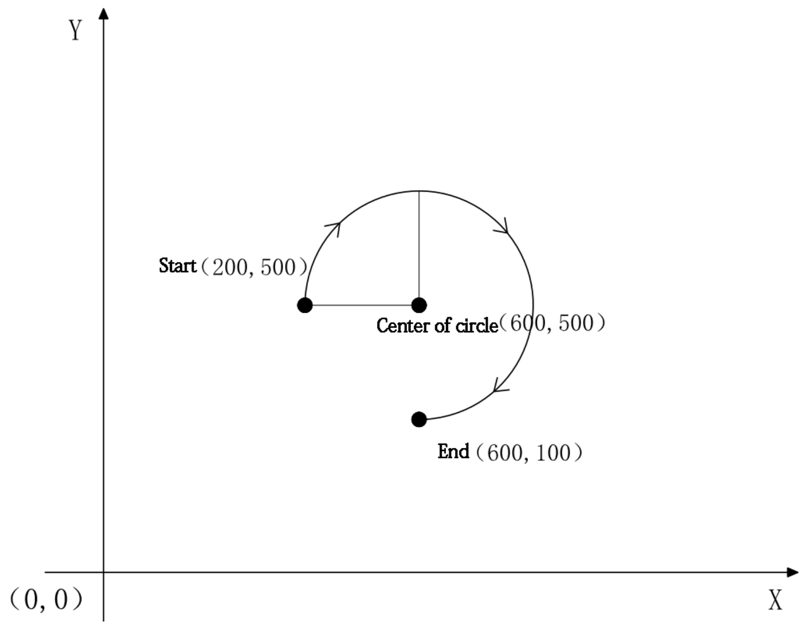
Description of functions and commands

MC_ Circle_ CW command is used to control the axis circular arc interpolation function, and the rising edge is valid. On the specified plane and the circular arc determined by the center or radius, the workpiece is processed in a clockwise circular arc at the feed rate set by the parameter Velocity.

A. Selection of interpolation type

1. PosType=0 represents circular arc interpolation based on the relative center.

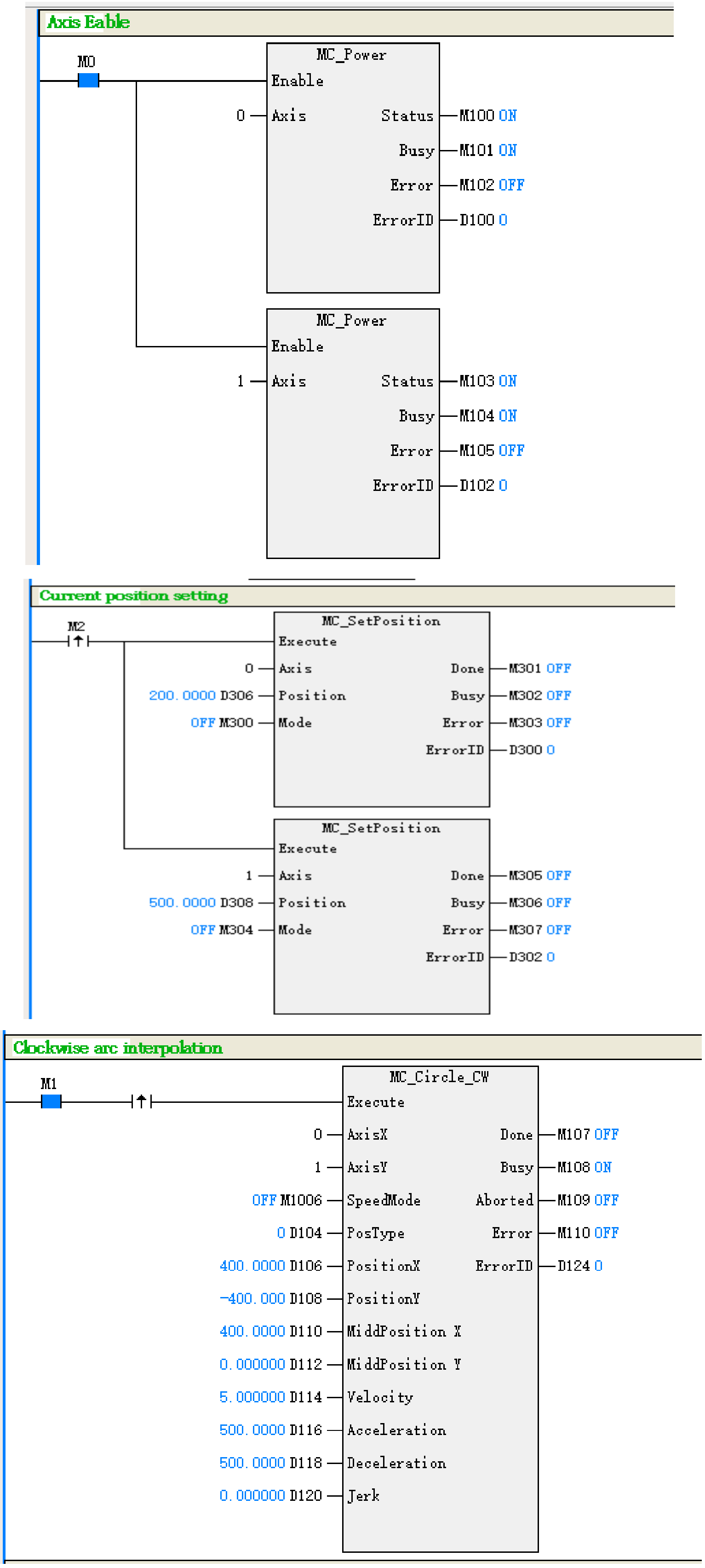
Assuming the current axis position is (200500), you want to draw the circular arc shown in the following figure:



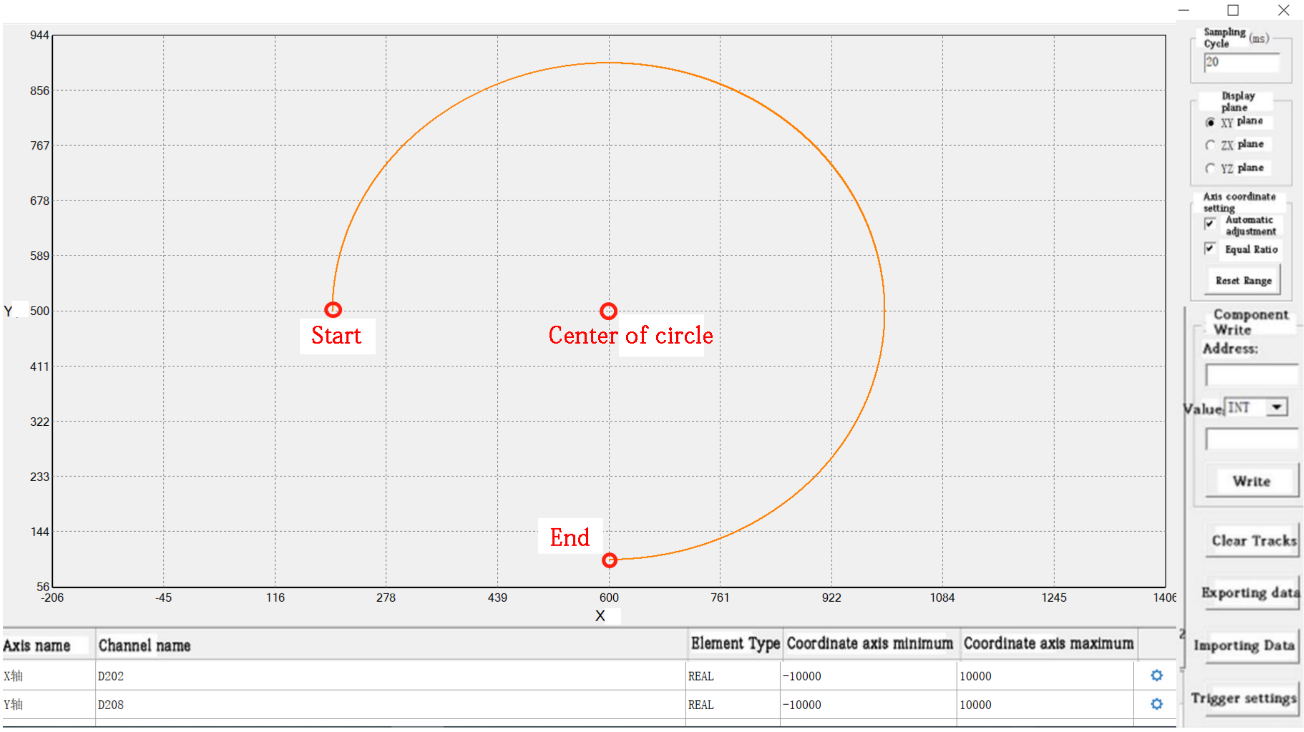
Programming:

The relative center mode is adopted, and the track target adopts the relative address, which refers to the distance of X and Y axes during the movement from the current position to the target.

The displacement of the end point relative to the starting point in the above figure is (400, - 400), and the displacement of the circle center relative to the starting point is (400, 0).

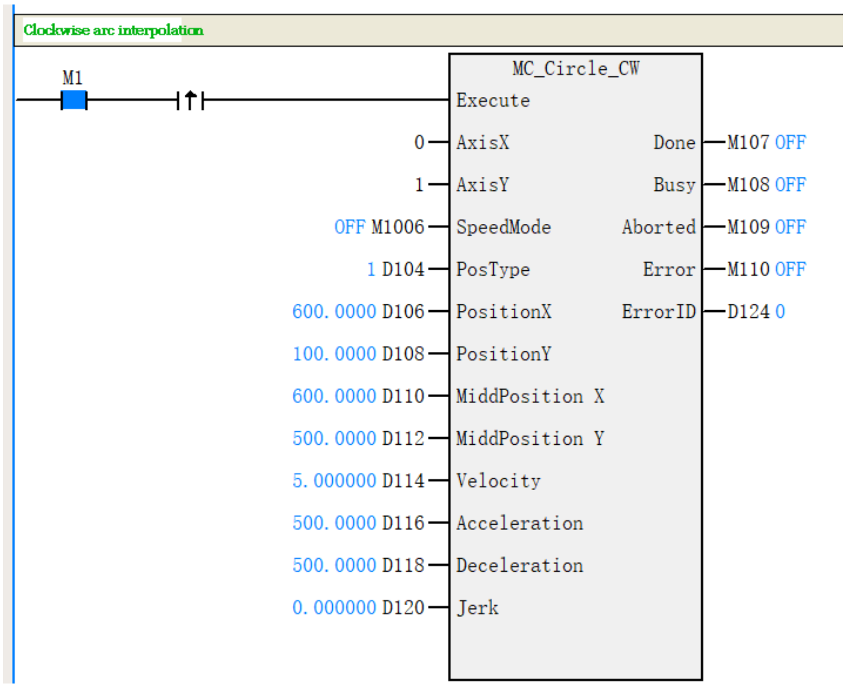


Circular arc interpolation track monitoring:



2. PosType=1 represents circular arc interpolation based on the absolute center.

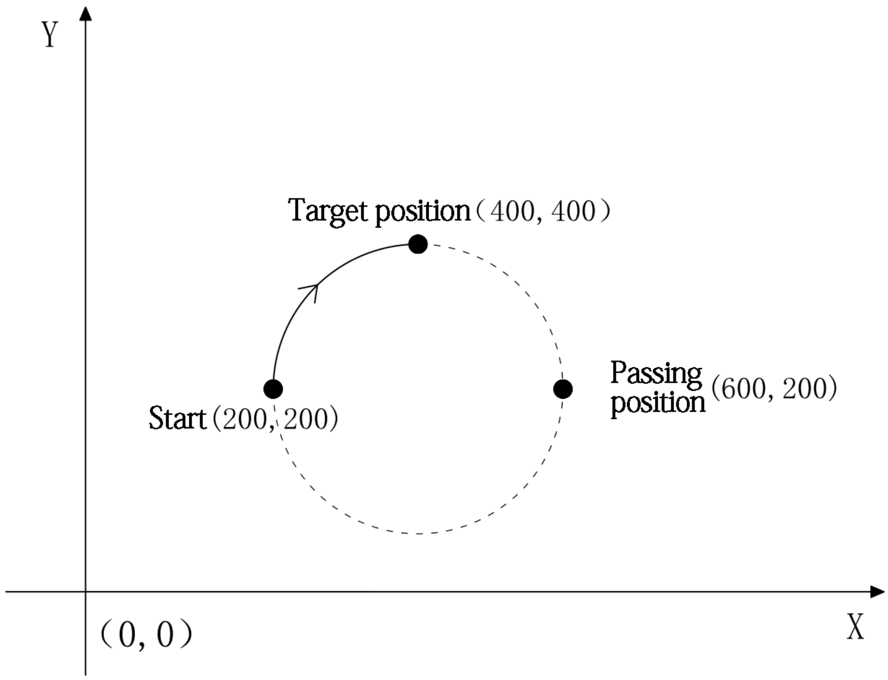
If you still draw an circular arc with PosType=0, you only need to change PosType to 1. The track target uses an absolute address, which refers to the absolute position coordinates of the target position on the X and Y axes. Other programs remain unchanged, and the interpolation command program is changed as follows:



The track monitored by interpolation is the same as when PosType=0.

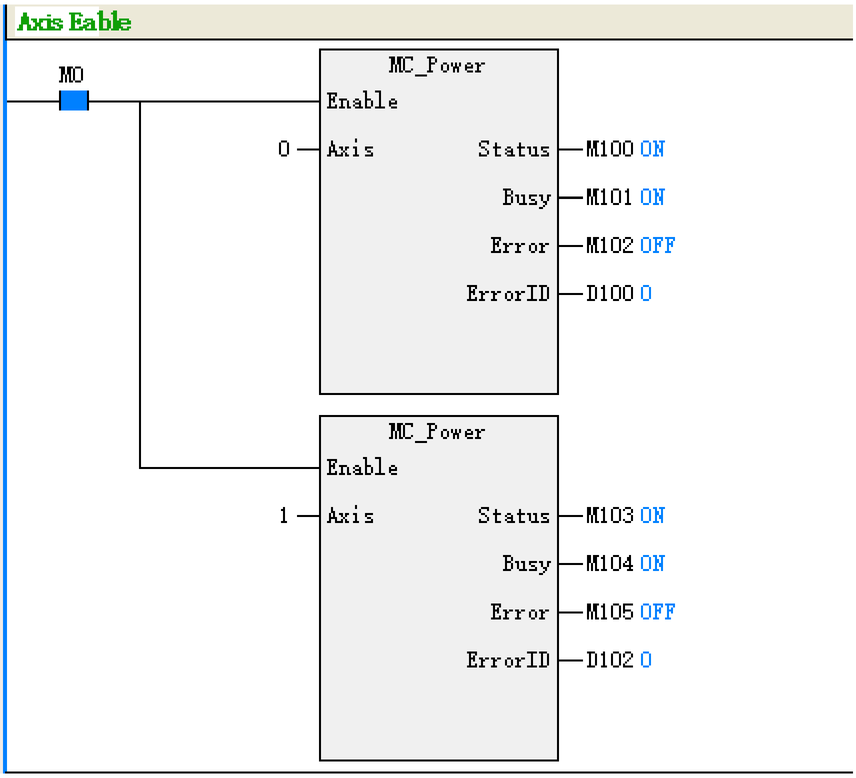
3. PosType=2 means the circular arc interpolation is performed according to the point passing through the relative circular arc .

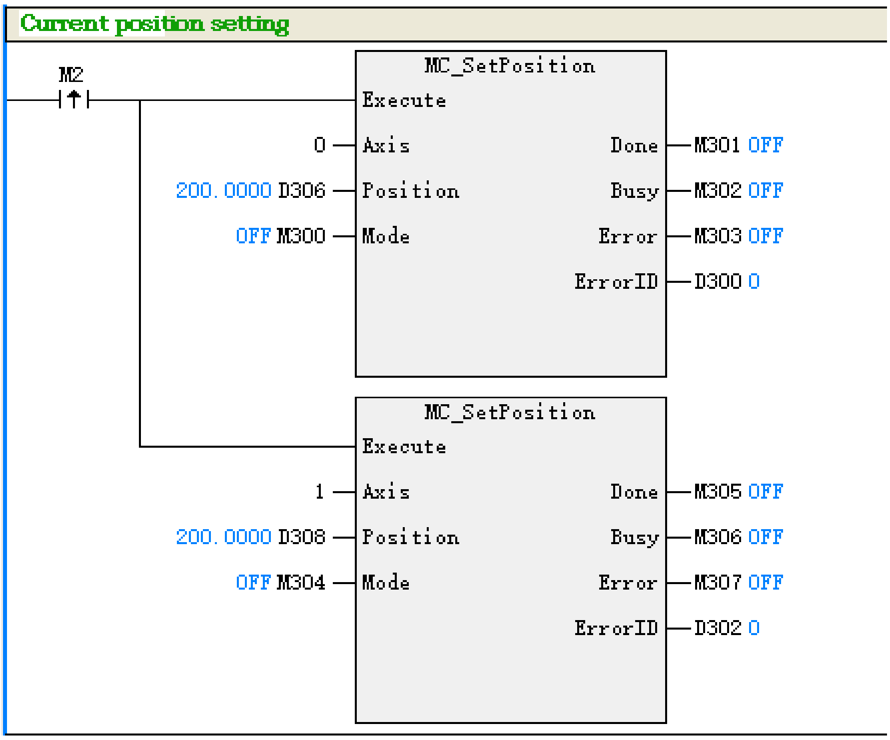
Assuming the current axis position is (200200), you want to draw the circular arc shown in the following figure:

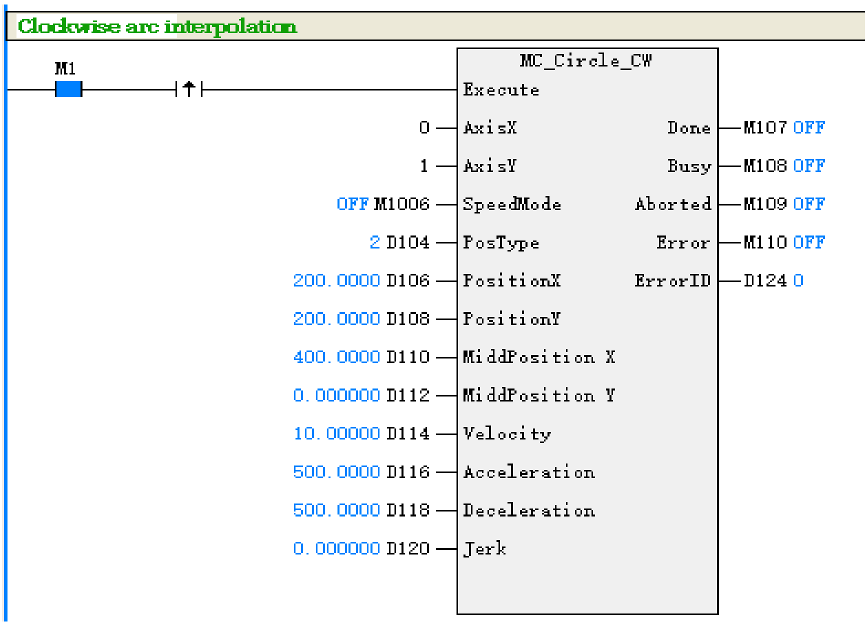


Programming:

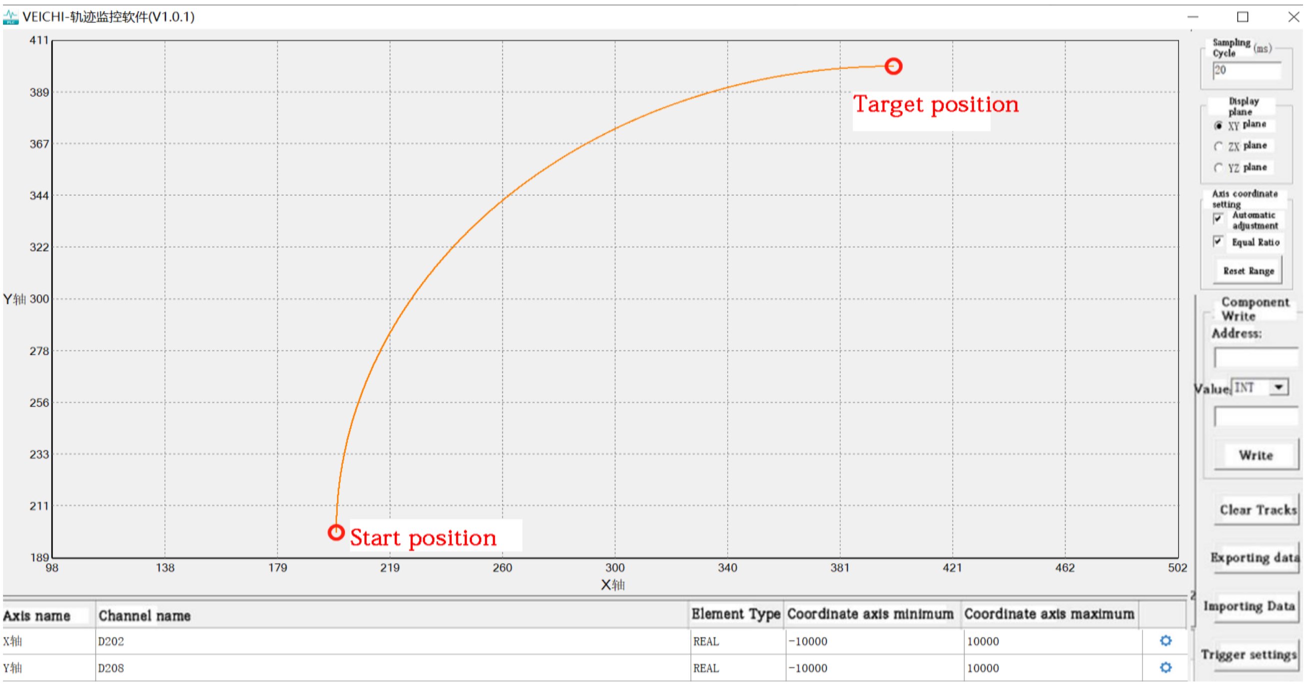
The mode of passing a point on a relative circular arc is adopted. The displacement of the target position relative to the starting point is (200200), and the displacement of the passing point relative to the starting point is (400, 0).





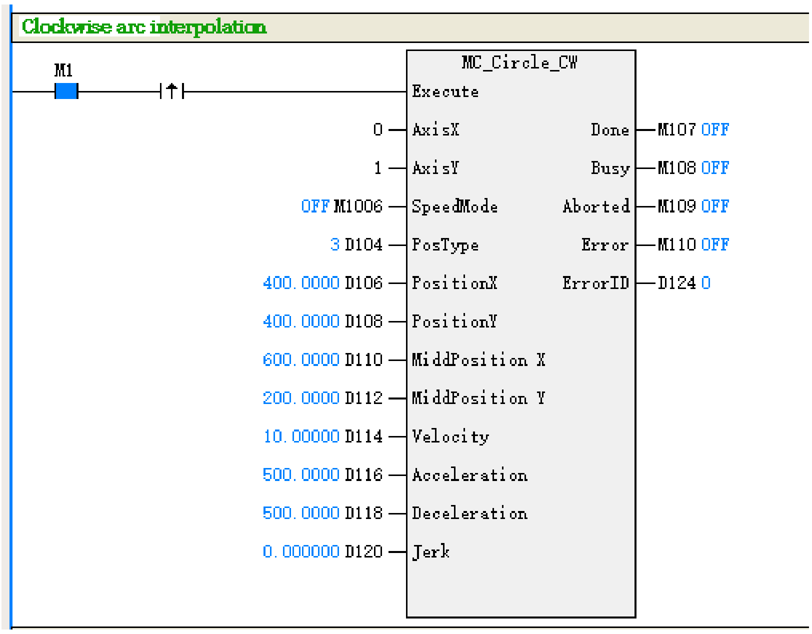


Circular arc interpolation track monitoring:



4. PosType=3 means the circular arc interpolation is performed according to the point passing through the absolute circular arc.

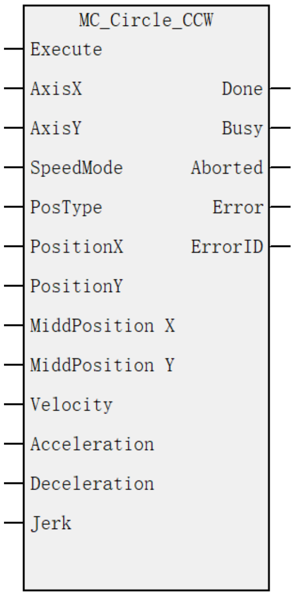
If you still draw an circular arc with PosType=2, you only need to change PosType to 3. The X axis target position, Y axis target position, X axis circular arc passing point coordinates, and Y axis circular arc passing point coordinates are absolute cordinates.other programs remain unchanged, and the interpolation command program is changed as follows：



The track monitored by interpolation is the same as when PosType=2.

4.3.27 MC_Circle_CCW (Counterclockwise circular arc interpolation command)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	AxisX	WORD	Constant/D/R/W	0~71	X axis name/axis number	Execute
IN	AxisY	WORD	Constant/D/R/W	0~71	Y-axis name/axis number	
IN	SpeedMode	BOOL	M/S	TRUE,FALSE （defaultFALSE）	Speed Type: Type 0T, Type 1S	
IN	PosType	WORD	Constant/D/R/W	0~3（default 0）	Location Type: 0 relative to the center of the circle 1 Absolute center of the circle 2 points on the relative circular arc 3 points on the absolute circular arc	
IN	PositionX	REAL	Constant/D/R/W	Positive/Negative Number/0	The target position of the X axis	
IN	PositionY	REAL	Constant/D/R/W	Positive/Negative Number/0	Y-axis target position	
IN	MiddPosition	REAL	Constant/D/R/W	Positive/Negative	X-axis center	

	X			Number/0	position/longitude coordinates	
IN	MiddPosition Y	REAL	Constant/D/R/W	Positive/Negative Number/0	Y-axis center position/longitude point coordinates	
IN	Velocity	REAL	Constant/D/R/W	Positive Number	The target speed	
IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	
IN	Deceleration	REAL	Constant/D/R/W	Positive Number	Decelerate	
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed step	
OUT	Done	BOOL	M/S	TRUE, FALSE	Finish stopping	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interrupts	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

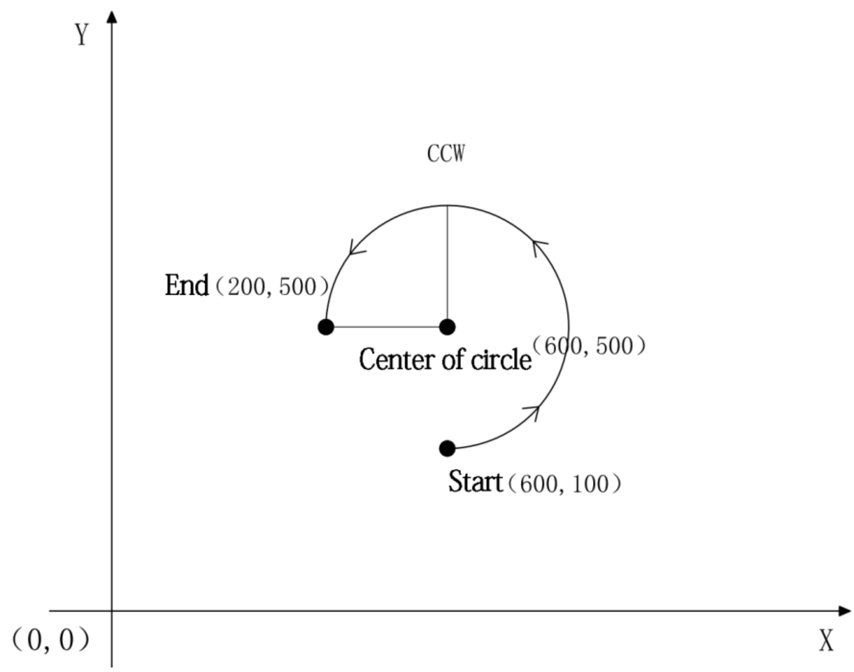
Description of functions and commands

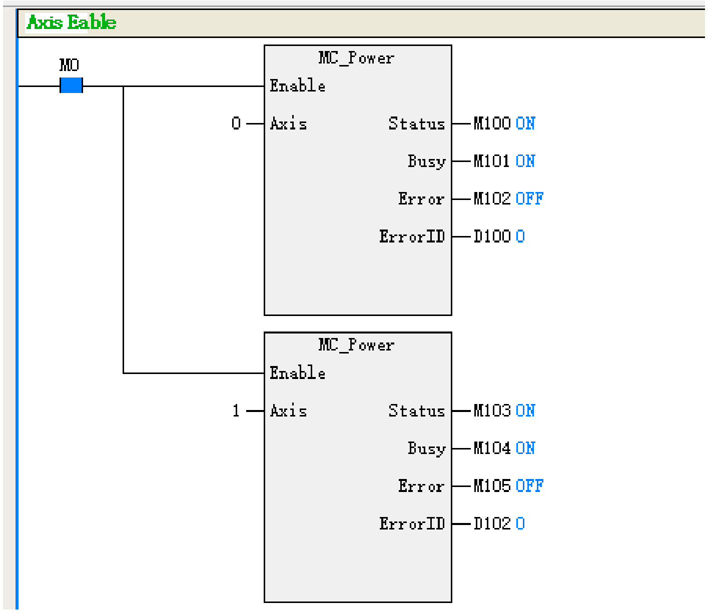
The MC_Circle_CCW command is used to control the axis circular arc interpolation function, and the rising edge is valid. The object is machined counterclockwise on the specified plane and on the circular arc determined by the center or radius of the circle, at the feed rate set by the parameter Velocity.

● Interpolation type selection

1. PosType = 0 represents counterclockwise circular arc interpolation according to the relative circle center.

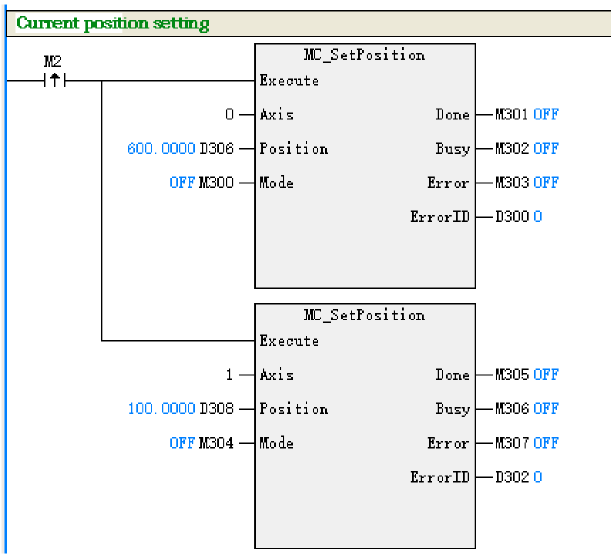
Suppose the current axis position is (600, 100) and wish to draw the circular arc as shown in the following figure:

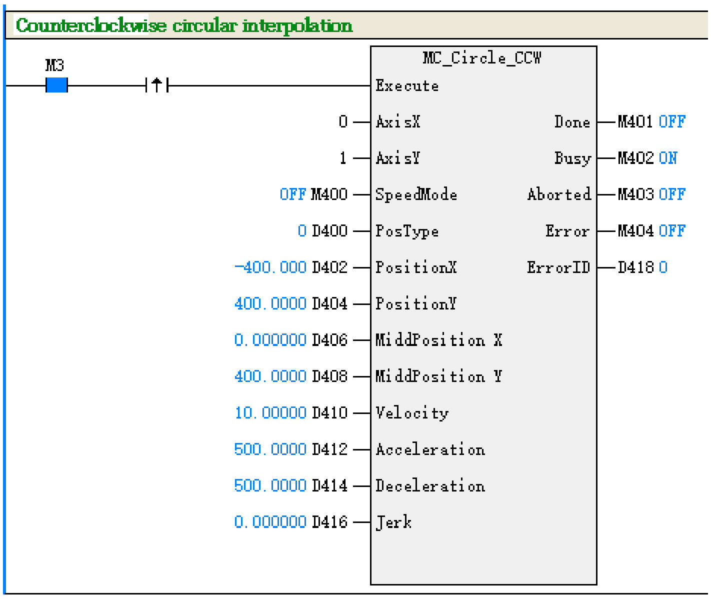




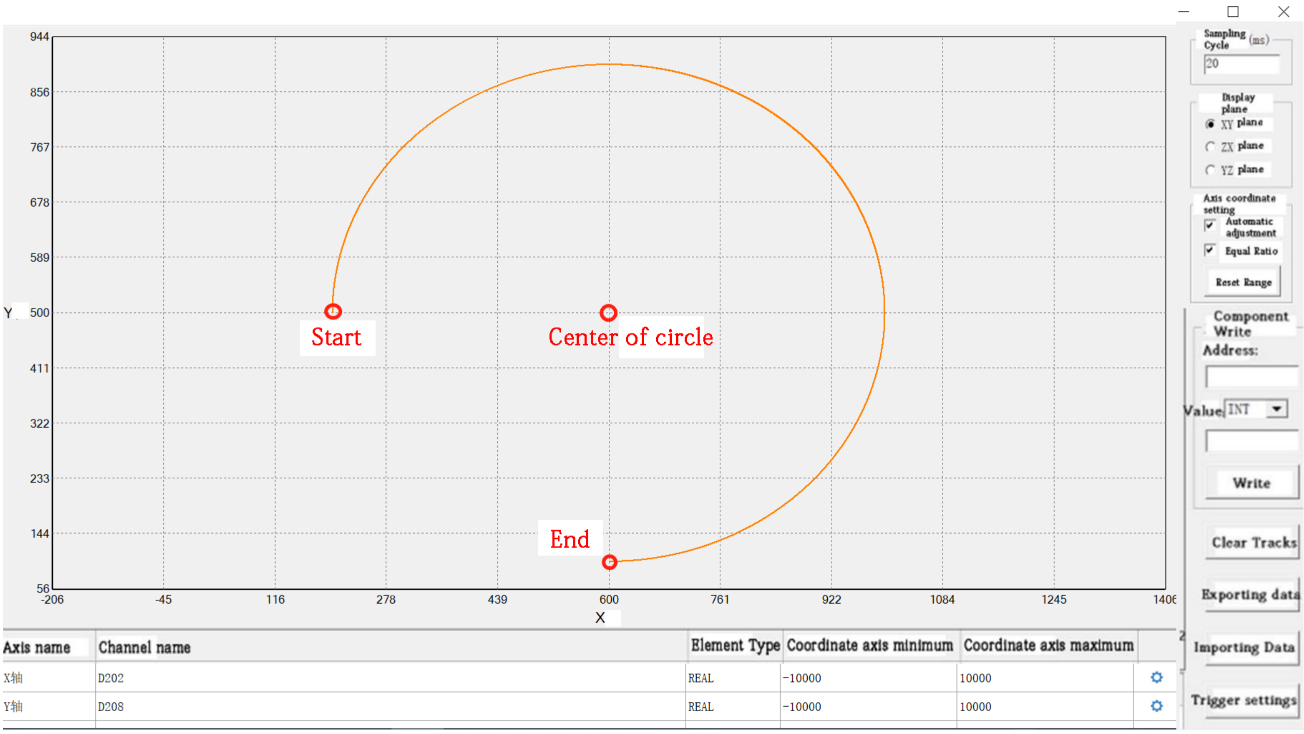
Program programming:

Using the relative circle center mode, the displacement of the end point with respect to the starting point is (-400, 400) and the displacement of the center of the circle with respect to the starting point is (0, 400).



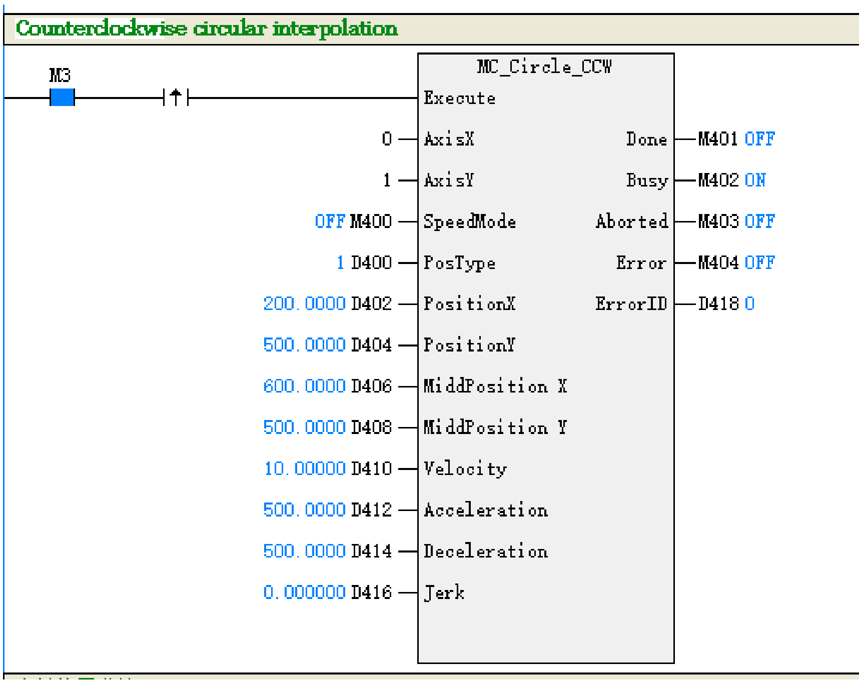


Circular arc interpolation trajectory monitoring.



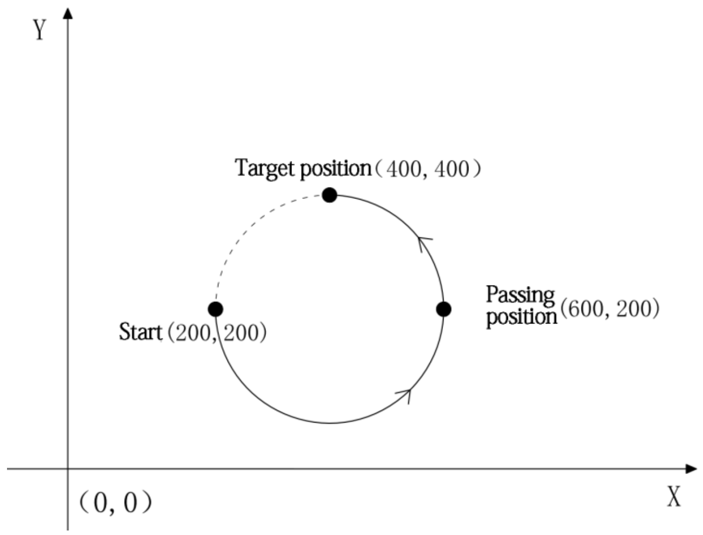
2. PosType = 1 represents the counterclockwise circular arc interpolation according to the absolute circle center.

If you still draw the circular arc with PosType = 0, then you only need to change PosType to 1, where X-axis target position, Y-axis target position, X-axis circle center position, Y-axis circle center position are all absolute value coordinates. Other procedures remain unchanged, and the interpolation command procedure is changed to the following.



The interpolated monitored trajectory is the same as when PosType = 0.

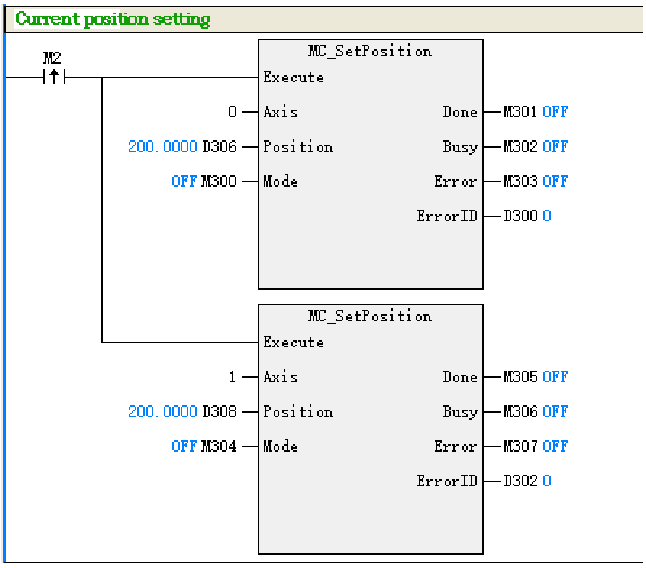
3. PosType = 2 represents counterclockwise circular arc interpolation according to the points passing on the relative circular arc.

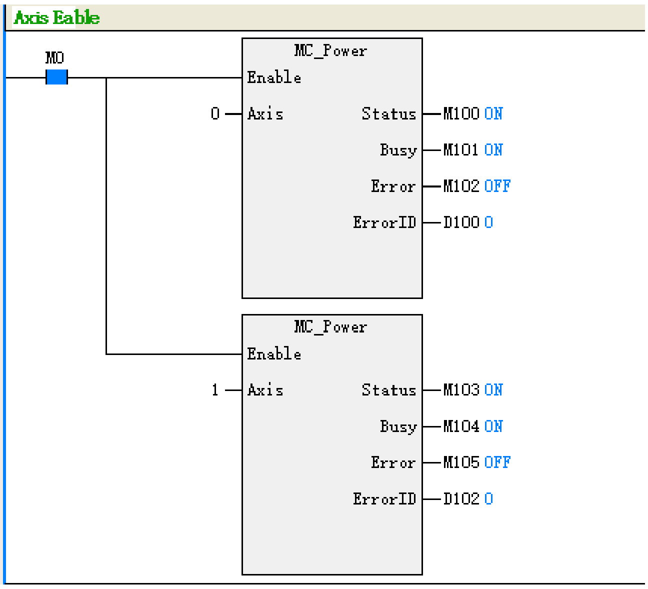


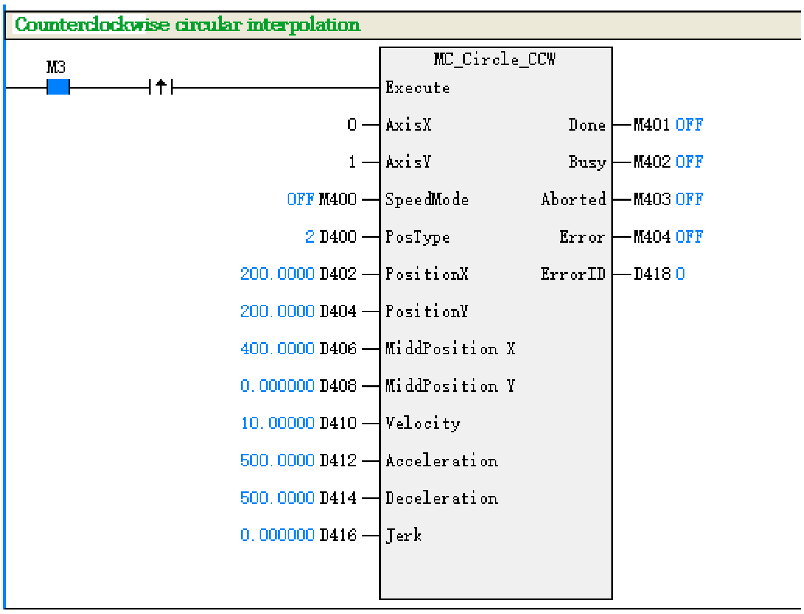
Assuming the current axis position is (200, 200), it is desired to draw the circular arc shown in the following figure:

Program programming:

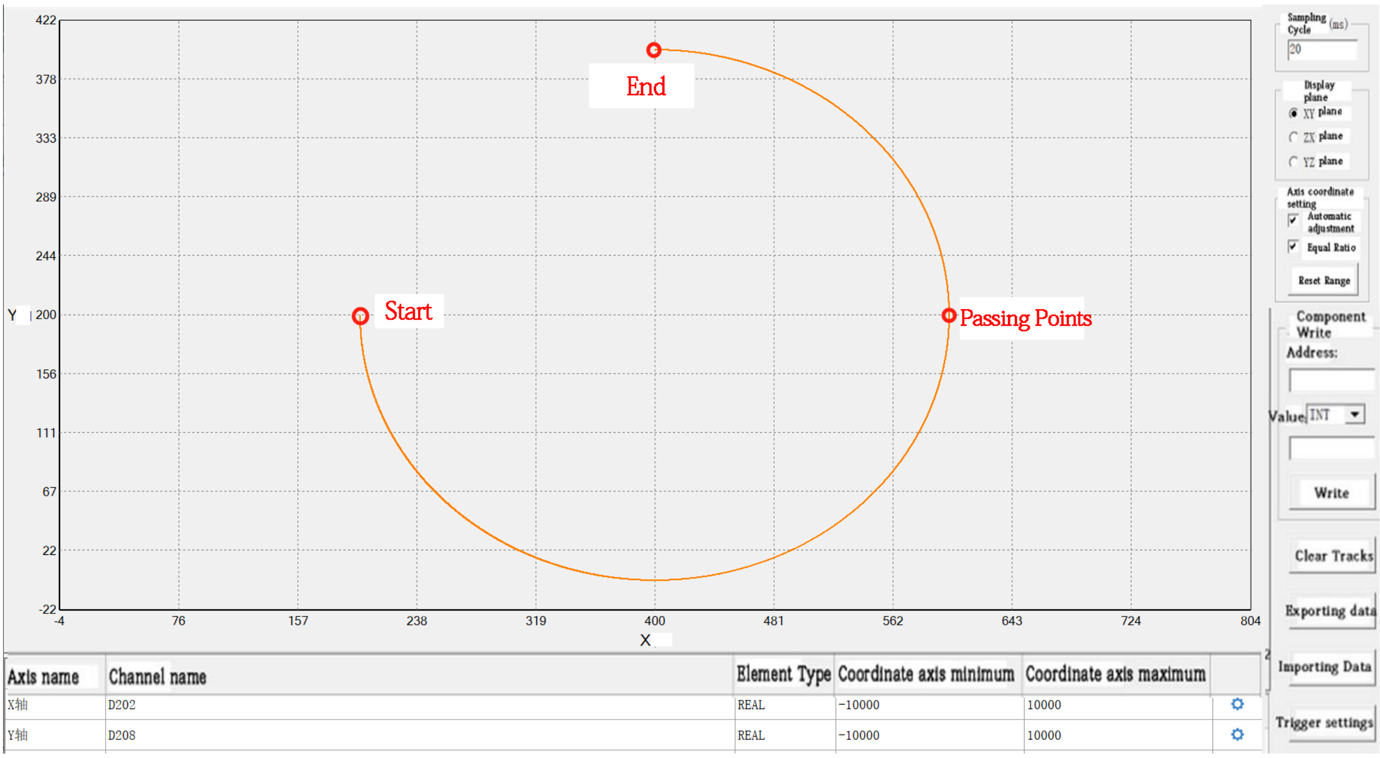
Using the mode of passing through points on the relative circular arc, the displacement of the target position with respect to the starting point is (200, 200) and the displacement of the passing point with respect to the starting point is (400, 0).





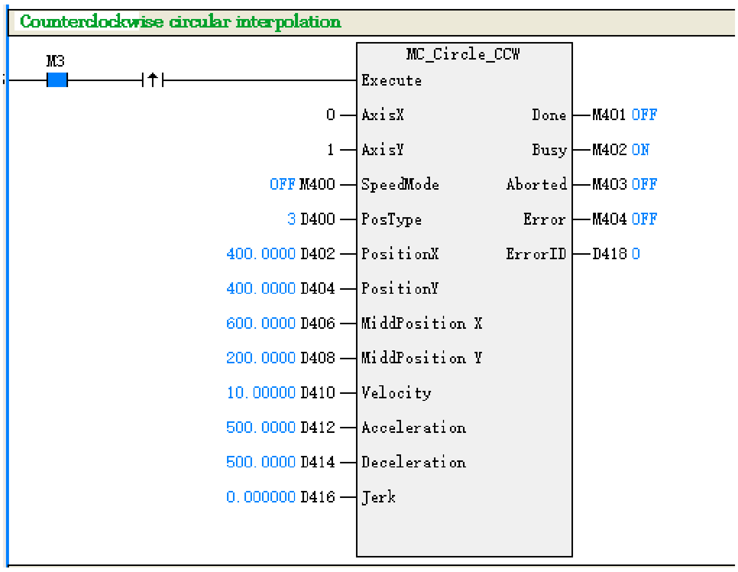


Circular arc interpolation trajectory monitoring:



4. PosType = 3 represents counterclockwise circular arc interpolation based on the points passing over the absolute circular arc.

If you still draw the circular arc with PosType = 2, then you only need to change PosType to 3, where X-axis target position,

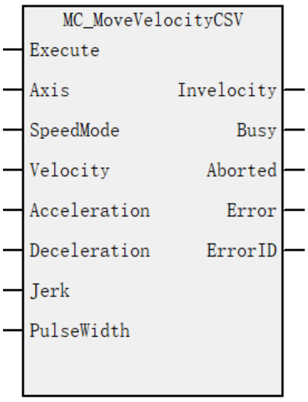


Y-axis target position, X-axis circular arc passing point coordinates, and Y-axis circular arc passing point coordinates are all absolute value coordinates. Other procedures remain unchanged, and the interpolation command procedure is changed to the following:

The interpolated monitored trajectory is the same as when PosType = 2.

4.3.28 MC_MovevelocityCSV(Speed command basing on adjustable CSV pulse width)

Drawing Block:



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Decription	Enable
IN	Axis	WORD	Constant/D/R/W	0~63	Axis name/axis number	Execute
IN	SpeedMode	BOOL	M/S	TRUE,FALSE (DEFAULT FALSE)	Speed type 0T type, 1S type	
IN	Velocity	REAL	Constant/D/R/W	Positive Number	Target speed	
IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	
IN	Deceleration	REAL	Constant/D/R/W	Positive Number (Default Acceleration)	Deceleration	
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed step	
IN	PulseWidth	Word	常数/D/R/W	1-9999 (Default：5000)	Pulse width in 0.01%	
OUT	Invelocity	BOOL	M/S	TRUE,FALSE	Reaching the target speed marker	
OUT	Busy	BOOL	M/S	TRUE,FALSE	Running flags	
OUT	Aborted	BOOL	M/S	TRUE,FALSE	Execution interruptions	
OUT	Error	BOOL	M/S	TRUE,FALSE	Wrong sign	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

This command controls the bus-type servo axis using the cyclic synchronous velocity (CSV) mode, which puts the axis' PLCOpen state machine in Continuous Motion mode, with the same basic function as MC_MoveVelocity.

The PDO needs to add three object dictionaries 0x6060,0x6061 and 0x60ff.

This command first writes the object dictionary 0x6060 to 9, switches the drive to CSV mode, and then converts the target speed into 32 shaped data in the object dictionary 0x60FF, where the target speed is increased or decreased according to the set acceleration or deceleration.

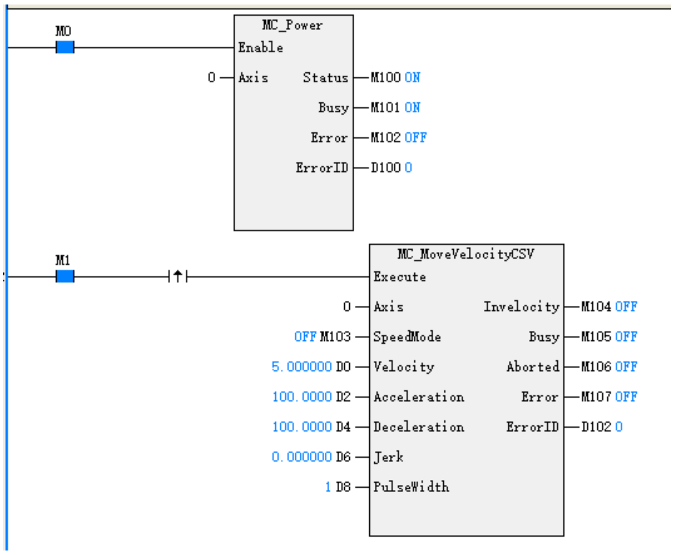
If you want to stop the axis motion after calling this instruction, you can call MC_Stop and Mc_Halt instructions.

Note that when this instruction is called, the MC_MoveSuperImposed instruction cannot be called to perform motion superimposition.

Program examples

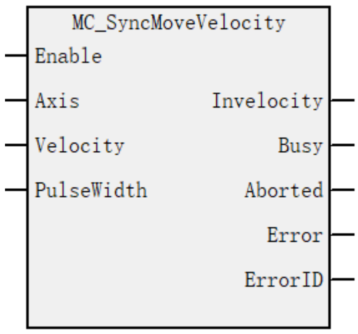
The following routine implements velocity motion function basing on the CSV mode

1. Enable axis Axis_0 after setting M0 to ON.
2. Set M1 to ON Rear Axle Axis_0 run at setting speed.



4.3.29 MC_SyncMoveVelocity（Synchronous speed control command basing on adjustable CSV pulse width）

Drawing Block



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant/D/R/W	0~63	Axis name/axis number	Enable
IN	Velocity	REAL	Constant/D/R/W	Positive Number	The target speed	
IN	PulseWidth	Word	Constant/D/R/W	1-9999	Pulse width ,the unit is	

				(Default : 5000)	0.01%.	
OUT	Invelocity	BOOL	M/S	TRUE, FALSE	Reaching the target speed flag	
OUT	Busy	BOOL	M/S	TRUE, FALSE	Running sign	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interruptions	
OUT	Error	BOOL	M/S	TRUE FALSE	Error flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

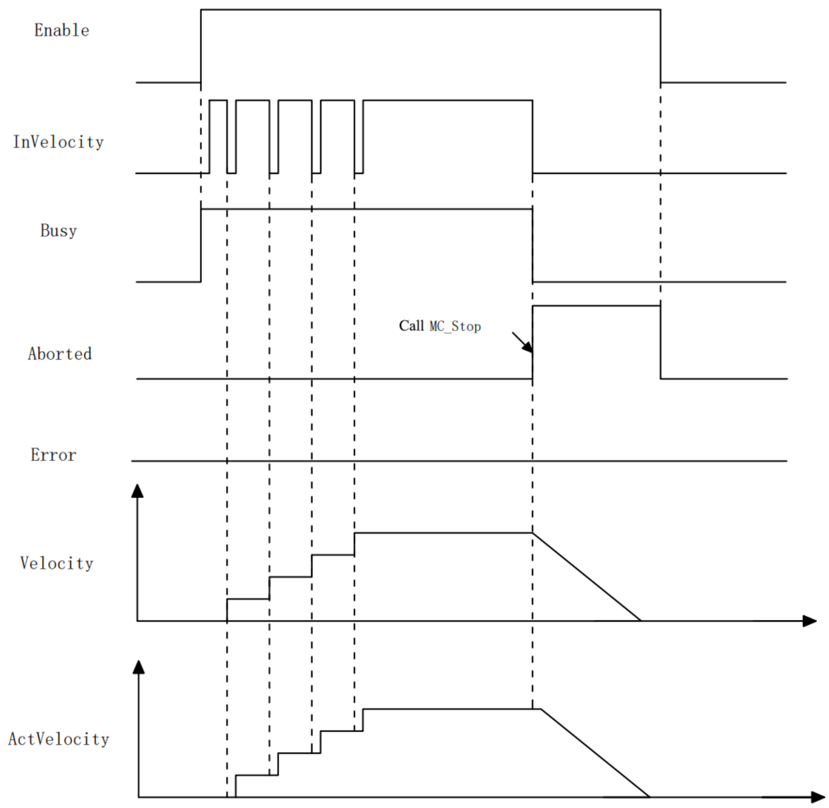
This command uses CSV mode to control bus-type servo axis, which puts the axis in continuous motion mode.

The PDO needs to add three object dictionaries 0x6060,0x6061 and 0x60ff.

This command first writes the object dictionary 0x6060 to 9, switches the drive to CSV mode, and then converts the target speed to 32 shaped data into the object dictionary 0x60FF.

This command allows the axis speed to be modified in real time in the program without re-triggering. The modified speed has no acceleration or deceleration process and is directly converted to pulse units and written into 0x60FF.

If you want to stop the axis motion after calling this command, you can call MC_Stop and Mc_Halt commands.

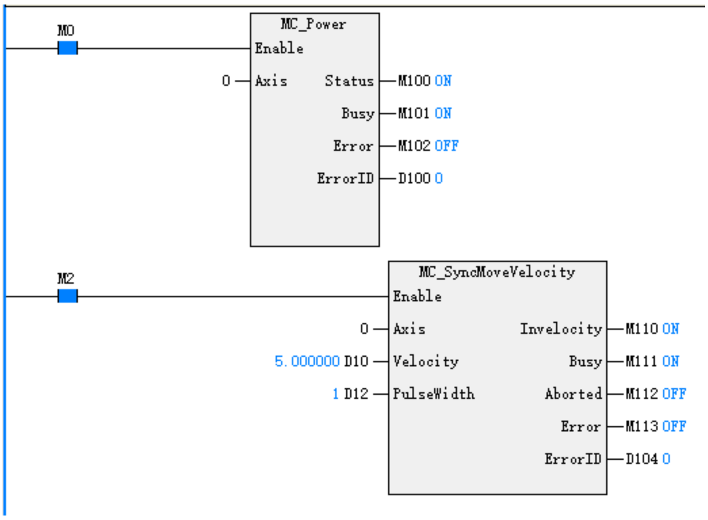


Note that when this command is called, the MC_MoveSuperImposed command cannot be called for motion overlay actions.

Program examples:

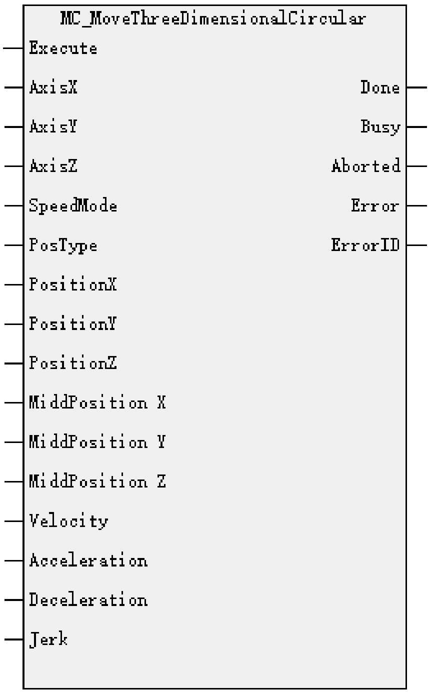
The following routine implements the function of synchronous speed motion based on CSV mode with adjustable speed during running.

1. Enable axis Axis_0 after setting M0 to ON.
2. Set M2 to ON Rear Axle Axis_0 run at setting speed



4.3.30 MC_MoveThreeDimensionalCircular（3D circular arc interpolation command）

Draw Block



Command list format

Input/Output	Name	Date Type	Applicable (soft components)	Range	Desription	Enable
IN	AxisX	WORD	Constant/D/R/W	0~63	X axis name/axis number	Execute
IN	AxisY	WORD	Constant/D/R/W	0~63	Y axis name/axis number	
IN	AxisZ	WORD	Constant/D/R/W	0~63	Z axis name/axis	

					number	
IN	SpeedMode	BOOL	M/S	TRUE,FALSE (DEFAULT FALSE)	Speed type. 0 T-type, 1 S-type	
IN	PosType	WORD	Constant/D/R/W	0~1 (Default 0)	Position Type. 0Relative position 1Absolute position	
IN	PositionX	REAL	Constant/D/R/W	Positive/Negative Number/0	X axis target position	
IN	PositionY	REAL	Constant/D/R/W	Positive/Negative Number/0	Y axis target position	
IN	PositionZ	REAL	Constant/D/R/W	Positive/Negative Number/0	Z axis target position	
IN	MiddPosition X	REAL	Constant/D/R/W	Positive/Negative Number/0	Position on the X-axis circular arc	
IN	MiddPosition Y	REAL	Constant/D/R/W	Positive/Negative Nuer/0	Position on the Y-axis circular arc	
IN	MiddPosition Z	REAL	Constant/D/R/W	Positive/Negative Number/0	Position on the Z-axis circular arc	
IN	Velocity	REAL	Constant/D/R/W	Positive Number	Target speed	
IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	
IN	Deceleration	REAL	Constant/D/R/W	Positive Numeber (Degault Acceleration)	Deceleration	
IN	Jerk	REAL	Constant/D/R/W	Constant	Speed step	
OUT	Done	BOOL	M/S	TRUE, FALSE	Finish Stopping	
OUT	Busy	BOOL	M/S	TRUE, FALSE	Running flag	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interruption	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

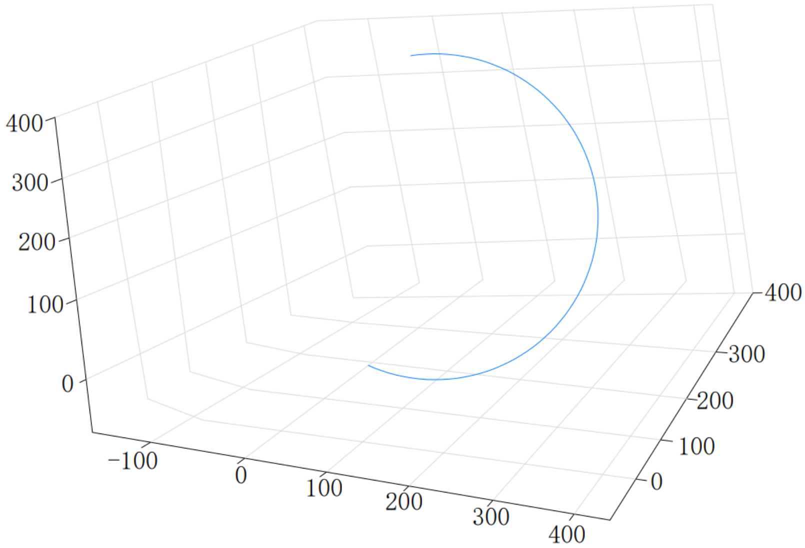
Description of functions and directives

MC_MoveThreeDimensionalCircular command is used for 3-dimensional circular arc interpolation function, and the rising edge is valid.

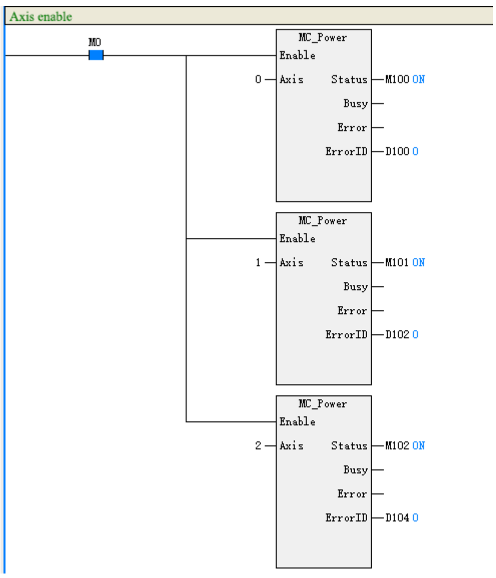
Spatial circular arc interpolation function is based on the current point and circular arc command parameters set the target and intermediate points, from three points to determine the circular arc, and realize the spatial circular arc interpolation motion, the coordinates are three-dimensional coordinates, at least three axis need to move along the X-axis, Y-axis and Z-axis respectively.

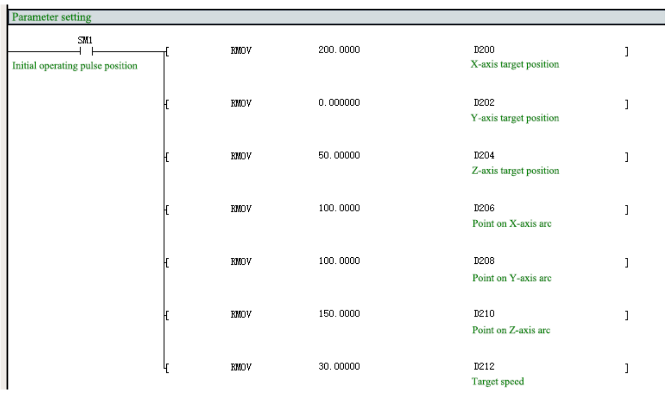
Example of program:

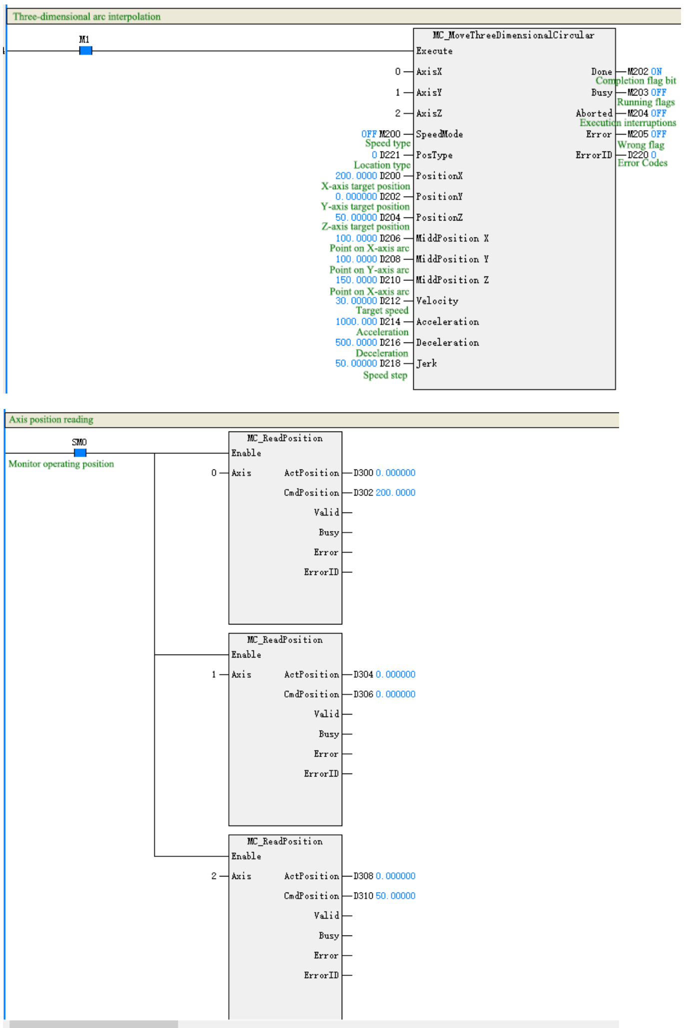
Set the 3-dimensional circular arc passing through the point (100,100,150), the target position (200,0,50), and the target



speed 30.

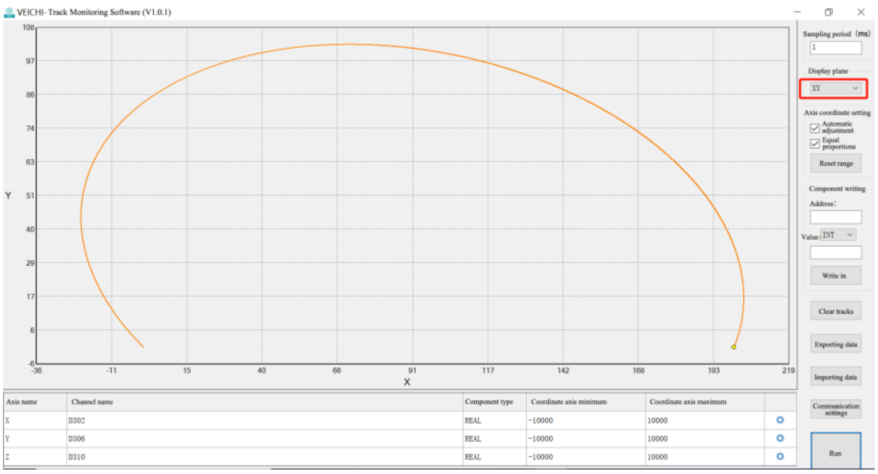




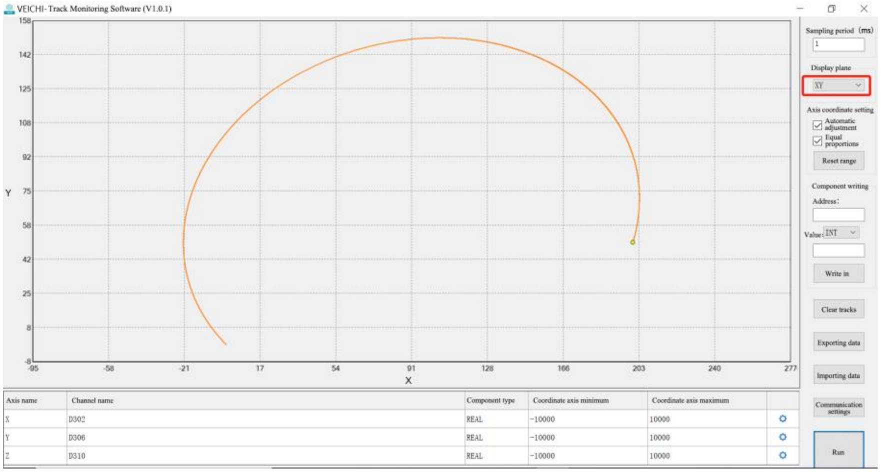


Track monitoring

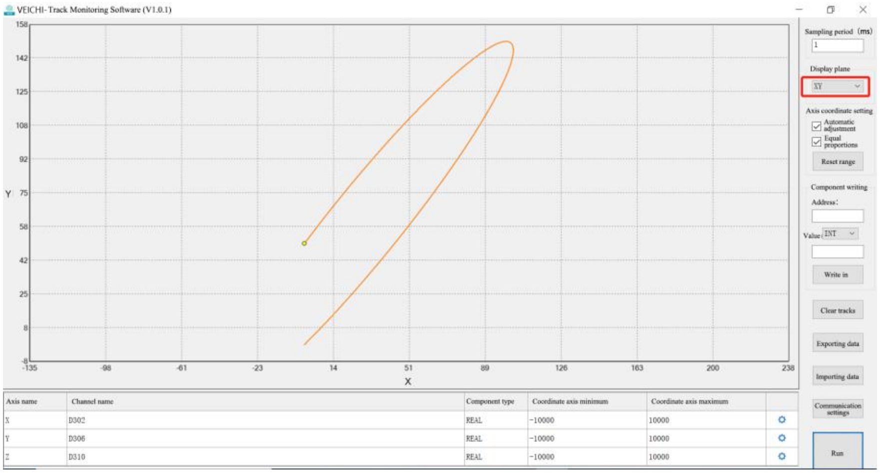
XY plane



XZ plane

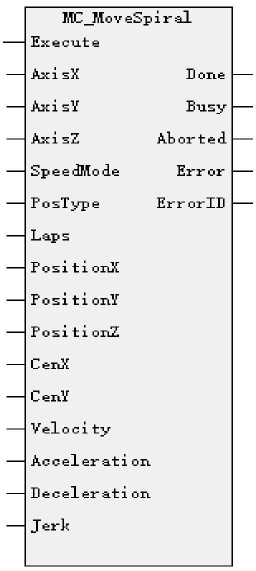


YZ plane



4.3.31 MC_MoveSpiral（Helical interpolation command）

Drawing Block



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	AxisX	WORD	Constant/D/R/W	0~63	X-axis name/axis number	Execute
IN	AxisY	WORD	Constant/D/R/W	0~63	Y-axis name/axis number	
IN	AxisZ	WORD	Constant/D/R/W	0~63(No such axis by default)	Z-axis name/axis number	
IN	SpeedMode	BOOL	M/S	TRUE,FALSE (defaultFALSE)	Speed type. 0T type, 1S type	
IN	PosType	WORD	Constant/D/R/W	0~1 (default 0)	Position Type. 0 Relative position 1 Absolute position	
IN	Laps	DWORD	Constant/D/R/W	(default 0)	Number of revolutions. Positive or negative is the direction of rotation (0 is equal to 1, such as: n = | La'p's | n is n turns or less than n turns greater than n-1 turns)	
IN	PositionX	REAL	Constant/D/R/W	Positive/Negative Number/0	X-axis target position	
IN	PositionY	REAL	Constant/D/R/W	Positive/Negative Number/0	Y-axis target position	
IN	PositionZ	REAL	Constant/D/R/W	Positive/Negative Number/0 (No such axis by default)	Z-axis target position	
IN	CenX	REAL	Constant/D/R/W	Positive/Negative Number/0	X-axis circle center position	
IN	CenY	REAL	Constant/D/R/W	Positive/Negative Number/0	Y-axis circle center position	
IN	Velocity	REAL	Constant/D/R/W	Positive Number	Target speed	
IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	
IN	Deceleration	REAL	Constant/D/R/W	Positive Number (Default Acceleration)	Deceleration	
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed step	
OUT	Done	BOOL	M/S	TRUE, FALSE	Finish Stop	
OUT	Busy	BOOL	M/S	TRUE, FALSE	Running flag	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interruptions	

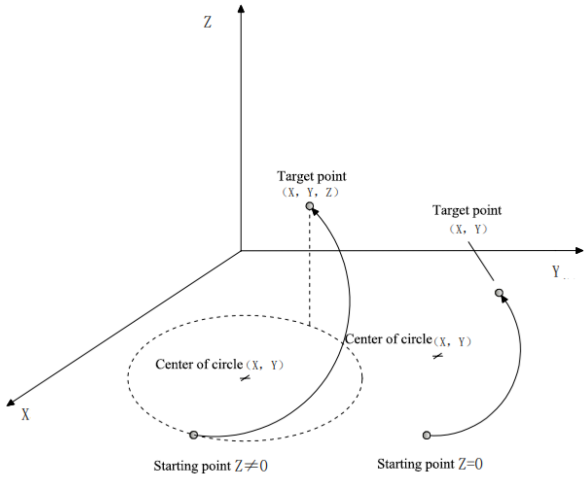
OUT	Error	BOOL	M/S	TRUE, FALSE	Wrong sign	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

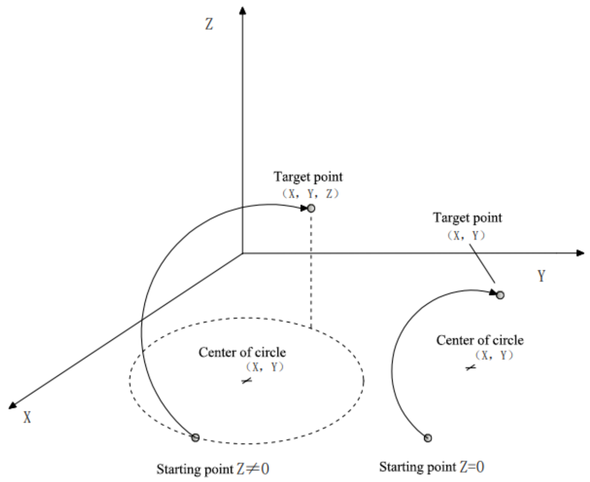
The MC_MoveSpiral command is used for the spiral interpolation function and is valid on the rising edge.

This function is a two-axis circular arc interpolation, and then add a vertical two-axis circular arc plane height movement, and the three axis is the same motion running, its for the extension of the circular arc interpolation function. When the spiral interpolation command is executed and the height value change is zero, it is the circular arc function.

●The following figure shows the spiral interpolation circle center in the XY plane, if the Z axis no change amount (Z = 0), the spiral interpolation running with XY axis two-axis circular arc interpolation.

