

1.3inch LCD HAT

来自Waveshare Wiki

跳转至： [导航](#)、 [搜索](#)

说明

产品简介

产品参数

- 工作电压： 3.3V
- 通信接口： SPI
- 屏幕类型： IPS
- 控制芯片： ST7789VM
- 分辨率： 240(H)RGB x 240(V)
- 显示尺寸： 23.4 (H) x 23.4 (V) mm
- 像素大小： 0.0975 (H) x 0.0975 (V) mm
- 产品尺寸： 65 x 30.2(mm)



功能引脚

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
CS	P8/CE0	片选
DC	P25	数据/命令选择
RST	P27	复位
BL	P24	背光

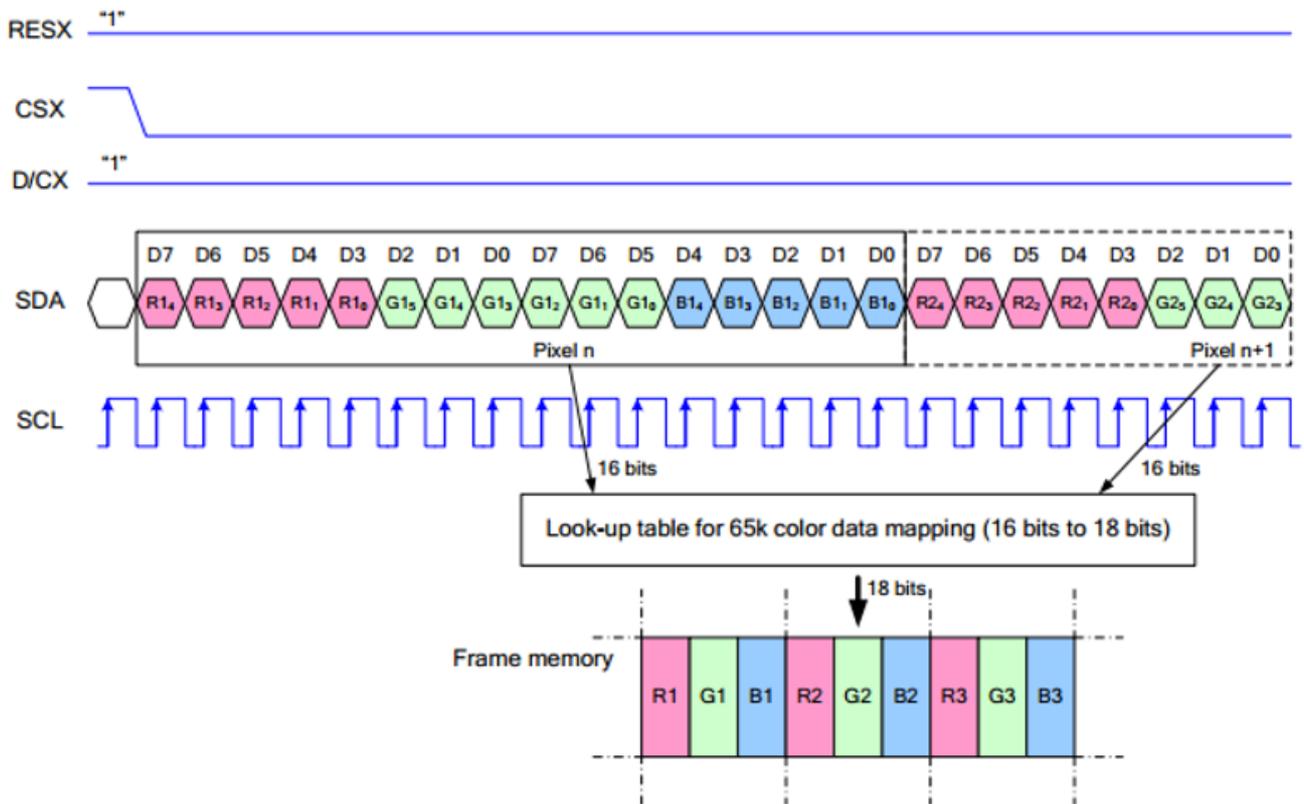
LCD 及其控制器

本款LCD使用的内置控制器为ST7789VM，是一款240 x RGB x 320像素的LCD控制器,而本LCD本身的像素为240(H)RGB x 240(V),同时由于初始化控制可以初始化为横屏和竖屏两种，因此LCD的内部RAM并未完全使用。

