

10. 5.8 Kinco Motion Control Functions

10. 5.8. 1 Overview

Kinco motion control functions are used to control Kinco's motion control products (servo and stepper drives) with CAN interface. It is based on the CANOpen protocol, encapsulates the CANOpen communication details with the driver, and combines the actual application requirements to provide users with a set of motion control instructions and corresponding network configuration tools. This function is easy to use, even if the user is not familiar with the details of the CANOpen protocol, it is easy to communicate with the drive and position control.

This function supports parameter upload (download), motor lock shaft, loose shaft, return to origin, jog (speed mode), absolute positioning, relative positioning and other operations for motion control products, and does not support operations such as torque mode and master-slave follow mode. In addition, this function can be used in principle for all third-party motion control products that support the standard CANOpen protocol, **please contact with Kinco before use.**

The number of motion control axis that can be controlled by different models of CPUs with this function is different, please refer to the following table. In practical applications, users can determine the actual number of connected units according to the needs of program space, network load rate, etc. At present, only the K209M and KS101M program storage is 8K instructions, and the other models are 4K instructions (a contact, a coil, and a function block instruction in the K series PLC are counted as one instruction for rough calculation).

Model	Controls the number of axes	The maximum number of MC instructions
K5 Series	Not supported	
K2 Series (K209M only)	32	1024
KS Series (KS101M only)		
KS Series (except KS101M)	16	192
KW series		

Calculation of the CAN bus load rate:

The network load rate is the percentage of bandwidth occupied by the network bus transmission data within 1s.

Calculated with standard frames and baud rate 1M, each PDO has 111 data bits, 1M baud rate is 1000000 bits /S, assuming that there are N PDOs on the bus, and the interval time sent by each PDO is t1, t2... tn ms.

$$\text{CAN bus load rate} = (1000\text{ms}/t_1 + 1000\text{ms}/t_2 + \dots) \cdot 111/1000000.$$

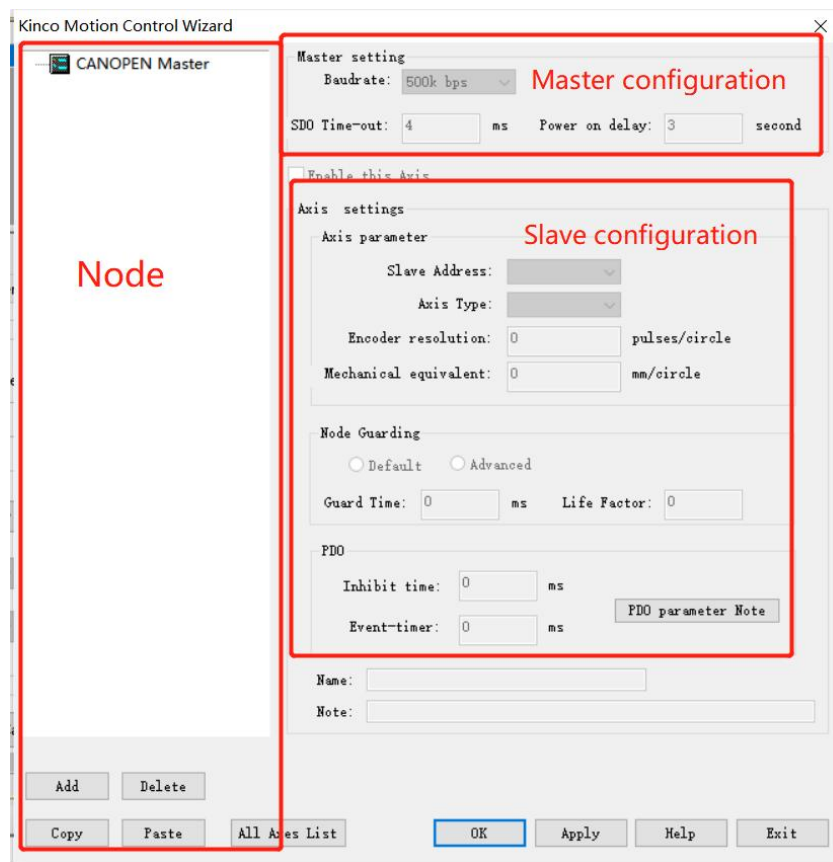
Users follow these steps to use the Kinco Motion Control feature:

- 1) In the user project, go to the Kinco Motion Control Network Configuration wizard window to complete the basic configuration of the network and axis parameters.
- 2) Call motion control instructions for programming according to actual needs.
- 3) Download the project to a PLC, which will run as a master station after starting, manage communications for the entire network, and execute positioning control procedures.

10. 5.8. 2 Kinco Motion Control Network Configuration

Kinco motion control functions using the CANOpen protocol, with the PLC as the master and the individual drivers as the slave. Before invoking an instruction, the user must first configure the canOpen network actually used. As is customary for field applications, we refer to slaves as "shafts" in software.

In the Project Manager of the Kincobuilder software, double-click the Kinco Motion Control Network Configuration node to enter the configuration window and complete the network configuration in this window.



The window is divided into three areas: a tree list of network nodes, master parameters, and parameters

for axis (slaves).

1) The operation of the network node tree

In the network node tree, the root node is the CANOpen master, and the following child nodes are the axes (slaves) in the network.

The following four buttons are provided: Add, [Delete], [Copy], and [Delete], while the software also provides corresponding shortcut keys and right-click menu functions. Users can take advantage of these features to operate on network nodes.

- Add a new axis

Click the Add button, right-click on either node and execute the Add menu command, or use the ALT+N shortcut. The new axes using the above three methods initially use the default parameters.

- Copy, paste

The user can first copy an existing axis, then paste it into the network to generate a new axis, the new axis in addition to the axis number (slave address), the other parameters are consistent with the copied axis. This feature is convenient for projects where all axis in a network have the same function.

Click an axis in the tree to select it, then click the Copy button, or use the Ctrl+C shortcut, or right-click an axis to execute the Copy menu command. All of these methods can replicate this axis.

When the copy is complete, click the Paste button, or use the Ctrl+P shortcut, or right-click on either axis and execute the Paste menu command to generate a new axis in the network.

- Delete an axis

You can delete an axis by clicking on it first, then clicking the Delete button, or by using the DELETE shortcut.

You can also delete an axis by right-clicking and executing the Delete menu command

2) Master parameters

Click the [CANOpen Master] node, all parameters of the master will be able to be modified, and all parameters of the axis (slave) will be grayed out and cannot be modified.

- Baud rate: Select the baud rate used by the master. Note that the baud rate of all nodes on the network (master and slave) must be consistent.
- [SDO timeout]: The timeout wait time after the master PLC sends the SDO request packet, if the corresponding slave does not receive the corresponding slave reply message after this time, a timeout error will be reported. When selecting a different baud rate, the software automatically recommends an SDO timeout period, which the user can modify based on this value.

3) Parameters of the axis (slave).

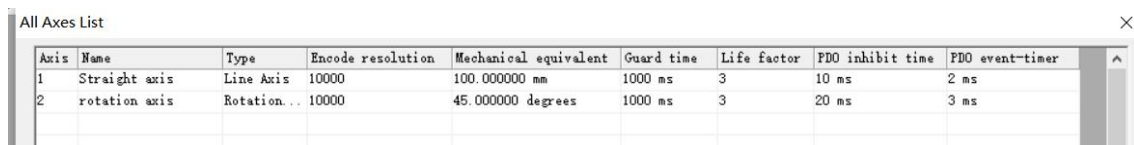
Click an axis node and all parameters of that axis will be modifiable, and all parameters of the master will be grayed out and cannot be modified.