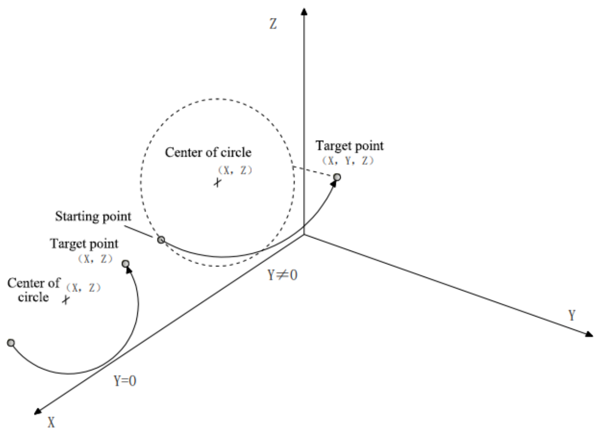


Counterclockwise (Laps are negative)

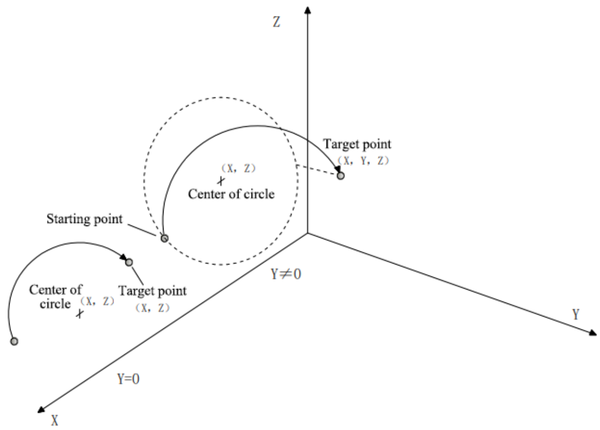


Clockwise (Laps are positive)

●The following figure shows the spiral interpolation in the XZ plane, if there is no change in the Y axis (Y = 0), the spiral interpolation operation with the XZ axis two-axis circular arc interpolation

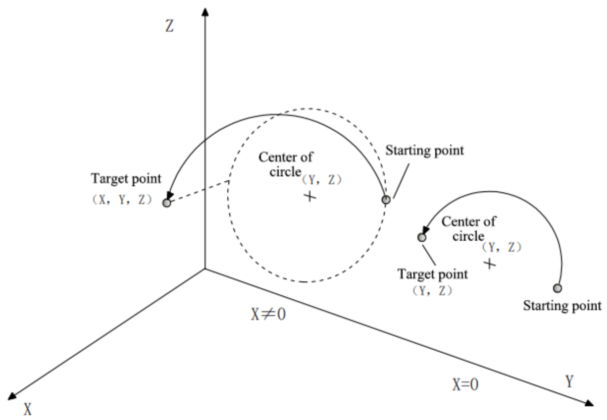


Counterclockwise (Laps are negative)

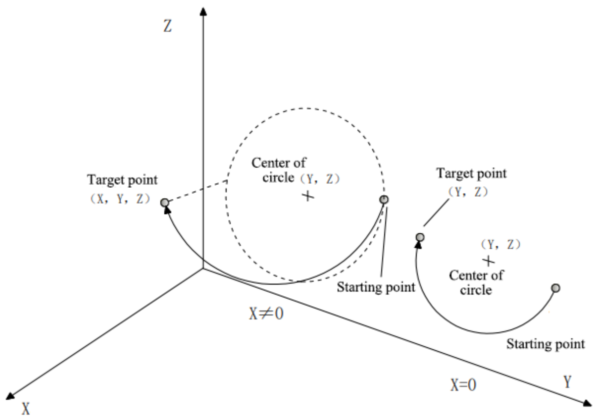


Clockwise (Laps are positive)

●The following figure shows the spiral interpolation in the YZ plane, if there is no change in the X-axis (Y = 0), the spiral interpolation operation with YZ axis two-axis circular interpolation



Counterclockwise (Laps are negative)

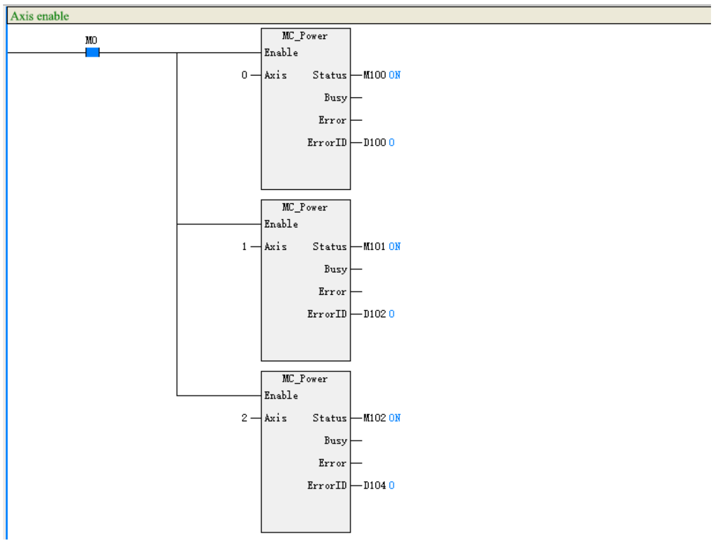


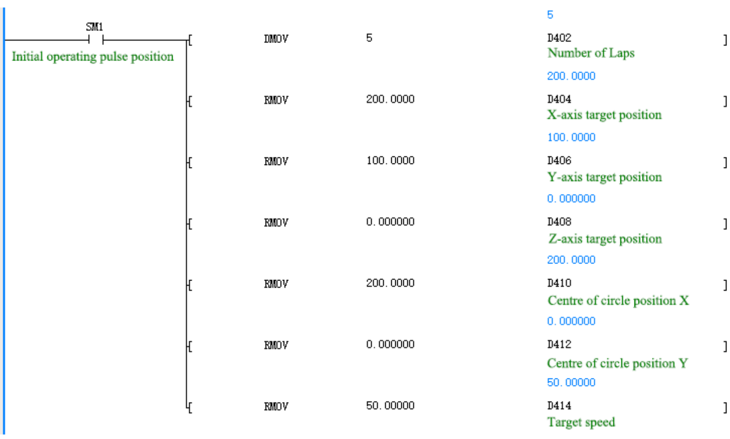
Clockwise (Laps are positive)

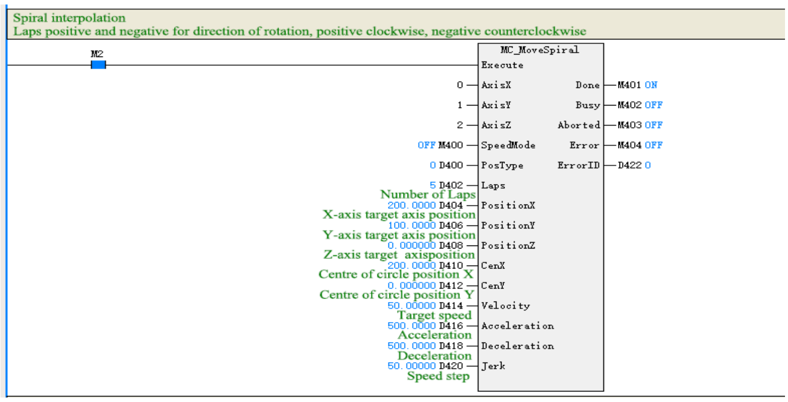
Program example 1

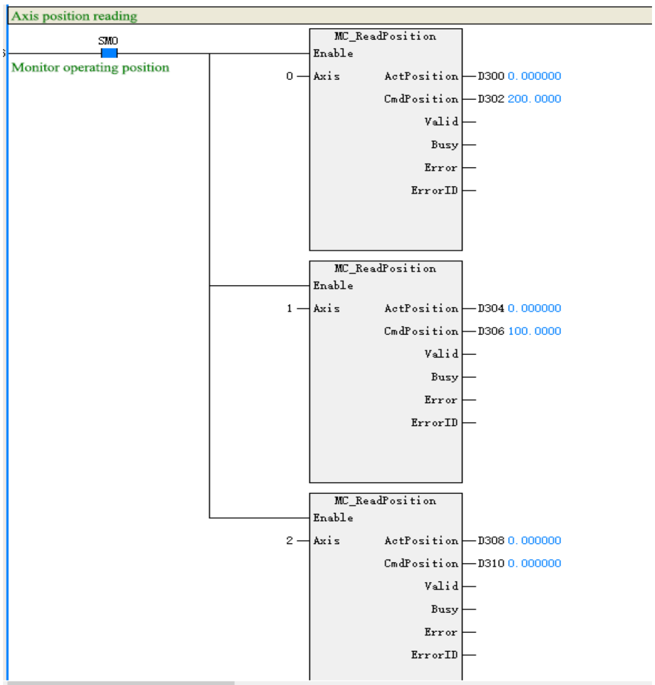
Plane spiral Z = 0 - the radius of the starting point and the end point to reach the center of the circle is not the same

Only set X axis, Y axis, the starting point (0,0), set the center of the circle (200,0), the target position (200,100), the number of circles 5, the starting point and the end point to reach the center of the circle does not coincide with the radius, then the execution of 5 circles of the plane spiral arc, the final arrival position (200,100)

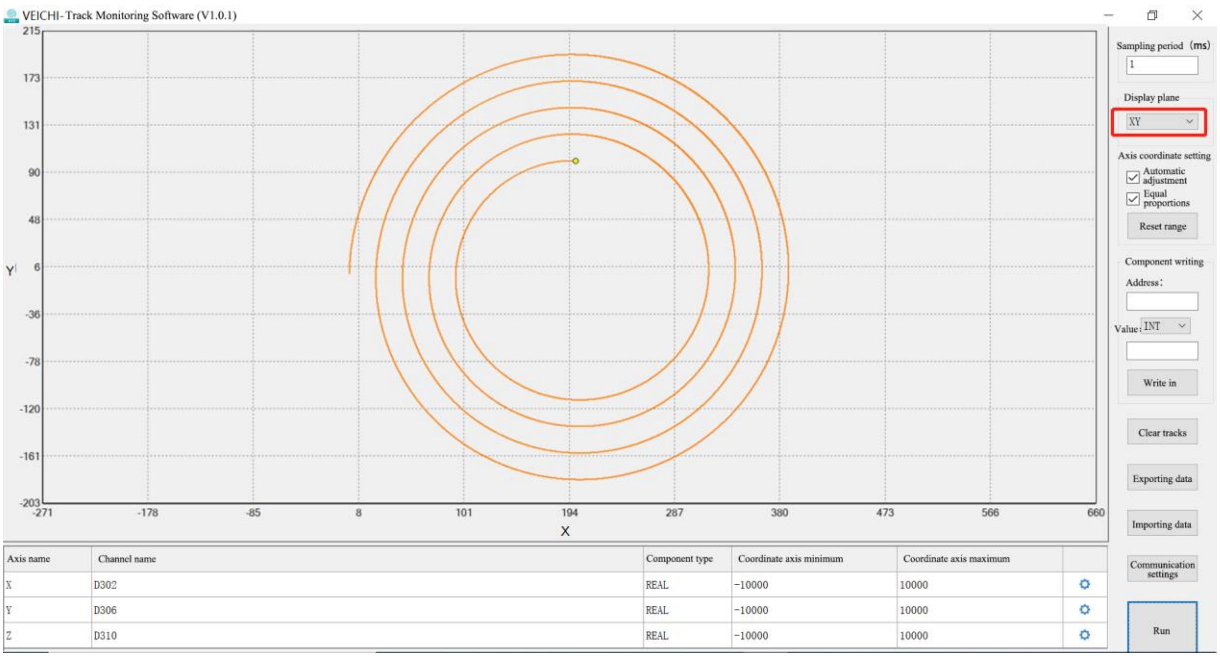








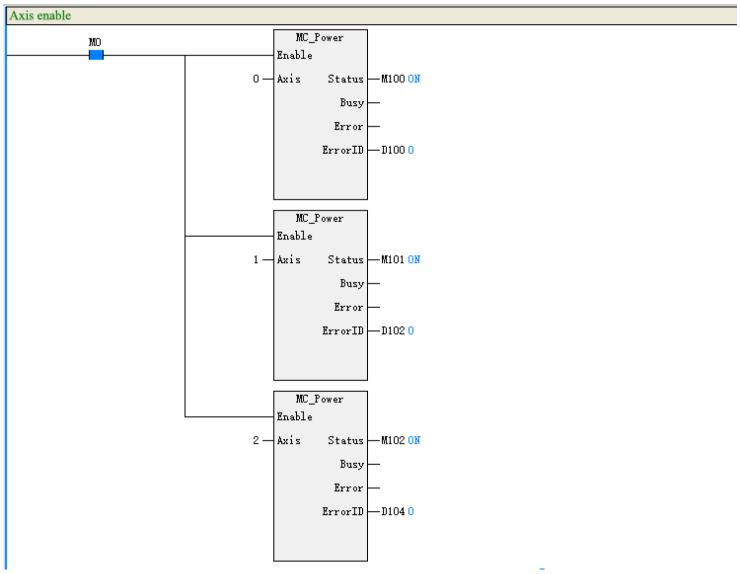
Curve Monitoring

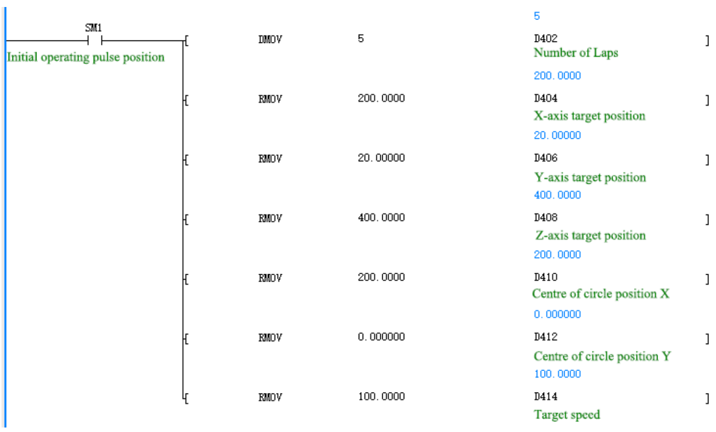


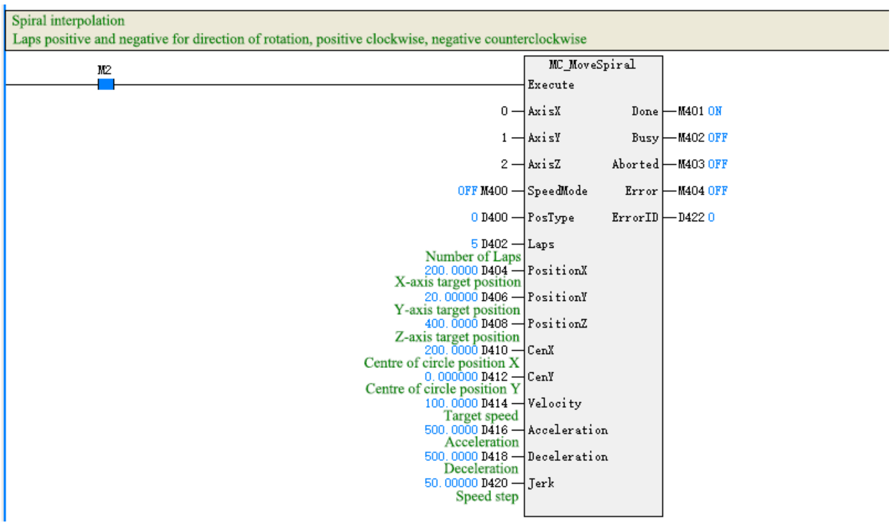
Program example 2

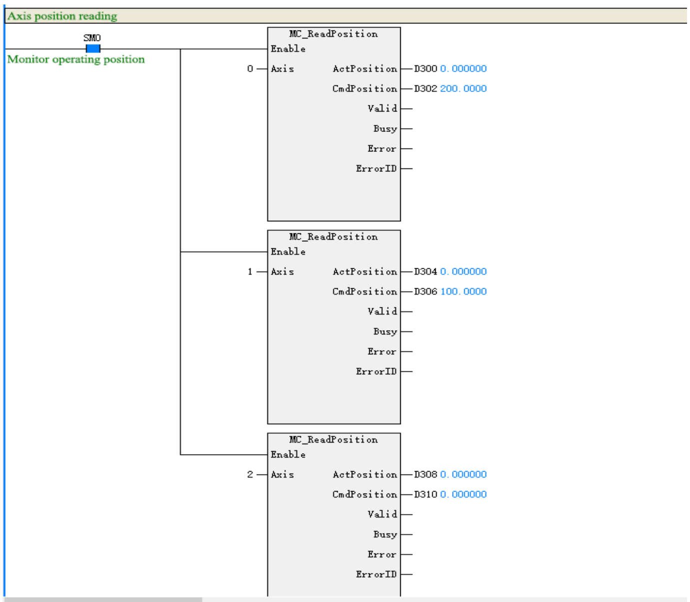
Reducing spiral Z ≠ 0 - The radius of the starting point and the end point to reach the center of the circle are not the same, and the 3D effect is conical.

The program starts with setting the relative coordinates, in XY plane mode, clockwise spiral, the starting position of the spiral is (0, 0, 0), the ending position is (200, 20, 400), the center of the spiral circle is (200, 00, 0), and the target speed is 100 Unit/s for 5 revolutions.



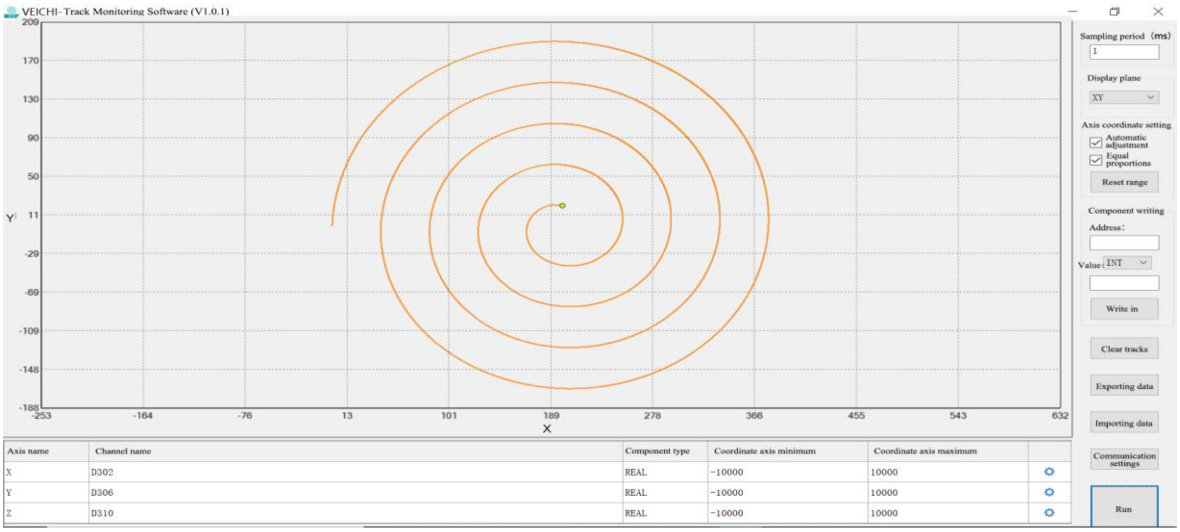




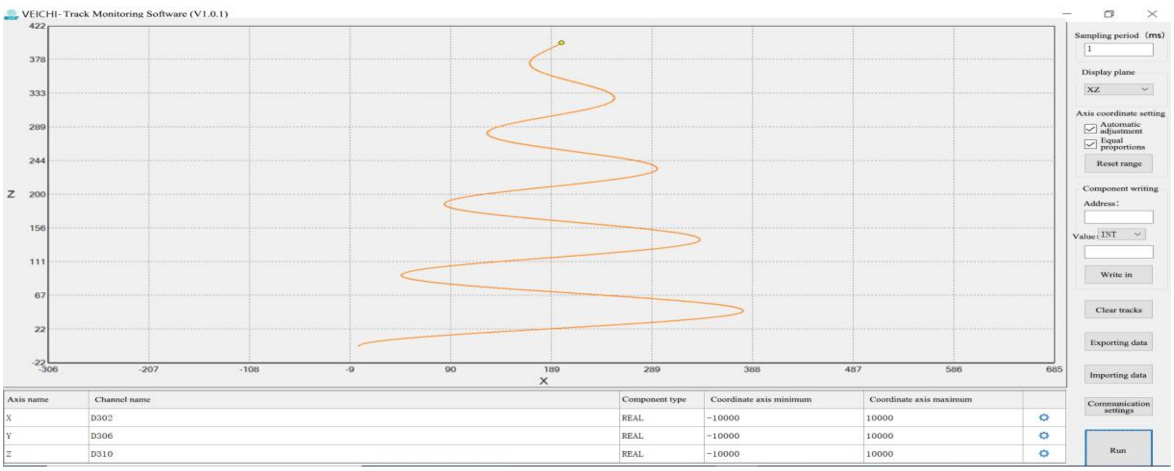


The track monitoring is as follows:

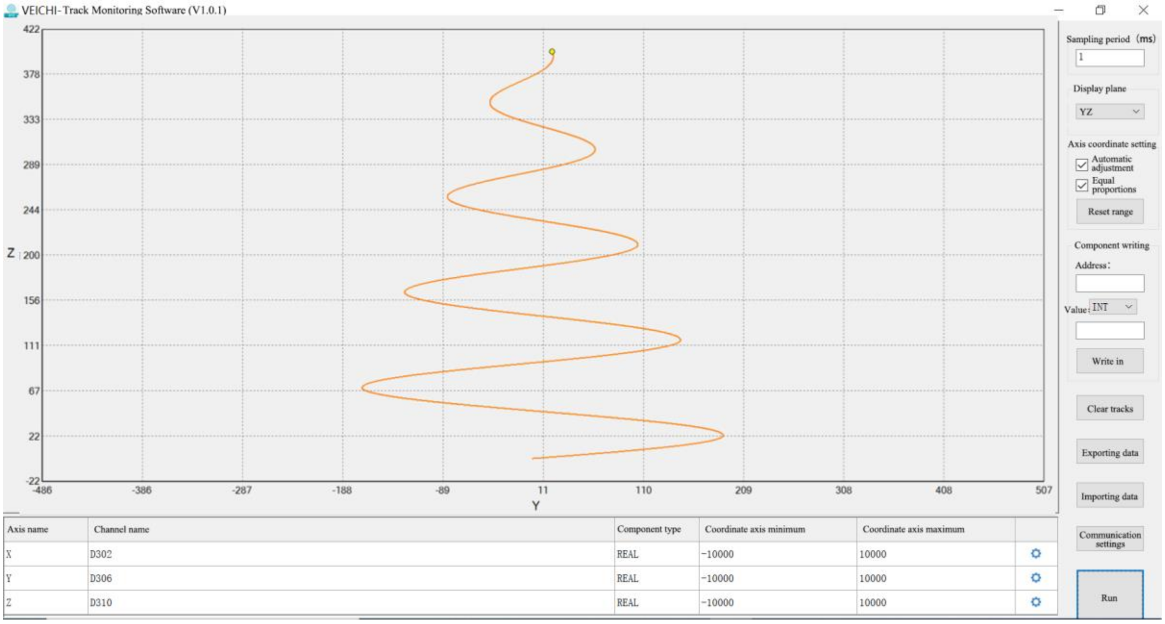
XY plane：



XZ plane：



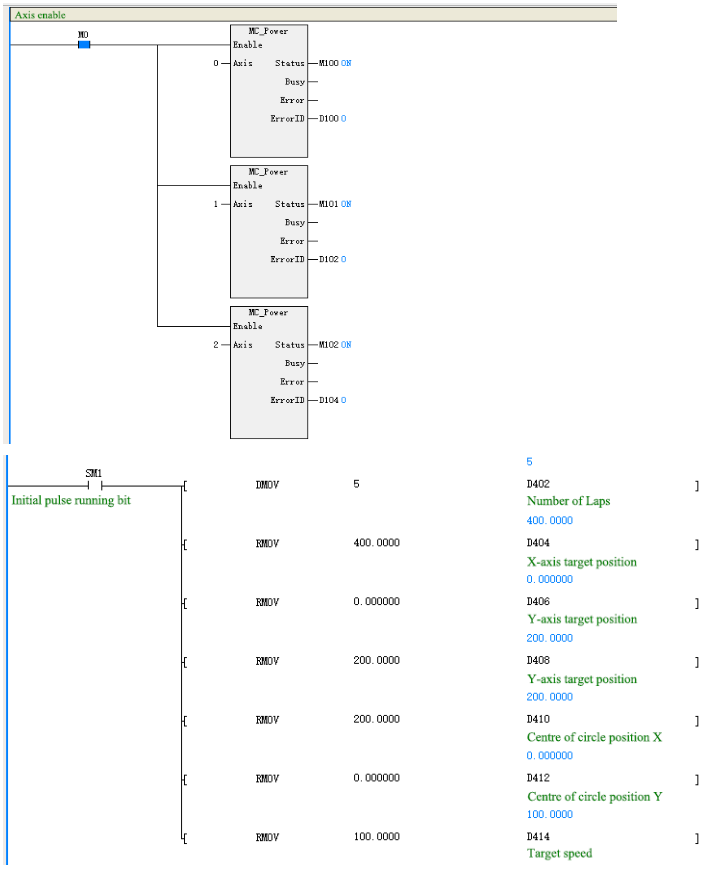
YZ plane：

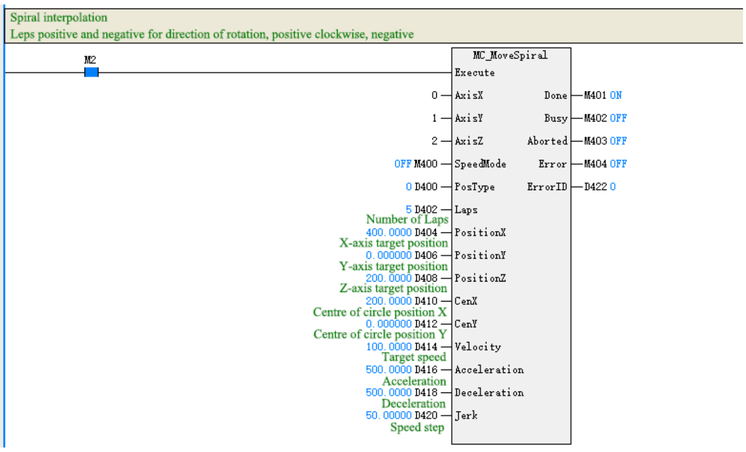


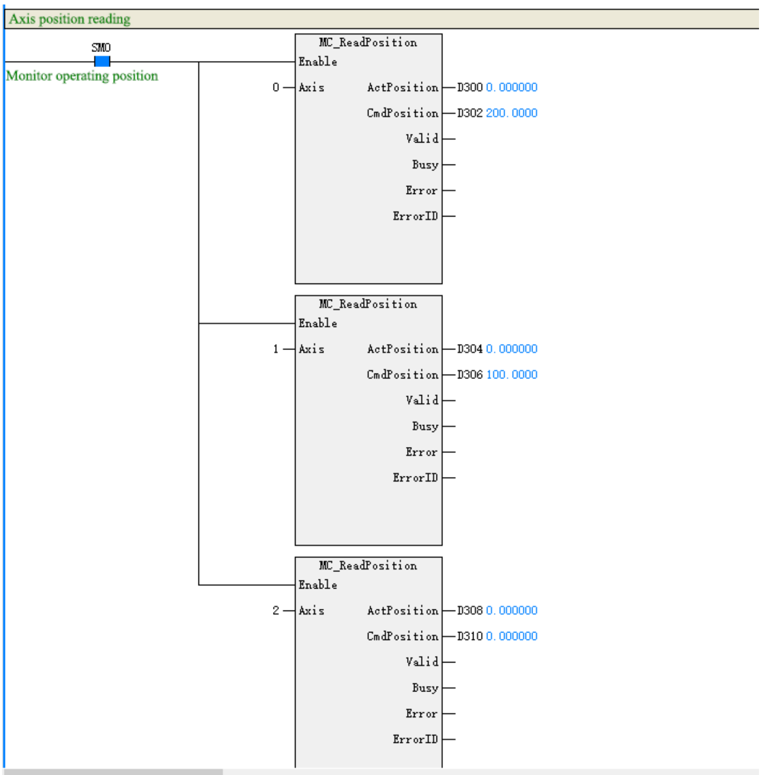
Program example 3

Non-variable spiral Z ≠ 0 - the radius of the starting point and the end point to the center of the circle is the same, and the 3D effect is cylindrical.

The program starts with setting the relative coordinates, in XY plane mode, clockwise spiral, the starting position of the spiral is (0, 0, 0), the ending position is (400, 0, 200), the center of the spiral circle is (200, 0, 0), and it moves 5 revolutions at the target speed of 100 Unit/s.

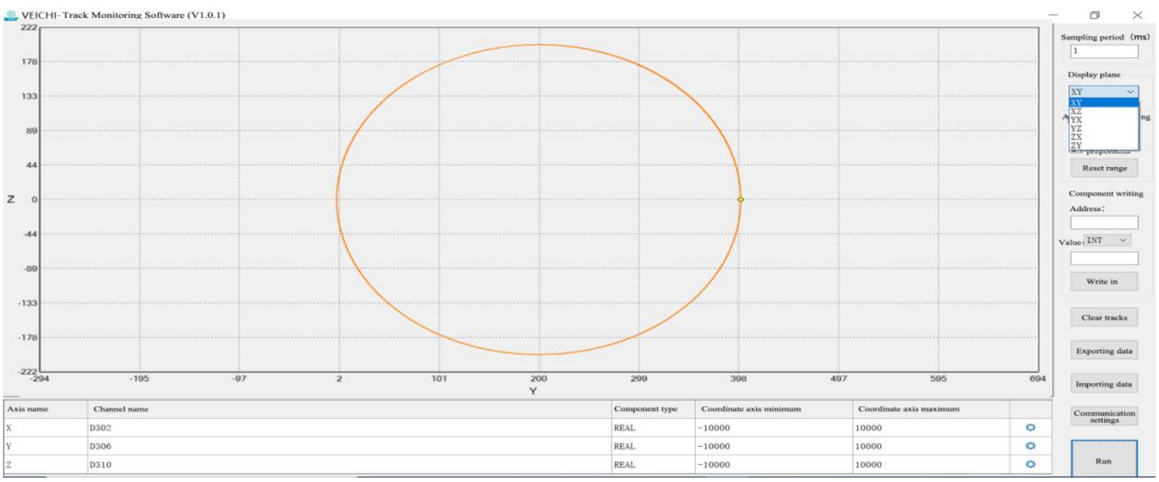




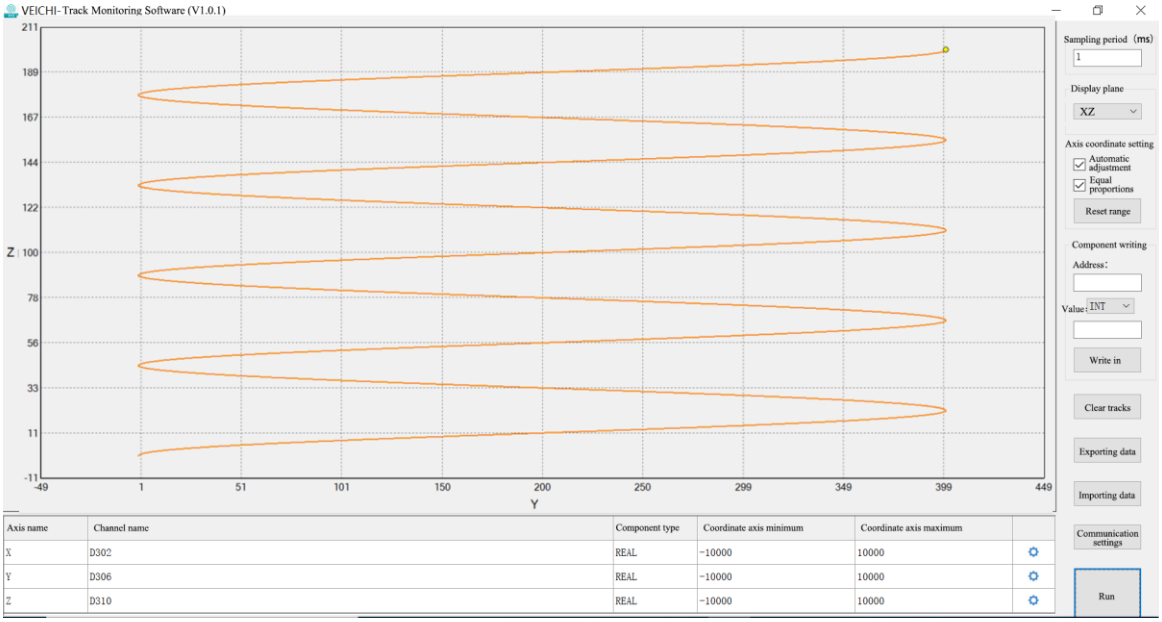


The track monitoring is as follows:

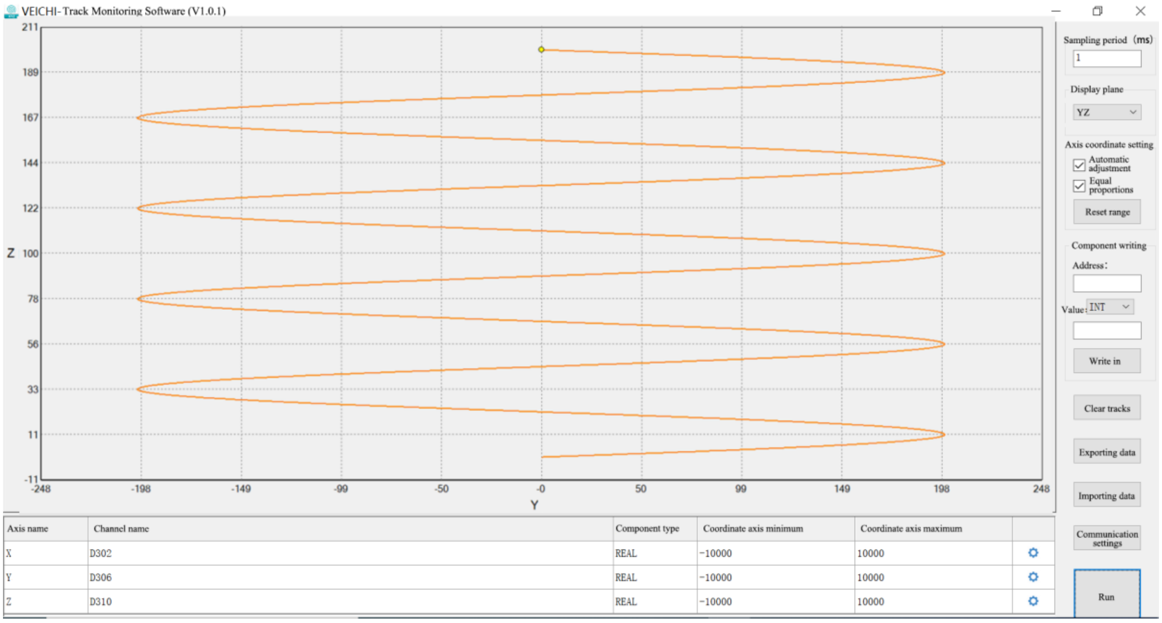
XY plane - 5 circles of non-reducer arc：



XZ plane：

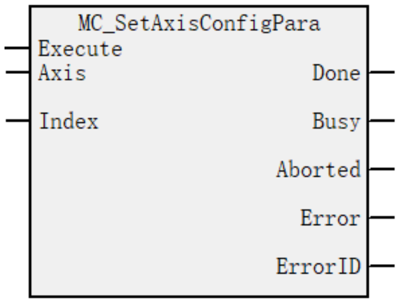


YZ plane：



4.3.32 MC_SetAxisConfigPara（Axis parameter setting commands）

Drawing Block



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	Axis	WORD	Constant /D/R/W/Var	0~63	Axis number	Execute
IN	Index	WORD	Constant /D/R/W/Var	1~9	Parameter serial number (reference axis system variable McAxis axis operating status group)	
OUT	Done	BOOL	M/S/Var	TRUE ,FALSE	Finish stop	
OUT	Busy	BOOL	M/S/Var	TRUE, FALSE	Running flag	
OUT	Aborted	BOOL	M/S/Var	TRUE, FALSE	Execution interrupted	
OUT	Error	BOOL	M/S/Var	TRUE, FALSE	Error flags	
OUT	ErrorID	WORD	Constant /D/R/W/Var	0x0~0xffff	Error ID	

Description of functions and commands

MC_SetAxisConfigPara this command is used to set the axis parameters and is valid on the rising edge

The user can modify the configuration parameters of individual axes through the PLC program to meet the needs of different applications, such as software limits, the amount of movement for one revolution of the table, etc., as described in ***section 4.1.5 Axis system variables.***

● To modify the axis configuration parameters in the PLC program, you need to use the axis configuration related system structure variable _SYS_MC_AXIS to set the parameters, note that this structure is not saved when power is lost

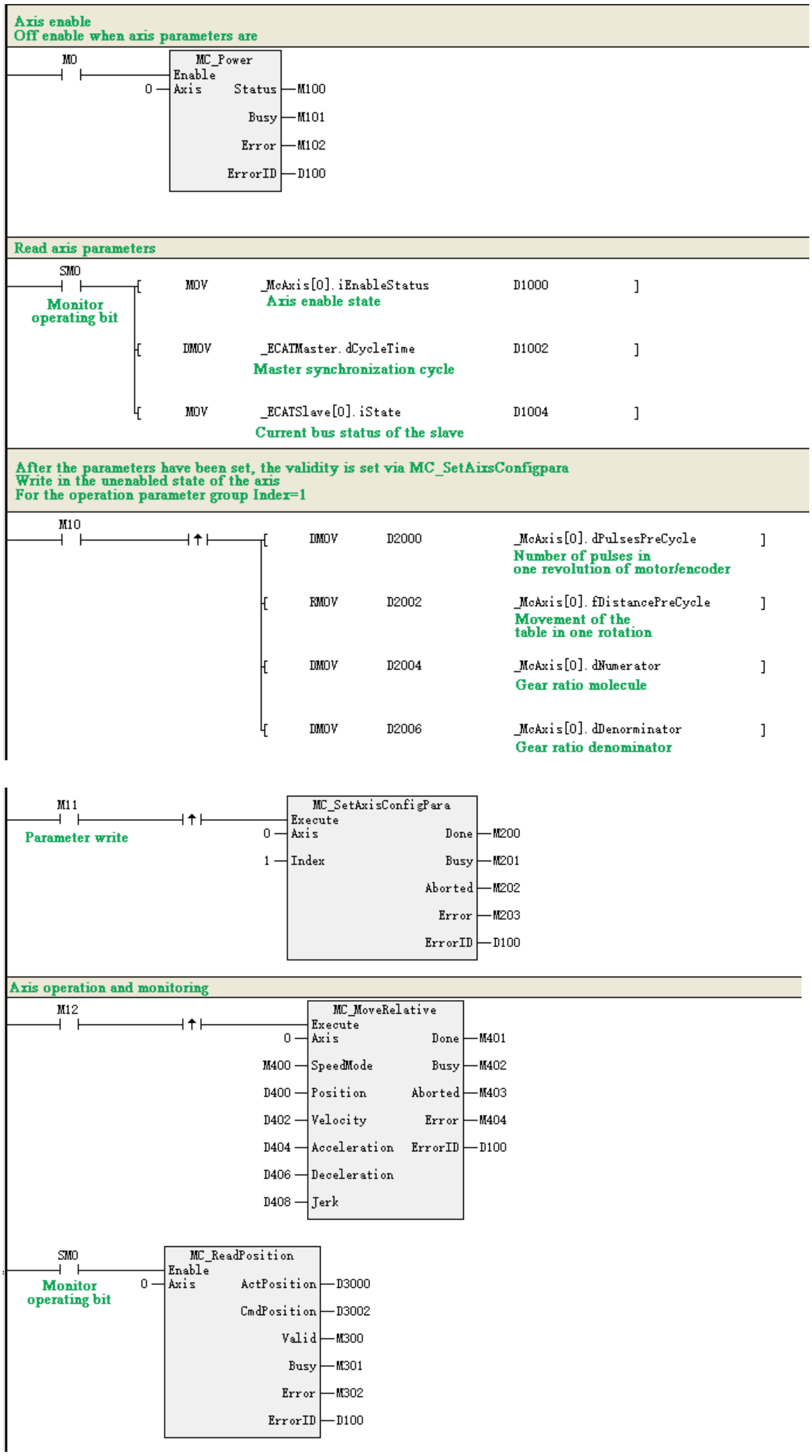
● When the PLC is powered up, this structure variable is initialised according to the axis configuration in the software backend.

● Modify the initialised variable values according to the application requirements and call the MC_SetAxisConfigPara command to check the settings to make them effective.

Example of program:

The following routines are used to modify the axis parameters, the number of pulses in one revolution of the motor/encoder, the amount of movement in one revolution of the table and the electronic gear ratio.

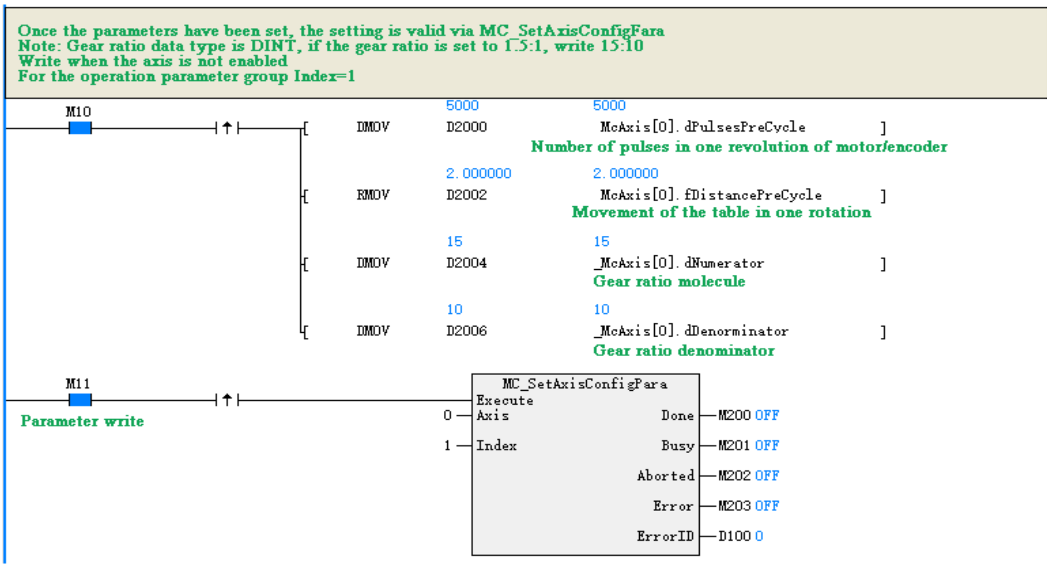
Note: Gear ratio setting, note the data type DINT, for example if you need to set the gear ratio to 1.5:1, you can set the parameter 15:10, or 3:1 or 30,000:10,000



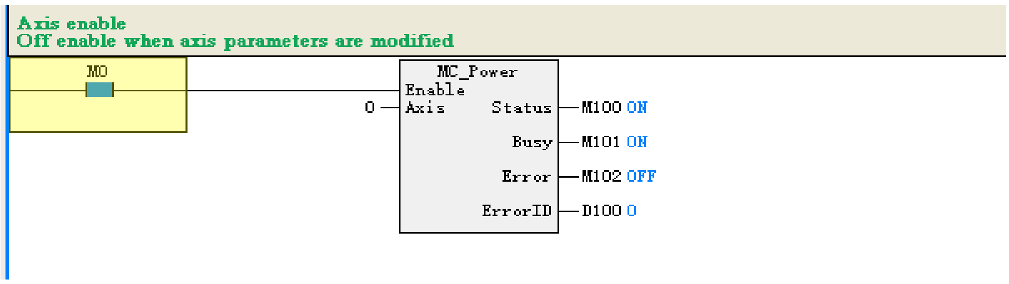
Program execution:

1. After setting M10 to ON, assign the new axis parameters to the system variables；

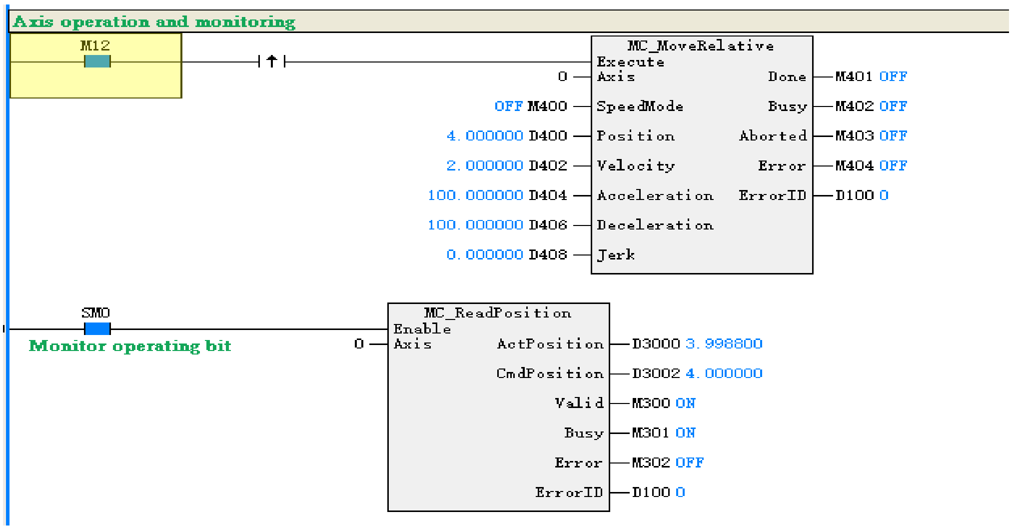
2. After setting M11 to ON, the assigned system variable is made valid by the parameter setting instruction；



3. After setting M0 to ON, the axis is enabled；

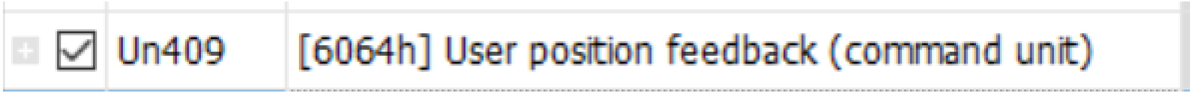


4. After setting M12 to ON, the axis runs according to the written parameters



The position monitoring of the servo software is as follows:

The number of pulses for one revolution of the motor is 5000, the movement is 2Unit, the gear ratio is 15:10 (1.5:1) and the target position is set to 4Unit, the servo monitoring is 15000.

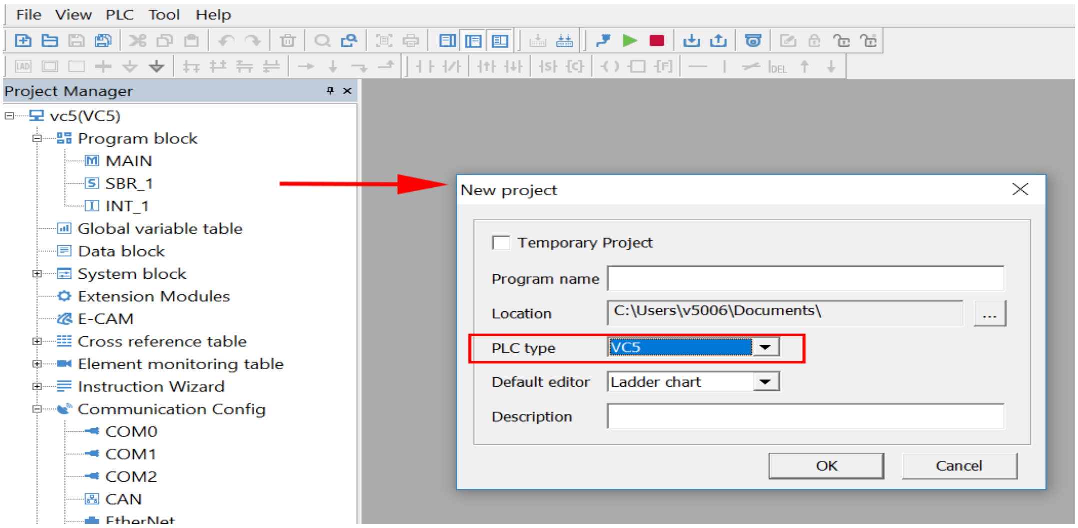


4.4. EtherCAT Motion Control Axis Settings

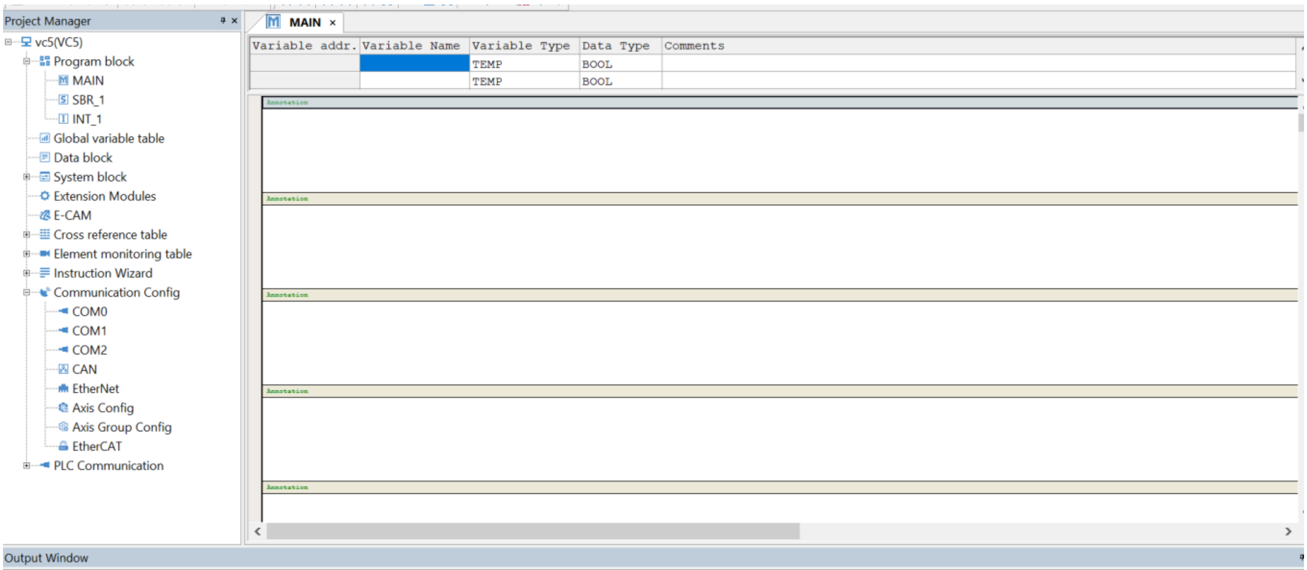
4.4.1 New project

To control the axis correctly, first create a configuration according to the needs of the project, and then set the relevant parameters according to the working conditions, download the project through online debugging for simple operation, judge whether the parameter setting is reasonable, whether the hardware connection is reliable, and finally write a PLC program to complete the overall control logic function. Here are some examples. This routine will create a new bus servo axis and a local pulse axis, and achieve simple motion through the online debugging interface and commands.

1. Open Auto Studio software, build a new project, and select VC5 for PLC type.



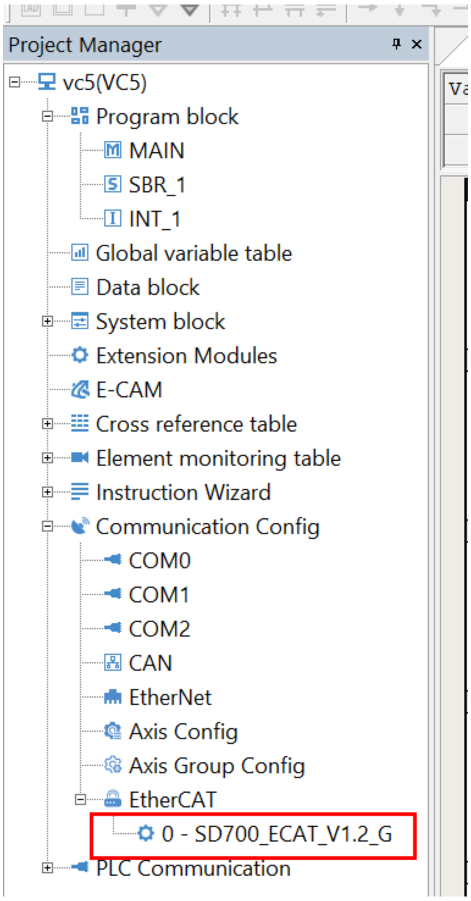
2. After the project is created, enter the program editing interface.



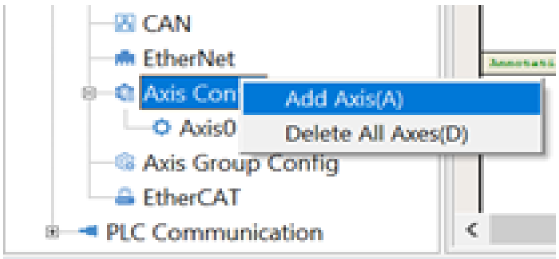
4.4.2 Create a project

For motor movement, it is necessary to configure a servo drive and a bus servo axis and link the two together.

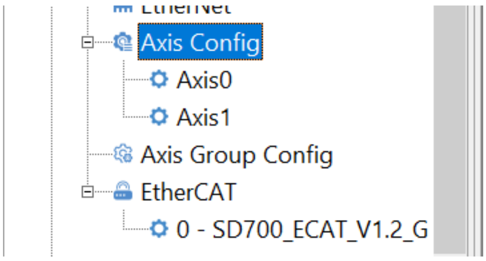
1. Add the device, if not, you need to import the XML file of the device；



2.You can add a motion control axis by selecting Axis Configuration in the device tree, right-click the mouse to select Add Axis, and repeat to create a second axis；

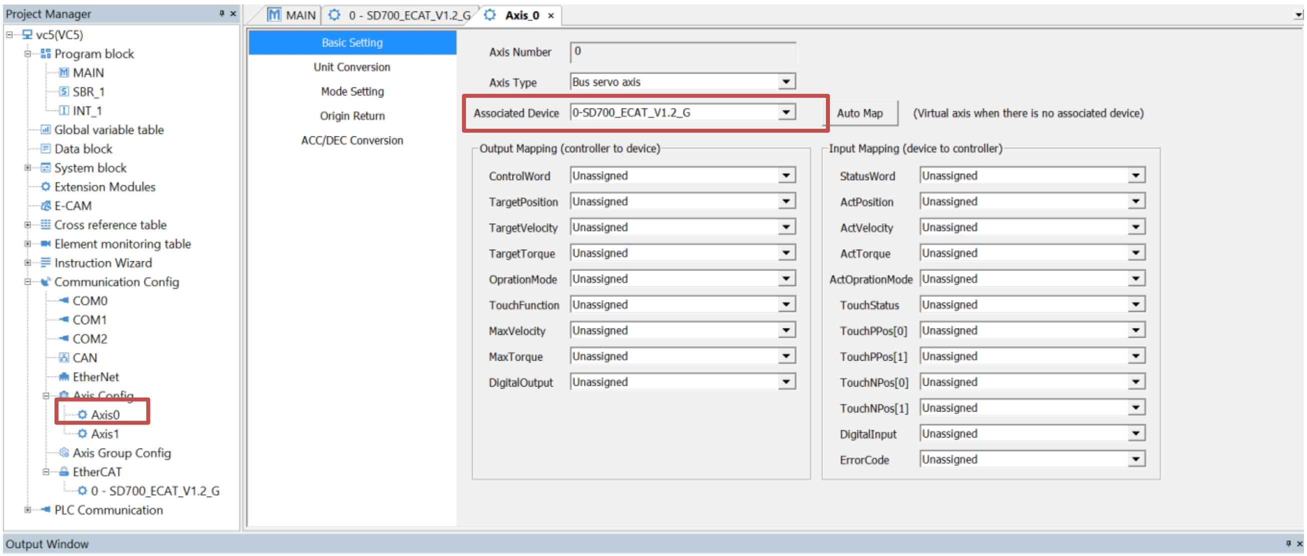


After adding the axis

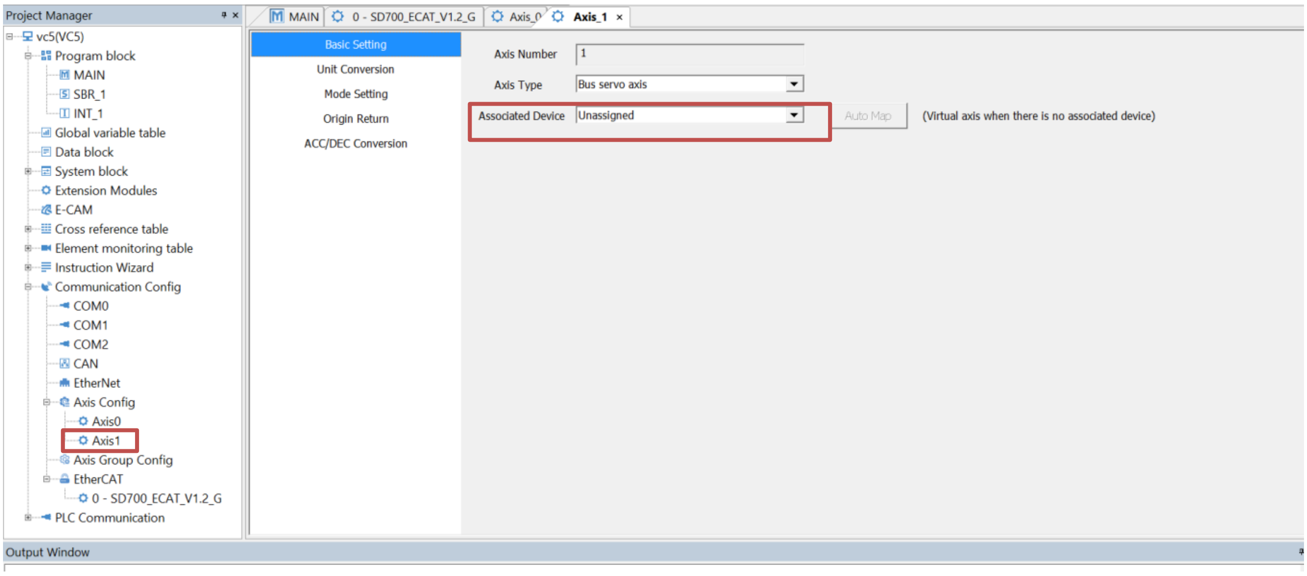


3.Set the first axis as the bus servo axis, and associate the SD700; Set the second axis as the local pulse axis, and associate the Y0/Y1 channels;

Note: There is no need to add an XML file for the local pulse.



Bus servo axis



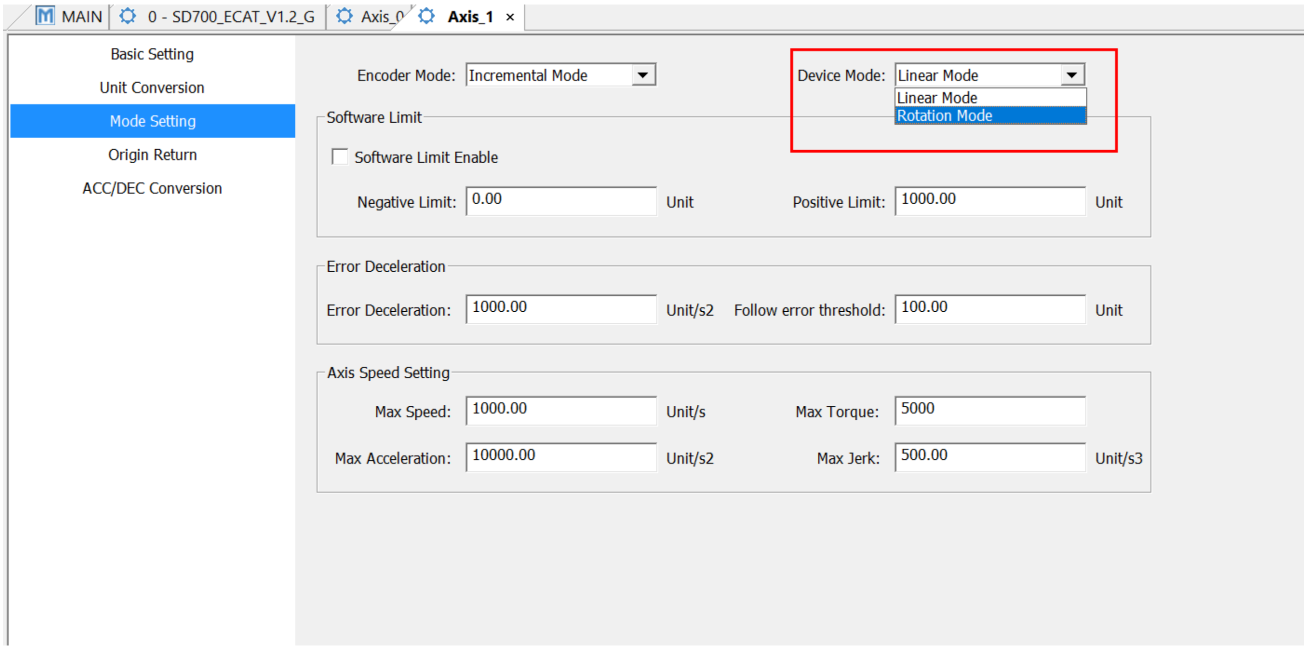
Local output shaft

4.4.3 Axis parameter configuration

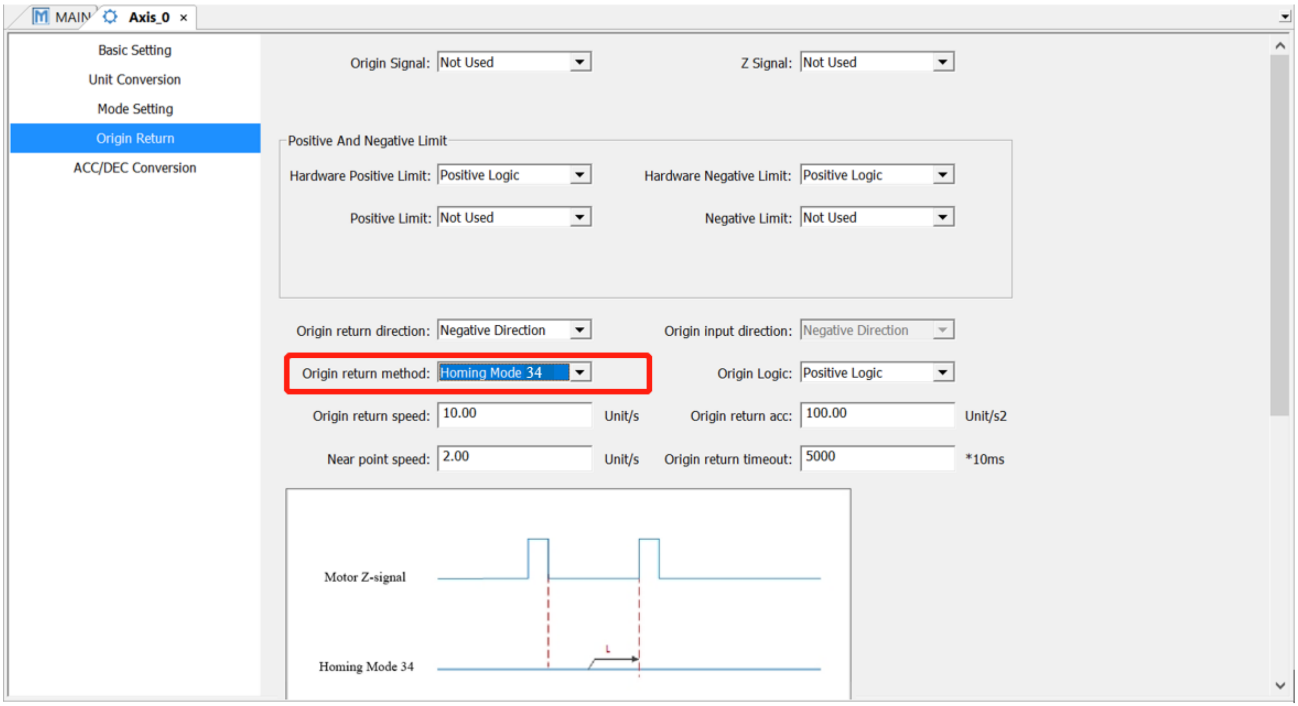
4.4.3.1 Bus servo axis

Set the relevant parameters of the axis according to the actual working conditions and requirements, and set them as follows in this routine:

1. Set to straight line mode



2. Set the origin regression method to 34.



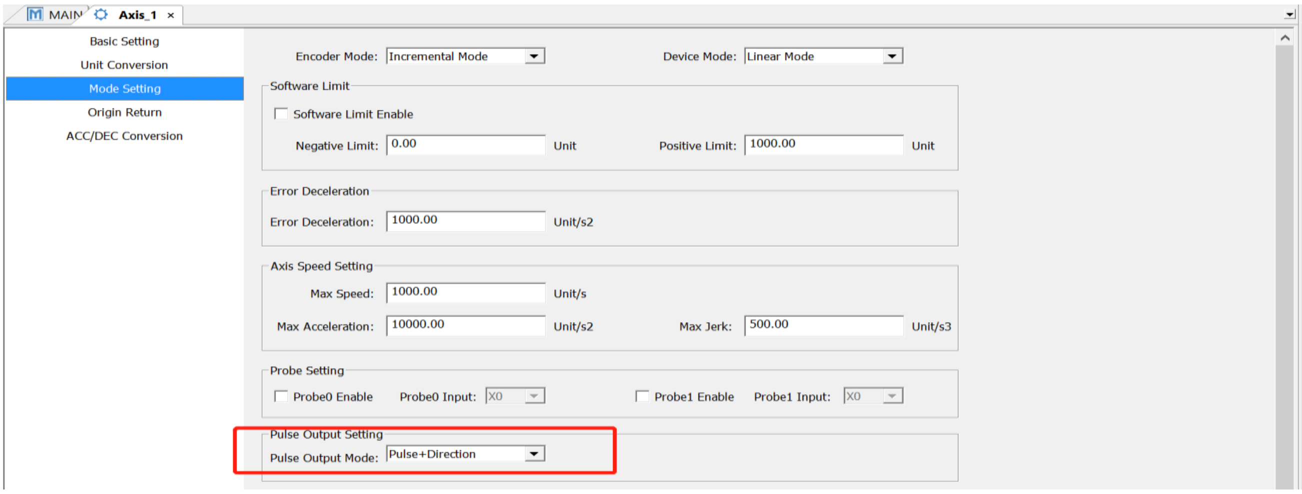
4.4.3.2 Local pulse axis

Set the relevant parameters of the shaft according to the actual working conditions and requirements, the local pulse axis uses the Waitron SD700 driver, and the driver and the host computer are set as follows in this routine:

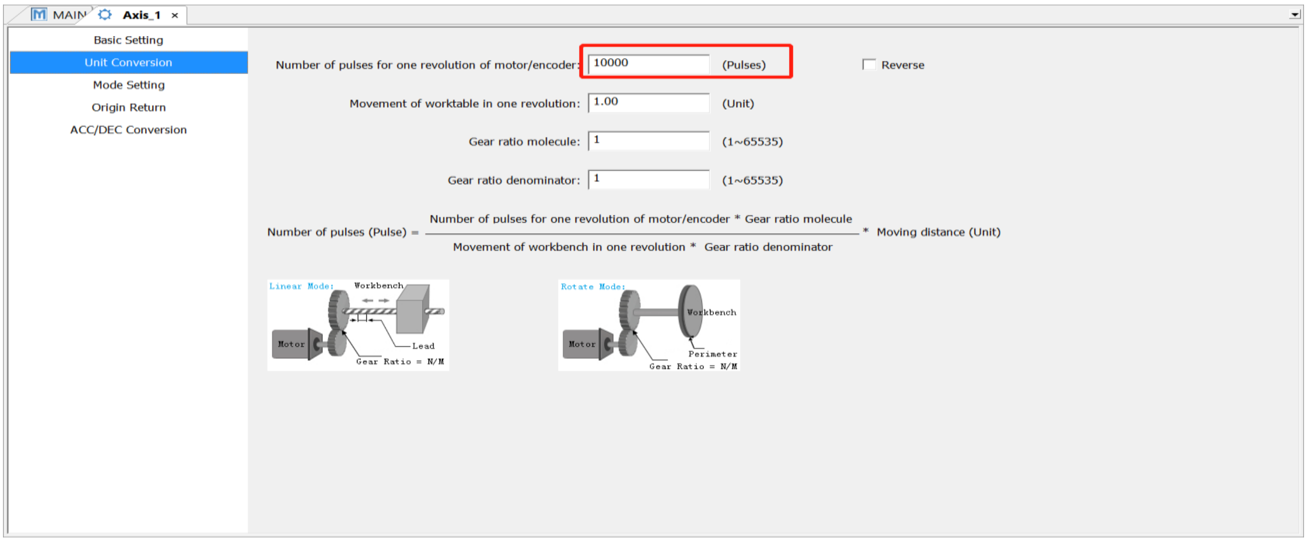
1. Internal settings of the servo drive

Function code	The parameter name	Description of the feature
Pn208=0	Internal position directives	The position command selects an external pulse input
Pn001=1	Internal enablement	Enable

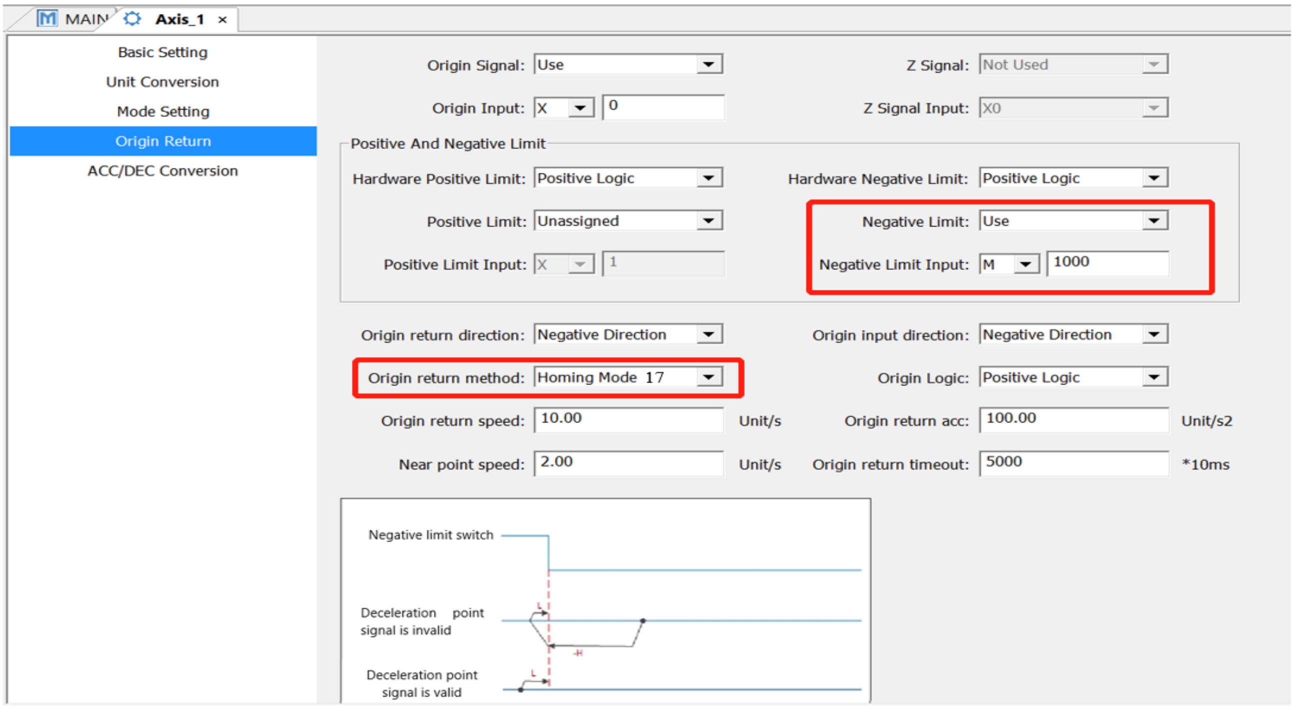
2. The pulse output mode is pulse + direction.



3. Change the "Number of command pulses for a rotation of the motor/encoder" to 10000.

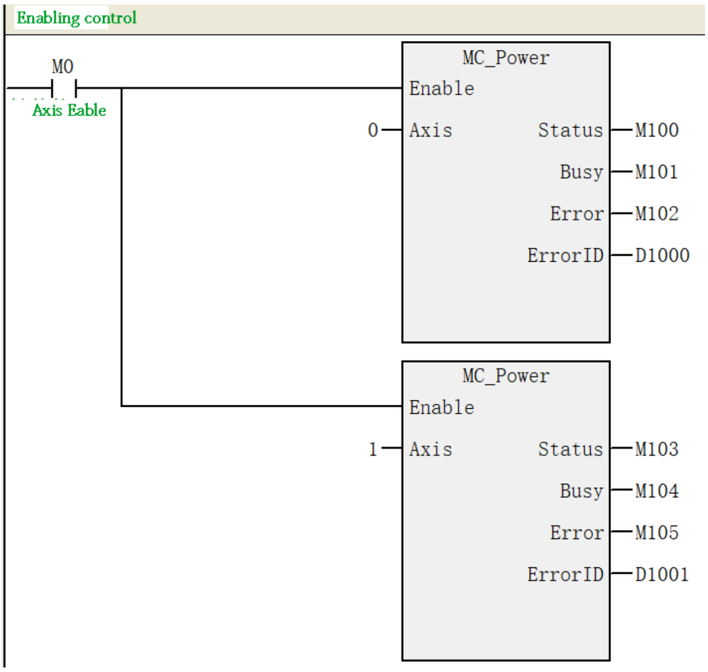


4. Modify the origin regression method to 17, and set the negative limit to M1000.

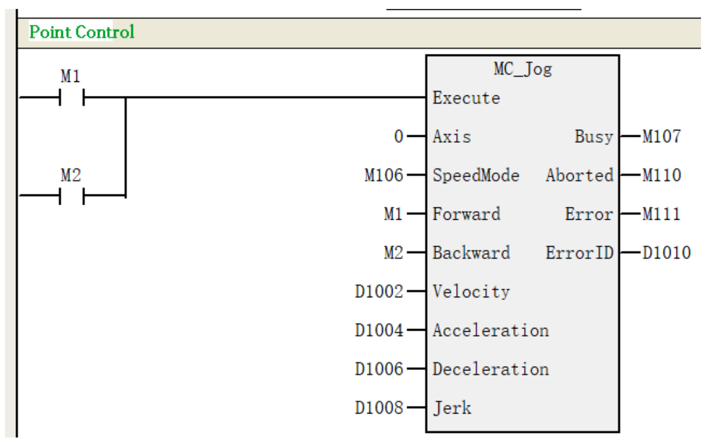


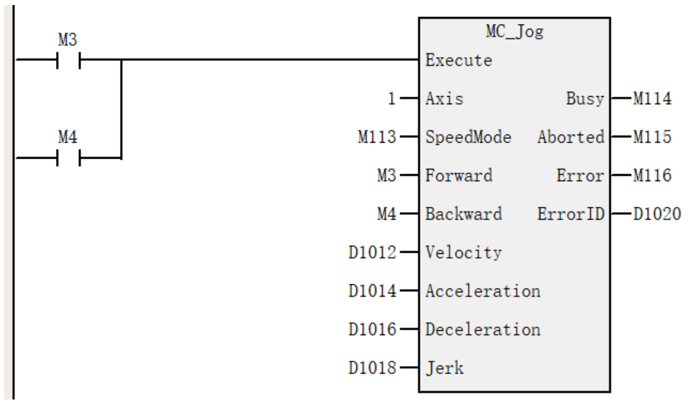
4.4.4 Program writing

1. The MC_Power function block is used to control the enablement of the axis.

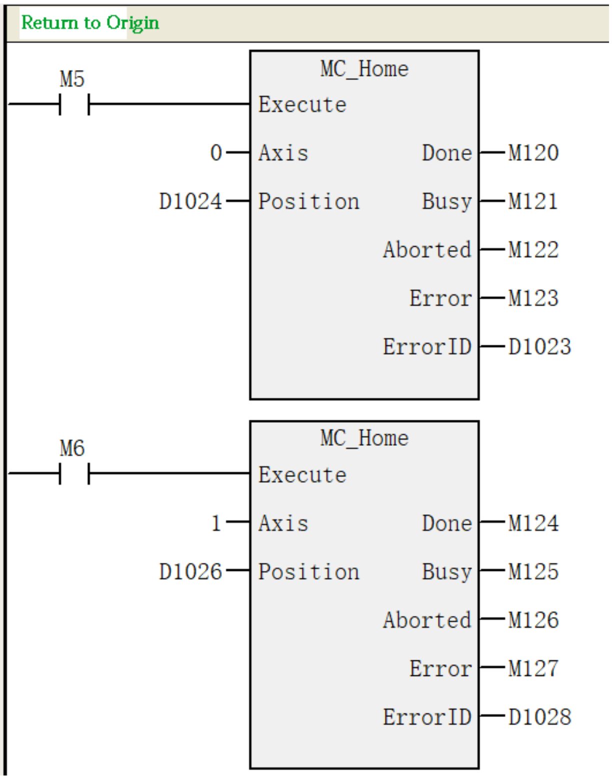


2. Call MC_Jog command test.

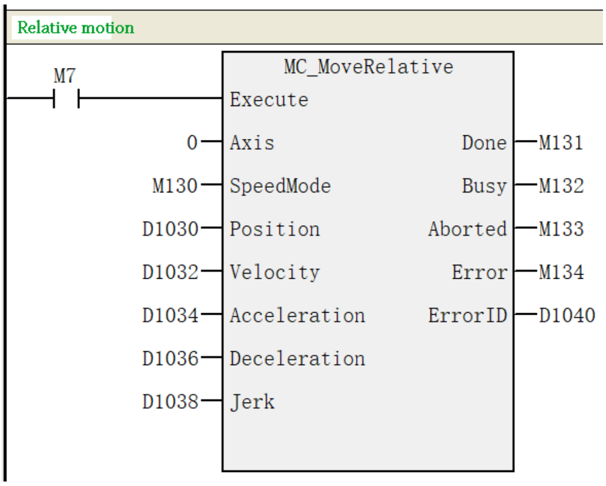


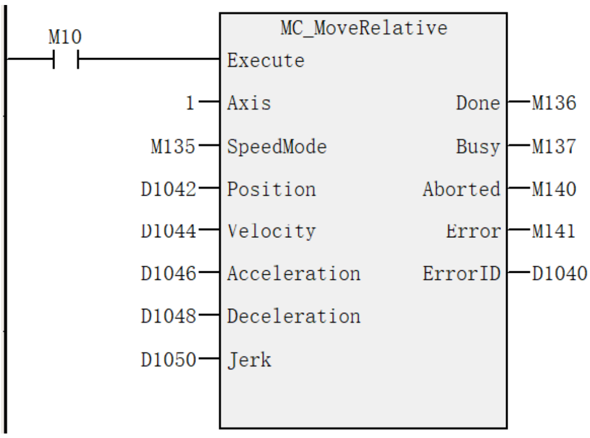


3. Call MC_Home command test.

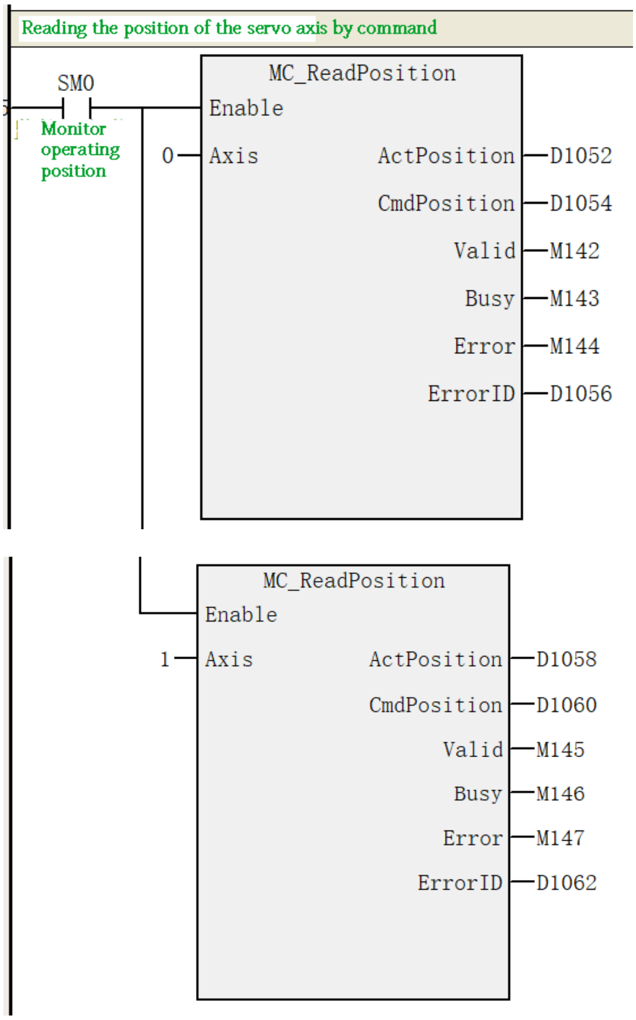


4. Call MC_Move Relative relative motion command to test point motion.





5. Shaft position monitoring, in PLC programs can be monitored by function blocks or axis system variables.

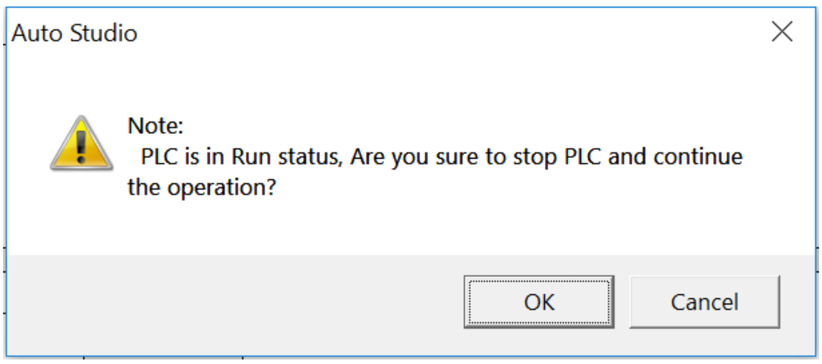


4.4.5 Project download

After completing the programming and engineering settings, perform the download operation as follows:

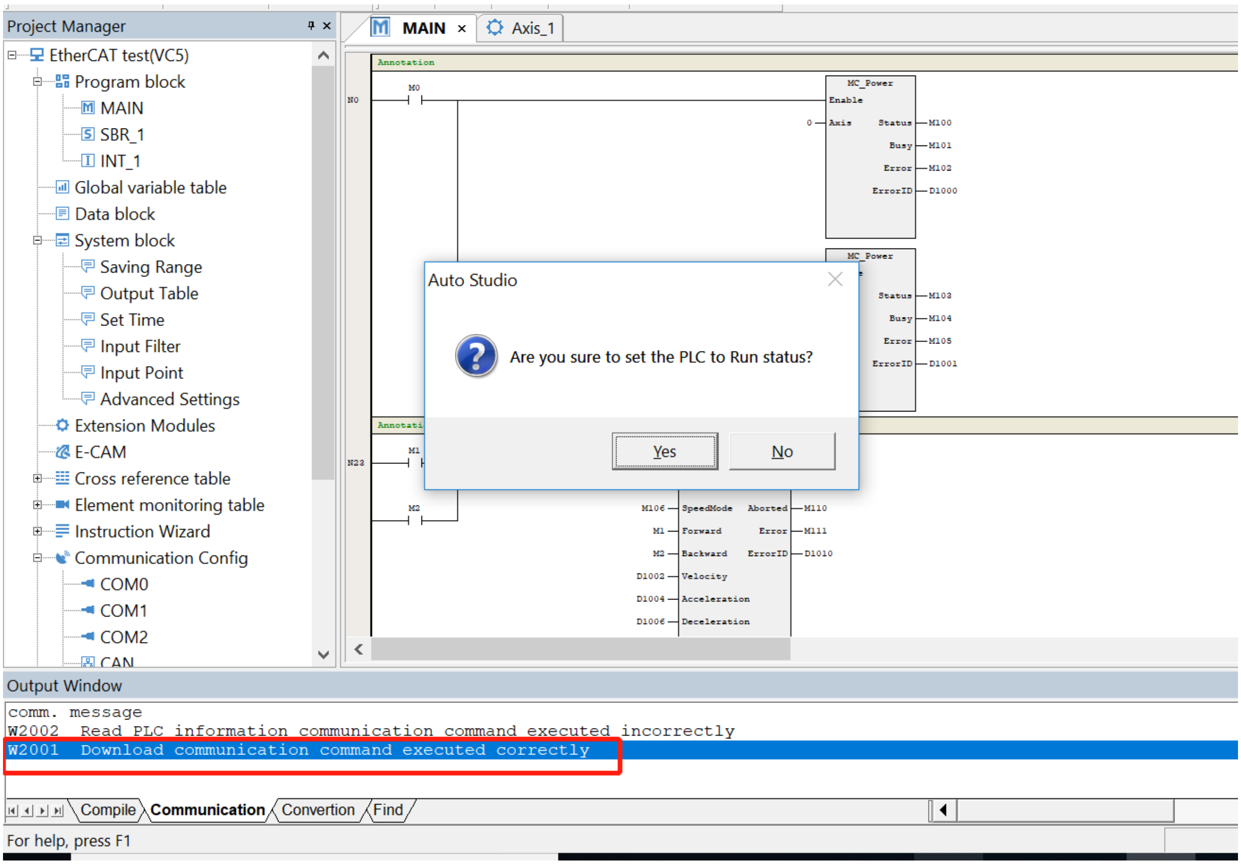
1. Click the download button and the compilation operation will be performed first.

2. After compilation, if the PLC is in a running state, the following dialog box pops up, select "OK" to go to step 3, if the PLC is in a stopped state, then download it directly, go to step 4.

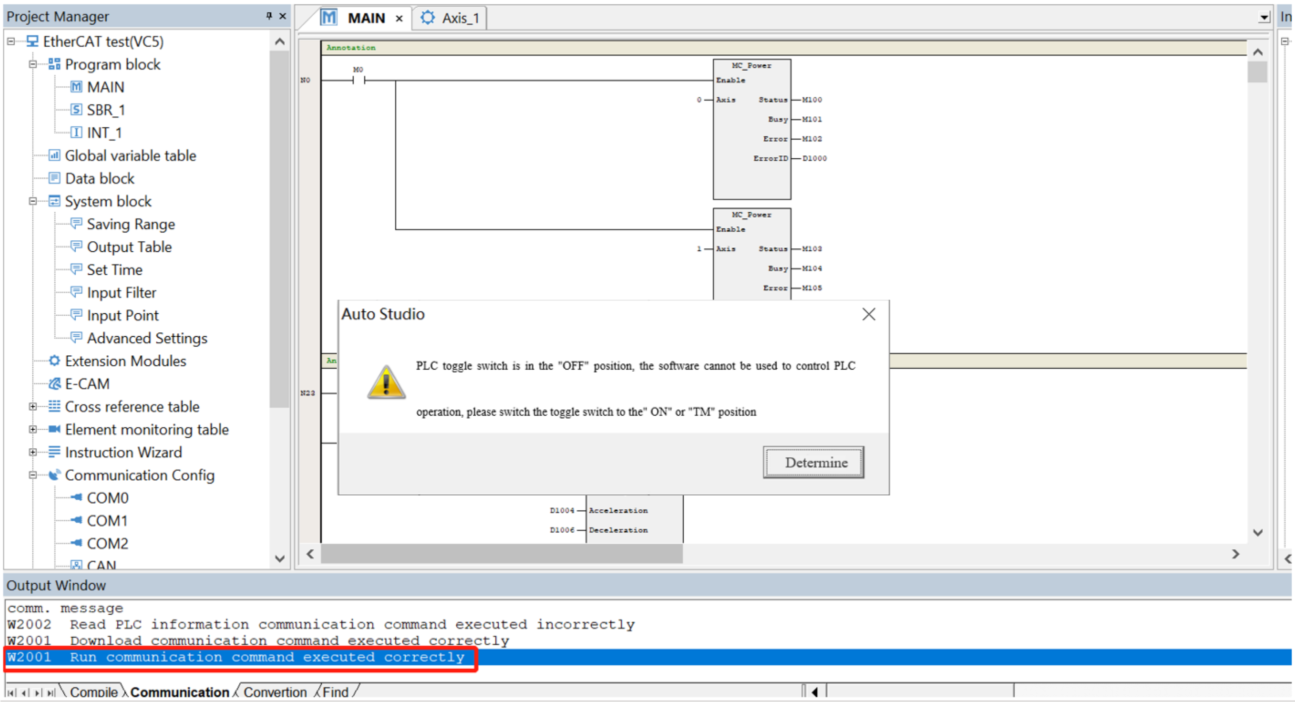


3. If the PLC is running before downloading the program, after the download is completed, you can see the prompt that the download communication command executes correctly, and a dialog box pops up, select "OK" to switch the PLC to the

running state.

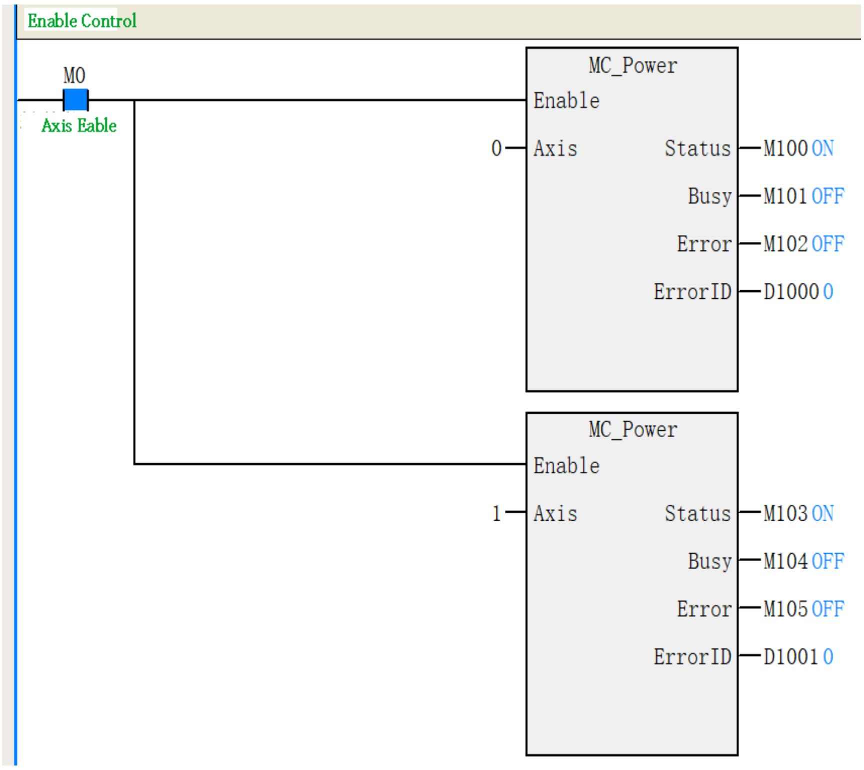


4. If the PLC is in a stopped state before downloading the program, the information output window pops up after the download is completed and the download communication command is prompted to execute the correct one. The PLC switch needs to be manually switched to the operating state.



