

```

/*****/
//Project:12864B
//Dots:128*64
//Driver_IC:ST7565P
//Panel_Mode:STN
//Writer:JWan
//Vdd:3.3V
//Vop:10V(0x25;0x81;0x2a)          9.69V(0x25;0x81;0x25)          9.48(0x25;0x81;0x22)
9.42(0x25;0x81;0x21)
//Date:2011.4.7
/*****/

```

```

#include<REG51.H>
#include<INTRINS.H>

```

```

#define unchar unsigned char
#define uint unsigned int
#define horizontal 64
#define vertical 128

```

```

unchar Vop=0x2a;

```

```

sbit A0 = P3^0;          //A0          EQU      P3.4
sbit CS = P3^1;          //CS          EQU      P3.6
sbit RW = P3^7;          //RW          EQU      P3.7
sbit RES = P3^2;          //RST          EQU      P3.1
sbit PSB = P3^5;
sbit C86 = P3^6;
sbit E = P3^4;           //E          EQU      P3.0

```

```

sbit AOUTOB = P2^1;      //Pause the display when it is '0'
sbit STEP = P2^0;        //Step
sbit UP = P0^0;          //Increase by voltage when it is '1'
sbit DOWN = P0^1;        //Decrease by voltage when it is '1'

```

```

void Write_Data(unchar dat);
void Write_Command(unchar dat);
void Inter_code();
void Delay_ms(uint tt);
void Display(unchar dat1,unchar dat2);
void Test(unchar dat);
//void Frame();

```

[illegible]

