Programmable Logic Controller Experiment Box

(SiemensS7-1200)

Experimental Manual

Content

Chapter One System Introduction……………………………………………………………..........……………2

一、PLC Mainframe………………………………………………………………............………2

二、Programming Device……………………………………………………….........…...……4

Chapter Two Software Installation and Use……………………………………………………..…………4

一、Software Installation……………………………………………………………………...……4

二、Use of Software…………………………………......………………………………………4

Chapter Three PLC Control Experiment ……………………………......…………………………………5

Chapter Four PLC Control Hardware Experiment………………………………………………........………23

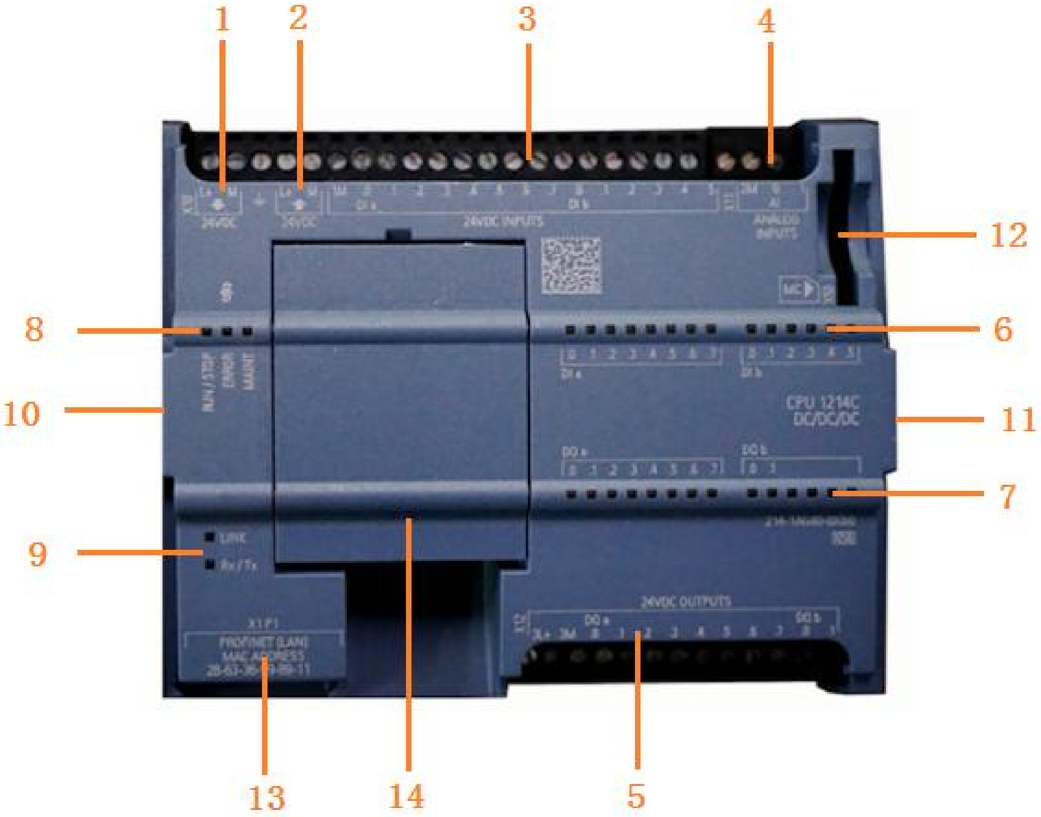
Chapter One System Introduction

Siemens (SIMATIC) S7-1200 series PLC (Micro PLC) can be applied to various automation systems. The compact structure, low cost and powerful instructions make the S7-1200 PLC an ideal solution for various automation control tasks. The diversification of S7-1200 products and Windows-based programming tools enable you to complete automation tasks more flexibly.

S7-1200 has strong functions and small size. The AC power supply can be varied within the range of 85-265V, and there is also a DC-24V power supply for input. Programmable controller (referred to as PC) is required to send the code table of the user program to the program memory of the PC during production control or experiment. When running, the PC performs calculation and judgment based on the detected input signal and program, and then Control the object through the output circuit. So a typical PC system consists of the following three parts: input/output interface, PC host, and communication port.

一、PLC Mainframe

In our experiment box, the selected PC host is SIMATIC S7-1200 CPU1214C, with 14 input points, 10 output points, and IO expansion module SM1223, with 8 input points and 8 output points. Two programming methods, mnemonic and ladder diagram, are available.



1、PLC power supply terminal。

Depending on the PLC model, it can be DC 24V power supply or AC 220V power supply. If the PLC model is DC/DC/DC, DC/DC/RLY uses DC 24V power supply, if the PLC model is AC/DC/RLY uses AC 220V power supply.

2、24V output power。 PLC provides a 24V power output, which can be used to power sensors or modules.

CPU1211C and CPU1212C can provide 300mA current, CPU1214C/1215C/1217C can provide 400mA current. Because the current capacity is limited, when the current capacity of the sensor or module we use exceeds the specified value, the built-in power supply cannot be used. So I suggest not to use it, but to connect the 24V switching power supply externally.

3、Digital input terminal。Switches, buttons, sensors, encoders and other digital signals or pulse signals can be connected to the PLC through the digital input terminals. The input connection of 1200PLC can support source connection and sink connection.

4、Analog input terminal。S7-1214C supports two analog inputs of 0-10V voltage signals. When you need to use the analog input function, connect some sensors to this input terminal.

5、Digital output terminal。Digital output terminals are used to connect external loads, such as indicator lights, relays, solenoid valves, etc. Depending on the PLC output type, the wiring method is also different. The PLC output type mentioned here refers to whether the PLC is a transistor output or a relay output. The transistor output is connected to a DC load. The relay output can be connected to an AC or a DC load.

6、Input indicator。When there is a signal input, the corresponding input indicator will light up in green.

7、Output indicator。When there is a signal output, the corresponding output indicator will light up in green.

8、Status Indicator。There are three status indicators on the PLC, STOP/RUN indicator, ERROR indicator and MAINT indicator. When the STOP/RUN indicator is green, it means that the PLC is in RUN mode. When it is orange, it means that the PLC is in STOP mode. If it is blinking alternately between green and orange, it means that the CPU is starting. When the ERROR indicator flashes in red, it means there is an error, such as CPU internal error, configuration error, etc. When it is red, it means hardware failure. The MAINT indicator flashes every time a memory card is inserted.

9、Network status indicator。Including LINK and Rx/Tx indicator lights, mainly used to display the network connection status. If there is no problem with the hardware connection, the LINK indicator is always on, and the Rx/Tx indicator will flash when data is being exchanged.

10、Communication module expansion port。S7-1200 PLC can expand up to 3 communication modules, which are expanded on the communication module expansion port on the left side of the CPU.

11、Signal module expansion slot。Signal modules include digital input, digital output, digital input/output, analog input, analog output, analog input/output and other modules. These signal modules are extended to this module expansion slot.

12、MC card slot。S7-1200 PLC provides a dedicated MC card, which can be used as a program card, transfer card, or to update hardware firmware, as well as to clear passwords and restore factory values. This MC card can support 4M, 12M, 14M, 2G.

13、PROFINET/IE interface。 This interface can support PROFINET communication and Ethernet communication, and can be used for communication connection between PLC and programming software, communication connection between PLC and touch screen/host computer, and Ethernet communication between PLC and PLC. In addition to supporting the S7 protocol, this port also supports communication protocols such as TCP/IP, UDP, ISO_on_TCP, and MODBUS TCP.

14、Signal board expansion slot。 S7-1200 PLC can expand the signal board, including digital input signal board, digital output signal board, digital input and output mixed signal board, analog input signal board, analog output signal board, communication signal board and battery signal board . These signal boards are extended directly above the PLC. The advantage of using these signal boards is that input and output points can be added without taking up space.

二、Programming device

Usually a microcomputer is used as a programming device. Install the PLC compiling and debugging software STEP 7 BASIC V14 from SIEMENS, connect the computer's network port and the programming interface of the PLC host with a super five network cable, and run STEP 7 BASIC V14 to download the coding table of the PLC program to the PLC Various control experiments can be performed by running the program in the memory. For the installation and use of STEP 7 BASIC V14 application software, please refer to the separate volume STEP _7_BASIC_V14_1_enUS.pdf.

Chapter Three　PLC Control Experiment

Experiment 1　Basic Instruction Experiment

一、Experiment Purpose

1. Master the usage of basic instruction.
2. Learn how to program the basic logic combinational circuit by using basic instruction.
3. Usage of programming and debugging software.

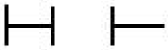
二、Experiment Material

1、 PC
2、 PLC experimental platform
3、 Communication cable
4、 Wires

三、Experiment Content

There are 10 common basic instructions of SIEMENS S7-1200 series programmable controller. This experiment carries out the programming operation training of commonly used basic instructions LD, LDN, A, AN, NOT, O, ON, ALD, OLD, =. A brief introduction is as follows:

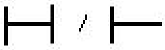
1、Instruction fetch

Mnemonic symbol：LD　　　Ladder Symbol： 

Data: contact number. Except for the data channel, the rest of the relay numbers of the PC are available.

Function: Read in the first normally open contact of a logic line (also called branch).

2、Invert instruction

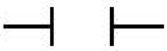
Mnemonic symbol：LDN　　　Ladder Symbol： 

Data: same as LD instruction

Function: Read the first normally closed contact of the logic line.

In the ladder diagram, each logic line must start with a contact, so the LD or LDN instruction must be used. In addition, this instruction is also used for the start of each branch in the circuit block, or the start of the branch circuit after the branch point, and is used in conjunction with other instructions.

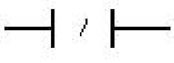
3、With instructions

Mnemonic symbol：A　　　Ladder Symbol： 

Data: contact number.

Function: logical AND operation, that is, a normally open contact in series.

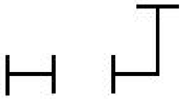
4、NAND instruction

Mnemonic symbol：AN　　　Ladder Symbol： 

Data: contact number, same as A command.

Function: logical NAND operation, that is, a normally closed contact in series.

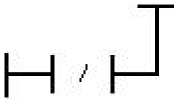
5、Or instruction

Mnemonic symbol：O　　Ladder Symbol：

Data: contact number, the range is the same as the A command.

Function: logical OR operation, that is, a normally open contact in parallel.

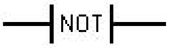
6、Or non-instruction

Mnemonic symbol：ON　　　Ladder Symbol：

Data: contact number, the range is the same as the A command.

Function: logical NOR operation, that is, a normally closed contact in parallel.

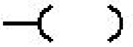
7、Non-command

Mnemonic symbol：NOT　　　Ladder Symbol：

Data: contact number, the range is the same as the A command.

Function: Logic non-operation, that is, a normally closed contact in parallel.

8、Output instruction

Mnemonic symbol：=　　　Ladder Symbol：

Data: Relay coil number.

Function: output the operation result of the logic line.

9、Circuit blocks and instructions

Mnemonic symbol：ALD　Ladder Symbol：none

Data: None

Function: Connect two circuit blocks in series.

10、Circuit block or instruction

Mnemonic symbol：OLD　Ladder Symbol：none

Data: None

Function: Connect two circuit blocks in parallel.

Description：

LD、A、O：Normally open contact instruction；

LDN、AN、ON：Normally closed contact instruction；

When the bit value is 1, the normally open contact is closed; when the bit value is 0, the normally closed contact is closed.

四、Experimental Steps

1. Before the experiment, first use the download cable to connect the PC network port to the PORT1 port of the S7-1200-CPU1214C host, then power on the experiment box and turn on the 24V power switch. The indicator lights of the host and 24V power supply are on, indicating that it is working normally and you can enter the next experiment.

2. Enter the compiling and debugging environment, and enter the following exercise programs with command symbols or ladder diagrams.

