CONTENT

Chapter 1 General introduction....6

1.1 Production introduction....6

1.2 Porduct list....6

1.3 Normal working conditions....6

Chapter 2 Usage of LPWAN....7

2.1 Generic wireless technology words....8

2.2 Usage of LoRa communication interface....8

2.2.1 Configure LoRa parameters....9

2.2.1.1 Use parameters to configure tools....9

2.2.1.2 Use parameters read/write instruction....11

2.2.2 Usage of LoRa....11

2.2.2.1 Build LoRa communication net....11

2.2.2.2 Usage of LoRa....11

2.2.2.2.1 Programming protocol....12

2.2.2.2.2 Kinco PLC interconnection protocol & interlink wizard....12

2.2.2.2.3 Modbus RTU protocol....15

2.2.2.2.4 Free communication function....16

2.3 LoRa function related instructions....17

2.3.1 LoRa Port Dedicated Instructions....17

2.3.1.1 LORA_RPARAS (read LoRa parameter)....17

2.3.1.2 LORA_WPARAS（modify LoRa parameter）....19

2.3.1.3 LORA_STATUS（Get LoRa signal quality）....21

2.3.1.4 Automatically reset LoRa communication port....21

2.3.2 Kinco Interconnection protocol specific instruction....23

2.3.2.1 SPS_SLV_OP（The master station pauses or restarts communication with the slave station）.23

2.3.2.2 SPS_MSLAVE（Read the communication status of the slave）....24

2.3.3 Modbus RTU Master Instruction....25

2.3.3.1 MBUSR (Read Slave Data) Instruction....25

2.3.3.2 MBUSW (write data to slave) instruction....27

2.3.4 Free communication..... *28*

2.3.4.1 COM_XMT (send datas)..... 29

2.3.4.2 COM_RECV (receive datas)..... 30

2.3.5 Reset communication port..... *31*

2.3.5.1 COM_RESET (send data)..... 31

Chapter 3 Standard CPU module..... 33

3.1 structure appearance..... 33

3.2 Technical Parameters..... 33

3.3 Functions..... 34

3.3.1 CPU Status and LEDs..... *34*

3.3.2 I/O function..... *34*

3.3.3 Serial port..... *35*

3.3.4 Can bus and extended bus port..... *36*

3.3.5 USB Port..... *36*

3.3.6 Data Retentive and Data Backup..... *38*

3.3.7 Real-time Clock (RTC)..... *38*

3.3.8 Backup Battery..... *38*

3.4 Wiring diagram..... 39

3.5 Dimension..... 40

3.6 Technical Specification of IO channel..... 40

Chapter 4 Simple CPU module..... 41

4.1 Technical Parameter..... 41

4.2 Function..... 42

4.2.1 CPU Status and LEDs..... *42*

4.2.2 I/O channel..... *43*

4.2.3 Serial port..... *43*

4.2.4 USB port..... *44*

4.2.5 Data Backup..... *44*

4.2.6 Switch..... *44*

4.3 Wiring diagram....44
4.4 Other....47
Chapter 5 The usage of high speed counter....47
5.1 High Speed Counter Work Modes and Input Signal....48
5.2 Control Register and Status Register....49
5.3 Preset Value(PV) setting....51
5.4 “CV=PV” Interrupt Numbers....52
5.5 The Methods to use the High Speed Counter....53
Chapter 6 How to Use the Functions of High Pulse Output....55
6.1 Motor Direction Control Signal....55
6.2 Positioning Control Instruction....55
6.2.1 Positioning Control Model....55
6.2.2 Control Register and Status Register....56
6.2.3 Error Code....58
6.2.4 PHOME（Homing）....58
6.2.5 PABS（Absolute Motion）....61
6.2.6 PREL（Relative Motion ）....62
6.2.7 PJOG（Inching）....64
6.2.8 PSTOP（Stop）....65
6.3 PLS Instruction....66
6.3.1 PWM & PTO....67
6.3.2 PTO/PWM register....68
6.3.3 Using PTO Function....69
6.3.4 Using PWM function....70
Chapter 7 CAN bus communication port....71
7.1 Interface....71
7.2 Extended bus....72
7.2.1 How to use EX_ADDR instruction to extension module....72
7.3 Kinco Motion control....73

7.3.1 Kinco Motion control network configuration ... *73*
7.4 CANOpen Master ... 75
7.4.1 CANOpen Introduction ... *76*
7.4.1.1Network Management Tool（NMT） ... 76
7.4.1.1.1 NMT Node Control ... 76
7.4.1.1.2 NMT Error Control ... 76
7.4.1.2 SDO (Service Data Object) ... 77
7.4.1.3 PDO(Process Data Object） ... 77
7.4.2 CANOpen Master Feature ... *78*
7.4.2.1 CANOpen Network Management Tool ... 78
7.4.2.2 EDS file ... 79
7.4.2.3 CANOpen Network configuration process ... 79
7.5 CAN Free communication function ... 83
7.6 CANBus related instruction ... 83
7.6.1 Kinco Motion control instruction ... *83*
7.6.1.1 Review ... 83
7.6.1.2 MC_RPARAS（Read parameter）and MC_WPARAS（Change parameters） ... 85
7.6.1.3 MC_POWER（Lock shaft and loose shaft） ... 91
7.6.1.4 MC_RESET（Reset drive alarm） ... 92
7.6.1.5 MC_HOME（Homing） ... 93
7.6.1.6 MC_MABS（PABS） ... 94
7.6.1.7 MC_MREL（PREL） ... 95
7.6.1.8 MC_JOG（JOG） ... 96
7.6.1.9 MC_STATE（Read the status of the drive） ... 97
7.6.1.10 MIOT_MC（Read Kinco servo driver information） ... 98
7.6.2 SDO Instruction ... *100*
7.6.2.1 SDO_WRITE ... 101
7.6.2.2 SDO_READ ... 102
7.6.3 CAN Free communication instruction ... *104*
7.6.3.1 CAN_INIT（Initialize the CAN interface） ... 104

7.6.3.2 CAN_TX（Send CAN message automatically）.....105
7.6.3.3 CAN_WRITE（Send CAN message once）.....106
7.6.3.4 CAN_RX（Receive specific ID CAN message）.....107
7.6.3.5 CAN_READ（Receive CAN message once）.....108
7.6.4 Extend Bus Instruction.....109
7.6.4.1 EX_ADDR（Modify the extension module configuration）.....110

Chapter 1 General introduction

1.1 Production introduction

KW series is wireless data collection & control products designed by Kinco.It mix up LPWAN communication technology and PLC technology. On the one hand, it's advantages such as user-programming, multiple IO modules, colorful functions, strong performance and high relieability are kept, which can easily finish tasks of data collection and terminal calculation. On the other hand, it also provides some wireless communication interfaces such as LoRa or NB-IoT and network protovols. The communication distance is long. The penetrate ability and anti-interference ability are strong, which solve the problem of construction, period and cost. So KW series is very suitable in big-range cover and big-scale connection factory IoT applications.

1.2 Porduct list

Name	Model	Description of functions
Standard CPU modules		
KW103	KW103-12DT-LoRa	DC24V power supply ，DI 8*DC24V，DIO 4*DC24V Wireless communication interface： 1*LoRa，working frequency range 410～493MHz Wired communication interface： 1*RS232，1*RS485，1*CAN Expansion：Support MAX 12 expansion modules. Poragmming port：MicroUSB
KW203	KW203-12DT-R2	DC24Vpower supply，DI 8*DC24V，DIO 4*DC24V Wireless communication interface： 1*LoRa，working frequency range 2400～2500MHz Wired communication interface： 1*RS232，1*RS485，1*CAN Expansion：Support MAX 12 expansion modules. Poragmming port：MicroUSB
Simple CPU module		
KW213	KW213-08DTX-R2	DC9～24V power supply，DIO 8*transistor Wireless communication interface： 1*LoRa，working frequency range 2400～2500MHz Wired communication interface：1*RS485 Expansion：No. Poragmming port：MicroUSB

1.3 Normal working conditions

Kinco-KW accords with GB/T 15969.3-2007（idt IEC61131-2：2007）standard and test specifications.

The following table lists the conditions and requirements for Kinco-KW to work properly. It is the user's responsibility to ensure that the service conditions are not exceeded.

Transport & Storage		
Ambient conditions	temperature	-40℃～+70℃ .
	Relative humidity	10%～95%，no condensation.
	Altitude	Up to 3000 m
Mechanical conditions	free falls	With manufacturer's original packaging, 5 falls from 1m of height
Normal operation		
Ambient condition	Air temperature	Open equipment : -10 ～ +55° C
	Relative humidity	10%～95%，no condensation.
	Altitude	Up to 2000 m
	Pollution degree	for use in pollution degree 2.
Mechanical conditions	Sinusoidal vibrations	5<f<8.4Hz, Occasional: 3.5mm amplitude; Continuous:1.75mm mplitude. 8.4<f<150, Occasional: 1.0g acceleration; Continuous:0.5g acceleration.
	shock	occasional excursions to 15g, 11 ms, half-sine, in each of 6 mutually perpendicular axes.
EMC	Electrostatic discharge	±4kV Contact, ±8kV Air. Performance criteria B.
	High energy surge	a.c. main power: 2KV CM, 1KV DM; d.c. main power: 0.5KV CM, 0.5KV DM; I/Os and Communication port: 1KVCM. Performance criteria A.
	Fast transient bursts	main power: 2KV, 5KHz. I/Os and Communication port: 1KV, 5KHz. Performance criteria A.
	Voltage drops and interruptions	a.c. supply: at 50Hz, 0% voltage for 1 period; 40% voltage for 10 periods; 75% voltage for 20 periods. Performance criteria A.
Ingress Protection Rating	Water & dust protection	IP20

Chapter 2 Usage of LPWAN

LoRa is one of LPWAN （Low Power Wide Area Network， lower power consumption WAN） communication technologies, which is a kind of narrowband wireless scheme based on spread spectrum technology and very long distance.Its advantages are wide cover range, better anti-interference and lower transmit power. Lora works in unlicensed band and it is flexible to deploy. Users can buld private net.

KW1 and KW2 series products provide different working frequenc band LoRa communication interface. This chaoter will introduce its functions and usage in detail.

2.1 Generic wireless technology words

- **dBm（dB milliwatt）**

dBm（dB milliwatt）is the value which stands for power absolute value. The standard is 1Mw.Its formula is 10*lg(P) and P is power（unit is mW）. For dBm，there are some direct but useful values:

- 1mW's power is 0dBm；1W's power is 30dBm.
- Add 3dBm means power multiply 2; Decrease 3dBm means power divide 2.
- Add 10dBm means power multiply 10; Decrease 10dBm means power divide 10.

- **dB（Decibel）**

dB is a abosulte value.

When users think about the difference (dB) between power of A and power of B, use this formula: 10*lg(A/B).

For example, power of A is twice larger than power of B, So 10*lg （power of A/power of B）=10*lg2=3dB.

- **Bandwidth（BW）**

Bandwidth is the difference between down frequency and upfrequency for allowable signals of channel.

LoRa's bandwidth is double-side bandwidth. It is supposed that center frequency of LoRa channel is f and bandwidth is BW. Then actual frequency range of this channel is $[f-\frac{BW}{2}, f+\frac{BW}{2}]$.

- **Coding rate（CR）**

LoRa use loop checking code to operate front checking and back checking.Under heavy interference, it can improve reliability of link but add redundancy information when coding.

Coding rate is the ratio of useful part in communication data. If coding rate is $\frac{k}{n}$, theneachKbit is useful. Encoder produces n bit data. n-k is redundant.

- **Receiver Sensitivity**

Receiver sensitivity is the smallest power of received right signal. When signal power is smaller than standard receiver sensitivity, receiver will not receive.

Receiver sensitivity is ability of weak signal received by testing receiver. The smaller the value is, the bigger receive ability is. For example, receiver sensitivity of A is -85dBm and B is -140dBm, the receive ability if B is far better than A and the cover range of B is bigger than A.

- **Link Budget**

The allowable and biggest transmit budget in communication link when communication is stable.

The link budget is the main way that access covering ability of wireless communication ability. In a specific environment, the bigger the link budget is, the bigger the covering range is.

2.2 Usage of LoRa communication interface

The working frequency band of KW1's LoRacommuncation port isdifferent from KW2 and occasions are different. But their operation methods are same, take KW2 for example.

LoRa communication interface supports programming protocol, Kinco interconnection protocol, Modbus RTU protocol (Master & slave) and free protocol. In programming software, it provides different usage wizards, parameter confirguration, net status monitoring and etc to as to make it easier for users. In addition, it also provides many communication instructions so as users can call these instructions to finish online operation for LoRa.

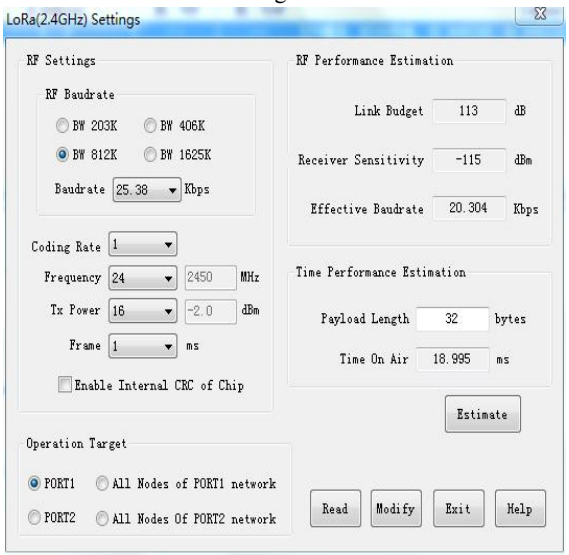
2.2.1 Configure LoRa parameters

KW provides two LoRa parameters configuration ways:

- Use KincoBuilder's parameters to configure tools;
- Use parameters to read/write instruction in user program;

2.2.1.1 Use parameters to configure tools

In KincoBuilder，operate【Tools】->【LoRa(2.4GHz）settings…】，users can make settings for LoRa interface.In addition, users can directly view estimated result of wireless communication when use different communication parameters so as to choose suitable settings.



➢ **Operate target**

Choose modified interface or device.

Different model's KW modules provide 1 or 2 LoRa ports and distribute different number.If module only has 1 LoRa port, the number of this port is 1. If this module has 2 LoRa ports, then users can find number of each port in silk screen.

- 【Local port 1】: The parameters of connected module's LoRa 1 port
- 【Local port 2】: The parameters of connected module's LoRa 2 port
- 【Port 1 net all nodes】: This option only support【Write】operation，the LoRa port parameters of all modules that can communicate in wireless net for LoRa port 1 of connected module in PC
- 【Port 2 net all nodes】: This option only support【Write】operation，the LoRa port parameters of all modules that can communicate in wireless net for LoRa port 2 of connected module in PC

When choose to modify "All net nodes" parameters, please note:

1） Modifying process will last for around 3 seconds.

2） Modifying parameters might influence on normal data communication. Please modify under safety!

3） If users modify all nodes'parameters in the net, it is not allowable for other nodes to send data in this net. Otherwise, modification might fail. And as LoRa is half-duplex communication way, users cannot judge if there are failed nodes if setting in software!

It is suggested to use master to operation this function in the net.

- **Communication parameters**

- 【RF baudrate】: is wireless data transmit rate，which is same as baudrate in wired communication. Air rate is bigger and data transmit rate is higher.But normally, communication distance is short.
  LoRa provides many multiple bands and each bandwidth provides lots of air transmit rate. In software, users firstly need to choose a bandwidth and software will automatically list out all air rates I this group. Then users can choose.
  Factory default value is 25.38Kbps
- 【Coding rate】: LoRa's coding rate. The bigger the value is, the lower the coding rate is. Default value is 1. The lower the coding rate is, the reliability of communiaiton is. But valid data communicate rate will be lower. Under heavy interference occasion, it is suggested to decrease coding rate (Choose bigger value)
- 【Frequency channel】: Center frequency of LoRa working. Default value is 2450MHz.
  2.4GHz is global free frequency band. WiFi, Bluetooth, Zigbee and some other devices use this frequency band. LoRa has strong immunity ability for these communications. If this occurs, it is suggested to adjust KW frequency channel so as to avoid used frequency band of other devices within scales.
- 【RF power】: LoRa's RF power. Default value is 16（-2dBm）.
- 【Message interval】: is the biggest time interval between two frame message. If KW module doesn't receive data any more in this time, it will deal with one complete message of received data. Default value is 1ms.
  The longer message interval is, the lower communication frequency arrives.
  In actual application, the transmission is not stable caused by long communication distance, many blocks, heavy interference and etc. Then KW will make wrong separation for message. If this occurs, it is suggested to increase message interval.
- 【Build-in CRC in enable chip】: LoRa chip build-in CRC function，If enable CRC，then transmitter LoRa will make CRC for data every time before sending and sned out after adding checking code to message. When receiver LoRa receive message, it will firstlt make CRC. If checking is right, messge will be saved. Otherwise, message will be taken away.
  Kinco interconnection protocol and Modbus protocol provide by KW all bring with CRC function. So defaut setting disables CRC inside the chip.

- **Performance access**

According to communication parameters chosed by users, software will automatically access potential wirelss communication performance and display results for reference.

- 【Link budget】: If use current communication parameters, LoRa will arrive link budget theoretically. The bigger the budget is, the wider the cover range is.
- 【Receiver sensitivity】: If use current communication parameters, the receiver sensitivity of LoRa. The smaller the sensitivity is, the stronger ablity the receving ability is. So the distance between transmitter and receiver can be longer.

- 【Valid baudrate】: If use current communication parameters, the rate of LoRa theroretically transmit user's valid data. Efficient rate is realted with air transmit rate and coding rate. The higher the air transmit rate is, the higher the efficient rate is; There are more redundant information in communication data if coding rate is low. So efficient rate is low.

  - **Time access**

  After choose communication parameters, user can input【baudrate distance】and single click【Time access】, this software will automatically calculate【air transmit time】 between transmitter and receiver.

2.2.1.2 Use parameters read/write instruction

KW provides LORA_RPARAS （read LoRa parameters） and LORA_WPARAS （Modify LoRa parameters） .In user project, users can call these instructions so as to online read or modify for LoRa parameters. The reference value in instructions all correspond to each reference value in LoRa setting tools.

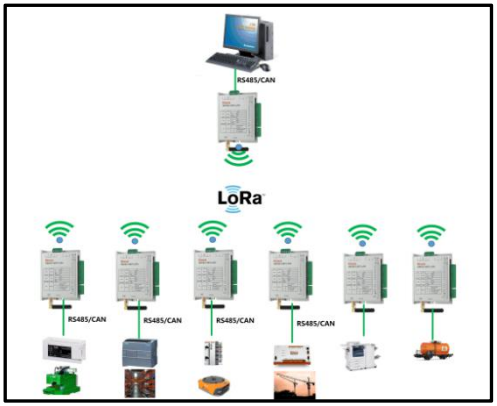
Please refer to 2.3 LoRa function related instructions.

2.2.2 Usage of LoRa

Before using LoRa communication, users need to configure LoRa port parameters. In same communication net, the parameters of all nodes below must be same: Frequency, air transmit rate, coding rate, if enable build-in CRC in chip.

2.2.2.1 Build LoRa communication net

LoRa is half duplex communication mode. Receive and transmit cannot operate at one time. In a LoRa net, only 1 node is under transmit status at one time. So it is suggested that users use LoRa to build a wireless communication net of master-slave or star topology.In net, there is only one master node and other nodes are slave; Master is used to call and manage net. It will check each slave in turn. Slave will respond only when it receive request from master. It is shown in figure.



In actual applications, users can divide different wireless net in modules in one area by setting different communication channels or air transmit baudrate. For example, if there are 100 pcs devices in a workshop, the communication frequency of 50 pcs device can be set to 2410MHz and another 50 pcs can be set to 2430MHz. Then these devices can build 2 separate wireless communication net.

2.2.2.2 Usage of LoRa

LoRa port support programming protocol（Master-slave mode）、Kinco PLC interconnection protocol（Master-slave mode）、Modbus RTU protocol（Master-slave mode）and free communication.

If there is no communication function in user program, the default is that KW will support programming

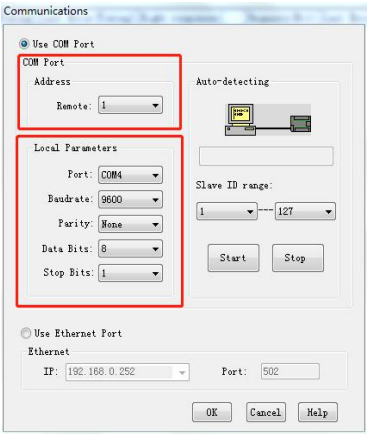
protocol after powering up. At the same time, it can be used as Kinco PLC interconnection slave and Modbus RTU slave.

2.2.2.2.1 Programming protocol

LoRa port supports programming protocol instruction in KincoBuilder (in【PLC】and【test】), including upload, download, online monitoring, force and etc. At the same time, it can also modify LoRa communication parameters of all nodes in the net.

Kinco provide test module to achieve functions above. Usage is shown below:

1) Connect a Kinco test module in PC's USB or RS232.
2) Open 【Communications】, choose 【Remote】station number and 【port】 of PC. Communication parameters of test module is: baudrate 115200, None, data bits 8, stop bit1.If user choose RS232 port, users need to change parameters above of PC com port. If users use USB, no need to do this.
3) Configure LoRa port communication parameters according to ways above and ensure it can communicate with target PLC.Users don't need to care about【Remote】station number，choosing any station number will not influence settings for this module.
4) At last，operate instructions needed in KincoBuilder.



Please note when use wireless programming function： LoRa port of target PLC cannot use free communication;In LoRa net, it is not allowable to send message from other nodes except for test modules. So users must stop master's communication in net!

2.2.2.2.2 Kinco PLC interconnection protocol & interlink wizard

Kinco PLC interconnection protocol is designed between Kinco PLC network communication. This protocol uses master-slave mode. There must be only one master in next and other nodes must be slave. Master visit slave according to setted period and slave could only response after receiving request from master. In addition, master send "Write data"broadcast message and all slaves will deal with it after receiving broadcast.

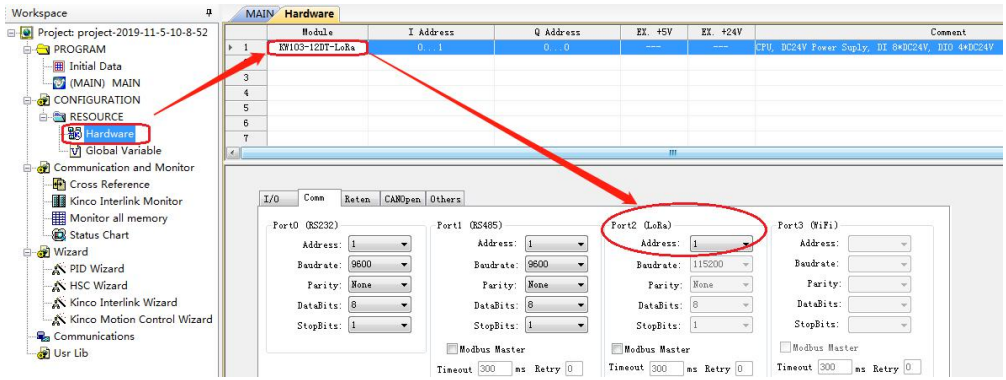
If users not enable other communication protocol functions, PLC will be Kinco interconnection slave after power on and users don't need to program. For master, Kinco interconnection protocol function has the highest priority. Once PLC work as interconnection master, this communication prot will not deal with message from other communication protocols.

- **Usage**

Users could enable Kinco PLC interlink function according to steps below.

1) Set LoRa communication parameters of each node according to preamble.

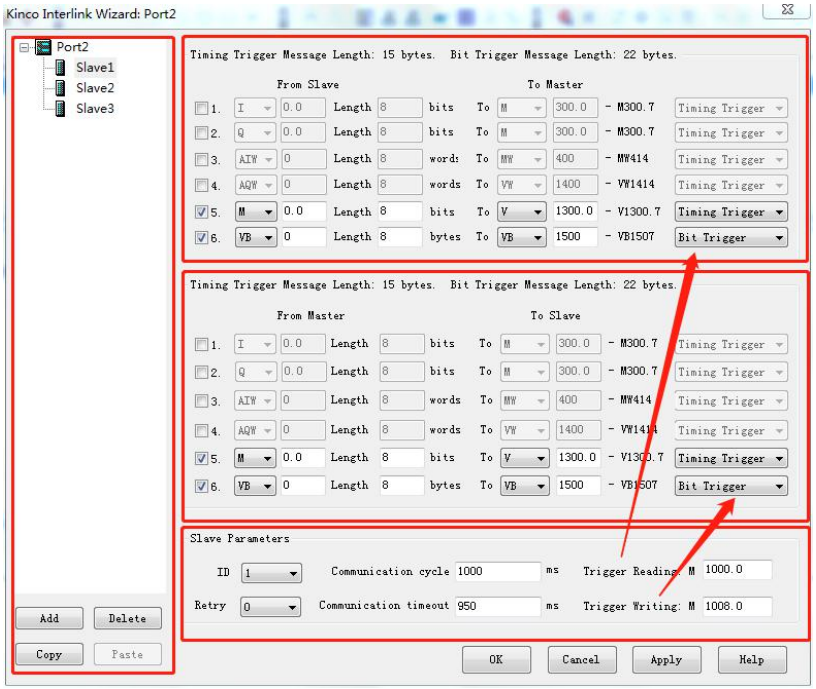
2) Each slave need to have a station number for LoRa port【Hardware】->【Comm】,valid station number is 1～63. In one net, each slave's station number must be unique.



3) In master's user project, use Kinco PLC interlink guide to set for each slave. After this, downloading project into a PLC and this PLC will run as a master. Then enable and arrange communication of the whole net.

➢ **Kinco PLC interlink wizard**

In KincoBuilder，double click node in【Wizard】-【Kinco PLC interlink wizard】，set communication parameters of all nodes in net.



- All slave list

  Edit connected slaves in this net and list out.

  Users must add all nodes in the net. The master only visit listed slaves.

  Single click one slave node and right side of window will show communication parameters and data of this node.**Each slave need to set a data message at least and slave without any data message will be ignored.After modifying【**slave ID**】**，node's topic will change automatically with new ID.

  If setting of slave is in trouble, a mark"?"will be shown in front of node topic，because when data read by multiple slaves are written in master slave, some ID are overlapping.

- Communication settings

  Slave visit slaves according to period set by users and visit period of each slave can be set alone.

  Data transmission message of each slave can be set to【Timing trigger】 and【Bit trigger】.When timing of one slave arrives, master will devide if it will transmit data with this slave according to trigger modes set by users:For data triggered by time, master will enable transmission at once; For data triggered by bit, master will judge trigger bit. If trigger bit is "0", master will not enable transmission. If trigger bit is "1", master will enable transmission and master will automatically reset trigger bit to "0".

【Slave ID】: Current slave ID

【Communication cycle】: Timing which master visit this slave，Unit：ms. The longest timing allowed by KW is about 49 days.

Users set visit period according to requirements.If it is necessary to update slave data rapidly, then communication cycle need to be short.If visit period is shorter than time of data transmission, then actual visit period is time of data transmission. If it is unnecessary to update data rapidly, users can make visit period bigger so as to decrease slave's communication frequency and reduce net pressure.

【Communication timeout】: After master sends out request, it must receive response message from slave within this time. Otherwise, slave will record communication timeout error.

【Retry】: After master send out requests, if there is communication error(no response for timeout or error in received message), master will retry request until communication is successful or repeat time of user setting is enough.

【Trigger reading】: Specify a variable in M area of master, which is used to trigger to master to read data set as "Bit trigger"from slave.After reading, master will clear this bit automatically. So in master user program, it is necessary to set this position to "1"and avoid keeping it to "1" all the time.

【Trigger writing】: Specify a variable in M area of master, which is used to trigger to master to write data set as "Bit trigger"from slave.After reading, master will clear this bit automatically. So in master user program, it is necessary to set this position to "1"and avoid keeping it to "1" all the time

- Data transmission setting

User set transmitted data of this slave. The data area which is used for master to read and write can be set to 6 groups at most. Data of each group can be set trigger mode (Timing trigger or bit trigger).

In order to improve communication efficiency, master will combine data area of user setting. Amoung them, all data of"Timing trigger"will be combined to a message and "Bit trigger" will be combined to another message. **LoRa communication message allow 246 bytes at most!** When users set communication data, software will automatically combine message distance. Please note that each mode is not over than 246 bytes. One message includes users' efficient data, extra protocol data of software and etc. **So the user valid data is about 226 bytes at most.**

- Broadcast message

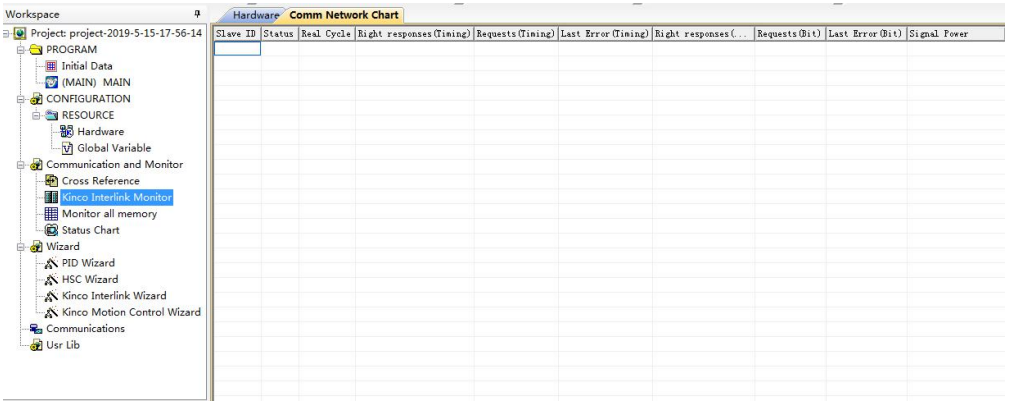
  Station No.64 is used to send broadcast message and modify all slaves'data of net.

