CONTENT

Chapter 1 General Introduction

1.1 Summary

Kinco KS series PLC is a small and integrated PLC .It is Kinco new thin and high performance PLC.

Based on high performance, high reliability and powerful functions of K5/K2, KS series use higher level CPU. KS has CANopen port, higher speed input and output, small size for installation.It can meet more user's requirement.

1.2 Product List

Name	Type	Description
CPU Module		
CPU101	KS101M-04DX	DC 24V, DI 4*DC24V，1*MicroUSB,1*Ethernet 1* RS232(programming port),1*RS485, 2*CAN Expandable(max 14 modules)
CPU105	KS105-16DT	DC 24V, DI 8*DC24V，DO 8*DC24V. 1* RS232(programming port),1*RS485. Expandable(max 14 modules)
	KS105C1-16DT	DC 24V, DI 8*DC24V，DO 8*DC24V 1* RS232(programming port),1*RS485,1*CAN
	KS105C2-16DT	DC 24V, DI 8*DC24V，DO 8*DC24V 1* RS232(programming port),1*RS485,2*CAN Expandable(max 14 modules)
Expansion modules		
PM121	KS121-16DX	DC24V , DI 16*DC24V,Modbus slave
PM122	KS122-12XR	DC24V, DO 12*relay, Modbus slave
PM122	KS122-14DT	DC24V，DO 14* Transistor,Modbus slave
PM123	KS123-14DR	DC24V, DI 8*DC24V, DO 6*relay, Modbus slave

| PM131 | KS131-04RD | 4-channel RTD input, two-wire, three-wire or four-wire， PT100、PT1000、Cu50、R,Modbus slave |
| PM133 | KS133-06IV | DC24V, 4 analog input/2 analog output
4-20mA/1-5V/0-20mA/0-10V as option, Modbus slave |

1.3 Environmental Condition

Kinco-KS 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-KS to work properly. It is the user's responsibility to ensure that the service conditions are not exceeded.

Transport and storage		
Ambient conditions	temperature	-40 --- +70 °C
	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 conditions	air temperature	Open equipment : -10 --- +55°C; Enclosed equipment: -10 --- +40°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 3 mutually perpendicular axes.

	Electrostatic discharge	±4kV Contact, ±8kV Air. Performance criteria B.
Electromagnetic compatibility (EMC)	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 B.
	Fast transient bursts	main power: 2KV, 5KHz. I/Os and Communication port: 1KV, 5KHz. Performance criteria B.
	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		IP20

Chapter 2 CPU Module Introduction

2.1 Overview

2.1.1 Structure
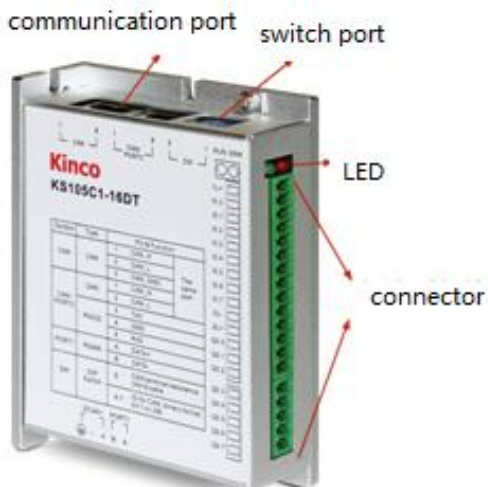
2.1.2 CPU Types

Kinco-KS provides different CPU models with a diversity of features and capabilities, all the CPU use DC24V power supply. The following table describes main technical data of each CPU model.

Parameters	KS101M-04DX
Power supply	
Rated voltage	DC24V Note: The USB port can also be directly powered for CPU operation.
Voltage range	DC20.4V－28.8V
Rated power	5W
I/O	
Digital	4*DI
Expansion	14
Programming port	USB2.0 (micro USB)
Ethernet	1，support programming protocol, Modbus TCP Server
CAN	2 *CAN CAN1 can support Extended protocol and free protocol CAN2 can support Kinco Motion control instruction、CANopen master and free protocol.
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
High speed counter	2,Max 200KHz,support single and double
Storage	
Programming	Max 8K bytes instruction
Data	M area 4K bytes；V area 16K bytes
Data backup	E2PROM，The last 1K bytes of the V area, Persistent storage.
Data retention	V area. Lithium battery ，3 years at normal environment
Other	
Timer	256

Timer interruption	2，0.1ms
Counter	256
RTC	yes，the difference is 5 min/month at 25℃

Parameters	KS105-16DT	KS105C1-16DT	KS105C2-16DT
Power supply			
Rated voltage	DC24V		
Voltage range	DC20.4V – 28.8V		
I/O			
Digital	8*DI / 8*DO		
Analog	--		
Expansion	14	---	14
CAN	---	Support Kinco Motion control instruction、CANopen master、slave and free protocol	CAN1 can support Extended protocol and free protocol CAN2 can support Kinco Motion control instruction、CANopen master and free protocol.
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		
High speed counter	4, Max 200KHz,support single and double		
High speed output	4 Channel 0&1&2 Max 200KHz（load resistance is less than 1.5KΩ at highest frequency）. Channel 3 Max 10KHz		
Interrupt	4，I0.0-I0.3 interrupt up and down		

Storage	
Programming	Max 4K bytes instruction
Data	M area 1K bytes；V area 4K bytes
Data backup	E2PROM，448 bytes
Data retention	1K bytes. Lithium battery ，3 years at normal environment
Other	
Timer	256 1ms ：4 10ms ：16 100ms ：236
Timer interruption	2，0.1ms
Counter	256
RTC	yes，the difference is 5 min/month at 25℃

2.2 Functions

2.2.1 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 KS provides two ways for manually changing the CPU status: Using the operation switch (RUN/STOP); Executing [Debug] -> [RUN] or [STOP] menu command in Kincobuilder.

Usually when the PLC is power on, the PLC status is based on switch and Kincobuilder

[RUN] or [STOP]. If switch and Kincobuilder are RUN, the PLC real status is RUN. The PLC real status is STOP for all other situation.

In below situation, the PLC status depend on single control way.

(1) The PLC status only depend on Kincobuilder RUN/STOP setup. The switch won't be valid.

    a— PLC RUN mistake (strong mistake) will stop the PLC

    b— The user use Kincobuilder [setup], PLC is RUN/STOP status

    c— Users use STOP instruction to stop PLC

    d— If downloading project failed, PLC will keep STOP status. Until below situation happen.

（2）PLC status only depend on switch, Kincobuilder RUN/STOP won't be valid.

    a—PLC is power on again. PLC status depend on the switch

    b—Kincobuild execute clean PLC instruction. After cleaning PLC status depend on switch.

    c—When downloading, if switch is RUN, then PLC will be defaulted STOP for downloading. After downloading, PLC will be RUN again. If switch is STOP,PLC will be STOP after downloading.

    d—Anytime switch can change PLC RUN/STOP status

➢ **CPU Status LED**

The CPU module provides 2 status LEDs: **RUN**, and **Err.**

Run, Err LEDs show the CPU operation status.

    【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.

KS separates errors into three levels: Fatal error, Serious error, Normal error. When CPU detects an error, it will use different way to handle according to error level and turn on Err LED, then it will save the error code in sequence for user analysis.

2.2.2 Programming port and serial port

KS provides 2 communication ports,PORT0 and PORT1.It supports baud rate up to 115.2kbps.PORT0 can be used as programming port and also supports Modbus RTU slave protocol and free protocol.PORT1can be used as programming port and also supports Modbus RTU protocol (as a slave or master) and free protocol.

RJ45 port. Pins and functions as below:
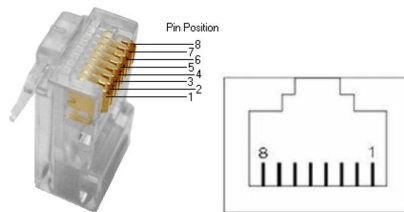
⚠RS232 can't insert and release with power. So we should turn off power of CPU or PC, otherwise it will break ports.

2.2.3 Ethernet communication port

KS101M-04DX products provide Ethernet interfaces that comply with the standard IEEE802.3 specifications. This interface supports the programming protocol and can be used as a programming port. In addition, it also supports MODBUS TCP Sever, commonly known as the "slave" function.

The communication cable can be a straight-through cable (straight wire) or a cross cable (cross wire). The Ethernet interface of K2 provides an "auto-negotiation" function. When a cable is inserted, K2 will automatically negotiate with the communication partner to determine the type of cable used.

Users can modify the parameters of the Ethernet interface through the USB port, serial port or Ethernet port itself: execute the [Tool]-[TCP / IP parameter configuration] menu command in the KincoBuilder software, the following dialog box will pop up, and the user can read Or modify the parameters.
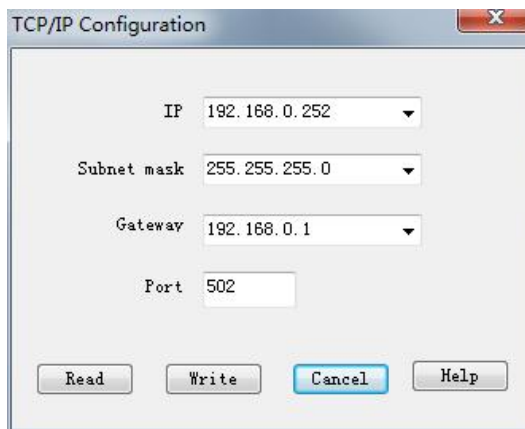
In a local area network, PLCs and PCs located in the same network segment (that is, the first three numbers of the IP address must be the same and the last one different) can communicate with each other.

2.2.4 CAN port

KS105C1-16DT has 1 CAN port , CAN.It can support Support Kinco motion control instructions CANopen master、slave and free protocol.

KS105C2-16DT /KS101M-04DX has 2 CAN ports, CAN1 and CAN2. CAN2 can support Kinco Motion Control Instructions CANopen master and free protocol.CAN1 can support Extended protocol (connectable expansion module - up to 14 expansion modules) and free protocol.

2.2.5 Expansion modules

KS105-16DT has expansion port, it can connect KS series expansion modules

CAN1 port of KS105C2-16DT/KS101M-04DX can work as expansion port, also it support protocol. Users can use them directly without setup, PLC can identify it automatically.(only one between expansion or CANopen communication)

CPU can connect expansion modules with standard double wires directly. Expansion modules have one input port and one output port. It is convenient to connect multiple expansion modules. Pls reference below figure.

We don't need to setup address for connecting expansion modules. The real address in the net depend on the turn that KincoBuilder configurate hardware.

2.2.6 High Speed Counter and High Speed Pulse Output

KS 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 can support up to 200KHz(Include single phase and AB phase).

KS105 provides 4 high speed pulse outputs(Q0.0,Q0.1 and Q0.4, Q0.5).All support PTO and PWM.Q0.0 and Q0.1 ,Q0.4support up to 200KHz (The resistor of load should be less than 1.5KΩ),Q0.5 supports up to 10KHz.

2.2.7 Edge Interrupts

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.

2.2.8 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 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.

Note: Because E^{2}PROM 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.KS 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.

2.2.9 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.

2.2.10 Backup Battery

KS 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 in figure, user can order the battery separately.

2.3 Wiring diagram

CAN1
CAN (in RJ45)

	Pin	Function
	1	CAN_H
CAN1	2	CAN_L
	3	CAN_GND

The CAN1 can also used as expansion bus

CAN2/PORT0
(in RJ45)

	Pin	Function
	1	CAN_H
CAN2	2	CAN_L
	3	CAN_GND
Port0 RS232	4	GND
	6	TXD
	8	RXD

Ethernet

MicroUSB

KS101M-04DX

DI :Real line means source input.
Broken line means sink input.

