



# 1.3inch OLED HAT

## 用户手册

### 产品概述

本产品是 1.3 寸树莓派专用 OLED 显示屏，分辨率为 128\*64，带有内部控制器，可使用 I2C 或 SPI 进行控制，同时外扩了一个摇杆和三个按键，方便扩展应用。

### 产品特性

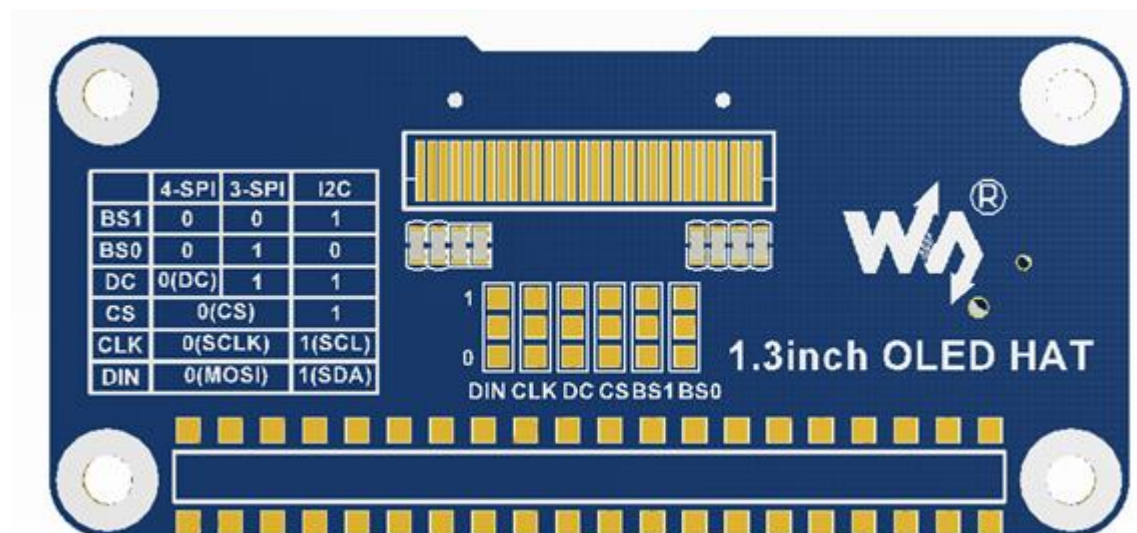
驱动芯片：SH1106  
 支持接口：3-wire SPI, 4-wire SPI, I2C  
 分辨率：128\*64  
 显示尺寸：1.3inch  
 外形尺寸：65mm\*30mm  
 显示颜色：蓝色  
 视角：>160°  
 工作温度：-20°C ~ 70°C  
 存储温度：-30°C ~ 80°C

### 管脚配置

功能引脚	树莓派接口 (BCM)	描述
KEY1	P21	按键 1GPIO
KEY2	P20	按键 2GPIO
KEY3	P16	按键 3GPIO
摇杆 Up	P6	摇杆上
摇杆 Down	P19	摇杆下
摇杆 Left	P5	摇杆左
摇杆 Right	P26	摇杆右
摇杆 Press	P13	摇杆按下
SCLK	P11/SCLK	SPI 时钟线
MOSI	P10/MOSI	SPI 数据线
SCL	P3/SCL1	I2C 时钟线
SDA	P2/SDA1	I2C 数据线
DC	P24	数据/命令线
CS	P8/CE0	片选
RST	P25	复位

## 硬件配置

OLED 模块提供三种驱动接口：分别为 3-wire SPI、4-wire SPI 和 I2C 接口，模块背面有六个可选择焊接的电阻，通过电子的选择来选择相应的通信方式，如图：



模块出厂默认使用 4 线 SPI 通信模式，即 BS0，BS1，DC，CS，CLK，DIN 默认接 0（1 和 0 不全代表电平，只是电阻接上或者接下的焊接方式，具体硬件链接见下表）：

注：上图为硬件上的焊接，下表为实际的硬件连接

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

具体硬件配置，如下：

**使用 4 线 SPI：**即出厂程序设置：BS0 与 BS1 接 0 连接到地，CS 接 0 连接到树莓派 CE0（使能管脚），D/C 接 0 连接到树莓派的 P24（数据/命令管脚），DIN 接 0 连接至树莓派 MOSI，CLK 接 0 连接至树莓派 SCLK；

**使用 3 线 SPI：**BS0 接 1 连接到 VCC，BS1 接 0 连接至 GND，CS 接 0 连接至树莓派 CE0，D/C 接 1 连接值 GND，DIN 接 0 连接至树莓派 MOSI，CLK 接 0 来连接至树莓派 SCLK；

**使用 I2C：**BS0 接 0 连接至 GND，BS1 接 1 连接至 VCC（3.3V），CS 接 1 连接 GND，D/C 接 1 来连接至 GND，DIN 接 1 连接至树莓派 SDA，CLK 接 1 连接至树莓派 SCL；使用 I2C 时：DC 的高低状态可以控制从设备的地址，这里接到了 GND，那么 I2C 的 7 为地址为：0x3C

