```
/*******************************************************;
;*          IC                  : ST75256                                  *;
;*      MCU type      : W78E52B(8K ROM)            *;
;*      Data                    : 2014.01.11      spi mode            *;
;*                                                                         *;
;*******************************************************/
#include<reg51.h>
#include<intrins.h>
#include<P1.h>
#define uint unsigned int
#define uchar unsigned char
//=========================================================
sbit RES=P3^1;
sbit CS1=P3^2;
sbit A0=P3^0;
sbit SCL=P1^0;
sbit SDI=P1^1;
sbit key1=P3^7;
sbit key2=P3^6;
sbit key3=P3^4;
//=========================================================
void init();
void clealddram();
void font1();
void font2();
void font3();
void showpic(char *k);
void write_com(int para);
void write_data(int para);
```

```
void delay_ms(int t);
void kscan1();
void kscan2();
void kscan3();
//=========================================================
void init()
{
        RES=1;
        delay_ms(20);
                RES=0;
        delay_ms(20);
        RES=1;
        delay_ms(200);

        write_com(0x30);   //Extension Command1
                write_com(0x6e);   //Enable Master

                write_com(0x31);   //Extension Command2
                write_com(0xd7);   //Disable Auto Read
                write_data(0x9f);  //Extension Command
                write_com(0xe0);   //Enable OTP Read
                write_data(0x00);  //Extension Command
                delay_ms(20);
                write_com(0xe3);   //OTP Up-Load
                delay_ms(20);
                write_com(0xe1);   //OTP Control Out

                write_com(0x30);   //Extension Command1
                write_com(0x94);   //Sleep Out
```

```
                write_com(0xae);   //Display off
                delay_ms(50);

                write_com(0x20);   //Power Control
                write_data(0x0b);  //VB,VR,VF AllON

                write_com(0x0C);         // D0=0 LSB on top

                write_com(0x81);   //Set VOP
//write_data(0x3e);  //0x03/0x3f=13.8V 0x2b=13.0 ;0x1d/0x04=15.0V
                write_data(0x11);  //0x03/0x3f=13.8V 0x2b=13.0 ;0x1d/0x04=15.0V
                write_data(0x03);  //

                write_com(0x31);         // Extension Command 2
                write_com(0x32);   //Analog Circuit Set
                write_data(0x00);  //
                write_data(0x01);  //Booster Efficiency-6KHz
                write_data(0x05);  //0x05=Bias=1/9 0x02=1/12 0x00=1/14

                write_com(0x51);   //Booster Level
                write_data(0xfb);  //*10

                write_com(0x30);   //Extension Command1

                write_com(0xf0);   //Display Mode
                write_data(0x10);  //Mono Mode

                write_com(0xca);   //Display Control
                write_data(0x00);  //CL Dividing Ratio---Not Divide
```

```
                write_data(0xa1);  //Duty Set---- 1/162
                write_data(0x00);  //Frame Inversion

                write_com(0x31);         // Extension Command 2

                write_com(0xf2);   //Temperature Range
                write_data(0x1E);  // TA=-10
                write_data(0x28);  // TB=0
                write_data(0x32);  // TC=10

        write_com(0xf0);   //Frame rate
                write_data(0x16);  //
                write_data(0x16);  //
                write_data(0x16);  // 0X18
                write_data(0x16);  //102Hz

                write_com(0x30);   //Extension Command1
                write_com(0xbc);   //Data Scan Direction
                write_data(0x00);  //Address direction

                write_com(0xa6);  //Normal display

                write_com(0x31);   //Extension Command2
                write_com(0x40);   //Internal Power Supply

                clealddram();

                write_com(0xaf);  //Display ON
}
```

```
//=====================================================
void clealddram()
{
        int i,j;
        write_com(0x30);
        write_com(0x15);        // Column Address Setting
        write_data(0x00);       // SEG0 -> SEG255
        write_data(0xFF);       //0XBF=192