DC24V

1M
I0.0
I0.1
I0.2
I0.3

PORT1
B
A

B
A

0V
24V

DC24V
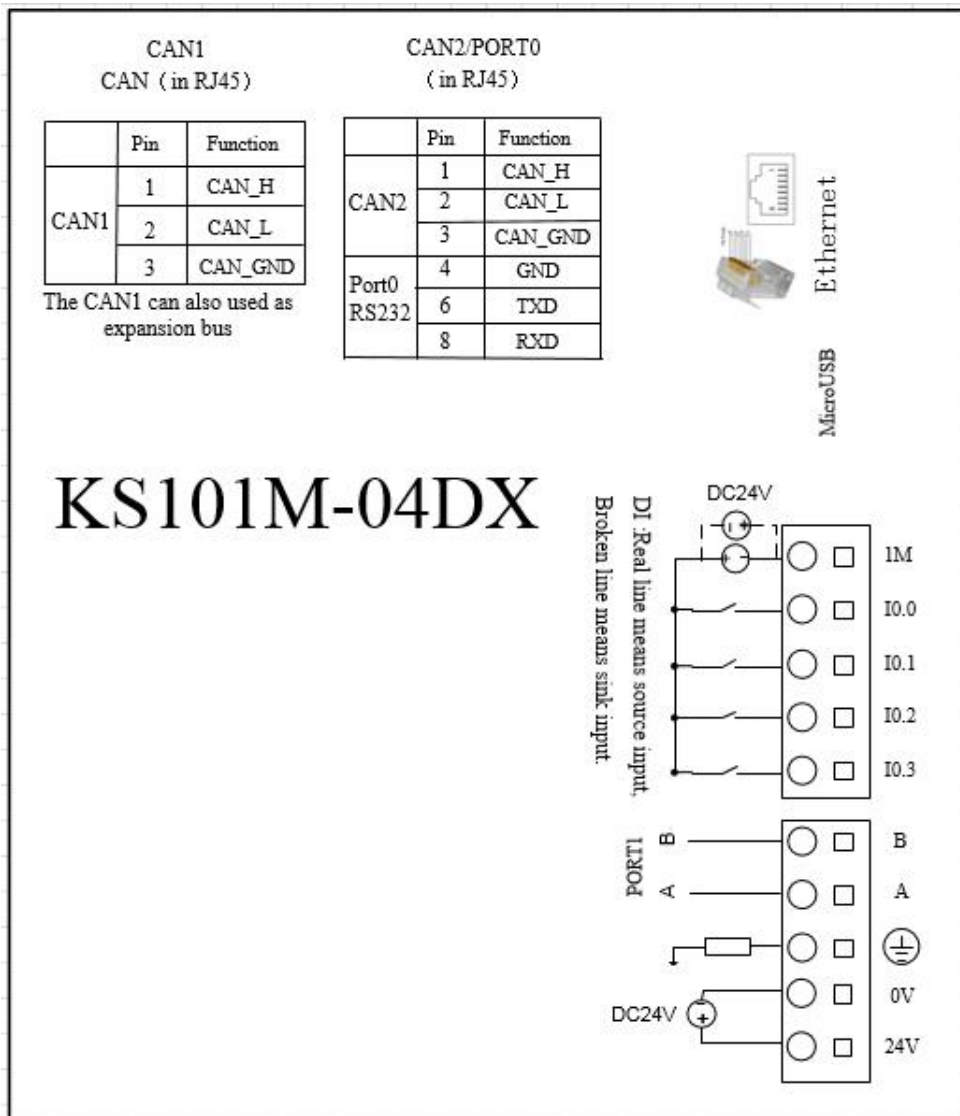
Ext.
Expansion Bus（in RJ45）

PORT0
RS232 (in RJ45)

	Pin	Function
RS232	6	RXD
	3	TXD
	4	GND

KS105-16DT

DC24V

D1 :Real line means source input, Broken line means sink input.

1M
I0.0
I0.1
I0.2
I0.3
I0.4
I0.5
I0.6
I0.7
2L-

DC24V

2L+

DO: transistor, source type.

Q0.0
Q0.1
Q0.2
Q0.3
Q0.4
Q0.5
Q0.6
Q0.7

RS485
B A

DC24V

0V   24V   B   A
PORT1
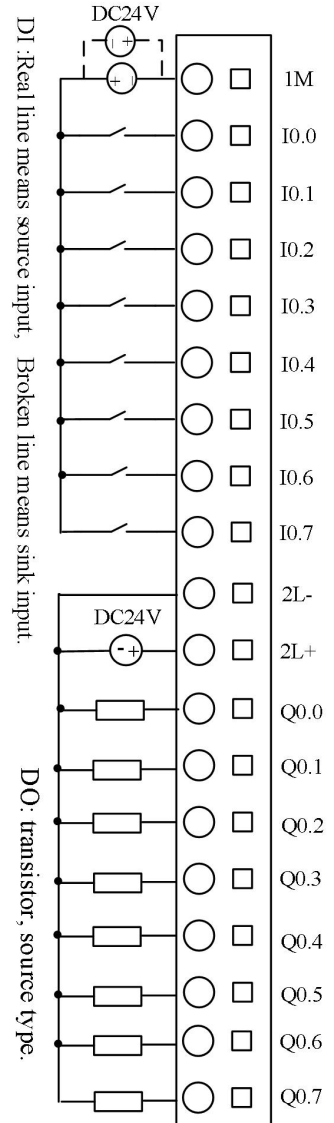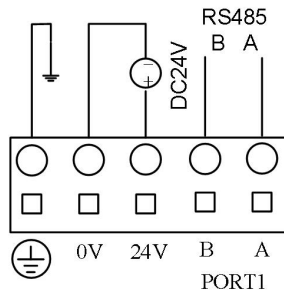
CAN
CAN (in RJ45)

	Pin	Function
CAN	1	CAN_H
	2	CAN_L
	3	CAN_GND

PORT0
RS232 (in RJ45)

	Pin	Function
RS232	6	RXD
	3	TXD
	4	GND

There is one same CAN in the two RJ45 interfaces.

KS105C1-16DT

DI :Real line means source input, Broken line means sink input.

DC24V

1M
I0.0
I0.1
I0.2
I0.3
I0.4
I0.5
I0.6
I0.7
2L-

DC24V
2L+

DO: transistor, source type.

Q0.0
Q0.1
Q0.2
Q0.3
Q0.4
Q0.5
Q0.6
Q0.7

RS485
B   A

DC24V

0V   24V   B   A
PORT1
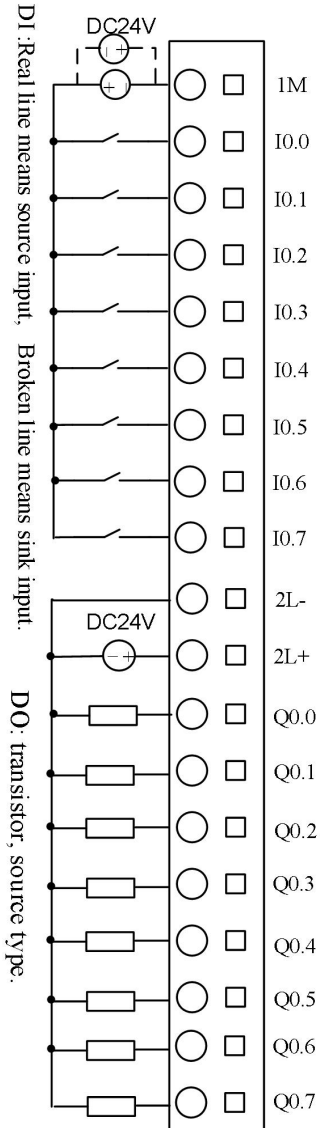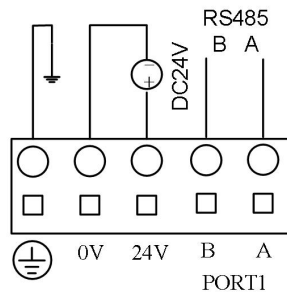
CAN1
CAN（in RJ45）

	Pin	Function
CAN	1	CAN_H
	2	CAN_L
	3	CAN_GND

The CAN1 can also used as expansion bus

CAN2/PORT0
(in RJ45)

	Pin	Function
CAN	1	CAN_H
	2	CAN_L
	5	CAN_GND
RS232	6	RXD
	3	TXD
	4	GND

KS105C2-16DT

DC24V

DI :Real line means source input, Broken line means sink input.

1M
I0.0
I0.1
I0.2
I0.3
I0.4
I0.5
I0.6
I0.7
2L-

DC24V

2L+

DO: transistor, source type.

Q0.0
Q0.1
Q0.2
Q0.3
Q0.4
Q0.5
Q0.6
Q0.7

RS485
B A

DC24V

0V 24V B A
PORT1
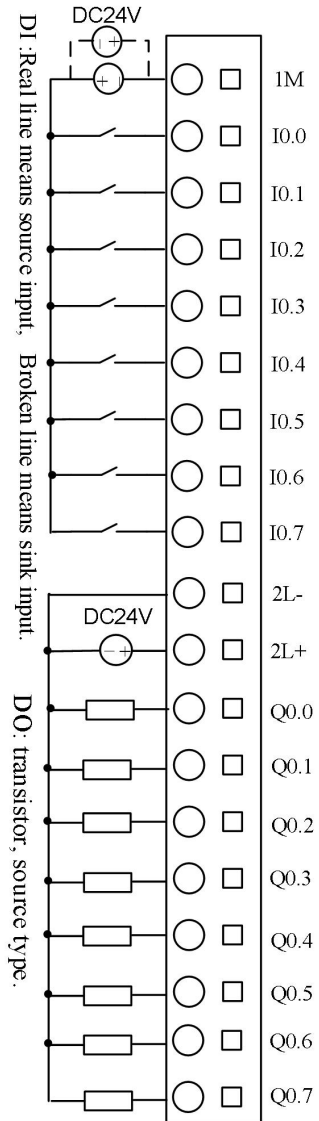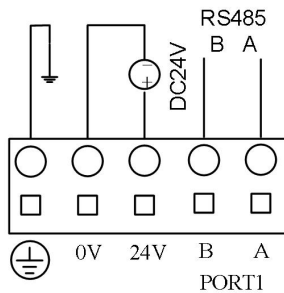
2.4 Dimension
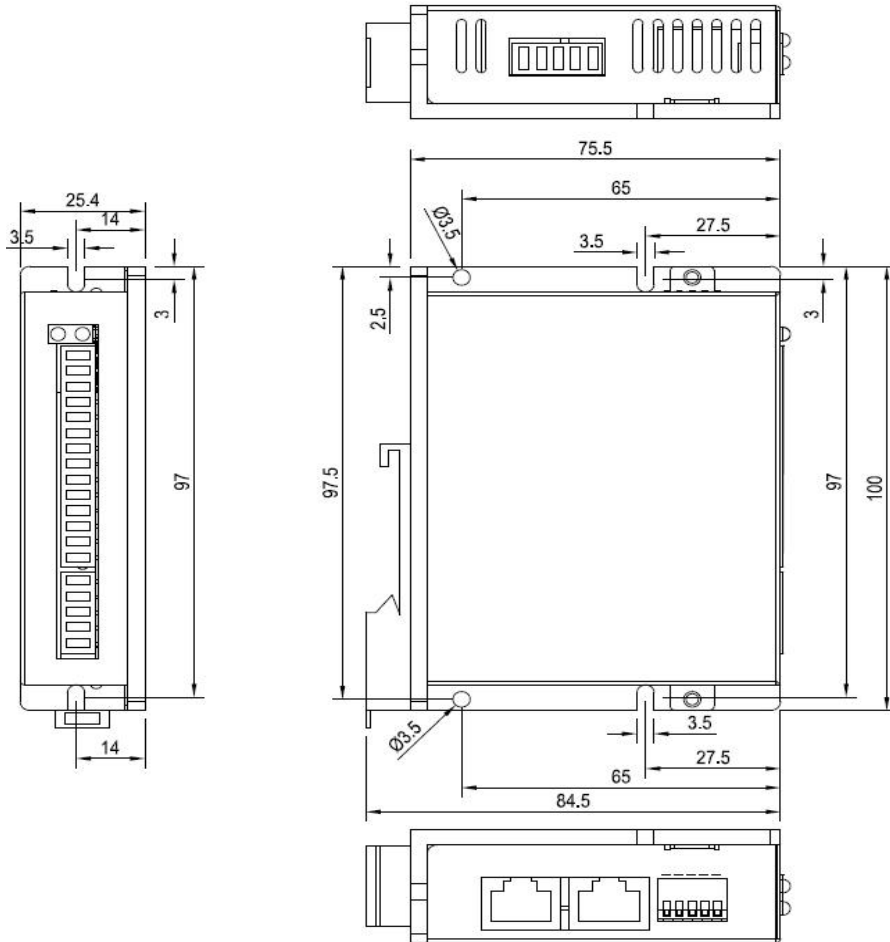
2.5 Technical Specification

➢ 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	Common channel:11V@2.0mA
Input filter time delay · off-to-on · on-to-off	 1.2μs 0.5μs;
Isolation between input and internal circuit · Mode · Voltage	 Opto-electrical isolation 500VAC/1 min

➢ DO Specifications(Transistor type)

Output type	Source
Rated power supply voltage	DC24V,allowance range: DC20.4V—DC28.8V.(Same as power supply) Note:Limited by size, The supply of transistor output has no filtering circuit and protected-circuit. If the voltage exceeds the allowable range,It might damage the internal compinents; When the power supply are not stable, please make sure the voltage between 1L+ and 1L- are within allowable range,which average voltage should not over 28.8V , Maximum peak voltage should not over 32V.
Output current per channel	Rated current:200mA,max.300mA @24VDC
Instant impulse current per channel	1A,less than 1s
Output leakage current	Max.0.5цA
Output impedance	Max. 0.2Ω
Output delay · off-to-on · on-to-off	 Common channel: 12μs; HSC channel: 0.5μs; Common channel: 35μs; HSC channel: 1μs;
Protection： ·Reverse polarity protection of power supply	 No Yes

·Inductive load protection	Yes
·Short-circuit protection	Yes, less than 10s.
·Reverse polarity protection of output	
Isolation between output and internal circuit · Mode · Voltage	Opto-electrical isolation 500VAC/1 min

Chapter 3 Expansion modules

3.1 Introduction

KS series can expand with expansion modules. The power supply for KS expansion modules is 24VDC. So we don't need power supply from expansion CANopen.

KS expansion modules have one standard RS485 port. It can work as Modbus slave.

Note: Some address of Modbus RTU master start from 1,so the data in the below form should add 1 directly.

Type	Mold	MODBUS coder	Area	MODBUS Address
KS122-14DT	DO（0XXXX）	01，05，15	Q0.0-Q1.5	0-13
KS122-12XR			Q0.0-Q1.3	0-11
KS123-14DR	DI（1XXXX）	02	I0.0 --- I0.7	0-7
	DO（0XXXX）	01，05，15	Q0.0-Q0.5	0-5
KS133-06IV	AI（3XXXX）	04	AIW0 --- AIW6	0-3
	AO（4XXXX）	03，06，16	AQW0 --- AQWQ	0-1
KS121-16DX	DI（1XXXX）	02	I0.0 --- I1.7	0-15
KS131-04RD	AI（3XXXX）	04	AIW0 --- AIW6	0-3

3.2 DO Expansion module

3.2.1 DO 14×transistor

The order number of the module is ： Kinco-KS122-14DT.

