Value & Technology

Programmable Controller

NK1 series
USER MANUAL

[The third edition]

KOYO ELECTRONICS (WUXI) CO., LTD.

Preface

Thank you for purchasing NK1 series PLC. We want your new automation equipment to operate safely. Anyone who installs or uses this equipment should read this publication (and any other relevant publications) before installing or operating the equipment.

If you have any questions or need additional information when you are reading this manual or using NK1 series PLC, please contact the company headquarters or offices around in order to get the service as soon as possible.

Due to product improvements and other reasons, the content of this document will be laid down some differences with the actual product.

Our Company reserves patents of all information including this manual!

Manual name:"NK1 series PLC user's manual"(edition history)

Profile ID	Date of establishment	Description
KEW-M2511A	June 2014	Originals, the first edition
KEW-M2511B	December 2014	The second edition, the number of expansion units is up to 14; supports two extension cables; high-speed I / O function increases parameter writing confirmation register; support BCD / HEX two kinds of count values / pulse output position value table mode and multi preset value end code changing; Ethernet port supports RX / WX communication under ECOM UDP protocol.
KEW-M2511B1	June 2015	Add four binary compare instructions: BCMPR, BCMPRD, BSCMPR, BCMPRC
KEW-M2511B2	November 2015	1. Add the description of NK1L-CPU40DR 2.Add the description of IBOX instructions (KPPSoftsoftware V1.0.0.19 or later.)

Manual name:"NK1 series PLC user's manual" (edition history continuation)

Profile ID	Date of establishment	Description
KEW-M2511C2	**January 2020**	**1.NK1L-CPU40DR add a RS-485 communication port.** **2. the communication ports do not support multiprotocol auto selection** **3. RX/WX instruction does not supprt MODBUS RTU protocol** **4. Add the description of NK1H-CPU20** **5.Etherport support non-protocol communication (Firmware V20171212 or later, KPPSoftsoftware V1.5.0.5 or later)** **6.Add nineteen new instructions (Firmware V20190716 or later, KPPSoftsoftware V1.6.4.7 or later)** **7 . open the RTS ON delay time setting for RS-232C (Firmware V20190806 or later, KPPSoftsoftware V1.6.6.9 or later)**

Table of contents

Chapter 1 System components

1-1 Brief introduction

NK1 series PLC is a new series of programmable controller developed by our company, this series PLC is integral and can be extended with expansion modules. All CPUs are built-in communication ports, high-speed counter, PID regulation, pulse output function, etc. NK1 CPUs have maximum 40 I/O points, and can be extended up to 14 I/O expansion units（You can use up to 2 expansion cables in one CPU system.）, plus a function board, so the system can be extended up to 496 I/O points (40 points CPU +32 points expansion unit × 14 + 8 points function board × 1).

NK1 series CPUs are built-in high-speed counter input circuits, supporting up to three A/B two-phase or five single-phase counting inputs. The counting speed is up to 100kcps. Single-phase and two-phase counting loops can be freely combined and support external hardware reset and count prohibiting (only two-phase counter). In addition, there is a set of input points can achieve external interrupt, pulse catch function, the supported signal width is not less than 0.05ms (and the pulse frequency is less than 10kHz).

Transistor output and mixed output type NK1 PLCs support up to two channels CW/CCW (or pulse+direction) high-speed pulse outputs. The pulse output speed is up to 100 kHz. Pulse output can chieve home search, trapezoidal positioning, interrupt positioning, multi-segment trapezoidal positioning and velocity control function, and can change the velocity during positioning.

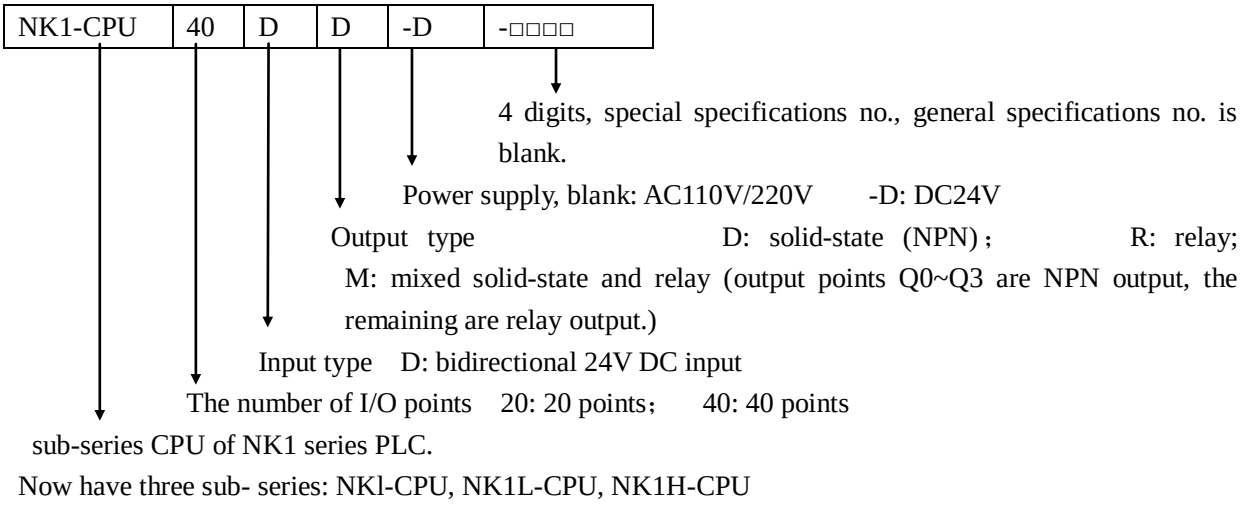
NK1 PLCs have multiple communication ports: 20-point models have two RS232C serial ports, one RS485 serial port and one USB port; 40-point models have one Ethernet port, one RS232C serial port and one RS485 serial port. In addition 40-point models can also be expanded one RS232C or one RS485 port via function board. You can configure the PLC by KPPSoft software via any of these communication ports. These communication ports support ECOM (Ethernet port), MODBUS TCP/IP (Ethernet port), Ethernet-non-sequence(Ethernet port),K-sequence (slave serial port), CCM2/DirectNET (slave serial port), MODBUS RTU (M/S serial port), non-sequence (M/S serial port) protocol and so on. In addition, you can directly download user program to NK1 PLC by using USB flash disk through the USB port instead of connetcting KPPSoft software(Need a OTG cable).

1-2 Composition of unit model

1-2-1 Composition of CPUs' unit

According to output type, NK1 series CPUs are divided into DM, DR and DD type; according to the number of I/O points, they are divided into 20-point and 40-point type. Here list the composition of CPU's unit.

Composition of CPU's unit



The following table lists all Units of the NK1 series CPUs. **Note:** Due to product improvements or other reasons, the unit listed in the table may be changed. In addition, CPU's Unit is listed in the table that does not mean it is in stock, please contact the sales department for details.

No.	Unit	specifications	remark
1	NK1-CPU20DR	12 points DC24V input and 8 points relay output, power supply is AC85V~264V without Ether port	
2	NK1-CPU20DD	12 points DC24V input and 8 points solid-state output, power supply is AC85V~264V without Ether port	
3	NK1-CPU20DM	12 points DC24V input, 4 points solid-state output and 4 points relay output, power supply is AC85V~264V without Ether port	
4	NK1-CPU40DR	24 points DC24V input and 16 points relay output, power supply is AC85V~264V with Ether port	
5	NK1-CPU40DD	24 points DC24V input, 16 points solid-state output, power supply is AC85V~264V with Ether port	
6	NK1-CPU40DM	24 points DC24V input, 4 points solid-state output and 12 points relay output, power supply is AC85V~264V With Ether port	
7	NK1-CPU20DR-D	12 points DC24V input and 8 points relay output, power supply is DC24V without Ether port	
8	NK1-CPU20DD-D	12 points DC24V input and 8 points solid-state output, power supply is DC24V without Ether port	
9	NK1-CPU20DM-D	12 points DC24V input, 4 points solid-state output and 4 points relay output, power supply is DC24V without Ethe port	
10	NK1-CPU40DR-D	24 points DC24V input and 16 points relay output, power supply is DC24V with Ether port	
11	NK1-CPU40DD-D	24 points DC24V input and 16 points solid-state output, power supply is DC24V with Ether port	
12	NK1-CPU40DM-D	24 points DC24V input, 4 points solid-state output and 12 points relay output, power supply is DC24V With Ether port	
13	NK1L-CPU40DR	24 points DC24V input and 16 points relay output, power supply is AC85V~264V without Ether port	

NOTE: the unit list of NK1H-CPU20 sub-series , please see Chapter 8.

1-2-2 Composition of NK1 expansion units

NK1 series PLC supports multiple expansion untis including a variety of normal digital I/O expansion units, analog expansion units, high-speed counter expansion units, etc. Here list their model composition.

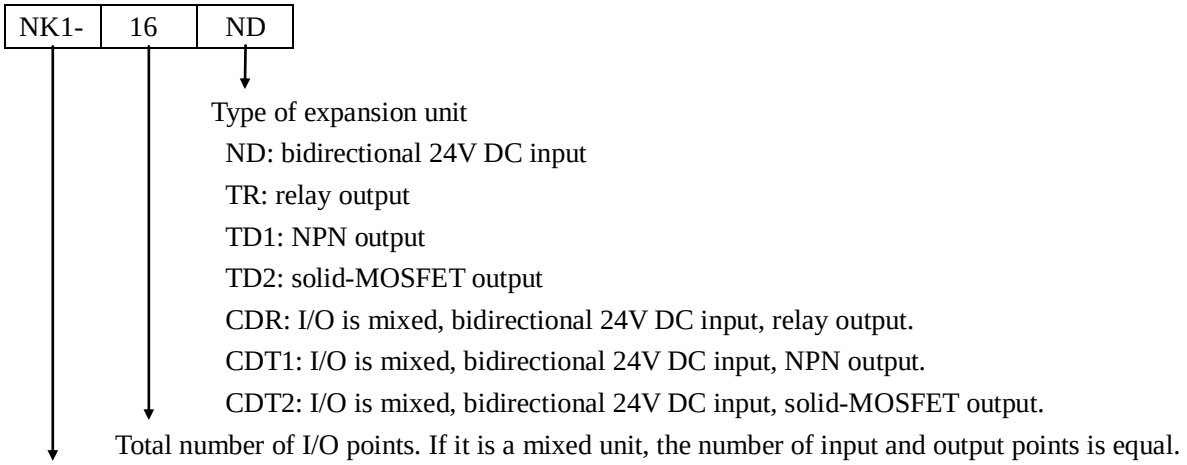
Normal digital I/O expansion units are named following the model table of digital I/O expansion units.

Special function expansion units, such as analog expansion units and high-speed counter expansion units are named with NK1- plus specialized name. For example:

NK1-8AD4DA: 8-channel input and 4-channel output analog expansion unit;

NK1-03Z: 3-channel A/B phase high-speed input expansion unit.

Composition of normal digital I/O expansion units' model



The following table lists all Models of the NK1 series expansion units. **Note:** Due to product improvements or other reasons, the models listed in the table may be changed. In addition, expansion unit's Model is listed in the table that do not mean it is in stock, please contact the sales department for details.

No.	Model	Specifications	Remark
1	NK1-08ND	8 points bidirectional DC24V input	
2	NK1-08TR	8 points relay output	
3	NK1-08TD1	8 points NPN output	
4	NK1-08TD2	8 points solid-MOSFET PNP output	
5	NK1-08CDR	4 points bidirectional DC 24V input, 4 points relay output	
6	NK1-08CDT1	4 points bidirectional DC24V input, 4 points NPN output	
7	NK1-08CDT2	4 points bidirectional DC24V input, 4 points solid-MOSFET PNP output	*1
8	NK1-16ND	16 points bidirectional DC24V input	
9	NK1-16TR	16 points relay output	
10	NK1-16TD1	16 points NPN output	
11	NK1-16TD2	16 points solid-MOSFET PNP output	
12	NK1-16CDR	8 points bidirectional DC 24V input, 8 points relay output	
13	NK1-16CDT1	8 points bidirectional DC24V input, 8 points NPN output	
14	NK1-16CDT2	8 points bidirectional DC24V input, 8 points solid-MOSFET PNP output	*1
15	NK1-32ND	32 points bidirectional DC24V input	
16	NK1-32TR	32 points relay output	
17	NK1-32TD1	32 points NPN output	
18	NK1-32TD2	32 points solid-MOSFET PNP output	*1
19	NK1-32CDR	16 points bidirectional DC24V input, 16 points relay output	
20	NK1-32CDT1	16 points bidirectional DC 24V input, 16 points NPN output	
21	NK1-32CDT2	16 points bidirectional DC 24V input, 16 points solid-MOSFET PNP output	*1
22	NK1-8AD4DA	8-channel input/4-channel output analog unit, module resolution is 12-bit, current and voltage is generic.	
23	NK1-4AD2DA	4-channel input/2-channel output analog unit, module resolution is 12-bit, current and voltage is generic.	
24	NK1-8AD4DA-H	8-channel input/4-channel output analog unit, module resolution is 16-bit, current and voltage is generic.	
25	NK1-4AD2DA-H	4-channel input/2-channel output analog unit, module resolution is 16-bit, current and voltage is generic.	
26	NK1-03Z	3-channel single-phase or A/B phase high-speed counter input unit, 100kHz	
27	NK1-04THM	4-Channel thermocouple input unit	
28	NK1-03SSI	3-Channel SSI sensor signal input unit	

***1: These expansion units are not to be ready now.**

1-2-3 Composition of NK1 function boards' model

40-point NK1 CPUs can install a function board to extend the functionality of the PLC. These boards include battery boards, ordinary input/output boards, analog boards, communication boards, and so on. Note that all function boards have a battery installing position, in other words，all function boards have the function of battery board.

The composition of NK1 function boards' model is similar with NK1 expansion units'. Normal digital I/O function boards are named according to the following rule. Special function boards are named with NK1-B plus specialized name. For example:

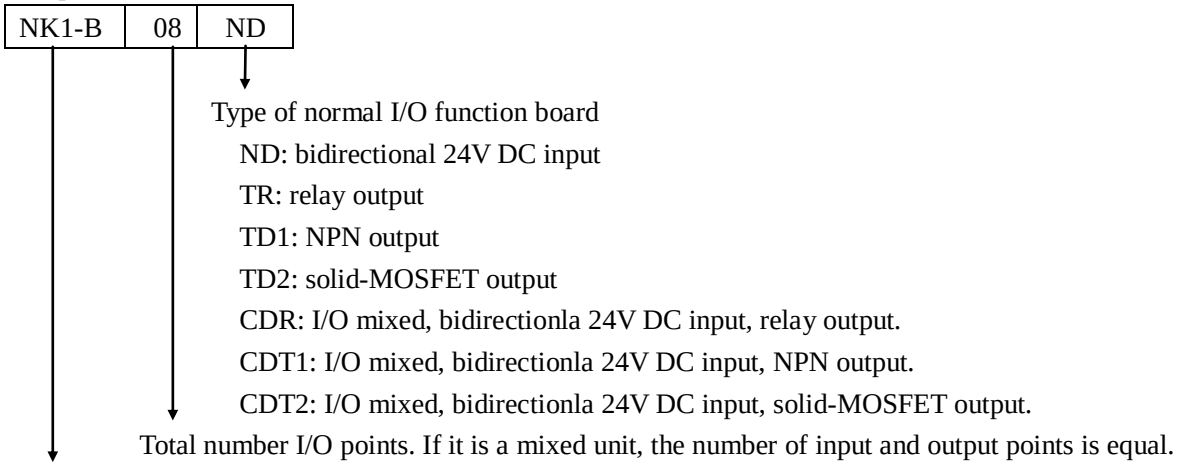
NK1-B4AD2DA-2: 4 channels input and 2 channels output analog voltage module;

NK1-BDCM : either RS232C or RS485 communication function board；

NK1-BAT: battery function board.

Note: the model of battry function board is somewhat special: it is named NK1-BAT instead of NK1-BBAT!

Composition of normal NK1 function boards' model



The following table lists all Models of the NK1 series function boards. **Note:** Due to product improvements or other reasons, the models listed in the table may be changed. In addition, function board which is listed in the table does not mean it is in stock, please contact the sales department for details.

No.	Model	Specifications	Remark
1	NK1-BAT	Battry board, retentive calendar and clock data	
2	NK1-B08ND	8 points bidirectional DC24V input board	*1
3	NK1-B08CDT1	4 points bidirectional DC24V input, 4 points NPN output board	*1
4	NK1-B4AD2DA-1	4-channel input/2-channel output analog board, 12-bit resolution, current type.	*1
5	NK1-B4AD2DA-2	4-channel input/2-channel output analog board, 12-bit resolution, voltage type.	*1
6	NK1-BDCM	serial communication board, one port 3-wire RS-232C/RS-485 selectable	*1
7	NK1-BCAN	CAN-bus communication (slave) board	*1

NOTE: *1 The function board include the function of battery board

1-3 NK1 PLC system configuration

1-3-1 NK1 PLC basic system configuration

NK1 CPU includes a variety of function components required to form a PLC system, including the CPU, input/output interface, system program, user program, function memory, etc, so a single CPU can constitute an independent PLC control system.

The following figure shows the shape of NK1-CPU40DM:

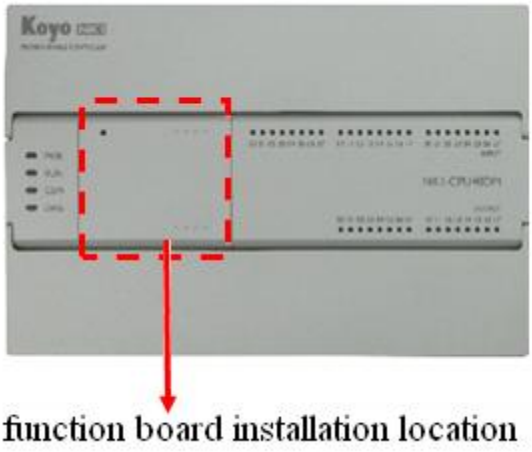
1-3-2 NK1 PLC expansion system configuration

NK1 PLC supports up to fourteen expansion units, including general I/O expansion units, analog expansion units, high-speed counter units and other special expansion units. And the installation location of each expansion unit has no special requirements.

The following figure shows a NK1-CPU40DM with 4 expansion units.

40 points NK1 CPU can also be expanded by adding an expansion board in addition to adding expansion units. Features of function boards include adding battry to retain calendar and clock data, increasing a small amount of I/O points, simple analog function, adding a serial communication port, and so on.

The installation location of a function board is shown on the right. When installing the function board, you must first remove the blank cover board and then install the function board in the location of the blank cover board.



1-3-3 Composition of NK1 PLC network

NK1 PLCs have Ethernet, RS232C, RS485 communiction ports.
The RS232C port is a 6-pin female modular (RJ12 phone jack) type connector. It is a programming port that can connect to KPPSoft software; it is also a universal communication port that can connect to HMI, PLC, etc. The protocols that it can use are CCM2, modbus RTU, non-sequence, and so on.
The RS485 port is a two-wire terminal. The protocols that it can use are CCM2, K-sequence, modbus RTU, non-sequence, and so on.

Note: RS485 port on NK1 PLC has no built-in termination resistors. If necessary, you can wire two termination resistors at both ends of RS-485 network, the standard terminal resistor value is 120 ohms.

40-point NK1 PLCs have a 10M/100M adaptive standard Ethernet port that supports ECOM, modbus TCP protocol,and the Ethernet non-sequence-protocol communication. The port can connect to KPPSoft software. The PLC can constitute a network by itself and also can connect to PLC which has an Ethernet port and supports ECOM UDP, modbus TCP protocol or non-sequence-protocol.
addition:The later NK1H-CPU20 sub-series PLCs also have a 10M/100M adaptive standard Ethernet port.

1-4 Peripheral devices

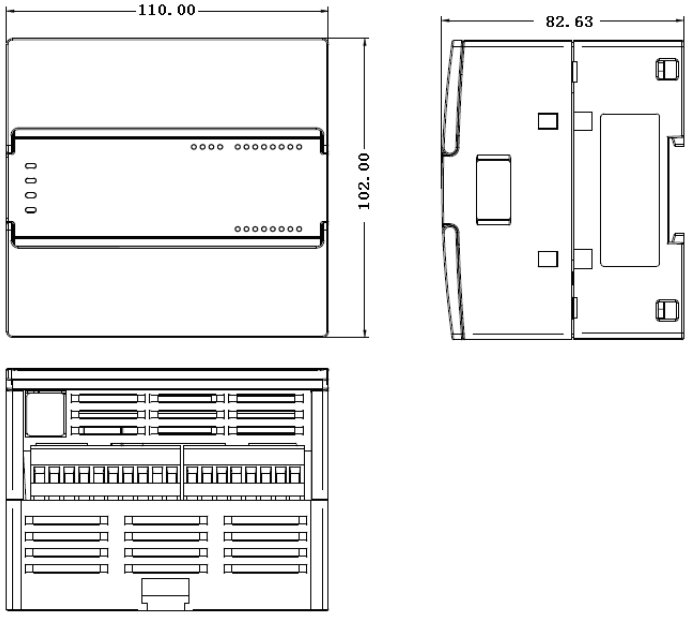
NK1 CPU has multiple communication ports which can connect a variety of communication type peripherals, including: KPPSoft programming software, industrial touch screen, operation display unit such as GT03, DV-1000(S-10D),GC-A2 series and other devices to exchange information.
The following table lists the functions of KPPSoft software.

No.	function	No.	Function
1	Program edit (modify, append, delete, etc) in ladder or mnemonic	22	Edit, display description text of function definition No., including aliases, wiring information and comments information
2	Program display in ladder view	23	Monitor program during PLC running
3	Program display in stage view	24	Edit password
4	Program display in mnemonic view	25	Display PLC system version information
5	Definition No. display in cross reference view	26	Readout scan time
6	Program syntax check	27	Setup retentive range
7	Retrieve instructions and definitions No.	28	Setup watchdog timer
8	Replace batch definition No.	29	Clear all registers
9	Compare two program content	30	Delete all program
10	Compare current program content with program in disk	31	Clear the current values of timer/counter
11	Inserts the specified file into the current program	32	Copy config data from PLC to disk
12	Save program to disk and read program from disk	33	Copy config data from disk to PLC
13	Write program to PLC and read program from PLC	34	Monitor I/O status during PLC running
14	Download program that has been locked to PLC	35	Forced ON/OFF
15	Edit and display memory contents	36	Forced to write register data
16	Monitor and change PLC operating mode	37	Monitor data during PLC running
17	Print program, notes, indicating content	38	Initialize system parameters
18	Display and edit program information	39	The cursor jumps to the specified address
19	Restore factory settings	40	Set the color and style of software
20	Edit and display loops and function blocks comments	41	User management, function limitations
21	Edit comments in multiple languages	42	Generate and transmit program files transferred via USB port

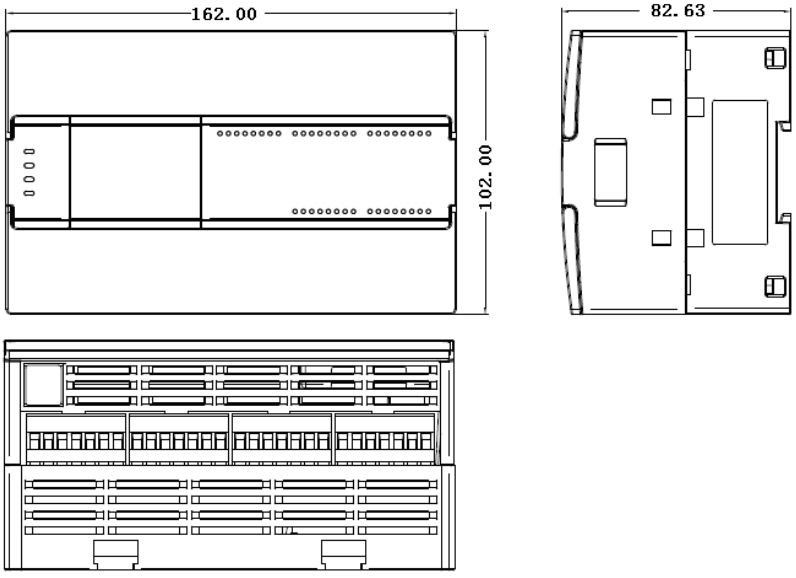
1.5 NK1 PLC dimensions

NK1 CPUs with same number of I/O points have the same dimensions; NK1 CPUs with different number of I/O points have different dimensions. 16-piont and 32-point expansion units have different dimensions.

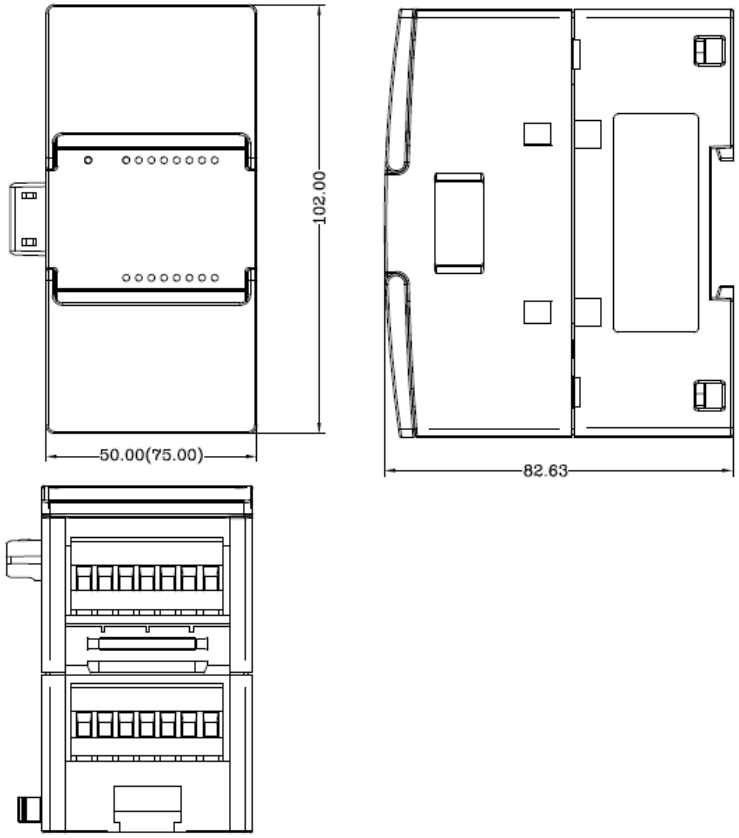
1） NK1-CPU20DM/DR/DD dimensions（unit：mm）



2）NK1-CPU40DM/DR/DD dimensions（unit：mm）



4）Expansion unit dimensions (unit：mm)



Chapter2 System specifications

2-1 General specifications

Items	Specifications	
Ambient operating temperature	0℃~55℃	
Storage temperature	-40℃~70℃	
Ambient humidity	30~95% (non-condensing)	
Atmosphere	No corrosive gases	
Rated supply voltage	AC power supply models: AC110V/220V	DC power supply models: DC24V
Power supply	85~265VAC 47~63Hz	DC19.2V~28.8V
Frequency	50/60Hz	---
Momentary power failure time	100ms，interval≥1s	
Voltage withstand	AC 1500V 1min, apply parts: between AC terminal/FG terminal/input terminal/output terminal	
Insulation resistance	DC 500V more than 20MΩ	
Noise immunity	1000V 1μs INPULSE	
Electrostatic discharge	8KV	
Shock resistance	At each of three vertical axes, haphazard amplitude is 15g with 11ms duration and impulses two times individually.	
Vibration resistance	Complicance with GB2423.10-81FC examinations prescribe. 10~57Hz, displacement amplitude is 0.075mm; 57~150Hz, acceleration is 10m/s^2. Scan 10 times individually at X, Y, Z directions in one octave per minute.	
Free fall	1000mm, 5 times, with product packaging.	
Maximum consume power	30W	
Sersor power supply	300mA 24VDC±15%	
Mounting methods	DIN rails mounting or screws mounting	
Weight (g)	NK1-CPU20DR: 422 NK1-CPU20DD: 410 NK1-CPU20DM: 422 NK1-CPU20DR-D: 417 NK1-CPU20DD-D: 415 NK1-CPU20DM-D: 417 NK1-CPU40DR: 590 NK1-CPU40DD: 548 NK1-CPU40DM: 590 NK1-CPU40DR-D: 585 NK1-CPU40DD-D: 548 NK1-CPU40DM-D: 585	
Dimensions (length×width×high)(mm）	NK1-CPU20 110L×102W×82.63H NK1-CPU40 162L×102W×82.63H 16 points expansion unit 50L×102W×82.63H 32 points expansion unit 75L×102W×82.63H	
Electromagnetic radiation	IEC 61000-6-4	
Certification	CE ROHS	

2-2 CPU specifications

Items		Performances
Control method		Cyclic execution method, time execution method
I/O transfer method		In a block transferring combine with immediate output and interrupt
Program language		Ladder /RLL Pluse stage programming
Mode switch		STOP/RUN（TERM-STOP/TERM-RUN）
Number of instruction		285 (please refer to “Appendix 1 NK1 RLL instructions” for details）
Processing velocity	Sequential instruction	0.1μs～
	Data processing instruction	1.0μ s ～
Scan time		1msec／1k words
Program size	Built-FROM	32k words（The using of KPPSoft software can lock the downloaded program for the purpose of protection）
I/O（I/Q）		I: 1024 points／Q: 1024 points
Communication I/O（GI/GQ）		GI: 2048 points／GQ: 2048 points（can be used as control relay）
Control relay (M)		2048 points
Timer (T)	Number of points	512 points
Counter (C)	Number of points	512 points
Special relay (SP)		1024 points
Stage (S)		1024 points
Data register（R/P） （Include system parameters, timer/counter current values）		16384 words 1 word = 16 bits
Accumulator (ACC)		1 (32 bits): for data processing
Data stack（DS1~8）		1 (32 bits): for auxiliary data processing
I/O assignment		I / O definition number is assigned automatically with configuration monitoring function
Calendar function	Points	1 point
	Specifications	Year , month, date, day, hour, minute, second, 1/100 second
Interrupt function		Yes, including external interrupt/ high-speed counter/ timed interrupt, a total of 14 interrupts
Immediate output		Yes, instruction support (only the output points of CPU are valid)
Data operation function		Support BCD, BIN arithmetic, floating point arithmetic, trigonometric, exponential logarithmic operation
Program protection		Password: 8 digitals BCD (In stop mode, enter 8 “C” restore factory settings) After you enter the wrong password three times, you must to re-power the NK1 CPU to enter the password again. Lock protection: When you are downloading the program, select program locked, the program downloaded to the NK1 CPU will not be read.

Continued

Items	Performances
Retentive data	Parameters (M，R，T，C，S) can be arbitrarily set. (Not require battery)
Battery (order separately)	Used for retentive calendar and clock data, battery model: RB-50
Self-diagnosis function	Communication error checking I/O configuration exception checking WatchDog timer checking Memory exception checking CPU' battery check (when equipped with battery)
Monitoring function	・block monitoring ・ON/OFF monitoring ・data monitoring ・I/O module operation display ・status monitoring ・self-diagnosis exception information display ・self-diagnosis exception history information display
Debugging function	・Forced SET/RESET ・Forced data is written ・Scanning stops ・Off-line simulation function（achieved by KPPSoft software）
Special function	The follwing functions can be selectd by setting the special register: ・5 channels single-phase or 3 channels two-phase high-speed counter (100 kcps) 4 channels single-phase or 2 channels two-phase high-speed counter (20-point CPU) (100 kcps) Interrupt function when the count reaches the preset value; high-speed counter reset function and counting inhibit function (hardware/software)； optional single octave, 4 octave ways to count. ・8 channels pulse catch or 8 channels external interrupt (with high-speed counting simultaneously) (4 channels on 20 points I/O CPU) ・2 channels pulse output (100 kHz) (with high-speed counting simultaneously) ・1 channel interrupt timer, interrupt time Optional (1ms-9999ms)
PID	Up to 16 PID loops
Peripheral equipment	Programming Tools Software : KPPSOFT HMI: CL-02DS, GT03, CMORE, EA7E, GC, GC-A2, etc.

Continued

Itmes		performance	
serial communication port（*Note）	Signal	RS-232C (non-insulated) (two ports on 20-point CPU)	RS-485 (non-insulated)
	Connector	6-pin phone connector	3-wire terminal block
	Transfer distance	Within 10 meters (without CPU power supply)	Within 1000 meters
	Protocol	K-sequence (slave), CCM2 (slave), MODBUS RTU (master/slave), non-sequence (master/slave)，KPPSOFT dedicated protocol	
	Speed	2400/4800/9600/14400/19200/28800//38400/57600/115200bps	
	Parity	None/odd/even	
	Station number	Set by special register	
	Data style	CCM2 protocl, HEX/ASCII (optional)	
	Parameter settings	Set by KPPSoft or by special registers	
Ethernet communication	Communication port	1	
	mode	10/100M adaptive	
	Transfer distance	Single-segment cable length is less than 100 meters (STP), can add HUB	
	Communication number	4 (MODBUS TCP/IP) (master/slave), 1(programming software), multiple ECOM UDP	
	Protocol	ECOM protocol (including KPPSoft dedicated protocol, MODBUS TCP/IP protocol, ECOM UDP protocol)	
USB communication port	Communication port	1	
	mode	Mini USB B type, USB2.0 FULL Speed	
	Transfer distance	Less than 5 meters	
	Connector	Standard USB2.0	
	Protocol	USB OTG supports point to point communication, downloading user program and system parameters (FAT format, 32G or less) to the NK1 PLC and system firmware upgrade; programming software KPPSoft dedicated protocol.	
NK1 series expansion	Expansion units	Each CPU can take up to fourteen expansion units.	
	Function board	40/64-point CPUs have a location to install function board, you can install a function board.	

***Note: The following is a few additional description for serial communication port:**

1）40/64-point CPU can add a RS-232C or RS-485 communication port by install an optional function board, the specifications of the additional communication port are same with the port's of CPU.

2）In order to facilictate the description, the communication ports of NK1 PLC are numbered in this manual: RS-232C port built-in CPU is called PORT0 (referred to as P0); 3-wire RS-485 port is called PORT1 (referred to as P1); the port on the optional function board is called PORT2 (referred to as P2) (the second RS-232C port on 20-point CPU is also called P2 port); Ethernet port is called PORT3 (referred to as P3). P0 is called programming port, other communication ports are called

universal communication port.

3）In TERM-STOP mode, the programming port (P0) acts according to the following fixed communications parameters: station No. = 1; communication speed = 19200bps; Stop bits = 1; Parity = no parity.

In other cases, P0 acts according to the communication parameters setup in its register.

Other communication ports act according to the communication parameters setup in their registers.

According to the different number of output points, the NK1 series PLCs are divided into NK1-CPU20, NK1-CPU40 and NK1-CPU64 sub-series, each sub-series has some different features, listed in the following table.

CPU	NK1-CPU20	NK1-CPU40	NK1-CPU64
I/O points number	20	40	64
Supports max. expansion units number	14	14	14
The number of expansion unit's I/O points	8/16/32	8/16/32	8/16/32
Support function board installed	No	Yes	Yes
Maximum I/O number of function board	-------	8	8
Maximum I/O number	468	496	520
Number of high-speed counter	4 channels single-phase or 2 channels A/B phase counter	5 channels single-phase or 3 channels A/B phase counter	5 channels single-phase or 3 channels A/B phase counter
Maximum pulse catch, external interrupt number	4 points	8 points	8 points
Maximum pulse output number	2 channels/4 points	2 channels/4 points	2 channels/4 points
Maximum PID loop number	16	16	16
Ethernet communication port on the CPU	None	1	1
RS-232C communication port on the CPU	2	1	1
RS-485 communication port on the CPU	1	1	1
Serial communication port on the functional board	None	1（RS-232C or RS485）	

Note：You can only install no more than one function board, if you have installed any other I / O function board, you can not install the serial communication port functional board.

2-3 I/O specifications of the NK1 series PLC

Before introducing I/O specifications of the NK1 series PLC, we need to introduce the concepts of sinking and sourcing, I/O common terminal and wiring strategies.

2-3-1 Concepts of sinking (Sink) and sourcing (Soure)

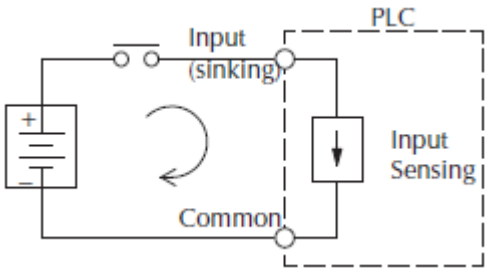
"Sinking" and "sourcing" and other terms often appear in the description of typical input or output circuits. It is the goal of this section to make these concepts easy to understand. First, we give the following short definitions, followed by practical applications.

Sinking = Path to supply ground (–)

Sourcing = Path to supply source (+)

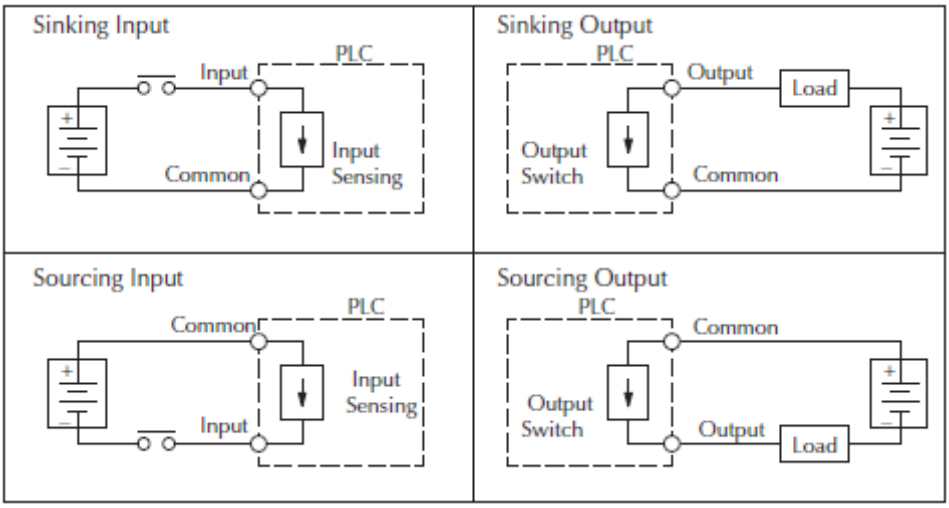
Notice the reference to (+) and (–) polarities. Sinking and sourcing terminology applies only to DC input and output circuits. Input and output points that are either sinking or sourcing can conduct current in only one direction. This means it is possible to connect the external supply and field device to the I/O point with current trying to flow in the wrong direction, and the circuit will not operate. However, we can successfully connect the supply and field device every time by understanding sourcing and sinking.

For example, the figure to the right depicts a sinking input. To properly connect the external supply, we just have to connect it so the input provides a path to ground (–). So, we start at the PLC input terminal, follow through the input sensing circuit, exit at the common terminal, and connect the supply (–) to the common terminal. By adding the switch, between the supply (+) and the input, we have completed the circuit. Current flows in the direction of the arrow when the switch is closed.



By applying the circuit principle above to the four possible combinations of input/output sinking/sourcing types, we have the four circuits as shown below. Note that the current direction in the figure is the current direction of the PLC internal circuit

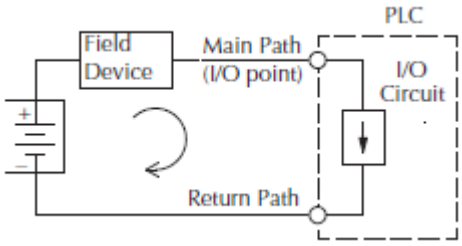
In the following description of I/O specifications, you will find that all of the inputs on the NK1 series CPU are bidirectional. That is, these inputs can be connected to either sinking or sourcing circuit loops; the transistor outputs on the CPU are generally sinking outputs.

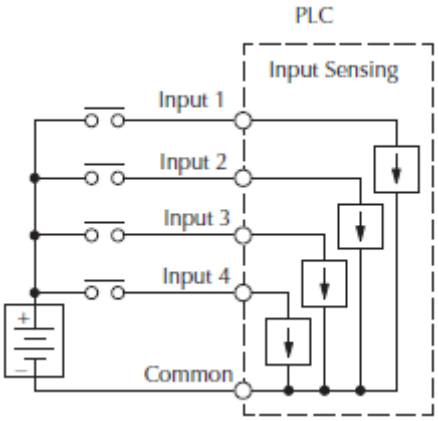


We referred to I/O common terminal concepts in the above description. Then what is I/O terminal? The follwing is the description of I/O common terminal concepts.

2-3-2 I/O common terminal concepts

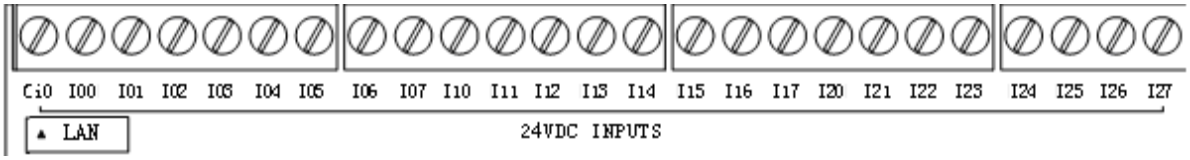
In order for a PLC I/O circuit to operate, current must enter at one terminal and exit at another. This means at least two terminals are associated with every I/O point. In the figure to the right, the input or output terminal is the main path for the current. One additional terminal must provide the return path to the power supply.
Most input or output point groups on PLCs share the return path among two or more I/O points. The figure to the right shows a group (or bank) of 4 input points which share a common return path. In this way, the four inputs require only five terminals instead of eight.



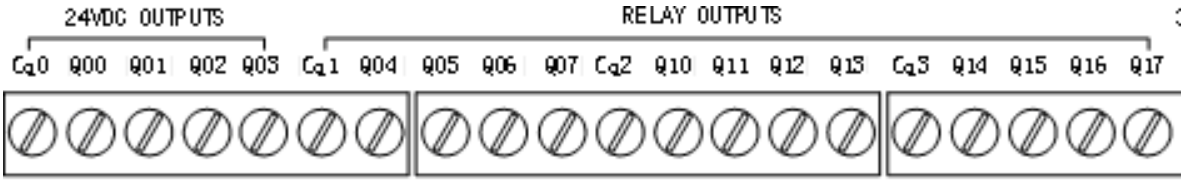


NOTE: In the circuit to the right, the current in the common path is 4 times any channel's input current when all inputs are energized. This is especially important in output circuits, where heavier gauge wire is sometimes necessary on commons.

All NK1 PLC's input and output circuits are using the method described above, that is, multiple inputs or outputs share a common terminal, note that the common terminal is electrically isolated from each other. This can clearly be seen in the following NK1 PLC wiring diagram.
The following figure is input terminal diagram of 40-point CPU, you can see that I0~I27 share a common terminal Ci0.



The following figure is the complete output terminals diagram of NK1-CPU40DM, 4 point are grouped into a bank that shares a common terminal. The banks provide Cq0, Cq1, Cq2, Cq4 four common terminals. Banks are electrically isolated from each other.



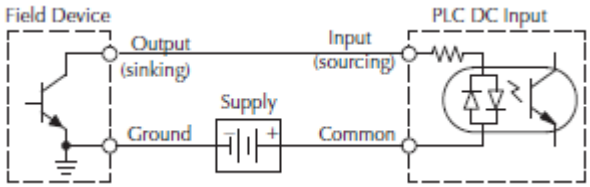
2-3-3 Connecting DC I/O to Solid State Field Devices

In the previous section on sinking and sourcing concepts, we discussed DC I/O circuits that only allow current to flow in one way. This is also true for many of the field devices which have solid-state (transistor) interfaces. In other words, field devices can be sourcing or sinking. When connecting the two

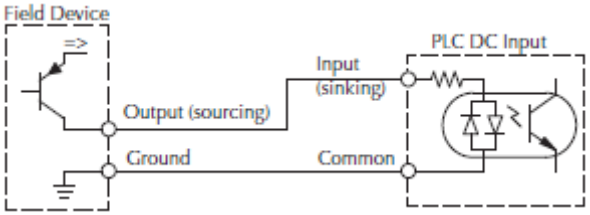
devices in a series DC circuit (as is the case when wiring a field device to a PLC DC input or output), one must be wired as sourcing and the other as sinking.

1. Solid state input sensors

The NK1's DC inputs are flexible that they detect current flow in either direction, so they can be wired as either sourcing or sinking. In the following circuit, a field device has an opencollector NPN transistor output. It sinks current from the PLC input point, which sources current. The power supply is 24 VDC.



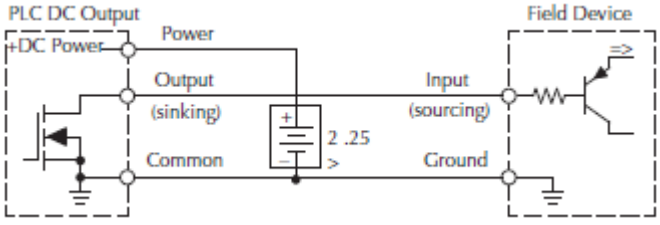
In the next circuit, a field device has an open-emitter PNP transistor output. It sources current to the PLC input point, which sinks the current back to ground. Since the field device is sourcing current, no additional power supply is required between the device and the PLC DC Input.



Since the NK1's DC inputs can be wired as either sourcing or sinking, so for 20 points/40 points PLC which has multiple input banks, you can wire one bank sourcing and other bank sinking. Note that inputs in the same bank must be wired as the same soucing or sinking.

2. Solid State Output Loads

Sometimes an application requires connecting a PLC output point to a solid state input on a device. This type of connection is usually made to carry a low-level signal, not to send DC power to an actuator.
There is only sinking output on NK1 series CPU, so each output provide a path to supply ground; 4 solid state outputs are grouped into a bank that share a common terminal.
In the following circuit, the PLC output point sinks current to the output common when energized. It is connected to a sourcing input of a field device input.



2-3-4 Relay output wiring methods

DR/DM models have a certain number of relay outputs. Relays are best for the following applications:

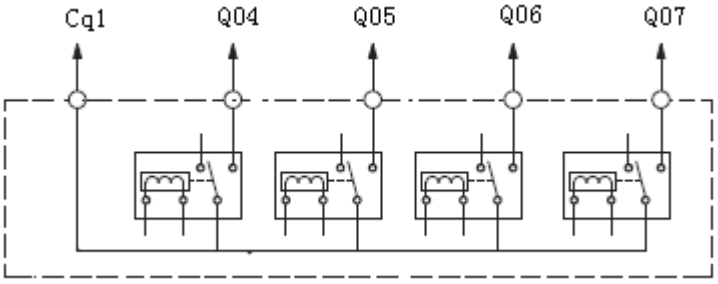
- Loads that require higher currents than the solid-state outputs can deliver;
- Loads which be switched at low speed;
- Some output channels need isolation from other outputs (such as when some loads require AC while others require DC).

Some applications in which NOT to use relays:

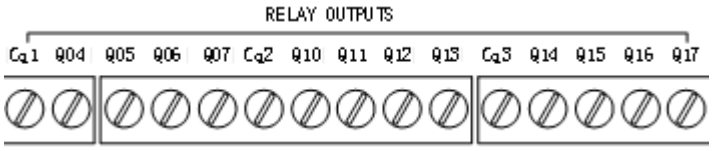
- Loads that require currents under 10 mA;
- Loads which must be switched at high speed and duty cycle.

The following describes methods to wire relay outputs.

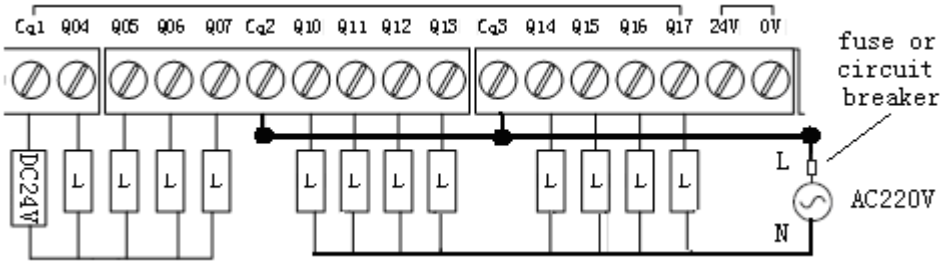
Every 4 relay outputs on NK1 series PLC are grouped into a bank, each bank is electrically isolated from the other bank of outputs. For example, NK1-CPU40DR has 16 relay outputs which be grouped into 4 banks, NK1-CPU40DM has 12 relay outputs which be grouped into 3 banks (the first 4 outputs are transistor outputs). The figure below shows internal circuit of the first bank of relay outputs on NK1-CPU40DM.



The figure below shows terminals of all relay outputs on NK1-CPU40DM.



When connecting field devices with relay outputs, you can power different banks with different power supply; you can also power different banks with the same power supply. For example, in the circuit below, the output points in the firs bank (Cq1 bank) use a separate DC24V supply; the output points in the second bank (Cq2 bank) and the third bank (Cq3 bank) use the same AC220V power supply. The common terminal Cq2 and Cq3 are connected together. In this case, common terminal Cq1 is independent and can be used to connect separate loads.



When using relay outputs to drive the field device, you need to pay special attention to the following two

points.

1. Surge Suppression For Inductive Loads

Inductive load devices (devices with a coil) generate transient voltages when de-energized with a relay contact. When a relay contact is closed it bounces, which energizes and deenergizes the coil until the bouncing stops. The transient voltages generated are much larger in amplitude than the supply voltage, especially with a DC supply voltage.

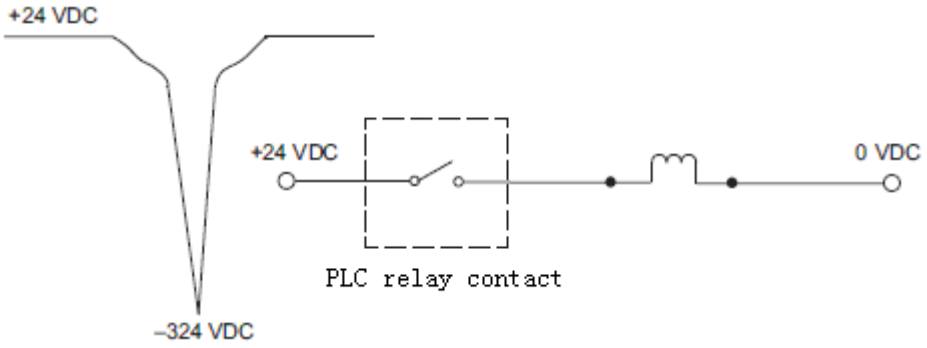
When switching a DC-supplied inductive load the full supply voltage is always present when the relay contact opens (or "bounces"). When switching an AC-supplied inductive load, there is one chance in 60 (60 Hz) or 50 (50 Hz) that the relay contact will open (or bounce) when the AC sine wave is zero crossing. If the voltage is not zero when the relay contact opens there is energy stored in the inductor that is released when the voltage to the inductor is suddenly removed. This release of energy is the cause of the transient voltages.

When inductive load devices (motors, motor starters, interposing relays, solenoids, valves, etc.) are controlled with relay contacts, it is recommended that a surge suppression device be connected directly across the coil of the field device. If the inductive device has plugtype connectors, the suppression device can be installed on the terminal block of the relay output.

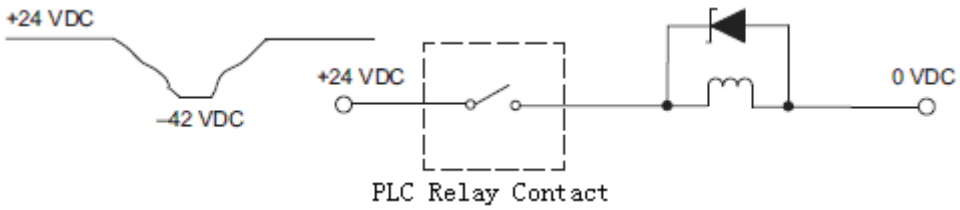
Transient Voltage Suppressors (TVS or transorb) provide the best surge and transient suppression of AC and DC powered coils, providing the fastest response with the smallest overshoot.

Metal Oxide Varistors (MOV) provides the next best surge and transient suppression of AC and DC powered coils.

For example, the waveform in the figure below shows the energy released when opening a contact switching a 24 VDC solenoid. Notice the large voltage spike.



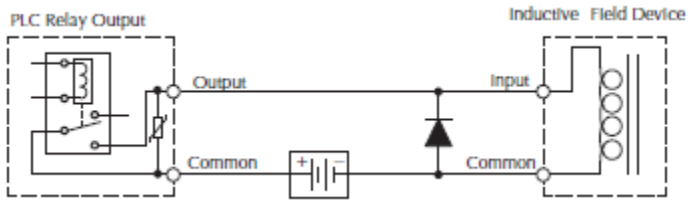
This figure shows the same circuit with a transorb (TVS) across the coil. Notice that the voltage spike is significantly reduced.



2. Prolonging Relay Contact Life

Relay contacts wear according to the amount of relay switching, amount of spark created at the time of open or closure, and presence of airborne contaminants. There are some steps you can take to help prolong the life of relay contacts, such as switching the relay on or off only when it is necessary, and if

possible, switching the load on or off at a time when it will draw the least current. Also, take measures to suppress inductive voltage spikes from inductive DC loads such as contactors and solenoids.
For inductive loads in DC circuits we recommend using a suppression diode as shown in the following diagram (DO NOT use this circuit with an AC power supply). Use a fast-recovery type (such as Schottky type). DO NOT use a small-signal diode such as 1N914, 1N941, etc. Be sure the diode is in the circuit correctly before operation. If installed backwards, it shortcircuits the supply when the relay energizes.



2-3-5 Inputs specifications

All inputs on NK1 CPU are bidirectional transistor inputs, require external DC24V power supply. All inputs on 40-point CPUs share an input common terminal, all inputs must use the same wiring methods. 20-point CPUs have two input common terminals. The two common terminals are electrically isolated. Different wiring method can be used between different common terminal and inputs.

Items	Specifications
Inputs number per module	12 points NK1-CPU20DR/DD/DM（I0~I3 high-speed counter inputs）
	24 points NK1-CPU40DR/DD/DM（I0~I5 high-speed counter inputs）
Open circuit voltage of input terminal	DC +24V
Rated input current	TYP 4mA(DC 24V)
Input impedance	Appro. 6KΩ
OFF voltage level	≤DC 5V
ON voltage level	≥DC 15V
Input response (normal)	OFF→ON ≤5ms (software filter can be set) ON→OFF ≤5ms (software filter can be set)
Input response(special)	High-speed counter *1: OFF→ON ≤5μs ON→OFF ≤5μs Pulse catch *2: OFF→ON ≤50μs ON→OFF ≤50μs
Insulation method	Optocoupler isolation
commons	1 common（20-point/64-point CPU，2 commons）
Commons polarity	none，bidirectional
Action indicates	LED indicates
Terminal type	Removable
Wire gauge	12~28AWG

*1：When I0~I5 are used as high-speed counter input（20- point CPU: I0~I3）;

*2：When I10~I17 are used as external interrupt, pulse catch input（20- point CPU: I0~I3）;

Inputs on NK1 CPU are divided into special and general inputs, I0~I5 (20-point CPU: I0~I3) are special high-speed counter input, the others are general input.

All inputs can be used as general inputs used to connect signal of switches or sensors.

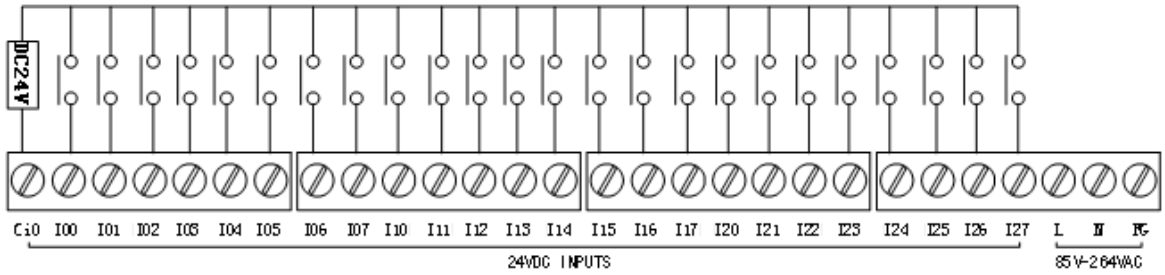
In addition to be used as general inputs, special inputs can also be used as high-speed counter inputs.

When you are using high-speed counter function, general inputs I10~I17 can be used as counting prohibition and counting reset inputs, they can also be used as external interrupt and pulse catch inputs.

Please refer to chapter 5 “High-speed input and pulse output feature” for more information of special inputs.

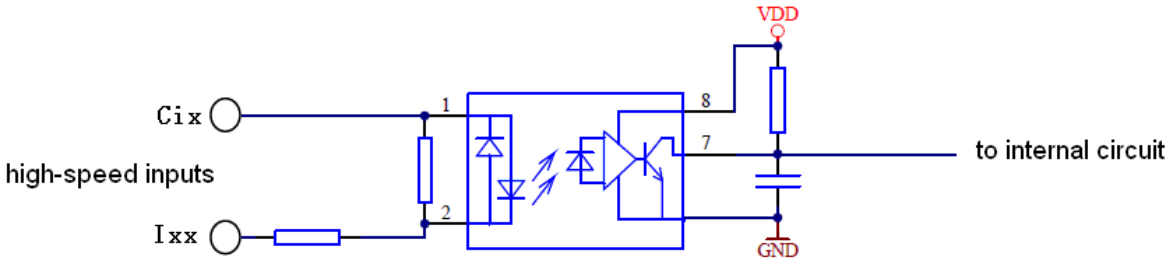
All inputs on NK1-CPU40DM share an input common terminal, which is the same with NK1-CPU40DR and NK1-CPU40DD.

The figure below is a wiring example of inputs on NK1-CPU40DM:

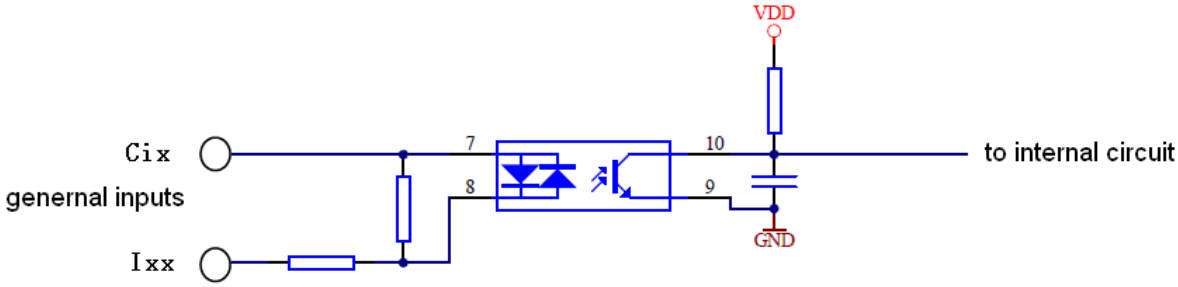


Note: 20-point and 64-point CPU have two or more common terminals, which are independent from each other.

The figure below shows internal electrical schematics of special inputs I0~I5 (20-point CPU: I0~I3)

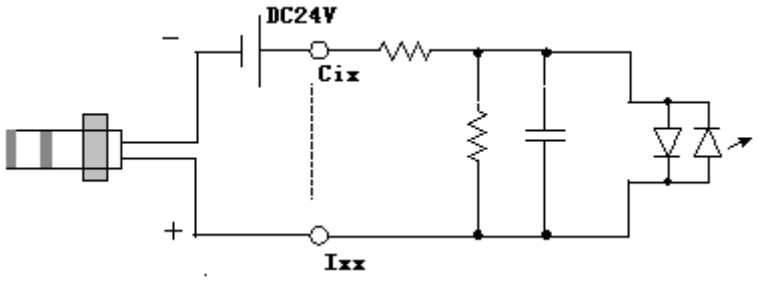


The figure below shows electrical schematics of general input:

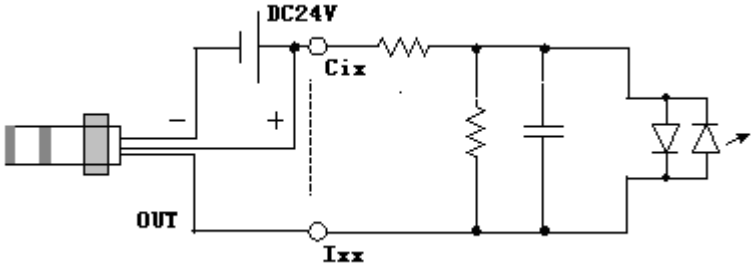


Here are some wiring diagram examples (sensors' inputs are all NPN type):

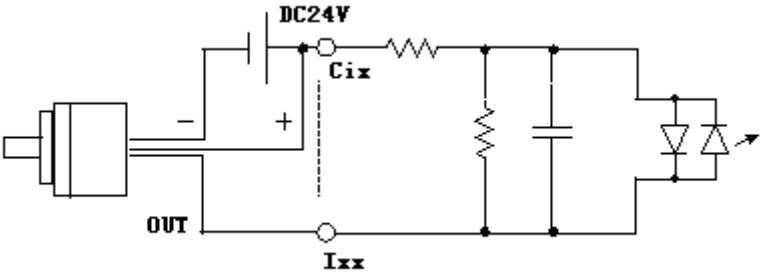
Wiring example with 2-wire proximity switch



Wiring example with 3-wire NPN type proximity switch



Wiring example with rotary encoder



2-3-6 Relay outputs specifications

DR/DM models have a certain number of relay outputs.

Items	Specifications
Outputs per module	4 points NK1-CPU20DM
	8 points NK1-CPU20DR
	12 points NK1-CPU40DM
	16 points NK1-CPU40DR
Operating voltage range	AC 85V~264V DC 5V~30V
Frequency	47~63Hz
Maximum load current	2A/point (resistive); maximum 8 A per bank
Maximum leakage current(cross contact)	0.1mA（AC 220V）
Maximum switching capacity	a. Resistive load: 220V AC 2A b. Resistive load: 28V DC 2A c. Inductive load: 220V AC 0.25A d. Inductive load: 28V DC 0.25A
Minimum off the load voltage/current	AC/DC 5V/5mA
Output response time	OFF→ON: ≤10ms ON→OFF: ≤10ms
ON/OFF times	200,000（current is 2A）
Surge suppression circuit	none (the user can increase outside)
Protection circuit	None (the user can increase protection devices such as varistors)
Common	1 common per 4 points (commons are electrically isolated from each other)
Common polarity	None
Action indicates	LED indicates
Terminal type	Removable
Wire gauge	14~28AWG

Every 4 relay outputs on NK1 series PLC are grouped into a bank. The 4 relay outputs share a common terminal Cqx. The number of common terminals equal to the number of banks. Outputs in the first bank on DM model are transistor outputs, the rest are relay outputs. All outputs on DR model are relay outputs. Banks are electrically isolated from each other.

The figure below is a wiring example of relay outputs on NK1-CPU40DM:

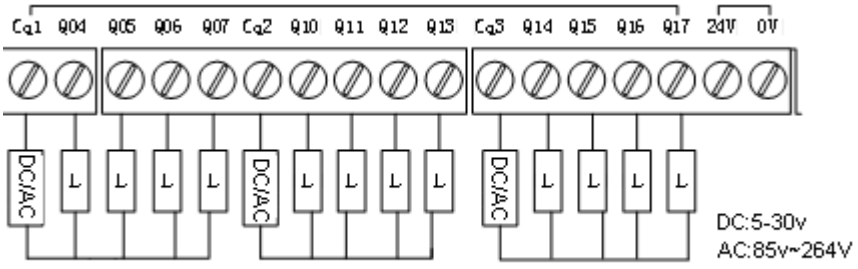
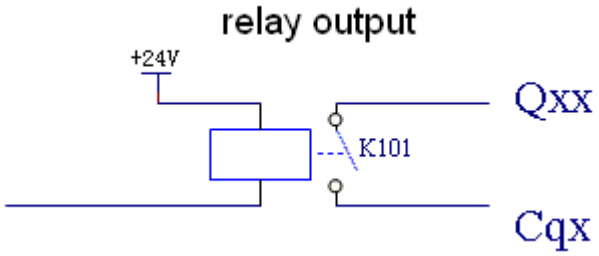


Figure above, if necessary, the banks can share a power supply. To do this, you need to connect the common terminals of each bank together, then connect it to one end of power supply, the other end of

power supply is connected to the loads and then to output terminals.

The figure below shows internal electrical schematics of relay outputs on NK1 CPU:



2-3-7 Transistor outputs specifications

Outputs in the first bank on DM model are transistor outputs. All outputs on DD model are transistor outputs. Transistor outputs are all NPN open collector outputs.

Items	Specifications
Outputs per module	4 points NK1-CPU20DM
	8 points NK1-CPU20DD
	4 points NK1-CPU40DM
	16 points NK1-CPU40DD
Output mode	NPN open collector output
Switching capacity	DC24V 0.3A
Maximum leakage current	100μA（DC 40V）
Maximum residual voltage	0.5V（0.3A）
Allow peak voltage	DC 40V（include peak voltage of ripple）
Output response （general ）	OFF→ON: ≤0.1ms ON→OFF: ≤0.1ms
Output response（pulse output）*1	OFF→ON: ≤5μS ON→OFF: ≤5μS
Protection circuit	3A（soldering a fuse in common side every 4 outputs,user increase outside）
Common	1 common per 4 points (commons are electrically isolated from each other)
Common polarity	Common emitter
Action indicates	LED indicates
Terminal type	Removable
Wire gauge	14~28AWG

*1: It applies only when the four transistor output points in the first bank using the pulse output function.

Transistor outputs in the first bank on DM/DR model are special transistor outputs. In addition to connect ordinary transistor input field devices, these outputs also can be set to special pulse outputs outputing high-speed pulse, to dirve pulse input stepper or servo motors and form a simple pulse positioning system. Please refer to chapter 5 “NK1 series PLC high-speed I/O function” for details.
The other transistor outputs on DD model are ordinary open collector outputs.

Every 4 transistor outputs on NK1 series PLC are grouped into a bank. The 4 relay outputs share a common terminal Cqx. The number of common terminals equal to the number of banks. Outputs in the first bank on DM model are transistor outputs, the rest are relay outputs. All outputs on DR model are relay outputs). Banks are electrically isolated from each other.

