

0.96inch OLED 用户手册

1. 产品特性：

驱动芯片	SSD1306
支持接口	3-wire SPI、4-wire SPI、I2C
分辨率	128*64
显示尺寸	0.96 inch
外形尺寸	29mm*33mm
颜色	黄色、蓝色
视角	>160°
工作温度	-20℃~70℃
存储温度	-30℃~80℃

表 1. 产品特性

2. 应用案例：

- 智能手表
- MP3 播放器
- 温度计
- 仪器仪表
- 便携医疗仪

3. 实验现象：

1) 硬件配置

本模块提供三种驱动接口：分别为 3-wire SPI、4-wire SPI 和 I2C 接口，模块出厂设置 BS0/BS1 置为 0/0，选通 4-wire SPI。

通过 BS0/BS1 的跳线配置模块工作模式和引脚功能。（需要用到焊接器材，在没有我司工作人员指导下擅自更改，将视为自动放弃保修）

BS0/BS1	工作模式	CS	D/C	DIN	CLK
0/0	4-wire SPI	CS	D/C	MOSI	SCLK
0/1	I2C	0	0/1	SDA	SCL
1/0	3-wire SPI	CS	0	MOSI	SCLK

表 2. 设置工作模式

2) 实验现象

这里使用 STM32 开发板为例介绍 4-wire SPI（默认）驱动方式。

- a) 将配套程序下载到相应的开发板中。
- b) 将串口线和模块接入开发板，给开发板上电，打开串口调试软件。
模块与开发板连接如下表所示：

引脚号	标识	描述
1	VCC	电源正(3.3~5V 电源输入)
2	GND	电源地
3	NC	NC
4	DIN	模块数据输入
5	CLK	模块时钟信号输入
6	CS	模块片选信号，低电平有效
7	D/C	模块命令信号，低电平表示命令，高电平表示数据
8	RES	模块复位信号，低电平有效

表 3. 引脚连接

串口配置如下表所示：

波特率	9600
数据位	8
停止位	1
奇偶校验	无

表 4. 波特率设置

- c) 串口接收到如下数据：
This is OLED 4-Wire SPI mode test
- d) OLED 显示如图：

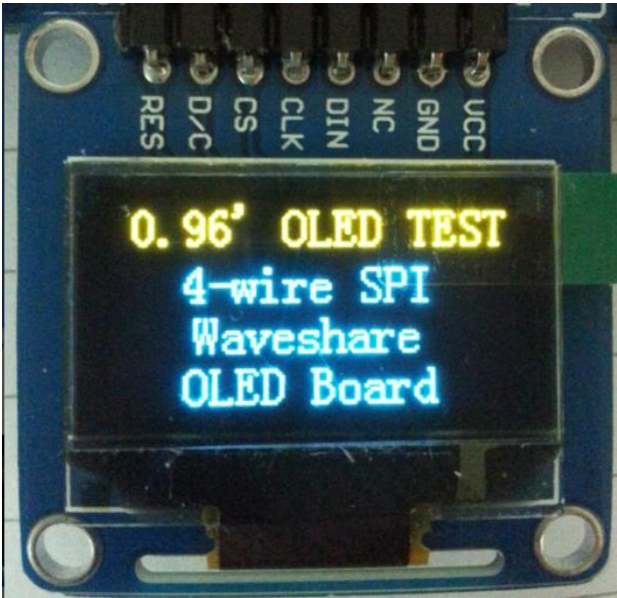


图 1. SPI 模式下 OLED 显示

4. 选择 I2C 驱动方式

注意：模块通过更改 BS0/BS1 的跳线配置为 I2C 驱动方式。此操作需要用到焊接器材，在没有我司工作人员指导下擅自更改，将视为放弃保修。

1) 硬件设置

把模块电路板上的 BS0 和 BS1 改为分别连接低电平和高电平（BS0: 0, BS1: 1, 丝印标识“0”表示低电平, “1”表示高电平）即可, 其余不变。如图:

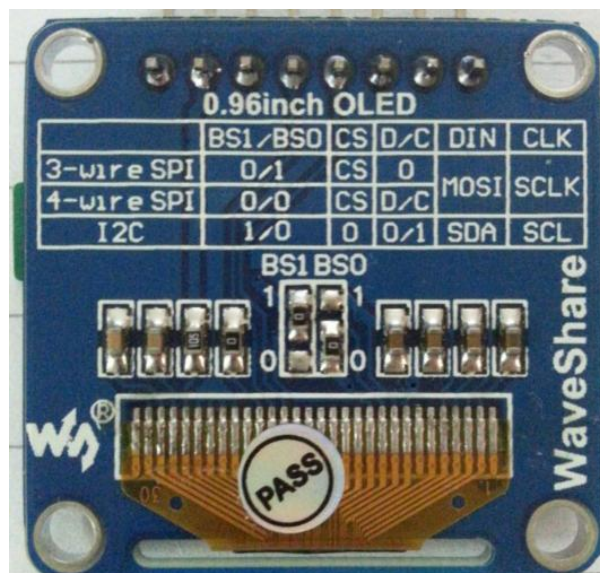


图 2. 设置 BS0、BS1 跳线

2) 软件设置

“SSD1306_Cfg.h”文件中的“#define OLED_MODE OLED_SPI_4LINE_MODE”
改为“#define OLED_MODE OLED_IIC_MODE”, 其他不变。编译下载之后, 连接模块到开发板, 并把接入串口到开发板, 串口接收到如下数据:

This is OLED I2C mode test

OLED 显示如图:

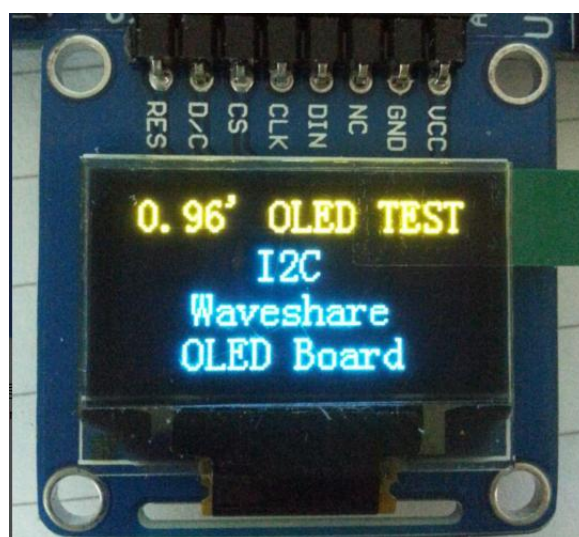


图 3. I2C 模式下 OLED 显示

5. 原理简介

本模块提供了 3 种驱动接口, 在这里简单介绍其中 4-wire SPI 和 I2C 接口的驱动的实现

方法，详细请参考《SSD1306-Revision_1.1.pdf》文档中的“8.1 MCU Interface selection”部分。

在 4-wire SPI 模式中，D0 相当于 SCLK，D1 相当于 SDIN。对于没有使用的引脚，D2 应该悬空，引脚从 D3 到 D7，E 和 RW#(WR#)应连接到外部电源地。

4-wire SPI 接口控制引脚表：

Function	E(RD#)	R/W#(WR#)	CS#	D/C#	D0
Write command	Tie LOW	Tie LOW	L	L	↑
Write data	Tie LOW	Tie LOW	L	H	↑

表 5. 4-wire SPI 接口控制引脚表

SDIN 上的数据按 MSB 在前 LSB 在后为次序在每个 SCLK 的上升沿到来时被移入一个 8-bit 移位寄存器。D/C#在每第 8 个时钟被采样，并且移位寄存器里的数据被写入图形显示内存（GDDRAM）或者命令寄存器，在同样的计数时钟下。

在串行模式下，仅允许写操作。

在 4-wire SPI 模式下的写操作过程，如图：

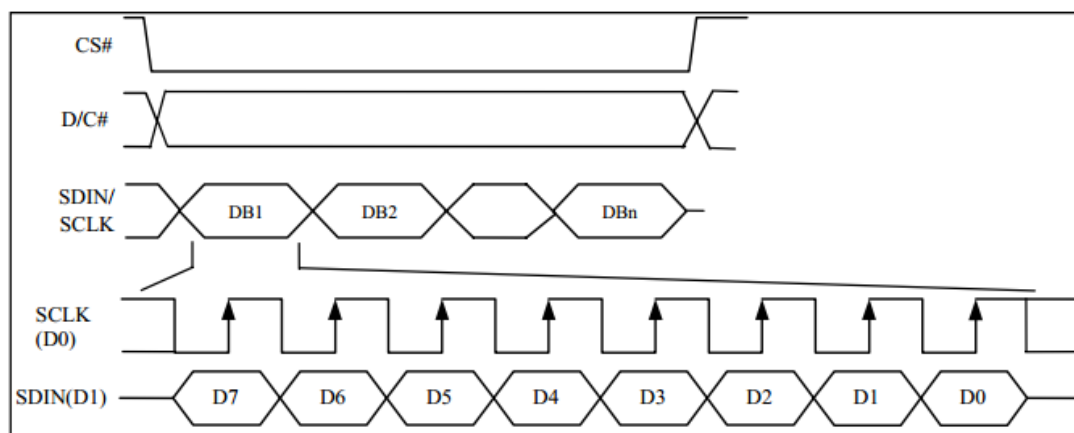


图 4. 4-wire SPI 模式下的写操作过程

I2C 通信接口包括从机地址位 SA0，I2C 总线信号 SDA(SDAOUT/D2 为输出并且 SDAIN/D1 为输入)和 I2C 总线时钟信号 SCL(D0)。

a) 从机地址位（SA0）

SSD1306 通过 I2C 总线传送或者接收任何信息之前，必须确认从机地址。在下列的字节格式中，SSD1306 将回应跟随着从机地址位和读/写选择位的从机地址，

```

b7 b6 b5 b4 b3 b2 b1 b0
0  1  1  1  1  0 SA0 R/W#