This is a DO module with 14 channels, where the DO 14× transistor.

3.2.1.1 Wiring diagram

3.2.1.2 Technical data

Rated power supply	DC 24V，≥100mA
Electrical parameters	
Number of output channels	14（8 channels/group）
Output type	Source type
Rated output voltage	DC24V. Allowable range: DC20.4V—DC28.8V (consistent with supply voltage).
Output current of per channel	Max 500mA@24VDC
Output leakage current	Max 0.5цA
Output impedance	Max 0.2Ω
Output delay time·connect delay · Disconnect delay	0.3--5μs 5μs
Protective function: ·Power supply polarity protection ·Inductive load output protection ·Short circuit protection ·Output polarity reverse protection	Yes Yes Yes Yes，Allow reverse polarity signal to be added to the output for no more than 10s.
Isolation between　Output and internal logic Circuit · Isolation method · Isolation voltage	Photocoupler 500VAC/1 minute
Size and weight	
Size (length × width × height)	100×84.5×25.4mm
weight	200g

3.2.2 DO，DO 12×Relay

Type：Kinco-KS122-12XR.

This modules is DO 12*Relay.

3.2.2.1   Wiring
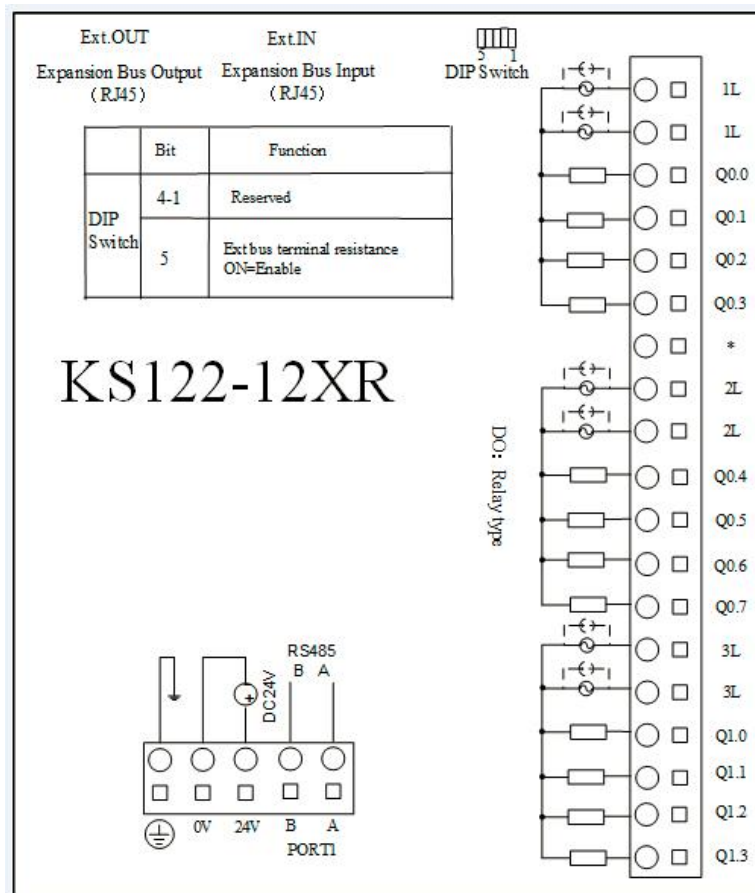
3.2.2.2 Technical data

Rated power supply	DC 24V，≥100mA
Electrical parameters	
Number of output channels	12 relay（4 channel/group）
Allowable load voltage	DC 30V/AC250V
Allowable load current	2A（DC 30V/AC250V）
Maximum output current per group	10A
Output turn-on delay time	10ms（MAX）
Output off delay time	5ms（MAX）

Relay contact life expectancy · Mechanical life (no load) · Electrical life (rated load)	20，000，000 time（1200/minute） 100，000 time（6/minute）
Output isolation characteristic · Isolation method · Isolation voltage of the coil and the contact	Relay 2000Vrms
Size and weight	
Size (length × width × height)	100×84.5×25.4mm
Weight	200g

3.3   DI Expansion module

3.3.1 DI 16×DC24V

The order number of the module is：Kinco-KS121-16DX.

This is a DI module with 16 channels, where DI 16 x DC24V.

3.3.1.1 Wiring diagram

	Bit	Function
DIP Switch	4-1	Reserved
	5	Ext bus terminal resistance ON=Enable

KS121-16DX
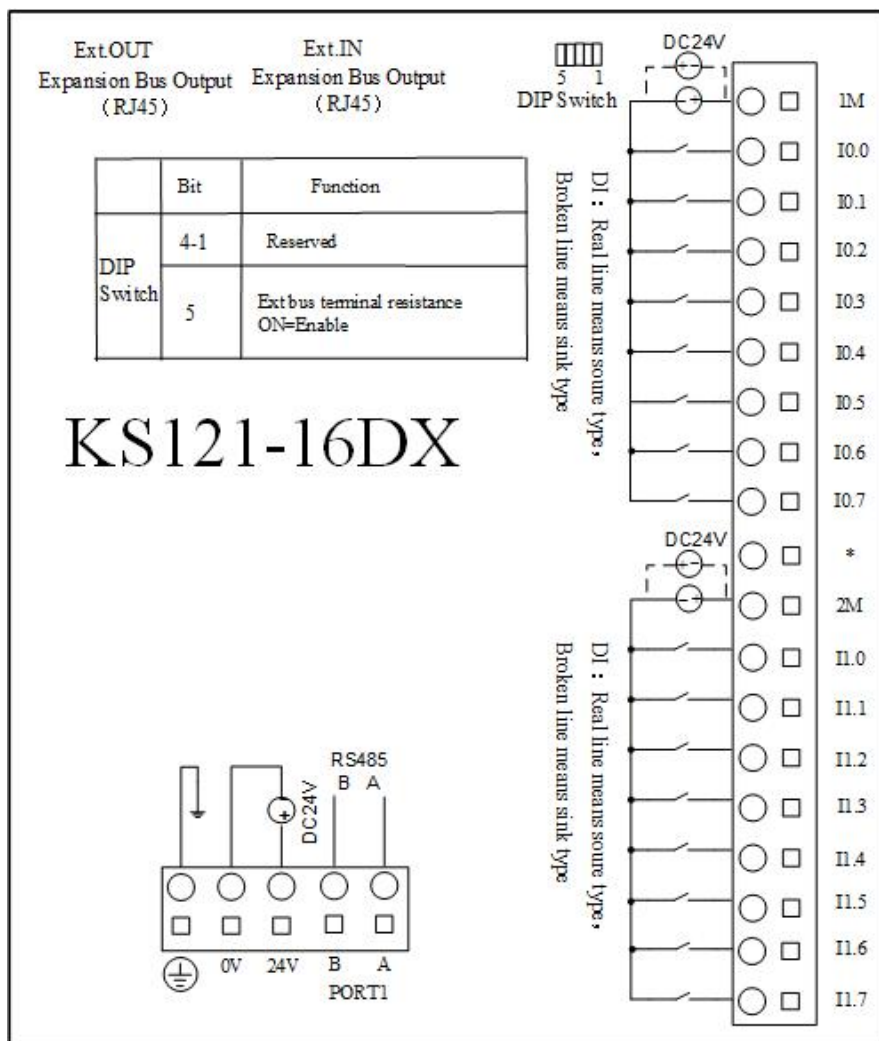
3.3.1.2 Technical data

Rated power supply	DC 24V，≥100mA
Electrical parameters	
Number of input channels	16（8 Channel/group）
Input type	NPN/PNP
Rated input voltage	DC 24V
Rated input current	3.5mA@24VDC
Logic "0" maximum input voltage	5V
Logic "1" minimum input voltage	11V@2.0mA
Input delay time ·Connection delay ·Disconnection delay	12μs 40μS
Input isolation from internal logic ·Isolation method ·Isolation voltage	Photocoupler 500VAC/1 minutes
Size and weight	
Size (length × width × height)	100×84.5×25.4mm
Weight	200g

3.4　　DI/O Expansion module

3.4.1 DI/O，　DI 8×DC24V　DO 6×Relay

Type：Kinco-KS123-14DR.

This module is 14IO,DI 8*DC24V,DO 6*Relay.

3.4.1.1 Wiring

KS123-14DR

	Bit	Function
DIP Switch	4-1	Reserved
	5	Ext bus terminal resistance ON=Enable
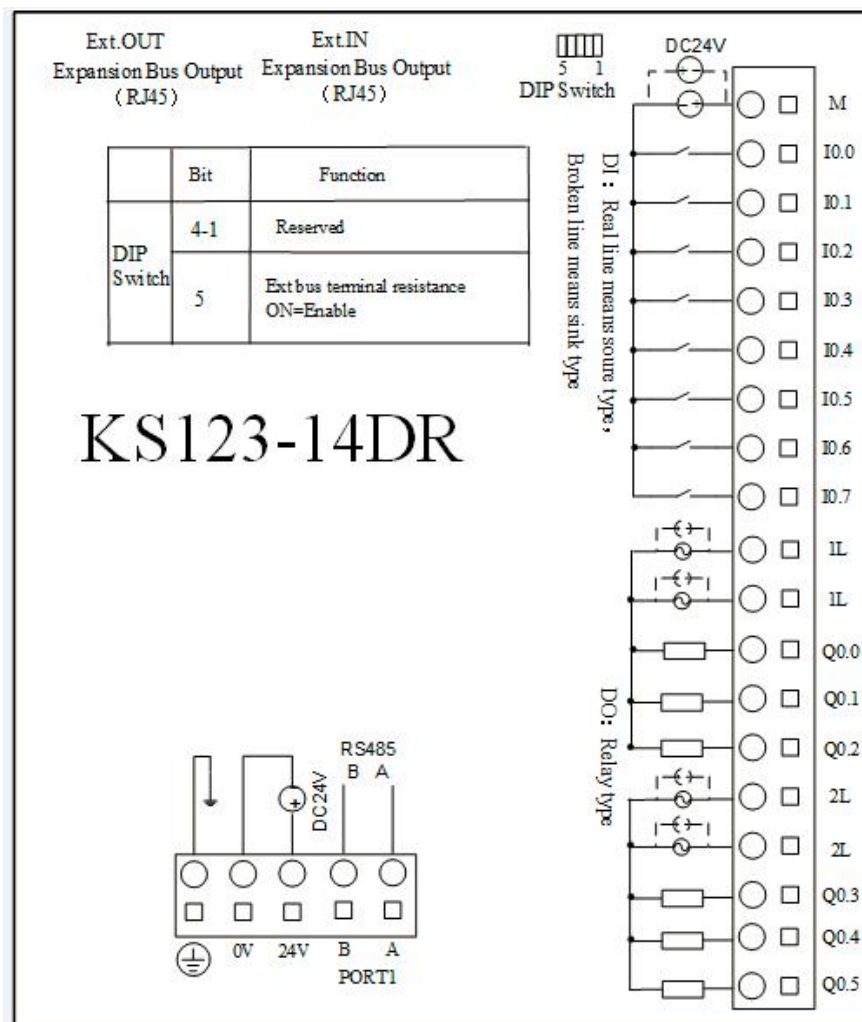
3.4.1.2　Technic specification

Rated voltage	DC 24V，≥100mA
Electric parameters	
input	8（8 channel/group）
Input type	source/drain
Rated voltage	DC 24V
Rated input current	3.5mA@24VDC
Digital "0"max. input voltage	5V
Digital"1"min. input voltage	11V@2.0mA
Isolation between input and inside power ·isolation type ·isolation voltage	Photoelectric couple 500VAC/1 minutes
Output	6 relay（3channel/group）
Max. loading voltage	DC 30V/AC250V
Loading current	2A（DC 30V/AC250V）
Max. output current	10A
Delay time for output on	10ms（max.）
Delay time for output off	5ms（max.）
Life of relay ·Mechanic life（no loading） · Electric life（rated loading）	20，000，000 times（1200times/min） 100，000times（6times/min）
Output isolation · isolation mode · isolation voltage	relay 2000Vrms
Measurement and weight	
Measurement(length ×width× height)	100×84.5×25.4mm
Net weight	200g

3.5 AI Expansion module

3.5.1 AI 4×RD Thermal resistance input

The order number of the module is：Kinco-KS131-04RD.

The module has 4 channels, which can be connected to the thermal resistance (Pt100, Pt1000, Cu50, R) to measure the temperature, or directly measure the resistance of the resistor. The signal form of the channel is configured in the programming software. Each channel can be mixed with different resistance models and supports two-wire and three-wire versions.