  Station No.64 slave doesn't exist in net. Master will send data set in No.64 slave as broadcast message.All slaves in net will deal with after receiving message and it is uncessary to reponse.

➢ **Kinco PLC interlink slave status**

"Kinco PLC interconnection status" is used to monitor running information of all slaves.

In KincoBuilder – "Communication and Monitor", users double click【Kinco PLC Interlink Monitor】.



【slave ID】: Slave ID.

【Status】: Communication status of this slave, including: Normal, offline, no successful setting.

When net is enabled, master will send message data communication data area to each slave.After it is successful, this slave will have normal data link. If it fails, this master will show "No successful setting" and set this slave repeatedly until it is successful.

When data are interacted normally, master will shown "Offline" when this slave has wrong response. If it is normally next time, it will show "Normal" after communication.

【Real cycle】: The actual period which master make polling visit on slave，unit ms.

【Right Response】: Every time when master receives message from slave, it will detect and record strength of wireless signals. When signal strength is negative value, the signal strength is higher when it is near to 0. In applications, if users find communication error ratio of one slave is high but signal strength is low, it is necessary to improve installation of this slave. For example, users can adjust position of antenna or use higher-gain antenna.

【Communicate correct rate】: The correct message rate received from slave when master and slave powers on.

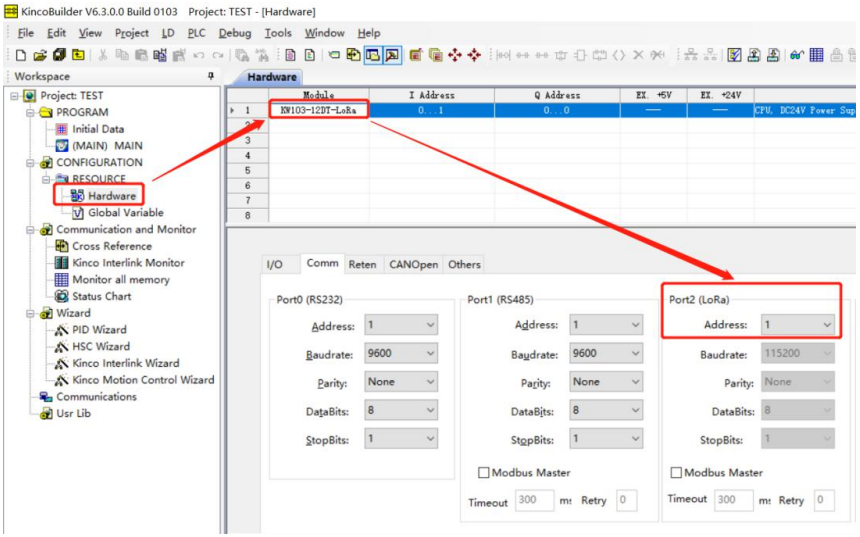
$$\text{Communicate correct rate} = \frac{\text{Total amount of correct message responsed by slave}}{\text{Total amount of request message from master}} \times 100\%$$

Correct rate of wireless communication might not 100%. It is good if correct rate is over 95%.

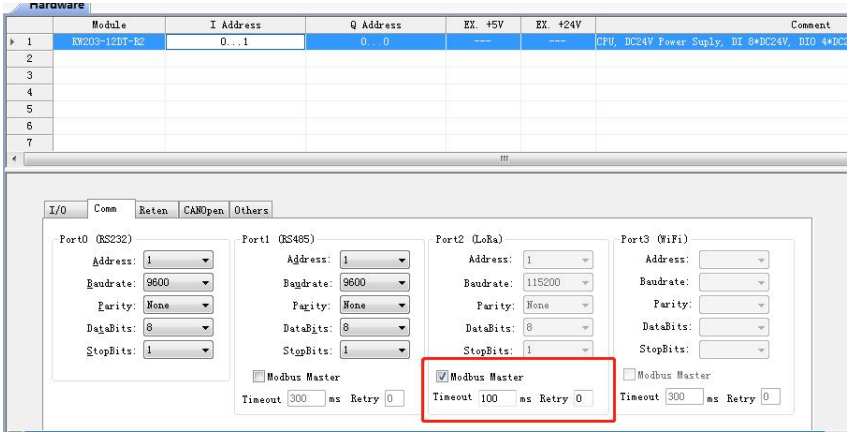
2.2.2.2.3 Modbus RTU protocol

KW supports to use Modbus RTU to communicate via LoRa communication interface.Steps are shown below:

1) Configure LoRa port communication parameters according to ways of preamble.
2) Each slave need to specify a station number for LoRa communication port in 【Hardware】-> 【Comm】 and its valid station number is 1~64.In one net, the station number of each slave must be unique and same station number is not allowe.



3) In user project of master, enable ModbusRTU master function, shown below:
Then users can use MBUSR、MBUSW to visit each slave.



Usage of instructions - please refer to2.3.3 Modbus RTU Master Instruction

2.2.2.2.4 Free communication function

The free communication means that the communication process and communication data are completely controlled by the user program. Users can use free communication to write various custom communication protocols to communicate with other devices.

When the free communication instructions in the user program is executed, the free communication mode is activated, and the corresponding communication port is completely occupied by the free communication.

After the free communication command is completed, the CPU switches the communication port to the default protocol automatically. If the PLC is in STOP, the free communication program is disabled and the PLC will restore the default programming protocol and Modbus RTU slave function.

For the instructions of the instruction, please refer to 2.3.4 Free communication.

2.3 LoRa function related instructions

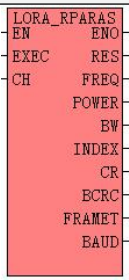
All the instructions described in this section are located in the 【Instruction Set】->【Communication Command】 group.

In the program written with IL language, the functions of all LoRa instructions are consistent with those in the LD program, and both follow the execution principle of the IL program: if the CR value is 1, the instruction is scanned and executed, and the execution result does not affect the CR value. .

Therefore, in the below text we only describe the LD format instructions , and will not describe the IL format again.

2.3.1 LoRa Port Dedicated Instructions

2.3.1.1 LORA_RPARAS (read LoRa parameter)

	name	Usage	Suitable for
LD	LORA_RPARAS		• KW203 • KW213

Parameter	Input/Output	Data Type	Acceptable Memory Areas
EXEC	Input	BOOL	I、Q、V、M、L、SM
CH	Input	INT	V、M、L、constant
RES	Output	BYTE	V、M、L
FREQ	Output	INT	V、M、L
POWER	Output	INT	V、M、L
BW	Output	INT	V、M、L
INDEX	Output	INT	V、M、L
CR	Output	INT	V、M、L

BCRC	Output	BOOL	V、M、L
FRAMET	Output	INT	V、M、L
BAUD	Output	REAL	V、L

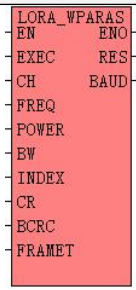
The LoRa parameter values read by this instruction correspond to the parameter options in the LoRa parameter configuration tool of Kincobuilder software.

Parameter	Description
EXEC	If a rising edge transition of EXEC is detected, the instruction is triggered to execute.
CH	LoRa port number which to be read parameter. 0 means the local port 1, 1 means the local port 2.
RES	The latest implementation result. Its composition is as follows: 7^{th} bit: instruction status. This bit is set to 0 if the instruction is executing and is immediately set when the instruction completes. 0^{th} bit :illegal LoRa port. This bit is set to 1 if the LoRa port number specified by the CH is an illegal value. Other bits are reserved
FREQ	The frequency channel option value read.
POWER	The value of the transmit power option read.
BW	The value of the bandwidth option read. This value corresponds to the BW option in the parameter configuration tool: 0 means BW203, 1 means BW406, 2 means BW812, 3 means BW1625,and 4 means FLRC. 0-3 means work in LoRa mode，4 means work in FLRC mode
INDEX	The value of the air transmission rate option read. This value corresponds to the option in the "Air Transfer Rate" list under the current BW option in the parameter configuration tool: 0 for the first item in the current list, 1 for the second item, and so on.
CR	Read encoding rate value.This value corresponds to the option in the "Encoding rate" list in the parameter configuration tool: LoRa mode:Valid value range is:1-4.1 for the first item in the current list, and so on. FLRC mode:Valid value range is:0-2.0 for the first item in the current list, and so on.
BCRC	Read the "Enable chip built-in CRC" option value.
FRAMET	The value of the packetized time option read.
BAUD	The air transmission rate calculated from the read BW and INDEX option values.

When the EN value is 1, if the rising edge of the EXEC input is detected. the instruction is triggered to execute once. RES is set to 0 when the instruction is executed. When the instruction is completed (either success or failure), the 7th bit of the RES is immediately set to 1. If the parameter is read successfully, KW will update the corresponding output parameter of this instruction according to the value of the LoRa parameter read, otherwise the value of the relevant output parameter will remain unchanged.

The user can judge whether the instruction is completed according to the rising edge of the 7th bit of the RES in the program, and then judge whether the reading is successful according to the error value indicated by other bits.

2.3.1.2 LORA_WPARAS（modify LoRa parameter）

	name	Usage	Suitable for
LD	LORA_WPARAS		• KW203 • KW213

Parameter	Input/Output	Data Type	Acceptable Memory Areas
EXEC	Input	BOOL	I、Q、V、M、L、SM
CH	Input	INT	V、M、L、constant
FREQ	Input	INT	V、M、L
POWER	Input	INT	V、M、L
BW	Input	INT	V、M、L
INDEX	Input	INT	V、M、L
CR	Input	INT	V、M、L
BCRC	Input	BOOL	V、M、L
FRAMET	Input	INT	V、M、L
RES	Output	BYTE	V、M、L
BAUD	Output	REAL	V、L

The LoRa parameter input values in this instruction correspond to the parameter options in the LoRa parameter configuration tool of Kincobuilder software.

Parameter	Description
EXEC	If a rising edge transition of EXEC is detected, the instruction is triggered to execute.
CH	The LoRa port number to be modified. The meaning of the input values is as follows: 0 :local port 1; 2 :port 1 network all nodes; 1 :local port 2; 3 :port 2 network all nodes
FREQ	New frequency channel option value.
POWER	New transmit power option value.
BW	New bandwidth option value. This value corresponds to the BW option in the parameter configuration tool:

	0 means BW203, 1 means BW406, 2 means BW812, 3 means BW1625,and 4 means FLRC. 0-3 means work in LoRa mode，4 means work in FLRC mode
INDEX	New air transfer rate option value. This value corresponds to the option in the "Air Transfer Rate" list under the current BW option in the parameter configuration tool: 0 for the first item in the current list, 1 for the second item, and so on.
CR	The new encoding rate option value.This value corresponds to the option in the "Encoding rate" list in the parameter configuration tool: LoRa mode:Valid value range is:1-4.1 for the first item in the current list, and so on. FLRC mode:Valid value range is:0-2.0 for the first item in the current list, and so on.
BCRC	New "Enable Chip CRC" option value.
FRAMET	New subcontract time option value.
RES	The latest implementation results. Its composition is as follows: 7^{th} bit: instruction status. This bit is set to 0 if the instruction is executing and is immediately set when the instruction completes. 1^{th} bit : the result of the modification. This bit is set if the parameter modification fails. 0^{th} bit : illegal LoRa port. This bit is set to 1 if the LoRa port number specified by the CH is an illegal value. Other bits are reserved
BAUD	The air transmission rate calculated from the input BW and INDEX option values.

When the EN value is 1, if the rising edge of the EXEC input is detected. the instruction is triggered to execute once, and the new parameter values are used to update the communication parameters of the Lora port CH.

RES is set to 0 when the instruction is executed. When the instruction is completed (either success or failure), the 7th bit of the RES is immediately set to 1. The user can judge whether the instruction is completed according to the rising edge of the 7th bit of the RES in the program, and then judge whether the modification is successful according to the error value indicated by other bits.

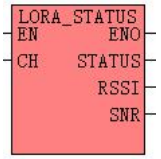
Users should pay attention to the following points when using this instruction:

1) At the same time, only one LORA_WPARAS instruction is allowed to be executed!

2) Modifying the parameters may affect the current normal data communication. Please make changes under the premise of ensuring security!

3) If you modify the parameters of all nodes in the network, it is recommended that the user execute this instruction through the master station in the network, and ensure that all nodes can communicate normally! In addition, since LoRa is a half-duplex communication method, KW cannot accurately determine whether the parameters of each node have been successfully modified!

2.3.1.3 LORA_STATUS（Get LoRa signal quality）

	name	Usage	Suitable for
LD	LORA_STATUS		• KW203

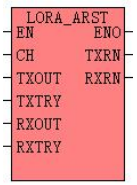
Parameter	Input/Output	Data Type	Acceptable Memory Areas
CH	Input	INT	V、M、L、constant
STATUS	Output	WORD	V、M、L
RSSI	Output	INT	V、M、AQW、L
SNR	Output	INT	V、M、AQW、L

Parameter	Description
CH	LoRa port number which to be used. 0 means the local port 1, 1 means the local port 2.
STATUS	The working status of the LoRa port. This parameter is reserved.
RSSI	The intensity value of the received signal. The value is a negative number, and the closer to 0, the higher the signal strength.
SNR	The signal to noise ratio of the received signal. The larger the signal to noise ratio, the better the signal quality.

When the EN value is 1, the instruction is executed. This instruction acquires the signal strength RSSI and signal to noise ratio SNR detected when the data was last received. The user can judge the quality of wireless communication according to these parameters: the larger the RSSI value, the higher the received signal strength, the smaller the attenuation of the signal during transmission; the larger the signal-to-noise ratio, the better the signal quality. The interference received is smaller.

2.3.1.4 Automatically reset LoRa communication port

- **LORA_ARST（Automatic reset caused by send and receive timeouts）**

	name	Usage	Suitable for
LD	LORA_ARST		• KW203 • KW213

Parameter	Input/Output	Data Type	Acceptable Memory Areas
CH	Input	INT	V、M、L、constant
TXOUT	Input	DWORD	V、M、L、constant
TXTRY	Input	INT	V、M、L、constant
RXOUT	Input	DWORD	V、M、L、constant
RXTRY	Input	INT	V、M、L、constant
TXRN	Output	WORD	V、M、L
SNR	Output	WORD	V、M、L

Parameter	Description
CH	LoRa port number which to be used. 0 means the local port 1, 1 means the local port 2.
TXOUT	Send timeout , ms.
TXTRY	The number of times the continuous timeout is allowed. If the number of consecutive transmission timeouts exceeds this value, the LoRa port will be automatically reset.
RXOUT	Recieve timeout , ms.
RXTRY	The number of allowed consecutive timeouts. If the number of continuous receive timeouts exceeds this value, the LoRa port will be automatically reset.
TXRN	After this power-on, the number of times the LoRa port was reset due to a failure in the transmission timeout.
RXRN	After this power-on, the number of times the LoRa port was reset due to the failure of the receiving timeout.

This instruction is used to specify the conditions of the LoRa communication port to be automatically reset. The rising edge of EN triggers the execution of this instruction. After the instruction is executed, the PLC will store the reset condition specified by the input parameter. In the later LoRa communication process, the PLC will automatically detect the timeout error of transmission and reception. If the reset condition is met, The PLC will automatically reset the LoRa communication port. **Therefore, if you do not need to adjust the reset condition multiple times, this instruction can be executed once!**

The TXOUT, TXTRY parameters are used to specify the condition that causes a continuous transmission timeout to cause a reset. When a transmission is initiated, the PLC starts to perform timeout detection. If the "successful transmission" signal is not detected within the TXOUT time, it is considered to be a transmission timeout. If the transmission timeout occurs continuously and the number of timeouts reaches the TXTRY value, the PLC will automatically reset the LoRa communication port and increase the TXRN value by 1.

The RXOUT and RXTRY parameters are used to specify conditions for continuous reception timeouts that cause a reset. When a reception is initiated, the PLC starts timeout detection. If a message is received within RXOUT time, it is considered to be a reception timeout. If the transmission and reception timeout occurs continuously and the number of timeouts reaches the RXTRY value, the PLC will automatically reset the LoRa communication port and increase the RXRN value by 1.

➢ Automatic reset of consecutively received message errors

SMB26 is used to store the maximum allowable number of consecutive error packets received by the LoRa communication port (local port 1). If the value is 0, this function is not enabled. If the value is greater

than 0, the PLC will automatically detect the number of message errors (CRC check errors, etc.). If the received message error occurs continuously and the number of errors reaches the value of SMB26, the PLC will automatically reset the LoRa port.

SMB28 is used to store the number of LoRa port resets caused by consecutively receiving message errors.

This reset condition is independent of the LORA_ARST instruction and can be used alone.

- **Timing automatic reset**

SMB24 is used to store the period of timing reset of LoRa communication port (local port 1). If the value is 0, this function is not enabled. If the value is greater than 0, the PLC will automatically start a timer. When the timing reaches the SMB24 value, the LoRa port will be automatically reset once.

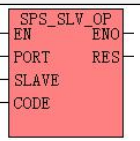
This reset condition is independent of the LORA_ARST instruction and can be used alone.

2.3.2 Kinco Interconnection protocol specific instruction

This group of instructions is only applicable to the "Kinco PLC Interconnection Protocol".

In the [Communication Settings], select [CPU], find out the [PLC Hardware Configuration], the user can see the specific number of each communication port of the machine.

2.3.2.1 SPS_SLV_OP（The master station pauses or restarts communication with the slave station）

	name	Usage	Suitable for
LD	SPS_SLV_OP		• KW103 • KW203

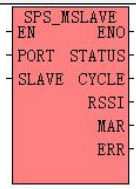
Parameter	Input/Output	Data Type	Acceptable Memory Areas
PORT	Input	INT	constant
SLAVE	Input	INT	V、M、L、constant
CODE	Input	BYTE	V、M、L、constant
RES	output	BYTE	V、M、L

Parameter	Description
PORT	The number of the communication port used. 0 means PORT0, 1 means PORT1, 2 means PORT2, and so on. If the parameter value specifies a communication port that does not exist, it is an illegal value, causing the instruction to report an error.
SLAVE	The station number of the target slave. The slave must already be present in the network, ie it must be configured in the [Kinco PLC Interconnect Wizard].
CODE	The operation options have the following meanings: 1 ~ The master starts communication with the target slave.

	2 ～ The master stops communication with the target slave.
RES	The result of the execution has the following meanings: 1 ~ Execution succeeded. 2 ~ The communication port used does not exist. 3 ～ The target slave station does not exist in the network. 4 ～ Illegal operation options.

When the EN value is 1, the instruction is executed. For the Kinco internetwork running on the PORT communication port, the user can call this instruction in the master station to suspend or restart communication with the target slave SLAVE.

2.3.2.2 SPS_MSLAVE（Read the communication status of the slave）

	name	Usage	Suitable for
LD	SPS_MSLAVE		• KW203

Parameter	Input/Output	Data Type	Acceptable Memory Areas
PORT	Input	INT	constant
SLAVE	Input	INT	V、M、L、constan
STATUS	output	BYTE	V、M、L
CYCLE	output	DWORD	V、M、L
RSSI	output	INT	V、M、L
MAR	output	WORD	V、M、L
ERR	output	BYTE	V、M、L

Parameter	Description
PORT	The number of the communication port used. 0 means PORT0, 1 means PORT1, 2 means PORT2, and so on. If the parameter value specifies a communication port that does not exist, it is an illegal value, causing the instruction to report an error.
SLAVE	The station number of the target slave. The slave must already be present in the network, ie it must be configured in the [Kinco PLC Interconnect Wizard].
STATUS	The operating status of the slave has the following meanings: 1 ~ Normal operation. 2 ～ The master station failed to configure the slave station. 3 ~ Offline (that is, the last time request did not receive a response message). 4 ~ The master station calls the SPS_SLV_OP instruction to suspend communication with

	the slave.
CYCLE	The true polling period of the master station to the slave station, unit: ms.
RSSI	The last time master station received the signal strength of the slave
MAR	the correct rate of the responds message by the slave station. The output value is a 16-bit integer. Meaning: Correct rate × 100. Correct rate = total number of packets correctly responded by the slave ÷ total number of request packets from the master to the slave
ERR	The most recent communication error occurred with this slave. 0 ~ No error. 1 ~ This slave has not responded with a timeout. 2 ~ The length of the "write data" message is incorrect. 3 ~ "Read data" message length is wrong. 4 ~ Incorrect configuration message. 5 ～ Incorrect communication area configuration.

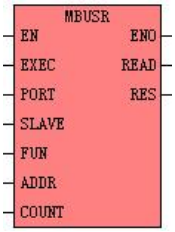
When the EN value is 1, the instruction is executed. For the Kinco internetwork running on the PORT communication port, the user can call this command in the master station to read the communication of the slave SLAVE.

In the [Communication Settings], select [CPU], find out the [PLC Hardware Configuration], the user can see the specific number of each communication port of the machine.

2.3.3 Modbus RTU Master Instruction

All communication-enabled communication ports (RS485, LoRa, etc.) support the Modbus RTU master function.

2.3.3.1 MBUSR (Read Slave Data) Instruction

	name	Usage	
LD	MBUSR		☑ K5 ☑ K2 ☑ KS ☑ KW

Parameter	Input/Output	Data Type	Acceptable Memory Areas
EXEC	Input	BYTE	I、Q、V、M、L、SM、RS、SR
PORT	Input	INT	constant
SLAVE	Input	BYTE	I、Q、M、V、L、SM、constant
FUN	Input	INT	constant（MODBUS function code）

ADDR	Input	INT	I、Q、M、V、L、SM、constant
COUNT	Input	INT	I、Q、M、V、L、SM、constant
READ	output	BOOL、WORD、INT	Q、M、V、L、SM
RES	output	BYTE	Q、M、V、L、SM

Note: 1) The parameters SLAVE, ADDR, COUNT must be either constant type or memory type at the same time.

2) The READ and COUNT parameters together form a variable-length memory block, which must all be in the legal memory area, otherwise the result is unpredictable.

When the PLC is used as the Modbus RTU master, the MBUSR instruction is used to read the data in the slave. The function codes for this instruction are 1 (read DO), 2 (read DI), 3 (read AO), and 4 (read AI).

The parameter PORT defines the communication port used. SLAVE defines the station number of the target slave. The allowed station number ranges from 1 to 255. FUN defines the function code. ADDR defines the starting address of the register to be read. COUNT defines the number of registers read, and the maximum allowed is 32.

A rising edge transition on the EXEC input is used to initiate communication. When the MBUSR instruction is executed, if the rising edge of EXEC is detected, MBUSR will perform a communication: organize the message according to the parameters such as the station number and function code input by the user and complete the CRC check, and then send the message and Waiting for the response from the slave station; after receiving the message returned by the slave station, it performs CRC check, address check and function code check. If the message proves to be correct after verification, the required data will be written to the data buffer, otherwise the received message will be discarded.

The parameter READ defines the starting address of the data buffer, and the read data (the number is COUNT) is stored in this area. READ must match the function code. If the function code is 1, 2, then enter the BOOL type address variable; if the function code is 3,4, enter the INT or WORD type address variable. The parameter RES is used to store the current state information and the fault information of the latest communication. Its composition is as follows:

MSB LSB

7	6	5	4	3	2	1	0

7:current communication status.when communicating is set 0, other sets 1.

6: reserved

5: illegal salve station number

4:number of data error

3:illegal address of register

2:communicaton port is using by other communication task

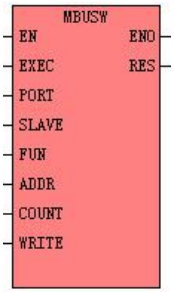
1:communication timeout. The user did not receive a response from the slave within the specified time.

0:received message error.Include CRC error,function code error etc.

• LD format instruction description

If EN is 1, the instruction is executed, otherwise it is not executed.

2.3.3.2 MBUSW (write data to slave) instruction

	name	Usage	
LD	MBUSW		☑ K5 ☑ K2 ☑ KS ☑ KW

Parameter	Input/Output	Data Type	Acceptable Memory Areas
EXEC	Input	BYTE	I、Q、V、M、L、SM、RS、SR
PORT	Input	INT	constant
SLAVE	Input	BYTE	I、Q、M、V、L、SM、constant
FUN	Input	INT	constant（MODBUS function code）
ADDR	Input	INT	I、Q、M、V、L、SM、constant
COUNT	Input	INT	I、Q、M、V、L、SM、constant
WRITE	Input	BOOL、WORD、INT	I、Q、RS、SR、V、M、L、SM、T、C
RES	output	BYTE	Q、M、V、L、SM

Note: 1) The parameters SLAVE, ADDR, COUNT must be either constant type or memory type at the same time.

2) The READ and COUNT parameters together form a variable-length memory block, which must all be in the legal memory area, otherwise the result is unpredictable.

When the PLC is used as the Modbus RTU master, the MBUSW instruction is used to write data to the slave. The function codes for this instruction are 5 (write one DO), 6 (write one AO), 15 (write multiple DOs),

and 16 (write multiple AOs).

The parameter PORT defines all the communication port used. SLAVE defines the station number of the target slave. The allowed station number range is 1~31. FUN defines the function code. ADDR defines the starting address of the register to be written. COUNT defines the number of write registers, the maximum allowed is 32.

The parameter WRITE defines the starting address of the data buffer, and the data to be written to the slave is stored in this area. WRITE must match the function code. If the function code is 5 or 15, enter the BOOL type address variable; if the function code is 6, 16, enter the INT or WORD type address variable.

A rising edge transition on the EXEC input is used to initiate communication. When the MBUSW instruction is executed, if the rising edge of EXEC is detected, MBUSW will perform a communication: organize the message according to the target station number, function code, target register, quantity, written data and other parameters input by the user and complete the message CRC check, then send the message out and wait for the response from the slave; after receiving the message returned by the slave, it will perform CRC check, address check and function code check, and judge whether the slave is correct execute the previous write command.

The parameter RES is used to store the current state information and the fault information of the latest communication. Its composition is as follows:

MSB LSB

7	6	5	4	3	2	1	0

7:current communication status.when communicating is set 0, other sets 1.

6: reserved

5: illegal salve station number

4:number of data error

3:illegal address of register

2:communicaton port is using by other communication task

1:communication timeout. The user did not receive a response from the slave within the specified time.

0:received message error.Include CRC error,function code error etc.

- LD format instruction description

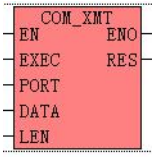
If EN is 1, the instruction is executed, otherwise it is not executed.

2.3.4 Free communication

The free communication command applies to all LoRa, RS232 and RS485 communication ports.

The free communication means that the communication process and communication data of the communication port are completely controlled by the user program. Users can use free communication commands to write various custom communication protocols to communicate with other devices.

2.3.4.1 COM_XMT (send datas)

	name	Usage	Suitable for
LD	COM_XMT		• KW203

Parameter	Input/Output	Data Type	Acceptable Memory Areas
EXEC	input	BOOL	I、Q、V、M、L、SM
PORT	input	INT	constant
DATA	input	BYTE	V、M、L
LEN	input	INT	V、M、L、constant
RES	output	BYTE	V、M、L

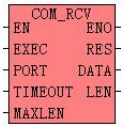
Parameter	Description
EXEC	If a rising edge transition of EXEC is detected, the instruction is triggered to execute.
PORT	The number of the communication port used. 0 means PORT0, 1 means PORT1, 2 means PORT2, and so on. If the parameter value specifies a communication port that does not exist, it is an illegal value, causing the instruction to report an error.
DATA	The first address of the storage area of the data to be sent.
LEN	Length of data to be sent, unit: bytes. **A maximum of 248 bytes of data can be sent at a time.**
RES	The latest implementation results. Its composition is as follows: 7^{th}: command status. This bit is set to 0 if the instruction is executing and is immediately set when the instruction completes and this bit is set 1 immediately. 4^{th}: Send error. If an error occurs during transmission, the transmission is stopped and this bit is set 1. 3^{th}: The communication port is busy. If the PORT is in the process of transmitting, this time does not initiate transmission, this bit is set to 1. 2^{th}: Data length error. If the LEN value is equal to 0 or exceeds the maximum length,it will set 1. 1^{th}: send timeout error. If the data has not been sent for more than 2 seconds, the

	transmission is stopped and the bit is set to 1. 0^{th}: illegal communication port. This bit is set 1 if PORT is an illegal value. Other bits are reserved

When the EN value is 1, if the rising edge of the EXEC input is detected, the instruction is triggered to execute once.and send user datas from the PORT.

RES is set to 0 when the instruction is executed. When the instruction is completed (either success or failure), the 7th bit of the RES is immediately set to 1. The user can judge whether the instruction is completed according to the rising edge of the 7th bit of the RES in the program, and then, based on the error value indicated by other bits, it is judged whether the transmission is successful.

