

测试平台介绍:

开发板: Raspberry Pi 各个型号的开发板

系统: Raspbian for Raspberry Pi

GPIO库: bcm2835、wiringpi

接线说明:

wiringPi 编码	BCM 编码	功能名	物理引脚 BOARD编码		功能名	BCM 编码	wiringPi 编码
		3.3V	1	2	5V		
8	2	SDA.1	3	4	5V		
9	3	SCL.1	5	6	GND		
7	4	GPIO.7	7	8	TXD	14	15
		GND	9	10	RXD	15	16
0	17	GPIO.0	11	12	GPIO.1	18	1
2	27	GPIO.2	13	14	GND		
3	22	GPIO.3	15	16	GPIO.4	23	4
		3.3V	17	18	GPIO.5	24	5
12	10	MOSI	19	20	GND		
13	9	MISO	21	22	GPIO.6	25	6
14	11	SCLK	23	24	CE0	8	10
		GND	25	26	CE1	7	11
30	0	SDA.0	27	28	SCL.0	1	31
21	5	GPIO.21	29	30	GND		
22	6	GPIO.22	31	32	GPIO.26	12	26
23	13	GPIO.23	33	34	GND		
24	19	GPIO.24	35	36	GPIO.27	16	27
25	26	GPIO.25	37	38	GPIO.28	20	28
		GND	39	40	GPIO.29	21	29