3.5.1.1 Wiring diagram
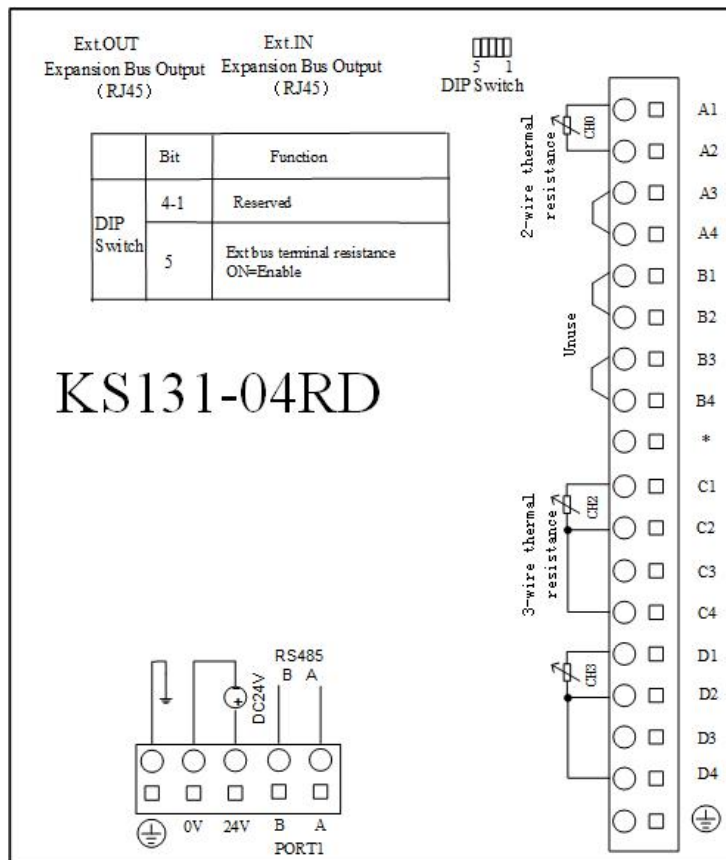
3.5.1.2 Measurement range and measured value representation format

The input signal of each channel is sampled and calculated by the ADC to obtain the resistance value, and then the corresponding temperature value is calculated according to the temperature-resistance characteristic formula of the selected thermal resistance, and sent to the AI area of the CPU module via the expansion bus for the user program. access.

Various signal forms have a certain measurement range. If the measured value exceeds the measurement range, the AI value will remain at the corresponding upper or lower limit value, and the module will alarm and send a fault report to the CPU module through the expansion bus.**It is recommended that the user short-circuit the terminals of the unused channels and configure their signal form to [Resistance] in the programming software, then these unused channels will not cause an alarm.**

The following table shows the measurement range and measured value representation format, where T is the measured temperature value and R is the measured resistance value..

Signal form	Measuring range	Measured value representation format
Pt100	-200～850℃	
Cu50	-50～150℃	T×10
Pt1000	-50～300℃	
Resistance	0～2000Ω	R×10

3.5.1.3 Technical data

Rated power supply	DC 24V，≥100mA
Electrical parameters	
Number of channels	4
Signal form	Pt100、Cu50、Pt1000、Resistance
Wiring form	Two-wire or three-wire system
Resolution (with sign bit)）	24 bit
measurement accuracy	temperature：±0.6℃；　resistance：±1Ω
Conversion rate (per channel)	About 1 time/second
Size and weight	
Size (length × width × height)	100×84.5×25.4mm
Weight	200g

3.6　AI/O Expansion module

3.6.1　AI/O, AI 4×IV　AO 2×IV, current/voltage input/output

type：Kinco-KS133-06IV.

This module have 4*AI and 2*AO. Measure and output standard voltage and current signal　（4-20mA、1-5V、0-20mA、0-10V. All channel in one module can choose different signal.

3.6.1.1 Wiring

KS133-06IV
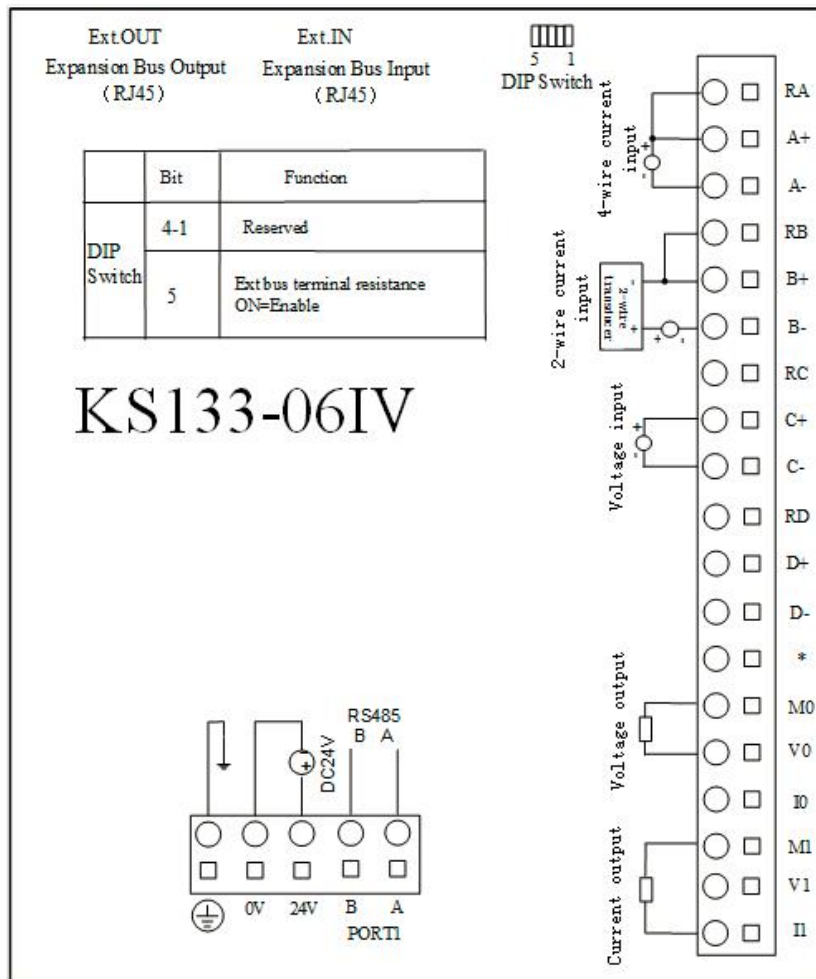
3.6.1.2 measure range and measure value for AI

The input signal in each channel will sample ADC and counter. The results will be send to CPU AI area from expansion CAN. Then user programming can visit it.

All the signal types have detection range. If the value is over range, the modules will warn, and the LED will be on. Meanwhile it will send problem file to CPU by expansion module. **Pls connect all the channels that is not used, also setup signal type to 【0-20mA】 or 【0-10V】,then these channels won't warn.**

Below is detection range and value format. I is input current, V is input voltage.

Signal type	Range	Format
4-20mA	3.92-20.4mA	I×1000
0-20mA	0-20.4mA	
1-5V	0.96-5.1V	V×1000
0-10V	0-10.2V	

3.6.1.3 AO output range and value format

AQ output will be send AO modules by expansion bus, then counter and change. After that it will output from channel by DAC.

The output range of all signal types is limited. If the output is over range, the output will keep the up/down limited value.

Below form is output range and value format., I is real current, V is real voltage

Signal types	Output range	Ourput value
4-20mA	3.92-20.4mA	I×1000
0-20mA	0-20.4mA	
1-5V	0.96-5.1V	V×1000
0-10V	0-10.2V	

3.6.1.4 Technical specification

Electric parameters	
AI channels	4
Rated supply current	DC 24V，≥100mA
Signal type	4-20mA、1-5V、0-20mA、0-10V
Resolution	12 bit
Accuracy	0.3% F.S.
Signal limitation	Current is not over 24mA，voltage is not over 12V
Change speed（each channel）	15 times /S
Input resistor	Current mode：<=250Ω

	Voltage mode：>4MΩ
AO channel	2
Rated supply current	DC 24V ≥100mA
Signal type	4-20mA、1-5V、0-20mA、0-10V
Resolution	12
Output accuracy	0.3% F.S.
Change speed（each channel）	15times /S
Extra loading	Current mode： Max. 500Ω
	Voltage mode： Min. 10KΩ
Dimension and weight	
Dimension(length×width× height)	100×84.5×25.4mm
Net weight	200g

Chapter 4 Software Introduction

4.1 Overview

Based on K5, KS use same Kincobuilder software and instructions. Users can reference K5/K2 manual for most functions. The main difference is the new functions.

4.2 Use Modbus TCP protocol to communicate with third-party devices

The memory areas accessible to the Modbus TCP master are classified as follows:

Type	Modbus function code	Corresponding PLC memory area
DO（Digital output , 0XXXX）	01，05，15	Q area，M area
DI（Digital input , 1XXXX）	02	I area，M area
AO（Analog output ,4XXXX）	03，06，16	AQ area，V area
AI （Analog input , 3XXXX）	04	AI area，V area
Error record(16-bit unsigned integer)	03,04	PLC Error record area

The maximum resistor number which one instruction can visit:

1. Read Bit: read 1600 bits(200 bytes) once at most.(function code 01,02)

2. Write Bit: write 800 bits once at most.(function code 15)

3. Read Word: read 100 words once at most.(function code 03,04)

4. Write Word: write 100 words once at most.(function code 16)

5. If the memory range is smaller than the above maximum value, user can not only read or write the whole memory, but cannot read or write the maximum register number, for example, user cannot read 90 words in AI (analog input) area, because there are only 32 words in this area.

4.2.1 Modbus Register Number

Due to the different memory areas of CPUs of various specifications, the allowed access range is also limited. For other specifications, Appendix A in the help topic in the programming software KincoBuilder uses Modbus RTU protocol for communication.

In some equipment, Modbus RTU registers begin with 1, so 1 should be added to each data in this column

➢ KS101M-04DX

Area	Range	Type	Corresponding Modbus Registers*
I	I0.0 --- I0.3	DI	0 --- 3
M	M0.0 --- M4095.7	DI/DO	320 -- 33087
V	VW0 --- VW16382	AI/AO	100 -- 8291

In addition to the above memory area supporting MODBUS access, the K series PLC also provides an error recording memory area for users to read and view through MODBUS. For details, see Appendix D in the help topic in the programming software KincoBuilder error diagnosis function

4.3 High speed counter

KS provides 4 high speed counters HSC0-HSC3.All can support up to 200KHz

High speed counter supports multiple modes: single phase, CW/CCW,AB phase (1 multiplication and 4 multiplication).

All high speed counter can support maximum 32 PV and support 32 "CV=PV" interrupts. PV can be set as relative value or absolute value. If it is relative value,

4.3.1 Operation Modes and Inputs of the High-speed Counters

Input signals of high-speed counter include: clock (input impulse), direction, start and reset.

In different operation modes input signals is different. Please see below:

HSC 0				
Mode	Description	I0.1	I0.0	I0.5
0	Single-phase up/down counter with internal direction control: SM37.3	Clock		
1			Reset	
2			Reset	Start
3	Single-phase up/down counter with external direction control	Clock		Direction
4			Reset	Direction
6	Two-phase counter with up/down clock inputs	Clock Down	Clock Up	
9	A/B phase quadrature counter	Clock A	Clock B	

HSC1					
Mode	Description	I0.4	I0.6	I0.3	I0.2
0	Single-phase up/down counter with internal direction control: SM47.3			Clock	
1		Reset			
2		Reset	Start		
3	Single-phase up/down counter with external direction control			Clock	Direction
4		Reset			Direction
6	Two-phase counter			Clock	Clock Up

7	with up/down clock inputs	Reset		Down	
9	A/B phase quadrature counter			Clock A	Clock B
10		Reset			

HSC 2			
Mode	Description	I0.4	I0.5
0	Single-phase up/down counter with internal direction control:SM57.3		Clock
9	A/B phase quadrature counter	Clock A	Clock B

HSC 3			
Mode	Description	I0.6	I0.7
0	Single-phase up/down counter with internal direction control:SM127.3		Clock
9	A/B phase quadrature counter	Clock A	Clock B

4.3.2 Control Byte and Status Byte

➢ **Control Byte**

In SM area,each high-speed counter is assigned control byte to save its configuration data: one control word (8 bit), current value and pre-set (double-integer with 32 bit). Initial value of current assigned value. If the current value is written in the high-speed counter, it will start counting from that value. Please see below:

HSC0	HSC1	HSC2	HSC3	Description
SM37.0	SM47.0	SM57.0	SM127.0	Effective electrical level of reset signal : 0=high；1=low
SM37.1	SM47.1	SM57.1	SM127.1	Effective electrical level to start signal：0=high；1=low
SM37.2	SM47.2	SM57.2	SM127.2	Orthogonal counter rate：0=1x rate；1=4x rate[*]
SM37.3	SM47.3	SM57.3	SM127.3	Counting direction:0=Decrease；1=Increase
SM37.4	SM47.4	SM57.4	SM127.4	Write counting direction in HSC? 0= NO；1= Yes

SM37.5	SM47.5	SM57.5	SM127.5	Write new pre-set value in HSC? 0= NO; 1= Yes
SM37.6	SM47.6	SM57.6	SM127.6	Write new current value in HSC? 0= NO; 1= Yes
SM37.7	SM47.7	SM57.7	SM127.7	Allow this high-speed counter? 0=NO; 1= YES
HSC0	**HSC1**	**HSC2**	**HSC3**	Description
SMD38	SMD48	SMD58	SMD128	Current value
SMD42	SMD52	SMD62	SMD132	Pre-set value

HSC0	**HSC1**	**HSC2**	**HSC3**	**Description**
SM141.0	SM151.0	SM161.0	SM171.0	Use multiple preset value:0=No. 1=Yes.
SM141.1	SM151.1	SM161.1	SM171.1	Preset value type:0=Absolute value. 1=Relative value.
SM141.2	SM151.2	SM161.2	SM171.2	Preset value comparison interrupt ("CV=PV") cyclic execution. 0=No. 1=Yes. Note:Only valid when preset value is relative value.
SM141.3	SM151.3	SM161.3	SM171.3	Reserved
SM141.4	SM151.4	SM161.4	SM171.4	Update multiple PV segment and preset value:0=No. 1=Yes
SM141.5	SM151.5	SM161.5	SM171.5	Reset interrupt variable:0=Yes. 1=No.
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	Starting value of preset value table （It is offset corresponding to VB0),it must be odd value.