4.4.6 PLC program monitoring

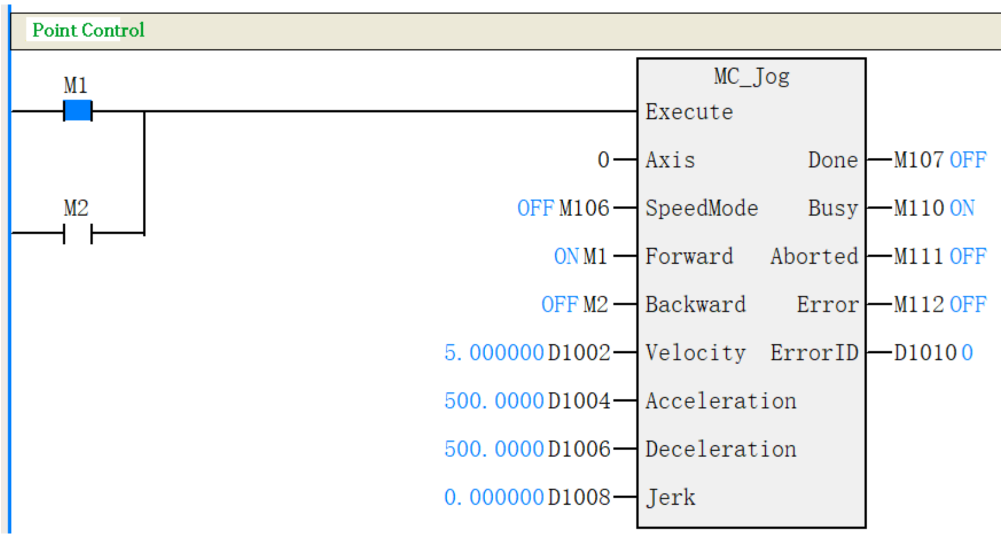
Enable control, set M0 to TRUE, the bus servo axis and the local pulse axis are enabled, and the axis enable status status shows ON.



4.4.6.1 PLC Bus servo axis commissioning monitoring

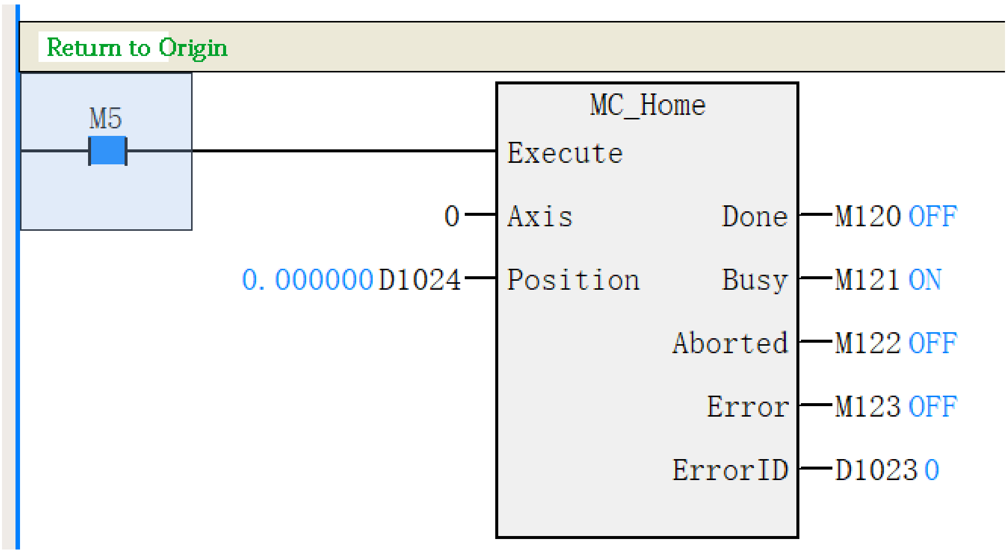
1. Bus servo axis jogging operation.

● Set the variable M1 to TRUE, at this time the bus servo axis starts to run forward at the set speed, and the axis driven by the actual servo drive is 5Unit/sec forward operation.

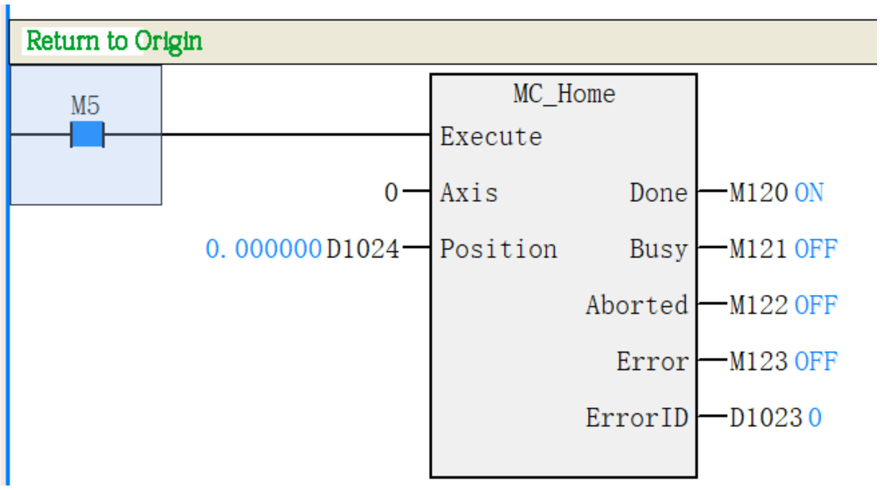


2. Bus servo axis origin regression test.

● Put M5 to ON, Home_busy will automatically set ON.

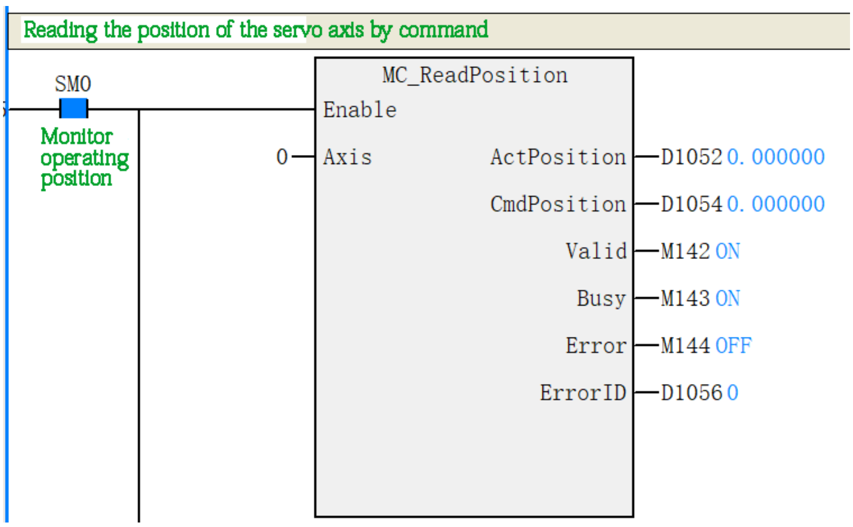


● When the servo motor touches the Z signal, the origin return is automatically completed, at this time Home_done ON, Home_busy will automatically set OFF.

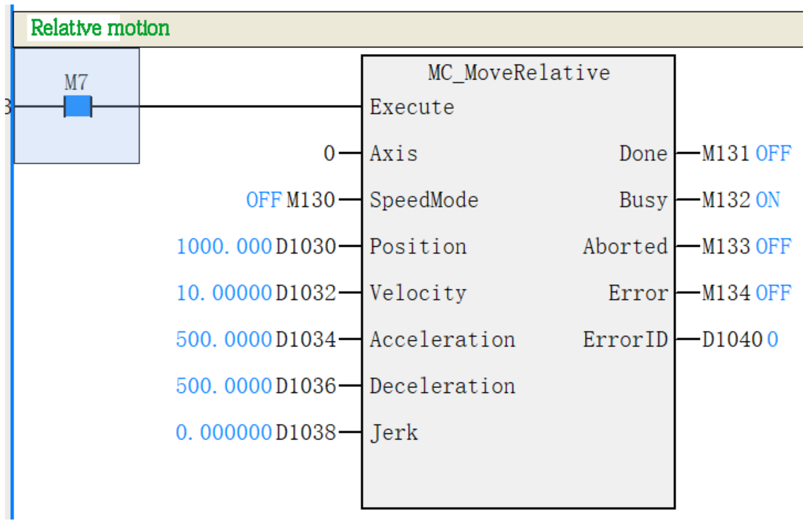


3. Bus servo axis relative positioning test.

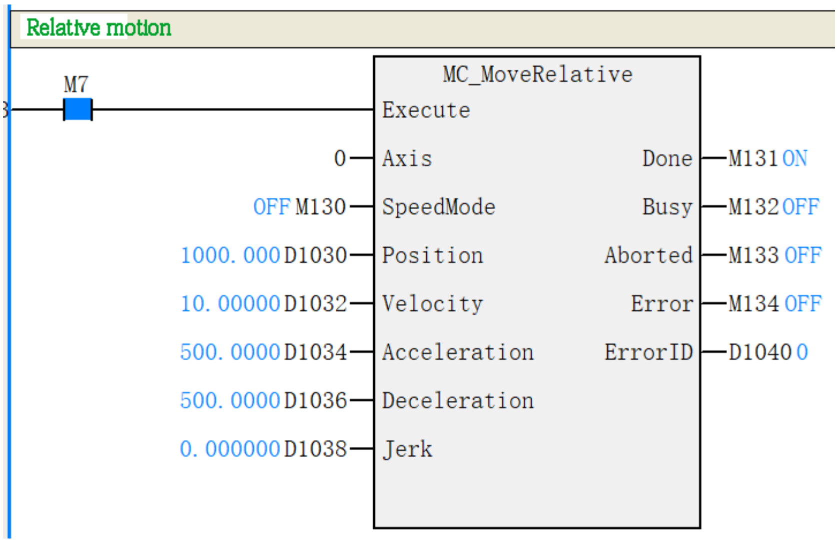
● The current position of the bus servo axis can be obtained by reading the axis position command：



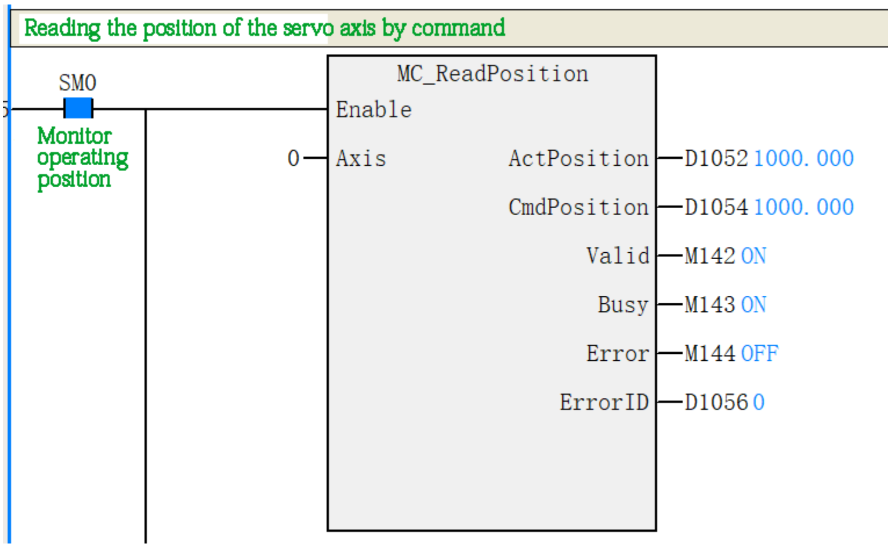
● Set the variable M7 to TRUE and the function block output variable busy output to TRUE, at which point the bus servo axis begins to move.



● After the positioning is complete, the done variable outputs as TRUE.



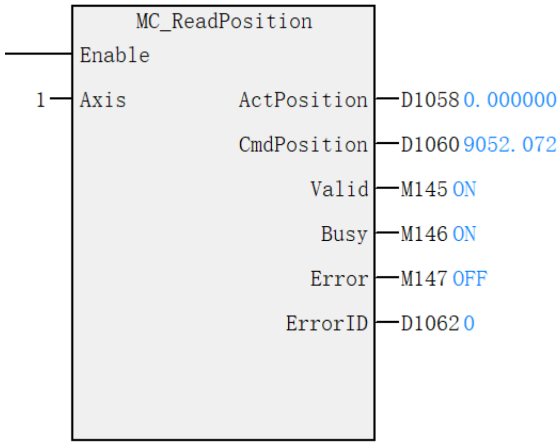
● By reading the position command, you can see that the position of the servo at this time is at the set target position.



4.4.6.2 PLC Local pulse axis commissioning monitoring

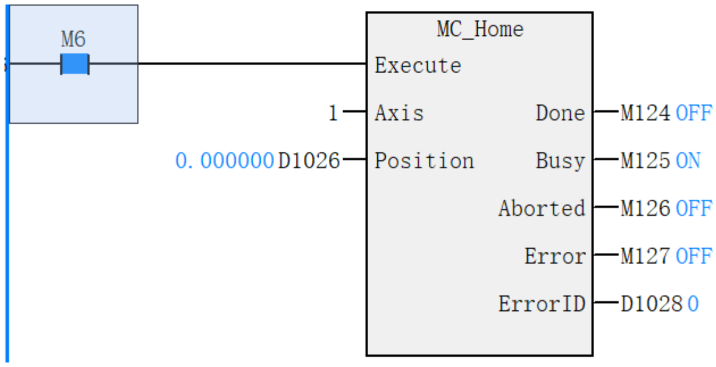
1. Local pulse axis current position monitoring.

● The current position of the local servo axis can be obtained by reading the axis position command：

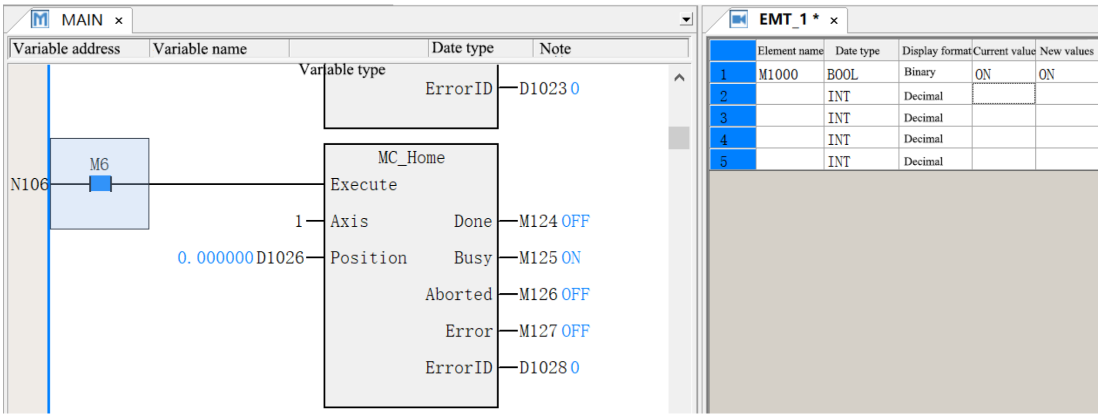


2. Click the "Origin Regression" button to start the origin regression and correctly manipulate the M1000 variable to complete the origin return action.

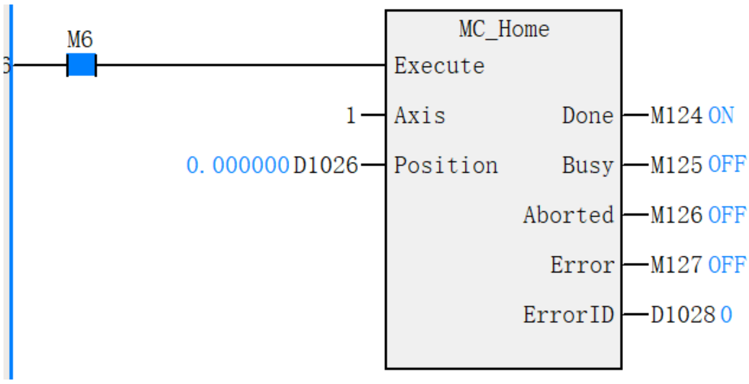
● Start origin regression when M1000=FALSE.



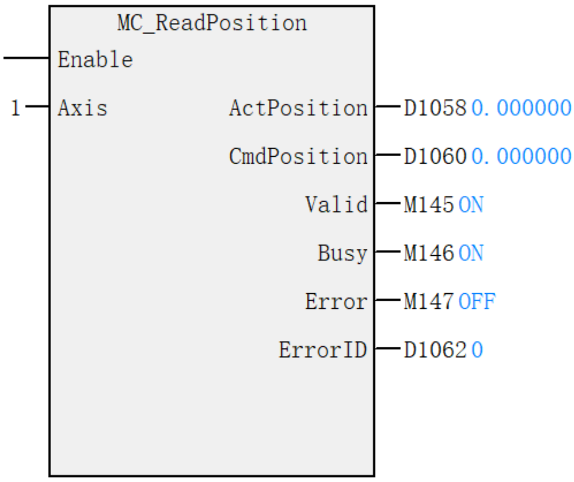
● Set M1000 to TRUE.



● Set M1000 to FALSE and origin regression is complete.



3. Monitors the position after the origin is regressed.



4.5 Motion Control Axis Group

4.5.1 Introduction of interpolation function

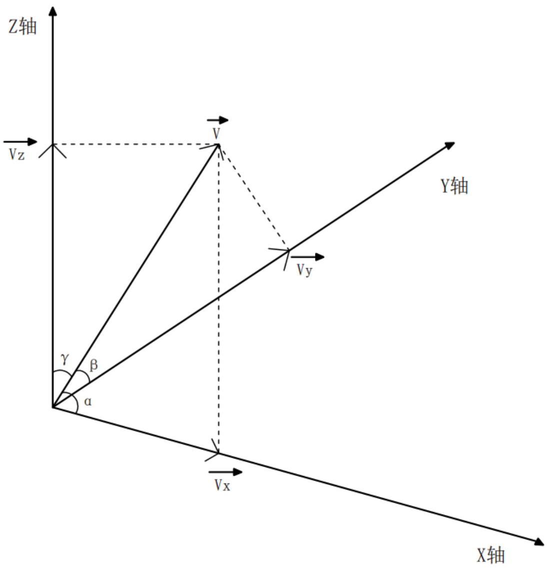
4.5.1.1 Overview

VC5 series PLC interpolation adopts space right angle coordinate system, supports linear interpolation and circular interpolation, and the interpolation function is realized in the way of axis group.

● Each axis group can control up to 8 motion control axes (bus servo axes or local pulse axes), including three coordinate axes X, Y, Z and five auxiliary axes A, B, C, H, W。

● VC5 supports up to 8 axis groups, each of which can be set to 2 axes (XY axis), 3 axes (XYZ), 4 axes (XYZ and auxiliary axes), and 8 axes, etc.

● Linear interpolation and circular interpolation support buffering mode. Up to 8 curves can be buffered for each axis group, and the transition mode between curves can be set individually.



Spatial Cartesian Coordinate System

In the figure, Vx, Vy, and Vz represent the fractional velocities of the three axes, which are also the actual operating velocities of the servo axes. v represents the real-time velocity of the interpolation curve. α, β, and γ represent the angle between the velocity of the interpolation curve and the coordinate axes, respectively.

Auxiliary axis linear coordinate system

In linear interpolation, the motion control axes representing x, y and z axes move along the coordinate axes, and the auxiliary axes move from the starting position to the end position along a straight line. When circular interpolation, one of xy-axis plane, yz-axis plane and xz-axis plane can be selected to do circular interpolation. At this time, if other axes are also configured in the axis group, the other axes move along the straight line from the starting position to the end position.

4.5.1.2 Axis group command list

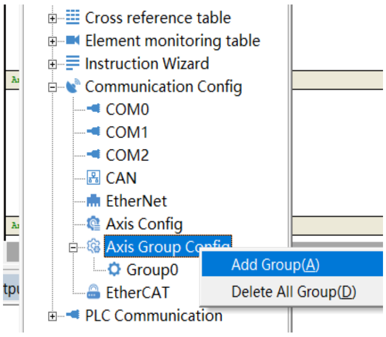
The list of axis group control commands supported by VC5 is as follows：

Command	Function Description
MC_MoveLinear	Linear interpolation command
MC_MoveCircular	Circular arc interpolation command
MC_GroupStop	Stop axis group running
MC_GroupPause	Pause axis group running
MC_AddAxisToGroup	Add axis to an axis group
MC_RemoveAxisFromGroup	Remove axis from axis group

MC_PathAdd	Add path
MC_PathMove	Path movement
MC_SetForwardlookingPara	Forwardlooking parameter setting command

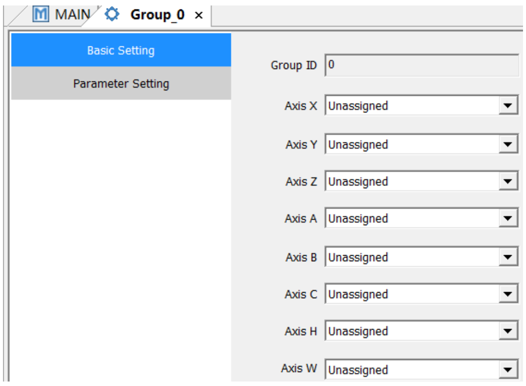
4.5.1.3 Configuration Interface

Right-click on the "Axis Group Config" option, "Add Group", create a new axis group, and double-click to open the configuration interface of the axis group：



The axis group config screen contains two parts: "Basic Setting" and "Parameter Setting".

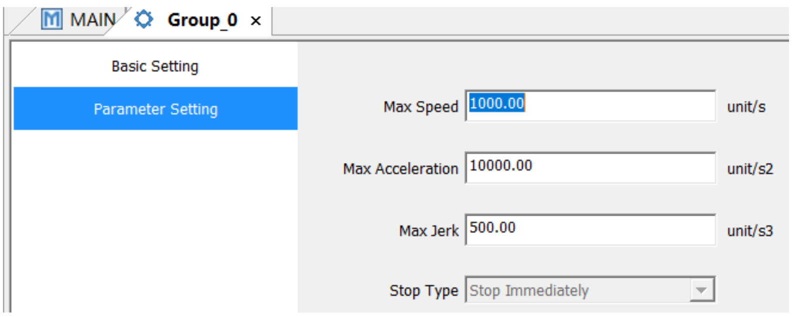
Basic Setting



● Axis group number: used to number the axis group

● Coordinate axis selection： Used to select coordinate axis, where x-axis and y-axis are mandatory and z-axis and auxiliary axis can be empty.The axis is allowed to be repeated between different axis groups.

Parameter Setting



● Maximum speed: linear interpolation mode indicates the maximum speed of interpolation for spatial straight lines; circular arc interpolation indicates the maximum linear speed of circular arcs.

●Maximum acceleration: linear interpolation mode indicates the interpolated maximum acceleration value of spatial straight line; circular interpolation indicates the maximum acceleration value of circular arc.

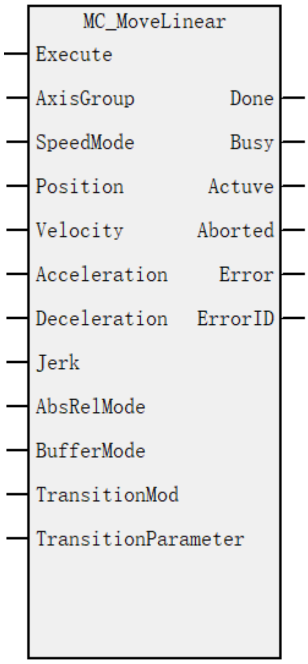
● Maximum jerk：The linear interpolation mode indicates the interpolated maximum jerk value of a straight line in space; the circular arc interpolation indicates the maximum jerk value of a circular arc.。

● Fault stop method：Refers to the stopping method in the event of an axle set failure。

4.5.2 Axis group command

4.5.2.1 MC_MoveLinear（Linear interpolation）

Drawing Block



Command list format

Input/output	Name	Data Type	Applicable（soft components)	Range	Description	Enable
IN	AxisGroup	WORD	Constant/D/R/W	0~7	Axis group	Execute
IN	SpeedMode	BOOL	M/S	TRUE,FALSE（Default 0)	Speed Type: Type 0T, Type 1S	
IN	Position	REAL	Constant/D/R/W	Positive/Negative Number/0	Array of target positions for each axis Array arrangement：（X、Y、Z、A、B、C、H、W)	

IN	Velocity	REAL	Contstant/D/R/W	Positive Number	Target speed	
IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	
IN	Deceleration	REAL	Constant/D/R/W	Positive Number (Default Acceleration)	Deceleration	
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed step	
IN	AbsRelMode	WORD	Constant/D/R/W	0: Absolute positioning 1: relative positioning (default: 0))	Absolute position and relative position mode	
IN	BufferMode	WORD	Constant/D/R/W	0:Interrupt + no transition 1:Buffer + no transition 2: Reserved 3: Previous speed + no transition 10: Buffer+transition (Default 0)	Buffer mode	
IN	TransitionMod	WORD	Constant/D/R/W	0: Speed transition (default: 0))	Transition Mode	
IN	TransitionParameter	REAL	Constant/D/R/W	Positive number/0 (default: 0) Value range 0~1	Transition parameter	
OUT	Done	BOOL	M/S	TRUE, FALSE	Finish Stoppping	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Actuve	BOOL	M/S	TRUE, FALSE	ON at the start of this curve	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interruption	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error Flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

MC_MoveLinear instruction is used to control the linear interpolation function of the axis group, and the rising edge is valid (the flag is not updated after the energy flow is invalidated).

● Setting of axis number

Latch the Group on the rising edge of the Execute input. Modifying the Group during Execute=ON is invalid. Modifying Group during Execute=OFF is invalid.

● Relationship to single-axis control commands

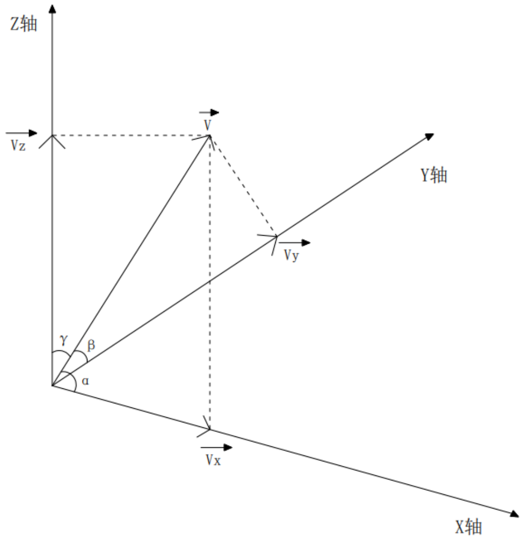
This command can only be triggered by calling the MC_Power command to switch all axis in the axis group to the StandStill state.

Triggering this instruction during single-axis motion (e.g., point motion, torque control, return to home, stop, etc.) is not valid.

After this instruction is triggered, the PLCOpen state machine of a single axis is in synchronous motion mode, and cannot be interrupted by single-axis motion commands during the motion.

● Parameter Description

Position is used to set the target position or shift, Position[0] denotes the position displacement component of x-axis, Position[1] denotes the position displacement component of y-axis, Position[2] denotes the position displacement component of z-axis, Position[3] denotes the position displacement component of auxiliary axis.



$$Vx = V \times \cos\alpha \quad (1)$$

$$Vy = V \times \cos\beta \quad (2)$$

$$Vz = V \times \cos\gamma \quad (3)$$

$$V = \sqrt[2]{Vx^2 + Vy^2 + Vz^2 + Va^2 + Vb^2 + Vc^2 + Vh^2 + Vw^2} \quad (4)$$

Velocity denotes the target rate of the interpolator. The velocities of X, Y and Z axis in the figure are decomposed according to equations (1), (2) and (3), and the velocities of auxiliary axis A, B, C, H and W are calculated in the same way as those of X, Y and Z axis.

The interpolation speed of the auxiliary axis is divided into two cases:

1. When the point on the coordinate axis is not moved and the auxiliary axis is moved alone, the auxiliary axis moves according to the target speed set by Velocity。

2. When the point on the coordinate axis moves, the auxiliary axis and will reach the target position at the same time as the point on the coordinate axis. Suppose the length of the interpolation line is L1, the target displacement of the auxiliary axis is L2, and the rate of the interpolation line at a certain moment is V0, then the velocity Va of the auxiliary axis is calculated as follows：

$$Va = V0 \times \frac{L2}{L1} \quad (5)$$

● Relative absolute mode selection

AbsRelMode = 0 Indicates absolute positioning mode. After triggering this instruction, the three axis groups finally move to the position specified by (Position[0], Position[1], Position[2]), and the auxiliary axis moves to the position specified by Position[3].

AbsRelMode = 1 Indicates relative positioning mode. Let the positions of the three axes of the axis group be (Px,Py,Pz) and the current position of the auxiliary axis is Pa. After triggering this instruction, the three axes finally move to (Px+ Position[0],Py+ Position[1],Pz+ Position[2]). The final position of the auxiliary axis is Pa+ Position[3].

● The following four modes are available for buffering and transition buffering and transition：

Serial number	Buffer mode	Description
0	Interruption + no transition	Immediate switch to the next function block action, no transition curve
1	Buffered + No Transition	The first deceleration period is completed and the function block of buffering is executed without transition curve.
3	Previous Speed + No Transition	Go to the end of the first segment at the current speed and start the second segment at the rate of the first segment
10	Buffer + Transition	There are transition curves that join the acceleration of the second segment at the beginning of the first segment to perform the deceleration

When the three buffering modes of 1, 3 and 10 are selected, the interpolation instruction allows up to 8 curves to be buffered. When the instruction enters the buffered state, the Busy signal is valid, the Active output is valid when this instruction starts to execute, and the Done signal output is valid when the instruction execution is completed.

When 0 (interrupt + no transition) mode is selected for the newly added interpolation instruction, this instruction will interrupt all interpolation instructions being executed and in buffer, and the interrupted interpolation instruction CommandAborted output is valid.

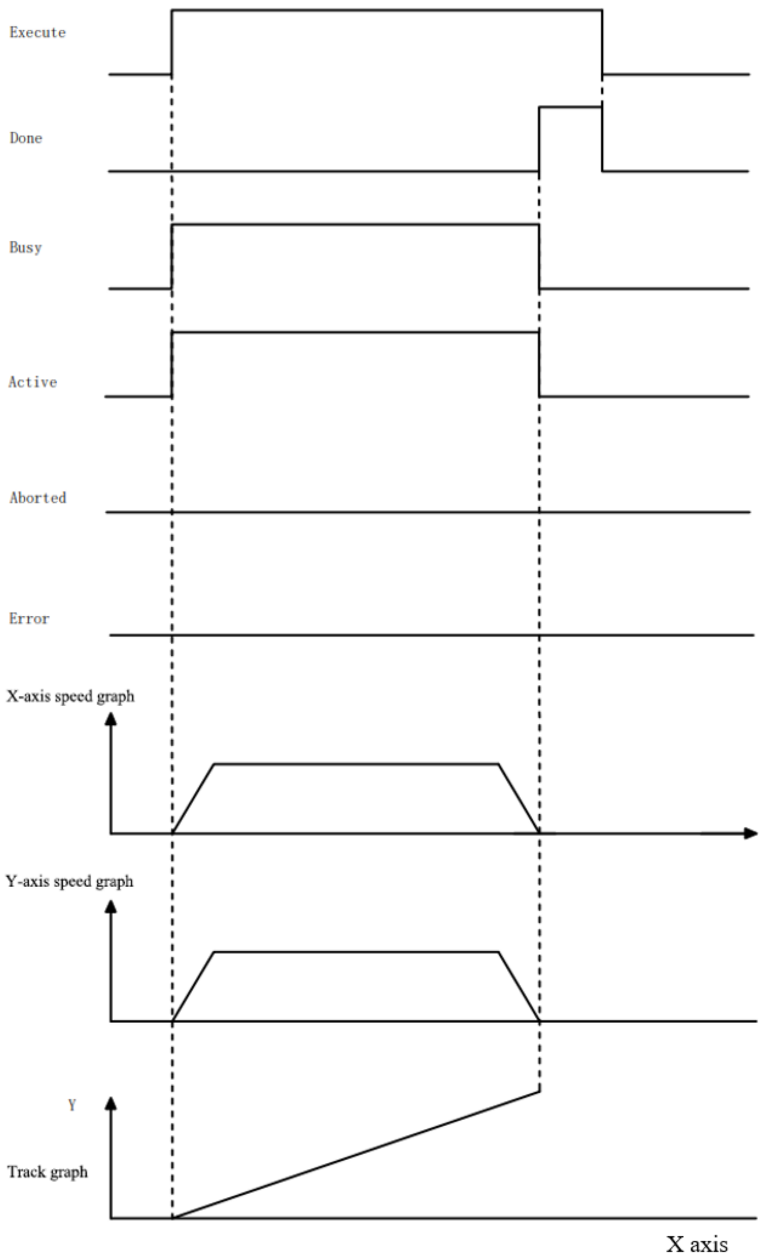
● Restart this command

Unable to restart this command.

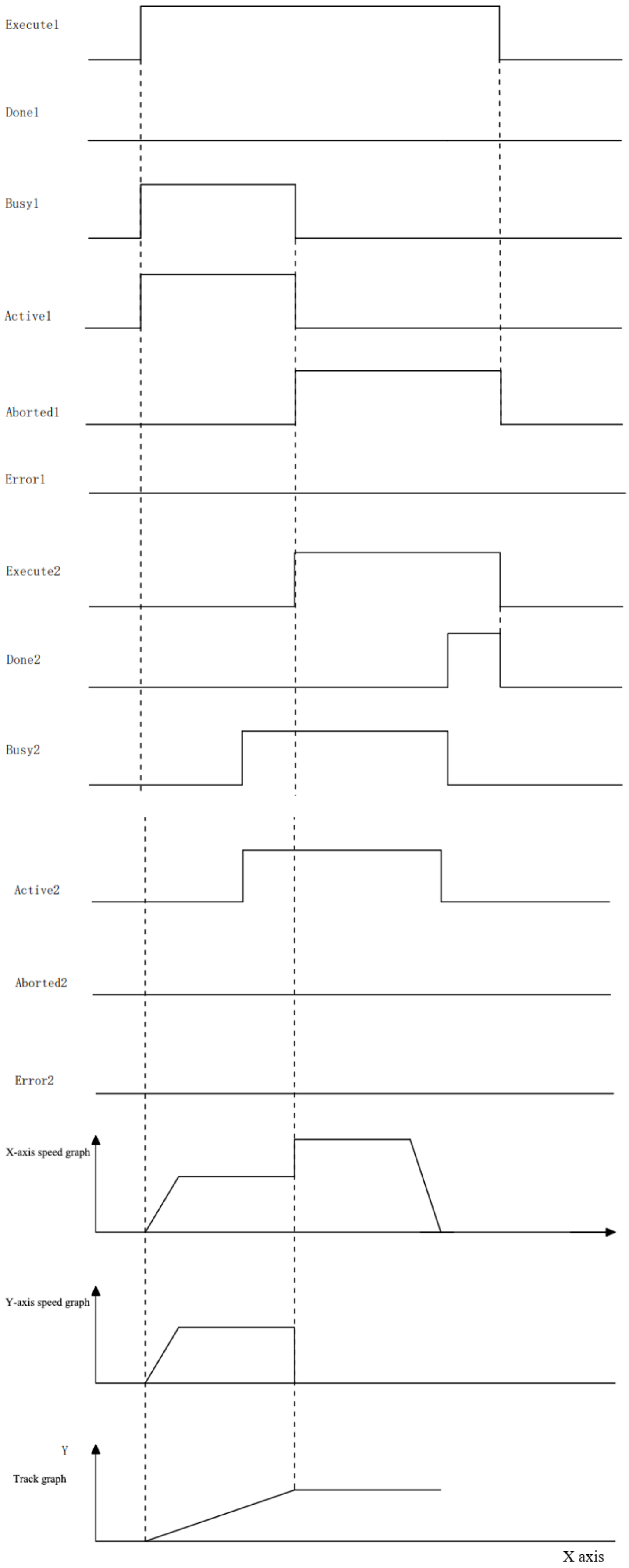
If this instruction is repeatedly triggered when Busy output is ON, the axis will report a fault (repeatedly triggered instruction fault) and will re-execute the instruction once again when the execution of this instruction is completed.

Timing diagram

● Call a linear interpolation command to interpolate in the X-axis and Y-axis directions.

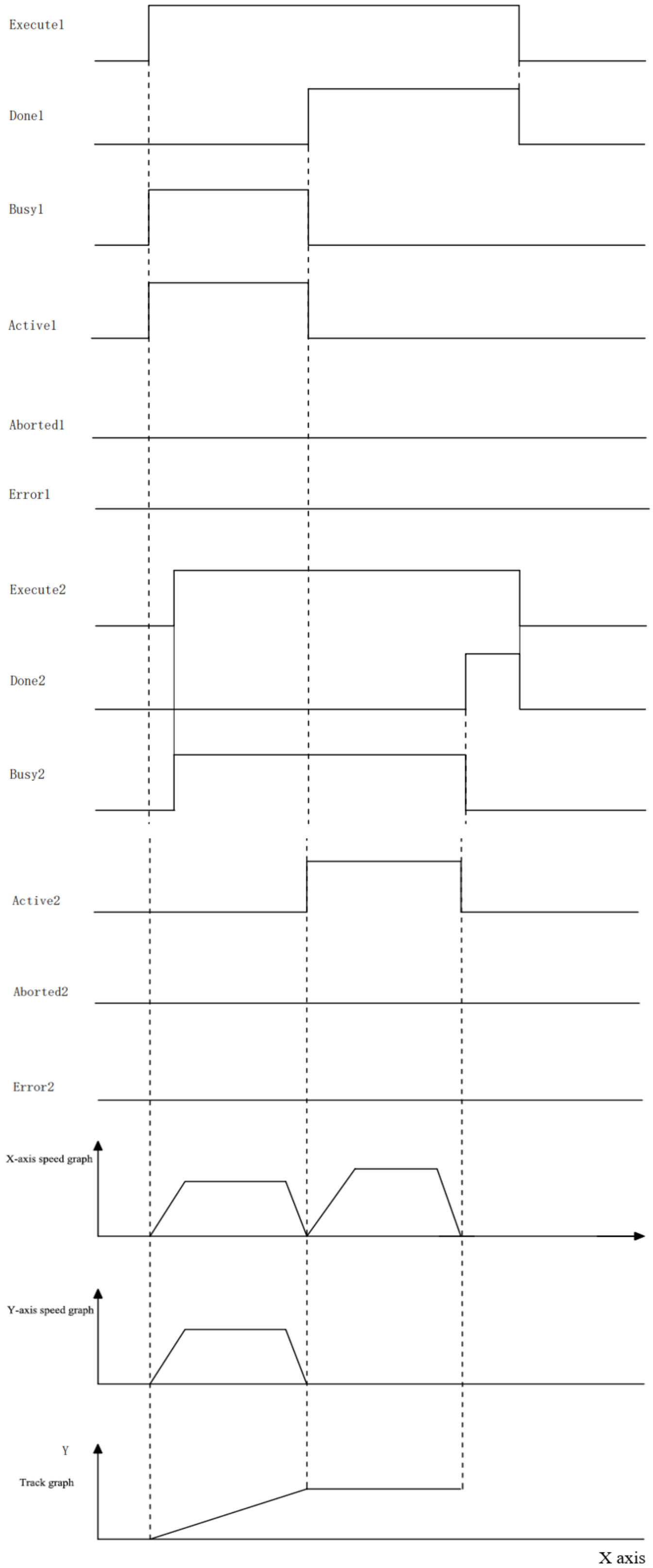


● Two linear interpolation instructions are called, and the second interpolation instruction is triggeredduring the first interpolation instruction to interrupt the first interpolation instruction.

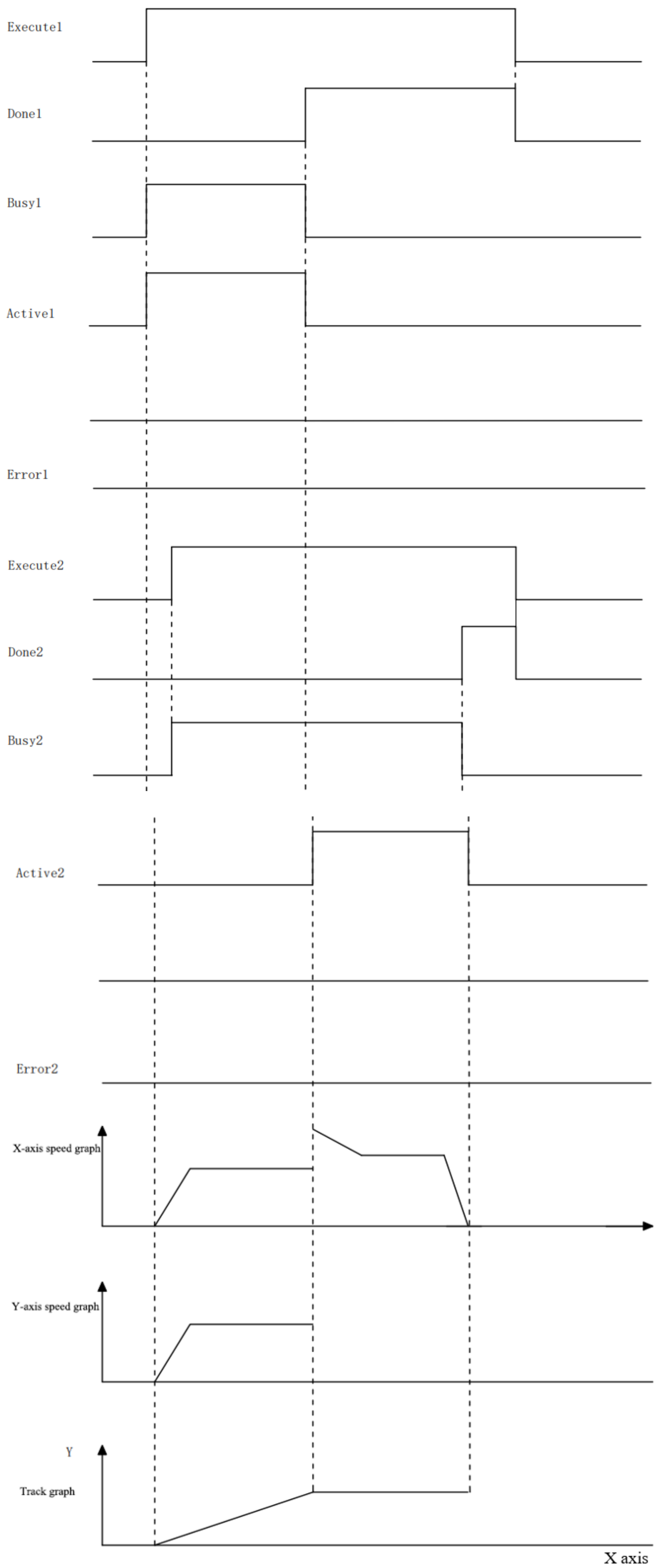


Execute1
Done1
Busy1
Active1
Aborted1
Error1
Execute2
Done2
Busy2
Active2
Aborted2
Error2
X-axis speed graph
Y-axis speed graph
Y
Track graph
X axis

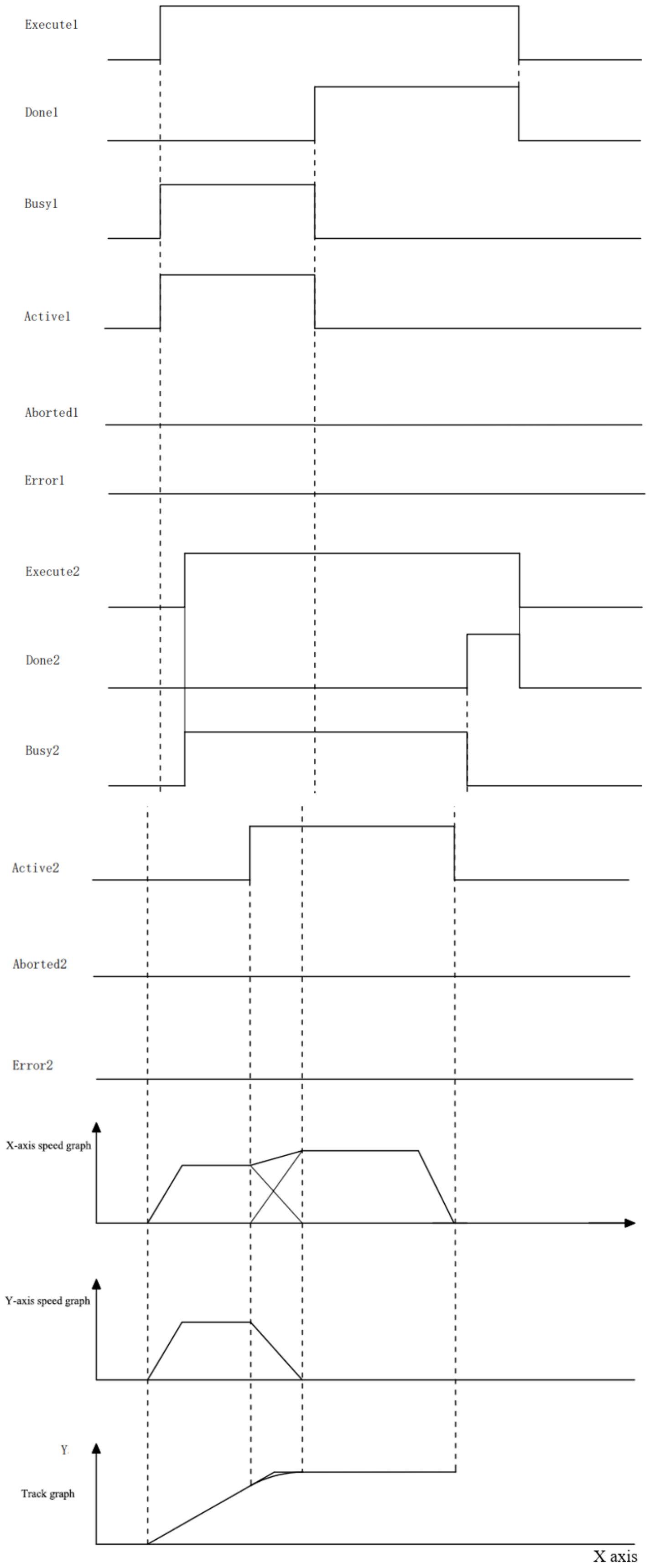
●Two linear interpolation instructions are called, and the second instruction is executed in "buffer + no transition" mode.



● Calling two linear interpolation instructions, the second instruction is executed in "previous speed + no transition" mode.

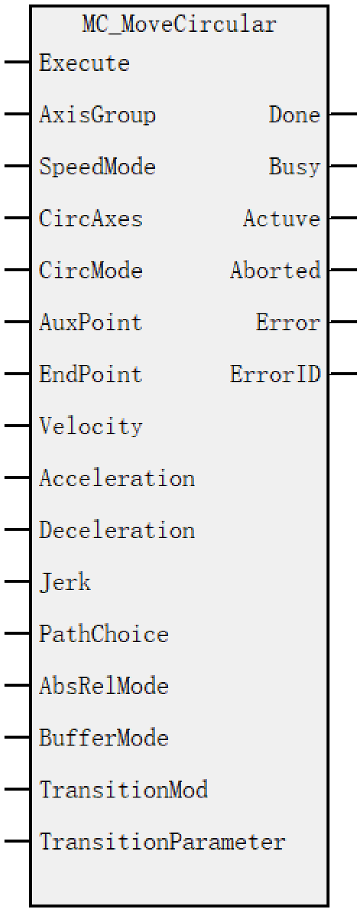

Execute1
Done1
Busy1
Active1
Error1
Execute2
Done2
Busy2
Active2
Error2
X-axis speed graph
Y-axis speed graph
Y
Track graph
X axis

● Calling two linear interpolation instructions, the second instruction is executed in "buffer + transition" mode.



4.5.2.2 MC_MoveCircular（Circular arc interpolation）

Drawing block



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	AxisGroup	WORD	Constant/D/R/W	0~7	Axis group	Execute
IN	SpeedMode	BOOL	M/S	TRUE,FALSE（Default 0）	Speed type. 0T type, 1S type	
IN	CircAxes	WORD	Constant/D/R/W	0：x-y axis plane 1：y-z axis plane 2：x-z axis plane （Default：0）	Circular arc designation	
IN	CircMode	WORD	Constant/D/R/W	0：Designated as a pass-through point 1：Designated as the center point	Circular arc interpolation mode	

				2：Designated as radius (Default：0)		
IN	AuxPoint	REAL	Constant/D/R/W	Positive/Negative Number/0	X/Y/Z axis auxiliary position array array arrangement:（X、Y、Z）	
IN	EndPoint	REAL	Constant/D/R/W	Positive/Negative Number/0	Array arrangement of target position arrays for each axis：（X、Y、Z、A、B、C、H、W）	
IN	Velocity	REAL	Constant/D/R/W	Positive Number	Target speed	
IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	
IN	Deceleration	REAL	Constant/D/R/W	Positive Number (Default Acceleration)	Deceleration	
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed step	
IN	PathChoice	WORD	Constant/D/R/W	0：CW 1:：CCW (Default：0)	Path choice	
IN	AbsRelMode	WORD	Constant/D/R/W	0：Absolute positioning 1：Relative positioning (Default：0))	Absolute positioning and relative positioning modes	
IN	BufferMode	WORD	Constant/D/R/W	0：Interruption + no transition 1：Buffering + no transition 2：Reserved 3：Previous speed + no transition 10：Buffer + Transition (Default 0)	Buffer mode	
IN	TransitionMod	WORD	Constant/D/R/W	0：Speed transition (Default：0))	Transition mode	
IN	TransitionParameter	REAL	Constant/D/R/W	Positive Number/0 (Default：0)	Transition parameter	
OUT	Done	BOOL	M/S	TRUE,FALSE	Finish stopping	
OUT	Busy	BOOL	M/S	TRUE,FALSE	In-Run flag	

OUT	Actuve	BOOL	M/S	TRUE,FALSE	ON at the start of this curve	
OUT	Aborted	BOOL	M/S	TRUE,FALSE	Execution interruption	
OUT	Error	BOOL	M/S	TRUE,FALSE	Error flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

MC_MoveCircular is used to control the axis group circular arc interpolation function, and the rising edge is valid.

● Setting of axis number

Latch Group on rising edge of Execute input.

Modifying a Group during Execute=ON is not valid.

Modifying a Group during Execute=OFF is not valid。

●Relationship to single-axis control commands

This instruction can only be triggered by calling the MC_Power instruction to switch the axis to the StandStill state.

Triggering this command during single-axis motion (e.g., point motion, torque control, return to home, stop, etc.) is not valid.

After this instruction is triggered, the PLCOpen state machine of single-axis is in synchronous motion mode, and it cannot be interrupted by single-axis motion instructions during the motion.

● CircAxes specifies the axis plane. The meaning is as follows：

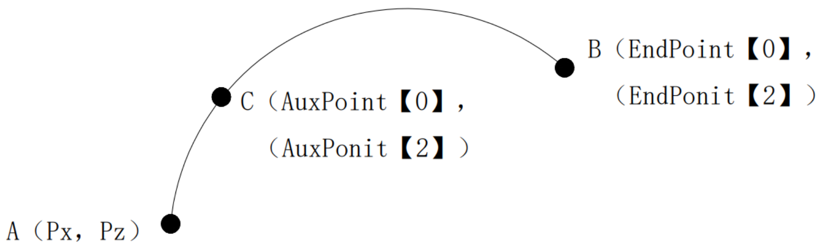
CircAxes = 0 means the x-y axis plane is selected. the axis specified by AxisID_x and AxisID_y perform circular arc interpolation, and the axis specified by AxisID_z and AxisID_a are auxiliary axes, which perform linear interpolation.

CircAxes = 1 means the y-z axis plane is selected. the motion axis specified by AxisID_y and AxisID_z performs circular arc interpolation, and the axis specified by AxisID_x and AxisID_a is the auxiliary axis, which performs linear interpolation.

CircAxes = 2 means the x-z axis plane is selected. the axis of motion specified by AxisID_x and AxisID_z perform circular arc interpolation, and the axis specified by AxisID_y and AxisID_a are auxiliary axis to perform linear interpolation.

● Interpolation mode selection

1. CircMode = 0 means circular arc interpolation according to the pass point.



Starting and finishing points are not at the same point

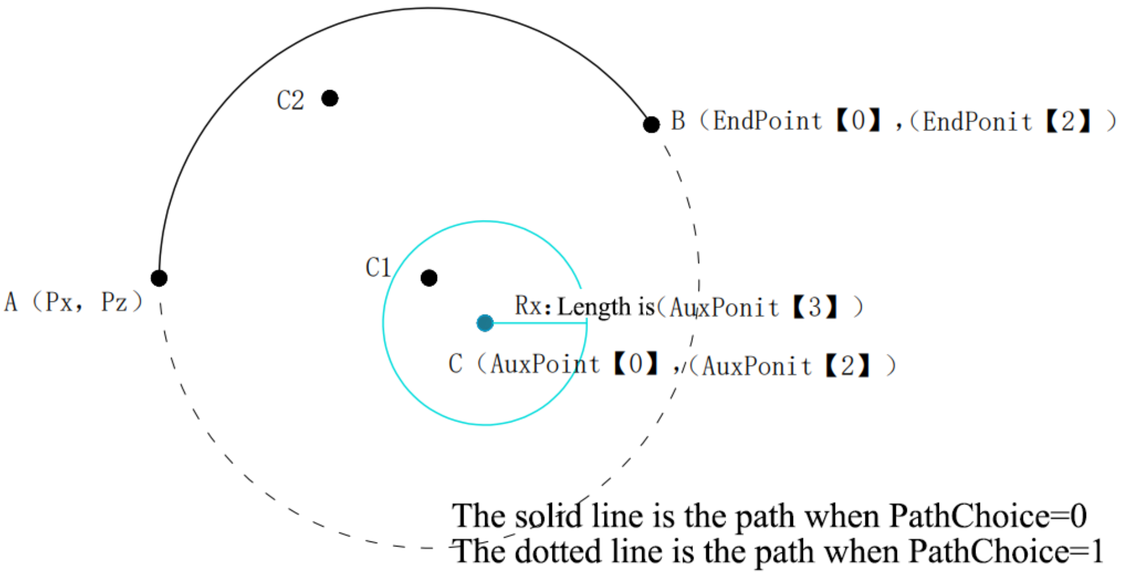
Select the x-y plane through the point (AuxPoint[0], (AuxPoint[1]) and the end point (EndPoint[0], EndPoint[1]);

The y-z plane is selected by (AuxPoint[1], (AuxPoint[2]) and the endpoint is (EndPoint[1], EndPoint[2])；

The x-z plane is selected by (AuxPoint[0], (AuxPoint[2]) and the end point is (EndPoint[0], EndPoint[2]).

Take x-y plane as an example, the starting position of x-axis is Px, and the starting position of y-axis is Py. After triggering

the command, a circular interpolation with (Px, Py) as the starting point and (EndPoint[0], EndPoint[1]) as the ending point and passing through the point (AuxPoint[0], (AuxPoint[1]) will be executed.
When the starting point, passing point and end point are on the same line, it cannot form a circle, the instruction reports an error and stops the execution of the interpolation instruction.
When the passing point and the end point are the same point or the starting point and the passing point are the same point, the instruction reports an error and stops the execution of the interpolation instruction.
2. CircMode = 1 means interpolation of circular arcs according to the center point.

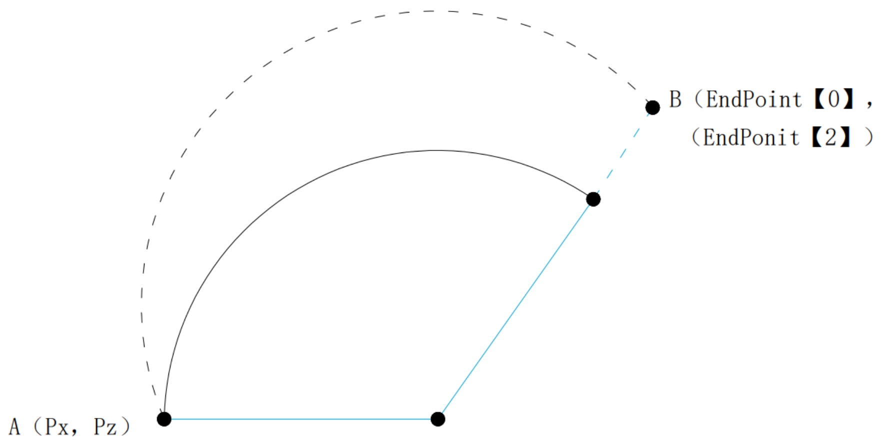


The center point is (AuxPoint[0], (AuxPoint[1]) and the end point is (EndPoint[0], EndPoint[1]) when the x-y plane is selected;

The center point is (AuxPoint[1], (AuxPoint[2]) and the end point is (EndPoint[1], EndPoint[2]) when the y-z plane is selected;

The center point is (AuxPoint[0], (AuxPoint[2]) and the end point is (EndPoint[0], EndPoint[2]) when the x-z plane is selected;

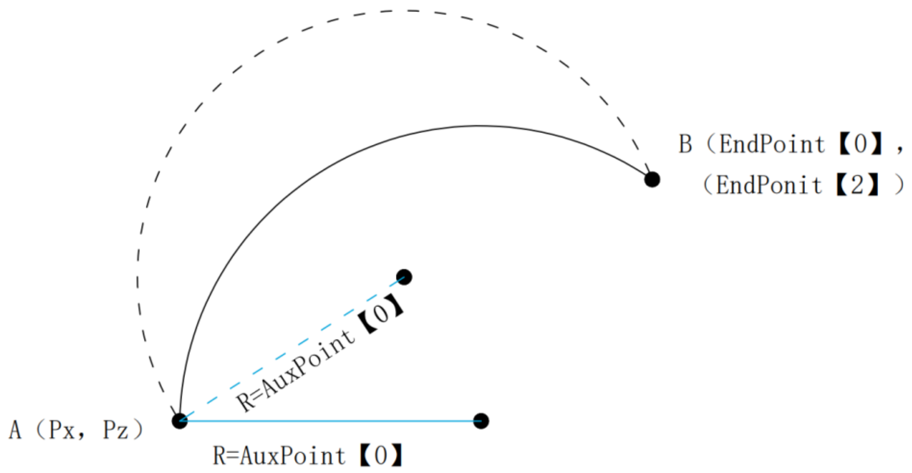
Take x-axis and z-axis as an example, the starting position of x-axis is Px, the starting position of z-axis is Pz, after triggering the command, the arc interpolation will be executed with (Px, Pz) as the starting point, (AuxPoint[0], (AuxPoint[2]) as the center and (EndPoint[0], EndPoint[2]) as the end point, the direction of rotation of the arc drawing PathChoice decides.
When the distance R1 from the specified center position (AuxPoint[0], AuxPoint[2]) to the starting point (Px, Pz) is different from the distance R2 to the end point (EndPoint[0], EndPoint[2]) (the difference between R1 and R2 is greater than 1), the radius will not be adjusted when the radius is not the same, and the actual target position of the drawn circle is the position of the corresponding angle of the target point on the circular arc.



The solid line is the actual interpolated path

The dotted line is the path of the interpolation setup

3. CircMode = 2 means interpolation of the circular arc according to the specified radius.



The solid line is the path when R = positive

The dashed line shows the path when R = negative

The radius of the arc is always determined by |AuxPoint[0]|, regardless of which plane is selected. The end point is (EndPoint[0], EndPoint[1]) when choosing the x-y plane; the end point is (EndPoint[1], EndPoint[2]) when choosing the y-z plane; the end point is (EndPoint[0], EndPoint[2]) when choosing the x-z plane; take the y-axis and z-axis for example, the starting position of y-axis is Py, the starting position of z The starting position of y-axis is Py and the starting position of z-axis is Pz. The y-axis and z-axis perform circular arc interpolation with (Py, Pz) as the starting point, |AuxPoint[0]| as the radius, and (EndPoint[1], EndPoint[2]) as the end point.

When the radius sign is negative, a longer circular arc is drawn; when the radius sign is positive, a shorter circular arc is drawn. The direction of rotation of the circular arc is specified by PathChoice.

● Choice of positioning mode

1. Absolute mode

When passing points are selected, the auxiliary and end points represent absolute points in the coordinate system。

When the center point is selected, the center point and the end point represent the absolute points in the coordinate system. When radius is selected, the end point represents the absolute point in the coordinate system.

2. Relative mode

When a pass point is selected, the auxiliary point and the end point represent relative points to the start point.

When the center point is selected, the center point and the end point represent relative points with respect to the start point.

When selecting a radius, the end point represents the relative point with respect to the start point.

● Buffering and Transition

When the buffering modes of 1, 2, and 3 are selected, the interpolation instruction allows up to 8 curves to be buffered. The Busy signal is valid when the instruction enters the buffered state, the Active output is valid when this instruction starts to execute, and the Done signal is valid when the instruction execution is completed.

When 0 (interrupt + no transition) mode is selected for the newly added interpolation instruction, this instruction will interrupt all interpolation instructions being executed and in buffer, and the interrupted interpolation instruction CommandAborted output is valid.

● Restart this command

Unable to restart this command.

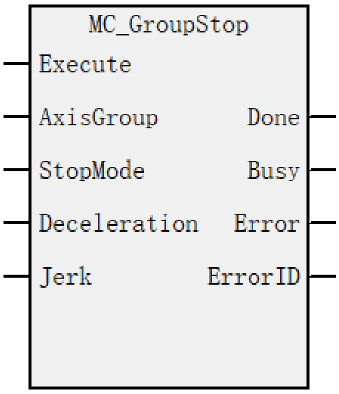
If this command is repeatedly triggered when the Busy output is ON, the axis will report fault 9421 (repeatedly triggered command fault), and all axis will stop operation immediately and enter the ErrorStop state.

Timing diagram

Reference linear interpolation instruction.

4.5.2.3 MC_GroupStop（Stop axis group running）

Drawing Block



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	AxisGroup	WORD	Constant/D/R/W	0~7	Axis group	Execute
IN	StopMode	WORD	Constant/D/R/W	0: Slow down and stop 1: Stop immediately (Default: 0)	Stop method	
IN	Deceleration	REAL	Constant/D/R/W	Positive Number	Deceleration	
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed step	

OUT	Done	BOOL	M/S	TRUE, FALSE	Finish stopping flag	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Rub flag	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

MC_GroupStop is for axis group to achieve stop function, command rising edge is valid.

● Setting of axis number

Latch the Group on the rising edge of the Execute input.

Modifying the Group during Execute=ON is invalid.

Modifying the Group during Execute=OFF is invalid.

● Effective scope

The MC_GroupStop instruction can only stop interpolation-type instructions (such as MC_MoveLinear), but not single-axis motion-type instructions (such as MC_ MoveAbsolute).

MC_Stop instruction can stop the corresponding interpolation instruction, and the whole axis group will slow down and stop when it is executed.

●State transition

On the rising edge of Execute, the interpolator executes a stop according to the stop mode set by StopMode and interrupts all interpolation instructions that are in the buffered state.The Done signal output is valid after the shutdown is completed, and the single-axis PLCOpen state machine is still in the Synchronized Motion state.

During Execute=ON, the interpolator remains stopped, and triggering a new interpolation instruction is not valid at this time.

On the falling edge of Execute, the interpolator will switch to the non-stop state and the single axis will enter the StandStill state, where a new interpolation instruction can be triggered.

● Stop mode

When StopMode is set to 0, deceleration will be stopped at the deceleration rate set by Deceleration；

When StopMode is set to 1, it will stop immediately without deceleration process.

● Restart this command

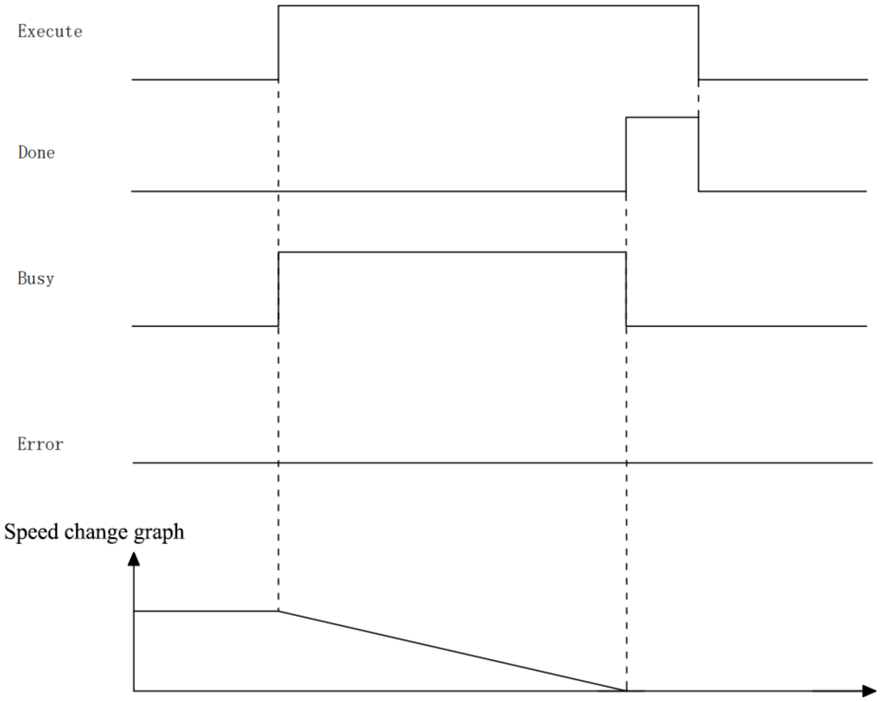
When this command is triggered again during axis deceleration, the axis in the axis group will be decelerated according to the new deceleration speed.

● Multiple calls

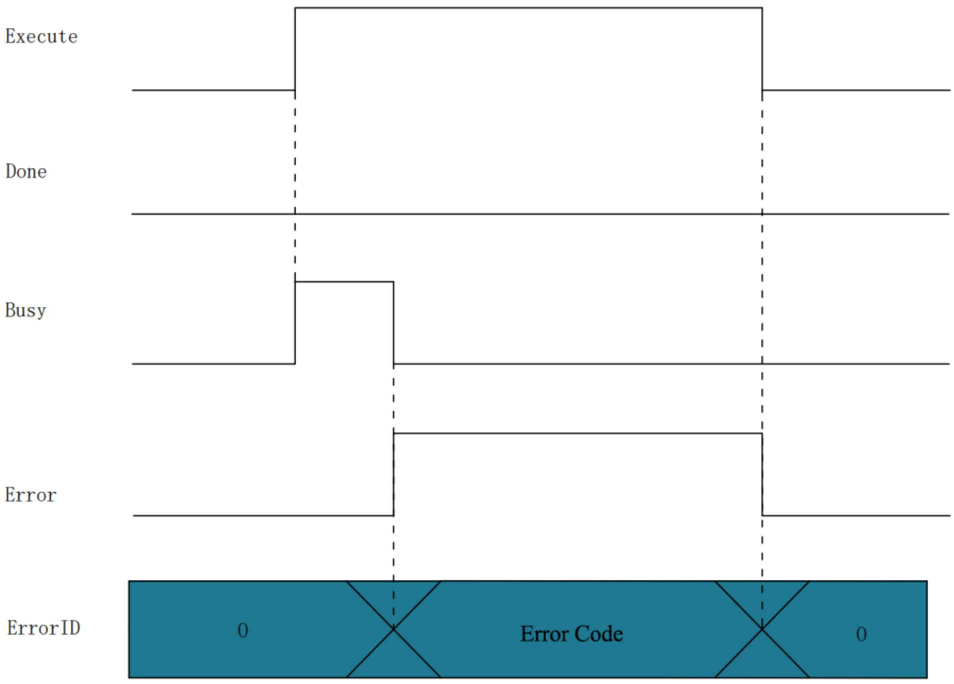
Multiple calls to this command are not allowed. Other MC_ GroupStop instructions are triggered when an MC_ GroupStop instruction is executing and the Execute input is ON, and the newly triggered MC_ GroupStop instruction reports an error (the axis is in the stopped state).

Timing diagram

● When there is a deceleration process and can stop normally.

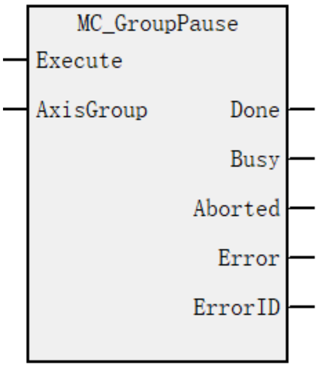


● When there is a axis failure in the deceleration process.



4.5.2.4 MC_GroupPause（Pause axis group running）

Drawing Block



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	AxisGroup	WORD	Constant/D/R/W	0~7	Axis group	Execute
OUT	Done	BOOL	M/S	TRUE,FALSE	Finish stopping	
OUT	Busy	BOOL	M/S	TRUE,FALSE	In-Run flag	
OUT	Aborted	BOOL	M/S	TRUE,FALSE	Execution interruption	
OUT	Error	BOOL	M/S	TRUE,FALSE	Error flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

MC_GroupPause command is used to implement the pause function for the axis group, level valid。

● Setting of axis number

Latch Group on rising edge of Execute input.

Modifying the Group during Execute=ON is invalid.

Modifying a Group during Execute=OFF is not valid.

● Effective scope

The MC_GroupPause instruction can only pause interpolation-type instructions (e.g. MC_MoveLinear) and cannot stop single-axis motion-type instructions (e.g. MC_ MoveAbsolute).。

● State transition

When all the axis in the axis group are in the StandStill state:

When Enable is set to ON, the axis group is still in the StandStill state. If the linear interpolation instruction or circular interpolation instruction is triggered at this time, the axis group will switch to the Synchronized Motion state, but in the pause state, and no interpolation algorithm will be executed. Only when the Enable signal of MC_ GroupPause instruction is set to OFF, the interpolation algorithm will be started.

When all axis within an axis group are in Synchronized Motion.

On the rising edge of Enable, the interpolator executes the deceleration process according to the deceleration speed set by Deceleration, and the Done signal output is valid after the deceleration is completed, and the PLCOpen state machine of the single axis remains in the Synchronized Motion state, and the Busy signal and Valid signal of the interpolation instruction being executed during the pause remain valid.

During Enable=ON, the interpolator is always suspended, when a new interpolation instruction is triggered new interpolation instruction is in buffered state.

On the falling edge of Enable, the interpolator restarts the execution of the previously paused interpolation instruction.

● Restart this command

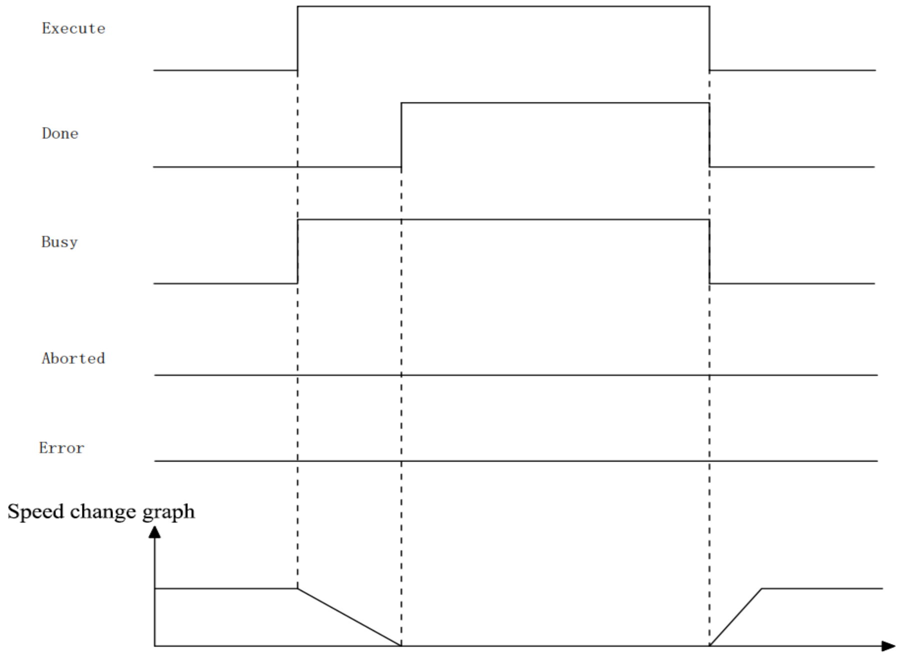
When this command is triggered again during axis deceleration, the axes in the group will be decelerated according to the new deceleration speed.

● Multiple calls

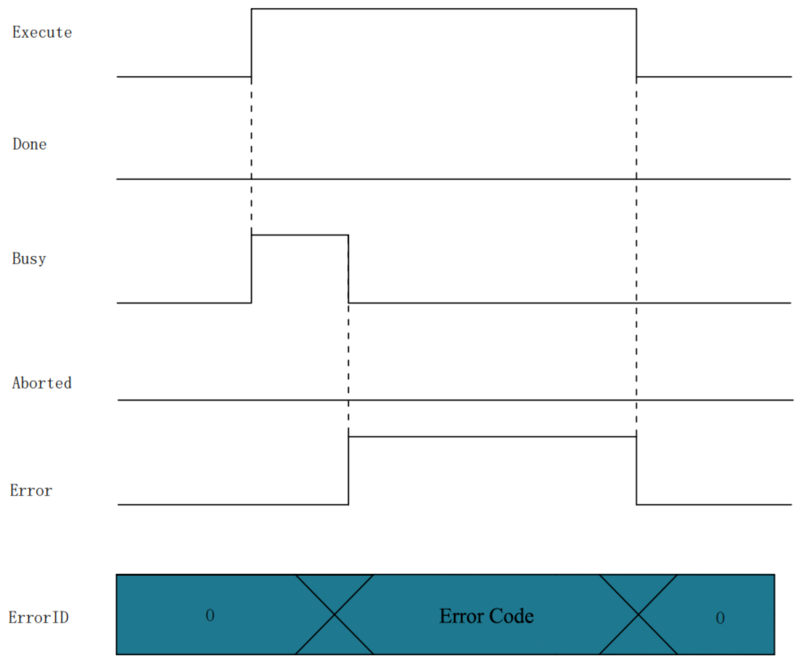
The second MC_GroupPause instruction is triggered during the execution of the first MC_GroupPause instruction, the first pause instruction is interrupted, and the interpolator starts to decelerate according to the deceleration rate of the second instruction.

Timing diagram

● When there is a deceleration process and can stop normally.

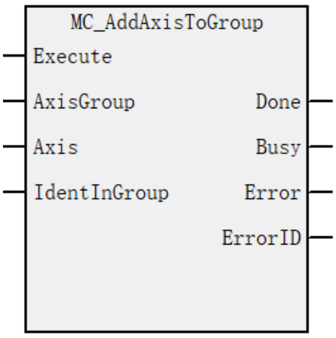


● When there is a axis failure in the deceleration process.



4.5.2.5 MC_AddAxisToGroup（Add axis to an axis group）

Drawing Block



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	AxisGroup	WORD	Constant/D/R/W	0~7	Axis group	Execute
IN	Axis	WORD	Constant/D/R/W	0~63	Axis number to be added	
IN	IdentInGroup	WORD	Constant/D/R/W	0~7	Position of this axis in the axis group	
OUT	Done	BOOL	M/S	TRUE,FALSE	Finish stopping	
OUT	Busy	BOOL	M/S	TRUE,FALSE	In-Run flag	
OUT	Error	BOOL	M/S	TRUE,FALSE	Error flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

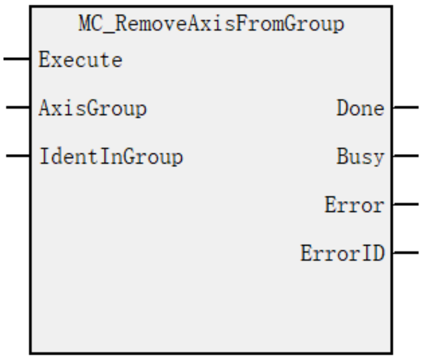
Description of functions and commands

This function block adds an axis to a group in the structure AxesGroup, valid on the rising edge.

1.When the Done bit of the command becomes TRUE, the axis is successfully added to the axis group; setting the Execute bit of the command to FALSE does not remove the axis from the axis group; if the axis needs to be removed from the axis group, the MC_RemoveAxisFromGroup command should be used. 2. IdentInGroup indicates the position of this axis in the axis group, ranging from 0 to 7. 0 represents X-axis, 1 represents Y-axis, 2 represents Z-axis, 3 represents A-axis, 4 represents B-axis, 5 represents C-axis, 6 represents H-axis, and 7 represents W-axis.

4.5.2.6 MC_RemoveAxisFromGroup（Remove axis from axis group）

Drawing Block



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	AxisGroup	WORD	Constant/D/R/W	0~7	Axis group	Execute
IN	IdentInGroup	WORD	Constant/D/R/W	0~7	Position of the axis to be removed in the axis group	
OUT	Done	BOOL	M/S	TRUE, FALSE	Finish stopping	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

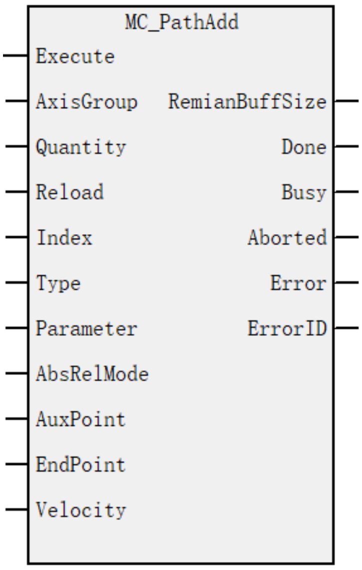
Description of functions and commands

This function block removes an axis from the group AxesGroup, valid on the rising edge.

This command is used to remove an axis from the axis group. The valid range of the input parameter IdentlnGroup is 0 to 7.

4.5.2.7 MC_PathAdd（Add path）

Drawing Block



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Descripyion	Enable
IN	AxisGroup	WORD	Constant/D/R/W	0~7	Axis group	Execute
IN	Quantity	WORD	Constan/D/R/W	Positive Number	Number of rows of data added	
IN	Reload	WORD	Constant/D/R/W	0 ： Continue	Whether to reload	

				loading 1： Reload (valid when stopped)	array type (each 1 line of data occupies one address)	
IN	Index	WORD	D/R/W	0Xffff automatically connects to the previous line number	The segment track data line number, the value of this parameter requires greater than the previous line number (starting from 0) array type (each 1 line of data occupies an address)	
IN	Type	WORD	D/R/W	0:Linear interpolation 1：CW circular arc interpolation 2: CCW circular arc interpolation 200: Termination line	Operation type array type (each 1 line of data occupies one address)	
IN	Parameter	WORD	D/R/W	Lower 8 bits. 0:x-y axis plane 1:y-z axis plane 2:x-z axis plane High 8 bits. 0:Designated as the passing point 1:Specify as center point 2:Specify as radius	Auxiliary parameters array type (each 1 line of data occupies one address)	(Only circular arc interpolation is valid, when interpolating straight lines fill in 0 can)
IN	AbsRelMode	WORD	D/R/W	0:Absolute positioning 1:Relative positioning	Absolute positioning and relative positioning modes array type (each 1 line of data occupies one address)	
IN	AuxPoint	REAL	D/R/W	Positive/Negitive	X/Y/Z axis	（仅圆弧插

				Number/0	auxiliary position arrays Array arrangement:(X, Y, Z) array type (each 1 line of data occupies 3 consecutive addresses)	补有效)
IN	EndPoint	REAL	D/R/W	Positive/Negitive Number/0	Array of target positions for each axis Array arrangement: (X, Y, Z, A, B, C, H, W) Array type (each 1 line of data occupies 8 consecutive addresses)	
IN	Velocity	REAL	D/R/W	Positive Number	Target speed Array type (each 1 line of data occupies one address)	
OUT	RemianBuffSize	WORD	D/R/W	Positive Number/0	Remaining cache size	
OUT	ErrsSerialNumber	WORD	D/R/W	Positive Number/0	The serial number where the error is located	
OUT	Done	BOOL	M/S	TRUE, FALSE	Finish stopping	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interruption	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

MC_PathAdd instruction is used to plan the axis group circular arc path function, the rising edge is valid, the edge latch data.

● The data line value must be greater than or equal to 0, but not exceed the remaining buffer size, cache size 2000.

● If the parameter of whether to reload is set to 0, the instruction execution will store the data in the buffer, and the MC_PathMov instruction will move with the data in the buffer; if the parameter is set to 1 (valid when stopping, reloading is prohibited during execution), the instruction execution will clear the data in the buffer and reload the current data.

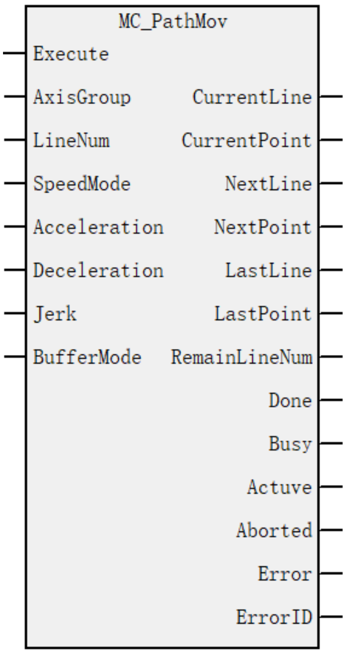
● The line number is set by the customer, but the set line number must be monotonically increasing, the line number starts from 0, the maximum is 1999, and the line number is 0xFFFF (-1) automatically connects to the number of the previous line.

● Different operation types, execute different curves, operation type set to 200 means the end of the run, execute MC_PathMov can have no termination line.

● The position parameter, which occupies a continuous floating point address, will be assigned and occupied even if the axis is not in the path planning, so please be careful when planning to prevent data address conflicts.