该LCD支持12位，16位以及18位每像素的输入颜色格式，即RGB444，RGB565，RGB666三种颜色格式，本例程使用RGB565的颜色格式，这也是常用的RGB格式

LCD使用四线SPI通信接口，这样可以大大的节省GPIO口，同时通信是速度也会比较快

通信协议



注：与传统的SPI协议不同的地方是：由于是只需要显示，故而将从机发往主机的数据线进行了隐藏，该表格详见Datasheet Page 66。

RESX为复位，模块上电时拉低，通常情况下置1；

CSX为从机片选，仅当CS为低电平时，芯片才会被使能。

D/CX为芯片的数据/命令控制引脚，当DC = 0时写命令，当DC = 1时写数据

SDA为传输的数据，即RGB数据；

SCL为SPI通信时钟。

对于SPI通信而言，数据是有传输时序的，即时钟相位（CPHA）与时钟极性(CPOL)的组合：

CPHA的高低决定串行同步时钟是在第一时钟跳变沿还是第二个时钟跳变沿数据被采集，当

CPHA = 0，在第一个跳变沿进行数据采集；

CPOL的高低决定串行同步时钟的空闲状态电平, CPOL = 0, 为低电平。

从图中可以看出, 当SCLK第一个下降沿时开始传输数据, 一个时钟周期传输8bit数据, 使用SPI0, 按位传输,高位在前,低位在后。

树莓派使用

提供C语言与python例程

树莓派使用教程

开启SPI接口

PS: 如果使用的是Bullseye分支的系统,需要将" apt-get "改成 "apt", Bullseye分支的系统只支持Python3。

- 打开树莓派终端, 输入以下指令进入配置界面

```
sudo raspi-config  
选择Interfacing Options -> SPI -> Yes 开启SPI接口
```

```
1 Change User Password Change password for the current user
2 Network Options      Configure network settings
3 Boot Options         Configure options for start-up
4 Localisation Options Set up language and regional settings to match your location
5 Interfacing Options  Configure connections to peripherals
6 Overclock            Configure overclocking for your Pi
7 Advanced Options     Configure advanced settings
8 Update               Update this tool to the latest version
9 About raspi-config  Information about this configuration tool
```

```
P1 Camera      Enable/Disable connection to the Raspberry Pi Camera
P2 SSH         Enable/Disable remote command line access to your Pi using SSH
P3 VNC         Enable/Disable graphical remote access to your Pi using RealVNC
P4 SPI         Enable/Disable automatic loading of SPI kernel module
P5 I2C         Enable/Disable automatic loading of I2C kernel module
P6 Serial      Enable/Disable shell and kernel messages on the serial connection
P7 1-Wire      Enable/Disable one-wire interface
P8 Remote GPIO Enable/Disable remote access to GPIO pins
```

Would you like the SPI interface to be enabled?

<Yes>

<No>

然后重启树莓派：

```
sudo reboot
```

请确保SPI没有被其他的设备占用，你可以在/boot/config.txt中间检查

安装库

BCM2835

```
#打开树莓派终端，并运行以下指令
wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.71.tar.gz
tar zxvf bcm2835-1.71.tar.gz
cd bcm2835-1.71/
sudo ./configure && sudo make && sudo make check && sudo make install
# 更多的可以参考官网：http://www.airspayce.com/mikem/bcm2835/
```

```
#打开树莓派终端，并运行以下指令
sudo apt-get install wiringpi
#对于树莓派2019年5月之后的系统（早于之前的可不用执行），可能需要进行升级：
wget https://project-downloads.drogon.net/wiringpi-latest.deb
sudo dpkg -i wiringpi-latest.deb
gpio -v
# 运行gpio -v会出现2.52版本，如果没有出现说明安装出错

#Bullseye分支系统使用如下命令：
git clone https://github.com/WiringPi/WiringPi
cd WiringPi
./build
gpio -v
# 运行gpio -v会出现2.60版本，如果没有出现说明安装出错
```