It needs to pay attention that not all the control bits of the control byte is suitable for all operation mode. For example, "Counting direction" and "Write counting direction in HSC" can be only used in mode 0,1 and 2 (Single-phase up/down counter with internal direction control),if the operation mode is with external direction control,

then these two bits will be ignored.

The control byte, current value and preset value are 0 by default after power on.

➢ **Status Byte**

In SM area, each high-speed counter has a status byte, which indicates 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	Fault in multiple PV value table: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 = Down; 1= Up
SM36.6	SM46.6	SM56.6	SM126.6	Current value equal to preset value: 0 = No,1 = Yes
SM36.7	SM46.7	SM56.7	SM126.7	Current value greater than preset value: 0 = No,1 = Yes
HSC0	HSC1	HSC2	HSC3	Description
SMB140	SMB150	SMB160	SMB170	Current PV segment No.(Start from 0)

4.3.3 Preset value (PV value) setting

KS supports up to 32 PV value for each high speed counter, and supports setting PV value as relative value or absolute value. It supports "CV=PV" interrupt cyclic execution.

Follows take HSC0 as example to describe PV value function and setting.

➢ **How to select "multiple PV" mode**

In the control byte of each high speed counter, there is one control bit for enable multiple preset value.

In HSC0, this control bit is SM141.0.

If SM141.0 is 0,it will use single PV value, same as K5 PLC.SMD42 is for new PV

value,SM37.5 is to update this new PV value.

If SM141.0 is 1,it will use multiple PV values. In this situation,SM37.5 and SMD42 is invalid. All the PV values will be in the PV table(SMW142 is for starting address of the table),SM141.4 defines whether it use the data in PV table or not.If SM141.4 is 1,it means when HSC starts, it will get the data from PV table. If SM141.4 is 0,when HSC starts,it will ignore the data in PV table and get the data from last preset value.

➢ **Multiple PV table**

If using PV table,all the PV value will get from PV table.

Each HSC provides one control word which is used to set the starting address of PV table.If using multiple PV,then all PV value will get from PV table.The starting address of PV table is odd address of V area,such as 301(Means VB301).

The format of PV table is as follows.

OFFSET [1]	Data type	Description
0	BYTE	Quantity of PV
1	DINT	First PV
5	DINT	Second PV
…	DINT	…

（**1**）　All the offset value are the offset bytes related to the table.

（**2**）　When it is set as relative value,then the absolute value of PV data must be greater than 1,or PLC will consider the segment of multiple PV finish and count the number of PV according to this(Higher priority than setting quantity of PV).

When it is set as absolute value,the difference between two adjacent PV's absolute value must be greater than 1,or PLC will consider the segment of multiple PV finish and count the number of PV according to this(Higher priority than setting quantity of PV).

（**3**）　"CV=PV" interrupts must execute in sequence,it means that after the counter reaches the first PV and executes interrupt,then it will compare with the second PV and so forth.

（4） PV must be set reasonably.Here takes relative value as example,if it is positive counting,PV must be greater than 0,otherwise the "CV=PV"interrupt will never execute.If it is negative counting,PV must be less than 0,otherwise the "CV=PV"interrupt will also never execute.

➢ **Relative value and absolute value**

In the control byte of each high speed counter, there is one control bit which is used to set PV as relative value or absolute value.

For HSC0,the control bit is SM141.1.

If SM141.1 is 0,it means PV is absolute value. When counting value is equal to PV,it will execute "CV=PV" interrupt. For example,if it sets 3 PV values,such as 1000,2000 and 3000,then when counting value reaches 1000,it will execute the first "CV=PV"interrupt. When the counting value reaches 2000,it will execute the second "CV=PV" interrupt and so forth.

If SM141.1 is 1,it means PV is relative value.If counter takes current counting value as reference,when the value it continues to count is equal to PV,it will execute "CV=PV" interrupt.For example,if it sets 3 PV values,such as 10,1000 and 1000,and the current counting value is 100 before HSC starts,then when the counting value reaches 110,1110 and 2110,it will execute corresponding"CV=PV" interrupt.

➢ **"CV=PV"interrupt cyclic execution**

"CV=PV"interrupt cyclic execution is only valid when PV is set as relative value.

If SM141.0 is 0,it means "CV=PV" interrupt only executes once.When all interrupts finish execution,then it will stop.If it needs to execute again, it must modify the related registers and execute HSC instruction again.

If SM141.0 is 1,it means "CV=PV" interrupt is cyclic execution.When the last PV interrupt finishes execution,PLC will take the current counting value as reference to calculate new value for PV interrupt,then it will start to compare the counting value and execute "CV=PV" interrupt and so forth.This process will execute cyclically.

For example,it sets 3 PV values,such as 10,1000 and 1000.And the current counting

value is 100 before HSC starts,then the value for every interrupt is as following table.

Current counting value	Interrupt times	First value	Second value	Third value
100	1st time	110	1110	2110
2110	2nd time	2120	3120	4120
4120	3rd time	4130	5130	6130
…	N time	…	…	…

4.3.4 "CV=PV" Envent No.

When it uses single PV mode, the HSC will be fully compatible with K5 (Include "CP=PV" event No.).

When it uses multiple PV mode, the HSC will assign a new event No. for 32 PV, as shown in following table.

High speed counter	Interrupt No.	Description
HSC0	64	"CV=PV"interrupt of 1st PV
	65	"CV=PV"interrupt of 2nd PV
	…	…（Plus 1）
	95	"CV=PV"interrupt of 32nd PV
HSC1	96	"CV=PV"interrupt of 1st PV
	97	"CV=PV"interrupt of 2nd PV
	…	…（Plus 1）
	127	"CV=PV"interrupt of 32nd PV
HSC2	128	"CV=PV"interrupt of 1st PV
	129	"CV=PV"interrupt of 2nd PV
	…	…（Plus 1）
	159	"CV=PV"interrupt of 32nd PV
HSC3	160	"CV=PV"interrupt of 1st PV
	161	"CV=PV"interrupt of 2nd PV
	…	…（Plus 1）
	191	"CV=PV"interrupt of 32nd PV

4.3.5 How to use high speed counter

➢ **Method 1:Use instructions for programming**

1)Configure the control byte of HSC and define the current value (i.e. starting value) and the preset value.

2)Use HDEF instruction to define the counter and its operation mode.

3)(Optional) Use ATCH instruction to define the interrupt routines.

4)Use HSC instruction to start the high-speed counter.

➢ **Method 2:Use wizard of HSC**

In KS PLC, it provides configuration wizard for high speed counter. Users can use the wizard to configure all high speed counters and don't need to program. The wizard is as following figure:

After using wizard to configure HSC, user also can use "Method 1" to modify the parameters of HSC.
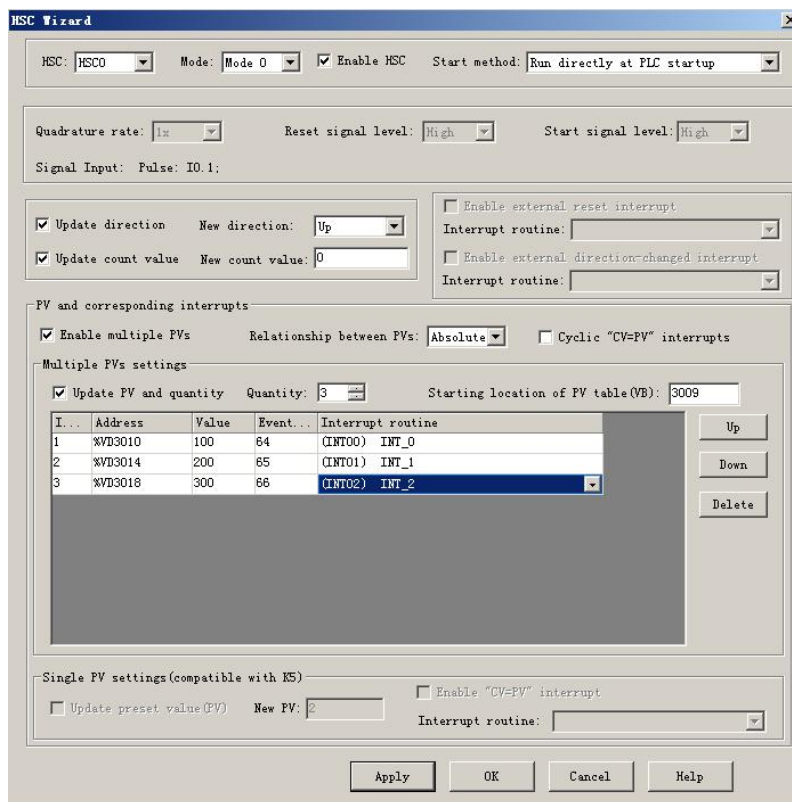
How to use HSC wizard:

1）Select the counter in【HSC】.

2）Check【Enable HSC】, and then continue following configuration.

3）Select counter mode in【Mode】.

4）Select the starting mode in【Start method】.

There are two starting method:

"Using HSC instruction": If selecting this method, then it needs to execute HSC instruction to start the HSC. Before executing HSC instruction, it doesn't need to configure the registers and execute HDEF instruction.

"Run directly at PLC startup": If selecting this method, then the HSC will start automatically after PLC power on without executing any instructions.

5）If user needs to use multiple PV mode, then check 【Enable multiple PVs】 and continue to configure all PV values and related 'Value' and 'Interrupt subroutine'. If checking

【Update PV and quantity】, then it can adjust the value in 【Quantity】to modify the number of PV.

6）If user needs to use single PV mode, then check 【Update preset value(PV)】 in 'Single PV settings' and modify the PV value and related interrupt subroutine.

7）For other options, please refer to the descriptions to HSC.

4.4 How to use high speed pulse output

Kinco-KS provides 4 channels for high speed pulse output, they are Q0.0,Q0.1 and Q0.4,Q0.5.All support PT0 and PWM output.

. Q0.0 and Q0.1,0.4 support maximum 200KHz, and Q0.5supports maximum 10KHz.

KS have one direction output channel for every high speed output. KS provide 1 direction enable control in SM area.

	Q0.0	Q0.1	Q0.4	Q0.5
Direction output channel	Q0.2	Q0.3	Q0.6	Q0.7
Direction enable Control	SM201.3	SM231.3	SM251.3	SM221.3

Direction output channel output motor direction control signal, corotation output 0,inversion output 1.

Direction enable control can forbid or allow direction output channel. It is highest primary.

If it is forbidden, it won't output direction control signal. The channel will work as common DO.

4.4.1 High speed pulse output instruction

KS provides 3types of instructions for high speed pulse output.

1）PLS: it is used to output PTO(Single segment or multiple segments) and PWM.

2）Position control: There are 5 instructions, include PREL(Relative positioning), PABS(Absolute positioning) ,PHOME(Homing), PJOG(Jogging) and PSTOP(Emergency stop). User can use these instructions to achieve positioning control easily .**Note: When**

using position control instructions, the frequency of output pulse must be not less than 80Hz.

3）Following instruction PFLO_F: There are parameters such as input frequency(F),electronic gear ratio($NUME$、$DENOM$), pulse number($COUNT$) and so on, these parameters can be used as variable. The frequency of pulse output is equal to F multiple by electronic gear ratio. When the pulse number reaches the value $COUNT$, then it will stop output and set $DONE$ bit. **Note: When using following instruction, the frequency of output pulse must be not less than 30Hz.**

4.4.2 How to use PLS instruction

PLS instruction can implement PTO and PWM output function.

- PTO：Pulse Train Output.
- PWM：Pulse-Width Modulation.
- ➢ Descriptions

	Name	**Usage**	**Group**	**Suitable for**
LD	PLS	PLS EN ENO Q		K2 K5 K6 KS
IL	PLS	PLS Q	U	KW

Opera nds	**Input/O utput**	**Data Type**	**Description**
Q	Input	INT	Constant（0、1 or 2）

The PLS instruction is used to load the corresponding configurations of the PTO/PWM from the specified SM registers and then start outputting pulse until it finish outputting pulse. The pulse output channel is specified by parameter Q, 0 means Q0.0,1 means Q0.1,2 means Q0.4,3 means Q0.5.

Note: In user program, it only needs to execute PLS instruction once when it is required.

50

It is suggested to use edge instruction to execute PLS instruction. If executing PLS executing all the time, then it can't output normally.

- **LD**

    If EN is 1,then PLS is executed.

- **IL**

    If CR is 1,then PLS is executed. It won't influence the value of CR.

4.4.2.1 High-speed Pulse Output Function

The Kinco-KS provides 4 PTO/PWM pulse generators that can be used to output PTO/PWM. Thereof, one generator is assigned to Q0.0, called PWM0 or PTO0; the second one is assigned to Q0.1, called PWM1 or PTO1,and the third one is assigned to Q0.4,called PWM2 or PTO2. The forth one is assigned to Q0.5,called PWM3 or PTO3.