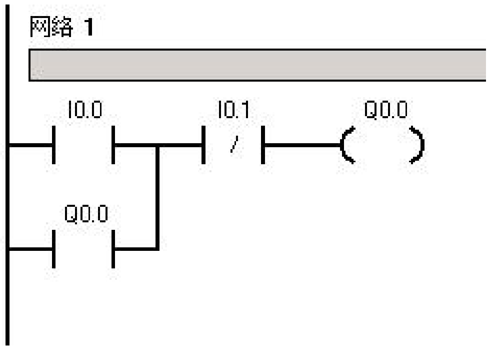
3. Make corresponding connections according to the procedure. (For wiring, see "How to Use Input/Output Ports" in Chapter One)

4. Download the program and run it, and observe the results.

Exercise1：

Network 1

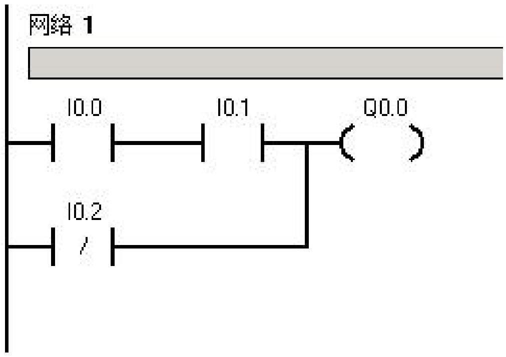
```
LD     I0.0
O      Q0.0
AN     I0.1
=      Q0.0
```



Exercise2：

Network 1

```
LD     I0.0
A      I0.1
ON     I0.2
=      Q0.0
```

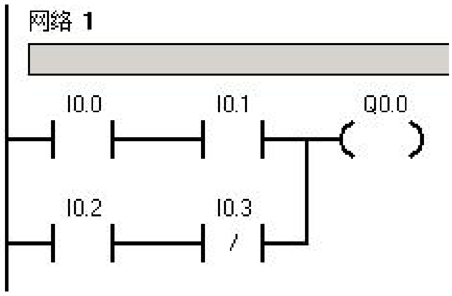


Exercise 3： When you want to connect two program segments (also called circuit blocks) in the program, you need to use circuit block connection instructions. Each circuit block starts with the LD or LDN instruction.

ALD instruction：

Network 1

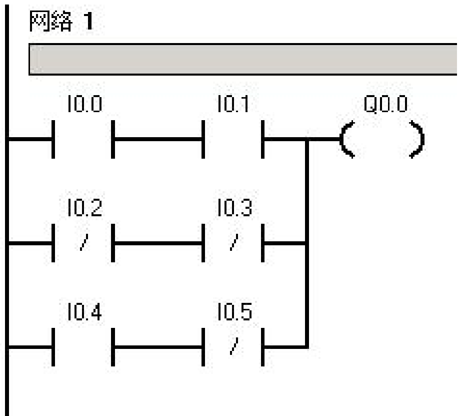
```
LD     I0.0
A      I0.1
LD     I0.2
AN     I0.3
OLD
=      Q0.0
```



OLD instruction：

Network 1

```
LD     I0.0
A      I0.1
LDN    I0.2
AN     I0.3
```



```
OLD
LD        I0.4
AN        I0.5
OLD
=         Q0.0
```

Experiment 2　Timer and Counter Experiment

一、Experiment Purpose

1. Master the use of common timing commands.
2. Master the use of counter instructions
3. Master the setting of the internal time base pulse parameters of the timer/counter.
4. Familiar with the use of compiling and debugging software.

二、Experiment Material

1、PC

2、PLC experimental platform

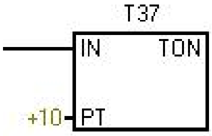
3、Communication cable

4、Wires

三、Experiment Content and Step

1、Timer：

Mnemonic symbol：TONxx　　　Ladder Symbol：



Data: xx (37): is the selected timer number; PT (+10): is the set value of the timer, expressed in a 4-digit decimal number, the timing unit is 0.1 seconds, so the lowest digit is tenths. For example, the setting value for 5 seconds is +50.

The timing range is 0.1 to 3276.7 seconds.

Function: Timed time to turn on the timer contact.

The timer is an increment timer. When the input condition is ON, the timer starts to increase by 1, and the current value of the timer is increased by 1 every 0.1 seconds. When the current value of the timer is equal to the set value, the timer expires and the timer contact is turned on and maintained. When the input condition is OFF, no matter what state the timer is in, it will be reset, the current value will be restored to 0, and the corresponding moving and closing contact will be disconnected. The timer is equivalent to the time relay. When the power is off, the timer is reset.

2、Counter：

Mnemonic symbol：CTU、CTD　Ladder Symbol：

among them:

Cxx: xx is the counter number 00～255;

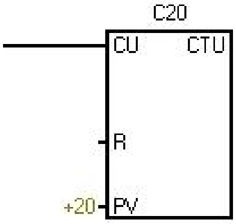
PV: is the count setting value, expressed by a 4-digit decimal number, and the count range is 1 to 32767.

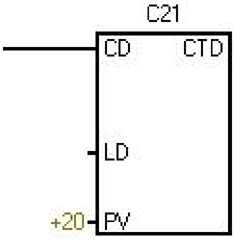
R, LD: is the reset terminal.

CTU is an up counter; CTD is a down counter.

Features:：

Up-counter: Starting from the current count value, it will count up when each (CU) input state goes from low to high. When the current value of Cxx is greater than or equal to the preset value PV, the counter Cxx is set. When the reset terminal (R) is turned on or the reset instruction is executed, the counter is reset. When it reaches the maximum value (32767), stop counting.





Down counter: Starting from the current count value, count down when each (CD) input state goes from low to high. When the current value of Cxx is equal to 0, the counter Cxx is set. When the load input (LD) is turned on, the counter is reset and the current value of the counter is set to the preset value PV. When the count value reaches 0, the counter stops counting and the counter bit Cxx turns on.

四、Experimental Steps

1. Before the experiment, first use the download cable to connect the PC network port to the PORT1 port of the S7-1200-CPU1214C host, then power on the experiment box and turn on the 24V power switch. The indicator lights of the host and 24V power supply are on, indicating that it is working normally and you can enter the next experiment.

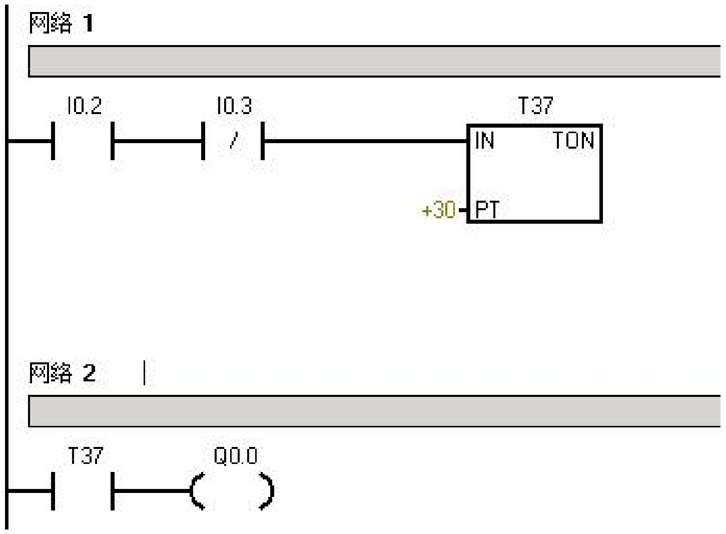
2. Enter the compiling and debugging environment, and enter the following exercise programs with command symbols or ladder diagrams.

3. Make corresponding connections according to the procedure. (For wiring, see "How to Use Input/Output Ports" in Chapter One)

4. Download the program and run it, and observe the results.

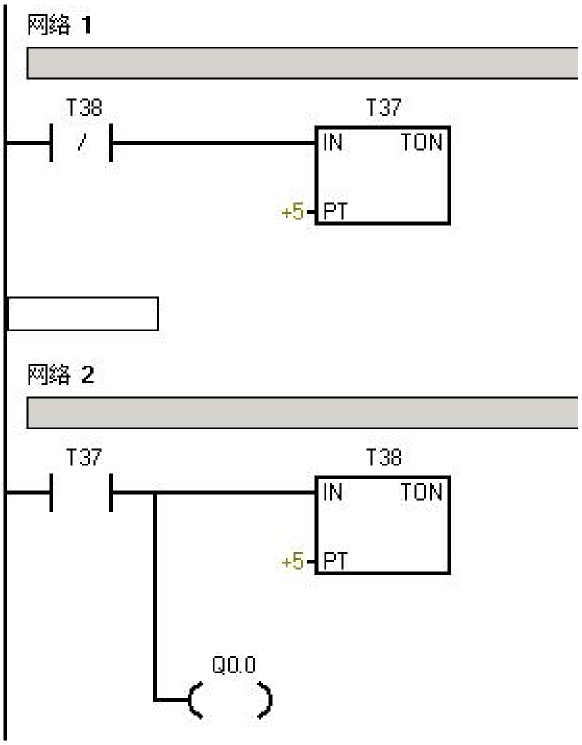
Exercise1：Delay

```
    Network 1                                                      LD
I0.2
      AN        I0.3
      TON       T37, +30
    Network 2
      LD        T37
      =         Q0.0
```



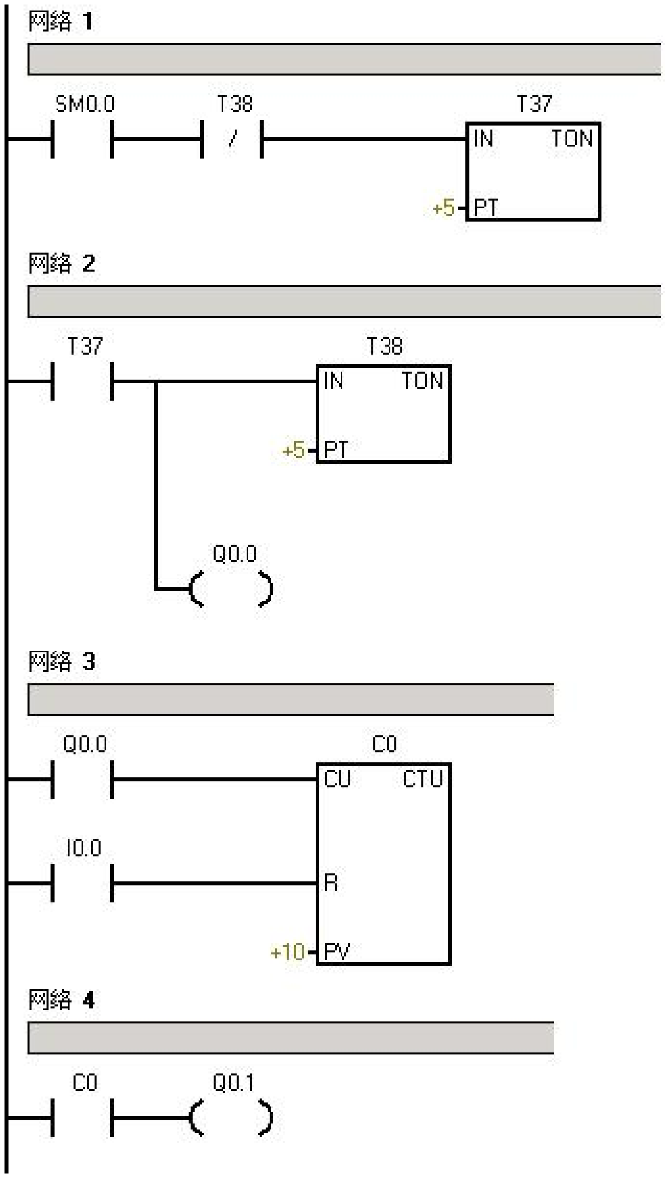
Exercise2：Second pulse generator

```
    Network 1
        LDN       T38
        TON       T37, +5
    Network 2
        LD        T37
        TON       T38, +5
        =         Q0.0
```



Exercise3：Up counter

```
    Network 1
        LD      SM0.0
        AN      T38
        TON     T37, +5
     Network 2
        LD      T37
        TON     T38, +5
        =       Q0.0
     Network 3
        LD      Q0.0
        LD      I0.0
        CTU     C0, +10
     Network 4
        LD      C0
        =       Q0.1
```



Exercise4：Design down counter by yourself (refer to up counter)

Experiment 3　Jump Instruction Experiment

一、Experiment Purpose

1、Master the use of jump instructions and label instructions.

2、Usage of programming and debugging software.

二、Experiment Material

1、PC

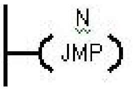
2、PLC experimental platform

3、Communication cable

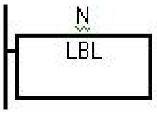
4、Wires

三、Experiment Content

1、Jump instruction

Mnemonic symbol：JMP　（N）　　Ladder Symbol：

2、Label instruction

Mnemonic symbol：LBL　（N）　　Ladder Symbol：

Jump instruction: can make the program flow jump to the program branch at the designated label N. Can be used for working mode selection.

Label instruction: mark the position N of the jump destination.