- (1) **【Axis number】** : The CANOpen slave address of the axis, and the **slave station number in this system must be continuously assigned from 1.**

- (2) **【Type】** : Depending on the function of the axis, the user can choose a linear axis or a rotary axis.
- (3) [Encoder Resolution]: The resolution of the encoder of the shaft or stepper driver, that is, the number of pulses emitted by the encoder rotating one turn.
- (4) **【Mechanical equivalent per turn】**: For each turn of the motor shaft, the length of the mechanical load moving (linear shaft, mm) or the angle of rotation (rotation axis, °).
- (5) Node Protection: Set the node protection time for the axis. Users can use the default values or click "Advanced" to modify themselves.
- (6) [PDO prohibition time]: PLC for each axis automatically established a plurality of PDO for the transmission of position, speed, status and other information, because the position of the axis, speed and other changes quickly, so PDO transmission is very frequent, must set the PDO prohibition time. Users can use the default values or modify them themselves.

4) Other operations

- (7) OK: Save the parameters configured by the current interface and exit the interface
- (8) [Cancel]: Only save the parameters configured by the current interface and have clicked on the application, and then exit the interface
- (9) **【Application】** : Save the parameters configured in the current interface
- (10) **【Axis List】** : The Axis List is mainly used to facilitate viewing the parameters of all configured and enabled axis configurations for easy verification



Axis	Name	Type	Encode resolution	Mechanical equivalent	Guard time	Life factor	PDO inhibit time	PDO event-timer
1	Straight axis	Line Axis	10000	100.000000 mm	1000 ms	3	10 ms	2 ms
2	rotation axis	Rotation...	10000	45.000000 degrees	1000 ms	3	20 ms	3 ms

10.5.8.3 Kinco Motion Control Instructions

10.5.8.3.1 Overview of Motion Control Instructions

The following commands are located in the Kinco Motion Control group of the instruction set.

name	Feature description
MC_RPARAS	Read the parameters within the shaft drive (see the parameter table below)
MC_WPARAS	Modify the parameters within the shaft drive
MC_POWER	Control the lock shaft, loose shaft
MC_RESET	Resets the error message on the axis to set the axis state to a quiescent wait state
MC_HOME	The control axis returns to the origin
MC_JOG	Controls the jog of the axis
MC_MABS	The control axis performs absolute positioning motion
MC_MREL	The control axis performs relative positioning motions
MC_STATE	Reads the values for each status of the drive

MC_RESTART	Reconfigure and start the slave
MC_MIOT	Read device information such as the serial number, software version, IIT, temperature, etc. of the target axis

1) Precautions

When using these instructions, users need to pay attention to the following points:

- The maximum number of axes allowed in a user project: 16 KS and KW series, 32 KS101M and K209M.
- In a user project, the total number of dedicated instructions used is limited to: KS (except KS101M) and KW series up to 192, KS101M and K209M maximum 1024. Among them, only 1 MC_MIOT instruction is allowed per axis.
- For the same axis, when a private instruction is executing and has not yet completed, it is not allowed to start executing another private instruction. If the user program starts another private instruction at this time, the instruction will directly end and report an error.
- For the same axis, MC_MIOT instructions have the lowest priority: if other instructions are running, the MC_MIOT instructions will not be executed; if the MC_MIOT instructions are in the process of being executed, the program is in the process Other instructions in the startup, the MC_MIOT instruction will be terminated directly.
- For the same axis, before the user program executes the motion instruction (without reading and writing parameter instructions), it must first execute the MC_POWER instruction to lock the shaft, and after the lock shaft is successful, it can continue to execute back to the origin, relative motion, absolute motion or jog instruction. If there is no key lock, then the execution of these instructions will directly end and report an error.
- For the same axis, the user program uses the MC_ RESET command to reset, after the reset is successful, the shaft will be in a resting waiting state of the loose shaft, and the MC_POWER instructions need to be executed to lock the shaft before continuing to execute back to the origin, relative motion, absolute motion or jog instructions.
- For the return-to-origin, relative motion, absolute motion, or jog command, the acceleration and deceleration speed used is the acceleration and deceleration speed set internally by the driver, and the user can also set it by MC_WPARAS command.
- The output of invoking individual motion control instructions in a user program is irrelevant. If an instruction is executed incorrectly, its output parameter ERRID will give an error code, and the error result will not be refreshed again until the next time the instruction is executed again, and the result of other instruction execution will not affect the execution result of the instruction!
- After the bus is dropped (the ONLINE output of the MC_STATE instruction is 1), for safety reasons, this set of instructions will not be automatically reconnected! The user must eliminate the error before the instruction can be re-executed after the power failure restart PLC!
- As shown in the error code 3 of the following table, this set of instructions does not support the simultaneous execution of multiple operation control actions. At the same time, the output completion parameters of this set of instructions are different from the input specified in the advanced motion

control specifications, such as the positioning completion of the DONE signal may have a certain delay output, that is, the current state of the motor shaft cannot be highly responsive, and the user should consider the delay if the signal is used as the basis for judging the mutual action of each command, otherwise there may be results that are inconsistent with the expected process!

2) Instruction output parameter ERRID

Each instruction provides ERRID output parameters. If the instruction is executed successfully, the ERRID output is 0. If the instruction execution fails, ERRID is set to a different error code value to explain the cause of the error.

Below is a description of each error code value (Note that **the error code here is not suitable for MC_RPARAS and MC_WPARAS instructions, the error codes of these two instructions have a special meaning**, please refer to the instruction description separately).

Error code	description
0	inerrancy
1	The target axis is not enabled, or it does not exist in the network
2	The target axis is not in the locked shaft state.
3	The target axis is executing other motion control commands and is not stationary.
4	The CAN packet send buffer inside the PLC is full and cannot be sent
5	The PLC sent an SDO request message to the target axis, but the timeout did not receive a response
6	The PLC sent an SDO request message to the target axis, but received an incorrect response message
7	The instruction executes normally, but the PLC continuously detects the state returned by the target axis and ultimately does not detect the correct state value

In motion control, the axis is divided into several logical states, and the direct transfer of each logical state requires specific conditions or specified MC operation command. The advantage of this division processing is that it is convenient for the axis to be classified and controlled according to the motion mode, the axis can only be in one logical state at a time, and the transfer of the logical state needs to be carried out according to the rules, and will not be chaotic due to the accidental triggering of different MCs.

The logical states of an axis are divided into the following six possible states:

0:Power_off (Disabled): The shaft is not powered on, or is not enabled, and the MC_Power instructions need to be executed