The figure below is a wiring example of transistor outputs on NK1-CPU40DM:

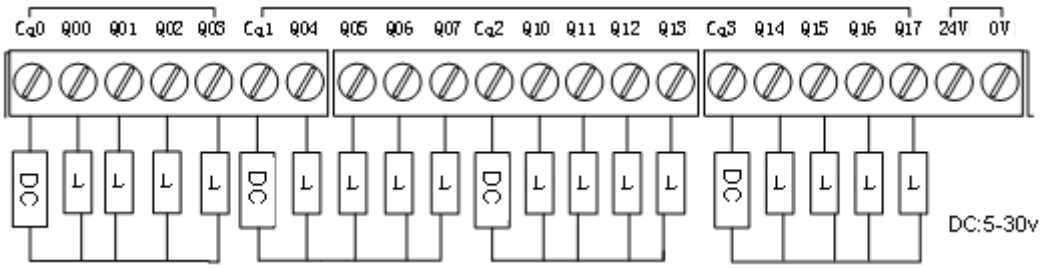
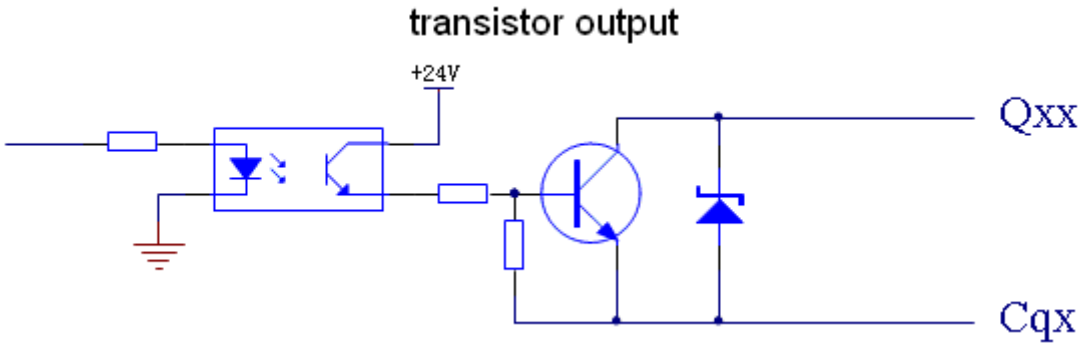
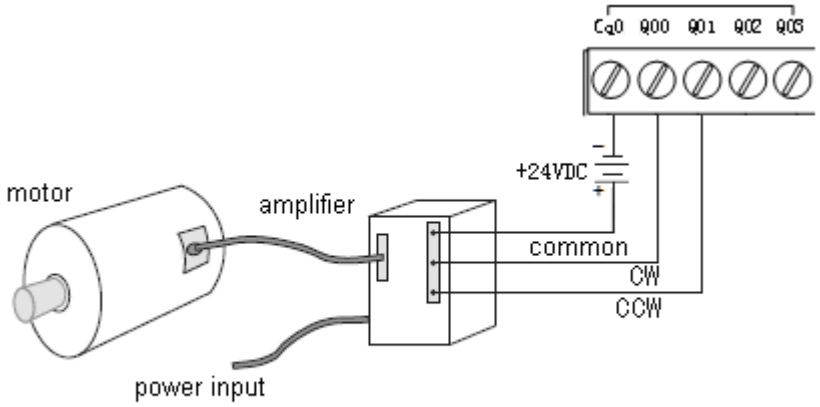


Figure above, if necessary, the banks can share a power supply. To do this, you need to connect the common terminals of each bank together, then connect it to one end of power supply, the other end of power supply is connected to the loads and then to outputs.

The figure below shows internal electrical schematics of transistor outputs on NK1 CPU:



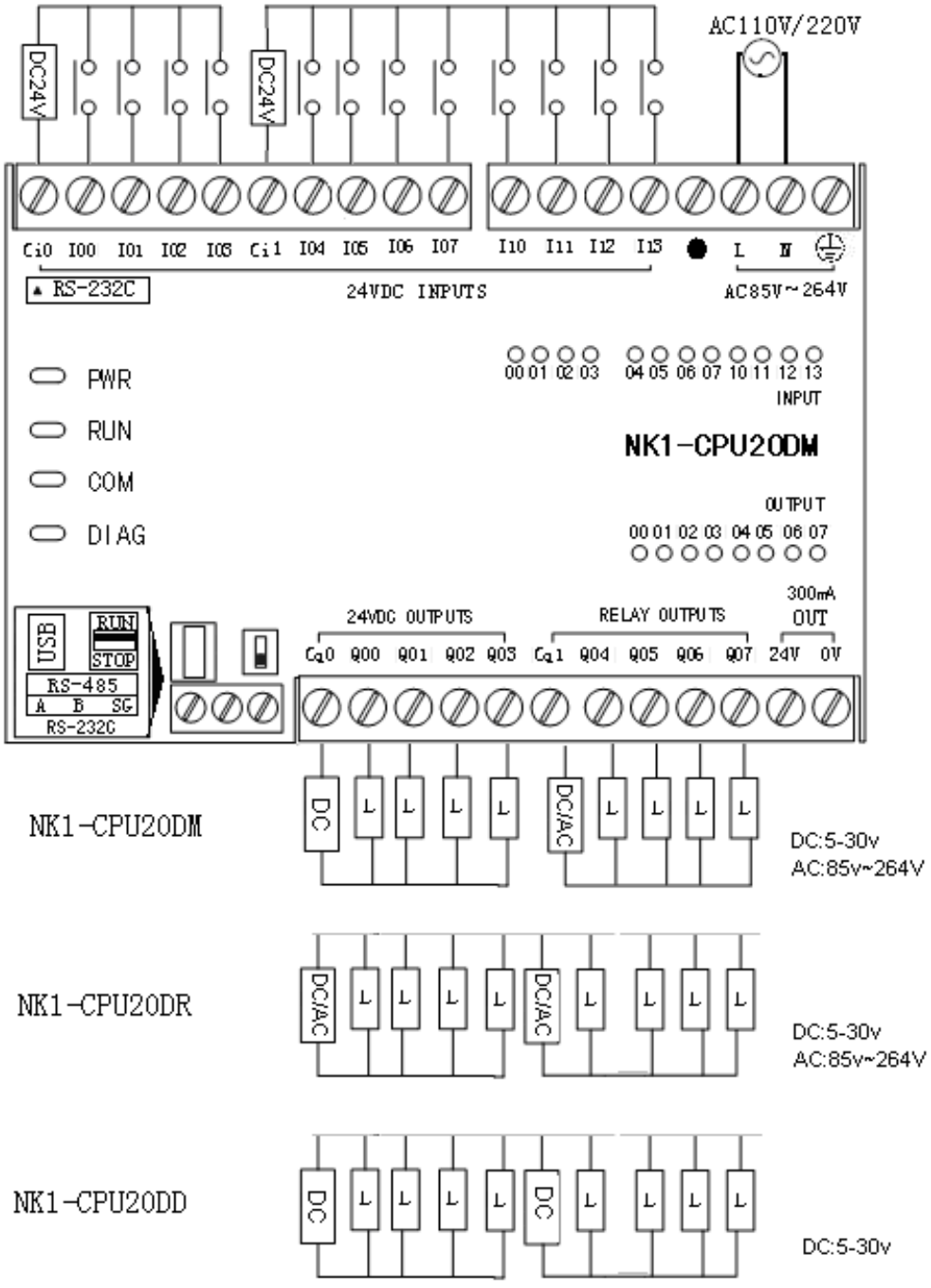
The figure below is a wiring example of using pulse output function. Note that since transistor outputs on NK1 series PLC are NPN open collector sinking output, the amplifier inputs need to be soucing inputs.



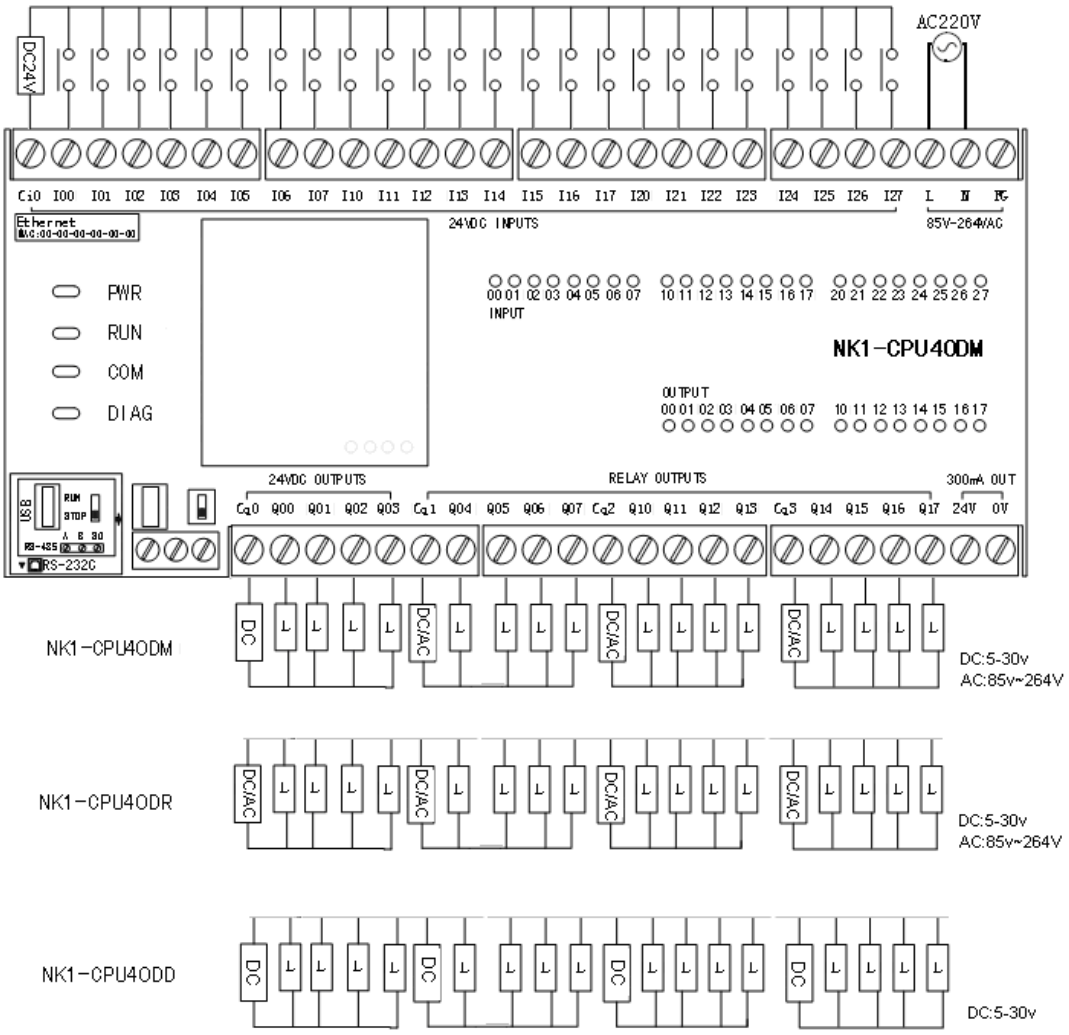
wiring example of pulse output

2-3-8 NK1 CPU I/O wiring diagram

1、 NK1-CPU20DM/DR/DD I/O wiring example diagram



2、 NK1-CPU40DM/DR/DD I/O wiring example diagram



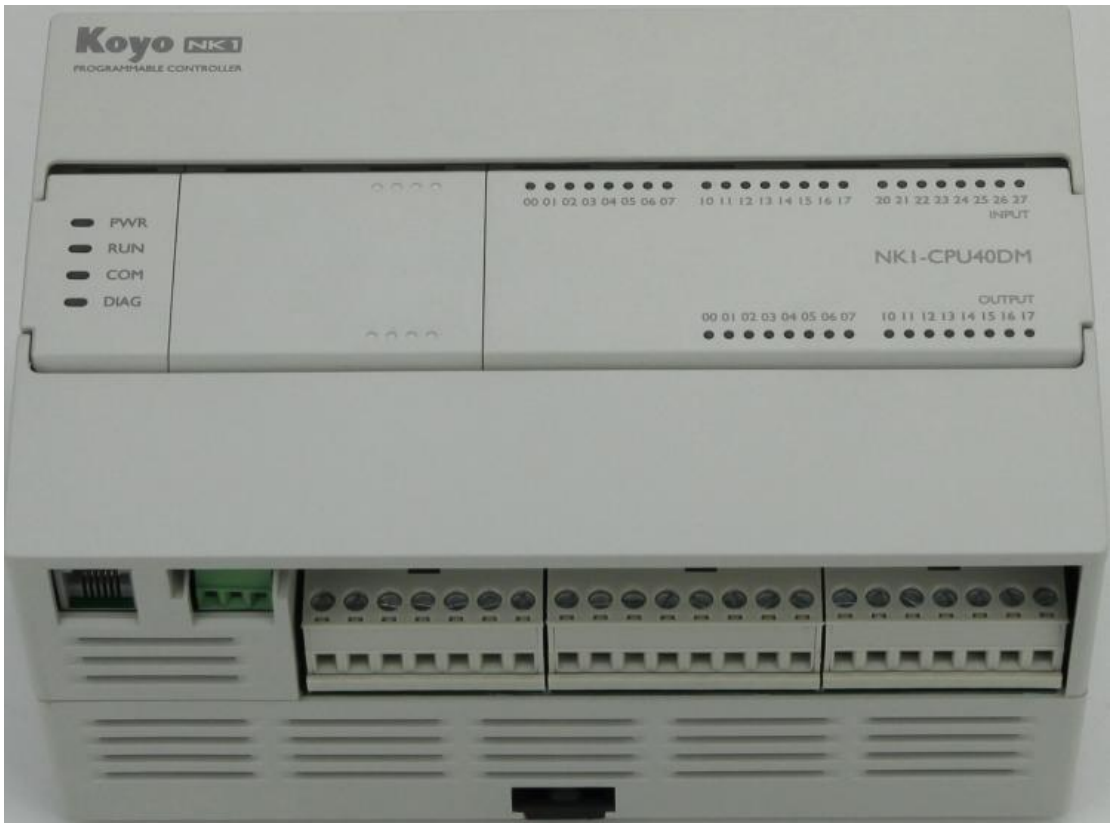
Chapter 3CPU specifications and operation

3-1 CPU hardware

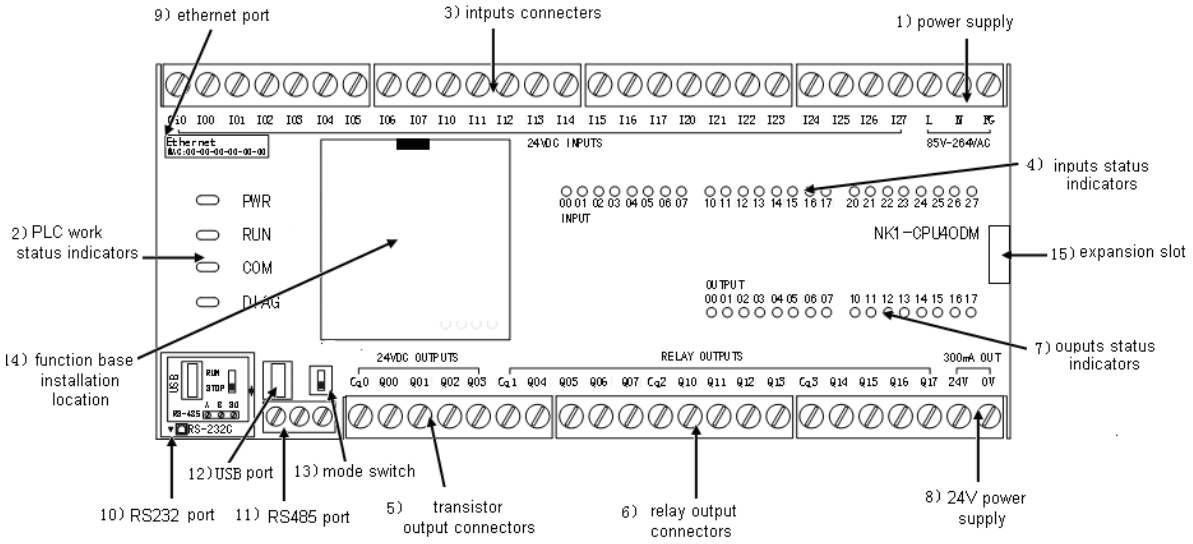
NK1 series PLCs are integral and extendable. A NK1 CPU module includes all the components of a digital PLC control system, such as CPU, memory, communication ports, inputs/outputs, and so on. Therefore, a separate CPU module can constitute an application system. The using of expansion units and function board can improve the performance of NK1 PLC, such as increasing digital/analog inputs/outputs, high-speed counter inputs, communication ports, etc.

NK1 CPUs are divided into three sub-series: CPU20, and CPU40 according to the difference of CPUs' I/O number. Each sub-series are divided into DR, DM and DD according to the difference of output type. The shape of CPU modules with same I/O number is basically the same. The shape of CPU modules with different I/O number is different, but the distribution position of each function parts is basically the same.

Here we take NK1-CPU40DM as an example, introduce the hardware configuration of NK1 CPUs. The figure below shows the appearance of a NK1-CPU40DM module. There are two terminal covers on upper and lower front panel.



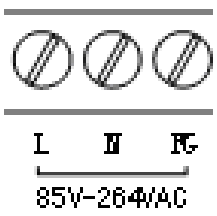
The figure below shows the appearance of a NK1-CPU40DM module without the two covers.



The above figure shows that two rows of terminals are located on upper and lower front panel of NK1 CPU. The upper row of terminals is power supply and inputs connectors; the lower row of terminals is outputs and DC24V connectors which powers the sensors. There are three groups of LED indicators on front panel of NK1 CPU; the four LED indicators on the left side indicate the working status of PLC. The upper right set of LEDs indicates the status of inputs. The lower right set of LEDs indicates the status of outputs. The position at the right of PLC working status LEDs is mounting position of function board (factory filled with an empty cover). There is an expansion slot on the right side of the CPU for plugging expansion unit. There are multiple communictation ports on CPU: 8-pin RJ45 port on the upper left side of CPU is Ethernet communication port (NK1-CPU20 here is a 6-pin RS232C communication port); a RJ12 port on the lower left side of CPU is a RS232 communication port; in addition, there is a three terminals RS485 communication port, a mini USB port and 2-position mode switch on the front panel.

The following is detailed description of the function parts of NK1 CPU.

1） Power supply connectors



Power supply connector is used for connecting external power supply. For AC power supply models, NK1 CPUs provide L, N and ⏚ terminals, allow the range of operating power is AC85V~264V. For DC power supply models, NK1 CPUs provide D+, D- and ⏚ terminals; allow the operating power is DC24V.

2） PLC work status indicators

The four LED indicators are used to indicate the basic working status of NK1 CPU. The group LEDs from top to bottom is PWR (power), RUN (run), COM (communication) and DIAG (diagnostics).

PWR: green power indicator is used for monitoring the status of the internal power supply.

ON: the internal power supply is normal;

Off: the internal power supply is abnormal, possible cause is not connected to an external power source or NK1 CPU's power failure.

RUN: green running indicator is used for monitoring the operation of user program in the CPU.

ON: the user program is running;

OFF: the user program is not running.

COM: red communication indicator is used for monitoring communication status of ports built-in NK1

CPU.

Flash: flashing alone no rules indicates that at least one communication port is in communication state.

Off: all communication ports built-in NK1 CPU is not in communication state.

DIAG: red diagnostic status indicator is used for monitoring operating status of the system.

On: an error occurred when power-up or when the system is running.

Flash: flashing alone at 1 second intervals indicates: in battery mode, no battery is installed or battery voltage is low, the battery needs to be replced.

Off: the system is in normal working state.

Further, in the case of PWR indicator on, combination flashing of RUN, COM and DIAG indicator has some special meaning:

RUN indicator off, COM and DIAG indicator flashing together at 1 second intervals, which means that the firmware system self-test error.

COM and DIAG indicator off, RUN indicator flashing alone no rules, which means that the system is being upgraded.

DIAG indicator off, RUN and COM indicator flashing together no rules, which means that the expansion unit is being upgraded.

RUN, COM and DIAG indicator flashing together at 1 second intervals, which means that the system upgrade is complete.

COM indicator off, RUN and DIAG flashing together no rules, which means that the project file package is downloading.

RUN indicator on, COM and DIAG indicator flashing together at 1 second intervals, which means that the project file package has been downloaded completely.

RUN, COM and DIAG indicator flashing together at 100ms intervals, which means that PF signal is detected. The cause is often that the USB port is connected with a USB cable, please unplug the USB cable.

RUN and COM indicator on, DIAG indicator flashing at 1 second intervals, which means that special feature within the system is in use.

Status	PWR	RUN	COM	DIAG
No external power or low power supply voltage	Off	Off	Off	Off
Powerup/stop	On	Off	———	———
Run	On	On	———	———
PF signal is detected	On	Flash(100ms interval)	Flash(100ms interval)	Flash(100ms interval)
System upgrade is complete	On	Flash(1s interval)	Flash(1s interval)	Flash(1s interval)
expansion unit is being upgraded	On	Flash	Flash	Off
system is being upgraded	On	Flash	Off	Off
communication port is in communication state	On	———	Flash	———
In battery mode, no battery is installed or battery voltage is too low to be replced.	On	———	———	Flash(1s interval)

Expansion system check error	On	Off	————	On
The firmware system self-test error.	On	Off	Flash(1s interval)	Flash(1s interval)
The project file package is downloading.	On	Flash	Off	Flash
The project file package has been downloaded completely.	On	On	Flash(1s interval)	Flash(1s interval)
Spcial feature within the system is in use.	On	On	On	Flash(1s interval)

3） Inputs terminals

Inputs terminals are used for connecting various external switches and sensor signals. All inputs on NK1 CPU are bidirectional DC24V input circuits which can connect to sinking or sourcing signal. Sub-series 40-point CPUs only provide an input common terminal, all inputs use it. So, all inputs on 40-point CPUs can only use the same input stype, either all soucing or all sinking. Sub-series 20-point and 64-point CPUs provide more common terminals, the common terminals are electrical isolated from each other. All inputs are grouped into banks which has an independent common terminal. Inputs in different banks can be different input stype, but inputs in the same bank must be the same input stye.

Also note: inputs on NK1 CPUs don’t provide internal 24V connection, so you should supply external DC24V power.

4) Input status indicators

Red indicators are used for monitoring internal on/off status of inputs. An indicator on means its corresponding input is on. You can simply determine input circuits are good or bad by input indicators.

5) Transistor output terminals

There are four transistor outputs on DM/DD type CPUs.The four transistor outputs are NPN open collector type outputs and generally used to connect high-speed pulse measurement devices.

Of course, the four outputs can also be used as ordinary digital outputs and connect transistor input type loads.

Transistor outputs are best for the following applications:

- Loads that require lower current
- Loads which be switched at high speed

Cq0 terminal is the common terminal of the four transistor outputs. It is independent from other three relay output common terminals.

Since DR type NK1 CPUs have no transistor outputs, so they have no high-speed pulse output function.

6) Relay output terminals

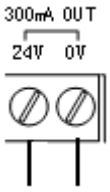
Relay output terminals are used to connect a variety of common loads. Relay outputs can be connected to an AC load and a DC load. When connecting the DC load, the outputs can be bidirectional. Relay outputs are best for the following applications:

- Loads that require higher currents than the solid-state outputs can deliver
- Loads which be switched at low speed

7) Output status indicators

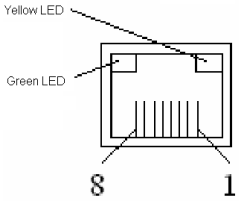
Red indicators are used for monitoring internal on/off status of outputs. An indicator on means its corresponding output is on. You can simply determine output circuits are good or bad by output indicators.

8) 24V power supply



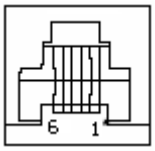
For the convenience of users, NK1 series PLCs provide a DC24 power supply which current capacity is 300mA for external sensors. When the system needs DC24V power supply, this DC24V power supply can be used to power the sensors, the prerequisite is ensuring that the total current consumption is less than 300mA.

9) Ethernet communication port



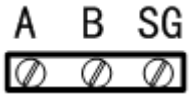
There is a RJ45 ethernet communication port in the top left side of NK1 CPU. The port meets the specifications of IEEE802.3 and can select 10M or 100M speed depending on the connecting network. It supports ECOM UDP, MODBUS TCP/IP protocol and can be used to connect KPPSOFT or other communication devices which support Ethernet MODBUS TCP/IP. Note: there is a RJ12 RS232C serial communication port instead of an Ethernet communication port in the same location of 20-point NK1 CPU. Its specifications are same with the other RS232C port.

10) RS232C communication port



There is a RJ12 RS232C serial communication port in the lower left side of NK1 CPU. It is both a programming port that can connect KPPSPFT and a serial communication port. It supports K (slave), CCM2 (slave), MODBUS RTU (master/slave), non-sequence (master/slave) protocol, etc.

11) RS485 communication port



RS485 port is a 3-terminal block. It supports protocol of K (slave), CCM2 (slave), MODBUS RTU (master/slave), non-sequence (master/slave), etc.

12) USB port

USB port is a mini D-type connector. It is a standard MINI USB port, supports standard OTG protocol. Communication speed meets the standards of USB2.0 FULL SPEED. You can write the program and system parameters to NK1 CPU via this USB port.

13) Mode switch

There are two operating mode: TERM-RUN (program run) and TERM-STOP (program stop). You can change the operating mode by mode switch or KPPSOFT.

Note: because no matter which position the mode switch is in, the operating mode of NK1 CPU can be changed by KPPSOFT, so it is possible that mode switch is in TERM-RUN and the program is not running, or mode switch is in TERM-STOP and the program is running. In this case, you can change the operating mode by KPPSOFT or toggling mode switch. For example, when the mode switch is in "RUN" position and the user program is in TERM-STOP mode, if you want to run the program, you can toggle the mode switch in the follwing order: first toggle mode switch in "STOP" position, and then

toggle mode switch in “RUN”position. If the program still is not in TERM-RUN mode, it shows than there is no user program in the PLC or the system has fatal errors.

When NK1 CPU is power up, if there is user program in CPU or system has no fatal error, the system will set the operating mode of NK1 CPU according to the follwing table.

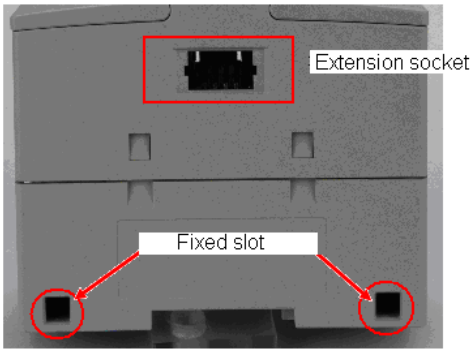
Mode swith position	Operating mode	Description
“STOP”	TERM-STOP	Stop the user program
“RUN”	TERM-RUN	Run the user program

14) Function board installation location

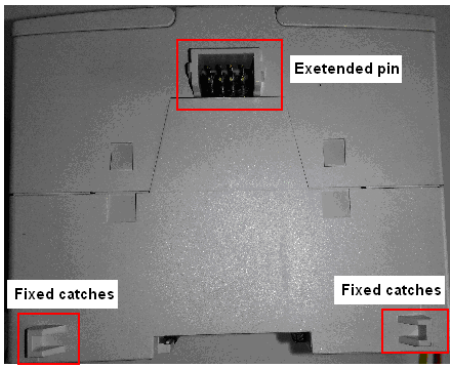
There is a function board installation location in 40-point and 64-point NK1 CPU, you can install a function board here for the following purpose: increasing a battry for retentive calendar and clock data, or increasing a small amount of I/O points, or using non-isolated AD/DA loop, or increasing a serial communication port, and so on.

15) Expansion unit interface

NK1 CPUs can take up to fourteen expansion units depending on the application. Expansion units are installed sequentially on the right side of the CPU.



Right side view of NK1 CPU



Left side view of the expansion unit

When you use the expansion cable, you can connect the expansion cable to the unit just like you do with the expansion unit.

Please see the photo on the right for details.

Connection box on the lower side of expansion cable Connection box on the upper side of expansion cable

3-2 I/O component

3-2-1 I/O system component

Each system of NK1 PLC requires a CPU module, and the system can be expanded by adding expansion units and a function board. You also require KPPSOFT for programming and debugging. When using different CPU the maximum I/O configuration supported by the system is listed in the following table.

CPU	NK1-CPU20	NK1-CPU40
I/O number of CPU	20	40
Support maximum expansion units	14	14
Number of I/O per expansion unit	8/16/32	8/16/32
Function board installation	Not supported	Supported *1
Maximum I/O of function board	-------	8
Maximum I/O	468	496
Number of high-speed counters	4 channels single phase or 2 channels A/B phase counters	5 channels single phase or 3 channels A/B phase counters
Maximum pulse catch, external interrupt	4	8
Maximum pulse output	2 channels or 4 points	2 channels or 4 points
PID loops	16	16
Ethernet communication port on the CPU	None	1
RS-232C communication port on the CPU	2	1
RS-485 communication port on the CPU	1	1
USB communication port on the CPU	1	1
Serial communication port on the functional board	Not supported	1（RS-232C or RS-485）

*1 Note: you can only install one function board. If you have installed an I/O function board, you can not install a serial port function board.

3-3 Identify I/O and assign I/O definition number

3-3-1 Identify I/O

In order to identify inputs and outputs f NK1 series PLC effectively, we give the distinguishing mark "I" for inputs and "Q" for outputs. We identify I/O by "distinguishing mark + digital", "distinguishing mark+digital" is called I/O definition number.

Each input / output point has a corresponding internal memory in the PLC for storing the point status. We call these internal memory function memory. In addition to the input / output function memory, there is other function memory, such as: control relays, timers, counters, special relays, data registers, and so on. In order to identify the various funcion memory types, we need to number them. All memory locations and resources are numbered octal starting from "0". This means that the octal system does not contain any numbers with the digits 8 or 9.

3-3-2 Assign I/O definition number

I/O definition number of NK1 PLC is assigned in the following principle: each unit specifies the starting I/O definition number, inputs and outputs are assigned automatically. NK1 CPU will assign definition number for I/O of each unit at powerup based on the CPU, expansion units and function board. The assigning principle is: inputs and outputs of CPU are automatically assigned from I0/Q0; inputs and outputs of function boards are automatically assigned from I100/Q100; for expansion units, CPU assigns the specified I/O definition number according to different installation location. Each expansion unit assigns I/O definition number for their I/O according to the specified starting address. In addition, you will use some special setup registers when you are using analog expansion units and high-speed counter expansion units. The registers will be different depending on the location of the expansion units.

The following table lists I/O definition number of each unit.

Unit type	Definition number of input	Definition number of output	remark
CPU	I000~I077	Q000~Q077	Definition number (I, Q) which be not assigned to inputs and outputs can be used as internal control relays in the program.
Function board	I100~I177	Q100~Q177	
The first expansion unit	I200~I277	Q200~Q277	
The second expansion unit	I300~I377	Q300~Q377	
The third expansion unit	I400~I477	Q400~Q477	
The fourth expansion unit	I500~I577	Q500~Q577	
The fifth expansion unit	I600~I677	Q600~Q677	
The sixth expansion unit	I700~I777	Q700~Q777	
The seventh expansion unit	I1000~I1077	Q1000~Q1077	
The eighth expansion unit	I1100~I1177	Q1100~Q1177	
The ninth expansion unit	I1200~I1277	Q1200~Q1277	
The tenth expansion unit	I1300~I1377	Q1300~Q1377	
The eleventh expansion unit	I1400~I1477	Q1400~Q1477	
The twelfth expansion unit	I1500~I1577	Q1500~Q1577	
The thirteenth expansion unit	I1600~I1677	Q1600~Q1677	
The fourteenth expansion unit	I1700~I1777	Q1700~Q1777	

Note: the above assignment of I/O definition number is only effective for ordinary I/O expansion (function board), special expansion units (function boards) such as analog expansion units and high-speed counter units don't need I/O definition number, so the I/O definition number according to the

module installation position can be used as internal relays in program.

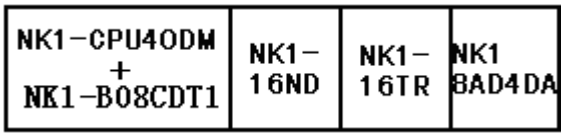
The flowing table lists the I/O definition number that each model of NK1 CPU assigns to local inputs and outputs.

CPU	Local inputs	Definition number	Local outputs	Definition number
NK1-CPU20*	12	I0~I13	8	Q0~Q7
NK1-CPU40*	24	I0~I27	16	Q0~Q17

3-3-3 Example of I/O definition number assigning

In the following example, the three NK1 systems are all composed by NK1-CPU40DM+NK1-16ND+NK1-16TR+NK1-8AD4DA. However, due to the different location of each unit is installed, the allocation of I/O definition number is different. The following examples are three kinds of assignment of I/O definition number.

The first installation sequence:



The I/O definition number assignment of each unit:

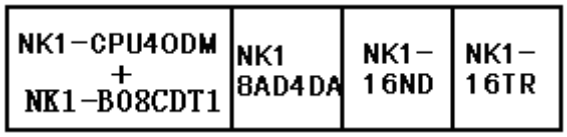
Installation sequence	CPU	Function board	The first expansion unit	The secong expansion unit	The third expansion unit
Module	NK1-CPU40DM	NK1-B08CDT1	NK1-16ND	NK1-16TR	NK1-8AD4DA
Input definition number	I0~I27	I100~I104	I200~I217	--------	--------
Output definition number	Q0~Q17	Q100~Q104	----------	Q300~Q317	----------

The second installation sequence:

The I/O definition number assignment of each unit:

Installation sequence	CPU	Function board	The first expansion unit	The secong expansion unit	The third expansion unit
Module	NK1-CPU40DM	NK1-B08CDT1	NK1-16ND	NK1-8AD4DA	NK1-16TR
Input definition number	I0~I27	I100~I104	I200~I217	--------	--------
Output definition number	Q0~Q17	Q100~Q104	----------	----------	Q400~Q417

The third installation sequence:



The I/O definition number assignment of each unit:

Installation sequence	CPU	Function board	The first expansion unit	The secong expansion unit	The third expansion unit
Module	NK1-CPU40DM	NK1-B08CDT1	NK1-8AD4DA	NK1-16ND	NK1-16TR
Input definition number	I0~I27	I100~I104	--------	I300~I317	--------
Output definition number	Q0~Q17	Q100~Q104	----------	----------	Q400~Q417

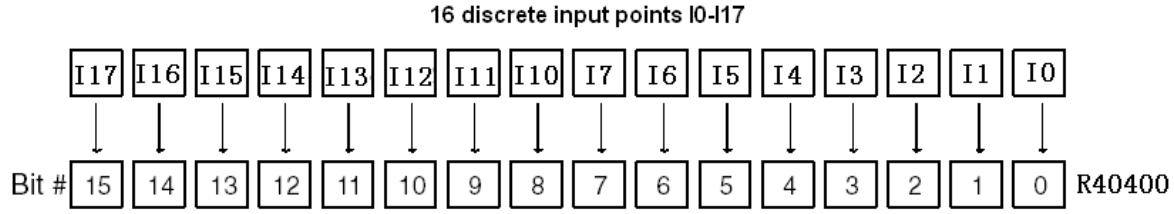
3-4 Function memory

With any PLC system, you generally have many different types of information to process. This includes input device status, output device status, various timing elements, parts counts, etc. It is important to understand how the system represents and stores the various types of data. For example, you need to know how the system identifies input points, output points, data words, etc. For NK1 PLC, the various information processed by the system are stord in internal memory, we called the internal memory function memory. All function memory locations are numbered in Octal (board 8).

Two types of memory are in NK1 PLC, discrete and word memory. Discrete memory is one bit that can be either a 1 or a 0. Word memory is referred to as R-memory (variable) and is a 16-bit location normally used to manipulate data/numbers, store data/numbers, etc.

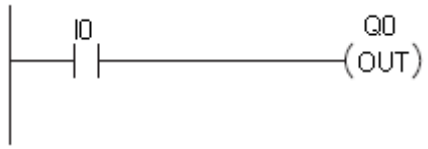
Some information is automatically stored in R-memory. For example, the timer current values are stored in R-memory.

The discrete memory area is for inputs, outputs, control relays, special relays, stages, timer status bits and counter status bits. However, you can also access the bit data types as a R-memory word. Each R-memory location contains 16 consecutive discrete locations. For example, the following diagram shows how the I input points are mapped into R-memory locations.



Input Points (I Data Type)

The discrete input points are noted by an I data type. In this example, the output point Q0 will be turned on when input I0 energizes.

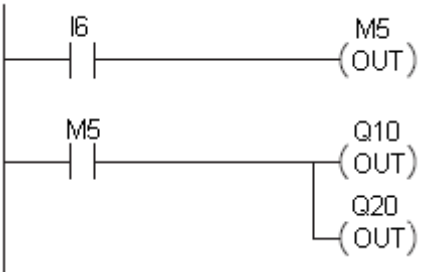


Output Points (Q Data Type)

The discrete output points are noted by a Q data type. In this example, output point Q1 will be turned on when input I1 energizes.

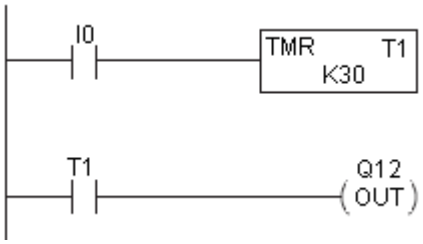
Control Relays (M Data Type)

Control relays are discrete bits normally used to control the user program. The control relays do not represent a real world device, that is, they cannot be physically tied to switches, output coils, etc. Because of this, control relays can be programmed as discrete inputs or discrete outputs. These locations are used in programming the discrete memory locations (M) or the corresponding word location which contains 16 consecutive discrete locations. In this example, memory location M5 will energize when input I6 turns on. The second rung shows a simple example of how to use a control relay as an input.



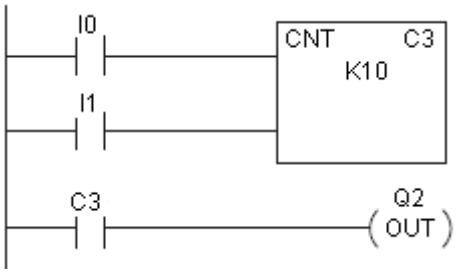
Timers and Timer Status Bits (T Data Type)

Timer status bits reflect the relationship between the current value and the preset value of a specified timer. The timer status bit will be on when the current value is equal or greater than the preset value of a corresponding timer. When input I0 turns on, timer T1 will start. When the timer reaches the preset of 3 seconds (K of 30) timer status contact T1 turns on. When T1 turns on, output Q12 turns on. Turning off I0 resets the timer.



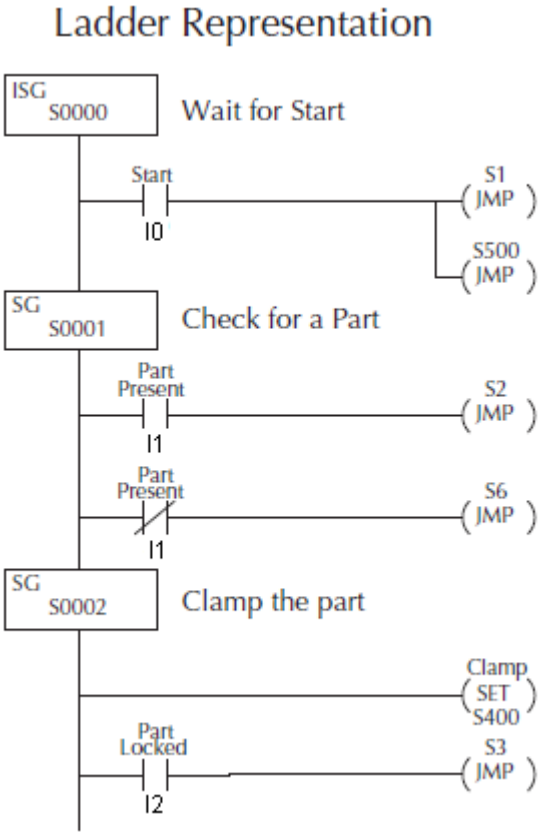
Counters and Counter Status Bits (C Data type)

Counter status bits that reflect the relationship between the current value and the preset value of a specified counter. The counter status bit will be on when the current value is equal to or greater than the preset value of a corresponding counter. Each time contact I0 transitions from off to on, the counter increments by one. (If I1 comes on, the counter is reset to zero.) When the counter reaches the preset of 10 counts (K of 10) counter status contact C3 turns on. When C3 turns on, output Q2 turns on.



Stages (S Data type)

Stages are used in RLLPLUS programs to create a structured program, similar to a flowchart. Each program stage denotes a program segment. When the program segment, or stage, is active, the logic within that segment is executed. If the stage is off, or inactive, the logic is not executed and the CPU skips to the next active stage. Each stage also has a discrete status bit that can be used as an input to indicate whether the stage is active or inactive. If the stage is active, then the status bit is on. If the stage is inactive, then the status bit is off. This status bit can also be turned on or off by other instructions, such as the SET or RESET instructions. This allows you to easily control stages throughout the program.

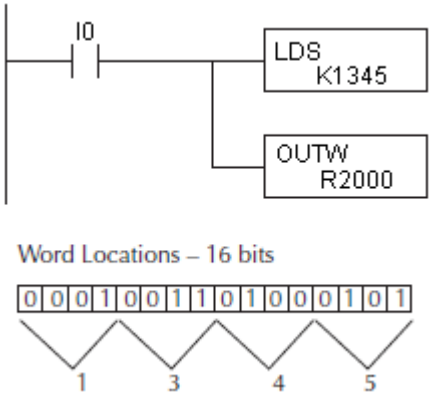


Special Relays (SP Data Type)

Special relays are discrete memory locations with predefined functionality. There are many different types of special relays. For example, some aid in program development, others provide system operating status information, etc.

Word Memory (R Data Type)

Word memory is referred to as R-memory (variable) and is a 16-bit location normally used to manipulate data/numbers, store data/numbers, etc. Some information is automatically stored in R-memory. For example, the timer current values are stored in R-memory. The example shows how a four-digit BCD constant is loaded into the accumulator and then stored in a R-memory location.



Remote input (GI)/ output (GQ)

GI and GQ are used as remote inputs and outputs, are mainly used to control the status of remote I/O. GI and GQ can be used as intermediate coils when they are not using in remote I/O system. NK1 series PLCs don't currently support remote I/O system, so GI and GQ are used as intermediate coils.

3-4-1 Memory map

Memory type	Memory name	Memory reference (octal)	Qty. decimal	Remark
I	Input points	I 0～I 1777	1024	
Q	Output points	Q 0～Q 1777	1024	
M	Control relays	M 0～M 3777	2048	Can be set to Retentive memory
S	Stages	S 0～S 1777	1024	Can be set to Retentive memory
T	Timers	T 0～T 777	512	Can be set to Retentive memory
C	counters	C 0～C 777	512	Can be set to Retentive memory
GI	Link relays	GI 0～GI 3777	2048	NK1 series PLCs don't support remote I/O system currently, GI and GQ are used as internal coils.
GQ		GQ 0～GQ 3777	2048	
SP	Special relays	SP 0～SP 1777	1024	System-defined flags such as clock, bit flags, and so on.
R/P	Timer current values	R 0～R 777	512 words	A timer corresponds to a word, can be set to Retentive memory
		P 0～P 777		
	Counter current values	R 1000～R 1777	512 words	A counter corresponds to a word, can be set to Retentive memory
		P 1000～P 1777		
R	Special registers	R 7400～R 7777	256 words	System-defined special registers
		P 7400～P 7777		
	Data registers	R2000～R 7377	2816 words	Can be set to retentive memory
		P 2000～P 7377		
		R 10000～R 36777	11776 words	Can be set to retentive memory
		P 10000～P 36777		
	Special registers	R 37000～R 37777	512 words	Be used to special expansion units, can be set to retentive memory
		P 37000～P 37777		

Continued

R	Link coil registsters	R 40000～R 40377	256 words	Registers corresponding to Link input
	Input registers	R 40400～R 40477	64 words	Registers corresponding to input points
	Output registers	R 40500～R 40577	64 words	Registers corresponding to output points
	Control relays registers	R 40600～R 40777	128 words	Registers corresponding to control relays
	Stage registers	R 41000～R 41077	64 words	Registers corresponding to stages
	Timer registers	R 41100～R 41117	16 words	Registers corresponding to timers
	Counter registers	R 41140～R 41157	16 words	Registers corresponding to counters
	Special coil registers	R 41200～R 41237	32 words	Registers corresponding to special relays
LBL	Interrupts	ILBL O 0～20	17	
	Subroutines	CLBL K 0～FFFF	256	
	GOTO blocks	GLBL K 0～FFFF	256	
	Data blocks	DLBL K 0～FFFF	256	

3-4-2 Special relays

Special relays are pre-defined internal coils, can only be used as a condition contact. The following table lists these special relays:

Address	Funciton	Descriptions
Startup and Real-Time Relays		
SP000	First scan	On for the first scan after a power cycle or program to run transition only. The relay is reset to off on the second scan. It is useful where a function needs to be performed only on program startup.
SP001	Always ON	Provides a contact to insure an instruction is executed every scan.
SP002	Always OFF	Provides a contact that is always off.
SP003	1 minute clock	On for 30 seconds and off for 30 seconds.
SP004	1 second clock	On for 0.5 second and off for 0.5 second.
SP005	100 ms clock	On for 50 ms. and off for 50 ms.
SP006	50 ms clock	On for 25 ms. and off for 25 ms.
SP007	Alternate scan	On every other scan.
CPU Status Relays		
SP010	undefined（Always OFF）	undefined
SP011	undefined（Always OFF）	undefined
SP012	TERM RUN mode	On when the CPU is in the RUN mode
SP013	TERM RUN mode	On when the CPU is in the TEST RUN mode
SP014	undefined（Always OFF）	undefined
SP015	TERM STOP mode	On when the CPU is in the TEST STOP mode
SP016	TERM STOP mode	On when the CPU is in the STOP mode
SP017	undefined（Always OFF）	Undefined
SP020	Forced stop mode	On when the STOP instruction is executed.
SP021	undefined（Always OFF）	undefined
SP022	Interrupt enabled relay	On when interrupts have been enabled using INE instruction.
SP025	CPU battery disabled	On when the CPU battery is disabled.
SP030	SP_DATA_SETTING_ERR	On when the Ethernet data is setting error.
SP031	SP_NO_PROGRAM	On when there is no user program.
SP032	SP_IP_READY	On when IP address is ready, instructions Ethernet-related need to check the flag for operation.
SP033	SP_TCP_CONNECTION_FULL	On when the number of communication connections that MRX/MWX instructions may use overflow.
System Monitoring Relays		
SP040	Critical error	On when a critical error such as I/O communication loss has occurred.
SP041	Warning	On when a non-critical error such as a low battery has occurred.
SP043	Battery low/dead	On when the CPU battery voltage is low or dead. Note: The CPU must have a battery installed.
SP044	Program memory error	On when a memory error such as a memory parity error has occurred.
SP045	I/O error	On when an I/O error occurs. For example, an I/O module is withdrawn from the base, or an I/O bus error is detected.
SP046	Communications error	On when a communications error has occured on any of the CPU ports.
SP047	I/O configuration error	I/ On if an I/O configuration error has occured. The CPU power-up I/O

		configuration check must be enabled before this relay will be functional.
SP050	undefined（Always OFF）	undefined
SP051	Watch Dog timeout	On if the CPU Watch Dog timer times out.
SP053	Math pointer error	On if there is math execution error
SP054	Communication port parameters error	On if there is parameter setting error of any one of communication port
Accumulator Status Relays		
SP060	Value less than	On when the accumulator value is less than the instruction value.
SP061	Value equal to	On when the accumulator value is equal to the instruction value.
SP062	Greater than	On when the accumulator value is greater than the instruction value.
SP063	Zero	On when the result of the instruction is zero (in the accumulator).
SP064	Half borrow	On when the 16-bit subtraction instruction results in a borrow.
SP065	Borrow	On when the 32-bit subtraction instruction results in a borrow.
SP066	Half carry	On when the 16-bit addition instruction results in a carry.
SP067	Carry	On when the 32-bit addition instruction results in a carry.
SP070	Sign	On anytime the value in the accumulator is negative.
SP071	Invalid octal number	On when an Invalid octal number was entered. This also occurs when the R-memory specified by a pointer (P) is not valid.
SP072	Floating Point	On when the numerical value in the accumulator is not a floating point number.
SP073	Overflow 1	On if overflow occurs in the accumulator when a signed addition or subtraction results in an incorrect sign bit.
SP074	Overflow 2	On when a floating point math operation results in an overflow error.
SP075	Data error	On when a BCD instruction is executed and a NON-BCD number was encountered.
SP076	Load zero	On when any instruction loads a value of zero into the accumulator.
Pulse catch status relays		
SP100	I10=ON	When using pulse catch function, On when I0=ON
SP101	I11=ON	When using pulse catch function, On when I1=ON
SP102	I12=ON	When using pulse catch function, On when I2=ON
SP103	I13=ON	When using pulse catch function, On when I3=ON
SP104	I14=ON	When using pulse catch function, On when I4=ON
SP105	I15=ON	When using pulse catch function, On when I5=ON
SP106	I16=ON	When using pulse catch function, On when I6=ON
SP107	I17=ON	When using pulse catch function, On when I7=ON
Communications Monitoring Relays *note 1		
SP110	Port0 communication	On when port 0 is communicating with another device and port 0 is master station.
SP111	Port0 comm error	On when port 0 has encountered a communication error and port 0 is master station.
SP112	Port1 communication	On when port 1 is communicating with another device and port 1 is master station.
SP113	Port1 comm error	On when port 1 has encountered a communication error and port 1 is master station.
SP114	Port2 communication	On when port 2 is communicating with another device and port 2 is master station.

SP115	Port2 comm error	On when port 2 has encountered a communication error and port 2 is master station.
SP116	Port3 communication	On when port 3 is communicating with another device and port 3 is master station.
SP117	Port3 comm error	On when port 3 has encountered a communication error and port 3 is master station.

Note 1:

The communication ports of NK1 PLC are numbered in this manual: RS-232C port built-in CPU is called PORT0 (referred to as P0); 3-wire RS-485 port is called PORT1 (referred to as P1); the port on the optional function board is called PORT2 (referred to as P2) (the second RS-232C port on 20-point I/O CPU is also called P2 port); Ethernet port is called PORT3 (referred to as P3). P0 is called programming port, other communication ports are called universal communication port.

Equal relays for high-speed counter		
SP540	Current = target value	On when the counter current value of 1# single-phase or two-phase channel equals the first preset value
SP541	Current = target value	On when the counter current value of 1# single-phase or two-phase channel equals the second preset value
SP542	Current = target value	On when the counter current value of 1# single-phase or two-phase channel equals the third preset value
SP543	Current = target value	On when the counter current value of 1# single-phase or two-phase channel equals the fourth preset value
SP544	Current = target value	On when the counter current value of 2# single-phase channel equals the first preset value
SP545	Current = target value	On when the counter current value of 2# single-phase channel equals the second preset value
SP546	Current = target value	On when the counter current value of 2# single-phase channel equals the third preset value
SP547	Current = target value	On when the counter current value of 2# single-phase channel equals the fourth preset value
SP550	Current = target value	On when the counter current value of 3# single-phase channel or 2# two-phase channel equals the first preset value
SP551	Current = target value	On when the counter current value of 3# single-phase channel or 2# two-phase channel equals the second preset value
SP552	Current = target value	On when the counter current value of 3# single-phase channel or 2# two-phase channel equals the third preset value
SP553	Current = target value	On when the counter current value of 3# single-phase channel or 2# two-phase channel equals the fourth preset value
SP554	Current = target value	On when the counter current value of 4# single-phase channel equals the first preset value
SP555	Current = target value	On when the counter current value of 4# single-phase channel equals the second preset value
SP556	Current = target value	On when the counter current value of 4# single-phase channel equals the third preset value
SP557	Current = target value	On when the counter current value of 4# single-phase channel equals the fourth preset value

Continued

SP560	Current = target value	On when the counter current value of 5# single-phase channel or 3# two-phase channel equals the first preset value
SP561	Current = target value	On when the counter current value of 5# single-phase channel or 3# two-phase channel equals the second preset value
SP562	Current = target value	On when the counter current value of 5# single-phase channel or 3# two-phase channel equals the third preset value
SP563	Current = target value	On when the counter current value of 5# single-phase channel or 3# two-phase channel equals the fourth preset value

3-4-3 Special control relays

Some control relays are pre-defined specific uses when using some special functions. These pre-defined uses control relays can not be used for any other purpose.

When using pulse output function, control relays listed in the following are pre-defined special uses.

Special features	Special M
1# channel pulse output（Q0/Q1）	M1700~M1717
2# channel pulse output（Q2/Q3）	M1720~M1737

3-4-4 Special registers

Special registers are pre-defined use by system. The range of system registers are R7400-R7777, R37000-R37777. Some reserve registers may be difined special use in the future, so do not use it for other purpose in the application.

System registers	Function	Default value
R7400 ↓ R7417	Mode setting registers for I0~I17, each register corresponds to an input point.	1007
R7420 ↓ R7427	Input filter time setting registers for I20~I27	
R7430 ↓ R7437	Undefined, reserved	
R7440	The first register location to store the preset value of 1# two-phase counter or 1# single-phase counter.	798 （R3630）
R7441	The first register location to store the preset value of 2# single-phase counter.	7A0（R3640）
R7442	The first register location to store the preset value of 2# two-phase counter or 3# single-phase counter.	7A8（R3650）
R7443	The first register location to store the preset value of 4# single-phase counter.	7B0（R3660）
R7444	The first register location to store the preset value of 3# two-phase counter or 5# single-phase counter.	7B8（R3670）
R7445	Flag that forces to write the basic parameters of pulse output	
R7446	Undefined, reserved	
R7447	Undefined, reserved	
R7500 ↓ R7547	Special registers for pulse output function. Please refer to chapter 5 high-speed input and pulse output features for details.	

Continued

R7550 ↓ R7577	Undefined, reserved	
R7600	Most significant bit of IP address（BCD）	
R7601	higher bit of IP address（BCD）	
R7602	Lower bit of IP address（BCD）	
R7603	The least significant bit of IP address (BCD）	
R7604	Most significant bit of MASK address（BCD）	
R7605	higher bit of MASK address（BCD）	
R7606	Lower bit of MASK address（BCD）	
R7607	The least significant bit of MASK address（BCD）	
R7610	Most significant bit of gateway IP address（BCD）	
R7611	higher bit of gateway IP address（BCD）	
R7612	Lower bit of gateway IP address（BCD）	
R7613	The least significant bit of gateway IP address (BCD）	
R7614 ↓ R7617	Undefined, reserved	
R7620	Sets the R-memory location that contains the value.	
R7621	Sets the R-memory location that contains the message.	
R7622	Sets the total number (1 – 16) of R-memory locations to be displayed.	
R7623	Sets the R-memory location that contains the numbers to be displayed.	
R7624	Sets the R-memory location that contains the character code to be displayed.	
R7625	Contains the function number that can be assigned to each key.	
R7626	Sets the power up mode.	
R7627	Change preset value password.	0000
R7630	Undefined, reserved	
R7631	Undefined, reserved	
R7632	Undefined, reserved	
R7633	Set BIN operation sign and battery mode Bit15 = 0: unsigned BIN operation（default value） Bit15 = 1: signed BIN operation Bit12 = 0: no battry mode（default value） Bit12 = 1: batty mode	0000
R7634	Set timed interrupt time (0~9999ms) (0:timed interrupt is not enabled)	0000
R7640	Stores an octal number identifying the first user R-memory location for setting PID parameters	
R7641	Stores number of PID loops	
R7642	Stores PID setting error code	
R7643 ↓ R7647	Undefined, reserved	

Continued

R7650 ↓ R7661	Store communication parememters of port0.	
R7662 ↓ R7673	Store communication parememters of port1.	
R7674 ↓ R7705	Store communication parememters of port2.	
R7706 ↓ R7717	Store communication parememters of port3.	
R7720	Titled timer preset value pointer	
R7721	Title counter preset value pointer	
R7722	HiByte-Titled Timer preset block size, LoByte-Titled Counter preset block size	
R7723	Undefined, reserved	
R7724	Undefined, reserved	
R7725	Undefined, reserved	
R7726	Stores system program version number (higer bit) (XXXX: year)	
R7727	Stores system program version number (lower bit) (YYZZ YY: month, ZZ: date)	
R7730 ↓ R7744	Undefined, reserved	
R7745	Stores USB port status	
R7746	Undefined, reserved	
R7747	10ms timer, is incremented by 1 every 10 ms, in the range 00~99	
R7750	Undefined, reserved	
R7751	Undefined, reserved	
R7752	Stores number of base (0)/slot which I/O configuration is abnormal. Slot number: the number range of expansion unit is 1~14, the number of function board is 15. You need to reconfigure I/O after power off when the error occurred, or you can cancel “power up I/O configuration check”.	---
R7753	Stores expansion unit model code registered in the system	---
R7754	Stores the current actual expansion unit model code	---
R7755	Stores the error code when a fatal error occurs	---
R7756	Stores the error code when a non-fatal error occurs	---
R7757	Stores the error code when a warning error occurs	---
R7760 ↓ R7764	Undefined, reserved	---
R7765	Stores the number of program scan after RUN (HEX)	---
R7766	Clock (second) 0~59（BCD）	---
R7767	Clock (minute) 0~59（BCD）	---

Continued

R7770	Clock (hour) 0~23（24-hour）（HEX）	---
R7771	Calendar (week) 0: Sunday 1: Monday 2: Tuesday 3:Wednesday 4: Thursday 5: Friday 6: Saturday	---
R7772	Calendar (day) 1~31（BCD）	---
R7773	Calendar (month)1~12（BCD）	---
R7774	Calendar (year) 00~99（BCD）	---
R7775	Current scan time (ms)	---
R7776	The shortest scan time (ms)	---
R7777	The longest scan time (ms)	---
R37000 ↓ R3777	Stores expansion unit/function board parameters settings and user data. For details, please see document of the unit.	

3-4-5 FlashROM type special register

Normally, special registers of NK1 PLC are RAM type memory. If not set to be retentive, their contents will be lost after the power failure. However, many applications need some special registers to be retentive even if these special registers are not in retentive range. For the purpose of setting these registers to be retentive, these special registers are designed two data storage location: RAM and FlashROM. When writing data to these special registers by the PLC program, only the data of RAM are modified. When writing data to these special registers by KPPsoft tools, other computer host software, touch screens, etc, the data of RAM and FlashROM are both modified at the same time.

When power up, NK1 PLC will automatically read data from flashROM and write data to RAM type registers. Data in flashROM will not copy to RAM registers when operation mode is changed.

Note that if there are instructions that write data in special registers in the user program, after the user program is running, the data in RAM registers will be modified by the user program according to the operating results, but the data in flashROM can not be modified by the user program.

The following table lists all special registers supporting flashROM.

Special registers	Uses	Remark
R7400 ~ R7444	Special function setting of input points	
R7500 ~ R7544	Special function setting of output points	
R7620 ~ R7627	Setting of DV-1000	
R7633	Set BIN operation sign and battery mode	
R7634	Set timed interrupt time	
R7640 ~ R7642	Setting of PID	
R7650 ~ R7717	Setting of communication ports	
R7720 ~ R7722	Setting of DV-1000	
R37560 ~ R37777	Setting of expansion units	

3-4-6 Data registers pre-difined uses

The data registers in the following table have been defined special uses. Please don’t change the uses of them if not very necessary.

Registers	Uses	Data	Details
R3630～R3637	Store the preset value of 1# two-phase counter or 1# single-phase counter.	Multi-preset value	Set in R7440
R3640～R3647	Store the preset value of 2# single-phase counter.	Multi-preset value	Set in R7441
R3650～R3657	Store the preset value of 2# two-phase counter or 3# single-phase counter.	Multi-preset value	Set in R7442
R3660～R3667	Store the preset value of 4# single-phase counter.	Multi-preset value	Set in R7443
R3670～R3677	Store the preset value of 3# two-phase counter or 5# single-phase counter.	Multi-preset value	Set in R7444

3-5 User memory

User memory stores user program and system parameters defining basic system configuration.

Program memory		
	32Kwords	Store the user program, including the main program and subroutines.
System parameters memory		
	512words	Store system parameters

3-5-1 Program memory

Program memory is mainly used to store user-written program, user program consists of main program and subroutines.

0 address END
…
CLBL CEND
…
ILBL IEND
…
DLBL
ACON
NCON

Main program area:
Program between 0 address and END instruction, CPU cyclic scan and execute the main program.

Subroutine area:
Subroutine is a program segment in the case of a specific condition is met before the execution, it is behind END instruction and must be written in ladder language (you can not use stage instruction)

Note of the interrupt subroutine:

1） Do not use the following command in the interrupt subroutine:
SG, ISG, JMP, NJMP, CVJMP, BREQ, BSTART, BEND, PD, FOR, NEXT, TMR, HTMR, ATMR, AHTMR

2） Interrupt subroutine is defined by the function number O0~O4 (high-speed counter), O10~O17 (external interrupt), O20 (timed interrupt). O10~O14 can also be used as high-speed counter reset interrupt program (when I10~I14 are used as high-speed counter external reset terminal).

3）The length of the interrupt subroutine is not generally limited, but best not to be too long, so as not to affect the interrupt response speed. Instruction execution time of interrupt subroutine must be less than scan cycle of user program.

3-5-2 System parameters area

System parameters area is used to store the most basic system information including user program name, password, I/O configuration information, retentive parameters, CPU communication ports parameters, monitoring timer settings, and so on.

The main system parameters

User program name	…	The user program name is 8 or less letters, numbers
Password	…	Program password, 8 BCD digital
Indicate whether to check I/O module configuration	…	Indicate whether to check I/O module configuration on power-up.
I/O configuration information	…	Remember the current configuration of system
Retentive parameters	…	Set retentive range of memory
Common communication port setting parameters	…	Set parameters of 4 communication ports
Watchdog timer	…	Set watchdog timer

1 User program name

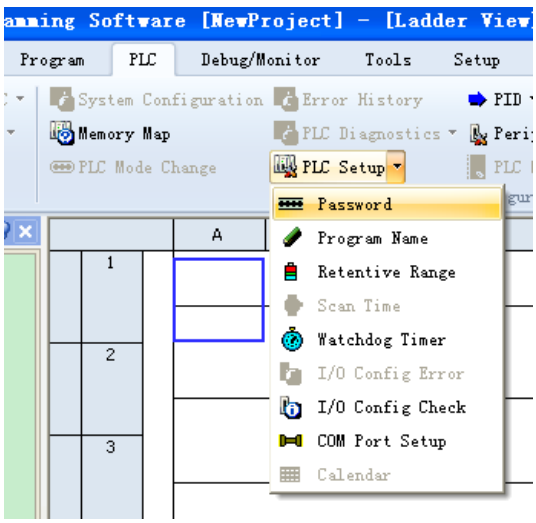
The user program name is 8 or less letters, numbers used to mark user program.

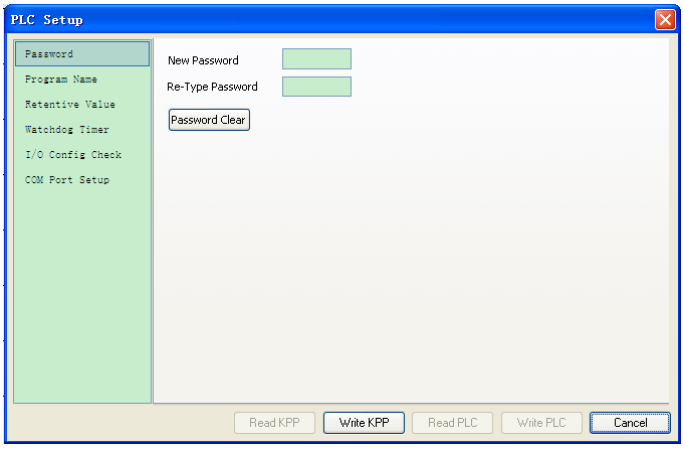
2 Password

The password is used to limit the operating functions of KPPSOFT or other software. The facory setting is eight zero, indicating no password.

1）Set password

You can set a password by KPPSoft. When you set the password, you need to establish the communication connection of KPPSoft and NK1 PLC, then you select the submenu [Password] under the menu [PLC Step], open the password input window (PLC setup window), enter a new password and re-enter the new password for confirmation, press [write PLC] key to write the password into PLC, the password has been writed. When the connection of KPPSoft and NK1 PLC is disconnected (or when KPPSoft software is closed), the password effect automatically.





2） Super password

If you forget your password and can not read the program in NK1 CPU, you can use super password introduced here to remove all the contents of NK1 CPU, and return to the initial state of the product at the factory.

Setting method: toggle mode switch to “STOP” position and confirm that NK1 CPU is in “STOP” state, then connect KPPSoft, enter 8 ‘C’ when you are prompted to enter a password, confirming the prompt will clear the password and the program and data in registers are cleared at the same time, the system parameters are restored to factory settings.

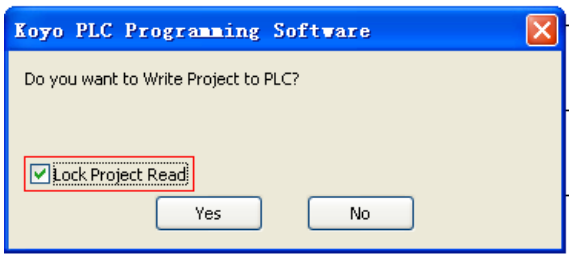
To use this feature, NK1 PLC must be in password valid state.

Note: after using super password to restore the factory settings, the program and data which has been cleared will not be restored.

3） Lock program

In some occasions we want that other people can monitor and modify data by KPPSoft, but do not want them to read the user program for the purpose of protecting software intellectual property.

In order to achieve the above functions, NK1 CPU provides a program lock protection functions: when you are downloading the program to PLC, a confirmation window will pop up, check [Lock Project Read] option in the window, the program which downloaded to the PLC will be locked and can not be read.



Note: due to the locked program can not be read again, be sure to save the project file before downloading it to the PLC with checking [Lock Project Read] option.

3 I/O configuration check indication and I/O configuration information

I/O configuration check indication indicates whether to check the I/O configuration information when power up.

NK1 factory default setting is not to check I/O configuration information, in the first power-up, NK1 will remember the current I/O configuration information.

Whether or not set to check I/O configuration information, NK1 always monitors each unit of the system during operation. Once problems are detected, NK1 will report the error immediately and even stop the operation of user program if the situation is serious.