■ python

```
#Python2
sudo apt-get update
sudo apt-get install ttf-wqy-zenhei
sudo apt-get install python-pip
sudo pip install RPi.GPIO
sudo pip install spidev
#Python3
sudo apt-get update
sudo apt-get install ttf-wqy-zenhei
sudo apt-get install python3-pip
sudo pip3 install RPi.GPIO
sudo pip3 install spidev
```

下载例程

在树莓派终端运行：

```
sudo apt-get install p7zip-full -y
wget http://www.waveshare.net/w/upload/b/bd/1.3inch_LCD_HAT_code.7z
7z x 1.3inch_LCD_HAT_code.7z -r -o./1.3inch_LCD_HAT_code
sudo chmod 777 -R 1.3inch_LCD_HAT_code
cd 1.3inch_LCD_HAT_code
```

运行测试程序

■ C语言

```
cd c
make clean
make
sudo ./main
```

- 对于树莓派4B及raspbian_lite-2019-06-20系统之后需要设置如下，按键才能正常输入

```
sudo nano /boot/config.txt
#添加如下:
gpio=6,19,5,26,13,21,20,16=pu
```

■ python

```
cd python
sudo python main.py
sudo python key_demo.py
```

FBCP移植

Framebuffer 是用一个视频输出设备从包含完整的帧数据的一个内存缓冲区中来驱动一个视频显示设备。简单的来说，就是使用一个内存区来存储显示内容，改变内存的数据就可以改变显示的内容。

在 github 上有一个开源工程：fbcp-ili9341，该工程相比其他类fbcp工程，利用了局部刷新和DMA实现高达60fps的速度

编译与运行

```
cd ~
sudo apt-get install cmake -y
sudo apt-get install p7zip-full -y
wget https://www.waveshare.net/w/upload/f/f9/Waveshare_fbcf.7z
7z x Waveshare_fbcf.7z -o./waveshare_fbcf
cd waveshare_fbcf
mkdir build
cd build
#以1.44inch_LCD_HAT为例
cmake -DSPI_BUS_CLOCK_DIVISOR=20 -DWAVESHARE_1INCH44_LCD_HAT=ON -DBACKLIGHT_CONTROL=ON -DSTATISTICS=0 ..
#以1.3inch_LCD_HAT为例
cmake -DSPI_BUS_CLOCK_DIVISOR=20 -DWAVESHARE_1INCH3_LCD_HAT=ON -DBACKLIGHT_CONTROL=ON -DSTATISTICS=0 ..
make -j
sudo ./fbcf
```

设置开机自启动

```
sudo cp ~/waveshare_fbcf/build/fbcf /usr/local/bin/fbcf
sudo nano /etc/rc.local
```

在 `exit 0` 前面添加 `fbcf&`。注意一定要添加"&" 后台运行，否则可能会出现系统不能启动的情况。

```
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

fbcf&
exit 0
```

设置用户界面显示尺寸

在 `/boot/config.txt` 文件中设置用户界面显示尺寸。

```
sudo nano /boot/config.txt
```

添加

```
hdmi_force_hotplug=1  
hdmi_cvt=300 300 60 1 0 0 0  
hdmi_group=2  
hdmi_mode=87  
display_rotate=0
```

这里是设置系统界面分辨率，最后显示的效果是按照比例缩放显示在1.3inch LCD上。这里设置分辨率应稍大于LCD的分辨率，过高会导致字体显示模糊。

重启设备即可

```
sudo reboot
```

启动树莓派后可以发现屏幕已经出现用户界面了。



模拟鼠标

模块上面板子一个摇杆和三个按键，我们可以用来控制树莓派的鼠标

- 安装库并下载并运行例程

```
#python2
sudo apt-get install python-xlib
sudo pip install PyMouse
wget http://www.waveshare.net/w/upload/d/d3/Mouse.7z
7z x Mouse.7z
sudo python mouse.py

#python3
sudo apt-get install python3-xlib
sudo pip3 install PyMouse
sudo pip3 install unix
sudo pip3 install PyUserInput
wget http://www.waveshare.net/w/upload/d/d3/Mouse.7z
7z x Mouse.7z
sudo python3 mouse.py
```