2.3.4.2 COM_RECV (receive datas)

	name	Usage	Suitable for
LD	COM_RCV		• KW203

Parameter	Input/Output	Data Type	Acceptable Memory Areas
EXEC	input	BOOL	I、Q、V、M、L、SM
PORT	input	INT	constant
TIMEOUT	input	INT	V、M、L、constant
MAXLEN	input	INT	V、M、L、constant
RES	output	BYTE	V、M、L
DATA	output	BYTE	V、M、L
LEN	output	INT	V、M、L

Parameter	Description
EXEC	If a rising edge transition of EXEC is detected, the instruction is triggered to execute.
PORT	The number of the communication port used. 0 means PORT0, 1 means PORT1, 2 means PORT2, and so on. If the parameter value specifies a communication port that does not exist, it is an illegal value, causing the instruction to report an error.
TIMEOUT	Indicates the maximum time allowed for this reception. Unit: ms. If the TIMEOUT value is exceeded, the receiving status is exited and the corresponding flag of the RES is set.

MAXLEN	Indicates the maximum number of bytes allowed for this reception. The maximum value of **MAXLEN is 248**. Unit: Byte. If the number of bytes received by the PLC exceeds MAXLEN, only the first MAXLEN bytes are reserved and processed.
RES	The latest implementation results. Its composition is as follows: 7^{th}: command status. This bit is set to 0 if the instruction is executing and is immediately set 1 when the instruction completes. 4^{th}:receive error. If an error occurs during reception, reception is stopped and this bit is set 1. 3^{th}: communication port is busy. If the PORT is in the process of receiving, the reception will not be initiated this time, and this bit is set to 1. 2^{th}: MAXLEN values error. This bit is set 1 if the MAXLEN value is greater than 248 or less than zero. 1^{th}: receive timeout error. If the receiving process time reaches TIMOUT, the reception is stopped and this bit is set 1. 0^{th} :illegal communication port. This bit is set 1 if PORT is an illegal value. Other bits are reserved
DATA	The first address of the storage area where the data is received
LEN	The length of received. Unit：byte.

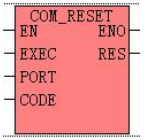
When the EN value is 1, if the rising edge of the EXEC input is detected, the instruction is triggered to execute: the PORT enters the receiving state. If a frame message is received, the receiving is stopped and the received data is stored in the receiving buffer. Medium (the first address of the buffer is DATA and the length is LEN). The condition for the end of the receiving process is: if the new character is not continuously received within the "3.5 character length" time (the instruction will automatically calculate according to the baud rate), the instruction considers that a complete message has been received and the reception is completed; If the reception time reaches TIMOUT, the instruction will also complete the reception immediately. After the reception is completed, the instruction will judge the number of characters received. If it exceeds MAXLEN, the instruction will only retain the first MAXLEN characters, and the rest will be discarded.

RES is set to 0 when the instruction is executed. When the instruction is completed (either success or failure), the 7th bit of the RES is immediately set to 1. The user can judge whether the instruction is completed according to the rising edge of the 7th bit of the RES in the program, and then judge whether the reception is successful according to the value of other bits: if the value of each bit is 0, the 1st bit is 1 or the 2nd bit is 1 , can be considered successful.

2.3.5 Reset communication port

2.3.5.1 COM_RESET (send data)

This directive applies to all LoRa, RS232 and RS485 communication ports.

	name	Usage	Suitable for
LD	COM_RESET		• KW203

Parameter	Input/Output	Data Type	Acceptable Memory Areas
EXEC	input	BOOL	I、Q、V、M、L、SM
PORT	input	INT	constant
CODE	input	BYTE	V、M、L、constant
RES	output	BYTE	V、M、L

Parameter	Description
EXEC	If a rising edge transition of EXEC is detected, the instruction is triggered to execute.
PORT	The number of the communication port to be reset. 0 means PORT0, 1 means PORT1, 2 means PORT2, and so on. If the parameter value specifies a communication port that does not exist, it is an illegal value, causing the instruction to report an error.
CODE	The reset option has the following meanings: 0 ~ Only reset all software variables such as cache and status used by this communication port. 1 ~ Reset all software variables used in this communication port, and perform hardware reset and initialization on the chip. Other values are illegal and will cause the command to report an error.
RES	The latest implementation results. Its composition is as follows: 7^{th}: command status. This bit is set to 0 if the instruction is executing and is immediately set when the instruction completes. 1^{th}: illegal reset option. This bit is set if CODE is an illegal value. 0^{th}:illegal communication port. This bit is set if PORT is an illegal value.

When the EN value is 1, if the rising edge of the EXEC input is detected, the instruction is triggered to execute once, and the PORT communication port is reset by pressing the CODE option. RES is set to 0 when the instruction is executed. When the instruction is completed (either success or failure), the 7th bit of the RES is immediately set to 1. The user can judge whether the instruction is completed according to the 7th bit of the RES in the program, and then judge whether the success is successful according to the error value indicated by other bits.

Chapter 3 Standard CPU module

3.1 structure appearance

The KW series adopts ultra-thin design, small size and convenient installation. Its appearance structure is as follows:

3.2 Technical Parameters

KW standard type provides a variety of CPU specifications, the following table describes the main technical parameters of various types of CPU.

Parameter	KW103	KW203
Power supply		
Rated power supply	DC24V.Note: The USB port can also be directly powered for CPU operation.	
Supply voltage range	DC20.4V～28.8V.	
Wireless communication port		
Type	LoRa，Working frequency 410～493MHz	LoRa,Working frequency 2400 ～ 2500MHz
Transmit power	160mW	100mW
Air transmission rate	0.81-18.23Kbps	0.59～253.91Kbps
Reference communication distance	1000m（Air speed 0.81Kbps）	3000m（Air speed 12.69Kbps）
	Fine weather, open and unobstructed environment, antenna gain 3.5dBi, antenna height 1.5m	
letter of agreement	Support programming protocol, Kinco interconnection protocol, Modbus RTU master and slave, free communication	
I/O and expansion		
IO channel	8 * DI (source / sink) / 4 * DIO (transistor)	
High-speed pulse input	4 channels, the highest frequency is all 200KHz, support single-phase and AB-phase counting.	
High speed output	2 channel，Max 200KHz（load resistance is less than 1.5KΩ at highest frequency）.	
Interrupt	4 channel,I0.0-I0.3 interrupt up and down.	
Expansion	Support KS series expansion，max 12.	
Communication port		
Serial port	PORT0,RS232,support programming protocol, MODBUS RTU slave, free protocol PORT1,RS485，support programming protocol, MODBUS RTU (as a master or slave), free protocol.	
CAN port	1 channel,support CANOpen master,Kinco Motion control and free protocol.	
USB port	1 channel,MicroUSB,support programming protocol.	
Storage		
Programming	Max 4K bytes instruction.	
Data	M area：1K bytes,V area：4K bytes.	

Data backup	E^2PROM，448 bytes,permanent storage.
Data retention	V area: 2K bytes（VB0-VB2047），C area.3 years at normal environment.
Other	
Timer	256.1ms: 4,10ms: 16, 100ms: 236.
Counter	256.
Timer interruption	2，0.1ms.
RTC	Yes,the difference is 5 min/month at 25℃.
Installation dimension	100*25.4*84.5mm

3.3 Functions

3.3.1 CPU Status and LEDs

The CPU has two modes: STOP mode and RUN mode.

In RUN mode, the CPU executes the main scan cycle and all interrupt tasks.

In STOP mode, the CPU will set all output channels (including DO and AO) to the known values which are specified in the [Hardware Configuration] through Kincobuilder, and only process communication requests which comes from KincoBuilder software and other Modbus RTU master device.

- **Change CPU status**

Kinco KW provides two ways for manually changing the CPU status:

1) Put operation switch(1-3) all to OFF,if anyone of those is ON,then CPU ON.

2) Executing [Debug] -> [RUN] or [STOP] menu command in Kincobuilder.

In addition,PLC RUN mistake (strong mistake) will stop the PLC.

- **CPU Status LED**

The CPU module provides 4 status LEDs: RUN, STOP, Comm. and Err.

LED	Description
Run	If CPU is in RUN status,it will turn on. If CPU is in STOP status,it will turn off.
Err	If CPU detects error in user program or module, it will turn on.
Com	Any serial port（PORT0/1）read or send data,it will twinkle.
Net	When wireless port read or send data,it Will twinkle.

3.3.2 I/O function

- **DI channel**

KW CPU provides some DI channel.

DI channel can detect common DI signals.And KW also provides [input filtering] function,User can select filter delay for DI point (max 12.8 ms),In this way, the DI signal at least maintain the configured delay time before being considered effective by the CPU，therefore, it is helpful to enhance the anti-interference ability of the system.

I0.0-I0.3 in CPU support edge interrupt function,it can execute interrupt by rising edge and falling edge of input signal. By using this function, it can capture the rising edge and falling edge of input signal quickly. For some input signal whose pulse width is less than the CPU scan time, it can respond quickly.

- **DIO channel**

KW CPU provides some DIO channels.. Each DIO channel can be used as DI(Source type) or DO (Source type) ,its signal is DC24V. User doesn’t need to configure the DIO channels ,it will self-adapt according to the wiring.

Each DIO channel occupies two address, one is for DI(in I area) and the other one is for DO(in Q area).Take kw203 for example,the first DIO channel address is Q0.0 and I1.0，.If you need to use it as DI, then you can connect input signal to this channel directly, and use I1.60 in the program.If you need to use it as DO, then user can connect this channel to output device and use Q0.0 in the program,

Note: If a channel is used as DI, you shouldn’t use DO address of this channel. If a channel is used as DO, you shouldn’t use DI address of this channel.

- **High Speed Counter**

A part of the DI channels of the KW body can also be used for the channels of high-speed pulse counters. But the same channel, high-speed pulse input function and ordinary DI function cannot be used at the same time!

KW provides 4 high speed counters (HSC0~HSC3).High speed counter supports multiple modes: single phase, CW/CCW(Up/Down),AB phase (1 multiplication and 4 multiplication),all max 200KHz.

There are two ways to use the high-speed counter:

1) Use [HSC Wizard] to configure in KincoBuilder programming software, and download user project to CPU.

2) The user configures the relevant control words in the program and calls the HDEF and HSC instructions.

- **High Speed Pulse Output**

A part of the DO (or DIO) channels of the KW body can also be used for the channels of high-speed pulse counters. But the same channel, high-speed pulse output function and ordinary DO function cannot be used at the same time!

KW provides 2 high speed pulse outputs(Q0.0,Q0.1).All support PTO and PWM.Q0.0 and Q0.1 support up to 200KHz (The resistor of load should be less than 1.5KΩ).

Users can call PLS, PFLO_F or location control instructions in the program to use the high-speed output function.

3.3.3 Serial port

KW provides 2 communication ports,PORT0 and PORT1,PORT0 is RS232,PORT1 is RS485.

All serial ports support programming protocol,can be used as programming port. In addition,PORT1 supports Modbus RTU(Master and Slave) and free protocol,PORT0 supports Modbus RTU Slave,free protocol.Each serial port is regarded as MODBUS slave by default,it can be used without programming.**The Modbus RTU master and free communication functions have exclusive communication ports. If one of them is used in the user program, the serial port used cannot be used for other protocol communication.**

When using PORT1 (RS485) interface for communication, a bus topology is recommended . In addition，PORT1 built-in 120 Ω terminal resistance,controlled by the **fourth switch:**put it to ON，then use terminal resistance,put it to OFF，then cancel terminal resistance.

PORT0 and PORT1 are in the RJ45 port. Pins and functions as below：

	Pin	Function
RS232 PORT0	3	TXD
	4	GND
	6	RXD
RS485 PORT1	1	A(+)
	2	B(-)
	5	GND

⚠ RS232 can't insert and release with power. So we should turn off power of CPU or PC, otherwise it will break ports.

3.3.4 Can bus and extended bus port

KW provides 1 CAN Bus port,support for Extended Bus Protocol,CANopen (Master) protocol,Kinco motion control protocol and Free protocol function.Free protocol can be used with any other protocol in the same time,however, except for free protocol,other protocols can not be used at the same time.

When use Can port,in order to eliminate the signal reflection on the communication cable,120 Ω terminal resistance is usually added to the first, last or one end of the bus.The CAN port of KW has a built-in 120 Ω terminal resistance,controlled by the **5th switch**：put the switch to ON,the terminal resistance is added；put the switch to OFF,the terminal resistance is canceled.

CAN1 is located in the RJ45 interface. The pin is defined as follows:

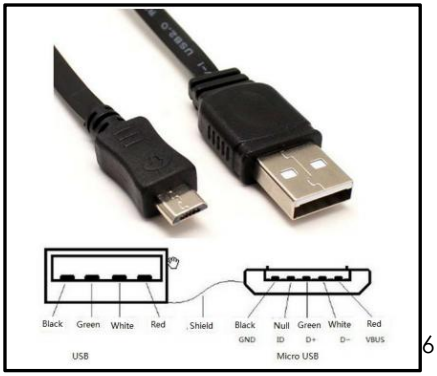
	Pin	Function
CAN1	1	CAN_H
	2	CAN_L
	3	GND

Refer to Chapter 7 the use of CAN bus communication ports to learn more about the use of various functions.

3.3.5 USB Port

KW provides 1 USB2.0 port,adopt Micro USB port.USB port supports programming protocol,it can be used as programming port.Micro USB port is widely used in smart phones,Users can directly use the same interface of mobile phone data lines as programming cables.**But it should be noted that some mobile phone data lines can only be used for charging, can not be used!**The pin definition of the USB port is shown in the figure.

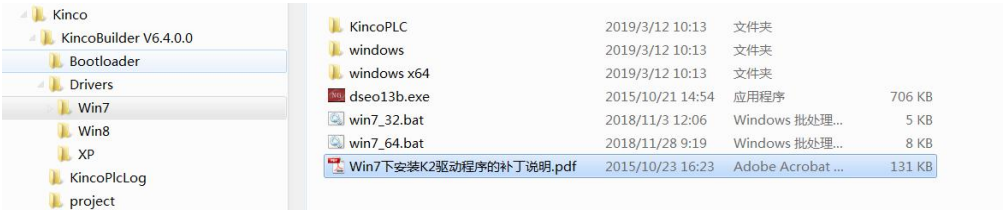
The USB programming interface is used as a virtual serial port on the computer,Its drivers are stored under the "Driver"



directory of the Kincobuilder software installation directory.Windows XP, Win7 and Win8 (Win10) systems are currently supported.When the user first uses the programmable data line to connect KW to the computer, the Windows system will automatically detect the new hardware and prompt the installation of the driver. At this time, the user can choose the driver under the corresponding directory according to his own version of Windows.

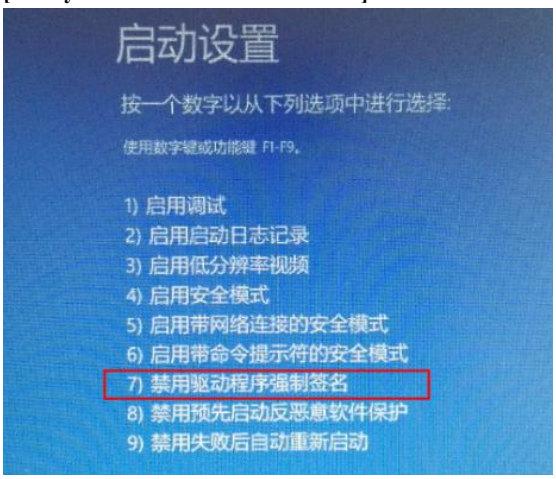
➢ **What if the driver can't be installed in Windows 7?**

This is due to the use of a simple Win7 system, the system lacks some necessary system files, resulting in the inability to install virtual serial ports.Under "…\Drivers\Win7" directory,We produced a batch file for this situation,Users can execute batch files according to their own Windows system type (32 or 64 bits),specific usage can refer to the description document <Patch Instructions for Installing K2 Driver under Win7>.



➢ **How to install driver in Win8 (or Win10) system?**

Windows 8 and above require third-party inf files to contain digital signature information certified by WHQL (Windows Lab), otherwise it is prohibited to install.The solution is to execute the command "Start Up Now" in Advanced Boot, then select "Disable Driver Forced Signature" in the subsequent boot setup interface and restart the computer,**The following diagram. This will temporarily disable the driver's mandatory signature checking function of Win8, and then continue to install the driver, and click Select in the pop-up prompt dialog box [Always install this driver software].**



➢ **What if the driver can't be installed by the above methods?**

When the above methods are unable to install drivers, the reason is that the WINDOWS operating system is not a full-featured installation version and lacks the necessary files of the system. At this time, the third-party

driver installation software can be used for automatic search and installation.

3.3.6 Data Retentive and Data Backup

Data retentive means the data in RAM can retain after power failure.CPU provides a lithium battery (Replaceable but un-rechargeable) for data retentive.When CPU loses power, the data in the RAM will be maintained by the lithium battery, and the retentive ranges will be left unchanged at next power on. Through [Hardware] configuration in KincoBuilder, user can select the type of data retentive (Such as V,C area) and the range.The retentive range in V area is from VB0 to VB2047.The life of battery is 5 years and the retaining duration is 3 years at normal temperature.

Data backup is that CPU provides an E2PROM to store data permanently. At power on, the CPU will restore the data from E2PROM into RAM to execute.**Because E2PROM has a writing limit of 1 million times, users should avoid to write data into data backup area frequently.**

There are 448 bytes in V area for data backup (VB3648--VB4095),the data in this area will save in E2PROM automatically.**.KW sets VB3648--VB3902 as data backup by default,if user needs to use VB3903--VB4095 for data backup,it needs to configure in 【PLC hardware configuration】.The configuration interface is as following figure.**

3.3.7 Real-time Clock (RTC)

The real-time clock built in the all CPU modules can provide real-time clock/calendar indication. Users need to use KincoBuilder【PLC】->【Time of Day Clock...】to set the clock when using RTC first time. Then users can use real-time clock instructions（READ_RTC、 SET_RTC、RTC_W、RTC_R）.

After CPU power off, the real-time clock can be maintained by lithium battery. The life of battery is 5 years and the retaining duration is 3 years at normal temperature.

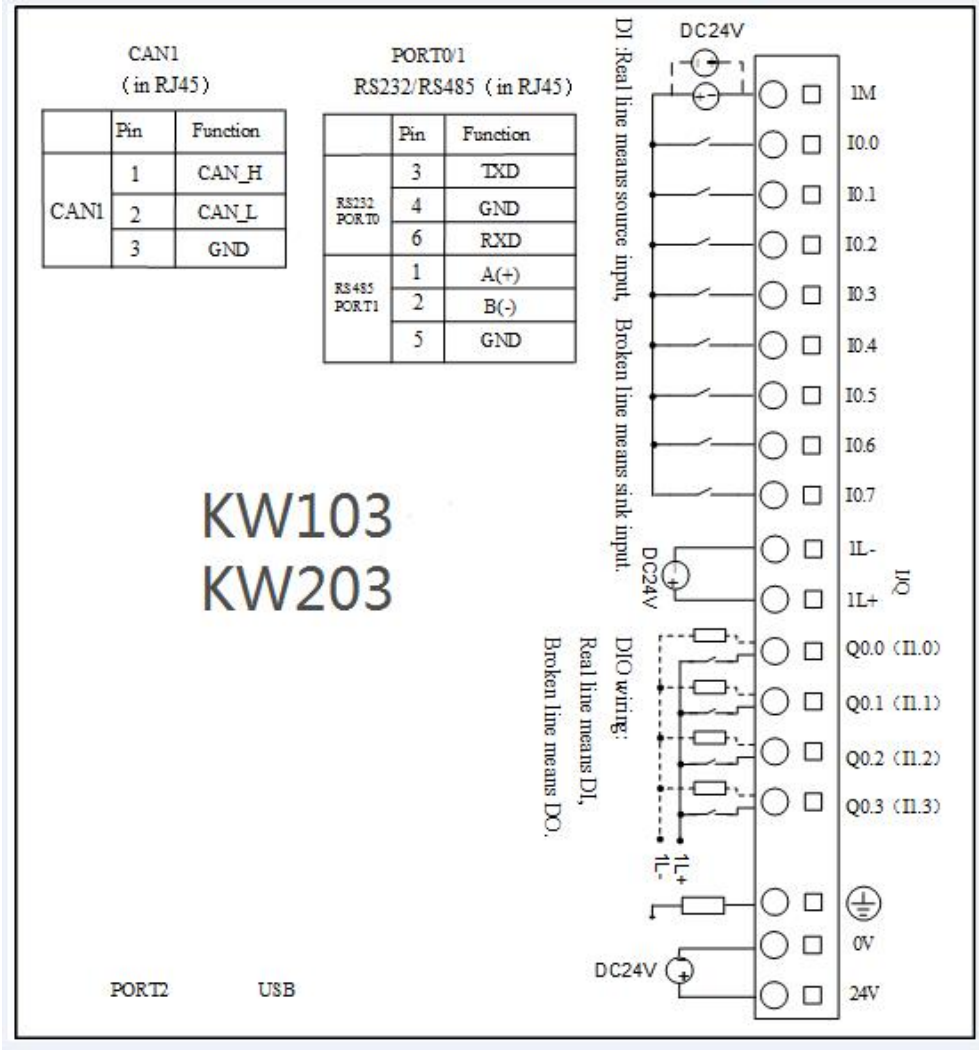
3.3.8 Backup Battery

KW can use certain specification lithium battery as backup battery. When PLC is power-off, it will use the backup battery to maintain real-time clock and RAM.

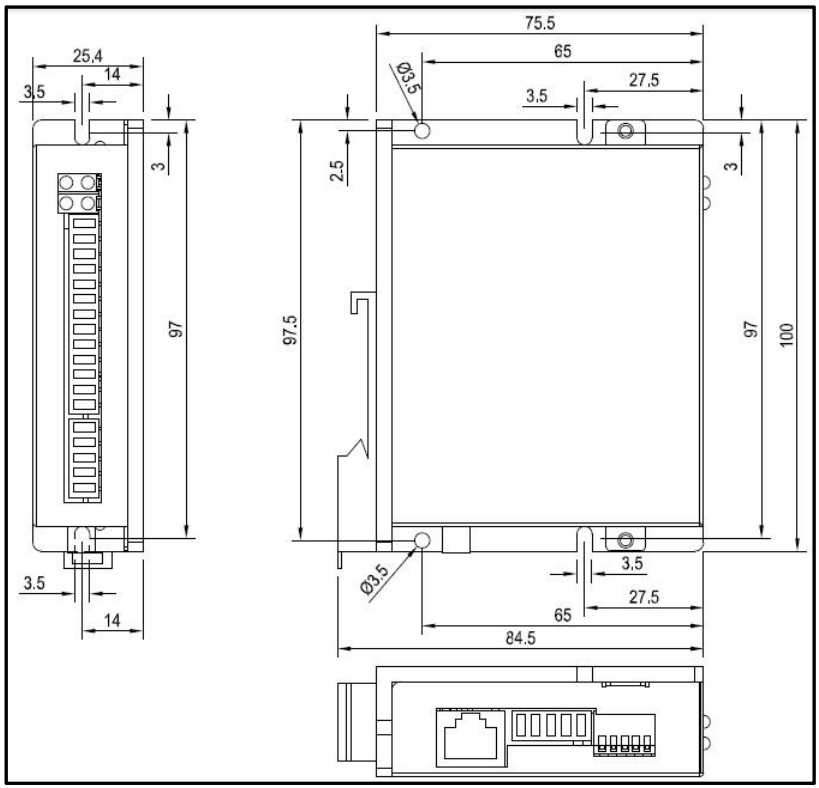
The backup battery is removable, user can replace new battery by themselves when the battery is empty. The lithium battery is CR2032(3V) with connector. As shown infigure,user can order the battery separately.

3.4 Wiring diagram

Mixed use of DI and DO,dotted line represents DO usage,sold line represents DI usage.Sold line represents source input,dotted line represents source input.



3.5 Dimension



3.6 Technical Specification of IO channel

➢ DI Specifications

Input type	Source/Sink
Rated input voltage	DC 24V (Max. 30V)
Rated input current	3.5mA@24VDC
Max input voltage of logic 0	5V@0.7mA
Minimum input voltage of logic 1	11V@2.0mA
Input filter time delay • off-to-on • on-to-off	 1.2 μ s 0.5 μ S
Isolation between input and internal circuit • Mode • Voltage	 Opto-electrical isolation 500VAC/1 min

➢ DIO Specifications(Transistor type)

Input/output type	Source

Rated input voltage	DC 24V (Max. 30V)
Rated input current	3.5mA@24VDC
Max input voltage of logic 0	5V@0.7mA
Minimum input voltage of logic 1	11V@2.0mA
Rated output voltage	DC24V,allowance range: DC20.4V—DC28.8V.(Same as power supply)
Output current per channel	Rated 200mA @24VDC
Instant impulse current per channel	1A,less than 1S
Output leakage current	Max 0.5μA
Output impedance	Max 0.2Ω
Input delay time • off to on • on to off	 1.2μS 0.5μS
Output delay time • off to on • on to off	 Common channel: 12μs; HSC channel: 0.5μs; Common channel: 35μs; HSC channel: 1μs;
Protection： • Inductive load protection • Short-circuit protection •Reverse polarity protection of output	 Yes Yes Yes,less than 10s.
Isolation between output and internal circuit • Mode • Voltage	 Opto-electrical isolation 500VAC/1 min

Chapter 4 Simple CPU module

The simple CPU is a simplified version of the standard,CPU provides some IO,RS485 port and LPWAN port,it allows users to use Bit Logic Instructions,Numeric Instructions,Logical Operations and Timers to program and realize simple control and calculation.

4.1 Technical Parameter

Parameter	KW213
Power supply	
Voltage range	DC10V～30V. Note：USB port can also be directly powered for CPU operation.
Wireless port	
Type	LoRa，Working frequency range 2400 ～ 2500MHZ
Max transmit power	100mW
RF Baudrate	0.59～253.91Kbps
Communication	3000m（RF 12.69Kbps）

distance	Clear weather, open and unobstructed environment, aerial gain 3.5dbi, aerial height 1.5m.
Protocol	Support programming protocol,Kinco Interlink、Modbus RTU Slave.
I/O and Expansion	
IO channel	8*DIO（transistor）. If all channel are only used as DI,then they support source or sink for input. If used as DI and DO,then DI only support source input.
High speed input	1 channel,max 10KHz,support single and AB phase.
Edge interrupt	4 channels,I0.0-I0.3 interrupt up and down.
Expansion	No support.
Wire port	
Serial port	PORT1:RS485,support Modbus RTU Slave.
USB port	1 channel,MicroUSB type,support programming protocol.
Storage	
Programming	Max 144 bytes instructions.
Data	M area：16 bytes,V area：256 bytes.
Data backup	VB224～VB255,32 bytes,written to permanent memory automatically by PLC. **Note: the lifetime of permanent memory is 250000 times. Therefore, it is necessary to avoid frequently modifying the value of the backup area!**
Other	
Timer	32. 1 ms : 4, 10 ms : 16，100 ms : 12.
Timer interruption	2， 0.1ms
RTC	No
Installation dimension	100*25.4*84.5mm

4.2 Function

4.2.1 CPU Status and LEDs

The CPU has two modes: STOP mode and RUN mode.

In RUN mode, the CPU executes the main scan cycle and all interrupt tasks.

In STOP mode, the CPU will set all output channels (including DO and AO) to the known values which are specified in the [Hardware Configuration] through Kincobuilder, and only process communication requests which comes from KincoBuilder software and other Modbus RTU master device.

- **Change CPU status**

Kinco KW provides two ways for manually changing the CPU status:

1) The 8th switch is used as RUN/STOP switch,put it to OFF,PLC STOP,put it to ON,PLC RUN.

2) Executing [Debug] -> [RUN] or [STOP] menu command in Kincobuilder.

In addition,PLC RUN mistake (strong mistake) will stop the PLC.

- **CPU Status LED**

The CPU module provides 2 status LEDs: RUN and Net.

LED	Description
Run	If CPU is in RUN status,it will turn on. If CPU is in STOP status,it will turn off
Net	When wireless port read or send data,it Will twinkle.

4.2.2 I/O channel

- **DIO channel**

KW CPU provides some DIO channels.. Each DIO channel can be used as DI(Source type) or DO (Source type) ,its signal is DC24V. User doesn’t need to configure the DIO channels ,it will self-adapt according to the wiring.

If all channel are only used as DI,then they support source or sink for input.If used as DI and DO,then DI only support source input.

Each DIO channel occupies two address, one is for DI(in I area) and the other one is for DO(in Q area).Take kw203 for example,the first DIO channel address is Q0.0 and I1.0，.If you need to use it as DI, then you can connect input signal to this channel directly, and use I1.60 in the program.If you need to use it as DO, then user can connect this channel to output device and use Q0.0 in the program.

Note: If a channel is used as DI, you shouldn’t use DO address of this channel. If a channel is used as DO, you shouldn’t use DI address of this channel.

The DI channel supports [input filtering] function,User can select filter delay for DI point (max 12.8 ms),In this way, the DI signal at least maintain the configured delay time before being considered effective by the CPU，therefore, it is helpful to enhance the anti-interference ability of the system.