4 retentive parameters

Retentive range	Control relay	M300~M377	M0000~M3777
	Data registers	R2000~R7377	R2000~R37777
	Timer	None	T000~T777（R0~R777）
	Counter	C000~C077	C000~C777（R1000~R1777）
	stage	None	S000~S1777

5 Universal communication port setting parameters

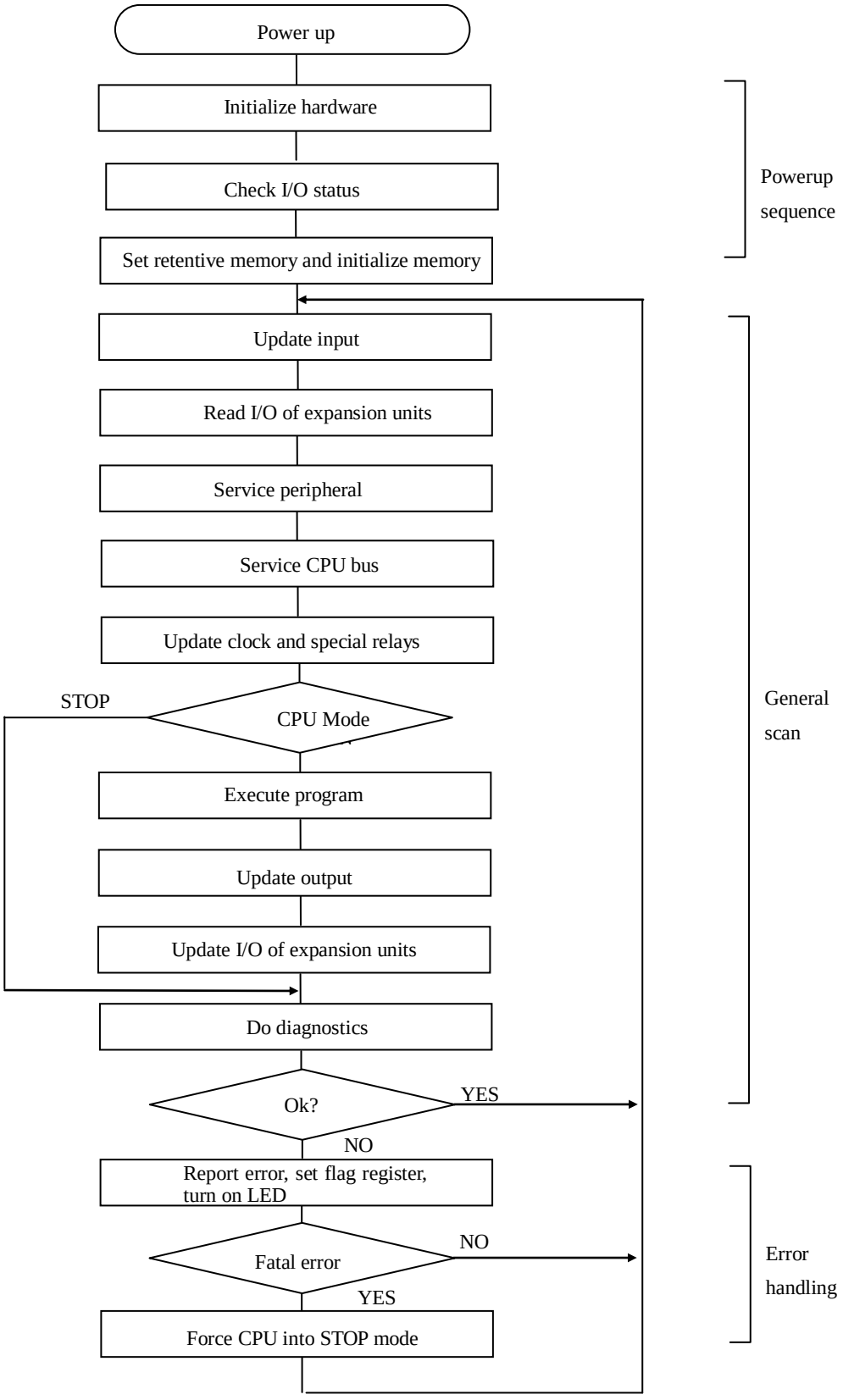
There are multiple communication ports on NK1 PLC. Communication parameters of these ports can be set by KPPSoft or special registers where store all the setting parameters.

6 Watchdog timer

Watchdog timer is used to monitor delay time of the dedicated processor during user program execution. It can detect the situation that user program enter an infinite loop resulting in loss of control.

3-6 CPU scan and input/output transfer

After power up, the CPU executes in the following order: first executes power-up sequence program, then exectutes general scanning process including I/O transmission, user program and error handling, etc.



3-7 CPU operation mode

NK1 PLC has five operating modes: TERM-RUN，TERM-STOP，TEST-RUN，TEST-HALT，TEST-STOP. You can select the operation mode by mode switch on NK1 CPU or KPPSoft programming software. TERM-RUN and TERM-STOP are official RUN operating mode, they can be selected by mode switch on NK1 CPU or KPPSoft programming software; TEST-RUN, TEST-HALT, TEST-STOP are debug operating mode, they can only be selectd by KPPSoft programming software.

The following table lists the actions of CPU in each mode:

Operation mode	CPU action
TERM-RUN	·Execute user program and update input and output cyclically. ·The mode can be changed by programming software, communication and mode switch.
TEST-RUN	·The mode is used for program debugging and trial operation. ·Execute user program and update input and output cyclically. ·You can set breakpoints execution program and can debug the program.
TEST-HALT	·The mode is a pause operating mode used for program debugging and trial operation. ·Program temporarily stops during execution. ·The status of each function memory and output point remains.
TEST-STOP	·The mode is a stop mode used for program debugging and trial operation. ·Program can be written and modified. ·CPU does not initialize function memory, you can continue to debug the program after it has been modified. ·Each function memory remains its original status, each output is OFF.
TERM-STOP	·The mode is used for program editing and parameters setting. ·The status of function memory is initialized except retentive range. ·All outputs are OFF. ·The mode can be changed by programming software, communication and mode switch.

1 Select mode by mode switch

1) Mode switch

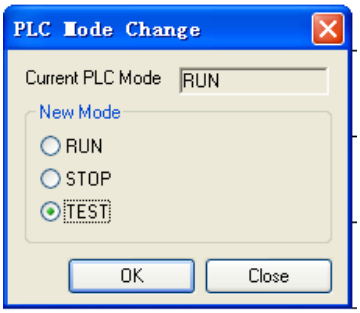
There are two positions: RUN (program run) and STOP (program stop). When the mode switch is toggled to STOP position, NK1 CPU enters TERM-STOP mode; when the mode switch is toggled to RUN position, NK1 CPU enters TERM-RUN mode if no errors are encountered.

Note:

1) No matter which position the mode switch is in, the operating mode of NK1 CPU can be changed by KPPSOFT software.
2) If the mode switch is in TERM-RUN and the program is not running, you can toggle the mode switch in the follwing order to run the program: first toggle mode switch in “STOP” position, then toggle mode switch in “RUN”position.
3) If the mode switch is in TERM-STOP and the program is running. You can toggle the mode switch in the follwing order to stop the program: first toggle mode switch in “RUN” position, then toggle mode switch in “STOP”position.

2 Select mode by KPPSoft programming software

You can select three operating mode: RUN, TEST and STOP by KPPSoft, as shown to the right. RUN is TERM-RUN mode; STOP is TERM-STOP mode; TEST includes three modes: TEST-STOP, TEST-HALT and TEST-RUN. In TEST mode, you can choose a variety of debugging operations: scan execution, 1 scan/n scans, single-step execution, execution to a specified position, etc.



Note: no matter which position the mode switch is in, the operating mode of NK1 CPU can be changed by KPPSOFT. In other words, the operating mode of NK1 CPU and the position of the mode switch is not one to one relationship.

3 Mode of NK1 PLC operation at powerup

NK1 PLC is in the operating mode according to the position of the mode switch when power up.

Position of mode switch	Internal status in NK1 CPU	NK1 mode at power-up
STOP	unrelated	TERM-STOP
RUN	No user program or system self-diagnostic error.	TERM-STOP
	The user program is correct, there is no self-diagnostic error.	TERM-RUN

4. The status of function memory in different modes

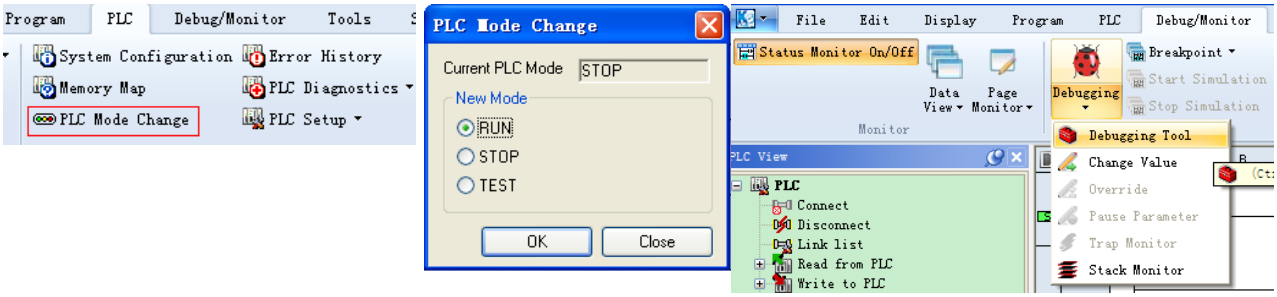
CPU operating mode	Inputs	Outputs	Other function memory and data registers
TERM-RUN	Update	The result of the program execution	The result of the program execution
TEST-RUN	Update	The result of the program execution	The result of the program execution
TEST-HALT	Update	The result of the program execution	The result of the program execution
TEST-STOP	Update	All OFF	The result of the program execution before stopping
TERM-STOP	Update	All OFF	OFF, 0（Data in retentive range will be retained）

3-8 NK1 CPU features

NK1 CPU is a complete PLC unit with all of PLC control functions, including online program debugging function, offline simulation run function, project file package generation/ download function, battry/ no battry mode, inputs software filter function, timed interrupt function, self-diagnostic function, history errors save/display function, communication function, high-speed I/O function, PID feature and other features.

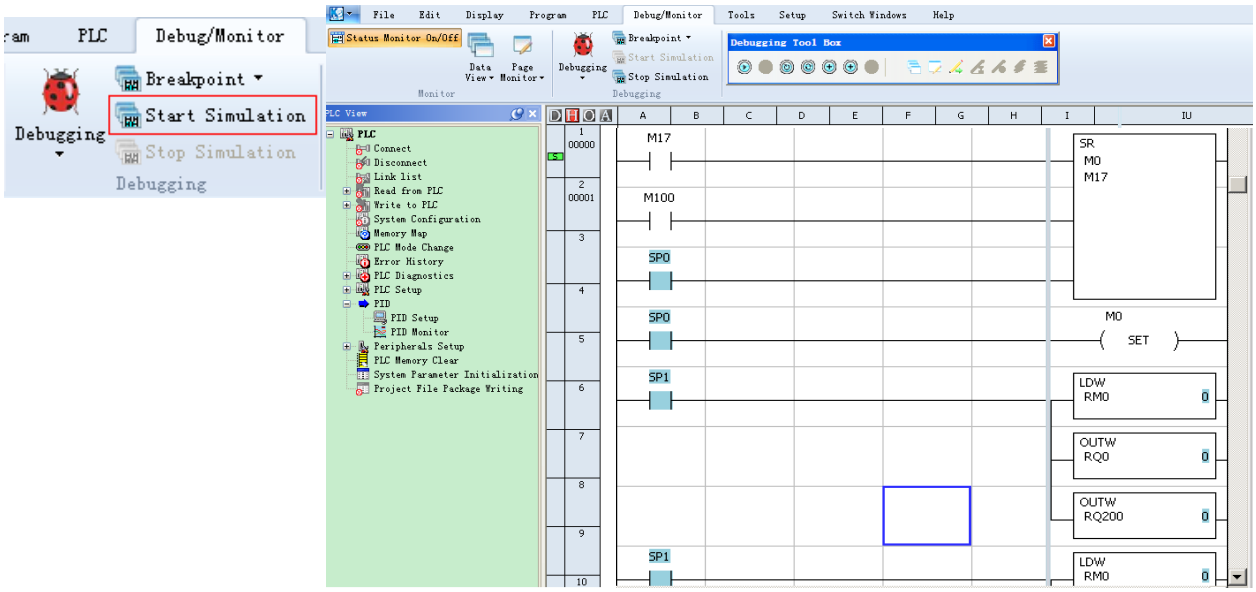
3-8-1 Online program debug

There are 3 online program debugging mode: TEST-RUN, TEST-HALT and TEST-STOP.
After the KPPSoft software is connected with NK1 PLC, open [PLC Mode Change] window under [PLC]. Note: NK1 PLC will enter TEST-STOP mode if it is in STOP mode before selecting TEST mode; NK1 PLC will enter TEST-RUN mode if it is in RUN mode before selecting TEST mode; TEST-HALT mode can not be directly selected.



3-8-2 Offline simulation run function

You must activate the feature before using it in KPPSoft software. Click [Debug/Monitor->Start Simulation] menu, offline simulation run function is activated.
Offline simulation run function can achieved all program debugging function achieved by online program debugging function.



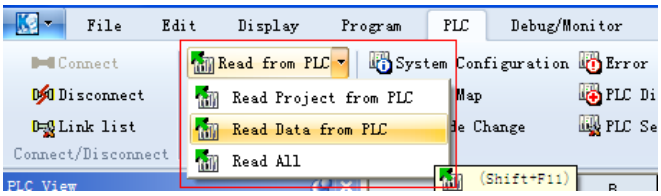
3-8-3 Generate and download project file package

You can write a user project file by KPPSoft software. The user project file can be downloaded to NK1 PLC and can also be stored on the disk as a backup (.KPP file). The .KPP file can be opened and edited again by KPPSoft software.

Sometimes we want to generate a project file package. The package can be downloaded to NK1 PLC by someone else, but it can not be opended and read by someone else. In this case we can use project files package generation and download function of KPPSoft.

You can generate a project file package (.KAD file) by KPPSoft. The project file package can only be downloaded to PLC instead of being opened and edited by KPPSoft; you can also rename the .KAD file and copy it to a USB flash disk, then you can connect the USB flash disk to the USB port on NK1 PLC by an OTG cable, the .KAD file can be downloaded to NK1 PLC automatically.

1. Generate project file package (.KAD file)

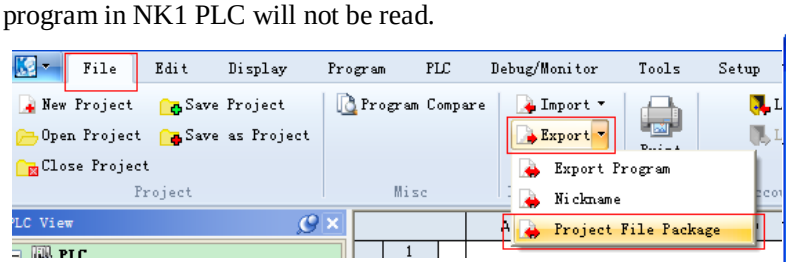


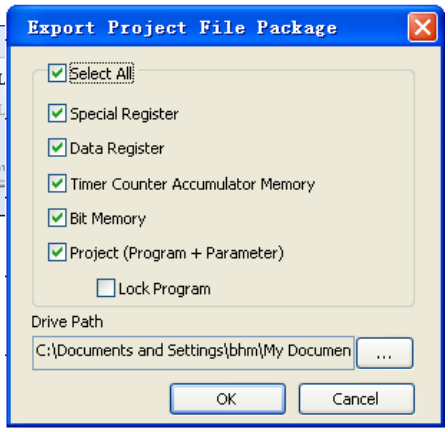
The project file package can include a user program and the information data reading from NK1 PLC.

You can read the user program from NK1 PLC through clicking ‘read from PLC’ in the right figure. Note that you can not read the user program if the program in NK1 PLC has been locked. But you can click ‘Read Data from PLC’ to read memory data and write them to the specified generated .KAD file.

Then, select ‘File→Export→Project File Package” menu, [Export Project File Package]window will pop up, you can select the contents to be wrote into the project fill package and select the name and storage path of the generation project fill package, click ‘OK’ button to close the window. KPPSoft software will automatically generate a project file package with a specified file name in the specified location.

Note: if you check ‘Lock Program’ item, after downloading the .KAD file to NK1 PLC, the user program in NK1 PLC will not be read.

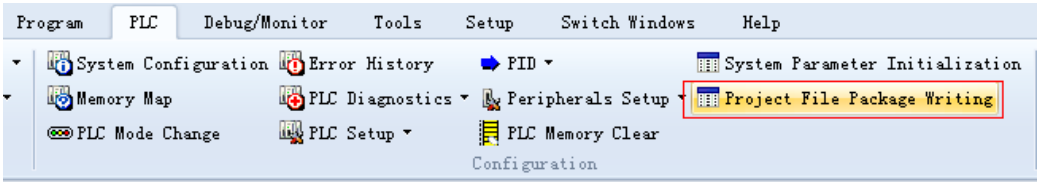




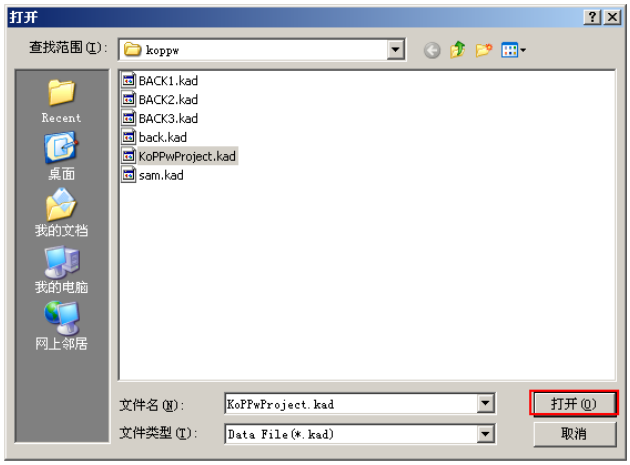
2. Download .KAD project file package by KPPSoft software

You can download the .KAD project file package to NK1 PLC by KPPSoft software.

When downloading, first communicate KPPSoft with NK1 PLC, then select 'PLC→ project file package writing' menu.



Pop 'file open' selection window, select the .KAD package, click on '打开' button, the selected .KAD package will be downloaded to NK1 PLC automatically.



Note that after the downloading, KPPSoft will no longer open and display the downloaded project file package.

3. Download ‘NK1PRJDT.KAD’ project file package by USB flash disk

There is a USB port on NK1 PLC. You can download the .KAD file to NK1 PLC through this USB port. Follow the following steps to download:

① Change the name of the.KAD package to ‘NK1PRJDT.KAD’, and then copy the package from disk to the root directory of USB flash disk;

② Toggle the mode switch to ‘STOP’ position and make sure that NK1 PLC is in stop mode;

③ Connect the USB flash disk to USB port by using a dedicated OTG cable, the package ‘NK1PRJDT.KAD’ will be downloaded to NK1 PLC automatically.

In the downloading process, the four LED indicators:

“PWR” on
“RUN” flash
“COM” off
“DIAG” flash

After a successful download, the four LED indicators:

“PWR” on
“RUN” on
“COM” flash (1 second)
“DIAG” flash (1 second)

④ Unplug the USB flash disk after a successful downloading. Re-power NK1 PLC, NK1 PLC will quit downloading mode.

Note: when downloading ‘K1PRJDT.KAD’ project file package by using USB flash disk, you should make sure that no files named ‘NK1SYSBK.BIN’ and ‘NK14064.BIN’ in the root directory of USB flash disk.

3-8-4 Set battry mode/ no battry mode

NK1 PLCs can be set to battry mode or no battry mode by setting a special register. In the standard configuration, NK1 PLCs have no back-up battry, and its corresponding special registers are set to no battry mode.

NK1 PLCs support calendar and clock function, but its long-term retaining need back-up battry. So, if you want to use calendar and clock function, you must install the backup battry and set the PLC to battry mode.

NK1 battry is not standard configuration. You need to purchase it separately.

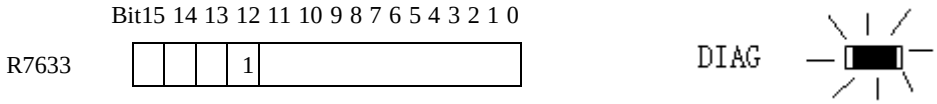
Battry type: RB-50 button battry

In addition, because the battry is installed in the function board (the empty board has no place to install the battry), so if you do not buy any function board, you need to buy a battery board.

Battry board type: NK1-BAT.

1、 Set battry mode

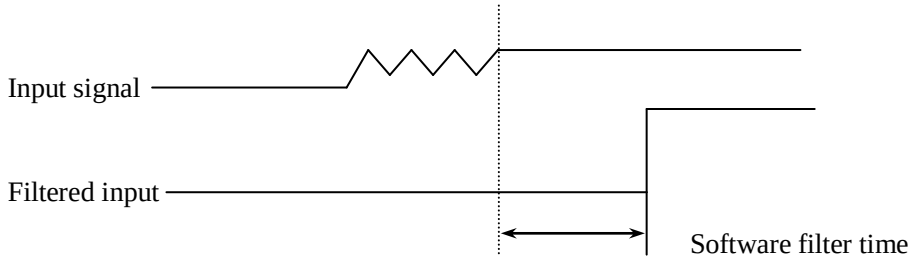
You can set NK1 PLC to battry mode by setting bit12 of R7633 ON. When NK1 PLC is in battry mode, if the battry is not installed or if the battry voltage is too low, NK1 PLC will detect ‘BATT ERROR’. (SP43=ON, DIAG LED flashes alone every one second.)



When bit12 of R7633 is set to OFF, NK1 PLC is in no battry mode. When NK1 PLC is in no battry mode, calendar and clock data is stored by a large-capacity capacitor. DIAG LED will never flash alone every one second when NK1 PLC is in no battry mode.

3-8-5 Input points software filter

Software filter is the feature to filter out the input signal glitch. It can filter out glitches in different widths by setting the filter time in the special register.



All input points on NK1 CPU can be set to software filtered input, the fliter time range is 0~99ms. Note that NK1 PLC inputs only support software filter when they are used as common inputs; they don't support software filter when they are used as high-speed counter inputs, pulse catch input, external interrupt input, etc.

There is a group of special registers corresponding to I0~I17 in NK1 PLC (20-point CPU: I0~I13). If you want to set an input as an ordinary input, you need write a data 'XX07' (hex) into the corresponding registers. '07' indicates that the input is an ordinary filtered input, 'XX' represents the filter time.

The following table lists the special registers corresponding to I0~I17 on 40-point NK1 CPU:

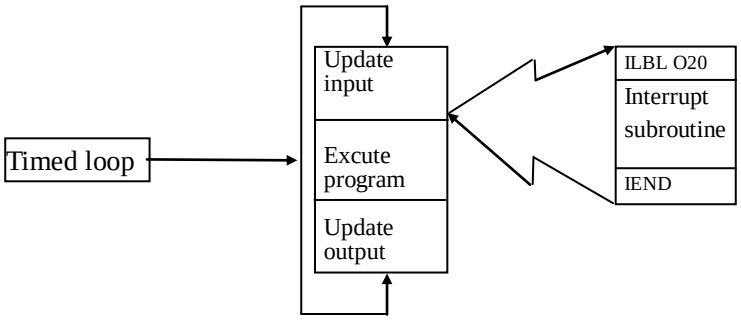
Inputs	Register	Setting content	Description
I000	R7400	XX07	XX: filter time, range:00~99
I001	R7401	XX07	XX: filter time, range:00~99
I002	R7402	XX07	XX: filter time, range:00~99
I003	R7403	XX07	XX: filter time, range:00~99
I004	R7404	XX07	XX: filter time, range:00~99
I005	R7405	XX07	XX: filter time, range:00~99
I006	R7406	XX07	XX: filter time, range:00~99
I007	R7407	XX07	XX: filter time, range:00~99
I010	R7410	XX07	XX: filter time, range:00~99
I011	R7411	XX07	XX: filter time, range:00~99
I012	R7412	XX07	XX: filter time, range:00~99
I013	R7413	XX07	XX: filter time, range:00~99
I014	R7414	XX07	XX: filter time, range:00~99
I015	R7415	XX07	XX: filter time, range:00~99
I016	R7416	XX07	XX: filter time, range:00~99
I017	R7417	XX07	XX: filter time, range:00~99
I20	R7420	00XX	XX: filter time, range:00~99
I21	R7421	00XX	XX: filter time, range:00~99
I22	R7422	00XX	XX: filter time, range:00~99
I23	R7423	00XX	XX: filter time, range:00~99
I24	R7424	00XX	XX: filter time, range:00~99
I25	R7425	00XX	XX: filter time, range:00~99
I26	R7426	00XX	XX: filter time, range:00~99
I27	R7427	00XX	XX: filter time, range:00~99

Note: 1) 20-point CPU only has 12 input points (I0~I13).

2) for detail setting of I0~I17,please see the Chapter 5.

3-8-6 Timed interrupt

Timed interrupt is also named timed scan. It is independent of the PLC scan, executes once every a certain time.



- When the current value of the interrupt timer reaches the set value, the interrupt subroutine is executed.
- No matter where the program is executing, the interrupt subroutine can interrupt it and executes the program behind ILBL instruction.
- After executing IEND instruction, interrupt subroutine is completed, the program execution will continue from where it was before the intettupt occurred.

1. Enable timed interrupt function and set the time

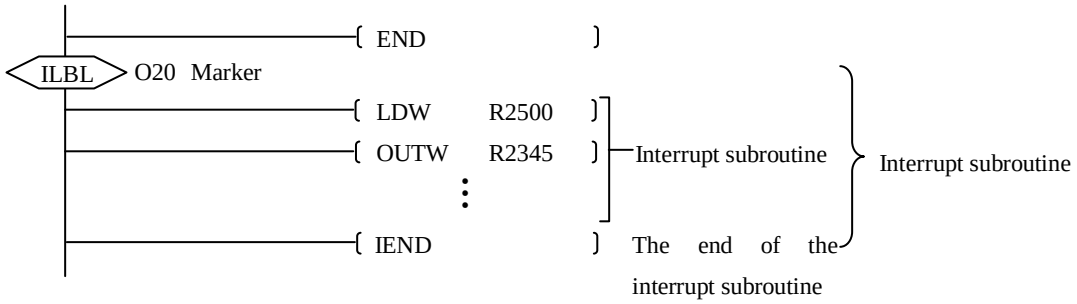
When using timed interrupt function, the time needs to be set previously in R7634, the range of time is 0~9999 (BCD), in milliseconds. 0 indicates timed interrupt is not enabled, other digital indicates timed interrupt is enabled and it is the interrupt time.

R7634: ××××（ms）

2. Write timed interrupt subroutine

Timed interrupt subroutine needs to be placed behind END instruction, its markers is O20.

The following example is a timed interrupt subroutine:

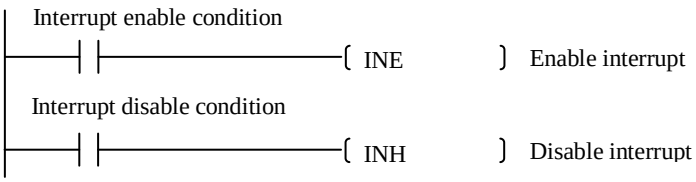


Note: the execution time of interrupt subroutine must be less than scan cycle time.

3. Enable and disable interrupt

Interrupt subroutine can't be execute until it is enabled.

When entering RUN mode, interrupt is disabled. In order to execute the interrupt subroutine, you must add the INE coil as enabling condition in the program. Once the coil has been energized, interrupts will be enabled until they are disabled by the INH coil.



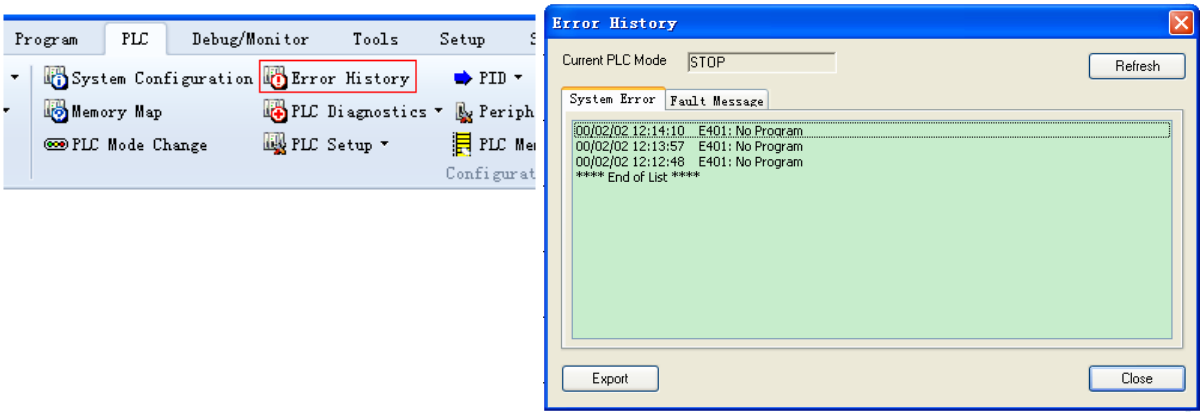
3-8-7 Self-diagnosis

NK1 PLCs have self-diagnosis feature. After power-up, NK1 PLC will diagnose its current state, the user program and the state of expansion modules every regular intervals. When errors are encountered, NK1 PLC will report the error codes and store them in the special registers. You can read the error codes by programming software, in addition, LEDs on the CPU panel or related special relays will be ON.
If fatal error is detected, CPU will stop scanning and enter into STOP mode (TERM-STOP).

Items	contents	Detection time	CPU mode	Special relays	Registers storing error codes
CPU error	CPU watchdog timer is abnormal	Always	Stop	—	R7755 （E003）
BATT error	DIAG LED flashes alone every on second (when in battery mode)	Always	Unchanged	SP43	R7757 （E041）
MEN error	Program memory parity error	Always	Stop	SP44	R7755 （E155）
	No user program	RUN beginning	Stop	SP44	R7755 （E401）
I/O error	I/O bus is abnormal	Always	Stop	SP45	R7755 （E250）
	I/O configuration is abnormal	Power up	Stop	SP47	R7755 （E252）
Communication error	Receive data error （When connect with KPPSoft via Ethernet）	Always	Unchanged	SP46	R7756 （E312）
	Receive address error	Always	Unchanged	SP46	R7756 （E313）
	Communication mode error	Always	Unchanged	SP46	R7756 （E316）
	SMTP communication error	Always	Unchanged	SP46	R7756 （E321）

3-8-8 History errors store and display

NK1 PLCs have the feature that they can automatically save the error messages detected by CPU.
You can open [PLC->Error History] window in KPPSoft to read the history error information, which is very useful for equipment repair and maintenance.



Chapter 4 NK1 PLC communication

4-1 NK1 CPU communication ports

NK1 CPU has 4 communication ports: one RS-232C port, one RS485 port, one 10M/100M adaptive Ethernet port (20-point NK1 CPU has no Ethernet, it has two RS-232C ports), one USB port. In addition, 40/60-point NK1 CPU supports the installation of a function board which can add a RS-232C or RS-485 port.

RS-232C, RS-485 and Ethernet communication port can be used to communicate with other devices with the same communication port. These ports can also be used to connect KPPSoft programming software. You can use KPPSoft software to edit user program or configure operating parameters of the PLC through any one of these ports.

USB port is mainly used to download user program or system parameters and upgrade PLC firmware.

In order to facilictate the description, the communication ports of NK1 PLC are numbered in this manual: RS-232C port built-in CPU is called PORT 0 (referred to as P0); 3-wire RS-485 port is called PORT 1 (referred to as P1); the port on the optional function board is called PORT 2 (referred to as P2) (the second RS-232C port on 20-point CPU is also called P2 port); Ethernet port is called PORT 3 (referred to as P3). P0 is called programming port, other communication ports are called universal communication port.

4-1-1 NK1 CPU ports specifications and wiring

1. PORT 0 specifications and wiring

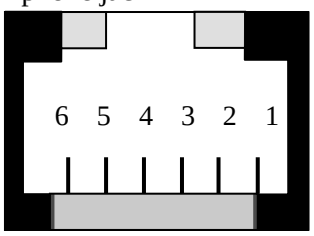
- 6-pin female modular（RJ12 phone jack）.
- Protocol: CCM2 (DirectNET) (slave only), K-sequence (slave only), MODBUS RTU（master/slave），non-sequence（master/slave），KPPSoft dedicated protocol.

  In TERM-STOP mode, P0 operates according to the following parameters:

  Station number = 1; communication speed = 19200bps; stop bits = 1; parity = no parity.

  In other modes, P0 operates according to parameters set up in its commnication registers.
- RS-232C
- Communication speed: up to 15200bps
- RS-232C communication distance: less than 10 meters.
- Nodes: one reception, one send, suitable for one pair connection.

6-pin phone jack



Pin No.	Signal name	Descriptions
1	0V	5V power (－) connection（GND）
2	5V	5V power(＋) connection
3	RX	Receive data (RS-232C)
4	TX	Transmit data(RS-232C）
5	NC	
6	0V	5V power (－) connection（GND）

RS-232C communication port is generally used to connect two RS-232C communication devices.
Wiring when NK1 PLC connects to other RS-232C communications devices through port P0:

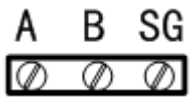
P0			RS-232C port on other devices
Pin No	signal		Signal
3	RX	———	TX
4	TX	———	RX
1 or 6	0V	———	0V

Wiring when NK1 PLC connects to computer's 9-pin serial port through port P0:

P0			9-pin RS-232C port on computer	
Pin No.	Signal		Signal	Pin No.
3	RX	———	TX	3
4	TX	———	RX	2
1 or 6	0V	———	GND	5
			DSR	6
			RTS	7
			CTS	8

2. PORT 1 specifications ang wiring

- 3-wire terminal
- Protocol: CCM2 (DirectNET) (slave only), K-sequence (slave only), MODBUS RTU (master/slave），non-sequence (master/slave), KPPSOFT dedicated protocol.
- RS-485
- Communication speed: up to 15200bps
- RS-485 communication distance: 9600bps or less, maximum 1km; above 19200bps，maximum 500m.
- Nodes: up to 32.
- Cable specifications: 16-26 AWG.



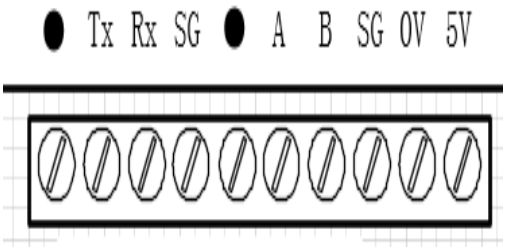
Terminal description of PORT 1		
1	A	Differential signal +
2	B	Differential signal -
3	SG	GND

Wiring when NK1 PLC connects to other RS-485 communications device through prot P1:

P1			RS-485 device	Remarks
Pin No.	signal		signal	Some RS-485 interface signals are labeled D+, D-. D+ is A，D- is B.
1	A	———	A	
2	B	———	B	
3	SG	———	SG	

3 PORT 2 specifications ang wiring

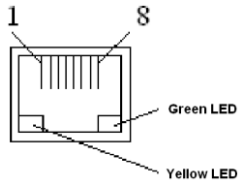
- It is a port on the optional function board NK1-BDCM.
- It can be RS-232C or RS-485.
- Protocol: CCM2（DirectNET）（slave only），K-sequence（slave only），MODBUS RTU（master/slave），non-sequence（master/slave），KPPSoft programming software dedicated communication protocol.
- Hardware performance meets the relevant communication specifications.
- Maximum communication speed is 115200bps.
- RS-232C communication distance: no more than 10 meters; RS-485 communication distance: 9600bps or less, no more than 1km; 19200bps above, no more than 500m.
- Nodes: it only supports one pair connection when using RS-232C communication; it supports more connection than one pair, up to 32 when using RS-485 communication.
- Cable specifications: 16-26 AWG.

Terminals assignment（either RS-232C or RS-485）	terminal description of Port 2		
	Terminal	descriptions	remark
	Tx	Data transmission	Suitable for RS-232C communication
	Rx	Data reception	
	SG	Ground	
	A	Differential signal +	Suitable for RS-485 communication. Some RS-485 interface signals are labeled D+, D-. D+ is A，D- is B.
	B	Differential signal -	
	SG	Ground	
	0V	5V power (－) connection	Suitabloe for RS-232C communication.
	5V	5V power（＋）connection	

When using RS-232C communication, wiring method refers to port P0 relevant content; when using RS-485 communication, wiring method refers to port P1 relevant content.

4 PORT 3 specifications and wiring

- Standard RJ45 plug and socket, meets IEEE802.3 standards.
- Protocols: ECOM (containing KPPSoft dedicated communication protocol, MODBUS TCP/IP protocol, and ECOM UDP protocol).
- 10/100M adaptively.
- Communication distance is not more than 100 meters between the two ends (STP).
- Number of connections: 4 MODBUS TCP/IP conncetions, multiple ECOM UDP and 1 programming software.



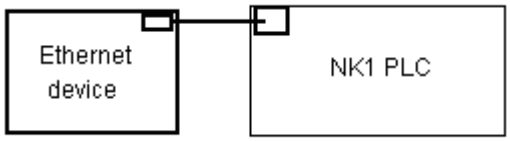
Pin No.	signal	Pin No.	Signal
1	TD+	5	Null
2	TD-	6	RD-
3	RD+	7	Null
4	Null	8	Null

Yellow LED: ON when sending and receiving signal; green LED: ON when properly connect.

Port 3 uses a standard Ethernet communication cable, you can buy a commercial standard Cat5e Ethernet communication cable, and you can also make an Ethernet communication cable yourself.

If the conection type of NK1 and Ethernet devices is different, the cables are different too.

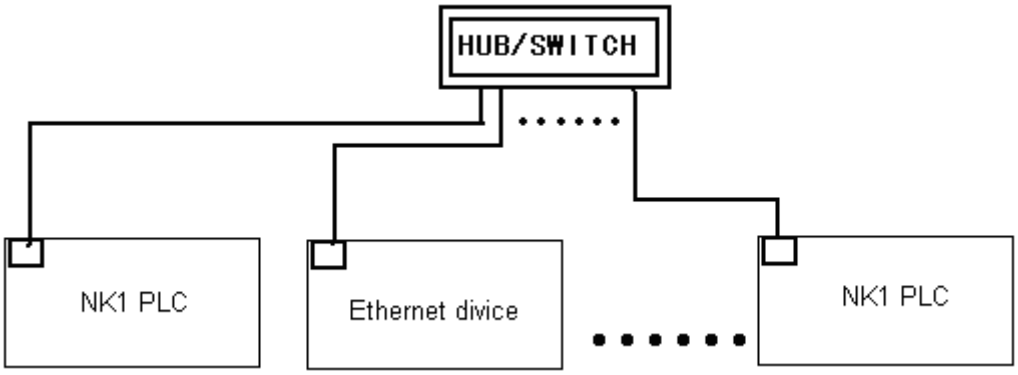
a) When NK1 and Ethernet device is one to one connection.



The following table shows cable wiring:

Port 3			Ethernet device		remark
Pin No.	signal		Pin No.	Signal	
1	TD+		1	TD+	
2	TD-		2	TD-	
3	RD+		3	RD+	
6	RD-		6	RD-	

b) When NK1 connects to Ethernet network by adding a HUB or other device.



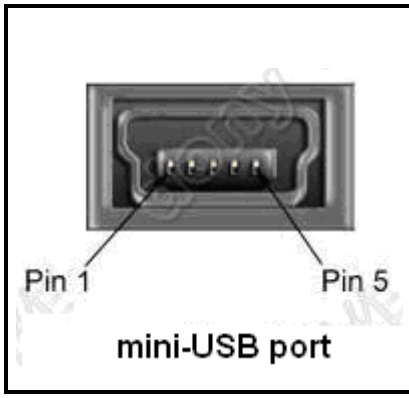
The following table shows cable wiring:

PORT 3			Other ethernet device		Remark
Pin No.	Signal		Pin No.	Signal	
1	TD+		1	TD+	
2	TD-		2	TD-	
3	RD+		3	RD+	
6	RD-		6	RD-	

Note: now, a lot of HUB / SWITCH equipments have automatic jumper function, so the signal line crossover cable can also be used in HUB connection.

3. Mini USB port specifications and wiring

- Standard mini USB B-type socket.
- OTG protocol (needs OTG cable).
- Communication speed: USB 2.0 FULL SPEED.
- Communication distance is not more than 5m.
- Be used to download user program and system parameters (FAT format, less than 32G) and upgrade system firmware.



Pin No.	Signal
1	VBus
2	Data-
3	Data+
4	ID
5	GND

4-2 Protocol and parameters setup of serial communication port on NK1 CPU

4-2-1 Protocol

The following table lists the protocols supported by each serial communication ports:

	P0（RS-232C）		P1（3-wire RS-485）		P2（function board）	
	RUN	STOP	RUN	STOP	RUN	STOP
KPPSoft dedicated protocol	Δ	○	○	○	○	○
K-sequence	○	×	○	○	○	○
CCM2（DirectNET）	○	×	○	○	○	○
MODBUS RTU	○	×	○	○	○	○
Non-sequence	○	×	○	○	○	○

○：support Δ：support in fixed communication parameters ×：don't support

Note：1）P0 is programming port; P1, P2 are general purpose.

2）P2 on 20-point CPU is a fixed 6-pin phone jack.

3）In RUN mode, P0 operates according to the communication parameters in setup registers;

If programming software dedicated protocol is using, P0 must use the fixed communication parameters of STOP mode. In STOP mode, P0 operates according to the fixed communication parameters (station number is 1/19200bps/1 stop bit/ no parity).

4）Using programming software dedicated protocol, you can read and write user program, system parameters, function memory,etc. Using other protocols, you can only read and write the contents in funciton memory.

5）K-sequence and directNET protocol support only slave station function.

4-2-2 Setting registers

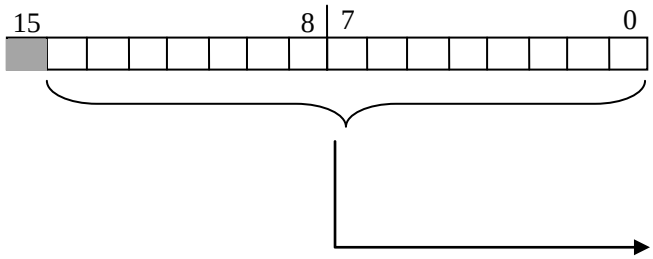
The setting data of each port is stord in some special registers. Each port on NK1 PLC is assigned a bank of 10 special registers.

The following table lists these registers:

Parameters	P0 special registers	P1 special registers	P2 special registers	The factory default value in register
Select the communication protocol	R7650	R7662	R7674	0x21
Parameters when have protocol ①/ Non-sequence parameters ①	R7651	R7663	R7675	00
Parameters when have protocol ②/ Non-sequence parameters ②	R7652	R7664	R7676	0x0701
Non-sequence parameters ③	R7653	R7665	R7677	0X70
Non-sequence parameters ④	R7654	R7666	R7700	0
Non-sequence parameters ⑤	R7655	R7667	R7701	0
Non-sequence parameters ⑥	R7656	R7670	R7702	0
Unused, reserved	R7657	R7671	R7703	0
Unused, reserved	R7660	R7672	R7704	0
Parameters setup completion flag	R7661	R7673	R7705	0

You can set the communicaiton parameter through the special register date setting(with ladder program) or through the KPPsoft communicatio parameter setting menu.

1. Communication protocol selection registers（R7650/R7662/R7674）
They are used to select protocol for each port.



Protocol
0x0080 = K-sequence
0x0040 = CCM2 (Direct net)
0x0020 = Modbus RTU
0x0010 = non-sequence
0x0008 = reserved
0x0001 = KPPSoft dedicated protocol

All communication ports of NK1 does not support multi-protocol auto-selection. It means that you can not select two or more protocol for one port of NK1. It is specially for KPPSoft dedicated protocol you can select this protocol and the Modbus RTU protocol for the same port.

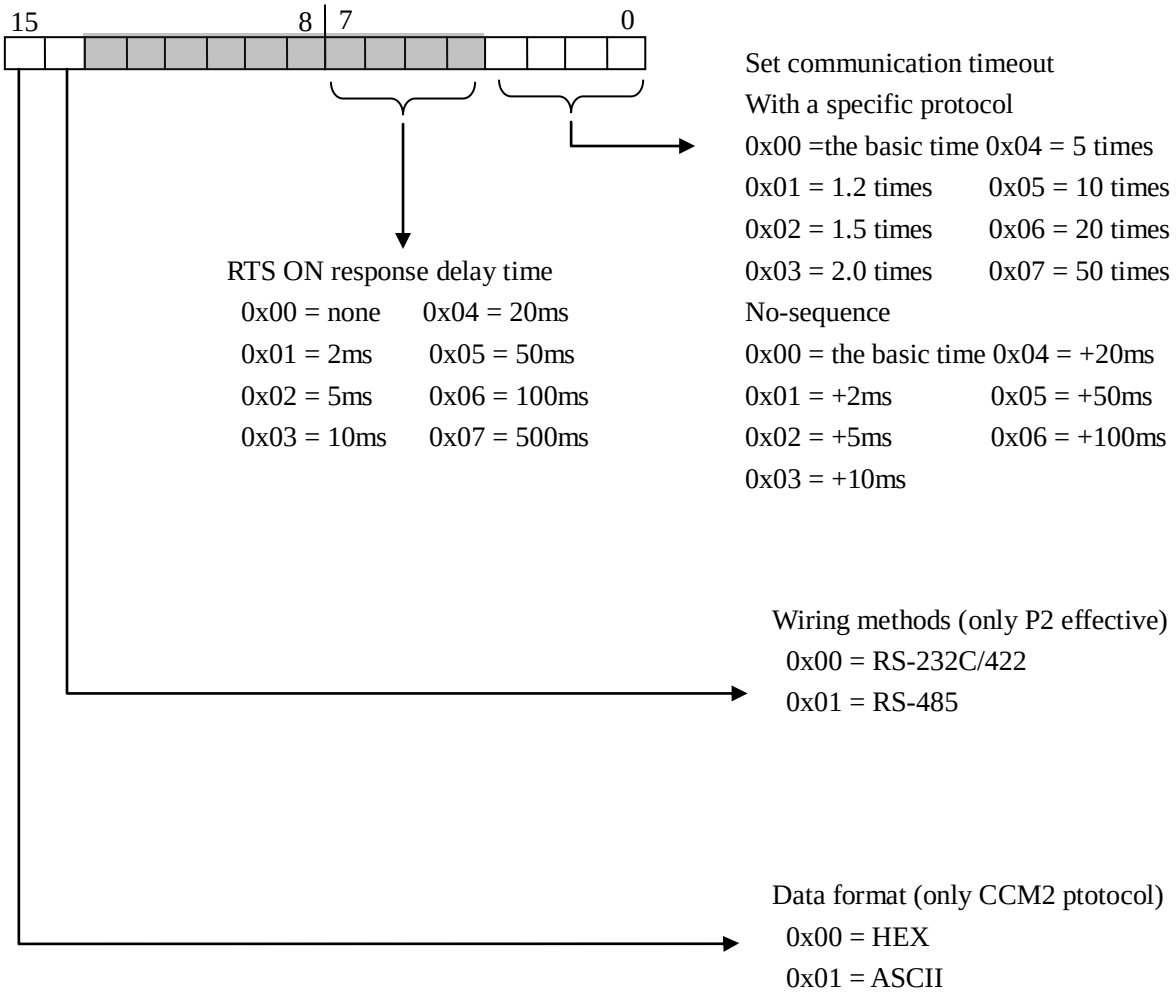
The default setting of the protocol selection registers is 0X21.It means the port is support the protcol Modbus RTU and KPPSoft dedicated protocol simultaneously.

So when you select the protocol of K-senquence, or CCM2(DirectNET) protocol, or non-senquence for a communication port ,you will can not connect KPPSoft with the NK1 series PLC through the port. But the Modbus RTU protocol is specially.You can connect KPPSoft with the NK1 series PLC through a port when you select the Modbus RTU protocol for the port.

Please remember that you always can connect KPPSoft with the NK1 series PLC through the RS-232C port when the NK1 is in "Stop" status.

2. Parameters setup registers when have protocol ①/ Non-sequence parameters setup registers ①（R7651/R7663/R7675）

They are used to select timeout time, wiring method of P2, and CCM2 communication data format when the protocol is K-sequence/ CCM2/ MODBUS RTU.



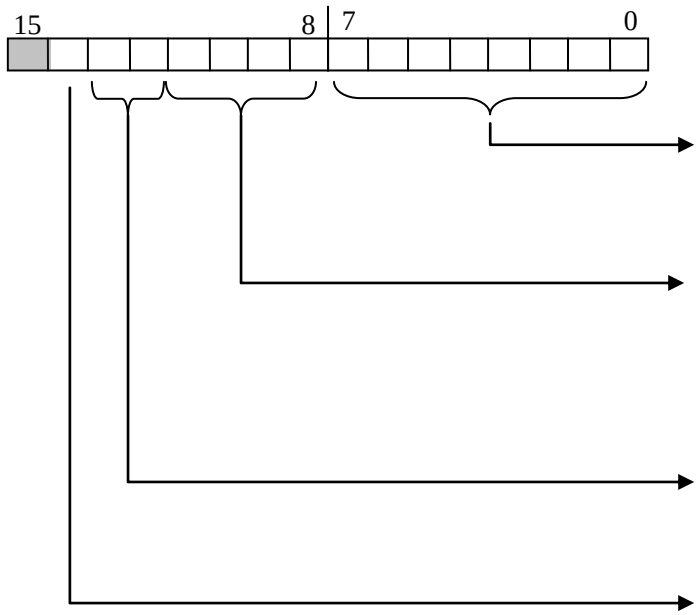
NOTE:

1) The "RTS ON response delay time" setting is only for RS-485 communication port when the system firmware of NK1 is V20190716 or before.

2) In order to set the "RTS ON response delay time" parameter for all RS-232C ports through the KPPsoft. You need the soft version of the KPPSoft is V1.6.6.9 or later.

3. Parameters setup registers when protocol communication ②/ Non-sequence parameters setup registers ②/（R7652/R7664/R7676）

They are used to select station number, communication speed, parity bits, stop bits and other communication parameters when using K-sequence/ CCM2/ MODBUS RTU. Note: although the range of station number is 0x01 ～ 0xFF, but in fact its range is 0x01 ～ 0x5A（01 ～ 90）for K-sequence and CCM2 protocol; its rang is 0x01 ～ 0XF7（01 ～ 247）for MODBUS RTU protocol.



Local station No. (0x01 ～ 0xFF)

Communication baud rate

0x03 = 2400bps 0x08 = 28800bps

0x04 = 4800bps 0x09 = 38400bps

0x05 = 9600bps 0x0A = 57600bps

0x06 = 14400bps 0x0B = 115200bps

0x07 = 19200bps

Parity

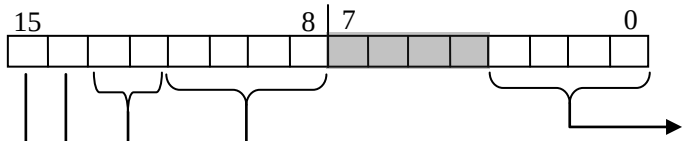
0x00 = none 0x01 = odd

0x02= even

Stop bits

0x0 = 1 bit 0x1 = 2 bits

They are used to select the communication speed, parity, stop bits, data bits, etc when non-sequence communication.



Spare

Communication baud rate

0x03 = 2400bps 0x08 = 28800bps

0x04 = 4800bps 0x09 = 38400bps

0x06 = 14400bps 0x0A = 57600bps

0x05 = 9600bps 0x0B = 115200bps

0x07 = 19200bps

Parity

0x00 = none 0x01 = odd

0x02= even

Stop bits

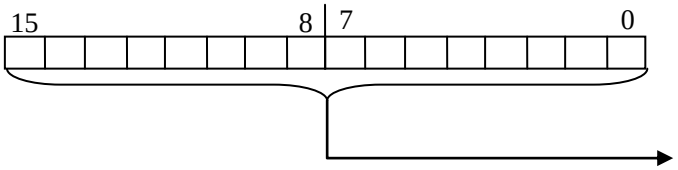
0x0 = 1bit 0x1 = 2bit

Data bits

0x0 = 7bit 0x1 = 8bit

4. Non-sequence parameters setup registers ③（R7653/R7665/R7677）

They are used to specify communication mode of no protocol communication. NK1 PLC supports mode 61, mode 62, mode 70 and mode 71.



Mode selection register

Mode 61

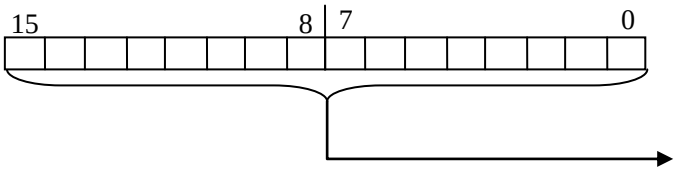
Mode 62

Mode 70

Mode 71

5. Non-sequence parameters setup registers ④（R7654/R7666/R7700）

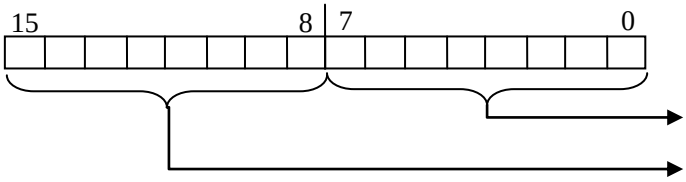
They are used to specify data reception start code when no protocol communication. If the beginning code can not be fixed, the data should be set to "0". NK1 PLC currently don't support mode 5E. These registers are reserved.



The beginning code

6. Non-sequence parameters setup registers ⑤（R7655/R7667/R7701）

They are used to specify data reception end code when mode 70 or 71.



End code 1 (0x00 ～ 0xFF)

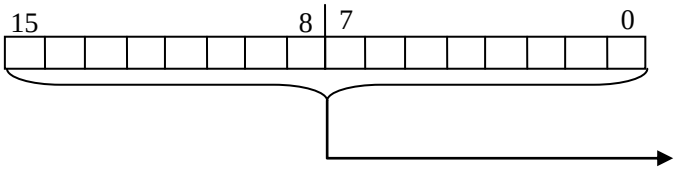
End code 2 (0x01 ～ 0xFF)

7. Non-sequence parameters setup registers ⑥（R7656/R7670/R7702）

They are used to specify address to store reception data

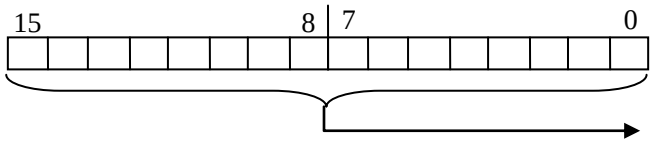
example） When no protocol communication through P0, the storage address starts from R2000, you should set: R7656 = 0400（H）

2000 in octal is 4000 in hexadecimal. 2000（O） = 400（H）



The beginning register to store reception data.

8. Parameters setup completion flag registers（R7661/R7673/R7705）



Parameter setup completion code is 0xA55A.

If the parameters are set correctly, it becomes 0x5555.

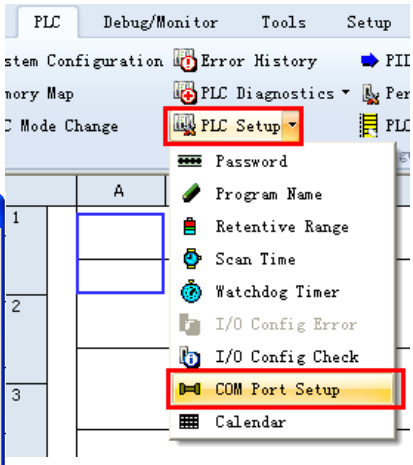
If the parameter is set incorrectly, the number of the register which stores the incorrect parameter will be written in this register.

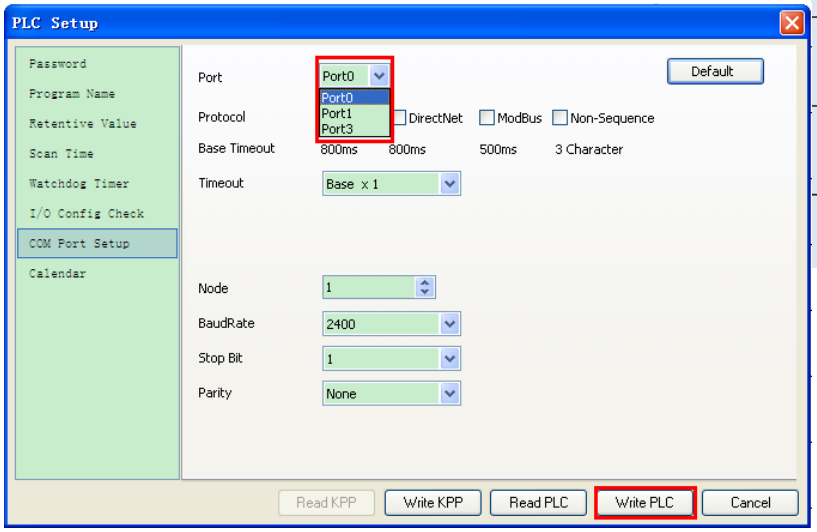
4-2-3 NK1 CPU ports configuration

There are two ways to configure ports on NK1 CPU:

1) Configure port by KPPSoft software:

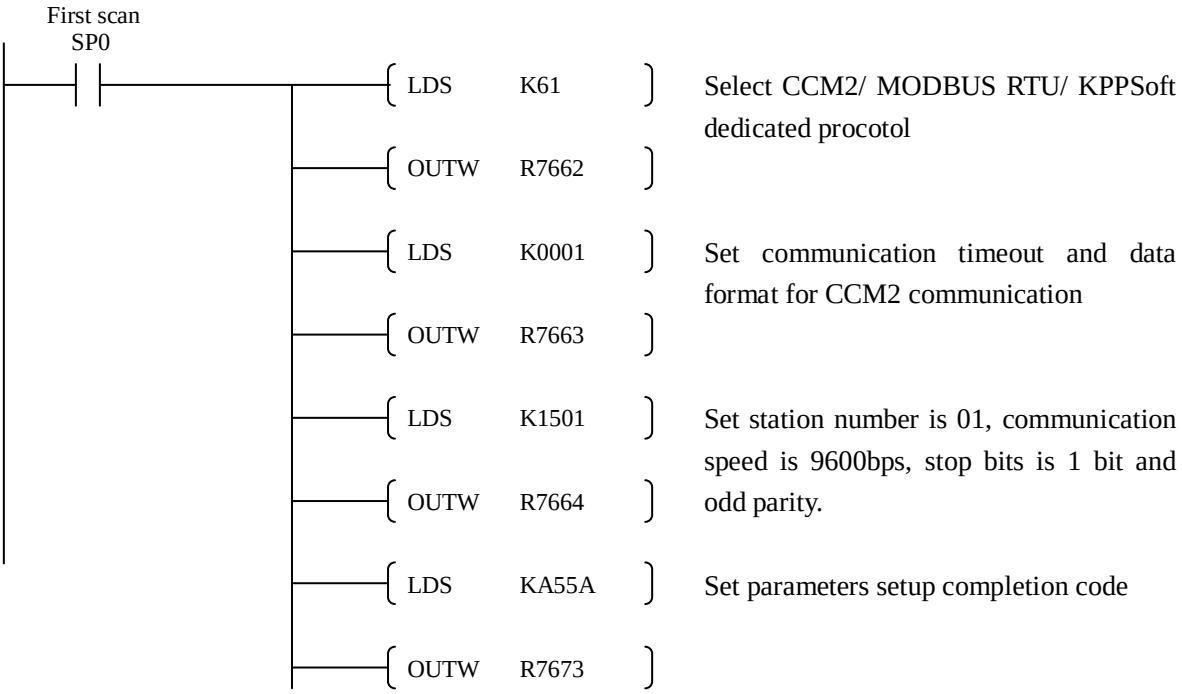
In KPPSoft, as shown on the right, choose PLC > PLC Setup > COM Port Setup, open PLC Setup window, select the communication port and setup the parameters, and then press the key [Write PLC], the setup of the port's parameters are finished.





2) Configure port by setup special registers in user program
The following example shows the setting program of PORT1.

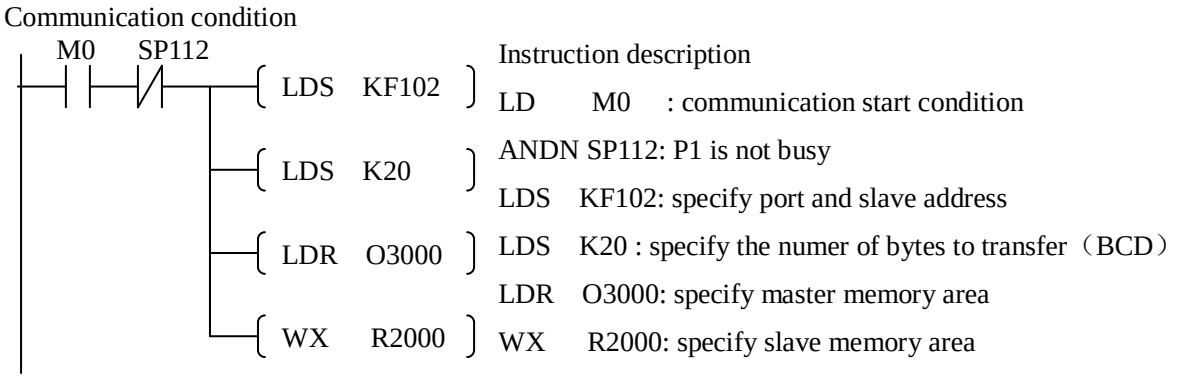
PORT : PORT1（P1）
Protocol : CCM2/MODBUS RTU/KPPSOFT dedicated procotol, adaptive
Communication timeout : 1.2 times of the predetermined time
Station number : 01
Data format : HEX （only CCM2）
Communication speed : 9600bps
Stop bits : 1
Parity : odd



4-2-4 Write communication program of serial port on NK1 CPU

NK1 PLC can be used as a master and a slave. When it is used as a slave, you only need to set up the relevant communication parameters without the need of writing a special communication program. When it is used as a master, you also need to write a special communication program besides setup the relevant communication parameters. This section teaches you how to design the required ladder logic for network master operation.When using NK1 PLC as the master station, simple RLL instructions are used to initiate the communication requests. The WX instruction initiates network write operations, and the RX instruction initiates network read operations. Before executing either the WX or RX commands, we will need to load data related to the read or write operation onto the CPU's accumulator stack. When the WX or RX instruction executes, it uses the information on the stack combined with data in the instruction box to completely define the task, which goes to the port.

The following procedure will provide you the information necessary to setup your ladder program to write data to a network slave.



1） SP112: port P1 is busy

Special relays	Functions
SP54	On if there is parameters setup error of any one communication port.
SP110	On when port P0 is communicating with another device.
SP111	On when port P0 has encountered a communication error.
SP112	On when port P1 is communicating with another device.
SP113	On when port P1 has encountered a communication error.
SP114	On when port 2 is communicating with another device.
SP115	On when port 2 has encountered a communication error.

2）Identify master port # and slave #

LDS KF102

The LDS instruction specifies the port number on the network master (NK1 PLC) and the address of the slave station.

We rewrite this instruction in the following format:

LDS KABCD

AB identifies master port #:

A B = F0 port P0

A B = F1 port P1

A B = F2 port P2

CD identifies slave # (hex). For example, 02 represents that slave address is 2, 10 represents that slave address is 16, etc. Note that slave address of MODBUS RTU is 1~247, so the range of CD is 0x01~0xF7.

3) Specify number of bytes to transfer

LDS K20

Specify number of bytes to transfer, the value is in BCD format, from 1 to 128 bytes.

4) Specify master memory area

LDR O3000

Its purpose is to load the starting address of the memory area to be transferred.

5) Specify slave memory area

WX R2000 （use WX to write to the slave, and RX to read from the slave）

NOTE: NK1 is not support the RX/WX instruction when use the MODBUS RTU protocol now! Please use the MRX/MWX instruction.

4-3 K-sequence/CCM2（DirectNET） protocol communication

All serial ports on NK1 PLC support k-sequence and CCM2 (DirectNET) protocol slave communication function.

If you want to use this communication function, you need first to set communication parameters of the port. Communication parameters are stord in a bank of registers.

Parameters	P0 special registers	P1 special registers	P2 special registers
Select communication protocol	R7650	R7662	R7674
Communication timeout, RTS ON time setup when RS-485 communication, P2 port wiring methods, CCM2 communication data format	R7651	R7663	R7675
Local station No., communication speed, parity, stop bits	R7652	R7664	R7676
Parameters setup completion flag	R7661	R7673	R7705

Please refer to section 4-2-2 for details.

When NK1 PLC is used as a slave, you only need to setup the relevant communication parameters without the need of writing a special communication program.

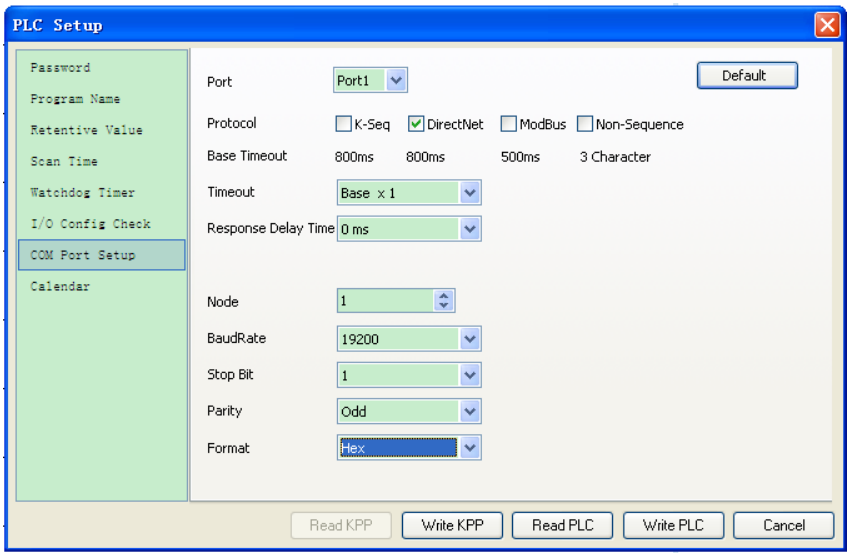
There are two methods to configure ports on NK1 CPU: by KPPSoftware and by setup special registers in user program.

The following example shows you how to set parameters of port P1:

Protocol : CCM2/KPPSOFT dedicated protocol, adaptive
Communication timeout : specified time
RTS ON time : 0ms
Station No. : 01
Data format : hex (only CCM2)
Communication speed : 19200bps
Stop bits : 1
Parity : odd

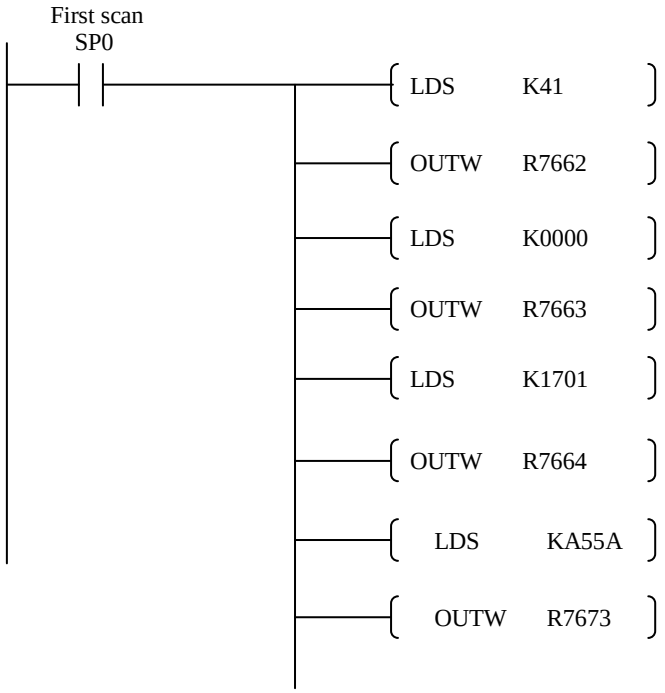
1) Configure port by KPPSoft software

In KPPSoft, as shown on the right， choose PLC > PLC Setup > COM Port Setup, open PLC Setup window, select the communication port and setup the appropritate parameters, and then press the key [Write



PLC], the setup of the port's parameters are finished.

2) Configure port by user program



Select CCM2/KPPSOFT dedicated protocol.

Set communication timeout, no RTS ON, data format is HEX (CCM2).

Station number is 01, communication speed is 19200bps, stop bit is 1 bit and odd parity.

Set parameters setup completion code.

Add the above initial program segment to the user program. After the execution of the user program, port P1 can be initialized.

4-4 MODBUS RTU protocol communication

All serial ports on NK1 PLC support MODBUS RTU protocol master and slave communication function. Use this communication function, you can read and write:

- Read the current value of timer and counter
- Read and write data in registers
- Read and write I/O internal relays and stages, etc.
- Read and wirte user program and system parameters.

4-4-1 MODBUS RTU protocol communication parameters setup

If you want to use MODBUS RTU communication function, you need to set communication parameters of the port. Communication parameters are stord in a bank of registers.