```

“SA0”位为从机提供一个扩展位。“0111100”或者“0111101”中的一个，可以被选择作为 SSD1306 从机的地址。D/C#脚相当于 SA0，作为从机地址的选择。

“R/W#”位用于决定 I2C 总线接口的操作模式。R/W#=1，I2C 处于读模式，R/W#=0，I2C 处于写模式。

b) I2C 总线数据信号（SDA）

“SDAIN”和“SDAOUT”已在模块内部短接，共同作为 SDA。

I2C 总线接口提供把数据和命令写入器件的访问，请参考下图的 I2C 总线在时间序列

下的写模式。

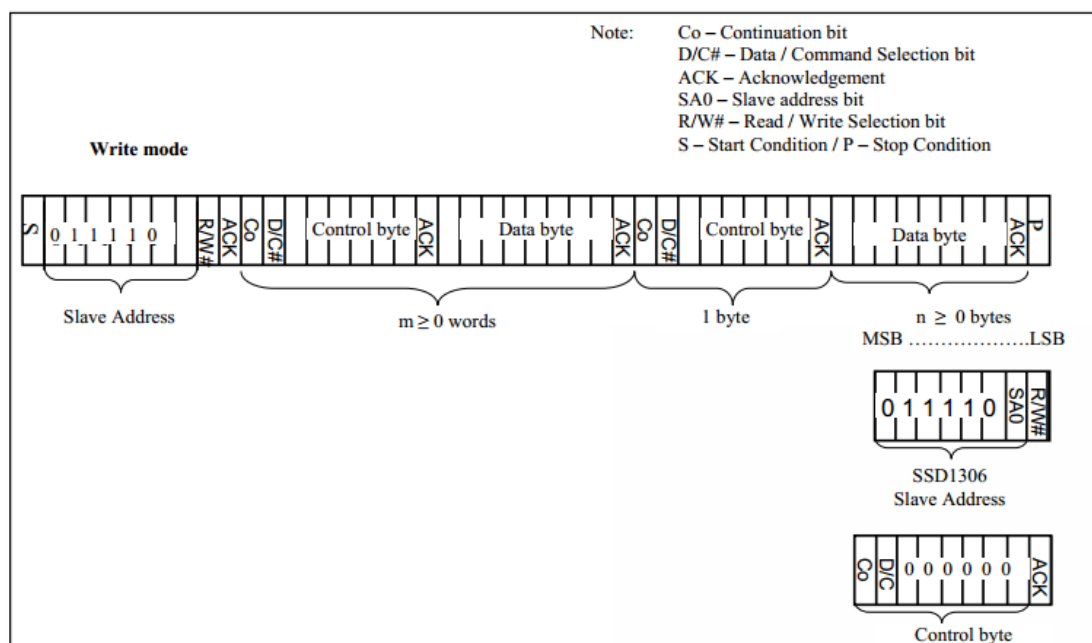


图 5. I2C 总线在时间序列下的写模式

- 1) 通过改变 SA0 置低或高电平（D/C 脚相当于 SA0），得到从机地址是“0111100”或者“0111101”中的一个。
- 2) 写模式的建立是通过设置 R/W# 位为逻辑“0”
- 3) 接收到数据的一个字节后，包括从机地址和 R/W# 位，将会产生一个应答信号。
- 4) 从机地址传送完之后，控制字节或数据字节中的一个会通过 SDA 发送，一个控制字节由 Co 和跟随着 6 个“0”的 D/C# 位组成。
 - a) 如果 Co 位被设置为逻辑“0”，跟随的信息将仅仅包含数据字节被传送。
 - b) D/C# 位决定下一个数据字节被作为一个命令还是一个数据，如果 D/C# 位被置逻辑“0”，它规定了跟随的数据字节作为一个命令，如果 D/C# 位被置逻辑“1”，它规定了跟随的数据字节作为一个将被存储在 GDDRAM 的数据。每一个数据写之后，GDDRAM 的列地址指针将会自动的加一递增。
- 5) 每接收一个控制字节或数据字节之后，将会产生一个应答位。