4.5.2.8 MC_PathMov（Path movement）

Drawing Block



Command list format

Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	AxisGroup	WORD	Constant/D/R/W	0~7	Axis group	Execute
IN	LineNum	WORD	Constant/D/R/W	0~1999（Default 0）	Line number to start execution	
IN	SpeedMode	BOOL	M/S	TRUE,FALSE （Default 0）	Speed type. 0T type, 1S type	
IN	Acceleration	REAL	Constant/D/R/W	Positive Number	Acceleration	When the path starts moving
IN	Deceleration	REAL	Constant/D/R/W	Positive Number （Default Acceleration）	Deceleration	When the path ends movement
IN	Jerk	REAL	Constant/D/R/W	Positive Number	Speed stp	
IN	BufferMode	WORD	Constant/D/R/W	Bit15 is 1, no look-ahead is turned on. Bit15 is 0 to turn on	Buffer mode	

				forwardlooking. (Default 0)		
OUT	CurrentLine	WORD	D/R/W	0~1999 No execution line at 0xFFFF	Line number of current execution	
OUT	CurrentPoint	REAL	D/R/W	Positive/Negitive Number/0	Target position of the current row Array alignment: (X, Y, Z, A, B, C, H, W)	
OUT	NextLine	WORD	D/R/W	0~1999 No next line at 0xFFFF	Line number of the next execution	
OUT	NextPoint	REAL	D/R/W	Positive/Negitive Number/0	Target position of the next row Array arrangement: (X, Y, Z, A, B, C, H, W)	
OUT	LastLine	WORD	D/R/W	0~1999	Line number of the previous execution	
OUT	LastPoint	REAL	D/R/W	Positive/Negitive Number/0	Target position of the previous row Array arrangement: (X, Y, Z, A, B, C, H, W)	
OUT	RemainLineNum	WORD	D/R/W	1~2000	Number of lines left to execute	
OUT	Done	BOOL	M/S	TRUE, FALSE	Finish stopping	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Active	BOOL	M/S	TRUE, FALSE	ON at the start of curve execution	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interruptions	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error flag	

OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Description of functions and commands

MC_PathMov instruction is used to start the path motion function, the rising edge is valid, the edge latch data.

● The starting row of the path movement can be customized.

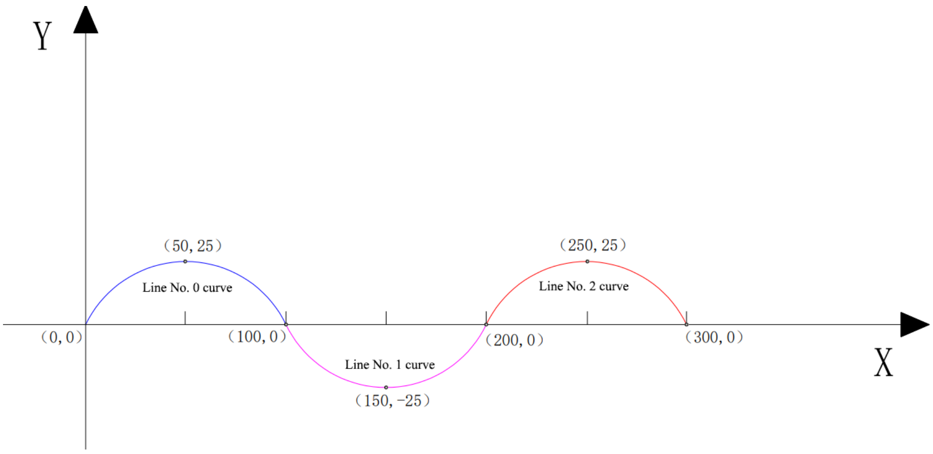
● The MC_PathMov command differs from other motion commands in that it is influenced by the look-ahead, the articulation between curves is more rounded, the look-ahead is activated, and the transition parameters between curves are calculated automatically according to the path planning.

● The MC_PathMov instruction can be paused by the MC_GroupPause instruction.

● The data that will be run, the interface will only show one line of data, but it will actually take up more registers afterwards, so please avoid it when planning to prevent data conflicts.

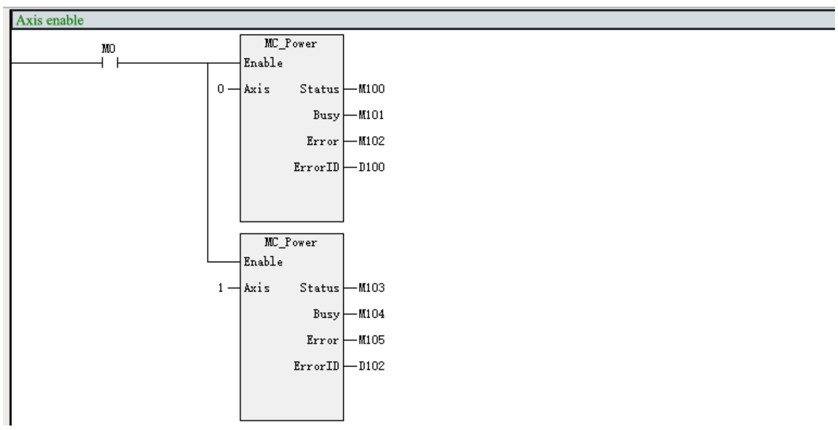
Application example 1

Add 4 lines of data (including the termination line), all XY plane mode circular arc (designated as the passing point), the



effect of the following schematic.

Program examples:



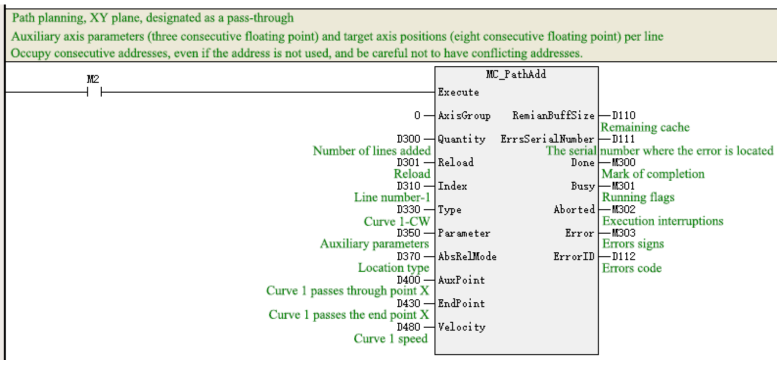
Three circular curves

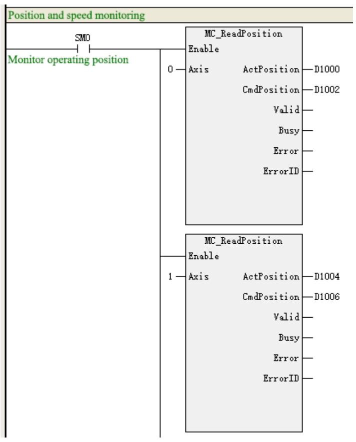
SM1
Initial pulse running bit

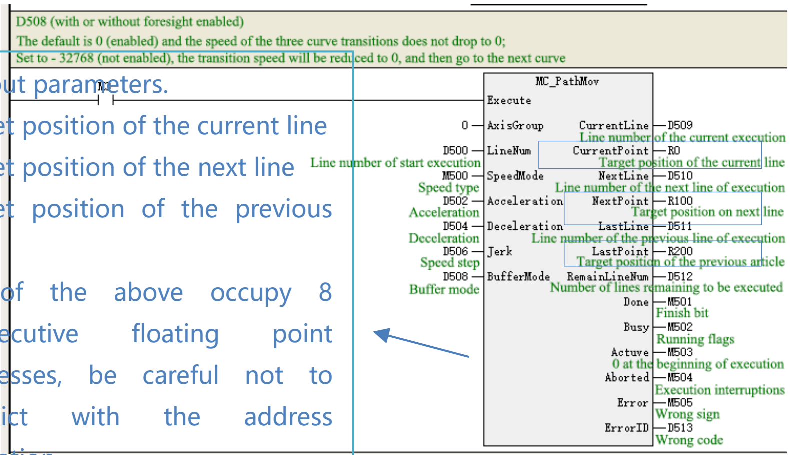
Instruction	Source	Destination	Comment
MOV	4	D300	Number of lines added
MOV	1	D301	Reload
MOV	0	D310	Line number-1
MOV	1	D311	Line number-2
MOV	2	D312	Line number-3
MOV	3	D313	End line number
MOV	1	D330	Curve 1-CW
MOV	2	D331	Curve 2-CCW
MOV	1	D332	Curve 3-CW
MOV	200	D333	Termination line
RMOV	50.00000	D400	Curve 1 passes through point X
RMOV	25.00000	D402	Curve 1 passes through point Y
RMOV	150.0000	D406	Curve 2 passes through point X
RMOV	-25.0000	D408	Curve 2 passes through point Y
RMOV	250.0000	D412	Curve 3 passes through point X
RMOV	25.00000	D414	Curve 3 passes through point Y
RMOV	100.0000	D430	Curve 1 end point X
RMOV	0.000000	D432	Curve 1 end point Y
RMOV	200.0000	D446	Curve 2 end point X
RMOV	0.000000	D448	Curve 2 end point Y
RMOV	300.0000	D462	Curve 3 end point X
RMOV	0.000000	D464	Curve 3 end point Y
RMOV	20.00000	D480	Curve 1 speed
RMOV	20.00000	D482	Curve 2 speed
RMOV	20.00000	D484	Curve 3 speed

Each line of the auxiliary axis position parameter occupies 3 consecutive floating point addresses, even if an axis is not used!

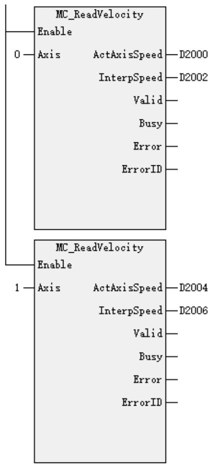
Each line of the axis target position parameter occupies 8 consecutive floating point addresses, even if the axis is not used, the address is still occupied!



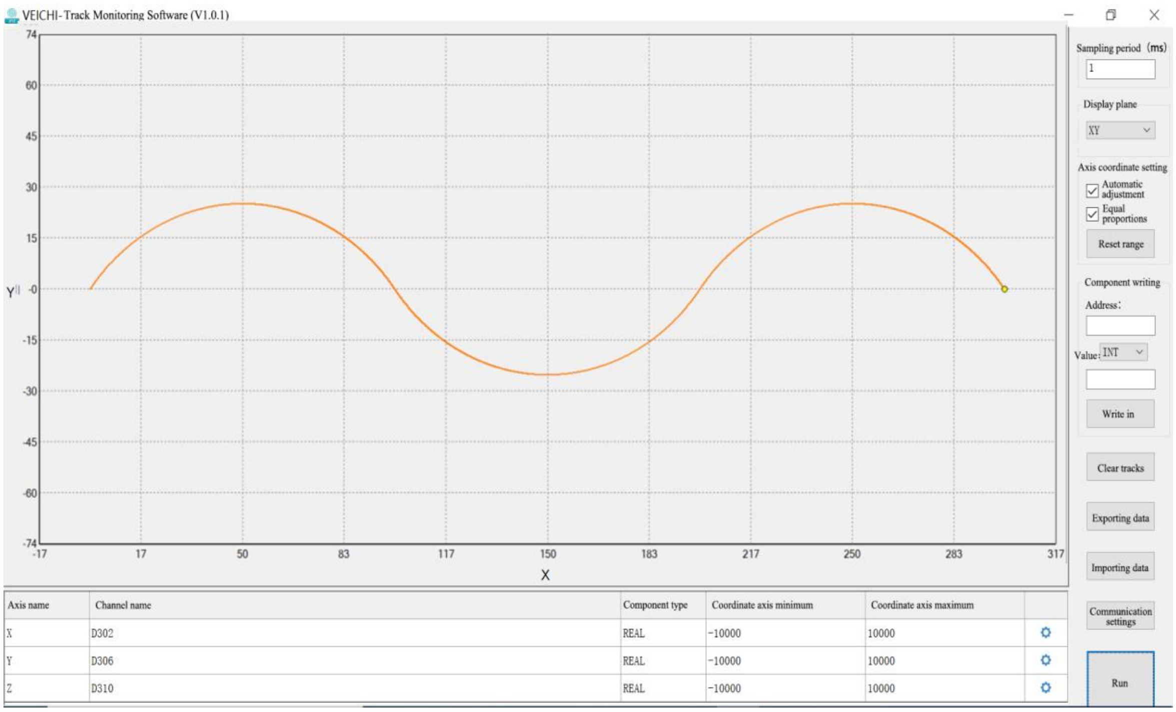


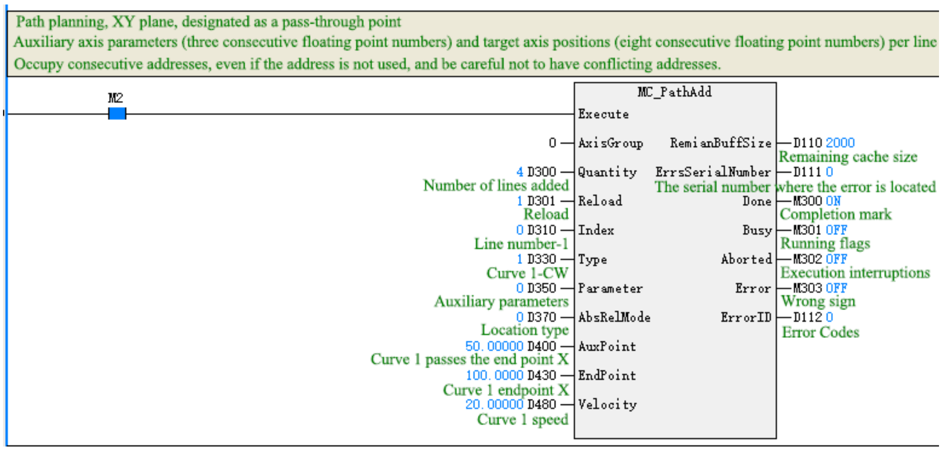


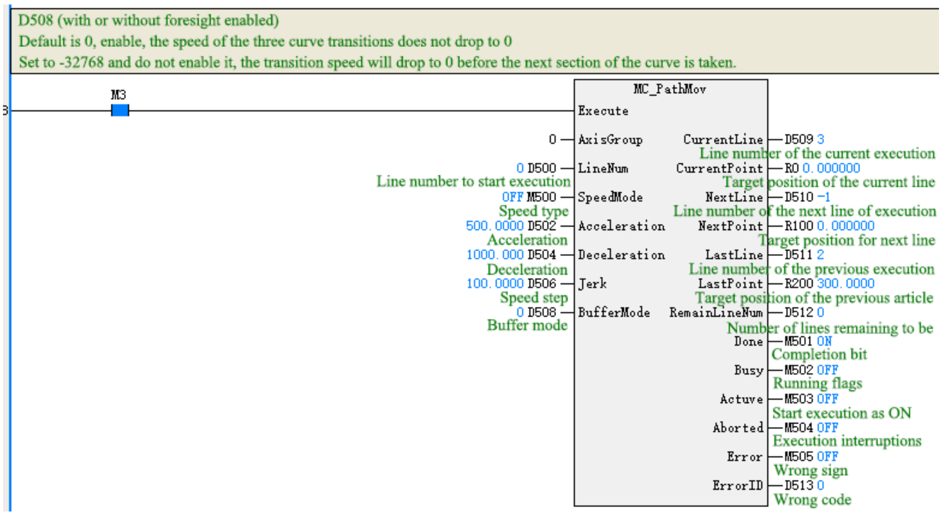
Output parameters.
Target position of the current line
Target position of the next line
Target position of the previous line
All of the above occupy 8 consecutive floating point addresses, be careful not to conflict with the address allocation



Track and parameter monitoring chart：



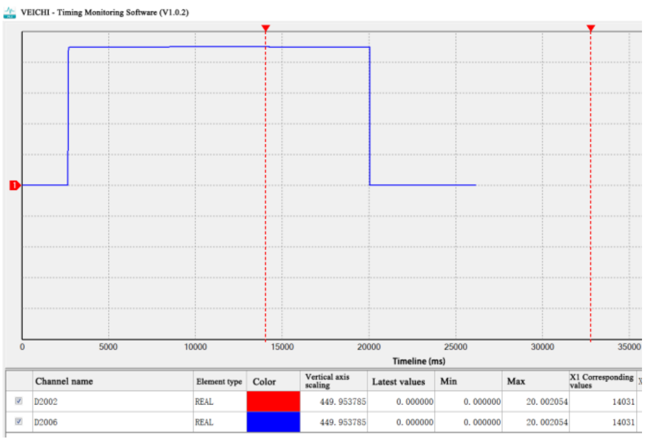




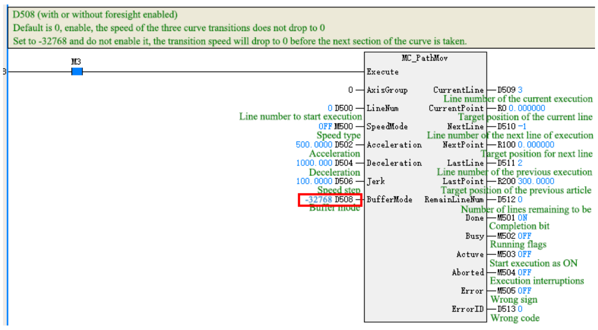
Forwardlooking parameters BufferMode

The trajectory is the same with and without forwardlooking, but the velocity profile will be different。

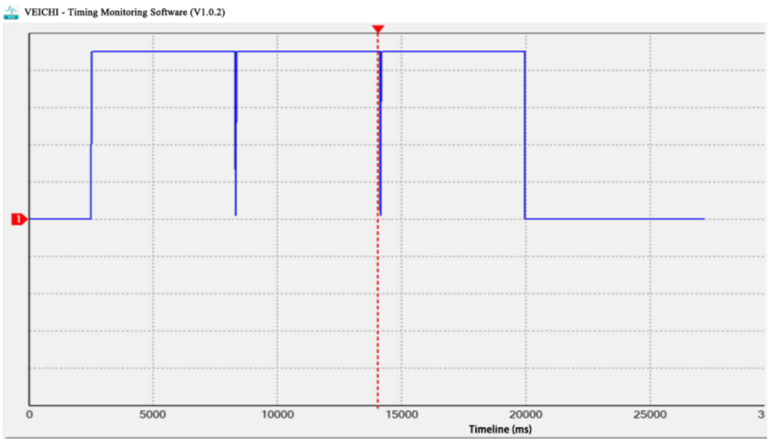
BufferMode =0， Turn on the foresight, the transition parameters between the curves will be calculated automatically according to the path planning, and the speed of the curve transition phase will not drop to 0. The above example turns on the forwardlooking, the speed of the three sections is the same, and its speed curve is as follows.



BufferMode =-32768， Without forwardlooking on, the speed of the curve transition phase will be reduced to 0. The above example is without forwardlooking on，



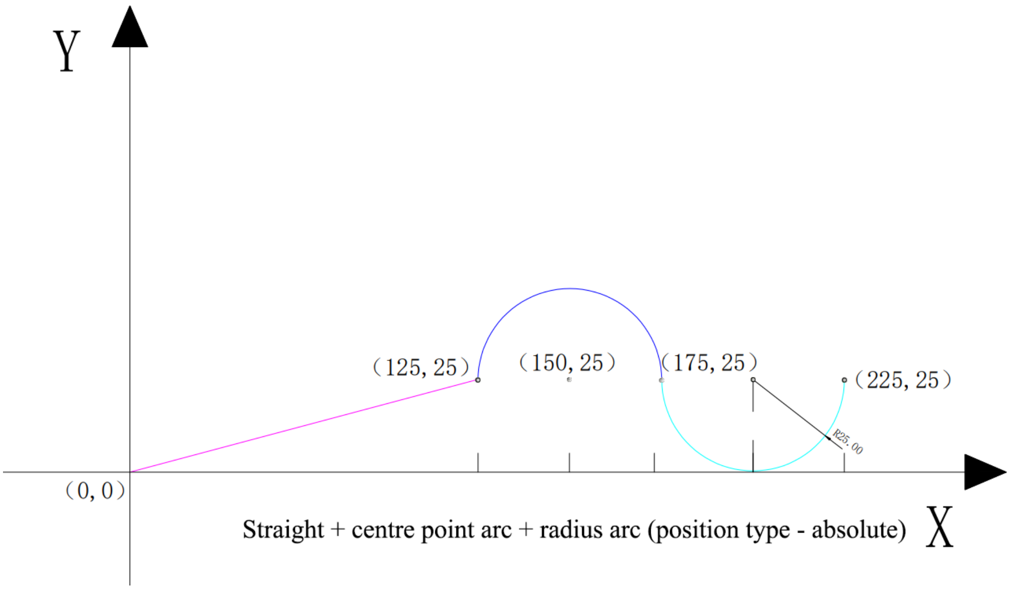
Its speed curve is as follows：



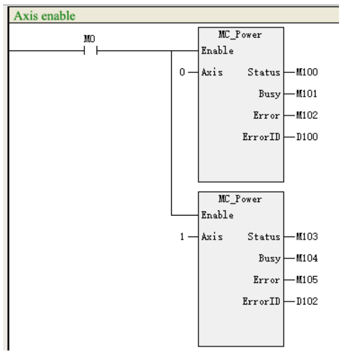
	Channel name	Element type	Color	Vertical axis scaling	Latest values	Min	Max	X1 Corresponding values
☑	D2002	REAL		450.000000	0.000000	0.000000	20.002054	14031
☑	D2006	REAL		450.000000	0.000000	0.000000	20.002054	14031

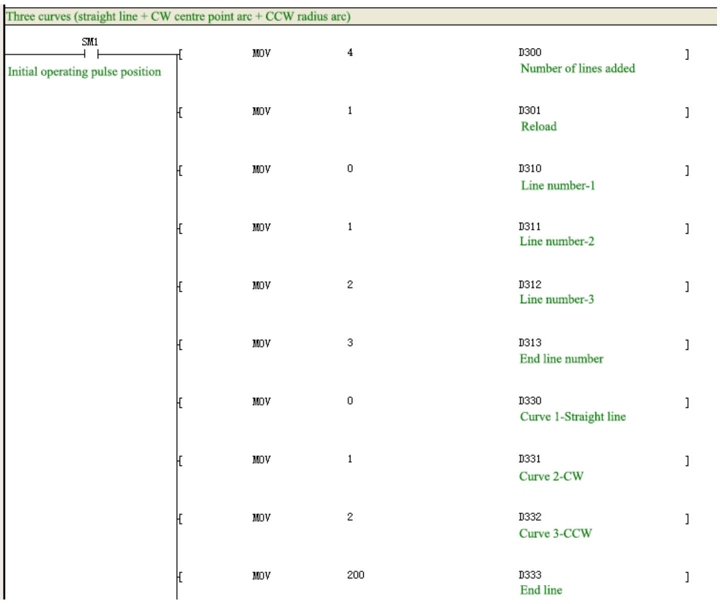
Application example 2

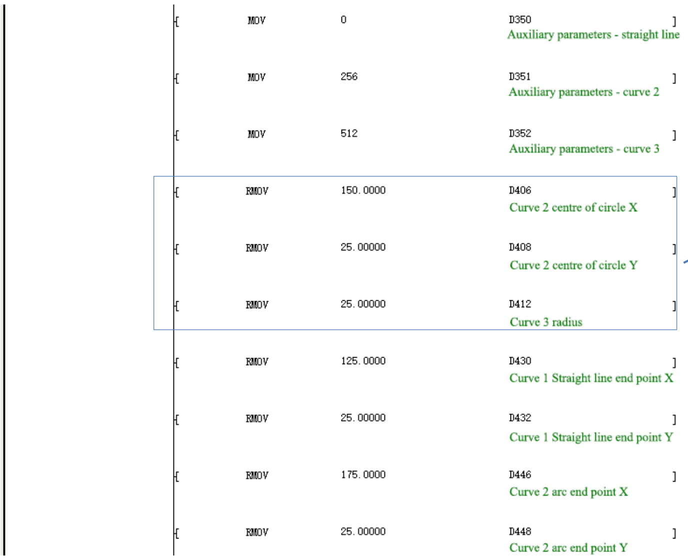
Add 4 lines of data (including the termination line), XY plane mode, straight line + CW center point arc + CCW radius arc effect schematic as follows.



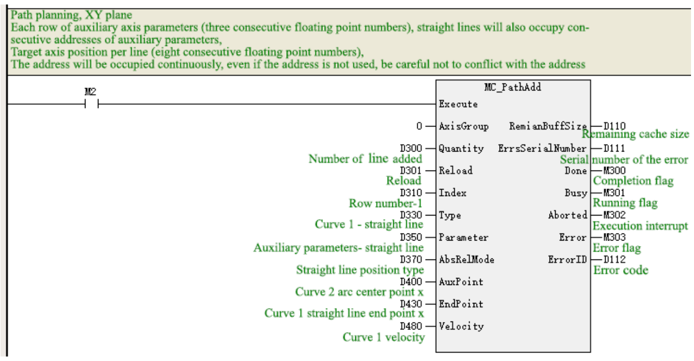
Program examples:

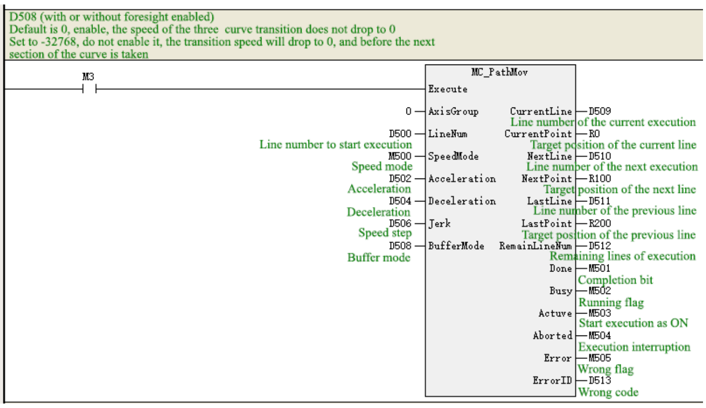


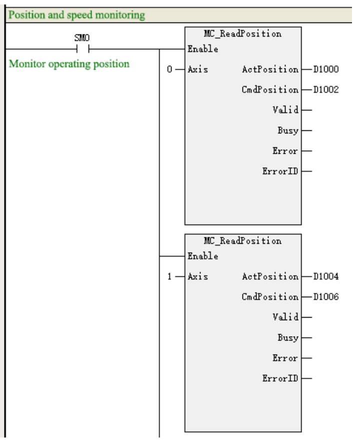


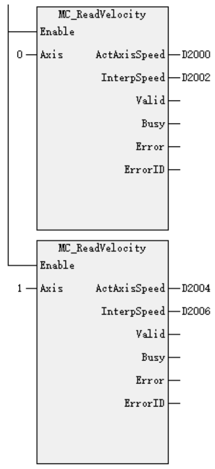


Auxiliary position parameter starting address D400
The first curve is a straight line, the auxiliary position parameter does not work, but the address is still occupied, so pay attention to the empty occupied address when setting parameters!

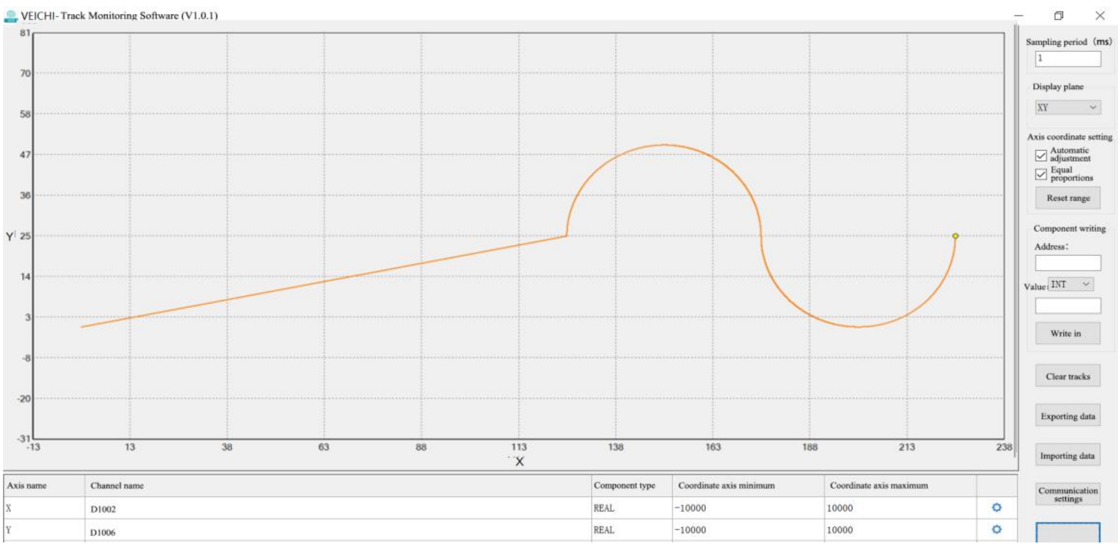


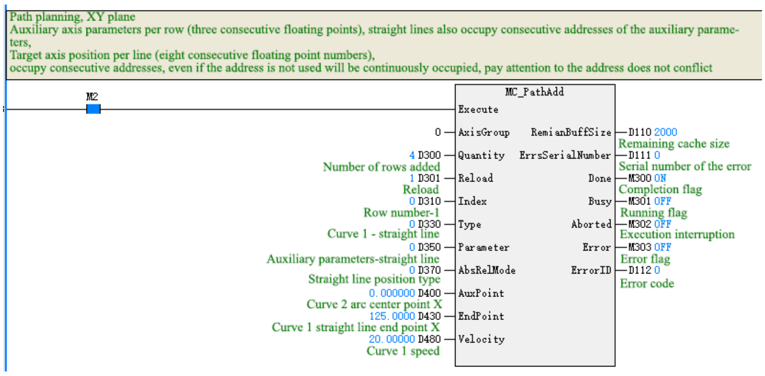


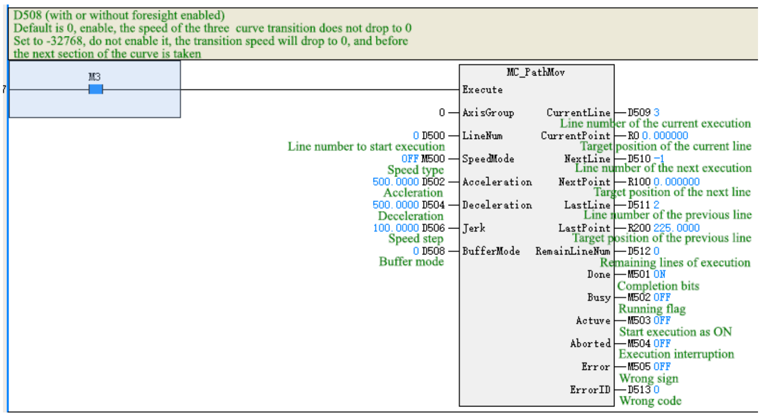




Track and parameter monitoring chart:

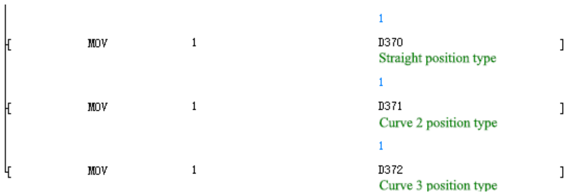


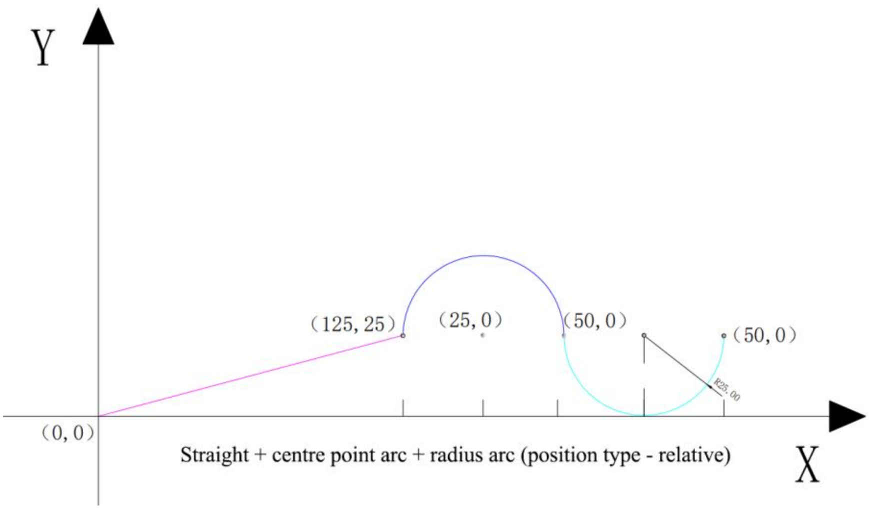




Position Type AbsRelMode

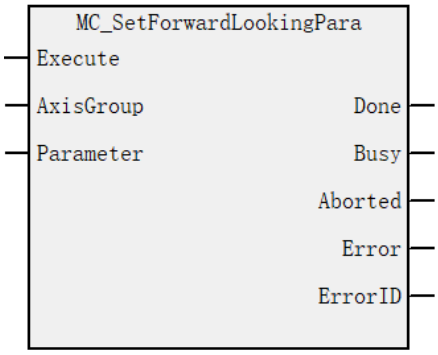
The above is the curve according to the absolute position type AbsRelMode=0 coordinates, if you choose the relative position type AbsRelMode=1, you can change the relative coordinates, the curve track is the same.





4.5.2.9 MC_SetForwardlookingPara（Forwardlooking parameter setting command）

Drawing Block



Command list format

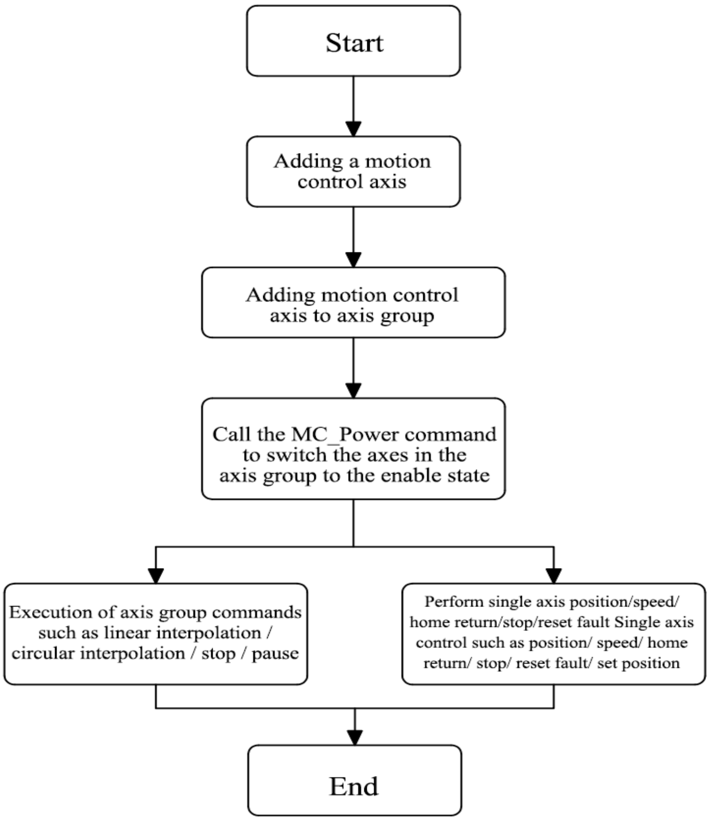
Input/Output	Name	Data Type	Applicable (soft components)	Range	Description	Enable
IN	AxisGroup	WORD	Constant/D/R/W	0~7	Axis group	Execute
IN	Parameter	REAL	D/R/W		Parameter start address	
OUT	Done	BOOL	M/S	TRUE, FALSE	Finish stopping	
OUT	Busy	BOOL	M/S	TRUE, FALSE	In-Run flag	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interruption	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Note: This command is temporarily unavailable and reserved.

4.6 Axis group interpolation operation

4.6.1 Overview

To execute the interpolation instruction correctly, first create the axis group and enable the axes within the axis group, the basic process is as follows：



Interpolation operation flow chart

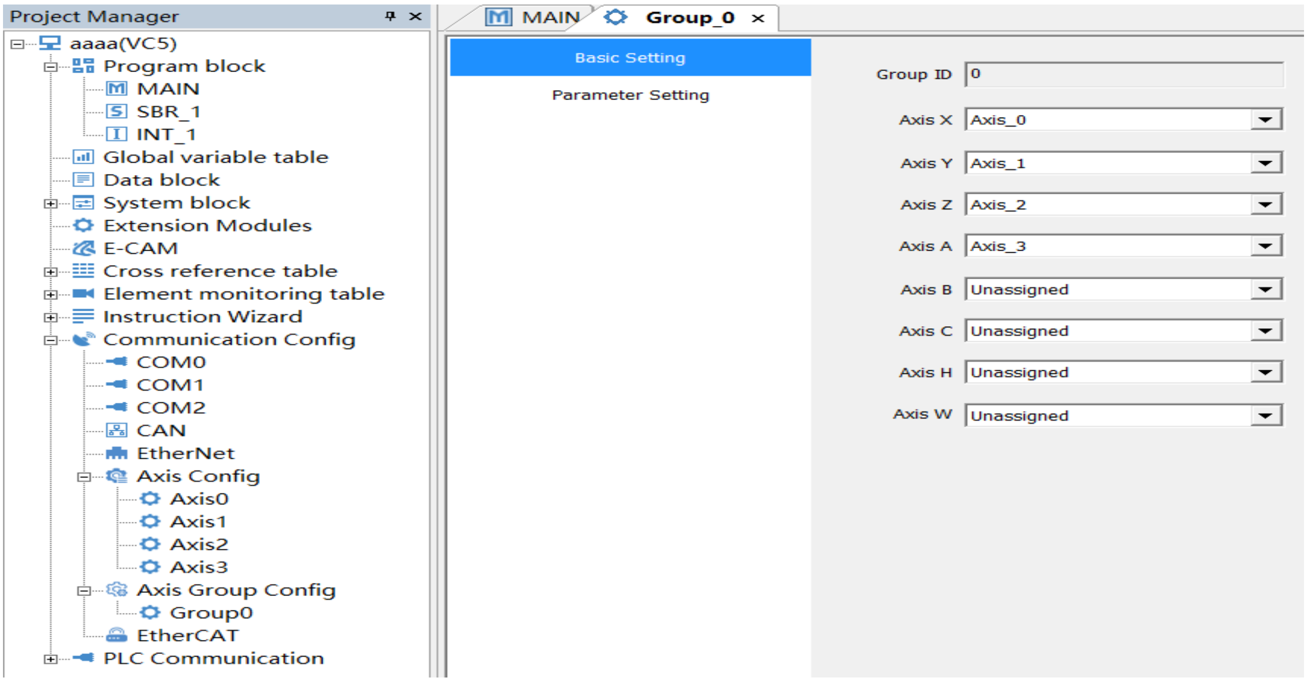
Description

Even after creating an axis group, the axis in the group can still execute single-axis motion and control commands, but the single-axis motion commands and the interpolation commands of the axis group are mutually exclusive and cannot be interrupted at the same time.

This section introduces the basic interpolation procedure in the form of a routine to combine Axis_0, Axis_1, Axis_2 and Axis_3 into an axis group and perform the related actions.

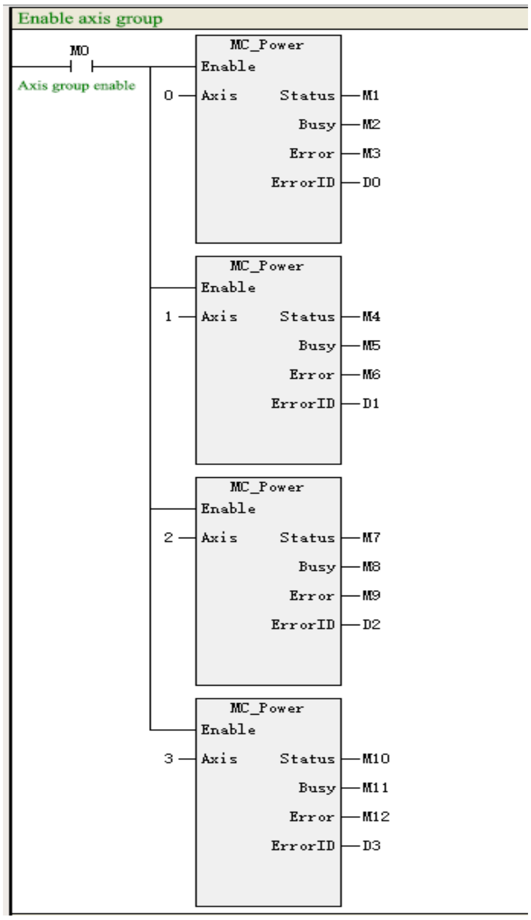
4.6.2 Creating axis groups

Right-click on "Axis Group Configuration", select "Add Axis Group", then you can select the axis and auxiliary axis and set the relevant parameters after creating the axis group.



4.6.3 Enabling Axis Group

Each individual axis in an axis group is enabled and de-enabled by the MC_Power instruction. Only when all axes in an axis group are enabled, the axis group instruction can be executed.

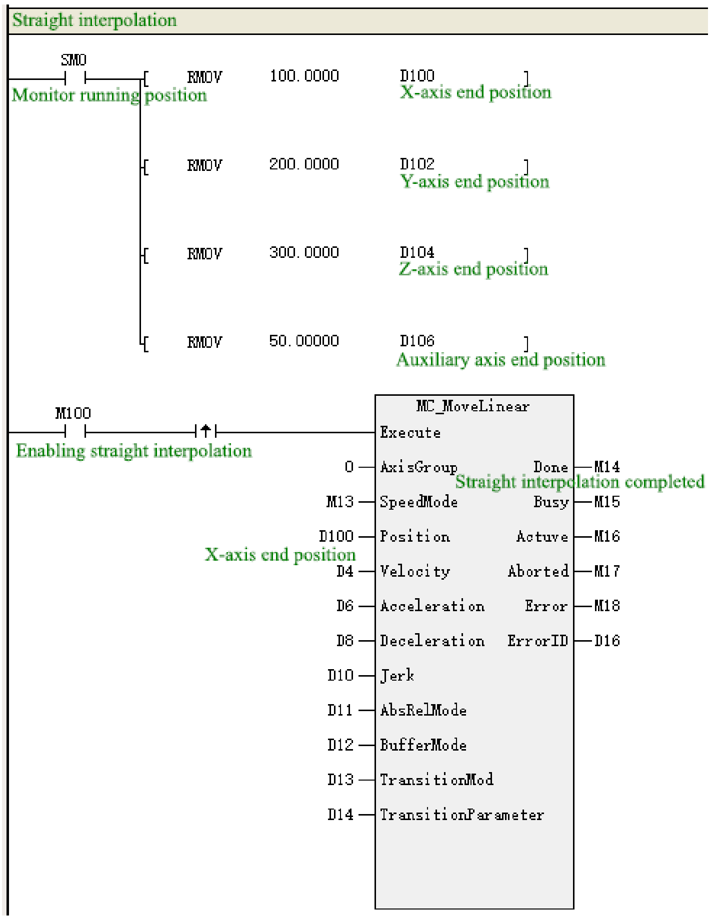


4.6.4 Linear interpolation

The linear interpolation function of the axis group is implemented by the instruction MC_MoveLinear.
When all axis groups are in the StandStill state, the Execute axis group is triggered to start linear interpolation, and all axis groups switch to the Synchronized Motion state. The execution of single-axis motion commands such as MC_MoveAbsolute, MC_Stop, etc. is not allowed at this time.
After linear interpolation is completed, all axis groups return to the StandStill state, and single-axis motion commands such as "MC_MoveAbsolute" and "MC_Stop" are allowed to be executed again.

Example

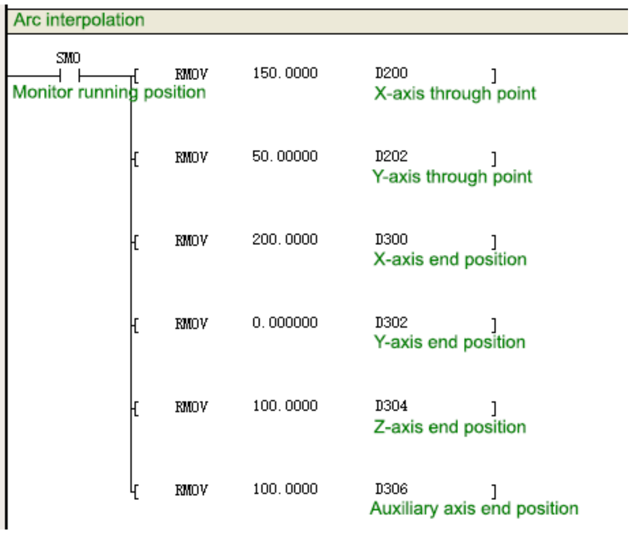
This routine uses absolute positioning to position the x-axis, y-axis, and z-axis to (100,200,300), and the auxiliary axis to 50.

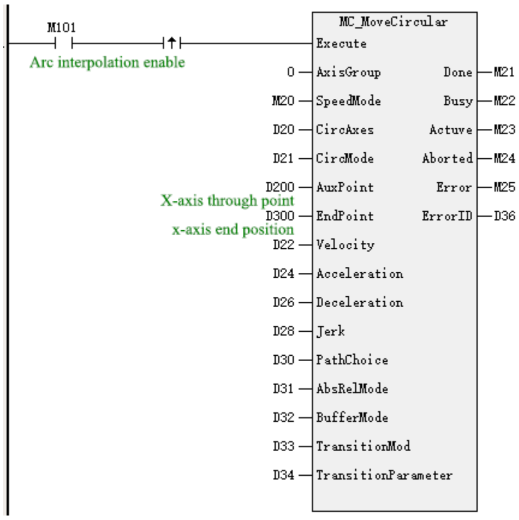


4.6.5 Circular arc interpolation

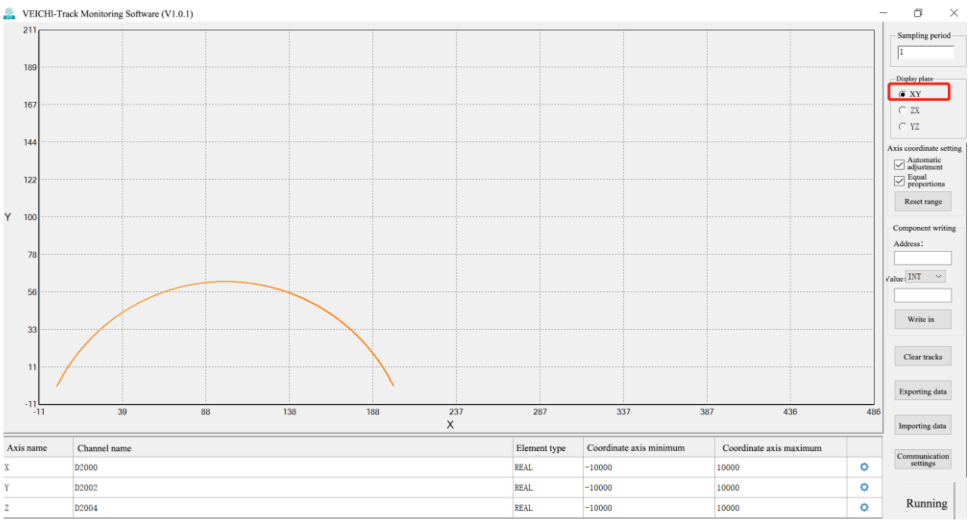
The circular arc interpolation function of the axis group is implemented by the instruction MC_MoveCircular. the transition rules of the PLCOpen state machine are the same as linear interpolation.
This routine performs circular arc interpolation in the XY-axis plane while the z-axis and auxiliary axes perform simultaneous linear motion. The circular interpolation uses the pass point mode, absolute positioning, first pass point (150,50) and then reach the position (200,0). z-axis and auxiliary axis reach the position 100.

Example

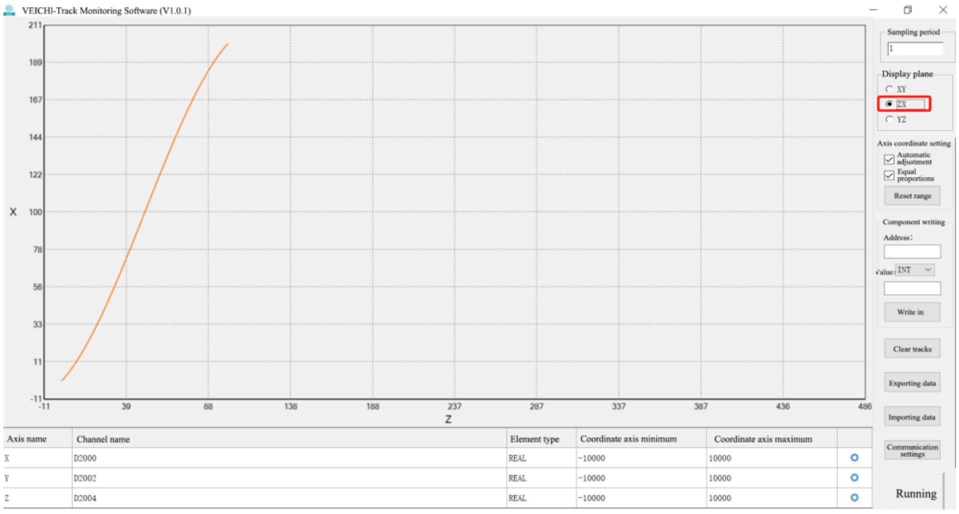




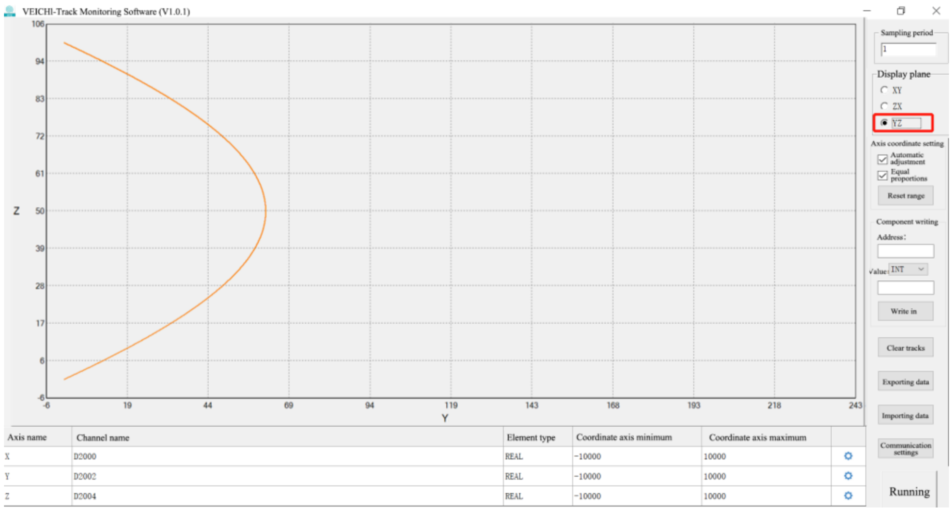
Floor plan monitoring map



X-Y plane



Z-X plane



Y -Z plane

4.6.6 Axis group stop

The execution of the interpolation curve is stopped by the instruction MC_GroupStop.在 Execute 的上 The rising edge interrupts the execution of the interpolation instruction, and the CommandAborted output of the interpolation instruction is valid.

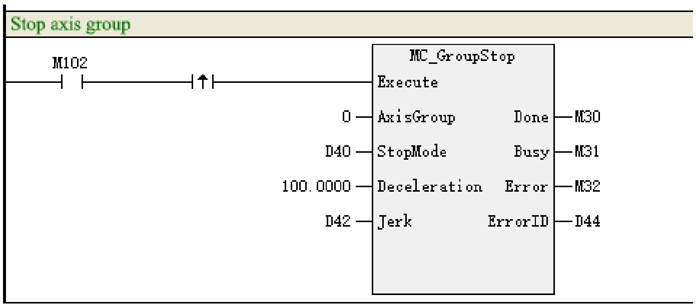
The interpolation instruction triggered with Execute=TRUE is invalid, and Execute must be set to False to re-execute the new interpolation instruction.

This command is allowed to be called only when all axes in the axis group are in StandStill or Synchronized Motion state, and the axes are in Synchronized Motion state while the Execute of the command is in effect.

MC_GroupStop can only stop the operation of interpolation curves, not single-axis motion commands (e.g. MC_MoveAbsolute).

Example

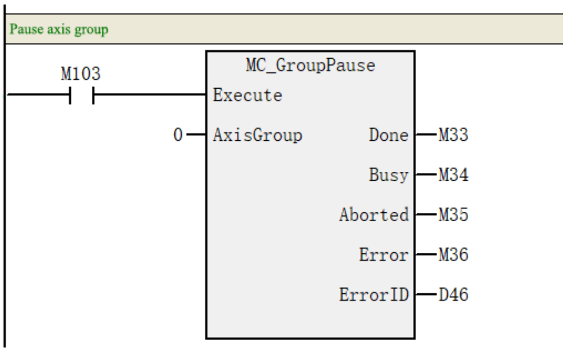
In this example, the following command is called to stop the interpolation curve during linear interpolation or circular interpolation, using the deceleration stop mode, and the stop deceleration speed is 100.



4.6.6 Pause axis group

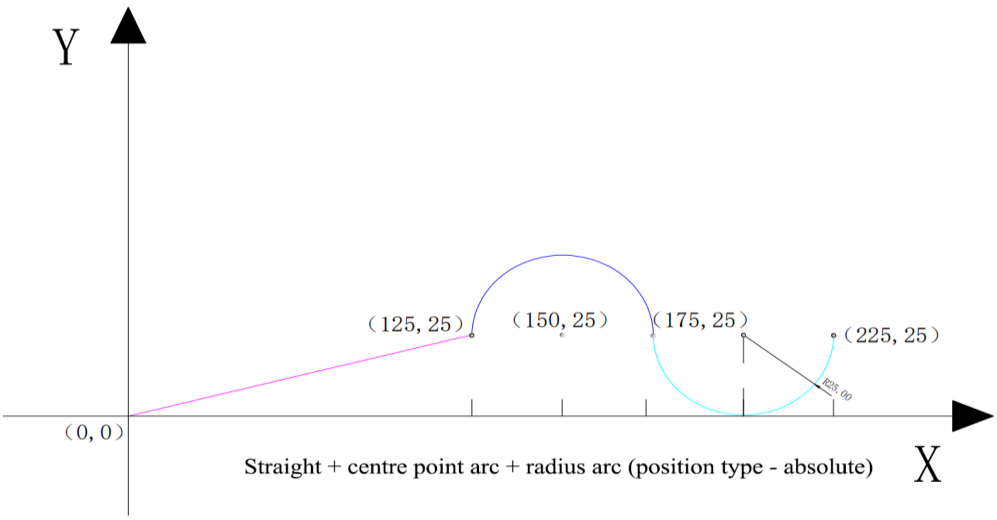
The interpolation curve is controlled by the command MC_GroupPause to pause.
Pause the interpolation curve when Enable=TRUE, resume the execution of the interpolation curve when Enable=False.
MC_GroupPause can only pause the interpolation curve, and cannot stop single-axis type motion commands (such as MC_MoveAbsolute).

Example



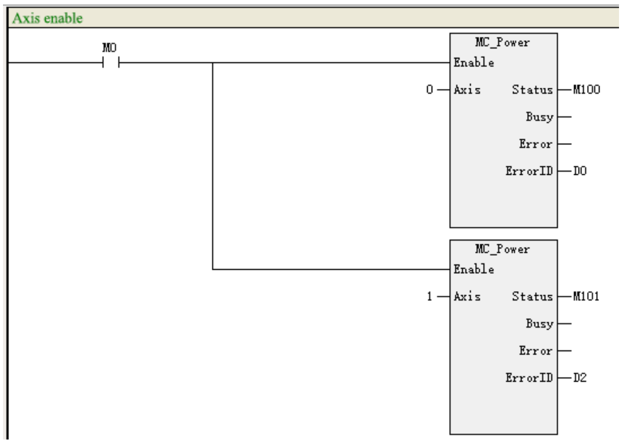
4.6.7 Continuous interpolation example

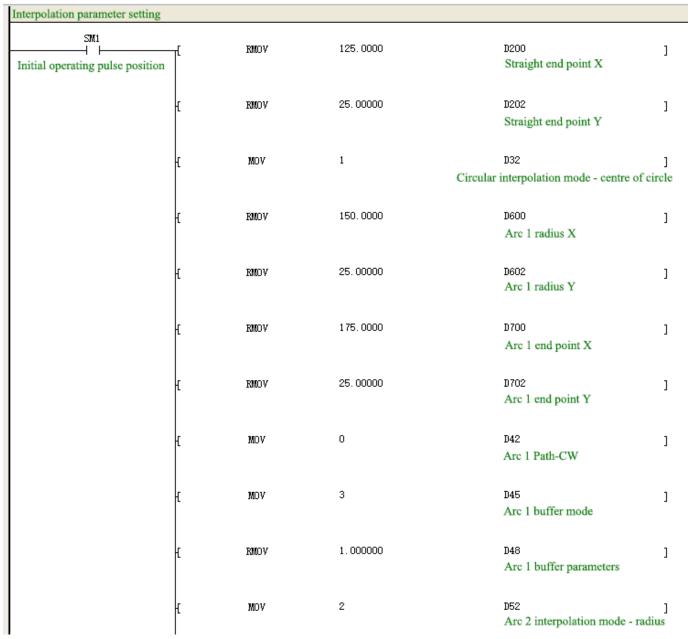
XY plane mode, straight line + CW center point arc + CCW radius circular arc effect diagram as follows：

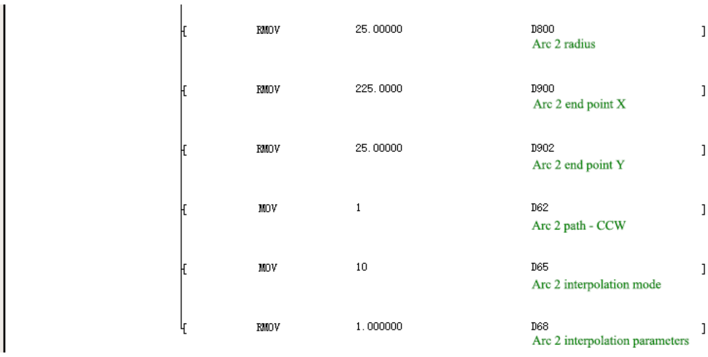


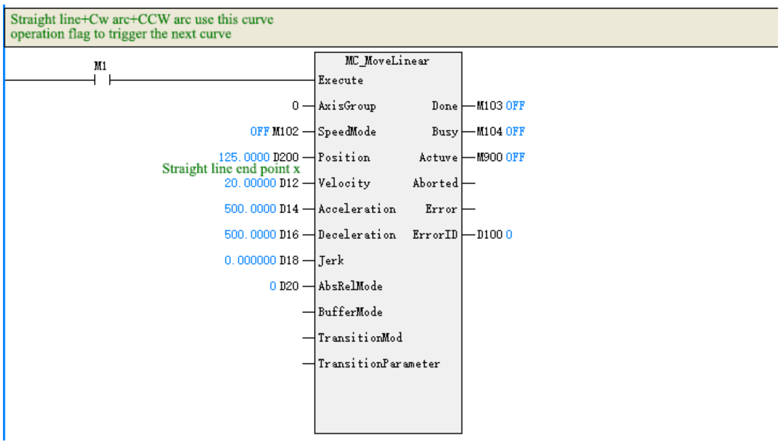
Straight + centre point arc + radius arc (position type - absolute)

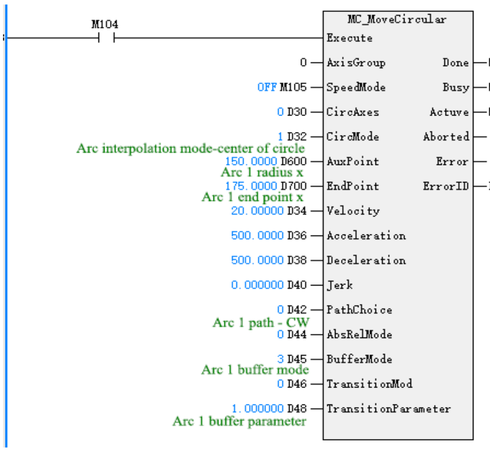
Program example：

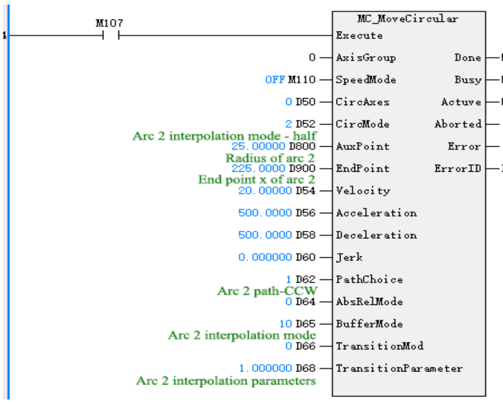


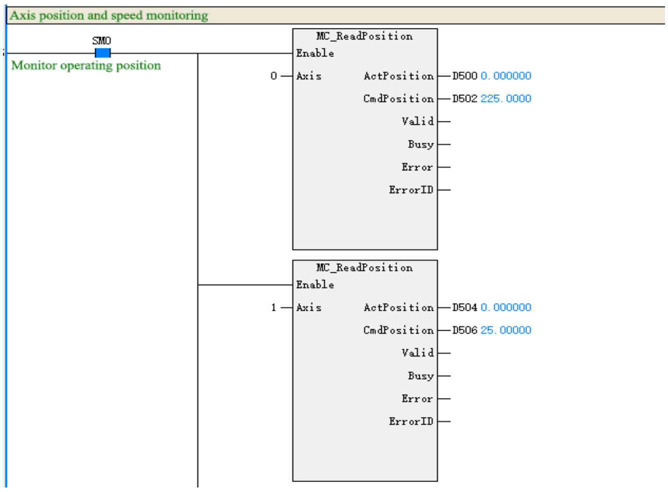




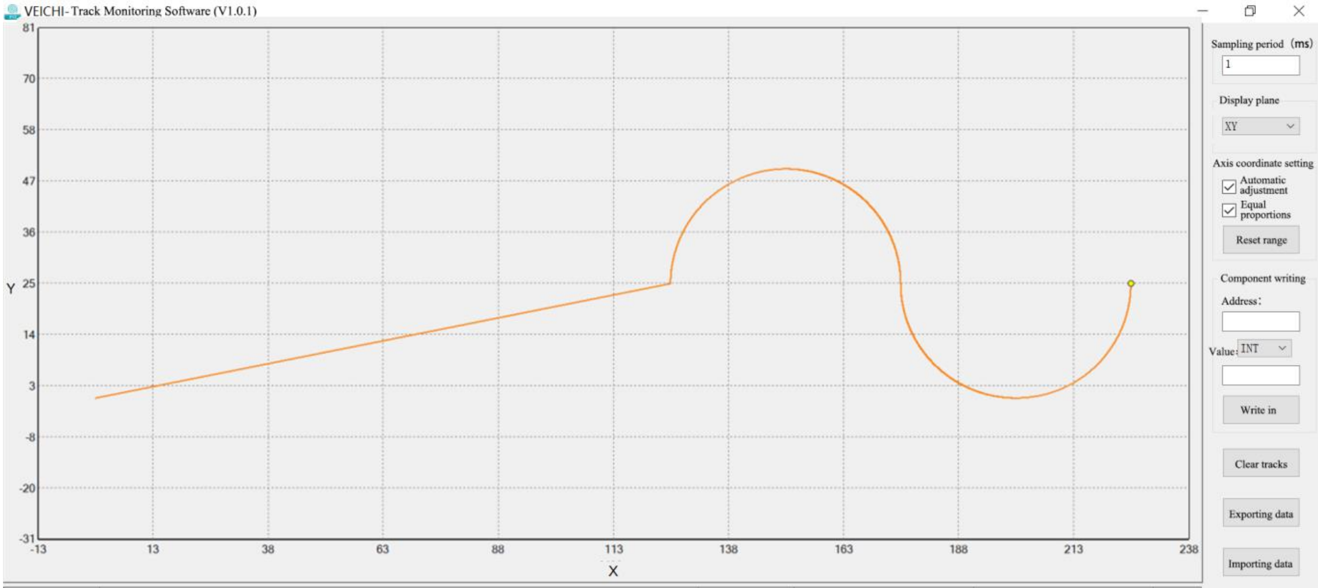

Straight line+Cw arc+CCW arc use this curve
operation flag to trigger the next curve
M1
MC_MoveLinear
Execute
0 AxisGroup Done M103 OFF
OFF M102 SpeedMode Busy M104 OFF
125.0000 D200 Position Actuve M900 OFF
Straight line end point x
20.00000 D12 Velocity Aborted
500.0000 D14 Acceleration Error
500.0000 D16 Deceleration ErrorID D100 0
0.000000 D18 Jerk
0 D20 AbsRelMode
BufferMode
TransitionMod
TransitionParameter


M104
MC_MoveCircular
Execute
0 AxisGroup Done M106 OFF
OFF M105 SpeedMode Busy M107 OFF
0 D30 CircAxes Actuve M901 OFF
1 D32 CircMode Aborted
Arc interpolation mode-center of circle
150.0000 D600 AuxPoint Error
Arc 1 radius x
175.0000 D700 EndPoint ErrorID D102 0
Arc 1 end point x
20.00000 D34 Velocity
500.0000 D36 Acceleration
500.0000 D38 Deceleration
0.000000 D40 Jerk
0 D42 PathChoice
Arc 1 path - CW
0 D44 AbsRelMode
3 D45 BufferMode
Arc 1 buffer mode
0 D46 TransitionMod
1.000000 D48 TransitionParameter
Arc 1 buffer parameter


M107
MC_MoveCircular
Execute
0 AxisGroup Done M111 OFF
OFF M110 SpeedMode Busy M112 OFF
0 D50 CircAxes Actuve M902 OFF
2 D52 CircMode Aborted
Arc 2 interpolation mode - half
25.00000 D800 AuxPoint Error
Radius of arc 2
225.0000 D900 EndPoint ErrorID D104 0
End point x of arc 2
20.00000 D54 Velocity
500.0000 D56 Acceleration
500.0000 D58 Deceleration
0.000000 D60 Jerk
1 D62 PathChoice
Arc 2 path-CCW
0 D64 AbsRelMode
10 D65 BufferMode
Arc 2 interpolation mode
0 D66 TransitionMod
1.000000 D68 TransitionParameter
Arc 2 interpolation parameters


Axis position and speed monitoring
SM0
Monitor operating position
MC_ReadPosition
Enable
0 Axis ActPosition D500 0.000000
CmdPosition D502 225.0000
Valid
Busy
Error
ErrorID
MC_ReadPosition
Enable
1 Axis ActPosition D504 0.000000
CmdPosition D506 25.00000
Valid
Busy
Error
ErrorID

Track monitoring map：



Axis name	Channel name	Component type	Coordinate axis minimum	Coordinate axis maximum
X	D502	REAL	-10000	10000
Y	D506	REAL	-10000	10000

5. Local High-Speed Counter Axis

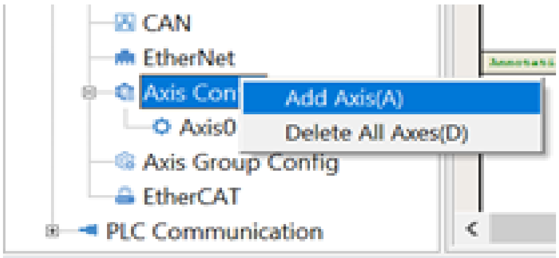
5.1 Introduction to high-speed counter axis

The counter is implemented in the form of encoder axis in Auto Studio software and engineering applications, and the counter is collectively referred to as the counter axis after being associated with the axis. VC5 supports 8-axis 32-bit high-speed counter, which can realize AB-phase 1/2/4 frequency multiplication, CW/CCW, pulse + direction and single-phase counting. The counting signal source can choose external pulse input or internal 1ms/1us clock counting; Other input signals can realize the preset and latch functions of the counter.

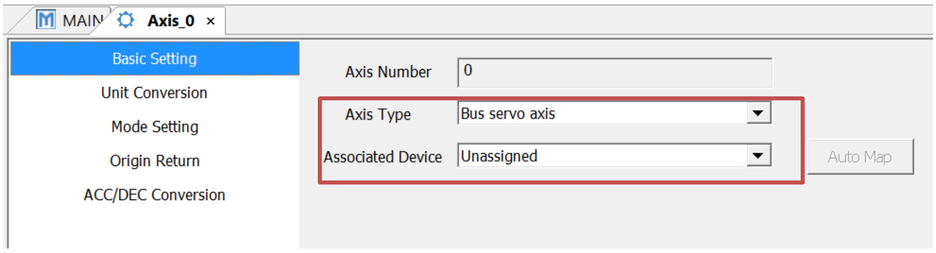
5.2 High-Speed Counter Axiscreate

Before using the counter in the Auto Studio programming software, it is necessary to associate the counter with the axis.

1. In "Project Managementdevice"bar, right-click"axisMotion Control Axis under Configuration, select Add Axis to create a motion control axis.

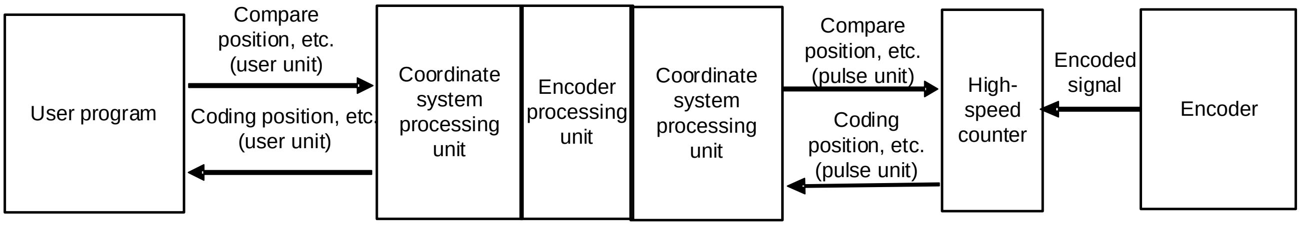


2. Double click on the newly added axis, Open the setting page, select "local encoder axis" as the axis type on the "basic setting" interface, and select "high-speed counter" as the input device to associate the axis and the counter. The axis number is used in the program as the axis identifier to realize the control of the corresponding counter axis.



5.3 Counter axis user unit and conversion

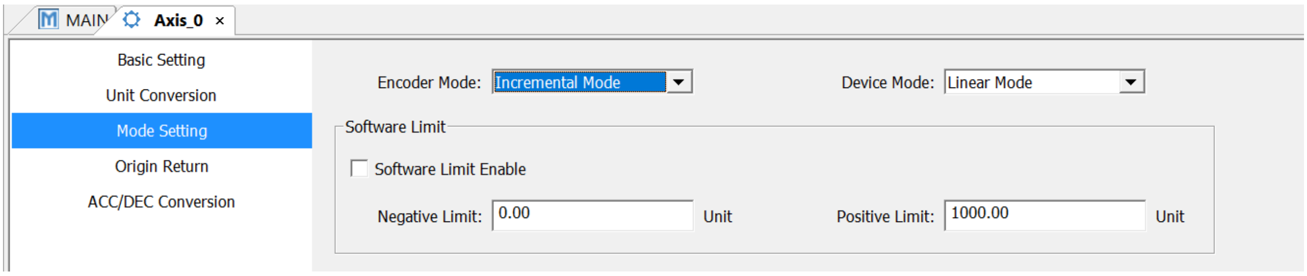
The high-speed counter uses the pulse unit when decoding the encoder signal, and the counter command uses the common measurement units such as millimeters, degrees, inches, etc., which we call the user unit (Unit). The number of pulses can be converted into a user unit (Unit) through unit conversion, and the user unit (Unit) is based on the actual application. Use can be defined as device-dependent units (millimeters, revolutions, etc.).



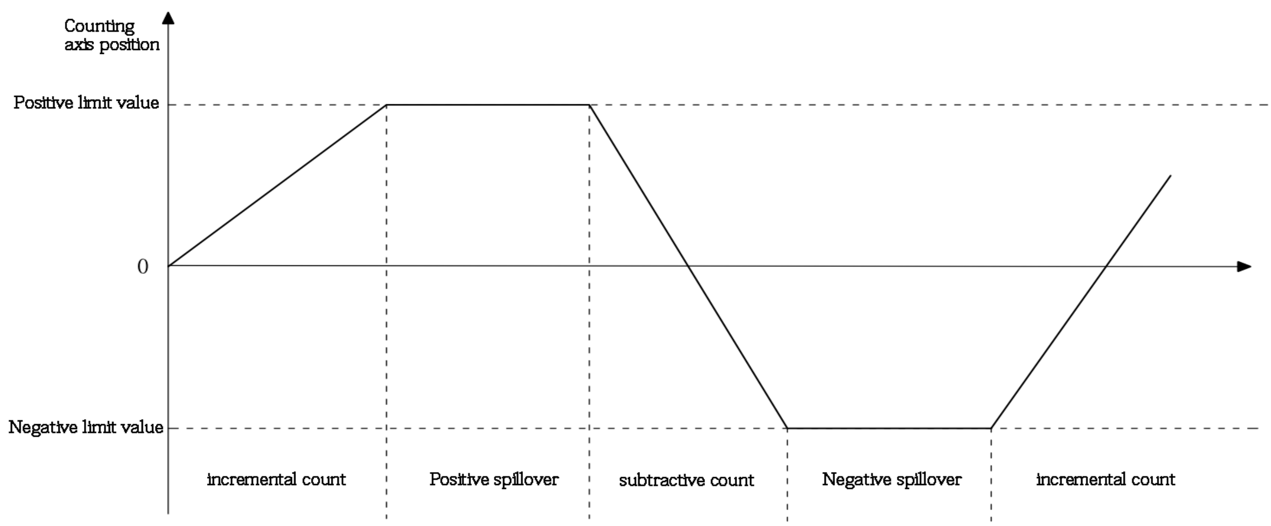
5.4 Set Working Mode

5.4.1 Linear mode

The position of the counter axis changes between the negative limit value and the positive limit value. After the position of the counter axis reaches the limit value, continue to input the same-direction pulse; the counter axis reports overflow, and the position of the counter axis remains unchanged. After the counter axis reports overflow, input the reverse pulse, the counter axis counts in the reverse direction, and the overflow error is cancelled. In linear mode, the negative and positive position limit values of the counter axis can be set in the interface, and the position unit is the user unit (Unit). The negative limit must be less than or equal to 0, and the positive limit must be greater than or equal to 0. Since the high-speed counter is a 32-bit counter, the negative limit value and the positive limit value must be in the range of 32-bit integers [-2147483648, 2147483647] after they are converted into pulse units.

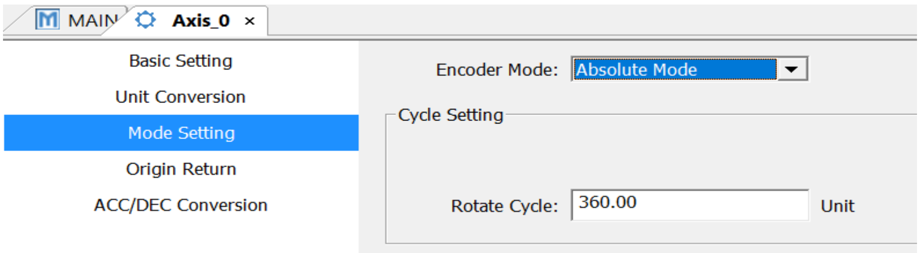


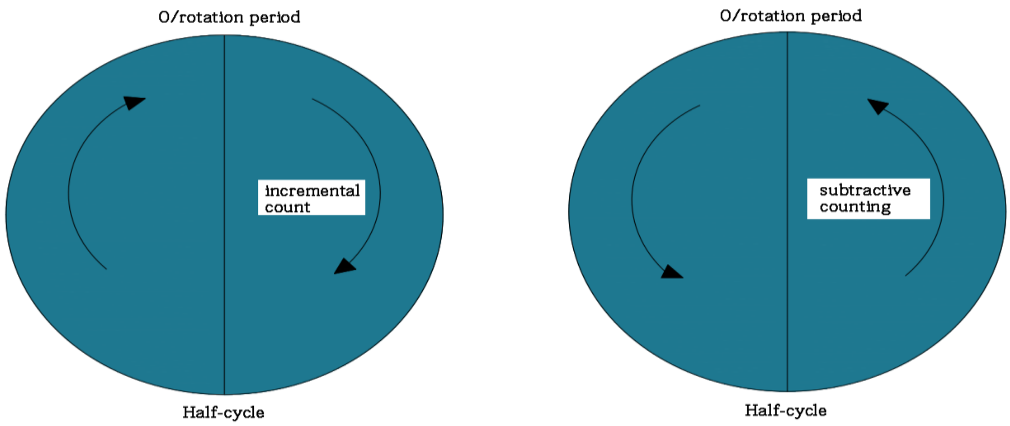
In linear mode, the high-speed counter operates within the closed range of [negative limit value, positive limit value]. When the direction is negative, the count value decreases in the negative direction, and after reaching the negative limit value, the count value does not decrease any more; when the direction is positive, the count value increases in the positive direction, and after reaching the positive limit value, the counte value is no longer incremented. The linear mode is as follows:



5.4.2 Rotation mode

The position of the counter shaft changes cyclically within the rotation period. When counting up, the position of the counter shaft reaches the maximum value of the rotation period and becomes 0. When counting down, the position of the counter shaft is 0 and then decreases from the maximum value of the rotation period. In rotation mode, the rotation period of the counter axis can be set in the interface, and the period unit is the user unit (Unit). Since the high-speed counter is a 32-bit counter, the rotation period must be in the range of 32-bit integers [-2147483648, 2147483647] after being converted into pulse units.

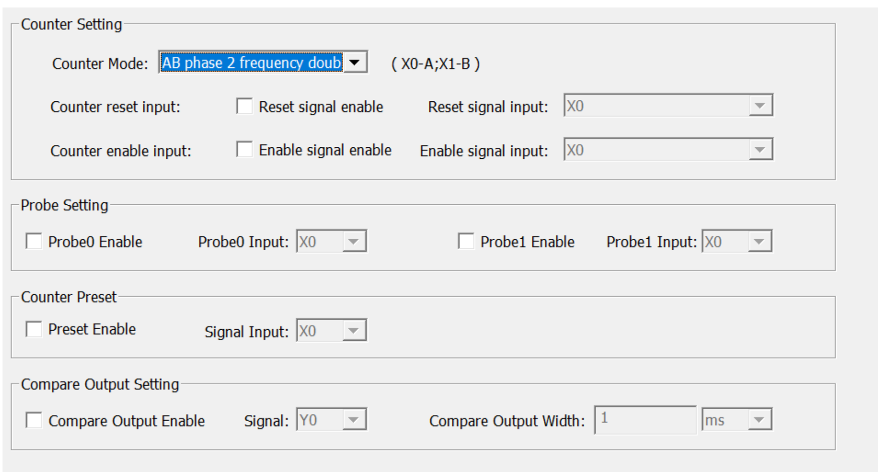




5.5 Set Counter Parameters

5.5.1 Overview

Parameter settings mainly include counting mode, probe, Enable, Compare output function.



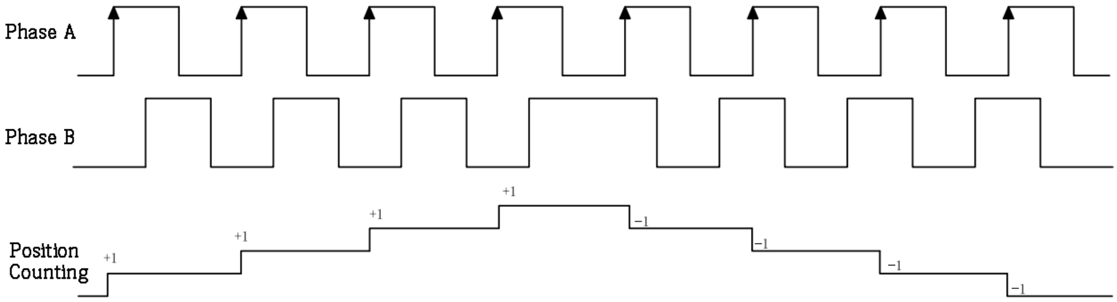
5.5.2 Count mode

The local encoder axis supports a variety of signal counting modes, A/B phase (1/2/4 frequency multiplication), CW/CCW, pulse + direction, singleMutuallycount.
VC5 supports 8 high-speed counters.

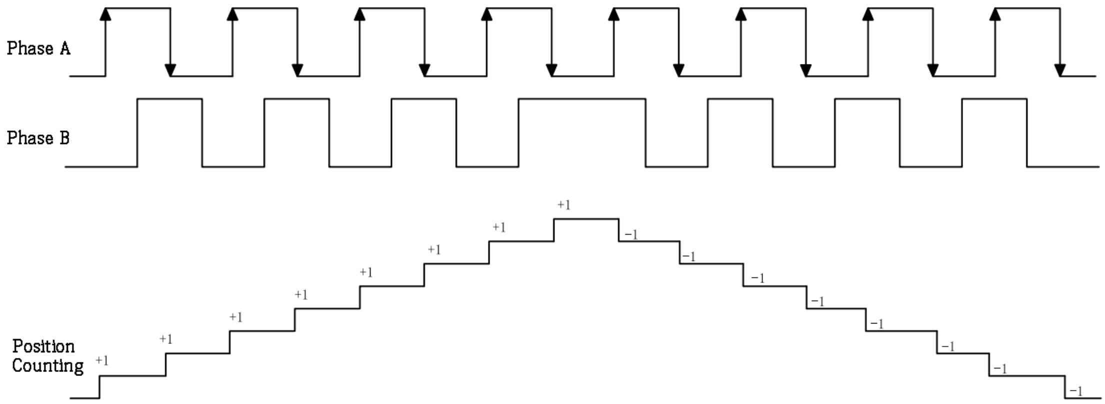
5.5.2.1 A/B phase mode

In A/B-phase mode, the encoder generates two quadrature-phase pulse signals with a phase difference of 90°, namely A-phase signal and B-phase signal. When the A-phase signal leads the B-phase signal, the counter counts up; when the B-phase signal leads the A-phase signal, the counter counts down.
Phase A/B pulses can be set to work in 1-multiplier, 2-multiplier or 4-multiplier mode.