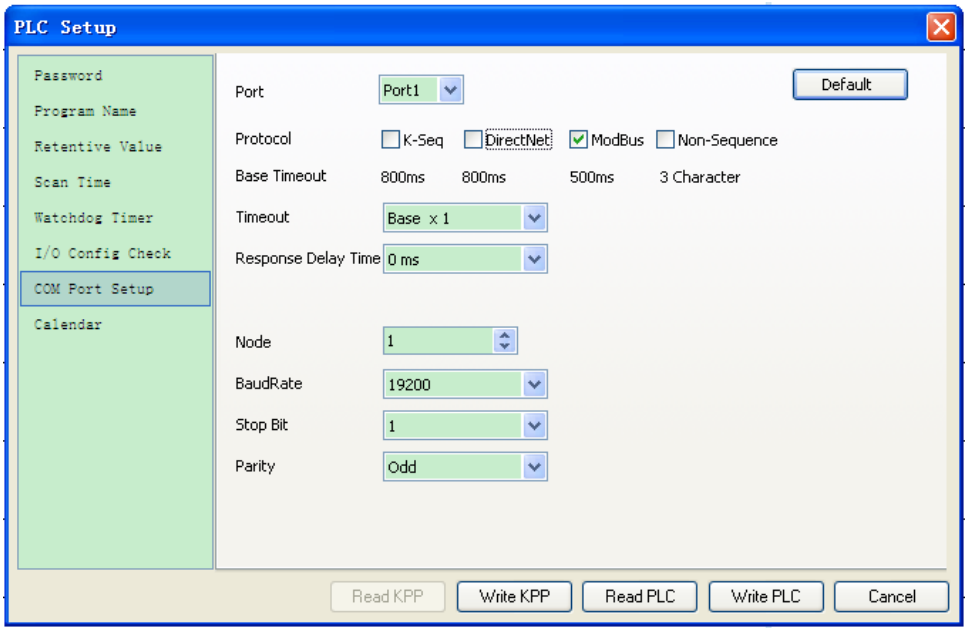
Parameters	P0 special registers	P1 special registers	P2 special registers
Select communication protocol	R7650	R7662	R7674
Communication timeout, RTS ON time setup when RS-485 communication, P2 port wiring methods	R7651	R7663	R7675
Local station No., communication speed, parity, stop bits	R7652	R7664	R7676
Parameter setup completion flag	R7661	R7673	R7705

Please refer to section 4-2-2 for details.

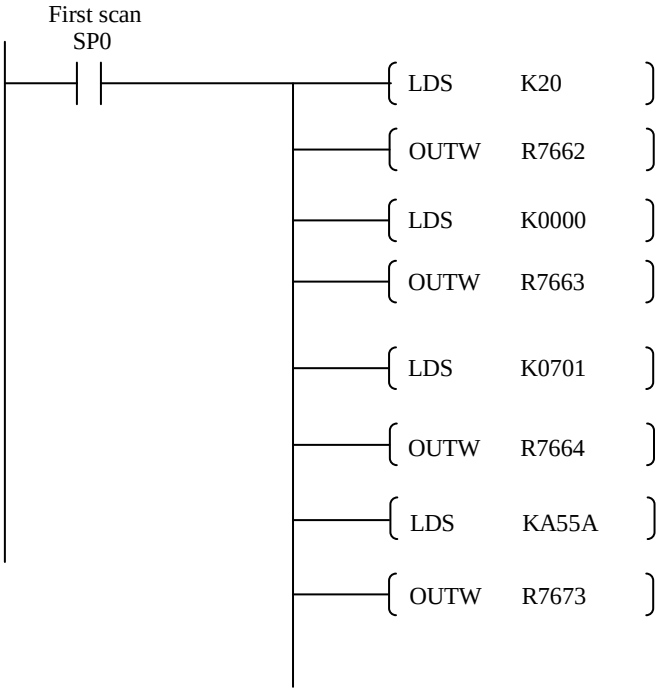
There are two methods to configure ports on NK1 CPU: by KPPSoftware and by setup special registers in user program.
The following example shows you how to setup parameters of P1 port:

Protocol : MODBUS RTU protocol
Communication timeout : specified time
RTS ON time : 0ms
Station No. : 01
Communication speed : 19200bps
Stop bits : 1
Parity : none

1) Configure port by KPPSoft software
In KPPSoft, as shown on the right, choose PLC > PLC Setup > COM Port Setup, open PLC Setup window, select the communication port and setup the appropritate parameters, and then press the key [Write PLC], the setup of the port's parameters are finished.



2) Configure port by user program



Select MODBUS RTU protocol. After the program is executed, port P1 will not be able to communicate with KPPSoft software.

Set communication timeout, no RTS ON.

Station number is 01, communication speed is 19200bps, stop bit is 1 bit and none parity.

Write parameters setup completion code.

4-4-2 MODBUS RTU protocol network slave operation

This section describes how other devices on a network can communicate with a CPU port that you have configured as a Modbus slave (NK1). A Modbus host must use the Modbus RTU protocol to communicate with the NK1 as a slave. The host software must send a Modbus function code and Modbus address to specify a PLC memory location the NK1 comprehends. No CPU ladder logic is required to support Modbus slave operation.

1. MODBUS function codes supported

The Modbus function code determines whether the access is a read or a write, and whether to access a single data point or a group of them. The NK1 PLC supports the Modbus function codes described below.

MODBUS function code	function	NK1 data types available
01	Read a group of coils	Q，M，T，C
02	Read a group of inputs	I，SP
05	Set / Reset a single coil (only slave)	Q，M，T，C
15	Set / Reset a group of coils	Q，M，T，C
03，04	Read a value from one or more registers	R
06	Write a value into a single register (only slave)	R
16	Write a value into a group of registers	R

2. Determining the Modbus Address

There are typically two ways that most host software conventions allow you to specify a PLC memory location. These are:

- By specifying the Modbus data type and address
- By specifying a Modbus address only

（1） If Your Host Software Requires the Data Type and Address...

Many host software packages allow you to specify the Modbus data type and the Modbus address that corresponds to the PLC memory location. This is the easiest method, but not all packages allow you to do it this way.

The actual equation used to calculate the address depends on the type of PLC data you are using. The PLC memory types are split into two categories for this purpose.

• Discrete – I, SP, Q, M, S, T, C (contacts)

• Word – R, Timer current value, Counter current value

In either case, you basically convert the PLC octal address to decimal and add the appropriate Modbus address (if required). The table below shows the exact equation used for each group of data.

NK1 memory type	QTY (Dec.)	PLC range (octal)	MODBUS address range (decimal)	MODBUS data type
For Discrete Data Types .... Convert PLC Addr. to Dec. + Start of Range + Data Type				
Global inputs (GI)	2048	GI0~GI3777	0~2047	Input
Inputs (I)	1024	I0~I1777	2048~3071	Input
Special relays (SP)	1024	SP0~SP1777	3072~4095	Input
Global outputs (Q)	2048	GQ0~GQ3777	0~2047	Coil
Outputs (Q)	1024	Q0~Q1777	2048~3071	Coil
Control relays (M)	2048	M0~M3777	3072~5119	Coil
Stage (S)	1024	S0~S1777	5120~6143	Coil
Timer (T)	512	T0~T777	6144~6655	Coil
Counter (C)	512	C0~C777	6656~7167	Coil
For Word Data Types .... Convert PLC Addr. to Dec. + Data Type				
Timer current values (R)	512	R0~R777	0~511	Input register
Counter current values (R)	512	R1000~R1777	512~1023	Input register
Data register 1(R)	2816	R2000~R7377	1024~3839	Holding register
Data registers 2(R)	11776	R10000~R36777	4096~15871	Holding register
Special registers 1(R)	256	R7400~R7777	3840~4095	Holding register
Special registers 2(R)	512 字	R37000~R37777	15872~16383	Holding register

The following examples show how to generate the Modbus address and data type for hosts which require this format.

Example 1: R2100

Find the Modbus address for
User R location R2100.

1. Find R memory in the table.
2. Convert R2100 into decimal (1088).
3. Use the Modbus data type from the table.

PLC Address (Dec) + Data Type

R2100=1088 decimal

1088+Hold.Reg.= Holding Reg. 1088

R Memory, user data（R）	2816	R2000-R7377	1024-3839	Holding Register

Example 2: Q20

Find the Modbus address for output Q20.

1. Find Q outputs in the table.
2. Convert Q20 into decimal (16).
3. Add the starting address for the range (2048).
4. Use the Modbus data type from the table.

PLC Address (Dec) + Start Addr.+ Data Type

Q20 = 16 decimal

16 + 2048+ Coil = Coil 2064

Outputs (Q)	1024	Q0~Q1777	2048-3071	Coil

Example 3: T10 Current Value

Find the Modbus address to obtain the current value from Timer T10.

1. Find Timer Current Values in the table.
2. Convert T10 into decimal (8).
3. Use the Modbus data type from the table.

PLC Address (Dec) + Data Type

T10 = 8 decimal

8 + Input Reg = Input Reg.8

Timer current vaulues (R)	512	R0~R777	0~511	Input register

Example 4: M54

Find the Modbus address for Control Relay M54.

1. Find Control Relays in the table.
2. Convert M54 into decimal (44).
3. Add the starting address for the range (3072).
4. Use the Modbus data type from the table.

PLC Address (Dec) + Start Addr.+ Data Type

M54 = 44 decimal

44 + 3072 + Coil = Coil 3116

Control relays (M)	2048	M0~M3777	3072~5119	Coil

（2）If Your Modbus Host Software Requires an Address ONLY

Some host software does not allow you to specify the Modbus data type and address. Instead, you specify an address only. This method requires another step to determine the address, but it's still fairly simple. Basically, Modbus also separates the data types by address ranges as well. So this means an address alone can actually describe the type of data and location. This is often referred to as "adding the offset". One important thing to remember here is that two different addressing modes may be available in your host software package. These are:

• 484 Mode

• 584/984 Mode

We recommend that you use the 584/984 addressing mode if your host software allows you to choose. This is because the 584/984 mode allows access to a higher number of memory locations within each data type. If your software only supports 484 mode, then there may be some PLC memory locations that will be unavailable. The actual equation used to calculate the address depends on the type of PLC data you are using. The PLC memory types are split into two categories for this purpose.

• Discrete – I, SP, Q, M, S, T (contacts), C (contacts)

• Word – R, Timer current value, Counter current value

In either case, you basically convert the PLC octal address to decimal and add the appropriate Modbus addresses (as required). The table below shows the exact equation used for each group of data.

NK1 Memory Type	QTY (Dec.)	NK1 PLC range （octal）	MODBUS address range （decimal）	Address (484 Mode)	Address (584/984 Mode)	Modbus Data Type
For Discrete Data Types .... Convert PLC Addr. to Dec. + Start of Range + Mode Address						
Global inputs (GI)	2048	GI0~GI3777	0~2047	1001	10001	Input
Inputs (Ⅰ)	1024	I0~I1777	2048~3071	1001	10001	Input
Special relays (SP)	1024	SP0~SP1777	3072~4095	1001	10001	Input
Global outputs （GQ）	2048	GQ0~GQ3777	0~2047	1	1	Coil
outputs（Q）	1024	Q0~Q1777	2048~3071	1	1	Coil
Control relays （M）	2048	M0~M3777	3072~5119	1	1	Coil
Timer (T)	512	T0~T777	6144~6655	1	1	Coil
Counter (C)	512	C0~C777	6656~7167	1	1	Coil
Stages (S)	1024	S0~S1777	5120~6143	1	1	Coil
For Word Data Types .... Convert PLC Addr. to Dec. + Mode Address						
Timer current values (R)	512	R0~R777	0~511	3001	30001	Input register
Counter current values (R)	512	R1000~R1777	512~1023	3001	30001	Input register
Data register 1(R)	2816	R2000~R7377	1024~3839	4001	40001	Holding register
Data registers 2(R)	11776	R10000~R36777	4096~15871	4001	400001	Holding register
Special registers 1(R)	256	R7400~R7777	3840~4095	4001	40001	Holding register
Special registers 2(R)	512	R37000~R37777	15872~16383	4001	400001	Holding register

Discrete data types:

NK1 memory types	PLC range（octal）	484 mode address	584/984 mode address	Modbus data type
Global inputs (GI)	GI0 ~ GI1746	1001 ~ 1999	10001 ~ 10999	Input
	GI1747 ~ GI3777	——	11000 ~ 12048	Input
Inputs (Ⅰ)	I0 ~I1777	——	12049 ~ 13072	Input
Special relays (SP)	SP0~ SP1777	——	13073 ~ 15120	Input
Global outputs (GQ)	GQ0 ~ GQ1746	1 ~ 999	1 ~ 999	Output
	GQ1747 ~ GQ3777	——	1000 ~ 2048	Output
Outputs (Q)	Q0 ~ Q1777	——	2049 ~ 3072	Output
Control relays (M)	M0 ~ M3777	——	3073 ~ 5120	Output
Stages (S)	S0 ~ S1777	——	5121 ~ 6144	Output
Timer contacts (T)	T0 ~ T777	——	6145 ~ 6656	Output
Counter contacts (C)	C0 ~ C777	——	6657 ~ 7168	Output

Word data types:

Registers (R)	PLC range (octal)	Input /holding (484 mode） *	Input / holding (584/984 mode)*
R-memory (timers)	R0~R377	3001/4001~3512/4512	30001/40001~30512/40512
R-memory (counters)	R1000~R1746	3513/4513~3999/4999	30513/40513~30999/40999
	R1747~R1777	---	31000/41000~31024/41024
R-memory (data words)	R2000~R7377	---	41025~43840
	R10000~R36777	---	404097~415872
R-memory (special parameters)	R7400~R7777	---	43841~44096
	R37000~R37777	---	415873~416384

Example 1: R2100 584/984 Mode

Find the Modbus address for user R-memory R2100.

1. Find R memory in the table.
2. Convert R2100 into decimal (1088).
3. Add the Modbus starting address for the mode (40001).

PLC address (Dec) + Mode Address
R2100 = 1088 decimal
1088 + 40001 = 41089

For Word Data Types .... Convert PLC Addr. to Dec. + Mode Address						
Timer current values (R)	512	R0~R777	0~511	3001	30001	Input register
Counter current values (R)	512	R1000~R1777	512~1023	3001	30001	Input register
Data register 1(R)	2816	R2000~R7377	1024~3839	4001	40001	Holding register

Example 2: Q20 584/984 Mode

Find the Modbus address for output Q20.

1. Find Q outputs in the table.
2. Convert Q20 into decimal (16).
3. Add the starting address for the range (2048).
4. Add the Modbus address for the mode (1).

PLC address (Dec) + Start Add + Mode
Q20 = 16 decimal
16 + 2048 + 1 = 2065

Outputs（Q）	1024	Q0~Q1777	2048~3071	1	1	Coil
Control relays（M）	2048	M0~M3777	3072~5119	1	1	Coil
Timer (T)	512	T0~T777	6144~6655	1	1	Coil

Example 3: T10 Current Value 484 Mode

Find the Modbus address to obtain the current value for Timer T10.

1. Find Timer Current Values in the table.
2. Convert T10 into decimal (8).
3. Add the Modbus starting address for the mode (3001).

PLC address (Dec) + Mode Address
T10 = 8 decimal
8 + 3001 = 3009

For Word Data Types .... Convert PLC Addr. to Dec. + Mode Address						
Timer current values (R)	512	R0~R777	0~511	3001	30001	Input register
Counter current values (R)	512	R1000~R1777	512~1023	3001	30001	Input register
Data register 1(R)	2816	R2000~R7377	1024~3839	4001	40001	Holding register

Example 4: M54 584/984 Mode

Find the Modbus address for Control Relay M54.

1. Find Control Relays in the table.
2. Convert M54 into decimal (44).
3. Add the starting address for the range (3072).
4. Add the Modbus address for the mode (1).

PLC address (Dec) + Start Add + Mode
M54 = 44 decimal
44 + 3072 +1 = 3117

Outputs（Q）	1024	Q0~Q1777	2048~3071	1	1	Coil
Control relays（M）	2048	M0~M3777	3072~5119	1	1	Coil
Timer (T)	512	T0~T777	6144~6655	1	1	Coil

4-4-3 MODBUS RTU protocol network master operation

For Modbus networks, it uses the Modbus RTU protocol.Modbus networks is a single master/multiple slave networks. The master is the only member of the network that can initiate requests on the network. When using the NK1 PLC as the master station, simple RLL instructions are used to initiate the requests. The WX instruction initiates network write operations, and the RX instruction initiates network read operations. Before executing either the WX or RX commands, we will need to load data related to the read or write operation onto the CPU's accumulator stack. When the WX or RX instruction executes, it uses the information on the stack combined with data in the instruction box to completely define the task, which goes to the port.

NOTE: NK1 is not support the RX/WX instruction when use the MODBUS RTU protocol now! Please use MRX/MWX instruction.

The following example provides you the information necessary to setup your ladder program to receive data from a network slave.

The NK1 PLC is the master station using port P1 (RS-485), the SN PLC is the slave station.

Communication parameters:

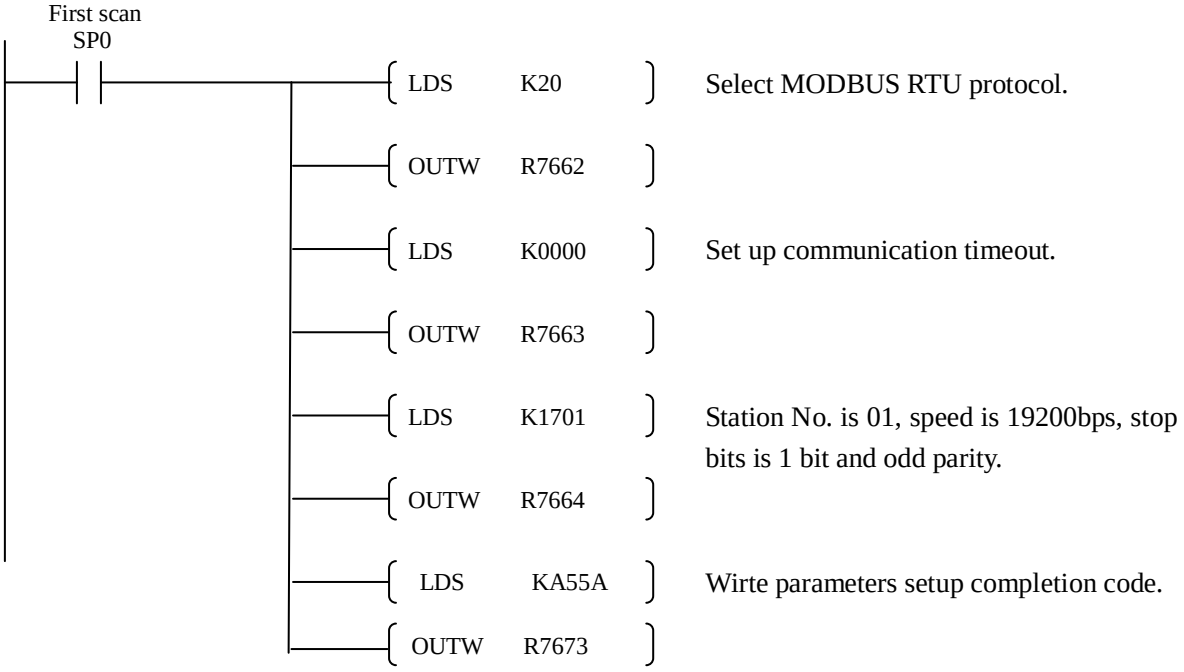
NK1 (master)	SN (slave)
Communication timeout: specified time	Station No.: 02
RTS ON: none	Communication timeout: specified time
Station No.: 01	Response delay time: 0ms
Speed: 19200bps	Speed: 19200bps
Stop bits: 1	Stop bits: 1
Parity: ODD	Parity: ODD

1） The communication speed, stop bits and parity of the master and slave station must be the same.

2） The slave station No. must be the same with the slave station No. in RX/WX instructions.

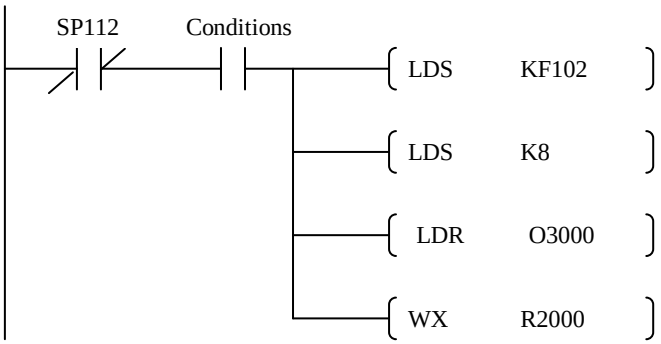
3） The master station No. can be any valid number, but it can not be the same with the slave stion No. in communication instructions.

A） the master station parameters setup procedure:



B）The master station communication procedure:

① Write data Wirte four words starting from R3000 in the master to four registers starting from R2000 in the slave.



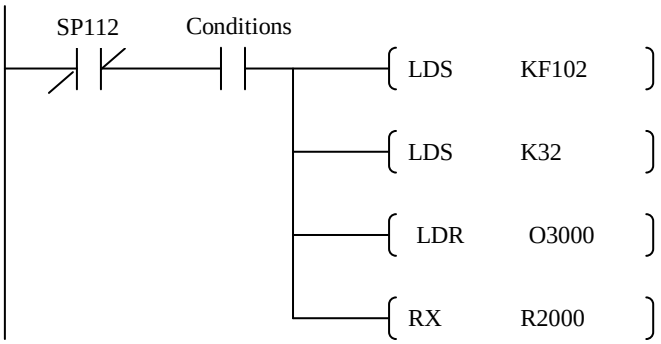
Use port P1; slave station No. is 02.

Number of bytes to transfer is 8.

Load the address of the data in the master that is to be written to the network into the accumulator.

Specify the starting R-memory location where the data will be written to the slave.

② Read data Read data from R2000~R2017 in the slave and write them to R3000~R3017 in the master.



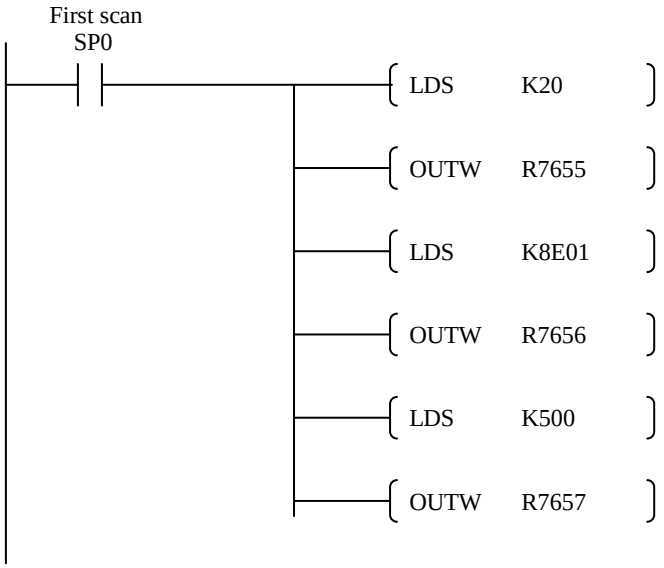
Use port P1; slave station No. is 02.

Number of bytes to transfer is 32.

Load the address in the master that stores data reading from the slave into the accumulator.

Specifies the staring R-memory location where the data will be read from in the slave.

C）The slave station parameters setup procedure:



Select MODBUS RTU protocol, set up communication timeout and response delay time.

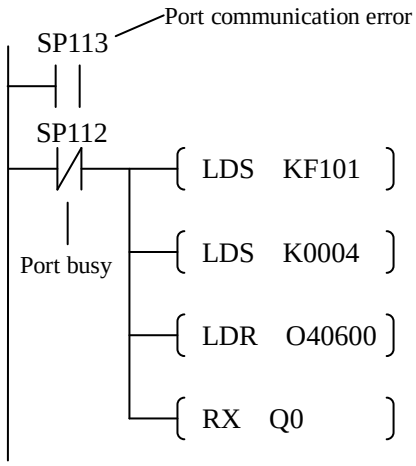
Station No. is 01, speed is 19200bps, stop bits is 1 bit and odd parity.

Wirte parameters setup completion code.

4-4-4 Communications from a Ladder Program

Typically network communications will last longer than 1 scan. The program must wait for the communications to finish before starting the next transaction. Port P1 (RS485), which can be a master, has two Special Relay contacts associated with it. One indicates "Port busy"(SP112), and the other indicates "Port Communication Error" (SP113).

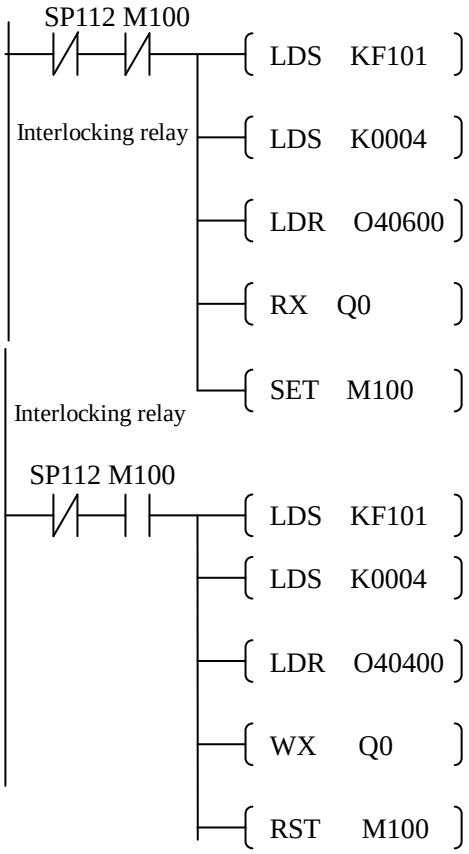
The example right shows the use of these contacts for a network master that only reads a device (RX). The "Port Busy" bit is on while the PLC communicates with the slave. When the bit is off the program can initiate the next network request. The "Port Communication Error" bit turns on when the PLC has detected an error. Use of this bit is optional. When used, it should be ahead of any network instruction boxes since the error bit is reset when an RX or WX instruction is executed.



4-4-5 Multiple Read and Write Interlocks

If you are using multiple reads and writes in the RLL program, you have to interlock the routines to make sure all the routines are executed. If you don't use the interlocks, then the CPU will only execute the first routine. This is because each port can only handle one transaction at a time.

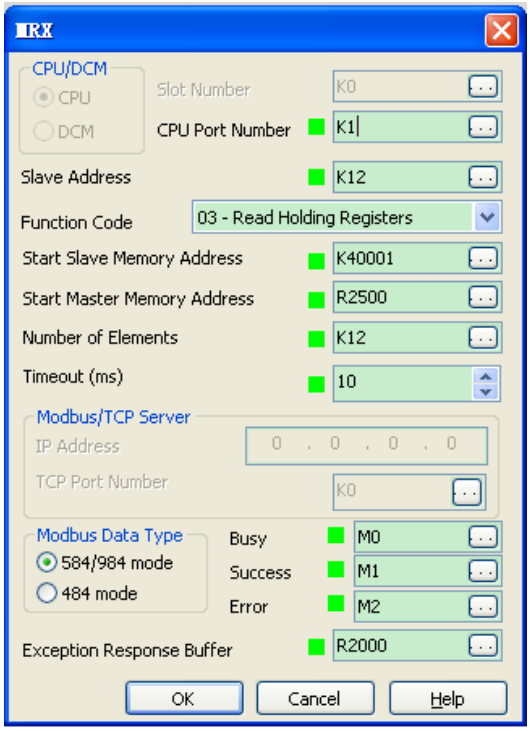
In the example on the right, after the RX instruction is executed, M100 is set. When the port has finished the communication task, the second routine is executed and M100 is reset.



4-4-6 MODBUS RTU protocol dedicatied communication instruction MRX/MWX

1. MODBUS read from network (MRX)

The Modbus Read from Network (MRX) instruction is used by the NK1 network master to read a block of data from a connected slave device and to write the data into R–memory addresses within the master. The instruction allows the user to specify the Modbus Function Code, slave station address, starting master and slave memory addresses, number of elements to transfer, Modbus data format and the Exception Response Buffer.



- **CPU port number:** specify NK1 port number, optional K0~K3，K0 is P0, K1 is P1, etc.
- **Slave address:** specify a slave station address (0~247). The parameters is invalid for P3 (Ethernet) port.
- **Function code:** the following modbus function codes are supported by the MRX instruction:

  01—— read a group of coils
  02—— read a group of inputs
  03—— read holding registers
  04—— read input registers
  07—— read exception status

- **Start slave memory address:** specifies the starting slave memory address of the data to be read.
- **Start master memory address:** specifies the starting memory address in the master where the data will be placed.
- **Number of elements:** specifies how many coils, input, holding registers or input registers will be read.
- **Timeout (ms):** specifies timeout to wait a communication partner response.
- **IP address:** only port P3 (Ethernet port), specifies the IP address of the communication partner.
- **TCP port number:** only port P3 (Ethernet port), specifies the TCP port number of the communication partner. (Note: when NK1 is a slave, its TCP address is fixed at 502).
- **MODBUS data format:** specifies modbus 584/984 or 484 data format to be used.
- **Communication implement flag:** the flag is set when start MRX instruction and is reset when communication is completed.
- **Communication completion flag:** the flag is reset when start MRX instruction and is set when communication is completed.
- **Communication error flag:** the flag is reset when start MRX instruction and is set when a communication error is encountered.
- **Exception response buffer:** specifies the master memory address where the exception response will be placed.

  Exception response code = 1 indicateds that the specified function code is not supported.

Exception response code = 2 indicates that the specified address or range is error.

Exception response code = 3 indicates that data is abnormal. Data that is prohibited to readand write may be specified in current operation mode.

Exception response code = 4 indicates that the communication process is abnormal.

1）MRX slave memory address

Function code	MODBUS data format	Slave address range
01— read coil	484 mode	1 ~ 999
	584/984 mode	1 ~ 65536
02— read coil	484 mode	1001 ~ 1999
	584/984 mode	10001 ~ 19999（5 digit）or 100001 ~ 165536（6 digit）
03 — read holding register	484 mode	4001 ~ 4999
	584/984 mode	40001 ~ 49999（5 digit） or 400001 ~ 465536（6 digit）
04—read input register	484 mode	3001 ~ 3999
	584/984 mode	30001 ~ 39999（5 digit）or 300001 ~ 365536（6 digit）
07 — read exception status	484 and 584/984 mode	N/A

2）MRX master memory address

Operand data type	NK1 range
Global inputs GI	GI0 ~ GI3777
Inputs I	I0 ~ I1777
Global ouputs GQ	GQ0 ~ GQ3777
Outputs Q	Q0 ~ Q1777
Control relays M	M0 ~ M3777
Stages S	S0 ~ S1777
Timer bits T	T0 ~ T777
Counter bits C	C0 ~ C777
Special relays SP	SP0 ~SP1777
Data registers R	R2000 ~ R7377 R10000~36777

3）MRX number of elements

Operand data type	NK1 range
R-memory R	R2000 ~ R7377 R10000~36777
constant K	Bits: 1~2000 Registers: 1~125

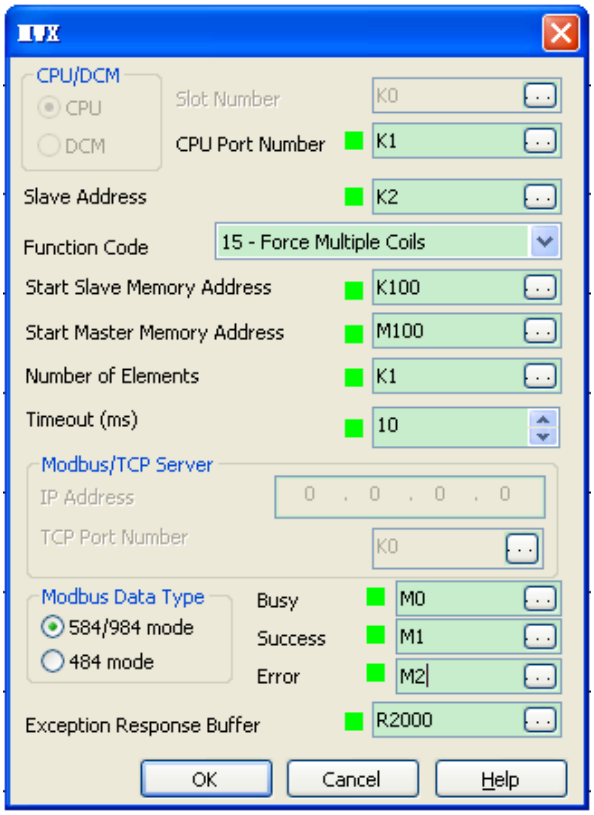
4）MRX exception response buffer

Operand data type	NK1 range
R-memory R	R2000 ~ R7377 R10000~36777

Note: when using MRX/MWX communication instruction, special relays SP110~SP117 will not work.

2. MODBUS wirte to network MWX

The Modbus Write to Network (MWX) instruction is used to write a block of data from the network masters's memory to Modbus memory addresses within a slave device on the network. The instruction allows the user to specify the Modbus Function Code, slave station address, starting master and slave memory addresses, number of elements to transfer, Modbus data format and the Exception Response Buffer.



- **CPU port number:** specify NK1 port number, optional K0~K3，K0 is P0, K1 is P1, etc.
- **Slave address:** specify a slave station address (0~247). The parameters is invalid for P3 (Ethernet) port.
- **Function code:** the following modbus function codes are supported by the MWX instruction:

  05—— force single coil
  06—— preset single register
  15—— force multiple coils
  16—— preset multiple registers

- **Start slave memory address:** specifies the starting slave memory address where the data will be written.
- **Start master memory address:** specifies the starting address of the data in the master that is to written to the slave.
- **Number of elements:** specifies how many consecutive coils or registers will be written to.
- **Timeout (ms）:** specifies timeout to wait a communication partner response.
- **IP address:** only port P3 (Ethernet port), specifies the IP address of the communication partner.
- **TCP port number:** only port P3 (Ethernet port), specifies the TCP port number of the communication partner. (Note: when NK1 is a slave, its TCP address is fixed at 502).
- **MODBUS data format:** specifies modbus 584/984 or 484 data format to be used.
- **Communication implement flag:** the flag is set when start MWX instruction and is reset when communication is completed.
- **Communication completion flag:** the flag is reset when start MWX instruction and is set when communication is completed.
- **Communication error flag:** the flag is reset when start MWX instruction and is set when a communication error is encountered.
- **Exception response buffer:** specifies the master memory address where the exception response will be placed.

  Exception response code = 1 indicateds that the specified function code is not supported.
  Exception response code = 2 indicates that the specified address or range is error.

Exception response code = 3 indicates that data is abnormal. Data that is prohibited to readand write may be specified in current operation mode.

Exception response code = 4 indicates that the communication process is abnormal.

1）MWX slave memory address

Function code	MODBUS data format	Slave address range
05－ force single coil	484 mode	1 ~ 999
	584/984 mode	1 ~ 65536
06－ force single coil	484 mode	4001 ~ 4999
	584/984 mode	40001 ~ 49999（5 digit）or 400001 ~ 465536（6 digit）
15－ force multiple coils	484 mode	1 ~ 999
	584/984 mode	1 ~ 65536
16－ preset multiple registers	484 mode	4001 ~ 4999
	584/984 mode	40001 ~ 49999（5 digit）or 400001 ~ 465536（6 digit）

2）MWX master memory address

Operand data type	NK1 range
Global inputs GI	GI0 ~ GI3777
Inputs I	I0 ~ I1777
Global outouts GQ	GQ0 ~ GQ3777
Outputs Q	Q0 ~ Q1777
Control relays M	M0 ~ M3777
Stages S	S0 ~ S1777
Timer bits T	T0 ~ T777
Counter bits C	C0 ~ C777
Special relays SP	SP0 ~SP1777
Data registers R	All

3）MWX number of elements

Operand data type	NK1 range
R-memory R	R2000 ~ R7377　R10000~36777
Constant K	Bits: 1~2000 Registers: 1~125

4）MWX exception response buffer

Operand data type	NK1 range
R-memory R	R2000 ~ R7377　R10000~36777

Note: when using MRX/MWX communication instruction, special relays SP110~SP117 will not work.

3. MRX/MWX example in KPPSoft

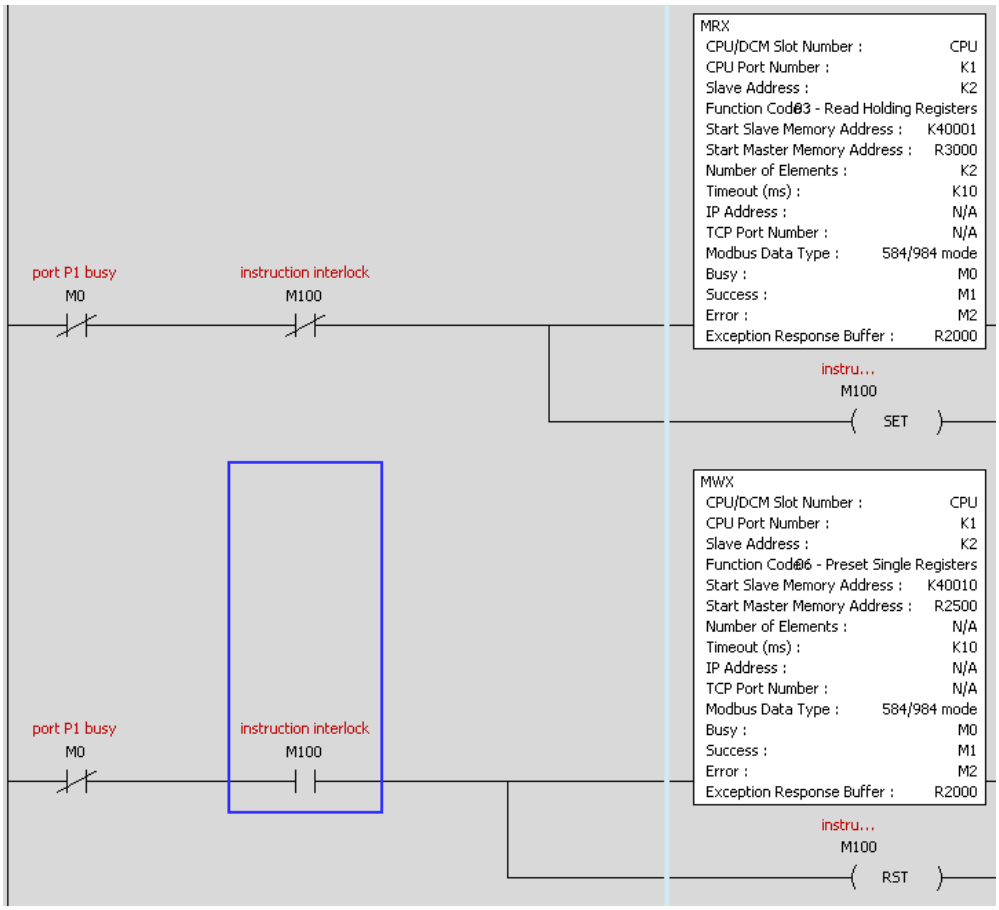
In the following program segment:

NK1 PLC communicates with the device which has RS-485 serial communication port and supports MODBUS RTU communication protocol through port P1.

The function code of MRX instruction is 03. MRX instruction is used to read out the data of holding registers 40001~40002 in slave station which station number is 2 and wirte them to R3000 and R3001 in NK1 PLC.

The function code of MWX instruction is 06. MWX instruction is used to wirte the data in R2500 of the master station to the holding registers R40010 of the slave station.

If you are using multiple reads and writes instructions in the RLL program, you need to interlock the routines to make sure all the routines are executed. If you don't use the interlocks, the CPU will only execute the first routine. This is because each port can only handle one transaction at a time. In the example, rungs 1 and 2 show that M100 will get set after the MRX instruction has been executed. When the port has finished the communication task, the second routine is executed and M100 is reset. If you're using RLLPLUS stage programming, you can put each routine in a separate program stage to ensure proper execution and switch from stage to stage allowing only one of them to be active at a time.



4-5 NK1 non-sequence serial communication

All NK1 PLC serial communication ports can be set to no-sequence communication mode and can connect to intelligent terminal with a serial communication port.

4-5-1 NK1 PLC supports non-sequence serial communication function

1. Non-sequence serial communication classification

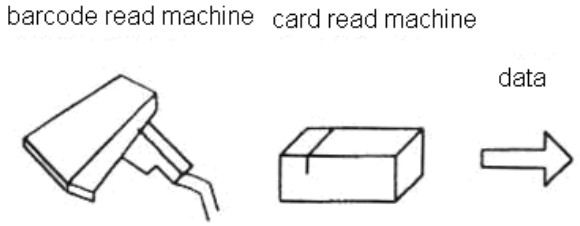
①Serial data receive…receive serial data, write data to registers of PLC.

②Serial data transmit…transmit data stored in registers of PLC.

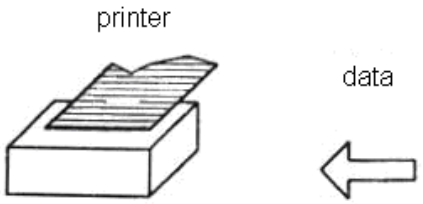
③Serial data receive/transmit… can transmit and receive serial data.

2. Devices can be connected

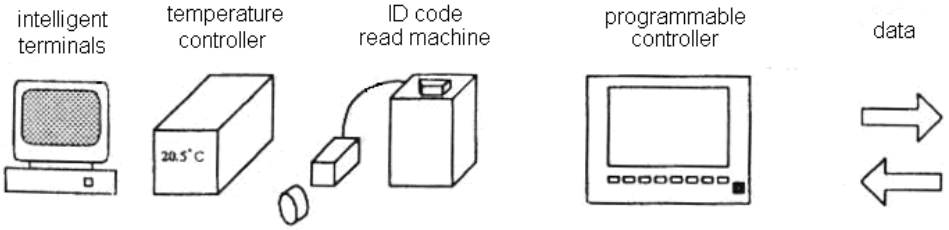
①Serial data receive: barcode read machine, card read machine.



②Serial data transmit: printer



③Serial data transmit/receive: intelligent terminals, temperature controller, ID code read machine, programmable controller.



3. Communication process control

When using non-sequence serial communication, there are two kinds of communication flow control functions: software flow control and hardware flow control.

①Software flow control

Use code X-ON (0x11H) and X-OFF (0x13H) to control communication process.

② hardware flow control

Use hardware signals RTS/CTS to control the transmission of data. All ports of NK1 PLC do not support hardware flow control.

4. Non-sequence communication mode supported by NK1 PLC

NK1 PLC supports the following four non-sequence communication mode.

Non-sequence communication mode	Set mode code	Remark
A type serial data receive mode(mode 61)	0x0061	Fixed format data receive
B type serial data receive mode (mode 62)	0x0062	Fixed format data receive
A type variable format serial data transmit and receive mode (mode 70)	0x0070	RTS control is invalid, X-ON/X-OFF control is invalid
B type variable format serial data transmit and receive mode (mode 71)	0x0071	RTS control is invalid, X-ON/X-OFF control is invalid

The following content is a brief description of these four kinds of non-sequence communication.

① A type serial data reception (mode 61)

It is a data receiving communication mode for some specific bar code read machine and other equipment.

STX	ASCII data （up to 127 bytes）	ETX	BCC

Data format:

BCC：STX（02H）→ETX（03H）, XOR

Transmission format：7 bits, even

Up to 127 ASCII characters

NK1 PLC sends ACK when it receives data normally, conversely sends NAK.

② B type serial data reception (mode 62)

It is also a data receiving communication mode for some specific bar code read machine and other equipment.

STX	ASCII data (up to 127 bytes)	ETX	BCC

Data format：

BCC：STX（02H）→ETX（03H）, XOR

Transmission format：7 bits, even

Up to 127 ASCII characters

NK1 PLC sends ACK when it receives data normally, conversely sends NAK.

③ A type variable format serial data transmission and reception (mode 70) (default mode)

In mode 70, data reception doesn't need end code. If PLC do not receive a new character, after waiting for a specified period of time (default is the communication time of 3 characters), automatically ends data reception; Of course, if the end code is received, it will end data reception.

ASCII data	Communication code 1	Communication code 2	Data format

Data format:

Data bits: 7 bits, 8 bits

Parity: none/ odd/ even

Stop bits 1 bit, 2 bits

Up to 128 bytes

Communication end code: 1 byte, 2bytes (any code except for 0) or no end code

In this mode, the communication software flow control code X-ON/X-OFF is invalid, and the communication hardware flow control mode RTS is also invalid.

④ B type variable format serial data transmission and reception (mode 71)

Mode 71 is basically same with mode 70, the only difference is that software flow control code X-ON/X-OFF is valid in mode 71.

ASCII data	Communication code 1	Communication code 2	Data format

In mode 71, NK1 data transmission process will be ended by X-OFF (0x13) signal. Once NK1 receives X-OFF (0x13) signal, it will stop transmitting data, it continues to transmit data until receiving the X-ON (0x11H) signal.

4-5-2 Set non-sequence protocol communication parameters

If you want to use non-sequence protocol communication function, you need to set communication parameters of the port. Communication parameters are stord in a group of registers.

Parameters	P0 special registers	P1 special registers	P2 special registers
Select communication protocol	R7650	R7662	R7674
Communication timeout, RTS ON delay time when RS-485 communication, port P2 wiring methods	R7651	R7663	R7675
Data bits, stop bit, parity, baud rate, response delay time	R7652	R7664	R7676
Communication mode	R7653	R7665	R7677
Start code	R7654	R7666	R7700
End code	R7655	R7667	R7701
The start register which stores the received data	R7656	R7670	R7702
Parameters setup completion flag	R7661	R7673	R7705

There are two methods to configure ports on NK1 CPU: by KPPSoftware and by setting special registers in user program.

The following example shows you how to set parameters of port P1:

Protocol : non-sequence communication mode 71（X-ON/X-OFF control is valid.）

Data format : 8 data bits /1 stop bits / no parity / 9600bps

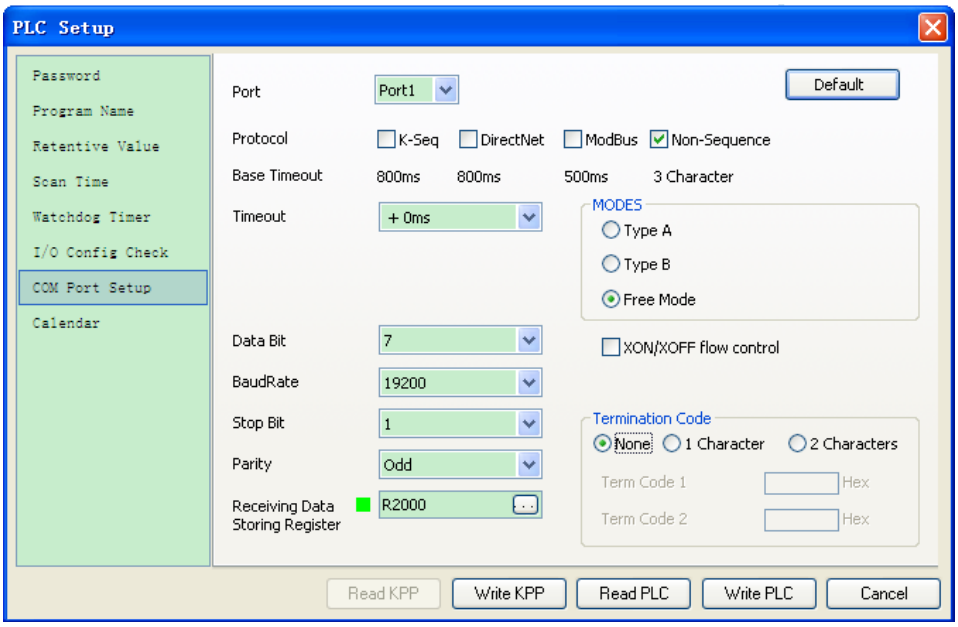
Register used to store received data : R2000

Response delay: 20ms

Communication end code: 0D

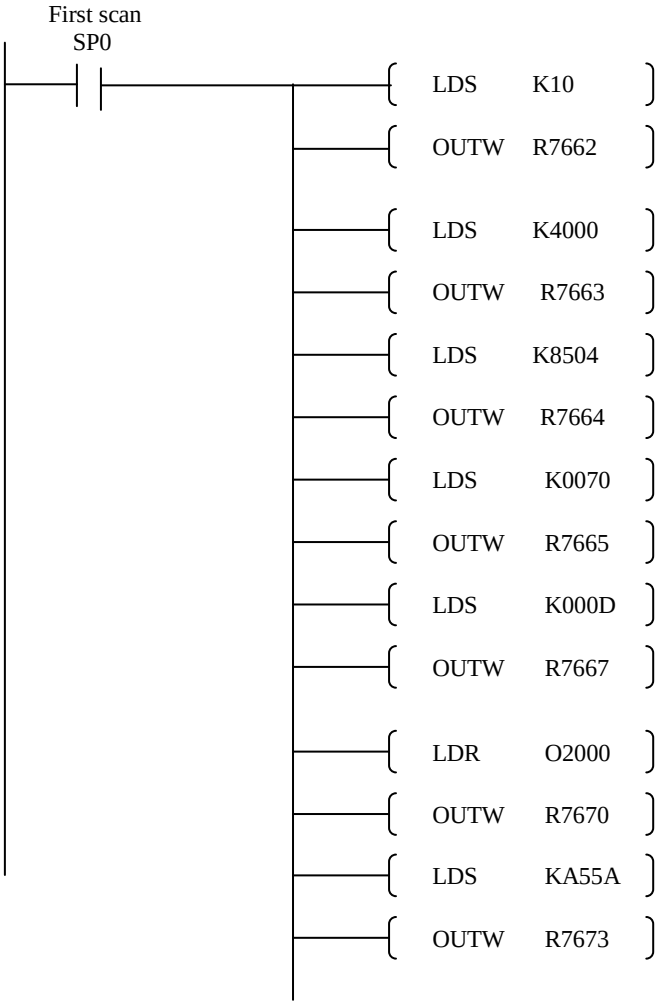
1) Configure port by KPPSoft software

In KPPSoft, as shown in the following diagram, choose PLC > PLC Setup > COM Port Setup, open PLC Setup window, select the communication port and setup the appropritate parameters, and then press the key [Write PLC], the setup of the port's parameters are finished.



Note: after a communication port is set to non-sequence communication by KPPSoft, the port will not be able to communicate with KPPSoft software. If the port which is connecting with KPPSoft is set to non-sequence communicatioon, the connection of NK1 PLC and KPPSoft will be interrupted immediately.

2) Configure port by user program



Select non-sequence communication mode. Port P1 will not be able to communicate with KPPSoft.

Set communication timeout.

8 data bits/ 1 stop bits/ no parity/ 9600bps, response delay time is 20ms.

Select mode 70.

An end code: 0D

Specify the starting R-memory location where the received data will be written.

Write parameters setup completion code.

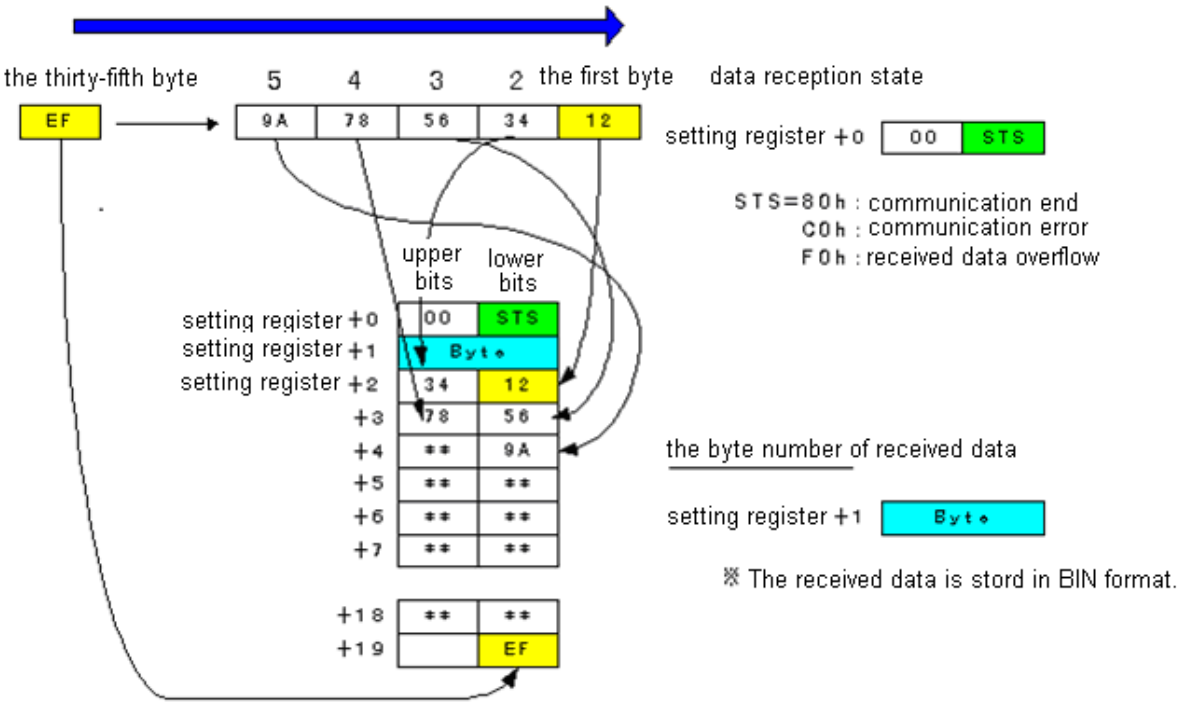
4-5-3 NK1 PLC non-sequence communication operation

After communication parameters setup using the above method, the communication port is in non-sequence communication mode.

No CPU ladder logic is required to support non-sequence data reception operation. When using data transmitting operation, CPU ladder logic is required to initiate the requests by using WX instruction.

1. Store format of received data

Received data is stored in a bank of registers, the data is stored in the following format:



- The lower bits of the first setting register is used to store data reception state (STS), the upper bits are 00;
  STS = 00H, indicates NK1 is well prepared to receive new data;
  STS = 80H, indicates data reception has been completed without errors;
  STS = C0H, indicates data reception encountered an error;
  STS = F0H, indicates data reception encountered overflow error.

- Setting register +1 stores byte number of received data (BIN), up to 128 bytes (80h);
- Setting register +2 is the first register to store received data.

2. Non-sequence communication operation in mode 61/62

No CPU ladder logic is required to support non-sequence data reception operation. You only need to setup communication parameters of corresponding port and reset some registers. The following example is a data reception operation in mode 61/62, the communication port is P1.

(1) Write 10h to R7662, choose non-sequence communication;
(2) Set communication timeout in R7663 (if communication port is P2, the wiring method also needs to be set.);
(3) Set transmission rate and response delay time parameter in R7664; (Data bits and parity bit are fixed)
(4) Write 61 or 62 in R7665, chose communication mode;
(5) Write the starting register number in R7670;
(6) Wirte setup completion code A55A (H) in R7673 to complete the setup;
(7) Reset the lower byte of the staring register;
(8) Receive data and store it automatically in registers designated by the prescribed format;
(9) Send ACK code when the end of normal data reception; Send NAK code when the abnormal end of data reception.
(10) The user program processed the received data. After the processing of received data, please reset the data reception state byte (STS).when the value of the status byte is not 0, the system will not receive the next frame data.

Data reception in mode 61/62 completes once.

3. Non-sequence communication operation in mode 70/71

In mode 70/71, NK1 communication port can receive and transmission data. Like with mode 60/61, No CPU ladder logic is required to support non-sequence data reception operation. Data transmission operation requires special user program.

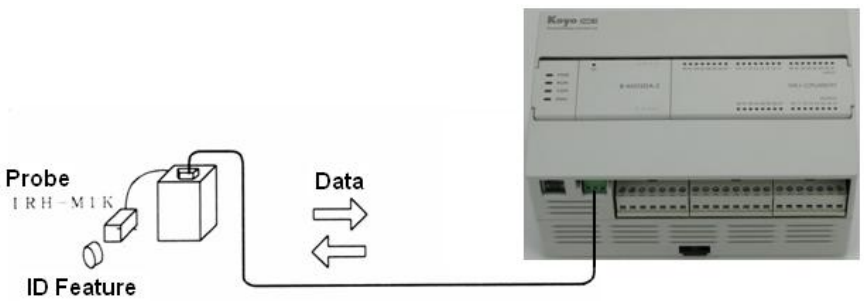
Here is an example:

(1) Function to be achieved

Connect port P1 with ID read machine, NK1 sends command to ID read machine, reads ID code and stores it in registers.

(2) System componets

ID read machine（IRI—KHA6—4H）

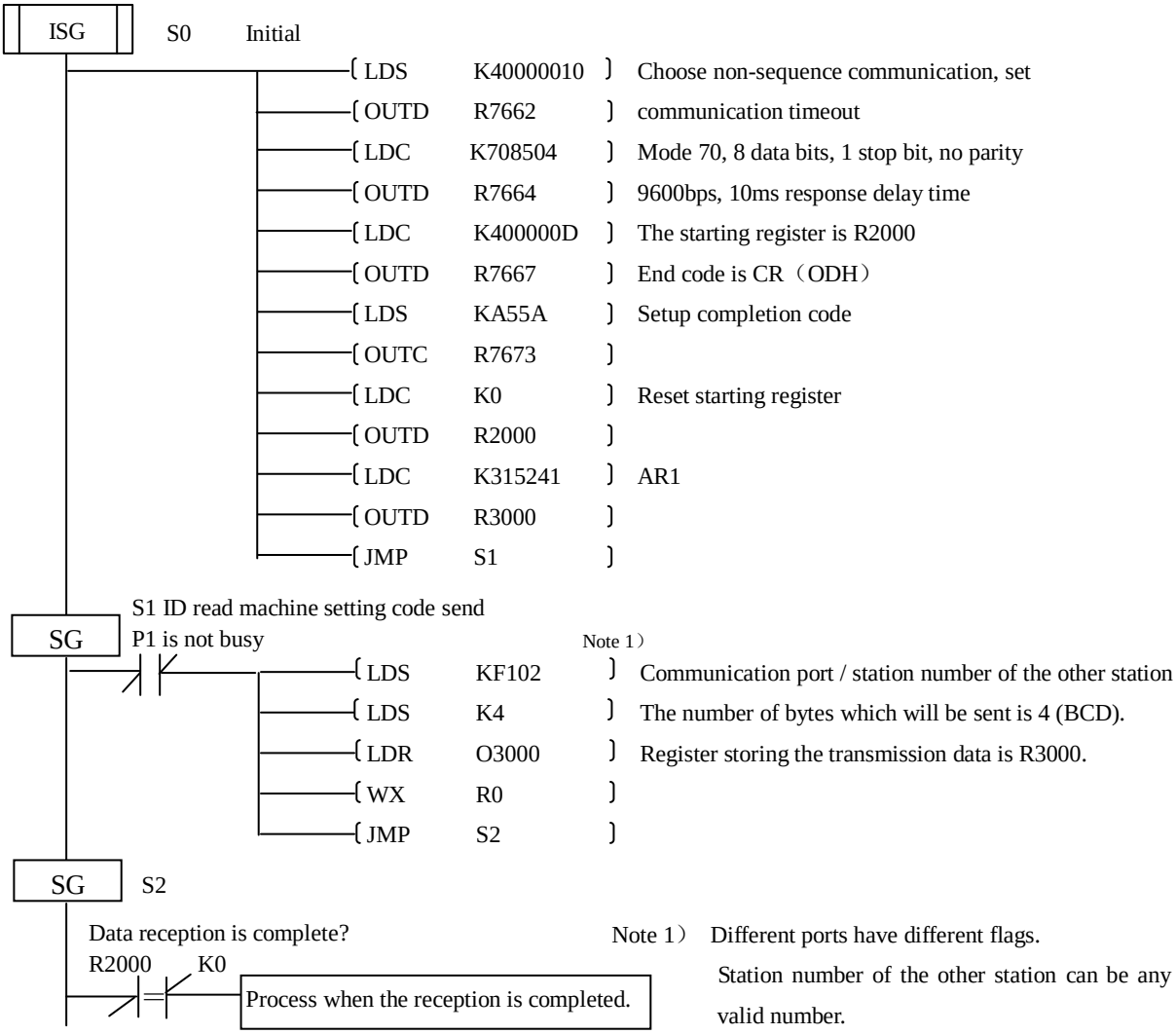


(3) Conditions

① R7662 = 10 : choose non-sequence communication
② R7663 = 4000: set communication timeout，
③ R7664 = 8504 : 8 data bits/ 1 stop bit/ no parity/ 9600bps/ 10ms response delay time

④ R7665 = 70 : variable format A (receive/ send)
⑤ R7667 = 0D : end code is CR（0DH）, 1 byte
⑥ R7670 = 400 : stores received data in registers starting from R2000

(4) User program



Note 1） Different ports have different flags.
Station number of the other station can be any valid number.

4-5-4 NK1 PLC non-sequence serial communication instruction (ASCII IN/OUT feature）

ASCII IN/OUT instructions supported by NK1 PLC are:

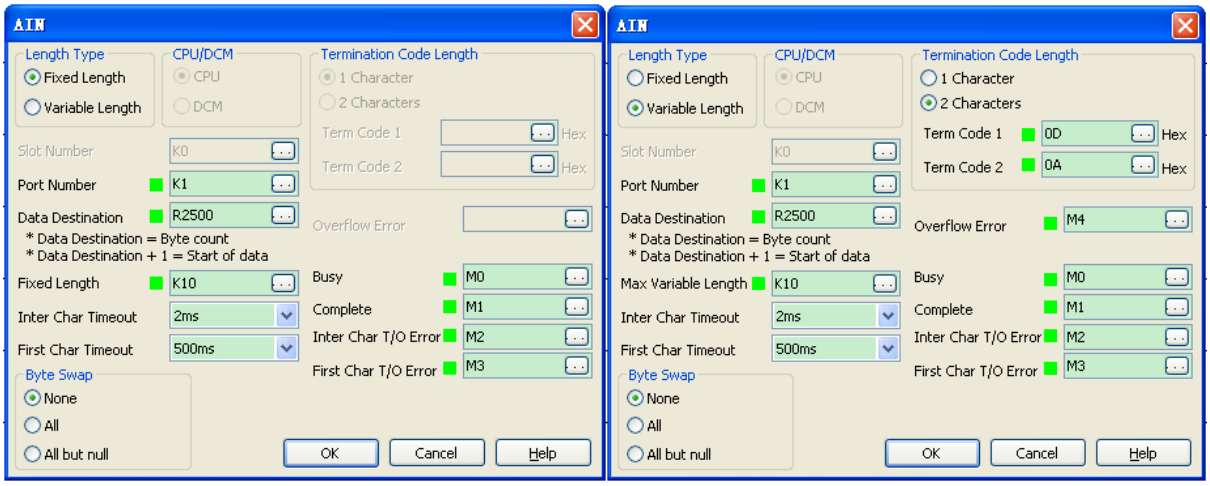
ASCII input： AIN

ASCII print from R-memory: PRINTV

For the purpose to use non-sequence communication instructions, you need first set the port to non-sequence communication mode (mode 70 or mode 71).

When using non-sequence communication instructions, SP54 will be ON when communication parameters or mode is set incorrectly.

1. AIN



The ASCII Input instruction allows the CPU to receive ASCII strings through the specified communication port and places the string into a series of specified R-memory registers. The ASCII data can be received as a fixed number of bytes or as a variable length string with a specified termination character(s). Other features include Byte Swap preferences, Character Timeout, and user defined flag bits for Busy, Complete and Timeout Error.

AIN configuration

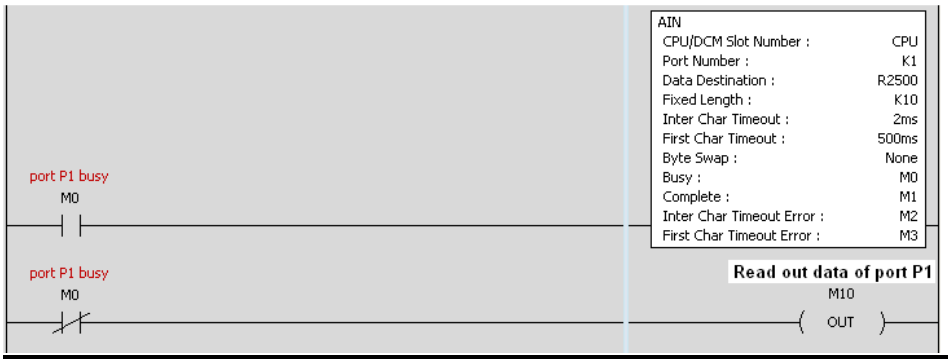
- **Length Type:** choose fixed length or variable length.
- **Port number:** choose communication port, K0 is P0, K1 is P1, K2 is P2.
- **Data Destination:** specifies where the ASCII string will be placed in R-memory.
- **fixed Length:** specifies the length, in bytes, of the fixed length ASCII string the port will receive.
- **Max variable Length:** specifies, in bytes, the maximum length of a variable length ASCII string the port will receive, up to 128 bytes.
- **Inter Char Timeout:** if the amount of time between incoming ASCII characters exceeds the set time, the timeout error bit will be set. No data will be stored at the data destination R-memory location. The timout error bit will reset when the AIN instruction permissive bits are disabled. None selection disables this feature.

- **First Char Timeout:** if the amount of time between incoming ASCII characters exceeds the set time, the timeout error bit will be set. No data will be stored at the data destination R-memory location. The timeout error bit will reset when the AIN instruction permissive bits are disabled. None selection disables this feature.
- **Byte Swap:** swaps the high-byte and low-byte within each R-memory register of the variable length ASCII string.
- **Busy:** is ON while the AIN instruction is receiving ASCII data.
- **Complete:** when select fixed length type, is set once the ASCII data has been received for the specified fixed length and reset when the AIN instruction permissive bits are disabled; when select variable length type, is set once the ASCII data has been received up to the termination code characters. It will be reset when the AIN instruction permissive bits are disabled.
- **Inter Char T/O Error:** the amount of time between incoming ASCII characters exceeds the set time, the Timeout Error bit will be set. No data will be stored at the Data Destination R–memory location. The Timeout Error bit will reset when the AIN instruction permissive bits are disabled. None selection disables this feature.
- **First Char T/O Error:** if the amount of time from when the AIN is enabled to the time the first character is received exceeds the set time, the specified First Character Timeout bit will be set. The bit will reset when the AIN instruction permissive bits are disabled. None selection disables this feature.

  Additionally, when selecting a "variable length" type, there are two parameters need to be set:
- **Termination Code:** is set once the ASCII data has been received up to the termination code characters. It will be reset when the AIN instruction permissive bits are disabled.
- **Overflow Error:** is set when the ASCII data received exceeds the Maximum Variable Length specified.

