

OLED-0.96 寸显示屏使用教程

1.1 OLED 简介

OLED，即有机发光二极管（Organic Light Emitting Diode）。OLED 由于同时具备自发光，不需背光源、对比度高、厚度薄、视角广、反应速度快、可用于挠曲性面板、使用温度范围广、构造及制程较简单等优异之特性，被认为是下一代的平面显示器新兴应用技术。

LCD 都需要背光，而 OLED 不需要，因为它是自发光的。这样同样的显示 OLED 效果要来得更好一些。以目前的技术，OLED 的尺寸还难以大型化，但是分辨率确可以做到很高。在此我们使用的是 0.96 寸 OLED 显示屏，内部驱动芯片是 SSD1306，如图 1.1.1 所示。该屏有以下特点：

1) 0.96 寸 OLED 有黄蓝，白，蓝三种颜色可选；其中黄蓝是屏上 1/4 部分为黄光，下 3/4 为蓝；而且是固定区域显示固定颜色，颜色和显示区域均不能修改；白光则为纯白，也就是黑底白字；蓝色则为纯蓝，也就是黑底蓝字。

2) 分辨率为 128*64

3) 多种接口方式；OLED 裸屏总共种接口包括：6800、8080 两种并行接口方式、3 线或 4 线的串行 SPI 接口方式、IIC 接口方式（只需要 2 根线就可以控制 OLED），这五种接口是通过屏上的 BS0-BS2 来配置的。

4) 本教程使用的是 4 线串行 SPI 接口方式的 OLED 屏，转接板已经将 BS0-BS2 设置好，如需使用其他接口方式，可以修改转接板上电阻位置即可。

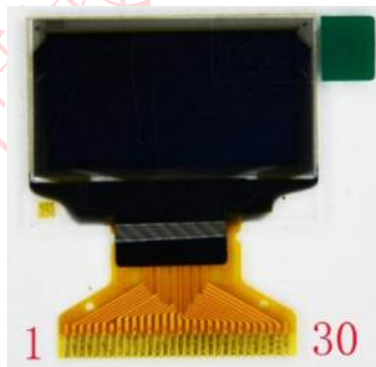


图 1.1.1 OLED-0.9 寸显示屏

1.2 OLED-0.96 寸显示屏模块使用方法

1.2.1 OLED 模块介绍

OLED-0.96 寸显示屏模块其实就是将 OLED-0.96 寸显示屏焊接在转接板上，通过转接板确定 BS0-BS2 的配置，并将 OLED 屏的驱动接口和电源接口引出。我

们使用的 OLED 模块默认采用 SPI 接口，如需使用 IIC 接口通信的朋友就需要把相应的电阻位置调换，具体的在模块背面丝印处标明。OLED-0.96 寸显示屏模块如图 1.1.2 所示：

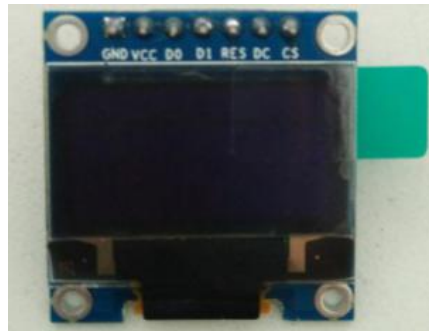


图 1.1.2 OLED-0.9 寸显示屏模块

从图 1.1.1 可以看到，OLED-0.96 显示屏总共有 30 个引脚，通常显示屏的排线上会有起始和终止管脚序号，如果自己设计转接板时，OLED 焊盘千万不要搞错方向，否则很有可能烧掉显示屏。