1: Errorstop: ————— Execute the MC_Reset instruction first

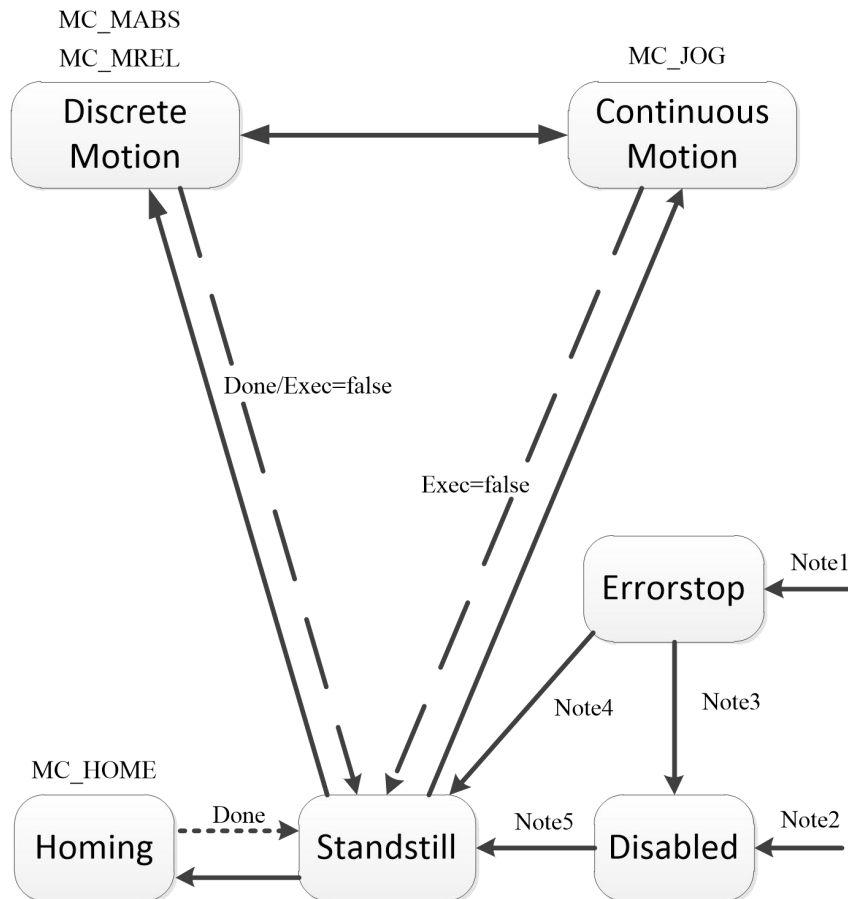
2: Standstill: ————— axis stops running

3:Discrete_Motion: the ——— axis is in discrete operation

4:Continuous_Motion:— The shaft is in continuous operation

5:Homing:————- Axis is in zero-to-zero operation, waiting for the zeroing operation to complete

The axis state transition diagram is as follows:



Note1: From any state. An error in the axis occurred.

Note2: From any state. MC_power.exec=false and there is no error in the axis.

Note3: MC_reset and MC_power.exec=false and there is no error in the axis.

Note4: MC_reset and MC_power.exec=true and there is no error in the axis.

Note5: MC_power.exec=true.

10.5.8.3. 2 motion control instructions

10.5.8.3. 2.1MC_RPARAS (read parameters) and MC_WPARAS (modify parameters).

The purpose of this set of instructions is to facilitate the user to operate the drive parameters in batches, for example, the user can set the parameters of the drive at one time at the beginning of debugging. How to set the specific parameters, please consult the drive operation manual, **improper settings may run abnormally,**

please be careful.

Note: The parameters not in the following list can be read and written by SDO instructions to send SDO messages, please refer to [10.5.4.5.2 SDO instructions for instructions](#) on the use of SDO instructions.

1) List of operational drive parameters

The following parameters of the drive can be operated on by the drive read and write instructions, all of which are readable and writable. Each instruction operates a maximum of 32 parameters at a time. In the table of process data types, REAL represents single-precision floating-point numbers, UINT32 represents unsigned 32 digits, INT32 represents signed 32 digits, and so on.

The "ordinal" value in **the** table is fixed, each parameter has an ordinal number, and the user can enter the ordinal number in the instruction to manipulate the corresponding parameter. "Process unit" refers to the unit used in the instruction parameters, and "drive value range" refers to the value range inside the drive (this instruction will automatically convert the actual process parameter values required by the user into the data format used internally by the drive, such as acceleration, speed, position, etc.).

serial number	Parameter name	CANOpen object	Process data type	Process units	The range of values in the drive
0	Trapezoidal acceleration	0x60830020	REAL	Linear axis: mm/s ² Axis of rotation: 1/s ²	[0,268435455]
1	Trapezoidal deceleration	0x60840020			
2	Find the origin speed	0x60990120	REAL	Linear axis: mm/min Axis of rotation: degrees/min	[-2147483648, 2147483647]
3	Find the origin mode	0x60980008	INT8	DEC	[-128,127]
4	The speed ring proportional gain is 0	0x60F90110	UINT16	DEC	[0,32767]
5	The velocity loop integral gain is 0	0x60F90210	UINT16	DEC	[0,32767]
6	The position ring proportional gain is 0	0x60FB0110	REAL	HZ	[0,32767]
7	Position ring speed feedforward	0x60FB0210	REAL	%	[0,1024]

8	The speed loop proportional gain is 1	0x23400410	UINT16	DEC	[0,32767]
9	The speed loop integral gain is 1	0x23400510	UINT16	DEC	[0,32767]
10	The position ring proportional gain is 1	0x23400610	REAL	HZ	[0,32767]
11	Target current limit	0x60730010	UINT16	DEC	[0,2048]
12	Maximum speed limit	0x607F0020	REAL	Linear axis: mm/min Axis of rotation: degrees/min	[-2147483648, 2147483647]
13	Origin offset mode	0x60990508	UINT8	No units	[0,255]
14	Motor orientation	0x607E0008	UINT8	No units	0 and 1
15	Motor model	0x64100110	UINT16	No units	[0,65535]
16	The soft limit is positively set	0x607D0120	REAL	Linear axis: mm Axis of rotation: Degrees	[-2147483648, 2147483647]
17	Soft limit negative setting	0x607D0220			
18	Smooth filtering	0x60FB0510	UINT8	No units	[0,255]
19	Maximum follow error	0x60650020	UINT32	DEC	[0,268435455]
20	The target location window	0x60670020	UINT32	DEC	[0,268435455]
21	Position window time	0x60680010	UINT16	DEC	[0,32767]
22	Speed feedback filtering	0x60F90508	REAL	HZ	[0,45]
23	Speed feedback mode	0x60F90608	UINT8	No units	[0,85]
24	Input port polarity	0x20100110	UINT8	No units	[0,255]
25	Input port 1 function	0x20100310	UINT16	No units	[0,65535]
26	Input port 2 function	0x20100410	UINT16	No units	[0,65535]
27	Input port 3 function	0x20100510	UINT16	No units	[0,65535]
28	Input port 4 function	0x20100610	UINT16	No units	[0,65535]
29	Input port 5	0x20100710	UINT16	No units	[0,65535]

	function				
30	Input port 6 function	0x20100810	UINT16	No units	[0,65535]
31	Input port 7 function	0x20100910	UINT16	No units	[0,65535]
32	Input port 8 function	0x20101D10	UINT16	No units	[0,65535]
33	Output port polarity	0x20100D10	UINT8	No units	[0,255]
34	Output port 1 function	0x20100F10	UINT16	No units	[0,65535]
35	Output port 2 function	0x20101010	UINT16	No units	[0,65535]
36	Output port 3 function	0x20101110	UINT16	No units	[0,65535]
37	Output port 4 function	0x20101210	UINT16	No units	[0,65535]
38	Output port 5 function	0x20101310	UINT16	No units	[0,65535]
39	Output port 6 function	0x20101E10	UINT16	No units	[0,65535]
40	Output port 7 function	0x20101F10	UINT16	No units	[0,65535]
41	Gear front pulse data	0x25080420	INT32	DEC	[-2147483648, 2147483647]
42	Pulse mode	0x25080308	UINT8	No units	[0,255]
43	Save the parameters	0x10100120	UINT32	No units	Only 16# 65766173 valid
44	Initialization parameters	0x10110120	UINT32	No units	Only 16#64616f6C is valid

2) Description of ERRID parameters

Both the Read and Write Parameters directive provide *ERRID* (DWORD type) output parameters.

This parameter value is an error code that indicates an error that occurred during instruction execution.