I0.0-I0.3 in CPU support edge interrupt function,it can execute interrupt by rising edge and falling edge of input signal,and execute the corresponding interrupt program in the user project.Kw213 can capture the edge of Di signal with a minimum pulse width of about 100us,by using this function, it can capture the rising edge and falling edge of input signal quickly.

- **High Speed Counter**

A part of the DI channels of the CPU body can also be used for the channels of the high-speed pulse counter. But the same channel, high-speed pulse input function and ordinary DI function cannot be used at the same time!

KW213 provides 1 high speed counter,HSC0.High speed counter supports single phase, ,AB phase (1 multiplication and 4 multiplication),max 10KHz.

Users can configure it with **HSC wizard in** KincoBuilder,and download project to CPU,then you can use it.

4.2.3 Serial port

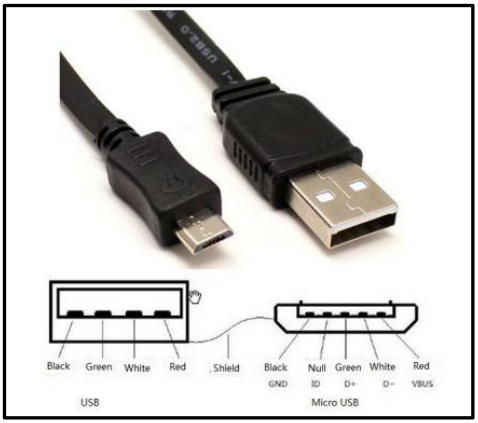
KW213 CPU provides 1 RS485 port, PORT1. PORT1 is on the terminal

PORT1 supports Modbus RTU Slave.When using PORT1 communication, a bus topology is recommended

4.2.4 USB port

KW provides 1 USB2.0 port,adopt Micro USB port.USB port supports programming protocol,it can be used as programming port.Micro USB port is widely used in smart phones,Users can directly use the same interface of mobile phone data lines as programming cables.**But it should be noted that some mobile phone data lines can only be used for charging, can not be used!**

The pin definition of the USB port is shown in the figure.



4.2.5 Data Backup

KW213 CPU does not provides a lithium Battery,so it does not support data retentive and RTC.

Data backup is that CPU provides an E2PROM to store data permanently. At power on, the CPU will restore the data from E2PROM into RAM to execute.**Because E2PROM has a writing limit of 200000 times, users should avoid to write data into data backup area frequently.**

VB224-VB255 in V area are set as data backup,the data in this area will save in E2PROM automatically.

4.2.6 Switch

KW simple CPU provides a set of 8 switches.

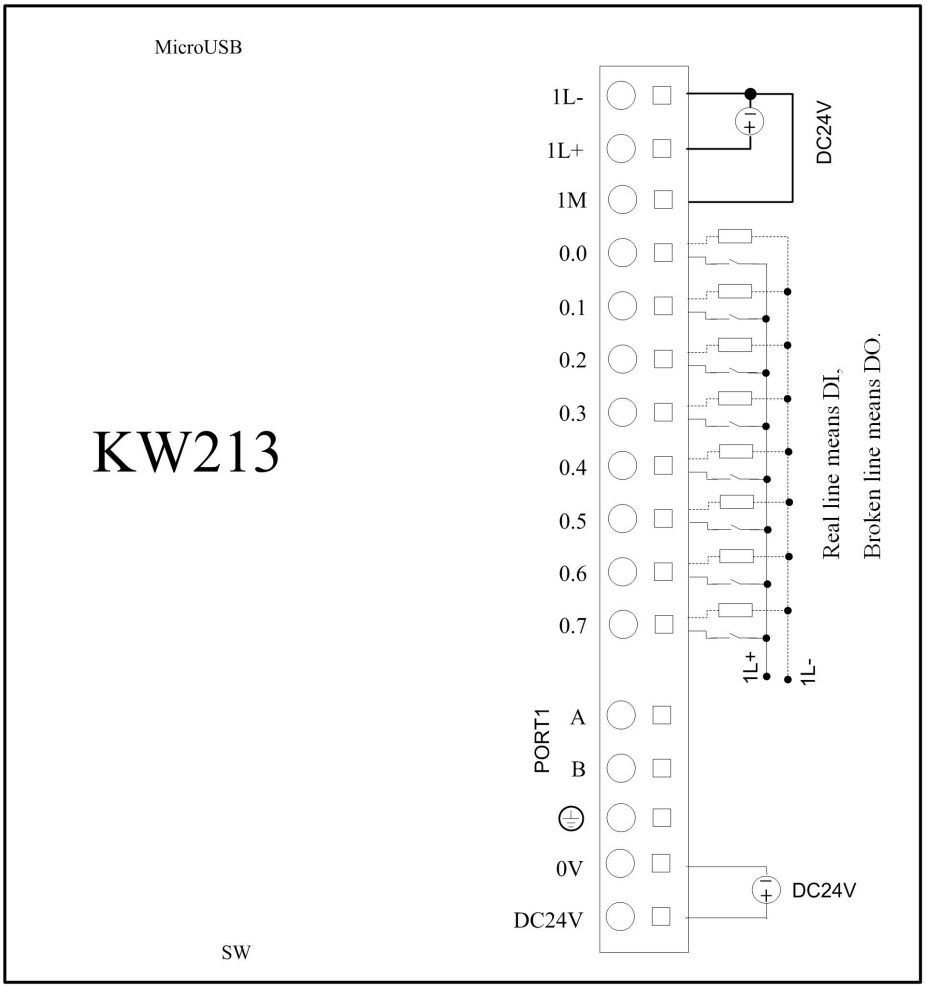
The 8th switch is used as RUN/STOP switch,put it to OFF,PLC STOP,put it to ON,PLC RUN.

The combined value of 1st to 7th of switch is the station number of Lora slave station. The combined value adopts binary mode, in which the 1st switch is the lowest bit and the 7th switch is the highest bit. Examples are as follows:

Station	1st	2nd	3rd	4th	5th	6th	7th
1	**ON**	OFF	OFF	OFF	OFF	OFF	OFF
2	OFF	**ON**	OFF	OFF	OFF	OFF	OFF
3	**ON**	**ON**	OFF	OFF	OFF	OFF	OFF
…	…	…	…	…	…	…	…
63	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**	OFF
64	OFF	OFF	OFF	OFF	OFF	OFF	**ON**

4.3 Wiring diagram

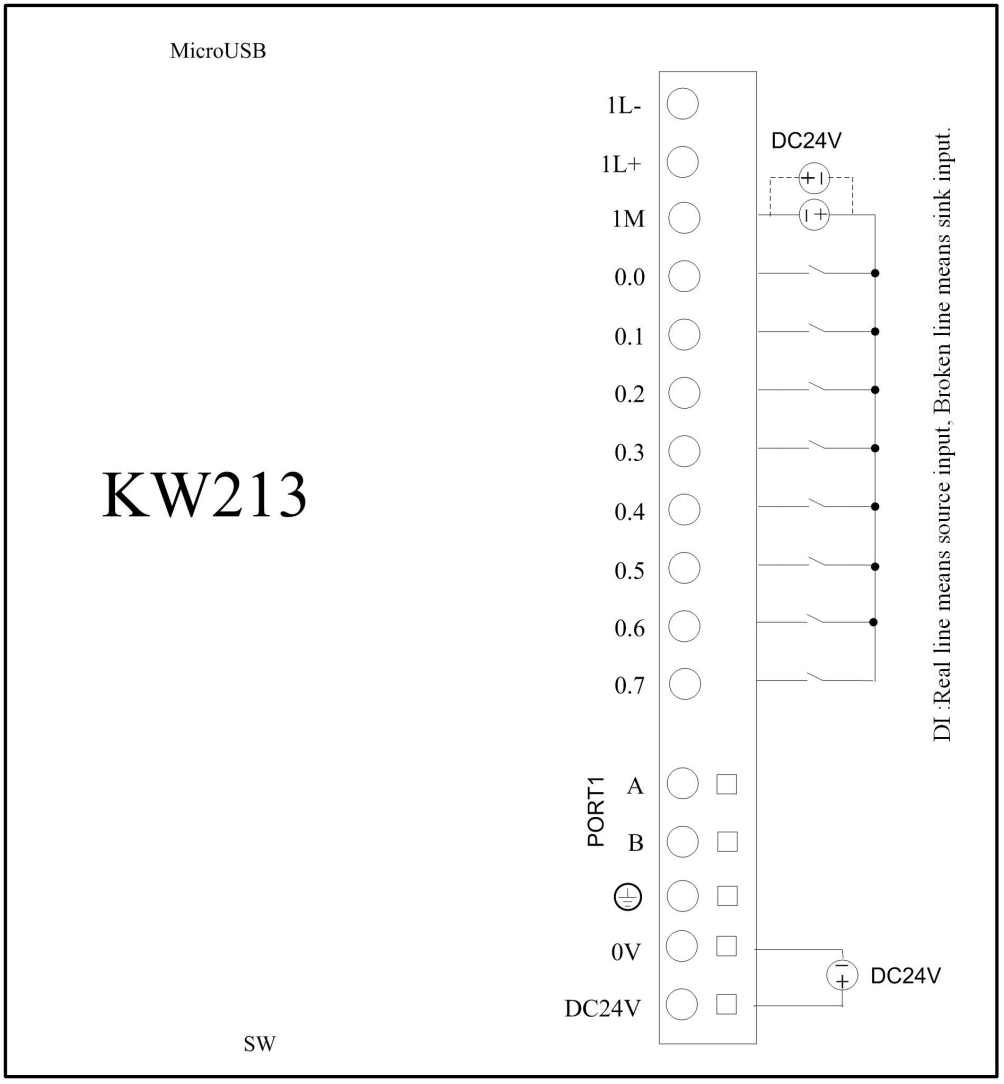
1）Mixed use of DI and DO,support source input and source output.



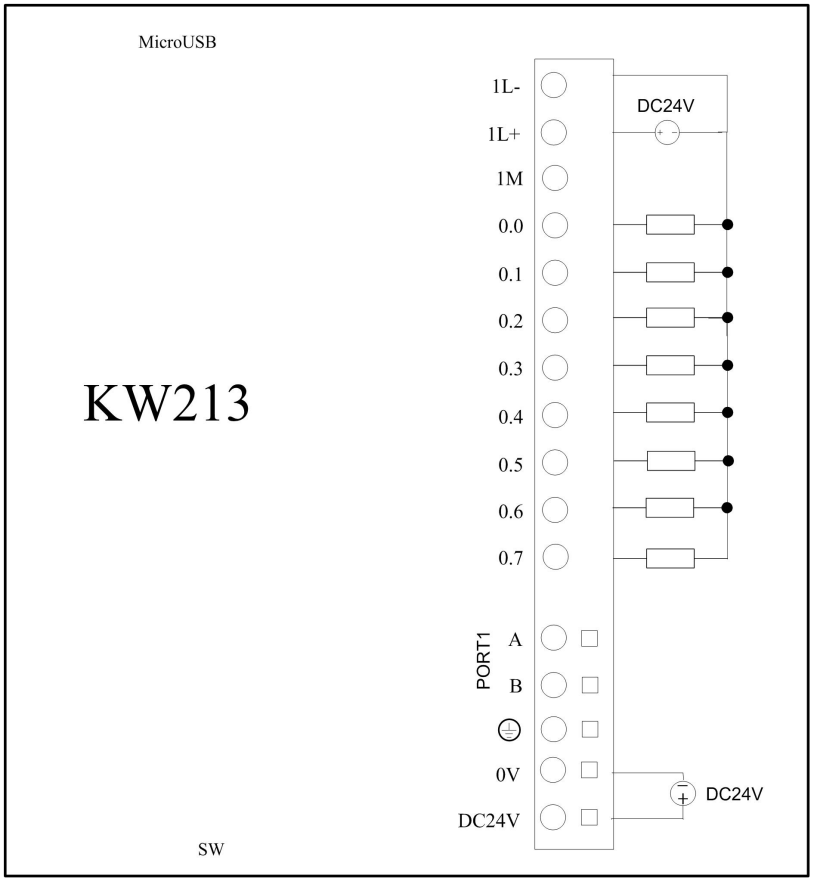
dotted line represents DO usage, solid line represents DI usage,

2）All are used as DI，support source and sink input.

Solid line represents source input,dotted line represents sink input.



3）All are used as DO,support source output.



4.4 Other

The dimension of KW213 is the same with KW203.
The I/O technical parameter of KW213 is the same with KW203.
Please refer to Chapter 3　Standard CPU module.

Chapter 5 The usage of high speed counter

KW standard CPU provides 4 high speed counter,HSC0 to HSC3,all max 200KHz.

KW simple CPU provides 1 high speed counter，HSC0，max 10KHz.

High Speed Counter has multiple work modes includes single phrase mode , 2 phrases(up/down) mode and AB phrases(1x rate and 4 x rate) mode. All the high speed counters have the same function working in the same mode.

All high speed counters has maximum 32 preset values(PV). Each PV supports the function of‘counting value= preset value’ interrupt .The PV is able to set to be absolute value mode or relative value mode. If the relative value mode is selected , it is allowed that the ‘counting value = preset value’ interrupt occurs cyclically.

5.1 High Speed Counter Work Modes and Input Signal

High Speed Counter includes the following input signals : clock(ie. Input pulses),direction, start and reset.

In different working modes , it is required to have different input signals.The table below describes the different working modes and the input signals allocation.

HSC 0				
Mode	**Description**	**I0.1**	**I0.0**	**I0.5**
0	Single phrase mode Internally change the counting value direction(increase /decrease) Direction change Bit：SM37.3	Clock		
1			Reset	
2			Reset	Start
3	Single phrase mode externally change the counting value direction(increase /decrease)	Clock		Direction
4			Reset	Direction
6	2 phrase mode Increase the counting value by phrase clock(plus) and Decrease the counting value by phrase clock (minus)	Clock（Minus）	Clock（Plus）	
9	Phrase A and B counting mode	Clock phrase A	Clock phrase B	

HSC1					
Mode	**Description**	**I0.4**	**I0.6**	**I0.3**	**I0.2**
0	Single phrase mode Internally change the counting value direction(increase /decrease) Direction change Bit：：SM47.3			Clock	
1		Reset			
2		Reset	Start		
3	Single phrase mode externally change the counting value direction(increase /decrease)			Clock	Reset
4		Reset			Reset
6	2 phrase mode Increase the counting value by phrase clock(plus) and Decrease the counting value by phrase clock (minus)			Clock（Minus）	Clock（Plus）
7		Reset			
9	Phrase A and B counting mode			Clock Phrase A	Clock Phrase B
10		Reset			

HSC 2			
Mode	**Description**	**I0.4**	**I0.5**
0	Single phrase mode Internally change the counting value direction(increase /decrease) Direction change Bit:SM57.3		Clock
9	Phrase A and B counting mode	Clock Phrase B	Clock Phrase A

HSC 3			
Mode	**Description**	**I0.6**	**I0.7**
0	Single phrase mode Internally change the counting value direction(increase /decrease) Direction change Bit：SM127.3		Clock
9	Phrase A and B counting mode	Clock Phrase B	Clock Phrase A

5.2 Control Register and Status Register

- Control Register

In SM zone, each high speed counter has their own control registers for storing data. The current value register is used to change the current counting value. When the corresponding current value is set , the HSC will start to count from this value.

The following table describes the control registers

HSC0	HSC1	HSC2	HSC3	Description
SM37.0	SM47.0	SM57.0	SM127.0	Reset Signal Active Level：0=active high；1=active low
SM37.1	SM47.1	SM57.1	SM127.1	Start Signal Active Level：0=active high；1=active low
SM37.2	SM47.2	SM57.2	SM127.2	Phrase A and B counting rate ：0=1xrate；1=4xrate*
SM37.3	SM47.3	SM57.3	SM127.3	Counting Direction：0=Decrease；1=Increase
SM37.4	SM47.4	SM57.4	SM127.4	Whether write the update direction to HSC： 0=NO；1=YES
SM37.5	SM47.5	SM57.5	SM127.5	Whether write the update preset value to HSC：0=NO；1=YES
SM37.6	SM47.6	SM57.6	SM127.6	Whether write the update current value to HSC：0=NO；1=YES
SM37.7	SM47.7	SM57.7	SM127.7	Allow the HSC to work：0=disable；1=allow
HSC0	**HSC1**	**HSC2**	**HSC3**	**Description**
SMD38	SMD48	SMD58	SMD128	Current value
SMD42	SMD52	SMD62	SMD132	Preset Value

HSC0	HSC1	HSC2	HSC3	Description
SM141.0	SM151.0	SM161.0	SM171.0	Allow the HSC to use multiple segment preset value ：0=NO；1=YES
SM141.1	SM151.1	SM161.1	SM171.1	Preset value relative value or absolute value: 0 =Absolute；1=Relative
SM141.2	SM151.2	SM161.2	SM171.2	Preset Value comparison （ “CV=PV”)interrupts cyclically option: 0=NO；1=YES. Attention：cyclical interruptions only in relative value mode
SM141.3	SM151.3	SM161.3	SM171.3	Reserved
SM141.4	SM151.4	SM161.4	SM171.4	Whether update NO. of stages and preset values：0=NO；1=YES
SM141.5	SM151.5	SM161.5	SM171.5	Whether reset the interrupted variable ： 0=NO；1=YES
SM141.6	SM151.6	SM161.6	SM171.6	Reserved
SM141.7	SM151.7	SM161.7	SM171.7	Reserved
HSC0	**HSC1**	**HSC2**	**HSC2**	**Description**
SMW142	SMW152	SMW162	SMW172	Preset value start position （ It is offset corresponding to VB0)，it must be odd value.

Attention! In those control registers ,not all control bits are suitable for all working modes. For example , ‘counting direction ‘ and ‘ Whether write the update direction to HSC’ are only able to be used in mode 0 ,1 and 2(Single phrase mode externally change the counting value direction). If the external direction signal mode of HSC is used, then those 2 internal control bits will be ignored.

The default value of control word ,current counting value and preset value are 0.

- Status Register

Each single high speed counter has their status registers in SM zone . Those status registers are used to indicate the current status of high speed counter.

HSC0	HSC1	HSC2	HSC3	Description
SM36.0	SM46.0	SM56.0	SM126.0	Reserved
SM36.1	SM46.1	SM56.1	SM126.1	Reserved
SM36.2	SM46.2	SM56.2	SM126.2	Reserved
SM36.3	SM46.3	SM56.3	SM126.3	Is there any error in multiple segments PV table setting ：0=No，1=Yes
SM36.4	SM46.4	SM56.4	SM126.4	Reserved
SM36.5	SM46.5	SM56.5	SM126.5	Current counting direction：0=decrease；1=increase
SM36.6	SM46.6	SM56.6	SM126.6	Is the counting value equal to Preset value：0=No；1=Yes
SM36.7	SM46.7	SM56.7	SM126.7	Is the counting value greater than preset value：0=No；1=Yes
HSC0	HSC1	HSC2	HSC3	Description
SMB140	SMB150	SMB160	SMB170	Current operating Preset value No.（From 0）

5.3 Preset Value(PV) setting

KW allows each HSC has maximum 32 preset values, allows to select the PV to be relative value or absolute value and allow 'CV=PV'intterupt happens cyclically. Meanwhile , KW series PLC is compatible to PV setting method of the old product.

The example is give in the following . It describes the function and setting method of HSC0 PV values in detail.

- **How to select the mutiple stages of PV**

Each HSC has their own control register bit to enable the multiple preset value.

The control register bit of HSC0 is SM141.0.

If SM141.0 is 0, it means single segment PV value is adopted. The setting method is consistent with the method of K5. It only needs to set smd42 for the new PV value, and set sm37.5 whether to use this update PV value.

If SM141.0 is 1, it means multiple segment PV value is adopted. And SM37.5 and SMD42 are invalid. Each PV value is stored in the PV value table (smw142 is the starting address of the table), and sm141.4 indicates whether to use the data in the PV value table. If sm141.4 is 1, the data in PV value table will be used for high-speed counter after this startup. If sm141.4 is 0, it means that after this startup, the high-speed counter will use the last PV value data, and ignore the data in the PV value table.

- **Multiple Segment PV value Table**

If multiple segment PV values are used, the data in PV value table will be used for each PV value.

A control word is provided in the control register of each high-speed counter to store the starting address of the PV value table. **The starting address of the table must be the odd address in V area, for example, 301 (representing VB301)**

PV Table as below.

Byte Offset (1)	Data Type	Description
0	BYTE	NO. Of PV
1	DINT	The first PV value
5	DINT	The second PV value
…	DINT	…

（1） All offsets are the number of offset bytes relative to the starting position of the table.

（2） When the relative value mode is adopted, the mathematical absolute value of PV value must be greater than 1. Otherwise, PLC considers that the number of segments ends here, and counts the number of PV values (prior to the number set value);

When the absolute value mode is adopted, the mathematical absolute value of the difference between two adjacent PV values must be greater than 1, otherwise the PLC considers that the number of segments ends here, and counts the number of PV values based on it (prior to the number set value);

（3） When setting the PV value, users need to pay attention to that "CV = PV" interrupt must be generated in order. That means the count value reaches the first PV value and generates an interrupt , then the PLC will compare with the second PV value, and so on.

（4） PV value setting must be reasonable. Take the relative value for example, when increasing the count,

the PV value must be greater than 0, Otherwise the "CV = PV" interrupt may never be occurred. When decreasing the count, the PV value must be less than 0, otherwise the "CV = PV" interrupt corresponding to the value may never be generated.

- **Relative and Absolute Mode**

Each high speed counter has a control bit that is used to select control the PV value relative or absolute. This control bit of HSC0 is sm141.1.

If SM141.1 is 0, the PV value is absolute. When the count value is equal to the PV value, the corresponding "CV = PV" interrupt will be generated. For example, if three PV values are set, 1000, 2000 and 3000 in sequence, the first "CV = PV" interrupt will be generated when the count value reaches 1000, the second "CV = PV" interrupt will be generated when the count value reaches 2000 and so on.

If SM141.1 is 1, it means that PV value is relative. If the counter is based on the current count value and continues counting so that the difference is equal to PV value, the corresponding "CV = PV" interrupt will be generated. For example, if three PV values are set, namely 10, 1000 and 1000 respectively, and the count value is 100 when the high-speed counter is started, when the count value reaches 110, 1110 and 2110 respectively, "CV = PV" interrupt will be generated.

- **"CV=PV" Cyclical Interrupt Generation**

Only when PV value in relative value mode allows cyclical interrupt setting.Otherwise it is invalid.

If SM141.0 is 0, "CV = PV" interrupt is generated only once. When all interrupts corresponding to PV value are completed, it will stop. To continue generation, you must modify the corresponding register value and call the HSC instruction again.

If sm141.0 is 1, it means that "CV = PV" interrupt will be generated cyclically. When the interrupt corresponding to the last PV value is completed, the PLC will add each PV value again based on the current count value to get the new interrupt value, and then continue to compare with the count value to generate the corresponding "CV = PV" interrupt. This process will continue to cycle and never stop.

For example, if three PV values are set, 10, 1000 and 1000 respectively, and the count value is 100 when the high-speed counter is started, the required values for each interrupt are as follows

Current value	No. Of Interrupt	The 1st Value	The 2nd Value	The 3rd Value
100	The fist time	110	1110	2110
2110	The second time	2120	3120	4120
4120	The third time	4130	5130	6130
…	The n time	…	…	…

5.4 "CV=PV" Interrupt Numbers

When the single PV value mode is selected, the high-speed counter is fully compatible with K5, including the number of "CV = PV" interrupt consistent with K5.

When the multiple segment PV value mode is adopted, the high speed counter assigns a new interrupt number to all 32 PV values, as shown in the following table.

High Speed Counter	Interrupt Event No.	Description
HSC0	64	The first PV value "CV=PV"interrupt
	65	The second PV value "CV=PV"interrupt

	…	…（+1 in sequence)
	95	The 32nd PV value “CV=PV”interrupt
HSC1	96	The first PV value “CV=PV”interrupt
	97	The second PV value “CV=PV”interrupt
	…	…（+1 in sequence)
	127	The 32nd PV value “CV=PV”interrupt
HSC2	128	The first PV value “CV=PV”interrupt
	129	The second PV value “CV=PV”interrupt
	…	…（+1 in sequence)
	159	The 32nd PV value “CV=PV”interrupt
HSC3	160	The first PV value “CV=PV”interrupt
	161	The second PV value “CV=PV”interrupt
	…	…（+1 in sequence)
	191	The 32nd PV value “CV=PV”interrupt

5.5 The Methods to use the High Speed Counter

- **Method 1: Programmed by the relevant instruction**

This method is also used in K3 and K5. KW also supports this method. The steps are as below:

1). configure the control byte of the high-speed counter, and specify the current value (i.e. the starting value of the count) and preset value;

2). use HDEF instruction to define a high-speed counter and its working mode;

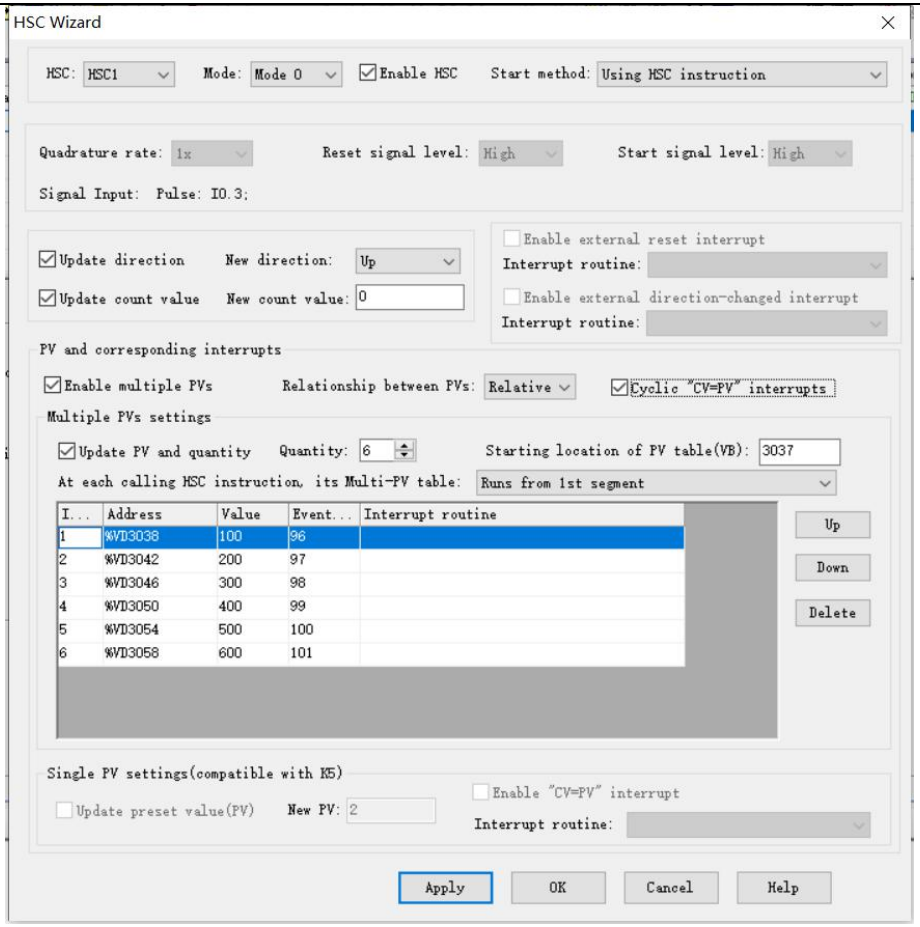
3). (optional) use the ATCH command to disconnect the corresponding interrupt service program for the high-speed counter;

4). use HSC instruction to configure and start high-speed counter

- **Method 2 :HSC Wizard**

In KW, it provides a configuration wizard for high-speed counter. Users can directly use the wizard to configure all the high-speed counters without complicated programming. The wizard is shown below.

Even if HSC is configured through the wizard, users can modify, start and stop the parameters of high-speed counter at any time according to "method 1" in the program



The HSC wizard is used as follows:

1) in 【Counter】, select the counter to be used.
2) select 【Enable this counter 】and subsequent configuration will be allowed.
3) 【Mode】, select the counter mode to be used.
4) in 【start mode】, select the start mode of the high-speed counter.

There are two ways to start:

"Call the HSC instruction in the program": if you choose this way, call the HSC instruction to start the counter in the user program. Before calling HSC instruction, there is no need to configure each register and call HDEF instruction.

"Start directly after PLC power on": if this mode is selected, the high-speed counter will run automatically after PLC power on, without calling any instructions.

5) to use multi segment PV value mode, select enable multi preset value (PV) function, and then configure PV value, quantity, and associated interrupt subroutines. If you select Modify PV value and quantity, you can adjust the value in PV value quantity to modify the number of PV values.

6) to use the single PV value mode, first select [modify PV value] in "single PV value setting (compatible

with K5)", then you can modify the PV value and the associated interrupt subroutine.

7) for other configuration items, please refer to the previous description and configure according to the actual requirements.

Chapter 6 How to Use the Functions of High Pulse Output

The KW standard CPU provides two channels of high-speed output, with channels of Q0.0 and Q0.1 respectively, which support PTO (pulse train) and PWM (pulse width modulation) mode output. The maximum output frequency can reach 200kHz (the load resistance is required to be no more than 3KΩ)

KW instruction set provides the following three instructions for high-speed output function:

1) positioning control instructions: there are 5 instructions in total, including PREL (relative motion), PABS (absolute motion), PHOME(return to origin), PJOG(inching), and PSTOP (emergency stop) instructions. Users can easily achieve simple positioning control functions. **Note: when using the positioning control command, the output pulse frequency cannot be lower than 80Hz!**

2)PLS command: PTO (single-stage or multi-stage) and PWM output functions can be achieved.