从图 1.1.2 可以看到，OLED-0.96 寸显示屏引出了 7 个管脚（其他的很多管脚我们没有引出），分别功能分别为：

GND：电源地

VCC：电源正（3~5.5V）

D0：OLED 的 D0 脚，在 SPI 和 IIC 通信中为时钟管脚

D1：OLED 的 D1 脚，在 SPI 和 IIC 通信中为数据管脚

RES：OLED 的 RES 脚，用来复位（低电平复位）

DC：OLED 的 D/C 脚，数据和命令控制管脚

CS：OLED 的 CS 脚，也就是片选管脚

SPI 通信比较简单，传输过程中每到一个时钟上升沿时，先传送高位后低位。DC 用于选择是发送命令还是数据，其时序图如图 1.1.3 所示：

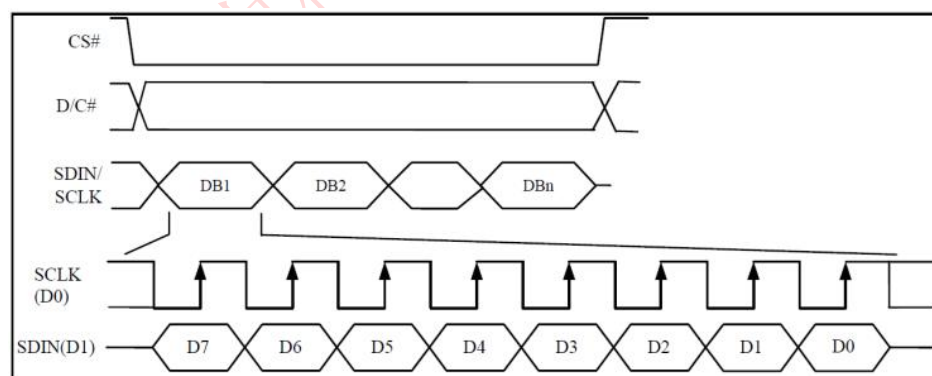


图 1.1.3 4 线 SPI 写操作时序图

IIC 通信的时序这里就不截取出来，不了解的大家可以百度下，也可以看下我们 STM32 的 IIC 视频教程。

如果有朋友从其他地方买的 OLED 模块，可能会有十多个管脚，那种的可能就是 8080 并口模式，OLED-0.96 寸显示屏驱动芯片 SSD1306 可以支持很多种模式的通信接口，只需要配置 BS0-BS2 就可以选择。SSD1306 芯片资料大家可以打开“[\模块芯片资料](#)”内对应数据手册了解。其实我们也只是了解下几个命令的功能，后面初始化程序讲解部分会说到。

使用的还就是它的初始化程序，不过这部分内容显示屏厂家都会提供，所以大家也没必要自己编写，你只需要根据初始化内使用到的命令到数据手册内查找，了解下功能即可。

1.2.2 OLED 模块与开发板连线说明

拿到这个模块应该怎么与开发板连接呢？我们可以使用杜邦线，将 OLED 模块的 7 个管脚对应连接到开发板上，其连接管脚说明如图 1.2.1 所示：

OLED-0.96寸模块		STM32芯片管脚
GND		GND
VCC		3.3V/5V
D0		PD6
D1		PD7
RES		PD4
DC		PD5
CS		PD3

图 1.2.1 模块连线图

如果使用 IIC 接口的朋友，先要将转接板背面电阻位置调换，然后只需要连接 GND、VCC、D0 和 D1 即可。

1.3 OLED-0.96 寸显示屏实验

本实验程序兼容 4 线 SPI 接口和 8080 接口的 OLED-0.96 寸显示屏，程序内已经预留好 IIC 接口的程序框架，如果大家有兴趣可以把 IIC 驱动程序加进去，使程序兼容性更强。

1.3.1 OLED 初始化

要让 OLED 显示屏显示，必须对其初始化，包括所用管脚时钟和 SSD1306 驱动芯片初始化，也就是要对 OLED 彩屏进行一些命令和数据的发送，由于我们采用的是 SPI 接口通信，所以需要根据 SPI 时序图编写一个 OLED 写函数，OLED 所有驱动代码都在 oled.c 和 oled.h 文件中，OLED 写函数如下：

```
//向 SSD1106 写入一个字节。
//dat:要写入的数据/命令
//cmd:数据/命令标志 0,表示命令;1,表示数据;
void OLED_WR_Byte(u8 dat,u8 cmd)
{
    u8 i;
    OLED_DC=cmd;
    OLED_CS=0;
    for(i=0;i<8;i++)
    {
```

```

        OLED_SCL=0;
        if(dat&0x80)
            OLED_SDA=1;
        else
            OLED_SDA=0;
        OLED_SCL=1;
        dat<<=1;
    }
    OLED_CS=1;
    OLED_DC=1;
}