Error code	meaning
0xFFFFFFFF	An error occurred that prevented the instruction from executing, including: 1) The user entered the wrong axis number and the number of parameters 2) There are other Kinco-specific instructions running 3) The instructions want to operate 32 parameters, and the result is that these 32 parameters fail to operate

Other values	Each bit of ERRID represents the operation result of the corresponding parameter, and each bit corresponds to the parameter specified in the ID parameter sequence table: bit0 represents the result of the first parameter of the operation, and bit1 represents the second the result of the operation of a parameter, and so on. A bit value of 1 indicates that the operation of the corresponding parameter failed, otherwise it indicates that the operation of the corresponding parameter was successful.
--------------	---

3) MC_RPARAS (read parameters).

	name	Directive format	Applies
LD	MC_RPARAS	<pre> MC_RPARAS EN ENO EXEC DONE AXIS ERR ID ERRID NUM PARAS </pre>	<input checked="" type="checkbox"/> KS <input checked="" type="checkbox"/> KW103 <input checked="" type="checkbox"/> KW203 <input checked="" type="checkbox"/> K209M

parameter	Input/Output	data type	Allowed memory areas	description
EXEC	input	BOOL	M、V、L、SM	If an ascending edge of EXEC is detected, the order is triggered for execution.
AXIS	input	INT	V、M、L、 constant	The axis number of the target axis (that is, the address of the CANOpen slave).
ID	input	BYTE	V、M、L、 constant	The starting address of the ordinal table of parameters to read.
A	input	INT	V、M、L、 constant	The number of parameters to read
DONE	output	BOOL	M、V、L	Completion flag bit. When the instruction execution is complete, <i>DONE</i> jumps from 0 to 1.
ERR	output	BOOL	M、V、L	Error flag bit. If an error occurs while the instruction is executing, it is set to 1.
ERRID	output	DWORD	V、M、L	Error code
BEST	output	DWORD	V、M、L	The storage start address of all parameter values read.



AXIS and NUM must be both constant types and memory types, and the ID and PAPAS parameters together form a variable-length memory block that must be All are in a legitimate memory region, otherwise the results are unpredictable.

The three parameters of ID, PARAS, and NUM together form a parameter table. Where ID is the starting address of the sequence number table, from this address onwards, the sequence number of each parameter to be operated is stored sequentially (that is, the "ordinal number" in the parameter list above), each sequence number occupies 1 byte; PARAS is the starting address of the parameter value table. From this

address onwards, the values of each parameter read are stored sequentially, each of which occupies 4 bytes; NUM is the number of parameters to be operated. For example, in the following example, it is assumed that the ID parameter is VB100, the PARAS parameter is VD1200, and the NUM parameter is 3, then VB100, VB101, VB102 respectively store the sequence numbers of the 3 parameters to be operated this time, and when the instruction is executed, read the 3 The parameter values are stored in VD1200, VD1204, and VD1208.

For PARAS, it should be noted that although the parameter value table uniformly adopts the DWORD address, the actual data type of each process parameter is not the same, so the user in the user program in the table should process the data in the parameter table according to the actual data type.

- ✧ If the actual process data type is REAL, then the parameter memory can be operated directly with the floating-point number address. For example, in the following example, the ordinal number 0 of the first parameter value is a REAL parameter, stored in VD1200, then VR1200 can be operated directly. Because VD1200 and VR1200 actually occupy the same memory address in the PLC.
- ✧ If the actual process data type is a data type other than REAL, and the corresponding parameter memory does not enforce the data type in the global variable table, then it is sufficient to read the parameter memory directly, because the instruction automatically handles various signed and unsigned integers. For example, in the following example, the sequence number of the third parameter value is 8 stored in VD1208, and the actual type is INT32 or UINT32, then directly operate VD1208.

- 1) Instruction description in LD format
- 2) If EN is 1, then on the rising edge of the EXEC input, the instruction is triggered to execute, and the instruction is sent to the SDO according to the ID, NUM specified by the parameter table to be read. Give the drive to read the corresponding object and put the read data into the paras-specified numerical table in turn, with the ERRID bit set to 0. If the SDO response of a parameter is incorrect or the timeout is not responding, the data of the corresponding address in PARAS remains unchanged, and the corresponding BIT OF ERRID is set to 1, and then the next parameter is read. When all the parameters are read, DONE is set to 1, and ERR and ERRID are set to different values according to the execution result.
- 3) If EN is 0, the instruction is not executed. When EXEC becomes 0 during the execution of the instruction, the instruction stops reading the parameters that have not yet been completed, sets THEONE to 1, and ERR and ERRID maintain the executed results.
- 4) If the PLC detects an error when the instruction is started (such as the axis is not enabled, the axis is executing other instructions, etc.), it directly exits, sets THE DONE, ERR to 1, AND ERRID to the corresponding error code.

➤ **example**

This example is in IL format. In Kincobuilder, select the IL format in the Project menu, then copy and paste the example into the editor, and then select LD Format, and the program can be displayed as LD

format.

(* Network 0 *)

(*Set the parameter table to indicate that you want to read parameters 0, 3, and 8.*)

```
LD      %SM0.0
MOVE    B#0,%VB100
MOVE    B#3,%VB101
MOVE    B#8,%VB102
```

(* Network 1 *)

(* Invoke instructions.) This time, AXIS and NUM parameters are constants, and they also support the format of full memory addresses. *)

```
LD      %M0.0
MC_RPARAS %M0.1, 1, %VB100, 3, %M0.2, %M0.3, %MD8, %VD1200
```

(* Network 2 *)

(*The read parameter values are stored in the table of parameter values at the beginning of the PARAS parameter (from 1200 in this case in the V zone).) The first data in the table is the first parameter value read, that is, parameter 0 in the table, because it is OF TYPE REAL, so the floating-point memory address is read.

*)

```
LD      %SM0.0
MOVE    %VR1200,%VR300
```

(* Network 3 *)

(*The second data in the table is the second parameter value read, that is, parameter 3 in the table.) This parameter is signed with 8 digits, and because this data type is not provided in the PLC, it is processed as an integer. *)

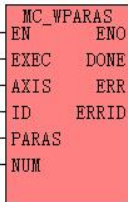
```
LD      %SM0.0
DI_TO_I %VD1204,%VW304
```

(* Network 4 *)

(*The third data in the table is the third parameter value read, which is parameter 8.) This parameter is unsigned 16 digits, but the maximum range is 32767, so the program can be treated by INT or WORD type, but it is best to judge whether the value is within the allowable range first. *)

```
LD      %SM0.0
DI_TO_I %VD1208,%VW308
BN %VW308, 0
ST      %M3.0
```

4) MC_WPARAS (modify parameters).

	name	Directive format	Applies
LD	MC_WPARAS		<input checked="" type="checkbox"/> KS <input checked="" type="checkbox"/> KW103 <input checked="" type="checkbox"/> KW203 <input checked="" type="checkbox"/> K209M

parameter	Input/Output	data type	Allowed memory areas	description
EXEC	input	BOOL	M、V、L、SM	If an ascending edge of EXEC is detected, the order is triggered for execution.
AXIS	input	INT	V、M、L、 constant	The axis number of the target axis (that is, the address of the CANOpen slave).
ID	input	BYTE	V、M、L	The starting address of the ordinal table of parameters to modify.
BEST	output	DWORD	V、M、L	Reads the storage start address of all modified parameter values.
A	input	INT	V、M、L、 constant	The number of parameters to modify
DONE	output	BOOL	M、V、L	Completion flag bit. When the instruction execution is complete, <i>DONE</i> jumps from 0 to 1.
ERR	output	BOOL	M、V、L	Error flag bit. If an error occurs while the instruction is executing, it is set to 1.
ERRID	output	DWORD	V、M、L	Error code



AXIS and NUM must be both constant types and memory types, and the ID and PAPAS parameters together form a variable-length memory block that must be All are in a legitimate memory region, otherwise the results are unpredictable.