        write_com(0x75);        // Page Address Setting
        write_data(0x00);       // COM0 -> COM161
        write_data(0x1f);
        write_com(0x5c);
        for(i=0;i<4;i++)
        {
        for(j=0;j<256;j++)
        {
        write_data(0x00);
        }
        }
}
//=================================================
void font1()
{
        int i,j;
        write_com(0x30);
        write_com(0x15);
        write_data(0x00);
        write_data(0xFF);
        write_com(0x75);
```

```
        write_data(0x00);
        write_data(0x1f);
        write_com(0x5c);
        for(i=0;i<4;i++)
        {
        for(j=0;j<256;j++)
        {
        write_data(0xff);
        }
        }
}
//=========================================================
void font2()
{
        int i,j;
        write_com(0x30);
        write_com(0x15);
        write_data(0x00);
        write_data(0xFF);
        write_com(0x75);
        write_data(0x00);
        write_data(0x1f);
        write_com(0x5c);
        for(i=0;i<4;i++)
        {
        for(j=0;j<256;j++)
        {
        write_data(0x55);
        }
```

```
        }
}
//=========================================================
void font3()
{
        int i,j;
        write_com(0x30);
        write_com(0x15);
        write_data(0x00);
        write_data(0xFF);
        write_com(0x75);
        write_data(0x00);
        write_data(0x1f);
        write_com(0x5c);
        for(i=0;i<4;i++)
        {
        for(j=0;j<128;j++)
        {
        write_data(0x55);
        write_data(0xaa);
        }
        }
}
//=========================================================
void font4()
{
        int i,j;
        write_com(0x30);
        write_com(0x15);
```

```
        write_data(0x00);
        write_data(0xFF);
        write_com(0x75);
        write_data(0x00);
        write_data(0x1f);
        write_com(0x5c);
        for(i=0;i<4;i++)
        {
        for(j=0;j<128;j++)
        {
        write_data(0xFF);
        write_data(0x00);
        }
        }
}
//=====================================================
void showpic(char *k)
{
        int i,j;
        write_com(0x30);
        write_com(0x15);
        write_data(0x00);
        write_data(0xFf);
        write_com(0x75);
        write_data(0x00);
        write_data(0x1f);
        write_com(0x5c);
        for(i=0;i<4;i++)
        {
```

```
        for(j=0;j<256;j++)
        {
        write_data(*k++);
        }
        }
}
//=====================================================
void write_com(int para)
{
        int j;
        j=8;
        CS1=0;
        A0=0;
        do
{
        if(para&0x80)
        SDI=1;
        else
        SDI=0;
        SCL=0;
        //delay(2);
        SCL=1;
        --j;
        para<<=1;
}
        while(j);
        CS1=1;
}
//=====================================================
```

```
void write_data(int para)
{
        int j;
        j=8;
        CS1=0;
        A0=1;
        do
{
        if(para&0x80)
        SDI=1;
        else
        SDI=0;
        SCL=0;
        //delay(2);
        SCL=1;
        --j;
        para<<=1;
}
        while(j);
        CS1=1;
}
//=========================================================
void kscan1()
{
        delay_ms(10);
        while(key1==1)
{
        delay_ms(10);
}
```

```
}
//=========================================================
void kscan2()
{
        delay_ms(10);
        while(key2==1)
{
        delay_ms(10);
}
}
//=========================================================
void kscan3()
{
        delay_ms(10);
        while(key3==1)
{
        delay_ms(10);
}
}
//=========================================================
void delay_ms(int t)
{
        register int i,j,k;
        for(i=0;i<t;i++)
        for(j=0;j<85;j++)
        for(k=0;k<2;k++);
}
//=========================================================
main()
```

```
{
        init();
        delay_ms(5);

        while(1)
{
   clealddram();
        showpic(bmp1);
        delay_ms(600);
        //kscan1();

        font1();
        delay_ms(600);
        //kscan1();

        font2();
        delay_ms(600);
        //kscan1();

        //font3();
        //delay_ms(1000);

        font4();
        delay_ms(600);

}
}
```