0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0x7F, 0x7F, 0x7F, 0x3F, 0x3F, 0x3F, 0x3F, 0x3F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x7F, 0xFF, 0xFF, 0x7F, 0x7F, 0xFF, 0xFF, 0xFF, 0x7F,
0xFF, 0xFF, 0x7F, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x7F, 0x7F, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0x7F, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x7F,
0x7F, 0x7F, 0x7F, 0x7F, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x7F, 0x7F, 0x7F,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00,
0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x3F, 0x0F, 0x0F, 0x07, 0x03, 0x03, 0xFF, 0x80, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0x03, 0x01, 0x00, 0x00, 0x20, 0xC0,
0x00, 0x01, 0x03, 0x0F, 0x1F, 0x1F, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF8, 0xC0, 0xE7, 0x09, 0x00, 0x40, 0x09, 0x09, 0xB9, 0xD2,
0x00, 0xB9, 0xD2, 0x00, 0x00, 0xFF, 0xFF, 0xF3, 0x81, 0xCF, 0xFF, 0x3E, 0x36, 0x62, 0x00, 0x09,
0xC9, 0xC0, 0x44, 0x7E, 0x3E, 0x3E, 0xFF, 0xFF, 0xFF, 0xFF, 0x03, 0x03, 0x67, 0x67, 0x00, 0x00,
0x66, 0x66, 0x66, 0x06, 0x06, 0xFC, 0xFC, 0xFF, 0xFF, 0x3F, 0x3F, 0x3F, 0x3F, 0x3E, 0x3E, 0x00,
0x00, 0x3F, 0x3F, 0x3F, 0x3F, 0x3F, 0x3F, 0x3F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00,
0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0x3F, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0xF8,
0xC0, 0x00, 0x00, 0x00, 0x80, 0x60, 0x30, 0x18, 0x0F, 0x3F, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00,
0x03, 0x3C, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x07, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xCC, 0xCC, 0xFF, 0xE4, 0x80, 0x81, 0xE4, 0xE4, 0xFF, 0xF0,
0x84, 0x9F, 0xC4, 0xE0, 0xF0, 0xFF, 0xFF, 0xE7, 0xE1, 0xF9, 0xFF, 0xFF, 0xF9, 0xC8, 0xC9, 0x81,
0x81, 0xC9, 0xC8, 0xC8, 0xC9, 0xF9, 0xFF, 0xFF, 0xFF, 0xFF, 0xF0, 0xF0, 0xF2, 0xF2, 0x80, 0x80,
0xF2, 0xF2, 0xF2, 0xF0, 0xF0, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xCF, 0xCF, 0xCF, 0xCF, 0xC8,
0xC8, 0xC3, 0xC7, 0xC7, 0xCF, 0xCF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00,
0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xC0, 0x80, 0x00, 0x00, 0x00, 0x00, 0x80, 0xC0, 0x40, 0x20,
0x13, 0x1F, 0x3E, 0x73, 0xE1, 0x80, 0x00, 0x00, 0x00, 0x00, 0x01, 0x07, 0x1C, 0x70, 0xC0, 0x00,
0x00, 0x00, 0x0F, 0xFC, 0x00, 0x00, 0x00, 0x80, 0xE0, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0x7F, 0x1F, 0x0F, 0x1F, 0x7F, 0xFF, 0xFF, 0x8F, 0x0F, 0x3F, 0x7F, 0x7F, 0x7F, 0x7F,
0x3F, 0x0F, 0x8F, 0xFF, 0xFF, 0x0F, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0x7F, 0x3F, 0x1F, 0x0F, 0xFF,
0xFF, 0xCF, 0xCF, 0xCF, 0xCF, 0xCF, 0xCF, 0x0F, 0x1F, 0x3F, 0xFF, 0xFF, 0xFF, 0xFF, 0x0F, 0x0F,
0x0F, 0xFF, 0xFF, 0xFF, 0x8F, 0x0F, 0x3F, 0x7F, 0x7F, 0x7F, 0x7F, 0x7F, 0x3F, 0x0F, 0x8F, 0xFF,
0xFF, 0x0F, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0x7F, 0x3F, 0x1F, 0x8F, 0xCF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00,
0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0xF4, 0xE6, 0xC3, 0x83, 0x01, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x01, 0x07, 0x0F, 0x1E, 0x3C, 0x78, 0xF0, 0xE0, 0xC0, 0x80, 0x00, 0x01,
0x80, 0xC0, 0xE0, 0xFF, 0xF8, 0xF8, 0xFC, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x7F, 0x0F,
0x81, 0xF0, 0xFE, 0xFF, 0xFE, 0xE0, 0x81, 0x1F, 0x7F, 0xFE, 0xF0, 0xC0, 0x0E, 0x3E, 0x0E, 0xC0,
0xF0, 0xFE, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x8F, 0xC7, 0xE1, 0xF0, 0xF8, 0xFE, 0x00, 0x00, 0xFF,
0xFF, 0x0F, 0x07, 0x07, 0x61, 0x71, 0x70, 0x70, 0x78, 0x7F, 0x7F, 0x7F, 0x7F, 0x7F, 0x00, 0x00,
0x00, 0x7F, 0x7F, 0x7F, 0x7F, 0xFE, 0xF0, 0xC0, 0x0E, 0x3E, 0x0E, 0xC0, 0xF0, 0xFE, 0xFF, 0xFF,
0xFF, 0x00, 0x00, 0x71, 0x71, 0x70, 0x70, 0x70, 0x07, 0x8F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00,
0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0xFC,
0xFC, 0xFC, 0xF8, 0xF8, 0xF8, 0xF8, 0xF8, 0xF8, 0xF8, 0xF8, 0xF8, 0xF8, 0xF9, 0xF9, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0xFE,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0xFE, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,


```
0x10, 0x08, 0x07, 0x00, 0x00, 0x1F, 0x11, 0x11, 0x11, 0x11, 0x10, 0x00, 0x00, 0x1F, 0x11, 0x11,
0x11, 0x11, 0x10, 0x00, 0x07, 0x08, 0x10, 0x10, 0x10, 0x10, 0x10, 0x00, 0x00, 0x1F, 0x01, 0x01,
0x01, 0x01, 0x1F, 0x00, 0x00, 0x10, 0x10, 0x1F, 0x10, 0x10, 0x00, 0x00, 0x00, 0x00, 0x10, 0x10,
0x10, 0x1F, 0x00, 0x00, 0x00, 0x1F, 0x01, 0x02, 0x04, 0x08, 0x10, 0x00, 0x00, 0x1F, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x1F, 0x1C, 0x03, 0x00, 0x03, 0x1C, 0x1F, 0x00, 0x00, 0x1F, 0x0C, 0x03,
0x00, 0x00, 0x1F, 0x00, 0x07, 0x08, 0x10, 0x10, 0x10, 0x08, 0x07, 0x00, 0x00, 0x1F, 0x10, 0x10,
0x10, 0x11, 0x0F, 0x00, 0x07, 0x08, 0x10, 0x10, 0x10, 0x08, 0x07, 0x00, 0x00, 0x00, 0x00, 0xFF,