3) Tracking instruction PFLO_: this instruction can make the frequency of output pulse change with the frequency of input. When the number of output pulses reaches the number specified by the user, the output will be stopped and the completion flag bit will be set. **Note: when using follow command, the output pulse frequency cannot be lower than 30Hz!**

Note: if the DO channel is of relay type, avoid using high-speed output function!

6.1 Motor Direction Control Signal

For the positioning control command, PLC specifies a direction output channel for each high-speed output, and also provides a direction prohibition control bit in SM area to prohibit or allow the use of the corresponding direction output channel.As the following table

	Q0.0	**Q0.1**
Direction Output Channel	Q0.2	Q0.3
Direction enable bit	SM201.3	SM231.3

The direction output channel is used to output the direction control signal of the motor. The output is 0 when rotating forward and 1 when reversing.

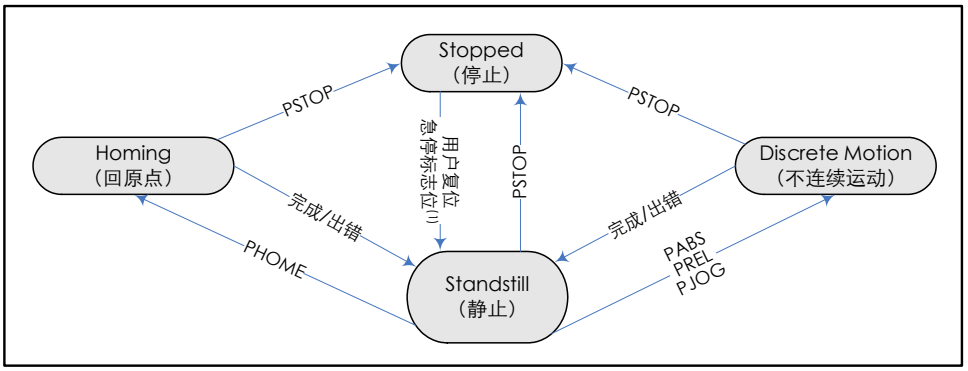
The direction inhibit control bit is used to disable or enable the corresponding direction output channel. If it is set to disable, the positioning control instruction will not output the direction control signal, and the corresponding direction output channel can be used as a common D0.

6.2 Positioning Control Instruction

6.2.1 Positioning Control Model

The following figure is the model diagram of positioning control. Users can learn the following information from the figure: when the positioning control command is executed, the status of the corresponding

high-speed output channel; in each state, the positioning control command that PLC allows to execute.



(1) The Estop flags are SM201.7/ SM231.7. When PSTOD is executed ， this bit is set .

6.2.2 Control Register and Status Register

For the positioning control command, PLC allocates a control byte for each high-speed output in SM zone. Users need to pay attention to set the control byte in the program. In addition, a current value (Dint type) register is allocated. The following table.

Q0.0	**Q0.1**	**Description**
SMD212	SMD242	Read Only.Current Value represents the current numbers of pulses has been outputed .(CW increases，CCW decrease）
SMD208	SMD238	Read Only.New Current Value.It is matching with flag bit to modify current value
Q0.0	**Q0.1**	**Description**
SM201.7	SM231.7	Read and Write.Estop Flag If this bit is 1, it means it is in emergency stop state and no positioning control command is executed. This bit will automatically set to 1 when the PSTOP command is executed. The user needs to use the program to clear this bit.
SM201.6	SM231.6	Read and write .Used to decide whether reset the current value. 1 - Set the current value to be 0 0 - the current value is maintained
SM201.4	SM231.4	Read and write .Used to decide whether change the current value. 1 - Set the current value to be 0 0 - the current value is maintained
SM201.3	SM231.3	Direction Control Enable Bit 1- Inhabit direction output,Direction Channel as Normal D0 0 - Enable Direction Output
Other Bit	Other bit	Reserved

➢ **How to modify the current value**

The current value registers (smd212 and smd242) contain the number of pulses that have been output by the corresponding channel.

The current value register is read-only and cannot be assigned directly in the user program. In order to

facilitate the user to modify the current value, PLC provides several methods, which can be used flexibly in the program. Note: please avoid the reset operation during the movement (including the execution of PHOME, PREP, PABS, PJOG, PFLO_F instructions), so as to avoid the error of current value count.

The following example programs are all in IL format. In KincoBuilder, the user creates or opens a new program, and then executes the menu command [Engineering] - > IL language], which will enter the format of IL editor, then copies and pastes the sample program into IL editor, and then executes the menu command [LD language], which can be converted to LD program.

- **Method 1**

Use reset control bits (sm201.6 and sm231.6) to clear the current value to 0.

As long as the reset control bit is "1", the PLC will clear the corresponding current value register. Therefore, the reset control bit only needs to be kept for one scanning cycle ,and avoid keeping "1" for a long time.

Example given as below is to demonstrate how to clear the current value

```
(* Network 0 *)
(*Based on the origin. When it is moved to origin，it is required to clear and set to be 0.*)
LD          %SM0.0
PHOME
0, %M0.0, %M0.1, %M0.2, %VW0, %VW2, %VW4, %VD6, %VW10, %M0.4, %M0.5, %MB1
(* Network 1 *)
(*PHOME completed，use DONE flag to clear the current value and set to be 0.*)
LD          %M0.4
R_TRIG
ST          %SM201.6
```

- **Method 2**

Use the following registers. It may change the current value to be any values.

Q0.0	Q0.1	Description
SMD208	SMD238	Read and Write.New Current Value.It is matching with flag bit to modify current value
SM201.4	SM231.4	Read and write .Used to decide whether change the current value. 1 - Set the current value to be 0 0 - the current value is maintained

Take channel 0 as an example to illustrate the use method: if sm201.4 is 0, the current value smd212 will remain unchanged. If sm201.4 is 1, the value in smd208 is assigned to the current value SMD212.

```
(* Network 0 *)
(*Based on the origin. When it is moved to origin，it is required to    set to be 100.*)
LD          %SM0.0
PHOME
0, %M0.0, %M0.1, %M0.2, %VW0, %VW2, %VW4, %VD6, %VW10, %M0.4, %M0.5, %MB1
(* Network 1 *)
(*PHOME completed，use DONE flag to modify the current value *)
LD          %M0.4
R_TRIG
MOVE        DI#100, %SMD208
ST          %SM201.
```

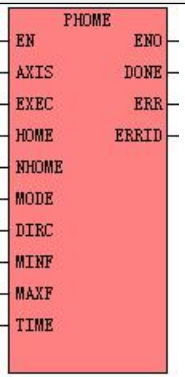
6.2.3 Error Code

When the positioning control instruction is executed, it may generate non fatal errors. At this time, the CPU will generate an error code and output it to the ERRID parameter of the instruction. The following table lists these error codes and their descriptions.

Error Code	Description
0	No Error
1	The acceleration / deceleration time is too short or the initial speed is too low, resulting in the initial pulse period exceeding each period of time.
2	The initial speed minf exceeds the maximum allowable speed (200kHz)
3	The initial speed minf is lower than the minimum allowable speed (80Hz)
4	The number of pulses required for initial acceleration and deceleration exceeds the total number of pulses
5	Initial speed minf exceeds max speed MAXF

6.2.4 PHOME（Homing）

➢ Instructions and operands

	Name	Instruction Format	Influence CR value	Available in
LD	PHOME			☑ K5 ☑ K2 ☑ KS ☑ KW
IL	PHOME	PHOME *AXIS, EXEC, HOME, NHOME, MODE, DIRC, MINF, MAXF, TIME, DONE, ERR, ERRID*		

Parameter	Input/Output	Data Type	Allowable memory type
AXIS	Input	INT	Constant
EXEC	Input	BOOL	I、Q、V、M、L、SM、RS、SR
HOME	Input	BOOL	I 、Q、V、M、L、SM、RS、SR
NHOME	Input	BOOL	I 、Q、V、M、L、SM、RS、SR
MODE	Input	INT	I、Q、V、M、L、SM、T、C、AI、AQ、Constant
DIRC	Input	INT	I、Q、V、M、L、SM、T、C、AI、AQ、Constant

MINF	Input	WORD	I、Q、M、V、L、SM、Constant
MAXF	Input	DWORD	I、Q、M、V、L、SM、Constant
TIME	Input	WORD	I、Q、M、V、L、SM、Constant
DONE	Output	BOOL	Q、M、V、L、SM
ERR	Output	BOOL	Q、M、V、L、SM
ERRID	Output	BYTE	Q、M、V、L、SM

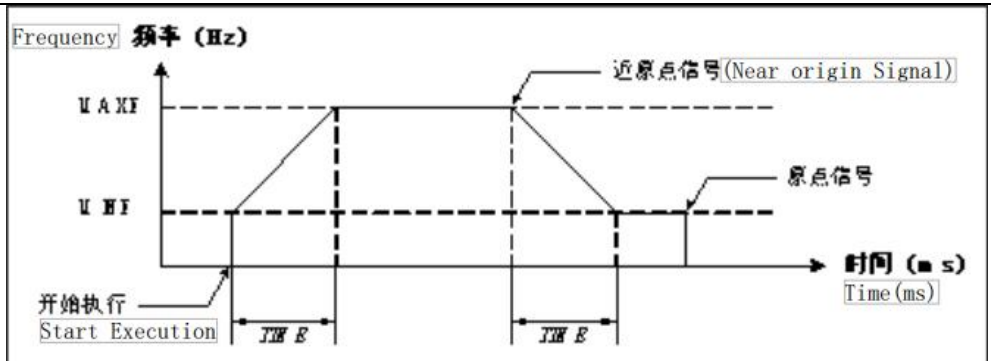
MODE,DIRC,MINF,MAXF,TIME must be all in the same data type (all in constant or all in internal memory type)at the same time.

The following table describes the function of each parameter in detail.

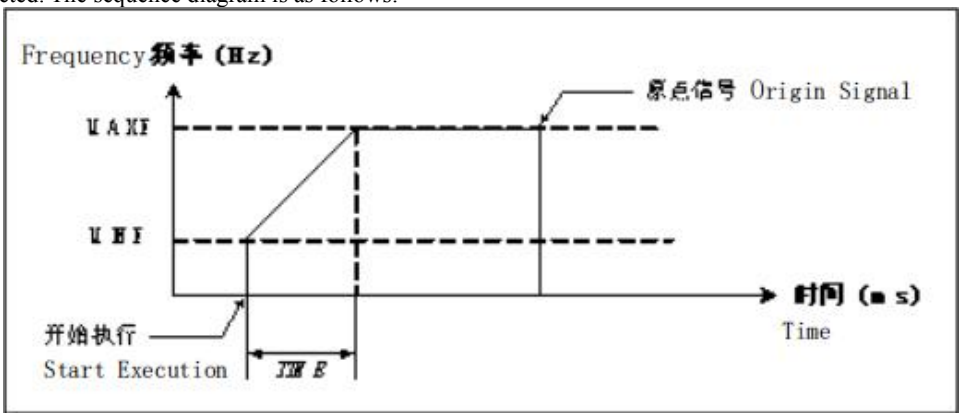
Parameters	Description
AXIS	High speed output channel used. 0 means Q0.0, 1 means Q0.1, and 2 means Q0.4.
EXEC	If the rising edge jump of exec is detected, the PHOME instruction is triggered.
HOME	Origin input signal
NHOME	Near origin input signal
MODE	Returning origin control method: 0 use origin and near origin input to control;1 use origin input to control;
DIRC	Motor rotational direction：0 CW；1 CCW Regards to motor direction control signals refers to 6.1 Motor Direction Control Signal
MINF	The initial speed (i.e. initial frequency) of the output pulse, in Hz. Minf is not allowed to be lower than 125Hz or higher than MAXF.
MAXF	The highest speed (i.e. the highest frequency) of the output pulse, in Hz.The allowable range of MAXF is 125Hz ~ 200kHz. MAXF must be greater than or equal to MINF
TIME	Acceleration / deceleration time, unit: ms. This command uses the same acceleration time and deceleration time. Acceleration time is the time required for MINF to accelerate to MAXF, and deceleration time is the time required for MAXF to decelerate to MINF.
DONE	Completion flag bit. When the normal execution of the instruction is completed, done jumps from 0 to 1.
ERR	Error flag bit. If an error occurs during the execution of an instruction, the flag bit is set to 1.
ERRID	Error Code

PHOME command uses the near origin and origin input signals to control the return to the origin. The parameter mode defines the control mode:

1) If the near origin signal and the origin signal are used for control, when the near origin signal is detected, it starts to slow down, and when the origin signal is detected, it stops the pulse output. The sequence diagram is as follows:



2). If only the origin signal is used for control, the pulse output will be stopped when the origin signal is detected. The sequence diagram is as follows:



When the PHOME command is executed and the direction enable control bit is set to 0, the direction control signal will be output in the corresponding direction. if the DIRC is set to positive rotation, the direction channel will output the positive rotation signal, and the current value will increase. if the dirc is set to reverse, the direction channel will output the reverse signal, and the current value will decrease.

After PHOME instruction is operated in highest frequency before returning to the origin and generating the origin signal, it will read the MAXF value in real time, and automatically calculate the number of acceleration or deceleration segments according to the new frequency value. Then accelerate or decelerate to the new frequency value and maintain the uniform output.

It should be noted that the current value register (SMD212 / SM242) does not automatically clear after the return to the origin movement is completed, and the user needs to change the current value according to the actual requirements.

- LD

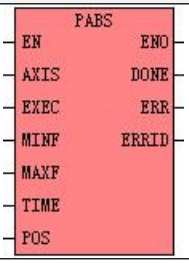
When en is 1, if the rising edge of EXEC input is detected, the command is triggered.

- IL

When the CR value is 1, if the rising edge of exec input is detected, the command is triggered. The execution of this instruction does not affect the CR value. If only the origin signal is used for control, the pulse output will be stopped when the origin signal is detected. The sequence diagram is as follows:

6.2.5 PABS（Absolute Motion）

➢ Instructions and operands

	Name	Instruction Format	Influence CR value	Available in
LD	PABS			☑ K5 ☑ K2 ☑ KS ☑ KW
IL	PABS	PABS AXIS, EXEC, MINF, MAXF, TIME, POS, DONE, ERR, ERRID		

Parameter	Input/Output	Data Type	Influence CR value
AXIS	Input	INT	Constant
EXEC	Input	BOOL	I、Q、V、M、L、SM、RS、SR
MINF	Input	WORD	I、Q、M、V、L、SM、Constant
MAXF	Input	DWORD	I、Q、M、V、L、SM、Constant
TIME	Input	WORD	I、Q、M、V、L、SM、Constant
POS	Input	DINT	I、Q、M、V、L、SM、HC、Constant
DONE	Output	BOOL	Q、M、V、L、SM
ERR	Output	BOOL	Q、M、V、L、SM
ERRID	Output	BYTE	Q、M、V、L、SM

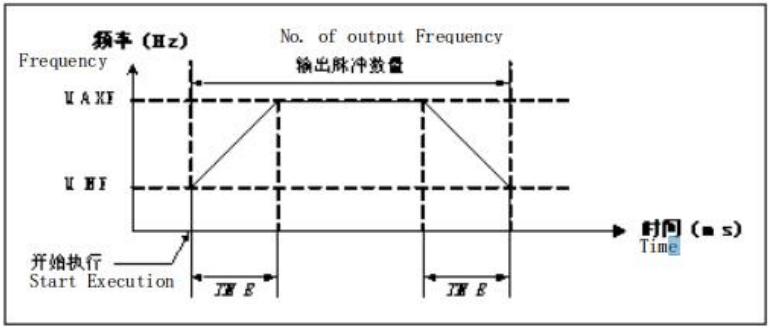
MINF,MAXF,TIME,POS must be all in the same data type (all in constant or all in internal memory type)at the same time.

The following table describes the function of each parameter in detail.

Parameter	Description
AXIS	High speed output channel used. 0 means Q0.0, 1 means Q0.1, and 2 means Q0.4.
EXEC	If the rising edge jump of exec is detected, the PABS instruction is triggered.
MINF	The initial speed (i.e. initial frequency) of the output pulse, in Hz. Minf is not allowed to be lower than 125Hz or higher than MAXF.
MAXF	The highest speed (i.e. the highest frequency) of the output pulse, in Hz.The allowable range of MAXF is 125Hz ~ 200kHz. MAXF must be greater than or equal to MINF
TIME	Acceleration / deceleration time, unit: Ms. This command uses the same acceleration time and deceleration time. Acceleration time is the time required for MINF to accelerate to MAXF, and deceleration time is the time required for MAXF to decelerate to MINF.
POS	The target value.the value to be reached by the current value register.

DONE	Completion flag bit. When the normal execution of the instruction is completed, done jumps from 0 to 1.
ERR	Error flag bit. If an error occurs during the execution of an instruction, the flag bit is set to 1.
ERRID	Error Code

PABS instruction uses absolute positioning . The outputs pulses is based on the difference between the current value and the target value POS. The difference between the current value and the target value is the number of output pulses. The sequence diagram of PABS instruction execution is as follows:



When PABS instruction is executed and the direction enable control bit is set to 0, the PAbs instruction will output the direction control signal of the motor in the corresponding direction output channel. When the target value is greater than the current value, the output forward signal, the current value will increase. When the target value is less than the current value, the output reverse signal, the current value will decrease.

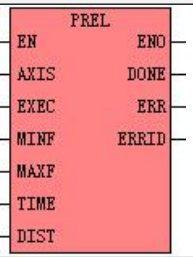
- LD

  When en is 1, if the rising edge of exec input is detected, the command is triggered.
- IL

  When the CR value is 1, if the rising edge of exec input is detected, the command is triggered. The execution of this instruction does not affect the CR value.

6.2.6 PREL（Relative Motion ）

➢ Instructions and operands

	Name	Instruction Format	Influence CR value	Available in
LD	PREL			☑ K5 ☑ K2 ☑ KS ☑ KW
IL	PREL	PREL AXIS, EXEC, MINF, MAXF, TIME, DIST, DONE, ERR, ERRID		

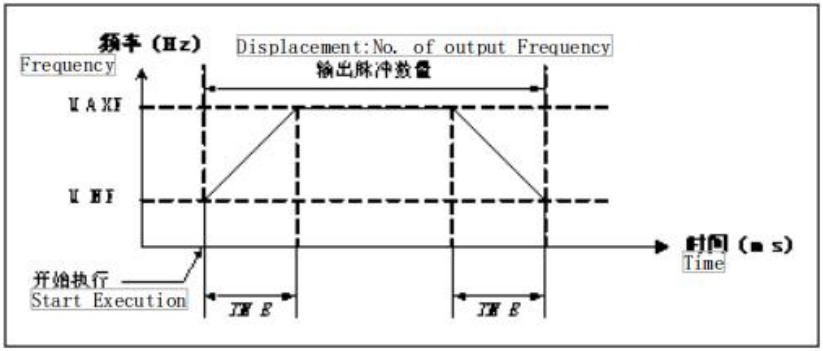
Parameter	Input/Output	Data Type	Influence CR value
AXIS	Input	INT	Constant
EXEC	Input	BOOL	I、Q、V、M、L、SM、RS、SR
MINF	Input	WORD	I、Q、M、V、L、SM、Constant
MAXF	Input	DWORD	I、Q、M、V、L、SM、Constant
TIME	Input	WORD	I、Q、M、V、L、SM、Constant
DIST	Input	DINT	I、Q、M、V、L、SM、HC、Constant
DONE	Output	BOOL	Q、M、V、L、SM
ERR	Output	BOOL	Q、M、V、L、SM
ERRID	Output	BYTE	Q、M、V、L、SM

MINF, MAXF, TIME, DIST must be all in the same data type (all in constant or all in internal memory type)at the same time.

The following table describes the function of each parameter in detail.

Parameter	Description
AXIS	High speed output channel used. 0 means Q0.0, 1 means Q0.1, and 2 means Q0.4.
EXEC	If the rising edge jump of exec is detected, the PABS instruction is triggered.
MINF	The initial speed (i.e. initial frequency) of the output pulse, in Hz. Minf is not allowed to be lower than 125Hz or higher than MAXF.
MAXF	The highest speed (i.e. the highest frequency) of the output pulse, in Hz.The allowable range of MAXF is 125Hz ~ 200kHz. MAXF must be greater than or equal to MINF
TIME	Acceleration / deceleration time, unit: Ms. This command uses the same acceleration time and deceleration time. Acceleration time is the time required for MINF to accelerate to MAXF, and deceleration time is the time required for MAXF to decelerate to MINF.
DIST	Displacement. that is number of pulses to be output.
DONE	Completion flag bit. When the normal execution of the instruction is completed, done jumps from 0 to 1.
ERR	Error flag bit. If an error occurs during the execution of an instruction, the flag bit is set to 1.
ERRID	Error Code

PREL instruction uses relative positioning. The number of output pulses is the displacement DIST. The PREL instruction sequence diagram of the prior instruction execution is as follows



When PREL command is executed and the direction enable control bit is set to 0, PREL command will output the direction control signal of the motor in the corresponding direction output channel. When the target value is greater than the current value, the output forward signal, the current value will increase. When the target value is less than the current value, the output reverse signal, the current value will decrease.

- LD

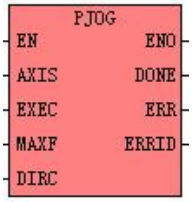
When EN is 1, if the rising edge of EXEC input is detected, the command is triggered.

- IL

When the CR value is 1, if the rising edge of EXEC input is detected, the command is triggered. The execution of this instruction does not affect the CR value.

6.2.7 PJOG（Inching）

➢ Instruction and operands

	Name	Instruction Format	Influence CR value	Available in
LD	PJOG			☑ K5 ☑ K2 ☑ KS ☑ KW
IL	PJOG	PJOG AXIS, EXEC, MINF, DIRC, DONE, ERR, ERRID		

Parameter	Input/Output	Data Type	Allowable memory type
AXIS	Input	INT	Constant（0 or 1）
EXEC	Input	BOOL	I、Q、V、M、L、SM、RS、SR
MAXF	Input	DWORD	I、Q、M、V、L、SM、Constant
DIRC	Input	INT	I、Q、V、M、L、SM、T、C、AI、AQ、Constant
DONE	Output	BOOL	Q、M、V、L、SM
ERR	Output	BOOL	Q、M、V、L、SM
ERRID	Output	BYTE	Q、M、V、L、SM

MAXF, DIRC must be all in the same data type (all in constant or all in internal memory type)at the same time.

The following table describes the function of each parameter in detail.

Parameter	Description
AXIS	High speed output channel used. 0 means Q0.0, 1 means Q0.1, and 2 means Q0.4
EXEC	If EXEC is 1,it will continuously output pulses;If it is 0, the output stops
MINF	The initial speed (i.e. initial frequency) of the output pulse, in Hz. Minf is not allowed to be lower than 125Hz.

DIRC	Motor rotational direction：0 CW；1 CCW Regards to motor direction control signals refers to 6.1 Motor Direction Control Signal
DONE	Completion flag bit. When the normal execution of the instruction is completed, done jumps from 0 to 1.
ERR	Error flag bit. If an error occurs during the execution of an instruction, the flag bit is set to 1.
ERRID	Error Code

If exec input is 1, PJOG instruction continuously outputs pulse train with frequency of MAXF from specified channel AXIS. During the execution, the PJOG instruction will read the input frequency parameter (MAXF) value in real time and adjust the frequency of the output pulse according to the new frequency value. If EXEC input is 0, the output is stopped immediately.

When the PJOG instruction is executed and the direction enable control bit is set to 0, then the direction control signal will be output in the corresponding direction. If the DIRC is set to forward, then the direction channel will output the forward signal, and the current value will increase. If the DIRC is set to reverse, then the direction channel will output the reverse signal, and the current value will decrease.

- LD

When en is 1, if exec input is 1, the instruction will be executed; if exec input is 0, the execution will be stopped immediately.

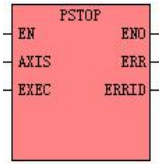
- IL

When CR value is 1, if exec input is 1, the instruction is executed.

The execution of this instruction does not affect the CR value. IL

6.2.8 PSTOP（Stop）

➢ Description

	Name	Usage	Group	Used for
LD	PSTOP			☑ K5 ☑ K2 ☑ KS ☑ KW
IL	PSTOP	PSTOP AXIS, EXEC, ERR, ERRID		

Operands	Input/Output	Data type	Acceptable Memory Areas
AXIS	Input	INT	Constant
EXEC	Input	BOOL	I、Q、V、M、L、SM、RS、SR
ERR	Output	BOOL	Q、M、V、L、SM
ERRID	Output	BYTE	Q、M、V、L、SM

The following table describes all the operands in detail.

Operands	Description
AXIS	The high-speed output channel, 0 means Q0.0, 1 means Q0.1, and 2 means Q0.4
EXEC	If EN is 1,the EXEC would stop current motion on the rising edge.
ERR	Indicates that error has occurred during the execution. 0= no error,1= an error has occurred .
ERRID	Error identification.

The PSTOP instruction is used to stop the pulse output of the AXIS channel immediately, causing the current movement to come to an emergency stop and at the same time set the Emergency- stop flag is set to 1.

Users need to clear the Emergency-stop fage to 0 , otherwise the CPU will not execute any positioning control instructions.

- LD

When *EN* is 1,If the *EXEC* detected rising edge, the instruction is triggered .

- IL

When CR is 1, if of EXEC detect the rising edge , the instruction is triggered.

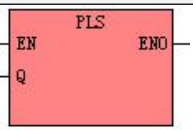
The execution of this instruction does not affect the CR value.

6.3 PLS Instruction

PLS instruction can realize PTO or PWM output function.

- PTO：Pulse Train Output
- PWM：Pulse-Width Modulation

➢ Description

	Name	Usage	Group	Used for
LD	PLS			☑ K5 ☑ K2 ☑ KS ☑ KW
IL	PLS	PLS *Q*	U	

Operand	Intput/Output	Data type	Acceptable Memory area
Q	Input	INT	Constant(0 ,1 or 2)

The PLS instruction reads the values of the corresponding control registers in the SM region and configure the characteristics of the high-speed pulse output, then starts the high-speed pulse output until the specified pulse output function is completed.The pulse output channel is specified by the parameter Q, 0 means output with Q0.0, and 1 means output with Q0.1.

Note: in user programs, PLS instruction can be executed only once when needed. It is recommended to use the output of edge instruction to invoke PLS instruction.If the EN end of PLS is kept at 1, the PLS

instruction will not output normally.

- **LD**

If *EN* is 1，PLS instruction is executed.

- **IL**

If CR is 1, PLS instruction is executed.

The execution of this instruction does not affect the CR value.

6.3.1 PWM & PTO

➢ **PWM**

PWM function means the Pulse-Width Modulation(PWM).The user can control the cycle and pulse width of the output.

Cycle and pulse width can be measured in microseconds (μ s) or milliseconds (ms), with a maximum cycle value of 65535.When the pulse width is equal to the cycle, the duty cycle is automatically set to 100% and the output is always switched on.When the pulse width is 0, the duty cycle is 0% and the output is disconnected.

➢ **PTO**

The PTO function is able to generate a specific number of pulses in a square pulse train (50% duty cycle).

The user can control the period of the output square wave and the number of output pulses.

The pulse cycle is measured in microseconds (mus) or milliseconds (ms), with a maximum period of 65535.The number of pulses ranges from 2 to 4,294,967,295.If you specify a pulse number less than 2, KW sets the appropriate error flag bit and disables output.

The PTO function provides two modes of single-segment operation and multi-segment operation.

- **Single-stage operation**

Single-stage operation mode means that PLS instruction outputs only one pulse train at a time, i.e. a set of pulses with the same frequency.

- **Multi-stage Operation**

Multi-stage operation mode means that PLS instruction will output multiple stage pulse train in turn each time.

The preset values of each segment are stored in the envelop table of the v-region, and each segment occupies 8 bytes in the envelop table, including a periodic value (16-bit unsigned integer), a reserved value (temporarily unused, 16-bit signed integer) and an impulse value (32-bit unsigned double integer). In other words, all the pulses have the same output frequency in the same segment. Multistage operations are configured and started using PLS instructions.

The initial position of the envelope table is stored in SMW168 (corresponding to PTO0) and SMW178 (corresponding to PTO1), and the time base can be selected in microseconds or milliseconds through setting of SM67.3 (corresponding to PTO0) and SM77.3 (corresponding to PTO1). All periodic values in the envelope table must use the same time base and cannot be changed during envelope execution.

Envelope table format are as follow:

Offset (1)	Length	Segments	Description
0	8bit		Number of segments (1 to 64)

1	16bit	1st.	Initial period (2 to 65535 time basis)
3	16bit		None
5	32bit		Number of pulses (1 to 4,294,967,295)
9	16bit	2nd.	Initial period (2 to 65535 time basis)
11	16bit		None
13	32bit		Number of pulses (1 to 4,294,967,295)
…		…	…

（**1**）All offsets are the bytes numbers relative to the starting position of the envelope table.

Note: The starting position of the envelope table must be an odd number of addresses in the V region，E.g:VB3001.