- 注意：mouse.py需要的图形化界面下运行,SSH登录命令行无法使用，可以直接跳过这一步骤，通过开机自启动运行程序。
- 对于树莓派4B及raspbian_lite-2019-06-20系统之后需要设置如下，按键才能正常输入

```
sudo nano /boot/config.txt
#添加如下：
gpio=6,19,5,26,13,21,20,16=pu
```

使用摇杆上下左右，就可以看到鼠标在动了。

- 配置成开机自启

注意不要加到/etc/rc.local中，因为rc.local在系统还没有进入桌面之前会执行，而PyMouse模块在命令行界面运行的话会报错没有鼠标这个事件，所以我们需要执行如下：

```
cd .config/
mkdir autostart
cd autostart/
sudo nano local.desktop
```

加入如下(添加对应的即可，不用全部添加进去)：

```
#python2
[Desktop Entry]
Type=Application
Exec=python /home/pi/mouse.py
#python3
[Desktop Entry]
Type=Application
Exec=python3 /home/pi/mouse.py
```

然后重新启动树莓派，即可使用按键控制鼠标了。

```
sudo reboot
```

其他

有兴趣弄游戏机的小伙伴们可以参考下面这两个链接,这里不再详细讲解。

<https://www.sudomod.com/forum/viewtopic.php?f=11&t=5371&start=10>

<https://pi0cket.com/guides/tiny-software-for-tinypi/#more-99>

资料

文档

- 原理图

预安装镜像

链接: https://pan.baidu.com/s/1S_D-TyvWgP6vH38OBNfW4Q

提取码: wxdz

程序

- 示例程序
- 摇杆模拟鼠标程序

软件

- 汉字取模软件
- Image2Lcd 图片取模软件

数据手册

- ST7789 数据手册

相关资料

- 图形界面(客户作品)
- 外壳图纸(PiSugar)

FAQ

问题： 设置fbtft没有识别到fb1,或者屏幕不显示？

这里是两行语句，不能分开成多行。

```
options fbtft_device name=flexfb gpios=reset:27,dc:25,cs:8,led:
24 speed=40000000 bgr=1 fps=60 custom=1 height=240 width=240
options flexfb setaddrwin=0 width=240 height=240 init=-1,0x11,-2,120,-1,0x36,0x70,-
1,0x3A,0x05,-1,0xB2,0x0C,0x0C,0x00,0x33,0x33,-1,0xB7,0x35,-1,0xBB,0x1A,-1,0xC0,0x2C,-
1,0xC2,0x01,-1,0xC3,0x0B,-1,0xC4,0x20,-1,0xC6,0x0F,-1,0xD0,0xA4,0xA1,-1,0x21,-
1,0xE0,0x00,0x19,0x1E,0x0A,0x09,0x15,0x3D,0x44,0x51,0x12,0x03,0x00,0x3F,0x3F,-
1,0xE1,0x00,0x18,0x1E,0x0A,0x09,0x25,0x3F,0x43,0x52,0x33,0x03,0x00,0x3F,0x3F,-1,0x29,-3
```

这里需要注意一下·这里是以 options 开头的两行语句。

问题： 设置fbtft后为什么程序控制了，报错没有SPI设备

这是两种控制方式，设置为fbtft方式控制后，SPI被系统占用了，不能再通过操作IO控制屏幕。

需要屏蔽/etc/module中的把

```
flexfb
fbtft_device
```

改成

```
#flexfb  
#fbtft_device
```

然后重启树莓派

问题： 如何播放视频

播放视频必须先设置fbtft，并且通过f设置fbtft显示桌面后，再运行播放视频命令才会有显示