图1. GPIO map

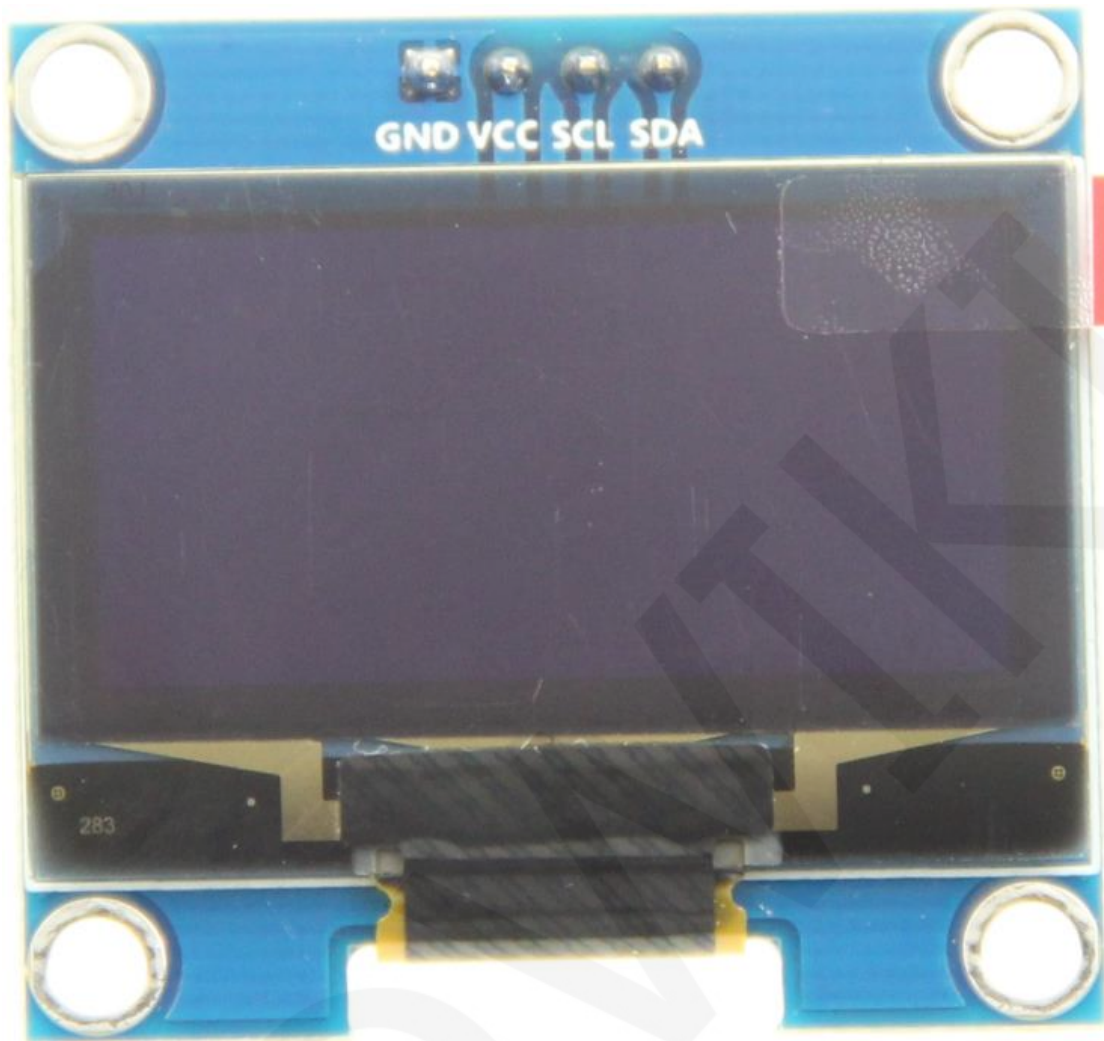


图2. 模块引脚丝印图（1脚为GND）

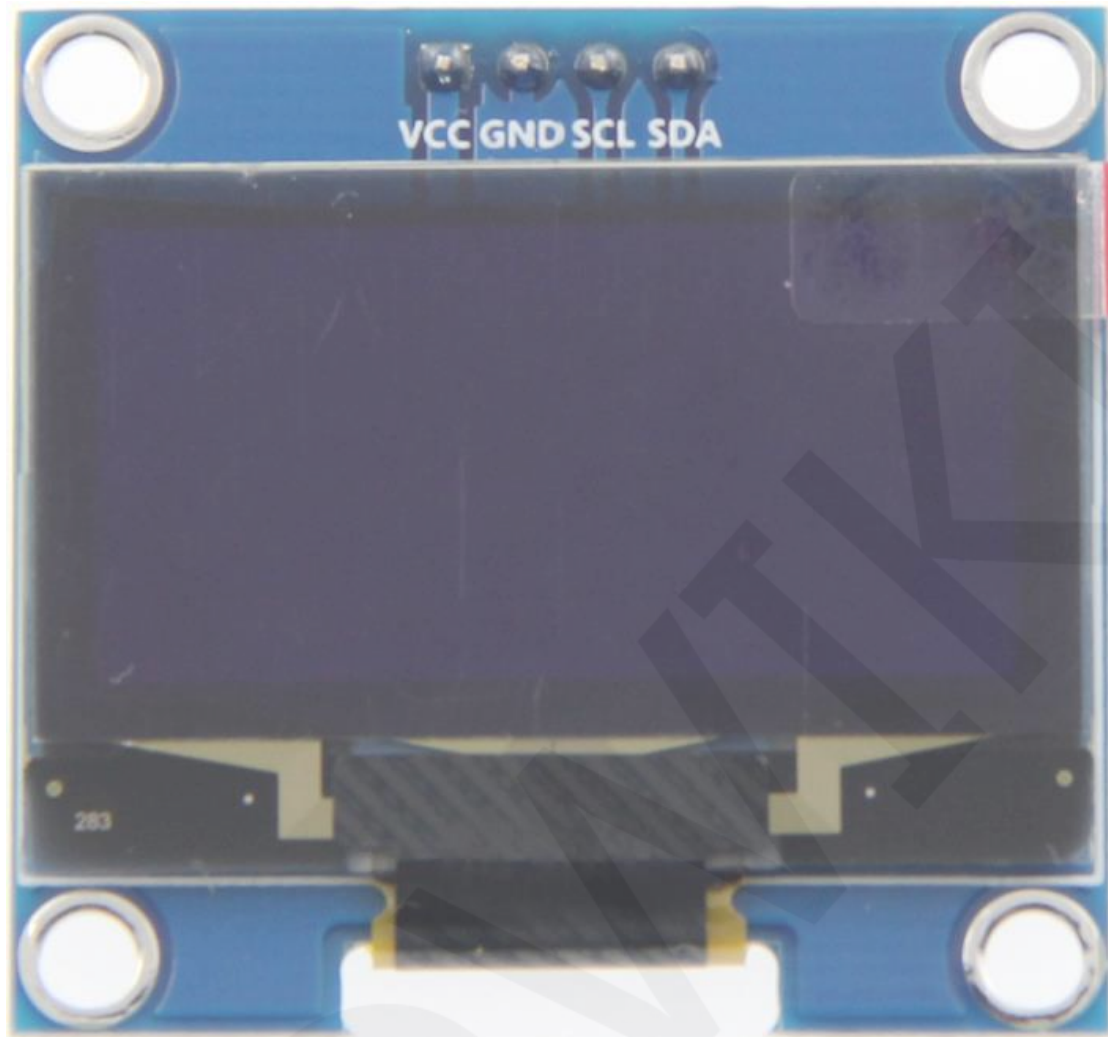


图2. 模块引脚丝印图（1脚为VCC）

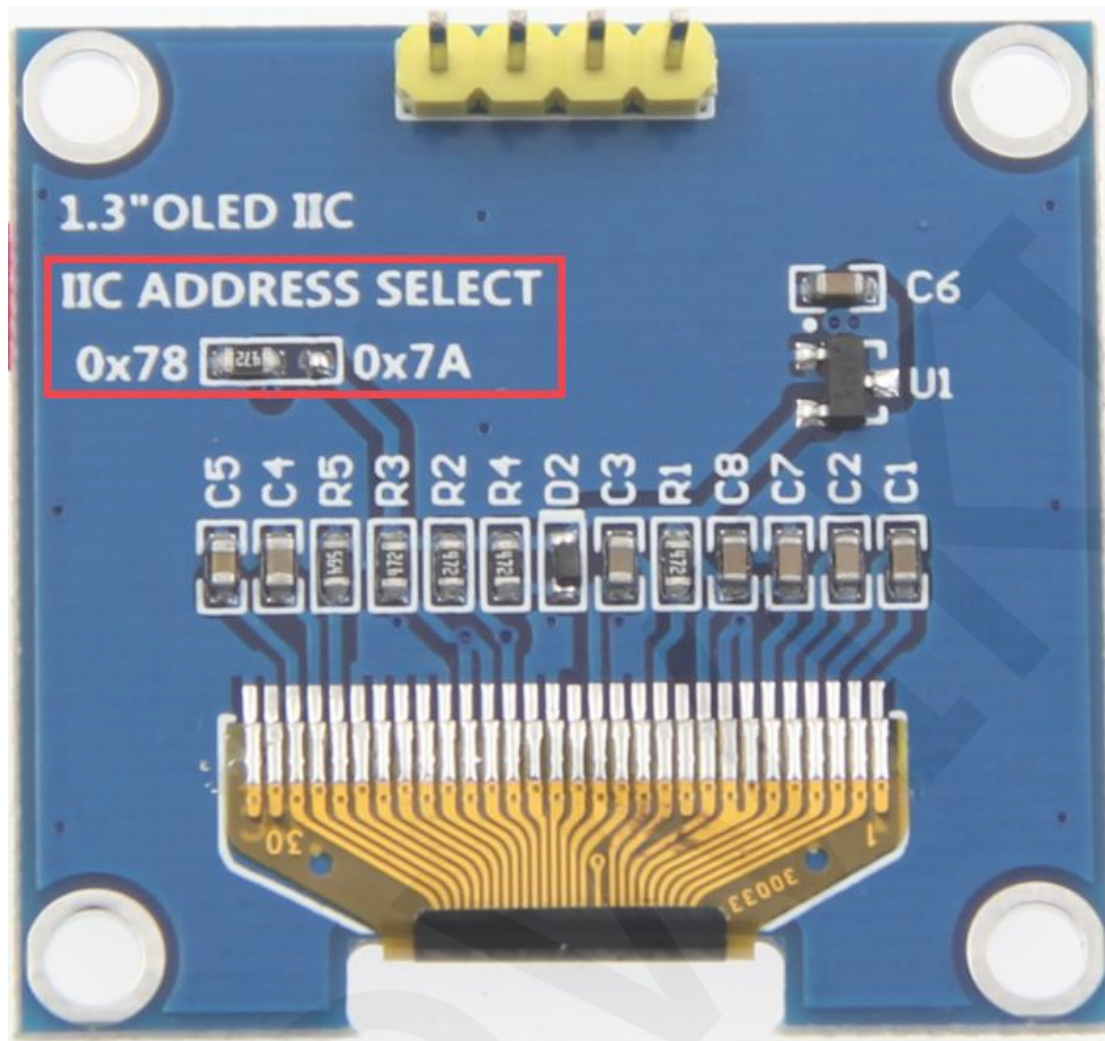


图4. 模块背面图

注意：

- 1、本模块支持IIC从设备地址切换（如图4红框内所示），具体说明如下：
 - A、焊接0x78一侧电阻，断开0x7A一侧，则选择0x78从设备地址（默认）；
 - B、焊接0x7A一侧电阻，断开0x78一侧，则选择0x7A从设备地址；
- 2、硬件切换了IIC从设备地址，软件上也要做相应的修改，具体修改方法见以下IIC从设备地址修改说明；

Raspberry Pi测试程序接线说明			
序号	模块引脚	对应开发板接线引脚	备注
1	GND	GND	OLED电源地

		(物理引脚: 6, 9, 14, 20, 25, 30, 34, 39)	
2	VCC	5V/3.3V (物理引脚: 1, 2, 4)	OLED电源正(3.3V~5V)