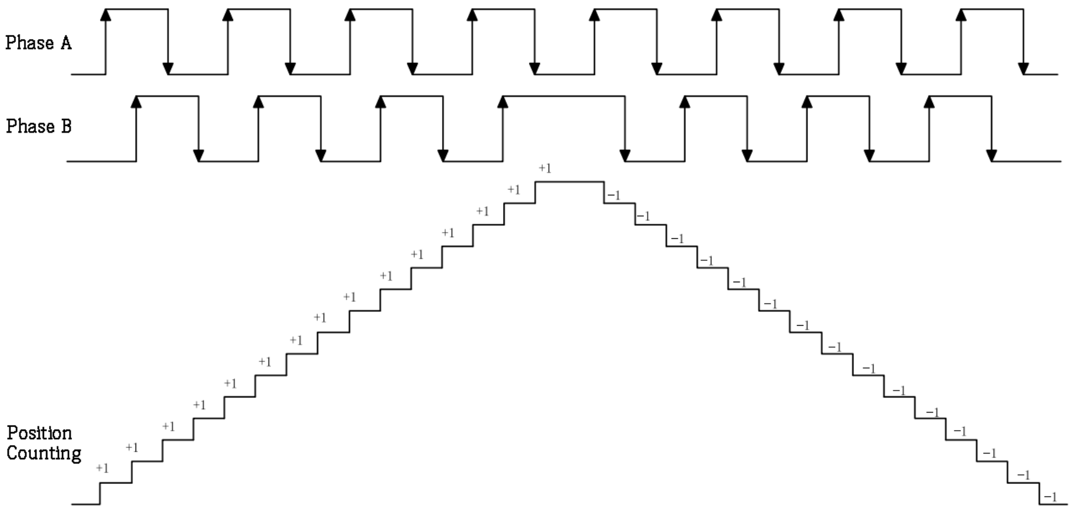
● In the A/B phase 1 multiplication mode, only the rising edge of the A phase pulse is counted, as shown in the following figure:



● In A/B phase 2 frequency multiplication mode, count the rising/falling edge of A phase pulse, as shown in the following figure:

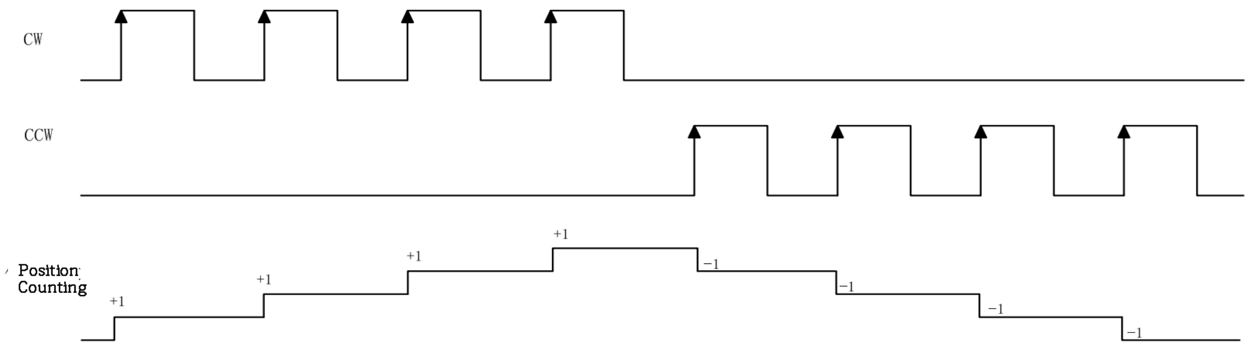


● In A/B phase 4 multiplier mode, count the rising/falling edges of A-phase pulse and B-phase pulse, as shown in the following figure:



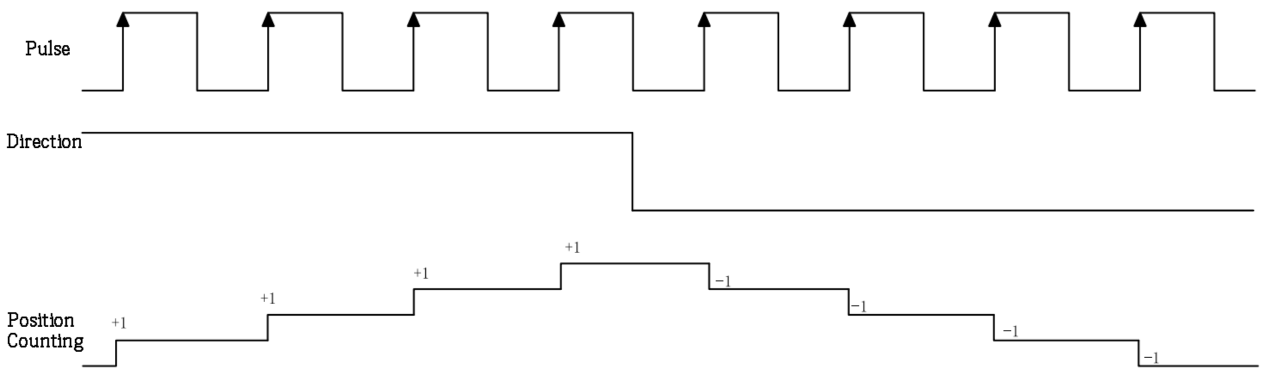
5.5.2.2 CW/CCW mode

CW (Clock Wise) is a forward rotation pulse signal, and CCW (Counter Clock Wise) is a reverse rotation pulse signal. When the encoder rotates forward, CW outputs pulse signal; when the encoder rotates reversely, CWW outputs pulse signal. When the local encoder axis works in this counting mode, the high-speed counter counts up the CW signal and counts down the CCW signal, as shown in the figure below.



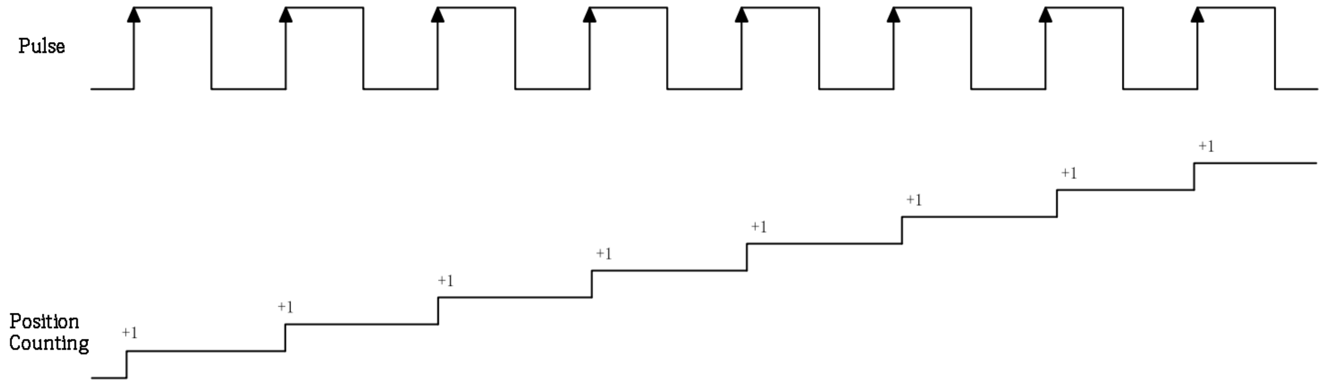
5.5.2.2 Pulse + Direction mode

In this mode, when the direction signal is ON, the high-speed counter counts up the pulse signal, and when the direction signal is OFF, the high-speed counter counts down the pulse signal, as shown in the figure below.



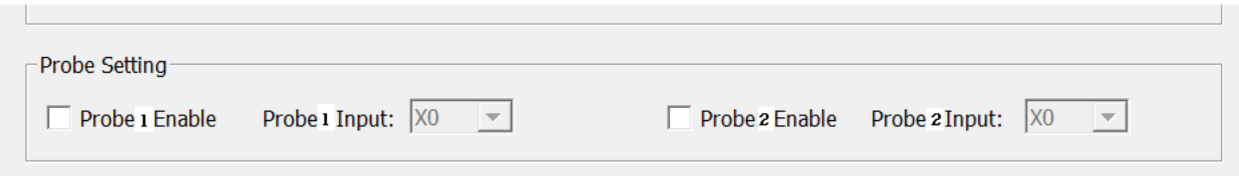
5.5.2.3 Single phase count

In this mode, the high-speed counter counts up the pulse signal, and when the input pulse rises, the position count increases by 1.



5.5.2.4 Probe terminal setup

Each counter supports 2 external inputs to latch the current value of the counter to realize the probe function. By checking the probe enable to enable the external input counter axis position latch, the input terminal can choose X0~X7 input arbitrarily. After enabling the probe, read the probe position of the counter axis through the HC_TouchProbe function block command.



5.5.2.5 Comparison output terminal settin

After checking "Comparison Output Enable", the hardware output when the comparison is equal can be realized without software processing. The real-time performance is high, and the output response can reach the microsecond level.

● After starting the comparison output function, cooperate with the function block command, and control the output to be ON through the hardware circuit when the comparison is equal, the output terminal can be arbitrarily selected from Y0~Y3, and the pulse width of the output is ON can be selected as the time unit(ms)or user unit (Unit).

● Each local encoder axis is equipped with a comparison output function, and the input terminal and output pulse width can be configured according to requirements.

● After the configuration is completed, use the HC_Compare, HC_ArrayCompare, and HC_StepCompare function block commands to realize the axis position comparison output.

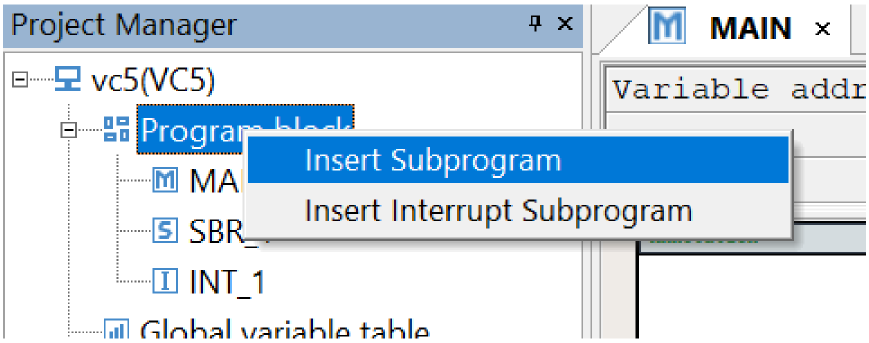
● When the unit is ms, the set time range is 0.1~6553.5ms. When selecting Unit as the unit, make sure that the set value is in the range of 1~65535 after converting it into pulse unit.

The comparison output is directly output through the hardware control port and is not processed by software, so the status of the comparison output cannot be displayed through the Y soft element in the program. The Y soft element and the comparison output control the output port in an OR relationship. If the Y soft element is continuously controlled to be in the ON state, the actual port output remains ON.

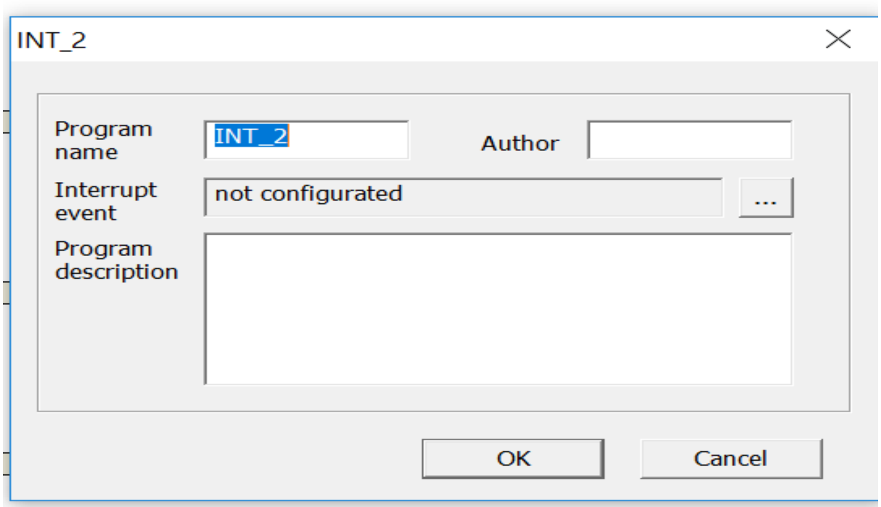
5.5.2.6 Compare interrupt

When the counter axis compares equal, the comparison interrupt can be associated to execute the interrupt subroutine. The specific operation steps are as follows:

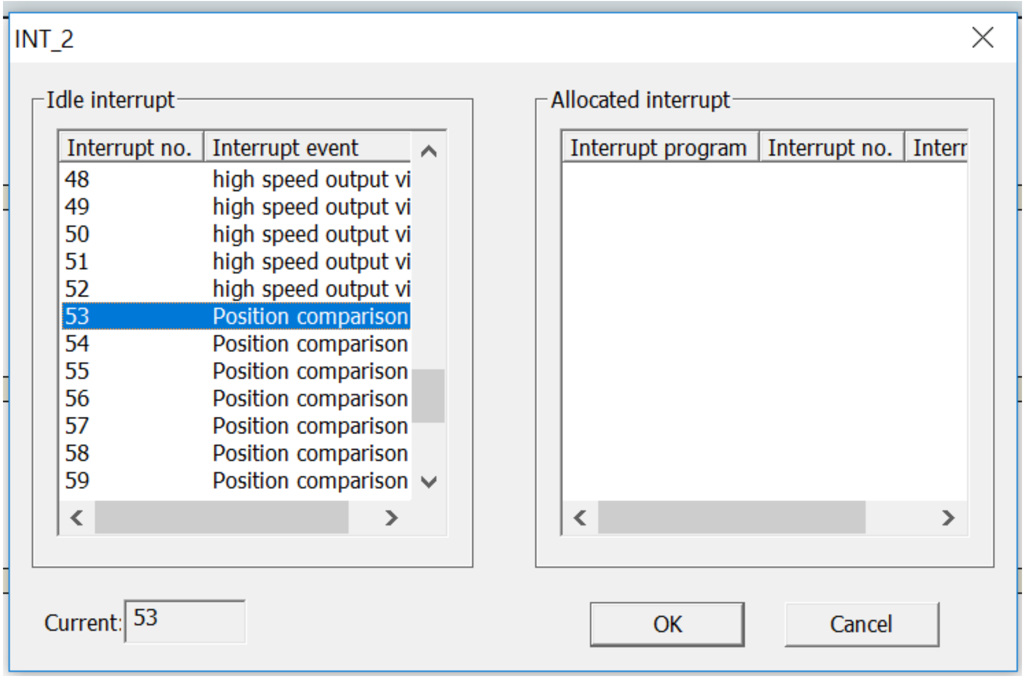
1. Under the "Programming" item of "Project Management", right-click "Program Block" and select "Insert Interrupt Subroutine " to interrupt the subroutine.



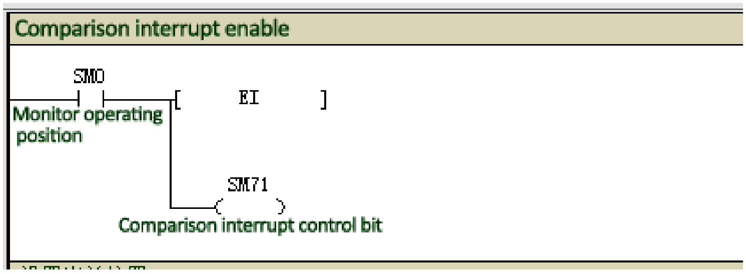
2. Right-click the inserted interrupt subroutine (INT_2 in the figure) and select "Properties" to open the interrupt subroutine setting page as shown in the figure below.

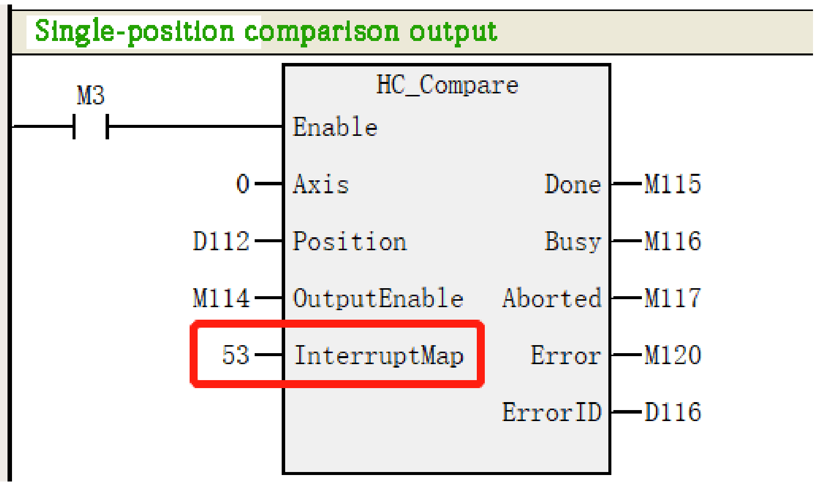


3. You can write the interrupt subroutine in INT_2 by clicking the selection icon after the "Interrupt Event" field and selecting the compare interrupt.



4. Call the MC_Compare, MC_StepCompare, MC_ArrayCompare commands in the main program or subprogram to associate the parameter InterruptMap with the compare interrupt number, i.e., the parameter InterruptMap is set to the compare interrupt number. The global interrupt EI is enabled in the program, and the compare interrupt control bit SM71 is enabled to trigger the execution of the corresponding compare interrupt subroutine when the instructions are equal.





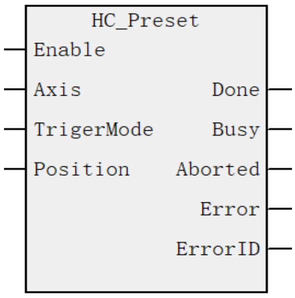
5.6 Local high-speed counter

The high-speed counter commands cover the following command entries:

Command category	Name	Function
High-speed counter	HC_Preset	High-speed counter preset value
	HC_Counter	High-speed counter enable
	HC_TouchProbe	High Speed Counter Probe
	HC_Compare	High-speed counter comparison
	HC_ArrayCompare	High-speed counter group comparison
	HC_StepCompare	High-speed counter equal-spaced comparison

5.6.1 HC_Preset (High-speed counter preset value)

Set the counter value to the preset value according to the trigger signal.
Drawing Block:



Command list format

Input Output	Name	Type of data	Applicable (soft components)	Scope	Describe	Enable
IN	Axis	WORD	constant/D/R/W	0~71	Axis name/Axis number	Execute rising edge
IN	TrigerMode	WORD	constant/D/R/W	0~1	Preset mode selection 0: command rising edge trigger 1: Triggered by the rising edge of the digital input terminal	
IN	Position	REAL	D/R/W		Default position	
OUT	Done	BOOL	M/S	TRUE, FALSE	Finish	
OUT	Busy	BOOL	M/S	TRUE FALSE	Running sign	
OUT	Aborted	BOOL	M/S	TRUE FALSE	execution interrupt	
OUT	Error	BOOL	M/S	TRUE, FALSE	error flag	

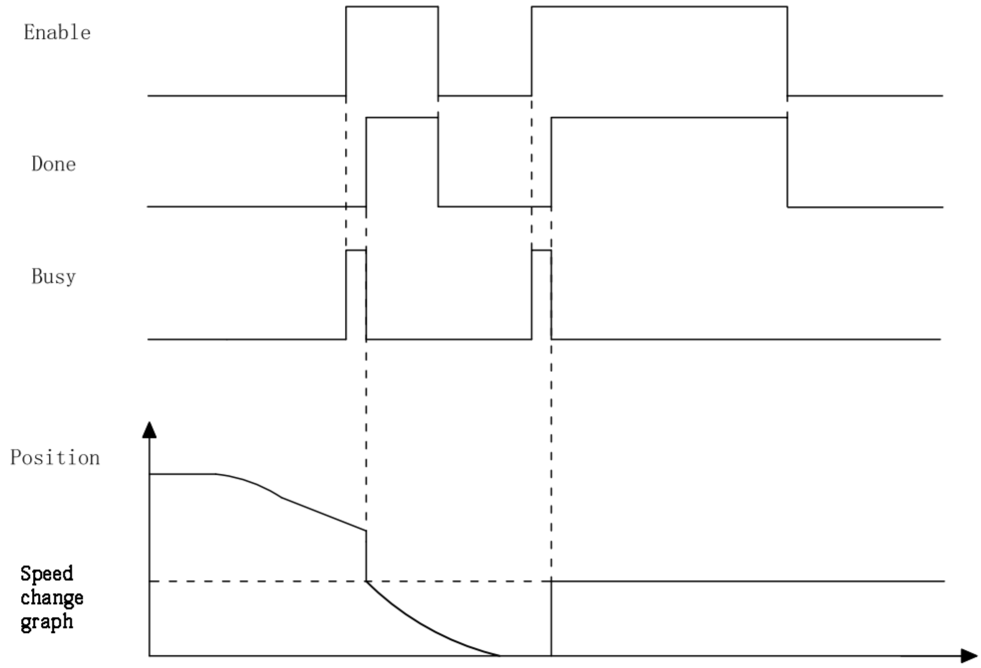
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	error ID	

Function and command description

Use the HC_Preset command to assign the counter axis position according to the preset conditions. Preset ConditionTrigerMode selects whether the rising edge of the command is triggered or the external X input is triggered.

Timing diagram

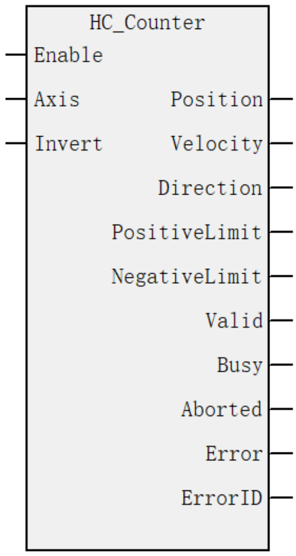
- The command is triggered by the rising edge, and the command timing diagram is shown in the figure below.



5.6.2 HC_Counter (High-speed counter enable)

Controls the high-speed counter to start or stop counting.

Drawing Block:



Command list format

Input Output	Name	Type of data	Applicable (soft components)	Scope	Describe	Enable
IN	Axis	WORD	constant/D/R/W	0~71	Axis name/Axis number	Enable is valid
IN	Invert	BOOL	M/S	TRUE, FALSE	direction control	
OUT	Position	REAL	D/R/W		current position	
OUT	Velocity	REAL	D/R/W		current speed	
OUT	Direction	BOOL	M/S	TRUE, FALSE	counting direction	
OUT	Positive Limit	BOOL	M/S	TRUE, FALSE	Positive limit value reached in linear mode	
OUT	Negative Limit	BOOL	M/S	TRUE, FALSE	Negative limit value reached in linear mode	
OUT	Valid	BOOL	M/S	TRUE, FALSE	active state	
OUT	Busy	BOOL	M/S	TRUE, FALSE	Running sign	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	execution interrupt	
OUT	Error	BOOL	M/S	TRUE, FALSE	error flag	
OUT	EerrorID	WORD	D/R/W	0x0~0xffff	error ID	

Function and command description

Using the HC_Counter command, the position count and speed measurement of the counter axis are possible.

The counter axis position value varies within the range of the counter axis mode according to the mode setting, and the position unit is Unit.

Invert (count inverse) parameter

The counting direction of the counter is set through the Invert parameter, and the definition of the counting direction of different counting modes is shown in the following table. After changing the setting of Invert, you need to re-enable the function block command to take effect.

Invert	A/B Phase	Pulse + direction	CW/CCW	Single phase count
0	A phase ahead B phase up count. B phase ahead A phase down count.	Direction signal low level counts down Direction signal high level counts up	A phase counts up B phase counts down	Count up
1	A phase ahead B phase down count. B phase ahead A phase increase count.	Direction signal low increase count Direction signal high level reduce count		Count down

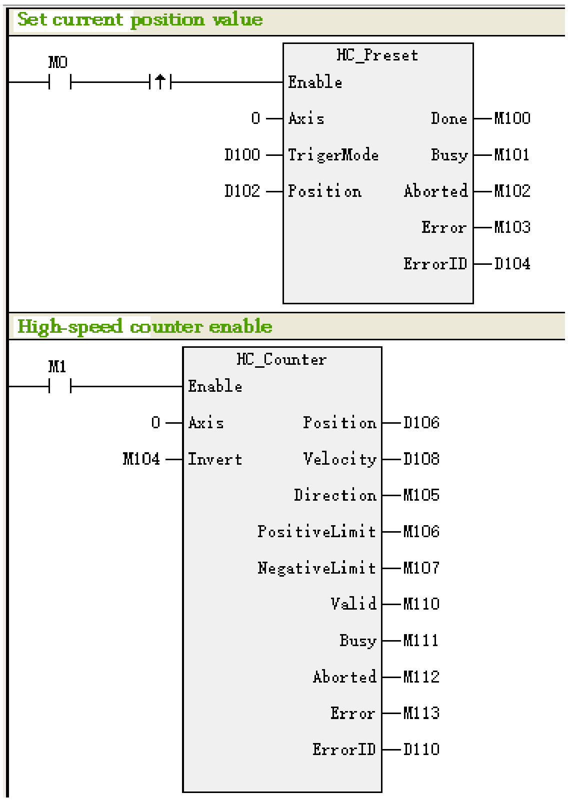
Program example

By the following routineRealize high-speed counter counting function.

Select high-speed counter 0, connect to terminal X0, single-phase counting.

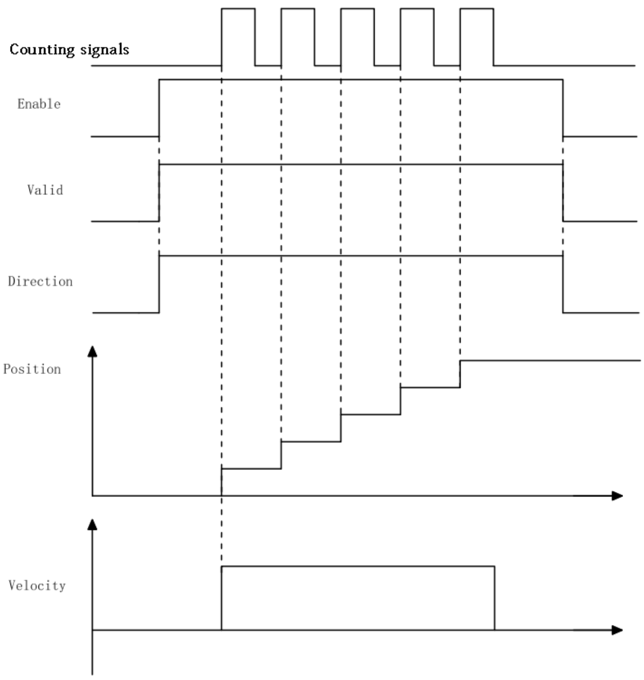
1. After setting M0 to ON, set the current high-speed counter axis position.

2. After setting M1 to ON, X0Received count signal, Axis_0Update current axis position.

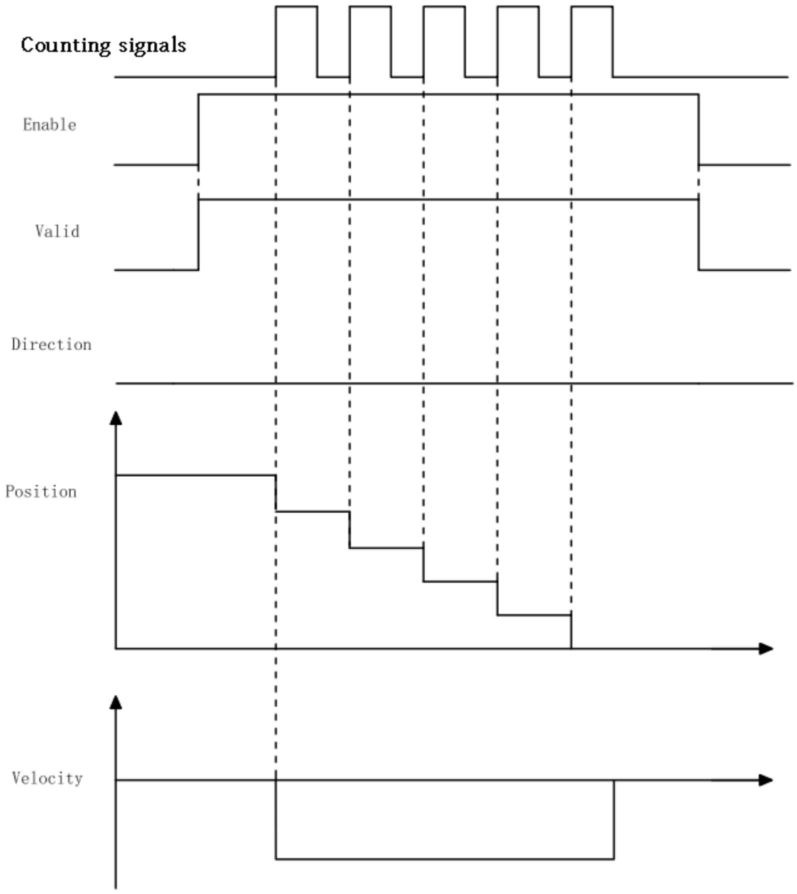


Timing diagram

● With "Pulse + Direction" mode as an example, the counter counts up when the direction signal=ON/Invert=0 or the direction signal=OFF/Invert=1, as shown in the following figure:

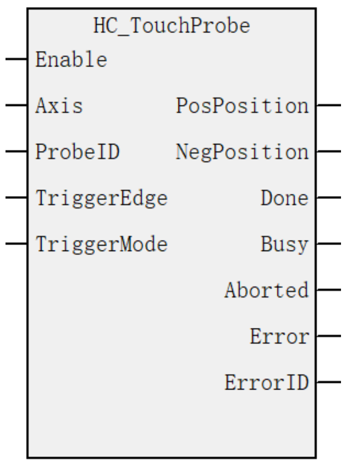


● With " Pulse + Direction” mode as an example, when the direction signal=ON/Invert=1 or the direction signal=OFF/Invert=0, the counter counts up, as shown in the figure below.



5.6.3 HC_TouchProbe (High-speed counter probe)

The counter value is recorded according to the occurrence of the trigger signal.
Drawing Block:



Command list format

Input Output	Name	Type of data	Applicable (soft components)	Scope	Describe	Enable
IN	Axis	WORD	constant/D/R/W	0~71	Axis name/Axis number	Execute rising edge
IN	ProbeID	WORD	constant/D/R/W	0 -1	Probe ID:	Execute rising edge
IN	TriggerEdge	WORD	constant/D/R/W	0~2	Edge trigger mode: 0rising edge 1 falling edge 2 any edge	Execute rising edge
IN	TriggerMode	WORD	constant/D/R/W	0~1	Trigger type: 0: single trigger 1: Continuous trigger	Execute rising edge
OUT	PosPosition	REAL	D/R/W	positive/negative/0	Rising edge capture position	
OUT	NegPosition	REAL	D/R/W	positive/negative/0	Falling edge capture position	
OUT	Done	BOOL	M/S	TRUE/FALSE	Complete stop	
OUT	Busy	BOOL	M/S	TRUE/FALSE	Running sign	
OUT	Aborted	BOOL	M/S	TRUE/FALSE	Execution interrupt	
OUT	Error	BOOL	M/S	TRUE/FALSE	Error flag	
OUT	EerrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Function and command description

Using the HC_TouchProbe function block command, the counter axis position value can be latched when the external input trigger condition is valid.

Each counter axis supports 2-way probes. When using, you need to select the corresponding probe function in the counter parameter setting, select the input terminal and trigger condition, and the input terminal can be arbitrarily set to choose X0~X7. The parameter TriggerEdge sets the probe trigger edge.

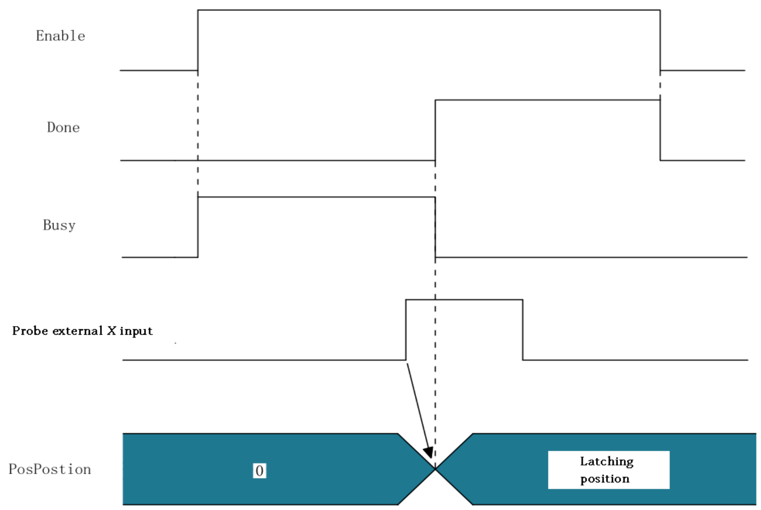
The rising edge trigger position is latched in the output parameter PosPosition, and the falling edge trigger position is latched in the output parameter NegPosition.

Timing diagram

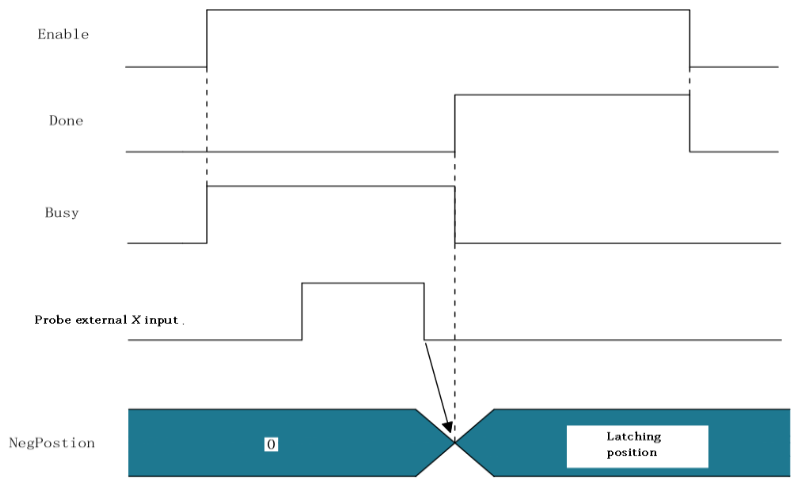
●ExternalXRising edge trigger (TriggerEdge=0), single trigger mode (TriggerMode=0), the command timing diagram is shown in the following figure.

Timing diagram

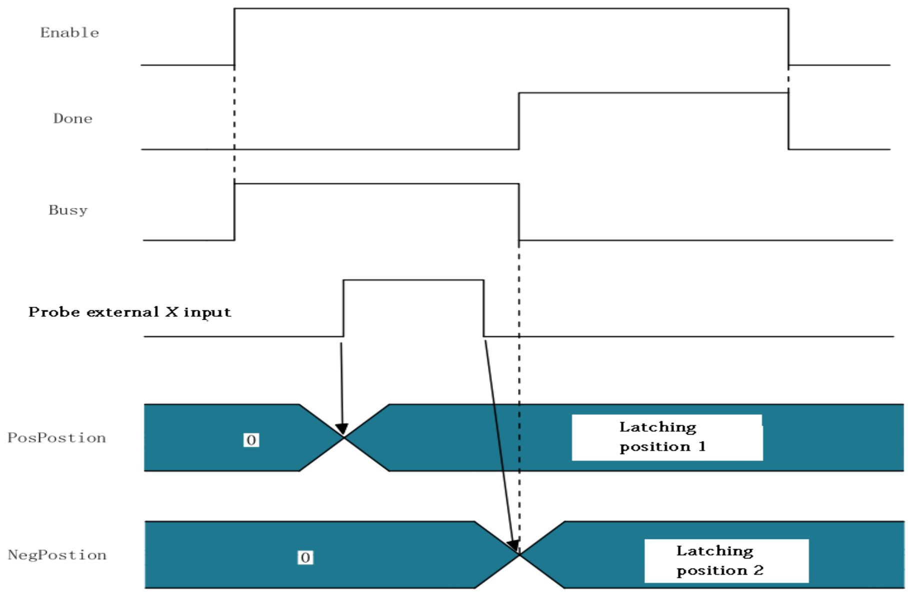
●ExternalXRising edge trigger (TriggerEdge=0), single trigger mode (TriggerMode=0), the command timing diagram is shown in the following figure.



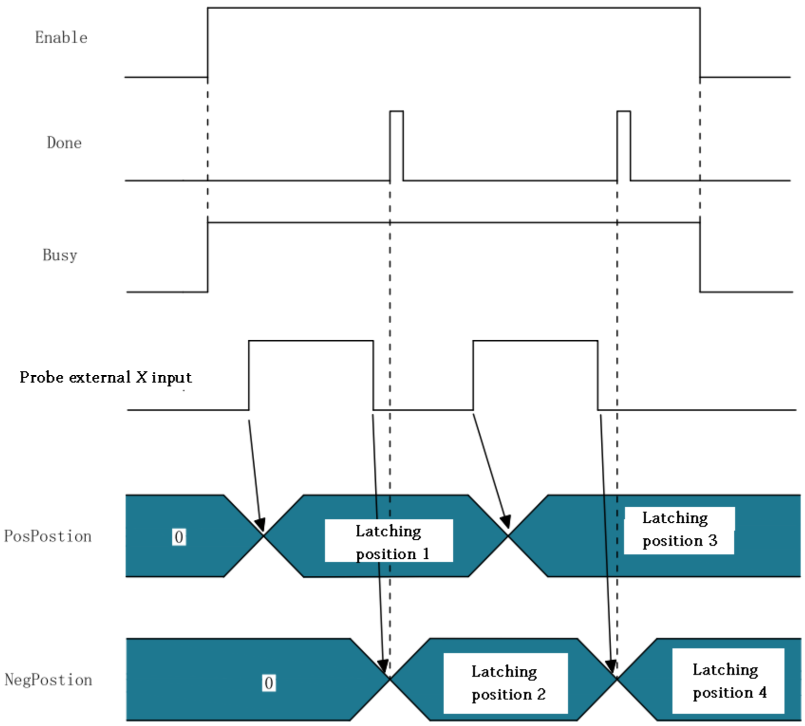
●ExternalXFalling edge trigger (TriggerEdge=1), single trigger mode (TriggerMode=0), the command sequence diagram is shown in the following figure.



● External X rising edge and falling edge trigger (TriggerEdge=2), single trigger mode (TriggerMode=0), the command sequence diagram is shown in the figure below.

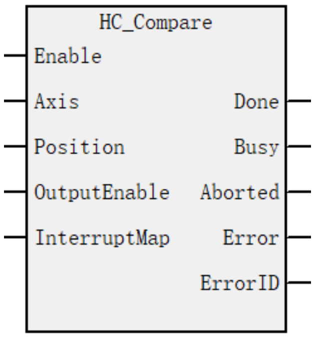


● External X rising edge and falling edge trigger (TriggerEdge=2), continuous trigger mode (TriggerMode=1), the command sequence diagram is shown in the figure below.



5.6.4 HC_Compare（High-speed counter comparison）

The probe counter count reaches the specified value.
Drawing Block:



Command list format

Input output	Name	Type of data	Applicable (soft components)	Scope	Describe	Enable
IN	Axis	WORD	Constants/D/R/W	0~71	Shaft name / shaft number	Enable is valid
IN	Position	REAL	D/R/W		Compare positions: Units: Unit	
IN	OutputEnable	BOOL	M/S	TRUE/FALSE	Compare	

					enable output	
IN	InterruptMap	WORD	Constants/D/R/W		Interrupt Map No.	
OUT	Done	BOOL	M/S	TRUE/FALSE	Completion Sign	
OUT	Busy	BOOL	M/S	TRUE/FALSE	Running sign	
OUT	Aborted	BOOL	M/S	TRUE/FALSE	Execution interruptions	
OUT	Error	BOOL	M/S	TRUE/FALSE	Error flag	
OUT	ErrorID	WORD	/D/R/W	0x0~0xffff	Error ID	

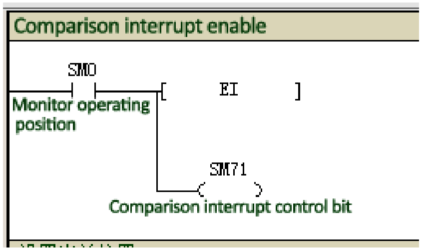
Function and command description

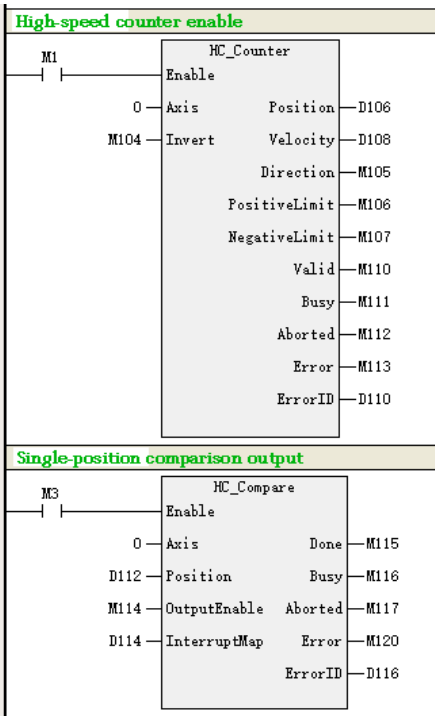
A single position comparison of a counter axis is possible using HC_Compare.

Program example

The single position comparison output function is realized through the following routine. Set the comparison position is D112=1000.

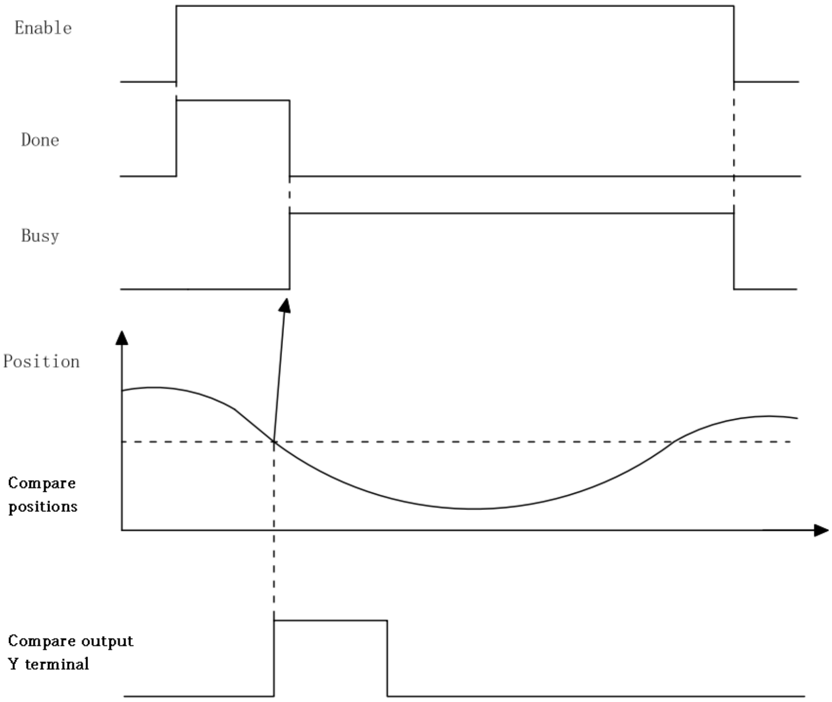
1. Interrupt enable
2. When M1 is set to ON, the high-speed counter axis is enabled. 3.
3. When M3 is set to ON, when count value = 1000, compare output Done is set, and when external output is enabled, external output is set.





Timing diagram

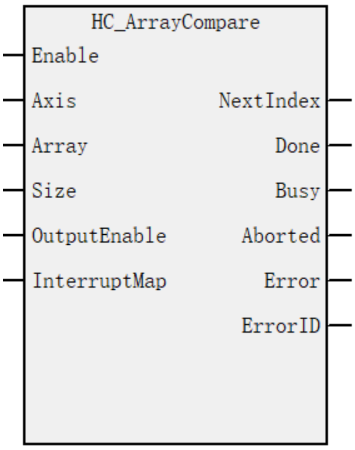
The command timing diagram is shown in the following figure.



5.6.5 HC_ArrayCompare (High-speed counter array comparison)

Continue to detect counter counts in order to reach the specified array sequence.

Drawing Block:



Command list format

Input Output	Name	Type of data	Applicable (soft components)	Scope	Describe	Enable
IN	Axis	WORD	constant/D/R/W	0~71	Axis name/Axis number	Enable is valid
IN	Array	REAL	D/R/W		Compare arrays:	
IN	Size	WORD	constant/D/R/W	1~1000	Number of comparison	

					values	
IN	OutputEnable	BOOL	M/S	TRUE/FALSE	Compare output enable	
IN	InterruptMap	WORD	constant/D/R/W		Interrupt map number	
OUT	NextIndex	WORD	constant/D/R/W	0~999	Next comparison value index	
OUT	Done	BOOL	M/S	TRUE/FALSE	Complete stop	
OUT	Busy	BOOL	M/S	TRUE/FALSE	Running sign	
OUT	Aborted	BOOL	M/S	TRUE/FALSE	Execution interrupt	
OUT	Error	BOOL	M/S	TRUE/FALSE	Error flag	
OUT	EerrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Function and command description

Use HC_ArrayCompare to achieve multi-position continuous comparison of counter axes.

When the command power flow is valid, the counter axis position is compared with the first position of the array, and after the comparison is equal, it is compared with the next position value of the array.

In the command, Size sets the array length. After the comparison of all array positions set by the array length is completed, the continuous output of signals is completed, and the continuous comparison of multiple positions is completed.

The output parameter NextIndex indicates the next comparison point index, that is, the number of completed comparisons for equality.

Program example

Set the number of comparison values to 3, that is, Size=3. Use the following routine to achieve the multi position comparison output function.

The three comparison positions are shown in the figure:

MAIN | EMT_1 * ×

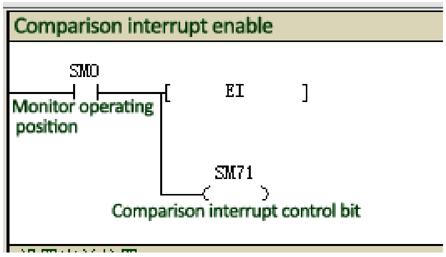
	Element Name	data type	display forma	current value
1	D144	REAL	Decimal	1000.000
2	D146	REAL	Decimal	3000.000
3	D148	REAL	Decimal	5000.000

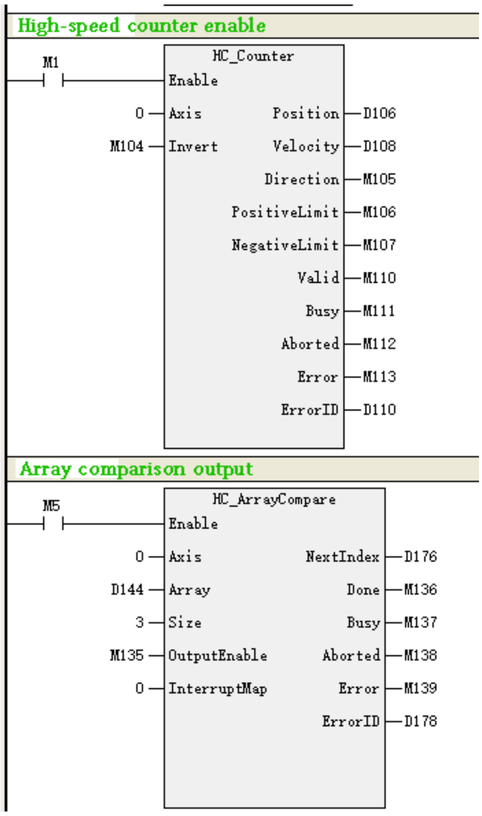
The function block command fills in the register that sets the first comparison position, and the consecutive register addresses set the continuous comparison position.

Program execution:

1. Interrupt enable
2. After setting M1 to ON, the high-speed counter axis is enabled.
3. After setting M5 to ON, the comparison output will be output according to the count position, and Position can monitor the current counter value.

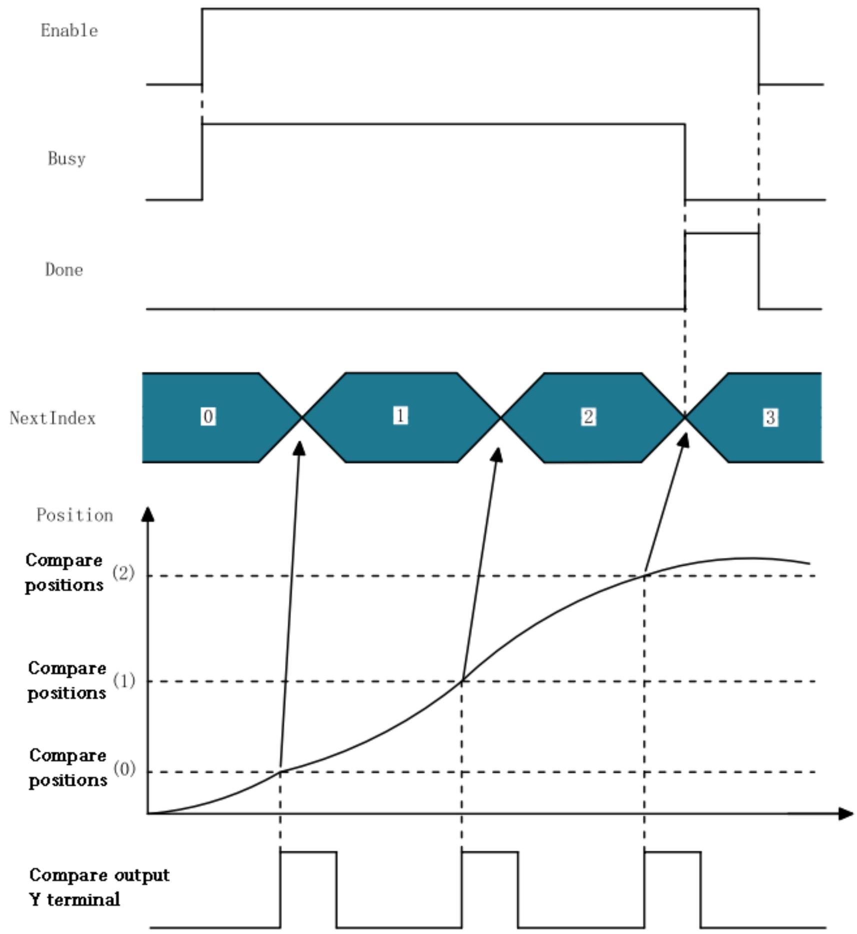
The next index value can be monitored by NexIndex. The program is set to 3 positions, when NexIndex=2, Done is set.





Timing diagram

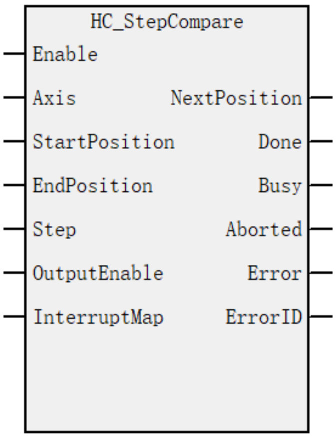
Compare 3 positions (Size=3), use the hardware output (OutputEnable=1), the command timing diagram is shown in the following figure.



5.6.6 HC_StepCompare (High-Speed counter equidistant distance comparison)

Continuously detect counters in sequence to count consecutive ordinal values to the specified range and spacing.

Drawing Block:



Command list format

Input Output	Name	Type of data	Applicable (soft components)	Scope	Describe	Enable
IN	Axis	WORD	constant/D/R/W	0~71	Axis name/Axis	Enable is valid

					number	
IN	StartPosition	REAL	constant/D/R/W		Start comparison position:	
IN	EndPosition	REAL	constant/D/R/W		End comparison position	
IN	Step	REAL	constant/D/R/W		Step size	
IN	OutputEnable	BOOL	M/S	TRUE/FALSE	Compare output enable	
IN	InterruptMap	WORD	constant/D/R/W		Interrupt map number	
OUT	NextPosition	REAL	D/R/W		Next comparison value	
OUT	Done	BOOL	M/S	TRUE/FALSE	Complete stop	
OUT	Busy	BOOL	M/S	TRUE/FALSE	Running sign	
OUT	Aborted	BOOL	M/S	TRUE/FALSE	Execution interrupt	
OUT	Error	BOOL	M/S	TRUE/FALSE	Error flag	
OUT	EerrorID	WORD	D/R/W	0x0~0xffff	Error ID	
OUT	EerrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Function and command description

Using HC_StepCompare, the counter axis can be continuously compared at equal intervals.

When the command power flow is valid, the counter axis position and the StartPosition position start to compare. After the comparison is equal, the comparison position increases or decreases the Step distance and continues the comparison.

After the last comparison position is compared at equal intervals, the completion signal is continuously output.

The output parameter NextPosition indicates the label of the next comparison point. The label is calculated from 0, that is, 0 represents the first comparison point. Therefore, this label is exactly equal to the number of comparison points that have been completed.

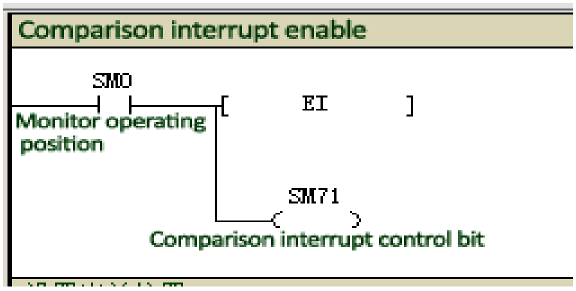
Program example

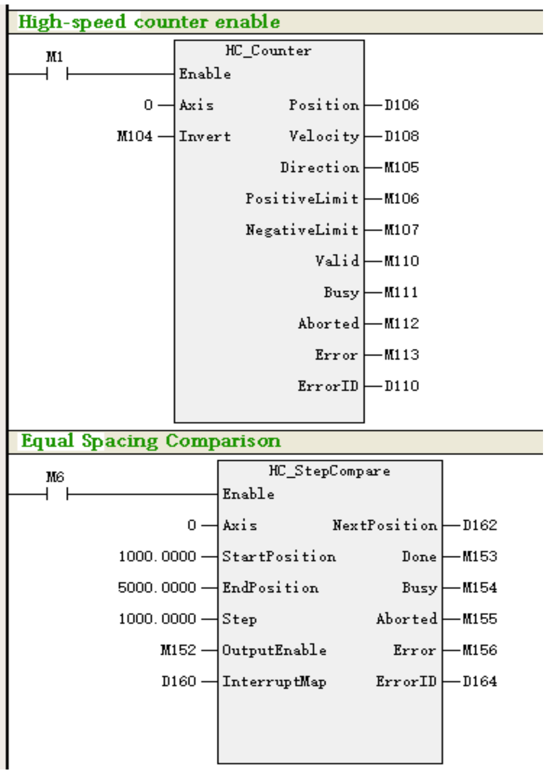
1. Interrupt enable

2. After setting M1 to ON, enable the high-speed counter axis.

3. After setting M6 to ON, the current position of counter Position=1000, comparison output is set, and the spacing Step=1000, comparison output is set at 2000, 3000, 4000, 5000.

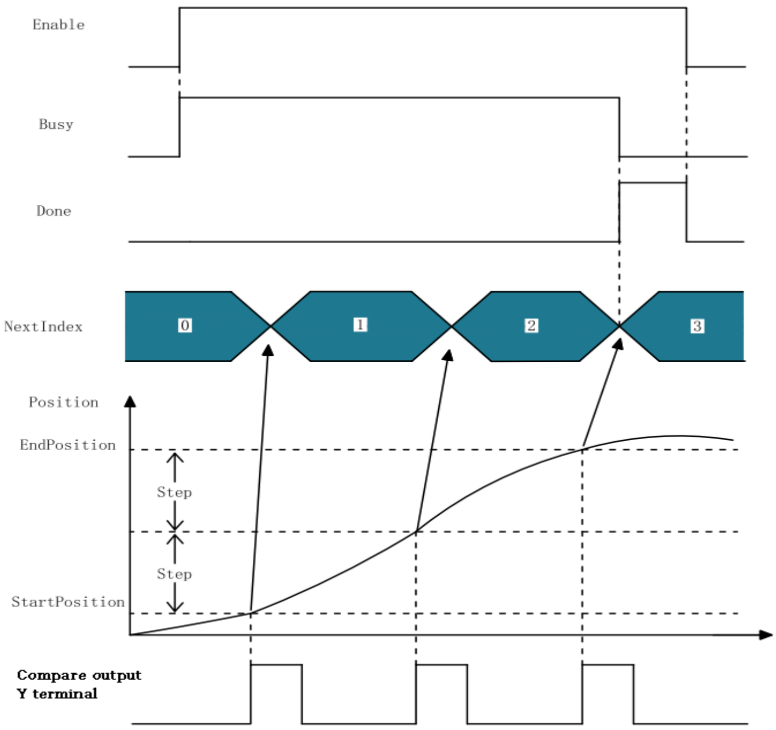
You can monitor NextPosition to check the current comparison point marker, and when the count value = 5000, Done is set.



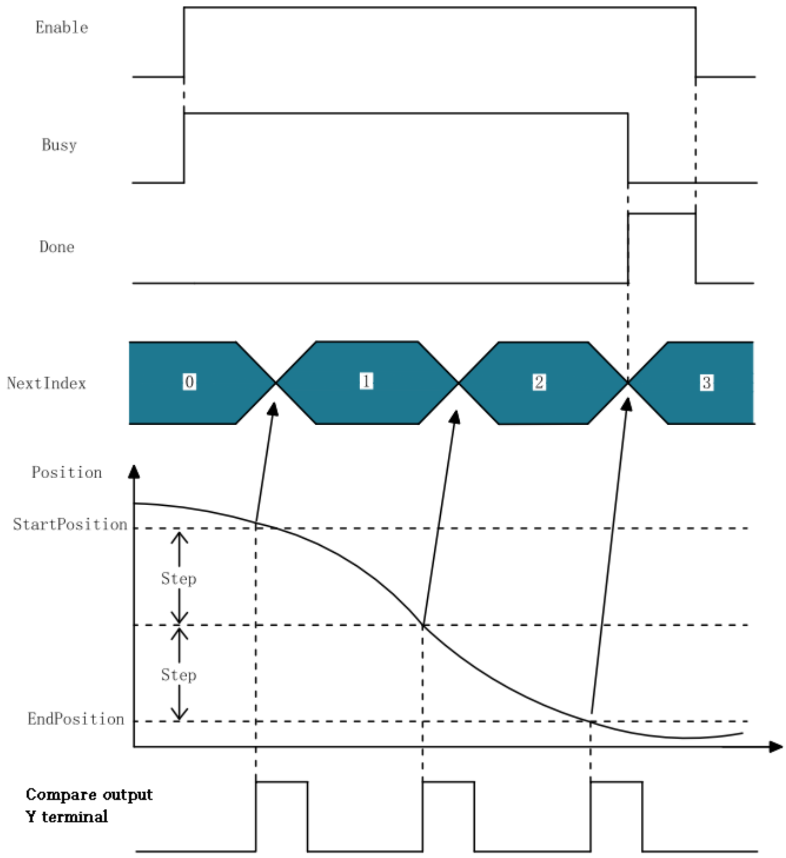


Timing diagram

● Use hardware output (OutputEnable=1), StartPosition < EndPosition, the command timing diagram is shown in the figure below.



● Use hardware output (OutputEnable=1), StartPosition>EndPosition, the command timing diagram is shown in the figure below.



6. Electronic Cam

6.1 Introduction to electronic cams

The electronic cam is essentially the movement of the slave axis following the master axis, and the motion relationship between the master axis and the slave axis can be represented by cam table data or electronic gear ratio.

● Use the electronic cam table to create a maximum of 361 key point data. With electronic gear ratios, there is only a fixed proportional relationship between the master and slave axes.

● If electronic gear is used, it is only necessary to set the numerator and denominator of the electronic gear ratio, and there is no need to set the cam table data. To use an electronic cam, it is necessary to set the electronic cam table data first.

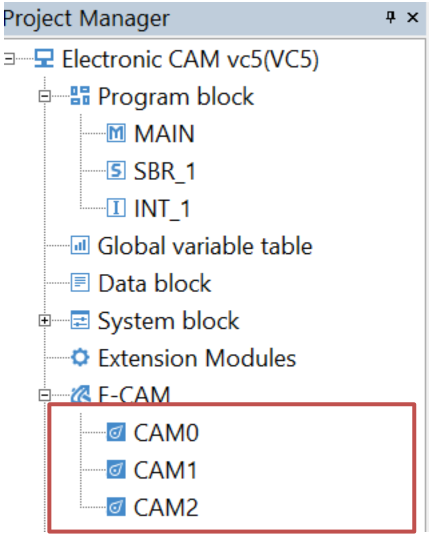
● The programming software can configure 16 cam tables, and 8 electronic cams can be used in the program at the same time. The maximum number of key points in each cam table is 361.

● During the cam execution process, it is allowed to add, delete and modify the key points of the cam table, and the modified cam table will take effect in the next cam cycle.

6.2 Software Settings

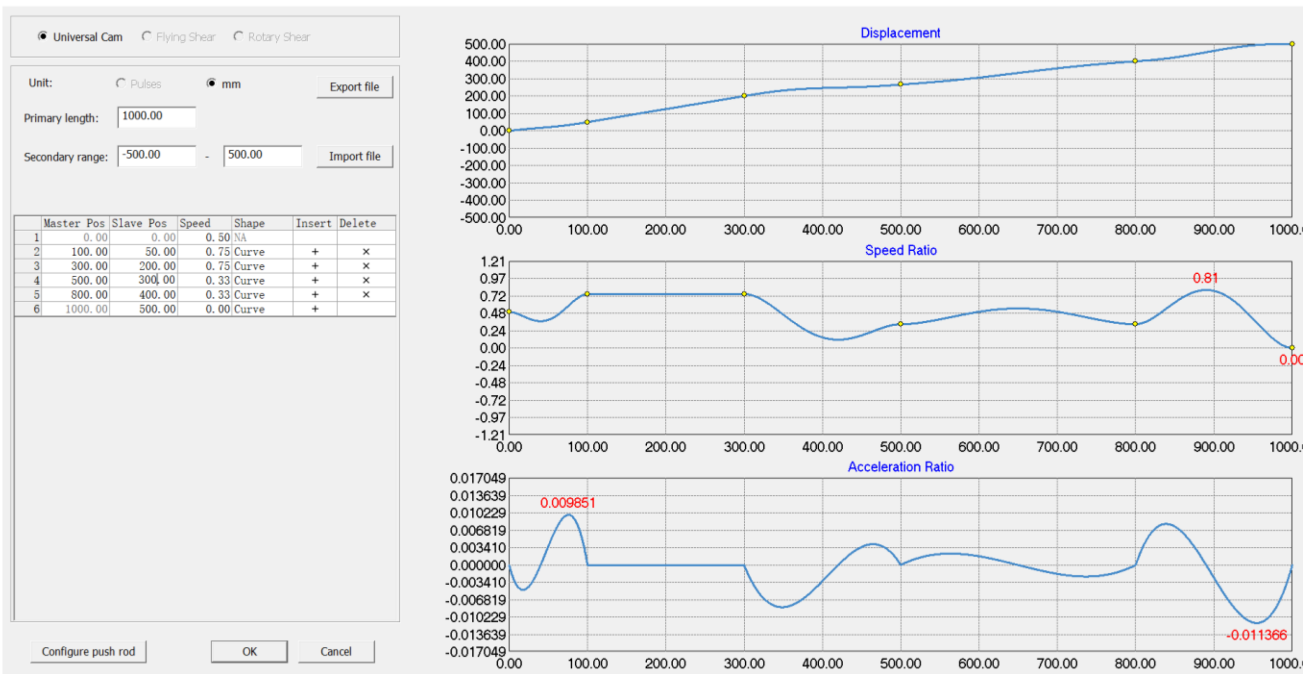
6.2.1 Overview

Expand the "Electronic Cam" item in the "Project Manager", right-click "Add Configuration" to create a new electronic cam, and double-click the electronic cam such as "CAM0" to open the relevant configuration interface.



The right side of the cam table interface is the graphic editing area, and the left side is the parameter point editing area.

Electronic CAM



Cam Table Specifications

When creating a cam table, follow these specifications:

Project	Illustrate
Total number of cam keys supported per cam table	361 Pcs
Total number of supported cam tables	16 Pcs

The number of cam tables allowed to execute simultaneously in the PLC	16 Pcs
Rules for cam table switching in cam action	Call the MC_CamIn command to switch the cam table and take effect in the next cam cycle.
Read and write cam data	The phase and displacement of the cam table can be obtained by command. The data of the cam key points in the cam table can be directly modified, and then take effect through the MC_GenerateCamTable command.

6.2.2 Cam node settings

Users can set cam nodes in the parameter point editing area according to application requirements. Click "+" under Insert to add a new cam node data row and edit the relevant data. Select the specified node data row and click "-" to delete the node.

	Master Pos	Slave Pos	Speed	Shape	Insert	Delete
1	0.00	0.00	0.50	NA		
2	100.00	50.00	0.75	Curve	+	×
3	300.00	200.00	0.75	Curve	+	×
4	500.00	300.00	0.33	Curve	+	×
5	800.00	400.00	0.33	Curve	+	×
6	1000.00	500.00	0.00	Curve	+	

Cam node parameter definition:

Parameter	Function
Spindle position	Spindle phase, Set the phase of the spindle (relative mode)
Slave position	Displacement from axis, used to set the slave axis offset (relative mode)
Speed	Connection speed, automatically generated when straight line is selected as curve type, manually set when quintic curve is selected
Shape	Set the curve type, straight line or 5 degree curve

Illustrate

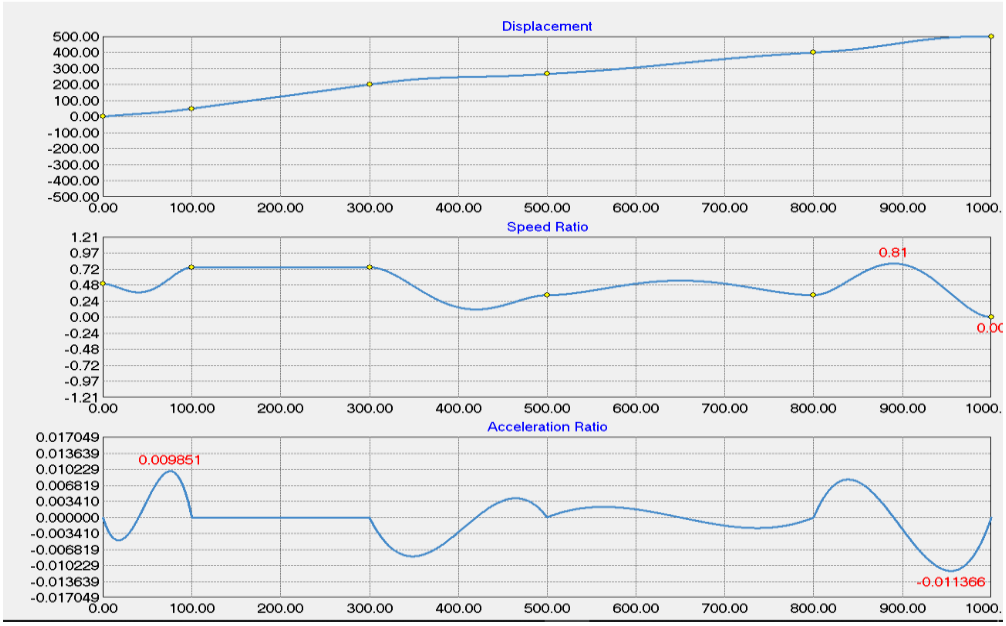
● The master axis phase and slave axis offset of the first point are 0 by default and cannot be changed.

● The spindle phases are arranged in ascending order.

● The last point of the spindle determines the size of the period of the spindle, and it is not necessary to set the period separately.

6.2.3 Cam curve settings

The user can set the cam curve in the graphic editing area according to the application requirements, including the curve of position, speed ratio and acceleration ratio.



Cam Curve Description

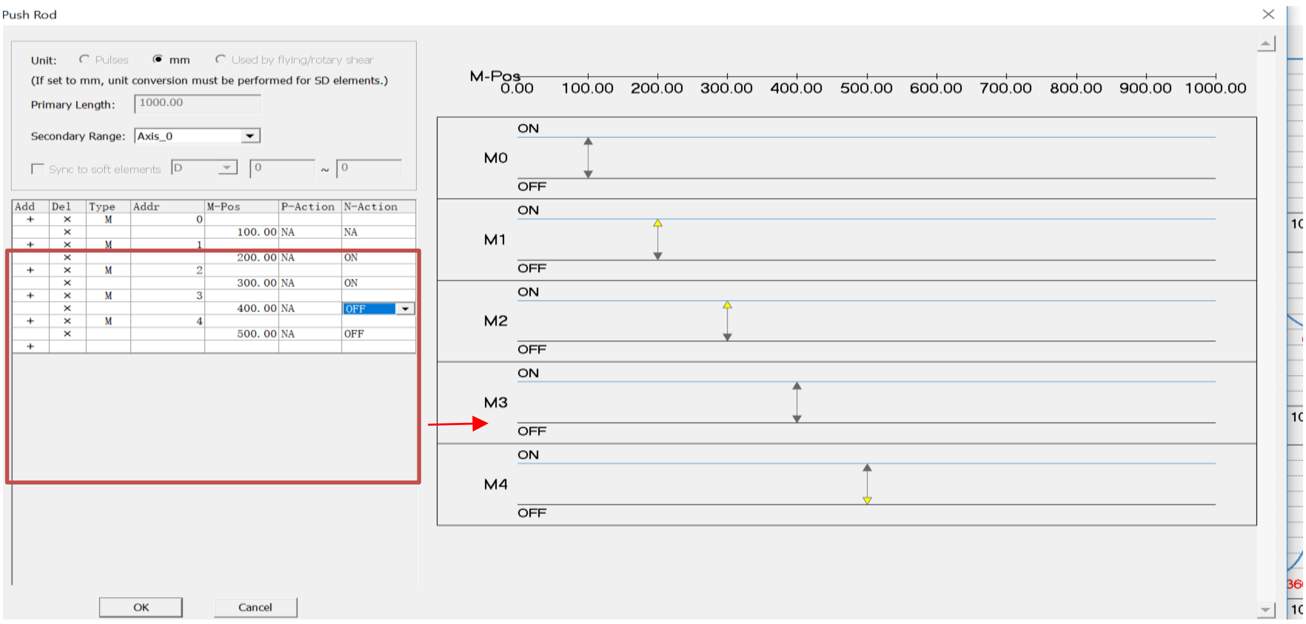
1. The position curve allows you to move the cam key points up, down, left and right. The speed ratio curve can only move up and down. The acceleration ratio curve is not allowed to change.

2. The last point can only be dragged up and down, not left and right. If you need to change the size left and right, you can manually modify the data of the last point on the right toolbar.

3. Move the mouse to any area of coordinates to prompt specific coordinate information.

6.2.4 Ejector settings

Definition of ejector rod: The ejector rod function can realize the cooperation between the bit element (M, Y) and the position of the electronic cam spindle, and control the ON/OFF change of the bit element with the change of the spindle position. The top rods are set up as follows:



Ejector data table setting parameters:

Parameter	Function
Slave Axis Properties	Indicates the axis attribute, that is, when the electronic cam slave axis and the axis attribute are the same, the ejector data takes effect.
Type	Displacement from axis, used to set the slave axis offset (relative mode)
Addr	Connection speed, automatically generated when straight line is selected as curve type, manually set when 5th degree curve is selected
M-Pos	Set the curve type, straight line or 5-degree curve
P-Action	The action of the component when the positive movement position of the spindle is equal to the M-Pos setting value; NA means no action; ON means set to ON; OFF means OFF; INV indicates negation.
N-Action	The counter-motion position of the spindle is equal to the action of the component when the M-POS setting value; NA means no action; ON means set ON; OFF means set OFF; INV means negation

6.3 Cam and Gear Instruction

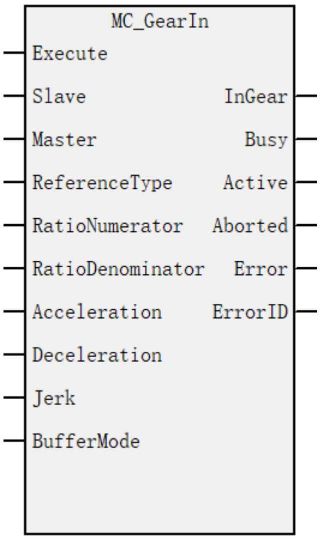
Command list

The electronic cam command covers the following command items

Command category	Command name	Command description
Electronic cam command	MC_GearIn	Gear operation start command
	MC_GearOut	Cancel gear action command
	MC_CombineAxes	Dual-spindle combined gear command
	MC_CamIn	Cam operation start command
	MC_CamOut	Cancel the cam action command
	MC_GenerateCamTable	Update Cam Table Command
	MC_GetCamTable	Get Cam Table Command
	MC_Phasing	Spindle phase shift command
	MC_GetCamTablePhase	Get cam table phase command
	MC_GetCamTableDistance	Get cam table displacement command
	MC_GenerateTappet	Updating ejector commands
	MC_RotaryCut	Flying shear command
	MC_ChasingCut	Trailing cut command

6.3.1 MC_GearIn (Gear operation start command)

Drawing Block:



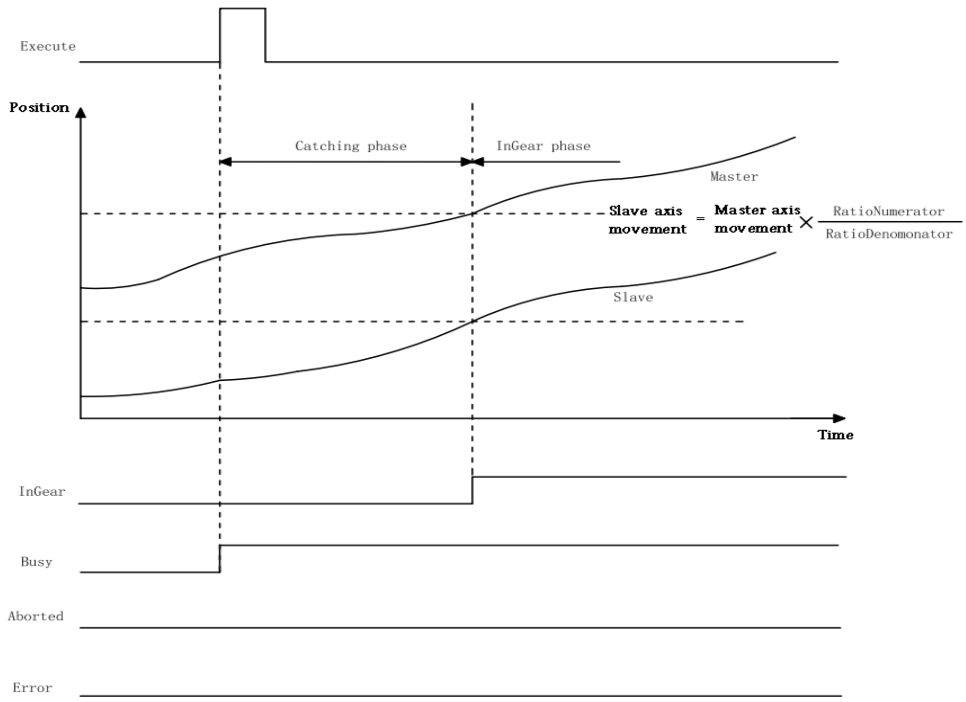
Command list format

Input Output	Name	Type of data	Applicable (soft components)	Scope	Describe	Enable
IN	Slave	WORD	constant/D/R/W	0~71	Slave axis name/axis number	Execute
IN	Master	WORD	constant/D/R/W	0~71	Spindle name/axis number	
IN	Reference Type	WORD	constant/D/R/W	0~2(default 1)	SpindleLocation type: 0: The command position of the previous cycle 1: Command position for this cycle 2: Feedback position of this cycle	
IN	Ratio Numerator	REAL	constant/D/R/W	positive/negative (default 1)	Electronic gear molecule	
IN	Ratio Denominator	REAL	constant/D/R/W	Positive number (default 1)	Electronic gear denominator	
IN	Acceleration	REAL	constant/D/R/W	Positive number (default 0)	Acceleration	
IN	Deceleration	REAL	constant/D/R/W	Positive number (default 0)	Deceleration	
IN	Jerk	REAL	constant/D/R/W	Positive number (default 0)	Speed step	
IN	Buffer Mode	WORD	Constant/D/R/W	- (default 0)	Buffer mode	
OUT	In Gear	BOOL	M/S	TRUE, FALSE	Sync arrival flag	
OUT	Busy	BOOL	M/S	TRUE, FALSE	Running sign	

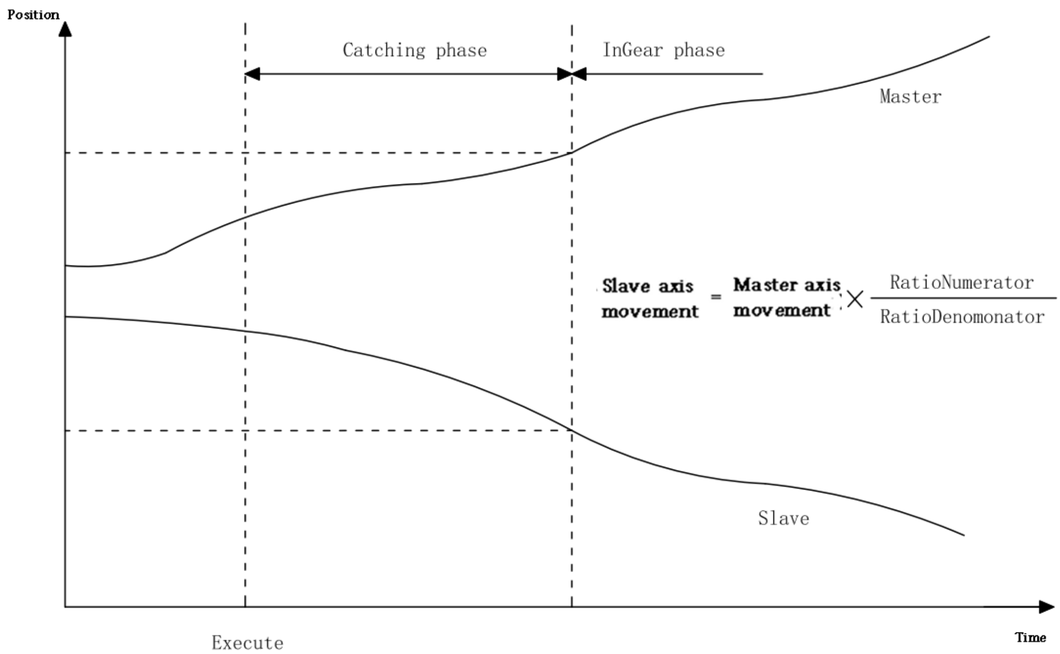
OUT	Active	BOOL	M/S	TRUE, FALSE	In control	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interrupt	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error flag	
OUT	Error ID	WORD	D/R/W	0x0~0xffff	Error ID	

Function and command description

After the operation starts, the slave axis performs acceleration/deceleration operation at the speed obtained by multiplying the main axis speed by the gear ratio as the target speed. Before reaching the target position, it is called Catching phase (in chasing), and after reaching it is called InGear phase (in gear synchronization). When the gear ratio is positive, the slave station and the main shaft move in the same direction.

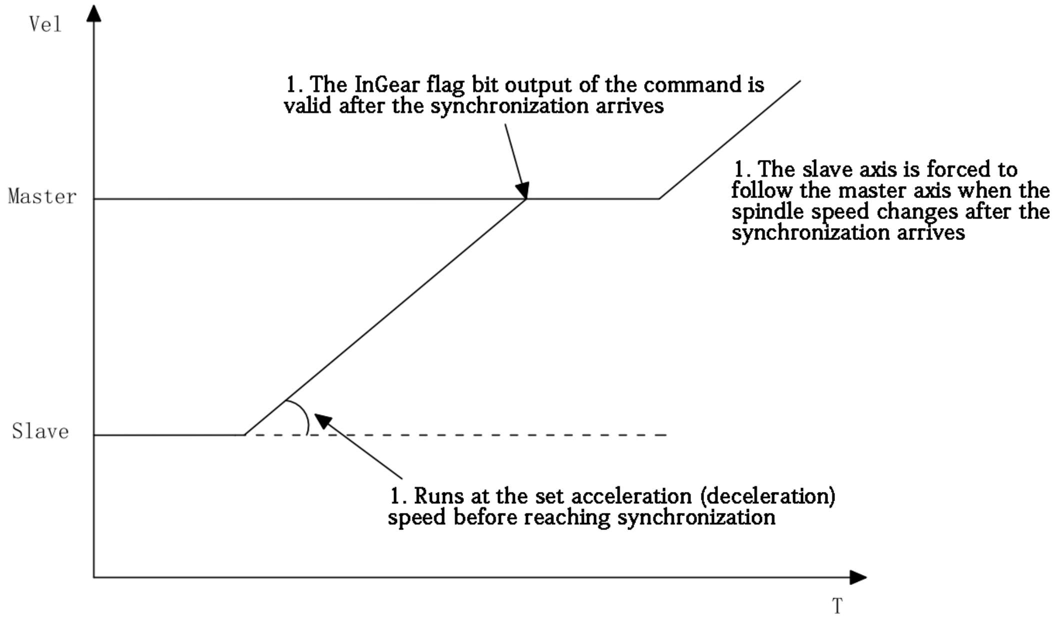


With a negative gear ratio, the slave axis moves in the opposite direction to the master axis.

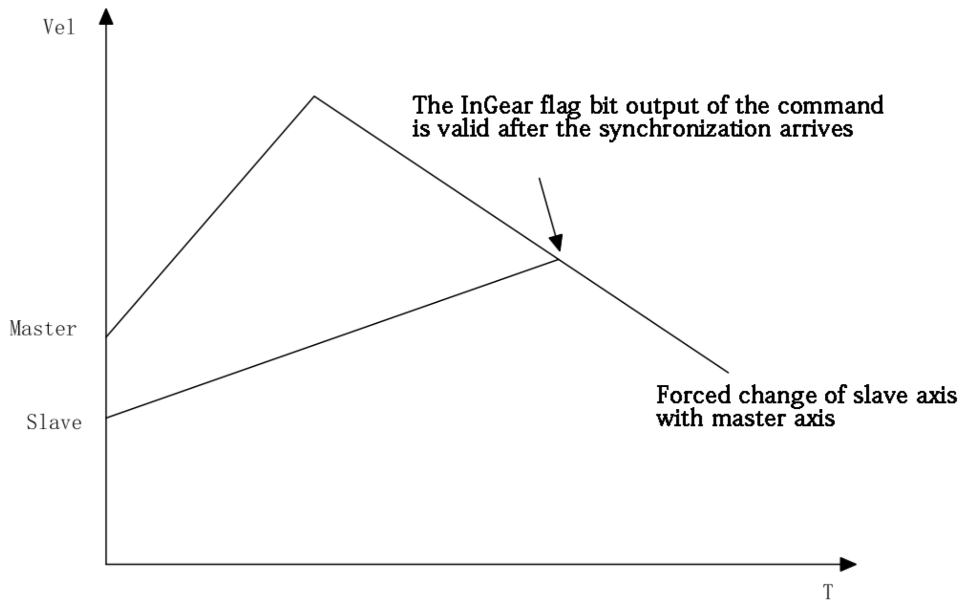


Before synchronization is achieved, the slave axis moves according to the set acceleration (deceleration) speed. When the speed of the slave axis is equal to the speed of the master axis multiplied by the gear ratio, it is considered that the gears are engaged. After that, the slave axis completely follows the change of the master axis.