0xFF, 0x00, 0x00, 0x00, 0xC0, 0x40, 0x40, 0x40, 0xF8, 0x40, 0x00, 0x00, 0x00, 0x08, 0x08, 0x08,
0x88, 0xF0, 0x00, 0x00, 0xE0, 0x90, 0x08, 0x08, 0x08, 0xF0, 0x00, 0x00, 0x00, 0x18, 0xE0, 0x00,
0x00, 0x00, 0x00, 0x70, 0x88, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0x00, 0x88, 0x88, 0x88,
0x10, 0xE0, 0x00, 0x00, 0xE0, 0x10, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x18, 0x18,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1A, 0x1C, 0x00, 0x00, 0x00, 0xE0, 0x30, 0x08, 0xE8, 0x28,
0x58, 0xE0, 0x20, 0x00, 0x02, 0x1C, 0xE0, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1A, 0x1C,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x18, 0x18, 0x00, 0x00, 0x00, 0xF8, 0x20, 0x20, 0x20, 0x20,
0x20, 0xF8, 0x00, 0x00, 0xF8, 0x88, 0x88, 0x88, 0x48, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,
0xFF, 0x00, 0x00, 0x00, 0x00, 0x01, 0x06, 0x08, 0x1F, 0x00, 0x00, 0x00, 0x00, 0x1F, 0x11, 0x11,
0x10, 0x10, 0x00, 0x00, 0x07, 0x08, 0x11, 0x11, 0x11, 0x00, 0x00, 0x00, 0x10, 0x10, 0x10, 0x11,
0x16, 0x18, 0x00, 0x00, 0x0E, 0x11, 0x11, 0x11, 0x12, 0x0C, 0x00, 0x00, 0x0F, 0x10, 0x10, 0x10,
0x09, 0x07, 0x00, 0x00, 0x07, 0x08, 0x10, 0x10, 0x08, 0x07, 0x00, 0x00, 0x00, 0x00, 0x03, 0x03,
0x80, 0x80, 0x00, 0x00, 0x80, 0x80, 0x03, 0x03, 0x00, 0x00, 0x00, 0x03, 0x0C, 0x18, 0x13, 0x14,
0x14, 0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x1C, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x04, 0x08, 0x10, 0x08,
0x04, 0x03, 0x00, 0x00, 0x0F, 0x08, 0x08, 0x08, 0x09, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,
0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF6, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x10, 0x96, 0xB8, 0xD0, 0x96, 0xB8, 0xD0, 0x90, 0x00, 0x82, 0x42, 0xFF, 0x22,
0x1C, 0x00, 0x00, 0x82, 0x44, 0x58, 0xA0, 0x5C, 0xA2, 0x22, 0x1C, 0x00, 0x1C, 0x22, 0xC2, 0xA2,
0x12, 0x0E, 0x0A, 0x12, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF8, 0x06, 0x01,
0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01, 0x06, 0xF8, 0x00, 0x00, 0x10, 0x10, 0x10, 0xFE, 0x10,
0x10, 0x10, 0x00, 0x00, 0x54, 0x38, 0x7C, 0x38, 0x54, 0x00, 0x00, 0x00, 0x00, 0x00, 0x36, 0x40,
0x80, 0x00, 0x00, 0x00, 0x02, 0x02, 0x02, 0xFE, 0x02, 0x02, 0x02, 0x00, 0x06, 0x0A, 0x12, 0x62,
0x82, 0x00, 0x00, 0x00, 0x00, 0x02, 0x42, 0x42, 0x42, 0xBC, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,
0xFF, 0x80, 0x80, 0x80, 0x80, 0x80, 0x87, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x8D, 0x8E, 0x80,
0x8D, 0x8E, 0x80, 0x80, 0x80, 0x80, 0x81, 0x86, 0x80, 0x81, 0x86, 0x80, 0x83, 0x84, 0x8F, 0x84,
0x84, 0x80, 0x80, 0x83, 0x84, 0x84, 0x83, 0x80, 0x81, 0x82, 0x84, 0x80, 0x80, 0x83, 0x84, 0x84,
0x83, 0x80, 0x80, 0x80, 0x80, 0x80, 0x8D, 0x8E, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x83, 0x84,
0x8C, 0x88, 0x80, 0x80, 0x88, 0x8C, 0x84, 0x83, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80,
0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x82, 0x84, 0x84, 0x84,
0x84, 0x83, 0x80, 0x80, 0x82, 0x82, 0x82, 0x87, 0x80, 0x80, 0x80, 0x80, 0x84, 0x84, 0x84, 0x84,
0x83, 0x80, 0x80, 0x80, 0x80, 0x84, 0x84, 0x84, 0x84, 0x83, 0x80, 0x80, 0x80, 0x80, 0x80, 0xFF,
};
```

```
unsigned char code pic4[]=
{
```



```

0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55,
0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55,
0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55,
0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55,
0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55,
0x42, 0x42, 0x40, 0x40, 0x00, 0x00, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55,
0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55,
0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55,
0x72, 0x01, 0xF2, 0x11, 0xF2, 0x01, 0x02, 0x01, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55,
0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55,
0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55,
0xF2, 0x01, 0xF2, 0x11, 0xF2, 0x01, 0xF2, 0x11, 0xF2, 0x01, 0x02, 0x01, 0xAA, 0x55, 0xAA, 0x55,
0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55,
0x08, 0x08, 0x00, 0x00, 0xC0, 0x00, 0x02, 0x01, 0x02, 0xF1, 0x02, 0x01, 0x02, 0x01, 0xC2, 0x01,
0x02, 0x01, 0x02, 0xF1, 0x02, 0x01, 0x02, 0x01, 0xC2, 0x01, 0x02, 0x01, 0x02, 0xF1, 0x02, 0x01,
0x02, 0x01, 0xC2, 0x01, 0x02, 0x01, 0x02, 0xF1, 0x02, 0x01, 0x02, 0x01, 0x02, 0x00, 0xC0, 0x00, 0x01, 0x01,
0x01, 0xF0, 0x01, 0x01, 0x01, 0x00, 0xC0, 0x00, 0x02, 0x01, 0x02, 0xF1, 0x02, 0x01, 0x02, 0x01,
0xC2, 0x01, 0x02, 0x01, 0x02, 0xF1, 0x02, 0x01, 0x02, 0x01, 0xC2, 0x01, 0x02, 0x01, 0x02, 0xF1,
0x02, 0x01, 0x02, 0x01, 0xC2, 0x01, 0x02, 0x01, 0x02, 0xF1, 0x02, 0x01, 0x02, 0x00, 0xC0, 0x00,
0x01, 0x00, 0x01, 0xF1, 0x01, 0x00, 0x01, 0x01, 0xC1, 0x00, 0x00, 0x00, 0x02, 0xF1, 0x02, 0x01,
0x02, 0x01, 0xC2, 0x01, 0x02, 0x01, 0x02, 0xF1, 0x02, 0x01, 0x02, 0x01, 0xC2, 0x01, 0x02, 0x01,
};