AIN fixed length example

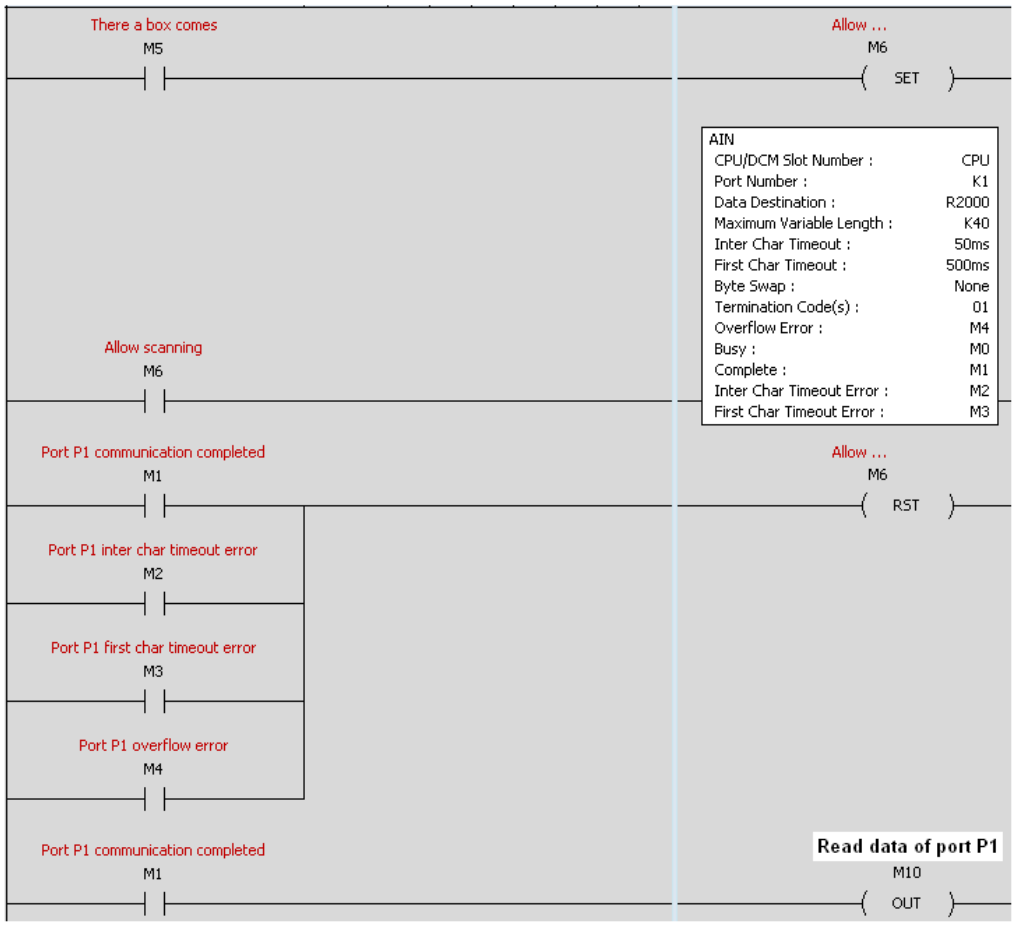
In the following program, NK1 PLC reads data of port P1 continuously and stores them to registers starting from R2500, read 10 bytes each time and the various timeout time is not critical.



AIN variable length example

AIN variable length example used to read the barcodes on boxes.

Assume that the maximum length of barcode is 40 characters，0DH is the completion code. When the received characters over 40 characters, M4 as the overflow flag is set. The received barcode data is stored in the register bank starting from R2000.



The following table lists the function memory rangs of AIN instruction:

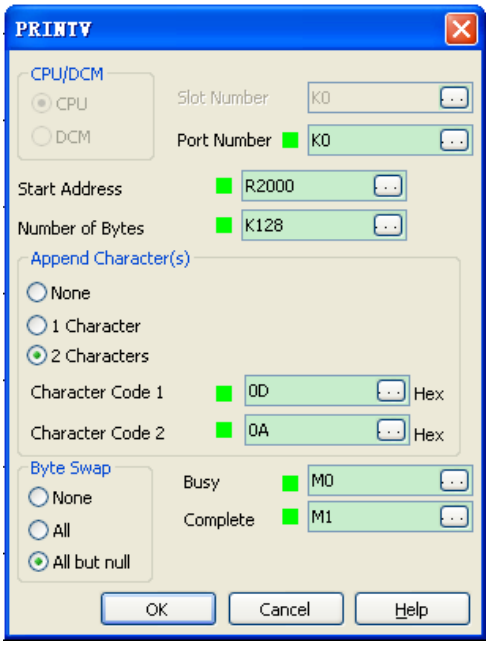
Parameters	NK1 range
Data destination	All R-memory
Fixed length（number of bytes）	K1~128
Maximum variable length（number of bytes）	K1~128
Bits: busy, complete, timeout error, overflow	M0~M3777

The following table lists the associated special relays of AIN instruction:

Discrete bit flages	Description
SP54	On when the communication port is not set to non-sequence or when instruction can not be executed correctly.
SP110	On when port P0 is communicating with another device.
SP111	On when port P0 has experienced a communication error.
SP112	On when port P1 is communicating with another device.
SP113	On when port P1 has experienced a communication error.
SP114	On when port P2 is communicating with another device.
SP115	On when port P2 has experienced a communication error.

2. PRINTV

The ASCII Print from R–memory instruction will send an ASCII string out of the designated communications port from a specified series of R–memory registers for a specified length in number of bytes. Other features include user specified Append Characters to be placed after the desired data string for devices that require specific termination character(s), Byte Swap options, and user specified flags for Busy and Complete.



- **Port Number:** choose communication port, K0 is P0, K1 is P1, K2 is P2.
- **Start Address:** specifies the begining of series of R–memory registers that contain the ASCII string to print.
- **Number of Bytes:** specifies the length of the string to print.
- **Append character(s):** specifies ASCII characters to be added to the end of the string for devices that require specific termination characters
- **Byte Swap:** swaps the high–byte and low–byte within each R–memory register of the string while printing.
- **Busy:** will be ON while the instruction is printing ASCII data.
- **Complete:** will be set once the ASCII data has been printed and reset when the PRINTV instruction permissive bits are disabled.

The following table lists the function memory rangs of PRINTV instruction:

Parameters	NK1 range
Start address	All R-memory
Number of bytes	K1~128
Bits: busy, complete	M0~M3777

The following table lists the associated special relays of AIN instruction:

Discrete bit flags	description
SP54	On when the communication port is not set to non-sequence or when instruction can not be executed correctly.

4-6 NK1 Ethernet communication

4-6-1 Communication protocols supported by Ethernet port of NK1 CPU

40-point NK1 PLCs have a 10M/100M adaptive standard Ethernet port, communication protocols supported by the port are:

Protocol	Ethernet port	
	RUN	STOP
MODBUS TCP	○	○
ECOM UDP	○	○
SMTP	○	○
DHCP	○	○
HAP+UDP	○	○
Programming software dedicated	○	○

○: support the connection

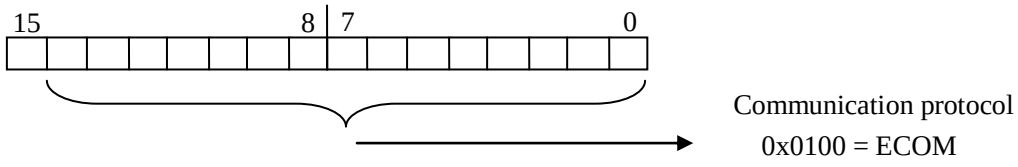
4-6-2 Parameters setup registers

The following table lists these registers:

Port P3 special registers	Parameters	The factory default value in register
R7706	Communication protocol, fixed at 0X0100, ECOM protocol	0X0100
R7707	Node number	0X0001
R7710	Specify IP address assignment	1
R7711	IP address low bits	0X0107
R7712	IP address high bits	0XC0A8
R7713	Subnet mask (MASK) address low bits	0XFF00
R7714	Subnet mask (MASK) address high bits	0XFFFF
R7715	GateWay address low bits	0X0101
R7716	GateWay address high bits	0XC0A8
R7717	Parameters setup completion flag	0

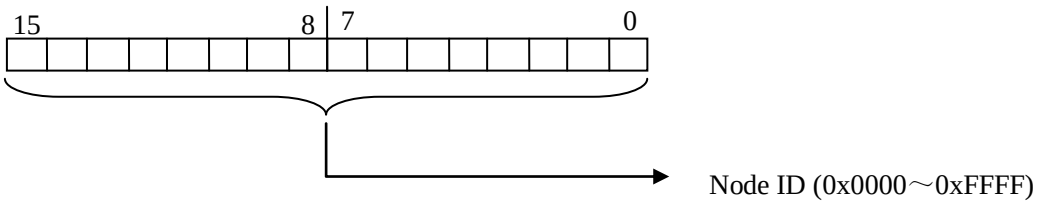
1. Communication protocol selection register R7706

Select protocol for NK1 Ethernet communication port which supports only ECOM protocol currently. The data in register is fixed at 0x0100. Note that NK1 ECOM protocol includes the supporting of MODBUS TCP/IP, DHCP, SMTP, KPPSoft dedicated communication protocol and the like.



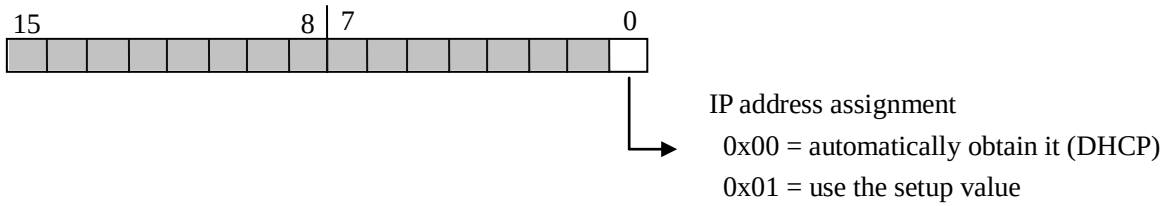
2. Node ID setup register R7707

The register is used to set node ID of Ethernet port, the range is 0x0000~0xFFFF (0~65535).



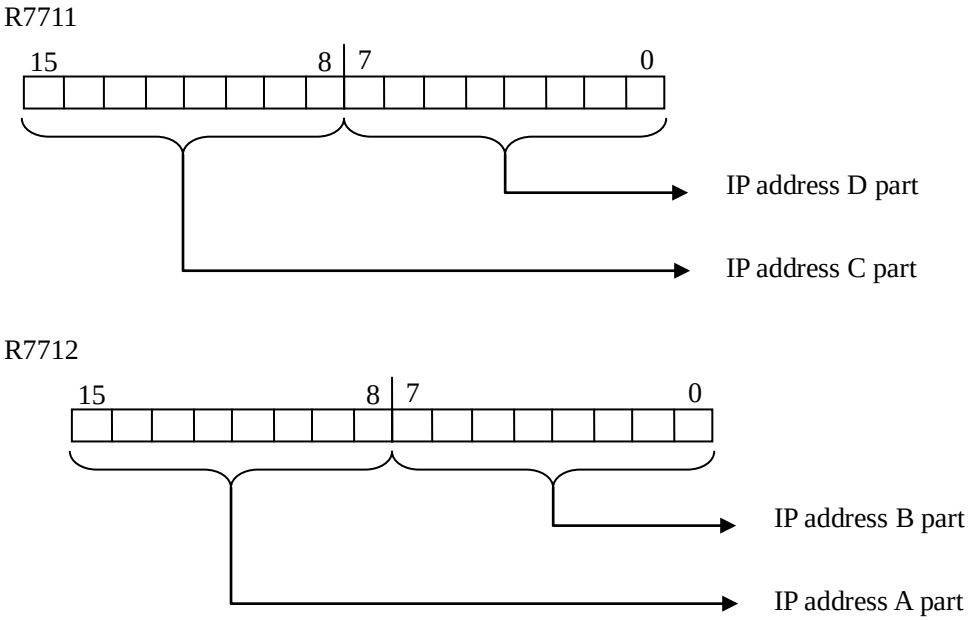
3. IP address assignment method setup register R7710

The register is used to select IP address assignment method, you can choose to automatically obtain DHCP or manually assign it.



4. IP address assignment registers R7711, R7712

Manually assign IP address. The four segment number of IP address is converted into a HEX number and stored in R7711 and R7712. Suppose IP address is A.B.C.D, the setting is R7711= CD, R7712=AB.



For example, if IP address is 192.168.11.1, the address is converted as:

A = 192（0xC0） B = 168(0xA8) C = 11 (0x0B) D = 1(0x01)

The values are stored as：

R7711 = 0x0B01 R7712 = 0xC0A8

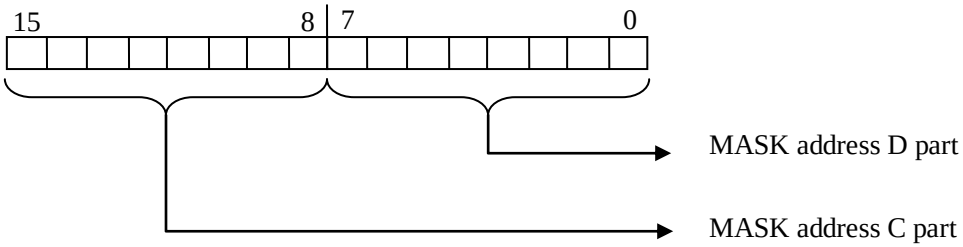
5. Subnet mask (MASK) address assignment registers R7713，R7714

Manually assign subnet mask address. The four segment number of MASK address is converted into a

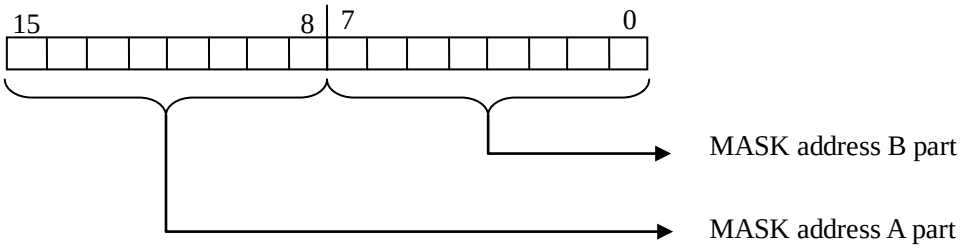
HEX number and stored in R7713 and R7714.

Suppose MASK address is A.B.C.D, the setting is R7713= CD, R7714=AB.

R7713



R7714



For example, if MASK address is 255.254.253.0, the address is converted as:

A = 255（0xFF） B = 254(0xFE) C = 253 (0xFD) D = 0(0x00)

The values are stored as：

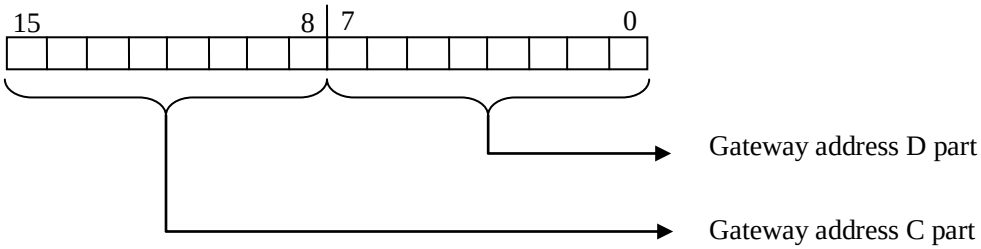
R7713 = 0xFD00 R7714 = 0xFFFE

6. Gateway address assignment registers R7715，R7716

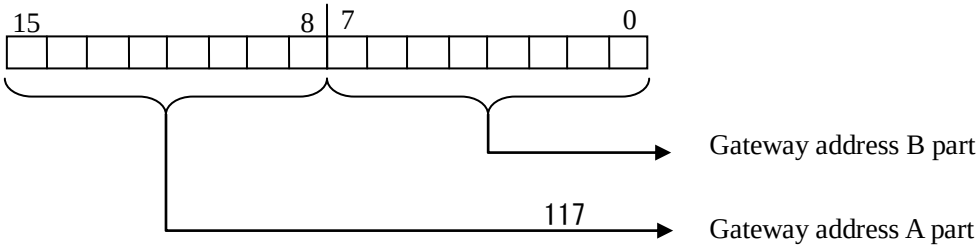
Manually assign gateway address. The four segment number of gateway address is converted into a HEX number and stored in R7715 and R7716.

Suppose gateway address is A.B.C.D, the setting is R7715= CD, R7716=AB.

R7715



R7716



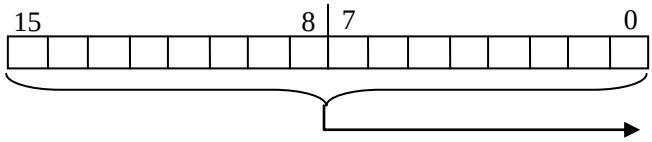
For example, if gateway address is 192.168.11.9, the address is converted as:

A = 192（0xC0） B = 168(0xA8) C = 11 (0x0B) D = 9(0x09)

The values are stored as：

R7715 = 0x0B09 R7716 = 0xC0A8

7. Ethernet communication parameters setup completion flag register R7717



Parameter setup completion code is 0xA55A.

If the parameters are set correctly, it becomes 0x5555.

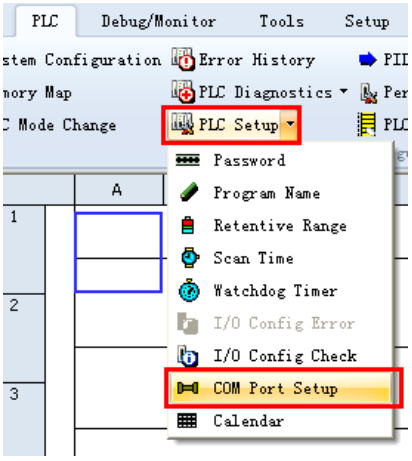
If the parameter is set incorrectly, the number of the register which stores the incorrect parameter will be written in this register.

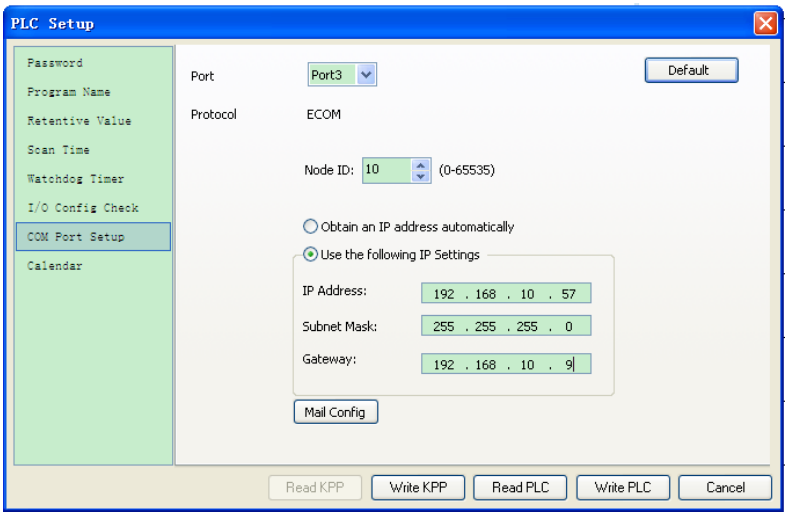
4-6-3 NK1 CPU Ethernet communication port configuration

There are two ways to configure ports on NK1 CPU:

1) Configure port by KPPSoft software:

In KPPSoft, as shown on the right, choose PLC > PLC Setup > COM Port Setup, open PLC Setup window, select the communication port and setup the appropritate parameters, and then press the key [Write PLC], the setup of the port's parameters are finished.





2) Configure port by setting special registers in user program

Assuming the setting parameters are:

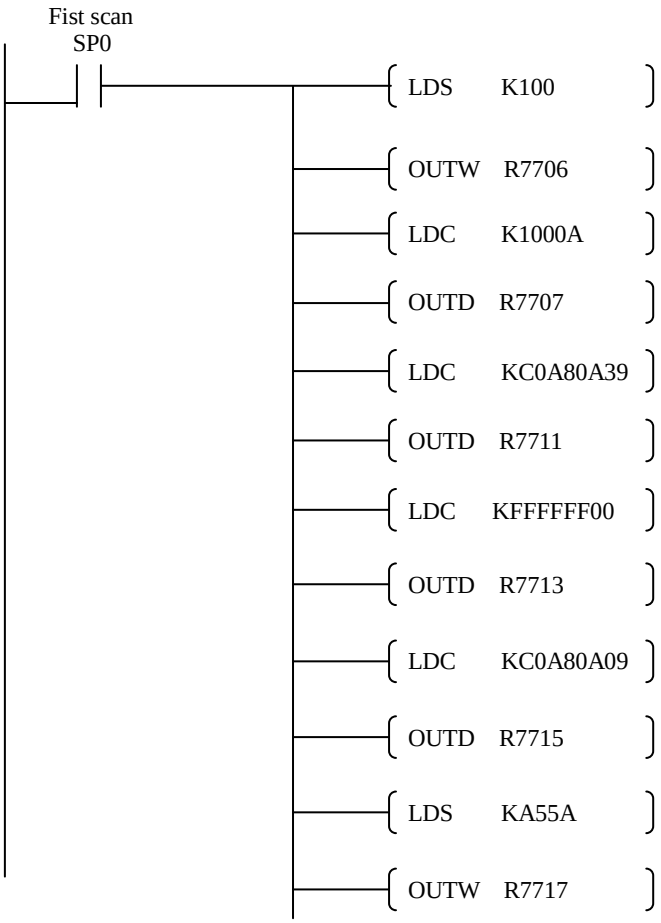
Node ID: 10;

IP address: 192.168.10.57; manually assign IP address;

MASK address: 255.255.255.0

Gateway: 192.168.10.9.

The setup program is:



4-6-4 NK1 Ethernet port MODBUS TCP communication

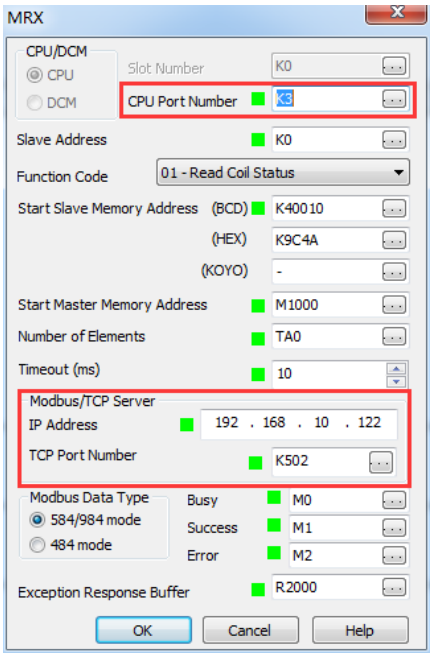
The Ethernet port of NK1 support the MODBUS TCP communication (Master/Slave). For Master, you can use MRX/MWX instruction to read data from MODBUS TCP Slaves or write data to MODBUS TCP Slaves. for Slave The other MODBUS TCP Master can read or write the data of the NK1 through the port.

When use the MRX/MWX instruciton through the Ethernet port ,the usage of the instruction is just the same as the usage when use the instruction through the serial port except the following three points.

1) the communication port number,please use K3;

2) please set the IP address and TCP Port Number(normally is K502);

3) All communication units in the same netware must be in the same network segment.



4-6-5 NK1 Ethernet port MODBUS UDP communication

The Ethernet port of NK1 support the ECOM UDP communication (Master/Slave). For Master, you can use RX/WX instruction to read data from ECOM UDP Slaves or write data to ECOM UDP Slaves. for Slave The other ECOM UDP Master can read or write the data of the NK1 through the port.

The usage of the RX/WX instructions within the ECOM UDP communication is just the same as the usage of the RX/WX instructions when using in a serial port(For detail , please see the Chapter4-2-4) except the following several points.

1. Please use F3 as the port flag in the communication program.(.it means use P3 Ethernet port)
2. The slave-node number in the communication program please use BIN format. But when you set the Ethernet port-to-port parameter in the tool soft **KewNetEdit**, please set the slave-node number in BCD format.
3. Inorder to use the NK1 Ethernet port as ECOM UDP communication Master,you must use the tool soft **KewNetEdit** to set the Ethernet port for using the RX/WX instructions.
   NOTE: The tool soft **KewNetEdit** version need V2.0.2.3 or later.
4. Using NK1as ECOM UDP communication Master to communication with theECOM UDP slaves. The numbers of the slaves must less then 30.
5. The Master and Slaves of one ECOM UDP communication netware must be int the same network segment

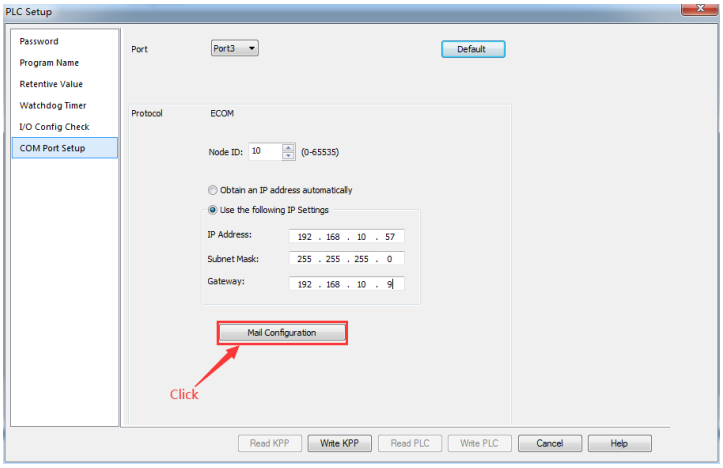
NOTE: The ECOM UDP protocol is the special communication protocol used by KOYO PLCs with Ethernet port.

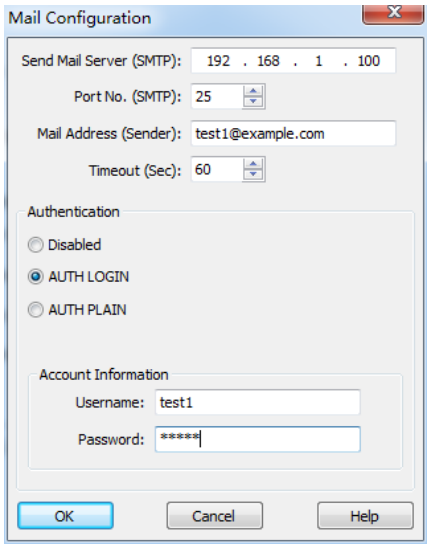
4-6-6 NK1 Ethernet port Email sending funtion

The Ethernet port of NK1 series supports the SMTP email delivery function. You can use the instruction EMAIL to send the email form NK1

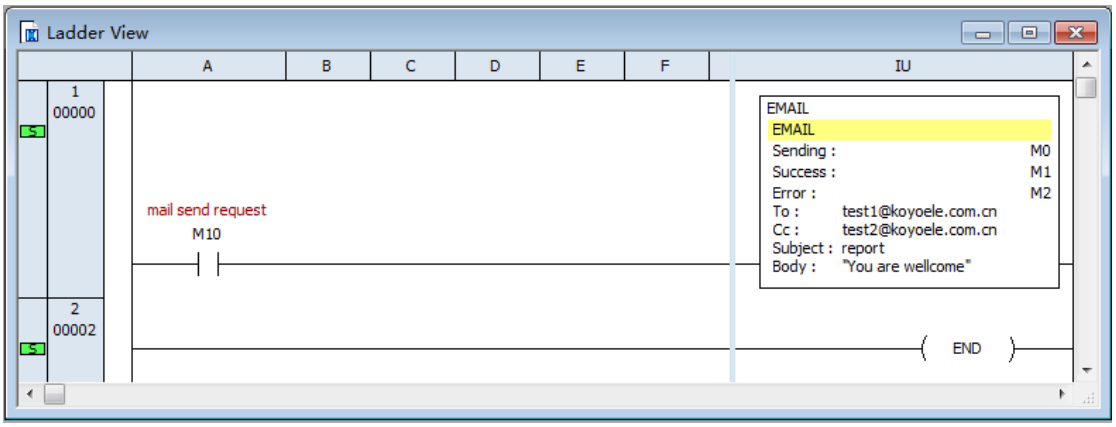
Before using the EMAIL instruction in the user program to send email, you must set the email parameter of the NK1 Ethernet port through the KPPSoft firstly. Of course, the essential parameter of the Ethernet prot like the node ID number, IP address,SubMask,and Gateway must be setup also.

In KPPSoft, you can click the [Mail configuration] in the [PLC Setup] window to open the [Mail configuration] window to set the email parameter.





After all parameter of the Ethernet port have been set, you can use the EMAIL instruction in the user program to achieve the sending email function.
Following is the sample program of sending email. Once M10 is ON, The program will send the message "You are wellcome" to two emailboxes for one time.
It is so simple!



4-6-7 NK1 Ethernet port non-sequence communication function

The Ethernet port in NK1 will support the non-sequence commnication function when the system fireware in it is V20171212 or later.
The NK1 will support the following three mode non-sequence commnication function.
1. UDP mode non-sequence commnication function.
2.TCP client mode non-sequence commnication function.
3.TCP server mode non-sequence commnication function.
You can select one mode of the three mode to achieve the special non-sequence commnication function.

The non-sequence commnication function is been included in the ECOM UDP protocol. And you need not to select the protocol specially when you want to use the NK1 Ethernet non-sequence commnication function.
For detail usage of the function,please see the document:
<NK1 Ethernet non-sequence commnication function >

4-7 Serial communication-related special relays

Each serial communication port of NK1 has two special relay contacts associated with it. One indicates "Port busy", and the other indicates "Port Communication Error".

The example right shows the use of these contacts for a network master that only reads a device (RX). The "Port Busy" bit is on while the PLC communicates with the slave. When the bit is off the program can initiate the next network request. The "Port Communication Error" bit turns on when the PLC has detected an error. Use of this bit is optional. When used, it should be ahead of any network instruction boxes since the error bit is reset when an RX or WX instruction is executed.

The following table lists all special relay contacts associated with communication.

Special relays	Description
SP54	On if there is parameters setup error of any one of communication port.
SP110	On when port 0 is communicating with another device and port 0 is master station.
SP111	On when port 0 has encountered a communication error and port 0 is master station.
SP112	On when port 1 is communicating with another device and port 0 is master station.
SP113	On when port 1 has encountered a communication error and port 0 is master station.
SP114	On when port 2 is communicating with another device and port 0 is master station.
SP115	On when port 2 has encountered a communication error and port 0 is master station.

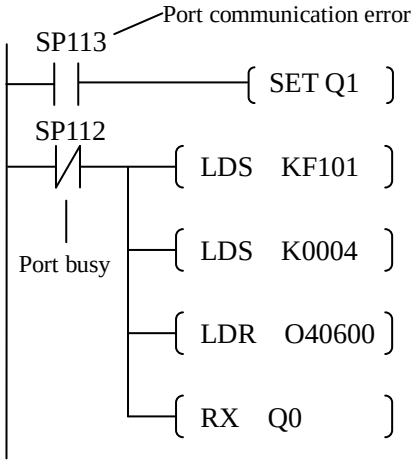
Note: the executing of MRX/MWX instructions does not affect the port busy contact and the communication error contact.

4-7-1 The subsequent processing after a communication error

For example, when port P1 is a master, it will affect two special relays SP112 and SP113.

Typically network communications will last longer than 1 scan. The program must wait for the communications to finish before starting the next transaction.

The example right shows the use of the two contacts for a network master. SP112 is on while the PLC communicates with the slave. When the bit is off the program can initiate the next network request. SP113 bit turns on when the PLC has detected an error. Use of this bit is optional. When used, it should be ahead of any network instruction boxes since SP113 is reset when port P1 excutes an RX or WX instruction.



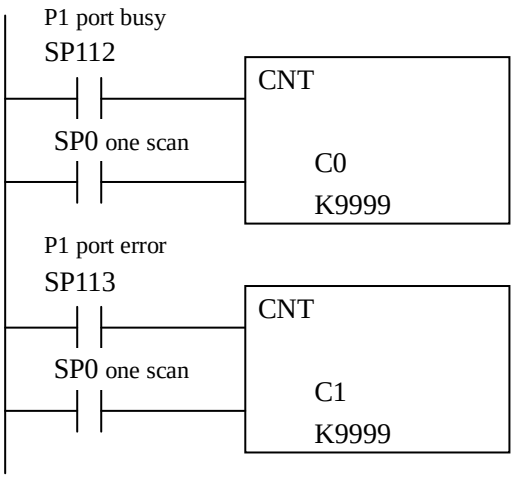
4-7-2 How to count the communication error

For example ,There are two special relays for the P1(RS-485 port). One is SP112("Port busy"), and the other is SP113("Port Communication Error").

The sample program in the right shows how to use the two special relays in your user program.

The counter C0 counts the times of communicaiton of WX/RX instruction through P1 port. The counter C1 counts the times of communicaiton error of WX/RX instruction through P1 port.

Theoretically, C0 is counting ,and C1 will keep 0. But in fact, for the reason of electromagnetic interference or nosie, C1 is usually with some number. You can get the correctness of the communication by comparing the data in C0 and C1.

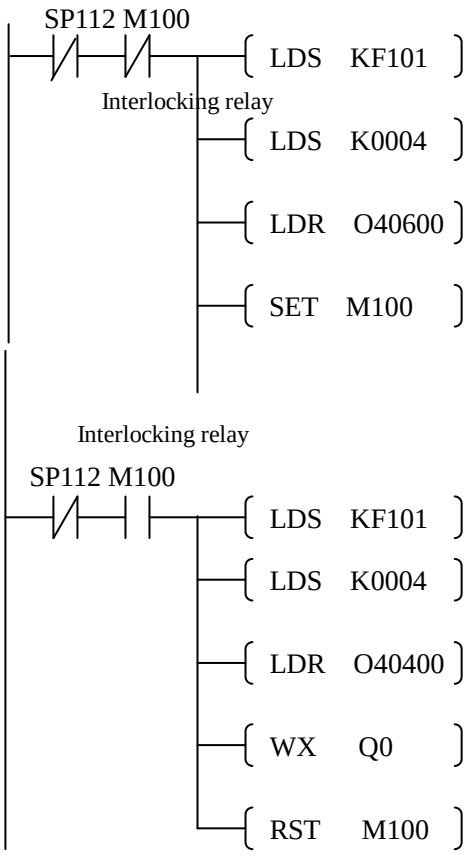


4-7-3 Multiple read and write interlocks

If you are using multiple reads and writes in the RLL program, you have to interlock the routines to make sure all the routines are executed. If you don't use the interlocks, the CPU will only execute the first routine. This is because each port can only handle one transaction at a time.

In the example on the right, after the RX instruction is executed, M100 is set. When the port has finished the communication task, the second routine is executed and M100 is reset.

If you're using RLLPLUS stage programming, you can put each routine in a separate program stage to ensure proper execution and switch from stage to stage allowing only one of them to be active at a time.



Chapter 5High-speed input and pulse output feature

NK1 CPUs have built-in high-speed I/O processing circuit. Its main feature is:

1) 5 channels single-phase high speed counters. (20-point model has 4 channels.)
2) 3 channels quadrature encoder inputs to measure counts and clockwise or counter-clockwise direction, counts up or down. Single-phase and two-phase may be used in combination. (20-point model has 2 channels.)
3) 8 channels high-speed interrupt inputs (can be used simultaneously with high-speed counters). (20-point model has 4 channels.)
4) 8 channels pulse catch (can be used simultaneously with high-speed counters). (20-point model has 4 channels.)
5) 1 channel timed interrupt (independent).
6) All high-speed inputs can be set to filter inputs.
7) 2 channels pulse catch (CW/CCW, direction/ pulse), can also be set to PWM output.
8) The data format BCD and HEX of pulse number are both supported by high-speed counter and pulse catch.

5-1 High-speed I/O terminal

I/O definition number	Function
I0~I5 (20-point model: I0~I3)	Single-phase, two-phase high-speed counter inputs
I6~I7	Interrupt input for pulse output
I10~I17 (20-point model: I10~I03)	External interrupt, pulse catch, high-speed counter external reset, count prohibit inputs
Q0~Q3	Pulse outputs (Q0 and Q1 is in a bank; Q2 and Q3 is in a bank)

5-2 Choose and setup the HSIO mode of inputs

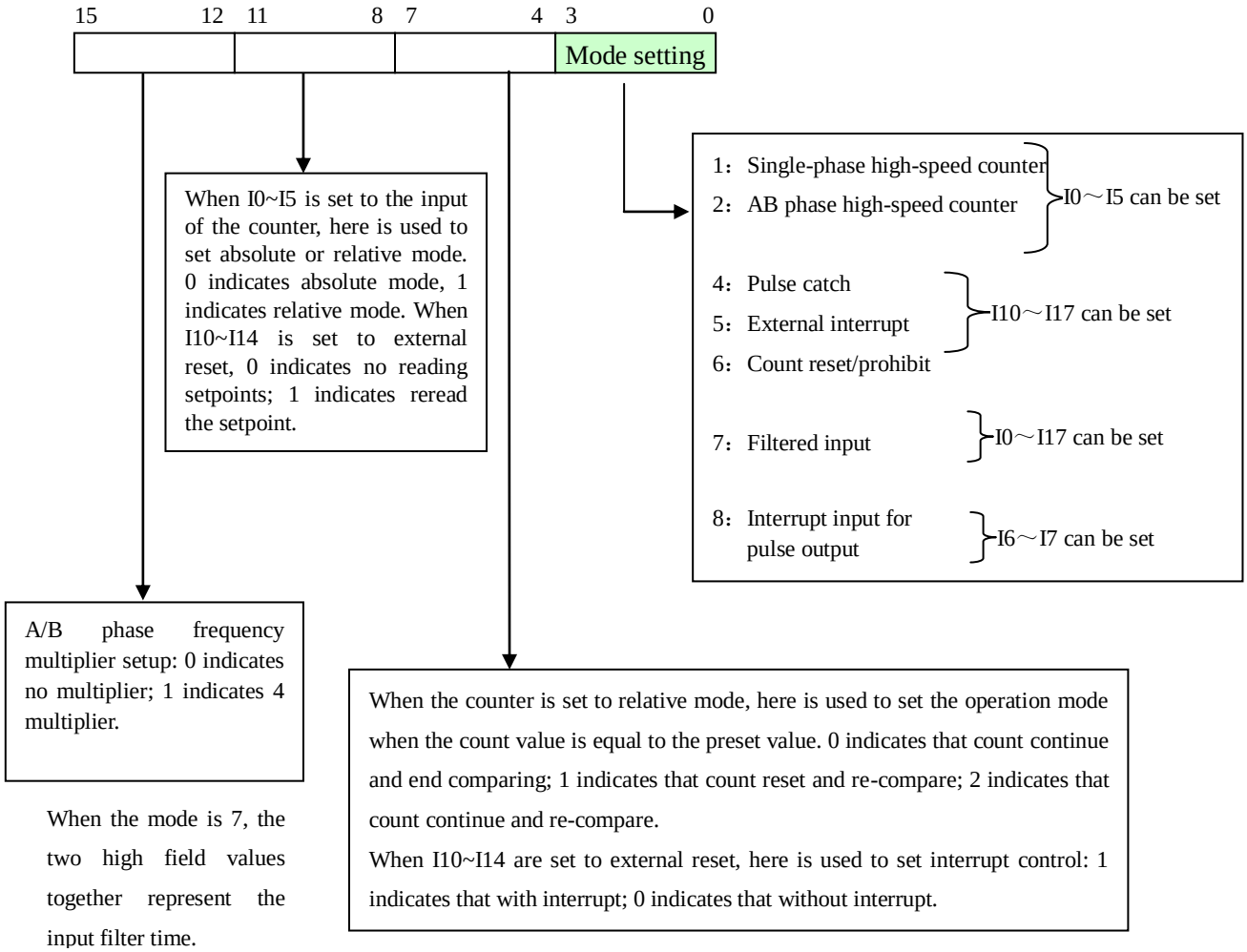
5-2-1 Inputs setup registers table

register	Function	Default value (HEX)
R7400	I0 mode setup register (high-speed counter)	1007
R7401	I1 mode setup register (high-speed counter)	1007
R7402	I2 mode setup register (high-speed counter)	1007
R7403	I3 mode setup register (high-speed counter)	1007
R7404	I4 mode setup register (high-speed counter)	1007
R7405	I5 mode setup register (high-speed counter)	1007
R7406	I6 mode setup register (pulse output interrupt)	1007
R7407	I7 mode setup register (pulse output interrupt)	1007
R7410	I10 mode setup register (pulse catch / external interrupt, etc.)	1007
R7411	I11 mode setup register (pulse catch / external interrupt, etc.)	1007
R7412	I12 mode setup register (pulse catch / external interrupt, etc.)	1007
R7413	I13 mode setup register (pulse catch / external interrupt, etc.)	1007
R7414	I14 mode setup register (pulse catch / external interrupt, etc.)	1007
R7415	I15 mode setup register (pulse catch / external interrupt, etc.)	1007
R7416	I16 mode setup register (pulse catch / external interrupt, etc.)	1007
R7417	I17 mode setup register (pulse catch / external interrupt, etc.)	1007
R7440	The first single-phase or two-phase counter multi-preset register.	798（R3630）
R7441	The second single-phase counter multi-preset register.	7A0（R3640）
R7442	The 3rd single-phase or the second two-phase counter multi-preset register.	7A8（R3650）
R7443	The 4th single-phase counter multi-preset register.	7B0（R3660）
R7444	The 5th single-phase or the 3rd two-phase counter multi-preset register.	7B8（R3670）
R7445	Modifying inputs mode parameters flag register in RUN mode.	5555

5-2-2 Input points mode setup

According to different function of input points, the setting content of each corresponding setup register is also different. We divided the 16 bits of setup register into four fields.

The following figure shows the function of the four fields:



Depending on its function, the inputs can be divided into the following five categories: high-speed counter inputs, interrupt inputs used for pulse output, count reset /disable inputs, external interrupt / pulse catch inputs, filtered inputs.

1．high-speed counter inputs (I0～I5) setup:

Mode	Data in R7400～R7405	Description
Single-phase counter (20-point model: I0~I3)	0001 (BCD, ABS absolute mode) 2001 (HEX, ABS absolute mode)	High-speed inputs (I0-I4) can be set to single-phase high-speed counter.
	0101（BCD, INC relative mode, count continue, end comparing） 0111（BCD, INC relative mode, count reset, re-compare） 0121（BCD, INC relative mode, count continue, re-compare） 2101（HEX, INC relative mode, count continue, end comparing） 2111（HEX, INC relative mode, count reset, re-compare） 2121（HEX, INC relative mode, count continue, re-compare）	
AB phase counter (20-point model: I0~I3)	0002 (BCD, ABS Absolute mode, no multiplier) 1002 (BCD, ABS Absolute mode, 4 multiplier) 2002 (HEX, ABS Absolute mode, no multiplier) 3002 (HEX, ABS Absolute mode, 4 multiplier)	Only I0, I2, I4 can be set to A/B phase count. When one of these points is set to A/B phase count, the next point can not be set (fixed to 0). Note: When set I0, I2, I4 that had been set to A/B phase count to other mode, you must also modify the mode of I1, I3, I5, otherwise it will report parameters setting error (0 is non-normal mode word).
	0102 (BCD, INC relative mode, no multiplier, count continue, end comparing) 0112 (BCD, INC relative mode, no multiplier, count reset, re-compare) 0122 (BCD, INC relative mode, no multiplier, count continue, re-compare) 1102 (BCD, INC relative mode, 4 multiplier, count continue, end comparing) 1112 (BCD, INC relative mode, 4 multiplier, count reset, re-compare) 1122 (BCD, INC relative mode, 4 multiplier, count continue, re-compare） 2102 (HEX, INC relative mode, no multiplier, count continue, end comparing) 2112 (HEX, INC relative mode, no multiplier, count reset, re-compare) 2122 (HEX, INC relative mode, no multiplier, count continue, re-compare) 3102 (HEX, INC relative mode, 4 multiplier, count continue, end comparing) 3112 (HEX, INC relative mode, 4 multiplier, count reset, re-compare) 3122 (BCD, INC relative mode, 4 multiplier, count continue, re-compare）	
filtered input	XX07（XX is filtered time）	Inputs I0~I5 can be set to filtered inputs.

2．Interrupt inputs (I6~I7) used for pulse output setup:

Mode	Data in R7406～R7407	description
Pulse catch	0008	Interrupt inputs used for pulse output, I6 corresponds to channel 0, I7 corresponds to channel 1.
Filtered input	XX07（XX：filtered inputs）	Inputs I6~I7 can be set to filtered inputs.

3. Inputs (I10~I17) used for external interrupt/ pulse catch setup (20-point models only I0~I3):

Mode	Data in R7410～R7417	Description
Pulse catch	0004	Inputs I10~I17 can be set to pulse catch
External interrupt	0005	Inputs I10~I17 can be set to external interrupt
Filtered input	XX07（XX: filtered time）	Inputs I10~I17 can be set to filtered input
Reset input or counting prohibition	0006 (without interruption, without re-reading the setpoint)	External reset or count prohibit (only I10~I15)
	0016 (with interruption, without re-reading the setpoint)	External reset（only I10~I14）
	0106 (without interruption, re-read the setpoint)	External reset（only I10~I14）
	0116 （ with interruption, re-read the setpoint）	External reset（only I10~I14）

The following table lists external reset and count prohibition inputs used for high-speed counter. Note that external reset and count prohibition feature is only valid for two-phase high-speed counter (20-point model: I0~I3).

Mode	I10	I11	I12	I13	I14	I15
Single-phase high-speed counter mode	Reset input for 1# channel single-phase high-speed counter	Reset input for 2# channel single-phase high-speed counter	Reset input for 3# channel single-phase high-speed counter	Reset input for 4# channel single-phase high-speed counter	Reset input for 5# channel single-phase high-speed counter	
Two-phase high-speed counter mode	Reset input for 1# channel two-phase high-speed counter	Count prohibition input for 1# channel two-phase high-speed counter	Reset input for 2# channel two-phase high-speed counter	Count prohibition input for 2# channel two-phase high-speed counter	Reset input for 3# channel two-phase high-speed counter	Count prohibition input for 3# channel two-phase high-speed counter

5-3 High-speed counter

5-3-1 Single-phase high-speed counter

1. Function description

The NK1 HSIO circuit contains five single-phase high-speed counters (C360~C371, each conter accoupies two counter location). The counters are 32-bit counter and count only upwards.

① 4 preset values
Each contrer has 4 preset values area (each preset value accoupies two registers).

② Interrupt
In RUN mode, the counter continuously compares the current count value with the preset. When the two are equal, a special relay contact is energized and program execution jumps to the interrupt routine. After the interrupt service routine is complete, the CPU returns to the ladder program, resuming program execution from the point of interruption.

③ Equal relay
Each preset is assigned a special relay which allocated from SP540.

④ Count range
The data format of Count value can be BCD or HEX. When the data format is BCD, the count range is 0~99999999, the count value will maintain 99999999 when it reaches the maximum count value. When the data format is HEX, the count range is 0~0xFFFFFFFF（0~4294967295）, the count value will maintain 0x FFFFFFFF（4294967295） when it reaches the maximum count value.

⑤ Reset count value
Count value can be reset by the program, it can also be reset by an external signal connetctd to external reset terminal.

⑥ Counting mode
Can be counted in absolute and relative value.

2. Special features setup

Input points I0~I4 can be set to single-phase high-speed counter input, the corresponding setup register (R7400 ~ R7404) can be set as follows:

Mode	Data in R7400～R7404	Description
Single-phase high-speed counter (20-point: I0~I3)	0001 (BCD, absolute value) 2001 (HEX absolute value)	I0~I4 can be set to single-phase high-speed counter input.
	0101 (BCD, relative value, counting continue, end comparing) 0111 (BCD, relative value, count is reset, re-compare) 0121 (BCD, relative value, counting continue, re-compare) 2101 (HEX, relative value, counting continue, end comparing) 2111 (HEX, relative value, count is reset, re-compare) 2121 (HEX, relative value, counting continue, re-compare)	

Each single-phase high-speed counter can be set to use an external reset input, these external reset terminal use input points I10~I14, the following settings are required when using these inputs as external reset terminal.

Mode	**Data in R7410～R7414**	**Description**
External reset	0006	Without interruption, without re-reading the preset value
	0016	With interruption, without re-reading the preset value
	0106	Without interruption, re-read the preset value
	0116	With interruption, re-read the preset value

3. Specifications

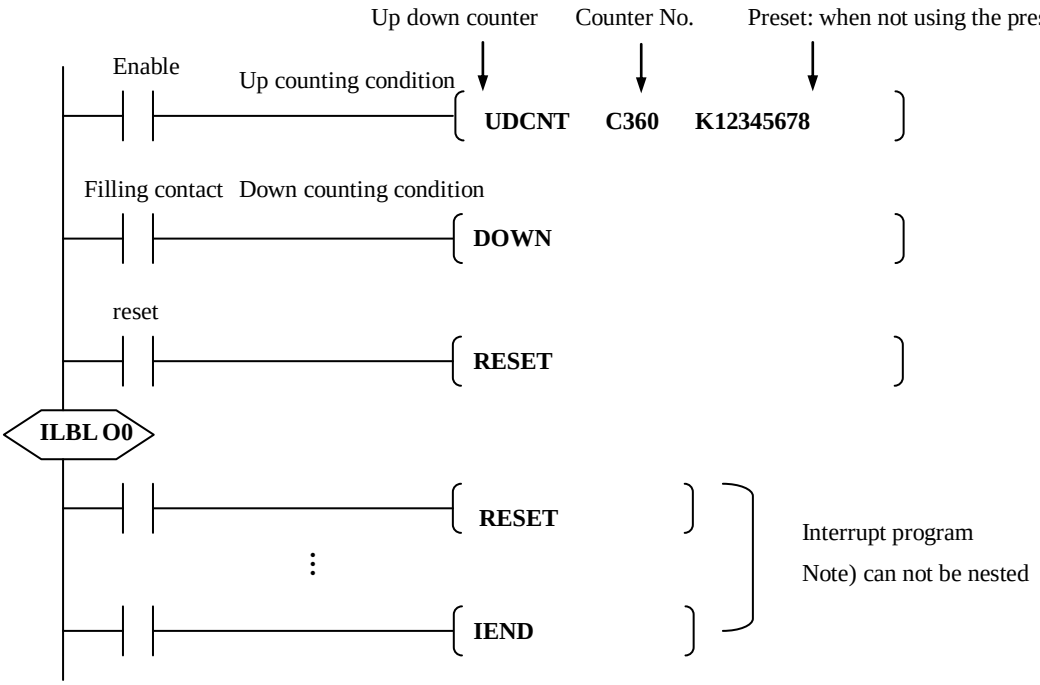
Item		Channel 1	Channel 2	Channel 3	Channel 4
Counting speed		100K	100K	100K	100K
Counter No.		C360, C361	C362, C363	C364, C365	C366, C367
Current value register		R1360, R1361	R1362, R1363	R1364, R1365	R1366, R1367
Count range	BCD	0-99999999	0-99999999	0-99999999	0-99999999
	HEX	0~FFFFFFFF	0~FFFFFFFF	0~FFFFFFFF	0~FFFFFFFF
Preset value mode		Relative or absolute	Relative or absolute	Relative or absolute	Relative or absolute
Number of preset value		4	4	4	4
Preset values address pointer register		R7440	R7441	R7442	R7443
Multi-preset value register (default value)		R3630-R3637	R3640-R3647	R3650-R3657	R3660-R3667
Equal relay		SP540-SP543	SP544-SP547	SP550-SP553	SP554-SP557
interrupt		ILBL O0	ILBL O1	ILBL O2	ILBL O3
Counting input terminal		I0	I1	I2	I3
External reset terminal		I10	I11	I12	I13
Counting enable (software)		Yes	Yes	Yes	Yes
Counting reset (software)		Yes	Yes	Yes	Yes

Items	Channel 5(20-point model has no this channel)			
Counting speed	100k			
Counter No.	C370, C371			
Current value register	R1370, R1371			
Counting range	0-99999999			
	0~FFFFFFFF			
Preset value mode	Relative or absolute			
Number of preset value	4			
Preset values address register	R7444			

Multi-preset value register (initial value)	R3670-R3677			
Special relay	SP560-SP563			
Interrupt	ILBL O4			
Counting input terminal	I4			
Counting reset terminal	I14			
Counting enable (software)	Yes			
Counting reset (software)	Yes			

4. Program example

High-speed counter program of NK1 PLC is achived by using up down counter instruction UDCNT.



Up counting terminal of UDCNT instruction is used for counting enable, reset terminal of UDCNT instruction is used for program reset. Because don't use down counting terminal, so it can connect any legal contact as a filling contact, e.g. SP1 (always on).

Counter number is selected according to counter channel. For example, C360 (the actual use of C360, C361) for 1# channel; 362 (the actual use of C362, C363) for 2# channel, etc. If the preset does not use UP/DOWN signal, you can use K0. In additon, when the data format of the preset is HEX, since the preset of UDCNT insturction does not support the full range of the count value, so the preset is up to 0x99999999 (2576980377) instead of 0xFFFFFFFF (4294967295).

After NK1 PLC enters RUN mode and the enable condition of UDCNT is ON, the high-speed counter starts counting. The counter continuously compares the current count value with the preset. When the two are equal, a special relay contact is energized and program execution jumps to the interrupt routine. After the interrupt service routine is complete, the CPU returns to the ladder program, resuming program execution from the point of interruption. The compare function is ready for the next preset event.

In the counting process, if the enable condition of UNCNT turns to OFF, then the counter stops counting (count value is held at the current value). When the enable condition turns to ON, the counter continues counting. (Count value is not reset.)

5. Use multi-preset value

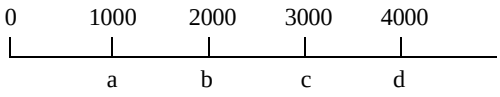
Each high-speed counter supports up to 4 presets, which you can program. Presets are double word so they occupy two R-memory registers. Therefore, multi-preset values of each counter occupie a contiguous set of eight registers.

Two preset modes are available, absolute and incremental.

A）absolute mode

- Set preset values in absolute mode.
- The current preset value must be greater than the previous preset.
- When the current preset is less than the previous preset, the CPU considers that setup is completed.

Example: in the following figure, starting from 0, set preset in the order of a→b→c→d.



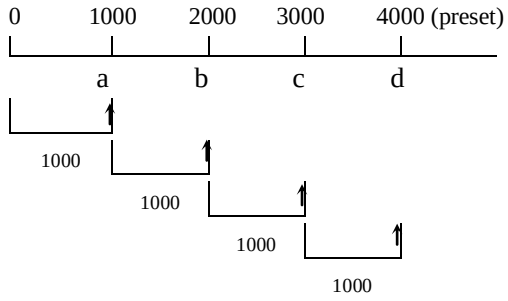
Data in preset registers (the initial value o f CH1)		
The first preset	R3630	1000
	R3631	0000
The second preset	R3632	2000
	R3633	0000
The 3rd preset	R3634	3000
	R3635	0000
The 4th preset	R3636	4000
	R3637	0000

Note: R7440 is the pointer to the R-memory location which contains the beginning of the preset data table. The default data in R7440 is 798, 798 is a hexadecimal number and it is 3630 in octal.

B）incremental mode

- Set presets in incremental mode.
- Starts sequentially from the first preset.

Example: in the following figure, starting from 0, set preset in the order of a→b→c→d.



Data in preset registers (the initial value o f CH1)		
The first preset	R3630	1000
	R3631	0000
The second preset	R3632	1000
	R3633	0000
The 3rd preset	R3634	1000
	R3635	0000
The 4th preset	R3636	1000
	R3637	0000

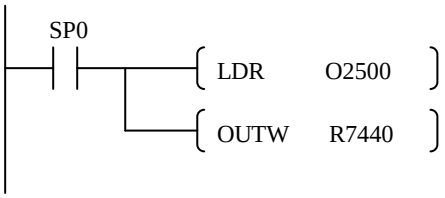
C）Change preset register area

The following table lists the default value in the pointer register.

Up counter channel	1# channel	2# channel	3# channel	4# channel	5# channel
pointer register	R7440	R7441	R7442	R7443	R7444
The default value	798 (R3630)	7A0 (R3640)	7A8 (R3650)	7B0 (R3660)	7B8 (R3670)

By changing the value in first address register, you can change preset register area by writing the appropriate value into R7440~R7444. In the following example, use a set of register starting from R2500 as preset registers of CH1.

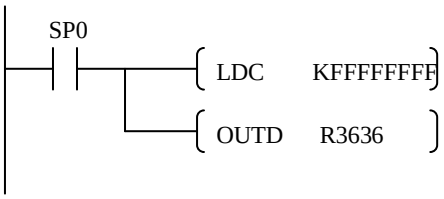
Example: a set of register starting from R2500 as preset registers of CH1.
Data in R7440:
0540(H) = 2500(O)
Registers effective range is:
R2000~R7370 R10000~R36770



D）Set multi-preset end code

When all 4 available presets are used, the CPU knows automatically when it reaches the end of the preset table. When using fewer than 4 presets, however, it is necessary to signal the CPU that it has reached the last preset. The way to signal the end of the block of presets is to insert a table-end code "FFFFFFFF" into the next available register.

Example:



Preset table example

R3631	R3630	0000	1000
R3633	R3632	0000	2000
R3635	R3634	0000	2500
R3637	R3636	0000	FFFF

In the example, three presets are used. The FFFF FFFF in R3637-R3636 indicates the previous preset was the last preset.

Note: Due to 0xFFFFFFFF is used to indicate the end of the multi-preset, so 0xFFFFFFFF can no longer be used as a preset.

E）Equal relays and interrupt

The following table lists the equal relays corresponding to each preset value.

CH1			CH 2			CH 3	
Equal relay	Preset value register		Equal relay	Preset value register		Equal relay	Preset value register
SP540	Preset 1（R3630）		SP544	Preset 1（R3640）		SP550	Preset 1（R3650）
SP541	Preset 2（R3632）		SP545	Preset 2（R3642）		SP551	Preset 2（R3652）
SP542	Preset 3（R3634）		SP546	Preset 3（R3644）		SP552	Preset 3（R3654）
SP543	Preset 4（R3636）		SP547	Preset 4（R3646）		SP553	Preset 4（R3656）

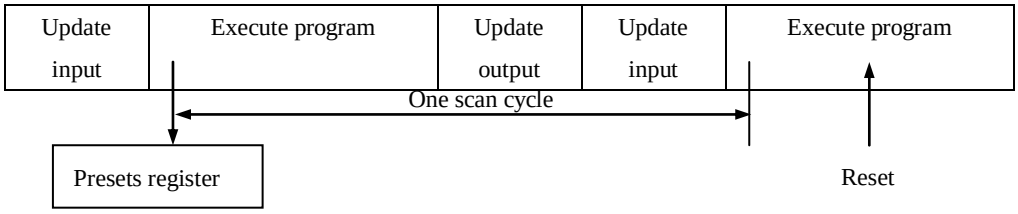
CH4			CH5	
Equal relay	Preset value register		Equal relay	Preset value register
SP554	Preset 1（R3660）		SP560	Preset 1（R3670）
SP555	Preset 2（R3662）		SP561	Preset 2（R3672）
SP556	Preset 3（R3664）		SP562	Preset 3（R3674）
SP557	Preset 4（R3666）		SP563	Preset 4（R3676）

NK1 PLC does not support interrupt nesting, it will handle the new interrupt until it has completed the current interrupt. It can record up to 24 new interrupts.

F）Program example: rewrite preset values

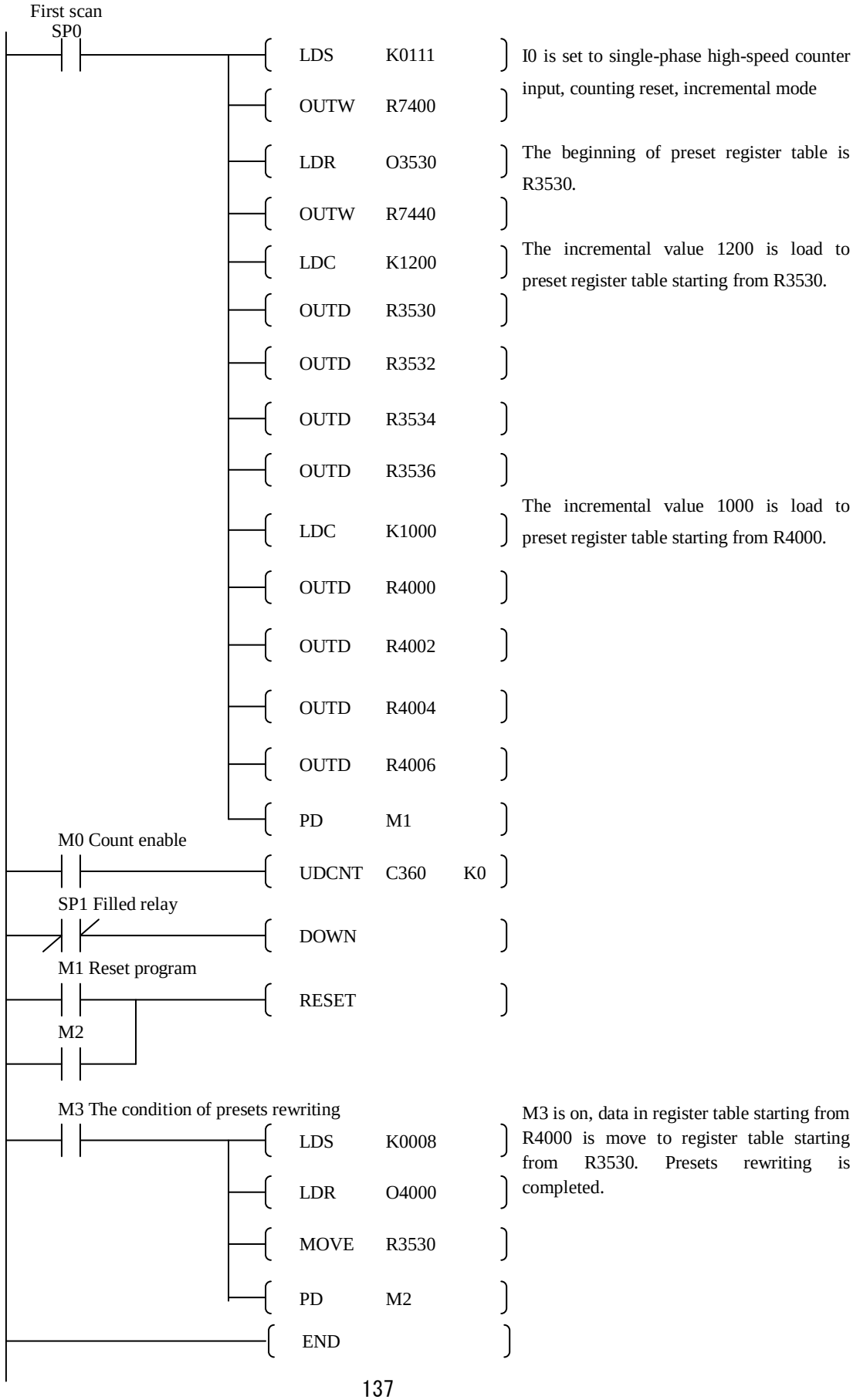
Preset values can not be freely rewritten during high-speed counter operation. If preset values have been rewrite during counter operation, the new preset values take effect only after the current counter value has been reset by an external reset signal or the program.

In addition, the processing of rewriting preset values requires more than one scan cycle, please wait for more than one scan cycle and then reset the current counter value.



In the following example, the initial preset is 1200, which will be rewritten to 1000 when M3 is on.

