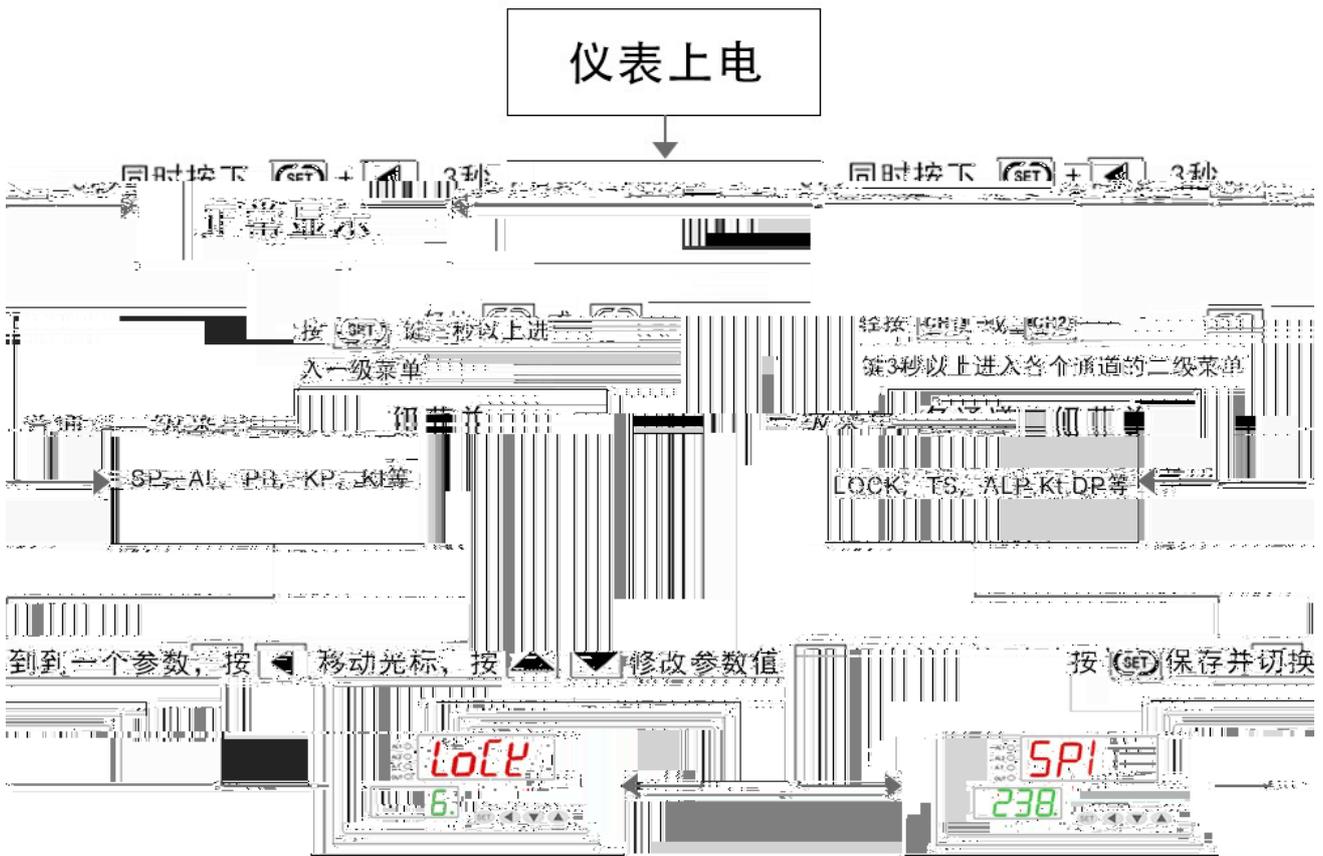


仪表上电



SET 3 1 2
SET

CH1 CH2

SET
5-1

		1-2		SET	
0	LOCK	0 50	18	18	18

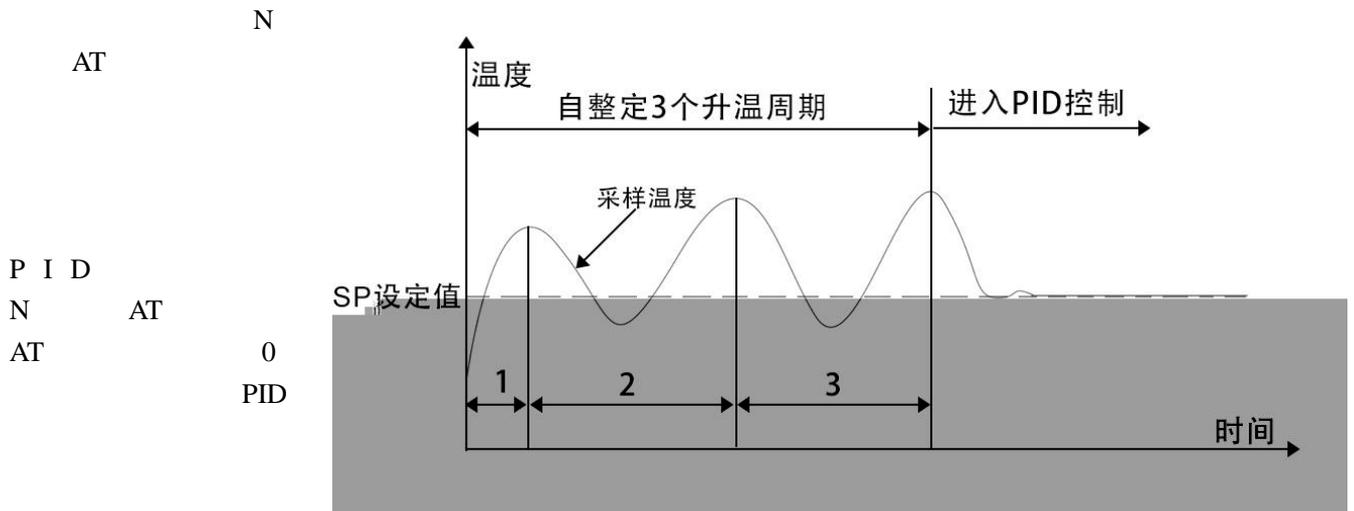
1	LS		-	CU50(LU50) PT100(PT100) K(L) E(E) J(J) T(T) S(S)	K(L)
2	ALP		0 6	0 1 2 3 4 5 6	1
3	LT		0 120 S	2S	10 S
4	dp		0~1	0. ; 1.	0

5 PS-H

SP AL

PT100

SP 20-60 Hy 0.5 1 , AT



AT 0

	ALP	ALI	ALI - HYI
	ALP	ALI	ALI + HYI
	ALP	SPI + ALI	SPI + ALI - HYI
	ALP	SPI - ALI	SPI - ALI + HYI
	ALP		SPI - ALI SPI + ALI
			SPI - ALI + HYI SPI + ALI - HYI
	ALP		SPI - ALI SPI + ALI
			SPI - ALI - HYI SPI + ALI + HYI
	ALP		- ALI
			- ALI - HYI
			SPI ALI HYI ALP

8-1

	1	1

' HH'	' LL'	2	2
-------	-------	---	---

1

A	B	C	D	E	F	G	H	I	J	K	L	M
<i>A</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>J</i>	<i>K</i>	<i>L</i>	<i>M</i>
N	O	P	Q	R	S	T	U	Y				
<i>n</i>	<i>o</i>	<i>P</i>	<i>q</i>	<i>r</i>	<i>S</i>	<i>t</i>	<i>u</i>	<i>y</i>				