The three parameters of ID, PARAS, and NUM together form a parameter table. Where ID is the starting address of the sequence number table, from this address onwards, the ordinal number of each parameter to be operated is stored sequentially (that is, the "ordinal number" in the preceding parameter list), each ordinal number occupies 1 byte; PARAS is the starting address of the parameter value table, from this address onwards, the values of each parameter are stored in turn. Each numeric value occupies 4 bytes; NUM is the number of parameters to be operated. For example, in the following example, suppose that the ID parameter is VB200 and the PARAS parameter is VD2000. The NUM parameter is 3, then VB200, VB201, VB202 stores the serial numbers of the three parameters to be operated this time, VD2000 and VD2004, VD2008 stores the parameter values to be modified.

For PARAS, it should be noted that although the parameter value table uniformly adopts the DWORD address, the actual data type of each process parameter is not the same, so the user in the user program in the table should assign the corresponding address in the parameter table according to the actual data type.

- ✧ If the actual process data type is REAL, then the parameter memory can be operated directly with the floating-point number address. For example, if the parameter value is expected to be stored in VD2000, vr2000 can be operated directly. VD2000 and VR2000 actually occupy the same memory address in the PLC, and the instruction automatically does type conversion.
- ✧ If the actual process data type is a data type other than REAL, then the parameter memory can be directly operated, and the instruction will automatically do type conversion according to the data type of the parameter. For example, in this example, the data types of parameters 3 and 8 to be

manipulated are UINT8 and UINT16, so you can directly assign a valid value to VD2004 and VD2008.

1) Instruction description in LD format

If EN is 1, then on the rising edge of the EXEC input, the instruction is triggered to execute, and the instruction will be executed according to the parameter table specified by ID, PARAS, NUM, etc. The values in paras are sent to the drive via SDO to modify the corresponding object, while the ERRID corresponding bit is set to 0. If the SDO response for a parameter is incorrect or the timeout is not responding, set the ERRID bit to 1 and continue writing to the next parameter. When all the parameters are written, DONE is set to 1, and ERR and ERRID are set to different values according to the execution result.

If EN is 0, the instruction is not executed. If EN changes to 0 during instruction execution, the instruction will stop writing the unfinished parameters, set the DONE to 1, and ERR and ERRID maintain the executed result.

If the PLC detects an error when the instruction starts to execute (such as the axis is not enabled, the axis is executing other instructions, etc.), it directly exits, sets the DONE, ERR to 1, and ERRID to the corresponding error code.

➤ example

This example is in IL format. In Kincobuilder, select the IL format in the Project menu, then copy and paste the example into the editor, and then select LD Format, and the program can be displayed as LD format.

(* Network 0 *)

(*Set the parameter table, indicating that you want to write parameters 0, 3, and 8.*)

```
LD      %SM0.0
MOVE    B#0, %VB200
MOVE    B#3, %VB201
MOVE    B#8, %VB202
```

(* Network 1 *)

(*Set the numeric value to be written for each parameter.) Note the data type with each parameter automatically occupies a 32-bit address. This example is equivalent to writing 1200.0 to parameter 0 in the table, 8 to parameter 3, and 2000* to parameter 8 in the table

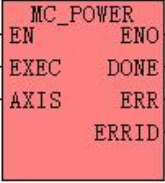
```
LD      %SM0.0
MOVE    1200.0, %VR2000
MOVE    DI#8, %VD2004
MOVE    DI#2000, %VD2008
```

(* Network 2 *)

(*Invoke Instruction*).

```
LD      %SM0.0
MC_WPARAS %M1.1, 1, %VB200, %VD2000, 8, %M1.2, %M1.3, %MD14
```

10.5.8.3. 2.2 MC_POWER (lock and loose shaft).

	name	Directive format	Applies
LD	MC_POWER R	 <pre> MC_POWER EN ENO EXEC DONE AXIS ERR ERRID </pre>	<input checked="" type="checkbox"/> KS <input checked="" type="checkbox"/> KW103 <input checked="" type="checkbox"/> KW203 <input checked="" type="checkbox"/> K209M

parameter	Input/Output	data type	Allowed memory areas	description
EXEC	input	BOOL	M、V、L、SM	The rising edge triggers the lock axis command, and the falling edge triggers the loose axis command.
AXIS	input	INT	V、M、L、 constant	The axis number of the target axis (that is, the address of the CANOpen slave).
DONE	output	BOOL	M、V、L	Completion flag bit. When the instruction execution is complete, done jumps from 0 to 1.
ERR	output	BOOL	M、V、L	Error flag bit. If an error occurs while the instruction is executing, it is set to 1.
ERRID	output	BYTE	V、M、L	Error code

1) Instruction description in LD format


If EN is 1, the execution of the axis lock command is triggered on the rising edge of EXEC and the loose axis command is triggered on the falling edge of EXEC.

When the instruction is executed, the PLC first sends the command control axis into the pending operation state, and checks the actual return state of the drive within the 5S timeout period, if the instruction is executed successfully, the DOL is set to 1, the ERR is set to 0, ERRID is set to 0. If an error occurs (it may be an error in the execution of the instruction itself, or it may be an error in which the drive did not perform the action correctly during execution, see the error code for details), the instruction execution fails, the instruction will stop executing, and the DONE is set to 1 and ERR to 1, ERRID is assigned the corresponding error code.

If EN is 0, the instruction is not executed.

10.5.8.3. 2.3 MC_RESET (reset drive alarm).

	name	Directive format	Applies
--	------	------------------	---------

LD	MC_RESET		<input checked="" type="checkbox"/> KS <input checked="" type="checkbox"/> KW103 <input checked="" type="checkbox"/> KW203 <input checked="" type="checkbox"/> K209M
----	----------	---	---

parameter	Input/Output	data type	Allowed memory areas	description
EXEC	input	BOOL	M、 V、 L、 SM	The rising edge triggers this order execution once.
AXIS	input	INT	V、 M、 L、 constant	The axis number of the target axis (that is, the address of the CANOpen slave).
DONE	output	BOOL	M、 V、 L	Completion flag bit. When the instruction execution is complete, <i>DONE</i> jumps from 0 to 1.
ERR	output	BOOL	M、 V、 L	Error flag bit. If an error occurs while the instruction is executing, it is set to 1.
ERRID	output	BYTE	V、 M、 L	Error code

When an error occurs in the operation of the shaft, this command can be invoked to reset the error message on the shaft and place the shaft in a resting state of the loose shaft. **If you need to continue executing other motion commands after the reset is successful, you should first call the MC_POWER instruction to lock the axis!**

Note: This instruction only resets the alarm error message of the driver, and does not reset the output of each command!

1) Instruction description in LD format

If EN is 1, then the execution of this order will be triggered on the rising edge of EXEC.

When the instruction is executed, the PLC first sends the instruction reset drive alarm, and checks the actual state of the drive within a 2-second timeout period, if the successful reset, it means that the instruction execution is successful, then THE Done is set to 1, and the ERR is set to 0 ERRID is set to 0. If an error occurs (it may be an error in the execution of the instruction itself, or it may be an error in which the drive did not perform the action correctly during execution, see the error code for details), the instruction execution fails, the instruction will stop executing, and the DONE is set to 1 and ERR to 1 , ERRID is assigned the corresponding error code.

If EN is 0, the instruction is not executed.