```

```

main()
{

    RES = 1;
    Delay_ms(10);
    RES = 0;          //复位，低有效
    Delay_ms(50);
    RES = 1;
    Delay_ms(100);

    C86=1;
    PSB=1;

    //P0=0x00;

    Inter_code();

    while(1)

```



```
{

    Test(0xff);
    Delay_ms(1000);
    Pause();

    Test(0x00);
    Delay_ms(1000);
    Pause();

    Display(0xaa, 0x55);
    Delay_ms(1000);
    Pause();

    Display(0xff, 0x00);
    Delay_ms(1000);
    Pause();

    Display(0xaa, 0xaa);
    Delay_ms(1000);
    Pause();

    //Frame();
    //Delay_ms(1000);
    //Pause();

    displaychar(pic1);
    Delay_ms(1000);
    Pause();

    displaychar(pic2);
    Delay_ms(1000);
    Pause();

    displaychar(pic3);
    Delay_ms(1000);
    Pause();

    displaychar(pic4);
    Delay_ms(1000);
    Pause();
}
}
```

```

/*****Delay_ms*****/
void Delay_ms(unint tt)
{
    unchar TT;
    while(--tt)
        for(TT=0;TT<100;TT++);

}

/*****Aouto and Step*****/
void Pause()
{
while(AOUTOB==1&STEP==0)
{
    if(UP==1)
    {
        Delay_ms(1000);
        if(Vop<0x3f)
        {
            Write_Command(0x81);
            Write_Command(++Vop);
        }
    }
    if(DOWN==1)
    {
        Delay_ms(1000);
        if(Vop>0x00)
        {
            Write_Command(0x81);
            Write_Command(--Vop);
        }
    }
}
if(STEP==1)
{
    Delay_ms(10);
    while(AOUTOB==0)
    {}
}
else if(AOUTOB==1)
{
    Delay_ms(10);
}
}