```

OLED 初始化代码如下:

```

void OLED_Init()
{
    GPIO_InitTypeDef  GPIO_InitStructure;

    #if OLED_MODE==0    //4 线 SPI 模式
        RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE);

        GPIO_InitStructure.GPIO_Mode=GPIO_Mode_OUT; //输出模式
        GPIO_InitStructure.GPIO_Pin=GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6|GPIO_Pin_7;//管脚设置
        GPIO_InitStructure.GPIO_Speed=GPIO_Speed_100MHz;//速度为 100M
        GPIO_InitStructure.GPIO_OType=GPIO_OType_PP;//推挽输出
        GPIO_InitStructure.GPIO_PuPd=GPIO_PuPd_UP;//上拉
        GPIO_Init(GPIOD, &GPIO_InitStructure); //初始化结构体

        GPIO_SetBits(GPIOD, GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6|GPIO_Pin_7); //拉高电平
    #endif

    #if OLED_MODE==1    //8080 模式

        RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD|RCC_AHB1Periph_GPIOC, ENABLE);

        GPIO_InitStructure.GPIO_Mode=GPIO_Mode_OUT; //输出模式
        GPIO_InitStructure.GPIO_Pin=GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6|GPIO_Pin_7;//管脚设置
        GPIO_InitStructure.GPIO_Speed=GPIO_Speed_100MHz;//速度为 100M
        GPIO_InitStructure.GPIO_OType=GPIO_OType_PP;//推挽输出
        GPIO_InitStructure.GPIO_PuPd=GPIO_PuPd_UP;//上拉
        GPIO_Init(GPIOD, &GPIO_InitStructure); //初始化结构体
    #endif
}

```

```

GPIO_SetBits(GPIOD, GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6|GPIO_Pin_7);

    GPIO_InitStructure.GPIO_Pin = 0xFF;    //PC0-7
    GPIO_Init(GPIOC, &GPIO_InitStructure);
    GPIO_SetBits(GPIOC, 0xFF); //PC0-7 输出高
#endif

#if OLED_MODE==2    //IIC 模式

#endif

    OLED_RST=1;
    delay_ms(100);
    OLED_RST=0;
    delay_ms(100);
    OLED_RST=1;

    OLED_WR_Byte(0xAE, OLED_CMD); //关闭显示
    OLED_WR_Byte(0xD5, OLED_CMD); //设置时钟分频因子, 震荡频率
    OLED_WR_Byte(80, OLED_CMD); // [3:0], 分频因子; [7:4], 震荡频率
    OLED_WR_Byte(0xA8, OLED_CMD); //设置驱动路数
    OLED_WR_Byte(0X3F, OLED_CMD); //默认 0X3F (1/64)
    OLED_WR_Byte(0xD3, OLED_CMD); //设置显示偏移
    OLED_WR_Byte(0X00, OLED_CMD); //默认为 0

    OLED_WR_Byte(0x40, OLED_CMD); //设置显示开始行 [5:0], 行数.

    OLED_WR_Byte(0x8D, OLED_CMD); //电荷泵设置
    OLED_WR_Byte(0x14, OLED_CMD); //bit2, 开启/关闭
    OLED_WR_Byte(0x20, OLED_CMD); //设置内存地址模式
    OLED_WR_Byte(0x02, OLED_CMD); //[1:0], 00, 列地址模式; 01, 行地址模式; 10, 页地址模式; 默认 10;
    OLED_WR_Byte(0xA1, OLED_CMD); // 段 重 定 义 设置, bit0:0, 0->0; 1, 0->127;
    OLED_WR_Byte(0xC0, OLED_CMD); //设置 COM 扫描方向; bit3:0, 普通模式; 1, 重定义模式 COM[N-1]->COM0; N: 驱动路数
    OLED_WR_Byte(0xDA, OLED_CMD); //设置 COM 硬件引脚配置
    OLED_WR_Byte(0x12, OLED_CMD); //[5:4] 配置

    OLED_WR_Byte(0x81, OLED_CMD); //对比度设置
    OLED_WR_Byte(0xEF, OLED_CMD); //1~255; 默认 0X7F (亮度设置, 越大越亮)

```

```

    OLED_WR_Byte(0xD9, OLED_CMD); //设置预充电周期
    OLED_WR_Byte(0xf1, OLED_CMD); //[3:0], PHASE 1; [7:4], PHASE 2;
    OLED_WR_Byte(0xDB, OLED_CMD); //设置 VCOMH 电压倍率
    OLED_WR_Byte(0x30, OLED_CMD); // [6:4]
    000, 0.65*vcc; 001, 0.77*vcc; 011, 0.83*vcc;