Case 1: The spindle keeps moving at a constant speed before synchronization (triggering the gear command according to the gear ratio of 1:1)

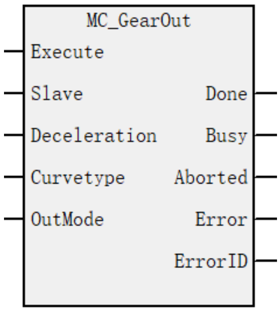


Case 2: Spindle variable speed motion before synchronization (gear command triggered according to 1:1 gear ratio)



6.3.2 MC_GearOut (Cancel gear action command)

Drawing Block:



Function and command description

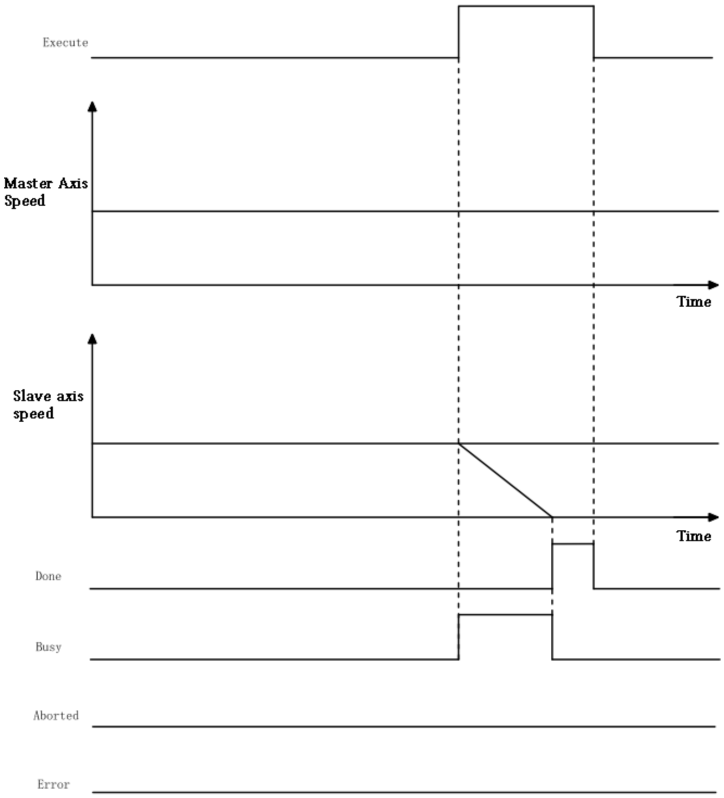
Input Output	Name	Type of data	Applicable (soft components)	Scope	Describe	Enable
IN	Slave	WORD	constant/D/R/W	0~71	Slave axis name/axis number	Execute
IN	Deceleration	REAL	constant/D/R/W	0~1(default 0)	Positive number of deceleration: stop according to deceleration 0: stop immediately	
IN	Curve type	WORD	constant/D/R/W	0 (default 0)	speedtype 0: T-shaped speed curve	
IN	OutMode	WORD	constant/D/R/W	0 (default 1)	Synchronous mode deselect: 0: Decelerate to stop 1: stop immediately	
OUT	Done	BOOL	M/S	TRUE/FALSE	complete stop	
OUT	Busy	BOOL	M/S	TRUE/FALSE	Running sign	
OUT	Aborted	BOOL	M/S	TRUE/FALSE	execution interrupt	
OUT	Error	BOOL	M/S	TRUE/FALSE	error flag	
OUT	Error ID	WORD	D/R/W	0x0~0xffff	error ID	

Function and command description

Specify the action target axis with the Slave (slave axis), specify the Deceleration (deceleration), and abort the MC_GearIn (gear motion start) command that is being executed. This command has no effect on the spindle operation of the MC_GearIn (gear operation start) command.

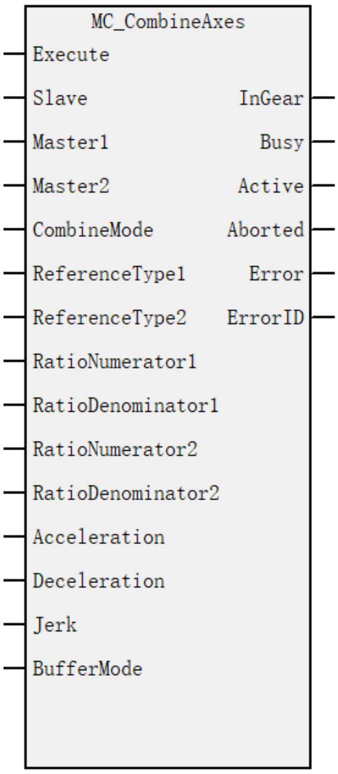
Timing diagram

Execute deceleration stop



6.3.3 MC_CombineAxes (Dual-spindle combined gear command)

Drawing Block:



Function and command description

Input Output	Name	Type of data	Applicable (soft components)	Scope	Describe	Enable
IN	Slave	WORD	constant/D/R/W	0~71	Slave axis name/axis number	Execute
IN	Master1	WORD	constant/D/R/W	0~71	No. 1 spindle name/axis number	
IN	Master2	WORD	constant/D/R/W	0~71	No. 2 Spindle name/axis number	
IN	Combine Mode	WORD	constant/D/R/W	0~1(default 0)	Synthetic mode: 0: 1+2 1: 1-2	
IN	ReferenceType1	WORD	constant/D/R/W	0~2(default 1)	Location type 0: The command position of the previous cycle 1: Command position for this cycle 2: Feedback position of this cycle	
IN	ReferenceType2	WORD	constant/D/R/W	0~2(default 1)	Location type 0: The command position of the previous cycle 1: Command position for this cycle 2: Feedback position of this cycle	
IN	RatioNumerator1	REAL	constant/D/R/W	positive/negative(default 1)	No. 1 electronic gear molecule	
IN	RatioDenominator1	REAL	constant/D/R/W	Positive number (default 1)	No. 1 Electronic gear denominator	
IN	RatioNumerator2	REAL	constant/D/R/W	positive/negative (default 1)	No. 2 electronic gear molecule	
IN	RatioDenominator2	REAL	constant/D/R/W	Positive number (default 1)	No. 2 Electronic gear	

					denominator	
IN	Acceleration	REAL	constant/D/R/W	Positive number (default 0)	Acceleration	
IN	Deceleration	REAL	constant/D/R/W	Positive number (default 0)	Deceleration	
IN	Jerk	REAL	constant/D/R/W	Positive number (default 0)	Speed step	
IN	BufferMode	WORD	Constant/D/R/W	- (default 0)	Buffer mode	
OUT	InGear	BOOL	M/S	TRUE, FALSE	Sync arrival flag	
OUT	Busy	BOOL	M/S	TRUE, FALSE	Running sign	
OUT	Active	BOOL	M/S	TRUE, FALSE	In control	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interrupt	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error flag	
OUT	Error ID	WORD	D/R/W	0x0~0xffff	Error ID	

Function and command description

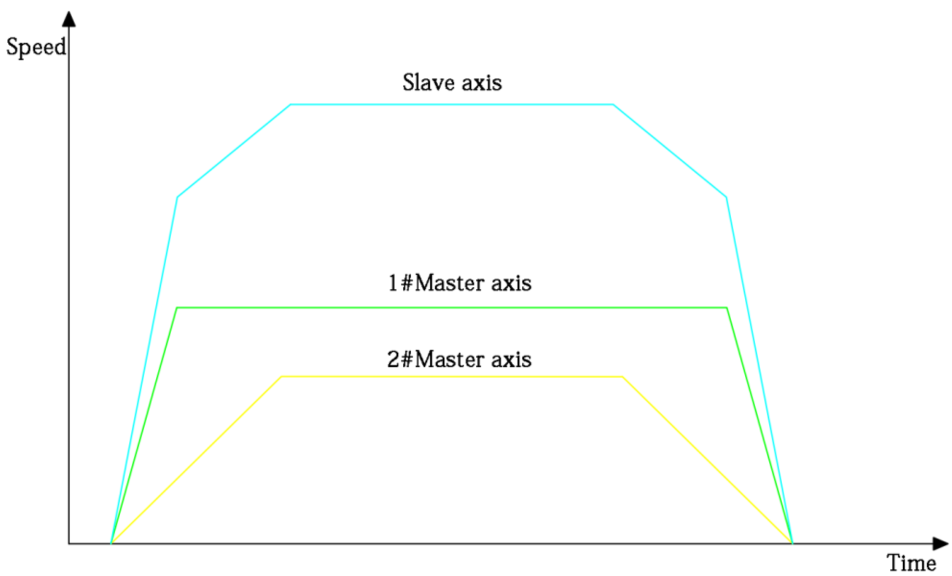
The value of adding or subtracting the positions of the two master axes is output as the slave axis position.

· There are two ways to synthesize this command: addition or subtraction

Add or subtract the position change of the No. 1 spindle and the position change of the No. 2 spindle and output the calculated value as the position change of the slave axis.

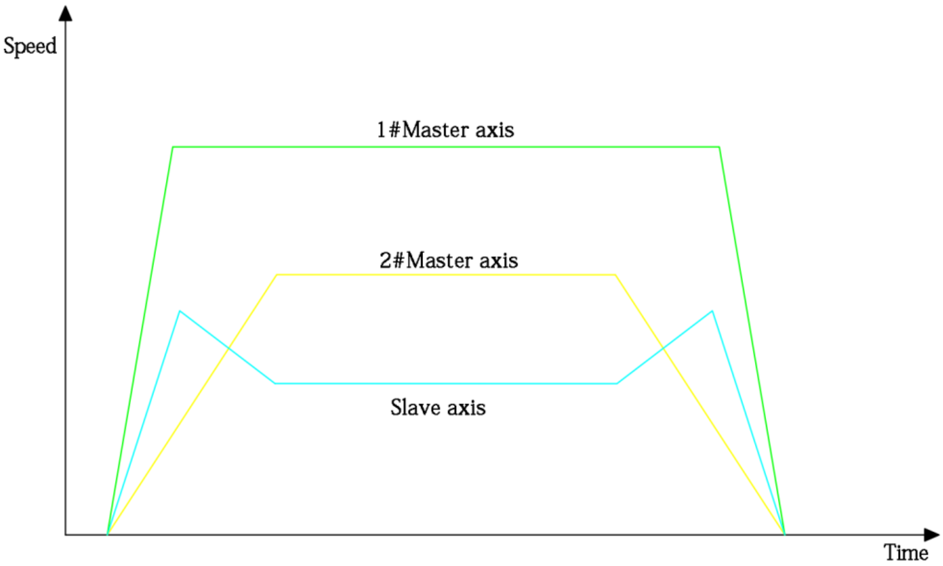
· When CombineMode value is 0

$$\text{Slave axis change amount} = \text{No.1 master axis change amount} \times \frac{\text{No.1 main wheel gear ratio molecule}}{\text{No.1 main wheel gear ratio denominator}} + \text{No.2 master axis change amount} \times \frac{\text{No.2 main wheel gear ratio molecule}}{\text{No.2 main wheel gear ratio denominator}}$$



· When CombineMode value is 1

$$\text{Slave axis change amount} = \text{No.1 master axis change amount} \times \frac{\text{No.1 main wheel gear ratio molecule}}{\text{No.1 main wheel gear ratio denominator}} + \text{No.2 master axis change amount} \times \frac{\text{No.2 main wheel gear ratio molecule}}{\text{No.2 main wheel gear ratio denominator}}$$

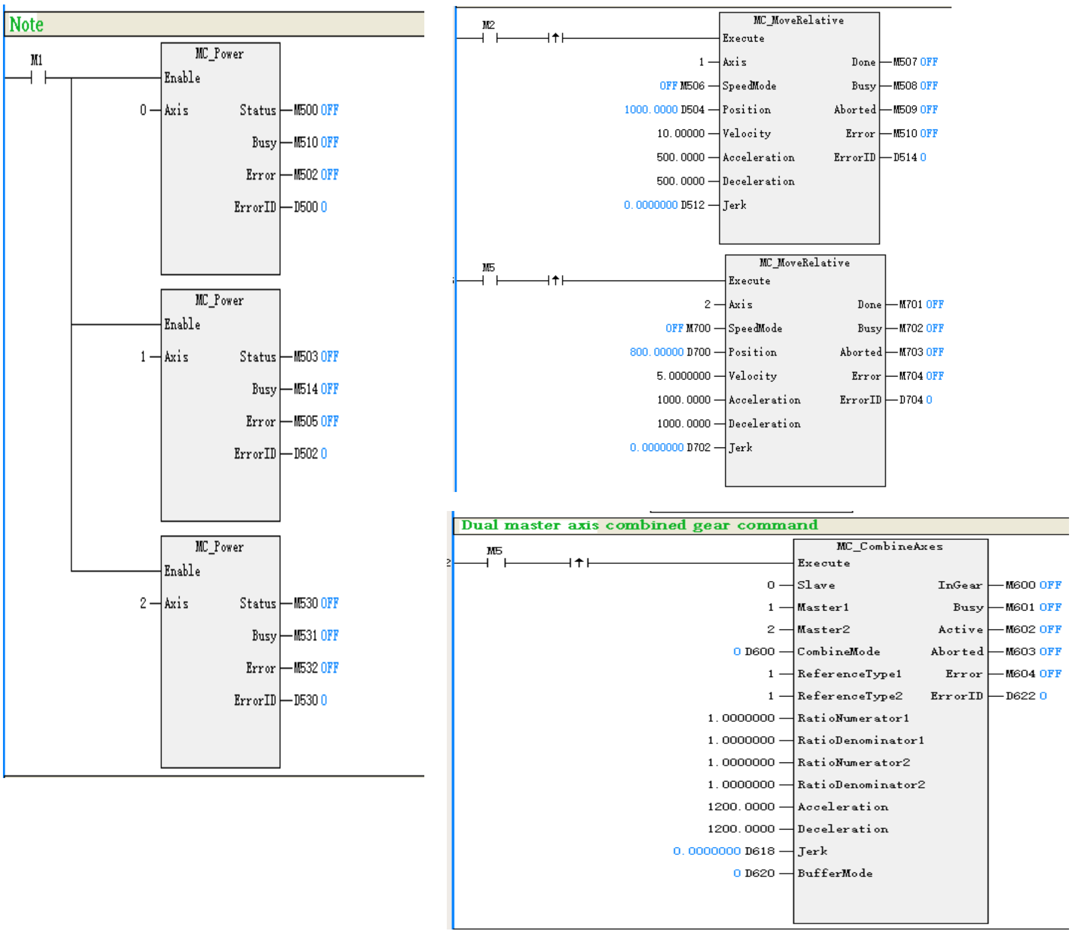


· The numerator and denominator settings of the main shaft gear ratio are the factors that adjust the position change of the two main shafts.

· Acceleration, decelerationand speed stepIndicates that the master axis is already in motion before this command is executed. At this time, if this command is executed, the slave axis willspeed stepAccelerate or decelerate to synchronize with the spindle position change. After synchronization, InSync is TRUE, and the command execution is completed.

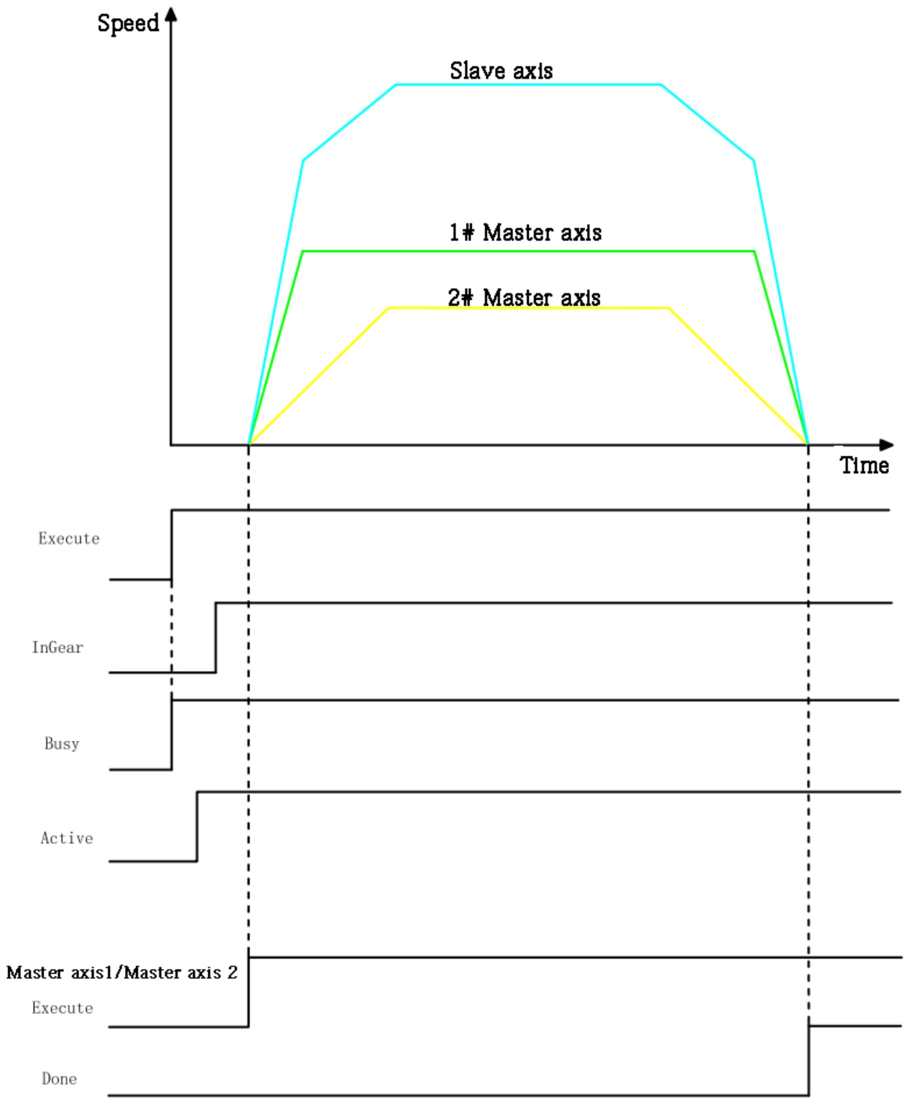
· To end the master-slave relationship of this command, use other motion commands to control the slave axis (such as MC_Stop).

Program example



Motion curve and timing diagram

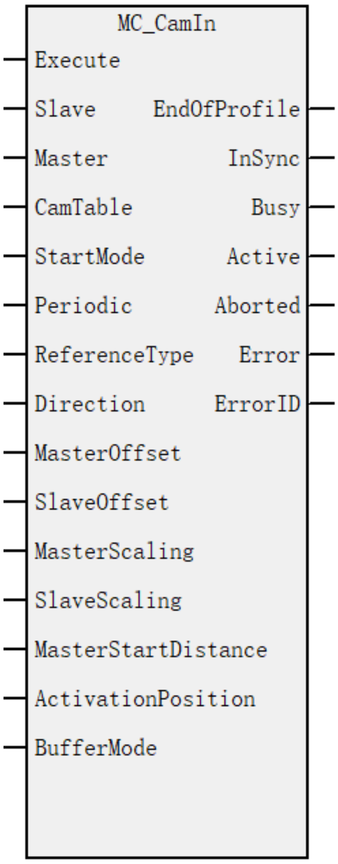
	Channel name	Component Type	Color	Vertical axis scaling	Min	Max
D800	Slave axis speed	REAL		600. 000000	0. 000000	15. 000000
D806	Master 1 axis speed	REAL		900. 000000	0. 000000	10. 000000
D814	Master 2 axis speed	REAL		1800. 000000	0. 000000	5. 000000



· When command Execute changes from FALSE to TRUE, the MC_CombineAxes command starts to execute, after a period of time, the command executes successfully, CombinA_InGearWhen it becomes TRUE, the three axes reach the synchronized motion state according to the command requirements. At this time, the Execute of the MC_MoveRelative command of the two spindles is changed to TRUE, and the two spindles start to move. At this time, the slave axis also starts to move according to the sum of the position changes of the two spindles. The position change of the slave axis per unit time is the sum of the position changes of the two spindles. After the execution of the spindle command is completed, the three axes are still in a synchronized state. To interrupt the synchronization state of the three axes, use the MC_Stop command to interrupt the slave axis to cancel the synchronization state.

6.3.4 MC_CamIn (Cam operation start command)

Drawing Block:



Function and command description

Input Output	Name	Type of data	Applicable (soft components)	Scope	Describe	Enable
IN	Slave	WORD	constant/D/R /W	0~71	Slave axis name/axis number	Execute
IN	Master	WORD	constant/D/R /W	0~71	Spindle name/axis number	
IN	CamTable	WORD	constant/D/R /W	0~15	Cam table	
IN	StartMode	WORD	constant/D/R /W	0~2(default 2)	Spindle tracking distance method: 0: absolute mode 1: Relative mode 2: start now	
IN	Periodic	WORD	constant/D/R /W	0-32767 (Default 0)	Repeat pattern: 0: keep looping Others: Loop the	

					specified number of times	
IN	ReferenceType	WORD	constant/D/R/W	0~2(default 1)	Spindle position type: 0: Command position in the previous cycle 1: Command position in this cycle 2: Feedback position in this cycle	
IN	Direction	WORD	constant/D/R/W	0~2(default 2)	Direction selection 0: Specify the positive direction 1: Specify the negative direction 2: No direction is specified	
IN	MasterOffset	REAL	constant/D/R/W	positive/negative/0(default 0)	Spindle position offset	
IN	SlaveOffset	REAL	constant/D/R/W	positive/negative/0(default 0)	Offset from axis position	
IN	MasterScaling	REAL	constant/D/R/W	Positive number(default 1)	Spindle position scaling ratio	
IN	SlaveScaling	REAL	constant/D/R/W	positive/negative(default 1)	Scaling from axis position	
IN	MasterStartDistance	REAL	constant/D/R/W	positive/negative/0(default 0)	Spindle tracking distance	
IN	ActivationPosition	REAL	constant/D/R/W	Positive number(default0)	Cam table start position	
IN	BufferMode	WORD	Constant/D/R/W	- (default 0)	Buffer mode 0: wait for the previous run to complete Other: reserved	
OUT	EndOfProfile	BOOL	M/S	TRUE/FALSE	End point in cam table	
OUT	InSync	BOOL	M/S	TRUE/FALSE	Sync arrival flag	
OUT	Busy	BOOL	M/S	TRUE/FALSE	Running sign	

OUT	Active	BOOL	M/S	TRUE, FALSE	In control	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interrupt	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Relative cam table

The phase and displacement of the cam table are specified in relative amounts from the starting point of 0.0. In each EtherCAT cycle, the cam calculation unit calculates the displacement of the slave axis corresponding to the phase of the master axis according to the selected cam curve type.

Cam table

	Phases	Displacement
Start	0	0
	80	30
	160	50
	240	20
End	360	0

Each task cycle calculates the command position when the electronic cam acts according to the curve type between the key points.

Command start condition

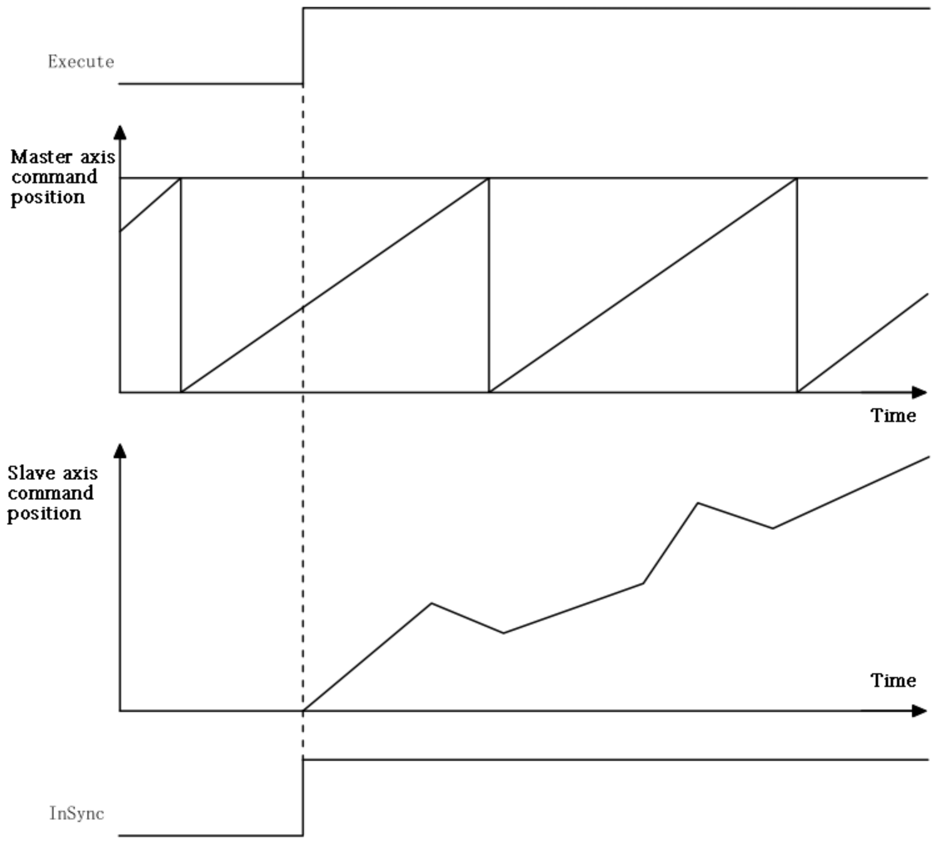
This command can be activated in any state of spindle stop, position control, speed control, or synchronous control. This command can be started when the slave axis is in StandStill state, Discrete Motion state, Continuous Motion state and Synchronized Motion (non-axis group motion).

Software limit

If the slave axis exceeds the software limit during the cam operation, an error will occur, and the operation will be stopped immediately.

Start of cam action

● When StartMode is set to 2 (start immediately), the cam action will be executed immediately after the command is started. The current position of the master axis is the phase 0 point of the cam, and the current position of the slave axis is the displacement zero point of the cam.



● When StartMode is set to 0-1 (specified position start)

After the start command, wait for the spindle to reach the StartPosition (cam table start position). When the spindle passes StartPosition (cam table start position), the start point of the cam table is executed. The phases and displacements of the cam table are specified in relative quantities from zero. Therefore, the absolute position of each axis of each phase is a relative value from the absolute position of each axis of the starting point of the cam table. For example, the count mode of the spindle is 0~360 ° rotation mode, and the cam table is shown in the figure below. StartPosition (cam table start position) =50. As shown below, the absolute position of the master axis is the phase of the cam table plus the value of StartPosition, and the absolute position of the slave axis is the displacement of the cam table plus the value of the absolute position of the slave axis at the starting point of the cam table.

Cam table

	Phases	Displacement
Start	0	0
	80	30
	120	50
	240	20
End	360	0

StartPosition=50 →

Absolute position of each axis

Master Axis	Slave Axis
50	0+ Absolute position of the slave axis at the start of the cam table
130	30+ Absolute position of the slave axis at the start of the cam table
170	50+ Absolute position of the slave axis at the start of the cam table
290	20+ Absolute position of the slave axis at the start of the cam table
50	0+ Absolute position of the slave axis at the start of the cam table

Then, when the MasterStartDistance (spindle tracking distance) is passed, the cam operation of the slave axis is started, and the output variable InSync (in synchronization) is turned on.

MasterStartDistance (spindle tracking distance) is specified as absolute position (StartMode = 0) or StartPosition (StartMode = 1) from StartPosition (cam table start position).

The set cam table is as follows:

Phase	Displacement
0	0
80	120
120	80
360	140

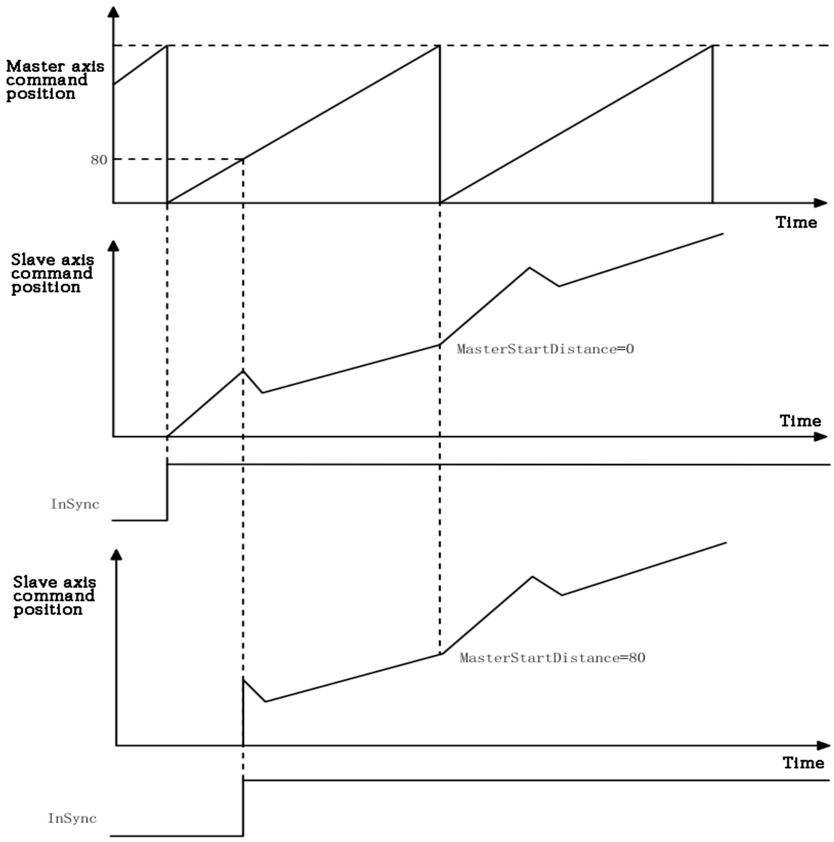
The starting conditions for the cam action are as follows:

Input variable	Condition 1	Condition 2
Periodic(repeat pattern)	0	0
StartMode(Start position mode command)	Relative position	Relative position
StartPosition (cam table start position)	0	0
MasterStartDistance (spindle tracking distance)	0	80

Under condition 1, when the master axis passes 0, the output variable InSync (in synchronization) turns on, and the slave axis starts cam operation.

Under condition 2, when the spindle passes 0, during spindle cam operation, when 80 is passed, the output variable InSync (in synchronization) turns ON, and the slave axis starts cam operation.

It should be noted that under condition 2, the cam action starts from the middle of the cam table, and the slave axis accelerates rapidly.



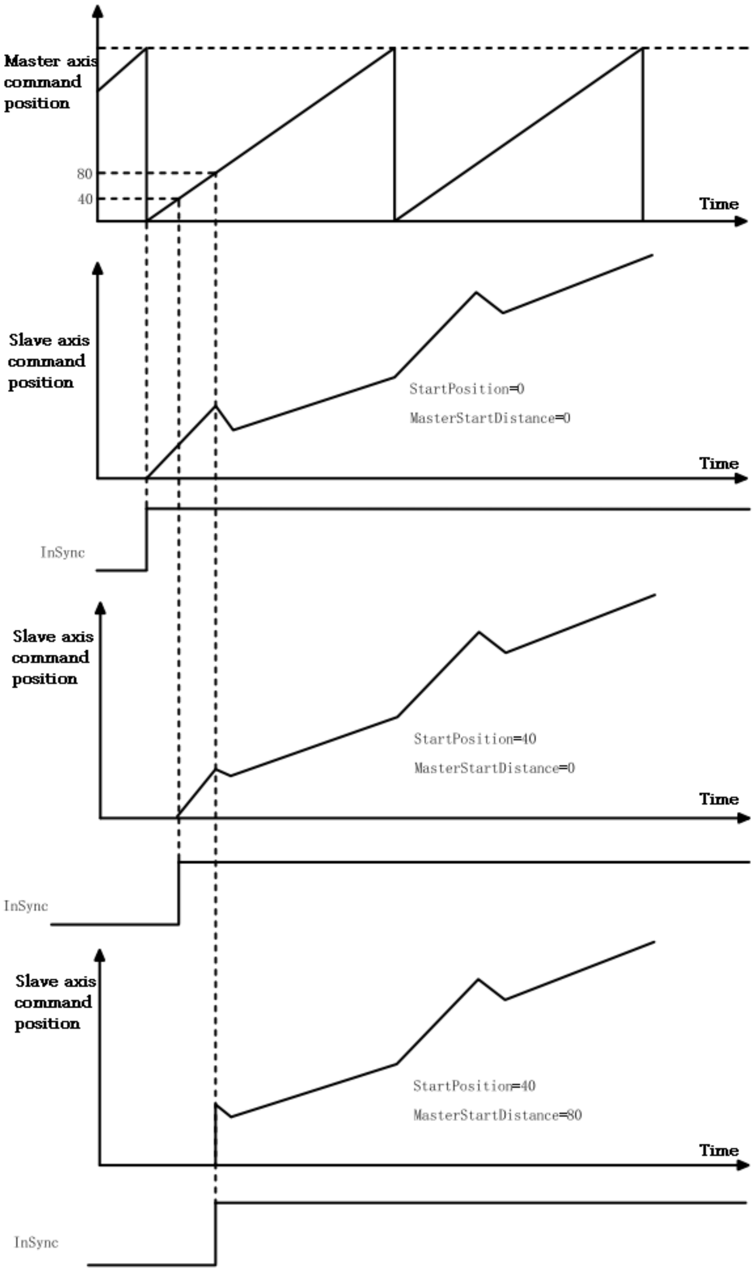
The above cam table, when modifying the starting conditions as follows:

Input variable	Condition 1	Condition 2	Condition3
Periodic(repeat pattern)	0	0	0
StartMode(Start position mode command)	Relative position	Relative position	Relative position
StartPosition (cam table start position)	0	40	40
MasterStartDistance (spindle tracking distance)	0	0	80

Under condition 1, when the spindle passes 0, Spindle Cam action, the output variable InSync (in synchronization) turns on, and the slave axis starts cam operation.

Under condition 2, when the spindle passes 40 specified by StartPosition (start position of cam table), Spindle Cam action, the output variable InSync (in synchronization) turns on, and the slave axis starts cam operation.

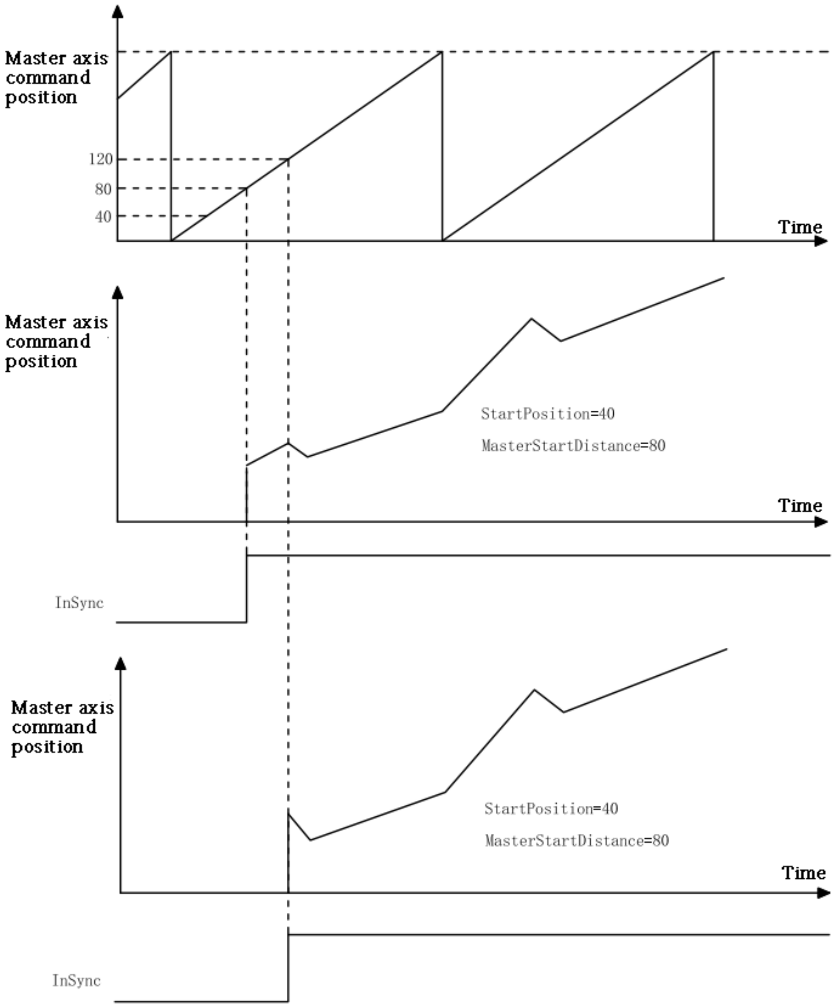
Under condition 3, when the spindle passes 40, the output variable InCam (in cam operation) turns ON, and when it passes 120, the output variable InSync (in synchronization) turns ON, and the slave axis starts cam operation.



With StartMode (start position specification), you can specify whether the specified value of MasterStartDistance (spindle tracking distance) should be treated as an absolute position or a relative position. The following explains the difference in the start of the cam operation of the slave axis due to the difference in StartMode (designation of the start position mode). The settings for the cam table are the same as in the previous example. The conditions for starting the cam operation are as follows.

Input variable	Condition 1	Condition 2
Periodic (repeat pattern)	0	0
StartMode (Start position mode command)	Absolute position	Relative position
StartPosition (cam table start position)	40	40
MasterStartDistance (spindle tracking distance)	80	80

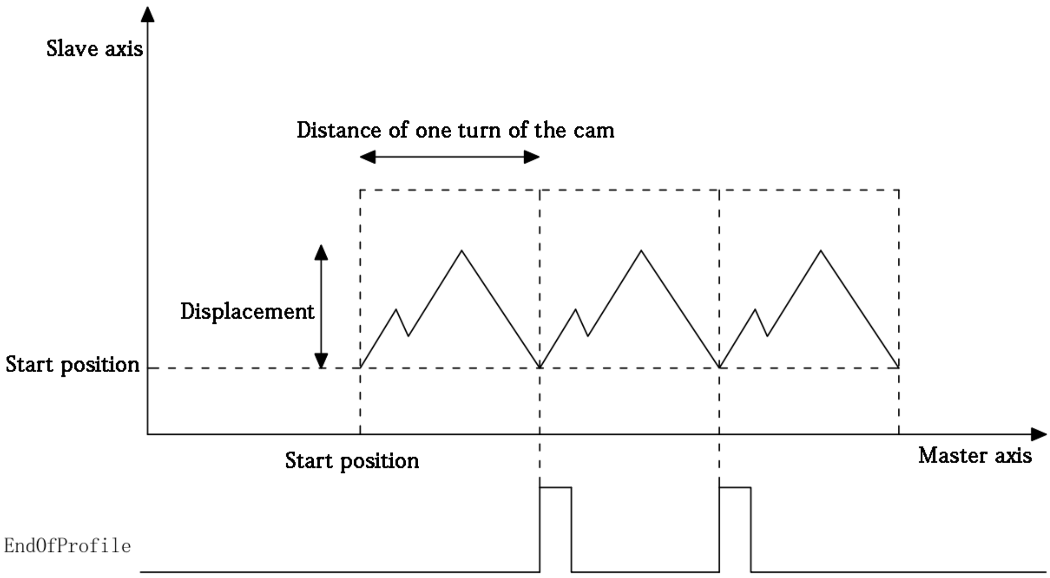
Under Condition 1 and Condition 2, when the spindle passes 40,SpindleCam action.
In Condition 1, since StartMode (designation of the start position method) is 0 (absolute position), when the master axis passes 80, the output variable InSync (in synchronization) turns on, and the slave axis starts cam operation.
In Condition 2, since StartMode (designation of the starting position method) is 1 (relative position), when the master axis passes 120 (=40+80), the output variable InSync (in synchronization) turns on, and the slave axis starts cam operation.



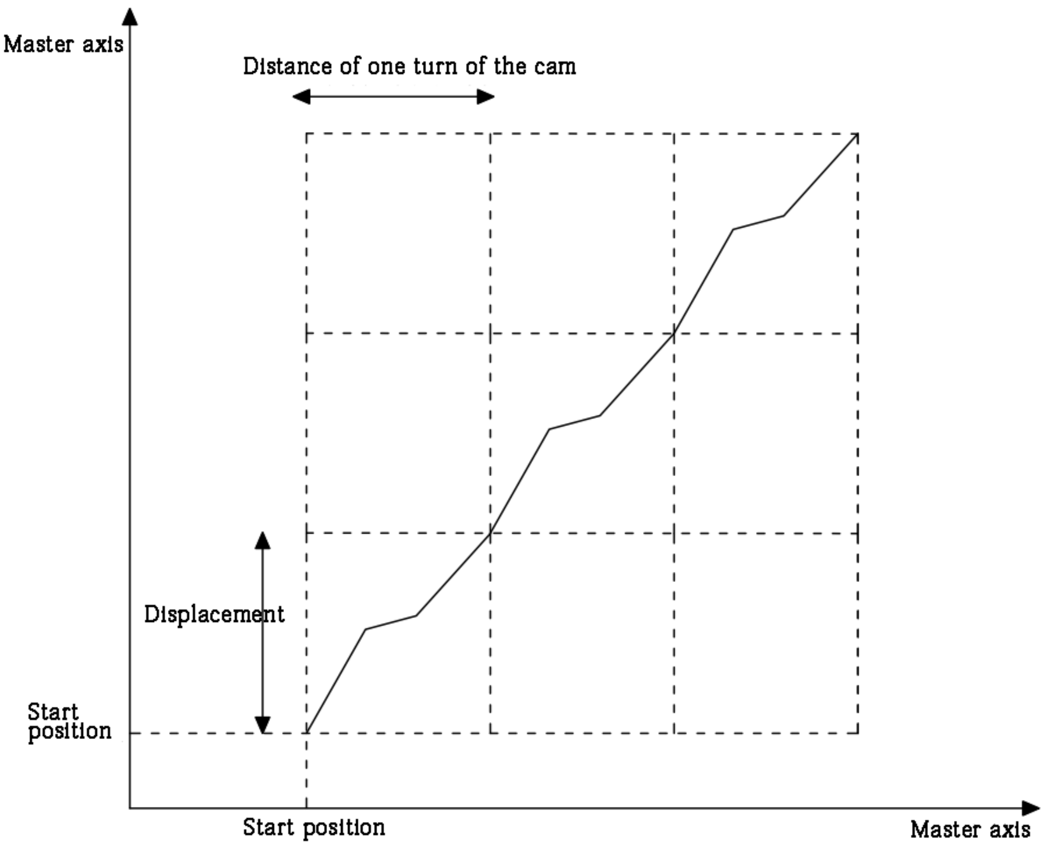
Repeat pattern

When 0 is specified in Periodic (repeated mode), the cam action is repeated from the start point to the end point of the cam table. After each cam period ends, EndOfProfile is set for one PLC scan period. When N (greater than 0) is specified, the cam action will end after N cycles. After the last cycle, if the Execute input is ON, the EndOfProfile will always be TRUE, and if the Execute is OFF, the EndOfProfile will be set for one PLC scan cycle . During the repeating process, if the stroke position of the slave axis is the same at the start and end points of the cam table, it operates as a reciprocating cam. If the stroke position of the slave axis is different at the start point and end point, it operates as a transfer cam.

・**Reciprocating cam action**



· **Transfer Cam action**

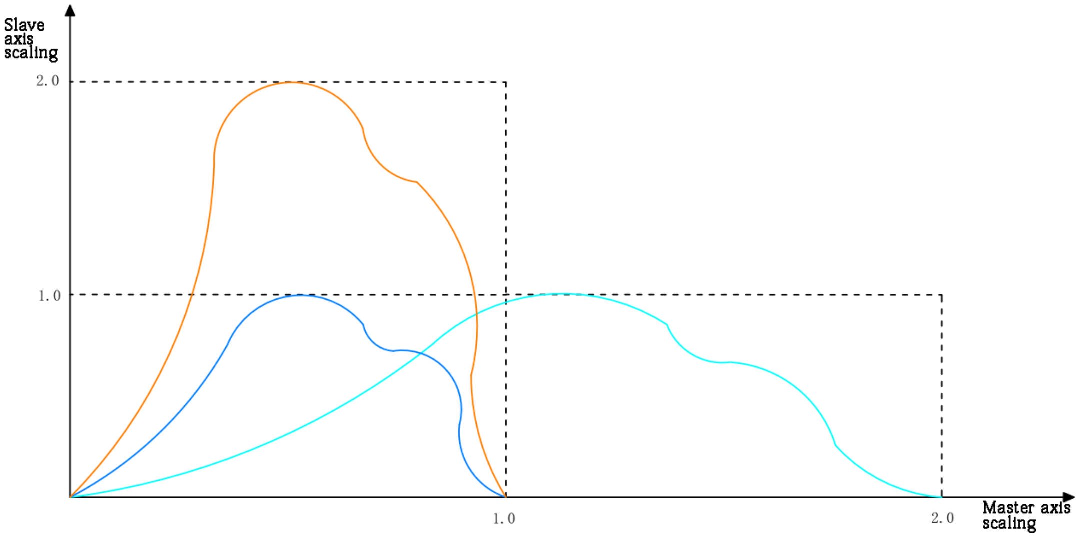


End of cam action

To end the cam operation in the middle, use the MC_CamOut (cam release) command or the MC_Stop (forced stop) command.

Coefficients (scaling)

The master phase and slave axis displacement of the specified cam table can be scaled according to the specified ratio. Different overrides can be set for the master and slave axes.

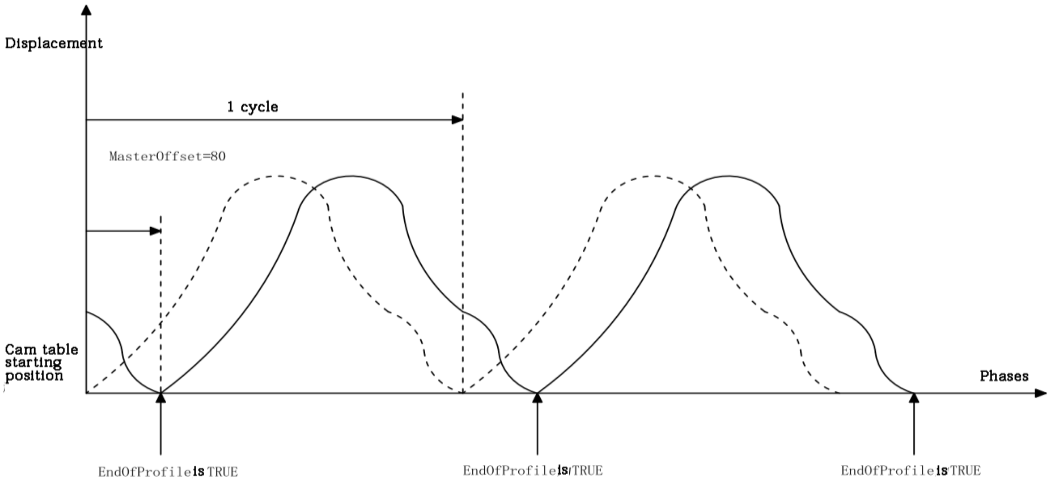


Bias

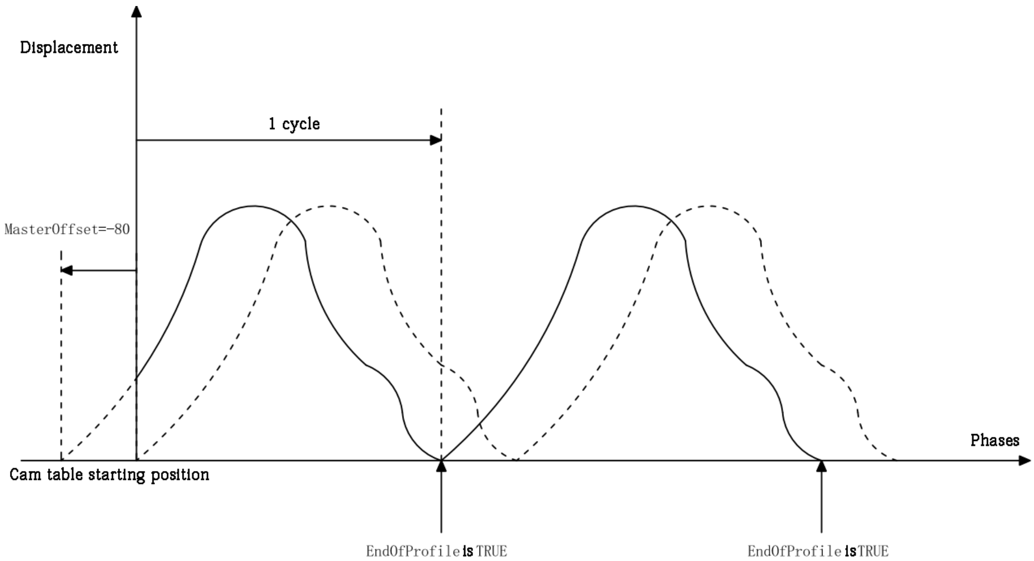
A function that shifts the phase and displacement by the offset amount for the specified cam table.

Different offsets can be specified for the master axis phase and slave axis displacement.

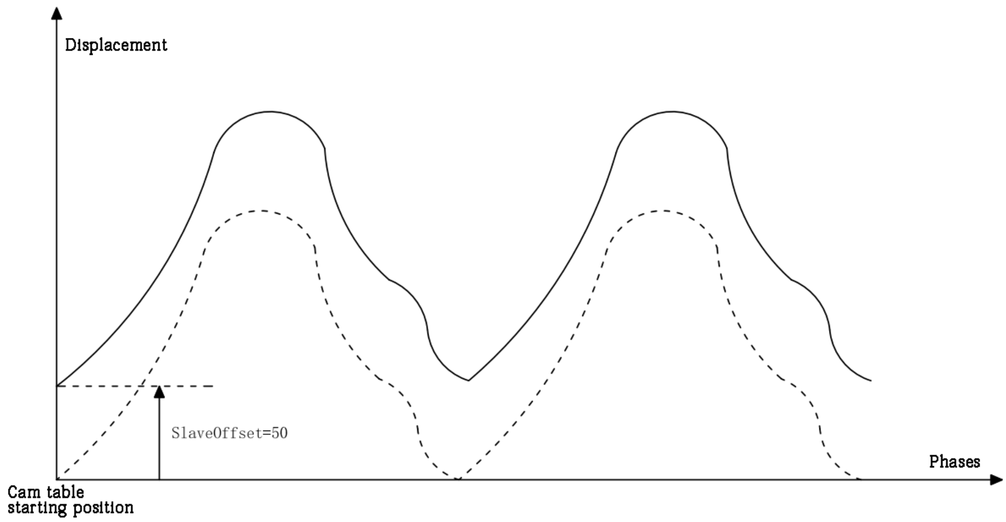
● When MasterOffset (Spindle offset)> 0



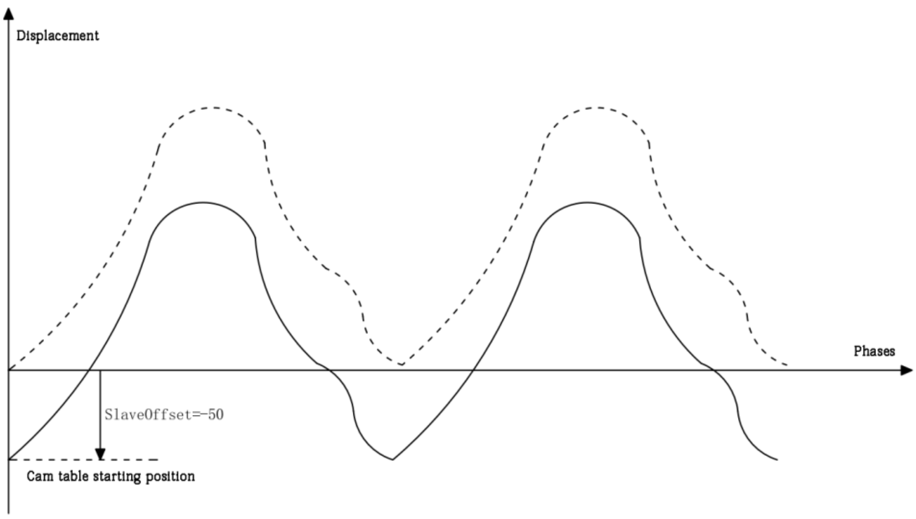
● When MasterOffset (spindle offset) < 0



● When SlaveOffset (slave axis offset) > 0



● When SlaveOffset (slave axis offset) < 0

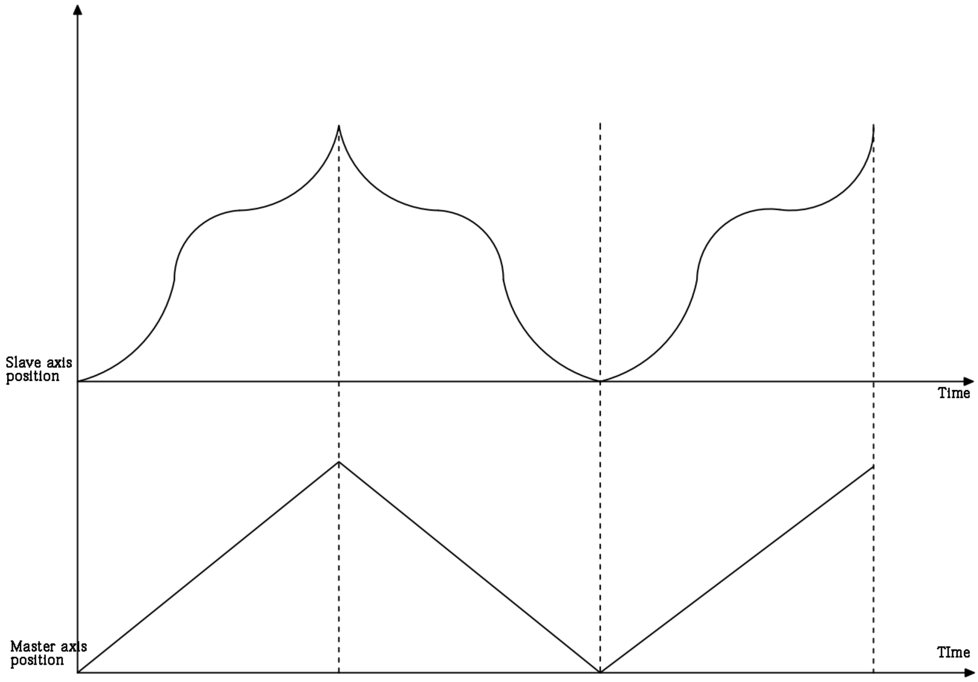


Direction (direction selection)

Only when the moving direction of the master axis is consistent with the Direction (direction selection), the cam action of the slave axis can be started.

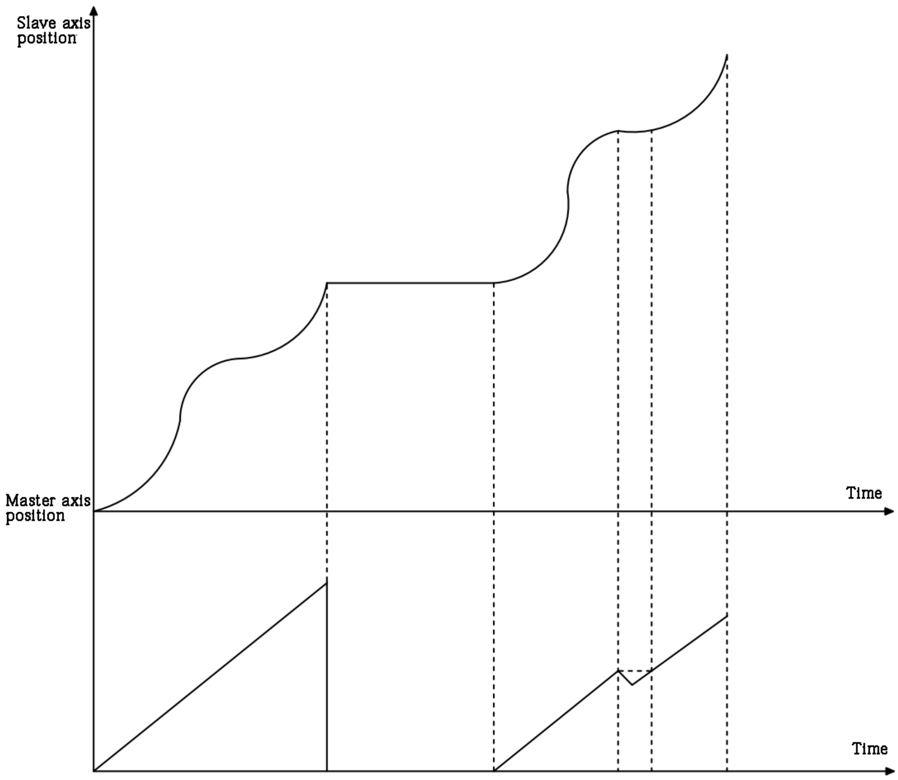
● No direction specification

The cam action starts regardless of whether the spindle moves in the positive or negative direction.



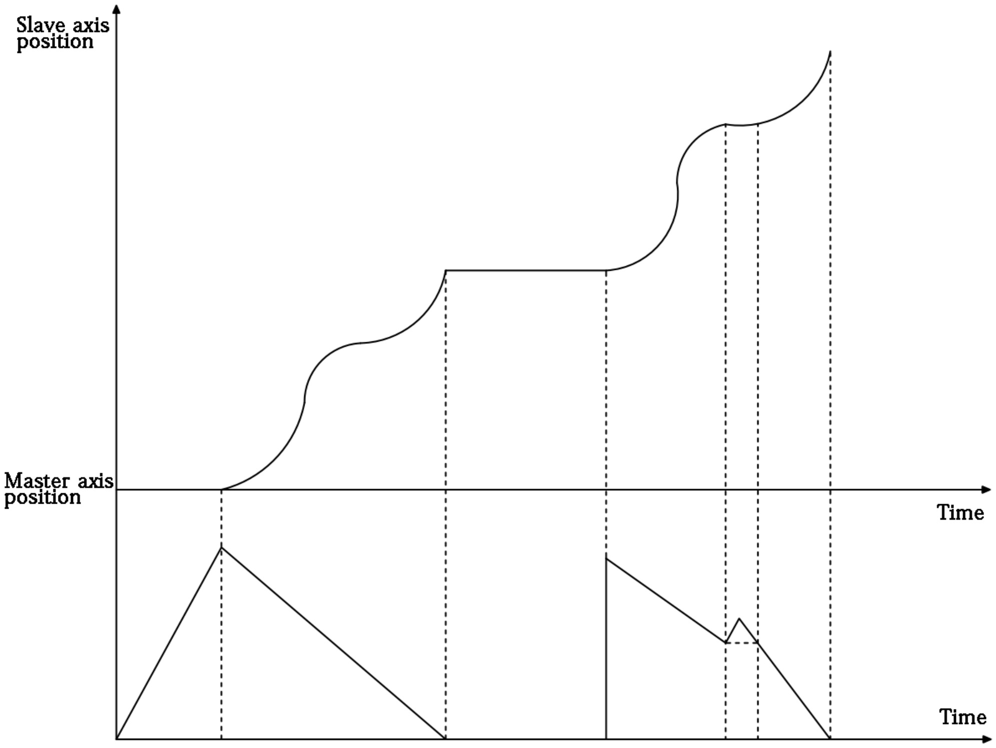
● Specify the positive direction

The cam action starts when the master axis moves in the positive direction. During a cam cycle, if the master axis is reversed, the slave axis will remain stationary until the master axis returns to its original position, and the slave axis will continue to follow the master axis to perform cam action.



● Specify the negative direction

The cam action starts when the master axis moves in the negative direction. In a cam cycle, if the master axis is reversed, the slave axis will remain stationary until the master axis returns to its original position, and the slave axis will continue to follow the master axis to perform cam action.



Position type selection

ReferenceType is used to set the data source of the spindle position.

When the spindle is the local encoder axis, the parameter setting is invalid, and it is always the feedback position of this cycle.

When the spindle is set as a bus servo axis and a local pulse axis, three modes can be set: the command position of the previous cycle, the command position of this cycle and the feedback position of this cycle.

Restart command

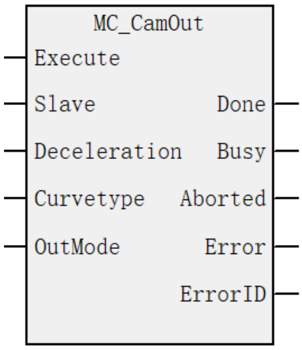
When the Busy signal of the MC_CamIn command is valid when the command is re-triggered, the parameters Periodic, MasterScaling, SlaveScaling, RefrenceType, and Direction will be cached and take effect in the next cam cycle.

Multiple boot

The second MC_CamIn command is triggered when the Busy signal of the MC_CamIn command is valid, the Busy signal of the second command is valid, and the Active signal is invalid. When a cam cycle ends, the first command is interrupted, and the Active signal of the second command is the output is valid, and the parameters Periodic, MasterScaling, SlaveScaling, ReferenceType, and Direction will take effect according to the parameters of the second command.

6.3.5 MC_CamOut (Cancle the cam action command)

Drawing Block:



Function and command description

Input Output	Name	Type of data	Applicable (soft components)	Scope	Describe	Enable
IN	Slave	WORD	constant/D/R/W	0~71	Slave axis name/axis number	Execute
IN	Deceleration	REAL	constant/D/R/W	Integer, 0	Positive number of deceleration: stop according to deceleration 0: stop immediately	
IN	Curvetype	WORD	constant/D/R/W	0 (default 0)	speedType 0: T-shaped speed curve	
IN	OutMode	WORD	constant/D/R/W	0~1(default 0)	Synchronous mode deselect: 0: Decelerate to stop 1: Stop immediately after executing the current cycle	
OUT	Done	BOOL	M/S	TRUE/FALSE	Complete stop	
OUT	Busy	BOOL	M/S	TRUE/FALSE	Running sign	
OUT	Aborted	BOOL	M/S	TRUE/FALSE	Execution interrupt	
OUT	Error	BOOL	M/S	TRUE/FALSE	Error flag	
OUT	Error ID	WORD	D/R/W	0x0~0xffff	Error ID	

Function description

Use this command to cancel the cam operation of the slave axis.

When Execute (start) is set to ON, the MC_CamIn command is interrupted and the interrupt flag is valid. If OutMode is set to 0, the deceleration action is performed according to Deceleration (deceleration). After deceleration to 0, the Done output is valid. Before the slave axis stops moving, the slave axis is in Continuous Motion mode; if OutMode is set to 1, the current Stop immediately after cyclic cam action, The slave axis is in synchronous motion mode until the cam action ends.

An error occurs when this command is activated for an axis that is not camming.

Repeat trigger

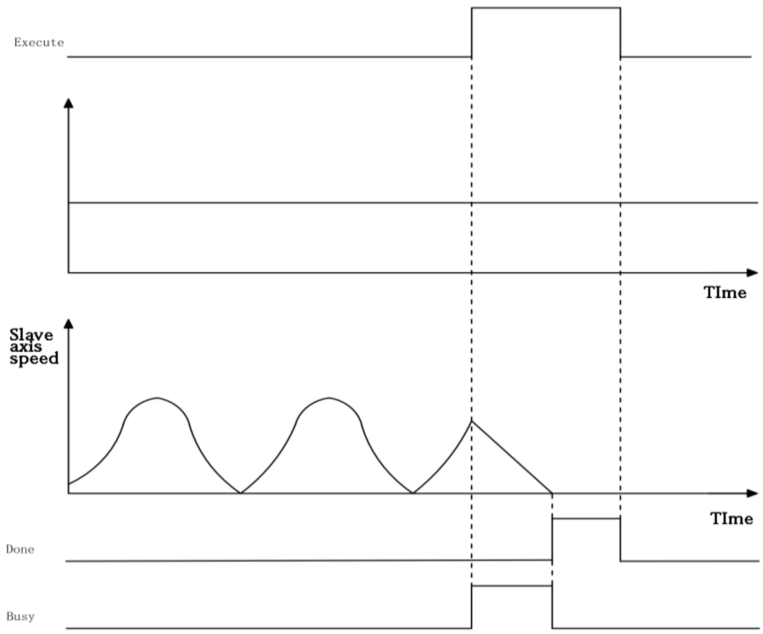
When the MC_CamOut command is re-triggered, the stop mode is performed according to the following rules:

Decelerate to stop	Stop immediately after executing the current cycle	The command reports an error, and the axis enters the standstill state after decelerating and stopping.
Decelerate to stop	Decelerate to stop	Stop at the new deceleration
Stop immediately after executing the current cycle	Decelerate to stop	Switch to deceleration stop mode

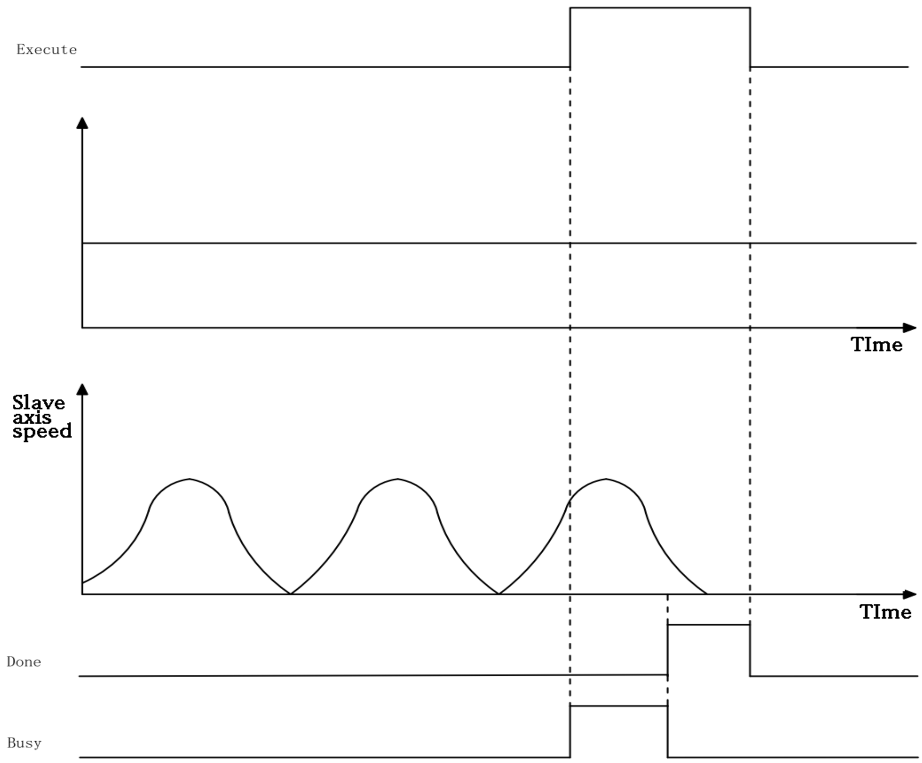
Stop immediately after executing the current cycle	Stop immediately after executing the current cycle	Stop after executing the current cycle
Decelerate to stop	Stop immediately after executing the current cycle	The command reports an error, and the axis enters the standstill state after decelerating and stopping.

Timing diagram

● Deceleration stop mode.

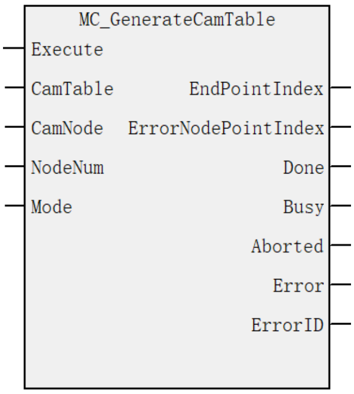


● Stop immediately after executing the current cycle.



6.3.6 MC_GenerateCamTable (Update cam table command)

Drawing Block:



Function and command description

Enter/output	Name	Type of data	Be applicable(device)	Scope	Describe	Enable
IN	CamTable	WORD	constant/D/R/W	0~15	Cam table	Execute
IN	CamNode	REAL	D/R/W		Cam node array: Array arrangement: (mPos[0],sPos[0], Speed[0],Acc[0],Type[0],…)	
IN	NodeNum	WORD	constant/D/R/W	0~361	Number of cam nodes: Less than 1 indicates that the original number of cam nodes is used	
IN	Mode	WORD	constant/D/R/W	0(Default 0)	Effective mode: 0: Take effect in the next cam cycle 1: Effective immediately 2: Valid at the next cam coupling	

OUT	EndPointIndex	WORD	D/R/W	WORD	End index	
OUT	EerrorNodePointIindex	WORD	D/R/W	WORD	Wrong node number	
OUT	Done	BOOL	M/S	TRUE/FALSE	Complete stop	
OUT	Busy	BOOL	M/S	TRUE/FALSE	Running sign	
OUT	Aborted	BOOL	M/S	TRUE/FALSE	Execution interrupt	
OUT	Error	BOOL	M/S	TRUE/FALSE	Error flag	
OUT	EerrorID	WORD	D/R/W	0x0~0xffff	Error ID	

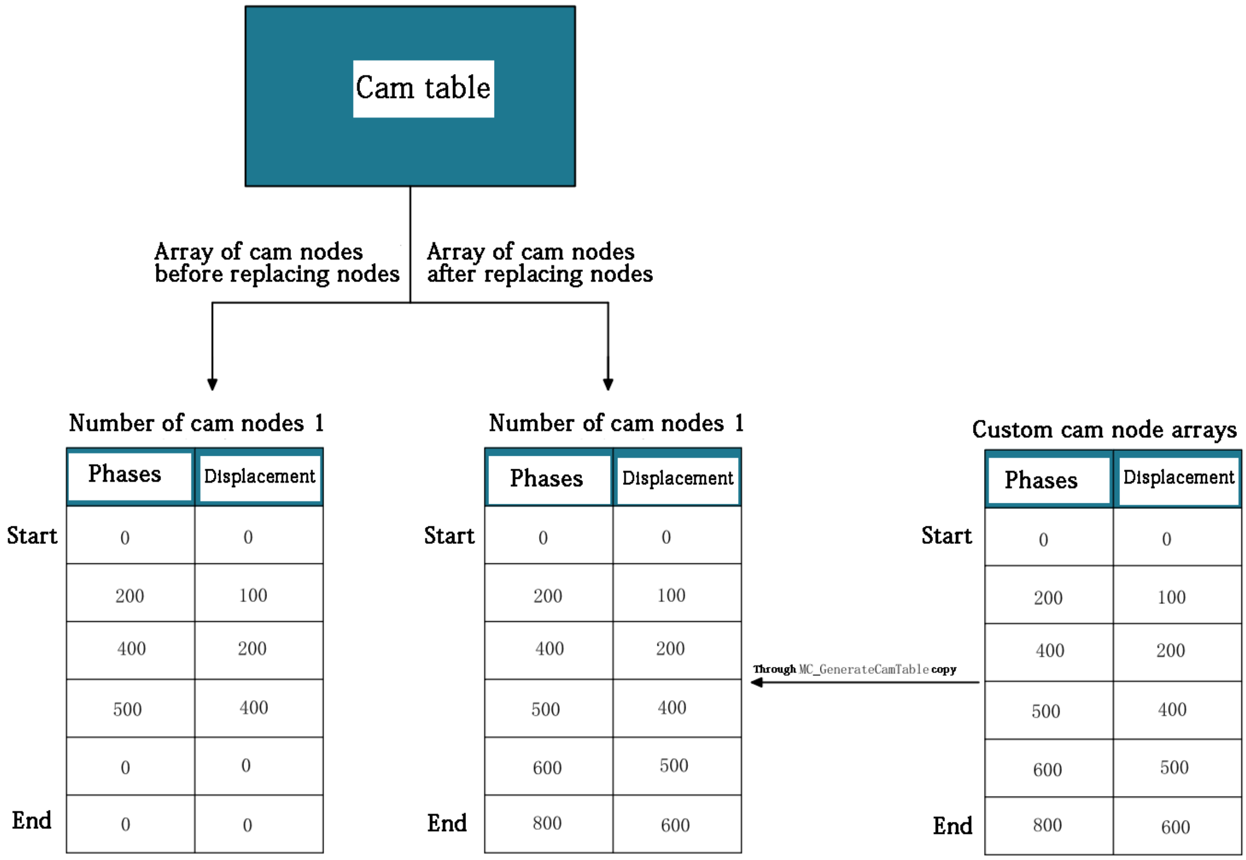
Functional Overview

At the rising edge of Execute (start), this command calculates the cam data based on the values of the input variables CamNode and NodeNum, and updates it to the cam table specified by CamTable, and takes effect in the next cam cycle.

CamNode variable function

The parameter CamNode is used to specify a new cam node array, the parameterCan not be empty.

Create a brand-new cam node array through the PLC program, copy the values in the cam node array to the cam table through the MC_GenerateCamTable command, and start executing in the next cam cycle.



Program example:

Number of cam nodes 1

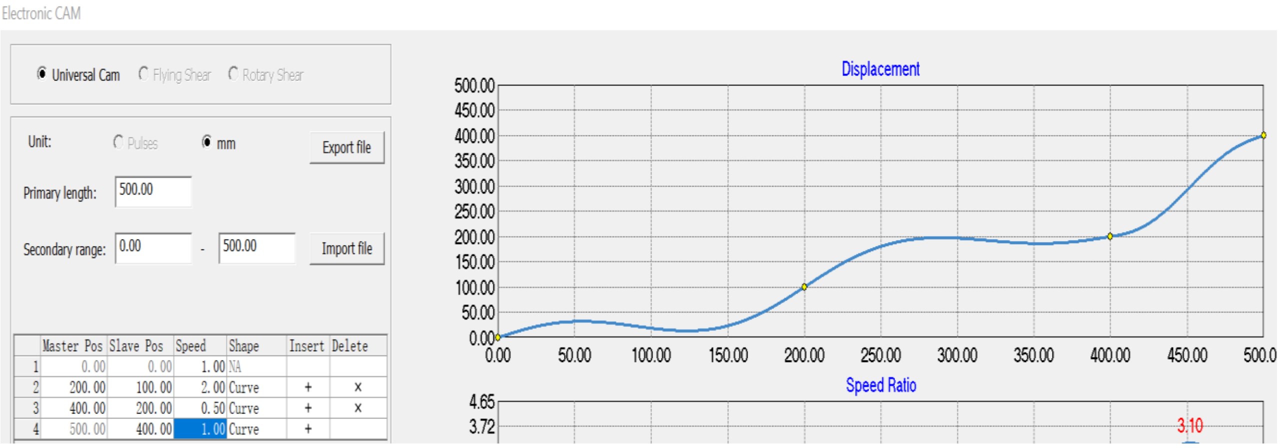
	Phases	Displacement
Start	0	0
	200	100
	400	200
	500	400
End	0	0
	0	0

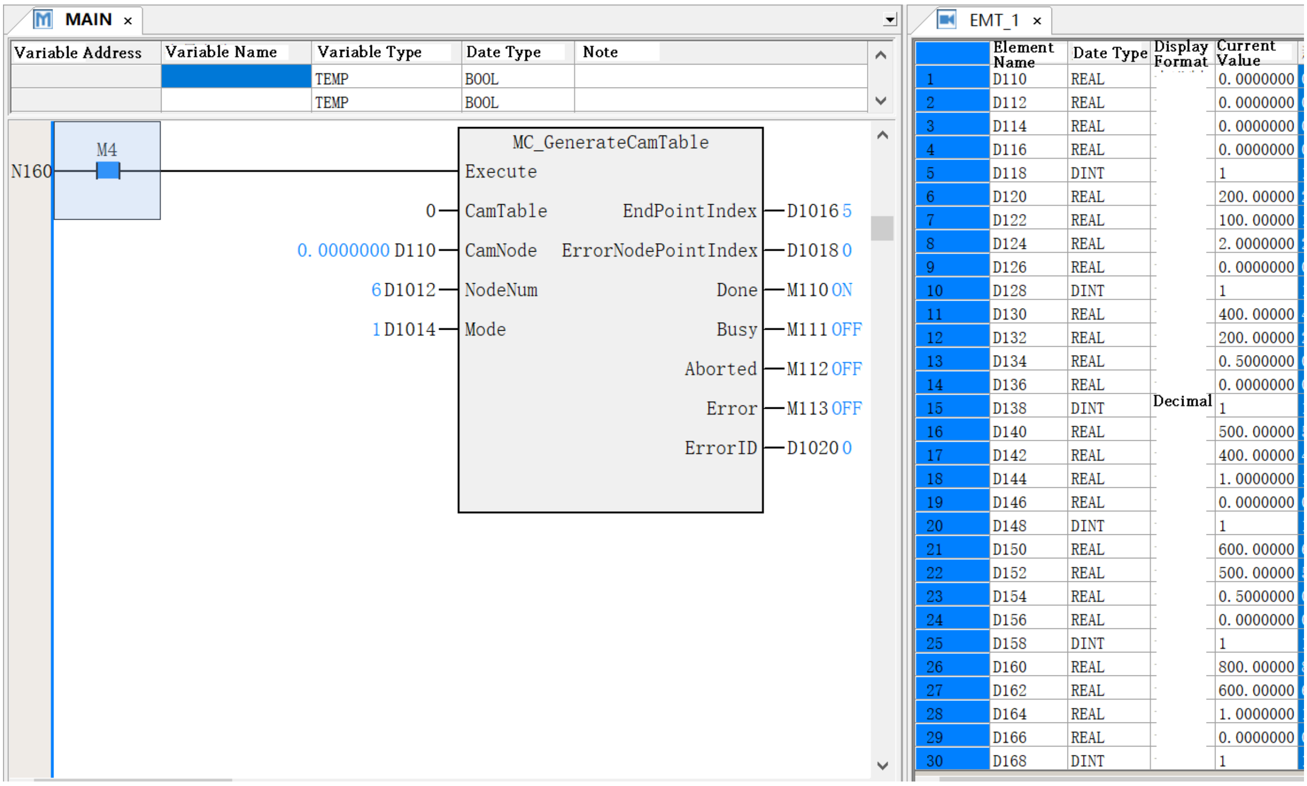
Add Node →

Number of cam nodes 1

	Phases	Displacement
Start	0	0
	200	100
	400	200
	500	400
	600	500
End	800	600

Replacing the cam before the node:

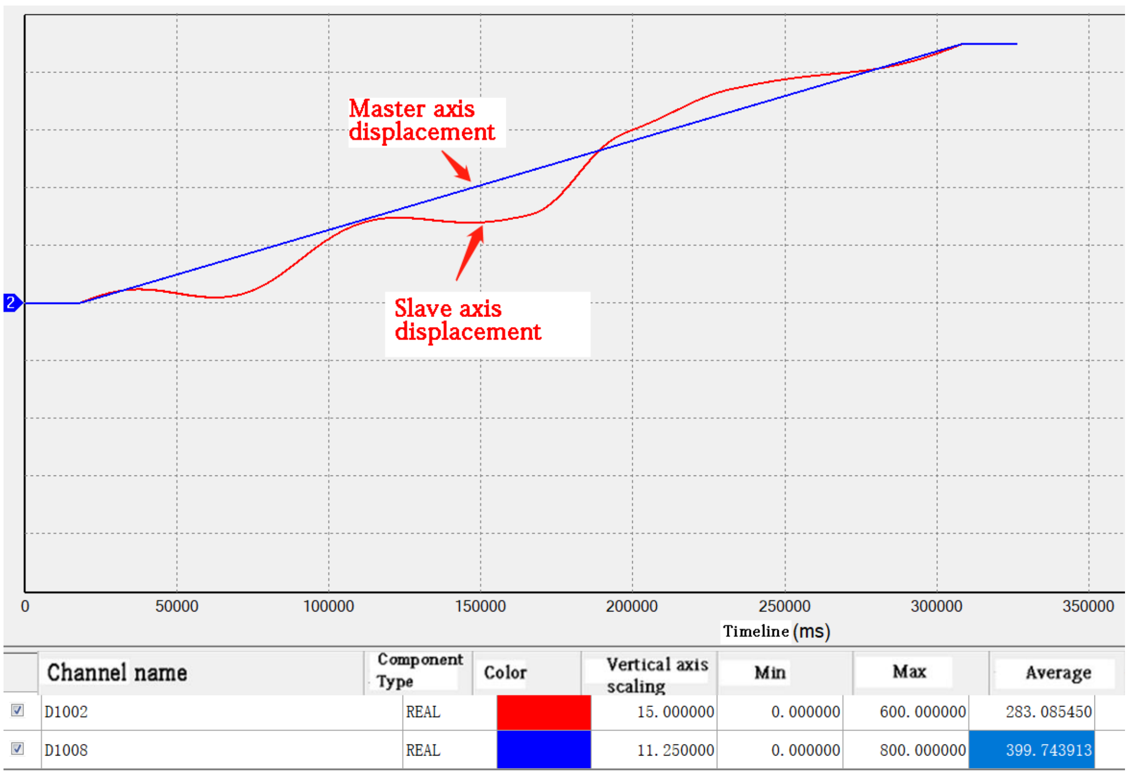




Node array data format

Ejector point	Address	Type of data		Notes
The first1nodes	D1	REAL		Spindle position
	D1+2	REAL		Slave position
	D1+4	REAL		speed
	D1+6	REAL		acceleration ratio
	D1+8	DINT		Curve type (0: straight line; 1: quintic curve)
The first2nodes	D1+10	REAL		Spindle position
	D1+10+2	REAL		Slave position
	D1+10+4	REAL		speed
	D1+10+6	REAL		acceleration ratio
	D1+10+8	DINT		Curve type (0: straight line; 1: quintic curve)
Nth node	D1+(n-1)*10	REAL		Spindle position
	D1+(n-1)*10+2	REAL		Slave position
	D1+(n-1)*10+4	REAL		speed
	D1+(n-1)*10+6	REAL		acceleration ratio
	D1+(n-1)*10+8	DINT		Curve type (0: straight line; 1: quintic curve)

Cams after node replacement:



NodeNum variable function

The parameter NodeNum is used to indicate the number of nodes in the newly generated cam table. When the parameter is empty, it means that the number of nodes in the cam table remains unchanged. When it is not empty, the value specified by

NodeNum is adopted.

Modify the number of key points in the cam table and take effect through the MC_GenerateCamTable command, and execute according to the new cam node in the next cam cycle.

Command parameter setting description

● The phase and displacement of the first point must be 0, otherwise an error will be reported.

● The absolute value of phase, displacement and speed ratio cannot be greater than 9999999, otherwise an error will be reported.

● The number of nodes cannot be greater than 361, otherwise an error will be reported.

● The number of nodes must be at least 2, otherwise an error will be reported.

● An error is reported when the phases are arranged in non-ascending order.

● It is required that the phase difference of two adjacent spindles must be greater than 0.0001, otherwise an error will be reported.