四、Experimental Steps

1. Before the experiment, first use the download cable to connect the PC network port to the PORT1 port of the S7-1200-CPU1214C host, then power on the experiment box and turn on the 24V power switch. The indicator lights of the host and 24V power supply are on, indicating that it is working normally and you can enter the next experiment.

2. Enter the compiling and debugging environment, and enter the following exercise programs with command symbols or ladder diagrams.

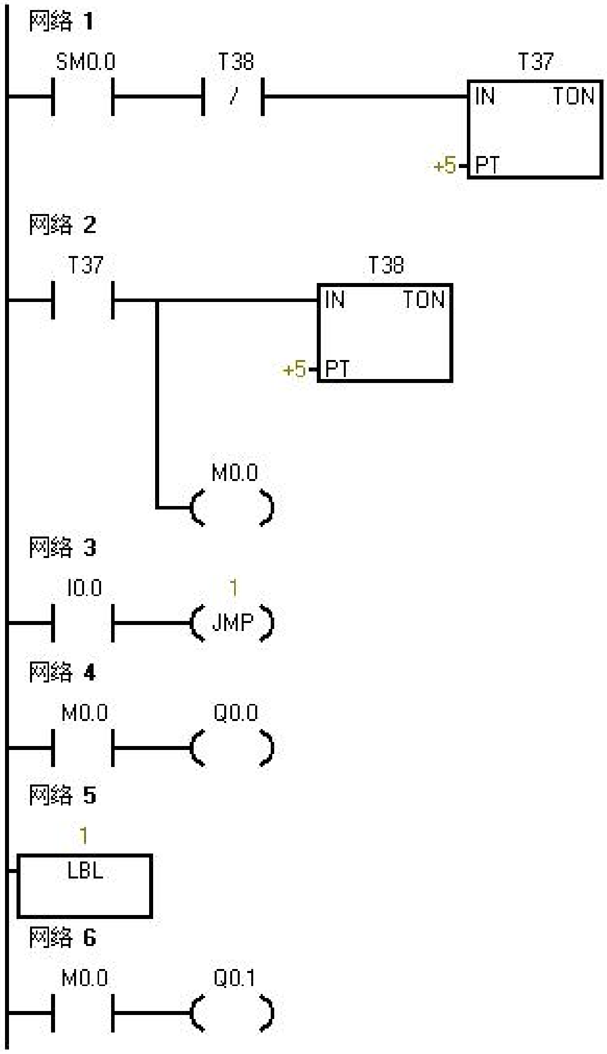
3. Make corresponding connections according to the procedure. (For wiring, see "How to Use Input/Output Ports" in Chapter One)

4. Download the program and run it, and observe the results.

Exercise1：

Network 1

```
        LD        SM0.0
        AN        T38
        TON       T37, +5
    Network 2
        LD        T37
        TON       T38, +5
        =         M0.0
    Network 3
        LD        I0.0
        JMP       1
    Network 4
        LD        M0.0
        =         Q0.0
    Network 5
        LBL       1
    Network 6
        LD        M0.0
        =         Q0.1
```



Exercise2： Design a program using jump instructions by yourself and verify it on the computer.

Experiment 4　Set/Reset and Pulse Command Experiment

一、Experiment Purpose

1、Master the use of S, R, EU and ED directives.

2、Usage of programming and debugging software.

二、Experiment Material

1、PC

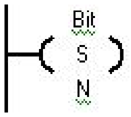
2、PLC experimental platform

3、Communication cable

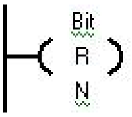
4、Wires

三、Experiment Content and Steps

1、Set:

Mnemonic symbol：S　　Ladder Symbol：

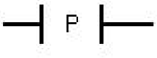
2、Reset:

Mnemonic symbol：R　　Ladder Symbol：

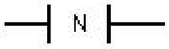
Function: It is equivalent to a latch, when it is set (ON), it will remain until reset.

Set (S) or reset (R) instructions will set or reset N points from the address specified by Bit. You can set or reset 1-255 points at a time. If the reset instruction specifies a timer or counter, the instruction not only resets the timer bit or counter bit, but also clears the current value of the timer or counter.

3、Positive transition instruction：

Mnemonic symbol：EU　　Ladder Symbol：

4、Negative transition instruction：

Mnemonic symbol：ED　　Ladder Symbol：

Function：

The positive transition instruction is at the leading edge of the input pulse (from 0 to 1), which makes the designated relay turn on for a scan period and then release, that is, the leading edge of the input state is differentially output to the designated relay.

The negative transition instruction makes the specified relay turn on for a scan cycle and then release it on the back edge of the input pulse (from 1 to 0), that is, the back edge differential of the input state is output to the specified relay.

Both of these differential instructions take effect when the input state changes. In the program running, the always-on input condition will not cause the execution of EU; the always-off input will not cause the execution of ED.

四、Experimental Steps

1. Before the experiment, use the download cable to connect the PC network port to the PORT1 port of the S7-1200-CPU1214C host, then power on the experiment box and turn on the 24V power switch. The indicator lights of the host and 24V power supply are on, indicating that it is working normally and you can enter the next experiment.

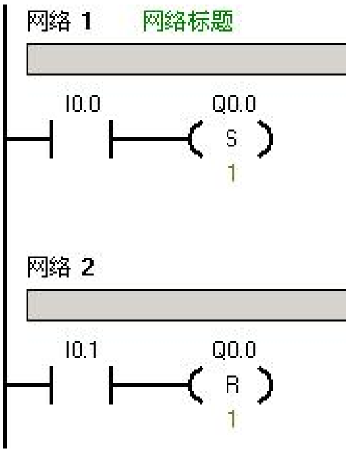
2. Enter the compiling and debugging environment, and enter the following exercise programs with instruction symbols or ladder diagrams.

3. Make corresponding connections according to the procedure. (For wiring, see "How to Use Input/Output Ports" in Chapter One)

4. Download the program and run it, and observe the results.

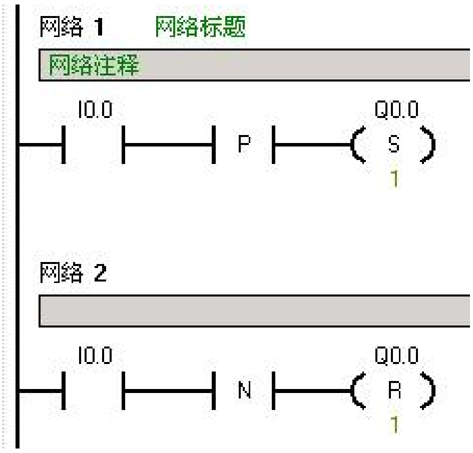
Exercise1: Programming of set (S) or reset (R) instructions

```
Network 1
        LD        I0.0
        S         Q0.0, 1
Network 2
        LD        I0.1
        R         Q0.0, 1
```



Exercise2: Programming of positive and negative transition instructions

```
Network 1
        LD        I0.0
        EU
        S         Q0.0, 1
Network 2
        LD        I0.0
        ED
        R         Q0.0, 1
```



In the example: I0.0 is connected to the button,

When the button is pressed, it is equivalent to a transition from 0 to 1 on I0.0, so the input is turned on for a scan period, and Q0.0 is set to light up.

When the button is released, it is equivalent to a transition from 1 to 0 on I0.0, so the input is turned on for a scan period, and Q0.0 is reset, that is, it goes out.

Experiment 5　Shift Register Instruction Experiment

一、Experiment Purpose

1、Master the use of shift instructions.

2、Usage of programming and debugging software.

二、Experiment Material

1、PC

2、PLC experimental platform

3、Communication cable

4、Wires

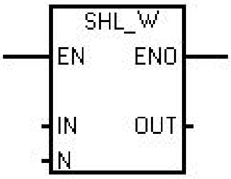
三、Experiment Content

S7-1200 provides a wealth of shift instructions, including left shift and right shift instructions, rotate left and rotate right instructions, as well as byte shift, word shift, and double word shift instructions. Here are a few commonly used commands, please refer to related manuals for others.

1、shift left

Mnemonic symbol：　SHL_W

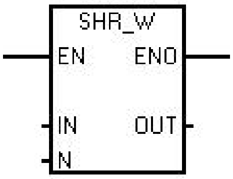
Ladder Symbol：



2、shift right

Mnemonic symbol：　SHR_W

Ladder Symbol：



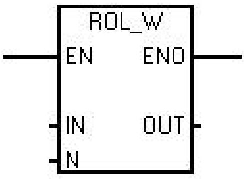
The shift instruction shifts the input value IN right or left by N bits, and loads the output result into OUT.

The shift instruction automatically adds zeros to the shifted bits. If the number of bits N is greater than or equal to the maximum allowable value (8 for byte operations, 16 for word operations, and 32 for double word operations), then the number of shift operations is the maximum allowable value. If the number of shifts is greater than 0, the overflow flag (SM1.1) is the value of the most recently shifted bit. If the result of the shift operation is 0, the zero memory bit (SM1.0) is set.

Byte operations are unsigned. For word or double word operations, the sign bit is also moved when using the symbol data type.

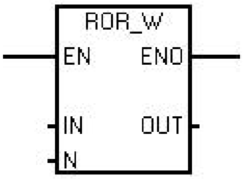
3、Rotate shift left

Mnemonic symbol：ROL_W　　　Ladder Symbol：



4、Rotate shift right

Mnemonic symbol：ROR_W　　　Ladder Symbol：



The rotation instruction rotates the input value IN to the right or left by N bits, and loads the output result into OUT. If the number of bits N is greater than or equal to the maximum allowable value (8 for byte operations, 16 for word operations, and 32 for double word operations), S7-1200 will perform a modulo operation before performing the cyclic shift, and get a The effective number of shifts. The result of the modulo operation is 0 to 7 for byte operations, 0 to 15 for word operations, and 0 to 31 for double word operations. If the number of shifts is 0, the rotation instruction is not executed. If the rotate instruction is executed, the value of the last bit will be copied to the overflow flag bit (SM1.1).

If the number of shifts is not an integer multiple of 8 (for byte operations), 16 (for word operations), or 32 (for double word operations), the last bit shifted out will be copied to the overflow flag (SM1.1). When the result of the cyclic shift is 0, the result is a zero flag (SM1.0) is set.

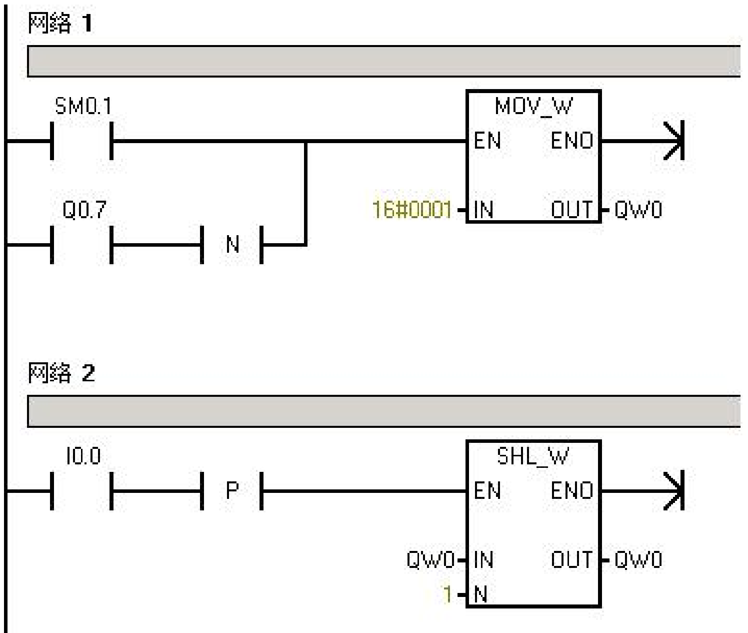
Byte operations are unsigned. For word or double word operations, the sign bit is also moved when using the symbol data type.

四、Experimental Steps

1. Before the experiment, use the download cable to connect the PC network port to the PORT1 port of the S7-1200-CPU1214C host, then power on the experiment box and turn on the 24V power switch. The indicator lights of the host and 24V power supply are on, indicating that it is working normally and you can enter the next experiment.
2. Enter the compiling and debugging environment, and enter the following exercise programs with instruction symbols or ladder diagrams.
3. Make corresponding connections according to the procedure. (For wiring, see "How to Use Input/Output Ports" in Chapter One)
4. Download the program and run it, and observe the results.