3	SCL	物理引脚: 5 BCM编码: 3 wiringPi编码: 9	OLED IIC总线时钟信号
4	SDA	物理引脚: 3 BCM编码: 2 wiringPi编码: 8	OLED IIC总线数据信号

注意:

物理引脚是指 **RaspBerry Pi** 开发板的 **GPIO** 引脚编码

BCM 编码是指使用 **BCM2835 GPIO** 库时的 **GPIO** 引脚编码

wiringPi 编码是指使用 **wiringPi GPIO** 库时的 **GPIO** 引脚编码

在代码里面使用哪个 **GPIO** 库, 引脚定义就需要使用相应的 **GPIO**

库编码, 详情见图 1 **GPIO map** 表

例程功能说明:

- 1、本套测试程序适用于Raspberry Pi 各个型号开发板;
- 2、本套测试程序包含使用bcm2835、wiringPi GPIO库的程序以及python程序, 具体的操作说明见模块用户使用说明文档;
- 3、本套测试程序使用RaspberryPi硬件IIC总线传输数据;
- 4、请选择相应测试程序和开发板按照上述接线说明进行接线;
- 5、本套测试程序包含如下测试项:
 - A、主界面显示测试;
 - B、简单的黑白刷屏测试;
 - C、矩形绘制及填充测试;
 - D、圆形绘制及填充测试;
 - E、三角形绘制及填充测试;

- F、英文显示测试;
- G、数字和符号显示测试;
- H、中文显示测试;
- I、BMP单色图片显示测试;
- J、菜单显示测试;

IIC从设备地址修改说明:

1、使用bcm2835、wiringPi GPIO库的程序IIC从设备地址修改

- A、打开程序source\include目录下的iic.h文件，找到以下内容:

```
#define IIC_SLAVE_ADDR 0x3C
```

- B、对IIC_SLAVE_ADDR宏定义进行修改(默认为0x3C(对应0x78)):

例如改为0x3D，那么IIC从设备地址就是0x3D(对应0x7A);

2、Python程序IIC从设备地址修改

- A、开程序source目录下的oled.py文件，找到以下内容

```
IIC_SLAVE_ADDR = 0x3C
```

- B、对IIC_SLAVE_ADDR宏定义进行修改(默认为0x3C(对应0x78)):

例如改为0x3D，那么IIC从设备地址就是0x3D(对应0x7A);