- Program example (Use SP0 on power-up, don't use R7445)



6. Writing your control program

Condition and function:

Absolute mode, BCD

CH1

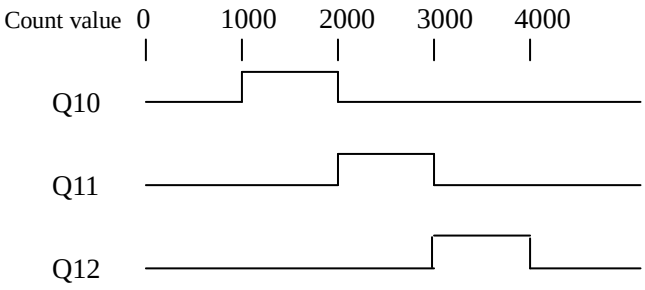
Preset register table starting from R3630

When the current count value is 1000, Q10 is ON

When the current count value is 2000, Q10 is OFF, Q11 is ON

When the current count value is 3000, Q11 is OFF, Q12 is ON

When the current count value is 4000, Q12 is OFF

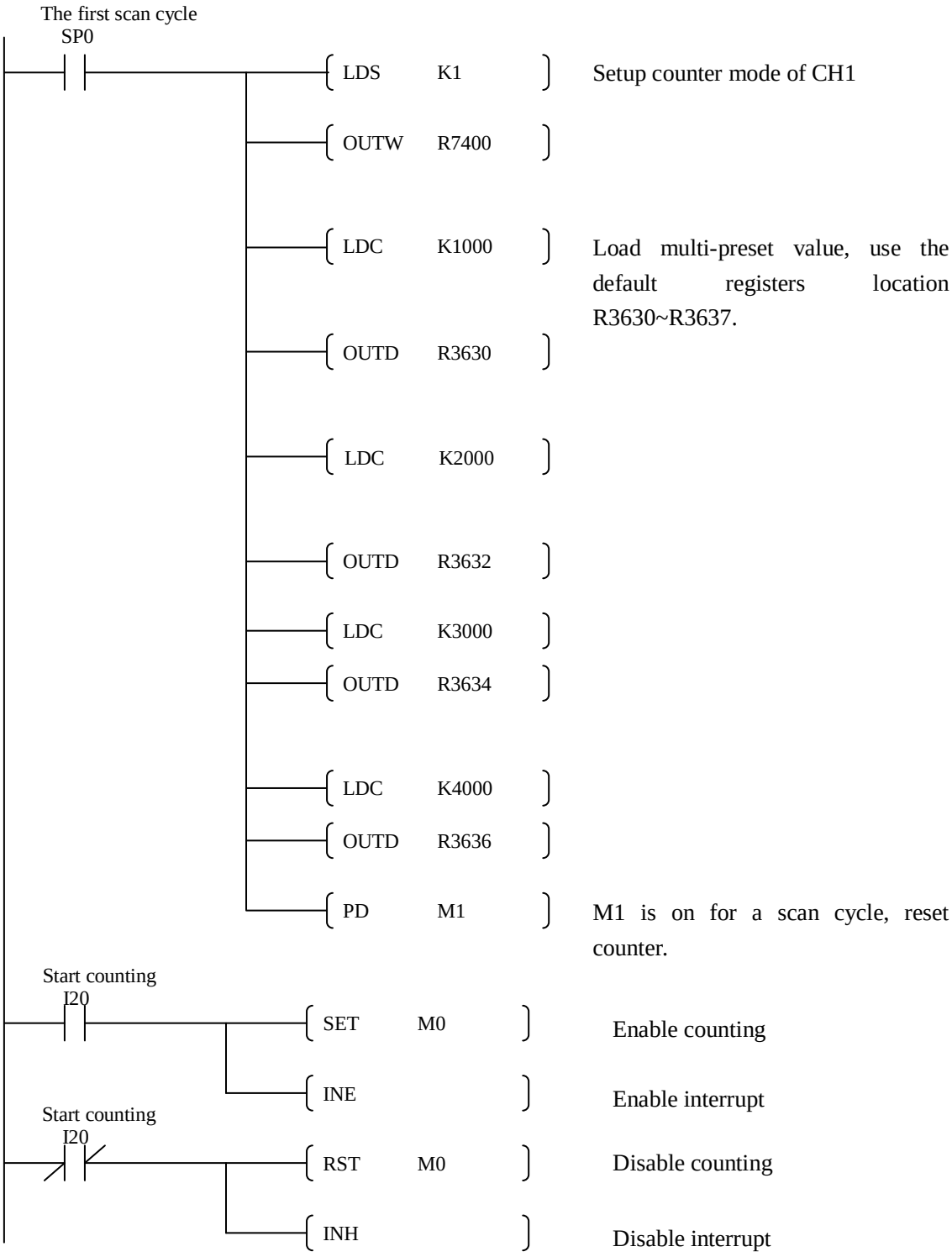


Data in preset registers（CH1）

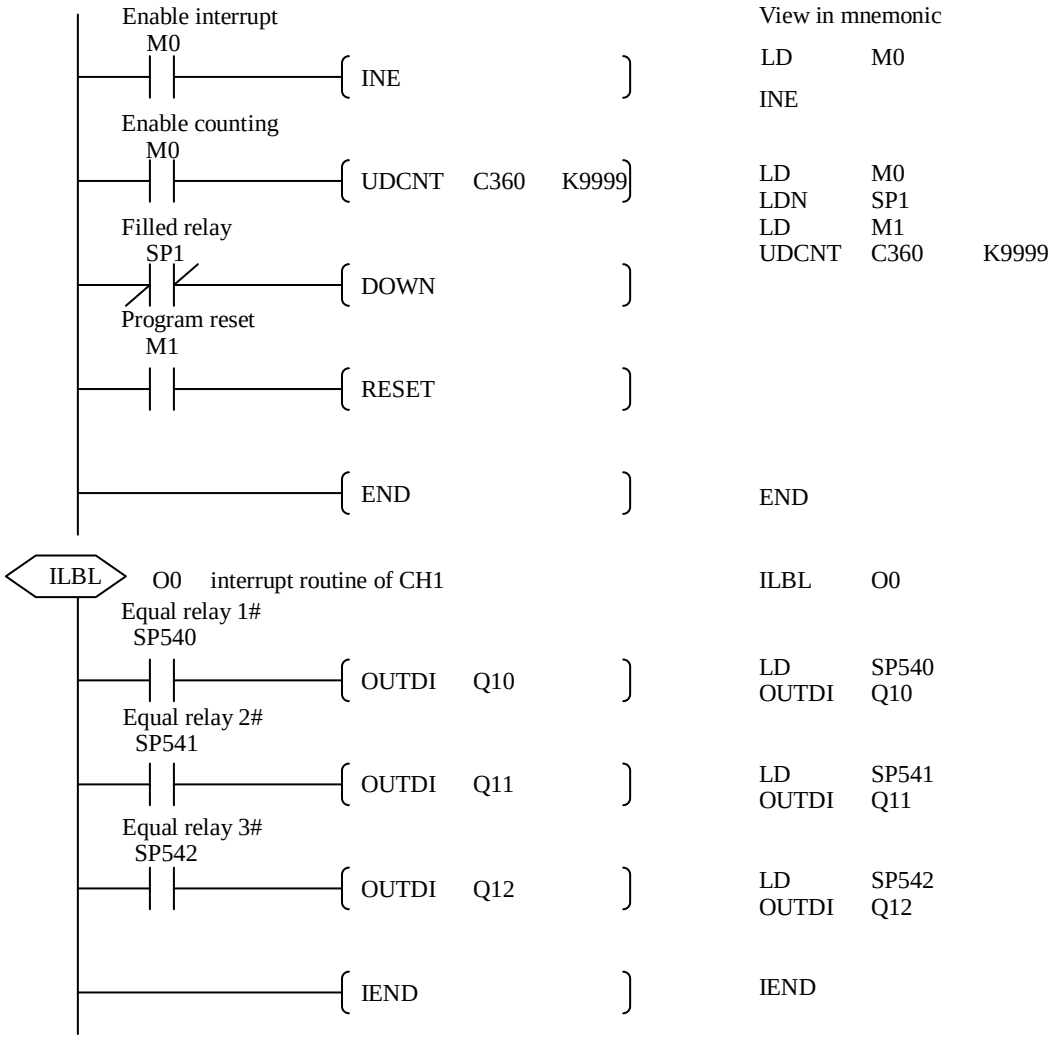
The first preset	R3630	1000
	R3631	0000
The 2nd preset	R3632	2000
	R3633	0000
The 3rd preset	R3634	3000
	R3635	0000
The 4th preset	R3636	4000
	R3637	0000

- Ladder program example

The initial setup routine (Use SP0 on power-up, don't use R7445):

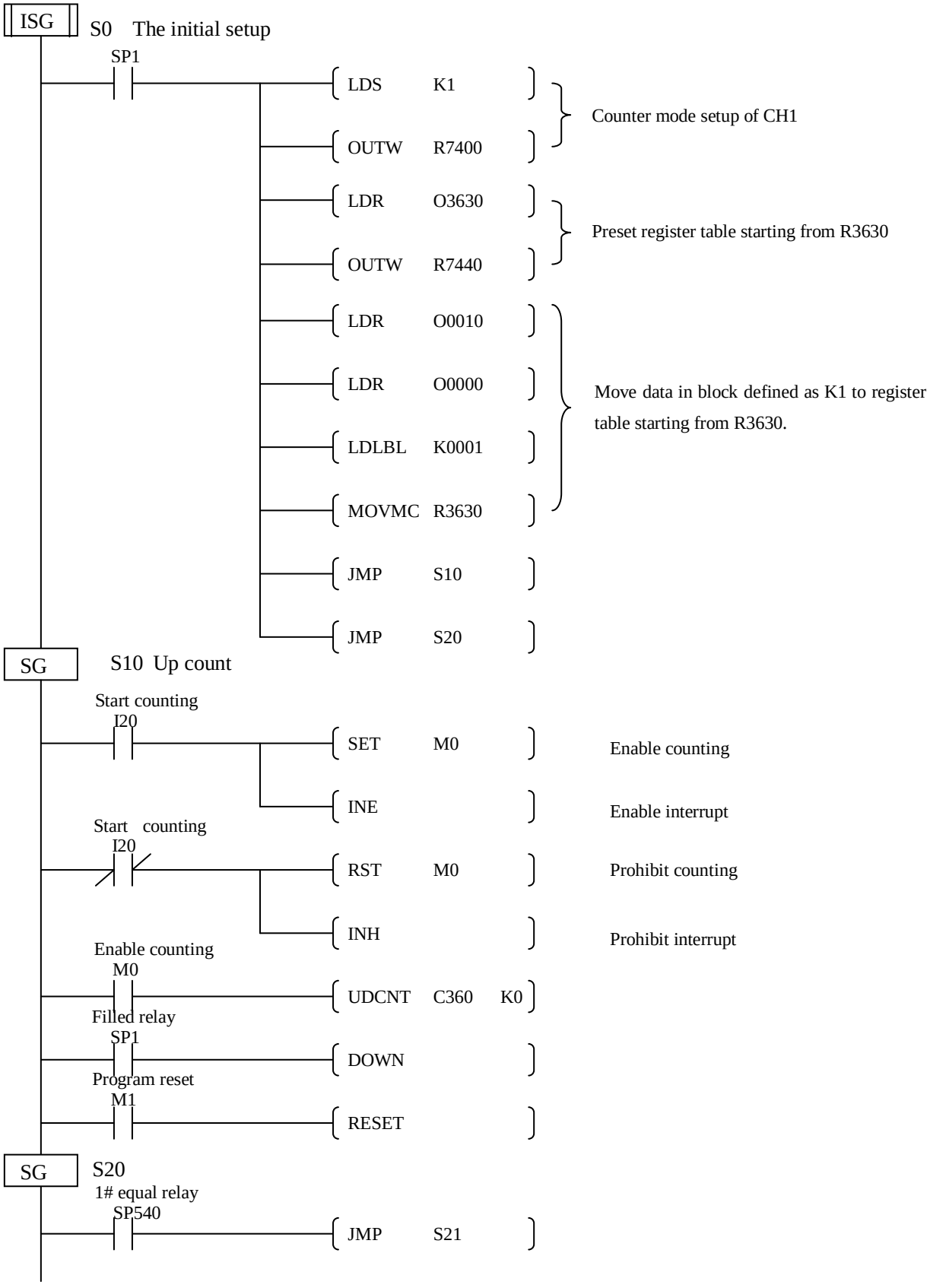


High-speed counter routine

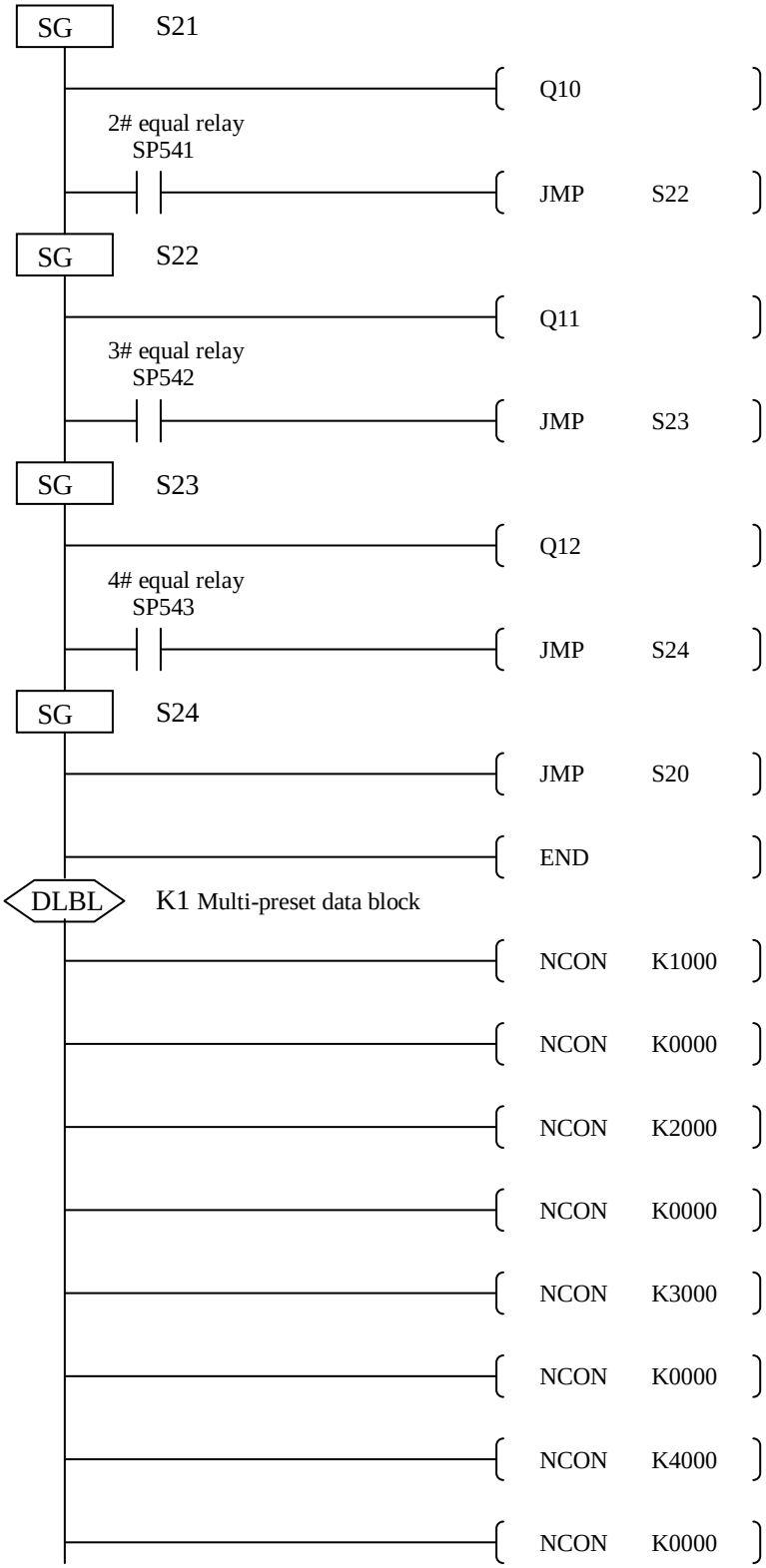


- Action description
  - ① When the count value reaches 1000, equal relay SP540 is ON, execute the interrupt routine, Q10 is ON.
  - ② When the count value reaches 2000, equal relay SP541 is ON, execute the interrupt routine, Q10 is OFF, Q11 is ON.
  - ③ When the count value reaches 3000, equal relay SP542 is ON, execute the interrupt routine, Q11 is OFF, Q12 is ON.
  - ④ When the count value reaches 4000, equal relay SP543 is ON, execute the interrupt routine, Q12 is OFF.

Stage program example (Setup in ISG stage on power-up, don't use SP0 and R7447)



Continued


SG S21
Q10
2# equal relay
SP541
JMP S22
SG S22
Q11
3# equal relay
SP542
JMP S23
SG S23
Q12
4# equal relay
SP543
JMP S24
SG S24
JMP S20
END
DLBL K1 Multi-preset data block
NCON K1000
NCON K0000
NCON K2000
NCON K0000
NCON K3000
NCON K0000
NCON K4000
NCON K0000

5-3-2 A/B phase high-speed counter

1. Function description

The counter in the HSIO circuit can count up/down signals from two quadrature signal pulses.

Quadrature signals are commonly generated from incremental encoders. The A/B phase counter is 32 bits, its maximum counting frequency is 100 kHz.

NK1 PLC supports up to 3 A/B phase counters (C360/C361, C364/C365, C370/C371, each conter accoupies two counter location). (20-point model has 2 A/B phase counters)

① 4 preset values

Each conter has 4 preset values (each preset value is 32-bit, accoupies two registers). When the preset value is negative, the most significant bit of the 32-bit preset value should be set to "1".

② Interrupt

In RUN mode, the counter continuously compares the current count value with the preset. When the two are equal, a special relay contact is energized and program execution jumps to the interrupt routine. After the interrupt service routine is complete, the CPU returns to the ladder program, resuming program execution from the point of interruption.

③ Preset equal relay

Each preset is assigned a special relay which allocated from SP540.

④ Count range

Count range is 88388608~8388607 (-8388608~8388607) in BCD format; It is 0xFFFFFFFF~0x7FFFFFFF (-2147483647~2147483647) in HEX format.

Counting forward, when the counter reaches the maximum value （BCD format: 8388607, HEX format: 0x7FFFFFFF）, if continues to receive the positive pulse signal, the count value will jump from the maximum to the minimum (BCD format: from 08388607 → -8388608; HEX format: from 0x7FFFFFFF → 0xFFFFFFFF); count reverse, when the counter count reaches the minimum -8388608, if continues to receive the reverse pulse signal, the count value will jump from the minimum to the maximum.

When the counter value of NK1 is an 8-bit BCD data: the highest bit is the sign bit, 8 represents the opposite direction pulse, 0 indicates the positive direction pulse. For example, -8388608 is displayed as 88388608, -1234 is displayed as 80001234, 8388607 is displayed as 8388607, and so on.

When the counter value of NK1 is a HEX data: it is displayed in the form of the sign bit (MSB BIT32) + data bit (bit31~0). The highest bit is the sign bit, 1 represents the opposite direction pulse, 0 indicates the positive direction pulse. For example, 0x7FFFFFFF (-2147483647) is displayed as 0xFFFFFFFF; 100000 is 0x186A0 in HEX format, -100000 is displayed as 0x800186A0, and so on.

⑤ Reset count value

Count value can be reset by the program; it can also be reset by an external signal connetctd to the external reset terminal.

⑥ Count inhibit

You can use the instruction inhibit function and can also use external hardware input inhibit function.

⑦ Counting mode

Can be counted in absolute and relative value.

2. Special features setup

Input points I0~I5 can be set to A/B-phase high-speed counter input, the corresponding setup register (R7400 ~ R7404) can be set as follows:

Mode	Data in R7400～R7405	Description
A/B-phase high-speed count	0002 (BCD, absolute value ABS, no multiplier) 1002 (BCD, absolute value ABS, 4 multiplier) 2002 (HEX, absolute value ABS, no multiplier) 3002 (HEX, absolute value ABS, 4 multiplier)	Only I0, I2, I4 can be set to A/B phase count mode. When one of these points is set to A/B phase count mode, the next point can not be set (fixed to 0).
	0102 (BCD, INC relative mode, no multiplier, count continue, end comparing) 0112 (BCD, INC relative mode, no multiplier, count reset, re-compare) 0122 (BCD, INC relative mode, no multiplier, count continue, re-compare) 1102 (BCD, INC relative mode, 4 multiplier, count continue, end comparing) 1112 (BCD, INC relative mode, 4 multiplier, count reset, re-compare) 1122 (BCD, INC relative mode, 4 multiplier, count continue, re-compare) 2102 (HEX, INC relative mode, no multiplier, count continue, end comparing) 2112 (HEX, INC relative mode, no multiplier, count reset, re-compare) 2122 (HEX, INC relative mode, no multiplier, count continue, re-compare) 3102 (HEX, INC relative mode, 4 multiplier, count continue, end comparing) 3112 (HEX, INC relative mode, 4 multiplier, count reset, re-compare) 3122 (HEX, INC relative mode, 4 multiplier, count continue, re-compare)	

Each A/B-phase high-speed counter can be set to use an external reset and count inhibition input, these external reset terminal use input points I10, I12 and I14, external count inhibition terminal use input points I11, I13, and I15. The corresponding parameters register (R7410 ~ R7415) need be set in accordance with the following table. (20 model: R7410~R7413)

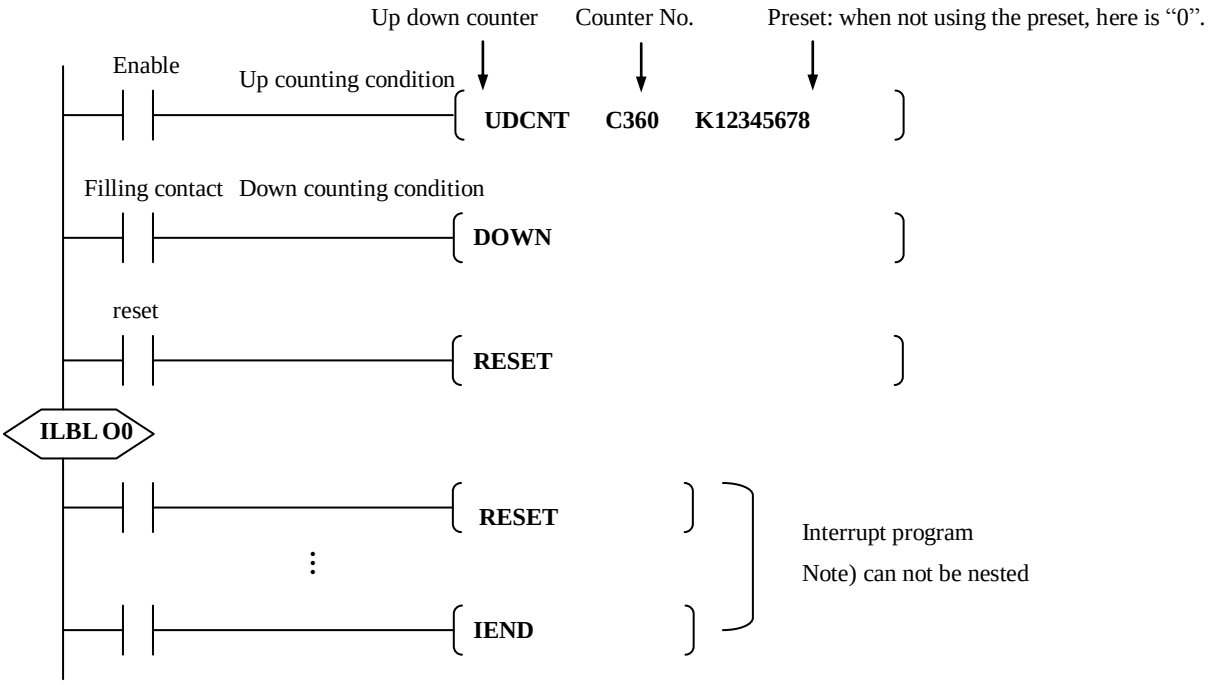
Mode	R7410 R7412 R7414	Description
External reset	0006	Without interruption, without re-reading the preset value
	0016	With interruption, without re-reading the preset value
	0106	Without interruption, re-read the preset value
	0116	With interruption, re-read the preset value
Mode	**R7411 R7413 R7415**	**Description**
External count inhibition	0006	

3. Specifications

Item		Channel 1	Channel 2	Channel 3
Counting speed		100k	100k	100k
Counter No.		C360,C361	C364,C365	C370,C371
Current value register		R1360,R1361	R1364,R1365	R1370,R1371
Count range	BCD	-8388608~8388607	-8388608~8388607	-8388608~8388607
	HEX	0xFFFFFFFF ~ 0x7FFFFFFF	0xFFFFFFFF ~ 0x7FFFFFFF	0xFFFFFFFF ~ 0x7FFFFFFF
Preset value mode		Relative or absolute	Relative or absolute	Relative or absolute
Count mode		A/B-phase	A/B-phase	A/B-phase
Number of preset value		4	4	4
Preset value address pointer register		R7440	R7442	R7444
Multi-preset value register (default value)		R3630-R3637	R3650-R3657	R3670-R3677
Equal relay		SP540-SP543	SP550-SP553	SP560-SP563
Interrupt		ILBL O0	ILBL O2	ILBL O4
Counting input terminal		I0, I1	I2, I3	I4, I5
external reset terminal		I10	I12	I14
External inhibit terminal		I11	I13	I15
Counting enable (software)		Yes	Yes	Yes
Counting reset (software)		Yes	Yes	Yes

4. Program example

A/B-phase high-speed counter program of NK1 PLC is achived by using up down counter instruction UDCNT.



Up counting terminal of UDCNT instruction is used for counting enable, reset terminal of UDCNT instruction is used for program reset. Because don't use down counting terminal, so it can connect any legal contact as a filling contact, e.g. SP1 (always on).

Conter number is selected according to counter channel. For example, C360 (the actual use of C360, C361) for channel 1; 364 (the actual use of C364, C365) for channel 2, 370 (the actual use of C370, C371) for channel 3, etc. If the preset does not use UP/DOWN signal, you can use K0.

After NK1 PLC enters RUN mode and the enable condition of UDCNT is ON, high-speed counter starts counting. The counter continuously compares the current count with the preset. When the two are equal, a special relay contact is energized and program execution jumps to the interrupt routine. After the interrupt service routine is complete, the CPU returns to the ladder program, resuming program execution from the point of interruption. The compare function is ready for the next preset event.

In the counting process, if the enable condition of UNCNT turns to OFF, then the counter stops counting (count value is held at the current value). When the enable condition turns to ON, the counter continues counting. (Count value is not reset.)

5. Use multi-preset value

Each counter supports up to 4 presets, which you can program. Presets are double word numbers so they occupy two R-memory registers. Therefore, multi-preset values of each counter occupy a contiguous set of eight registers.

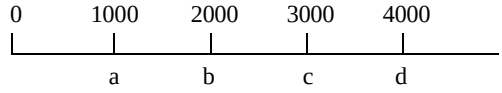
Two preset modes are available, absolute and incremental.

A) Absolute mode

- Set preset values in absolute mode.

- Set preset values staring from the first preset value.
  The first preset→the 2nd preset→……→the 4th preset

Example: in the following figure, starting from 0, set preset in the order of a→d→c→b.

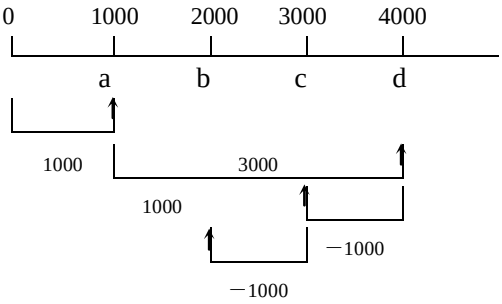


Data in preset registers (the initial value)		
The first preset	R3630	1000
	R3631	0000
The 2nd preset	R3632	4000
	R3633	0000
The 3rd preset	R3634	3000
	R3635	0000
The 4th preset	R3636	2000
	R3637	0000

B) Incremental mode

- Set presets in incremental mode.
- Starts sequentially from the first preset.

Example: in the following figure, starting from 0, set preset in the order of a→d→c→b.



Data in preset registers (the initial value)		
The first preset	R3630	1000
	R3631	0000
The 2nd preset	R3632	3000
	R3633	0000
The 3rd preset	R3634	1000
	R3635	8000
The 4th preset	R3636	1000
	R3637	8000

C) Change preset register area

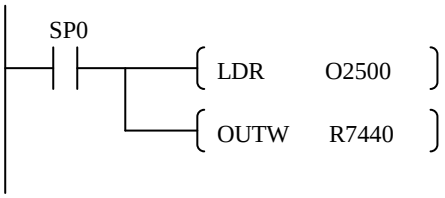
The following table lists the default value in the pointer register.

A/B-phase counter channel	Channel 1	Channel 2	Channel 3
Pointer register	R7440	R7442	R7444
The default value	798 (R3630)	7A8 (R3650)	7B8 (R3670)

By changing the value in first address register, you can change preset register area by writing the appropriate value into R7440, R7442 and R7444. In the following example, use a set of register starting from R2500 as preset registers of CH1.

Data in R7440: 0540(H) = 2500(O)

Registers effective range is:
R2000~R7370 R10000~R36770

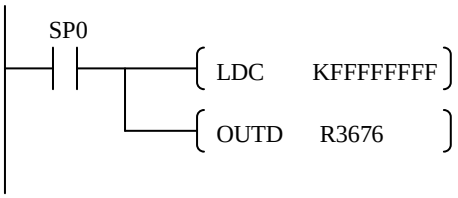


D) **Set multi-preset end code**

When all 4 available presets are used, the CPU knows automatically when it reaches the end of the preset table. When using fewer than 4 presets, however, it is necessary to signal the CPU that it has reached the last preset. The way to signal the end of the block of presets is to insert a table-end code 0xFFFFFFFF (H) into the next available register.

In the following example, three presets are used. The FFFF FFFF in R3677-R3676 indicates the previous preset was the last preset.

Example:



Preset table example

R3671	R3670	0000	1000
R3673	R3672	0000	2000
R3675	R3674	0000	2500
R3677	R3676	0000	FFFF

E) Equal relays and interrupt

The following table lists the equal relays corresponding to each preset value.

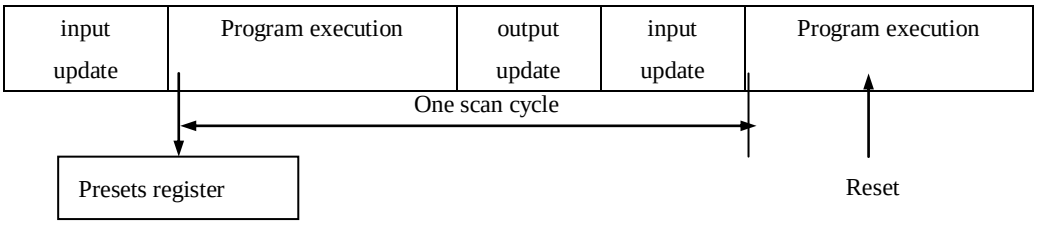
CH1			CH2			CH3	
Equal relay	Preset value register		Equal relay	Preset value register		Equal relay	Preset value register
SP540	Preset 1（R3630）		SP550	Preset 1（R3650）		SP560	Preset 1（R3670）
SP541	Preset 2（R3632）		SP551	Preset 2（R3652）		SP561	Preset 2（R3672）
SP542	Preset 3（R3634）		SP552	Preset 3（R3654）		SP562	Preset 3（R3674）
SP543	Preset 4（R3636）		SP553	Preset 4（R3656）		SP563	Preset 4（R3676）

NK1 PLC does not support nested interrupts, it will handle the new interrupt until it has completed the current interrupt. It can record up to 24 new interrupts.

F) Program example: rewrite preset values

Preset values can not be freely rewritten during high-speed counter operation. If preset values have been rewritten during counter operation, the new preset values take effect only after the current counter value has been reset by an external reset signal or the program.

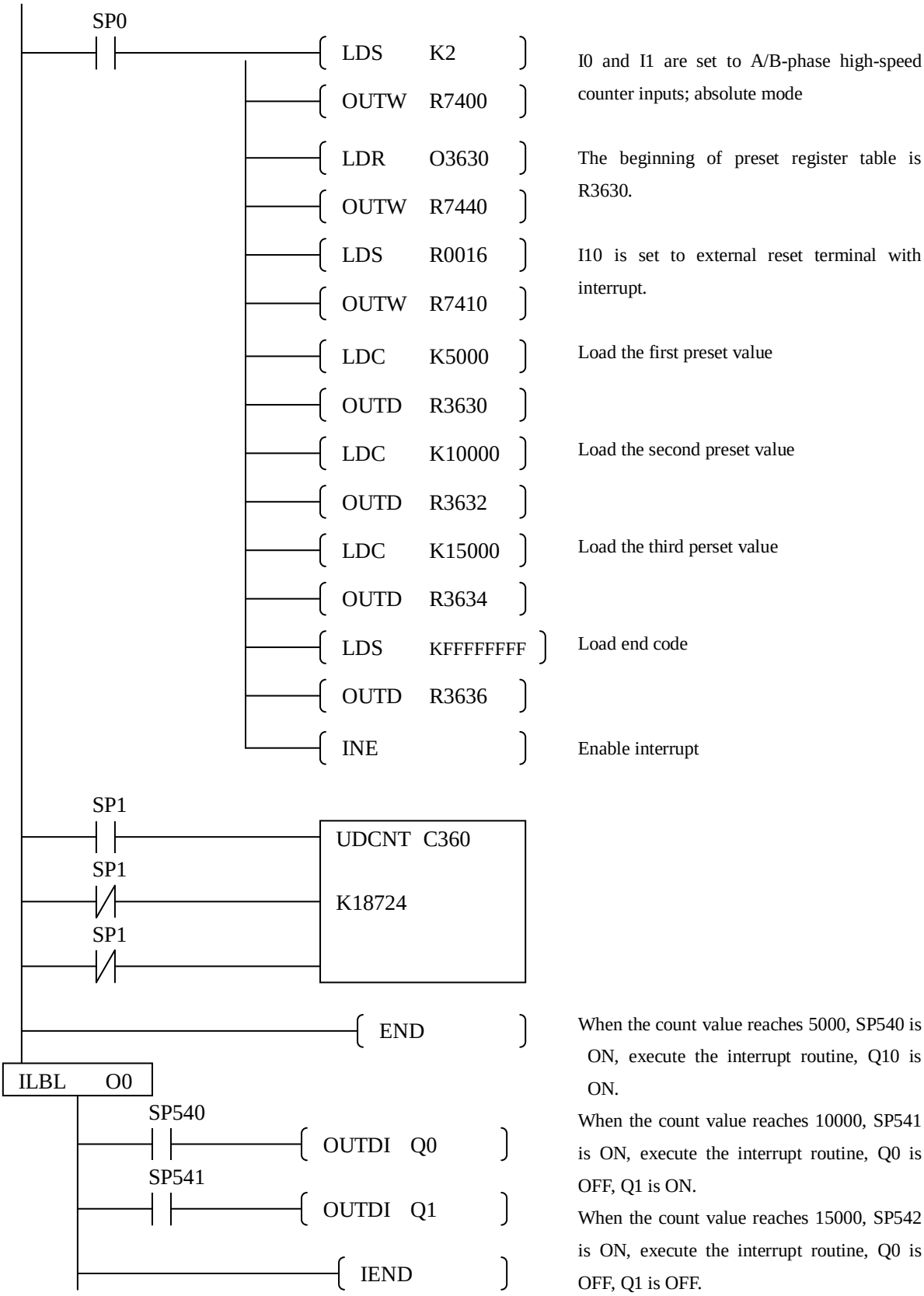
In addition, the processing of rewriting preset values requires more than one scan cycle, please wait for more than one scan cycle and then reset the current counter value.



The program that rewriting preset values of A/B-phase is same with the program of single-phase.

8. Writing your control program（Use SP0 on power-up, don't use R7445)

In the following example, I0 and I1 are the counter clock inputs; quadrature signals; absolute mode, with interrupt.



I0 and I1 are set to A/B-phase high-speed counter inputs; absolute mode

The beginning of preset register table is R3630.

I10 is set to external reset terminal with interrupt.

Load the first preset value

Load the second preset value

Load the third perset value

Load end code

Enable interrupt

When the count value reaches 5000, SP540 is ON, execute the interrupt routine, Q10 is ON.

When the count value reaches 10000, SP541 is ON, execute the interrupt routine, Q0 is OFF, Q1 is ON.

When the count value reaches 15000, SP542 is ON, execute the interrupt routine, Q0 is OFF, Q1 is OFF.

5-3-3 Mixed-use single-phase and A/B-phase high-speed counter

High-speed count circuit of NK1 series PLC supports a mix of single-phase and A/B-phase.
The following table lists the various combinations:

Item	Combination 1	Combination 2	Combination 3	Combination 4
A/B-phase high-speed counter	3	2	1	0
Single-phase high-speed counter	0	2	4	5

20-point model:

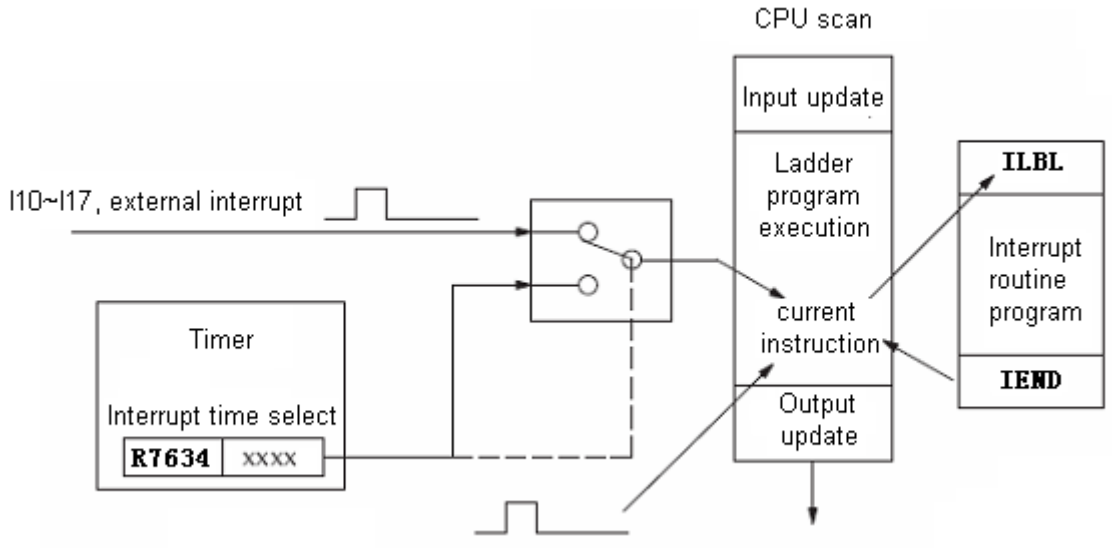
Item	Combination 1	Combination 2	Combination 3
A/B-phase high-speed counter	2	1	0
Single-phase high-speed counter	0	2	4

5-4 External interrupt

5-4-1 Function description

External interrupt is a system execution method by an external signal. When an external interrupt input signal from OFF to ON, program execution jumps to the interrupt routine. After the interrupt service routine is complete, the CPU returns to the ladder program, resuming program execution from the point of interruption.

NK1 PLC does not support interrupt nesting, it will handle the new interrupt until it has completed the current interrupt. It can record up to 24 new interrupts.



5-4-2 Specifications

Channel	1	2	3	4
Pulse width	0.05ms			
Pulse cycle	≥0.1ms			
Effective pulse	Rising edge			
Interrupt number	ILBL 10	ILBL 11	ILBL 12	ILBL 13
Interrupt input	I10	I11	I12	I13
Interrupt priotity	high->			
Channel	5	6	7	8
Pulse width	0.05ms			
Pulse cycle	≥0.1ms			
Effective pulse	Rising edge			
Interrupt number	ILBL 14	ILBL 15	ILBL 16	ILBL 17
Interrupt input	I14	I15	I16	I17
Interrupt priotity	->low			

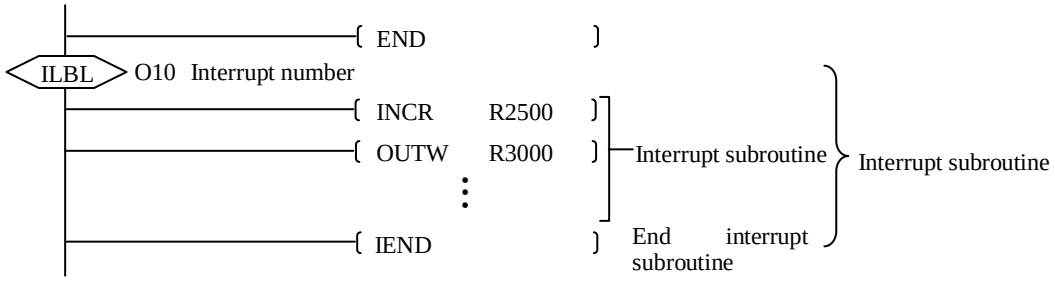
5-4-3 Function setting

NK1 PLC supports up to eight external interrupts and uses I10~I17 as external interrupt signal input point. You must first setup the corresponding registers before using the external interrupt function.

Input point	Setup register	Interrupt number	Set value
I10	R7410	ILBL O10	0005 (external interrupt)
I11	R7411	ILBL O11	0005 (external interrupt)
I12	R7412	ILBL O12	0005 (external interrupt)
I13	R7413	ILBL O13	0005 (external interrupt)
I14	R7414	ILBL O14	0005 (external interrupt)
I15	R7415	ILBL O15	0005 (external interrupt)
I16	R7416	ILBL O16	0005 (external interrupt)
I17	R7417	ILBL O17	0005 (external interrupt)

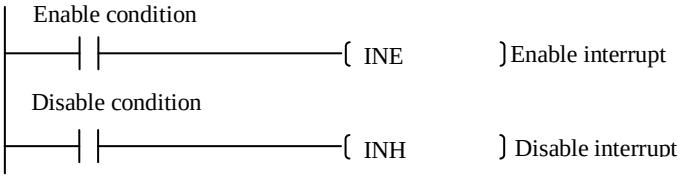
5-4-4 Writing your external interrupt subroutine

In the example, I10 is the external input point.



5-4-5 Enable/disable interrupt

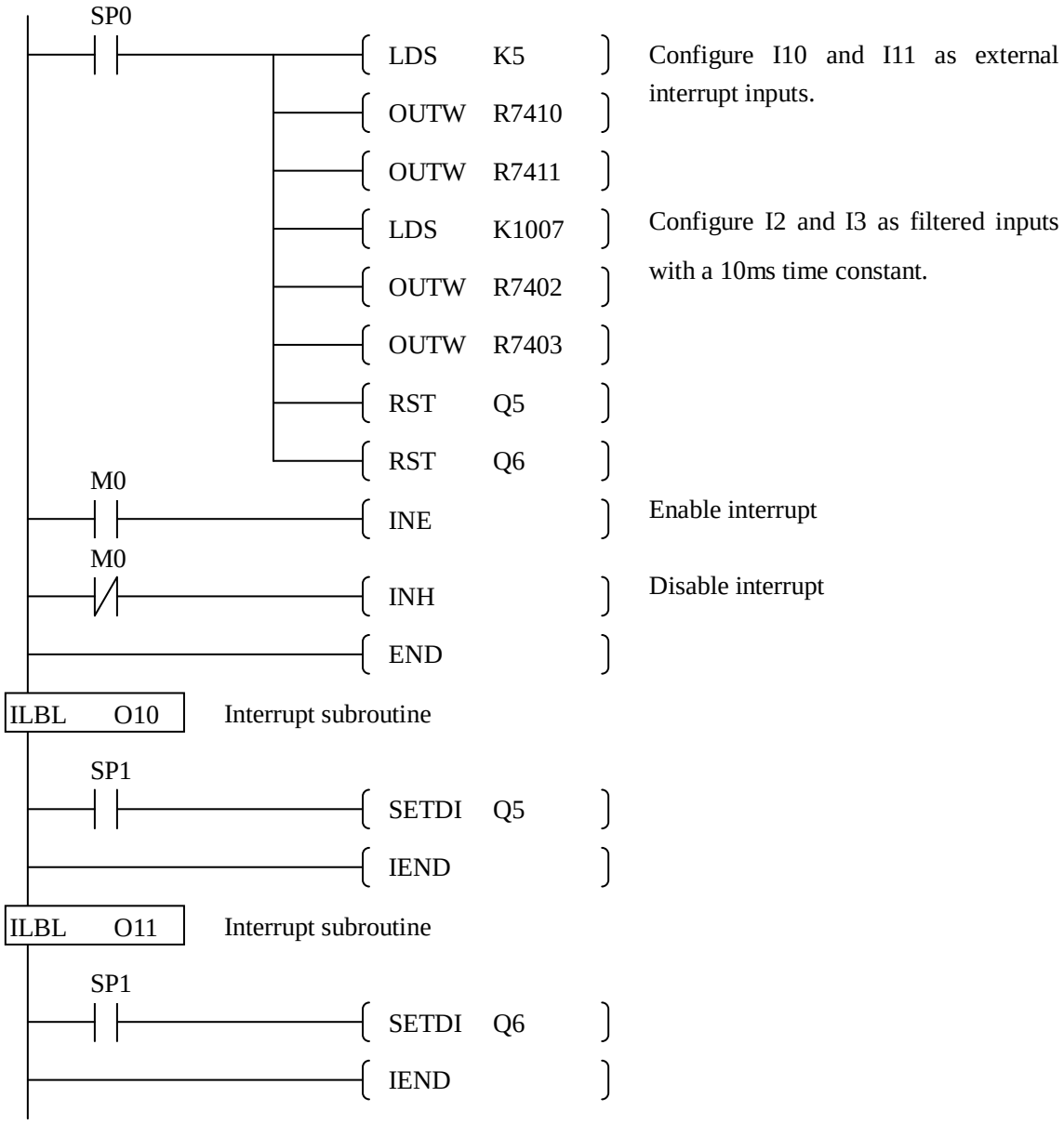
The interrupt remains disabled until the program executes an enable interrupt instruction INE. The interrupt remains enabled until the program executes a disable interrupt instruction.



- when entering RUN mode, interrupt is in disabled state.
- After INE is executed, interrupt is enabled (SP22 = ON).
- After INH is executed, interrupt is disabled (SP22 = OFF).

5-4-6 Program example: external interrupt

In the following example, I10 and I11 are configured as external inputs; I2 and I3 are configured as filtered inputs with a 10ms time constant. Set Q5 on if I10 is ON; set Q6 on if I11 is ON.

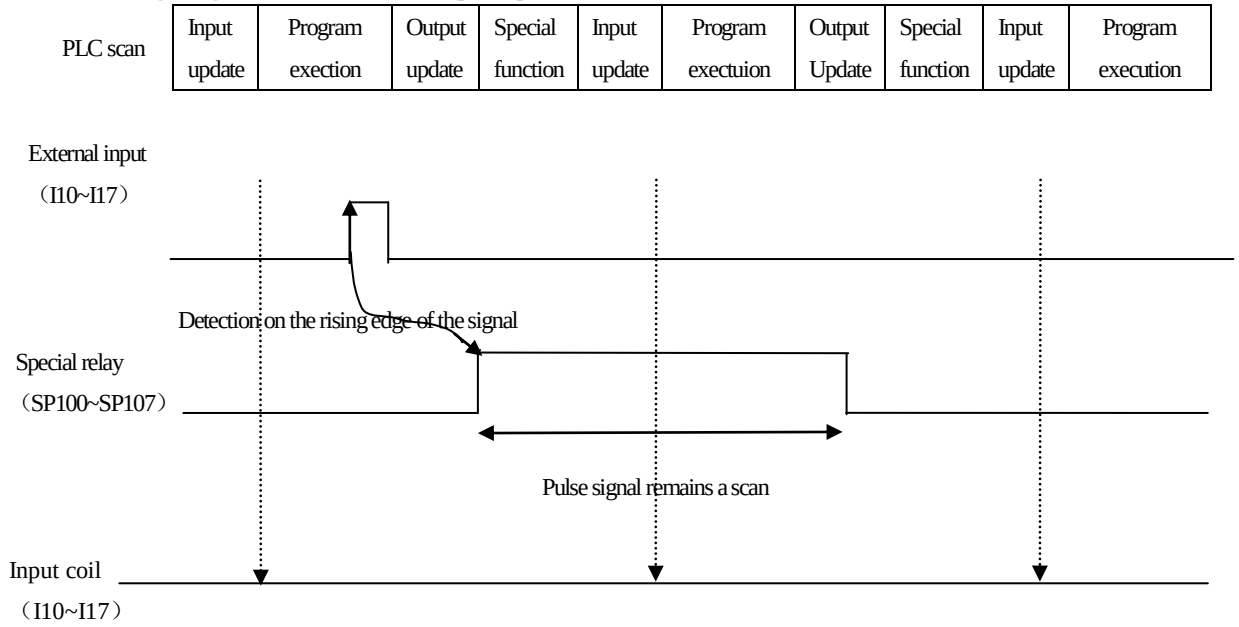


5-5 Pulse catch

5-5-1 Function description

The purpose of the pulse catch function is to enable the ladder program to see an input pulse which is shorter in duration than the current scan time. The HSIO circuit latches the input event for one scan. This contact automatically goes off after one scan.

The following diagram shows the concept of pulse catch.



In the process of input transmission, the input signal is not ON, so the input coil is always OFF.

5-5-2 Specifications

Channel	1	2	3	4
Pulse width T*	0.05ms			
Effective pulse	Rising edge			
Input point	I10	I11	I12	I13
State relay	SP100	SP101	SP102	SP103
Channel	5	6	7	8
Pulse width	0.05ms			
Effective pulse	Rising edge			
Input point	I14	I15	I16	I17
State relay	SP104	SP105	SP106	SP107

*Note: in order to ensure that pulses can be caught normally, the pulse width of each input point shall not be less than the pulse width T in the above table. In addition, in order to ensure that all pulses can be caught, the interval between two pulses of the same input signal shall not less than 2 scanning period.

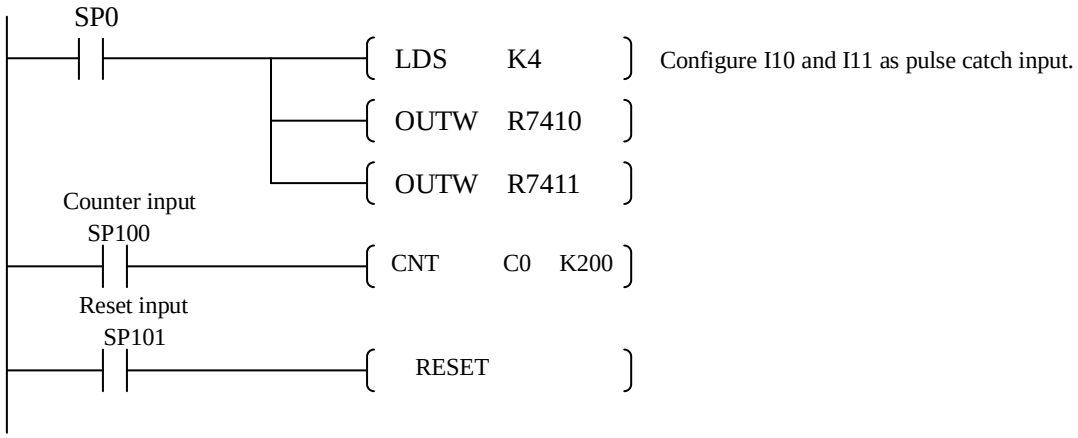
5-5-3 Function setting

NK1 PLC supports up to eight channels pulse catch input and uses I10~I17 as pulse catch input point. You must first setup the corresponding registers before using the pulse catch function.

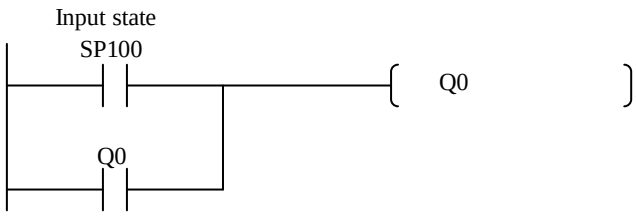
Input point	Setup register	Set value
I10	R7410	0004 (pulse catch)
I11	R7411	0004 (pulse catch)
I12	R7412	0004 (pulse catch)
I13	R7413	0004 (pulse catch)
I14	R7414	0004 (pulse catch)
I15	R7415	0004 (pulse catch)
I16	R7416	0004 (pulse catch)
I17	R7417	0004 (pulse catch)

5-5-4 Program example: pulse catch

A) Count the narrow pulse



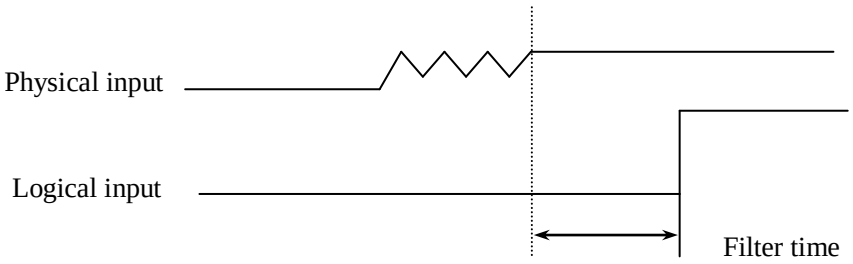
B) Use input state bit (SP100)



5-6 Inputs with filter

5-6-1 Common input with filter

Software filter is a function that rejecting narrow pulses and accept wide ones by software method. By setting the filter time in the special register, you can filter out the different width glitches in the input signal.



All high-speed inputs points I0~I17 on NK1 series PLC can be set to inputs with filter, its filter time range is 0~99ms.

High-speed inputs support software filter function only when they are configured as common inputs. When they are configured as high-speed count inputs, pulse catch inputs and external interrupt inputs or other special functions, they can not use software filter funciton.

5-6-2 Specifications

Input point	I00	I01	I02	I03
Register	R7400	R7401	R7402	R7403
Filter time	0～99ms			
Setting method	××07（××：0～99）			

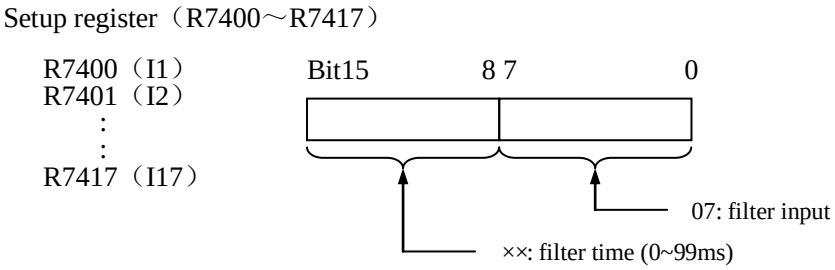
Input	I04	I05	I06	I07
Register	R7404	R7405	R7406	R7407
Filter time	0～99ms			
Setting method	××07（××：0～99）			

Input	I10	I11	I12	I13
Register	R7410	R7411	R7412	R7413
Filter time	0～99ms			
Setting method	××07（××：0～99）			

Input	I14	I15	I16	I17
Register	R7414	R7415	R7416	R7417
Filter time	0～99ms			
Setting method	××07（××：0～99）			

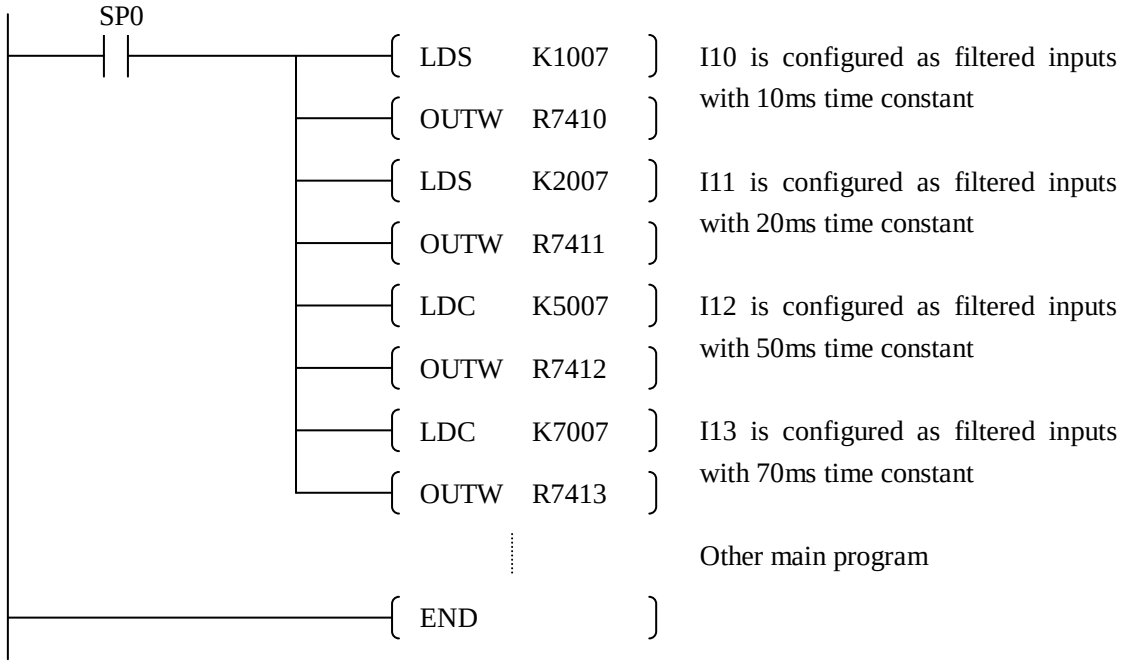
5-6-3 Filter function and filter time setup

On NK1 series PLC, input points I0~I17, each input has a special mode setup register used for setting its input function. To use the filter input function, you first need to write XX07 (HEX) into the corresponding register, 07 indicates that the input point is a filter input, XX is the filter time.



5-6-4 Program example: inputs with filter

In the following program, I10 is configured as filtered input with 10ms filter time; I11 is configured as filtered input with 20ms filter time; I12 is configured as filtered input with 50ms filter time; I13 is configured as filtered input with 70ms filter time.



5-7 Pulse output

5-7-1 Function description

DM/DD type NK1 CPU has 4 two-axis pulse output function points (Q0~Q3), they can achieve the following functions:

1. Positioning function which can achieve 2 channels CW/CCW or pulse/direction pulse output function, pulse rate is up to 100kHz;
2. PWM output function which can achieve 2 points PWM outuput (Q0 and Q2), the speed is up to 100kHz.
3. General NPN transistor output points (sink outputs).

Using pulse output positioning function of NK1 series PLC, NK1 PLC and pulse sequence type stepper or servo motor can constitute a simple positioning control system.

Using pulse output positioning function of NK1 series PLC, you can achieve home search, trapezoidal positioning, velocity control, interrupt positioning, multi-table positioning function and so on.

NK1series PLC achieves pulse output positioning function by using instructions. NK1 series PLC provides nine special positioning instructions which can achieve home search, trapezoidal positioning, velocity control, interrupt positioning, multi-table positioning, reset, stop and other functions. These instructions can be positioned anywhere in the main user program.

In order to use pulse output positioning function, you need to set some necessary parameters before writing the positioning control program. These parameters are stored in a group of special registers.

In addition, these registers also include current position, current speed, error or alarm information registers.

5-7-2 Specifications

Item	Specification
Control type	Position control (absolute or relative)
	Interrupt control
	Velocity control (speed and direction)
Number of axes	2
Position range	BCD: -9999999～9999999 HEX: -2147483647～2147483647
Pulse output type	CW/CCW Pulse/direction PWM output
Output point	Axis 1: Q0, Q1 Axis 2: Q2, Q3
Acceleration and deceleration mode	Trapezoidal acceleration and deceleration
Velocity range	1Hz ~ 100KHz (Resolution is 1Hz)
Acceleration and deceleration time	10ms ~ 9990ms
Software limit	Yes
Home search	Near point mode and the origin mode
External interrupt inputs for positioning	Axis 1：I6 Axis 2：I7
Abnormal stop	Yes
Change speed and position during operation	Yes

5-7-3 The basic parameters register

The following table lists the basic parameters registers.

Register	Use	Description
R7500	Operation mode	Set operation mode for each axis
R7501，R7502	Current position of 1# axis	Read-only (BCD or HEX)
R7503，R7504	Current speed of 1# axis	Read-only (BCD)
R7505，R7506	Maximum speed of 1# axis	1000Hz ~ 100000Hz (BCD)
R7507	Start-up speed of 1# axis	1Hz ~ 9999Hz (BCD)
R7510	Stop speed of axis 1# axis	1Hz ~ 9999Hz (BCD)
R7511	Acceleration time of 1# axis	10 ~ 9990ms (BCD)
R7512	Deceleration time of 1# axis	10 ~ 9990ms (BCD)
R7513	Abnormal stop time of 1# axis	10 ~ 9990ms (BCD)
R7514，R7515	Software positive limit of 1# axis	Minimum ~ maximum value allowed by the system *1
R7516，R7517	Software negative limit of 1# axis	Minimum ~ maximum value allowed by the system *1
R7520	Error and warning of 1# axis	Read-only, error and warning message of 1# axis
R7521，R7522	Current position of 2# axis	Read-only (BCD)
R7523，R7524	Current speed of 2# axis	Read-only (BCD)
R7525，R7526	Maximum speed of 2# axis	1000Hz ~ 100000Hz (BCD)
R7527	Start-up speed of 2# axis	1Hz ~ 9999Hz (BCD)
R7530	Stop speed of axis 2# axis	1Hz ~ 9999Hz (BCD)
R7531	Acceleration time of 2# axis	10 ~ 9990ms (BCD)
R7532	Deceleration time of 2# axis	10 ~ 9990ms (BCD)
R7533	Abnormal stop time of 2# axis	10 ~ 9999ms (BCD)
R7534，R7535	Software positive limit of 2# axis	Minimum ~ maximum value allowed by the system *1
R7536，R7537	Software negative limit of 2# axis	Minimum ~ maximum value allowed by the system *1
R7540	Error and warning of 2# axis	Read-only, error and warning message of 2# axis
R7541	Q0 PWM output pulse width time	HEX 0~FFFF (BCD：0~65535)
R7542	Q0 PWM output cycle time	HEX 0~FFFF (BCD：0~65535)
R7543	Q2 PWM output pulse width time	HEX 0~FFFF (BCD：0~65535)
R7544	Q2 PWM output cycle time	HEX 0~FFFF (BCD：0~65535)
R7545	Flag that forces to modify the basic parameters of pulse output	0x005A: force to write the parameters of 1# axis 0xA500: force to write the parameters of 2# axis 0xA55A: force to write the parameters of 1# and 2# axis 0x5555: Parameters normally writing

		is completer. Other: wrong parameter register ID

*1: The minimum and maximum value allowed by the system is the minimum and maximum position value of NK1 PLC pulse output function.

In the first scan after NK1 PLC entering RUN mode, the system automatically reads the basic parameters of two axes in registers to initialize the basic parameters of the pulse output operation. After running the program, these basic parameters can not be modified.

The basic parameters of pulse output can be set by a program segment enabled by SP0 or by writing them to the registers in PLC STOP mode. The parameters modified during the RUN mode of PLC can not immediately work, you need to re-run the user program or using R7545 (flag that forces to modify the basic parameter of pulse output).

The factory default values for all the basic parameters are all 0. R7545 = 5555 represents the basic parameters have been written correctly.

Note that the current position, current speed error and alarm information in the above table are read-only and do not need special setting.

1. Operation mode R-memory: R7500

The operation mode of the two axes is set separately. Each axis occupies 4 bits, BIT3 ~ 0 corresponds 1# axis; BIT7 ~ 4 corresponds 2# axis.

1# axis mode bit	BIT3	BIT2	BIT1	BIT0
2# axis mode bit	BIT7	BIT6	BIT5	BIT4
Function	Data format of positioning output: 0: BCD 1: HEX	Pulse output mode 0: positioning mode 1: PWM mode	Positioning pulse output mode 0: CW/CCW mode 1: pulse/direction mode	Bit indicates whether pulse output is enabled 0: pulse output is not enabled 1: pulse output is enabled

Example: 1# axis is set to CW/CCW position mode and BCD data format; 2# axis is set to PWM output mode. You need set:

R7500 = 0x0051

Note: after enabling PWM output function of the axis, only one point of the axis generates pulse output signal (1# axis: Q0, 2# axis: Q2), the other point of the axis has no effect (1# axis: Q1, 2# axis: Q3).

2. Current position R-memory: 1# axis: R7502 / R7501 2# axis: R7522 / R7521

Automatically store the current position value of pulse output.

3. Current speed R-memory: 1#axis: R7502 / R7501 2#: R7522 / R7521

Automatically store the current speed value of pulse output.

4. Maximum speed R-memory: 1# R7506 / R7505 2#: R7526 / R7525

Set the allowed maximum speed value during operation.

5. Start-up speed R-memory: 1# axis: R7507 2# axis: R7527

Set start-up speed. When NK1 PLC starts pulse output operation, its speed will directly jump to the start-up speed from zero.

6. Stop speed R-memory: 1# axis: R7510 2# axis: R7530

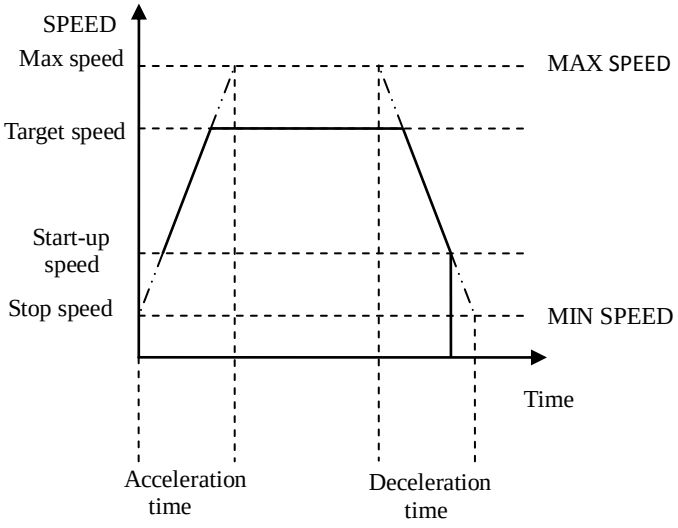
Set the minimum speed when the pulse output operation has been completed.

7. Acceletation / deceleration time R-memory: 1# axis: R7511/R7512 2# axis: R7531/R7532

Set acceletation and deceleration time.

After setting the maximum speed, stop speed, acceleration and deceleration time, the system will automatically calculate the acceletation and deceleration. All the acceleration and deceleration of NK1 PLC are calculated in accordance with the maximum speed and stop speed.

The right diagram shows the relationship among the various speeds.



8. Abnormal stop time R-memory: 1# axis: R7513 2# axis: R7533

Set abnormal stop time. Note that abnormal stop time is the time from maximum speed to stop speed, rather than the time from the current speed to stop speed.

9. Software positive limit R-memory: 1# axis: R7515/R7514 2# axis: R7535/R7534

Set the maximum number of positive pulse. When the current position value reaches the data in the positive limit R-memory, the system will immediately stop pulse output and set "positioning exception flag", store the error code into "error warning register" at the same time.

Note 1: if a positioning exception occurs, the positioning exception flag (1# axis: M1702; 2# axis: M1722) must be clear before starting a new positioning operation.

Note 2: during velocity control instruction operation, even if the current position value reaches the data in the positive limt R-memory, the system will not give the error and stop operation.

10. Software negative limit R-memory: 1# axis: R7517 R7516 2# axis: R7537 R7536

Set the minimum number of negative pulse. When the current position value reaches the data in the negative limt R-memory, the system will immediately stop pulse output and set "positioning exception flag", store the error code into "error warning register" at the same time.

Note 1: if a positioning exception occurs, the positioning exception flag(1# axis: M1702; 2# axis: M1722) must be clear before starting a new positioning operation.

Note 2: during velocity control instruction operation, even if the current position value reaches the data in the negative limit R-memory, the system will not give the error and stop operation.

11. Error and warning R-memory: 1# axis: R7520 2# axis: R7540

When using the NK1 pulse output function, if an error has occurred in the parameters setting or instruction operation, the system will immediately stop positioning operation and set "positioning exception flag", store the error code into "error warning register" at the same time. During normal operation, the system will send same value to indicate normal operation.

The system will automatically convert the error register number to hexadecimal number and write it into R7545 (flag that forces to modify the basic parameters of pulse output).

The following table lists the meaning of error and warning codes in registers.

1) Error description of basic parameters setup

Error description	Error code	
	Non-BCD error	Parameters out of setting range
Lower bits setting error of max speed	0001	0002
Upper bits setting error of max speed	0101	0102
Setting error of start-up speed	0201	0202
Setting error of stop speed	0301	0302
Setting error of acceleration time	0401	0402
Setting error of deceleration time	0501	0502
Setting error of abnormal stop time	0601	0602
High bits setting error of software positive limit	0701 *1	0702
Low bits setting error of software positive limit	0801 *1	0802
High bits setting error of software negative limit	0901 *1	0902
Low bits setting error of software negative limit	0A01 *1	0A02

*1: The none-BCD error is only detected when using BCD data format.

2) Error description of the instructions

Error description		Error code	
		Non-BCD error	Parameters out of setting range
Home search instruction error	Home search method	1121	1122
	Home search speed	1131	1132
	Crawling speed	1141	1142
Absolute positioning instruction error	Target location	1221 *1	1222
	Positioning velocity	1231	1232
Relative positioning instruction error	Relative position	1321 *1	1322
	Positioning velocity	1331	1332
Speed control instruction error	Direction	1421	1422
	Positioning velocity	1431	1432
Interrupt positioning instruction error	Direction	1521	1522
	Positioning velocity	1531	1532
	Interrupt positioning distance	1541 *1	1542
Current position change instruction error	New current location	1621 *1	1622
Stop instruction error	Stop code	1821	1822
Multi-position instruction error	Number of segments	1901	1902
	Target location	1911 *1	1912
	Positioning velocity	1921	1922
	Acceleration time	1931	1932
	Deceleration time	1941	1942
	Residence time	1951	1952
	Multi-step instructions	1960	1960

*1: The none-BCD error is only detected when using BCD data format.

3) Over limit error description

Error description	Error code
Reach positive limit error	1005
Reach negative limit error	1006
Home search failure	1007
Permitting condition is OFF during positioning	1008

4) Normal operation flag in positioning mode

Description	Code
Normal operation in positioning mode	0000

5) Normal operation flag in non-positioning mode

Description	Code
Q00~Q01PWM output mode / Q02~Q03PWM output mode	AAAA
Q00~Q01 normal output / Q02~Q03 normal output	FFFF

10. PWM output pulse width time R-memory: 1# axis: R7541 2# axis: R7543

When the relevant axis is the PWM output, set the number of pulse width time block. Time unit can select 1us or 40us.

11. PWM output pulse cycle time R-memory: 1# axis: R7542 2# axis: R7544

When the relevant axis is the PWM output, set the number of pulse cycle time block. Time unit can select 1us or 40us.

12. Flag R-memory forcing to modify the basic pulse output parameters during user program operation: R7545

R7545 is used to modify the basic pulse output parameters during user program operation. When you modifying the parameters, the system will forcibly stop pulse output operation.