Exercise 1: The following is an exercise for shift left instruction. Read and understand the program, and download the program to the host for verification.

```
Network 1
    LD      SM0.1
    LD      Q0.7
    ED
    OLD
    MOVW    16#0001, QW0
Network 2
    LD      I0.0
    EU
    SLW     QW0, 1
```



Experimental results： Connect I0.0 to the button. Each time you press the button, the next indicator light will be illuminated in cycles.

Exercise2： Change the N in exercise 1 to 2 and observe the experimental results.

Exercise3：Change the shift left instruction in exercise 1 to shift right instruction, rotate shift left instruction, and rotate shift right instruction, and observe the experimental results.

Exercise4： Change the word instruction in exercise 1 to byte instruction, double word instruction, and observe the experimental results.

Experiment 6　Common Function Instruction Experiment

一、Experiment Purpose

1、Master the use of data comparison instructions, data transfer instructions, addition instructions, and subtraction instructions.

2、Usage of programming and debugging software.

二、Experiment Material

1、PC

2、PLC experimental platform

3、Communication cable

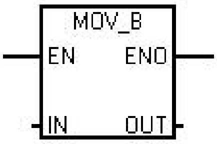
4、Wires

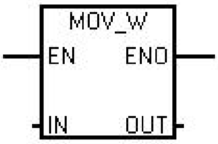
三、Experiment Content

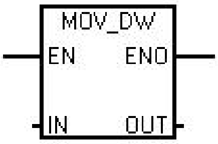
1、Transfer instructions

Mnemonic symbol：　MOV_B　MOV_W　MOV_DW

Ladder Symbol：





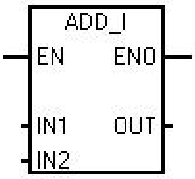


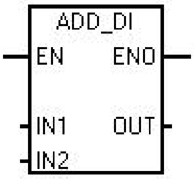
Transfer instructions can be divided into byte transfer (MOV_B), word transfer (MOV_W), double word transfer (MOV_DW) and real number transfer instructions. Transfer the value in IN to OUT without changing the original value.

2、Addition instruction

Mnemonic symbol：　ADD_I　ADD_DI

Ladder Symbol：

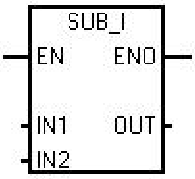


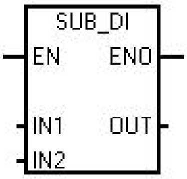


3、Subtraction instruction

Mnemonic symbol：　SUB_I　SUB_DI

Ladder Symbol：





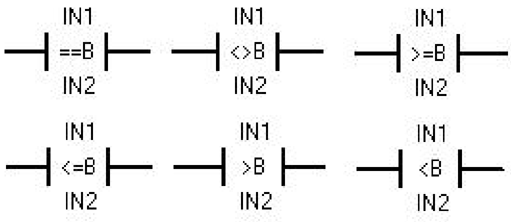
Integer addition (ADD_I) or integer subtraction (SUB_I) instructions add or subtract two 16-bit integers to produce a 16-bit result.

The double integer addition (ADD_DI) or double integer subtraction (SUB_DI) instructions add or subtract two 32-bit integers to produce a 32-bit result.

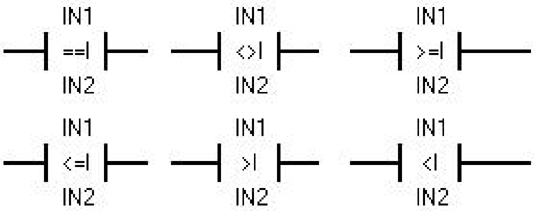
Addition	Subtraction
IN1+IN2 = OUT	IN1–IN2 = OUT

4、Comparison instruction

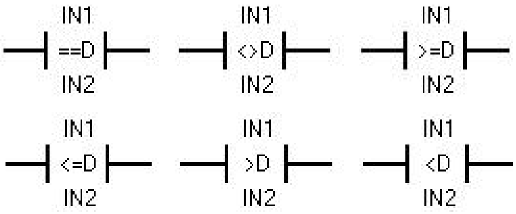
Byte comparison instruction：



Word comparison instruction：



Double word comparison instruction：



Comparison instructions are used to compare two values：

IN1 = IN2	IN1 >= IN2	IN1 <= IN2
IN1 > IN2	IN1 < IN2	IN1 <> IN2

The byte (B) comparison operation is unsigned.

Integer (I) comparison operations are signed.

Double word (D) comparison operation is signed.

When the comparison result is true, the comparison command enable point is closed or the output is connected.

四、Experimental Steps

1. Before the experiment, use the download cable to connect the PC network port to the PORT1 port of the S7-1200-CPU1214C host, then power on the experiment box and turn on the 24V power switch. The indicator lights of the host and 24V power supply are on, indicating that it is working normally and you can enter the next experiment.

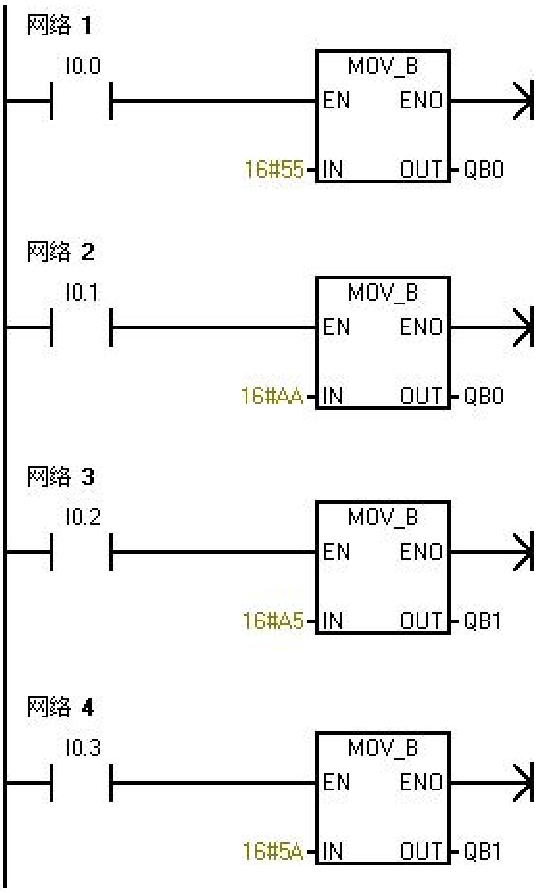
2. Enter the compiling and debugging environment, and enter the following exercise programs with instruction symbols or ladder diagrams.

3. Make corresponding connections according to the procedure. (For wiring, see "How to Use Input/Output Ports" in Chapter One)

4. Download the program and run it, and observe the results.

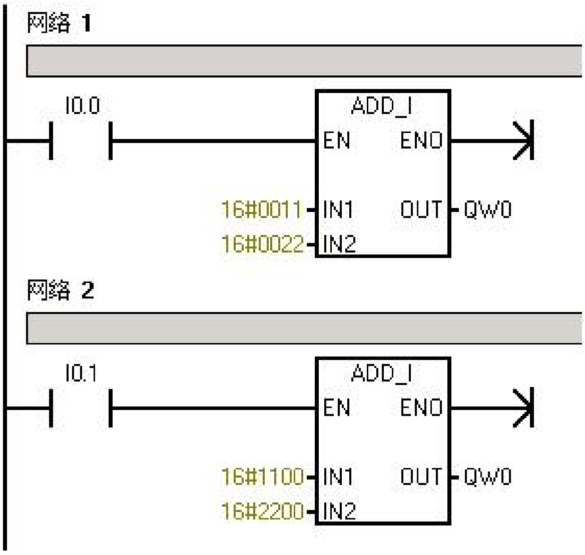
Exercise1、MOV instruction

```
Network 1
        LD      I0.0
        MOVB    16#55, QB0
Network 2
        LD      I0.1
        MOVB    16#AA, QB0
Network 3
        LD      I0.2
        MOVB    16#A5, QB1
Network 4
        LD      I0.3
        MOVB    16#5A, QB1
```



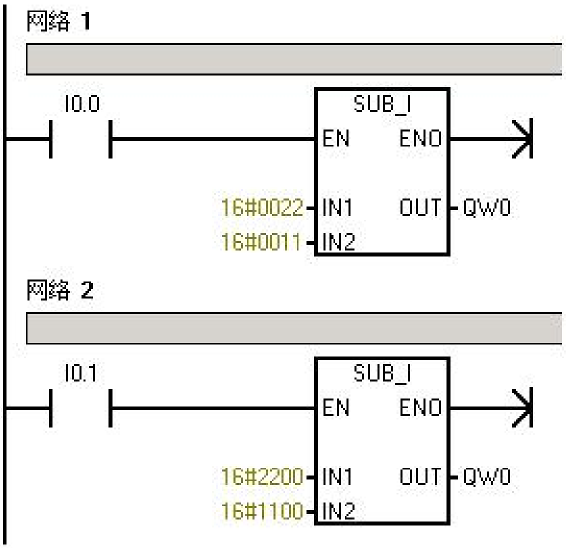
Exercise2、Addition instruction

```
Network 1
        LD      I0.0
        MOVW    16#0011, QW0
        +I      16#0022, QW0
Network 2
        LD      I0.1
        MOVW    16#1100, QW0
        +I      16#2200, QW0
```



Exercise3、Subtraction instruction

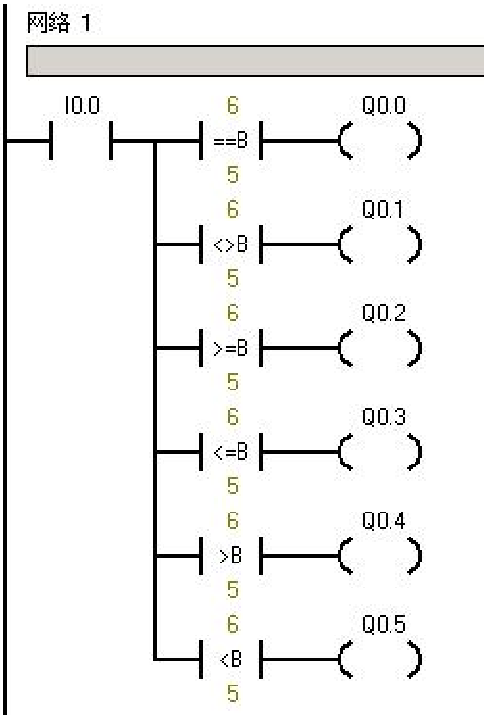
```
Network 1
        LD      I0.0
        MOVW    16#0022, QW0
        -I      16#0011, QW0
```



```
Network 2
        LD          I0.1
        MOVW        16#2200, QW0
        -I          16#1100, QW0
```

Exercise4、Comparison instructions

```
Network 1
        LD          I0.0
        LPS
        AB=         6, 5
        =           Q0.0
        LRD
        AB<>        6, 5
        =           Q0.1
        LRD
        AB>=        6, 5
        =           Q0.2
        LRD
        AB<=        6, 5
        =           Q0.3
        LRD
        AB>         6, 5
        =           Q0.4
        LPP
        AB<         6, 5
        =           Q0.5
```



Exercise5、Modify the instructions in the program of Exercise 1-Exercise 4 according to your own ideas, such as replacing byte instructions with word instructions or double-word instructions, or write another program yourself, and observe the running results. Familiar with the application of instructions.

Chapter Four PLC Control Hardware Experiment

Motor Automatic Control System

一、Experiment Purpose

Learn the sequential control and timing instruction of PLC controlled motor forward and reverse.

二、Experiment Material

1．PLC training platform

2．Reduced voltage starting and forward/reverse simulation panel

三、Experiment Content：

1．I/O（Input Output）distribution

Input		Output	
SB1（Forward）	I0.0	KM1	Q0.0
SB2（Stop）	I0.1	KM2	Q0.1
SB3（Reverse）	I0.2	KM3	Q0.2
		KM4	Q0.3

2．Input program (see reference program).

3．Operation：

Press the starting button to forward → KM1 on → KM2 off → KM3 on → motor reduced voltage starting forward → After 5 seconds → KM3 on → KM4 on → motor running normally

Press reverse button → KM1 off → KM2 on → KM3 on → after 5 seconds → KM3 off →KM4 on

Operate repeatedly.。

Water Tower Automatic Water Supply System

一、Experiment Purpose

Design a water tower automatic supply system by using PLC

二、Experiment Material

1．PLC training platform

2．Water tower water level automatic control simulation panel

三、Experiment Content

1．Wire according to the following table：

I/O distribution

Input		Output	
Water tower high water level boundaryS1	I0.1	Feed water solenoid valve Y	Q0.0
Water tower low water level boundaryS2	I0.2	Pumping motor M	Q0.1
Water tank high water level boundary S3	I0.3		
Water tank low water level boundaryS4	I0.4		

2．Input program (see reference program).