```

```

    OLED_WR_Byte(0xA4, OLED_CMD); //全局显示开启; bit0:1, 开启; 0, 关闭; (白屏/黑屏)

```

```

    OLED_WR_Byte(0xA6, OLED_CMD); //设置显示方式; bit0:1, 反相显示; 0, 正常显示

```

```

    OLED_WR_Byte(0xAF, OLED_CMD); //开启显示

```

```

    OLED_Clear();

```

```

}

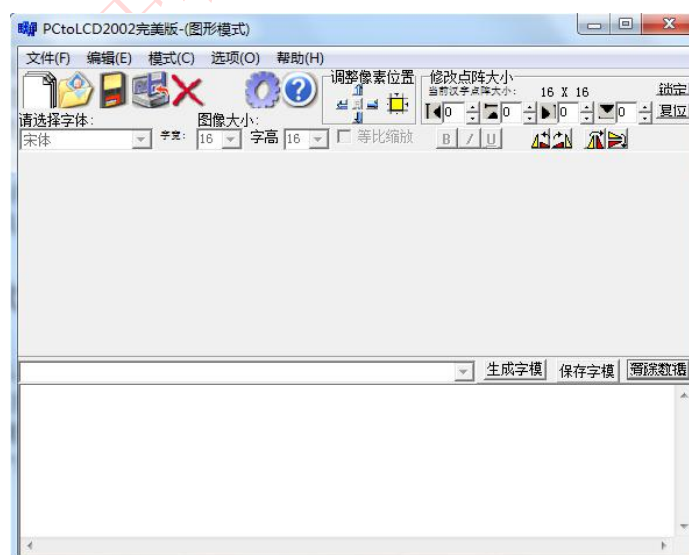
```

程序中可以看到使用了 `#if OLED_MODE...#endif`, 这个是条件编译, 方便根据不同的 `OLED_MODE` 值选择编译程序, 这样就可以实现兼容不同接口的 OLED 屏, 默认我们使用 4 线 SPI 模式, 程序中也为大家编写了 8080 接口的屏驱动。

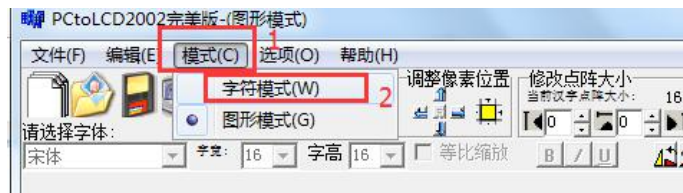
`OLED_WR_Byte()` 函数内传入的命令值, 这些都可以在 SSD1306 数据手册中查询到, 初始化函数一般显示屏厂家都会提供, 所以看不懂没有关系, 关键的是后面显示程序。具体的程序大家可以打开 “\OLED-0.96 寸显示屏程序” 查看。

1.3.2 取模软件的使用

知道了怎么驱动 OLED, 我们还需让它在上面显示一些信息, 比如汉字或者字符等, 那么这些显示内容的数据都是通过取模软件来完成, 所以还需要知道怎么使用这个取模软件。我们使用的取模软件在 “\取模软件\PCtoLCD2002 完美” 目录下, 双击 “PCtoLCD2002” 应用程序, 弹出如下界面:



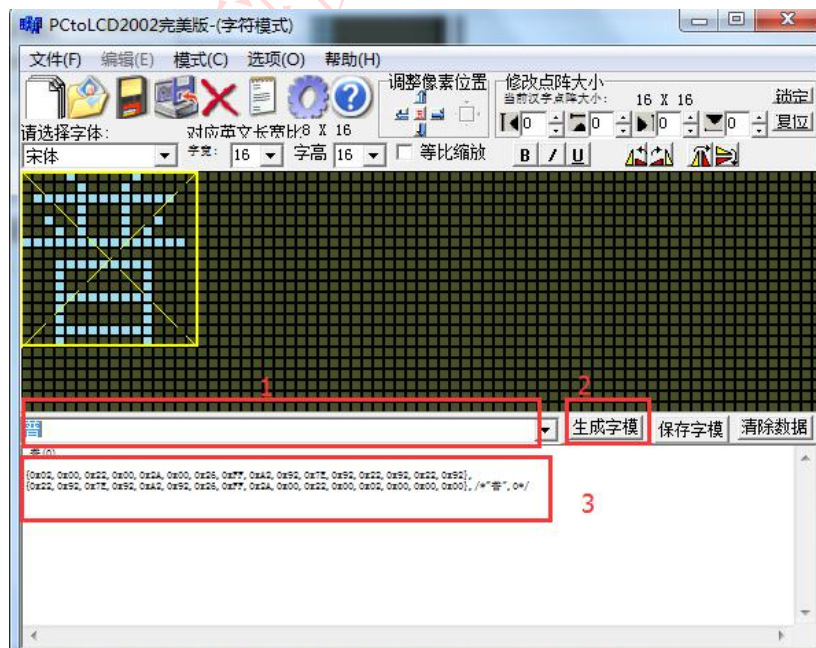
这里我们介绍下如何取汉字和字符字模数据, 点击 “模式”, 选择 “字符模式”, 操作如下: (如果要对图片取模, 就选择 “图形模式”)



然后点击“选项”，会弹出一个字模选项对话框，将其设置为“阴码+逐列式+顺向+C51 格式”，操作如下：



设置好以后点击“确定”按钮，在字符输入输入框内输入你要取的汉字或者字符，然后点击“生成字模”按钮，下面对话框中就会显示你所取的字模数据了，具体操作如下：



显示的内容字模数据取好后，你就可以将其保存在数组当中，注意取模后的汉字数据总共是 32 个字节，被分成了 2 个大数组，所以我们将它整合在一个数组内，具体的可以参考我们 oledfont.h 文件，里面保存的就是我们取模的数据，

可以看到，里面有 12*12、16*16、24*24 ASCII 字符集点阵和 16 号汉字字体点阵数据。

如果要对图片进行取模，方法是一样的，只是开始时选择好一张 BMP 图片，要注意，这个不是 TFT 显示屏，所以不能显示彩色的图片，要使用 OLED 显示图片，只能显示一些简单的图形，而且 OLED 显示屏的大小是 128*64，所以图片大小不能超过这个。点击下图红框位置加载图片，然后点击“模式”选择图片模式，后面操作同上。



在工程中我们也给大家取好了一张简单的图片数据，放在 picture.h 内。

1.3.3 主函数

主函数代码如下：

```
int main()
{
    u8 i;

    SysTick_Init(168);
    NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2); //中断优先级
    分组 分 2 组
    LED_Init();
    USART1_Init(9600);

    OLED_Init(); //OLED 初始化

    OLED_DrawBMP(0, 0, 128, 8, (u8 *)pic1); //如果要正着显示，需要
    将取模方式修改重新取模
    delay_ms(1500);
    OLED_Clear();

    OLED_ShowString(0, 0, "PRECHIN", 16);

    OLED_ShowFontHZ(16*0, 20, 0, 16, 1); //深圳普中科技
    OLED_ShowFontHZ(16*1, 20, 1, 16, 1);
    OLED_ShowFontHZ(16*2, 20, 2, 16, 1);
    OLED_ShowFontHZ(16*3, 20, 3, 16, 1);
    OLED_ShowFontHZ(16*4, 20, 4, 16, 1);
    OLED_ShowFontHZ(16*5, 20, 5, 16, 1);

    OLED_ShowString(0, 40, "www.prechin.cn", 16);
```



```

OLED_Refresh_Gram(); //刷新 GRAM 数组

while(1)
{
    i++;
    if(i%20==0)
    {
        led1=!led1;
    }
    delay_ms(10);
}
}

```

主函数代码功能很简单，将使用到的硬件进行初始化，然后显示调用显示图片函数显示一张图片，1.5 秒后清屏然后进入字符汉字的显示，ASCII 字符的显示大小默认是 16 号字体，如果需要修改此字体的，可以参考下函数内参数意义，ASCII 字体支持 12、16、24 这 3 种。显示汉字函数默认也是 16 号字体，如果需要显示其他字体，需要自己设定字体大小后取模。

1.3.4 实验现象

将模块连接好以后，把实验程序下载到开发板内即可看到 OLED-0.96 寸显示屏开始显示一张图片，1.5 秒后切换到汉字和字符显示界面，同时开发板上 D1 指示灯不断闪烁，表示程序正常运行。