6.3.2 PTO/PWM register

Control registers are provided in the SM area for each PTO/PWM generator to store its configuration data.As following table shown:

Q0.0	**Q0.1**	**Description**	
SM67.0	SM77.0	PTO/PWM	Whether to update cycle:0=No,1=Yes
SM67.1	SM77.1	PWM	Whether to update pulse width:0=No,1=Yes
SM67.2	SM77.2	PTO	Whether to update pulse number:0=No,1=Yes
SM67.3	SM77.3	PTO/PWM	Time base：0=1μs；1=1ms
SM67.4	SM77.4	PWM	Update method: 0= asynchronous update; 1= synchronous update
SM67.5	SM77.5	PTO	Operation mode: 0= single segment operation; 1= multistage operation
SM67.6	SM77.6	Function selection: 0= PTO;1 = PWM	
SM67.7	SM77.7	PTO/PWM	Enable or disable this function: 0= disable; 1 = enable
Q0.0	**Q0.1**	**Description**	
SMW68	SMW78	PTO/PWM	Cycle Value,Range:2~65535
SMW70	SMW80	PWM	Pulse Width,Range:0~65535
SMD72	SMD82	PTO	Pulse Number,Range:1~4,294,967,295
SMW168	SMW178	The starting position of the envelope table(represented by a byte offset from VB0), a PTO multisegment operation.	

The default value for all control bytes, cycles, and pulses is 0.

The method for users to modify the characteristics of PTO/PWM waveform is: first set the corresponding control register, if it is a PTO multi-stage operation, the envelop table must be set first, and then execute the PLS instruction.

In the SM area, a status byte is provided for each PTO/PWM generator. The user can access the status byte to understand the current status information of the PTO/PWM generator.

The following table.

Q0.0	Q0.1	Description
SM66.0	SM76.0	None
SM66.1	SM76.1	None
SM66.2	SM76.2	None
SM66.3	SM76.3	Whether PWM is idle: 0= No;1 = Yes
SM66.4	SM76.4	Is there any error in setting PTO cycle value and pulse number: 0= No;1 = Yes Note: the value of period and pulse must be greater than 1.
SM66.5	SM76.5	Whether PTO is terminated by user command: 0= No;1 = Yes
SM66.6	SM76.6	None
SM66.7	SM76.7	Whether PTO is idle:0=No,1=Yes

PTO idle bit and PWM idle bit indicate whether PTO output and PWM output have finished.

6.3.3 Using PTO Function

The following is an example of how to program PTO with PTO0.

In general, using a PTO involves two steps: Setting up the relevant control register and initializing the PTO;Execute PLS instruction.

Suggested that the user should write a separate initialization subroutine as much as possible in the project, which can make the whole project have a good structure. In addition, if possible, try to call the initialization subroutine in the main program with SM0.1 as the condition, so that the subroutine will only be called and executed once in the first scan after the CPU is powered up, which can reduce the CPU scan time.

- **Execute PTO（Single-stage operation）**

1) Set the control byte SMB67 based on the expected operation.

For example, SMB67 = B#16#85 indicates：

- Allow PTO/PWM function;
- Select to use PTO function, Single-stage operation；
- The time base is chosen as 1μs;
- Allow update pulse number and cycle.

2) Assign the expected period value to SMW68.
3) Assign the expected number of pulses to SMD72.
4) (optional) Add an interrupt program to the "PTO0 completed" interrupt event (event no. 27) with the ATCH instruction to achieve a quick response to the interrupt event.
5) Execute PLS instructions to configure and start PTO0.

- **Change the PTO cycle（Single-stage）**

Follow these steps to change the PTO0 cycle value:

1) Set the control byte SMB67 based on the expected operation:

For example, SMB67 = B#16#81 indicates：

- Allows PTO/PWM function；
- Select to use PTO function, Single-stage operation；
- The time base is chosen as 1μs；
- Allow update pulse number and cycle.

2) Assign the expected period value to SMW68.
3) Execute PLS instructions to configure and start PTO0, and PTO with new cycle values start immediately.

➢ **Change the number of PTO pulses（Single-Stage）**

Following step:

1) Set the control byte SMB67 based on the desired operation:
   For example, SMB67 = B#16#84 indicates:
- Allows PTO/PWM function;
- Select to use PTO function, Single-stage operation；
- The time base is chosen as 1μs；
- Allow update pulse number.

2) Assign the expected number of pulses to SMD72.
3) Executing PLS instructions to configure and start PTO0 will immediately follow with a new specified number of pulses.

➢ **Execute PTO（Multi-stage）**

1) Set the control byte SMB67 based on the desired operation.
   For example, SMB67 = B#16#A0 indicates:
- Allows PTO/PWM function;
- Select to use PTO function
- Select to use Multi-stage operation;
- The time base is chosen as 1μs；

2) Assign the start position of the envelop table (odd, indicating the byte offset of the start address of the envelop table from VB0) to SMW168.
3) Set the relevant values in the envelope table.
4) (optional) connect an interrupt service program to the "PTO0 completed" interrupt event (event no. 27) with the ATCH instruction to achieve a quick response to the interrupt event.
5) Execute PLS instructions to configure and start PTO0.

6.3.4 Using PWM function

Take PWM0 as an example to show how to program to use the PWM function.

Generally speaking, using PWM includes two steps: Setting up the relevant control register; Execute PLS instruction.

Suggested that the user should write a separate initialization subroutine as much as possible in the project, which can make the whole project have a good structure. In addition, if possible, try to call the initialization subroutine in the main program with SM0.1 as the condition, so that the subroutine will only be called and executed once in the first scan after the CPU is powered up, which can reduce the CPU scan time.

➢ **Execute PWM**

1) Set the control byte SMB67 based on the desired operation.
   For example, SMB67 = B#16#D3 indicates:
- Allows PTO/PWM function;
- Choose to use PWM function;
- Select the synchronous update mode;
- The time base is chosen as1μs;
- Allows update of pulse width and cycle.

2) Assign the expected cycle value to SMW68.
3) Assign the expected pulse width to SMW70.
4) Execute the PLS instruction to configure and start PWM0.

➢ **Change the pulse width**

How to change the pulse width of PWM0 is described below.

1) Set the control byte SMB67 based on the desired operation.

For example, SMB67 = B#16#D2 indicates：

- Allows PTO/PWM function;
- Choose to use PWM function;
- Select the synchronous update mode;
- The time base is chosen as 1μs；
- Allows pulse width values to be updated.

2) Assign the expected pulse width to SMW70.
3) Execute the PLS instruction to configure and start PWM0.

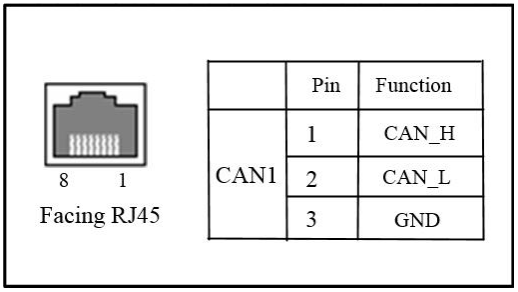
Chapter 7 CAN bus communication port

KW standard CPU module provides a CAN communication port, named CAN1.

CAN communication port support Extended Bus protocol,CANopen Master and slave protocol,Kinco motion control protocol,and free communication.Among them,The free communication could work with other protocol at the same time.While the other protocol could not work at the same time.

7.1 Interface

CAN1 are located in the RJ45 interface (female socket), pins are defined as follows：



	Pin	Function
CAN1	1	CAN_H
	2	CAN_L
	3	GND

When using the CAN communication interface, it is suggested that using the topology of the main line, and in order to eliminate the signal reflection on communication cable, usually need at the first, both ends or at the end of the end of the bus to join 120 Ω terminal resistance.

KW series PLC CAN interface with built-in 120 Ω resistor, controlled by the fifth dial the code switch: Puts dial the switch ON, It means termination resistors has joint;By setting the switch OFF, the terminal resistance is cancelled.

7.2 Extended bus

CAN communication port supports the extension bus protocol, could connect extension module of KS series.If the extension module is configured in the[**Hardware configuration**] of the user project, the CAN interface will work as the extension bus interface, and only free communication instructions are allowed to be used simultaneously in the user program.

The CPU is allowed to connect up to 14 extension modules, each extension module allows distributed installation, and the total length of communication cable between the CPU and one extension module at the end is not allowed to exceed 30 meters.When using the extension bus, it is recommended to add terminal resistances of the first terminal KW and the last terminal extension module to avoid signal reflection and enhance communication stability.

In addition, in the case of long-distance distributed installation, it is recommended to adopt a shielded twisted pair and a single end of the shielding layer is well grounded (controlled), and the communication cable should be away from strong interference sources, various high-power lines (including power cables of equipment), and frequently switched pulse signal lines.

In the factory default setting, the CPU module will automatically assign a unique ID and configure various parameters for each extension module when it is powered up, so it is required that the CPU and all extension modules are powered up at the same time or all extension modules are powered up before the CPU module, otherwise it may lead to program execution errors.

However, for the convenience of distributed applications, the CPU provides EX_ADDR instruction, which can be invoked by the user to modify the above default configuration, thus making the use of extension modules more flexible.

7.2.1 How to use EX_ADDR instruction to extension module

Detail of EX_ADDR please refer to 7.6.4 Extended bus instruction.

In practice, if the extension module is far away from the CPU module or installed on different devices, it may not be possible to guarantee the required power up order by default.

In this case, the user can use the EX_ADDR instruction as follows to modify the factory default configuration so that each extension module can be powered on or off at any time without causing PLC execution program errors.

1) In the user project, each extension module is added in the [**hardware configuration**] in the required order and configured according to actual requirements, and EX_ADDR instruction is called in the program (located in the [CAN instruction] group of the instruction set).
2) Connect the CPU module and all extension modules according to the order in [**hardware configuration**], and then power up according to the default order (the CPU and all extension modules power up at the same time or all extension modules power up before the CPU modules).
3) Download the user project to the CPU module.After the CPU is running properly, change the EX_ADDR instruction to 181 (decimal) and let the EX_ADDR instruction execute once.After successful execution of the instruction, each extension module will automatically save its own ID and various parameters (such as signal form, filtering mode, etc.).
4) Power off the PLC system.The user can then install the extension modules in the desired location, the order of the modules (starting with the CPU) is still the same as in the hardware configuration.After that, when the extension module is powered up again, it will automatically read the saved data and enter the running state automatically, without CPU configuration, so it can be powered on or off at any time independently of the CPU.

5) If the user needs to restore the factory default power-on order, change the parameter value of EX_ADDR instruction in the program to 99 (decimal), and then make EX_ADDR instruction execute once.After successful execution of the instruction, each extension module clears the saved ID and channel parameters and waits for the CPU to automatically assign the ID and configure the parameters when it is powered up again.

7.3 Kinco Motion control

Kinco motion control functions are used to control Kinco's motion control products (servo and step drivers) with CAN interface.Based on the CANOpen protocol, it combines the CANOpen communication details of the driver with 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 CANOpen protocol, It can easily communicate with the driver and conduct positioning control.

This function can control up to 32 axis.In practical application, users can determine the number of connections according to the required program space and network load rate.

This function supports the operation of parameter upload (download), motor lock shaft, loose shaft, return origin, inching (speed mode), absolute positioning and relative positioning of the motion control product, while torque mode and master-slave following mode are not supported for the moment.In addition, this function can be used in principle for all third-party motion control products that support the standard CANopen protocol，

Before use third-party, please consult the technical personnel of the Kinco.

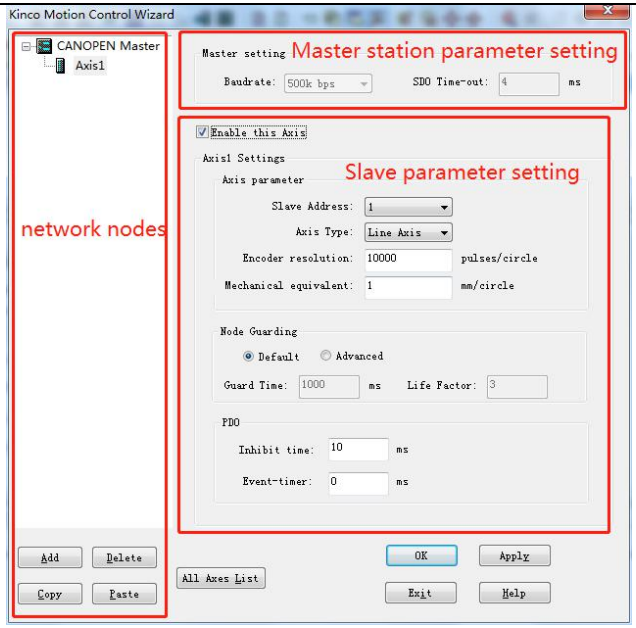
The user follows the steps below to use the Kinco motion control feature:

1) In the user project, enter the [**Kinco motion control wizard**] window to complete the network configuration.
2) Build the project with motion control instruction according to application.
   Description please review 7.6.1 Kinco MC Instruction.
3) Download the project to the PLC, the PLC will be started as the master station, management of the entire network communication, and the implementation of positioning control procedures.

7.3.1 Kinco Motion control network configuration

Kinco motion control function adopts CANOpen protocol, PLC as the master station, each driver as the slave station.Before invoking the directive, the user must first configure the actual CANOpen network being used.According to the custom of field application, we call the slave station "axis" in the software.

In the "project manager" of Kincobuilder software, double-click the "Kinco motion control network configuration" node to enter the configuration window and complete the network configuration in the window.



The window is divided into three parts: tree list of network nodes, parameters of master station and parameters of axis (slave station).

➢ **Network node tree operation**

In the network node tree, the root node is the [CANOpen master station], and each node below is the axis (slave station) in the network.

It provides [add], [delete], [copy] and [delete] 4 buttons, at the same time the software also provides the corresponding shortcut and right-click menu functions.Users can take advantage of these capabilities to operate on network nodes.

- Add a new axis

Click the [add] button;Or right-click on any node and execute the add menu command.Or use the ALT+N shortcut.The axes added using the above three methods all start with default parameters.

- Copy、Paste

The user can copy an existing axis and then paste it into the network to generate a new axis. The new axis has the same parameters as the copied axis except the axis number (slave station address).This is handy for projects where all the axes in the network function the same.

Click on an axis in the tree to select it, then click the copy button, or use Ctrl+C.Or right-click on an axis and execute the copy menu command.You can duplicate this axis in any of these ways.

After copying, click the paste button, or use the Ctrl+P shortcut, or right-click on any axis and execute the paste menu command to generate a new axis in the network.

- Delete

You can delete the axis by clicking on an axis to select it, then clicking the [DELETE] button, or by using the DELETE shortcut.You can also delete an axis by right-clicking on it and executing the delete menu command.

➢ **Master parameter**

Click on the [CANOpen master] node, all parameters of the master station will be modifiable, and all

parameters of the axis (slave station) will be grey and unmodifiable.

- 【Baudrate】: Select the baud rate used by the master station.Note that baud rates must be consistent across all nodes (master and slave) on the network.
- 【SDO Time-out】: The timeout waiting time after the master station PLC sends the SDO request message, if no reply message from the corresponding slave station is received after this time, the timeout error will be reported.When selecting a different baud rate, the software automatically recommends an SDO timeout that the user can modify.

➢ **Axis(Slave)**

When you click on an axis node, all the parameters of the axis will be modifiable, and all the parameters of the primary station will be grey and unmodifiable.

- 【Slave Address】: Axis of CANOpen slave station address,**Consecutive assignment from station number must begin with 1.**
- 【Axis type】: According to the function of the shaft, the user can choose either a straight or a rotating shaft.
- 【Encoder resolution】: The resolution of the encoder of the shaft or stepping driver, that is, the number of pulses emitted by the encoder in one rotation.
- 【Mechanical equivalent】: For each rotation of the motor shaft, the length (mm) or Angle (°) of movement of the mechanical load.
- 【Node Guarding】: 设 Set the node protection time of the axis.Users can use the default value or click "advanced" to modify it.
- 【PDO】: PDO are automatically established in the PLC for each axis to transmit information such as position, speed and status. Since the position and speed of the axis change rapidly, PDO is sent very frequently, so PDO must be set to prohibit time.The user can use the default values or modify them.

➢ **Other**

- 【OK】: Save parameter and exit.
- 【Cancle】: Not saving and Exit
- 【Apply】: Saves the parameters configured by the current setting
- 【All Axes List】: The axes checklist is designed to facilitate the inspection of all configured and enabled axis configuration parameters for verification

7.4 CANOpen Master

CANOpen has the advantages of good openness, high reliability, good real-time performance, strong anti-interference ability and low cost. It is a commonly used field bus in industrial control and has been applied more and more widely at present.

7.4.1 CANOpen Introduction

The CANOpen application layer and communication specification (CiA DS301) is the core of the CANOpen protocol and applies to all CANOpen devices.Various CANOpen communication objects are defined in DS301 and the services and protocols for these objects are described in detail.In order to facilitate the user's application, we will introduce several key objects and their communication protocols.

7.4.1.1Network Management Tool（NMT）

Network management (NMT) is a master-slave model for CANOpen devices.The NMT service can initialize, start, monitor, reset, or stop CANOpen devices.Within a network, there must be one NMT master station that has control over the entire network, the network management class (NMT) function..

Here are some common NMT services.

7.4.1.1.1 NMT Node Control

NMT master station controls the NMT state (including stop, preoperation, operation and initialization) of each slave station through the NMT Node Control message.Slave station devices must support NMT node control services.

The format of NMT node control message is as follows：

COB-ID	Byte 0	Byte 1
0x000	CS(Command Specifier)	Node ID

Node ID：Slave ID.If Node ID is 0，indicates that all slave stations on the network need to execute this command.

CS： Command Specifier,when it is： 1 Means Start target node；
2 Means Stop target node；
128 Means The target node has ready;
129 Means Reset target node
130 Means Target node reset communication parameters

7.4.1.1.2 NMT Error Control

Error control service used for detection of network failure, including Node protection (Node Guarding) and heart rate (Heartbeat) in two ways.In practice, you must choose an error control method for a node.By the way,The heartbeat service is added in a later version of the DS301 and is recommended by the CiA.

- **NMT Node Guarding**

NMTThe Master station sends remote frames (no data)：

COB-ID
0x700 + Node ID

NMT Slave station return frame：

COB-ID	Byte 0
0x700 + Node ID	Bit7：Trigger bit, must alternate "0" or "1" in each node protection reply. Bit0-6：The value of the combination represents the slave station state.Where, 0 means boot-up and 4 means STOPPED; 5 means Operational;127 means the Pre-Operational.

- **NMT Node Guarding**

If a node is configured as a heartbeat producer, it periodically sends heartbeat messages. Another node or nodes in the network act as heartbeat consumers to process heartbeat messages of each producer.In general, the master station acts as the heartbeat consumer and the other slave stations act as heartbeat producers.

The format of heartbeat message is as follows：

COB-ID	Byte 0
0x700 + Node ID	The status value of this node. Where, 0 means Boot-up ,4 means STOPPED;5 means Operational;127 means Pre-Operational.

7.4.1.2 SDO (Service Data Object)

SDO communication is based on a client-server model.

By using index and sub-index, SDO enables one CANOpen device (as a client) to directly access objects in the object dictionary of another CANOpen device (as a server).Generally, the primary station acts as the client.

SDO has two transport modes: Accelerated type, with up to four bytes of data transferred at a time;

Segmented type, allowing more than 4 bytes of data to be segmented.

The following is a brief description of the message format of the accelerated type transport mode:

SDO.requiring frame,Client -> Server：

COB-ID	Byte 0	Byte 1-2	Byte 3	Byte 4-7
0x600 + Node ID	SDO CS	Index	Sub-Index	data

Returning frame，Server -> Client：

COB-ID	Byte 0	Byte 1-2	Byte 3	Byte 4-7
0x580 + Node ID	SDO CS	Index	Sub-Index	data

7.4.1.3 PDO(Process Data Object）

PDO for real-time data transmission, a PDO message contains up to 8 bytes of data.

PDO communication is based on a producer-consumer model.In terms of sending or receiving data, PDO is divided into sending PDO (TPDO) and receiving PDO (RPDO).Producers support TPDO and consumers support RPDO.

PDO communication does not have a protocol that states that the contents contained in a PDO message are pre-defined.During network configuration, the user defines the cob-id of each PDO and the objects mapped in it, so that both the producer and consumer can know the content of the corresponding PDO and parse the message accordingly.

Each PDO is described in the object dictionary by communication and mapping parameters. The communication parameters of PDO are described below.

- **COB-ID**
  Indicates the COB-ID used by the PDO.

- **Transmission type**
  Indicates how the PDO is triggered to send (or receive).It is an 8-bit unsigned integer value.
  Transmission types fall into the following categories：
  - Synchronization: triggers send (or receive) based on the count of the SYNC object.

  If the transmission type value is 0 ,means "synchronous, non-cyclic" and if value is 1-240 ,means "synchronous, cyclic".
  - RTR-Only: Only applicable to TPDO, which is triggered by the received RTR message.

  If its value is 252 for the transmission type, means that the PDO is sent after receiving SYNC and RTR.A value of 253 means that PDO is sent immediately after RTR is received.
  - Event-driven: sends PDO immediately after an internal event occurs on CANOpen device.

  A transport type value of 254 indicates a custom event by the device manufacturer.A value of 255 indicates an event defined by the device subprotocol and the application layer protocol, typically a change in the data value in the PDO or the timer timing time.

- **Inhibit time**
  The disable time defines the minimum time interval between consecutive sends of the PDO.
  The disable time is configured to avoid the problem of high priority PDO being sent too frequently, always occupying the bus, and other lower priority packets not being able to use the bus.

- **Event-timer**
  Used to specify a periodic value that is timed to be sent.It is a 16-bit unsigned integer in ms.The PDO will trigger the send with this timing value as a period.If the value is 0, the event timer is not able to used.

7.4.2 CANOpen Master Feature

The KW CANOpen master station feature：

- Adopted standard Can2.0A.Compliant with CANOpen standard DS301 V4.2.0 protocol.
- Support NMT network management services, including NMT Node Control and NMT Error Control, as the NMT master station.
- Up to 32 CANOpen slave stations are supported.Allows users to configure the startup process for each slave site in KincoBuilder；
- Each slave station supports up to 8 TPDO and 8 RPDO;Up to 128 TPDO and 128 RPDO are supported.
- Supports client SDO and provides read and write instructions that support the standard accelerated type transmission mode；
- CANOpen predefined emergency messages are supported.

7.4.2.1 CANOpen Network Management Tool

In KincoBuilder,enter【Hardware】, Select the CPU module in the table at the top of the window, then click on the page at the bottom of the window [CANOpen master station] to enter the network configuration page of CANOpen.

7.4.2.2 EDS file

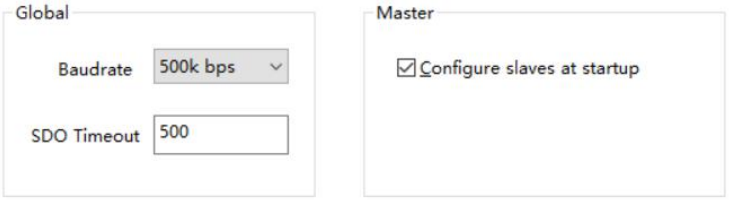
In【CANOpen】->【Network Setting】Window,The following buttons are provided to operate on EDS files：

- 【Import EDS】：Click the button to select EDS file,Import to Kincobuilder and save it.The relevant slave device would show up below【All types of device】.
- 【Delete】： Select a slave device from the list of all slave modules below and click the delete button to remove the device from the list as well as its EDS file from Kincobuilder.
- 【Export All EDS】: Could export all the existing slave site EDS files in Kincobuilder into one file (with the extension name.ALLEDS.).This feature is useful when uninstalling Kincobuilder. Users can use this feature to backup ALLEDS files of slave stations before uninstalling, and then import the backup.ALLEDS files directly.
- 【Import All EDS】: An EDS backup file (with the extension.alleds) can be imported into Kincobuilder, and all slave station devices contained in the file will be displayed below in the list of all slave station modules.

7.4.2.3 CANOpen Network configuration process

1) Configure global parameters

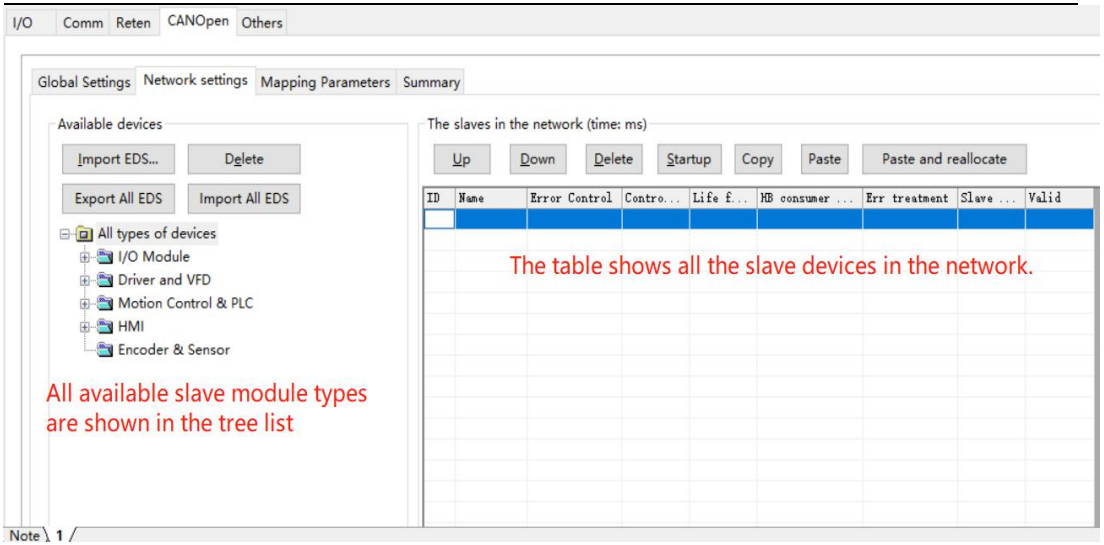
Go to the [Main Station and Global Configuration] page, as shown below:



- [Baud Rate]: Select the baud rate used by the master station. Note that the baud rates of all nodes on the network must be the same.
- [SDO Timeout]: Set the timeout waiting time after the primary station sends an SDO request message. If the response message of the corresponding slave is not received after this time, an error will be reported. The SDO timeout value setting generally does not need to exceed 100ms.
- [Configure each slave at startup]: If this option is selected, the master station will not only control the NMT state transition of each slave station, but also the master station will send the corresponding configuration according to the parameter configuration of each slave station at startup. Command to configure each slave (such as slave error control mode, PDO mapping, etc.). If this option is not selected, the primary station only controls the NMT state transition of each slave.

2) Configure each slave station

Go to the [Network Configuration] page and continue to configure the slave nodes and their parameters on the network, as shown below:



All function buttons on the page have corresponding right-click menu commands. When the user clicks the right mouse button at the relevant position, the corresponding right-click menu will pop up, and the menu command can be used at this time. The following describes the common process of configuring a slave.

a).Add a slave device to the network

Double-click the slave type you want to join the network from the tree list on the left, and add a slave device of that type to the network and display it in the table on the right.

b).Configure the station number (ID), supervision type and other parameters of the slave device:

The [Address] column in the table on the right is the station number (ID) of the slave. The first line is the location of station 1.When you add a slave device, it displays its default configuration parameters. When added, Kincobuilder defaults to adding devices to the table from top to bottom. The user can click on the row in the table to select a slave, and then click the [Up] and [Down] buttons to adjust its station number, or click the [Delete] button to take the device from the network. delete.

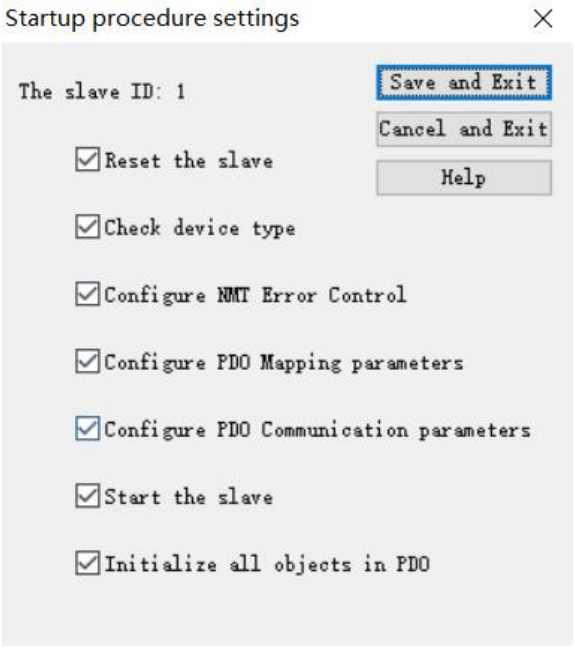
[Supervised Type] is used to configure the NMT Error Control mode of the node, including node protection and heartbeat. If the slave device supports both modes at the same time, it is recommended to use the heartbeat mode first.[Supervised time] indicates the period value of the node protection mode or heartbeat mode selected earlier. It is recommended that in practical applications, this period value should not be set too small, for example, it can be set above 2000.