3．Operation：

When the water level is immersed in S1, S2, S3, S4 switches, (the state is ON, otherwise it is OFF). When the pool water level is lower than the low water level boundary (S4 is OFF), valve Y opens for water intake; when the pool water level is higher than the high water level boundary (S3 is ON), valve Y is closed; when the water tower water level is lower than the low water level boundary (S2 is OFF) and the water displacement current of the pool is higher than the low water level boundary, the pump M is opened, and if the water tower water level is higher than the high water level boundary, M is closed. If during the pumping process, the pool water level drops to the low pool water level boundary, M is also closed.

Material Mixing Control Experiment

一、Experiment Purpose

1. Master the usage of function instructions
2. Master the design of material mixing control program

二、Experiment Material

1、PC

2、PLC experimental platform

3、Material mixing control module

4、Communication cable

5、Wires

三、Experiment Content and Steps

1. Design requirements: Design a material mixing control program. The schematic diagram of the module is as follows:

Press the start button, first add material 1, Y1 lights up, stop adding material 1, Y1 off when the water level L3 is reached, delay, add material 2, Y2 lights, when the water level L2 (or water level 1), stop adding material 2, Y2 Extinct. After the feed is completed, the mixer starts to work, M is on, delay, the mixer stops working, M is off, and H starts to heat, when the temperature sensor reaches T, the heating is stopped, H is off, the discharge valve is opened, and Y4 is on. After the material is finished, repeat the above process.

2. Determine the input and output ports and write programs.

3. Compile the program, download it to the memory of the PLC host after correct, and run the program.

4. Debug the program until it meets the design requirements.

5. Wiring table

Input			Output		
Host	Experimental module	Note	Host	Experimental module	Note
I0.0	Start	Start	Q0.1	Y1	
I0.5	Stop	Stop	Q0.2	Y2	
			Q0.3	Y3	
I0.1	Water level L3		Q0.4	Y4	Nesting
I0.2	Water level L2		Q0.5	M	Blender
I0.3	Water level L1		Q0.6	H	heating
I0.4	Temperature		Q0.6	POS	Forward
1M		24V	Q0.7	NEG	Reverse
	COM	GND	1L、		24V

Television Transmitting Tower Simulation

一、Experiment Purpose

Form an electric wave launching simulation by using PLC

二、Experiment Material

1.PLC training platform

2.Television transmitting tower simulation panel

三、Experiment Content

1．Wire according to the following table：

I/O distribution

Input		Output	
Start	I0.0	L0	Q0.0
		L1	Q0.1
		L2	Q0.2
		L3	Q0.3

2．Input program (see page 89 for reference program)

3．Operation:

Press ‘Start’ button, the ‘L0’ light on.

0.4 Second later, ‘L0’ light off and ‘L1’ light on.

0.4 Second later, ‘L1’ light off and ‘L2’ light on.

0.4 Second later, ‘L2’ light off and ‘L3’ light on.

0.4 Second later, ‘L3’ light off.

0.4 Second later, ‘L1’ light on.

The circulation will continue for 2 minutes, all light off after 2 minutes, and then the circulation restarts.

Automatic Goods Delivery and Loading System Simulation

一、Experiment Purpose

Form an automatic goods delivery and loading system by using PLC.

二、Experiment Material

1．PLC training platform

2．Automatic goods delivery and loading system simulation panel

三、Experiment Content

1．Wire according to the following table：

I/O distribution

Input		Output	
Full hopper detection switch	S1—I0.1	Hopper switch	K2—Q0.0
Car detection switch	S2—I0.2	Red lamp	L1—Q0.1
Start button (to button on platform)	X0—I0.0	Green lamp	L2—Q0.2
		Motor	M1—Q0.3
		Motor	M2—Q0.4
		Motor	M3—Q0.5
		Motor	K1—Q0.6

2．Input program (see reference program).

3．Operation：

When the start button X0 is pressed, the red light L1 is off, the green light L2 is on, and K1 is turned on, until S1 is closed, and K1 is turned off, which means that the car is allowed to open for charging. The hopper K2, motors M1, M2, and M3 are all OFF. When the car arrives (indicated by S2 on), L1 is on, L2 is off, and M3 is running. Motor M2 runs after M3 is on for 2 seconds, and K2 is on for 2 seconds after M1 is on. When the material is full (indicated by S2 off), the hopper K2 is closed, the motor M1 is turned off after a delay of 2 seconds, M2 stops after M1 stops for 2 seconds, M3 stops after M2 stops for 2 seconds, L2 is on, L1 is off, indicating the car You can drive away.

4. Programming practice

According to the following two control requirements, the control programs of the automatic feeding and loading system without vehicle counting and with vehicle counting are respectively compiled, and commissioned and run on the computer.

4.1 The initial state is the same as the previous experiment

When the material is not full (S1 is OFF, the light is off, the hopper switch K2 is closed (OFF), the light is off, and the material is not discharged, and the feed switch K1 is opened (K1 is ON) to feed the material, otherwise it will not feed. When the car comes, M3 runs, Motor M2 runs after M3 runs for 2 seconds, M1 runs after M2 runs for 2 seconds, K2 turns on and discharges after 2 seconds, and when the material is full (indicated by S2 off), motor M1 turns off after a delay of 2 seconds. M2 turns on discharging after M1 stops for 2 seconds. When the material is full (indicated by S2 off), the motor M1 is turned off after a delay of 2 seconds, and M2 turns on discharging after M1 stops for 2 seconds. When the material is full (use S2 Off means), the motor M1 is turned off after a delay of 2 seconds, M2 stops after M1 stops for 2 seconds, and M3 stops after M2 stops for 2 seconds.

4.2 The control requirements are the same as 1, but the statistical function of the number of vehicles loaded per day is added.

Traffic Light Control System Simulation

一、 Experiment Purpose

Form a traffic light control system by using PLC.

二、Experiment Material

1．PLC training platform

2．Traffic light control system simulation panel

三、Experiment Content

1．Wire according to the following table：

I/O distribution

Input		Output	
S1(on as automatic, off as manual)	I0.0	E&W red	Q0.0
S2 (manual, E&W green, S&N red)	I0.1	E&W yellow	Q0.1
S3(manual, S&N green, E&W red)	I0.2	E&W green	Q0.2
S4 (start and stop button)	I0.3	S&N red	Q0.3
		S&N yellow	Q0.4
		S&N green	Q0.5

Note: E=east, W=west, N=north, S=south

2．Input program (see reference program).

3．Operation：

The automatic switch S1 is closed and the start-stop switch S4 is disconnected. The green light is on for 4 seconds, flashes for 2 seconds, and the yellow light is on for 2 seconds, and the red light is on for 8 seconds. Correspondingly, when the east and west green lights and yellow lights are on, the north and south red lights are on for 8 seconds, then the green light is on for 4 seconds and flashes for 2 seconds, and after the yellow light is on for 2 seconds, the red light is on for 8 seconds.

Manual: Switch S1 off, press S2 to disconnect S3, east-west green, north-south red, press S3 to disconnect S2, north-south green, east-west red.

Automatic Punching Machine Control System Simulation

一、Experiment Purpose

Form an automatic punching machine control system by using PLC.

二、Experiment Material

1．PLC training platform

2．Automatic punching machine control system simulation panel

三、Experiment Content

1．Wire according to the following table：

I/O distribution

Input			Output		
Start	S0	I0.0	Front conveyor motor	M0	Q0.0
Detection	S1	I0.1	Electromagnet switch	KM	Q0.1
Downtime	S2	I0.2	Punching air-hammer	Y2	Q0.2
Workbench travel switch	S3	I0.3	Rear conveyor motor	M1	Q0.3
Workbench travel switch	S4	I0.4	Running motor forward	Y4	Q0.4
Workbench travel switch	S5	I0.5	Running motor reverse	Y5	Q0.5

2．Input program (see reference program).

3．Operation

3.1 Press the start button S0, the conveyor (front) (rear) motor runs, and the indicator light is ON.

3.2 When S1 detects the arrival of an object, the electromagnet breaker KM pulls in, the object is pulled up, and M0 stops at the same time.

3.3 When the magnet attracts the object, one second later, the motor on the electromagnet rotates forward, that is, the Y4 light is on, and the object is sent to the worktable.

3.4 When the motor on the electromagnet reaches the workbench, the workbench travel switch is depressed (S3 is ON) and the electromagnet loses electricity and the object is sent to the workbench. At the same time, the motor reverses, that is, the Y5 light is on, and it returns to the initial point. Lower the limit switch S5 and the motor stops.

3.5 After 2 seconds, the punching hammer switch is closed, that is, the Y2 light is on, and the punching work starts. After 20 seconds, it stops working and the Y2 light goes out.

3.6 The motor is rotating back to the table to absorb the body, and the electromagnet motor moves forward at the same time, sending the processed work to the belt M1.

3.7 After reaching the end stroke switch S4 voltage, the electromagnet is de-energized and put down to work, and the motor reverses at the same time, the electromagnet returns to the initial position, press the stroke switch (S5), and the conveyor motor runs again.

Automatic Vending Machine System Simulation

一、Experiment Purpose

Form an automatic vending machine system by using PLC.

二、Experiment Material

1．PLC training platform

2．Automatic vending machine system simulation panel

三、Experiment Content

1．Wire according to the following table

I/O distribution

Input (reset) SES		I0.5	Output		
$1 slot	S0	I0.0	Coffee outlet	Y0	Q0.0
$5 slot	S1	I0.1	Soda outlet	Y1	Q0.1
$10 slot	S2	I0.2	Coffee button indicator	L0	Q0.2
Coffee button	S3	I0.3	Soda button indicator	L1	Q0.3
Soda button	S4	I0.4	Change indicator	L7	Q0.7

2．Input program (see reference program).

3．Operation

3.1 The vending machine can input 1 yuan (press S0), 5 yuan (press S1), and 10 yuan coins (press S2).

3.2 When the total value of the inserted coins is equal to or more than RMB 12, the soda button indicator L1 lights up, indicating that you can buy soda; when the total value of the inserted coins exceeds 15 yuan, the soda and coffee indicator lights are both on, indicating that the soda and You can buy coffee.

3.3 When the indicator light of the soda button switch is on, press the soda button and the soda drain will automatically stop after 7 seconds. When the soda is discharged, the soda outlet Y1 indicator flashes.

3.4 When the coffee button indicator L0 is on, the action is the same as above.

3.5 If the total value of the inserted coins exceeds the amount of money required by the button (12 yuan for soft drinks, 15 yuan for coffee), the money-finding indicator L7 is on after shopping.

Automatic Molding System Simulation

一、 Experiment Purpose

Form an automatic molding machine system by using PLC.

二、Experiment Material

1．PLC training platform

2．Automatic molding machine system simulation panel

三、Experiment Content

1．Wire according to the following table：

I/O distribution

	Input		Output	
Start	S0	I0.0	Solenoid valveY1	Q0.1
	S1	I0.1	Solenoid valveY2	Q0.2
	S2	I0.2	Solenoid valveY3	Q0.3
	S3	I0.3	Solenoid valve Y4	Q0.4
	S4	I0.4		
	S5	I0.5		
	S6	I0.6		

2．Input program (see reference program).

3．Operation

3.1 Initial state, when the raw material is put into the forming machine, each cylinder is in the initial state: Y1=Y2=Y4=OFF, Y3=ON, S1=S3=S5=OFF, S2=S4=S6=ON.

3.2 Press the start key S0, Y2=ON, and the piston of the upper cylinder moves downwards to make S4=OFF.

3.3 When the piston of the cylinder drops to the end point, S3=ON, at this time, start the left cylinder, the piston of A moves to the right, and the piston of the right cylinder C moves to the left. When Y1=Y4=ON, Y3=OFF, Make S2=S6=OFF.

3.4 When the piston of cylinder A reaches the end point, S1=ON, and the piston of cylinder C reaches the end point S5=ON, the raw material has been formed, and each cylinder begins to return to its original position. First, the cylinders A and C return, Y1=Y4= OFF, Y3=ON, make S1=S5=OFF.