10.5.8.3. 2.4 MC_HOME (back to square one).

	name	Directive format	Applies
LD	MC_HOME	<pre> MC_HOME - EN ENO - EXEC DONE - AXIS ERR - POS ERRID - TIME </pre>	<input checked="" type="checkbox"/> KS <input checked="" type="checkbox"/> KW103 <input checked="" type="checkbox"/> KW203 <input checked="" type="checkbox"/> K209M

parameter	Input/Output	data type	Allowed memory areas	description
EXEC	input	BOOL	M、 V、 L、 SM	The rising edge triggers the execution of this order once; the falling edge triggers the pause motion
AXIS	input	INT	V、 M、 L、 constant	The axis number of the target axis (that is, the address of the CANOpen slave).
POS	input	REAL	V、 M、 L constant	Offset position of the origin in mm or °.
TIME	input	DWORD	V、 M、 L、 constant	Timeout period, if the origin is not found within this time, the error is reported to exit.
DONE	output	BOOL	M、 V、 L	Completion flag bit. When the instruction execution is complete, <i>DONE</i> jumps from 0 to 1.
ERR	output	BOOL	M、 V、 L	Error flag bit. If an error occurs while the instruction is executing, it is set to 1.
ERRID	output	BYTE	V、 M、 L	Error code

Execution of this instruction allows the target axis to return to the origin. *The POS* parameter sets the offset value of the origin coordinates.

Note: This instruction uses the drive internal back to the origin mode, you need to first set the drive 60980008 back to the origin mode and the speed back to the origin and other related parameters (can also be written through the MC_WPARAS instructions), for details, please refer to the driver manual.

1) Instruction description in LD format

If EN is 1, then the execution of this order will be triggered on the rising edge of EXEC.

When the instruction is executed, the PLC first sends a command to let the axis start to find the origin, and after the transmission is complete, check the status returned by the drive. The check will last *time* (user-set timeout, in ms), if the axis successfully finds the origin within this timeframe, it means that the instruction execution was successful, at which point the DONE is set to 1 and the ERR is set 0, ERRID is set to 0. If an error occurs (it may be an error in the execution of the instruction itself, or it may be an error in which the drive does not perform the action correctly during execution, see the error code) the instruction execution fails, the instruction will stop executing, and the DONE is set to 1 and THE ERR is set to 1 , ERRID is assigned

the corresponding error code.

If EN is 0, the instruction is not executed. If EN becomes 0 during execution, the command will stop executing and the axis will be in a static locking axis waiting state.

10.5.8.3. 2.5 MC_MABS (absolute motion).

	name	Directive format	Applies
LD	MC_MABS	<pre> MC_MABS EN ENO EXEC DONE AXIS ERR POS ERRID VEL ACT DIR MODE </pre>	<input checked="" type="checkbox"/> KS <input checked="" type="checkbox"/> KW103 <input checked="" type="checkbox"/> KW203 <input checked="" type="checkbox"/> K209M

parameter	Input/Output	data type	Allowed memory areas	description
EXEC	input	BOOL	M、V、L、SM	The rising edge triggers the execution of this order once; the falling edge triggers the pause motion
AXIS	input	INT	V、M、L、constant	The axis number of the target axis (that is, the address of the CANOpen slave).
POS	input	REAL	V、M、L、constant	Absolute target position in mm or °
WELL	input	REAL	V、M、L、constant	The maximum speed (>0) increased during motion in mm/min or °/min.
YOU	input	INT	V、M、L、constant	Direction of motion. Reserved, the function has not been implemented, and it can be maintained as 0.
MODE	input	INT	V、M、L、constant	Sport mode: single execution or permanent execution. 0 indicates a single execution, and the instruction exits after the axis executes this absolute positioning. 1 indicates permanent execution, when the axis performs an absolute positioning, the instruction does not exit, if a new target position is found, it will send a command to let the axis continue to perform a new absolute positioning.
DONE	output	BOOL	M、V、L	Completion flag bit. When the instruction execution is complete, <i>DONE</i> jumps from 0 to 1.
ERR	output	BOOL	M、V、L	Error flag bit. If an error occurs while the instruction is executing, it is set to 1.
ERRID	output	BYTE	V、M、L	Error code
ACT	output	BOOL	M、V、L	MODE=0, when executed in a single shot, the ACT

				indicates whether the single positioning instruction is correctly activated. 1 means active and 0 means inactive. MODE=1, when executed permanently, the ACT indicates whether the permanent positioning instruction is correctly activated. 1 indicates activation (it will always be at 1 when a single positioning is completed), and 0 means inactivation.
--	--	--	--	--

This directive controls the movement of the target axis to the target position (absolute position). When moving, the velocity starts at the current value of the VEL and reaches the target position at zero. This Directive permits suspension.

Note: This instruction uses the internal acceleration and deceleration of the drive, you need to set the 60830020 acceleration and deceleration and other related parameters in the driver first (can also be written through the MC_WPARAS instructions), please refer to the driver manual for details.

1) Instruction description in LD format

If EN is 1, then the execution of this order will be triggered on the rising edge of EXEC.

When the instruction is executed, the PLC control axis starts to position absolutely according to the user-entered target position (POS) and motion speed (VEL) parameter values. During the motion, the command will continuously scan the target position and the target speed parameter values, and if there is a change, it will be sent to the axis immediately, that is, the new speed parameter and position parameter values can be accepted at any time (for example, to perform a pause, pause by setting the speed to 0 during the movement, and resuming the motion by re-giving the speed value). At the same time, the PLC will continue to check the return status of the axis, if the target position of this positioning is successfully reached, indicating that the positioning is completed, then the DONE is placed 1, the ERR is placed 0, ERRID is set to 0. After the completion of this positioning, the instruction will judge the mode (MODE) value, if set to a single operation mode, then the instruction directly exits; if it is set to a permanent operation mode, then the instruction does not exit, scan the target position value at any time, if the target position changes, it will be sent to the axis, so that the axis can carry out a new absolute positioning.

If an error occurs (it may be an error in the execution of the instruction itself, or it may be an error in which the drive did not perform the action correctly during execution, see the error code for details), the instruction execution fails, the instruction will stop executing, and the DONE is set to 1, ERR to 1, ERRID Assign the value to the appropriate error code.

If EN is 0, the instruction is not executed. If EN becomes 0 during execution, the command will stop executing and the axis will be in a static locking axis waiting state.

10.5.8. 3. 2.6 MC_MREL (relative motion).

	name	Directive format	Applies
LD	MC_MREL	<pre> MC_MREL - EN ENO - EXEC DONE - AXIS ERR - POS ERRID - VEL ACT </pre>	<input checked="" type="checkbox"/> KS <input checked="" type="checkbox"/> KW103 <input checked="" type="checkbox"/> KW203 <input checked="" type="checkbox"/> K209M

parameter	Input/Output	data type	Allowed memory areas	description
EXEC	input	BOOL	M、V、L、SM	The rising edge triggers the execution of this order once; the falling edge triggers the pause motion
AXIS	input	INT	V、M、L、constant	The axis number of the target axis (that is, the address of the CANOpen slave).
POS	input	REAL	V、M、L、constant	The relative distance to be moved in mm or °. Positive numbers indicate positive motion; negative numbers indicate motion in the negative direction.
WELL	input	REAL	V、M、L、constant	The maximum speed (>0) increased during motion in mm/min or °/min.
DONE	output	BOOL	M、V、L	Completion flag bit. When the instruction execution is complete, <i>DONE</i> jumps from 0 to 1.
ERR	output	BOOL	M、V、L	Error flag bit. If an error occurs while the instruction is executing, it is set to 1.
ERRID	output	BYTE	V、M、L	Error code
ACT	output	BOOL	M、V、L	Whether the directive was activated correctly. 1 means active and 0 means inactive.

This directive controls the target axis to move the specified distance *POS* (using the current position as a reference, that is, using the current position as the starting position). When moving, the velocity starts at the current value of the *VEL* and reaches the target position at zero. This Directive permits suspension.