## 实验演示

本模块提供 BCM2835, WiringPi 以及 python 例程, 并实现常用屏幕操作功能: 画点、画线、画矩形, 画圆及他们的大小, 宽度。填充、显示英文字符并且提供 5 种常见字体, 以及画图等功能。

为了方便您的使用, 这里对例程使用进行了说明。

### 1. 树莓派开启 SPI 功能

```
sudo raspi-config
```

选择 Advanced Options -> SPI -> yes

启动 SPI 内核驱动

### 2. 函数库的安装

关于树莓派函数库的安装详细见微雪课堂:

<http://www.waveshare.net/study/article-742-1.html>

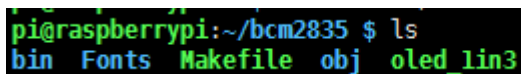
此处详细介绍了 WiringPi、bcm2835、python 的安装

### 3. 使用

对于 BCM2835 与 WiringPi 而言只需要把对应的例程拷贝进树莓派中(可通过 samba 或者直接复制到 SD)即可, 以下示例程序均复制到了树莓派 pi 用户目录下。

#### 3.1 BCM2835 使用

运行 ls 命令, 可见如下文件



```
pi@raspberrypi:~/bcm2835 $ ls
bin  Fonts  Makefile  obj  oled_lin3
```

其中: bin 文件夹中为项目生成的.o 文件, 通常情况下我们是不需要管的:

Fonts 文件夹中为 5 种常见字体。

Obj 文件夹中为项目文件, 其中有 mian.c, OLED\_Driver.c 及.h, DEV\_Config.c 及.h, 以及 OLED\_GUI.c 及.h

**main.c**: 主函数。需要注意的是, 虽然定义了 OLED\_ScanDir, 这个是控制扫描方向的, 但是此模块为树莓派专用, 同时也为了程序的兼容性, 此处虽然定义了函数, 但是不会影响扫描方式。

**DEV\_Config.c**：定义了树莓派的管脚及通信方式，若按照前面的硬件配置去更改了通信方式，只需要修改.h 文件的 USE\_SPI 与 USE\_IIC 这两个宏的相关定义；

**OLED\_Driver.c**：OLED 的驱动，通常情况下不需要做修改

**OLED\_GUI.c**：常用的画点，线，图，字函数，通常情况下你只需要修改 GUI\_Show()这个函数，这个函数为显示调用函数。

**Makefile**：工程的编译规则，如果更改了代码，需要先执行 make clear 清楚全部文件依赖以及产生的可执行文件，然后再执行 make，这样 makefile 就会自动编译整个项目，生成可执行文件。

**oled\_lin3**：可执行文件，通过 make 命令生成

用户使用的时候只需要执行 `sudo ./oled_lin3` 执行程序即可

### 3.2 WiringPi 使用

运行 ls 命令，可见如下文件

```
pi@raspberrypi:~/wiringPi $ ls
bin  Fonts  Makefile  obj  oled_lin3
```

WiringPi 与 BCM2835 文件目录相同，区别在两点：

一：WiringPi 是通过读取 Linux 系统的设备文件操作，而 bcm2835 则是树莓派 cpu 芯片的库函数，操作的是寄存器。因此如果先使用了 BCM2835 库，wiringpi 则会使用失败，此时重启系统可以运行；

二：由于第一个区别，他们的底层的配置不一样，在 DEV\_Config.c 中使用的为 WiringPi 及其相应的 wiringPiSPI 来提供底层接口

同样，只需运行 `sudo ./oled_lin3` 运行程序即可

### 3.3 Python

对于 python 的使用还需要装 OLED 驱动库，有两个库可以用：

1. Adafruit\_Python\_SSD1306 库->只支持 SSD1306
2. Luma.oled 库->支持 SSD1306/SSD1322/SSD1325/SSD1331/SH1106

因此我们需要安装 luma.oled 库，依次执行：

```
sudo apt-get install python-dev python-pip libfreetype6-dev libjpeg-dev
sudo -H pip install --upgrade pip
sudo apt-get purge python-pip
sudo -H pip install --upgrade luma.oled
```

注：如果安装 Luma.oled 库时红字错误，请继续执行命令重试，那是因为网络问题下载一个叫 Pillow 的库不成功

执行 `sudo python demo.py` 运行程序

### 3.4 开机自动运行

通过配置 `/etc/rc.local`，使代码在树莓派启动时运行

运行：

```
sudo vim /etc/rc.local
```

在 `exit 0` 前加上：

```
sudo python /home/pi/python/demo.py &
```

需要注意的是：`/home/pi/python/demo.py` 为放置例程所在目录位置，可以通过命令：`pwd` 来获取

还有务必在结尾加上 `&`，否则您可能需要重装系统（无法通过 `ctrl+c` 终止进程，无法登录到树莓派用户下）。