3.5 When the A and C cylinders return to the initial position, S2=S6=ON, the B cylinder returns, Y2=OFF, and S3=OFF.

3.6 When the oil cylinder returns to the initial state and S4=ON, the system returns to the initial state to take out the finished product. After feeding the raw material, press the start button to restart and start the processing of the next work piece.。

4.Programming exercise

1. Control requirements (fully automatic control)

(1) Initial state: When the raw material is put into the forming machine, each cylinder is in the initial state: Y1=Y2=Y4=OFF, Y3=ON, S1=S3=S5=OFF, S2=S4=S6=ON.

(2) Start and run

When the start button is pressed, the system action requirements are as follows:

(A) Y2=ON. Piston B of the upper cylinder moves downwards to make S4=OFF.

(B) When the piston of the cylinder drops to the end point, S3=ON, at this time, start the left cylinder, the piston of A moves to the right, and the piston of the right cylinder C moves to the right, Y1=Y4=ON, Y3=OFF, Make S2=S6=OFF.

(C) When the piston of cylinder A reaches the end point, S1=ON, and the piston of cylinder C reaches the end point S5=ON, the raw material has been formed, and the cylinders begin to return to their original positions. First, the A and C cylinders return, Y1=Y4=OFF, Y3=ON, so that S1=S5=OFF.

(D) When the A and C cylinders return to the initial position, S2=S6=ON, the B cylinder returns, Y2=OFF, and S3=OFF.

(E) When the B cylinder returns to the initial state and S4=ON, the system returns to the initial state. After a delay of 10 seconds, the finished product is taken out, and the raw material is put in, and then the processing of the next work piece is started.

2. Control requirements (full automatic control with counting):

(1) Initial state

When the raw materials are put into the forming machine, each oil cylinder is in the initial state Y1=Y2=Y4=OFF, Y3=ON.

(2) Start and run:

Press the start button QA, the system action requirements are as follows:

(A) Y2=ON The piston B of the upper cylinder moves downwards, so that S4=OFF.

(B) When the cylinder piston descends to the end point, S3=ON, at this time, start the left cylinder, the piston of A moves to the right, the piston of the right cylinder C moves to the left, Y1=Y4=ON, Y3=OFF, so that S2 =S6=OFF.

(C) When the piston of cylinder A runs to the end point S1=ON, and the piston of cylinder C also reaches the end point S5=ON, the raw material has been formed, and the cylinders begin to return to their original positions. First, the A and C cylinders return, Y1=Y4=OFF. Y3=ON, make S1=OFF, make S3=OFF.

(D) When the A and C cylinders return to the initial position, S2=S6=ON, the A cylinder returns, Y2=OFF, and S3=OFF.

(E) When the oil returns to the initial state S4=ON, the system returns to the initial state, delay 10 seconds, and take out the finished product.

(F) At this time, count the number of finished products, and then start the processing of the next workpiece after putting in the raw materials. Stop operation After pressing a stop button, after the current workpiece is processed, it returns to the initial state and stops running.

Elevator Automatic Control System Simulation

一、Experiment Purpose

Form an elevator automatic control system by using PLC.

二、Experiment Material

1．PLC training platform

2．Elevator automatic control system simulation panel.

三、Experiment Content

1．Wire according to the following table：

I/O distribution

	Input		Output	
	Call buttonPB4	I0.0	Up△	Q0.5
	Call buttonPB3	I0.1	Down▽	Q0.0
Outbound switch	Call buttonPB2	I0.2	First layer indicator	Q0.1
	Call buttonPB1	I0.3	Second layer indicator	Q0.2
	Leveling signal LS4	I0.4	Three-layer indicator	Q0.3
	Leveling signal LS3	I0.5	Four-layer indicator	Q0.4
Delivery switch in the floor	Leveling signal LS2	I0.6		
	Leveling signal LS1	I0.7		

2．Input program (see reference program).

3．Operation

3.1 When the car stops on the first or second or third floor, press the PB4 button to call the elevator, the car will rise, and the rising indicator light will be on. If it reaches the fourth floor (press LS4), the Y4 light will be on to indicate that it has reached every fourth floor. Level, the rising indicator light goes out.

3.2 When the car stops on the fourth or third floor, or the second floor, press the PB1 button to call the elevator, and the descending indicator light will be on. When the car descends to the first floor, press the LS1 button to indicate that it has reached the first floor and descend The indicator light is off and the Y1 indicator light is on, indicating that it has reached the first floor.

3.3 When the car stops on the first floor, if you press the PB2 button to call the elevator, the car rises to LS2 stop, if you press the PB3 button to call the elevator, the car rises to LS3 stop.

3.4 When the car stops on the fourth floor, if you press the PB3 button to call the elevator, the car will descend to LS3 stop, if you press the PB2 button to call the elevator, the car will descend to LS2 stop.

3.5 When the car stops on the first floor and there are people calling for the PB2, PB3 and PB4 buttons, the car will rise to LS2 and then continue to rise to LS3 after a 4-second pause, and continue to rise to LS4 after a 4-second pause.

3.6 When the car stops on the fourth floor and the PB1, PB2, PB3 buttons all call the elevator, the car drops to LS3 and pauses for four seconds before continuing to descend to LS2. After LS2 pauses for four seconds, continue to descend until LS stops.

3.7 The elevator stops running if the elevator car needs to run for more than 12 seconds in the stairwell.

3.8 When the car is ascending (or descending), any button call for descending (or ascending) in the opposite direction is invalid. When the floor indicator light is on, it means there is a signal request for that floor, and the light off means that the floor request signal is eliminated.

△Bright means the elevator is rising

▽Bright means the elevator descends

3.9 The outbound switch PB3 and PB2 are only used in the following procedures. If you want to expand, you must customize it.

4.Programming exercise

After the elevator starts, it is first on the first floor, there are

When calling, the elevator rises. The ascending process only responds to the internal call signal and the rising external call signal greater than or equal to the current floor, and remembers other signals, and stops at the internal call floor and the rising external call floor. And eliminate the internal call signal and rising signal of the floor. After that, if there is no call signal (including memory signal) on other floors, stop at this floor, and continue to run if there is. After reaching the fourth floor, the elevator descends when there are internal and external call signals from other floors. The descending process only responds to internal calls less than or equal to the current floor and descending external call signals, memorizes other signals and stops on internal call floors and descending external call floors. After that, if there is no internal/external call signal (including memory signal), stop at this floor and continue operation if there is one. After reaching the first floor, the above cycle starts again. Set the elevator to rise or fall at a constant speed, and it takes 3 seconds to rise or fall one floor. And ensure that the stay time of the electric building is not less than 2 seconds.

The floor indicator light indicates that the elevator door of the corresponding floor is open.

△Bright means the elevator is up. ▽Bright means the elevator descends.

2. Compile the four-story elevator control program according to the following control requirements, debug the program on the machine and run it.

a. After the elevator starts, the car is on the first floor. If there is a call signal for the first floor, the door will be opened.

b. It can memorize and respond to other signals during operation, and internal selection has priority. When the call signal is greater than the current floor, it goes up, and when the call signal is less than the current floor, it goes down.

c. After reaching the calling floor, open the door after leveling to clear the memory. The door can be closed with delay when the elevator is called on the current floor.

d. The door opening time can be selected for multi-layer call building. If the call signal comes from both sides of the current floor and the distance is equal, the original movement direction will be memorized and maintained. After reaching the call floor, it will run in reverse and respond to the call.

e. If the call signal comes from both sides of the current floor, and the distance is not equal, then remember and select the shortest floor to respond first.

f. If there is no call signal, the car stops at the current floor.

g. When the elevator is not in use, return to the first floor, cut off the power after opening the door, and restart when it is used again.

h. From the start of the elevator, the seven-segment code displays the floor until the end.

Automatic Washing Machine Control System Simulation

一、 Experiment Purpose

Form an automatic washing machine control system by using PLC.

二、Experiment Material

1．PLC training platform

2．Automatic washing machine control system simulation panel

三、Experiment Content

1．Wire according to the following table：

I/O(Input and output) distribution

Input	Output
Upper limitS0—I0.0	Left-rotation — Q0.0
Lower limitS1—I0.1	Right-rotation — Q0.1
StartS2—I0.2	Dehydration — Q0.2
StopS3—I0.3	Buzzer — Q0.3
Manual drainageS4—I0.4	Inlet valve — Q0.4
	Drain valve — Q0.5

2．Input program (see reference program).

3．Operation：

3.1 Press the start switch S2, the water inlet valve light is on, indicating that the washing tub has started to fill with water;

3.2 When the water level reaches the upper limit, the upper limit switch S0 is pressed, and the water inlet valve light is off at the same time, indicating that the water is full and no more water is needed.

3.3 Then the liquid wheel starts to rotate left and right, turn left for 2 seconds, that is, the turning indicator light will be on for 2 seconds. After a lapse of 0.5 seconds, the right turn light will be on for 2 seconds after the turn for 2 seconds, and the left turn light will be on for 2 seconds after 0.5 seconds, and the cycle is repeated.

3.4 After cycling for 3 minutes, the pulsator stops rotating, that is, the left and right turn lights are off, and the drain valve lights are on, indicating that the washing machine starts to drain water.

3.5 When the water level starts to drop, press the limit switch S1 to indicate that the water level reaches the lower limit, and then stop draining, that is, the drain valve is closed and the light is off.

3.6 The dehydration starts, that is, the dehydration light is red and lights up for 1 minute.

3.7 After one minute, the buzzer sounds, indicating that the entire washing process is complete.

3.8 Press the stop button S3, the washing machine stops working.

3.9 Press the manual drain switch S4, the washing tub starts to drain and the drain valve light is on.

Plating Control System Simulation

一、Experiment Purpose

Form a plating control system by using PLC.

二、Experiment Material

1．PLC training platform

2．Plating control system simulation panel

三、Experiment Content

1．Wire according to the following table：

I/O(Input and output) distribution

Input		Output		
Start	S1—I0.1	Crane advance	—	Q0.1
Home position limit switch	SQ4（S2）—I0.2	Crane retreat	—	Q0.2
Clear water bath limit switch	SQ3（S3）—I0.3	Crane up	—	Q0.3
Recycling wave trough limit switch	SQ2（S4）—I0.4	Crane down	—	Q0.4
Plating tank limit switch	SQ1（S5）—I0.5			

Lower limit switch	SQ6（S6）—I0.6
Upper limit switch	SQ5（S7）—I0.7
Inching advance	S8 — I0.10
Inching retreat	S9 — I0.12

2．Input program (see reference program)

3．Operation：

The electroplating production line uses a special crane, and the crane frame is equipped with a lifting hook. Both the crane and the hook are driven by a motor. The advance and retreat of the crane and the lifting of the hook are positioned by the corresponding limit switch SQ. The process is as follows:

(1) Press the start button S1, the hook rises, lifts the workpiece, and stops when it hits the upper limit switch SQ5, and then goes to the next process.

(2) Drive forward and stop until SQ1 is depressed, just above the plating tank.

(3) The hook descends and stops when SQ6 is depressed, and the workpiece penetrates into the plating wave for 280 seconds.

(4) At the end of electroplating, the hook lifts the work piece and stops when the SQ5 is pressed down, and stops at the top of the plating tank for 28 seconds to let the wave drop back into the tank.

(5) Drive back and stop when SQ2 is depressed, just above the echo groove.

(6) The hook descends, and after hitting SQ6, the workpiece is put back into the recovery groove K, and the time is 30 seconds.

(7) The hook rises and stops at SQ5 for 15 seconds.

(8) Drive back and stop at SQ3 above the clear water tank for 15 seconds.

(9) After the hook descends and hits SQ6, the workpiece is placed in the clean water tank and cleaned for 30 seconds.

(10) The hook rises to SQ5 and stops for 15 seconds.

(11) Drive back and stop above the original position at SQ4.

(12) The hook descends and returns to the original position at SQ6, and the plated workpiece is taken off.

(13) Press S8, S9, the motor can advance and retreat.

At this point, the entire electroplating production has completed a work cycle. Then connect the start button to start the second working cycle.

4.Debug and run the program

Stepping instruction is suggested (NSTP, SSTP, CSTP, STPE).

PLC Control Of Driving Stepping Motor

一、Experiment Purpose

1. Master the usage of PLC function instructions
2. Master the method of controlling stepper motor with PLC

二、Experiment Material

1.PC

2.PLC experimental platform

3. Stepper motor control module

4.Communication cable

5.Wires

三、Experiment Content and Step

1. Design requirements

The working mode of the stepping motor in the control module is four-phase eight-beat. The four-phase coils of the motor are represented by A, B, C, and D respectively, and the common terminal is grounded.