Note: This instruction uses the internal acceleration and deceleration of the drive, you need to set the 60830020 acceleration and deceleration and other related parameters in the driver first (can also be written through the MC_WPARAS instructions), please refer to the driver manual for details.

1) Instruction description in LD format

If *EN* is 1, then the execution of this order will be triggered on the rising edge of *EXEC*.

When the instruction is executed, the PLC control axis starts to position relatively (with the current position as a reference) according to the user-entered target position (*POS*) and speed of motion (*VEL*) parameter values. During the motion, the command will continuously scan the target speed parameter value, and if there is a change, it will be sent to the axis immediately, that is, the new speed parameter value can be accepted at any time (for example, to perform a pause, set the speed to 0 during the motion, and then re-give the speed value to resume the motion). At the same time, the PLC will constantly check the return status of the check axis, if the target position of the current positioning is successfully reached, indicating that the positioning is completed, then *THE DOL* is set to 1, and the *ERR* is set to 0, *ERRID* is set to 0. If an error occurs (it may be an error in the execution of the instruction itself, or it may be an error in which the drive did not perform the action correctly during execution, see the error code for details), the instruction execution fails, the instruction will stop executing, and the *DOONE* is set to 1 and *ERR* to 1, *ERRID* is assigned the corresponding error code.

If *EN* is 0, the instruction is not executed. If *EN* becomes 0 during execution, the command will stop

executing and the axis will be in a static locking axis waiting state.

10.5.8.3. 2.7 MC_JOG (jog).

	name	Directive format	Applies
LD	MC_JOG	<pre> MC_JOG EN ENO EXEC DONE AXIS ERR VEL ERRID DIR ACT </pre>	<input checked="" type="checkbox"/> KS <input checked="" type="checkbox"/> KW103 <input checked="" type="checkbox"/> KW203 <input checked="" type="checkbox"/> K209M

parameter	Input/Output	data type	Allowed memory areas	description
EXEC	input	BOOL	M、V、L、SM	The rising edge triggers the execution of this order once; the falling edge triggers the pause motion
AXIS	input	INT	V、M、L、constant	The axis number of the target axis (that is, the address of the CANOpen slave).
VEL	input	REAL	V、M、L、constant	Speed of motion in mm/min or °/min. Positive numbers indicate a positive direction and negative numbers indicate a negative direction.
YOU	input	INT	V、M、L、constant	Direction of motion. Reserved, the function has not been implemented, and it can be maintained as 0.
DONE	output	BOOL	M、V、L	Completion flag bit. When the instruction execution is complete, <i>DONE</i> jumps from 0 to 1.
ERR	output	BOOL	M、V、L	Error flag bit. If an error occurs while the instruction is executing, it is set to 1.
ERRID	output	BYTE	V、M、L	Error code
ACT	output	BOOL	M、V、L	Whether the directive was activated correctly. 1 means active and 0 means inactive.

This directive controls that the target axis runs at the target speed specified by Vel.

Note: This instruction uses the internal acceleration and deceleration of the drive, you need to set the 60830020 acceleration and deceleration and other related parameters in the driver first (can also be written through the MC_WPARAS instructions), please refer to the driver manual for details.

1) Instruction description in LD format

If *EN* is 1, then the execution of this order will be triggered on the rising edge of EXEC.

When the instruction is executed, the PLC control axis starts to jog according to the user input motion speed (*VEL*) parameter value. During axis motion, the command will continuously scan the target speed parameter value, and if there is a change, it will be immediately sent to the axis, that is, the new speed parameter value can be accepted at any time.

If an error occurs (it may be an error in the execution of the instruction itself, or it may be an error in which

the drive did not perform the action correctly during execution, see the error code for details), the instruction execution fails, the instruction will stop executing, and the DOONE is set to 1, ERR to 1, ERRID Assign the value to the appropriate error code.

If *EN* is 0, the instruction is not executed. If EXEC *becomes* 0 during execution, the instruction will stop executing and the axis will be in a static locking state.

10.5.8.3. 2.8 MC_STATE (reading the values of each status of the drive).

	name	Directive format	Applies
LD	MC_STATE	<pre> MC_STATE EN ENO - AXIS POS HOME CW CCW RUN FAULT INPUT LIMIT ERRCODE APOS AVEL ONLINE </pre>	<input checked="" type="checkbox"/> KS <input checked="" type="checkbox"/> KW103 <input checked="" type="checkbox"/> KW203 <input checked="" type="checkbox"/> K209M

parameter	Input/Output	data type	Allowed memory areas	description
AXIS	input	INT	V、M、L、constant	The axis number of the target axis (that is, the address of the CANOpen slave).
POS	output	BOOL	V、M、L	"Location to" signal
HOME	output	BOOL	V、M、L	"Origin found" signal
CW	output	BOOL	M、V、L	"Motor forward" signal
CCW	output	BOOL	M、V、L	Motor inversion signal
RUN	output	BOOL	M、V、L	"Motor in operation" flag
FAULT	output	BOOL	M、V、L	Axis Alarm flag
INPUT	output	WORD	V、M、L	The status of the switch input port of the shaft, BIT0 corresponds to the DIN1 of the shaft, and is stored in order, and the specific number of input ports is queried in the driver manual
LIMIT	output	BOOL	M、V、L	Limit to flag
ERRCODE	output	WORD	V、M、L	Alarm error code for the shaft
apos	output	REAL	M、V、L	The current actual position of the machine, mm or °.
AVEL	output	REAL	M、V、L	The current actual speed of the machine, mm/min or °/min.
ONLINE	output	BYTE	M、V、L	"Axis Online" logo. 1 means that the axis is not online, and 0 means that the axis is online.

This instruction scans the drive status all the way, gets flags for various different states and outputs them to the corresponding output parameters. **However, there may be a certain delay, and the user**


should be cautious when using each output as the basis for judging the action

Note: The signals "Position to" and "Origin Found" will change to 0 again during the execution of the action (locate or find the origin), and will not reset 1 until the action is performed correctly!

1) Instruction description in LD format

If *EN* is 1, this directive is executed. If *EN* is 0, the instruction is not executed and the various output parameters are not refreshed.

10.5.8.3. 2. 9 MC_RESTRAT (Reconfigure and Boot Slave).

	name	Directive format	Applies
LD	MC_RESTRAT		<input checked="" type="checkbox"/> KS <input checked="" type="checkbox"/> KW103 <input checked="" type="checkbox"/> KW203 <input checked="" type="checkbox"/> K209M

parameter	Input/Output	data type	Allowed memory areas	description
EXEC	input	BOOL	I•Q•M•M•L•SM	The rising edge of EXEC initiates this instruction, performing a reboot of the slave. Exec's falling edge stops the execution of this instruction and terminates the process of restarting the slave.
AXIS	input	INT	V、M、L、constant	The axis number of the target axis (that is, the address of the CANOpen slave).
ESTIMATES	input	INT	V、M、L、constant	The maximum time that a master waits for a slave to respond after sending an SDO request packet.
DONE	output	BOOL	Q、M、V、L	Instruction execution result flag bit. If the instruction is executing, the DONE output is 0. If the instruction execution ends (either successful or unsuccessful), done immediately becomes 1.
ERR	output	BOOL	Q、M、V、L	Instruction execution error flag bit. If an error occurs while the instruction is executing, it is placed 1.
ERRID	output	BYTE	V、M、L	The error code of the instruction execution. If an error occurs while the instruction is executing, ERRID is the specific error message.

Note: AXIS and STIME parameters must be both constants and variables at the same time. Up to 32 MC_Restart directives are allowed in a user project.