[Heartbeat Consumer Time] The primary station will periodically check whether the heartbeat message of the slave station is received. If the timeout of this "heartbeat consumer" is still not received, the slave station is considered to be offline and the corresponding fault handling is performed. It is recommended that in practical applications, this period value should not be set too small, for example, it can be set above 2000.

[Troubleshooting] is used to select the processing method adopted after the primary station detects the slave failure, including three options: “None”, “Stop Node” and “Stop Network”. The faults that the primary station can detect include the SDO command timeout not responding, the node protection or the heartbeat message timeout, and the receipt of some types of emergency messages sent by the secondary station.

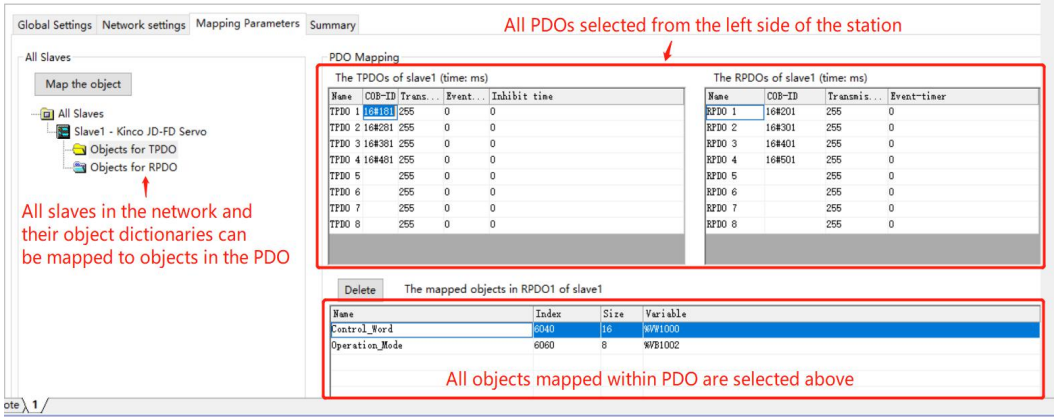
c). Configure the boot process of the slave:

Click on a slave in the table and click [Startup Process] to select what configuration the master needs to configure for the slave during network startup.



[Reset Node]: Whether the master station sends the "Reset Node" command before sending the configuration command to the slave station.
[Check device type]: Before the master station sends a configuration command to the slave station, whether to read the device information for checking.
[Configuration node supervision mode]: Whether the master station needs to configure the supervision type of the slave station and its parameters.
[Configure PDO Mapping Parameters]: Whether the primary station needs to configure the PDO mapping parameters of the secondary station.
[Configure PDO communication parameters]: Whether the primary station needs to configure the PDO communication parameters of the secondary station.
[Start this node]: After the configuration is completed, the master station needs to send the "start node" command to the slave station.
[Initialize the PDO data of the slave]: After starting the slave, whether the master needs to clear the data in all the RPDOs of the slave and send it immediately.

d).Configure the PDO of each slave：



Go to the Object Dictionary Mapping page to configure PDO for all slaves in the network.The left part of the page [Added Slave List] shows all the slaves that have joined the network, as well as the objects in the respective slave object dictionary that are allowed to be mapped to the PDO. Among them, the object in [Send PDO] can only be mapped to the TPDO of the slave station, and the object in the [Receive PDO] list can only be mapped to the RPDO of the slave station.

Click on a slave in the [added slave list], then all the PDOs of the slave will be displayed on the right side, and the user can configure each PDO:

- Communication parameters

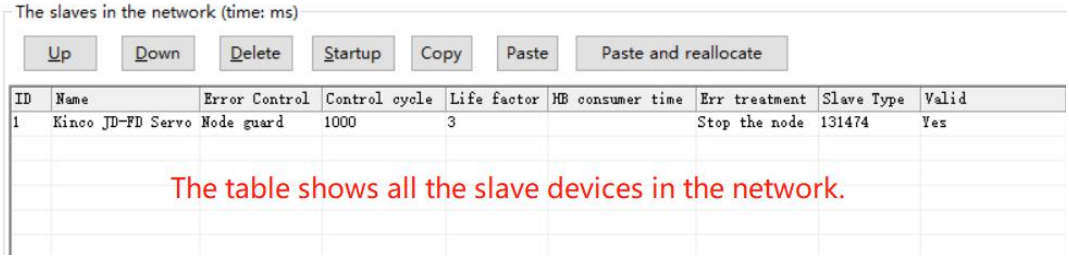
In the table on the right, select a PDO to modify the communication parameters such as timing time and prohibition time.Among them, the COB-ID of the first 4 TPDOs and RPDOs in the slave station are not allowed to be modified, and the default values in the predefined connection set in the DS301 are adopted. The last four TPDOs and RPDOs allow the user to enter the legal COB-ID value themselves.

- Mapping parameter

In the object list on the left, double-click an object, it will be added to the current PDO, and Kincobuilder will automatically assign a PLC V area address to the object, such as VW1006, the user operates the V area address in the program. It is equivalent to operating the corresponding object.

3） Copy slave, paste slave

In the [Network Configuration] page, three buttons of [Copy Slave], [Paste Slave], and [Paste Slave (Reshare Memory)] are provided. As shown below.



[Copy slave]: Select a configured slave and click this button to copy all the information of the slave (itsIt includes all PDO communication parameters, mapping parameters, etc.). If the selected slave does not have any PDO configured, the copy fails andPrompt for the corresponding information.

[Paste slave]: After copying a slave successfully, click on a blank line in the selected table, and then click this

button to paste the previously copied slave information into the line and generate a new slave. Note: The PLC memory address corresponding to each mapping object in the new slave PDO remains the same as that in the source slave. There is no reallocation, and the user needs to modify it.

[Paste slave station (reshare memory)]: The operation method is the same as [Paste slave], but the difference is that the PLC memory address of each mapping object in the new slave PDO is automatically allocated without user modification.

7.5 CAN Free communication function

KW provides a set of CAN communication commands, which can initialize the CAN port, send and receive data through the CAN port, etc., and users can use these commands to communicate with other devices. The CAN communication command supports the CAN2.0A and CAN2.0B standards. In addition, these commands only support data frames and do not support remote frames. The CAN data frame format is as follows:

ID	byte 1-8
11bit (CAN2.0A, standard frame) or 29 bits (CAN2.0B, extended frame)	Data of 1-8 bytes length

The CAN free communication function can be used simultaneously with other communication functions (extension bus, Kinc motion control, CANOpen master and slave), but it is important to note that the communication baud rate must be consistent.

Note: The ID number of the free communication message is not allowed to use the COB-ID number in the CANOpen protocol!

For details on the CAN communication instructions, refer to 7.6.3 CAN Free communication instruction.

7.6 CANBus related instruction

7.6.1 Kinco Motion control instruction

7.6.1.1 Review

The following instructions are located in the [Kinco Motion Control] group of the instruction set.

name	Functional description
MC_RPARAS	Read the parameters in the axis drive (see the parameter table below for details)
MC_WPARAS	Modify the axis parameters in the drive
MC_POWER	Control lock shaft, loose shaft
MC_RESET	Reset the error message on the axis and set the axis state to the static wait state
MC_HOME	Control axis homing
MC_JOG	Control axis jog
MC_MABS	Control axis for absolute positioning motion
MC_MREL	Control axis for relative positioning movement
MC_MIOT	Read the device number of the target axis, software version, IIT, temperature, etc.

➢ **Precautions**

Users should pay attention to the following points when using these instructions:

- In a user project, the maximum number of axes allowed: 16 for the KS and KW series and 128 for the KM series.• The total number of dedicated instructions used in a user project is limited to 192 for the KS and KW series and 1024 for the KM series. Among them, the MC_MIOT instruction only allows one for each axis.• For the same axis, when a dedicated instruction is being executed and not completed, it is not allowed to start another dedicated instruction. If the user program starts another dedicated instruction, then the instruction will end directly and report the error.
- For the same axis, the MC_MIOT instruction has the lowest priority: if other instructions are running, the MC_MIOT instruction will not be executed; if the MC_MIOT instruction is being executed and the program starts other instructions, the MC_MIOT instruction will terminate directly.• For the same axis, before the user program executes the motion command (without the read/write parameter command), the MC_POWER command must first be executed to lock the axis. After the lock axis is successful, the homing, relative motion, absolute motion or jog can be continued. instruction. If there is no lock axis, then executing these kinds of instructions will directly end and report the error.• For the same axis, the user program uses the MC_RESET instruction to reset. After the reset is successful, the axis will be in the static waiting state of the loose axis. The MC_POWER command must be executed to lock the axis before continuing to perform the homing, relative motion, absolute motion or point. Move instructions.
- For the homing, relative motion, absolute motion, or jog command, the acceleration and deceleration used is the acceleration and deceleration set internally by the drive, and can also be set by the MC_WPARAS command.• The output of each motion control instruction called in the user program is irrelevant. If an instruction executes an error, its output parameter ERRID will give an error code, and the result of this error will not be refreshed until the next time the instruction is executed again. The result of execution of other instructions will not affect the execution result of the instruction!
- After the bus is disconnected (the ONLINE output of the MC_STATE instruction is 1), for safety reasons, this group of instructions will not be automatically reconnected! After the user has to correct the error, the PLC can be restarted after the power is turned off!

➢ **Command output parameters ERRID**

Each instruction provides an ERRID output parameter. If the instruction is executed successfully, the ERRID output is 0. If the instruction fails to execute, the ERRID will be set to a different error code value to indicate the cause of the error.

The following is a description of each error code value (note that the error code here is not suitable for the MC_RPARAS and MC_WPARAS instructions, the error codes of these two instructions have special meaning, please refer to the instruction description):

Error code	Description
0	No error
1	The target axis is not enabled, or the axis does not exist in the network
2	The target axis is not in the lock axis state.
3	The target axis is executing other motion control commands and is not at rest.
4	The CAN message transmission buffer inside the PLC is full and CAN messages cannot be sent.
5	The PLC sent an SDO request message to the target axis, but did not receive a response after timeout.

6	The PLC sent an SDO request message to the target axis, but received an error response message.
7	The instruction is executed normally, but the PLC continuously detects the state returned by the target axis, and finally does not detect the correct status value.

7.6.1.2 MC_RPARAS（Read parameter）and MC_WPARAS（Change parameters）

The purpose of this group 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 the initial stage of debugging. Please refer to the driver operation manual for how to set the specific parameters. If the settings are not correct, the operation may be abnormal. Please operate with caution.

1) List of operable drive parameters

Through the drive read and write instructions, the following parameters of the drive can be operated, all parameters are readable and writable. Each instruction can operate up to 32 parameters at a time. The process data type in the table, REAL means single-precision floating-point number, UINT32 means unsigned 32-bit number, INT32 means signed 32-bit number, and so on.

The "serial number" value in the table is fixed, and each parameter has a serial number. The user can input the serial number in the instruction to operate the corresponding parameter. "Process Unit" refers to the unit used in the command parameters, "Drive Range" refers to the internal value range of the drive (this instruction will automatically convert the actual process parameter values required by the user to the internal use of the drive. Data format, such as acceleration, speed, position, etc.).

Serial number	parameter name	CANOpen Object	Process data type	Process unit	Range of values within the drive
0	**Profile_ACC**	0x60830020	REAL	Line axis：mm/s² Ratation Axis：1/s²	[0,268435455]
1	**Profile_DEC**	0x60840020			
2	**Home speed**	0x60990120	REAL	Line axis：mm/min Ratation Axis：degree/min	[-2147483648, 2147483647]
3	**Home mode**	0x60980008	INT8	DEC	[-128,127]
4	**Kvp 0**	0x60F90110	UINT16	DEC	[0,32767]
5	**Kvi 0**	0x60F90210	UINT16	DEC	[0,32767]
6	**Kpp 0**	0x60FB0110	REAL	HZ	[0,32767]
7	**K_Velocity_FF**	0x60FB0210	REAL	%	[0,1024]
8	**Kvp 1**	0x23400410	UINT16	DEC	[0,32767]
9	**Kvi 1**	0x23400510	UINT16	DEC	[0,32767]
10	**Kpp 1**	0x23400610	REAL	HZ	[0,32767]
11	**CMD_q_Max**	0x60730010	UINT16	DEC	[0,2048]
12	**Max_speed_RPM**	0x607F0020	REAL	Line axis：mm/min Ratation Axis：degree/min	[-2147483648, 2147483647]

13	**Home_Offset_Mode**	0x60990508	UINT8	No unit	[0,255]
14	**Motor direction**	0x607E0008	UINT8	No unit	0 and 1
15	**Motor_Num2**	0x64100110	UINT16	No unit	[0,65535]
16	**Soft_Positive_Limit**	0x607D0120	REAL	Line axis：mm Ratation Axis： Degree	[-2147483648, 2147483647]
17	**Soft_Negative_Limit**	0x607D0220			
18	**Pos_Filter_N**	0x60FB0510	UINT8	No unit	[0,255]
19	**Max_Following_Error**	0x60650020	UINT32	DEC	[0,268435455]
20	**Target_Pos_Window**	0x60670020	UINT32	DEC	[0,268435455]
21	**Position_Window_time**	0x60680010	UINT16	DEC	[0,32767]
22	**Speed_Fb_N**	0x60F90508	REAL	HZ	[0,45]
23	**Speed_Mode**	0x60F90608	UINT8	No unit	[0,85]
24	**Input port polarity**	0x20100110	UINT8	No unit	[0,255]
25	**DIN1**	0x20100310	UINT16	No unit	[0,65535]
26	**DIN2**	0x20100410	UINT16	No unit	[0,65535]
27	**DIN3**	0x20100510	UINT16	No unit	[0,65535]
28	**DIN4**	0x20100610	UINT16	No unit	[0,65535]
29	**DIN5**	0x20100710	UINT16	No unit	[0,65535]
30	**DIN6**	0x20100810	UINT16	No unit	[0,65535]
31	**DIN7**	0x20100910	UINT16	No unit	[0,65535]
32	**DIN8**	0x20101D10	UINT16	No unit	[0,65535]
33	**Output port polarity**	0x20100D10	UINT8	No unit	[0,255]
34	**OUT1**	0x20100F10	UINT16	No unit	[0,65535]
35	**OUT2**	0x20101010	UINT16	No unit	[0,65535]
36	**OUT3**	0x20101110	UINT16	No unit	[0,65535]
37	**OUT4**	0x20101210	UINT16	No unit	[0,65535]
38	**OUT5**	0x20101310	UINT16	No unit	[0,65535]
39	**OUT6**	0x20101E10	UINT16	No unit	[0,65535]
40	**OUT7**	0x20101F10	UINT16	No unit	[0,65535]
41	**OUT8**	0x25080420	INT32	DEC	[-2147483648, 2147483647]
42	**Pulse mode**	0x25080308	UINT8	No unit	[0,255]

43	**Save control parameter**	0x10100120	UINT32	No unit	Only 16#65766173 is valid
44	**Init control parameters**	0x10110120	UINT32	No unit	Only 16#64616f6C is valid

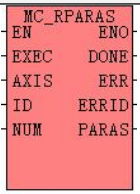
2) ERRID Parameter Description

Both the read and write parameter instructions provide the ERRID (DWORD type) output parameters.

This parameter value is an error code indicating an error that occurred during the execution of the instruction.

Error code	**Meaning**
0xFFFFFFFF	An error has occurred that caused the instruction to fail to execute, including: 1) The axis number entered by the user is incorrect and the number of parameters is incorrect. 2) There are other Kinco dedicated instructions running 3) The instruction has to operate 32 parameters, and the 32 parameters fail to operate.
Other value	Each bit of ERRID indicates the operation result of the corresponding parameter, and each bit corresponds one-to-one with the parameter specified in the ID parameter number table: bit0 indicates the result of the first parameter of the current operation, and bit1 indicates the second parameter. The result of the operation, and so on. A bit value of 1 indicates that the corresponding parameter operation failed, otherwise the corresponding parameter operation succeeds.

3) MC_RPARAS（Read parameter）

	Name	Instruction format	Suitable for
LD	MC_RPARAS		• KS • KW103 • KW203

Parameter	Input/Output	type of data	Allowed memory area	Description
EXEC	Input	BOOL	M、V、L、SM	If a rising edge of EXEC is detected, the instruction is triggered to execute.
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 sequence number table of the parameter to be read.
NUM	Input	INT	V、M、L、constant	The number of parameters to read
DONE	Output	BOOL	M、V、L	Complete the flag. When the instruction execution is completed, DONE transitions from 0 to 1.
ERR	Output	BOOL	M、V、L	Error flag. Set to 1 if an error occurs while executing the instruction.
ERRID	Output	DWORD	V、M、L	Error code
PARAS	Output	DWORD	V、M、L	The starting address of the storage of all parameter values read.

AXIS and NUM must be either constant type or memory type at the same time. In addition, the ID and PAPAS parameters together form a variable-length memory block. All of the memory blocks must be in the legal memory area, otherwise the result is unpredictable.

The three parameters ID, PARAS, and NUM together form a parameter table. The ID is the starting address of the sequence number table. The serial number of each parameter to be operated is stored successively from this address (that is, the "serial number" in the previous parameter list), each sequence number occupies 1 byte; PARAS is the parameter value. The starting address of the table, from which it successively stores the values of the read parameters, each of which takes up 4 bytes; NUM is the number of parameters to be operated. For example, suppose the ID parameter is VB100, the PARAS parameter is VD1200, and the NUM parameter is 3. Then VB100, VB101, and VB102 respectively store the serial numbers of the three parameters to be operated, and the three parameters read after the execution of the instruction is completed. The values are stored in VD1200, VD1204, and VD1208.

Note that for PARAS, although the parameter value table uses the DWORD address uniformly, the actual data types of the various process parameters are not the same. Therefore, in the user program, the user should process the data in the parameter table according to the actual data type.

- If the actual process data type is REAL type, then the parameter memory can be operated directly by the floating point address. For example, if the parameter value is 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 other than REAL, and the corresponding parameter memory does not force the data type to be defined in the global variable table, then the parameter memory can be read directly, because the instruction will automatically handle various signed and Unsigned integer. For example, if the parameter value is stored in VD1200 and the actual type is INT32 or UINT32, then VD1200 is directly operated.

- LD format instruction description

If EN is 1, then the instruction is triggered to execute on the rising edge of the EXEC input. The instruction sequentially sends the SDO to the driver to read the corresponding object according to the parameter table to be read specified by ID and NUM, and will read the corresponding object. The data is placed in the value table specified by PARAS in turn, and the corresponding bit of ERRID is set to 0. If the SDO response of a parameter is incorrect or the timeout does not respond, 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 parameters are read, 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. When EXEC becomes 0 during the execution of the instruction, the instruction will stop reading the parameters that have not been completed, and set DONE to 1, ERR and ERRID maintain the executed results.

If the PLC detects an error when the command is started (for example, the axis is not enabled, the axis is executing other commands, etc.), exit directly, set DONE, ERR to 1, and set ERRID to the corresponding error code.

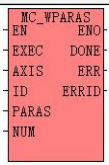
➢ **Example**

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

```
(* Network 0 *)
(*Set the parameter table to indicate that parameters 0, 3, and 8 are to be read this time..*)
```

```
LD          %SM0.0
MOVE        B#0, %VB100
MOVE        B#3, %VB101
MOVE        B#8, %VB102
(* Network 1 *)
(*Call the instruction. This time, AXIS and NUM parameters are constants, they also support the format of full memory address. *)
LD          %M0.0
MC_RPARAS %M1.1, 1, %VB100, 3, %M1.2, %M1.3, %MD8, %VD1200
(* Network 2 *)
(*The read parameter values are sequentially stored in the parameter value table known by the PARAS parameter. The first data in the table is the first parameter value read, that is, parameter 0. Because it is of type REAL, 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 0. This parameter is a signed 8-bit number. Since 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, parameter 8. This parameter is unsigned 16-bit number, but the maximum range is 32767, so the program can be processed by INT or WORD type, but it is best to judge whether the value is within the allowable range. *)
LD          %SM0.0
DI_TO_I %VD1208, %VW308
NE          %VW308, 0
ST          %M3.0
```

4) MC_WPARAS（Change parameters）

	Name	Instruction format	Suitable for
LD	MC_WPARAS		• KS • KW103 • KW203

Parameter	Input/Output	Date type	Allowed memory area	Description
EXEC	Input	BOOL	M、V、L、SM	If the rising edge of EXEC is detected, the instruction is triggered to execute..
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 sequence number table of the parameter to be modified.
PARAS	Output	DWORD	V、M、L	Read the starting address of the stored parameter values.
NUM	Input	INT	V、M、L、constant	The number of parameters to be modified
DONE	Output	BOOL	M、V、L	Complete the flag. When the instruction execution is completed, DONE transitions from 0 to 1.

ERR	Output	BOOL	M、V、L	Error flag. Set to 1 if an error occurs while executing the instruction.
ERRID	Output	DWORD	V、M、L	Error code

AXIS and NUM must be either constant type or memory type at the same time. In addition, the ID and PAPAS parameters together form a variable-length memory block. All of the memory blocks must be in the legal memory area, otherwise the result is unpredictable.

The three parameters ID, PARAS, and NUM together form a parameter table. The ID is the starting address of the sequence number table. The serial number of each parameter to be operated is stored successively from this address (that is, the "serial number" in the previous parameter list), and each serial number occupies 1 byte； PARAS is the starting address of the parameter value table. From this address, the values of each parameter are stored in succession, each value occupies 4 bytes; NUM is the number of parameters to be operated..For example, suppose the ID parameter is VB100, the PARAS parameter is VD1200, and the NUM parameter is 3. Then VB100, VB101, and VB102 respectively store the serial numbers of the three parameters to be operated, and VD1200, VD1204, and VD1208 respectively store the parameters to be modified. value.

Note: for PARAS, although the parameter value table uses the DWORD address uniformly, the actual data type of each process parameter is not the same, so in the user program in the table, the user should assign a value to the corresponding address in the parameter table according to the actual data type.

- If the actual process data type is REAL type, then the parameter memory can be operated directly by the floating point address. For example, if the parameter value is expected to be stored in the VD1200, then the VR1200 can be operated directly. VD1200 and VR1200 actually occupy the same memory address in the PLC, and the instruction will automatically perform type conversion.
- If the actual process data type is other than REAL, the parameter memory can be directly manipulated, and the instruction will automatically perform type conversion according to the data type of the parameter. For example, if the parameter data type is UINT16, then a legal value can be directly assigned to VD1200.

• LD format instruction description

If EN is 1, then the instruction is triggered to execute on the rising edge of the EXEC input. The instruction sequentially sends the value in PARAS to the drive through the SDO to modify the corresponding object according to the parameter table specified by ID, PARAS, NUM. The corresponding bit of ERRID is set to 0. If the SDO of a parameter responds incorrectly or the timeout does not respond, set the corresponding bit of the ERRID to 1, and then continue writing the next parameter. When all 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 becomes 0 during the execution of the instruction, the instruction will stop writing the parameters that have not been completed, and set DONE to 1, ERR and ERRID maintain the executed result. If the PLC detects an error when the command is started (for example, the axis is not enabled, the axis is executing other commands, etc.), exit directly, set DONE, ERR to 1, and set ERRID to the corresponding error code.

➢ **Example**

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

```
(* Network 0 *)
(* Set the parameter table to indicate that parameters 0, 3 and 8.* are to be read this time.*)
```

```
LD          %SM0.0
MOVE        B#0, %VB100
MOVE        B#3, %VB101
MOVE        B#8, %VB102
(* Network 1 *)
(* Set the value of each parameter to be written. Note the data type. *)
LD          %SM0.0
MOVE        1200.0, %VR1000
MOVE        DI#8, %VD1004
MOVE        DI#2000, %VD1008
(* Network 2 *)
(*Call instruction *)
LD          %SM0.0
MC_WPARAS %M0.1, 1, %VB100, %VD1000, 8, %M0.2, %M0.3, %MD4
```

7.6.1.3 MC_POWER（Lock shaft and loose shaft）

	Name	Instruction format	Suitable for
LD	MC_POWER		• KS • KW103 • KW203

Parameter	Input/Output	Date type	Allowed memory area	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	Complete the flag. When the instruction execution is completed, DONE transitions from 0 to 1.
ERR	Output	BOOL	M、V、L	Error flag. Set to 1 if an error occurs while executing the instruction.
ERRID	Output	BYTE	V、M、L	Error code

- LD format instruction description

If EN is 1, then the execution of the lock axis command will be triggered on the rising edge of EXEC, and the execution of the loose axis command will be triggered on the falling edge of EXEC. When the instruction is executed, the PLC first sends a command to control the axis to enter the standby state, and checks the actual return status of the drive within the 5S timeout period. If the execution is successful, the command is executed successfully, then DONE is set to 1, ERR is set to 0, and ERRID is Set to 0. If an error occurs (may be the execution of the instruction itself, or the error of the driver not performing the action correctly during the execution, see the error code), the instruction will fail to execute, the instruction will stop executing, and DONE will be set to 1, ERR. ERRID is assigned the corresponding error code.

If EN is 0, the instruction is not executed.

7.6.1.4 MC_RESET（Reset drive alarm）

	Name	Instruction format	Suitable for
LD	MC_RESET		• KS • KW103 • KW203

parameter	Input/ Output	Date type	Acceptable Memory Areas	Description
EXEC	Input	BOOL	M、V、L、SM	A rising edge triggers this instruction to execute 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	Done flag. When the instruction execution is completed, DONE transitions from 0 to 1.
ERR	Output	BOOL	M、V、L	Done flag. When the instruction execution is completed, DONE transitions from 0 to 1.
ERRID	Output	BYTE	V、M、L	Error code

When an axis makes an error during operation, this instruction can be called to reset the error information on the axis, and at the same time, set the axis to the loose shaft standstill waiting state. If you need to continue to execute other motion instructions 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 information of the driver, and does not reset the output results of each instruction!

- LD Format Instructions

If EN is 1, the execution of this instruction will be triggered on the rising edge of EXEC. When the instruction is executed, the PLC first sends a command to reset the drive alarm, and checks the actual state of the drive within a timeout period of 2 seconds. If the reset is successful, it indicates that the command was successfully executed, DONE is set to 1, ERR is set to 0, and ERRID is set 0. If an error occurs (it may be an error in the execution of the instruction itself or an error that 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 DONE is set to 1, ERR is set to 1, The ERRID is assigned the corresponding error code.

If EN is 0, the instruction is not executed.

7.6.1.5 MC_HOME（Homing）

	Name	Instruction format	Suitable for
LD	MC_HOME		• KS • KW103 • KW203

parameter	Input/Output	Date type	Acceptable Memory Areas	Description
EXEC	Input	BOOL	M、V、L、SM	A rising edge triggers this instruction to execute once; a falling edge triggers a pause in 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	Origin offset position, unit: mm or °.
TIME	Input	DWORD	V、M、L、constant	Timeout time. If the origin is not found within this time, it will exit with an error.
DONE	Output	BOOL	M、V、L	Done flag. When the instruction execution is completed, DONE transitions from 0 to 1.
ERR	Output	BOOL	M、V、L	Error flag. Set to 1 if an error occurs during instruction execution.
ERRID	Output	BYTE	V、M、L	Error code

By executing this command, the target axis can be returned to the origin. The POS parameter sets the offset value of the origin coordinates.

Note: This instruction uses the internal homing mode of the driver. You must first set the 60980008 homing mode on the driver (also written by the MC_WPARAS instruction). For details, please refer to the driver manual.

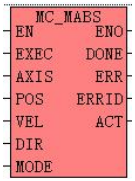
- LD Format Instructions

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

When the instruction is executed, the PLC first sends a command to start the axis to find the origin; after the transmission is completed, check the return status of the driver. The inspection will continue for TIME (timeout time set by the user, unit ms). If the axis successfully finds the origin within this time, it means that the instruction was executed successfully. At this time, DONE is set to 1, ERR is set to 0, and ERRID is set to 0. If an error occurs (it may be an error in the execution of the instruction itself or an error during the execution of the driver during execution, see the error code for details), the instruction execution fails, the instruction will stop executing, and DONE is set to 1, and 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 the execution, the instruction will stop executing and the axis will be in the state of static lock and waiting.

7.6.1.6 MC_MABS（PABS）

	Name	Instruction format	Suitable for
LD	MC_MABS		• KS • KW103 • KW203

parameter	Input / Output	Date type	Acceptable Memory Areas	Description
EXEC	Input	BOOL	M、V、L、SM	A rising edge triggers this instruction to execute once; a falling edge triggers a pause in 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, unit: mm or °
VEL	Input	REAL	V、M、L、constant	Maximum speed (> 0) increased during movement, unit: mm / min or ° / min.
DIR	Input	INT	V、M、L、constant	Direction of movement. It is reserved and has not implemented the function for the time being. It can be kept at 0.
MODE	Input	INT	V、M、L、constant	Movement mode: single execution or permanent execution.0 means single execution, the command will exit after the axis executes this absolute positioning.1 means permanent execution. After the axis performs an absolute positioning, the command does not exit. If a new target position is found, a command will be sent to allow the axis to continue to perform a new absolute positioning.
DONE	Output	BOOL	M、V、L	Done flag. When the instruction execution is completed, DONE transitions from 0 to 1.
ERR	Output	BOOL	M、V、L	Error flag. Set to 1 if an error occurs during instruction execution.
ERRID	Output	BYTE	V、M、L	error code
ACT	Output	BOOL	M、V、L	MODE = 0, single execution, ACT indicates whether the single positioning instruction is activated correctly. 1 means active, 0 means inactive.MODE = 1, when executed permanently, ACT indicates whether the permanent positioning instruction is correctly activated. 1 means active (it will remain at 1 when single positioning is completed), 0 means not active.