The PTO/PWM pulse generators and the DO mapping area share the memory address Q0.0 ,Q0.1 and Q0.4,Q0.5. When the user program executes the high speed pulse output instructions, then the PTO/PWM generator controls the output and prohibits the normal use of this output channel.

⚠ Notice: Make sure not to use the PTO and PWM functions if Q0.0 ,Q0.1 and Q0.4 , Q0.5 are relay-output!

➢ **PWM**

PWM provides a continuous pulse output with a variable duty cycle, and you can control the cycle time and the pulse width.

The unit of cycle time and pulse width time is microsecond(us) or millisecond(ms). The maximum value of cycle time is 65535. If the pulse width time is greater than the cycle time value, the duty cycle is set to be 100% automatically and the output is on continuously. If the pulse width time is 0, the duty cycle is set to be 0% and the output is off.

➢ **PTO**

PTO provides a square wave (50% duty cycle) output, and you can control the cycle time and the number of the output pulses. The unit of cycle time is microsecond(us) or millisecond(ms).The maximum value of cycle time is 65535.The range of pulse number is 2~4,294,967,295.If the specified pulse number is less than 2, then Kinco-KS will set related error bit and prohibit the output.

PTO function provides single segment of pulse and multiple segment of pulse.

● **Single segment pulse**

In single segment pulse mode, it only executes pulse train output once after executing PLS instruction.

● **Multiple segment pulse**

In multi-segment pulse mode, CPU automatically reads the configurations of each PTO segment from a profile table located in V area and executes the related PTO segment.

The length of each segment is 8 bytes, including a cycle time value (16-bit, WORD), a reserved value (It is not used now,16-bit, INT), and a pulse number value (32-bit, DWORD).Thereof, all the pulse output frequency are the same in same segment. It uses PLS instruction to start multiple segment pulse.

In this mode, the starting address of the profile table is stored in SMW168 (corresponding to PTO0) ,SMW178 (corresponding to PTO1) , SMW218(corresponding to PTO2) and SMW248(corresponding to PTO3) .Time base is configured by SM67.3 (corresponding to PTO0) ,SM77.3 (corresponding to PTO1) , SM97.3 (corresponding to PTO2) and SM107.3(corresponding to PTO3).　The time base can be in either microsecond or millisecond. All cycle values in the profile table must use same time base, and cannot be modified when the profile is executing.

The following table describes the format of the profile table.

Byte offset[1]	Length	Segment	Description
0	8-bit		The number of segments (1 to 64)
1	16-bit	1	Initial cycle time (2 to 65535 times of the time base)
3	16-bit		Reserved
5	32-bit		Pulse number(1 to 4,294,967,295)
9	16-bit	2	Initial cycle time (2 to 65535 times of the time base)
11	16-bit		Reserved
13	32-bit		Pulse number(1 to 4,294,967,295)
…		…	…

[1] All the offsets in this column are relative to the starting position of the profile table.

Notice: the starting position of the profile table must be an odd address in V area, e.g. VB3001.

4.4.2.2 PTO/PWM Register

Each PTO/PWM generator is provided with some registers in SM area to store its configurations, as shown in following table.

Q0.0	Q0.1	Q0.4	Q0.5	Description	
SM67.0	SM77.0	SM97.0	SM107.0	PTO/PWM	Whether to update the cycle time: 0 = No; 1 = Yes
SM67.1	SM77.1	SM97.1	SM107.1	PWM	Whether to update pulse width time: : 0=No；1=Yes
SM67.2	SM77.2	SM97.2	SM107.2	PTO	Wheter to update the pulse number: : 0=No；1=Yes
SM67.3	SM77.3	SM97.3	SM107.3	PTO/PWM	Time base: 0=1μs；1=1ms
SM67.4	SM77.4	SM97.4	SM107.4	PWM	Update method: 0 = asynchronous update; 1 = synchronous update
SM67.5	SM77.5	SM97.5	SM107.5	PTO	Operation mode:

				0 = single segment; 1 = multiple segment
SM67.6	SM77.6	SM97.6	SM107.6	Function selection: 0= PTO；1=PWM
SM67.7	SM77.7	SM97.7	SM107.7	PTO/PWM Enable/disable: 0=disable；1= enable
Q0.0	**Q0.1**	**Q0.4**		**Description**
SMW68	SMW78	SMW98	SMW108	PTO/PWM Cycle time , Range:2~65535
SMW70	SMW80	SMW100	SMW110	PWM Pulse width, Range: 0~65535
SMD72	SMD82	SMD102	SMD112	PTO Pulse number, Range:1~4,294,967,295
SMW168	SMW178	SMW218	SMW248	The starting location of the profile table (byte offset from V0)For multi-segment PTO operation only

All the default value for control byte, cycle time and pulse number are 0.The way to modify configuration of PTO/PWM is that configure related control registers first, if it is PTO multiple segment pulse, it also needs to configure profile table, and then execute PLS instruction.

Each PTO/PWM generator also provides a status bytes in SM area, user can get the status information of PTO/PWM generator from the status bytes, as shown in following table.

Q0.0	Q0.1	Q0.4	Q0.5	Description
SM66.0	SM76.0	SM96.0	SM106.0	Reserved
SM66.1	SM76.1	SM96.1	SM106.1	Reserved
SM66.2	SM76.2	SM96.2	SM106.2	Reserved
SM66.3	SM76.3	SM96.3	SM106.3	PWM idle: 0=No, 1=Yes
SM66.4	SM76.4	SM96.4	SM106.4	Whether the cycle time or pulse number of PTO is wrong: 0=No, 1=Yes **Note: Cycle time and pulse number must be greater than 1.**
SM66.5	SM76.5	SM96.5	SM106.5	PTO profile terminated due to user command: 0=No, 1=Yes
SM66.6	SM76.6	SM96.6	SM106.6	Reserved
SM66.7	SM76.7	SM96.7	SM106.7	PTO idle: 0=No, 1=Yes

The PTO idle bit or PWM idle bit indicate the completion of the PTO or PWM output.

4.4.2.3 PTO Operations

The fallowing takes PTO0 as an example to introduce how to configure and operate the PTO/PWM generator in the user program.

There are two procedures for using PTO: Configure related control registers and initialize PTO. Execute PLS instruction.

Use SM0.1 (the first scan memory bit) to call a subroutine that contains the initialization instructions. Since SM0.1 is used, the subroutine shall be executed only once, and this reduces CPU scan time and provides a better program structure.

➢ **Execute the PTO (Single-Segment Operation)**

1) Set control byte SMB67 according to the desired operation.

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

   - Enable the PTO/PWM function

   - Select PTO operation

   - Select 1μs as the time base

   - Allow updating the pulse number and cycling time.

2) Set SMW68 according to desired cycle time.

3) Set SMD72 according to desired pulse number.

4) (Optional) use ATCH to attach the PTO0-complete event (event 28) to an interrupt routine to respond in real time to a PTO0-complete event.

5) Execute the *PLS* instruction to configure PTO0 and start it.

➢ **Changing the PTO Cycle Time (Single-Segment Operation)**

   Follow these steps to change the PTO cycle time.

1) Set control byte SMB67 according to the desired operation.

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

   - Enable the PTO/PWM function

- Select PTO operation

- Select 1μs as the time base

- Allow updating the cycle time value.

2) Set SMW68 according to desired cycle time.

3) Execute the *PLS* instruction to configure PTO0 and start it, then a new PTO with the updated cycle time shall be generated.

➤ **Changing the PTO Pulse Number(Single-Segment Operation)**

Follow these steps to change the PTO pulse count:

1) Set control byte SMB67 according to the desired operation.

For example, SMB67 = B#16#84 indicates:

- Enable the PTO/PWM function

- Select PTO operation

- Select 1μs as the time base

- Allow updating the pulse number

2) Set SMD72 according to desired pulse number.

3) Execute the *PLS* instruction to configure PTO0 and start it, then a new PTO with the updated pulse number shall be generated.

➤ **Execute the PTO (Multiple-Segment Operation)**

1) Set control byte SMB67 according to the desired operation.

For example, SMB67 = B#16#A0 indicates:

- Enable the PTO/PWM function

- Select PTO operation

- Select multi-segment operation

- Select 1μs as the time base

2) Set an odd number as the starting position of the profile table into SMW168.

3)   Use V area to configure the profile table.

4)   (Optional) Use ATCH to attach the PTO0-complete event (event 28) to an interrupt

routine to respond in real time to a PTO0-complete event.

5)   Execute the *PLS* instruction to configure PTO0 and start it.

4.4.2.4 PWM Operations

Following takes PWM0 as an example to introduce how to configure and operate the

PTO/PWM generator in the user program.

There are two procedures for using PWM: Configure related control registers and

initialize PTO. Execute PLS instruction.

Use SM0.1 (the first scan memory bit) to call a subroutine that contains the initialization

instructions. Since SM0.1 is used, the subroutine shall be executed only once, and this

reduces CPU scan time and provides a better program structure.

➢   **Execute PWM**

1)   Set control byte SMB67 according to the desired operation.

For example, SMB67 = B#16#D3 indicates:

- Enable the PTO/PWM function

- Select PWM operation

- Select 1μs as the time base

- Allow updating the pulse width value and cycle time value

2)   Set SMW68 according to desired cycle time.

3)   Set SMW70 according to desired pulse width.

4)   Execute the *PLS* instruction to configure PWM0 and start it.

➢   **Changing the Pulse Width for the PWM Output**

The following steps describes how to change PWM output pulse width.

1)   Set control byte SMB67 according to the desired operation.

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

- Enable the PTO/PWM function

- Select PWM operation

- Select 1μs as the time base

- Allow updating the pulse width value and cycle time value

2) Set SMW70 according to desired pulse width.

3) Execute the *PLS* instruction to configure PWM0 and start it.

4.4.3 How to Use Position Control Instructions

4.4.3.1 How to Modify the Current Value of Position Control Instructions

➢ **Control Registers and Status Registers**

For the Position Control instructions,KS1 specifies a control byte for each high-speed output channel to store its configurations. Besides, it assigns a current value register(DINT) to store the pulse number which has outputted currently (This value will increase when run forward and decrease when run reverse).The following table describes the control byte and the current value.

Q0.0	Q0.1	Q0.4	Q0.5	**Description**
SMD212	SMD242	SMD262	SMD226	Read only. Current value (Increase when run forward, decrease when run reverse).It indicates the pulse number which has already outputted.
SMD208	SMD238	SDM258	SDM222	Read/Write. New current value. Use to modify the current value together with specific control bit.
Q0.0	Q0.1	Q0.4	Q0.4	Description
SM201.7	SM231.7	SM251.7	SM221.7	Read/Write. Emergency-Stop bit. If this bit is 1, no position control instructions can be executed. When executing the PSTOP instruction, this bit is

				set to 1 automatically, and it must be reset in the program..
SM201.6	SM231.6	SM251.6	SM221.6	Read/Write. Reset the current value or not 1 --- Clear the current value. 0 --- Maintain the current value..
SM201.5	SM231.5	SM251.5	SM221.5	Reserved
SM201.4	SM231.4	SM251.4	SM221.4	Read/Write. Use to modify current value. 1 - Modify current value. 0 - Maintain the current value.
SM201.3	SM231.3	SM251.3	SM221.3	Read/Write. Direction control bit. 1 --- Disable the direction output channel, it will be used as normal output. 0 --- Enable the direction output channel.
SM201.0 ~ SM201.2	SM231.0 ~ SM231.2	SM251.0 ~ SM251.2	SM221.0 ~ SM221.2	Reserved

➢ **How to modify current value**

Each high speed output channel has one register for current value, they are SMD212,SMD242 and SMD262,SMD226.The outputted pulse number are stored in these registers. Current value registers are read only, if user needs to modify the current value, it can use following methods.

• **Method 1**

User reset bit to clear current value.

The reset bits for 4 output channels are SM201.6、SM231.6 、SM251.6 and SM221.6.

When the reset bit is 1, PLC will set the current value as 0.Therefore, t only needs one scan time for the reset bit to activate. When it needs to use this bit, try to avoid to keep this bit always 1 and also and also avoid to set this bit while the Position Control instruction (Include PHOME, PREL, PABS, JOG and PFLO_F) is executing, otherwise the counting value may be wrong.

Following takes channel 0 as example to describe how to reset current value.

(* Network 0 *)

(*Based on homing signal, when it moves to homing, it requires to clear current value*)

LD   %SM0.0

PHOME

0, %M0.0, %M0.1, %M0.2, %VW0, %VW2, %VW4, %VD6, %VW10, %M0.4, %M0.5, %MB1

(* Network 1 *)

(*After PHOME finishing, it uses finishing bit "DONE" to clear current value*)

LD   %M0.4

R_TRIG

ST   %SM201.6

- **Method 2**

Modify current value by using following registers.

Q0.0	Q0.1	Q0.4	Q0.5	Description
SMD208	SMD238	SDM258	SDM222	Read/Write. New current value. Use to modify the current value together with specific control bit.
SM201.4	SM231.4	SM251.4	SM221.4	Read/Write. Use to modify current value. 1 - Modify current value. 0 - Maintain the current value.

Here takes channel 0 as example to describe the method: If SM201.4 is 0,then it will maintain the current value SMD212. If SM201.4 is 1, then it will move the value of SMD208 to SMD212.When it needs to use this bit, avoid to keep this bit always 1 and also avoid to set this bit while the Position Control instruction (Include PHOME, PREL, PABS, JOG and PFLO_F) is executing, otherwise the counting value may be wrong.