```

```
//Write Command for 6800 8 bits system in parellel
```

```
void Write_Command(unchar dat)
```

```
{
    A0=0;
    E=1;
    RW=0;
    CS=0;
    P1=dat;
    E=0;
    CS=1;
    RW=1;

```

```
}
```

```
//Write data for 6800 8 bits system in parellel
```

```
void Write_Data(unchar dat)
```

```
{
    A0=1;
    RW=0;
    E=1;
    CS=0;
    P1=dat;
    E=0;
    CS=1;
    RW=1;

```

```
}
```

```
/******Frame******/
```

```
/*
```

```
void Frame()
```

```
{
    uint i,j,k;
    unchar page,col; //页变量、列变量
    page=0xb0;
    k=horizontal/8-2;
    j=vertical-2;

```

```
    Write_Command(page++);
    Write_Command(0xb0);
    Write_Command(0x10);
    Write_Command(0x00);

```

```

        Write_Data(0xff);
    for(col=0;col<j;col++)
    {
        Write_Data(0x01);
    }
    Write_Data(0xff);

    for(i=0;i<k;i++)
    {
        Write_Command(page++);
        Write_Command(0x10);
        Write_Command(0x00);
        Write_Data(0xff);
        for(col=0;col<j;col++)
        {
            Write_Data(0x00);
        }
        Write_Data(0xff);
    }

    Write_Command(page++);
    Write_Command(0x10);
    Write_Command(0x00);
    Write_Data(0xff);
    for(col=0;col<j;col++)
    {
        Write_Data(0x80);
    }
    Write_Data(0xff);
    Delay_ms(100);
}
*/

//Test for all white and all black
void Test(unchar dat) //功能：在全屏中的每一列都写入同一个数据, dat 为待写入的数据
{
    uint i,k;
    unchar page,column;
    page=0xb0;
    k=horizontal/8;

    for(i=0;i<k;i++)
    {

```

```

        Write_Command(0x40);
        Write_Command(page++);
        Write_Command(0x10);
        Write_Command(0x00);

        for( column=0; column<vertical; column++ )
        {
            Write_Data(dat);
        }

    }
    Delay_ms(100);
}

/*****Display*****/
void Display(unchar dat1,unchar dat2) //写 2 列 8 行数据
{
    uint i,j,k;
    unchar page,column;
    page=0xb0;
    j=vertical/2;
    k=horizontal/8;
    for(i=0;i<k;i++)
    {
        Write_Command(0x40);
        Write_Command(page++);
        Write_Command(0x10);
        Write_Command(0x00);
        for(column=0;column<j;column++)
        {
            Write_Data(dat1);
            Write_Data(dat2);
        }
    }
    Delay_ms(100);
}

//-----

```

```

void displaychar(unsigned char *p)
{
    uint i,k;
    unchar page,col;
    page=0xb0;
    k=horizontal/8;

    for (i=0;i<k;i++)
    {
        Write_Command(0x40);
        Write_Command(page++);//set page address
        Write_Command(0x10);//set column high address
        Write_Command(0x00);//low address
        for(col=0; col<vertical; col++) //
        {
            Write_Data(*p++);
        }
    }
    Delay_ms(100);
}

/*****Inter_code*****/
void Inter_code()
{
    Write_Command(0xe2);          //IC 复位

    Write_Command(0xa2);          //BIAS 设置 a3: 1/7bias a2:1/9bias
    Write_Command(0xa0);          //ADC normal 设置 a0 正常 a1 COM 和 SEG 颠倒
    Write_Command(0xc0);          //COM 线扫描方向选择

    Write_Command(0x25);          //V5 电压 regulator(校准)内部电阻比率设置
    Ra/Rb

    Write_Command(0x81);          //设置电压值
    Write_Command(Vop);          //Vout 电压值大小设置

    Write_Command(0x2f);          //Power circuit 设置 bias 开启 regulator 开启
    follower 开启
    Write_Command(0xaf);          //显示打开
}

```