



液晶模块使用电压是2.5V-3.5V
 CPU是5V是用1117/3.3V，将电压直接转为3.3V
 CPU是3.3V是不用1117/3.3V，直接接3.3V
 所有电容的耐压值选25V或以上

```
#include <reg52.h>
#include <intrins.h>
#include <Ctype.h>
#include <fonit.h>

sbit cs1=P1^1;
sbit reset=P1^0;
sbit rs=P3^0;
sbit sclk=P3^1;
sbit sid=P3^2;
sbit key=P2^0;

void delay_us(int i);
void delay(int i);

//写指令到LCD 模块
```

```

void transfer_command(int data1)
{
    char i;
    cs1=0;
    rs=0;
    for(i=0;i<8;i++)
    {
        sclk=0;
        if(data1&0x80) sid=1;
        else sid=0;
        sclk=1;
        data1=data1<<=1;
    }
    cs1=1;
}

```

//写数据到LCD 模块

```

void transfer_data(int data1)
{
    char i;
    cs1=0;
    rs=1;
    for(i=0;i<8;i++)
    {
        sclk=0;
        if(data1&0x80) sid=1;
        else sid=0;
        sclk=1;
        data1=data1<<=1;
    }
    cs1=1;
}

```

//延时 1

```

void delay(int i)
{
    int j,k;
    for(j=0;j<i;j++)
    for(k=0;k<110;k++);
}

```

//延时 2

```

void delay_us(int i)
{

```

```

int j,k;
for(j=0;j<i;j++)
for(k=0;k<10;k++);
}

void waitkey()
{
repeat:
    if(key==1)goto repeat;
    else delay(2800);
}

//LCD 模块初始化
void initial_lcd()
{
    reset=0;          //低电平复位
    delay(800);
    reset=1;          //复位完毕
    delay(800);
    transfer_command(0xe2); //软复位
    delay(500);
    transfer_command(0x2f); //打开内部升压
    delay(500);

    transfer_command(0x81); //微调对比度
    transfer_command(0x3a); //微调对比度的值，可设置范围 0x00~0xFF
    transfer_command(0xe9); //1/7 偏压比 (bias)
    transfer_command(0xc4); //行扫描顺序：从上到下 0xc2
    transfer_command(0xf1);
    transfer_command(0x1f);
    transfer_command(0xaf); //开显示
}

void lcd_address(uchar page,uchar column)
{
    transfer_command(0xb0+page);          //设置页地址。每页是 8 行。一个画面的 32 行被分成 4 个页。
    我们平常所说的第 1 页，在 LCD 驱动 IC 里是第 0 页，所以在这里减去 1
    transfer_command(((column>>4)&0x0f)+0x10); //设置列地址的高 4 位
    transfer_command(column&0x0f);          //设置列地址的低 4 位
}

//全屏清屏
void clear_screen()
{
    unsigned char i,j;

```

```
for(i=0;i<8;i++)
{
    lcd_address(i,0);
    for(j=0;j<128;j++)
    {
        transfer_data(0x00);
    }
}

void display_graphic_128x32(uchar *dp)
{
    uchar i,j;
    for(i=0;i<4;i++)
    {
        lcd_address(i,0);
        for(j=0;j<128;j++)
        {
            transfer_data(*dp);
            dp++;
        }
    }
}
```