Following takes channel 0 as example to describe how to modify current value:

(* Network 0 *)

(*Based on homing signal, hen it moves to homing, t requires to set current value as 100.*)

LD   %SM0.0

PHOME

0, %M0.0, %M0.1, %M0.2, %VW0, %VW2, %VW4, %VD6, %VW10, %M0.4, %M0.5, %MB1

(* Network 1 *)

(*When PHOME instruction finishing, it uses finishing bit DONE to modify current value.*)

LD   %M0.4

R_TRIG

MOVE   DI#100, %SMD208

ST   %SM201.4

4.4.3.2 Can it change maximum output frequency when position control instruction is executing?

PREL（Relative position）and PABS（Absolute position） will not change maximum output frequency when it is executing. It will read the parameters minimum frequency, maximum frequency and acceleration/deceleration time parameters when it starts, and calculates suitable acceleration/deceleration segments according to the value of these parameters, then it will start outputting pulse. During pulse outputting, PREL and PABS will not read the parameters above again, therefore, changing these parameters will not affect the pulse output.

PJOG （Jogging） will read pulse input frequency(MAXF) all the time when it is executing, and adjust the pulse output frequency according to new setting frequency.

PHOME（Homing） will read the maximum frequency (MAXF) all the time when it is running at maximum frequency but hasn't found homing signal, and calculate acceleration or deceleration segment automatically according the new setting frequency, then it will accelerate or decelerate to new frequency to output pulse.

4.5 Use of CAN bus

Model	CAN communication port		Extended protocol	Free communication	CANOpen master station	Kinco Motion Control Protocol	CANOpen slave station
KS105C1-16DT	1*CAN port (named CAN)	CAN	not support	Support Note: You can only choose one of them.			
KS105C2-16DT KS101M-O4DX	2*CAN port （Named CAN1, CAN2）	CAN1	Support (up to 14 expansion modules)	support	Not support		
			Note: When the CAN1 interface is connected to the expansion module, the CAN communication function is invalid, that is, the CAN1 interface extension protocol or CAN communication function can only be selected one.				
		CAN2	not support	support Note: you can only choose one of them.			Not support

4.5.1 Hardware wiring

The KS105C1-16DT CPU module provides a CAN interface that is directly connected to the two RJ45s, allowing the user to use one of them. Refer to for wiring diagram.

The KS105C2-16DT /KS101M-04DX CPU module provides two CAN interfaces, which are divided into two RJ45s. Refer to for wiring diagram.

The CAN bus PLC side provides the DIP switch to select whether the terminal resistance is enabled. For details, please refer to the product silk screen marking!

Note: the interface provided by the module is an RJ45 interface. When multiple modules are networked themselves, the connection cable needs to be connected by a straight-through network cable instead of a crossover network cable.

4.5.2 Extended bus function

The CAN communication port supports the extended bus protocol. If the expansion module is configured in the [Hardware Configuration] of the user project, the CAN interface will work as an expansion bus interface. At this time, only the free communication command is allowed in the user program.

The CPU allows up to 14 expansion modules to be connected. Each expansion module allows for distributed installation. The total length of the communication cable between the CPU and the last expansion module is not allowed to exceed 30 meters. When using the expansion bus, it is recommended to add the termination resistance of the KS at the head end and the expansion module at the end to avoid signal reflection and enhance communication stability. In addition, when using long-distance distributed installation, the shielded twisted pair cable is recommended for the extended communication cable and the single-ended shield is well grounded (control ground), and the communication cable should be away from strong interference sources and various high-power lines (including the power of the equipment). Cable), frequent pulse signal lines, etc.

In the factory default setting, the CPU module automatically assigns a unique ID to each expansion module and configures various parameters when powering up. Therefore, the CPU and all expansion modules are required to be powered on at the same time or all of the expansion modules are prior to the CPU module. Power on, otherwise it may cause a program execution error. However, in order to facilitate the user's distributed application, the CPU provides the EX_ADDR instruction, and the user can modify the above default configuration by calling this instruction, thereby making the use of the expansion module more flexible.

4.5.2.1 How to use EX_ADDR instruction to extension module

For a description of the EX_ADDR instruction, see 4.5.7.4 Extend Bus Instruction .

In practical applications, if the expansion module is far away from the CPU module or installed on a different device, the default power-on sequence may not be guaranteed. In this

case, the user can use the EX_ADDR instruction to modify the factory default configuration as described below, so that each expansion module can be powered on or off at any time without causing the PLC to execute a program error.

1) In the user project, in the [Hardware Configuration], add each expansion module in the required order and configure it according to the actual requirements, and call the EX_ADDR instruction in the program (located in the [CAN command] group of the instruction set).

2) Connect the real CPU and all expansion modules in the order of [Hardware Configuration], and then power on in the default order (CPU and all expansion modules are powered on at the same time or all expansion modules are powered on before the CPU module).

3) Download the user project to the CPU. After the CPU is running normally, change the parameter value of the EX_ADDR instruction to 181 (decimal), and then let the EX_ADDR instruction execute once. After the instruction is successfully executed, each expansion module will automatically save its own ID and various parameters (such as signal form, filtering method, etc.).

4) Power off the PLC system. The user can then install the expansion module in the desired location, noting that the order of the modules (starting with the CPU) remains the same as in [Hardware Configuration]. After that, when the expansion module is powered on, it automatically reads the saved data and automatically enters the running state. It does not need to be configured by the CPU, 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 sequence, modify the parameter value of the EX_ADDR instruction in the program to 99 (decimal), and then let the EX_ADDR instruction execute once. After the instruction is successfully executed, each expansion module will clear the saved ID and channel parameters. After power-on, it will wait for the CPU to automatically assign an ID and configure parameters.

4.5.3 Kinco motion control function

The Kinco motion control function is used to control Kinco's motion control products (servo and stepper drives) with CAN interface. Based on the CANOpen protocol, it encapsulates the CANOpen communication details of the driver and provides a set of motion control instructions and corresponding network configuration tools for the user in combination with actual application requirements. This function is easy to use, and users can easily communicate with the drive and perform positioning control even if they are not familiar with the details of the CANOpen protocol.

This function can control up to 32 motion control products. In practical applications, the user can determine the actual number of connected units according to the required program space, network load rate, and the like.

This function supports parameter uploading (downloading), motor lock axis, loose shaft, homing point, jog (speed mode), absolute positioning, relative positioning, etc. for motion control products, and does not support torque mode and master-slave following mode. operating. In addition, this function can be used in principle for all third-party motion control products that support the standard CANopen protocol. **Please consult the step technician before use**.

The user uses the Kinco motion control function as follows：：

1) In the user project, enter the [Kinco Motion Control Network Configuration] wizard window to complete the network configuration.

2) Call the motion control instruction to program according to actual needs.For a description of the motion control instructions, see 4.4.7.1 Kinco Motion control instruction.

3) When the project is downloaded to the PLC, the PLC will run as the master station after starting up, manage the communication of the entire network, and execute the positioning control program.

4.5.3.1 Kinco Motion Control Network Configuration

The Kinco motion control function uses the CANOpen protocol, the PLC acts as the master station, and each driver acts as a slave. Before calling the command, the user must first configure the actual CANOpen network used. According to the habit of field application, we refer to the slave as "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, in which the network configuration is completed.
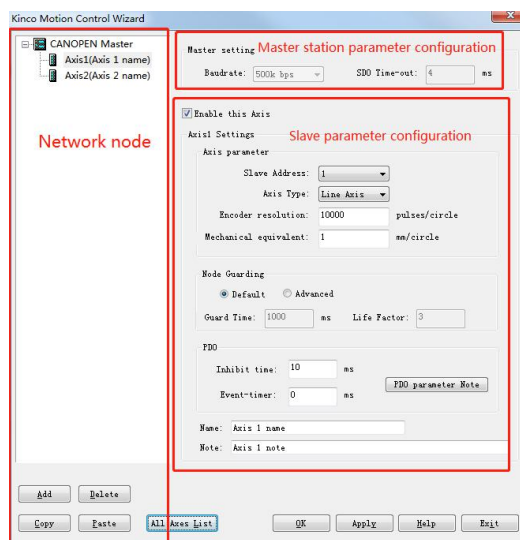
The window is divided into three parts: a tree list of network nodes, parameters of the master station, and parameters of the axis (slave).

➢ **Operation of the network node tree**

In the network node tree, the root node is [CANOpen Master], and each of the following child nodes is an axis (slave) in the network.

Four buttons of [Add], [Delete], [Copy] and [Delete] are provided below, and the software also provides corresponding shortcut keys and right-click menu functions. Users can use these features 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

4) 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.
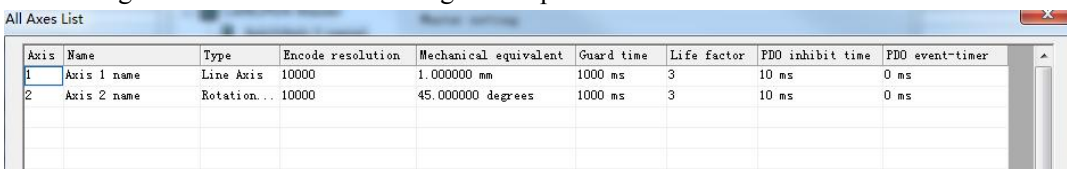
5) Axis (Slave parameter)

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.

6) **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

All Axes List

Axis	Name	Type	Encode resolution	Mechanical equivalent	Guard time	Life factor	PDO inhibit time	PDO event-timer
1	Axis 1 name	Line Axis	10000	1.000000 mm	1000 ms	3	10 ms	0 ms
2	Axis 2 name	Rotation...	10000	45.000000 degrees	1000 ms	3	10 ms	0 ms

4.5.4   CANOpen Master function

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.

4.5.4.1 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.

4.5.4.1.1 Network 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.

4.5.4.1.1.1 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

4.5.4.1.1.2 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.

4.5.4.1.2 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

4.5.4.1.3 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.

4.5.4.2 CANOpen Master Feature

The KS 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.

4.5.4.2.1   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.

4.5.4.2.2 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.

4.5.4.2.3 CANOpen   Network configuration process

1) Configure global parameters

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

Global

Baudrate   500k bps

SDO Timeout   500

Master

☑Configure slaves at startup

- [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:
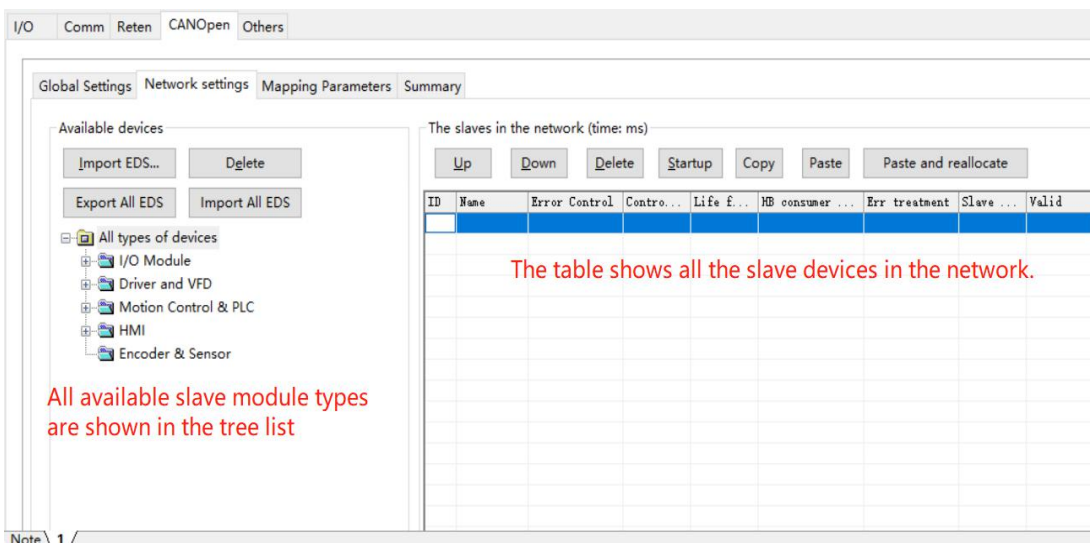
All function buttons on the page have corresponding right-click menu commands. When the user clicks the right mouse button at the relevant position, the corresponding right-click menu will pop up, and the menu command can be used at this time. The following describes the common process of configuring 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.
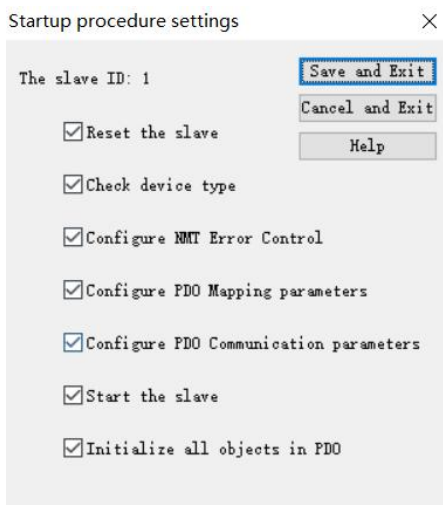
[Reset Node]: Whether the master station sends the "Reset Node" command before sending the configuration command to the slave station.