A description of the Fault Code (ERRID) is shown in the table below:

Error code	illustrate
0	inerrancy
1	The target axis number is incorrect

2	The target axis is performing the process of restarting the slave, which prohibits further execution until it is complete.
3	The master sends a CAN packet error (which may be due to a line, hardware interface, software buffer full, etc.).
4	The SDO request message sent by the master did not respond to the target axis timeout.
5	The SDO request packet sent by the master, the target axis responds incorrectly.
6	The user actively stops the execution of this instruction (set EXEC to 0).

1) Instruction description in LD format

If *EN* is 1, this directive is executed. If *EN* is 0, the instruction is not executed and the various output parameters are not refreshed.

During normal operation of the master, the user can invoke this directive to reconfigure and start the target axis (station number *AXIS*). Each parameter of the axis will use the parameters that have been configured for the slave in [Kinco Motion Control Network Configuration], including error supervision parameters and so on.

If *EN* is 1, then the rising edge of *EXEC* will trigger the execution of this instruction, first the *DONET* output is 0, and then the master will read the configured axis parameters and perform the following operations on the target axis:

- 1) Send a command to put the target axis into the pre-operation state .
- 2) Send SDO to configure the node protection parameters of the target axis.
- 3) Send SDO to configure the mapping parameters of all PDOs of the target axis.
- 4) Send SDO to configure the communication parameters of all PDOs of the target axis.
- 5) Send the "Start Node" command to the target axis.

When the above process is successfully executed, this instruction will exit and immediately output *DONE* to 1, and *ERR* and *ERRID* will both output 0.

If any error occurs during execution of the instruction, or IF *EXEC* becomes 0, the instruction also exits and the *DONE*, *ERR* is immediately output as 1, and *errid* outputs the corresponding error code.

10.5.8.3. 2. 10 MC_MIoT (Read Device Information).

	name	Directive format	Applies
--	------	------------------	---------

LD	MC_MIOT	<div style="border: 1px solid black; background-color: #f0f0f0; padding: 5px; display: inline-block;"> MC_MIOT - EN ENO - AXIS RES - EXECV VER - EXECS VLEN - TIMES SN SLEN DATAS DATAP </div>	<input checked="" type="checkbox"/> KS <input checked="" type="checkbox"/> K209M
----	---------	--	---

parameter	Input/Output	data type	Allowed memory areas	description
AXIS	input	INT	V、M、L、constant	The axis number of the target axis (that is, the address of the CANOpen slave).
EXECV	input	BOOL	M、V、L、SM	The rising edge triggers a read of the serial number, software version
EXECS	input	BOOL	M、V、L、SM	The rising edge triggers a read of the IIT, temperature, runtime
TIMES	input	INT	V、M、L	Timer, the timing time is up to read the IIT, temperature, running time once. If 0, the timer does not start.
NOTHING	output	BYTE	M、V、L	Execution results.
SEE	output	BYTE	M、V、L	The address from which the software version information is stored
WORTHIT	output	BYTE	M、V、L	The total length of the software version information, in bytes
SN	output	BYTE	M、V、L	The address from which the serial number information is stored
SLEN	output	BYTE	M、V、L	The total length of the serial number information, in bytes
DATES	output	BYTE	M、V、L	The location from which IIT, temperature, and runtime information are stored
DATAP	output	BYTE	M、V、L	The starting address where information such as status words, currents, and speeds are stored

This directive is used to read the product of the target axis, the operating status and other information. Only 1 per axis is allowed to use this directive. **This directive is limited to step servo as a slave shaft and does not support third-party products**

For the same axis, MC_MIOT instructions have the lowest priority: if other motion instructions are running, the MC_MIOT will not be executed; if other motion instructions are initiated, the instructions MC_MIOT Will be interrupted and terminated directly.

The information read by this directive is divided into 3 categories, which will be described in detail below.

1) Serial number, software version information

This information is triggered by the rising edge of the *EXECV* parameter to read this information once. These are fixed messages that can generally be read once at power-up.

The *VER* parameter specifies the starting address where software version information is stored, and the software version information is continuously stored in the area where the address begins. The *VLEN* parameter value indicates the total length of the software version information, that is, the number of bytes consumed.

The *SN* parameter specifies the starting address where the product serial number information is stored, and the serial number information is continuously stored in the area where the address begins. The *SLEN* parameter value indicates the total length of the serial number information, that is, the number of bytes occupied.

After each trigger, the PLC performs a read process, if all the readings are successful, the data and length information in the output parameters are updated, and if the reading fails, the output parameters are not updated. Regardless of success or failure, when the instruction completes, the corresponding bits in the *RES* are refreshed.

Bits in RES	description
Bit 7	Indicates whether the read completes this set of parameters. 0 indicates that the read is in progress; 1 indicates that the read is complete (whether successful or unsuccessful).
Bit 6	Indicates whether an error occurred in reading this set of parameters. 0 indicates a successful read; 1 indicates an error during the read.

2) IIT, drive temperature, runtime information

These are running information that needs to be read in real time, but not too frequently, otherwise it may affect other movements.

There are two trigger conditions for reading these parameters: the rising edge of the *EXECS* parameter triggers a read, and the *times* specify a timed read period, and the PLC triggers a read every other time. If the *TIMES* parameter value is 0, the timed reading is stopped.

The *DATAS* parameter specifies the starting address of the storage of this information, and each parameter information is stored in the following table:

Parameter name	object	The data type	length	The byte offset in the hold area
IIT	0x2FF010	UINT8	1	0
Drive temperature	0x60F70B	UINT16	2	1
Runtime	0x2FF700	UINT32	4	3

After each trigger, the PLC performs a read process, if all reads are successful, the data information in the output parameters is updated, and if the reading fails, the output parameters are not updated. Regardless of success or failure, when the instruction completes, the corresponding bits in the *RES* are refreshed.

Bits in RES	description
Bit 5	Indicates whether the reading of this set of parameters is complete.

	0 indicates that the read is in progress; 1 indicates that the read is complete (whether successful or unsuccessful).
Bit 4	Indicates whether an error occurred in reading this set of parameters. 0 indicates a successful read; 1 indicates an error during the read.

3) Status word, error word, actual current and other information

This information is automatically read by the instruction via the PDO without the user having to trigger it in the program.

The *DATAP* parameter specifies the starting address where this information is stored, and each parameter information is stored in the following table:

Parameter name	object	The data type	length	The byte offset in the hold area
Status word	0x604100	UINT16	2	0
Error word	0x260100	UINT16	2	2
Error word 1	0x260200	UINT16	2	4
Actual current	0x607800	INT16	2	6
Actual speed	0x606C00	INT32	4	8
Actual location	0x606300	INT32	4	12

4) Execution result parameter: *RES*

Bits in RES	description
Bit 7	Indicates whether to read the finished software version, serial number parameters. 0 indicates that the read is in progress; 1 indicates that the read is complete (whether successful or unsuccessful).
Bit 6	Indicates whether an error occurred in the reading software version and serial number parameters. 0 indicates a successful read; 1 indicates an error during the read.
Bit 5	Indicates whether the reading of THEIT, temperature, actual parameters is complete. 0 indicates that the read is in progress; 1 indicates that the read is complete (whether successful or unsuccessful).
Bit 4	Indicates whether the IIT of the reading group, the temperature, and whether an error actually occurred. 0 indicates a successful read; 1 indicates an error during the read.
Bit 3... 0	The combined values indicate an execution error: 0 --- indicates no errors 1 --- indicates that the target axis number is incorrect 2 --- indicates that other motion instructions are being executed and cannot be run.

5) Instruction description in LD format

If EN is 1, the corresponding device information is triggered according to the EXECV, EXECS, and TIMES parameter conditions.

If EN is 0, the instruction is not executed. If EN becomes 0 during execution, the instruction will stop executing.