● The curve type of the node can only be set to straight line or quintic curve, otherwise an error will be reported.

Speed ratio adjustment rules

When this instruction is called, if the speed ratio of the key point is not set properly, the speed ratio of the cam node will be adjusted automatically, and the modification rules are as follows.

● When the current segment is a straight line, the speed ratio is automatically adjusted according to the formula.

For example, if points A1 and A2 form a straight line, the calculated speed ratio will be written into A2.

A1 point coordinates (x1,y1),A2 point coordinates (x2,y2), then the speed ratio value of straight line A1-A2 is::

$$v2 = \frac{y2 - y1}{x2 - x1}$$

● No adjustment when a straight line is connected immediately after a 5-times curve

● No adjustment is made when a 5-times curve is connected immediately after a 5-times curve.

When a straight line is linked to a straight line, the link speed of each segment is calculated separately, and the link speed ratio is allowed to change abruptly.

For example, if A1-A2 is the first straight line and A2-A3 is the second straight line, the speed ratio of the first curve is calculated and written into A2, and then the link speed of the second straight line is calculated and written into A3, and there is a sudden change in speed caused by the unequal link speeds of the first and second straight lines.

Restart motion command

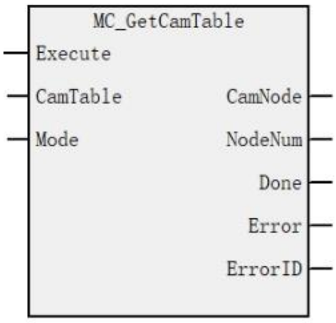
Re-trigger this command while the Busy signal of this command is valid, and the cam table will be modified according to the new parameters.

Multiple boot commands

When the Busy signal of this command is valid, a new MC_GenerateCamTable command is triggered, the command is interrupted, the CommandAborted signal output is valid, and the cam table will be modified according to the parameters of the newly triggered command.

6.3.7 MC_GetCamTable (Get cam table command)

Drawing Block:



Function and command description

Input Output	Name	Type of data	Applicable (soft components)	Scope	Describe	Enable
IN	CamTable	WORD	Constant/D/ R/W	0~15	Cam table	Execute
IN	Mode	WORD	Constant/D/ R/W	0(Default 0)	read model: 0:currentRunning cam table data Other: reserved	
OUT	CamNode	REAL	D/R/W		cam node array: Array arrangement: (mPos[0],sPos[0], Speed[0],Acc[0],Type[0] ,…)	
OUT	NodeNum	WORD	D/R/W	0~361	Number of cam nodes: less than 1Indicates that the original number of cam nodes is used	
OUT	Done	BOOL	M/S	TRUE, FALSE	completion sign	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error flag	
OUT	Error ID	WORD	D/R/W	0x0~0xffff	Error ID	

Functional Overview

This command reads the current cam data on the rising edge of Execute and updates it to the cam table specified by CamTable.

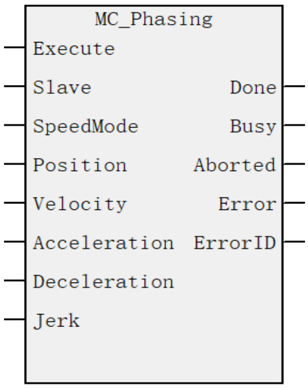
Node array data format

Node	Address	Type of data	Notes
	D1	REAL	Spindle position
	D1+2	REAL	Slave position
	D1+4	REAL	Speed

The 1st nodes	D1+6	REAL	Acceleration ratio
	D1+8	DINT	Curve type (0: straight line; 1: quintic curve)
The 2nd nodes	D1+10	REAL	Spindle position
	D1+10+2	REAL	Slave position
	D1+10+4	REAL	Speed
	D1+10+6	REAL	Acceleration ratio
	D1+10+8	DINT	Curve type (0: straight line; 1: quintic curve)
The Nth node	D1+(n-1)*10	REAL	Spindle position
	D1+(n-1)*10+2	REAL	Slave position
	D1+(n-1)*10+4	REAL	Speed
	D1+(n-1)*10+6	REAL	Acceleration ratio
	D1+(n-1)*10+8	DINT	Curve type (0: straight line; 1: quintic curve)

6.3.8 MC_Phasing (Spindle phase shift command)

Drawing Block:



Function and command description

Input Output	Name	Type of data	Applicable (soft components)	Scope	Describe	Enable
IN	Slave	WORD	constant/D/R/W	0~71	Slave axis name/axis number	Execute
IN	SpeedMode	BOOL	M/S	TRUE/FALSE (default FALSE)	Speed Type: 0T type, 1S type	
IN	Position	REAL	constant/D/R/W	Positive number	Distance or phase	
IN	Velocity	REAL	constant/D/R/W	Positive number	Target speed	
IN	Acceleration	REAL	constant/D/R/W	Positive number	Acceleration	
IN	Deceleration	REAL	constant/D/R/W	Positive number(defaultAcceleration)	Deceleration	

IN	Jerk	REAL	constant/D/R/W	Positive number	Speed step	
OUT	Done	BOOL	M/S	TRUE/FALSE	Complete stop	
OUT	Busy	BOOL	M/S	TRUE/FALSE	Running sign	
OUT	Aborted	BOOL	M/S	TRUE/FALSE	Execution interrupt	
OUT	Error	BOOL	M/S	TRUE/FALSE	Error flag	
OUT	EerrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Function description

If this command is started during single-axis synchronous control, according to the setPosition(phaseCompensation amount), Velocity (target speed), Acceleration (acceleration), Deceleration (deceleration), to compensate the spindle phase.

● When cooperating with cam motion, this command can be called after calling MC_CamIn command. When InSync=OFF of MC_CamIn command, the phase compensation command is in the buffer state, only the Busy signal is valid, and the Active signal output is invalid. When InSync=ON of the MC_Camin command, the cam is fully engaged, the Active signal output of the phase compensation command is valid, and the phase compensation action starts to execute.

● This command can be called only after the MC_GearIn command is called when moving with the gear. First trigger the MC_GearIn command, the master and slave axes establish a gear relationship, and after the slave axis is in the Synchronized Motion state, trigger the MC_Phasing command, and the phase compensation action begins to execute.

During the execution, the setting position (feedback position) of the spindle remains unchanged, and the value that the MC_Phasing command only compensates for the setting position (feedback position) by a relative amount is the "phase of the spindle". The slave axis is synchronized with the compensated "phase of the master".

Achieve Position (phaseCompensation amount), Done turns ON.

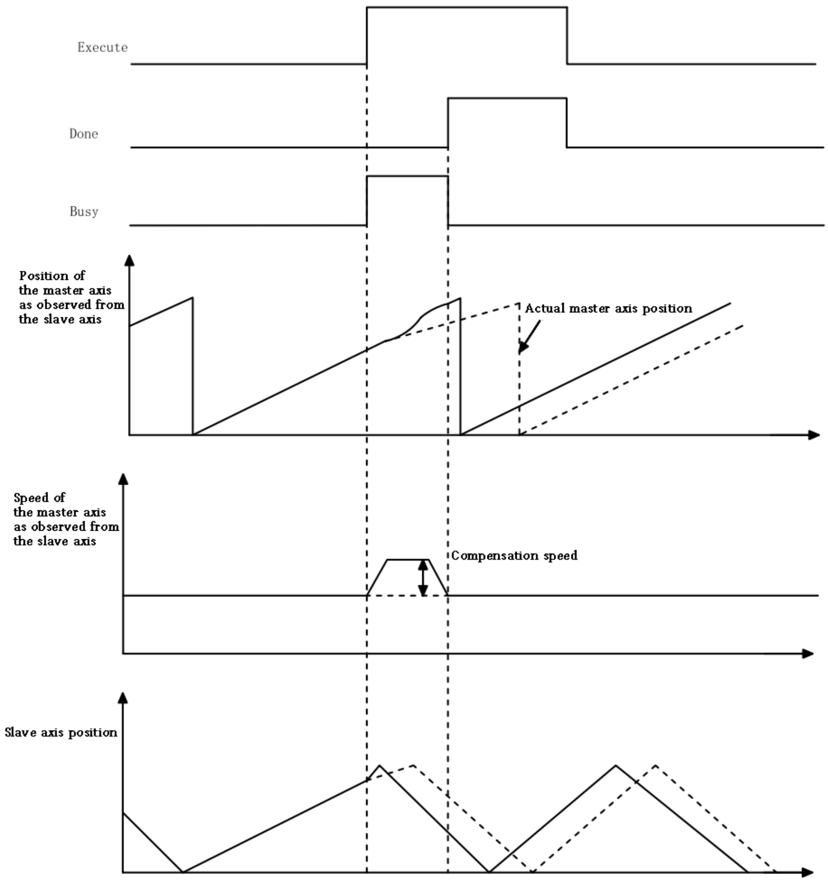
Compensation ends when the synchronous control command being executed is completed. When the synchronous control command is executed again, the previous compensation amount has no effect.

The synchronous control commands valid for spindle phase compensation are divided into two: MC_CamIn (cam action start) command and MC_GearIn (gear action start) command.

Mode selection

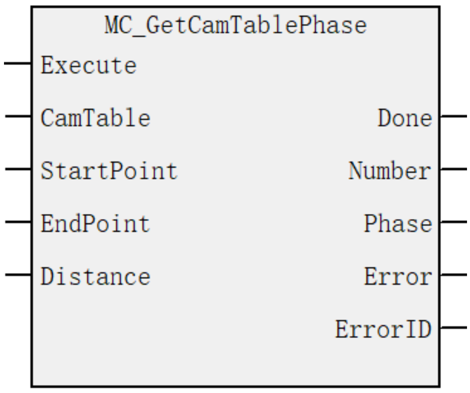
When Mode is set to 1, if the spindle stops running (the speed of the spindle is 0), the phase compensation will automatically stop. When the spindle restarts, the phase compensation will continue to run at the original paused position.

Timing diagram



6.3.9 MC_GetCamTablePhase (Get cam table phase command)

Drawing Block:



Function and command description

Enter/out put	Name	Type of data	Be applicable(dev ice)	Scope	Describe	Enable
IN	CamTable	WORD	constant/D/R/ W	0~15	Cam table	Execute
IN	StartPoint	REAL	constant/D/R/	- (default 0)	Starting point	

			W			
IN	EndPoint	REAL	constant/D/R/W	- (default 0)	End	
IN	Distance	REAL	constant/D/R/W	positive/negative/0	Displacement from axis	
OUT	Done	BOOL	M/S	TRUE/FALSE	Complete stop	
OUT	Number	WORD	D/R/W	WORD	The corresponding number of phases -1: an infinite number of identical solutions 0: None Greater than 0: the actual amount	
OUT	phase	REAL	D/R/W	positive number/0	Array of solved phase values	
OUT	Error	BOOL	M/S	TRUE/FALSE	Error flag	
OUT	Error ID	WORD	D/R/W	0x0~0xffff	Error ID	

Function description

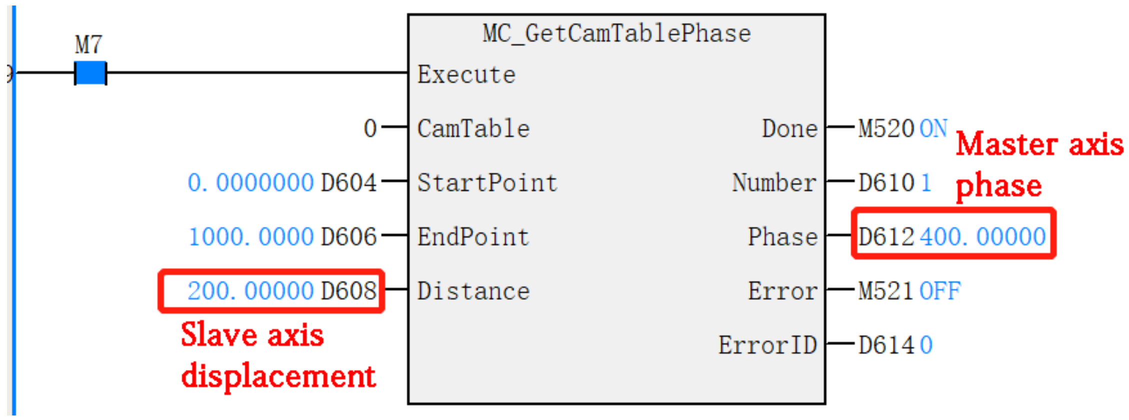
This command can obtain the corresponding master axis phase (Phase) according to the slave axis displacement (Distance) between two cam key points.

If the cam curve is a straight line and is parallel to the X axis, the given Distance in the command is on the straight line, the command output parameter Number outputs -1, and Phase [0] outputs the abscissa of the starting point.

If the cam curve is a 5th degree curve, there may be multiple solutions at this time, the command output parameter Number represents the number of solutions, and the Phase array stores the specific value obtained.

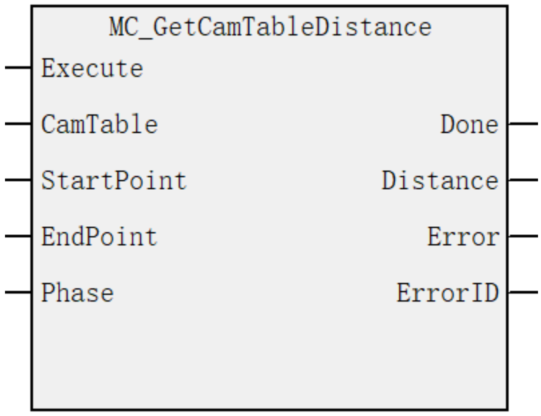
If there is no solution, the output parameter Number is equal to 0.

Program example



6.3.10 MC_GetCamTableDistance (Get cam table displacement command)

Drawing Block:



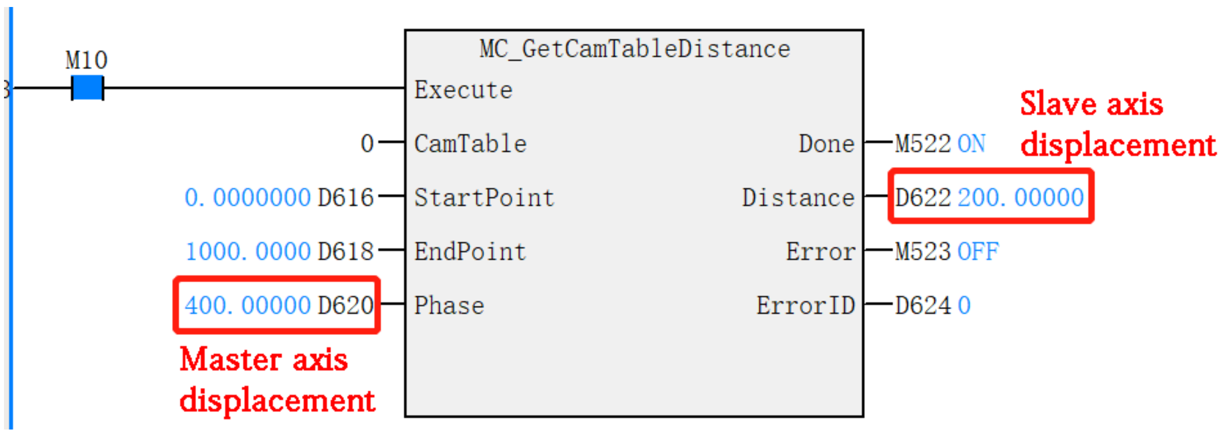
Function and command description

Enter/out put	Name	Type of data	Be applicable(device)	Scope	Describe	Enable
IN	CamTable	WORD	constant/D/R/W	0~15	Cam table	Execute
IN	StartPoint	REAL	constant/D/R/W	- (default 0)	Starting point	
IN	EndPoint	REAL	constant/D/R/W	- (default 0)	End	
IN	phase	REAL	constant/D/R/W	Positive number/negative number/0	Spindle phase shift	
OUT	Done	BOOL	M/S	TRUE/FALSE	Complete stop	
OUT	Distance	REAL	D/R/W	positive/negative/0	Solved slave displacement value	
OUT	Error	BOOL	M/S	TRUE/FALSE	Error flag	
OUT	Error ID	WORD	D/R/W	0x0~0xffff	Error ID	

Function description

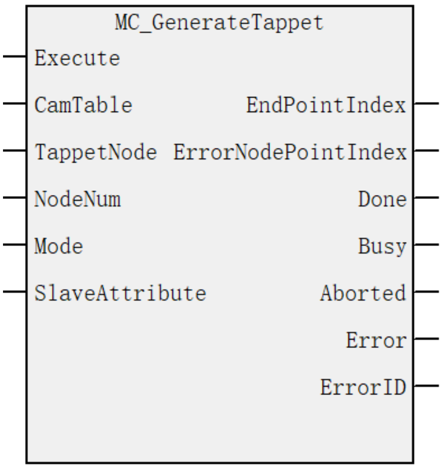
This command can obtain the corresponding slave axis displacement (Distance) according to the master axis phase (Phase) between two cam key points.

Program example



6.3.11 MC_GenearateTappet (Update ejector command)

Drawing Block:



Function and command description

Enter/ output	Name	Type of data	Be applicable(device)	Scope	Describe	Enable
IN	CamTable	WORD	constant/D/R/W	0~15	Cam table	Execute
IN	TappetNode	REAL	D/R/W		Pretty catchy array: Array arrangement: (Spindle position, element type, element address, forward pass action type, reverse pass action type,…)	
IN	NodeNum	WORD	constant/D/R/W	0~361	Pretty rush quantity	
IN	Mode	WORD	constant/D/R/W	0 (default 0)	Effective mode: 0: Take effect in the next cam cycle Other: reserved	
IN	SlaveAttribute	WORD	Constant/D/R/W	0~71	Slave axis properties	
OUT	EndPointIndex	WORD	D/R/W	WORD	End index	
OUT	EerrorNodePointIindex	WORD	D/R/W	WORD	Wrong node number	
OUT	Done	BOOL	M/S	TRUE/FALSE	Complete stop	
OUT	Busy	BOOL	M/S	TRUE/FALSE	Running sign	

OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interrupt	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error flag	
OUT	Error ID	WORD	D/R/W	0x0~0xffff	Error ID	

Function description

This command uses edge triggering.

Before the cam is enabled, it needs to be set by this command before writing.

Modifications are not allowed during operation.

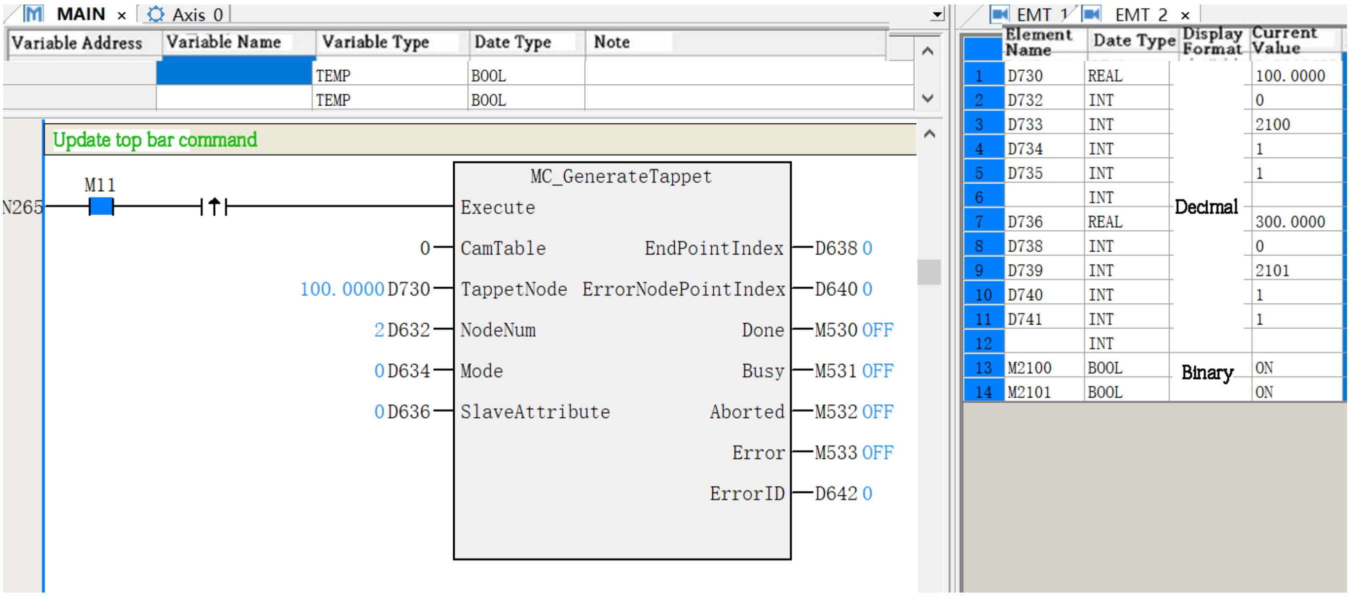
Ejector array data format

Ejector point	Address	Number of digits	Scope	Notes
The 1st point	D1	32bit		REAL,Spindle position when the ejector is moving, unit: Unit
	D1+2	16bit	0,1	Element type 0: M element 1: Y element
	D1+3	16bit	Variety	Element address: 0~7679 for element M, 0~191 for element Y
	D1+4	16bit	0,1,2,3	positiveAction type 0: OFF, 1: ON
	D1+5	16bit	0,1,2,3	reverseAction type 0: OFF, 1: ON
The 2nd points	D1+6	32bit		REAL,Spindle position when the ejector is moving, unit: Unit
	D1+6+2	16bit	0,1	Element type 0: M element 1: Y element
	D1+6+3	16bit	Variety	Element address: 0~7679 for element M, 0~191 for element Y.
	D1+6+4	16bit	0,1,2,3	Positive action type 0: OFF, 1: ON
	D1+6+5	16bit	0,1,2,3	Reverse action type 0: OFF, 1: ON
The n th point	D1+6*(n-1)	32bit		REAL,Spindle position when the ejector is moving, unit: Unit
	D1+6*(n-1) +2	16bit	0,1	Element type 0: M element 1: Y element
	D1+6*(n-1) +3	16bit	Variety	Element address: 0~7679 for element M, 0~191 for element Y
	D1+6*(n-1) +4	16bit	0,1,2,3	Positive action type 0: OFF, 1: ON
	D1+6*(n-1) +5	16bit	0,1,2,3	Reverse action type 0: OFF, 1: ON

Note: When the Y address is used, pay attention to the conversion between octal and decimal, Y10 corresponds to K8.

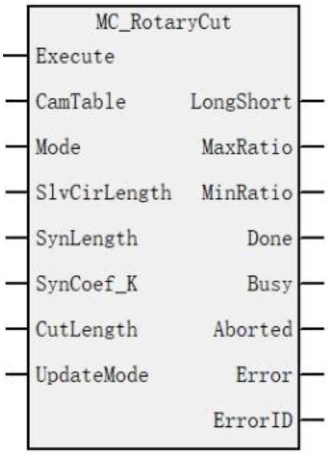
The ejector data of the same device needs to be arranged in ascending order.

Program example



6.3.12 MC_RotaryCut (Flying shear command)

Drawing Block



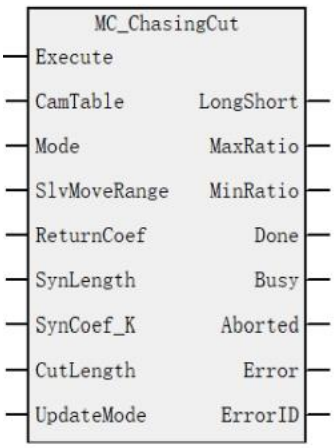
Function and command description

Input Output	Name	Type of data	Applicable (soft components)	Range	Describe	Enable
IN	CamTable	WORD	constant/D/R/W	0~15	Cam table	Execute
IN	Mode	WORD	constant/D/R/W	0(Default 0)	flying shears model: 0: The starting point is the midpoint of the asynchronous zone. 1: The midpoint of the starting synchronization zone.	

					Others: Reserved.	
IN	SlvCirLength	REAL	D/R/W		Length of one circle from the shaft	
IN	SynLength	REAL	D/R/W		Sync area length	
IN	SynCoef_K	REAL	D/R/W	0.8~1.2 (default 1)	Sync zone compensation factor	
IN	CutLength	REAL	D/R/W		Cutting length	
IN	UpdateMode	WORD	constant/D/R/W	0(Default 0)	Update mode: 0: Effective in the next cycle Other: reserved	
OUT	LongShort	BOOL	M/S	TRUE, FALSE	FALSE: short material TRUE: long material	
OUT	MaxRatio	REAL	D/R/W		Maximum speed ratio	
OUT	MinRatio	REAL	D/R/W		Minimum speed ratio	
OUT	Done	BOOL	M/S	TRUE, FALSE	Completion sign	
OUT	Busy	BOOL	M/S	TRUE FALSE	Running sign	
OUT	Aborted	BOOL	M/S	TRUE FALSE	Execution interrupt	
OUT	Error	BOOL	M/S	TRUE FALSE	Error flag	
OUT	Error ID	WORD	D/R/W	0x0~0xffff	Error ID	

6.3.13 MC_ChasingCut (Chasing shear command)

Drawing Block:

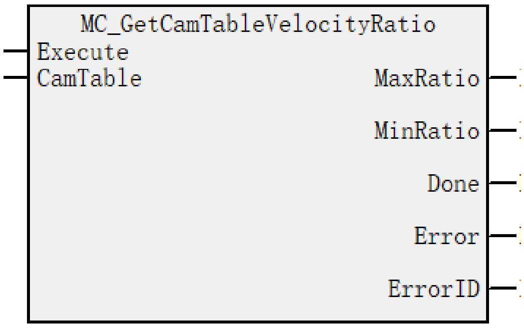


Function and command description

Input Output	Name	Type of data	Applicable (soft components)	Range	Describe	Enable
IN	CamTable	WORD	constant/D/R/W	0~15	Cam table	Execute
IN	Mode	WORD	constant/D/R/W	0(Default 0)	chasing shear model: 0:Start acceleration segment 1:Start synchronization segment Others: Reserved	
IN	SlvMoveRange	REAL	D/R/W		Slave axis movement range	
IN	ReturnCoef	WORD	constant/D/R/W	40-500(Default 100)	Return coefficient	
IN	SynLength	REAL	D/R/W		Sync area length	
IN	SynCoef_K	REAL	D/R/W	0.8~1.2 (default 1)	Sync zone compensation factor	
IN	CutLength	REAL	D/R/W		Cutting length	
IN	UpdateMode	WORD	constant/D/R/W	0(Default 0)	Update mode: 0: Effective in the next cycle Other: reserved	
OUT	LongShort	BOOL	M/S	TRUE, FALSE	FALSE: short material TRUE: long material	
OUT	MaxRatio	REAL	D/R/W		Maximum speed ratio	
OUT	MinRatio	REAL	D/R/W		Minimum speed ratio	
OUT	Done	BOOL	M/S	TRUE, FALSE	Completion sign	
OUT	Busy	BOOL	M/S	TRUE, FALSE	Running sign	
OUT	Aborted	BOOL	M/S	TRUE, FALSE	Execution interrupt	
OUT	Error	BOOL	M/S	TRUE, FALSE	Error flag	
OUT	Error ID	WORD	D/R/W	0x0~0xffff	Error ID	

6.3.14 MC_GetCamTableVelocityRatio（Getting Cam Table Speed Ratio Commands）

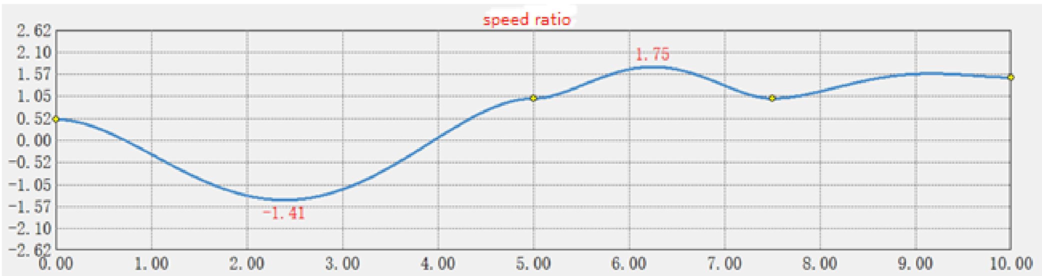
Drawing Block

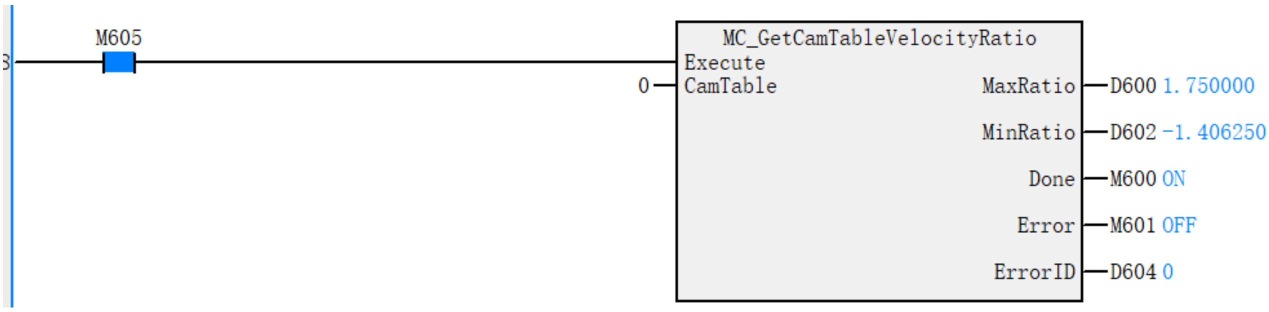


Function and command description

Input Output	Name	Type of data	Applicable（soft components)	Range	Describe	Enable
IN	CamTable	WORD	constant /D/R/W	0~15	cam table	Execute
OUT	MaxRatio	REAL	D/R/W	Positive/Negative	Max. speed ratio	
OUT	MinRatio	REAL	D/R/W	Positive/Negative	Min. speed ratio	
OUT	Done	BOOL	M/S	TRUE,FALSE	Completion sign	
OUT	Error	BOOL	M/S	TRUE,FALSE	Error flag	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

This command is active and reads the maximum and minimum speed ratio of the current cam.

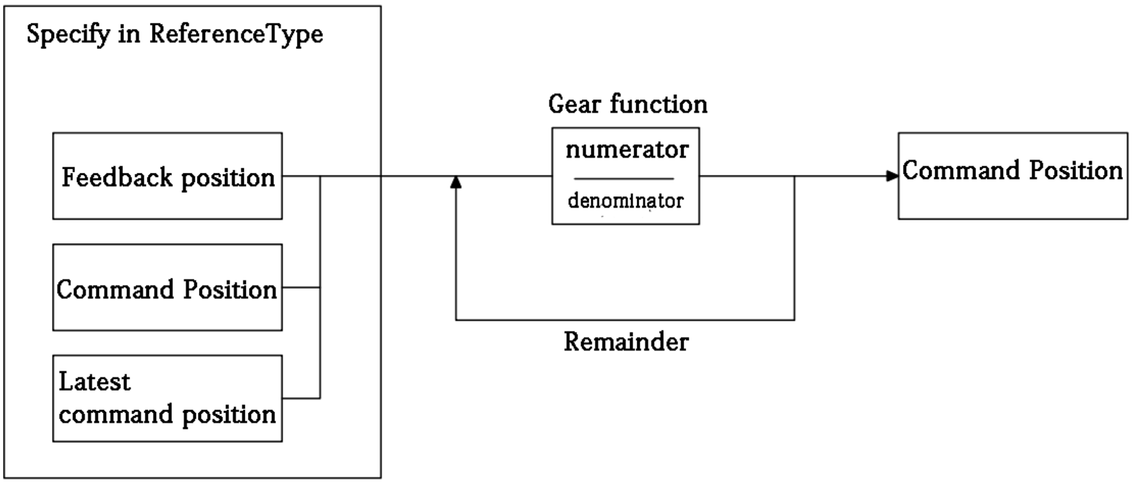




6.4 Electronic Cam Operation

6.4.1 Gear action

Basic Block Diagram:



Function description

The types of master and slave axes available for gear action are as follows:

● Spindle: bus servo axis, local pulse axis and local encoder axis, bus encoder axis.

● Slave axis: bus servo axis and local pulse axis.

The gear operation is started by the MC_GearIn (gear operation start) command, and the synchronization is released by the MC_GearOut (gear operation release) command or the MC_Stop (forced stop) command.

After the operation starts, the slave axis performs acceleration/deceleration operation at the speed obtained by multiplying the main axis speed by the gear ratio as the target speed.

Before reaching the target speed, it is called the Catching phase (in chasing), and after reaching the target speed, it is called the InGear phase (in gear synchronization).

Gear action is performed by setting the gear ratio between the master and slave shafts.

When the gear ratio is positive, the Slave (slave) moves in the same direction as the Master (spindle); when it is negative, the Slave (slave) moves in the opposite direction of the Master (spindle).

Program example

Program description:

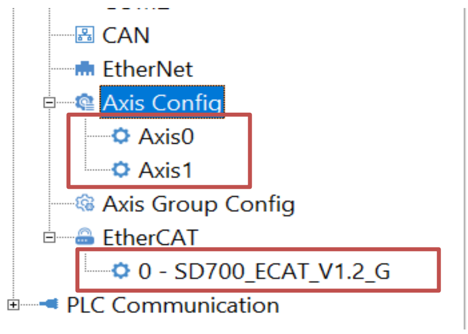
Create a new bus servo axis and a local pulse axis, with the local pulse axis as the main axis and the bus servo axis as the slave axis. The slave shaft follows the main shaft for gear action according to 1:1 gear ratio.

Program description:

Create a new bus servo axis and a local pulse axis, with the local pulse axis as the main axis and the bus servo axis as the slave axis. The slave shaft follows the main shaft for gear action according to 1:1 gear ratio.

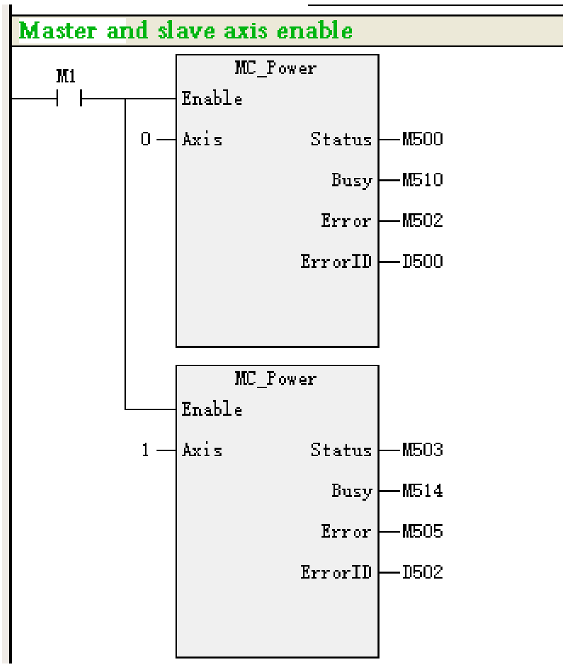
Steps:

1. New project, A new bus servo axes, A local pulse axis, the local pulse axis as the master axis, the bus servo axis as the slave axis.

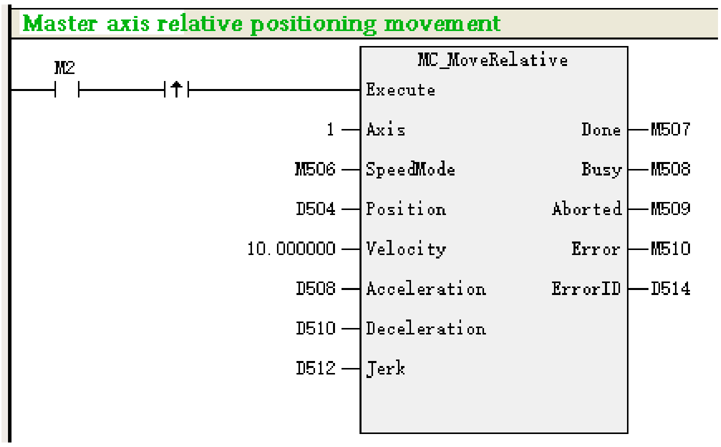


Bus servo axis for Axis_0, the local pulse axis is Axis_1.

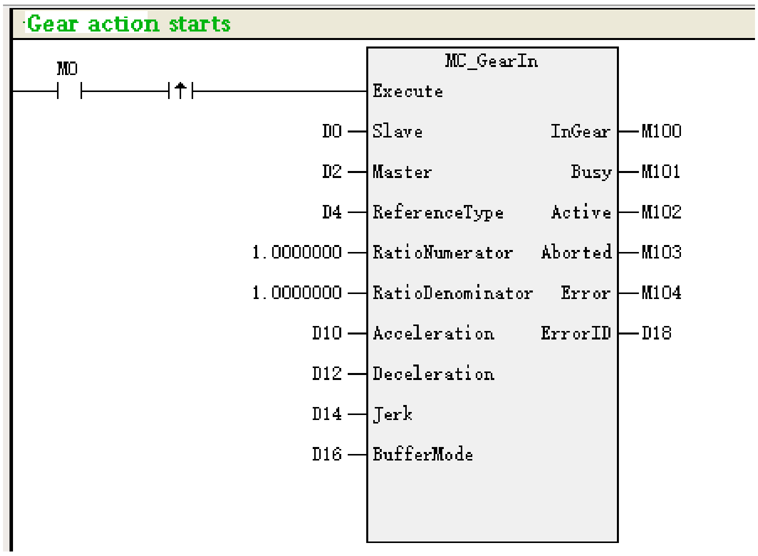
2. Call the MC_Power command to control the enabling of the master and slave axes.



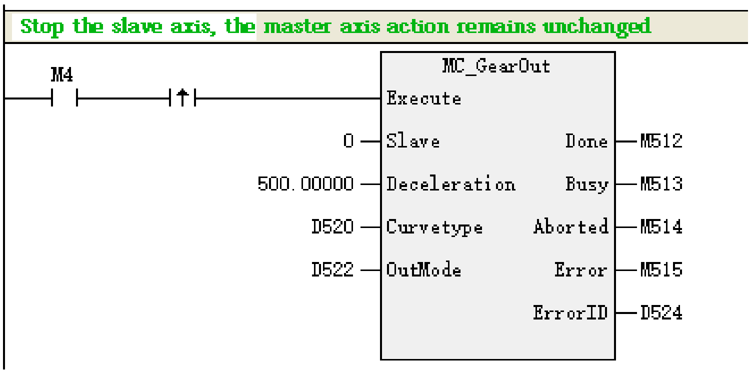
3. Call MC_ The MoveRelative command controls the spindle motion.



4. Call the MC_GearIn command to perform the gear operation, and set the gear ratio to 1:1.



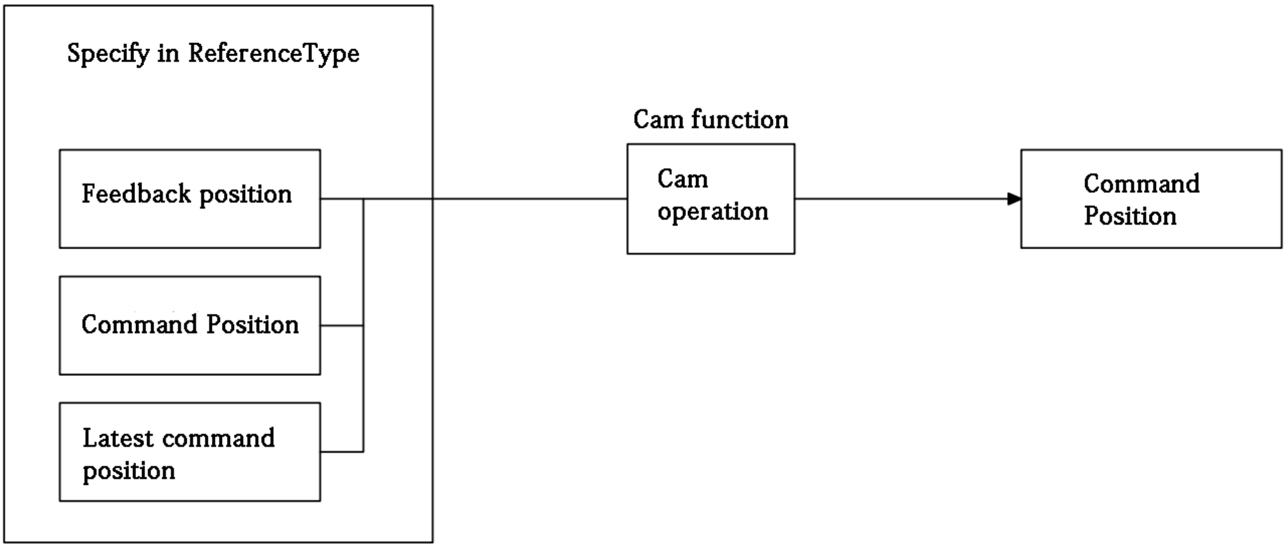
5. Call the MC_GearOut command to cancel the gear operation.



6.4.2 Cam action

Cam action means that the slave axis moves synchronously with the master position according to the cam table.

Basic Block Diagram:



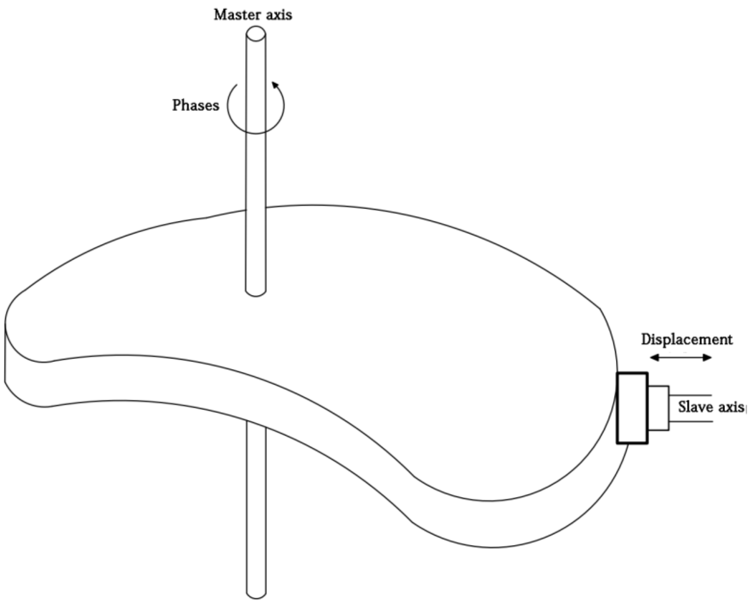
Function description

The types of master and slave axes available for cam action are as follows:

- Spindle: bus servo axis, local pulse axis, local encoder axis and remote encoder axis.
- Slave axis: motion control axis.

Start the cam operation or replace the cam table with the MC_CamIn (cam operation start) command, and cancel the cam operation with the MC_CamOut (cam operation cancel) command or MC_Stop (forced stop) command.

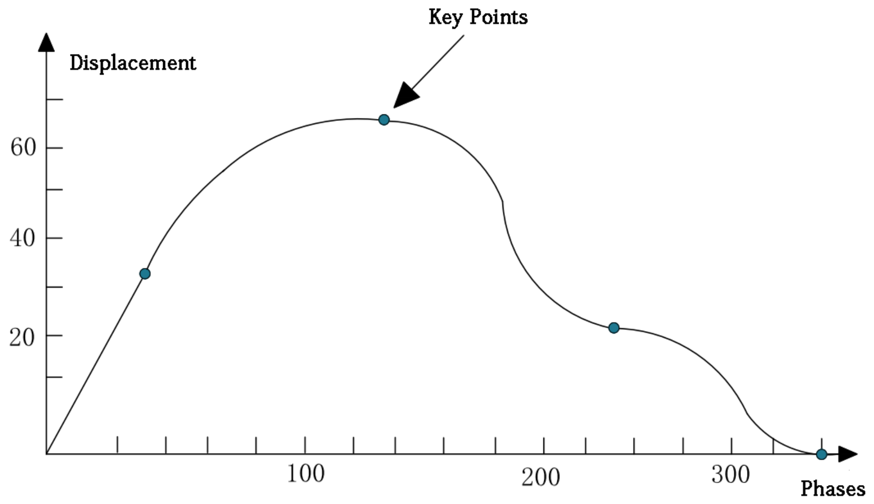
A typical cam structure is shown in the figure below. The main shaft rotates periodically, and the slave shaft reciprocates in one direction under the control of the main shaft.



Electronic cam similar in this structure, select an axis (bus servo axis, local pulse axis of the local encoder axis) is used as the master axis, and an axis (bus servo axis or local pulse axis) is selected as the slave axis, and the two move synchronously under the set cam curve.

Cam curve

The cam curve is a two-dimensional coordinate system, in which the abscissa represents the phase of the master axis, and the ordinate represents the displacement of the slave axis. Set some key points in the coordinate system and connect each two key points with a set curve (such as a straight line or a 5th curve) to form a cam curve.



Cam table

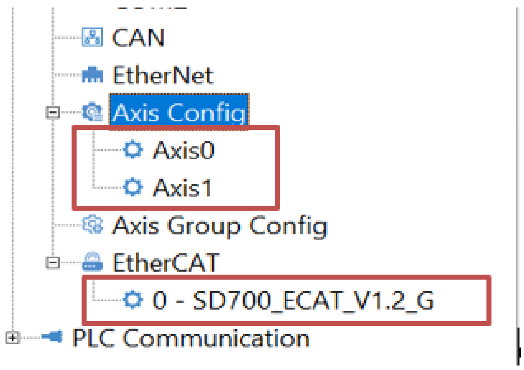
	Phases	Displacement
	0	0
Start	80	30
	160	50
	240	20
End	360	0

Program example

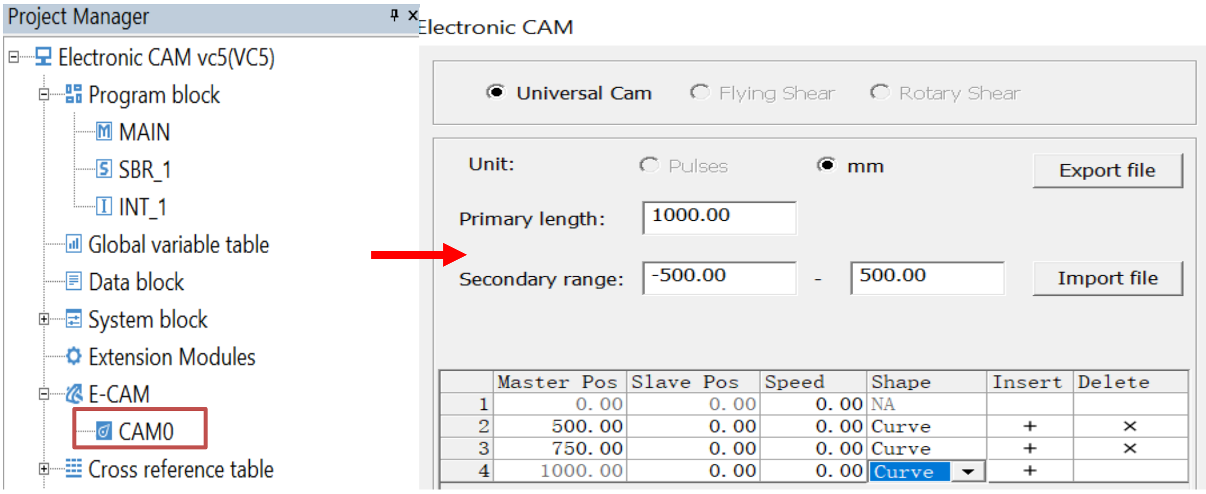
Description: Local pulse axisAxis_1 as the cam shaft, Bus servo axisAxis_0 acts as a cam slave axis to follow Axis_0 to perform cam action.

Steps:

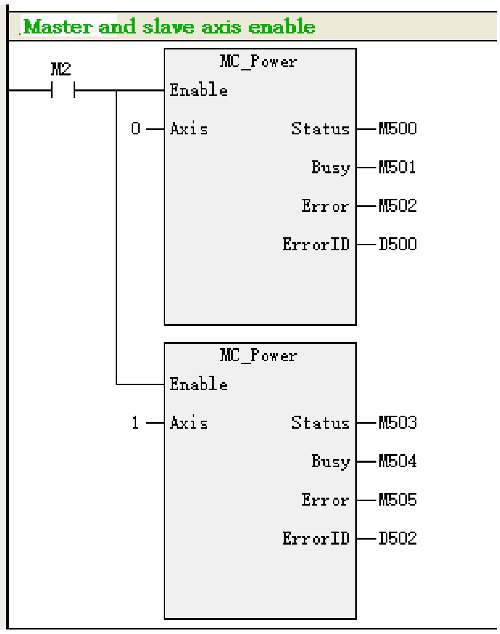
1. New Project, Establish a bus servo axis, A local pulse axis, local pulse axisas the main axis, bus servo axis for the slave axis.



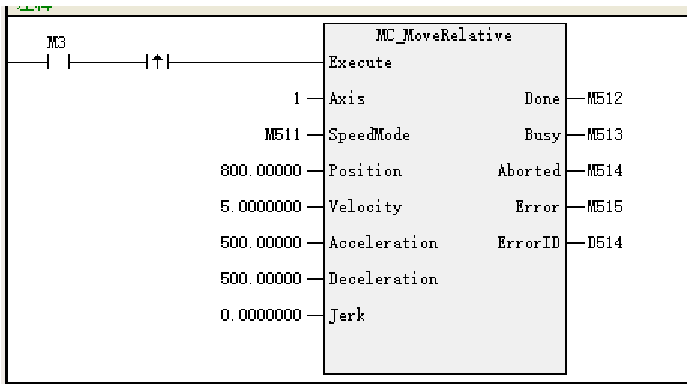
2. Create a new cam table.



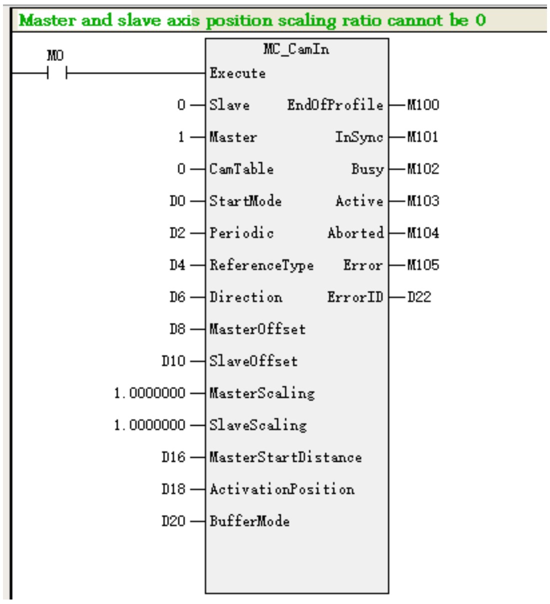
3. Call the MC_Power command to control the enabling of the master and slave axes.



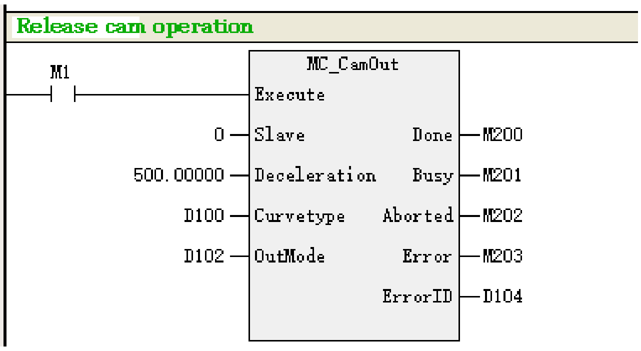
4. Call MC_MoveRelative command controls the spindle motion.



5. Call the MC_CamIn command to execute the cam action.



6. Call MC_CamOut to cancel the cam action.

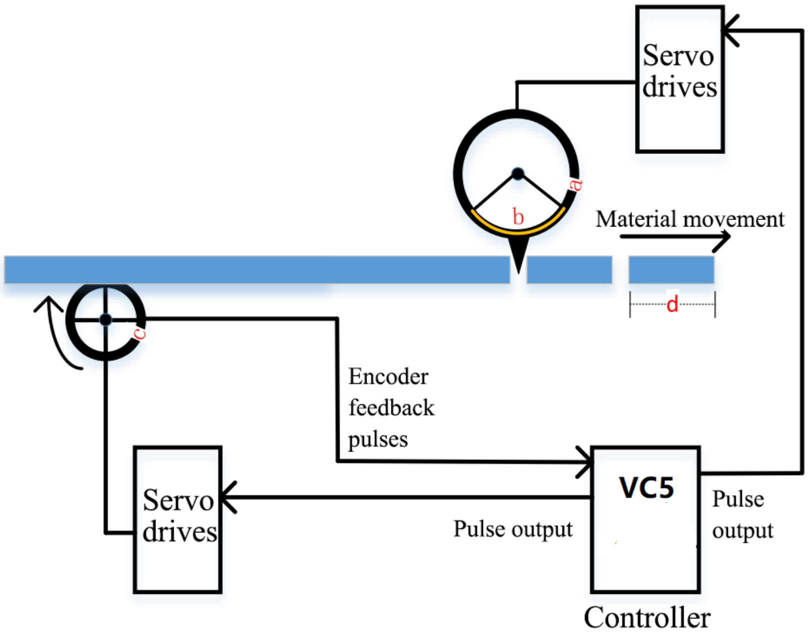


6.4.3 Flying shear function

6.4.3.1 Introduction

In the application of feeding and cutting, the traditional practice is to use the walk-stop type, where the feeding axis first walks to a fixed length, and then the shearing axis moves again, after which the process of "feeding stop" and "shearing stop" is repeated continuously. The disadvantage of this method is that the acceleration and deceleration required in the process of feeding axis walking stop makes the production efficiency cannot be improved, so the new practice is to use the way of feeding non-stop to achieve, there are generally two kinds of feeding and cutting methods are flying shear and chasing shear, the difference between the two is that chasing shear is a round trip movement, while flying shear is the same direction movement, the CAM table curve set is also different. The following is a detailed explanation of the flying shear function.

6.4.3.2 Principle



In the figure:

a: Length of one turn of slave axis (mm)

b: Length of slave axis synchronization zone (mm)

c: length of one turn of spindle (mm)

d: length of material (mm)

Slave axis one-turn length: the length of one turn of the slave axis, i.e. the slave axis circumference, which is equal to the slave axis diameter*π

Slave axis synchronous zone length: the area when the spindle speed and the slave axis speed are the same is the synchronous zone length.

Spindle one turn length: the length of one turn of spindle, i.e. spindle circumference, the length is equal to spindle diameter*π

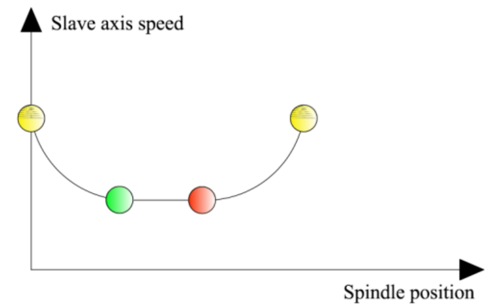
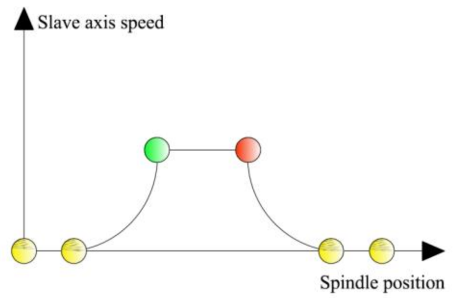
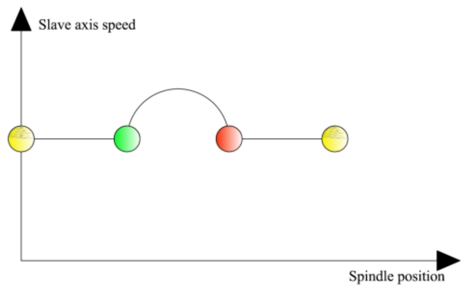
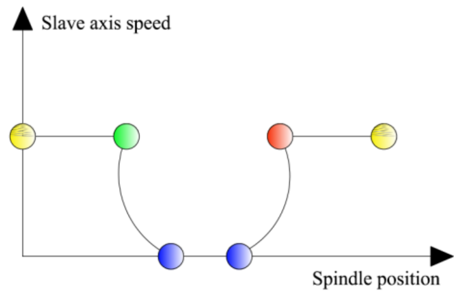
Material length: the length of the material to be cut.

6.4.3.3 Flying shear mode

NO	Take value	Description
1	0	Starting point non-synchronous zone midpoint
2	1	Starting point synchronization zone midpoint

A comparison of the master-slave relationship of the flying shear model is as follows：

Flying shear mode	Short material master-slave relationship curve	Long material master-slave relationship curve

0		
1		

6.4.3.4 Starting point non-synchronous zone midpoint

The starting point is the midpoint of the non-synchronous zone for the flying shear function mode 0, which is implemented in such a way that the starting point of the flying shear curve is the midpoint of the adjustment zone, and the speed of the slave axis synchronous zone can be compensated by the compensation factor. In the actual flying shear system, the slave axis follows the spindle in an electronic cam motion. The relationship between the material length and the cutter circumference is different and the key points of the electronic cam are different.

1、 Cutting short material

When the speed ratio of the adjustment area is greater than or equal to 0, the system will determine that the material is short material LongShort is OFF, according to the curve planning method of short material to plan, the planning point of short material is 4 points.
As shown in Figure 1.1 below, the starting point of cam engagement is the midpoint of the adjustment zone, and the slave axis runs from the yellow point to the green point by deceleration, then runs from the green point to the red point by synchronization again, and then accelerates from the red point to the yellow point to complete one cycle of movement.

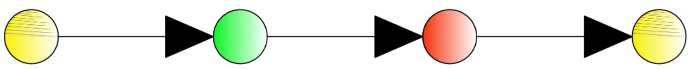
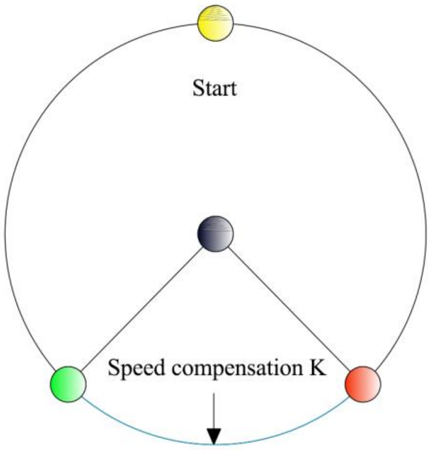


Figure 1.2 gives the curve relationship between master and slave when cutting short material. The green to red section is the synchronization zone; both the red to yellow section and the yellow to green section are the adjustment zone.



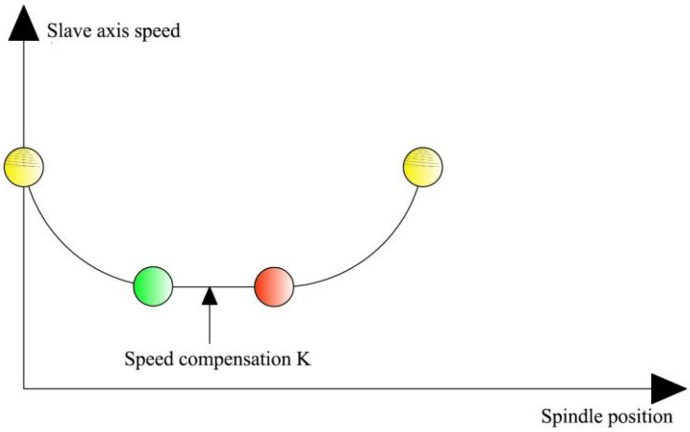


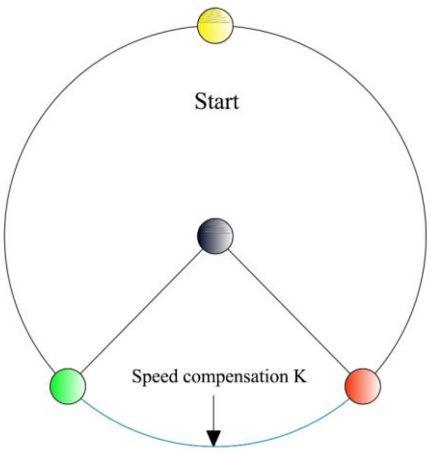
Figure 1.1 Short material point planning   Figure 1.2 Short material master-slave relationship curve

2、Cutting long material

When the speed ratio of adjustment zone is less than 0, the system will judge the material as long material and set LongShort to ON, according to the curve planning method of long material, the planning points of long material are 6 points. When cutting long material, the slave axis has a waiting area, at this time the spindle moves and the slave axis is in a stationary state.

As shown in Figure 1.3 below, the cam engagement point is the midpoint of the adjustment zone, and the slave axis is at standstill at the yellow point, then accelerates from the yellow point to the green point, then synchronizes from the green point to the red point, and finally decelerates from the red point to the yellow point.

Figure 1.4 gives the curve relationship between master and slave when cutting long material. Yellow point to green point and red point to yellow point are the adjustment zone, and between several yellow points is the stop zone. The green point to to the red point is the synchronization zone.



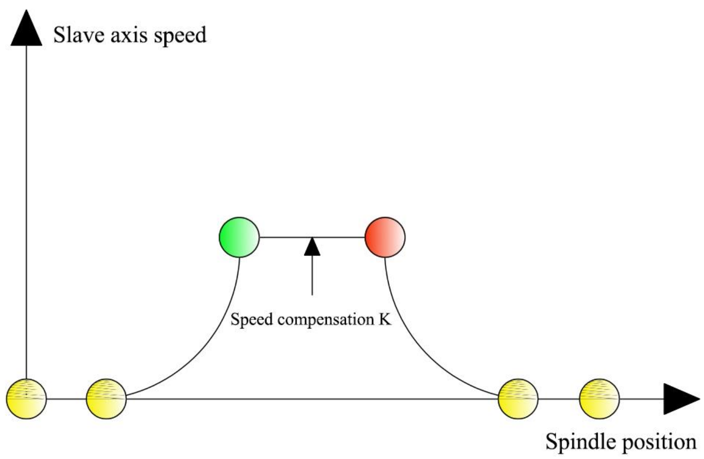


Figure 1.3 Long material point planning   Figure 1.4 Long material master-slave relationship curve

3、Parameter requirements

NO	Parameter Condition
1	Synchronization zone length (D) × compensation factor of synchronization zone (K) < cutting material length (L)
2	Length of synchronous zone (D) × compensation factor of synchronous zone (K) < length of one revolution from the shaft (L1)
3	Sync zone length (D) > 0
4	Synchronous zone compensation factor (K) >= 0.8
5	Synchronous zone compensation factor (K) <= 1.2

6.4.3.5 Starting point synchronization zone midpoint

When the starting point is the midpoint of the synchronous zone, it is the flying shear function mode 1, which is implemented in such a way that the starting point of the flying shear curve is the midpoint of the synchronous zone and the speed of the slave axis in the synchronous zone can be compensated by a compensation factor. In the actual flying shear system, the slave axis follows the spindle in an electronic cam motion. The relationship between the material length and the cutter circumference is different and the key points of the electronic cam are also different.

1、Cutting short material

When the speed ratio of adjustment zone is greater than or equal to 0, the system will judge the material as short material and set LongShort to OFF, and plan according to the curve planning method of short material, and the planning points of short material are 4 points.
As shown in Figure 1.1 below, the starting point of cam engagement is the middle point of synchronous zone, from the axis running synchronously from yellow point to green point, then moving from green point to red point again, and then moving from red point to yellow point to complete a cycle of moving.

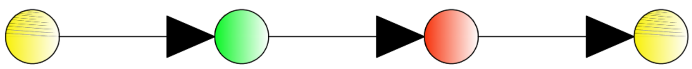


Figure 1.2 gives the curve relationship between master and slave when cutting short material. Both the yellow to green section and the red to yellow section are the synchronization zone; the green to red section is the adjustment zone.

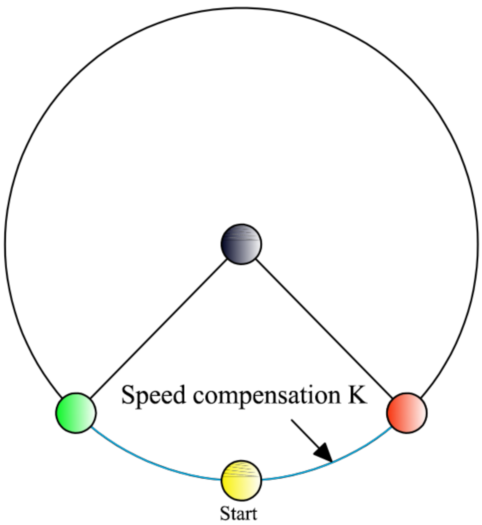


Figure 1.1 Short material point planning

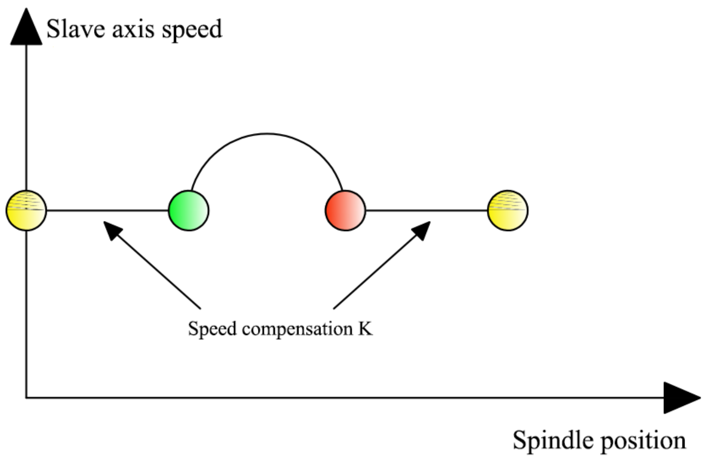


Figure 1.2 Short material master-slave relationship curves

2、Cutting long material

When the speed ratio of adjustment zone is less than 0, the system will judge the material as long material and set LongShort to ON, according to the curve planning method of long material, the planning points of long material are 6 points. When cutting the long material, the slave axis has a waiting area, at this time the spindle moves and the slave axis is in a stationary state.

As shown in Figure 1.3 below, the cam engagement point is the midpoint of the synchronization zone, and the slave axis runs synchronously from the yellow point to the green point, then from the green point to the blue point, then from the blue point to the red point, and finally from the red point to the yellow point.

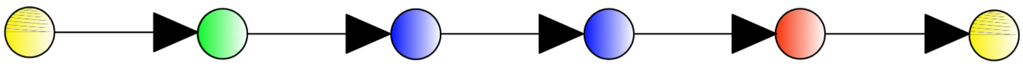
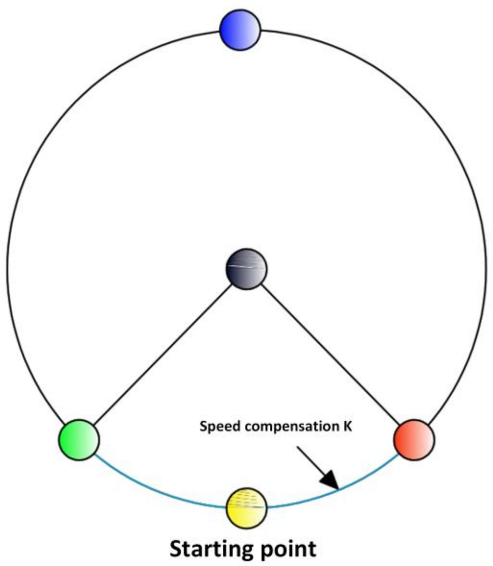


Figure 1.4 shows the relationship between the master and slave curves when cutting a long material, with the synchronization zone from the yellow point to the green point and the red point to the yellow point, and the stop zone between the two blue points. The green point to the blue point and the blue point to the red point are the adjustment zones.



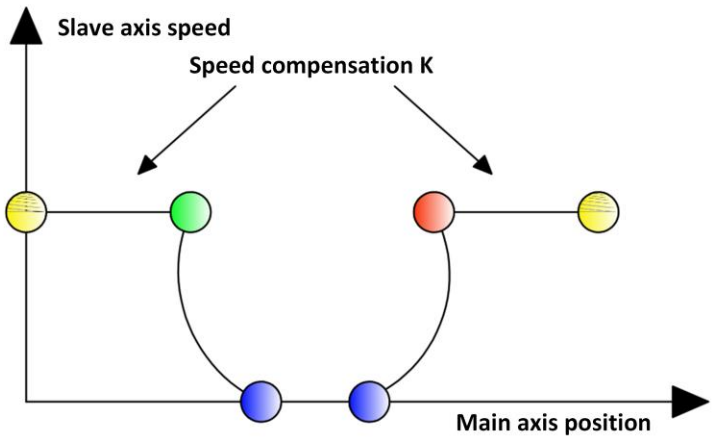


Figure 1.3 Long material point planning Figure 1.4 Long material master-slave relationship curve

3、Parameter requirements

NO	Parameter Condition
1	Synchronization zone length (D) × compensation factor of synchronization zone (K) < cutting material length (L)
2	Length of synchronous zone (D) × compensation factor of synchronous zone (K) < length of one revolution from the shaft (L1)
3	Sync zone length (D) > 0
4	Synchronous zone compensation factor (K) >= 0.8
5	Synchronous zone compensation factor (K) <= 1.2

6.4.3.6 Application Example

The flying shear mode is the use case for the midpoint of the starting point synchronization zone. Its flying shear parameters are as follows.

- Spindle diameter: 150mm
- Spindle drive ratio: 1:10
- Slave shaft diameter: 200mm
- Synchronous zone angle: 60°
- Slave drive ratio: 1:3
- Cutting material length: 500mm

Based on the above system parameters, calculate the values to be set for the MC_RotaryCut command.

(servo using SD700 servo, internal setting 10000 pulses servo motor select one turn)

Spindle one turn length=150*π=471

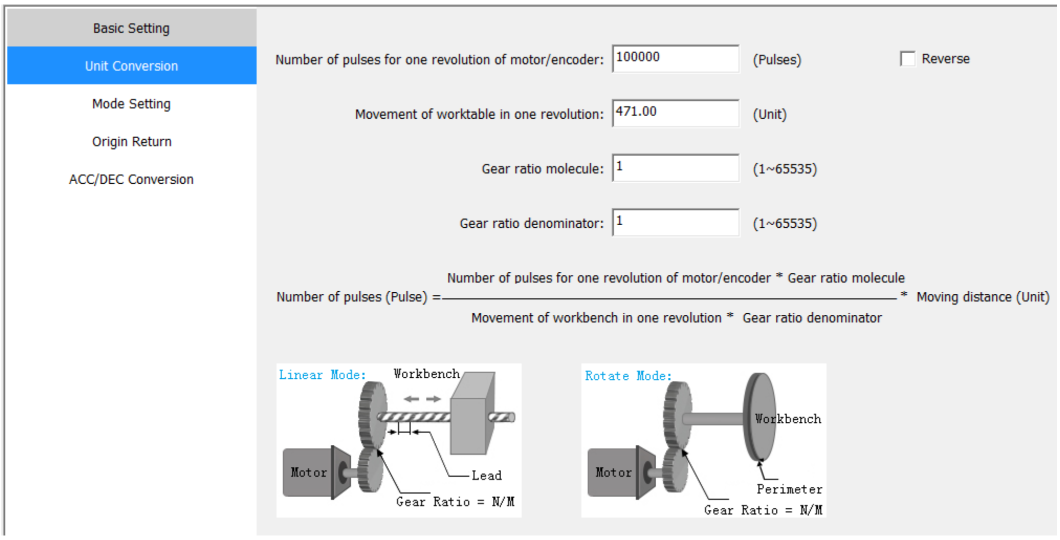
Number of pulses in one revolution of master axis=10000*10=100000

Slave axis one-turn length=200*π=628

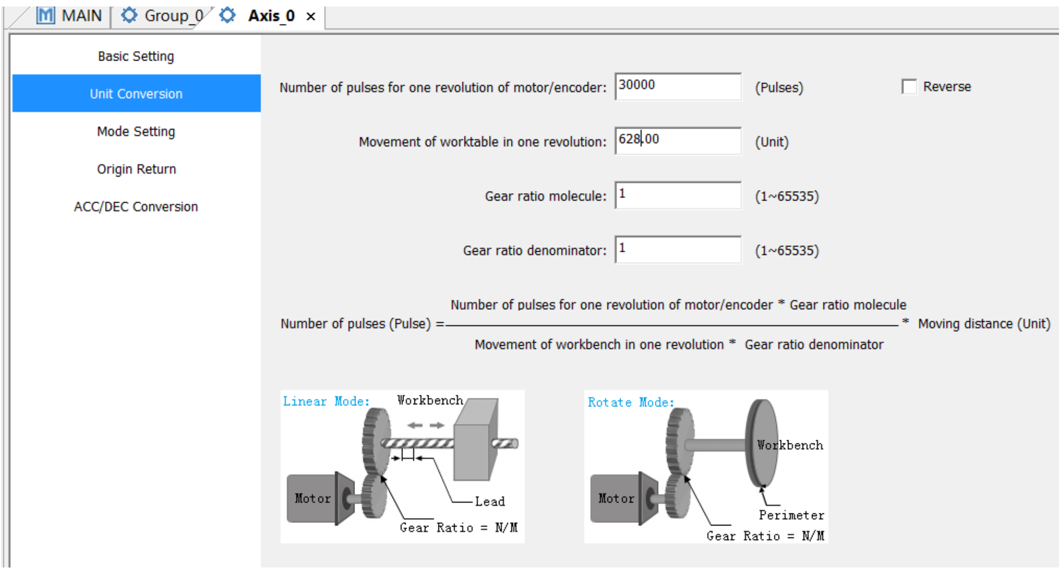
Number of pulses in one revolution of slave axis=10000*3=30000

Synchronization zone length=60°/360°*slave axis one revolution length=104.667

1、Spindle parameter setting

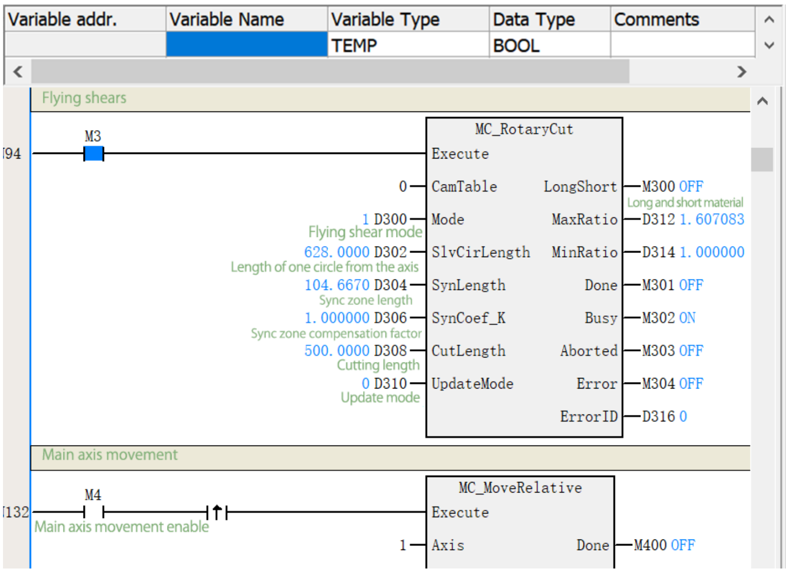


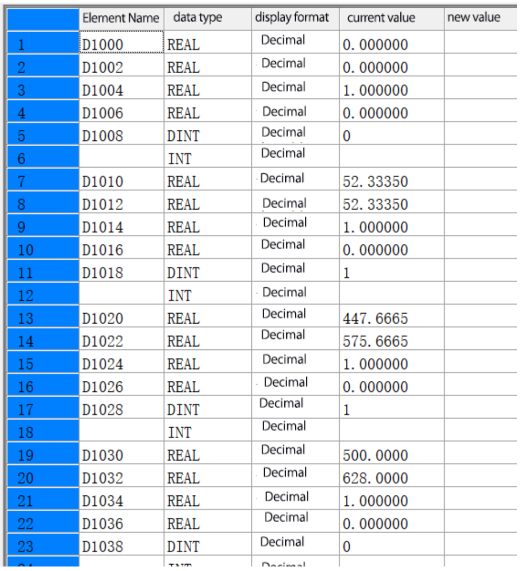
2、Slave axis parameter setting



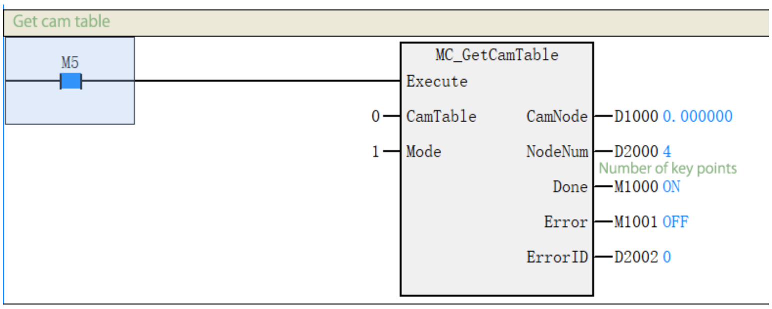
3、Flying shear curve

Obtain curve key point data with Get Cam Table (Mode 1)



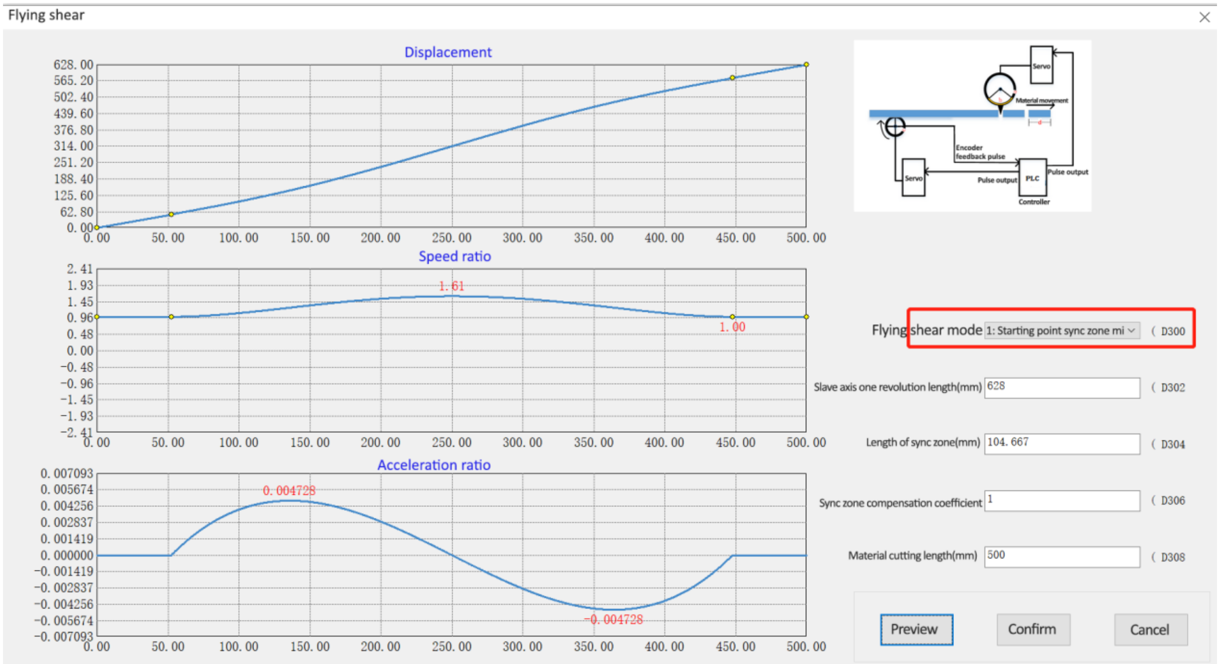


	Element Name	data type	display format	current value	new value
1	D1000	REAL	Decimal	0. 000000	
2	D1002	REAL	Decimal	0. 000000	
3	D1004	REAL	Decimal	1. 000000	
4	D1006	REAL	Decimal	0. 000000	
5	D1008	DINT	Decimal	0	
6		INT	Decimal		
7	D1010	REAL	Decimal	52. 33350	
8	D1012	REAL	Decimal	52. 33350	
9	D1014	REAL	Decimal	1. 000000	
10	D1016	REAL	Decimal	0. 000000	
11	D1018	DINT	Decimal	1	
12		INT	Decimal		
13	D1020	REAL	Decimal	447. 6665	
14	D1022	REAL	Decimal	575. 6665	
15	D1024	REAL	Decimal	1. 000000	
16	D1026	REAL	Decimal	0. 000000	
17	D1028	DINT	Decimal	1	
18		INT	Decimal		
19	D1030	REAL	Decimal	500. 0000	
20	D1032	REAL	Decimal	628. 0000	
21	D1034	REAL	Decimal	1. 000000	
22	D1036	REAL	Decimal	0. 000000	
23	D1038	DINT	Decimal	0	



4、Graphical drawing effect

Right click on the flycut command - Open to view the flycut effect curve.