[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：
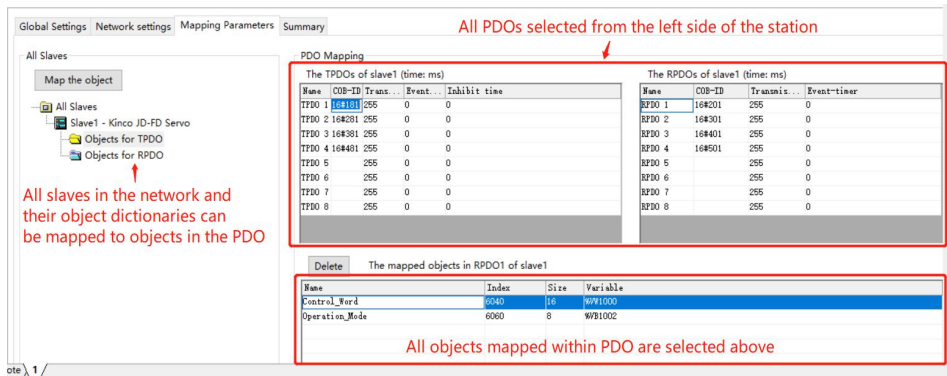
Enter the [Object Dictionary Mapping] page and configure PDO for all slaves in the network.

The left part of the page [list of added slaves] shows all the slaves that have been added to the network, as well as the objects in the slave object dictionary that can be mapped to the PDO. Among them, the objects in [Send PDO] can only be mapped to the TPDO of the slave, and the objects in the [Receive PDO] list can only be mapped to the RPDO of the slave.

Click a slave station in the [Added Slave Station List], then all PDOs of the slave station will be displayed on the right. You can configure each PDO:

- Communication parameters

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

- Mapping parameters

In the object list on the left, double-click an object to add it to the current PDO. At the same time, Kincobuilder will automatically assign a PLC's V area address to this object, such as VW1006. The user operates this V area address in the program It is equivalent to operating the corresponding object.

3) **Copy slave, paste slave**

On the [Network Configuration] page, there are 3 buttons: [Copy Slave], [Paste Slave], and [Paste Slave (Re-divide Memory)]. As shown below.

The slaves in the network (time: ms)

| | Up | Down | Delete | Startup | Copy | Paste | Paste and reallocate |

ID	Name	Error Control	Contro...	Life f...	HB consumer ...	Err treatment	Slave ...	Valid

Table shows all slaves in the network
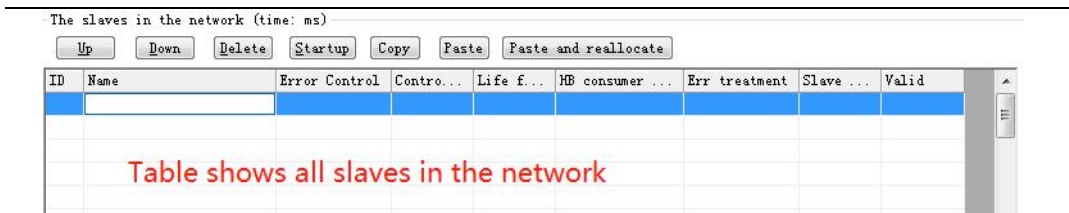
[Copy Slave]: Select a configured slave, and then click this button to copy all the information of the slave (itsIncluding all PDO communication parameters, mapping parameters, etc.). If the selected slave is not configured with any PDO, the copy fails andPrompt corresponding information.

[Paste Slave]: After copying a slave successfully, click to select an empty row in the table, and then click this button. The slave information you just copied will be pasted into the row and a new slave will be generated. Note: The PLC memory address corresponding to each mapping object in the new slave PDO is still the same as that in the source slave, there is no reallocation, and the user needs to modify it.

[Paste Slave (Re-separate 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 will be allocated automatically without user modification.

4.5.5 CANOpen slave function

4.5.5.1 Overview

KS supports CANOpen slave function and has the following characteristics:

- CAN interface adopts CAN2.0A standard.
- Compliant with DS301 V4.2.0 and DS405 V2.0 protocols
- Support NMT network management service, support heartbeat protocol.
- Support SDO Server and accelerate transmission mode.
- Supports up to 16 TPDO and 16 RPDO, PDO communication parameters and mapping parameters can be freely configured.

Note: When using, the serial number of PDO must be continuous and no interval is allowed.

For example, it is legal to use TPDO1, TPDO2, TPDO3, but it is illegal to use TPDO1, TPDO3, TPDO4.

4.5.5.2 Enable CANOpen slave function

In the user project, enter [PLC hardware configuration], and in the [Other] page of the CPU module, the user can configure the CANOpen slave function.
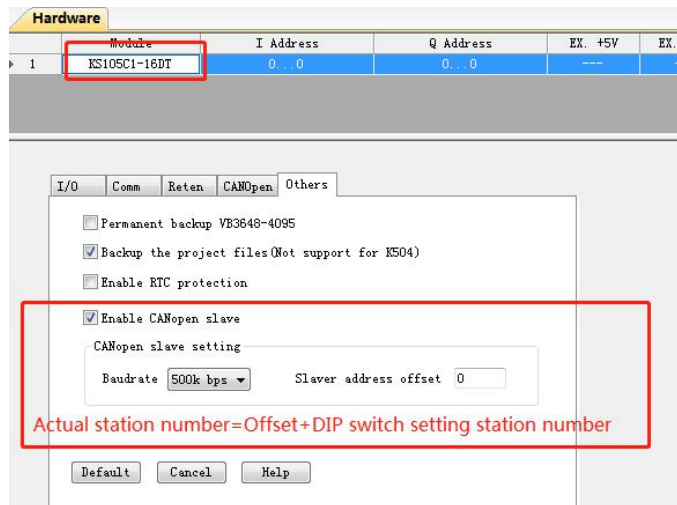
[Enable CANOPEN slave]: If this option is selected, this PLC will act as a CANOpen slave.
[Baud rate]: Select the baud rate used by the CAN interface. The baud rate of all nodes in a network must be consistent.
[Slave Address Offset]: The real station number of this slave is equal to the "combined value of the 1 to 3 digit DIP switches" plus "slave address offset". For example, if the combined value of the 1st to 3rd digit DIP switches is 5, and the slave address offset is set to 8, the slave station number is 13.

4.5.5.3 Object dictionary

Users can download the eds file of this slave device for free from the official website of Buco.The EDS file of this device has been integrated in KincoBuilder software: In the [CANOpen Master]-> [Network Configuration] page of the hardware configuration, this device

79

is displayed in the "Motion Controller and PLC" of "All Slave Module List" ", The name appears as" Kinco KPLC for Kinco. " If you use Kinco PLC as the master and slave, you do not need to import the eds file again.This device temporarily opens the following memory areas as CANOpen communication objects, which can be mapped to PDO for use. If you need to use a larger memory area, please contact Buco to provide another eds file.

Object index	type of data	PLC memory area	Quantity	Read and write attributes
0xA040	BYTE	IB0～IB9	10	Read-only
0xA0C0	INT16	AIW0～AIW18	10	Read-only
0xA4C0	BYTE	QB0～QB9	10	Read and write
0xA4C1	BYTE	MB0～MB9	10	Read and write
0xA4D2	BYTE	VB508～VB517	10	Read and write
0xA540	INT16	AQW0～AQW18	10	Read and write
0xA550	INT16	VW0～VW18	10	Read and write
0xA641	INT32	VD1016～VD1052	10	Read and write
0xA6C2	FLOAT	VR2032～VR2068	10	Read and write

4.5.6 CAN Free communication function

KS 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

4.5.7 CANBus related instruction

4.5.7.1　Kinco Motion control instruction

4.5.7.1.1 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!

1) **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.

4.5.7.1.2 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 parametersThrough 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	MC_RPARAS EN ENO EXEC DONE AXIS ERR ID ERRID NUM PARAS	• 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.

2) 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.

3) 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. *)
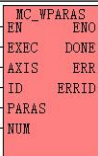LD   %SM0.0
DI_TO_I %VD1204, %VW304
(* Network 4 *)
(* The third data in the table is the third parameter value read, parameter 8. This parameter is unsigned 16-bit number, but the maximum range is 32767, so the program can be processed by INT or WORD type, but it is best to judge whether the value is 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	MC_WPARAS EN ENO EXEC DONE AXIS ERR ID ERRID PARAS NUM	• 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、	The axis number of the target axis (that is, the address

			constant	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.

4)   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 descriptionIf 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
```
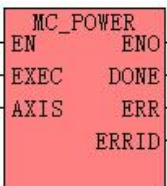
4.5.7.1.3 MC_POWER（Lock shaft and loose shaft）

	Name	Instruction format	Suitable for
LD	MC_POWER	MC_POWER EN ENO EXEC DONE AXIS ERR ERRID	• KS • KW103 • KW203
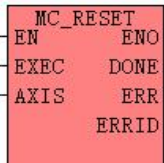
Parameter	Input/Output	Date type	Allowed memory area	Description
EXEC	Input	BOOL	M、V、L、SM	The rising edge triggers the lock axis command and the falling edge triggers the loose axis command.
AXIS	Input	INT	V、M、L、constant	The axis number of the target axis (that is, the address of the CANOpen slave)
DONE	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.

4.5.7.1.4 MC_RESET（Reset drive alarm）

	Name	Instruction format	Suitable for
LD	MC_RESET	MC_RESET EN ENO EXEC DONE AXIS ERR ERRID	• 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!
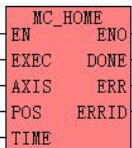
Note: This instruction only resets the alarm error information of the driver, and does not reset the output results of each instruction!

➢ 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.

4.5.7.1.5 MC_HOME（Homing）

	Name	Instruction format	Suitable for
LD	MC_HOME	MC_HOME EN ENO EXEC DONE AXIS ERR POS ERRID TIME	• 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.
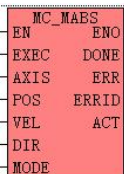
➢ LD Format Instructions

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.

4.5.7.1.6 MC_MABS（PABS）

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

param eter	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.
ERRI D	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.
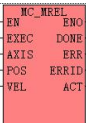
When the instruction is executed, the PLC controls the axis to start the absolute positioning of the axis according to the target position (POS) and motion speed (VEL) parameter values entered by the user. During the movement, the command will continuously scan the target position and the target speed 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.

4.5.7.1.7 MC_MREL（PREL）

	Name	Instruction format	Suitable for
LD	MC_MREL	MC_MREL EN ENO EXEC DONE AXIS ERR POS ERRID VEL ACT	• 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.

4.5.7.1.8 MC_JOG（JOG）

	Name	Instruction format	Suitable for
LD	MC_JOG	MC_JOG EN ENO EXEC DONE AXIS ERR VEL ERRID DIR ACT	• 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.

4.5.7.1.9 MC_STATE（Read the status of the drive）

	Name	Instruction format	Suitable for
LD	MC_STATE	MC_STATE EN ENO AXIS POS HOME CW CCW RUN FAULT INPUT LIMIT ERRCODE APOS AVEL ONLINE	• 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.
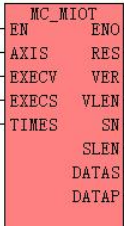
This instruction scans the driver status all the time, obtains the flags of various states and outputs them to the corresponding output parameters.

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.

4.5.7.1.10 MIOT_MC （Read Kinco servo driver information）

	Name	Instruction format	Suitable for
| LD | MC_HOME | ```
MC_MIOT
EN      ENO
AXIS    RES
EXECV   VER
EXECS   VLEN
TIMES   SN
        SLEN
        DATAS
        DATAP
``` | • 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.

4.5.7.2 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.

4.5.7.2.1 SDO_WRITE

	Name	Instruction format	Suitable for
LD	SDO_WRITE	SDO_WRITE EN ENO EXEC DONE NODE ERROR INDEX SUBINDEX DATA DATALEN	• 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.

4.5.7.2.2 SDO_READ

	Name	Instruction format	Suitable for
LD	SDO_READ	SDO_READ EN　　　　ENO EXEC　　　DATA NODE　　DATALEN INDEX　　　DONE SUBINDEX　ERROR	• 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.

4.5.7.3 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！

4.5.7.3.1 CAN_INIT（Initialize the CAN interface）

	Name	Instruction format	Suitable for
LD	CAN_INIT	CAN_INIT EN ENO CH ERR BAUD	• 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.

4.5.7.3.2 CAN_TX（Send CAN message automatically）

	Name	Instruction format	Suitable for
LD	CAN_TX	CAN_TX EN ENO CH DONE TIMER ERR ID FMT DATA LEN	• 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！**

4.5.7.3.3 CAN_WRITE（Send CAN message once）

	Name	Instruction format	Suitable for
LD	CAN_WRITE	CAN_WRITE EN　　ENO CH　　DONE ID　　ERR FMT DATA LEN	• 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！

4.5.7.3.4　CAN_RX（Receive specific ID CAN message）

	Name	Instruction format	Suitable for
LD	CAN_RX	CAN_RX EN　　ENO CH　　DONE ID　　ERR FMT　　DATA MODE　　LEN TIME	• 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！

4.5.7.3.5   CAN_READ（Receive CAN message once）

	Name	Instruction format	Suitable for

LD	CAN_READ	CAN_READ
		EN ENO
		CH DONE
		TIME ERR
		       ID
		       FMT
		       DATA
		       LEN
			• 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.

4.5.7.4 Extend Bus Instruction

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

4.5.7.4.1 EX_ADDR（Modify the extension module configuration）

	Name	Instruction format	Suitable for
LD	EX_ADDR	EX_ADDR EN　　ENO EXEC OP	• 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.