This command controls the target axis to move to the target position (absolute position). When moving, the speed starts from the current value, and it reaches zero when it reaches the target position. This instruction allows a pause.

- LD Format Instructions

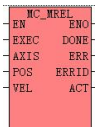
If EN is 1, the execution of this instruction will be triggered on the rising edge of EXEC.

When the instruction is executed, the PLC controls the axis to start the absolute positioning of the axis according to the target position (POS) and motion speed (VEL) parameter values entered by the user. During the movement, the command will continuously scan the target position and the target speed parameter value. If there are changes, the command will be sent to the axis immediately, that is, new speed parameters and position parameter values can be accepted at any time (for example, to perform a pause, during the movement Set the speed to 0 in the middle to pause, and resume the movement by re-setting the speed value). At the same time, the PLC will continuously check the return status of the inspection axis. If the target position of this positioning is successfully reached, indicating that the positioning is completed, DONE is set to 1, ERR is set to 0, and ERRID is set to 0. After the positioning is completed, the command will judge the MODE value. If it is set to the single operation mode, the command will exit directly; if it is set to the permanent operation mode, the command will not exit, and the target position value will be scanned at any time. When it changes, it will be sent to the axis, and the axis will perform a new absolute positioning.

If an error occurs (it may be an error in the execution of the instruction itself or an error that 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 DONE is set to 1, ERR is set to 1, The ERRID is assigned the corresponding error code.If EN is 0, the instruction is not executed.

If EN becomes 0 during the execution, the instruction will stop executing and the axis will be in the state of static lock and waiting.

7.6.1.7 MC_MREL（PREL）

	Name	Instruction format	Suitable for
LD	MC_MREL		• KS • KW103 • KW203

parameter	Input / Output	Date type	Acceptable Memory Areas	Description
EXEC	Input	BOOL	M、V、L、SM	A rising edge triggers this instruction to execute once; a falling edge triggers a pause in 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 move, in mm or °.A positive number indicates movement in the positive direction; a negative number indicates movement in the negative direction.
VEL	Input	REAL	V、M、L、constant	Maximum speed (> 0) increased during movement, unit: mm / min or ° / min.
DONE	Output	BOOL	M、V、L	Done flag. When the instruction execution is completed, DONE transitions from 0 to 1.
ERR	Output	BOOL	M、V、L	Error flag. Set to 1 if an error occurs during instruction execution.
ERRID	Output	BYTE	V、M、L	Error code
ACT	Output	BOOL	M、V、L	Whether the instruction is activated correctly. 1 means active, 0 means inactive.

This command controls the target axis to move a specified distance POS (using the current position as a reference, that is, using the current position as the starting position). When moving, the speed starts from the current value, and it reaches zero when it reaches the target position. This instruction allows a pause.

- LD format instruction description

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

When the instruction is executed, the PLC controls the axis to start relative positioning (using the current position as a reference) according to the target position (POS) and motion speed (VEL) parameter values entered by the user. During the movement, the command will continuously scan the target speed parameter value. If there is a change, the command 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 movement, that is, Yes, and then resume the movement by giving the speed value again). At the same time, the PLC will continuously check the return status of the inspection axis. If the target position of this positioning is successfully reached, indicating that the positioning is completed, DONE is set to 1, ERR is set to 0, and ERRID is set to 0. If an error occurs (it may be an error in the execution of the instruction itself or an error that 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 DONE is set to 1, ERR is set to 1, The ERRID is assigned the corresponding error code.If EN is 0, the instruction is not executed.

If EN becomes 0 during the execution, the instruction will stop executing and the axis will be in the state of static lock and waiting.

7.6.1.8 MC_JOG（JOG）

	Name	Instruction format	Suitable for
LD	MC_JOG		• KS • KW103 • KW203

parameter	Input / Output	Date type	Acceptable Memory Areas	Description
EXEC	Input	BOOL	M、V、L、SM	A rising edge triggers this instruction to execute once; a falling edge triggers a pause in 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	Movement speed, unit: mm / min or ° / min.A positive number indicates a positive direction, and a negative number indicates a negative direction.
DIR	Input	INT	V、M、L、constant	Direction of movement. It is reserved and has not implemented the function for the time being. It can be kept at 0.
DONE	Output	BOOL	M、V、L	Done flag. When the instruction execution is completed, DONE transitions from 0 to 1.
ERR	Output	BOOL	M、V、L	Error flag. Set to 1 if an error occurs during instruction

				execution.
ERRID	Output	BYTE	V、M、L	Error code
ACT	Output	BOOL	M、V、L	Whether the instruction is activated correctly. 1 means active, 0 means inactive.

This command controls the target axis to run at the target speed specified by Vel.

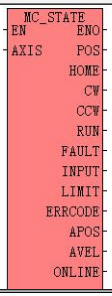
- LD Format Instructions

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

When the instruction is executed, the PLC controls the axis to start jogging according to the user-entered velocity (VEL) parameter value. During the axis movement, the command will continuously scan the target speed parameter value. If there is a change, the command will be sent to the axis immediately, that is, new speed parameter values can be accepted at any time.

If an error occurs (it may be an error in the execution of the instruction itself or an error that 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 DONE is set to 1, ERR is set to 1, The ERRID is assigned the corresponding error code.If EN is 0, the instruction is not executed. If EXEC becomes 0 during the execution, the instruction will stop executing and the axis will be in the state of static lock and waiting for the axis.

7.6.1.9 MC_STATE（Read the status of the drive）

	Name	Instruction format	Suitable for
LD	MC_STATE		• KS • KW103 • KW203

Operands	Input/ Output	Data Type	Acceptable Memory Areas	Description
AXIS	Input	INT	V、M、L、Constant	Axis number（CANOpen slave ID）
POS	Output	BOOL	V、M、L	"Position reached" 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 backward" signal
RUN	Output	BOOL	M、V、L	"Motor running" signal
FAULT	Output	BOOL	M、V、L	"Axis faulting" signal
INPUT	Output	WORD	V、M、L	The status of DI,BIT0 corresponding to DNI1 status.BIT1~BIT7 corresponding to DIN2~DIN8. The number of DIN please find detail on driver manual
LIMIT	Output	BOOL	M、V、L	"Limit reached" signal

ERRCODE	Output	WORD	V、M、L	Axis alarming error code
APOS	Output	REAL	M、V、L	Current actual position of the machine, unit in mm or °
AVEL	Output	REAL	M、V、L	Actual actual speed of the machine, Unit in mm / min or ° / min.
ONLINE	Output	BYTE	M、V、L	"Axis status" signal. 1 means the axis is offline, 0 means the axis is online.

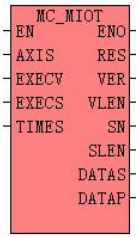
This instruction scans the driver status all the time, obtains the flags of various states and outputs them to the corresponding output parameters.

Note: The two signals "position to" and "origin found" will change to 0 again during the execution of the action (positioning or finding the origin), and will not be reset to 1 until the action is performed correctly!

- LD Instruction description

If EN is 1, this instruction is executed. If EN is 0, the instruction does not execute and the output parameters would not be refreshed.

7.6.1.10 MIOT_MC（Read Kinco servo driver information）

	Name	Instruction format	Suitable for
LD	MC_HOME		• KS

parameter	Input/ Output	Data Type	Acceptable Memory Areas	Description
AXIS	Input	INT	V、M、L、Constant	Axis number（CANOpen slave ID）
EXECV	Input	BOOL	M、V、L、SM	Rising edge triggers once reading of the serial number and software version
EXECS	Input	BOOL	M、V、L、SM	Rising edge trigger once reading of IIT, temperature, running time
TIMES	Input	INT	V、M、L	Timer, time to read IIT, temperature, running time. If it is 0, the timer does not enable.
RES	Output	BYTE	M、V、L	Result
VER	Output	BYTE	M、V、L	The software version information Location
VLEN	Output	BYTE	M、V、L	The total length of software version information, in bytes
SN	Output	BYTE	M、V、L	The serial number information Location
SLEN	Output	BYTE	M、V、L	The total length of serial number information, in bytes
DATAS	Output	BYTE	M、V、L	The IIT, temperature, running time information

				storage starting address
DATAP	Output	BYTE	M、V、L	State word, current, speed and other information stored at the starting address

This instruction is used to read the product, running state and other information of the target axis. Only one of these instructions is allowed per axis.

For the same axis, MC_MIOT instruction has the lowest priority: if other motion instruction is running, MC_MIOT will not execute; If other motion instructions are started, the MC_MIOT instruction is interrupted and terminated.

The information read by this instruction is divided into three categories, as detailed below.

- **Serial Number, Software Version**

Triggered by the rising edge of the EXECV parameter, this information is read once. These are fixed information, generally when the power can be read once.

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

The SN parameter specifies the starting location for the product serial number information, which is continuously stored of the starting address. The SLEN value specifies the total length of the sequence number information, that is, the number of bytes consumed.

After each trigger, the PLC performs a reading process. If all reads successfully, the data and length information in the output parameters are updated. If the read fails, the output parameters are not updated. Whether it succeeds or fails, the corresponding bit in the *RES* is flushed when the instruction completes.

Bit in RES	Description
Bit 7	Indicates whether the read completed. 0 means that it is in the process of reading; 1 means read complete (whether successful or failed)
Bit 6	Indicates whether an error occurred. 0 represents successful reading; 1 indicates an error in reading.

- **IIT, drive temperature, running time information**

These are information that needs to be read in real time, but not so frequently that it might interfere with other instruction.

There are two kinds of trigger conditions for reading these parameters: the rising edge of EXECS parameters triggers reading once; The timing reading cycle specified by TIMES, at which the PLC will trigger a read every cycle. If the TIMES parameter value is 0, the timing reading would not work.

The DATAS parameter specifies the starting address for storing this information, and each parameter information is stored as shown in the table below：

Parameter	Object	Data type	Length	Offset in byte
IIT	0x60F612	UINT16	2	0
Temperature	0x60F70B	UINT16	2	2
Running Time	0x2FF700	UINT32	4	4

After each trigger, the PLC performs a reading process. If all reads successfully, the data and length information in the output parameters are updated. If the read fails, the output parameters are not updated. Whether it succeeds or fails, the corresponding bit in the *RES* is flushed when the instruction completes

Bit in RES	Description
Bit 5	Indicates whether the read completed. 0 means that it is in the process of reading; 1 means read complete (whether successful or failed)
Bit 4	Indicates whether an error occurred. 0 represents successful reading; 1 indicates an error in reading.

- **State word, Error word, Actual Current, etc**

This information is automatically read by instructions through PDO, without the user having to trigger it in the program.

The *ATAP* parameters specify the starting address where the information will be stored, and each parameter information will be stored as shown in the table below：

Parameter	Object	Data type	Length	Offset in byte
State 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 Position	0x606300	INT32	4	12

- **Execution Result ： *RES***

Bit in RES	Description
Bit 7	Indicates whether the read of software version, serial number parameters completed. 0 means that it is in the process of reading; 1 indicates that the read is complete (whether it succeeds or fails).
Bit 6	Indicates whether an error occurred in reading the software version and serial number parameters. 0 represents successful reading; 1 indicates error happened.
Bit 5	Indicates whether the reading of IIT, temperature, and actual parameters completed. 0 means that it is in the process of reading; 1 indicates that the read is complete (whether it succeeds or fails).
Bit 4	Indicates whether there is an error in reading the group IIT, temperature. 0 represents successful reading; 1 indicates error happened.
Bit 3…0	The combined value represents an execution error： 0 --- Mean no error 1 --- Mean the axis number is wrong 2 --- Mean the other motion instructions are executing. This instruction cannot be run.

- **LD Instruction description**

If EN is 1, then the corresponding device information will be triggered to read according to the parameters of EXECV, EXECS and TIMES.

If EN is 0, the instruction is not executed. If EN goes to 0 during execution, the instruction stops executing.

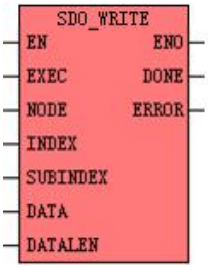
7.6.2 SDO Instruction

SDO instructions are located in the CAN instruction group.

SDO Instruction can be used when using the Kinco motion control function or the CANOpen master station function.

A maximum of 64 SDO instructions are allowed in a user project.

7.6.2.1 SDO_WRITE

	Name	Instruction format	Suitable for
LD	SDO_WRITE		• KS • KW103 • KW203

Operands	Input/output	Data Type	Acceptable Memory Areas
EXEC	Input	BOOL	I、Q、V、M、L、SM
NODE	Input	BYTE	I、Q、V、M、L、SM、Constant
INDEX	Input	WORD	I、Q、V、M、L、SM、Constant
SUBINDEX	Input	BYTE	I、Q、V、M、L、SM、Constant
DATA	Input	BYTE	I、Q、V、M、L、SM
DATALEN	Input	BYTE	I、Q、V、M、L、SM、Constant
DONE	Output	BOOL	Q、M、V、L、SM
ERROR	Output	DWORD	Q、M、V、L、SM

Note: NODE, INDEX, SUBINDEX, DATALEN must be constant or variable at the same time. The DATA and DATALEN parameters make up a variable length block of memory, which must all be in valid memory regions, otherwise the result is not expected.

The specific use of each parameter is shown in the table below:

Operands	Description
EN	The Enable bit. If EN is 1, the instruction is enabled, allowing execution
EXEC	The Execute Bit. The rising edge of EXEC triggers this instruction execute once, please make sure that EN is trigger before EXEC.
NODEID	The accessed Node ID.
Index	Index of the object to be accessed in OD
SubIndex	Sub-index of the object to be accessed in OD
Data	The initial byte address of data.
DataLen	The length of data, in bytes
DONE	Execution result indication. If SDO is executing, DONE is 0; If SDO communication ends (response received or timeout), DONE is 1
ERROR	Error message. See the table below

SDO Error message. See the table below：

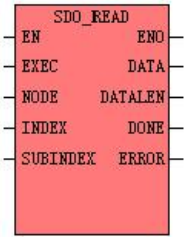
Error code	Description
0	No mistake.
1	The master station is not enabled
2	Target Node is no existed
3	Incorrect input parameter values (such as data length)
4	The last command on the target node has not response
5	The PLC send or receive buffer is full
6	The instruction timeout did not respond
7	Error received response message (not expected response message, length error, etc.)
8	Received termination message
9	In this project, the number of SDO instructions exceeded the limit

• LD Instruction description

If EN is 1, the instruction can be scanned, and if the rising edge of EXEC is detected, execution is started once.

If EN is 0, the instruction cannot be scanned and will not be executed.

7.6.2.2 SDO_READ

	Name	Instruction format	Suitable for
LD	SDO_READ		• KS • KW103 • KW203

Operands	Input/output	Data Type	Acceptable Memory Areas
EXEC	Input	BOOL	I、Q、V、M、L、SM
NODE	Input	BYTE	I、Q、V、M、L、SM、Constant
INDEX	Input	WORD	I、Q、V、M、L、SM、Constant
SUBINDEX	Input	BYTE	I、Q、V、M、L、SM、Constant
DATA	Output	BYTE	I、Q、V、M、L、SM

DATALEN	Output	BYTE	I、Q、V、M、L、SM
DONE	Output	BOOL	Q、M、V、L、SM
ERROR	Output	DWORD	Q、M、V、L、SM

Note: NODE, INDEX, SUBINDEX, DATALEN must be constant or variable at the same time. The DATA and DATALEN parameters make up a variable length block of memory, which must all be in valid memory regions, otherwise the result is not expected.

The specific use of each parameter is shown in the table below:

Operands	Description
EN	The Enable bit. If EN is 1, the instruction is enabled, allowing execution
EXEC	The Execute Bit. The rising edge of EXEC triggers this instruction execute once, please make sure that EN is trigger before EXEC.
NODEID	The accessed Node ID.
Index	Index of the object to be accessed in OD
SubIndex	Sub-index of the object to be accessed in OD
Data	The initial byte address of data.
DataLen	The length of data, in bytes
DONE	Execution result indication. If SDO is executing, DONE is 0; If SDO communication ends (response received or timeout), DONE is 1
ERROR	Error message. See the table below

SDO Error message. See the table below：

Error code	Description
0	No mistake.
1	The master station is not enabled
2	Target Node is no existed
3	Incorrect input parameter values (such as data length)
4	The last command on the target node has not response
5	The PLC send or receive buffer is full
6	The instruction timeout did not respond
7	Error received response message (not expected response message, length error, etc.)
8	Received termination message
9	In this project, the number of SDO instructions exceeded the limit

• LD Instruction description

If EN is 1, the instruction can be scanned, and if the rising edge of EXEC is detected, execution is started once.

If EN is 0, the instruction cannot be scanned and will not be executed.

7.6.3 CAN Free communication instruction

AN communication instructions are located in the CAN instruction group .

Note: the ID number of free communication message is not allowed to use the cob-id number in CANOpen protocol！

7.6.3.1 CAN_INIT（Initialize the CAN interface）

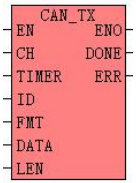
	Name	Instruction format	Suitable for
LD	CAN_INIT		• KS • KW103 • KW203

Operands	Input/output	Data Type	Acceptable Memory Areas
EN	Input	BOOL	I、Q、V、M、L、SM
CH	Input	INT	Constant
BAUD	Input	INT	L、M、V、Constant
ERR	Output	BOOL	L、M、V、Constant

Operands	Description
EN	The Enable Bit
CH	CAN interface used. 0 for CAN1, 1 for CAN2, 2 for K541 module
BAUD	CAN Baudrate 8 --- 1000K 7 --- 800K 6 --- 500K 5 --- 250K 4 --- 125K 3 --- 50K 2 --- 20K 1--- 10K
ERR	Instruction execution status. 0 means success, 1 means error (such as parameter error)

This instruction is triggered by rising edge of the EN input that initializes the specified CAN interface (CH) and sets the CAN baud rate same as BAUD value.

7.6.3.2 CAN_TX（Send CAN message automatically）

	Name	Instruction format	Suitable for
LD	CAN_TX		• KS • KW203

Operands	Input/output	Data Type	Acceptable Memory Areas
CH	Input	INT	Constant
TIMER	Input	INT	L、M、V、Constant
ID	Input	DWORD	L、M、V、Constant
FMT	Input	INT	L、M、V、Constant
DATA	Output	BYTE	M、V
LEN	Output	BYTE	L、M、V
DONE	Output	BOOL	L、M、V
ERR	Output	BOOL	L、M、V

Operands	Description
EN	The Enable Bit.
CH	CAN interface used. 0 for CAN1, 1 for CAN2, 2 for K541 module
TIMER	The period of time for the message to be sent,unit in *ms*. 0 means that timing sending is not enabled.
ID	ID of message to be sent
FMT	Format of message to be sent. 0 is the standard frame, and 1 is the extended frame.
DATA	The first address of the data store of the message to be sent.
LEN	Data length of message to be sent in bytes.
DONE	Transmission complete flag. After each successful transmission, DONE is automatically set to 1 and maintained for at least one scan period.
ERR	Send error flag. 1 indicates that the transmission failed.

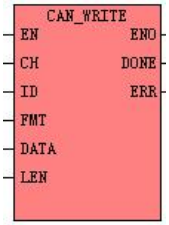
Note: ID, FMT, TIMER and LEN parameters must be both constant and variable. The DATA and LEN parameters form a memory block of variable length, which must all be in the legal memory area, otherwise the result is unpredictable.

The PLC maintains an automatic message sending list internally. When the message in the table meets the sending condition, the PLC will automatically send out the message.The condition for automatic transmission of a message is: if the data in the message changes, it is immediately sent once; If the set timing of the send cycle time, immediately sent once.After the message is sent, the output parameter "DONE" is automatically set to 1 and after a scan,set to 0 automatically. If the message sending fails (because the sending buffer is full or the message fails to send), the output parameter "ERR" is automatically set to 1.The maximum length of the sent message list is 48.

CAN_TX instruction is used to add a message to the message list, which is specified by message ID number, format (FMT, indicating the extended frame or standard frame), DATA (DATA, indicating the starting address where the message DATA is stored), and length LEN. The TIMER parameter value indicates the time period (ms) for timed delivery. If the TIMER value is 0, it means that it will not be timed delivery.

The rising edge of the *EN* input triggers the execution of the instruction. After the execution of this instruction, PLC will immediately add the message specified by the instruction to the automatic transmission list.**Therefore, a CAN_TX instruction needs to be executed only once in a project. In addition, up to 48 CAN_TX instructions can be invoked in a project! The total number of CAN_TX and CAN_WRITE instructions is allowed to be up to 64！**

7.6.3.3 CAN_WRITE（Send CAN message once）

	Name	Instruction format	Suitable for
LD	CAN_WRITE		• KS • KW103 • KW203

Operands	Input/Output	Data Type	Acceptable Memory Areas
CH	Input	INT	Constant
ID	Input	DWORD	L、M、V、Constant
FMT	Input	BYTE	L、M、V、Constant
DATA	Input	BYTE	L、M、V
LEN	Input	BYTE	L、M、V、Constant
DONE	Output	BOOL	L、M、V
ERR	Output	BOOL	L、M、V

Operands	Description
EN	The Enable Bit.
CH	CAN interface used. 0 for CAN1, 1 for CAN2, 2 for K541 module
ID	The ID number of the message which to be sent.
FMT	Message format. 0 represents standard frames and 1 represents extended frames.
DATA	The address of the beginning byte where the data is to be sent.
LEN	Length of data to be sent. Unit: bytes.
DONE	Whether the message has been sent. DONE is set to 0 when executing, and DONE is set to 1 after sent.
ERR	Whether the message was sent incorrectly. If the send fails (usually because the send buffer is full), ERR is set to 1.

Note: ID, FMT, and LEN must be both constant and variable. The DATA and LEN parameters make up a variable length memory block that must all be in a valid memory region, otherwise the result

is not expected.

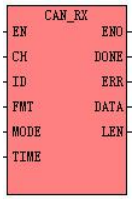
CAN message to be sent is specified by ID number, format (FMT, indicating extension frame or standard frame), DATA (indicating starting address of message DATA storage) and length LEN.

The ascending edge jump of EN input will trigger the execution of this instruction once, and the sent message will be written into the sending buffer inside PLC, and then sent out through the specified CAN interface CH by PLC scheduling.

If the instruction successfully writes the message into the send buffer, the execution is completed, and the DONE will be set to 1. If the buffer is full(sending fails), and the instruction will sets both DONE and ERR to 1.

The total number of CAN_TX and CAN_WRITE instructions allowed in a project is up to 64!

7.6.3.4 CAN_RX（Receive specific ID CAN message）

	Name	Instruction format	Suitable for
LD	CAN_RX		• KS • KW103 • KW203

Operands	Input/Output	Data Type	Acceptable Memory Areas
CH	Input	INT	Constant
ID	Input	DWORD	L、M、V、Constant
FMT	Input	INT	L、M、V、Constant
MODE	Input	INT	L、M、V、Constant
TIME	Input	INT	L、M、V、Constant
DONE	Output	BOOL	L、M、V
ERR	Output	BOOL	L、M、V
DATA	Output	BYTE	M、V
LEN	Output	BYTE	L、M、V

Operands	Description
EN	The Enable Bit.
CH	CAN interface used. 0 for CAN1, 1 for CAN2, 2 for K541 module
ID	The ID number of the message which to be sent.
FMT	Message format. 0 represents standard frames and 1 represents extended frames.
MODE	Receive mode. 0 represents the always receive mode, and 1 represents the single receive mode
TIME	Timeout ,Unit in ms.
DONE	In single receive mode, DONE is the receive success flag bit.
ERR	Receive the timeout flag bit.

DATA	The first address of the data stored in the last received message.
LEN	The data length of the last received message,Unit in bytes.

Note: ID, FMT, MODE and TIME parameters must be both constant and variable. The DATA and LEN parameters form a memory block of variable length, which must all be in the legal memory area, otherwise the result is unpredictable.

The PLC maintains a receiving message filter list automatically. The CPU will filter the received CAN message, and only the message ID and format (standard frame or extended frame) matching the list value will be received by this instruction.

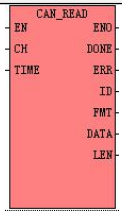
The CAN_RX instruction is used to add a message to the filter list, which is determined by the specified ID number (ID, CAN message ID) and format (FMT, indicating extended frame or standard frame).

The *MODE* parameter indicates the receiving MODE. If the MODE is 1, it is a single receiving MODE. The instruction only receives the specified message once, and exits after receiving it. If MODE is 0, it is a permanent receive MODE, and the instruction will always receive the specified message.

The rising edge signal at the *EN* input triggers the execution of the instruction. After the execution of this instruction, the specified ID number (*ID*) and format (*FMT*) values will be added to the receiving filter list immediately, and the PLC will enter the receiving state immediately, and the DONE and ERR will be cleared to 0 at the same time.If it is a single receive mode, then if the specified message is received within TIME, the DONE will be set to 1 and the instruction will exit the receiving state. If the specified message is not received within TIME, the DONE and ERR will be set to 1 and the instruction will exit the receiving state； If it is a permanent receiving mode, this instruction will always monitor the CAN interface CH and receive all the designated messages after starting. If the designated message is not received again in TIME after a successful receiving, ERR will be set to 1. If the designated message is successfully received again, ERR will be cleared to 0.Therefore，**In permanent receive mode, each CAN_RX instruction needs to be executed only once instead of repeatedly！**

A maximum of 64 CAN_RX instructions can be invoked in a project！

7.6.3.5 CAN_READ（Receive CAN message once）

	Name	Instruction format	Suitable for
LD	CAN_READ		• KS • KW103 • KW203

Operands	Input/output	Data Type	Acceptable Memory Areas
CH	Input	INT	Constant（1 Or 2）
TIME	Input	INT	L、M、V、Constant
DONE	Output	BOOL	L、M、V
ERR	Output	BOOL	L、M、V
ID	Output	DWORD	L、M、V

FMT	Output	BYTE	L、M、V
DATA	Output	BYTE	M、V
LEN	Output	BYTE	L、M、V

Operands	Description
EN	The Enable Bit.
CH	CAN interface used. 0 for CAN1, 1 for CAN2, 2 for K541 module
TIME	Timeout. After beginning to recive message, if no message is received within the specified time, the receiving state will exit over time and ERR will be set to 1
ID	The ID number of the message which to be sent.
FMT	Message format. 0 represents standard frames and 1 represents extended frames.
DATA	The address of the beginning byte where the data is to be sent.
LEN	Length of data to be sent. Unit: bytes.
DONE	Whether the message has been sent. DONE is set to 0 when executing, and DONE is set to 1 after sent.
ERR	Whether the message was sent incorrectly. If the send fails (usually because the send buffer is full), ERR is set to 1.

Note: the DATA and LEN parameters make up a variable length memory block, which must all be in valid memory regions, otherwise the result is not expected.

The EN input ascent jump will trigger the execution of this instruction: start the receiving state to receive any message from the specified CAN interface CH.

Begin to receive message, if receive a CAN message within the specified timeout TIME , the PLC would according to message,set the Output parameters such as ID (ID number), FMT (format, indicate the received is extended frame or standard frame), DATA (the received the starting address of the packet DATA storage), LEN (length).Meanwhile,Set DONE set to 1 after finish receiveing.

If no message is received within the specified timeout period, the received status will be quitted with DONE and ERR set to 1 both.

When started the CAN_READ Instruction ,It would receives any message of the CAN interface, so be careful do **NOT** mixing with other protocols, such as CANOpen.

Be careful when mixing.

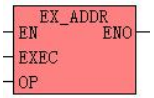
Note：1) CAN_READ instruction has a lower priority than CAN_RX instruction.

2) After CAN_READ instruction is started, any message on the bus will be received by it. Therefore, if multiple CAN_READ instructions are called in the program, the last one instruction will take effect.

7.6.4 Extend Bus Instruction

The extension bus instruction is located in the CAN instruction group .

7.6.4.1 EX_ADDR（Modify the extension module configuration）

	Name	Instruction format	Suitable for
LD	EX_ADDR		• KS • KW103 • KW203

Operands	Input/output	Data Type	Acceptable Memory Areas
EXEC	Input	BOOL	M、V、L、SM
OP	Input	INT	M、V、L、Constant

Operands	Description
EXEC	The Enable bit.If EN is 1, the instruction is enabled, allowing execution.
OP	Operation code. 181 --- Command all extension modules to save their ID and parameters. 99 ---Command all extension modules to clear their ID and parameters.

This instruction sends the appropriate command to the extension module based on the OP value：

- If the OP value is 181, the extension module will automatically save its ID and various parameters (such as signal form, filtering mode, etc.) after receiving the command.When it is powered on again later, the expansion module will automatically read the saved data and enter into the running state, which does not require CPU configuration, so it can power on or off independently of the CPU at any time.
- If the OP value is 99, the extension module will clear the saved ID and parameters after receiving the command, and will wait for the CPU module to assign the ID and configure the parameters after re-power on.

➢ **LD instruction description**

If EN is 1, the instruction is scanned and execution is started once if the rising edge of EXEC is detected.
If EN is 0, the instruction is not scanned and will not be executed.