When the motor is rotating forward, its working mode is as follows: A→AB→B→BC→C→CD→D→DA→A.

When the motor reverses, its working mode is as follows: A→-AD→D→DC→C→CB→B→BA→A.

The design program requires the ability to control the forward and reverse rotation of the stepper motor, and to control its speed.

2. Determine the input and output ports and write programs.

3. Compile the program, download it to the memory of the PLC host after correct, and run the program.

4. Debug the program until it meets the design requirements.

5. Wiring table

Input			Output		
Host	Experimental module	Note	Host	Experimental module	Note
I0.0	Start	Start	Q0.0	XA	Stepper motor phase A
I0.1	Stop	Stop	Q0.1	XB	Stepper motor phase B
I0.2	Forward	Forward	Q0.2	XC	Stepper motor phase C
I0.3	Reverse	Reverse	Q0.3	XD	Stepper motor phase D
I0.4	Fast	Fast			
I0.5	Slow	Slow			
1M	24V				
	COM1←→GND	Switch common			
			1L、2L		24V
				COM2←→GND	Motor common

Tool Magazine Shortcut Direction Selection Control

一、Experiment Purpose

1. Master the usage of data processing instructions in function instructions

2. Master the program design of the shortcut direction selection control of the tool magazine of the machining center

二、Experiment Material

1、PC

2、PLC experimental platform

3、Tool magazine control module

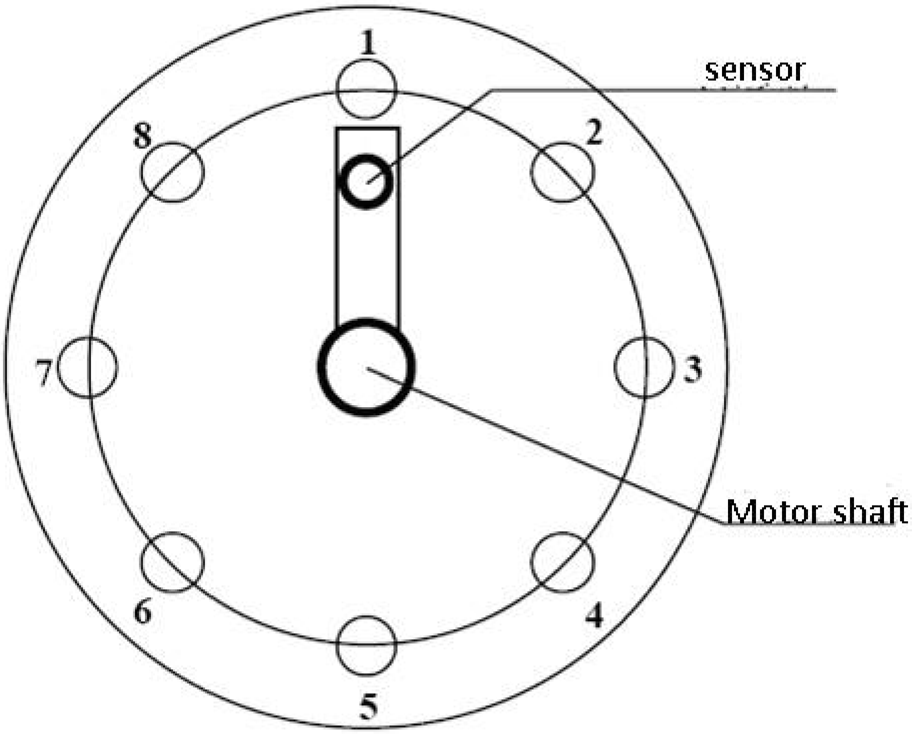
4、Communication cable

5、Wires

三、Experiment Content and Step

1. Design requirements

As shown in the figure, it is the tool magazine back-table simulation device of the machining center. There are 8 tools on it, and each tool has a corresponding tool number address, namely 1, 2, ..., 8. When the tool magazine rotates, the sensor will test its tool number position.



After starting, if the sensor does not find any knife, the motor will automatically rotate clockwise until the sensor detects a certain knife and the motor stops. The code switch is used to select the tool number. After selecting the tool number, press the "Send tool number" button again to send the selected tool number into the program. The program will compare and calculate the selected tool number with the current tool number, and then instruct the motor to rotate in the direction closest to the current tool number (ie the shortest direction, forward or reverse), and then turn to the selected tool number position , The motor stops rotating.

2. Determine the input and output ports and write programs.

3. Compile the program, download it to the memory of the PLC host after correct, and run the program.

4. Debug the program until it meets the design requirements.

5. Reference program (s7_226_knife.mwp) and wiring table

Input			Output		
Host	Experimental module	Note	Host	Experimental module	Note
I0.0	Start S1	Start button	Q0.0	DJB	Motor forward (clockwise)

I0.1	Give knife number S2	Give knife number button	Q0.1	DJA	Motor reverse (counterclockwise)
I0.2	1	BCD code 1			
I0.3	2	BCD code 2			
I0.4	4	BCD code 4			
I0.5	8	BCD code 8			
	COMB	GND			
I1.0	1	No. 1 knife position			
I1.1	2	No. 2 knife position			
I1.2	3	No. 3 knife position	1L		24V
I1.3	4	No. 4 knife position	2L		24V
I1.4	5	No. 5 knife position	3L		24V
I1.5	6	No. 6 knife position			
I1.6	7	No. 7 knife position			
I1.7	8	No. 8 knife position		24V	24V
				GND	GND
1M		24V			
2M		24V			

Automatic Mail Sorting System Simulation

一、Experiment Purpose

1. Master the usage of function instructions

2. Master the design of mail sorting control program

二、Experiment Material

1、PC

2、PLC experimental platform

3、Automatic mail sorting module

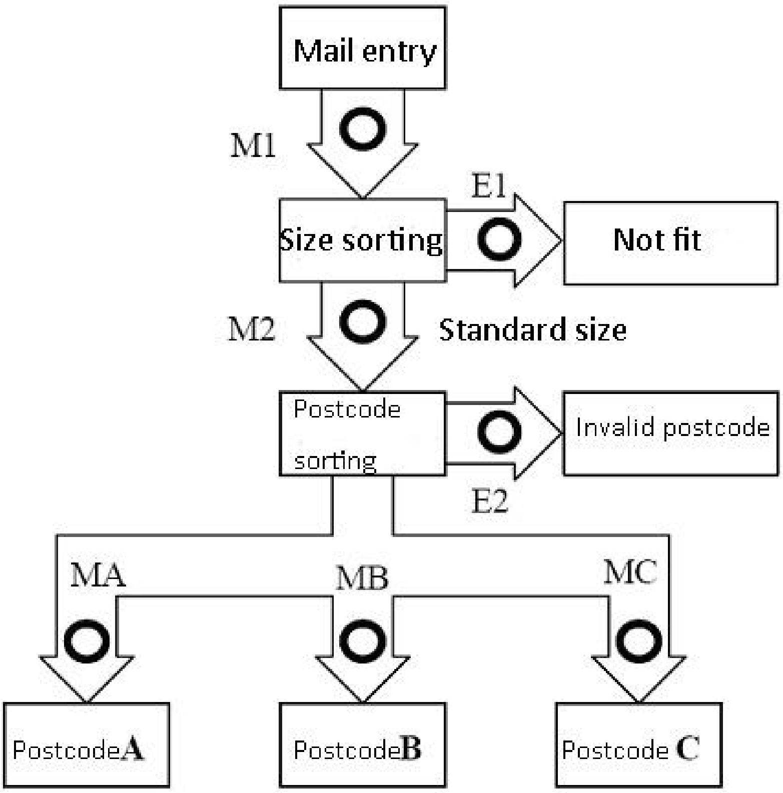
4、Communication cable

5、Wires

三、Experiment Content and Step

1. Design requirements

Design a mail sorting control program. The module diagram is as follows：



Schematic diagram of mail sorting system

After pressing the start button, the mail starts to enter the assembly line, the indicator light M1 lights up, after a delay of 2 seconds, it enters the appearance detection area, the indicator light M2 lights up, enters the postal code detection area after a delay of 2 seconds, and enters the postal code division after a delay of 2 seconds In the gear area, the indicators MA, MB, and MC in this area flash and light up.

Out-of-shape (E1) and invalid postal code (E2) buttons: used to simulate non-standard size signals and unrecognizable signals. When the button is pressed, it means that the signal is valid and the pipeline is suspended and waiting for processing. When the button is released, it means the signal is invalid and the pipeline continues to run.

The three buttons of Zip Code A, Zip Code B, and Zip Code C are designated to process the signals of the corresponding zip code. When a button is pressed, only the mails with the zip code corresponding to the button will flash in the zip code bin area.

2. Determine the input and output ports and write programs.

3. Compile the program, download it to the memory of the PLC host after correct, and run the program.

4. Debug the program until it meets the design requirements.

5. Reference program (s7_226_mail.mwp) and wiring table

Input			Output		
Host	Experimental module	Note	Host	Experimental module	Note
I0.0	Start	Start	Q0.1	M1	Size sorting
I0.1	Stop	Stop	Q0.2	M2	Postcode sorting
I0.2	K1	Postcode A	Q0.3	MA	Postcode A
I0.3	K2	Postcode B	Q0.4	MB	Postcode B
I0.4	K3	Postcode C	Q0.5	MC	Postcode C
I0.5	E1	Not fit	Q0.6	E1	Not fit
I0.6	E2	Invalid postcode	Q0.7	E2	Invalid postcode
1M		24V			
	COM	GND	1L		24V
	KCOM	GND	2L		24V

Note: K1, K2, K3 use the switches in the "Basic Command Module".

Four-stage Conveyor Belt Control

一、Experiment Purpose

1. Master the usage of function instructions

2. Master the design of the four-level conveyor belt control program

二、Experiment Material

1、PC

2、PLC experimental platform

3、Four-stage conveyor belt control module

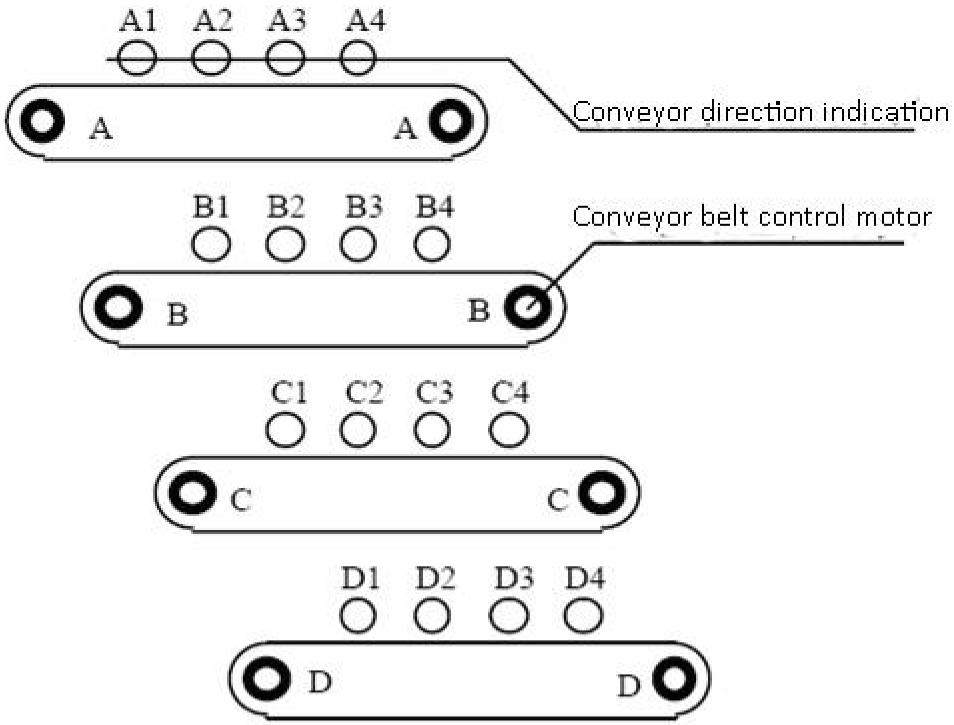
4、Communication cable

5、Wires

三、Experiment Content and Step

1. Design requirements

Design a four-level conveyor belt control program. The module diagram is as follows：



Schematic diagram of four-stage conveyor belt

The workflow of the experimental program:

From top to bottom in the control module are one, two, three, four (A, B, C, D) grade conveyor belts.