5、 Flying shear program

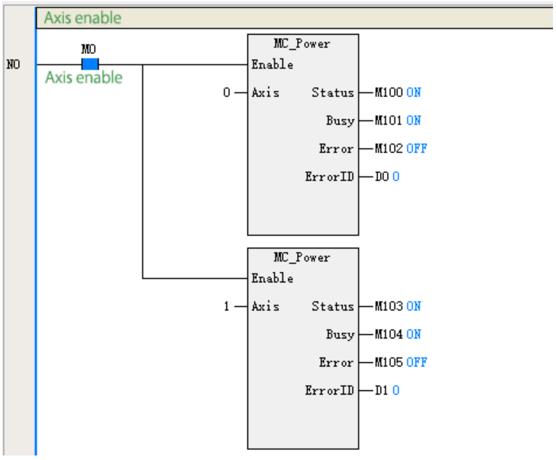
The program is as follows (add back to the original, axis stop and other instructions as required by the program):

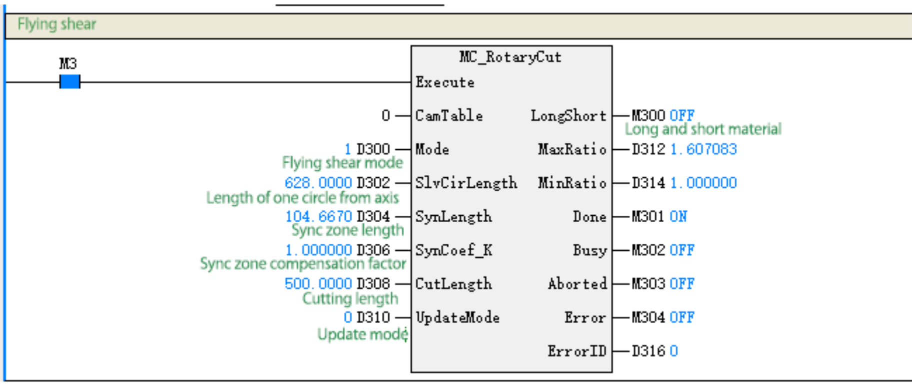
Step 1: Update the top bar

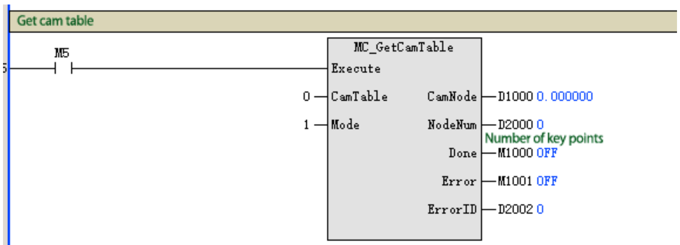
Step 2: The axis is enabled and the flying shear data is written via the flying shear command

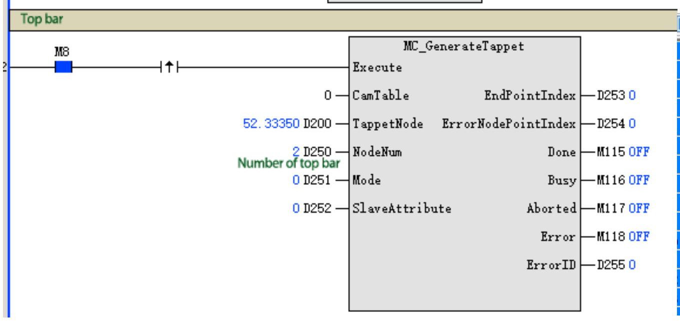
Step 3: Initialize the electronic cam

Step 3: Activate the electronic cam according to the current master-slave position relationship



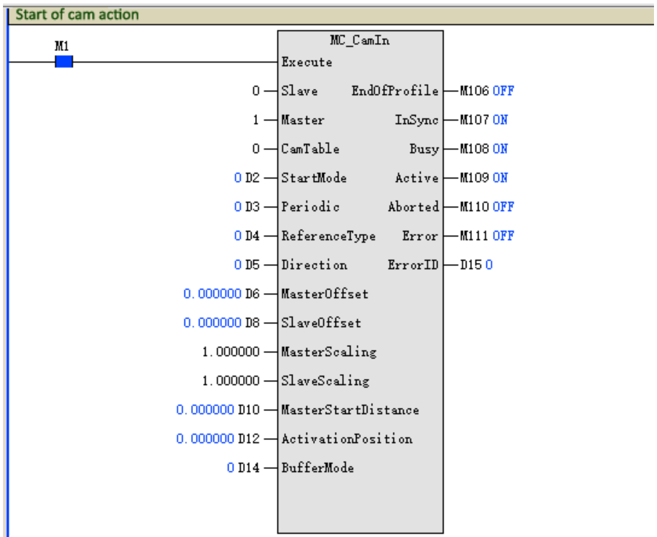


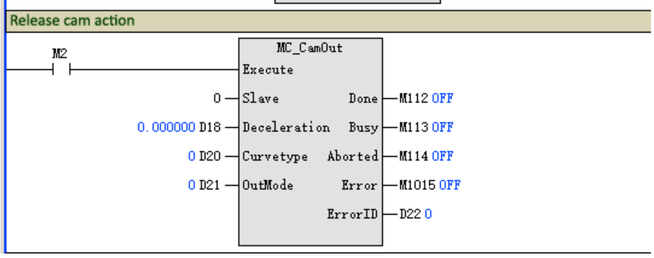


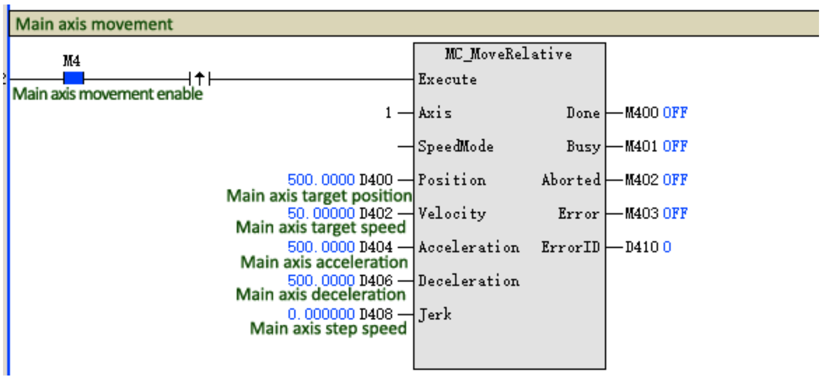


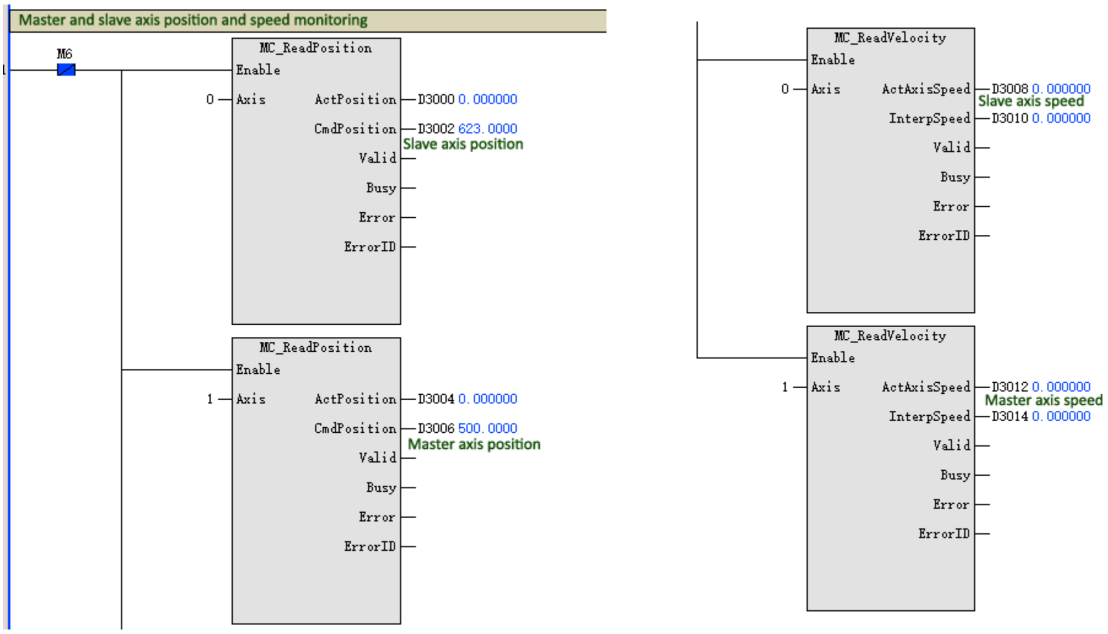
MAIN | EMT_1*

Element Name	data type	display format	current value	new value	element remark
D250	INT	Decimal	2	2	Number of top bar point
	INT	Decimal			
D200	REAL	Decimal	52.33350	52.33350	
D202	INT	Decimal	0		
D203	INT	Decimal	500	500	
D204	INT	Decimal	1	1	
D205	INT	Decimal	0	0	
	INT	Decimal			
D206	REAL	Decimal	447.6665	447.6665	
D208	INT	Decimal	0	0	
D209	INT	Decimal	501	501	
D210	INT	Decimal	1	1	
D211	INT	Decimal	0		

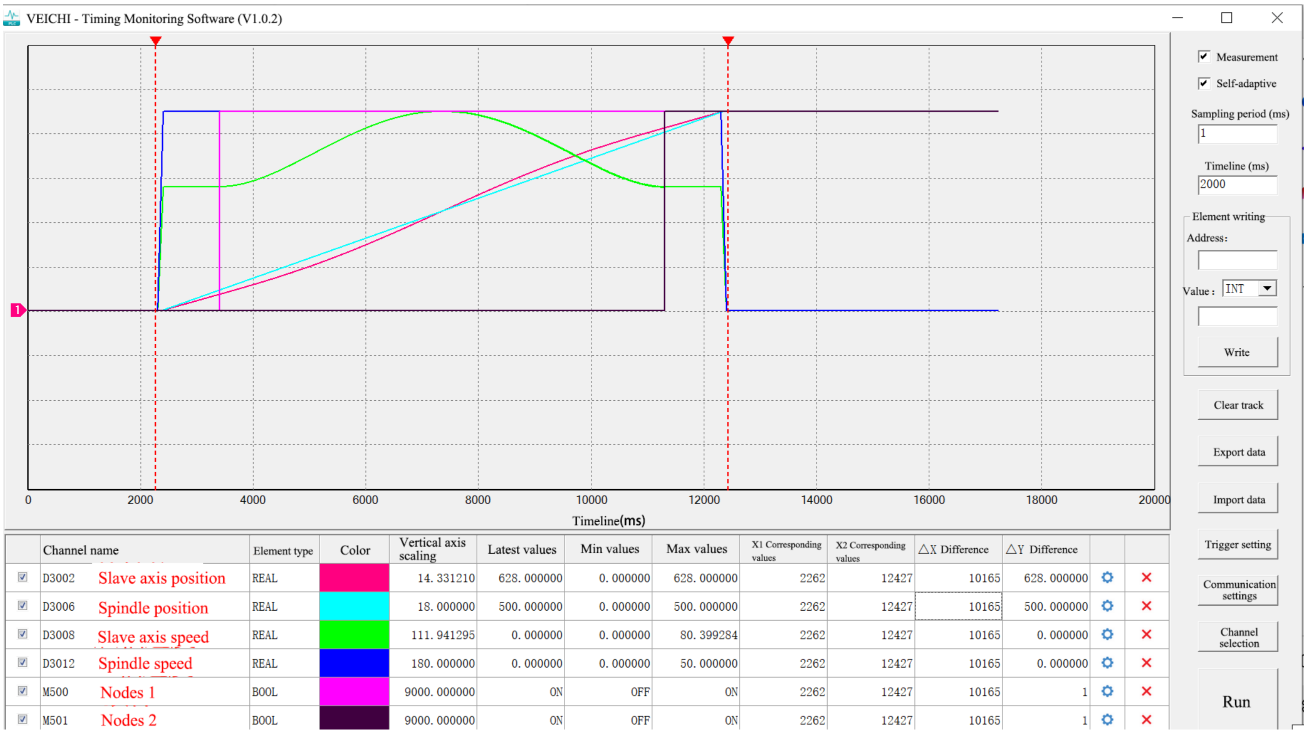








The actual results of the run are shown below:



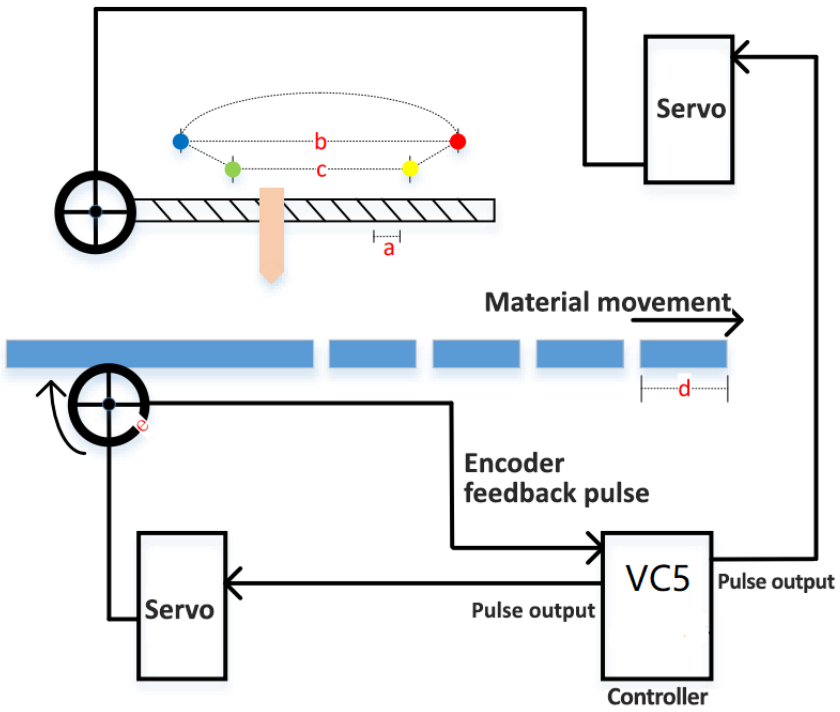
	Channel name	Element type	Color	Vertical axis scaling	Latest values	Min values	Max values	X1 Corresponding values	X2 Corresponding values	△X Difference	△Y Difference		
☑	D3002 Slave axis position	REAL		14.331210	628.000000	0.000000	628.000000	2262	12427	10165	628.000000		×
☑	D3006 Spindle position	REAL		18.000000	500.000000	0.000000	500.000000	2262	12427	10165	500.000000		×
☑	D3008 Slave axis speed	REAL		111.941295	0.000000	0.000000	80.399284	2262	12427	10165	0.000000		×
☑	D3012 Spindle speed	REAL		180.000000	0.000000	0.000000	50.000000	2262	12427	10165	0.000000		×
☑	M500 Nodes 1	BOOL		9000.000000	ON	OFF	ON	2262	12427	10165	1		×
☑	M501 Nodes 2	BOOL		9000.000000	ON	OFF	ON	2262	12427	10165	1		×

6.4.4 Chasing shear function

6.4.4.1 Introduction

Chasing shear is the process of cutting and filling the material in motion vertically. Its whole process is shown in the figure below. VC5 with two servo axis --- feeding axis and chasing shear axis, the feeding axis mainly pulls the material forward and the chasing shear axis mainly tracks the feeding axis so as to process in the synchronous area.

6.4.4.2 Principle



In the figure.:

a:One-turn lead from the shaft (mm)

b:Slave axis travel range (mm)

c:Synchronous zone(mm)

d:Material length(mm)

e:length of one turn of the spindle (mm)

Slave axis one-turn lead refers to the distance that the slave axis moves when the slave axis motor rotates for one week. If the slave axis adopts the screw structure, then this value refers to the lead of the screw.

Slave axis moving range refers to the moving range of the slave axis, the slave axis only moves within this safety range, the blue dot is the starting point of the slave axis moving range, the red dot is the end point of the slave axis moving range.

Synchronous zone refers to the area when the spindle and slave axis are at the same speed. In this area, the spindle moves at the same speed as the slave axis, and some cutting and other actions can be done in this area.

Material length refers to the length of the material to be cut.

The length of one revolution of the spindle refers to the circumference of the spindle drive wheel, and its value is equal to the diameter of the spindle drive wheel*π.

6.4.4.3 Chasing shear mode

The following table shows the supported modes for chase clipping：

NO	Take value	Description
1	0	Starting point acceleration section
2	1	Starting point synchronization segment

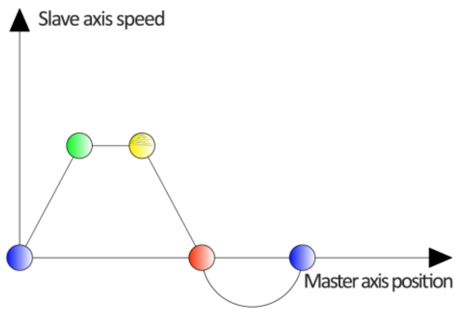
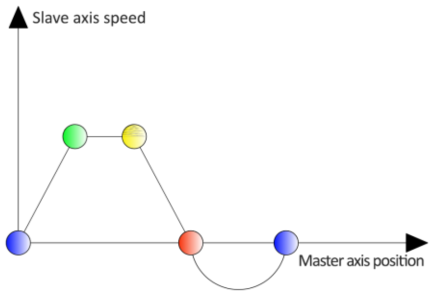
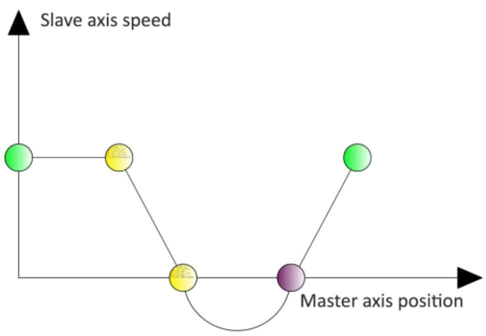
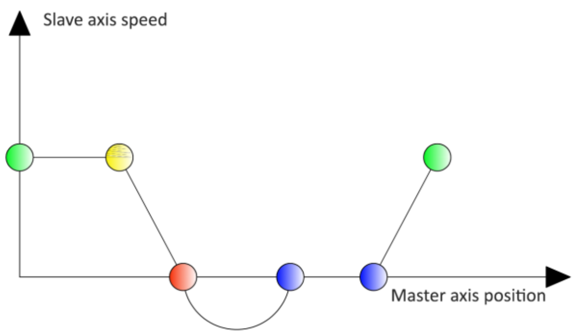
Symbols	Name
L2	From axis travel range（mm）
D	Synchronization Zone（mm）
K1	Synchronous zone compensation factor

Difference between mode 0 and mode 1

Chasing shear mode	Start point	
0	Accelerated segment	Short material range: 0 ~ L2 Long material range: 0 ~ L2
1	Synchronized segment	Short material range： $-(L2-(D\times K1))/4\sim \mathrm{D}+(L2-(D\times K1))/4$ Long material range： $-(L2-(D\times K1))/2\sim \mathrm{D}+(L2-(D\times K1))/2$

Note: In non-compensated mode, K1 can be brought into the calculation equal to 1.

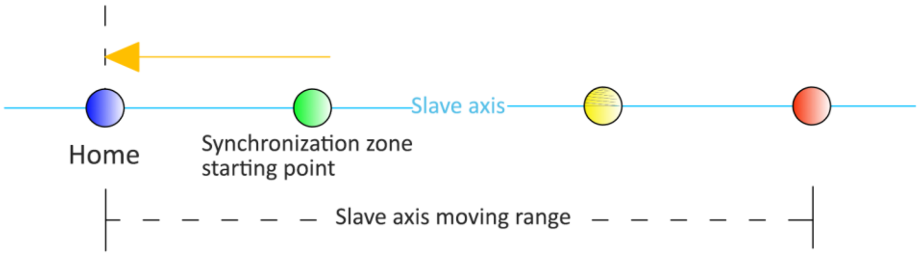
A comparison of the master-slave relationship of the chase cut model is as follows：

Chasing shear mode	Short material master-slave relationship curve	Long material master-slave relationship curve
0		
1		

6.4.4.4 Operation steps

1、 Start segment is in acceleration zone mode

For the starting section is the acceleration zone, the starting point of its entire operation is set at the beginning of the acceleration section, so there is no need to perform the from-axis front operation, i.e., the origin is the starting point.
Step 1: Return to home position. The home position command of the controller controls the return of the slave axis to the home position. Here we treat the blue point as the origin, i.e., we control the slave axis to return to the blue point position.

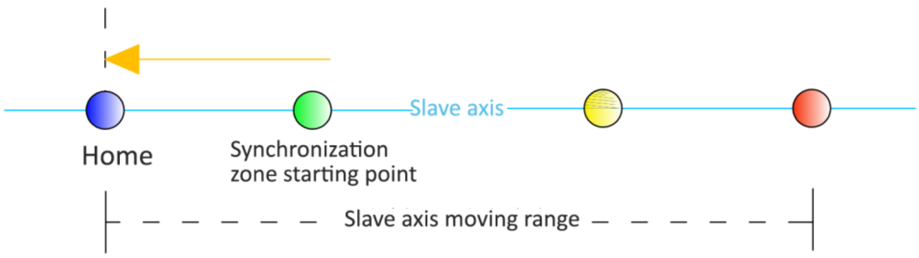


Step 2: Configure the generic cam and write the chasing parameters via the MC_ChasingCut instruction.
Step 3: Enable the cams. Make the spindle and slave axis engage.
Step 4: The controller controls the spindle movement. The master-slave relationship operates according to the set cam relationship.

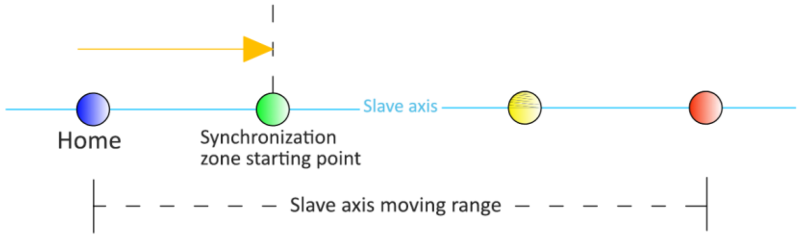
Note: No slave axis front is required in this mode, i.e. the origin is the starting point。

2、 Start segment is in synchronous zone mode

According to the actual application of chasing shear, there are long and short materials, but no matter long or short materials, the starting point of its synchronization zone is fixed, so we set the starting point of the whole operation here at the starting point of synchronization zone. The operation flow of the whole system is shown below.
Step 1: Return to home position. Control the return of the slave axis to the home position by the controller's return to home command. Here we treat the blue point as the origin, i.e., control the return of the slave axis to the blue point position.



Step 2: Slave axis front. After returning to the home position, the slave axis needs to be controlled to move to the starting position of the synchronous zone, which is the starting point of the cam engagement. Front amount = (slave axis moving range - (sync zone length * sync zone compensation factor))/2.



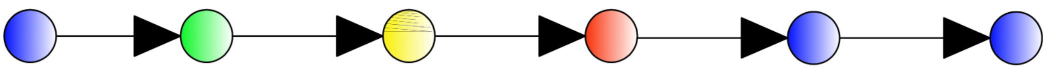
Step 3: Configure the generic cam and write the chasing parameters via the MC_ChasingCut instruction.
Step 4: Enable the cams. Make the spindle and slave axis engage.
Step 5: The controller controls the spindle movement. The master-slave relationship runs according to the set cam relationship.
Notes:
1.The first step and the second step generally only need to be executed once when commissioning the machinery, and the cam can be engaged immediately after power-up after the master and slave axis positions are subsequently saved.
2. In the actual application, the second step of the slave axis front operation can also be performed, but the slave axis will move a distance to the negative direction of the origin every time it moves.
The distance is the preload amount for long material and preload amount/2 for short material).
3.The front amount is related to the compensation coefficient of synchronous area, and the front amount is different for different compensation coefficients.

6.4.4.5 The starting segment is the acceleration zone mode description

When the starting point is in the acceleration zone, it is the chase shear function mode 0. It is implemented in such a way that the starting point of the chase shear curve is the starting point of the acceleration section and the speed of the slave axis synchronization zone can be compensated by a compensation factor.
In the actual chasing shear system, the slave axis follows the spindle in an electronic cam motion. The relationship between the length of the material and the range of movement of the slave axis is different, and the key points of the electronic cam are also different. The following describes how the slave axis follows the spindle in the starting point acceleration mode.

1、Cutting long material

When the material length > from the axis moving range × (2 + return coefficient / 50) - (2 × synchronous area compensation coefficient - 1) × synchronous length, the system will determine the material as long material and set LongShort to ON, according to the curve planning method of long material, the planning points of long material are 6 points. The specific sequence of moving from the axis is shown below：



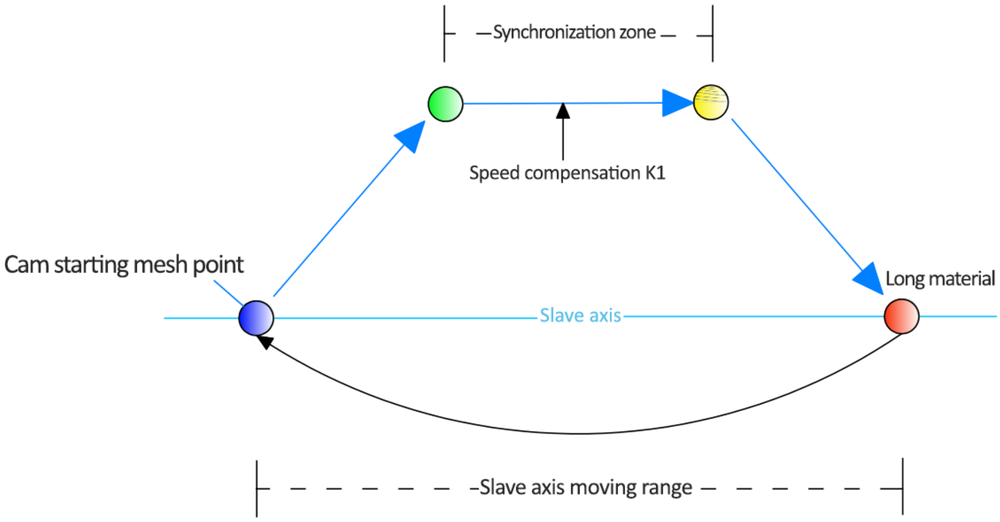
Point 1: The first point is the blue point, which is the beginning of the acceleration section, from the axis acceleration to the beginning of the synchronization zone.
Point 2: The second point is the green point, which is the starting point of the synchronization zone.
Point 3: The third point is the yellow point, i.e. the end of the synchronization zone, moving from the axis to the yellow point at a uniform speed from the green point at the speed of the synchronization zone.
Point 4: The fourth point is the red point, which is the end of the moving range from the axis, and the axis moves from the

third point to the fourth point with gradual deceleration.

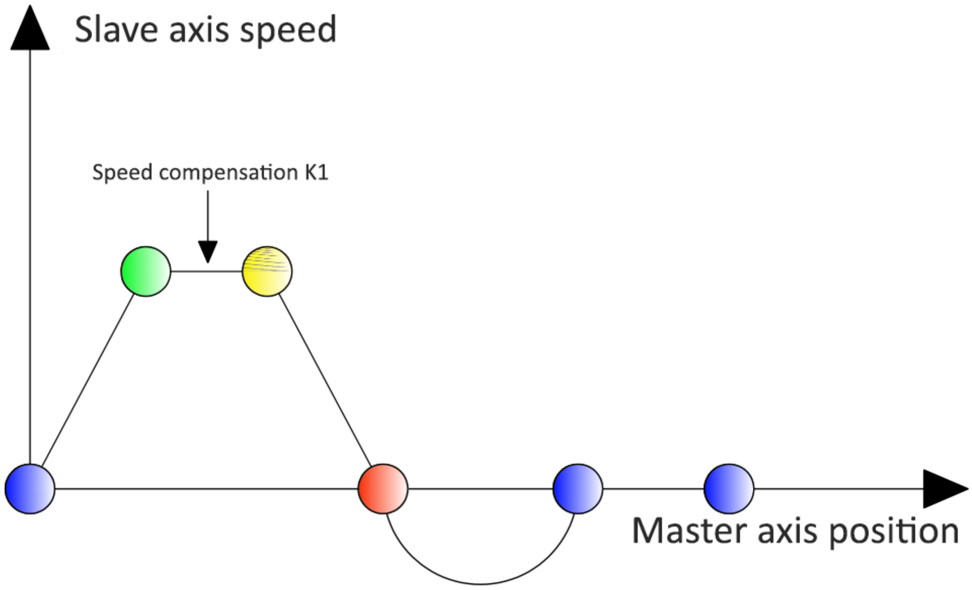
Point 5: The fifth point is the blue point, which is the starting point of the range of movement from the axis, from the axis from the fourth point back to the fifth point and stop waiting point.

Point 6: the sixth point is the blue point, the same as the first point.

The schematic diagram of the long material movement is shown below:



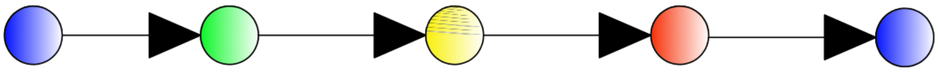
Long material point planning



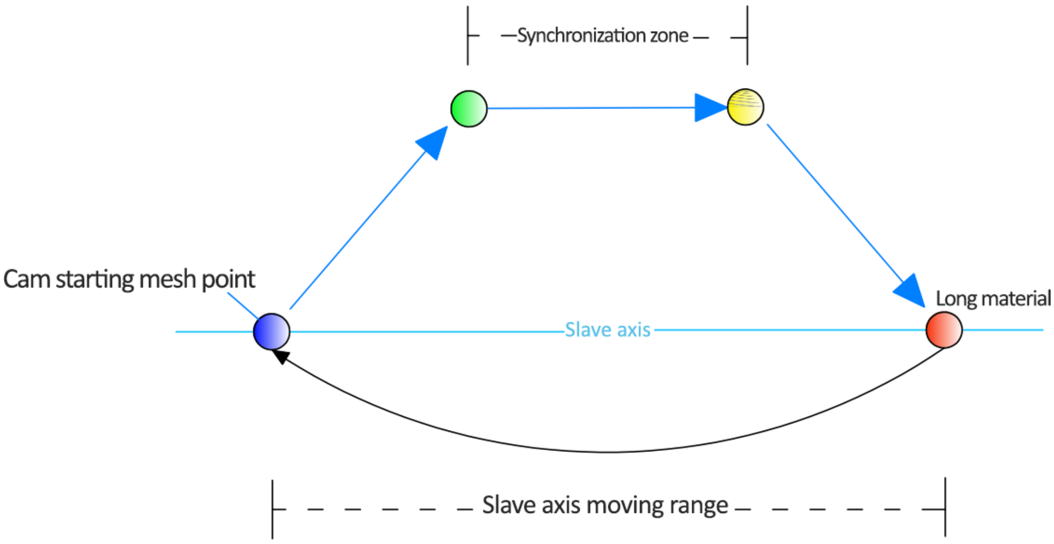
Long material master-slave relationship curve

2、Cutting short material

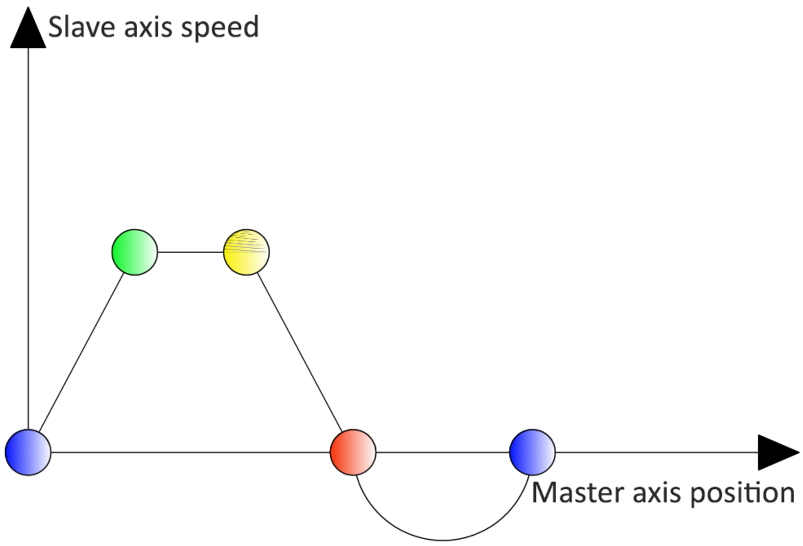
When the material is short, mode 0, because the cam engagement point is the starting point of the acceleration section, in order to ensure that the long and short materials can be switched normally during the running process, for the case of cutting short materials, the slave axis will no longer stop waiting, and its specific slave axis movement sequence is shown in the figure below:



The schematic diagram of the short material movement is shown below：



Short material point planning



Short material master-slave relationship curve

Note: The material length should not be too short. When the material length is too short, it may cause the slave axis to return too fast, which will lead to the servo alarm. Synchronous zone start point spindle position: from the axis moving range - synchronous zone length.

3、Parameter requirement

NO	Parameter Condition
1	Synchronization zone length (D) < slave axis travel range (L2)
2	Slave axis travel range (L2) < material length (L1)
3	Sync zone length (D) > 0

4	Synchronous zone compensation factor (K) >= 40
5	Synchronous zone compensation factor (K) <= 500
6	2 x slave axis travel range (L2) - length of synchronization zone (D) < material length (L1)

6.4.4.6 The starting segment is the synchronous zone mode description

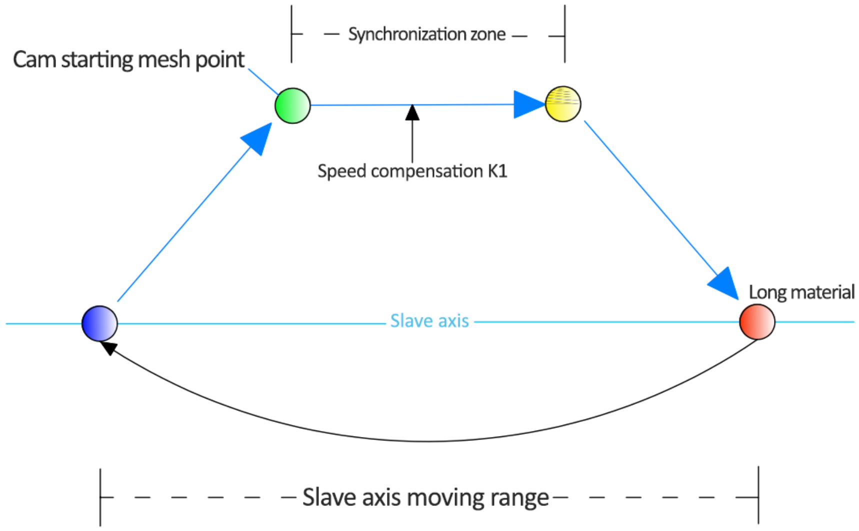
When the starting point is in the synchronous zone, it is the chasing shear function mode 1, which is implemented as follows: the starting point of the chasing shear curve is the starting point of the synchronous zone, and the speed of the slave axis synchronous zone can be compensated by the compensation coefficient.
In the actual chasing shear system, the slave axis follows the spindle in an electronic cam movement. The relationship between the length of the material and the range of movement of the slave axis is different, and the key points of the electronic cam are also different. The following describes how the slave axis follows the spindle movement in chasing shear mode 1.

1、 Cutting long material

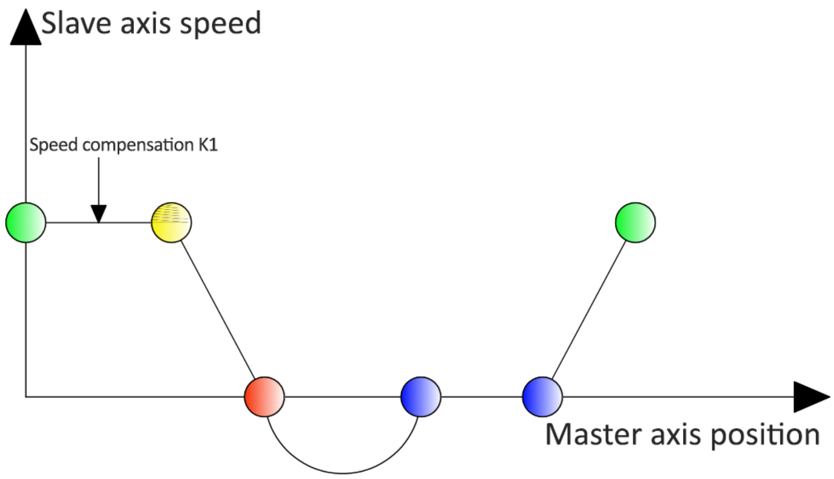
When the material length > from the axis moving range × (2 + return coefficient / 50) - (2 × synchronous zone compensation coefficient - 1) × synchronous length, the system will determine the material as long material and set LongShort to ON, according to the curve planning method of long material, the planning points of long material are 6 points. The specific sequence of moving from the axis is shown below：

Point 1: The first point is the green point, which is the starting point of the synchronization zone.
Point 2: The second point is the yellow point, which is the end of the synchronization zone. The slave axis moves from the green point to the yellow point at a uniform speed of the synchronization zone.
Point 3: The third point is the red point, which is the end point of the moving range from the axis, and the axis moves from the second point to the third point with gradual deceleration.
Point 4: The fourth point is the blue point, i.e. the starting point of the range of movement from the axis, from the axis from the third point back to the fourth point and stop waiting point.
Point 5: The fifth point is the blue point, which is the waiting time to the starting point of the acceleration from the axis to the synchronization area.
Point 6: the sixth point is the green point, the same as the first point.

The schematic diagram of the long material movement is shown below：



Long material point planning



Long material master-slave relationship

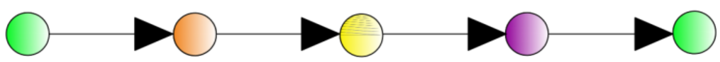
2、 Cutting short material

When the material is short, if the slave axis still moves according to the 6 points of the long material, then it may cause the slave axis to return too fast and the servo can't track. So for the case of short cut material, we modify the moving point.

Modification:

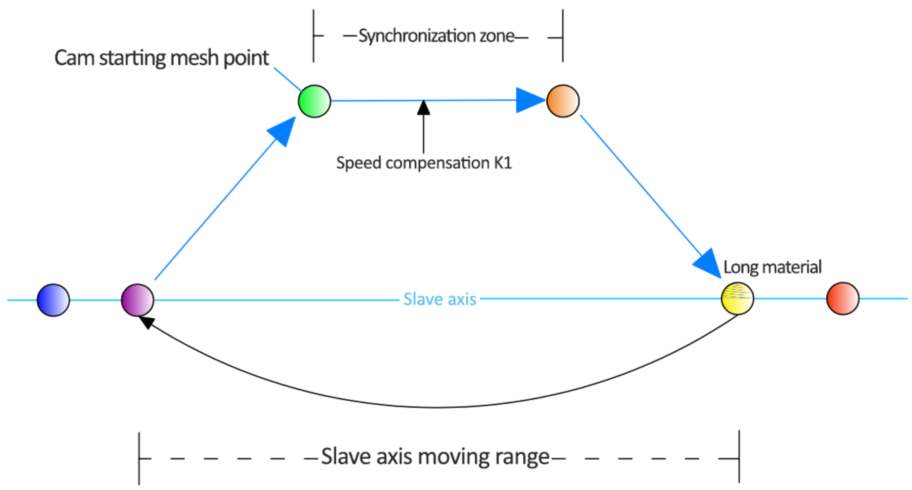
1) Reduce the deceleration time, i.e., decelerate from the axis from the yellow point, decelerate quickly to 0, not to the red point, but to the position behind the red point.

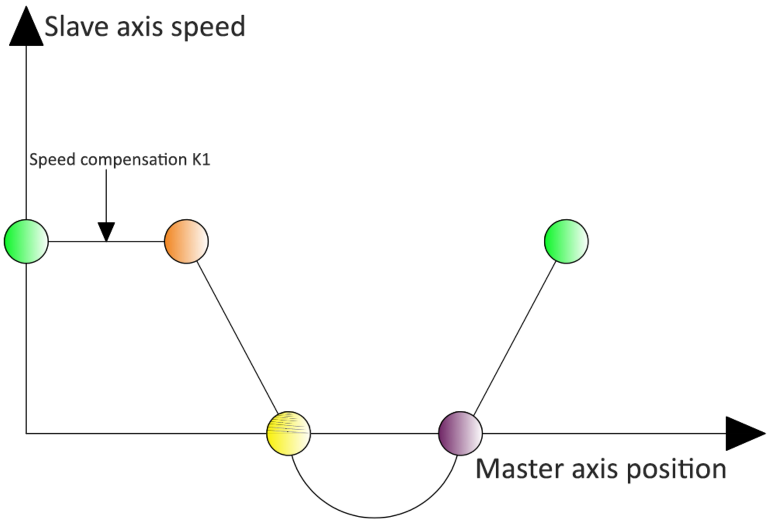
2) Reduce the acceleration time, i.e., the axis does not accelerate from the blue point, but starts to accelerate from the position before the blue point, so that when the axis returns, it does not return to the blue point position completely, but returns to the position before the blue point. The specific sequence of the slave axis movement is shown in the following figure:



The schematic diagram of the short material movement is shown in the following figure:

Short material point planning





Short material master-slave relationship curve

Note: The material length should not be too short. If the material length is too short, it may cause the return speed from the axis to be too fast, which will lead to the servo alarm.

3、Required parameters

NO	Parameter conditions
1	Length of synchronous zone (D) x compensation factor for synchronous zone (K1) < range of slave axis movement (L2)
2	Slave axis travel range (L2) < material length (L1)
3	Length of synchronous zone（D）>0
4	Return factor（K）>=40
5	Return factor（K）<=500
6	Synchronous zone compensation factor（K1）>=0.8
7	Synchronous zone compensation factor（K1）<=1.2
8	(Slave axis travel range (L2) + (1 - compensation factor for synchronisation zone (K1)) x length of synchronisation zone (D) < length of material (L1)

6.4.4.7 Application examples

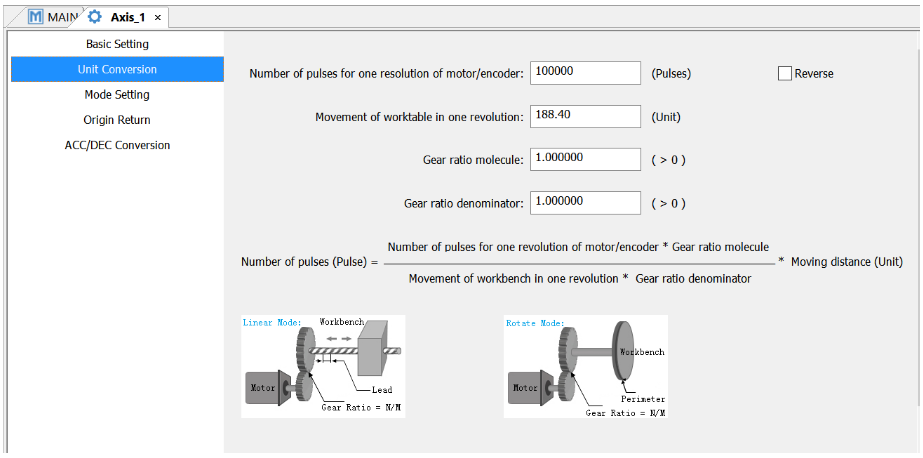
The function of chasing shear is illustrated below by an example of actual operation in chasing shear mode 1, with the system parameters shown below：

- Spindle diameter：60mm
- Spindle reduction ratio：1:10
- Spacing between slave axis screws：20mm
- Slave axis reduction ratio：1:1
- Moveable distance of the slave axis：400mm
- Length of slave axis synchronisation zone：300mm
- Cutting length：1320mm

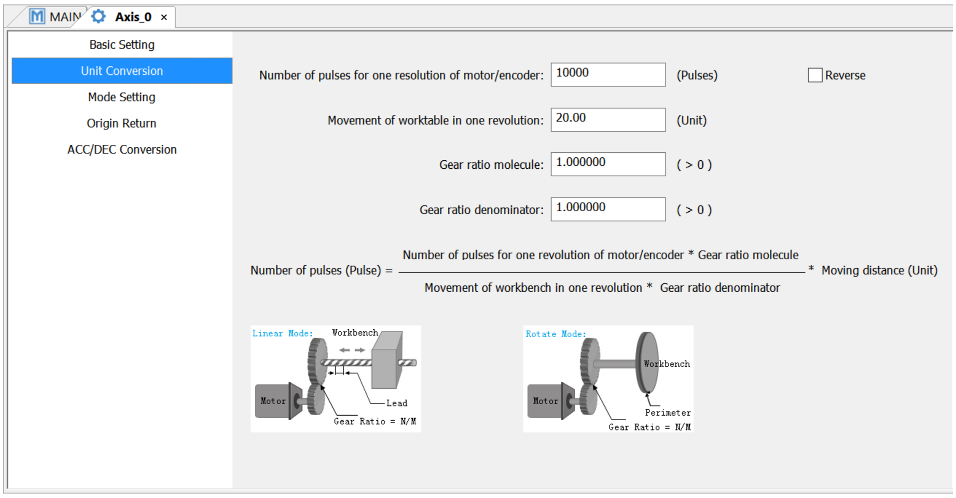
Parameters required to calculate chasing shear MC_ChasingCut command（using the drive as SD700 drive, 10,000 pulses input and one rotation of the motor）:

Spindle length in one revolution =60*π=188.4
Number of pulses per revolution of the spindle =10000*10=100000
Number of pulses per revolution of the slave axis =10000*1=10000
The return factor is recommended to be set to 100

1、Spindle parameter setting

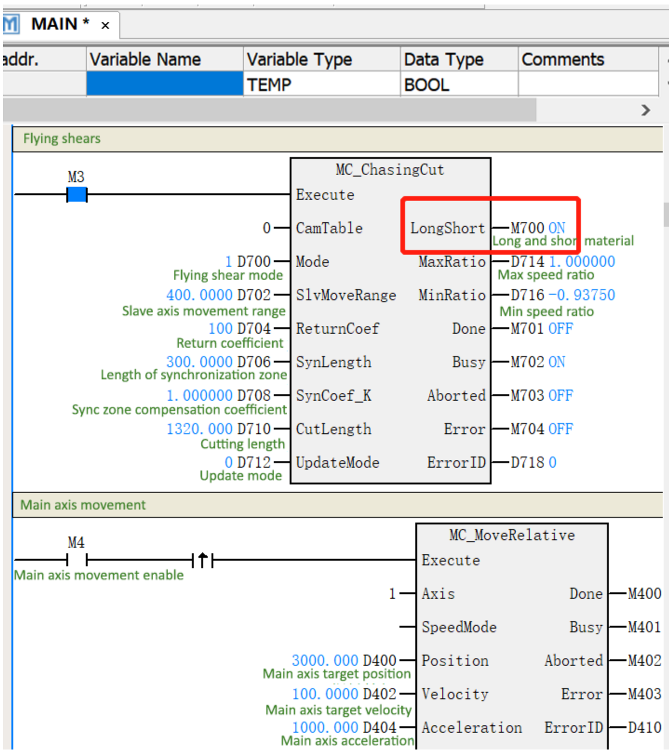


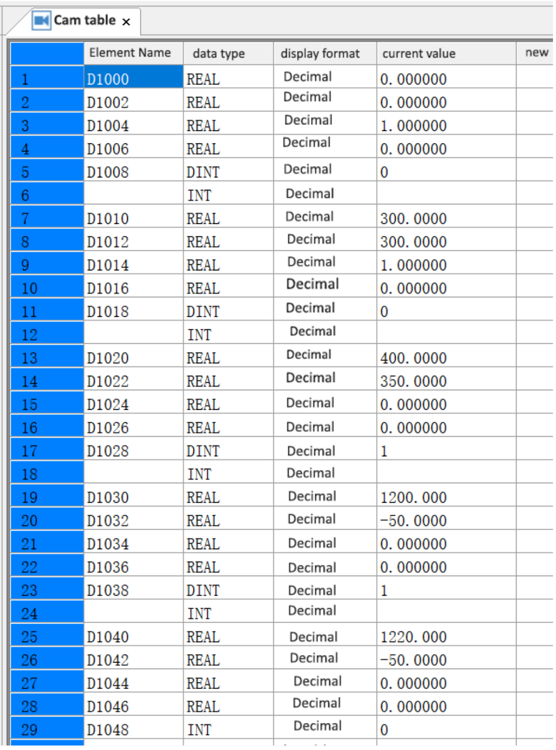
2、Slave axis parameter setting



3、Chasing shear curves

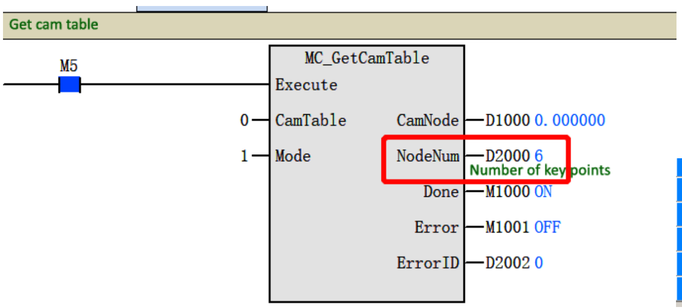
Acquisition of curve keypoint data using the acquisition cam table (mode 1)





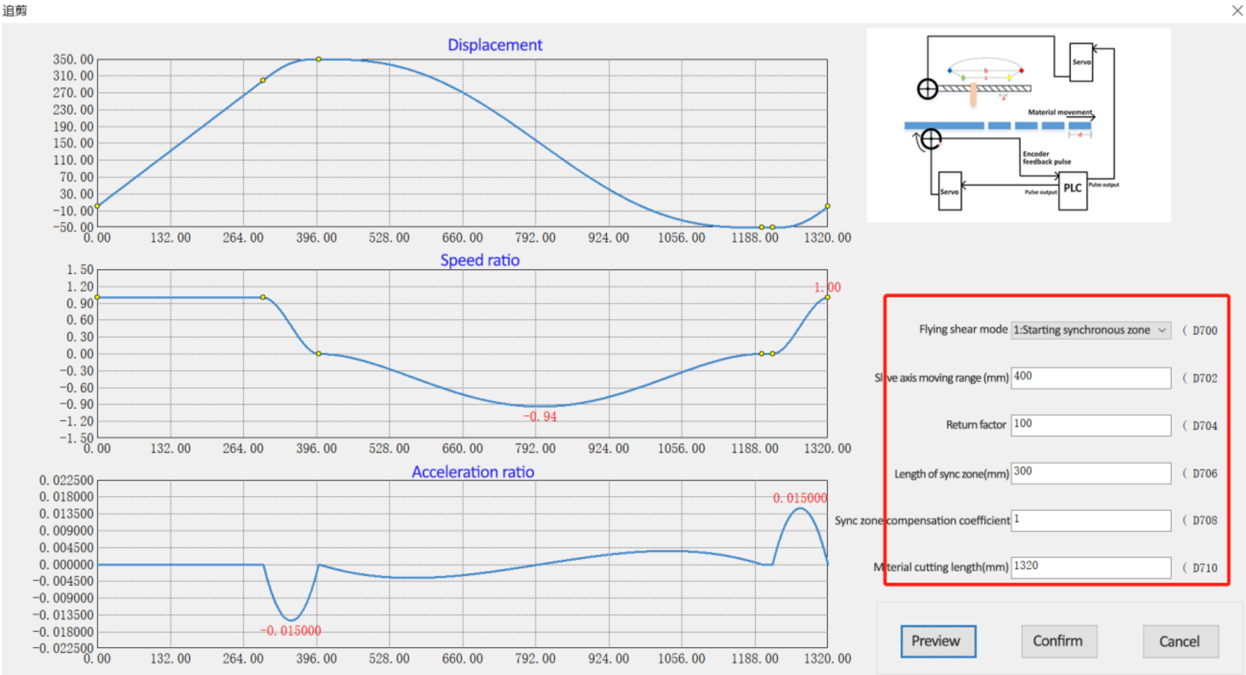
Cam table

	Element Name	data type	display format	current value
1	D1000	REAL	Decimal	0. 000000
2	D1002	REAL	Decimal	0. 000000
3	D1004	REAL	Decimal	1. 000000
4	D1006	REAL	Decimal	0. 000000
5	D1008	DINT	Decimal	0
6		INT	Decimal	
7	D1010	REAL	Decimal	300. 0000
8	D1012	REAL	Decimal	300. 0000
9	D1014	REAL	Decimal	1. 000000
10	D1016	REAL	Decimal	0. 000000
11	D1018	DINT	Decimal	0
12		INT	Decimal	
13	D1020	REAL	Decimal	400. 0000
14	D1022	REAL	Decimal	350. 0000
15	D1024	REAL	Decimal	0. 000000
16	D1026	REAL	Decimal	0. 000000
17	D1028	DINT	Decimal	1
18		INT	Decimal	
19	D1030	REAL	Decimal	1200. 000
20	D1032	REAL	Decimal	-50. 0000
21	D1034	REAL	Decimal	0. 000000
22	D1036	REAL	Decimal	0. 000000
23	D1038	DINT	Decimal	1
24		INT	Decimal	
25	D1040	REAL	Decimal	1220. 000
26	D1042	REAL	Decimal	-50. 0000
27	D1044	REAL	Decimal	0. 000000
28	D1046	REAL	Decimal	0. 000000
29	D1048	INT	Decimal	0
30		INT	Decimal	
31	D1050	REAL	Decimal	1320. 000
32	D1052	REAL	Decimal	0. 000000
33	D1054	REAL	Decimal	1. 000000
34	D1056	REAL	Decimal	0. 000000
35	D1058	INT	Decimal	0



4、Graphical drawing of a chasing shear effect

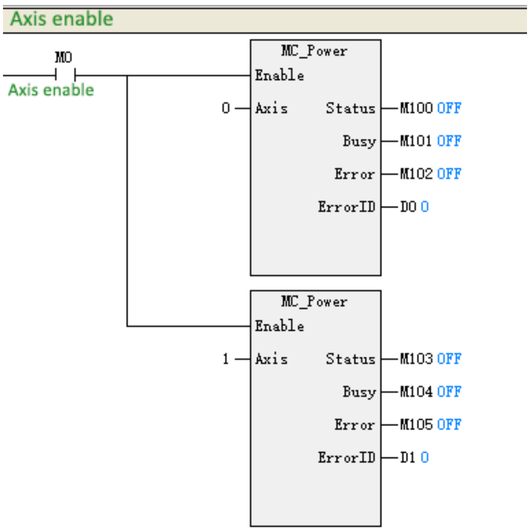
Right-click on the chase command - Open to view the chase effect curve.

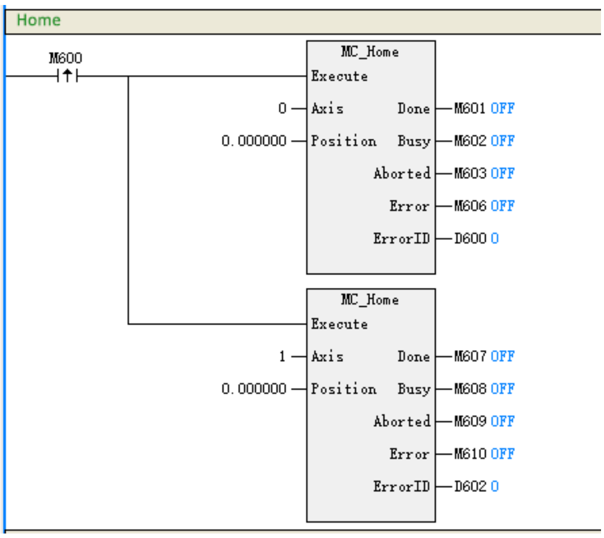


5、Procedure for chasing shear

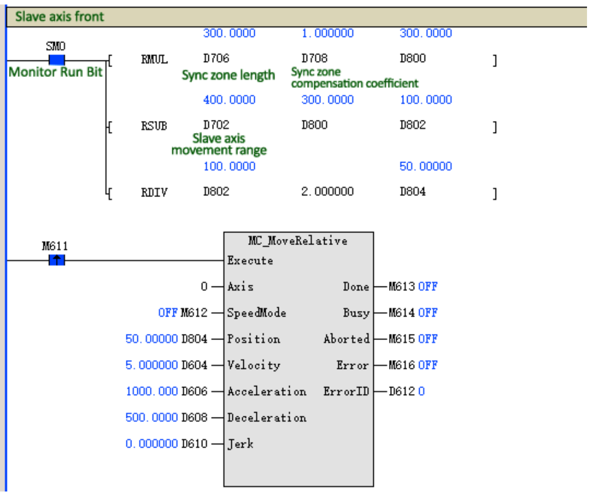
Next we will follow the step-by-step procedure to run the chasing shear program

Step 1: Home return.

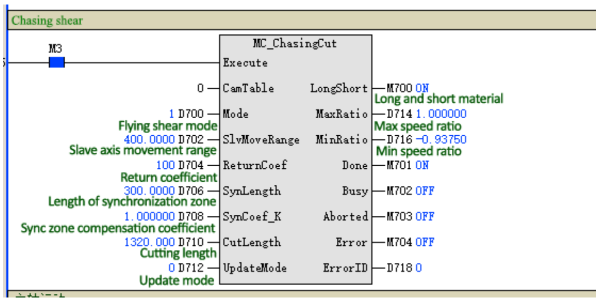




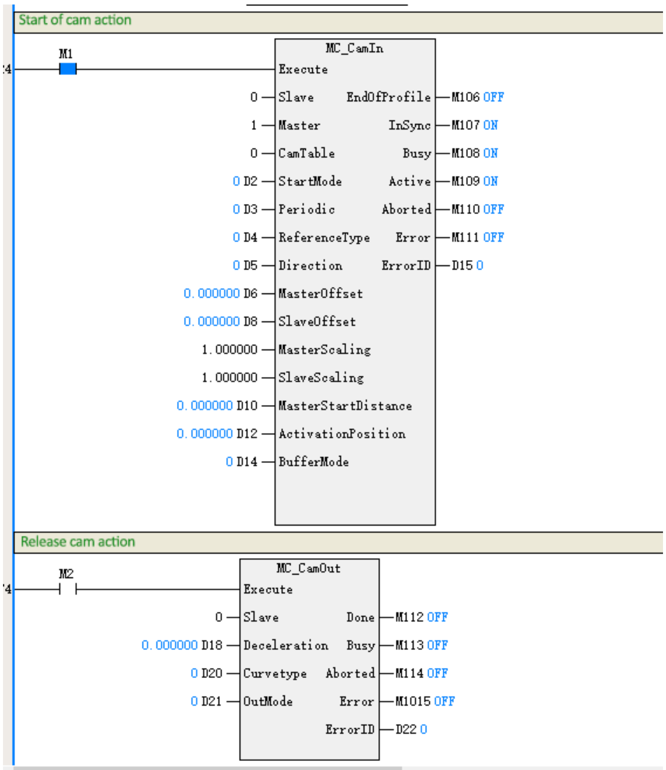
Step 2: After home return, the slave axis is advanced.



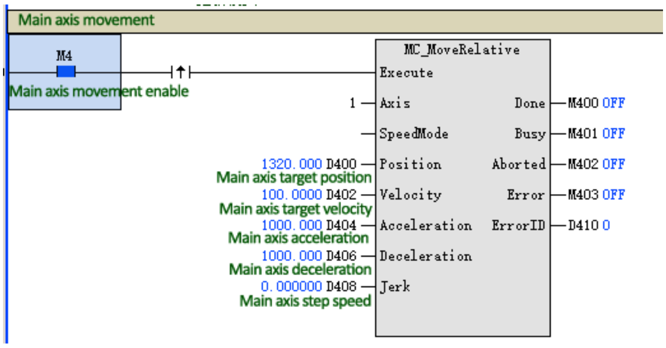
Step 3: Configure the generic cam and write the chasing shear parameters via the MC_ChasingCut command.

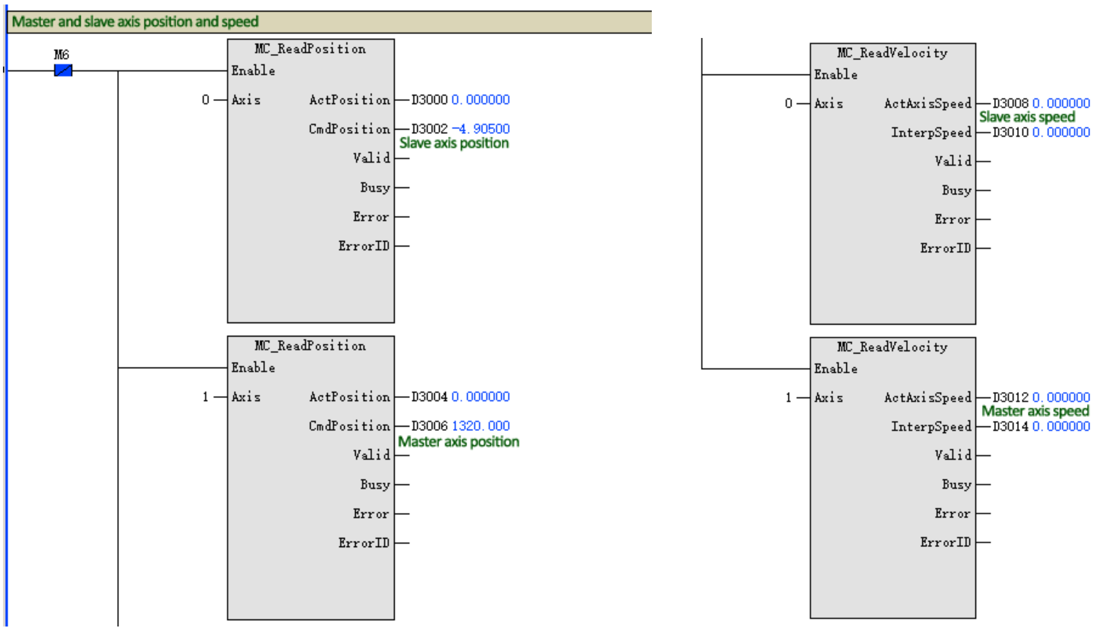


Step 4: Enable the cam. Engaging the master and slave axis.

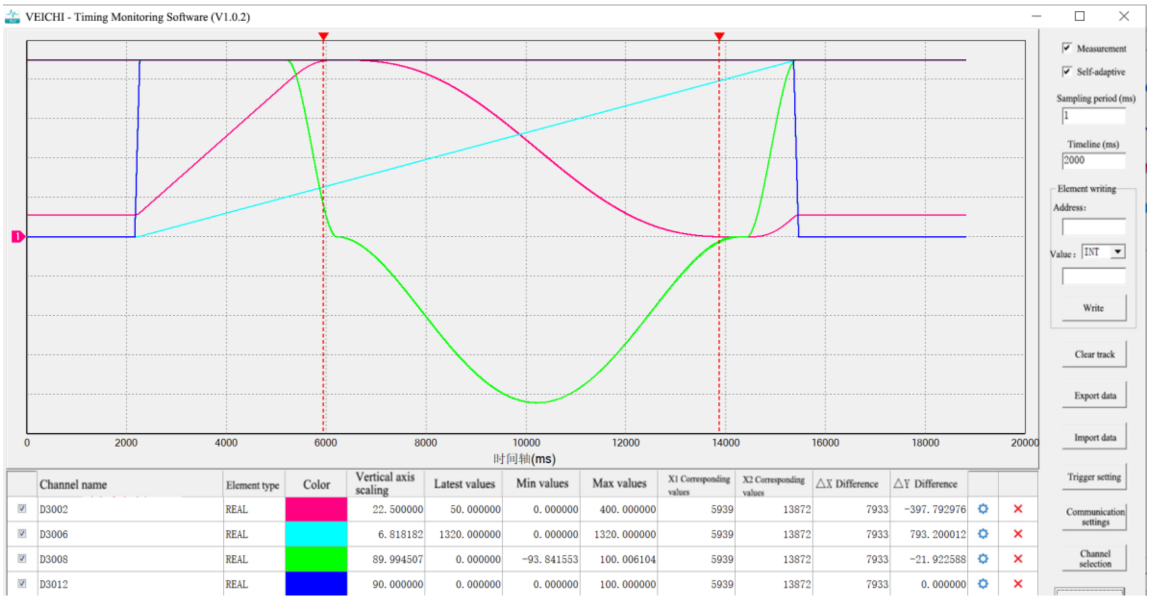


Step 5: The controller controls the spindle movement. The master-slave relationship operates according to the set cam relationship.





The actual results of the run are shown below:



7. EtherCAT communication commands

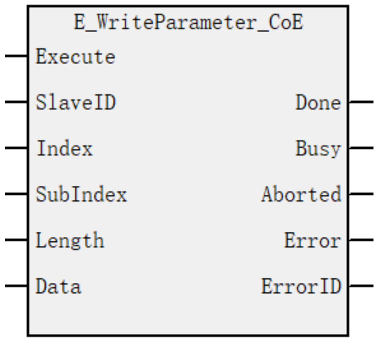
List of commands

The EtherCAT communication commands cover the following command entries

Command category	Command name	Command description
EtherCAT communication commands	E_WriteParameter_CoE	Write COE slave axis parameter command
	E_ReadParameter_CoE	Read COE slave axis parameter command

7.1 E_WriteParameter_CoE（Write COE slave axis parameter command）

Graphic block



Description of functions and commands

Input/Output	Name	Data type	Applicable (soft components)	Rage	Description	Enable
IN	SlaveID	WORD	Constants /D/R/W	0~71	EtherCAT Slave axis station number	Execute
IN	Index	WORD	Constants /D/R/W	0x0~0xffff	Index	
IN	SubIndex	WORD	Constants /D/R/W	0x0~0xff	Sub-index	
IN	Length	WORD	D/R/W	1/2/4	Parameter (byte) length	
IN	Data	DWORD	D/R/W	0x0~0xffff	Parameter values	
OUT	Done	BOOL	M/S	TRUE,FALSE	Sign of operation completion	
OUT	Busy	BOOL	M/S	TRUE,FALSE	Running flags	
OUT	Aborted	BOOL	M/S	TRUE,FALSE	Execution interruptions	
OUT	Error	BOOL	M/S	TRUE,FALSE	Wrong sign	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Function description

This command is used to write the object dictionary of an EtherCAT slave and is valid on the rising edge.

● SlaveID Used to specify the configuration address of the EtherCAT slave + 1.

● On the rising edge of Execute, the command latches the input parameters on the left and writes the data in Data to the dictionary of objects specified by Index and SubIndex.

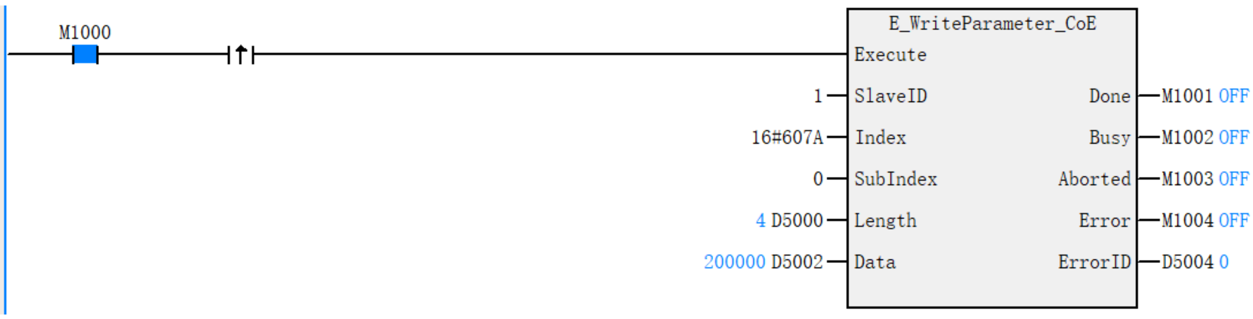
● Length is used to specify the length of the object dictionary to be written, in bytes (1, 2, 4 for BYTE, WORD, DWORD respectively).

● When a write succeeds, the Done signal is valid. When a write fails, the Error output is valid, and the Aborted and ErrorID are used in conjunction to determine the cause of the read failure.

Example program

The following routine implements the function of writing the target position by means of a command (in the case of drive SD700).

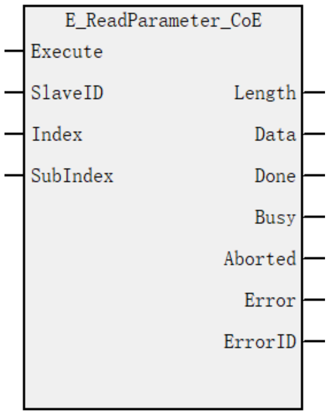
Target location object dictionary 16#607A, data type DWORD.



Note: SlaveID is the station number + 1 from the station configuration!

7.2 E_ReadParameter_CoE（Read COE slave axis parameters）

Graphic block



Description of functions and commands

Input/Output	Name	Data type	Applicable (soft components)	Rage	Description	Enable
IN	SlaveID	WORD	Constants /D/R/W	0~71	EtherCAT Slave axis station number	Execute
IN	Index	WORD	Constants /D/R/W	0x0~0xffff	Index	
IN	SubIndex	WORD	Constants /D/R/W	0x0~0xff	Sub-index	
OUT	Length	WORD	D/R/W	1/2/4	Parameter (byte) length	
OUT	Data	DWORD	D/R/W	0x0~0xffff	Parameter values	
OUT	Done	BOOL	M/S	TRUE,FALSE	Sign of operation completion	
OUT	Busy	BOOL	M/S	TRUE,FALSE	Running flags	
OUT	Aborted	BOOL	M/S	TRUE,FALSE	Execution interruptions	
OUT	Error	BOOL	M/S	TRUE,FALSE	Wrong sign	
OUT	ErrorID	WORD	D/R/W	0x0~0xffff	Error ID	

Function descriptions

This command is used to read the object dictionary of an EtherCAT slave.

● The SlaveID is used to specify the configuration address of the EtherCAT slave + 1.

● On the rising edge of Execute, the command latches the input parameters on the left and triggers the reading of the dictionary of objects specified by Index and SubIndex.

● Length is used to specify the length of the object dictionary to be read, in bytes (1, 2, 4 for BYTE, WORD, DWORD respectively).

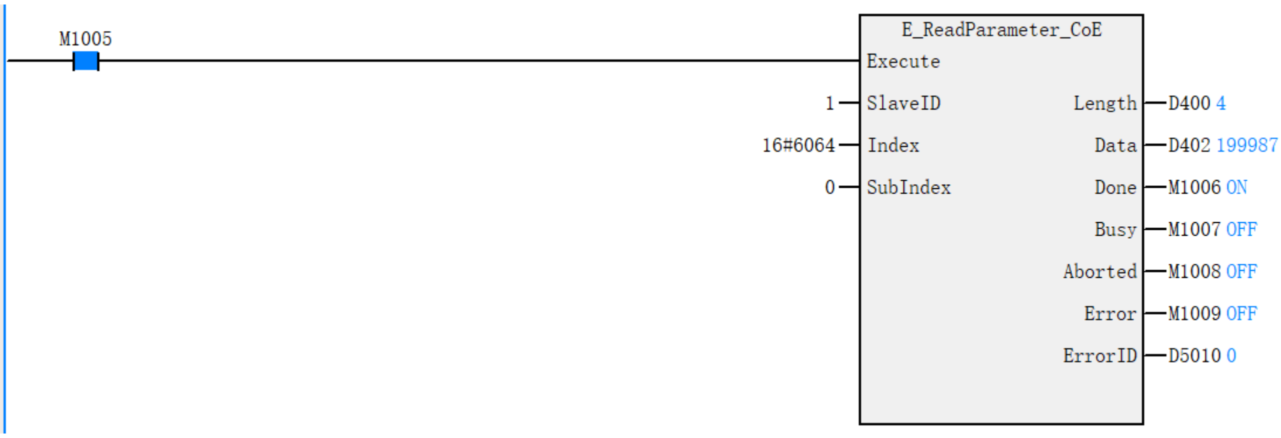
● When the read is successful, the Done signal is valid, Dtate is used to display the value read and Length is used to display the length of the actual object dictionary read. When the read fails, the Error output is valid and Aborted and ErrorID work together to determine the reason for the read failure.

● The Data parameter in this instruction is a DINT type parameter, occupying 4 bytes of space. When the object dictionary read is SINT or INT, the result of the read is placed in the lower 8 or lower 16 bits of the Data parameter, and then the unused higher 24 or higher 16 bits are filled with 0.

Example program

The following routines enable the feedback position to be read by command (in the case of the drive SD700).

Feedback position object dictionary 16#6064, data type DWORD.



Note: SlaveID is the slave shaft configuration station number + 1!

This command only supports level triggered reads once, not all the time, edge triggers are not valid!

8. Monitoring and Troubleshooting

8.1 EtherCAT master status monitoring register

<table>
<tr><td rowspan="3">EtherCAT Status</td><td>SD512</td><td rowspan="2">EtherCAT cycle time</td><td rowspan="2">RO</td><td rowspan="3">Overall status of EtherCAT</td></tr>
<tr><td>SD513</td></tr>
<tr><td>SD514</td><td></td><td>WR</td></tr>
</table>

Monitor Register	SD515			
	SD516			
	SD517		WR	
	SD518	EtherCAT task execution time		
	SD519		RO	
	SD520			
	SD521		WR	
	SD522			
	SD523		WR	
	SD524	EtherCAT master operating status	RO	
	SD525	Total number of currently connected slaves	RO	
	SD526	The current connection supports the number of DC slaves	RO	
	SD527	Cumulative period data packet loss times	WR	
	SD528	EtherCAT master PDO input byte length	RO	
	SD529	EtherCAT master PDO output byte length	RO	
	SD530	EtherCAT master sends jitter time periodically	RO	
	SD531			
	SD532~548	Reserve	WR	
	SD549	Monitor the slave station number: 1 ~ the maximum number of slave axes, which determines the ownership of the following parameters	WR	EtherCAT slave status
	SD550	Monitor the EtherCAT communication status of the slave axis number	RO	
	SD551	Monitor the AL status code of the slave axis number	RO	
	SD552	Monitor the configuration address of the slave axis number	RO	
	SD553	Monitor the alias of the slave axis number	RO	
	SD554	Monitors the manufacturer ID of the slave axis number	RO	
	SD555			
	SD556	Monitor the device ID of the slave axis number	RO	
	SD557			
	SD558	Monitor the input PDO bit length of the slave axis number	RO	
	SD559	Monitor the output PDO bit length of the slave axis number	RO	
	SD560	Monitor the topology of the slave axis number	RO	

	SD561	Monitor the active port of the slave axis number	RO	
	SD562~849	Reserve	WR	

8.2 Troubleshooting and Diagnosis

8.2.1 Fault indicator

The CAT light is used to display the fault status of the EtherCAT bus, and the meanings are as follows:

LED light status	Content
Extinguish	Communication not established
Flicker	Communication is normal with data receiving and sending
Always bright	Communication is normal without data receiving and transmitting

8.2.2 Error code

8.2.2.1 Motion control error codes

Error code	**Error message**	**Meaning**
2001	Axis number	Configure EtherCATServo axis and EtherCAT task not started
2002	Axis number	The configuration EtherCAT servo axis mapping EtherCAT slave station number is too large.
2003	Axis number	Axis running state error
2004	Axis number	Axis position out of tolerance alarm
2005	Axis number	Hardware positive limit
2006	Axis number	Hardware negative limit
2007	Axis number	Software positive limit
2008	Axis number	Software negative limit
2009	Axis number	The target speed exceeds the maximum value
2010	Axis number	Acceleration exceeds maximum value
2011	Axis number	Deceleration exceeds maximum value
2012	Axis number	The acceleration step value exceeds the maximum value
2013	Axis number	Torque exceeds maximum value
2014	Axis number	The axis is not mapped necessary PDO
2501	-	Cam channel error
2502	-	Cam table number setting error

2503	-	The Cam direction/tracking distance/starting position are not matched properly
2504	-	The number of cam points is wrong
2505	-	Effective mode error
2506	-	Cam point data error
2507	-	Wrong number of ejectors
2508	-	Ejector data error
2509	-	Incorrect camshaft number setting
2510	-	Cam calculation start and end point setting error
2511	-	More than 100 spindle phases
2512	-	CAM stop mode switch failed
2513	-	The shaft is already in the coupled state to repeat the conduction
2514	-	The cam master/slave scaling ratio is 0
2515		Get cam table data mode error
2516		No data in cam table cache
2517		No data in cam table operation
2520		Flying shear mode setting error
2521		The flying shear data setting is unreasonable
2522		Wrong shear mode setting
2523		Unreasonable setting of chasing data
2524		Flying shear return coefficient setting error
2551	-	Incorrect setting of gear axis number
2552	-	Gear denominator set incorrectly
2553	-	Gear acceleration setting is wrong
2554	-	Incorrect gear deceleration setting
2555	-	Wrong gear adding acceleration setting
2601	Axis number	Spindle position changes too much
2602	Axis number	Slave axis position changes too much
2603	Axis number	Slave axis not enabled
2604	Axis number	The slave axis acceleration exceeds the maximum acceleration
2605	Axis number	The deceleration of the slave axis exceeds the maximum deceleration
2606	Axis number	Slave axis adding acceleration exceeds maximum adding acceleration
2607	Axis number	Cam mode switch failed while axis is running
2700	Axis group quantity	Axis group amount out of range
2701	Axis group serial number	Axis group number configuration error
2702	-	Axis group number not existed or not configured
2703	Axis group serial number	Axis number setting out of range error
2704	-	The position of the axis in the axis group is set incorrectly

2705	-	Axis group or axis in non-stop condition
2706	-	Axis group command has repeated conduction
2707	-	Speed mode setting error
2708	-	Speed range setting error
2709	-	Acceleration range setting error
2710	-	Deceleration range setting error
2711	-	Add acceleration range setting error
2712	-	Speed over maximum
2713	-	Acceleration exceeding maximum acceleration
2714	-	Deceleration exceeding maximum deceleration
2715	-	Add acceleration exceeds maximum add acceleration
2716	-	Absolute relative mode setting error
2717	-	Cache mode setting error
2718	-	Transition mode setting error
2719	-	Transition parameter setting error
2720	-	Axis group instructions exceed the maximum number of cache entries
2721	-	Stop command execution in progress
2722	-	Stop mode setting error
2723	-	Not all axes in the axis group are enabled
2724	-	The circular interpolation plane is specified incorrectly.
2725	-	The circular interpolation mode is specified incorrectly.
2726	-	The circular interpolation path is specified incorrectly.
2727	-	Specified interpolation axis does not exist
2728	-	No interpolation axes in the axis group

2729	-	The starting point, ending point and passing position of circular interpolation are in a straight line
2730	-	Radius setting error
2731		Interpolation instruction type error
2732		Forward-looking calculations are not timely
2733		Greater than forward-looking cache space
2734		Forward-looking operation does not allow reloading
2735		Forward path line number error
2736		Forward-looking auxiliary parameters are set incorrectly
2737		Specify that the forward start line is out of range or invalid
2738	-	Forward interpolation parameter error
2800		Axis number out of range
2801	-	Axis in operation conduction command
2802	-	Axis not enabled
2803	-	60FF_PDO unmapped
2804	-	Pulse width over range

8.2.2.2 EtherCAT trouble codes（SD53）

Error code	Error message	Meaning
3001	High 16Bit: slave axis number	The EtherCAT configuration slave sends PDO entries that exceed the maximum limit
3002	High 16Bit: slave axis number	The index entry of PDO configuration object dictionary sent by EtherCAT configuration slave exceeds the maximum limit

	low 16Bit:PDOentry number	
3003	High 16Bit: slave axis number	The EtherCAT configuration slave receives PDO entries that exceed the maximum limit
3004	High 16Bit: slave axis number low 16Bit:PDOentry number	The EtherCAT configuration slave receives PDO configuration object dictionary index entries that exceed the maximum limit
3005	High 16Bit: slave axis number	EtherCAT configuration slave initialization parameter entries exceed the maximum limit
3006		The number of EtherCAT configured axes exceeds the maximum limit
3007	Axis number	EtherCAT configuration axis number exceeds the limit value
3008	Station no	EtherCAT configuration station number exceeds the limit
3009	Type	EtherCAT Configuration type error
3010	Counter number	Configure the local encoder axis counter number exceeds the limit value
3011	High-speed pulse port number	The port number of the configured local pulse axis exceeds the limit
3012	High 16Bit: slave axis number Low 16Bit: PDO entry number	Bus servo axis sends PDOMapping error
3013	High 16Bit: slave axis number Low 16Bit: PDO entry number	Bus servo axis receives PDOMapping error
3014	Axis number	Configure the local pulse axis Z pulse input out of port range
3015	Axis number	Error configuring local pulse axis origin input type
3016	Axis number	Error in configuring local pulse axis positive limit input type
3017	Axis number	Error in configuring local pulse axis negative limit input type
3018	Axis number	Configure local pulse axis probe input out of limits
3019	Axis number	Error configuring local pulse axis pulse output type
3020	Axis number	Configure the local pulse axis position comparison output port number exceeds the limit value or the output type is wrong
3021	Axis number	Error configuring local encoder type
3022	Axis number	Error configuring local encoder settings input port
3023	Axis number	Error configuring local encoder enable input port
3024	Axis number	Error configuring local encoder probe input

3025	Axis number	Error configuring local encoding position comparison output port or output type
3026	Axis number	Configure the rotation period of the rotary axis is too small error
3027	High 16Bits: Number of Configurations Low 16Bit: Number of connections	The number of configured sites does not match the actual number of connections
3028	Configure the slave number	Configuration DCMode actual slave does not support
3029	Configure the slave number	The configured slave device does not match the device identified by the connection
3030	Slave number	The mapped slave number exceeds the set value
3031	Slave number	Mapping slave sends PDOFailed to communicate abnormally
3032	Slave number	Mapping the slave to receive PDOFailed to communicate abnormally
3033	Slave number	Failed to initialize slave parameters
3034		System EtherCAT parameter is not configured
3035		EtherCAT slave PDO disconnected
3036	Slave number	EtherCAT slave system parameter configuration failed
3037		EtherCAT network no connection signal
3038		EtherCAT does not recognize the number of slave connections
3039	Slave number	Mapped element out of range
3040	Slave number	Error mapping component type
3041	Slave number	Error configuring local encoder preset input
3042	Axis number	Zero return speed setting error
3043	Slave station number	Connection port error
3044	High 16 bits：slave station number	Slot configuration exceeds maximum limit
3045	High 16 bits：slave station number	Slot configuration does not match actual connection
3046	Slave station number	Send Sync Manager configuration out of range
3047	Slave station number	Receive Sync Manager configuration out of range
3048	High 16 bits：slave station number Lower 16 bits: AL code	Slave status switching error
3049		Execution instruction error not initialized to completion

8.2.2.3 Command error code

Error code	Error message	Meaning
16	Command error	Command interrupt
17	Command error	Write SOD parameter error
18	Command error	Error reading SOD parameter
256	Command error	No configuration
257	Command error	Configuration error
258	Command error	Not running to running state
259	Command error	Command error
260	Command error	The bus servo axis has an alarm
261	Command error	Error writing SDO
262	Command error	Error reading SDO
263	Command error	Return to zero error
264	Command error	Timeout for return to zero
265	Command error	Command running abnormally
266	Command error	Axis in motion
267	Command error	Axis not enabled
268	Command error	Axis is already used
269	Command error	Axis stop error
270	Command error	Axis at stop
271 272	Command error Command error	Non-bus servo axes and local pulse axes There is no configuration related PDO, etc.
273	Command error	Reserve
274	Command error	Axis not running
275	Command error	Pause JOG movement
276	Command error	Non-bus servo axis
277	Command error	PDO does not support
278	Command error	Reserve
279	Command error	Reserve
280	Command error	Reserve
281	Command error	No negative limit input signal is used
282	Command error	Wrong axis type
288	Command error	Positive limit input signal is not used
289	Command error	No zero input signal is used
290	Command error	Negative limit input signal out of range
291	Command error	Positive limit input signal is out of range
292	Command error	Zero signal input is out of range
293	Command error	Z pulse input out of range
294	Command error	Counter type error

295	Command error	Probe without enable input
296	Command error	Wrong counter number
297	Command error	Compare channel is occupied
298	Command error	Comparison exception
299	Command error	Compare output not set
300	Command error	Illegal exit from Jog mode
301	Command error	Too low speed
302	Command error	SDO access is occupied
303	Command error	Not in location mode
304	Command error	Not done changing parameters
305	Command error	Data out of range
306	Command error	No enable input preset
307	Command error	Counter interrupt map number error
308	Command error	The number of comparisons is out of range
309	Command error	Both positive and negative directions are valid in jog mode
310	Command error	The starting point, end point and passing position of circular arc interpolation are on a straight line