2

第一步：修改仪表输入类型



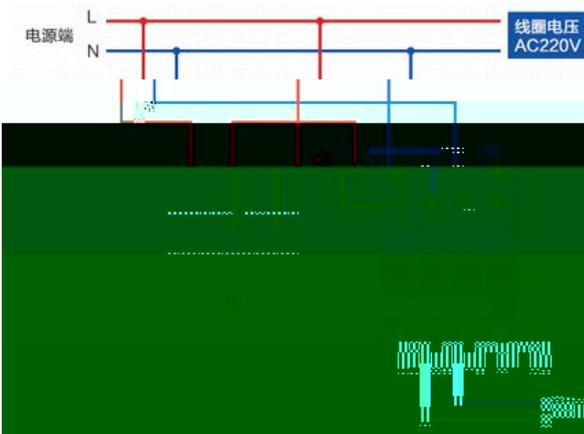
第二步：传感器接入仪表

三线制PT100/CU50接线方法

二线制PT100/CU50接线方法



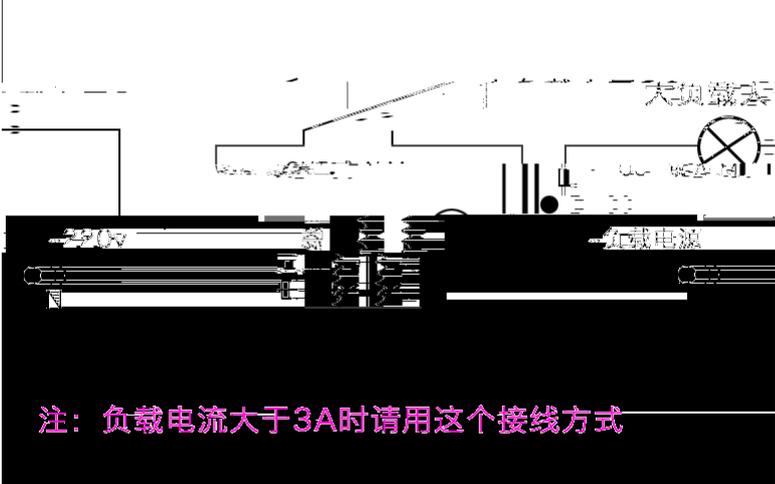
中间继电器接线方法



固态继电器接线方法



OUT ALM 继电器接中间继电器示意图



OUT ALM 继电器接负载示意图



4 Modbus-RTU

1	PC	PLC	RS485	RS232	255
2		1200 2400 4800 9600	1	8	1
1					

		2		CRC16
0103027FFFD834				
01	03	02(2)	7FFF	D834 CRC
7FFF	10	32767		

3

126

	(06)	00xx		CRC16
0106000A04ECAA85				
01	06	000A()	04EC	AA85 CRC
04EC	10	1260	10	12.5 125

3、仪表各种寄存器地址列表：

							PLC	
(PV)	YES	1001H~1004H						44098~44101
+	NO	1101H~1104H						44354~44357
	1101H	D15-D8	D3	D2	D1	D0		
		1			2	1		
		0~100	1 0					
5-1								
Lock	NO	0000H						40001
TS	NO	0001H						40002
BAUD	NO	0009H						40010
1	5-1							
SP1~ COL1	-	000AH~0012H						40011~40019
2	5-1							
SP2~ COL2	-	0013H~001BH						40020~40028

4

1).

ADDR CRC

300ms

2). PLC

PLC,

MODBUS-RTU

MODBUS

8

1

300ms ,

>2

7

5 CRC C++

```

void CRC16_S(byte[] data, int len)
{
    byte CRC16Lo;
    byte CRC16Hi; //CRC寄存器
    byte CL; byte CH; //多项式码&HA001
    byte SaveHi; byte SaveLo;
    int Flag;
    CRC16Lo = 0xFF;
    CRC16Hi = 0xFF;
    CL = 0x01;
    CH = 0xA0;
    for (int i = 0; i < len; i++)
    {
        CRC16Lo = (byte)(CRC16Lo ^ data[i]);
        CRC16Lo = (byte)(CRC16Lo >> 1); //低位右移一位
        if ((SaveHi & 0x01) == 0x01) //如果高位字节最后一位为1
        {
            CRC16Lo = (byte)(CRC16Lo ^ 0x80); //则低位字节右移后前两位
            //否则自动补0
        }
        if ((SaveLo & 0x01) == 0x01) //如果LSB为1,则与多项式码进行
        {
            CRC16Hi = (byte)(CRC16Hi ^ CH);
            CRC16Lo = (byte)(CRC16Lo ^ CL);
        }
        SaveHi = SaveLo;
        SaveLo = CRC16Lo & 0x01;
    }
    //如果是modbus协议的话, 应该是第一位是低位, 第二位是高位
    data[len++] = CRC16Lo; //CRC低位
    data[len] = CRC16Hi; //CRC 高位
}

```

2). PLC

PLC,

MODBUS-RTU

MODBUS

Æ