When the start button is pressed, the fourth-level conveyor belt starts to work, the red light is on, and the green light starts to circulate from left to right; after a delay of 5 seconds, the third-level conveyor belt starts to work; after a delay of 5 seconds, the second-level conveyor belt starts Work; after a delay of 5 seconds, the first level conveyor belt starts to work

Switch fault A, fault B, fault C, and fault D are respectively used to simulate the fault signals of the first, second, third, and fourth-level conveyor belts.

When fault A closes and the first-level conveyor belt fails, the first-level conveyor belt stops immediately. After a delay of 5 seconds, the second-level conveyor belt stops. After a delay of 5 seconds, the third-level conveyor belt stops. After a delay of 5 seconds, the fourth-level conveyor belt stops.

When fault B is closed, the second conveyor belt fails, the first and second conveyor belts stop immediately, after a delay of 5 seconds, the third conveyor belt stops, and after a delay of 5 seconds, the fourth conveyor belt stops.

When the fault C K3 is closed, the third-level conveyor belt fails, the first-level, second-level, and third-level conveyor belts stop immediately. After a delay of 5 seconds, the fourth-level conveyor belt stops.

When the fault D is closed, the fourth-level conveyor belt fails, and the first-level, second-level, third-level, and fourth-level conveyor belts stop immediately.

When the fault is eliminated, the switch is turned off and the four-stage conveyor belt starts to work at the same time.

When the stop button is pressed, the first-level conveyor belt stops immediately. After a delay of 5 seconds, the second-level conveyor belt stops. After a delay of 5 seconds, the third-level conveyor belt stops. After a delay of 5 seconds, the fourth-level conveyor belt stops.

2. Determine the input and output ports and write programs.
3. Compile the program, download it to the memory of the PLC host after correct, and run the program.
4. Debug the program until it meets the design requirements.
5. Reference program (s7_226_transfer.mwp) and wiring table

Input			Output		
Host	Experimental module	Note	Host	Experimental module	Note
I0.0	Start	Start	Q0.1	XL1	Belt conveying direction indication
10.1	Stop	Stop	Q0.2	XL2	
			Q0.3	XL3	
			Q0.4	XL4	
I0.4	S1	Primary conveyor belt failure simulation			
I0.5	S2	Secondary conveyor belt failure simulation	Q1.1	A	Level 1 conveyor motor
I0.6	S3	Three-level conveyor belt failure simulation	Q1.2	B	Level 2 conveyor motor
I0.7	S4	Four-stage conveyor belt failure simulation	Q1.3	C	Level 3 conveyor motor
			Q1.4	D	Level 4 conveyor motor
	COM	GND			
1M		24V	1L		GND
			2L		GND
			3L		24V

Assembly line control

一、Experiment Purpose

1. Master the application of shift register in control system and its programming method.

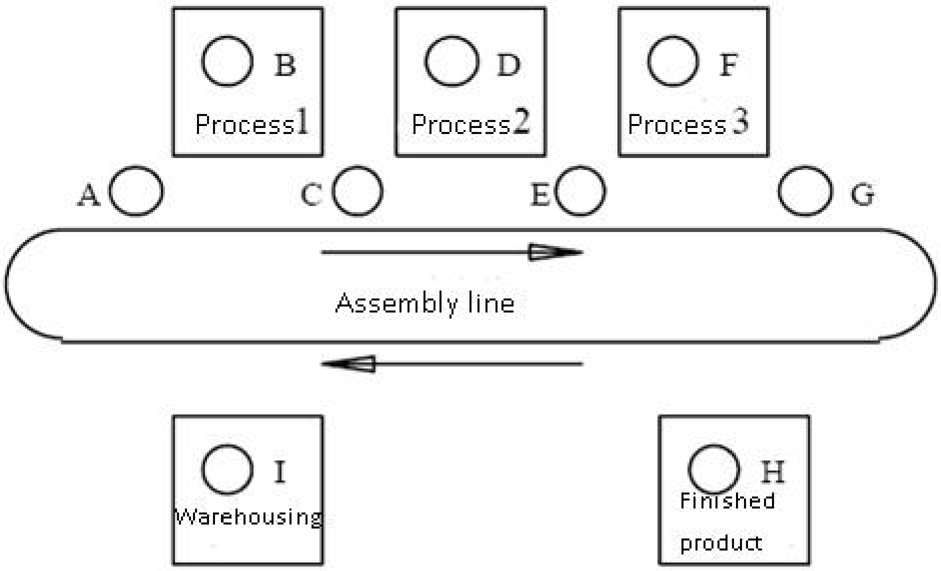
二、Experiment Material

1、PC
2、PLC experimental platform
3、Assembly line control module
4、Communication cable
5、Wires

三、Experiment Content and Step

1. Design requirements

There are four processes in the assembly line. A, C, E, G indicate the direction of the assembly line, and four processes of B (process 1), D (process 2), F (process 3), and H (finished product) complete four assembly tasks. Finally it is put into storage via I.



Schematic diagram of assembly line

2. Procedure flow:

Flowing water lamps A, C, E, G light up once when they are shifted, and B (process 1) light up with a delay of 3 seconds;

Flowing water lamp A, C, E, G light up once shifted, D (process 2) light up, delay 3 seconds;

Flowing water lamps A, C, E, G light up once shifted, and F (process 3) light up with a delay of 3 seconds;

Flowing water lights A, C, E, G light up once shifted, H (finished product) lights up, and the delay is 3 seconds;

Flowing water lamps A, C, E, G light up once when they are shifted, and I (in storage) light up with a delay of 3 seconds;

Repeat the cycle.

3. Determine the input and output addresses and write programs

4. Compile the program, download it to the memory of the PLC host after correct, and run the program.

5. Please design this experimental program by yourself according to the "program flow", and debug the program until it meets the design requirements.

Design steps: 1. Determine the input and output points.

6. Design procedure。

Input			Output		
Host	Experimental module	Note	Host	Experimental module	Note
I0.0	mobile		Q0.1	A	
I0.1	start		Q0.2	B	
I0.2	stop		Q0.3	C	
			Q0.4	D	
			Q0.5	E	
			Q0.6	F	

			Q0.7	G	
			Q1.0	H	
			Q1.1	I	

PLC Automatic Control of Manipulator

一、Experiment Purpose

1. Master the usage of PLC function instructions

2. Master the design of the stepping control program of the manipulator with PLC

二、Experiment Material

1、PC

2、PLC experimental platform

3、Manipulator control module

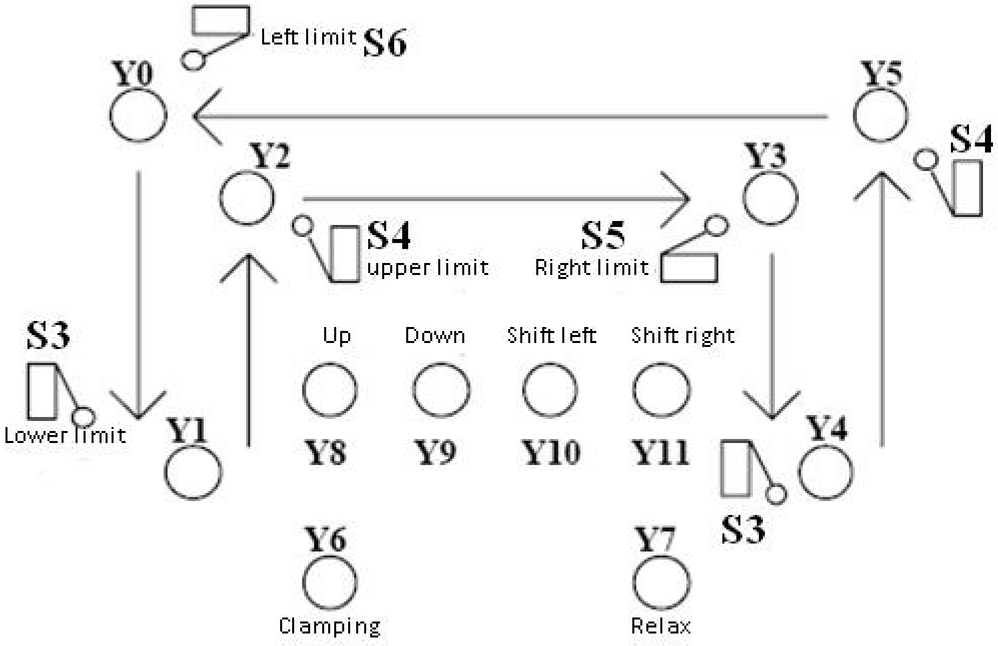
4、Communication cable

5、Wires

三、Experiment Content and Step

1. Design requirements

The working diagram of the manipulator is as follows：



Manipulator control diagram

The control process is as follows：

At the beginning, when the manipulator is in the original position and in a relaxed state, Y0 is on and Y7 is on. After pressing the start button, the manipulator starts to move sequentially:

1. Manipulator down: After a delay of 1 second, Y0 is off and Y9 is on. When the lower limit is reached, press S3, then Y9 is off and Y1 is on.
2. Manipulator clamping: After a delay of 3 seconds, Y7 is off and Y6 is on.
3. Manipulator up: Y1 is off and Y8 is on. After reaching the upper limit, press S4, then Y8 is off and Y2 is on.
4. The manipulator moves to the right: Y2 is off and Y10 is on. After reaching the right limit, press S5, then Y10 is off and Y3 is on.
5. Manipulator down: Y3 is off and Y9 is on. After reaching the lower limit, press S3, then Y9 is off and Y4 is on.
6. The manipulator relaxes: after a delay of 3 seconds, Y6 is off and Y7 is on.
7. Manipulator up: Y4 is off and Y8 is on. After reaching the upper limit, press S4, then Y8 is off and Y5 is on.
8. The manipulator moves to the left: Y5 is off and Y11 is on. After reaching the left limit, press S6, then Y11 is off and Y0 is on, returning to the original position.。

2. Determine the input and output ports and write programs.
3. Compile the program, download it to the memory of the PLC host after correct, and run the program.
4. Debug the program until it meets the design requirements.

5. Reference program (s7_226_machand.mwp) and wiring table

Input			Output		
Host	Experimental module	Note	Host	Experimental module	Note
I0.0	start	start	Q0.0	Y0	origin
I0.1	stop	stop	Q0.1	Y1	Relax and down in place
			Q0.2	Y2	Clamping rises in place
I0.4	upper limit	Upper limit switch	Q0.3	Y3	Clamping moves right into place
I0.5	Lower limit	Lower limit switch	Q0.4	Y4	Clamp down in place
I0.6	Left limit	Left limit switch	Q0.5	Y5	Relax and rise in place
I0.7	Right limit	Right limit switch	Q0.6	Y6	Clamping instructions
			Q0.7	Y7	Relaxation instructions
			Q1.0	Y8	Rising
			Q1.1	Y9	Downing
			Q1.2	Y10	Move right
			Q1.3	Y11	Move left
	COM	GND	1L		24V

1M		24V	2L		24V
			3L		24V

Note: The button switch is used during the experiment, and the button switch is not used, so the button switch must be pulled to the right (disconnected, no effect).

LED Digital Tube Display Control

一、Experiment Purpose

1. Master the application of shift register instructions

2. Master the use of PLC to control the digital tube display

二、Experiment Material

1、PC

2、PLC experimental platform

3、LED digital tube control module

4、Communication cable

5、Wires

三、Experiment Content and Step

1. Design requirements

Design a digital tube cycle display program. Display numbers 0-9. The digital tube is a common cathode type. A, B, C, D, E, F, G, Dp are digital tube segment codes, COM is the digital tube common end (bit code), when the segment code is input high level, and the bit code is input low level, the corresponding segment point bright.

2. Determine the input and output ports and write programs.

3. Compile the program, download it to the memory of the PLC host after correct, and run the program.

4. Debug the program until it meets the design requirements.

5. Reference program (s7_226_led8.mwp) and wiring table

Input			Output		
Host	Experimental module	Note	Host	Experimental module	Note
I0.0	start	start	Q0.0	A	Segment A
I0.1	stop	stop	Q0.1	B	Segment B
I0.2	K1	0-9 cycle	Q0.2	C	Segment C
I0.3	K2	Segment loop	Q0.3	D	Segment D
			Q0.4	E	Segment E
			Q0.5	F	Segment F
			Q0.6	G	Segment G
	COM1←→GND		Q0.7	DP	Decimal point
	KCOM←→GND			COM←→GND	LED common terminal
1M	24V		1L	24V	
			2L	24V	

Note: The start and stop buttons borrow the stepper motor module. K1, K2 in the basic command module.