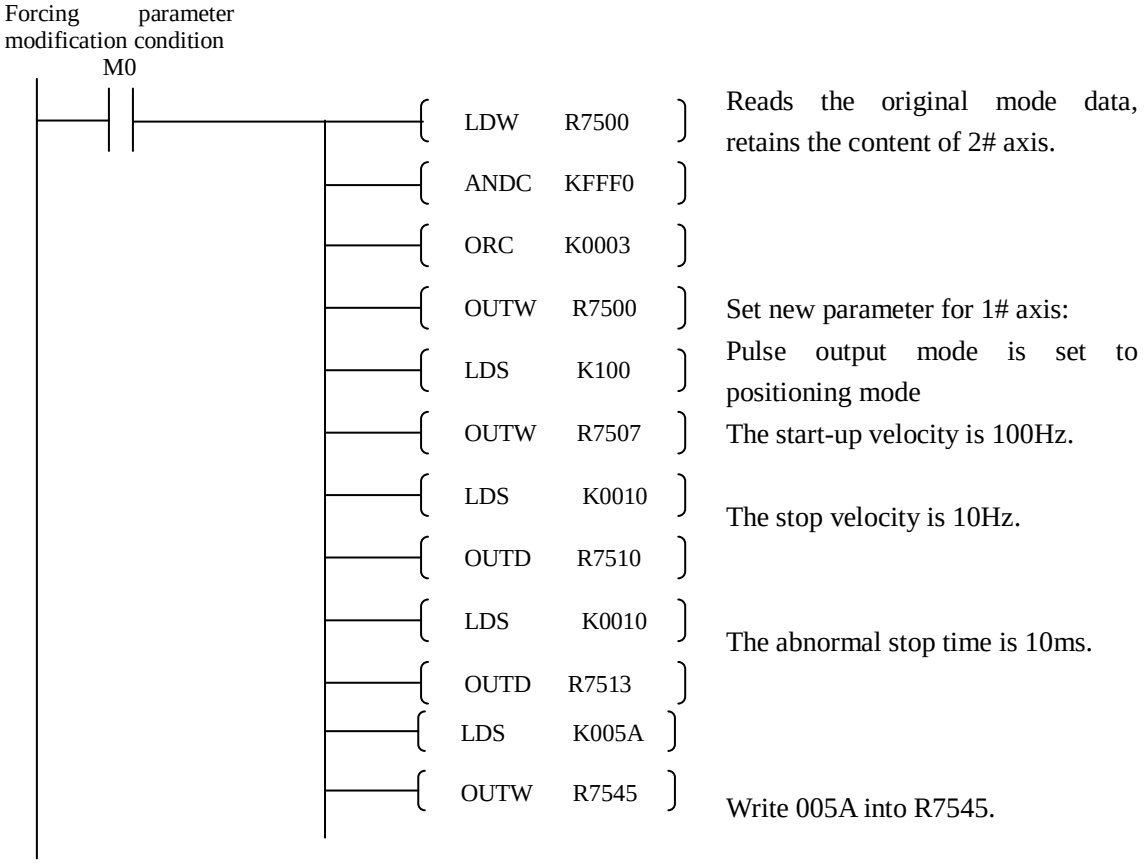
You can modify the parameters of two axes simultaneously, you can also modify the parameters of one axis (the other axis is not affected). The system is informed to modify the appropriate parameters by writing different values into R7545.

1） R7545 = 0XA55A, modify the parameters of two axes simultaneously, the corresponding parameter registers are:
R7500，R7505~R7517，R7525~R7537， R7541~R7544.

2） R7545 = 0X005A, only modify the parameters of 1# axis, the corresponding parameter registers are:
R7500，R7505~R7517， R7541~R7542.

3） R7545 = 0XA500, only modify the parameters of 2# axis, the corresponding parameter registers are:
R7500， R7525~R7537， R7543~R7544.

Program example: only modify the parameters of 1# axis.



Because the system will automatically stop the pulse output operation when the basic parameters are modifying (the influence for pulse output is the same with turning the operation mode from RUN to STOP and to RUN again), this may cause some unexpected operating state. To prevent the unexpected operating state, please do not modify the parameters during user program operation. If you have to modify the parameters, make sure that pulse output is not operating.

5-7-4 Special auxiliary relay of NK1 pulse output

NK1 PLC assigns a bank of 15 special relays for each pulse output channel, these relays are used to set or mark pulse output operation. The following table lists these relays.

Special relay		Name and function	Property
1# axis	2# axis		
M1700	M1720	Flag during positioning instruction execution 0: positioning instruction execution condition is not met. 1: positioning instruction execution condition is met.	Read-only
M1701	M1721	Flag during positioning instruction operation 0: positioning operation is not in progress. 1: positioning operation is in progress.	Read-only
M1702	M1722	Positioning exception flag (cleared by MT_RSTE instruction) 0: normal 1: abnormal	Read-only
M1703	M1723	Home search completion flag 0: unfinished 1：finished	Read-only
M1704	M1724	Specifies the direction of home search 0: positive direction 1: opposite direction	Read and write
M1705	M1725	Positive limit flag (Normal operation, immediately stop on the arrival of limit position) 0: does not arrive 1: arrives	Read and write
M1706	M1726	Reverse limit flag (Normal operation, immediately stop on the arrival of limit position) 0: does not arrive 1: arrives	Read and write
M1707	M1727	Near-point signal logic reverse 0：don't reverse 1：Reverse	Read and write
M1710	M1730	Interrupt signal logic reverse 0：don't reverse 1：Reverse	Read and write
M1711	M1731	Whether with the acceleration and deceleration in velocity control mode 0：with the acceleration and deceleration 1：without the acceleration and deceleration	Read and write
M1712	M1732	Interrupt signal input 0: untriggered 1: triggered	Read and write
M1713	M1733	Near-point signal input 0: near-point signal is not established 1: near-point signal is established	Read and write
M1714	M1734	Home signal input 0: home signal is not established 1: home signal is established	Read and write
M1715	M1735	PWM output enable control 0: stop PWM output 1: start PWM output	Read and write
M1716	M1736	Set PWM output time unit 0: 1μs 1: 40μs	Read and write

5-7-5 NK1 positioning instructions

NK1 series PLC provides a nine positioning instructions, the following table lists these instructions.

No.	Instruction name	Instruction code	Instruction function
1	Home search	MT_ORG	Execute home search operation
2	Absolute value positioning	MT_ABSC	Move to the target position in trapezoidal profile
3	Relative value positioning	MT_INCC	Output the corresponding pulses in trapezoidal profile
4	Velocity control	MT_SPDC	Output pulse according to the specified velocity, variable velocity.
5	Interrupt positioning	MT_INTC	Detect prescribed number of pulses.
6	Change current Position	MT_PSET	Change the current position value
7	Reset	MT_RSTE	Reset the error flag and clear error and waring R-memory.
8	Stop	MT_STPM	Stop positioning operation
9	Multi-positioning table	MT_TABL	Execute multi-stage ladder positioning operation

Notes when using the positioning instruction:

1) When a positioning instruction's execution condition is met, while executing the instruction, the system automatically set "flag during positioning instruction execution" (M1700 / M1720) and "flag during positioning instruction operation" (M1701 / M1721).

2) The positioning instruction's execution condition must always be kept true when the positioning instruction is executing. If the condition turns to OFF before the exection is completed, the system will stop the positioning operation, slow down and stop, automatically reset "flag during positioning instruction execution" (M1700 / M1720) and "flag during positioning instruction operation" (M1701 / M1721).

3) After completing the positioning operation specified by a positioning instruction, the system will automatically reset "flag during positioning instruction operation" (M1701 / M1721).

4) After a positioning command is executed, the user program requires resetting the positioning instruction's execution condition; the system will automatically reset "flag during positioning instruction execution" (M1700 / M1720) to prepare for the execution of the next positioning instruction.

5) In addition to the stop instruction (MT_STPM) can be executed during the execution of other positioning instructions, the other positioning instructions can only be executed alone at the same time. When there are multiple positioning instruction's execution conditions are met, according to the instruction's positon in the user program, they are executed in the following rules:

If it is the first execution of positioning instruction, the first positioning instruction whose execution condition is met will be executed; when a positioning instruction has been executed, the first positioning instruction whose execution condition is met behind it will be executed. If it is just the last positioning instruction has been executed, the first positioning instruction whose execution condition is met will be

executed.

6) Each positioning instruction has several action parameters, which are divided into two categories: can be changed and can not be changed during operation.

When the positioning instruction is started, if there is a parameter setting error, the positioning instruction will not be started. The system will automatically set "positioning exception flag" (M1702 / M1722), and store the error code into "error warning register".

When you change the parameters during operation, if there is a parameters setup error, the system will stop the positioning operation, slow down and stop, automatically set "positioning exception flag" (M1702 / M1722) and reset "flag during positioning instruction operation" (M1701 / M1721).

7) During positioning operation, if the current position reaches positive and negative limit position, the system will immediately stop the positioning operation, automatically set "positioning exception flag" (M1702 / M1722), store the error code into "error warning register" and reset "flag during positioning instruction operation" (M1701 / M1721).

8) The parameters specified by R or P can be modified during positioning operation. Only the content of the register can be modified, the instruction can not be modified; the parameter specified by K can not be specified.

9) The modification of the parameters which can not be changed during operation is not valid.

10) The positioning parameter also supports BCD and HEX data format.

When BCD data format, the data represent rang is 89999999~9999999, the corresponding data range is -9999999~9999999.

When HEX data format, the data represent range is 0xFFFFFFFF~0x7FFFFFFF, the corresponding data range is -2147483647~2147483647.

Chapter 6 PID loop operation

6-1 Main features

The NK1 CPUs process loop control offers a sophisticated set of features to address many application needs. The main features are:

• Up to 16 loops, individual programmable sample rates
• Manual/ automatic/ cascaded loop capability available
• Two types of bumpless transfer available
• Full-featured alarms
• Ramp/soak generator with up to 16 segments
• Auto Tuning

You can select and configure up to sixteen loops. All sensors and actuators are wired directly to NK1 CPU or expansion unions. All process variables, gain values, alarm levels, etc., associated with each loop reside in a Loop Variable Table in the CPU. The CPU reads process variable (PV) inputs during each scan. Then it makes PID loop calculations during a dedicated time slice on each PLC scan and update the control output value. The control loops use the Proportional-Integral-Derivative (PID) algorithm to generate the control output command.

The following table lists the feature and specifications of NK1 PID:

PID loop feature	Specifications
Number of loops	Up to 16
CPU R-memory needed	32 words (R locations) per loop selected, 64 words if using ramp/soak
PID algorithm	Position or velocity form of the PID equation
Control output polarity	Selectable direct-acting or reverse-acting
Error term curves	Selectable as linear, square root of error, and error squared
Loop update rate (time between PID calculation)	0.05 to 99.99 seconds, user programmable
Minimum loop update rate	0.05 seconds for 1 to 4 loops, 0.1 seconds for 5 to 8 loops, 0.2 seconds for 9 to 16 loops
Loop modes	Automatic, Manual (operator control), or Cascade control
Ramp/Soak generator	Up to 8 ramp/soak steps (16 segments) per loop with indication of ramp/soak step number
PV curves	Select standard linear, or square-root extract (for flow meter input)
Set point limits	Specify minimum and maximum setpoint values
Process variable limits	Specify minimum and maximum process variable values
Proportional Gain	Specify gains of 0.01 to 99.99
Integrator (Reset)	Specify reset time of 0.1 to 999.8 in units of seconds or minutes
Derivative (Rate)	Specify the derivative time from 0.01 to 99.99 seconds
Rate Limits	Specify derivative gain limiting from 1 to 20
Bumpless transfer I	Automatically sets the bias equal to the control output and the setpoint equal to the process variable when the control switches from manual to automatic.

Bumpless transfer II	Automatically sets the bias equal to the control output when control switches from manual to automatic
Step bias	Provides proportional bias adjustment for large setpoint changes
Anti-windup (Freeze Bias)	For position form of PID, this inhibits integrator action when the control output reaches 0% or 100% (speeds up loop recovery when output recovers from saturation)
Error deadband	Specify a tolerance (plus and minus) for the error term (SP-PV), so that no change in control output value is made

Alarm feature	Specifications
PV alarm deadband	0.1%-5%
PV alarm points	Select PV alarm settings for Low-low, Low, High, and High-high conditions
PV deviation	Specify alarms for two ranges of PV deviation from the setpoint value
Rate of change	Detect when PV exceeds a rate of change limit you specify

Loop table word definitions

These are the loop parameters associated with each of the sixteen loops. The parameters are listed in the following table. The address offset is in octal, to help you locate specific parameters in the loop table. For example, if a table begins at R2000, then the location of the reset (integral) term is Addr+11, or R2011. Do not use the Word # (in the first column) to calculate addresses.

Word #	Address+ Offset	Description	Format	Change On-the-fly
1	Addr+0	PID Loop Mode Setting 1	Bit	Yes
2	Addr+1	PID Loop Mode Setting 2	Bit	Yes
3	Addr+2	Setpoint Value (SP)	Word/Binary	Yes
4	Addr+3	Process Variable (PV)	Word/Binary	Yes
5	Addr+4	Bias (Integrator) Value	Word/Binary	Yes
6	Addr+5	Control Output Value	Word/Binary	Yes
7	Addr+6	Loop Mode and Alarm Status	Bit	—
8	Addr+7	Sample Rate Setting	Word/BCD	Yes
9	Addr+10	Gain (Proportional) Setting	Word/BCD	Yes
10	Addr+11	Reset (Integral) Time Setting	Word/BCD	Yes
11	Addr+12	Rate (Derivative) Time Setting	Word/BCD	Yes
12	Addr+13	PV Value, Low-low Alarm	Word/Binary	No*
13	Addr+14	PV Value, Low Alarm	Word/Binary	No*
14	Addr+15	PV Value, High Alarm	Word/Binary	No*
15	Addr+16	PV Value, High-high Alarm	Word/Binary	No*
16	Addr+17	PV Value, deviation alarm (YELLOW)	Word/Binary	No*
17	Addr+20	PV Value, deviation alarm (RED)	Word/Binary	No*
18	Addr+21	PV Value, rate-of-change alarm	Word/Binary	No*

Continued

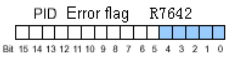
19	Addr+22	PV Value, alarm hysteresis setting	Word/Binary	No*
20	Addr+23	PV Value, error deadband setting	Word/Binary	Yes
21	Addr+24	PV low-pass filter constant	word/BCD	Yes
22	Addr+25	Loop derivative gain limiting factor setting	word/BCD	No**
23	Addr+26	SP value lower limit setting	Word/Binary	Yes
24	Addr+27	SP value upper limit setting	Word/Binary	Yes
25	Addr+30	Control output value lower limit setting	Word/Binary	No**
26	Addr+31	Control output value upper limit setting	Word/Binary	No**
27	Addr+32	Remote SP Value R-Memory Address Pointer	Word/Hex	Yes
28	Addr+33	Ramp/Soak Setting Flag	Bit	Yes
29	Addr+34	Ramp/Soak Programming Table Starting Address	Word/Hex	No**
30	Addr+35	Ramp/Soak Programming Table Error Flags	Bit	No**
31	Addr+36	Reserve		
32	Addr+37	Reserve		

* Read data only when alarm enable bit changes from 0 to 1.

** Read data only when PLC mode change..

There isn't a PID instruction that can be used in RLL, such as a block, to setup the PID loop control. Instead, the CPU reads the setup parameters from system R-memory locations. These locations are shown in the table below for reference only; they can be used in a RLL program if needed.

Address	Setup parameters	Data type	range	Read/Write
R7640	Loop parameter table pointer	Octal	R2000~R7340, R10000~R36740	Read/Write
R7641	Number of loops	BCD	0~16	Read/Write
R7642	Loop error flags	Bit	0 or 1	Read only



The following table lists errors recorded in R7642.

Bit	Error description (0 = no error, 1 = error)
0	The starting address (in R7640) is out of the lower R-memory range.
1	The starting address (in R7640) is out of the upper R-memory range.
2	The number of loops selected (in R7641) is greater than 16.
3	The loop table extends past (straddles) the boundary at R7377. Use an address closer to R2000.
4	The loop table extends past (straddles) the boundary at R37777. Use an address closer to R10000.

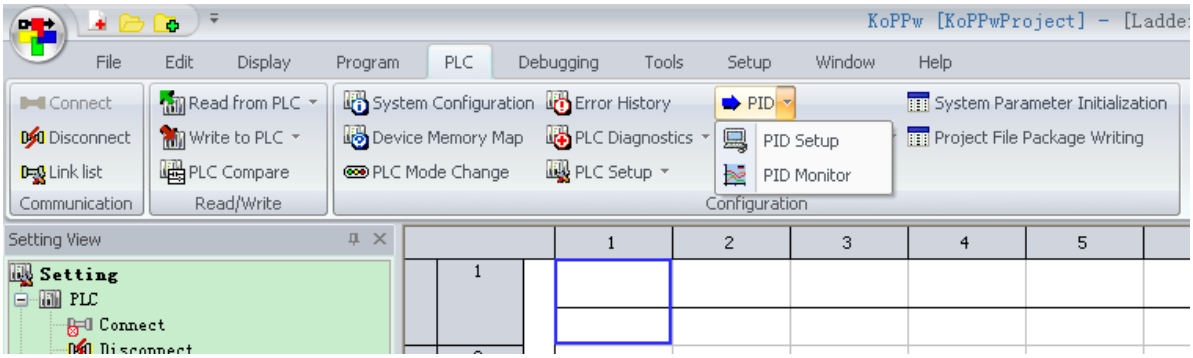
If the CPU is in Run mode and R7642=0000, there is no programming error.

You can setup and store the PID parameters either directly in your RLL program or by using the PID Setup in KPPSoft. Using KPPSoft is the simplest way to setup the parameters.

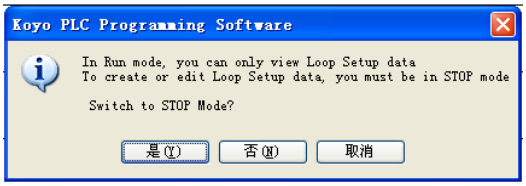
6-2 PID operation in KPPSoft

To setup the PID parameters, the NK1 PLC must be powered up and connected to the programming computer. The parameters can only be entered in PID setup when the PLC is in the program mode. Once the parameters have been entered and saved for each loop, changes through the PID setup can be made, but only in program mode. You can type the beginning address in the PID Table Address dialog found when the PID Setup is opened in KPPSOFT. This can be seen in the diagram below. After the address has been entered, the memory range will appear. The necessary PID parameters for a basic loop operation for each loop can be setup with the dialogs made available.

Click on PLC > PID> PID Setup to access the PID Setup dialog, as shown below:

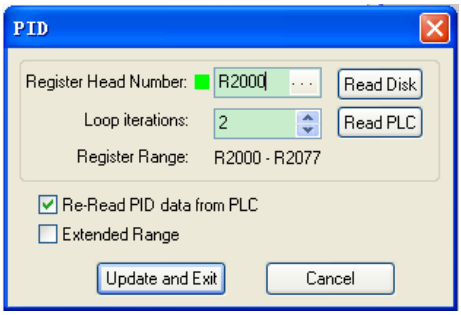


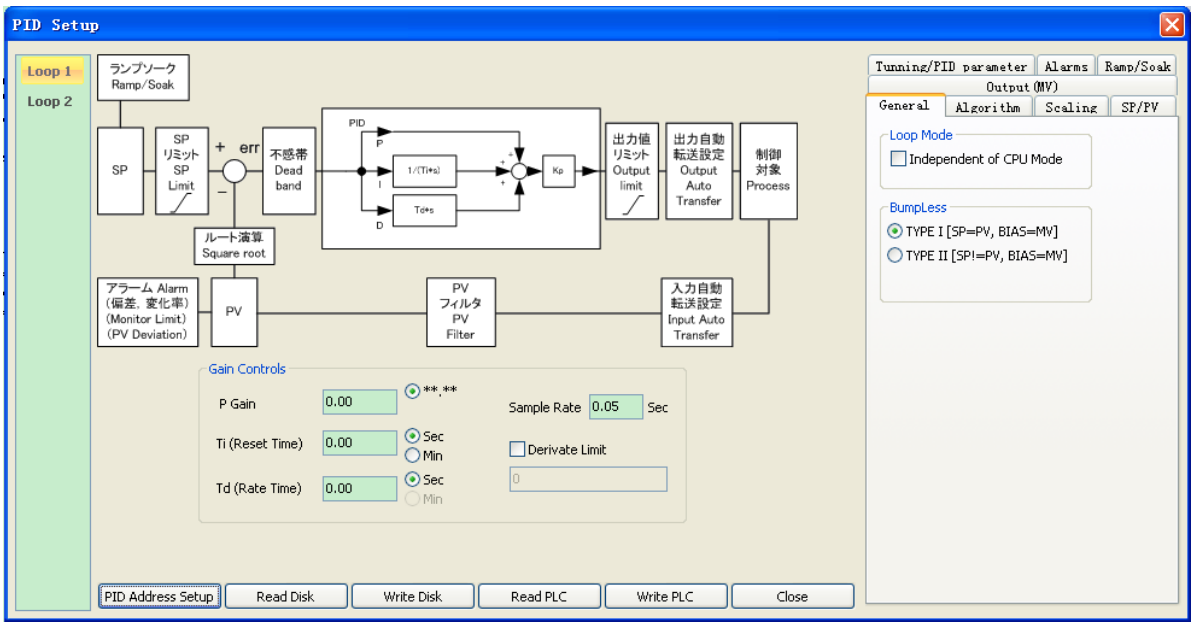
If NK1 PLC is in RUN mode, window shown at right will pop up, select [是], enter the next step.



If the connected PLC has not been setup the window shown at right will pop up.

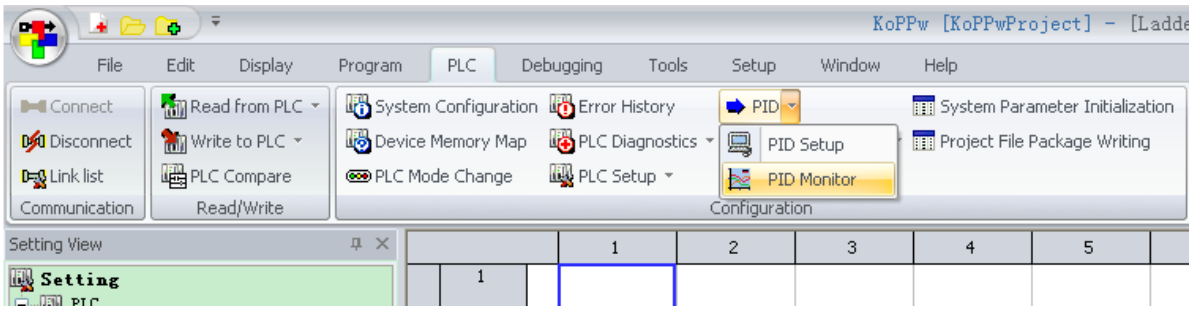
After setup, click “update and exit” button to save the settings. KPPSoft automatically pop “PID Setup” window.



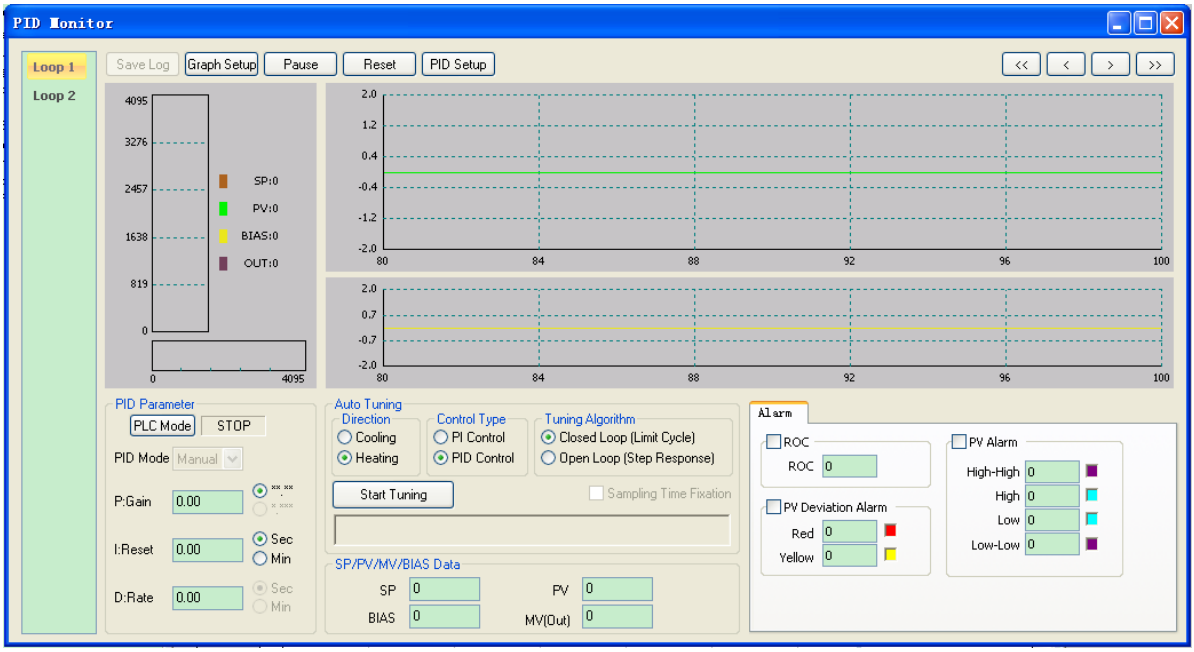


"PID Setup" includes eight parameter settings pages: general, algorithm, scaling, SP/PV, output, tunNing/PID parameter, alarms and pamp/soak.

After the PID parameters are setup, you can open the PID monitor window, monitor the status of the PID. Click on PLC > PID> PID Setup to access the PID Monitor dialog, as shown below:



"PID Monitor" window:



Please refer to relevant PID manual for details of PID Setup and PID Monitor window.

PID example

NK1 system consists of a CPU, one digiatal module NK1-16CDR and one analog module NK1-8AD4DA. The first input channel of NK1-8AD4DA is used for PV value of PID loop. The first output channel of NK1-8AD4DA is used for control output of PID loop.

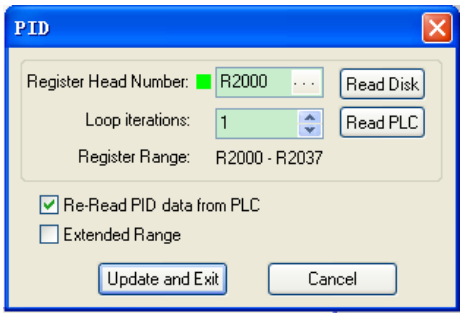
Steps of PID implementation:

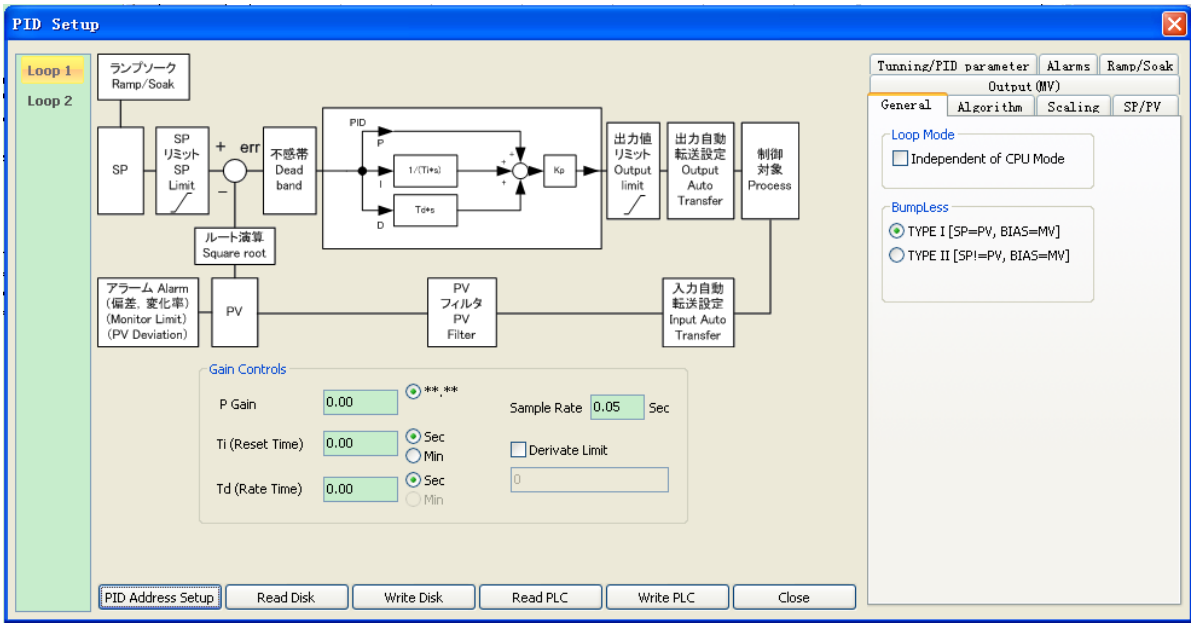
1. Setup PID parameter address and loop number

Register head number: R2000

Loop iterations: 1

Click “Update and Exit” button, access PID Setup window.

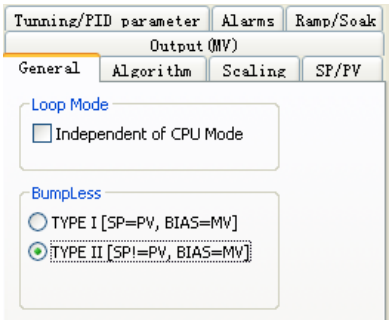




2. Setup 8 pages of parameters

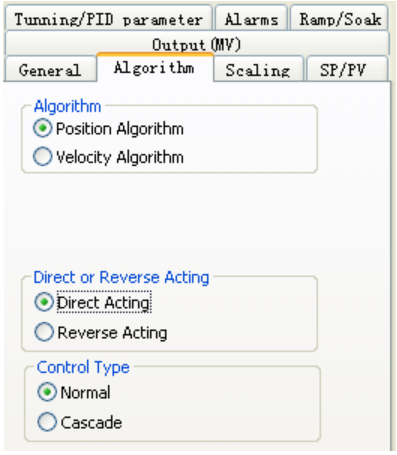
① “General” page

Setup parameters according to the following diagram:



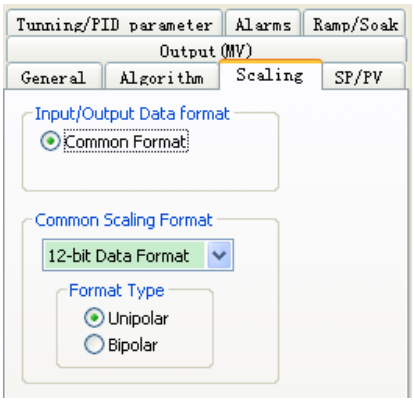
② “Algorithm” page

Setup parameters according to the following diagram:



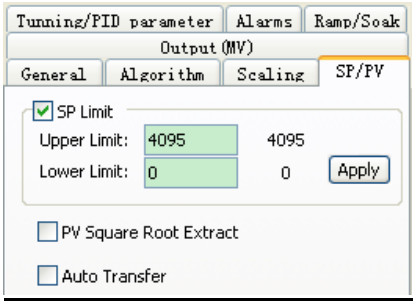
③ “Scaling” page

Setup parameters according to the following diagram:



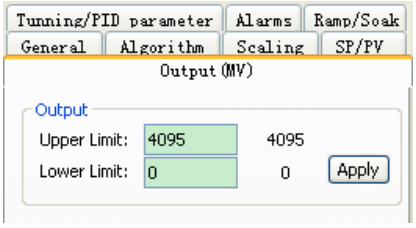
④ “SP/PV” page

Setup parameters according to the following diagram:



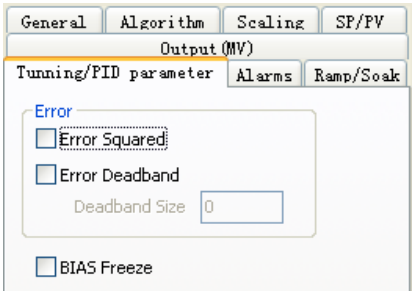
⑤ “Output” page

Setup parameters according to the following diagram:



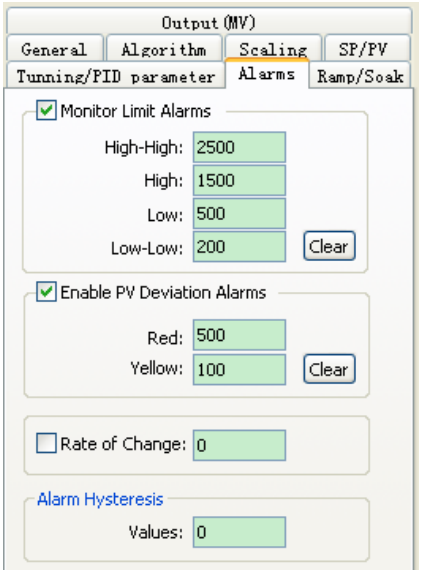
⑥ “Tunning/PID parameter” page

Setup parameters according to the following diagram:



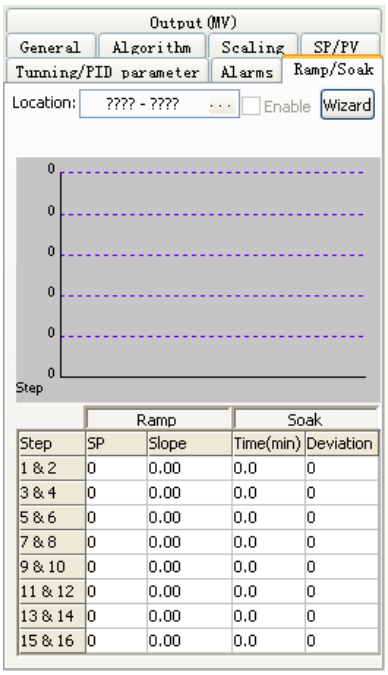
⑦ “Alarms” page

Setup parameters according to the following diagram:



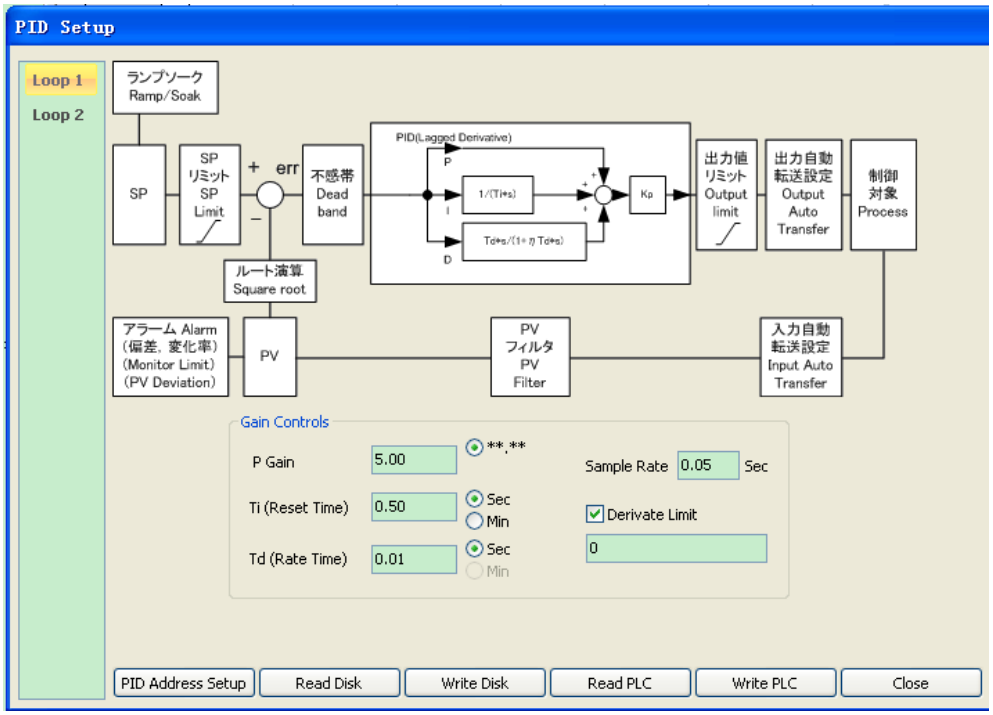
⑧ “Ramp/Soak” page

Setup parameters according to the following diagram:



	Ramp		Soak	
Step	SP	Slope	Time(min)	Deviation
1 & 2	0	0.00	0.0	0
3 & 4	0	0.00	0.0	0
5 & 6	0	0.00	0.0	0
7 & 8	0	0.00	0.0	0
9 & 10	0	0.00	0.0	0
11 & 12	0	0.00	0.0	0
13 & 14	0	0.00	0.0	0
15 & 16	0	0.00	0.0	0

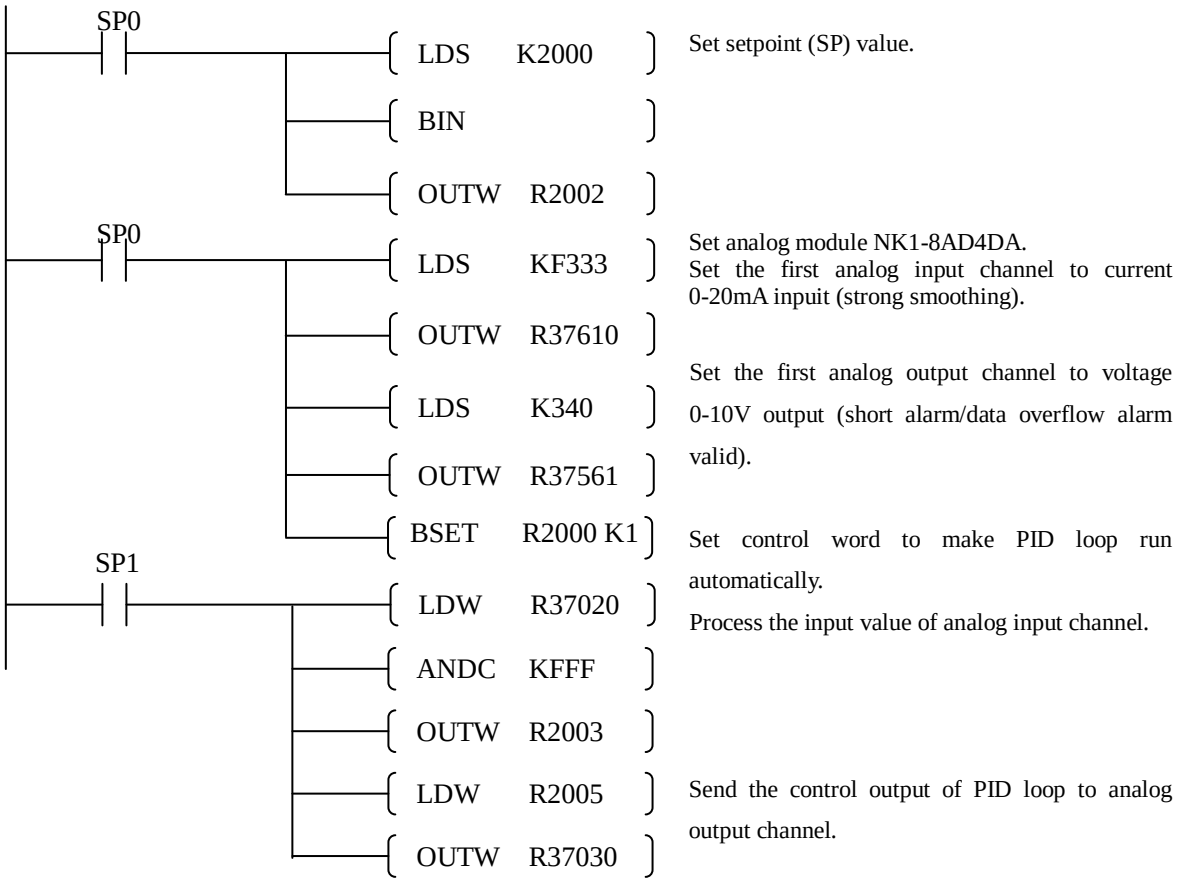
After all setting of the above parameters is completed, click on “Write PLC” button to save the setup data to the PLC. You can also click on “Write Disk” button to save the setup data to the computer disk.



3. PID user program

After the PID loop, or loops, has been setup with KPPSOFT, you will need to edit your RLL program to include the rungs needed to setup the analog I/O module to be used by the PID loop(s).

The following example program shows how an input/output analog module, NK1-8AD4DA, is used and setup for a PID loop. This example assumes that the PID table for loop 1 has a beginning address of R2000.



Note: the above example uses only one PID loop. When the application has multiple PID loops, you need to setup parameters for each PID loop and write the necessary user program.

Chapter 7 NK1L-CPU40DR introduction

NK1L-CPU40DR is the simplified version of PLC based on NK1-CPU40DR. Its main specifications are as follows:

- The CPU has bulit-in one RS-232C port ,one RS-485 port and one USB port. The specifications and the usage of this ports are the same as them in the NK1-CPU40DR.
- The CPU has extended functionality, supports up to 14 expansion units. The specifications and the usage are the same as NK1-CPU40DR's.
- The CPU has 6 high-speed inputs, supporting up to three channels A/B two-phase or five channels single-phase counters. The counting speed is up to 10kcps and the usage is the same as NK1-CPU40DR's.
- The 8 common inputs I10~I17 of the CPU can be set to external interrupt or pulse catch inputs. The specifications and the usage are the same as NK1-CPU40DR's.
- The CPU has one channel independent timed interrupt. The specifications and the usage are the same as NK1-CPU40DR's.
- The CPU has PID functionality. The specifications and the usage are the same as NK1-CPU40DR's.
- The CPU has no Ethernet port .
- The CPU does not support the function board and the battery, has no calendar functionality, and does not support TIME and DATE instructions.
- The CPU supports up to 1k words retentive memory. Five function memory M, R, T, C, S can be set to retain.

The retentive memory of NK1L-CPU40DR does not require battry. Note that the retentive range is no more than 1k words. When you setup the retentive memory in KPPSoft software, KPPSoft will prohibit writing the range parameter and report error if the retentive data exceeds 1k words.

The following table lists the initial value and the range of NK1L CPU's retentive memory.

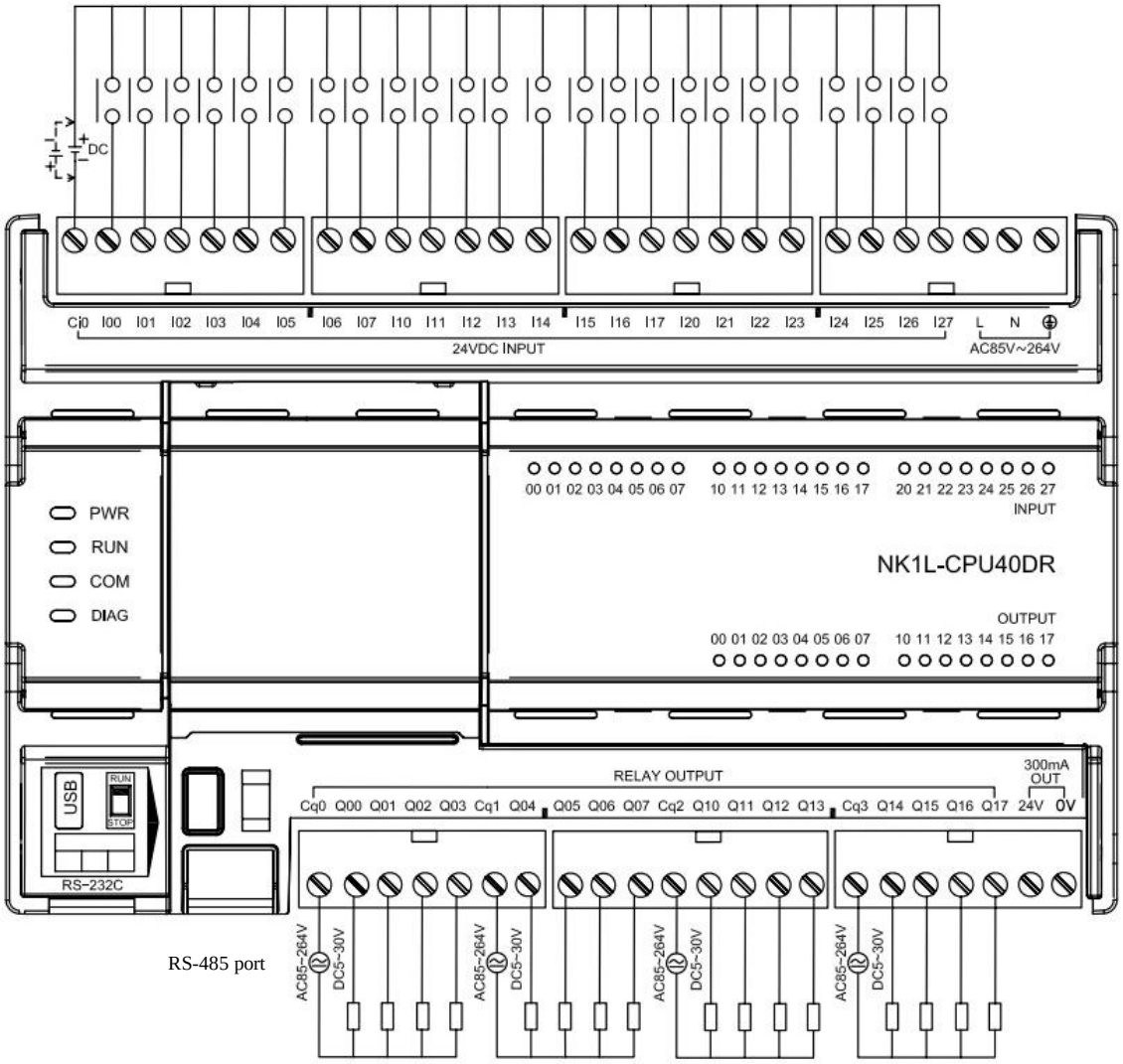
Item		Initial value	Set value
Retentive memory (*)	M	M300 – M377	M0000—M3777
	R	R2000 – R3677	R2000—R37777
	T	None	T000—T777
	C	C000 – C777	C000—C777
	S	None	S000—S1777

In addition to the specifications described above, the other specifications and usage of NK1L-CPU40DR are the same as NK1-CPU40DR's.

When using NK1L-CPU40DR, please refer to the information of NK1-CPU40DR described in other sections.

NOTE:Only the NK1L-CPU40DR with the production code 16ZB later is been installed the RS-485 prot!

The following figure shows the input/ output wiring diagram of NK1L-CPU40DR.



RS-485 port

Chapter 8 NK1H-CPU20 series introduction

NK1H-CPU20 sub-series PLCs is the new NK1 series PLCs which is besed on the NK1-CPU20 sub-series. NK1H-CPU20 sub-series PLCs includes following six types.

No.	Unit	specifications	remark
1	NK1H-CPU20DR	12 points DC24V input and 8 points relay output, power supply is AC85V~264V with Ether port	
2	NK1H-CPU20DD	12 points DC24V input and 8 points solid-state output, power supply is AC85V~264V with Ether port	
3	NK1H-CPU20DM	12 points DC24V input, 4 points solid-state output and 4 points relay output, power supply is AC85V~264V with Ether port	
4	NK1H-CPU20DR-D	12 points DC24V input and 8 points relay output, power supply is DC24V with Ether port	
5	NK1H-CPU20DD-D	12 points DC24V input and 8 points solid-state output, power supply is DC24V with Ether port	
6	NK1H-CPU20DM-D	12 points DC24V input, 4 points solid-state output and 4 points relay output, power supply is DC24V with Ethe port	

To compare with the NK1-CPU20 sub-series, NK1H-CPU20 sub-series PLCs changes one RS-232C port (P2) to a Ethernet port(P3).The specification of this Ethernet port is just as the same as it is in the NK1-CPU40 sub-series PLCs. For details of the Ethernet port specification, please see chapter4-7 of this manual.

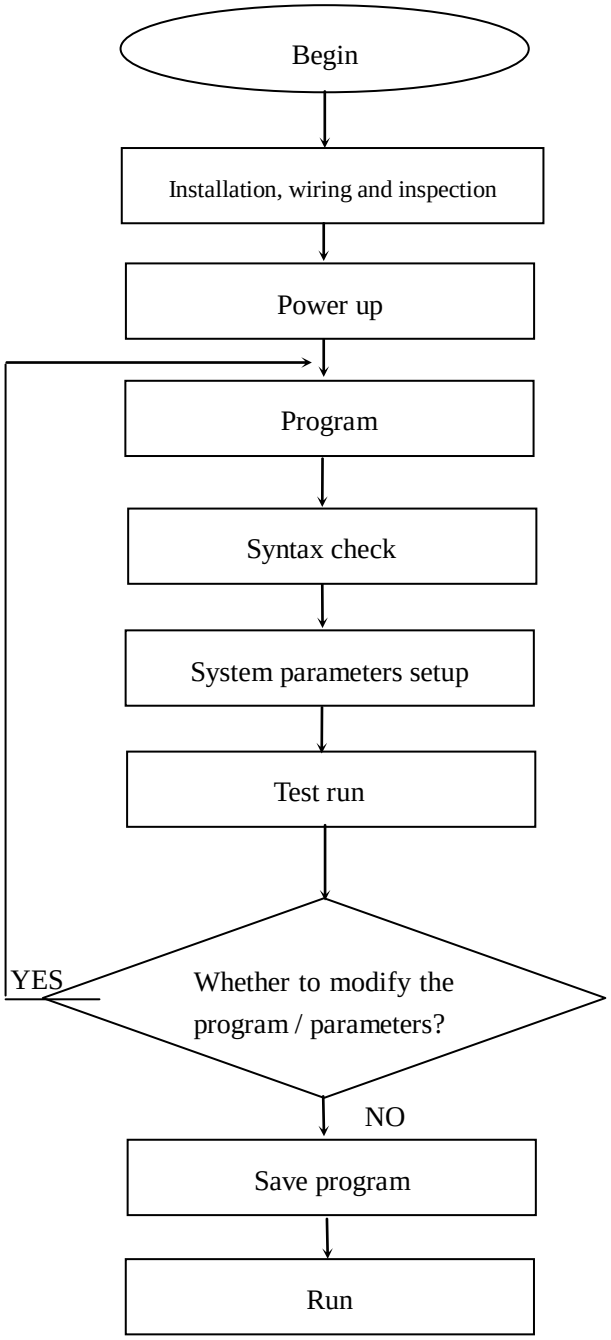
All other specification of NK1H-CPU20 sub-series PLCs is just the same as the specification of NK1-CPU20 sub-series PLCs. For detail please see the other chapters of this manual.

NOTE:

1. Only the system firmware(file name: NK14064.bin) version V20171212 or later supports NK1H-CPU20 sub-series PLCs.
2. For programming software, It needs the KPPSoft version V1.5.0.5 or later to support NK1H-CPU20 sub-series PLCs.

Chapter 9 Running preparation

9-1 Running steps

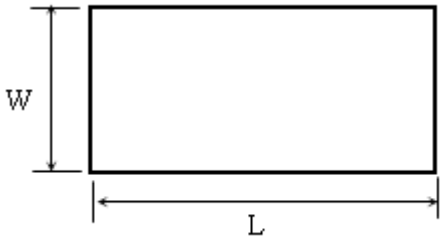


7－2
7－3
7－4
7－5
7－6
7－7
7－8
7－9

9-2 Installation wiring and inspection

9-2-1 Mounting dimensions and method

NK1 series PLC can be mounted by using screws or using mounting rails. Regardless of the mouting method, you first need to understand the length and width dimensions (L and W) of each NK1 modules.

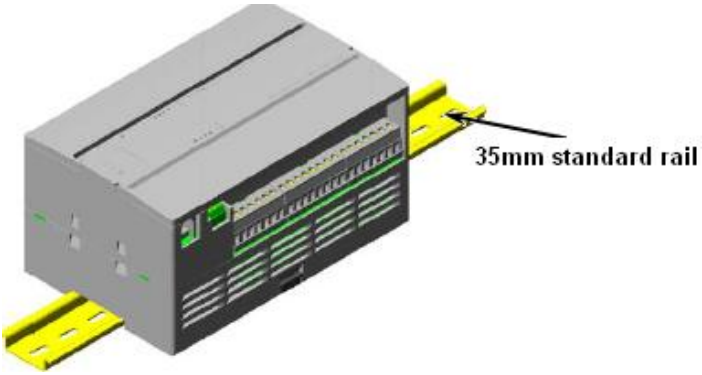


The following table lists the L and W of each NK1 modules (unit: ㎜).

NK1 modules	L	W
NK1-CPU20	110	102
NK1-CPU40	162	102
NK1-CPU64		
16-point expansion unit	50	102
32-point expansion unit	75	102

1. Use mounting rails

NK1 series CPU can also be mounted in the cabinet by using mounting rails. These rails are approximately 35mm high.



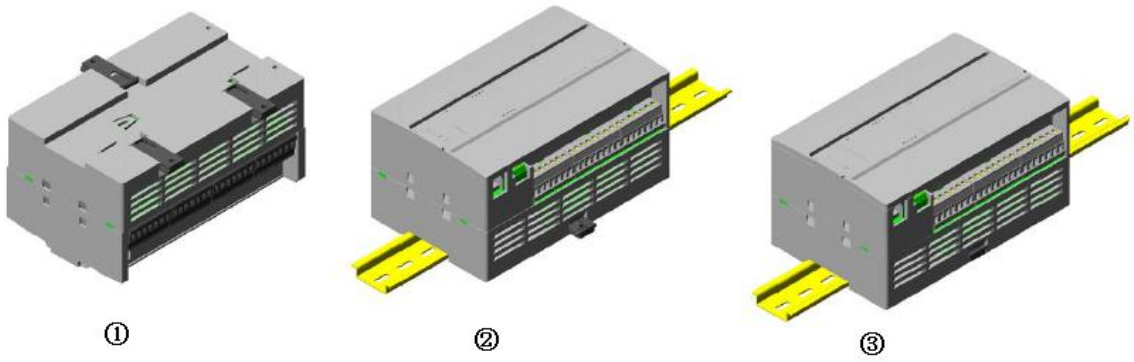
Rail mounting steps are:

① Pull open the three retaining clips on the back of CPU (expansion units have 2 retaining clips).

② Tilt NK1 CPU, first install the upper edge of the rail into the slot of NK1 CPU, then press NK1 CPU down hard, press the entire rail into the slot of NK1 CPU.

③ Withhold all the three retaining clips to complete mounting.

The following figure is the mounting diagram.



To remove the NK1 CPU, pull open the retaining clips, and pull it away from the rail.

When the system has expansion unit, its mounting steps are:

① Mount the CPU on the rail following the previous step ①, ② and ③;

② Mount the expansion unit on the rail following the previous setp ① and ②;

③ Remove the expansion socket cover in the right side of the CPU with a small screwdriver;

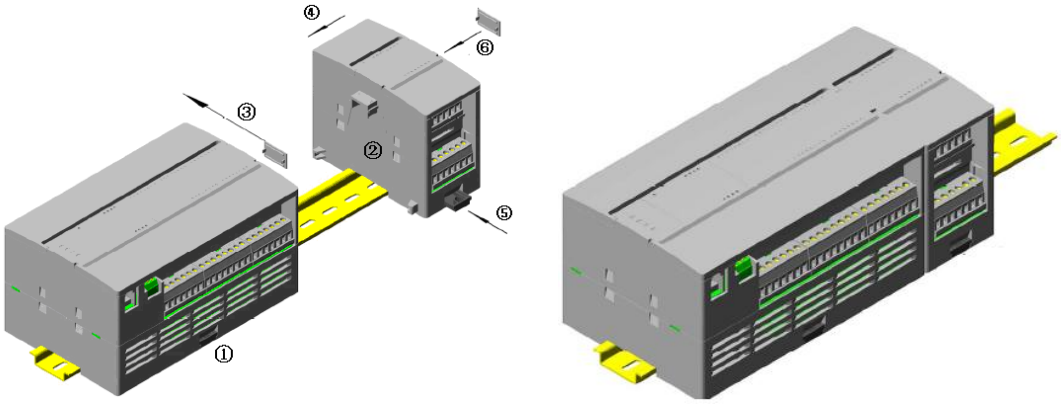
④ One hand holds the CPU, the other hand pushes the expansion unit to the CPU module (note that the expansion plug and fixed feet on the side of expansion unit should align with those on the side of CPU), until the two modules tightly connected together;

⑤ Withhold all the retaining clips of the expansion unit;

⑥ Install the expansion socket cover removed from the CPU into the expansion socket on the right side of expansion socket.

The following figure is the mounting diagram.

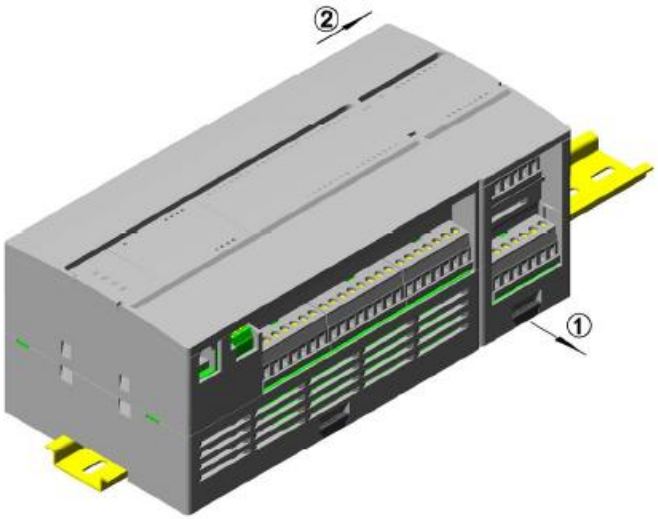
When install multiple expansion units, you can install them one by one according to the above method.



To remove the expansion unit: pull open the retaining clips, and pull it away from the rail.

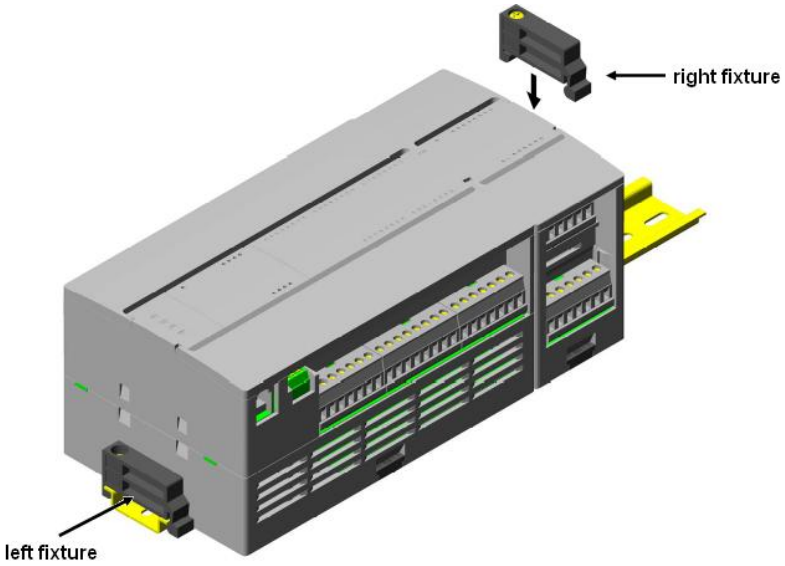
① Pull open the retaining clips of the expansion units;

② One hand holds the CPU, the other hand holds the expansion unit and pulls it to the right, the expansion unit is disengaged from the body.



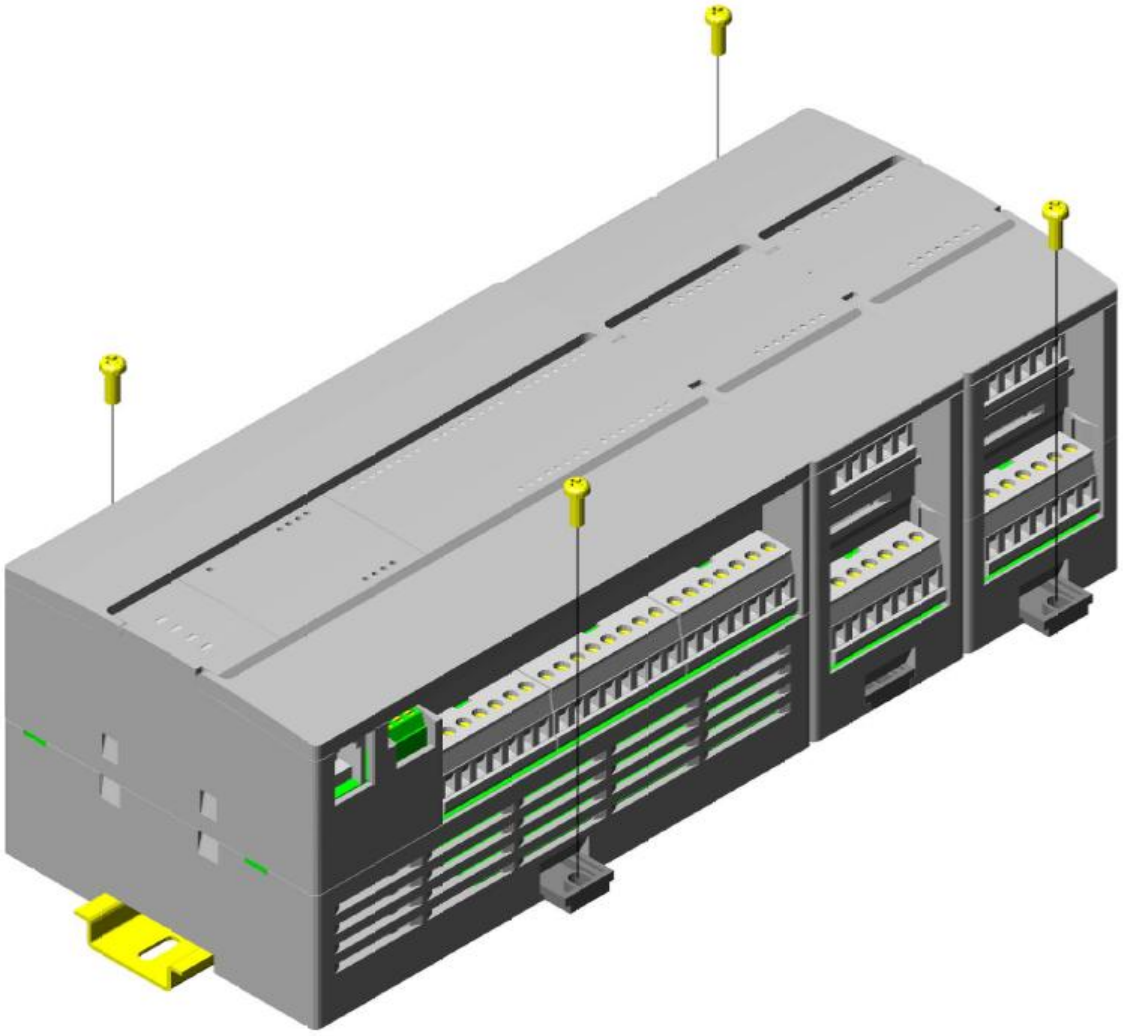
Rail mounting fixture

When using mounting rails, in order to better fixing the NK1 PLC, make sure to install a dedicated 35mm rail mounting fixture at both ends of the NK1 PLC, the following figure is the mounting diagram.



Our company sells this fixture, the model is NK1-Fixer.

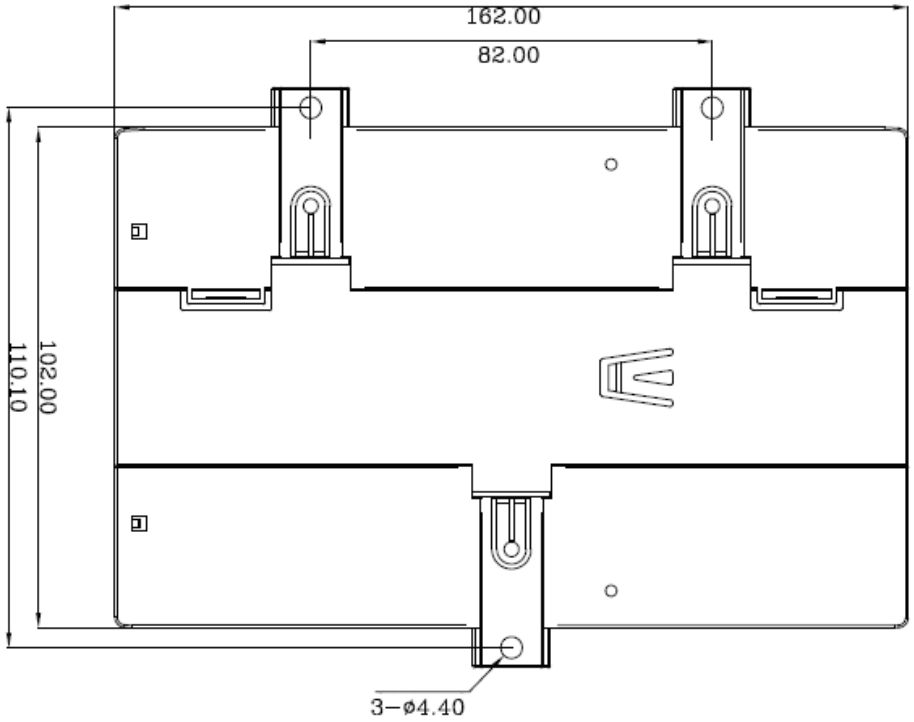
If you will not use dedicated fixtures, in order to better fixed NK1 PLC, we recommend that you fix the retaining clips on the CPU and the last expansion unit with screws.



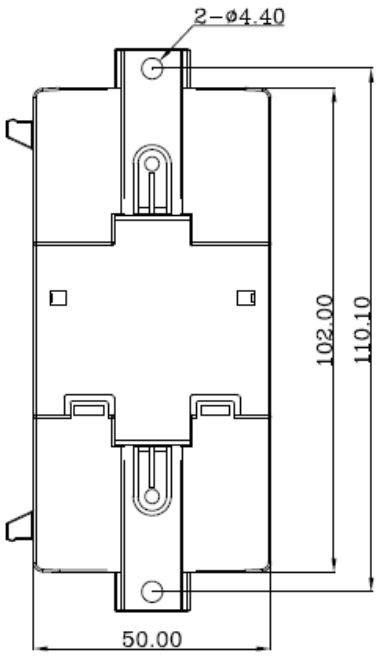
When the system has an extension cable, please fix the cable box with screws if you do not use the dedicated fixtures.

2. Use screws

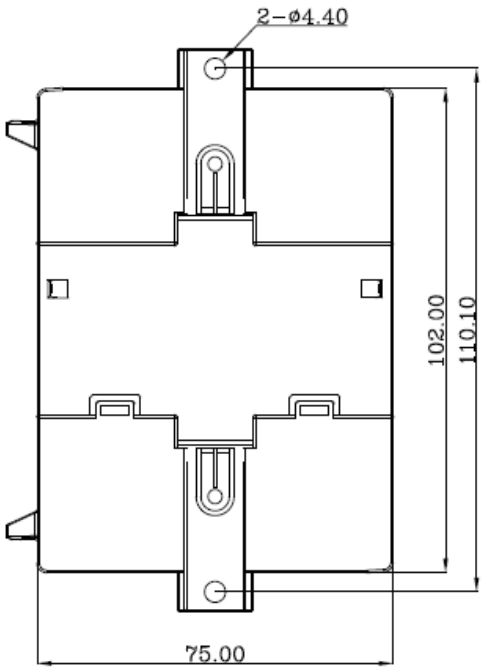
The retaining clips on NK1 series PLC have screw mounting holes, the hole diameter is ∅4.40mm, applicable M4 screws. The following figure is the screw mounting hole dimensions diagram.



40-point NK1 CPU (unit: mm)



16-point expansion unit (unit: mm)

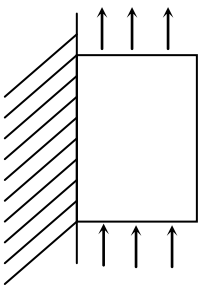


32-point expansion unit (unit: mm)

To mount the CPU and expansion units, please use the rail to hold them and then fix them with screws.

3. Panel Layout & Clearances

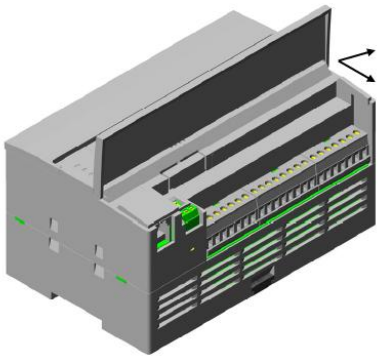
NK1 series PLC can be mounted vertically or horizontally. To ensure effective heat dissipation, please provide a minimum clearance of 50 mm between the units and all sides of the cabinet.



Heat dissipation diagram

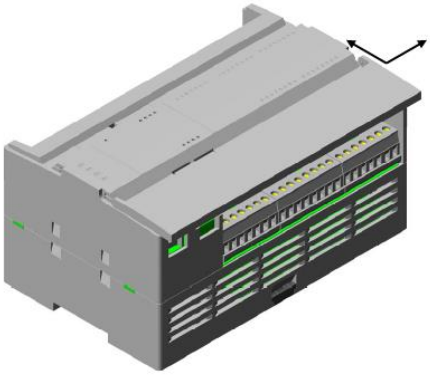
9-2-2 Install and remove the cover of terminal block

1. Remove the cover



① Lift the cover to the upright position (as shown on the right figure).
② One hand holds the side of the vertical cover, the other hand gently pull the lower end of the other side out, and then pull the other side out, you can remove the cover.
③ Remove the other cover with the same method.

2. Install the cover

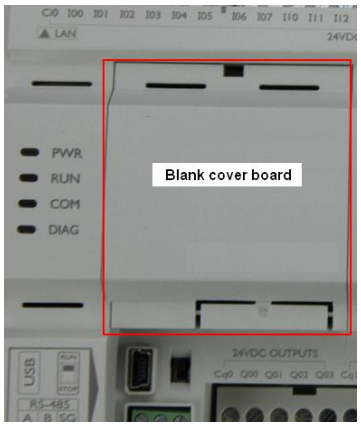


① Fist install the side of the cover (insert the small protrusion into the installation location);
② Pull out the other side of the cover, and also insert the other small protrusion into the other installation location.
③ Install the other cover with the same method.

9-2-3 Install and remove the option function board

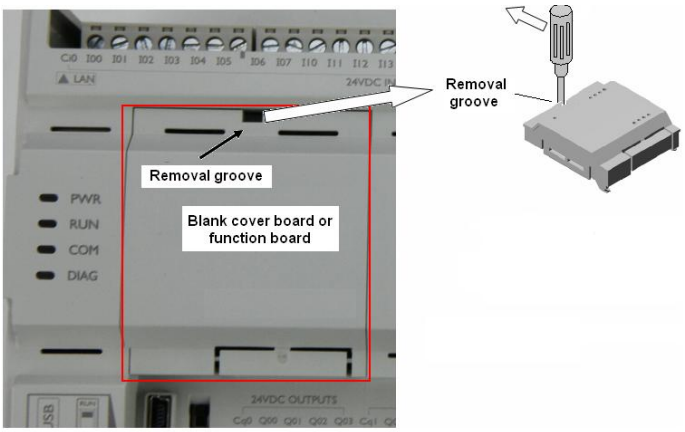
40-point/64-point NK1 CPUs can install a function board to extend the functionality of the PLC. These boards include battery boards, ordinary input/output boards, analog boards, communication boards, and so on.

The installation location of a function board is shown on the right. When installing function board, you must first remove the blank cover board on the CPU and then install the function board on the location of the blank cover board.



To remove the function board, you need first to remove the cover of terminal block, please refer to section 7-2-2 for details; the method of removing the blank cover board is:

1. Remove the blank cover board



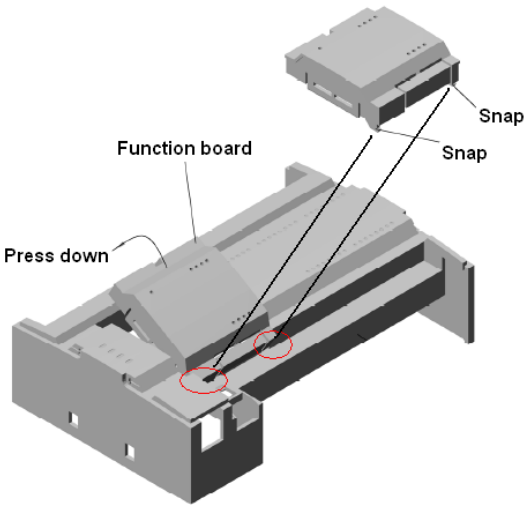
Insert a small flat-blade screwdriver into the "removal groove" in the upper end of blank cover board, gently pull the upper end of the screwdriver upwards and outwards, the blank cover board will float, then you can remove the function board.

The method of removing the function board and the blank cover board is the same.

2. Remove the function board

① Align the two snaps on function board to the snap installation location on the CPU, install the function board according to the right diagram.

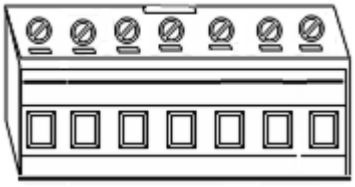
② Firmly press down the upper end of the function board, until it is firmly seated.



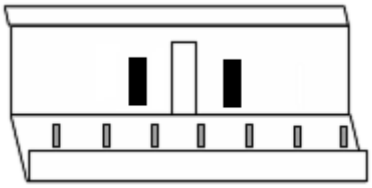
9-2-4 Install and remove terminal block

The terminal block is designed for easy removal with just a small screwdriver.

NK1 terminals are divided into different groups. Each group has its own terminal block.



Terminal block

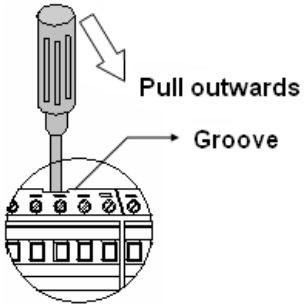


Terminal block slot

Note: the terminal block of RS-485 communications port on NK1 CPU and the terminal block on function board are not removable.

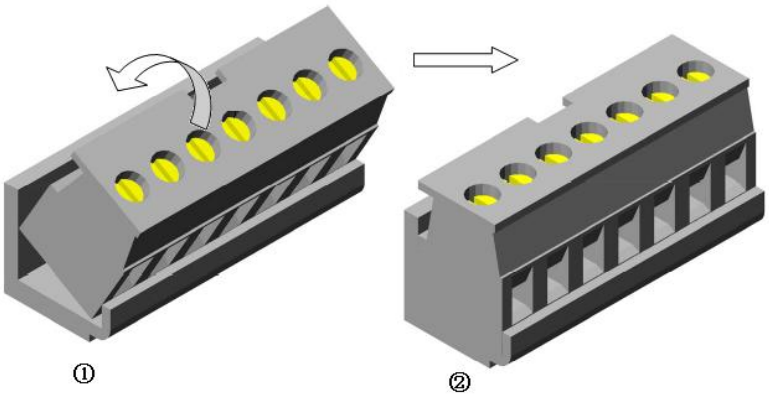
1. Remove the terminal block

There is a groove in the middle inside of each terminal block. Insert a small flat-blade screwdriver into the groove, gently pull the upper end of the screwdriver outwards, the terminal block will tilt, continue pull the screwdriver outwards, then you can remove the terminal block.



2. Install the terminal block

Align the little holes on the terminal block to the pins on the terminal block slot; align the little protrusion on the back of the terminal block to the little hole on the terminal block slot, press the terminal block downwards and inwards, you will hear a click, indicates that the terminal block is firmly seated.

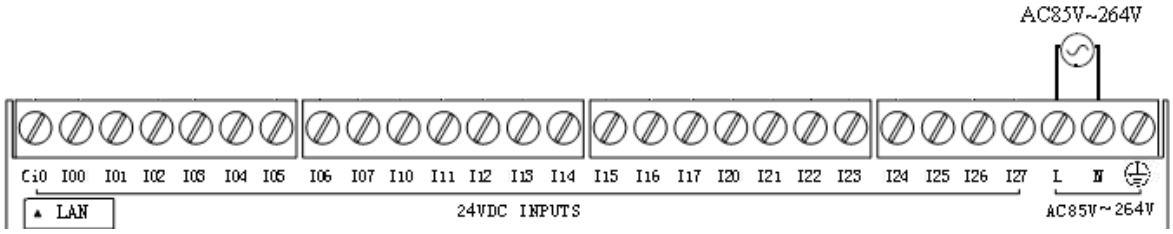


9-2-5 Wiring guidelines

1. Power input wiring

① 85～264VAC power input

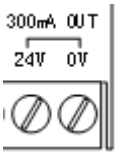
The following drawing shows the power input wiring of 40-point NK1 CPU.



② 24VDC power input

2. Auxiliary +24VDC supply

NK1 PLCs supply a limited amount of 24VDC power. The 24VDC output can be used to power external devices. In some cases, using the built-in auxiliary +24VDC supply can result in a cost savings for your control system. It can power combined loads up to 300mA.

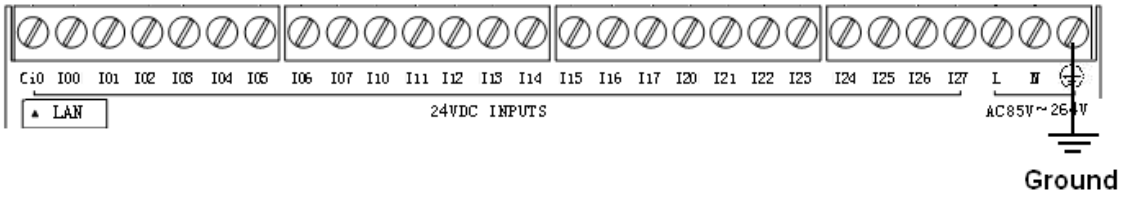


Note:

1) The 24VDC supply can not power the load that may become source of noise, such as a solenoid valve;
2) Do not exceed the maximum current capacity;
3) Be sure not to short-circuit the power.

3. Ground wiring

The terminal maked with ⏚ is the ground, the following diagram shows the method of wiring.

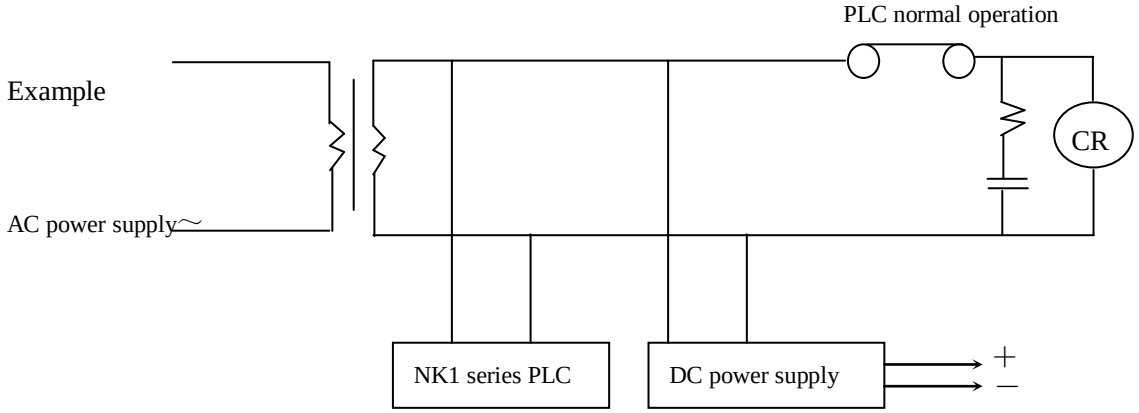


4. Caution for I/O wiring

1) Select the appropriate size wire.

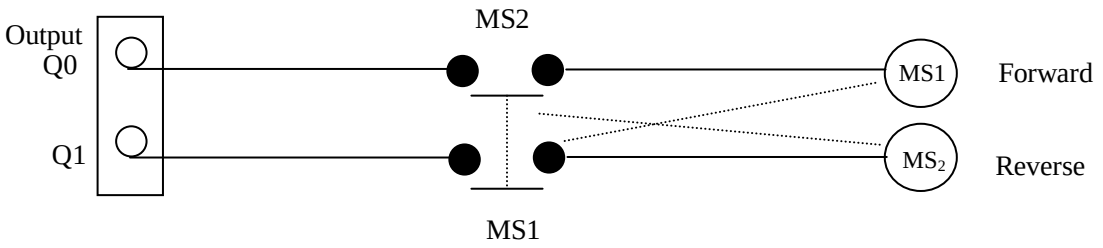
2) Avoid running wires near high energy wiring. Also, avoid running input wiring close to output wiring where possible. Avoid running DC wiring in close proximity to AC wiring where possible.

3) It is recommended that emergency stop circuits be incorporated into the system for every machine controlled by a PLC. For maximum safety in a PLC system, these circuits must not be wired into the controller, but should be hardwired external to the PLC.



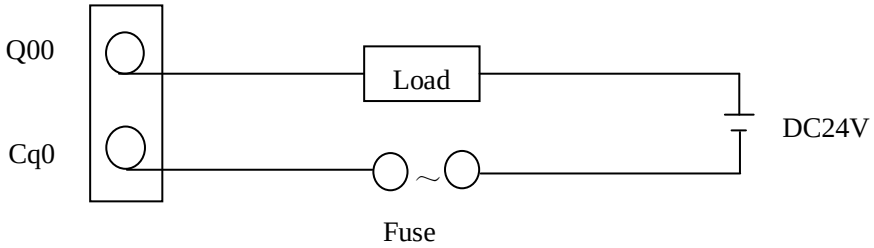
4） Interlock circuit

If use PLC to control the contrary action or avoid the machine from a serious accident or damage, please hardware an interlock circuit external to the PLC.



5） Fuse protection

To protect the external devices and the output points, we suggest you add external fuses to your output wiring.



9-2-6 Caution for PLC installation

(1) Check for damage and complying with order requirements before installing NK1 PLC.
(2) To ensure effective heat dissipation and maintenance, please provide a minimum clearance of 50 mm between the units and all sides of the cabinet.
(3) Please install the PLC on a flat surface.
(4) Please use the necessary wiring duct.
(5) If the PLC is installed under the panel, it is best to install a ventilation fan for heat dissipation.
(6) Avoid using this product in the following places:

- The place where ambient temperature is higher than 55℃ or lower than 0℃;
- The place where the relative humidity is not in the range of 30 to 95%, and the drastic temperature changes result in dew condensation;
- The place where there is dust, iron power, corrosive gas in the air;
- The place where there is excessive vibration and shock;
- The place where there are devices with strong magnetic and strong electrical interference.

(7) Avoid running input wiring close to output wiring where possible.

- Input, output and power wiring should be separated from each other;
- When using the high-speed counter, because the need of high-speed response, it is best to use shielded cable for high-speed counter input.

9-2-7 NK1 PLC installation check

(1) Check whether the power and I/O terminal connectors are firmly;
(2) Check whether the PLC is firmly;
(3) Check the power and I/O wiring;
(4) Battery confirmation, including whether the battery is installed, with or without a battery abnormality alarm.
(5) Check whether the debris into the PLC.

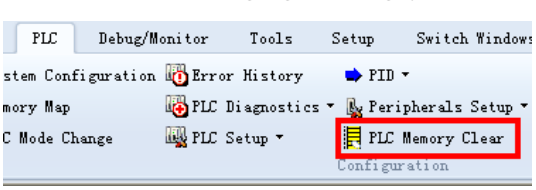
9-3 Power up the PLC

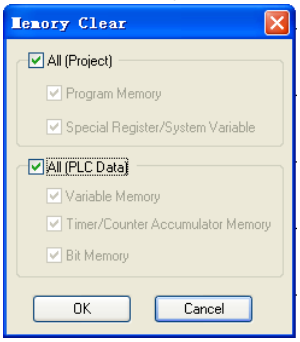
(1) Make sure that the power supply voltage is normal before power up;
(2) Make sure that the wiring of each connectors is correct;
(3) Power up the PLC;
Note: to ensure the system is in STOP mode, please toggle the mode swith to STOP position before power up the PLC.
(4) Check and ensure that PWR indicator (green) is on.
If the PWR indicator is not on, turn off the power immediately, find the cause and correct it.

9-4 Programming

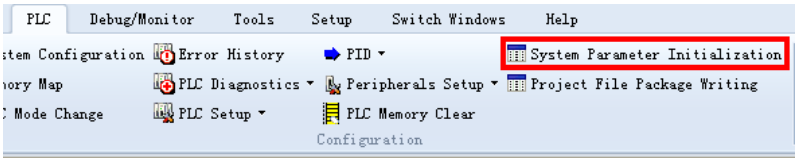
NK1 can be programmed by programming software KPPSoft. Please refer to KPPSoft software and NK1 system instructions manual for specific programming method.

Note: before the first grogramming, you'd better clear PLC memory.



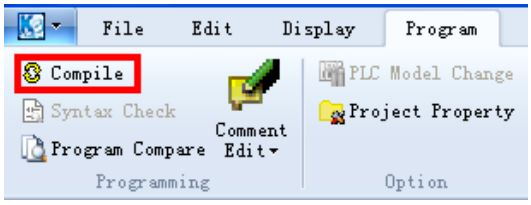


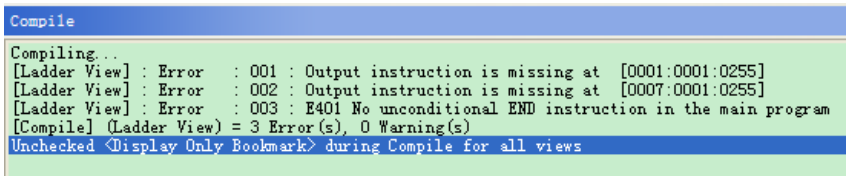
Then initialize the system parameter to ensure PLC is in standard factory state.



9-5 Program syntax checking

The finished program should be compiled for syntax checking, syntax checking can find the content against the rules. Syntax checking is done in KPPSoft software. In KPPSoft software, click "program --> compile" (the shortcut is F9). If there is syntax error in the user program, the error message will be displayed in the "compile information" window in the bottom of KPPSoft.





The major grammatical errors and its causes and solutions are:

(1) "Error"

▲E401 missing end

Write END instruction at the end of the main program.

▲E421 stage duplicate

Duplicate use of the same stage number for the SG and ISG instruction

▲E453 missing timer/counter

The T/C coil corresponding to the contact of timer or counter is not written.

▲E455 missing counter condition

Insufficient conditions of the CNT instruction. (Two conditions for the CNT and three conditions for the UDCNT are required.)

Write in the necessary conditions before the CNT or UDCNT.

▲E461 logic levels overflow

ANDLD or ORLD instruction is used continuously more than 9 times.

▲E462 an unmatched number of logic levels

The number of ANDLD or ORLD instruction exceeds the number required by the contact set.

▲E463 logic error

A LD instruction was not used to begin a rung of ladder logic.

▲E464 do not form a loop

No OUT or JMP instruction to end the loop.

(1) "Warining"

▲ E471 duplicate coil reference

Two or more OUT instructions reference the same I/O point.

▲ E472 duplicate timer reference

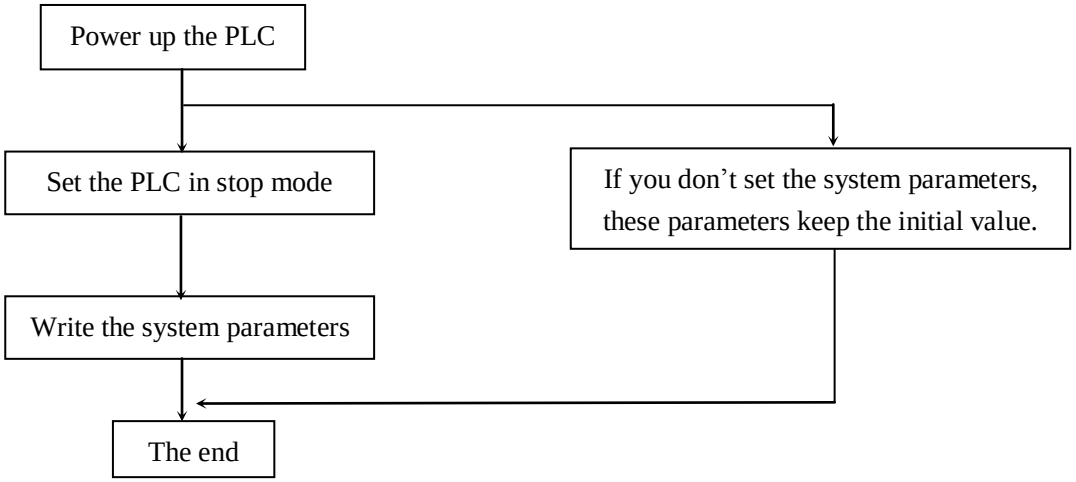
Two or more TMR instructions reference the same number.

▲ E473 duplicate counter reference

Two or more CNT instructions reference the same number.

9-6 System parameters setup

Before system operation, you need first to confirm whether the parameters listed in the following table and some special R-memory need to be set. Usually these parameters have an initial value, if the initial value is appropriate, you do not require to change it.



Note: if necessary, you can initialize the system parameter in KPPSoft before setup them.

The following table lists the initial value and the range of system parameter.

Item		Initial value	Set value
The name of user program		Blank	Eight letters, digitals
Password		00000000	Eight digitals (BCD)
Retentive memory (*)	M	M300 – M377	M0000—M3777
	R	R2000 – R7377	R2000—R37777
	T	None	T000—T777
	C	C000 – C777	C000—C777
	S	None	S000—S1777
W ĐOG time		200ms	2—9998ms

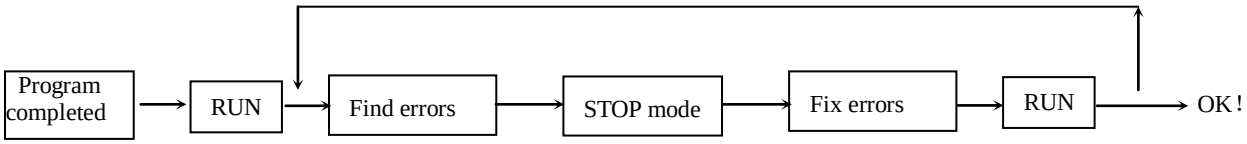
(*) Description:

The retentive memory of NK1 PLC does not require battry, but the retentive range supported by NK1 PLC is no more than 10k words. If the retentive memory exceeds 10k words, only the 10k words in front of rentive memory (the retentive memory is sorted by M, R, T, C and S) will be retentive.

9-7 Test run

Compile your program, make sure there is no syntax error in the program, you can download it to the NK1 PLC and can test run it.
Verify whether the program logic meets the control requirements by test run the program. If not, you should modify the program, until it meets the requirements.

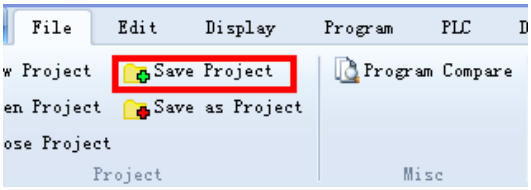
The basic steps of test run are:



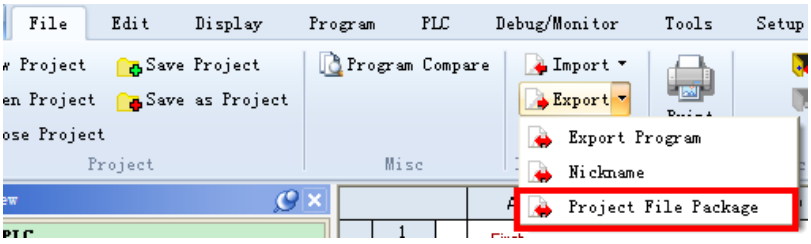
Note: Regardless of which position the mode switch is in, you can change the operating mode by NK1 KPPSoft!

9-8 Save program

Before the program formal running, if necessary, you should save the program for filing or backup.
You can save the program as a computer disk file (.KPP file) via "Save Project" menu in KPPSoft software.



You can also save the program as project file package dedicated for program transferring via "Project File Package" menu in KPPSoft software. The project file package can be transferred to NK1 PLC in KPPSoft software, after being renamed, it can also be transferred to NK1 PLC via USB port by using USB flash disk.



Please refer to <KPPSoft Software Getting Started Guide> for details.

9-9 Run

After the user program is debugged, it can officially operate on the NK1 PLC. Make sure the operation mode switch is in “RUN” position, if not, after power off and up again, NK1 PLC will automatically enter STOP mode.

If the mode switch is in “RUN” position and NK1 PLC is in STOP mode, there are two ways to run the program: 1) power NK1 off and on again, NK1 will automatically enter run mode; 2) toggle the mode switch to "STOP" position, then toggle it to "RUN" position, NK1 will enter run mode.

Chapter 10 System maintenance

No regular or preventative maintenance is required for this product. You only need to do a few periodic checks to help reduce the risks of problems.

When the PLC is running, you can moniter the state of the entire control system by KPPSoft programming software. Additionally indicators on PLC panel (each I/O LED, PWR LED, RUN LED, COM LED and DIAG LED) are helpful for monitoring PLC status and faults.

Failure occurs when the PLC is running, you can consider the following reasons:

⑴ PLC system power supply failure

○ No power supply;

○ Supply voltage is low;

○ Instantaneously power disconnect;

○ Large noise is mixed in power supply.

⑵ Machine damage due to accident, error and other causes.

○ Due to the superposition of high voltage (such as lightning, etc.);

○ Power units(such as valves, motors, etc) damage caused by mechanical failure;

○ Detection device damage caused by mechanical failure.

⑶ Incomplete control loop

○ Control circuit (PLC program, etc.) and mechanical dyssynchrony.

○ Control loop unexpected case.

⑷ Mechanical equipment aging and loss

○ Poor contact (limit switches, relays, solenoid valves, etc.);

○ Backup battery is abnormal;

○ Environmental deterioration of PLC caused by large noise;

⑸ Program changes due to large noise or misuse

○ Contrary to the operating requirements changes the program;

○ Replace the memory chips when power on;

○ Strong electrical noise changes the program.

10-1 Hardware maintenance

No regular or preventative maintenance is required for this product; however, a routine maintenance check (about every one or two months) of your PLC and control system is good practice, and should include the following items:

• Air temperature – Monitor the air temperature in the control cabinet, make sure that the operating temperature range of any component is not exceeded.

• Air filter – If the control cabinet has an air filter, clean or replace it periodically as required.

• Fuses or breakers – Make sure that all fuses and breakers are intact.

• Cleaning the Unit – Make sure that all air vents are clear. If the exterior case needs cleaning, disconnect the input power, and carefully wipe the case using a damp cloth. Do not let water enter the case through the air vents and do not use strong detergents because this may discolor the case.

10-2 CPU indicators

The indicators on the front panel of the NK1 PLCs can help you determine potential problems within the system. In normal RUN mode, the RUN and PWR indicators are on, the COM indicator flashes when NK1 is communicating with other devices, the DIAG indicator is off. The table below is a quick reference to potential problems.

Indicator status	Potential problems
PWR（OFF）	System voltage incorrect
	PLC power supply faulty
RUN（OFF）	CPU not in RUN mode
	CPU programming error
DIAG（ON）	An error occurred when power-up or when the system is running.
DIAG（flashing）	Flashing alone at 1 second intervals indicates: in battery mode, no battery is installed or battery voltage is low, the battery needs to be replced.
	COM and DIAG indicator flashing together at 1 second intervals. If RUN indicator is off, indicates: The firmware system self-test error. If RUN indicator is on, indicates: The project file package has been downloaded completely via USB port.
	RUN, COM and DIAG indicator are flashing together at 100ms intervals, indicates: PF signal is detected. (The cause is often that the USB port is connected with a USB cable, please unplug the USB cable.)

10-3 Communication problems

If you can not establish communications with the CPU, check these items:

• The cable is disconnected.

• The cable has a broken wire or has been wired incorrectly.

• The cable is improperly terminated or grounded.

• The connected device is not operating at the correct baud rate (9600 baud).

• The device connected to the port is sending data incorrectly, or another application is running on the device.

• A grounding difference exists between the two devices.

• Electrical noise is causing intermittent errors.

• The PLC has a bad communication port and should be replaced.

Refer to the KPPSoft programming user manual for details of the problems in communicating with KPPSoft on a personal computer. It includes a troubleshooting section that can help you diagnose PC problems in communications port setup, address or interrupt conflicts, etc.

10-4 I/O point troubleshooting

If you suspect an I/O error, there are several things that could be causing the problem.
• High-speed I/O configuration error
• A blown fuse in your machine or panel (the NK1 does not have internal I/O fuses)
• A loose terminal block
• The auxiliary 24 VDC supply has failed
• The input or output circuit has failed

When troubleshooting the NK1 PLCs, please be aware of the following facts which may assist you in quickly correcting an I/O problem.
• HSIO configuration errors are commonly mistaken for I/O point failure during program development. If the I/O point in question is in I0–I17, or Q0–Q3, check all parameter locations that apply to the HSIO mode you have selected.
• The output circuits cannot detect shorted or open output points. If you suspect one or more faulty points, measure the voltage drop from the common to the suspect point. Remember when using a Digital Volt Meter, leakage current from an output device such as a triac or a transistor must be considered. A point which is off may appear to be on if no load is connected the point.
• The I/O point status indicators are logic-side indicators. This means the LED which indicates the on or off status reflects the status of the point with respect to the CPU. On an output point the status indicators could be operating normally while the actual output device (transistor, triac etc.) could be damaged. With an input point, if the indicator LED is on, the input circuitry is probably operating properly. Verify the LED goes off when the input signal is removed.
• Leakage current can be a problem when connecting field devices to an I/O point. False input signals can be generated when the leakage current of an output device is great enough to turn on the connected input device. To correct this install a resistor in parallel with the input or output of the circuit. The value of this resistor will depend on the amount of leakage current and the voltage applied but usually a 10K to 20K resistor will work. Verify the wattage rating of the resistor is correct for your application.
• Because of the removable terminal blocks on the NK1, the easiest method to determine if an I/O circuit has failed is to replace the unit if you have a spare. However, if you suspect a field device is defective, that device may cause the same failure in the replacement PLC as well. As a point of caution, you may want to check devices or power supplies connected to the failed I/O circuit before replacing the unit with a spare.

10-5 Noise troubleshooting

Noise is one of the most difficult problems to diagnose. Electrical noise can enter a system in many different ways and falls into one of two categories, conducted or radiated. It may be difficult to determine how the noise is entering the system but the corrective actions for either of the types of noise problems are similar.
• Conducted noise is when the electrical interference is introduced into the system by way of an attached wire, panel connection, etc. It may enter through an I/O circuit, a power supply connection, the communication ground connection, or the chassis ground connection.
• Radiated noise is when the electrical interference is introduced into the system without a direct

electrical connection, much in the same manner as radio waves.

While electrical noise cannot be eliminated it can be reduced to a level that will not affect the system.
• Most noise problems result from improper grounding of the system. A good earth ground can be the single most effective way to correct noise problems. Ensure all ground wires are single point grounds and are not daisy chained from one device to another.
• Electrical noise can enter the system through the power source for the PLC and I/O circuits. Installing an isolation transformer for all AC sources can correct this problem. DC sources should be well-grounded good quality supplies.
• Separate input wiring from output wiring. Never run low-voltage I/O wiring close to high voltage wiring.

10-6 Install and replace the lithium battery

An optional lithium battery is available to use calendar and clock function when the NK1 system is without external power. Typically, CPU battery life is 2~3 years. However, poor working conditions (such as high temperature, high humidity) may shorten the battery life. To ensure NK1 clock is correct, you should replace battery quickly when the battery voltage is low. Lithium battery model is RB-50.

Install/replace battery in the NK1 CPU:

(1) Before replacing the battery, the PLC needs to be energized more than one hour (power is not required when installing the battery for the first time);

(2) After a power failure, first remove the two terminal cover (To replace function board, you must first remove the terminal cover);

(3) Remove the function board or blank cover board mounting on the CPU;

(4) Remove the old battery;

(5) Install the new battery in the function board, be careful not to mistake the positive and negative battery installation;

(6) Install the function board to the CPU;

(7) Cover the NK1 terminal cover;

(8) CPU power on and make sure the battery abnormality indicator goes out. If this is the first installation of a lithium battery, after power up, NK1 first needs to be set to battery mode (BIT12 of R7633= ON) and then verify CPU battery abnormality indicator (DIAG) goes out.

Note:

(1) When installing the battery, be careful not to mistake the battery positive and negative;

(2) Complete battery replacement within 10 minutes after a power failure;

(3) Do not litter the old batteries; do not litter the old batteries into the fire;

(4) Do not short-circuit, charge and disassemble the old battery.

10-7 NK1 PLC error code table

Errors in NK1 PLC mainly include two categories: errors occur during PLC operating; errors occur during writing program.

1. Errors occur during PLC operating

Error code	Description
E003	If the program scan time exceeds the time allotted to the watchdog timer, this error will occur. SP51 will be on.
E041	Battery voltage is low. SP43 will be ON.
E155	A checksum error has occurred in the system memory.
E250	A failure has occurred in the local I/O system. The problem could be in the I/O bus or equipments connecting with I/O bus.
E252	This error occurs when the auto configuration check is turned on in the CPU and the actual I/O configuration has changed either by moving modules in a base or changing types of modules in a base.
E312	A data error was encountered during communications (connects with KPPSoft via Ethernet).
E313	An address error was encountered during communication.
E316	A mode error was encountered during communication.
E321	An error was encountered during SMTP communication.
E401	No END instruction.

2. Errors occur during writing program

Error code	Description
E4××	No user program
E401	Missing end statement
E402	Missing GLBL, CLBL. A GOTO, CALL instruction was used without the appropriate label.
E403	Missing CEND. A subroutine in the program does not end with the CEND instruction.
E404	Missing FOR. A NEXT instruction does not have the corresponding FOR instruction.
E405	Missing NEXT. A FOR instruction does not have the corresponding NEXT instruction.
E406	MISSING CEND. An interrupt routine in the program does not end with the CEND instruction.
E411	There is greater than permitted number of stages in the program.
E412	There is greater than permitted number of GLBL/CLBL/DLBL instructions in the program.
E413	There is greater than 64 FOR/NEXT loops in the program.
E421	Duplicate stage reference. Two or more SG or ISG labels exist in the application program with the same number.
E422	Duplicate CLBL/GLBL reference. Two or more CLBL or GLBL instructions exist in the application program with the same number.

Continued

E423	Nested loops. Nested loops (programming one FOR/NEXT loop inside of another) are not allowed in NK1 PLCs.
E431	Invalid ISG/SG address. An ISG or SG must not be programmed after the end statement such as in a subroutine.
E432	Invalid JUMP (GOTO) address. A GLBL that corresponds to a GOTO instruction must not be programmed after the end statement such as in a subroutine.
E433	Invalid CLBL address. A CLBL must be programmed after the end statement, not in the main body of the program or in an interrupt routine.
E434	Invalid RET address.
E435	Invalid CEND address. A CEND must be programmed after the end statement, not in the main body of the program or in an interrupt routine.
E436	Invalid ILBL address. An ILBL must be programmed after the end statement, not in the main body of the program.
E437	Invalid RETI address. A RETI must be programmed in the interrupt routine instead of main program or subroutines.
E438	Invalid IEND address. An IEND must be programmed in the interrupt routine instead of main program or subroutines.
E440	The DLBL instruction is on a rung containing input contact(s).
E441	There are instructions except for ACON or NCON in data definition blocks.
E451	Bad MLS/MLR. MLS instructions must be numbered in ascending order from top to bottom.
E452	I as coil. An I data type is being used as a coil output.
E453	Missing T/C. A timer or counter contact is being used where the associated timer or counter does not exist.
E454	One of the contacts is missing from a ATMR and AHTMR instruction.
E455	One of the contacts is missing from a CNT or UDCNT instruction.
E456	One of the contacts is missing from the SR instruction.
E461	More than nine levels of logic have been stored on the stack.
E462	An unmatched number of logic levels have been stored on the stack.
E463	A LD instruction was not used to begin a rung of ladder logic.
E464	A rung of ladder logic is not terminated properly.
E471	Two or more OUT instructions reference the same I/O point.
E472	Two or more TMR instructions reference the same number.
E473	Two or more CNT instructions reference the same number.
E480	The CV instruction is used in a subroutine.
E481	An instruction exists between convergence stages.
E482	Number of CV instructions exceeds 17.
E483	CVJMP has been used in a subroutine.
E484	CVJMP is not preceded by the CV instruction.
E485	A CVJMP instruction is not placed between the CV and the SG, ISG, BSTART, BEND, END instruction.

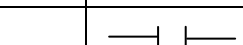
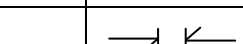
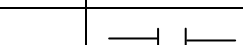
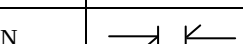
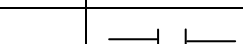
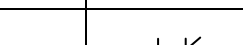
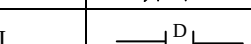
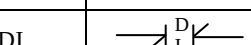
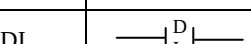
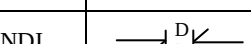
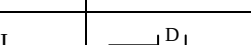
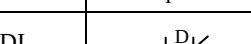
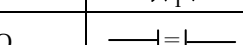
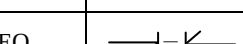
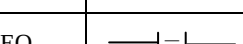
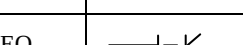
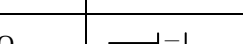
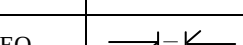
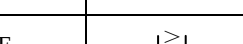
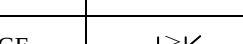
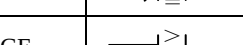
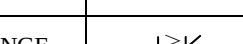
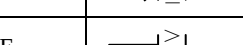
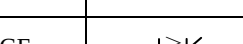
Continued

E486	A BREQ instruction is used in a subroutine.
E487	The BREQ instruction is not followed by a BSTART instruction.
E488	Another BSTART instruction is used between the BSTART and the BEND instructions.
E489	The BSTART instruction used the same number.
E490	The BSTART instruction is not immediately followed by the SG instruction.
E491	There is an ISG instruction between the BSTART and BEND instruction.
E492	The BEND instruction is not followed by a BSTART instruction.
E493	The BEND instruction is not immediately followed by CV, SG, ISG, BSTART, END instruction.
E494	There is not a BEND instruction between BSTART and END instruction.

Appendix 1 NK1 RLL insturctions

NK1 PLC has a total 265 instrucitons, which can be divided into four categories according to the function.

1. Sequential instructions (63)

Classification	Instruction name	Instruction	Symbol	Words number	Legal data types	Remark
Contact instruction	Store	LD		1	GI,GQ,I,Q,M,T,C,S,SP	
	Store not	LDN		1	GI,GQ,I,Q,M,T,C,S,SP	
	And	AND		1	GI,GQ,I,Q,M,T,C,S,SP	
	And not	ANDN		1	GI,GQ,I,Q,M,T,C,S,SP	
	Or	OR		1	GI,GQ,I,Q,M,T,C,S,SP	
	Or not	ORN		1	GI,GQ,I,Q,M,T,C,S,SP	
Immediate input contact instruction	Store immediate	LDDI		1	I	
	Store not immediate	LDNDI		1	I	
	And immediate	ANDDI		1	I	
	And not immediate	ANDNDI		1	I	
	Or immediate	ORDI		1	I	
	Or not immediate	ORNDI		1	I	
Comparative equal instruction	Store if equal	LDEQ		2	R,P [K,R,P]	*1
	Store if not equal	LDNEQ		2	R,P [K,R,P]	*1
	And if equal	ANDEQ		2	R,P [K,R,P]	*1
	And if not equal	ANDEQ		2	R,P [K,R,P]	*1
	Or if equal	OREQ		2	R,P [K,R,P]	*1
	Or if not equal	ORNEQ		2	R,P [K,R,P]	*1
Comparative greater and equal instruction	Store if greater or equal	LDGE		2	R,P [K,R,P]	*1
	Store if not greater or equal	LDNGE		2	R,P [K,R,P]	*1
	And if greater or equal	ANDGE		2	R,P [K,R,P]	*1
	And if not greater or equal	ANDNGE		2	R,P [K,R,P]	*1
	Or if greater or equal	ORGE		2	R,P [K,R,P]	*1
	Or if not greater or equal	ORNGE		2	R,P [K,R,P]	*1

*1 [ ]: the function memory that can be used by the second operand data.

Continued

Classification	Instruction name	Instruction	Symbol	Words number	Legal data types	Remark
Differential contact instruction	Store positive differential	LDPD	—| P |—	1	GI,GQ,I,Q,M,T,C,S	
	Store negative differential	LDND	—| N |—	1	GI,GQ,I,Q,M,T,C,S	
	And positive differential	ANDPD	—| P |—	1	GI,GQ,I,Q,M,T,C,S	
	And negative differential	ANDND	—| N |—	1	GI,GQ,I,Q,M,T,C,S	
	Or positive differential	ORPD	—| P |—	1	GI,GQ,I,Q,M,T,C,S	
	Or negative differential	ORND	—| N |—	1	GI,GQ,I,Q,M,T,C,S	
Register manipulation instruction	Store bit-of-word	BLD	—| |—	2	R,P [K](BCD0 ~ 15)	*1
	Store not bit-of-word	BLDN	—| |—	2	R,P [K] (BCD0 ~ 15)	*1
	And bit-of-word	BAND	—| |—	2	R,P [K] (BCD0 ~ 15)	*1
	And not bit-of-word	BANDN	—| |—	2	R,P [K] (BCD0 ~ 15)	*1
	Or bit-of-word	BOR	—| |—	2	R,P [K] (BCD0 ~ 15)	*1
	Or not bit-of-word	BORN	—| |—	2	R,P [K] (BCD0 ~ 15)	*1
	Out bit-of-word	BOUT	—{BOUT }—	2	R,P [K] (BCD0 ~ 15)	*1
	Set bit-of-word	BSET	—{BSET }—	2	R,P [K] (BCD0 ~ 15)	*1
	Reset bit-of-word	BRST	—{BRST }—	2	R,P [K] (BCD0 ~ 15)	*1
Logical group connection	And store	ANDLD		1		
	Or store	ORLD		1		
Output instruction	Or out	OUT	—( )—	1	GI,GQ,I,Q,M	
	Out	ZOUT	—{ZOUT }—	1	GI,GQ,I,Q,M	
	Out	ZOUTH	—{OUTH }—	1	I,Q,M	
	Not	NOT	—{NOT }—	1		
	Set	SET	—{SET }—	1（2）	GI,GQ,I,Q,M,S	
	Reset	RST	—{RST }—	1（2）	GI,GQ,I,Q,M,S	
Immediate output instruction	Or out immediate	OUTDI	—{OUTDI }—	1	Q	
	Out immediate	ZDI	—{ZDI }—	1	Q	
	Set immediate	SETDI	—{SETDI }—	1（2）	Q	
	Reset immediate	RSTDI	—{RSTDI }—	1（2）	Q	

*1 [ ]: the function memory that can be used by the second operand data.

Continued

Classification	Instruction name	Instruction	Symbol	Words number	Legal data types	Remark
Master line instruction	Master line set	MLS	—{MLS }—	1	K	
	Master line reset	MLR	—{MLR }—	1	K	
Differential outpout instruction	Positive differential	PD	—{PD }—	1	GI,GQ,I,Q,M	
Shift register instruction	Shift register	SR	—{SR }— —{CLOCK }— —{RESET }—	2	M ,[M]	*1
Timer instruction	Timer	TMR	—{TMR }—	2（3）	T [K，R，P]	*1
	Fast timer	HTMR	—{HTMR }—	2（3）	T [K，R，P]	*1
	Accumulating timer	ATMR	—{ATMR }— —{RESET }—	2（3）	T [K，R，P]	*1
	Fast accumulating timer	AHTMR	—{AHTMR }— —{RESET }—	2（3）	T [K，R，P]	*1
Counter instruction	Counter	CNT	—{CNT }— —{RESET }—	2（3）	C [K，R，P]	*1
	Stage counter	GCNT	—{GCNT }—	2（3）	C [K，R，P]	*1
	Up down counter	UDCNT	—{UDCNT }— —{DOWN }— —{RESET }—	2（3）	C [K，R，P]	*1
	Reset timer/counter	RSTTC	—{RSTTC }—	1（2）	T ,C [T,C]	*1

*1 [ ]: the function memory that can be used by the second operand data.

2. Program execution control instructions (26)

Classification	Instruction name	Instruction	Symbol	Words number	Legal data types	Remark
Stage instruction	Stage	SG	SG	2	S	
	Initial stage	ISG	ISG	2	S	
	Jump	JMP	JMP	1	S	
	Not jump	NJMP	NJMP	1	S	
	Convergence stage	CV	CV	1	S	
	Convergence jump	CVJMP	CVJMP	1	S	
	Block call	BREQ	BREQ	1	M	
	Block start	BSTART	BSTART	2	M	
	Block end	BEND	BEND	1		
Jump instruction	Goto label	GOTO	GOTO	2	K(1-FFFF)	
	Jump target mark	GLBL	GLBL	2	K(1-FFFF)	
Cycle instruction	Cycle starting point	FOR	FOR	1	K,R(1-9999)	
	Cycle return point	NEXT	NEXT	1		
Subroutine instruction	Goto subroutine	CAL	CAL	2	K(1-FFFF)	
	Subroutine	CLBL	CLBL	2	K(1-FFFF)	
	Subroutine return conditional	RET	RET	1		
	Subroutine return	CEND	CEND	1		
Interrupt instruction	Enable interrupts	INE	INE	1		
	Disable interrupts	INH	INH	1		
	Interrupt	ILBL	ILBL	1	O	
	Interrupt return conditional	RETI	RETI	1		
	Interrupt return	IEND	IEND	1		
other	Reset Watch Dog Timer	WDOGR	WDOGR	1		
	Stop	STOP	STOP	1		
	No operation	NOP		1		
	End of main program	END	END	1		

3. Data processing instructions (161)

Classification	Instruction name	Instruction	Symbol	Words number	Legal data types	Remark
Load instruction	Load	LDW	—〔LDW 〕⊢	1	R,P	
	Load double	LDD	—〔LDD 〕⊢	1	R,P	
	Load Formatted	LDF	—〔LDF 〕⊢	2	GI,GQ,I,Q,M,S,T,C,SP [K(1-32)]	*1
	Load Immediate	LDDW	—〔LDDW 〕⊢	1	R(I: input)	
	Load Immediate Formatted	LDDF	—〔LDDF 〕⊢	2	I, [K] (bit number)	*1
	Load Indexed	LDIX	—〔LDIX 〕⊢	2	R,P	
	pop	POP	—〔POP 〕⊢	1		
	Load 4 constant (Decimal, hexadecimal)	LDS	—〔LDS 〕⊢	1	K	
	Load 8 constant (Decimal, hexadecimal)	LDC	—〔LDC 〕⊢	2	K	
	Load real number(32 bits)	RLDD	—〔RLDD 〕⊢	3	R,P	
	Load real constant	RLDC	—〔RLDC 〕⊢	3	F	
	Load address	LDR	—〔LDR 〕⊢	1	O	
Out instruction	Out	OUTW	—〔OUTW 〕⊢	1	R,P	
	Out double	OUTD	—〔OUTD 〕⊢	1	R,P	
	Out formatted	OUTF	—〔OUTF 〕⊢	2	GI, GQ, I, Q, M [K] (bit number)	*1
	Out immediate	OUTDW	—〔OUTDW 〕⊢	1	R(Q：output)	
	Out immediate formatted	OUTDF	—〔OUTDF 〕⊢	2	Q, [K] (bit number)	*1
	Out most	OUTM	—〔OUTM 〕⊢	2	R	
	Out least	OUTL	—〔OUTL 〕⊢	2	R	
	Out indexed	OUTIX	—〔OUTIX 〕⊢	1	R,P	

*1 [ ]: the function memory that can be used by the second operand data.

Continued

Classification	Instruction name	Instruction	Symbol	Words number	Legal data types	Remark
BCD data add instruction	Add BCD	ADD	—〔ADD 〕—	1	R,P	*1
	Add double BCD	ADDD	—〔ADDD 〕—	1	R,P	*1
	Add formatted	ADDF	—〔ADDF 〕—	2	GI,GQ,I,Q,M,S,T,C,SP [K(1-32)]	*1
	Add top of stack	SADD	—〔SADD 〕—	1		
	Add 8 constant	ADDC	—〔ADDC 〕—	2	K	
BCD data sbutract instruction	Subtract BCD	SUB	—〔SUB 〕—	1	R,P	*1
	Subtract double BCD	SUBD	—〔SUBD 〕—	1	R,P	*1
	Subtract formatted	SUBF	—〔SUBF 〕—	2	GI,GQ,I,Q,M,S,T,C,SP [K(1-32)]	*1
	Subtract top of stack	SSUB	—〔SSUB 〕—	1		
	Subtract 8 constant	SUBC	—〔SUBC 〕—	2	K	
BCD data multiply instruction	Multiply BCD	MUL	—〔MUL 〕—	1	R,P	*1
	Multiply double BCD	MULD	—〔MULD 〕—	2	R,P	
	Multiply formatted	MULF	—〔MULF 〕—	2	GI,GQ,I,Q,M,S,T,C,SP [K(1-32)]	
	Multiply top of stack	SMUL	—〔SMUL 〕—	1		
	Multiply 4 constant	MULS	—〔MULS 〕—	1	K	
BCD data divide instruction	Divide BCD	DIV	—〔DIV 〕—	1	R,P	*1
	Divide double BCD	DIVD	—〔DIVD 〕—	2	R,P	
	Divide formatted (1~16bit)	DIVF	—〔DIVF 〕—	2	GI,GQ,I,Q,M,S,T,C,SP [K(1-32)]	
	Divide top of stack	SDIV	—〔SDIV 〕—	1		
	Divide 4 constant	DIVS	—〔DIVS 〕—	1	K	
BIN data add instruction	Add binary	BADD	—〔BADD 〕—	1	R,P	
	Add binary double	BADDD	—〔BADDD 〕—	1	R,P	
	Add binary top of stack	SBADD	—〔SBADD 〕—	1		
	Add 4 constant	BADDS	—〔BADDS 〕—	1	K	
	Add 8 constant	BADDC	—〔BADDC 〕—	1	K	

*1 [ ]: the function memory that can be used by the second operand data.

Continued

Classification	Instruction name	Instruction	Symbol	Words number	Legal data types	Remark
BIN data subtract instruction	Subtract binary	BSUB	—{BSUB }—	1	R,P	
	Subtract binary double	BSUBD	—{BSUBD }—	1	R,P	
	Subtract binary top of stack	SBSUB	—{SBSUB }—	1		
	Subtract 4 constant	BSUBS	—{BSUBS }—	1	K	
	Subtract 8 constant	BSUBC	—{BSUBC }—	1	K	
BIN data multiply instruction	Multiply binary	BMUL	—{BMUL }—	1	R,P	
	Multiply binary top of stack	SBMUL	—{SBMUL }—	1		
	Multiply 4 constant	BMULS	—{BMULS }—	1	K	
BIN data divide instruction	Divide binary	BDIV	—{BDIV }—	1	R,P	
	Divide binary top of stack	SBDIV	—{SBDIV }—	1		
	Divide 4 constant	BDIVS	—{BDIVS }—	1	K	
Register math instruction	Increment BCD	INCR	—{INCR }—	2	R,P	
	Decrement BCD	DECR	—{DECR }—	2	R,P	
	Increment binary	BINC	—{BINC }—	2	R,P	
	Decrement binary	BDEC	—{BDEC }—	2	R,P	
Real math instruction	Add real	RADD	—{RADD }—	1	R,P	
	Add real constant	RADDC	—{RADDC }—	3	F	
	Subtract real	RSUB	—{RSUB }—	1	R,P	
	Subtract real constant	RSUBC	—{RSUBC }—	3	F	
	Multiply real	RMUL	—{RMUL }—	1	R,P	
	Multiply real constant	RMULC	—{RMULC }—	3	F	
	Divide real	RDIV	—{RDIV }—	1	R,P	
	Divide real constant	RDIVC	—{RDIVC }—	3	F	

*1 [ ]: the function memory that can be used by the second operand data.

Continued

Classification	Instruction name	Instruction	Symbol	Words number	Legal data types	Remark
Logic and instruction	And (16-bit)	ANDW	—〔ANDW 〕—|	1	R，P	
	And double (32-bit)	ANDD	—〔ANDD 〕—|	1	R，P	
	And formatted	ANDF	—〔ANDF 〕—|	2	GI,GQ,I,Q,M,S,T,C,SP [K(1-32)]	*1
	And stack	SAND	—〔SAND 〕—|	1		
	And 8 constant	ANDC	—〔ANDC 〕—|	2	K	
Logic or instruction	Or (16-bit)	ORW	—〔ORW 〕—|	1	R，P	
	Or double (32-bit)	ORD	—〔ORD 〕—|	1	R，P	
	Or formatted	ORF	—〔ORF 〕—|	2	GI,GQ,I,Q,M,S,T,C,SP [K(1-32)]	*1
	Or stack	SOR	—〔SOR 〕—|	1		
	Or 8 constant	ORC	—〔ORC 〕—|	2	K	
Logic exclusive or instruction	Exclusive or (16-bit)	XORW	—〔XORW 〕—|	1	R，P	
	Exclusive or (32-bit)	XORD	—〔XORD 〕—|	1	R，P	
	Exclusive or formatted	XORF	—〔XORF 〕—|	2	GI,GQ,I,Q,M,S,T,C,SP [K(1-32)]	*1
	Exclusive or stack	SXOR	—〔SXOR 〕—|	1		
	Exclusive or 8 constant	XORC	—〔XORC 〕—|	2	K	
Compare instruction	Compare	CMPR	—〔CMPR 〕—|	1	R，P	
	Compare double	CMPRD	—〔CMPRD 〕—|	1	R，P	
	Compare formatter	CMPRF	—〔CMPRF 〕—|	2	GI,GQ,I,Q,M,S,T,C,SP [K(1-32)]	*1
	Compare stack	SCMPR	—〔SCMPR 〕—|	1		
	Compare 8 constant	CMPRC	—〔CMPRC 〕—|	2	K	
	Compare real	RCMPR	—〔RCMPR 〕—|	1	R，P	
	Compare real constant	RCMPRC	—〔RCMPRC〕—|	3	F	

*1 [ ]: the function memory that can be used by the second operand data.

Continued

Classification	Instruction name	Instruction	Symbol	Words number	Legal data types	Remark
ACC conversion instruction	Invert	INV	—〔INV 〕—	1		
	Decimal coded binary	BIN	—〔BIN 〕—	1		
	Binary coded decimal	BCD	—〔BCD 〕—	1		
	Tens complement	BCDCPL	—〔BCDCPL 〕—	1		
	Encode	ENCO	—〔ENCO 〕—	1		
	Decode	DECO	—〔ENCO 〕—	1		
	Segment	SEG	—〔SEG 〕—	1		
	Shift right	SHFR	—〔SHFR 〕—	2	R,K	
	Shift left	SHFL	—〔SHFL 〕—	2	R,K	
	Rotate right	ROTR	—〔ROTR 〕—	2	R,K	
	Rotate left	ROTL	—〔ROTL 〕—	2	R,K	
	Sum	SUM	—〔SUM 〕—	1		
	Square root	SQRT	—〔SQRT 〕—	1		
	Gray code	GRAY	—〔GRAY 〕—	1		
	Shuffle digits	SFLDGT	—〔SFLDGT 〕—	1		
ACC conversion instruction	Radian conversion	RAD	—〔RAD 〕—	1		
	Degree conversion	DEG	—〔DEG 〕—	1		
Trigonometric sine, cosine, and tangent, and also their inverses instruction	Sine	SIN	—〔SIN 〕—	1		
	Cosine	COS	—〔COS 〕—	1		
	Tangent	TAN	—〔TAN 〕—	1		
	Arc sine	ASIN	—〔ASIN 〕—	1		
	Arc cosine	ACOS	—〔ACOS 〕—	1		
	Arc tangent	ATAN	—〔ATAN 〕—	1		
Real trigonometric sine, cosine, and tangent, and their inverses instruction	Sine real	RSIN	—〔RSIN 〕—	1		
	Cosine real	RCOS	—〔RCOS 〕—	1		
	Tangent real	RTAN	—〔RTAN 〕—	1		
	Arc sine real	RASIN	—〔RASIN 〕—	1		
	Arc cosine real	RACOS	—〔RACOS 〕—	1		
	Arc tangent real	RATAN	—〔RATAN 〕—	1		

Continued

Classification	Instruction name	Instruction	Symbol	Words number	Legal data types	Remark
Real ACC conversion instruction	Binary to real	REAL	—{REAL }—	1		
	Real to binary	INT	—{INT }—	1		
	Square root real	RSQRT	—{RSQRT }—	1		
	Radian real conversion	RRAD	—{RRAD }—	1		
	Degree real conversion	RDEG	—{RDEG }—	1		
Real exponentiation and logarithm instruction	power operation	RPOW	—{RPOW }—	1		
	Exponentiation operation	REXP	—{REXP }—	1		
	Natural logarithm operation	RLN	—{RLN }—	1		
	Common logarithm operation	RLG	—{RLG }—	1		
Time and data Instruction	Time	TIME	—{TIME }—	2	R	
	Date	DATE	—{DATE }—	2	R	
Set and reset bit instruction	Set bit	BITSET	—{BITSET }—	2	R,O	
	Reset bit	BITRST	—{BITRST }—	2	R,O	
Data block processing instruction	Move	MOVE	—{MOVE }—	2	R,P	
	And move	ANDMOV	—{ANDMOV}—	2	R,P	
	Or move	ORMOV	—{ORMOV }—	2	R,P	
	Exclusive or move	XORMOV	—{XORMOV}—	2	R,P	
	Swap table data	SWAP	—{SWAP }—	2	R	
	ASCII to HEX	ATH	—{ATH }—	2	R	
	HEX to ASCⅡ	HTA	—{HTA }—	2	R	
	Fill	FILL	—{FILL }—	2	R,K	
	Table shift right	TSHFR	—{TSHFR }—	2	R,O	
	Table shift left	TSHFL	—{TSHFL }—	2	R,O	
Table search instruction	Find	SRCH	—{SRCH }—	2	R,K	
	Find block	BSRCH	—{BSRCH }—	2	R	
	Find greater than	CLASS	—{CLASS }—	2	R,K	

Continued

Classification	Instruction name	Instruction	Symbol	Words number	Legal data types	Remark
Table processing with pointer instruction	Table to destination	TTD	—{TTD }—|	2	R	
	Remove from bottom of table	RFB	—{RFB }—|	2	R	
	Remove from top of table	RFT	—{RFT }—|	2	R	
	Source to table	STT	—{STT }—|	2	R,K	
	Add to top of table	ATT	—{ATT }—|	2	R,K	
Data registration instruction	Data label	DLBL	—{DLBL }—|	2	K	
	Numeric constant	NCON	—{NCON }—|	1	K	
	ASCII constant	ACON	—{ACON }—|	1	A	
	Load indexed from constant	LDSIX	—{LDSIX }—|	2	K	
	Load label	LDLBL	—{LDLBL }—|	2	K	
	Move block	MOVAS	—{MOVAS }—|	2		
	Move memory cartridge	MOVMC	—{MOVMC }—|	2	R,K	
Drum control instruction	Timed drum	DRUM	—{DRUM }—|			
	Event drum	EDRUM	—{EDRUM }—|			
	Masked drum event discrete	MDRMD	—{MDRMD }—|			
	Masked drum event word	MDRMW	—{MDRMW }—|			

4. Special object instruction (15)

Classification	Instruction name	Instruction	Symbol	Words number	Legal data types	Remark
Communication instruction	Read from network	RX	—{RX }—|	2	GI,GQ,I,Q,M,S,T,C,SP,R,P	
	Write to network	WX	—{WX }—|	2	GI,GQ,I,Q,M,S,T,C,SP,R,P	
MODBUS RTU communication instruction	Read from MODBUS network	MRX	—{MRX }—|		GI,GQ,I,Q,M,S,T,C,SP,R,P	
	Write to MODBUS network	MWX	—{MWX }—|		GI,GQ,I,Q,M,S,T,C,SP,R,P	
ASCII code processing instruction	ASCII in	AIN	—{AIN }—|			
	ASCII print from R–Memory	PRINTV	—{PRINTV }—|			
Pulse output instruction	Home search	MT_ORG	—{MT_ORG }—|		K,R,P	
	Absolute value positioning	MT_ABSC	—{MT_ABSC}—|		K,R,P	
	Relative value positioning	MT_INCC	—{MT_INNC}—|		K,R,P	
	Velocity control	MT_SPDC	—{MT_SPDC}—|		K,R,P	
	Interrupt positioning	MT_INTC	—{MT_INTC}—|		K,R,P	
	Change current Position	MT_PSET	—{MT_PSET}—|		K,R,P	
	Reset	MT_RSTE	—{MT_RSTE}—|		K	
	Stop	MT_STPM	—{MT_STPM}—|		K,R,P	
	Multi-positioning table	MT_TABL	—{MT_TABL}—|		K,R,P	

5. Binary compare instruction (4)

The four instructions are supported by KPPSoft Version V1.0.1.10 (or later).

Classification	Instruction name	Instruction	Symbol	Words number	Legal data types	Remark
Binary compare instruction	16BIT	BCMPR	—{BCMPR }—|	1	R，P	
	32BIT	BCMPRD	—{BCMPRD}—|	1	R，P	
	Compare stack	BSCMPR	—{BSCMPR }—|	1		
	Compare 8 constant	BCMPRC	—{BCMPRC }—|	2	K	

Comparison with the sign bit or without the sign bit dedends on the data of R7633.15.

R7633.15=1, compare with the sign bit

R7633.15=0, compare without the sign bit

Note

1. Moving data from R-memory to user-memory by using MOVMC instruction will lengthen PLC scan time.
2. If the words number columns are blank, their words number is in relation to their content.

6. IBOX instruction (25)

supported by KPPSoft software version V1.0.1.19 or later.

Classification	Instruction name	Instruction Code	Instruction	Remark
Memory instruction	Move single word	IB-200	MOVEW	
	Move double word	IB-201	MOVED	
Discrete instruction	Push on/push off circuit	IB-300	PONOFF	
	On delay timer	IB-301	ONDTMR	
	Off delay timer	IB-302	OFFDTMR	
	One shot	IB-303	ONESHOT	
Analog instruction	Hi/low alarm-binary	IB-401	HILOALB	
	Filter over time-binary	IB-402	FILTERB	
	Anolog scale 12 bit binary to binarey	IB-403	ANSCLB	
	Hi/low alarm-BCD	IB-421	HILOAL	
	Filter over time-BCD	IB-422	FILTER	
	Anolog scale 12 bit BCD to BCD	IB-423	ANSCL	
Math instruction	Math-binary	IB-501	MATHBIN	
	Sum binary numbers	IB-502	SUMBIN	
	Square-binary	IB-503	SQUAREB	
	Math-BCD	IB-521	MATHBCD	
	Sum BCD numbers	IB-522	SUMBCD	
	Square-BCD	IB-523	SQUARE	
	Math-real	IB-541	MATHR	
	Sum real numbers	IB-542	SUMR	
	Square-binary	IB-543	SQUARER	
	BCD to real with implide decimal point	IB-560	BCDTOR	
	Real to BCD with decimal Point and rounding	IB-561	RTOBCD	
	Double BCD to real with implide decimal point	IB-562	BCDTORD	
	Real to double BCD with decimal point and rounding	IB-563	RTOBCDD	

7. double words contact / 32 bits math arithmetic instruction (16)

Need the system fireware version 20190716 or later , KPPSoft software version V1.6.4.7 or later.

(1) ordinary instruction (16)

Classification	Instruction name	Instruction	Symbol	Words number	Legal data types	Remark
Double words Comparative equal instruction	Store if equal	DLDEQ	—|=|—	2	R,P [K,R,P]	*1
	Store if not equal	DLDNEQ	—/=|—	2	R,P [K,R,P]	*1
	And if equal	DANDEQ	—|=|—	2	R,P [K,R,P]	*1
	And if not equal	DANDEQ	—/=|—	2	R,P [K,R,P]	*1
	Or if equal	DOREQ	—|=|—	2	R,P [K,R,P]	*1
	Or if not equal	DORNEQ	—/=|—	2	R,P [K,R,P]	*1
Double words Comparative greater and equal instruction	Store if greater or equal	DLDGE	—|≥|—	2	R,P [K,R,P]	*1
	Store if not greater or equal	DLDNGE	—/≥|—	2	R,P [K,R,P]	*1
	And if greater or equal	DANDGE	—|≥|—	2	R,P [K,R,P]	*1
	And if not greater or equal	DANDNGE	—/≥|—	2	R,P [K,R,P]	*1
	Or if greater or equal	DORGE	—|≥|—	2	R,P [K,R,P]	*1
	Or if not greater or equal	DORNGE	—/≥|—	2	R,P [K,R,P]	*1
32 bits Math arithmetic Instruction	BCD constant Multiply	MULC	—{MULC }—	2	K	
	BCD constant Divide	DIVC	—{DIVC }—	2	K	
	BIN Multiply	BMULD	—{BMULD }—	2	R,P,K	*2
	BIN Divide	BDIVD	—{BDIVD }—	2	R,P,K	*2

*1 [ ]: the function memory that can be used by the second operand data.

*2 R7633.15=1, operating with the sign bit

R7633.15=0, operating without the sign bit

(2) IBOX instruction (3)

Classification	Instruction name	Instruction Code	Instruction	Remark
Math Arithmetic instruction	32 bits BCD Arithmetic	IB-550	MATHBCDD	*1
	32 bits BIN Arithmetic	IB-551	MATHBIND	*1
	CRC generation instruction	IB-552	CRCXOR	

Appendix 2 Additonal information about software patents

This product uses TOPPERS software technology and TINET protocol stack technology of TOPPERS Association, and TINET is developed based on FreeeBSD.
Based on the requirements of the above software technology copyright owner, make the following copyright description for the use of the above software technology.

(1)FreeBSD

Copyright (c) 1980, 1986 and 1993

The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the regents and contributors. “As is” and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the regents or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services, loss of use, data, or profits; or business interruption). However caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

(2)KAME

(3) Ethernet・device driver

Device driver for National Semiconductor DS8390/WD83C690 based Ethernet adapters. By David Greenman, 29-April-1993

Currently supports the Western Digital/SMC 8003 and 8013 series, the SMC Elite Ultra (8216), the 3Com 3c503, the NE1000 and NE2000, and a variety of similar clones.

(4)/usr/sbin/ppp

User Process PPP

Written by Toshiharu OHNO (tony-o@iij.ad.jp)

(5)/usr/sbin/pppd

main.c - Point-to-Point Protocol main module

(6)TINET 和 TOPPERS

TINET (TCP/IP Protocol Stack)

Koyo ELECTRONICS （WUXI） CO., LTD.

Add: 21st Floor, Building 1, No.599, Jianzhuxi Road, Binhu District, Wuxi, Jiangsu, P.R.China
Pc: 214072
Tel: 0510-85167888 Fax: 0510-85161393
http://www.koyoele.com.cn

KEW-M2511C2

January 2020