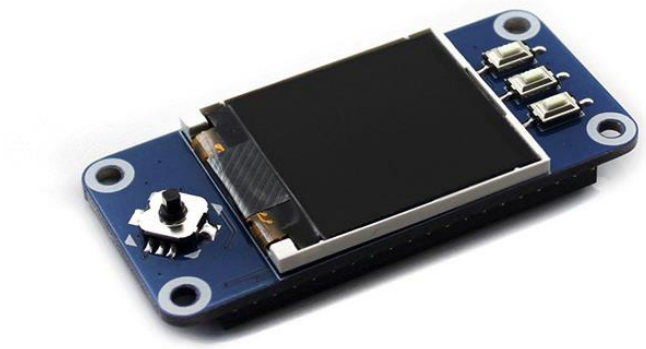
树莓派驱动 1.44inch LCD HAT

-FrameBuffer 驱动，使用 fbtft

自 1.44inch LCD HAT 这个迷你的屏幕上架后深受很多创客门喜爱，同时也有多客户反应我们提供的资料不够完善。特此写一篇教程讲解树莓派如何通过移植 fbtft 显示.



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: https://github.com/notro/fbtft, 完整的实现了 framebuffer 驱动，让树莓派完美支持 tft 液晶。下面来介绍一下如何使用 fbtft 驱动 1.44inch LCD HAT.

打开编辑配置文件，启用一些模块。

```
sudo nano /etc/modules
```

在文件后面添加如下两个语句，第一行是确保屏幕的 SPI 已经启动并正在运行，第二个命令实际是启动 fbtft 模块。

```
spi-bcm2835
fbtft_device
```

新建另外一个配置文件，配置 fbtft

```
sudo nano /etc/modprobe.d/fbtft.conf
```

将下面语句添加到新建的空白文件中，

```
# /etc/modprobe.d/fbtft.conf
options fbtft_device name=adafruit18_green
gpios=reset:27,dc:25,cs:8,led:24 speed=40000000 bgr=1 fps=60
custom=1 height=128 width=128 rotate=180
```

这里需要注意一下，name 需要根据屏幕的主控芯片型号选择，1.44inch LCD HAT 的主控芯片是 ST7735s。fbtft 是支持这个型号的。这里选择 adafruit18_green，因为芯片是和这个一样的。

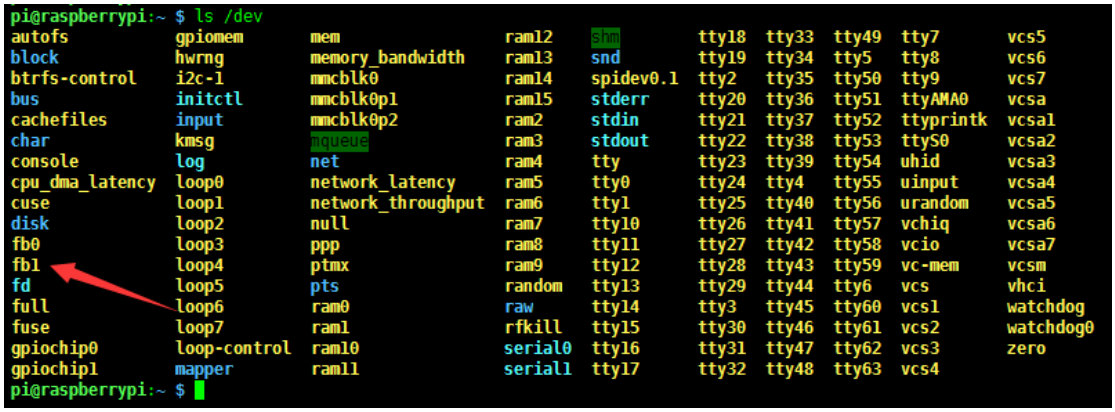
gpios=reset:27,dc:25,cs:8,led:24 这个设置屏幕对应的引脚, 这个根据板子的原理图设置。

height=128 width=128 rotate=180 设置屏幕分辨率大小和显示方向。

此时重启一下树莓派.如果屏幕显示全黑则屏幕已经工作了。

```
sudo reboot
```

查看设备可以发现多了一个 fb1 设备，则说明设备已经成功启动了

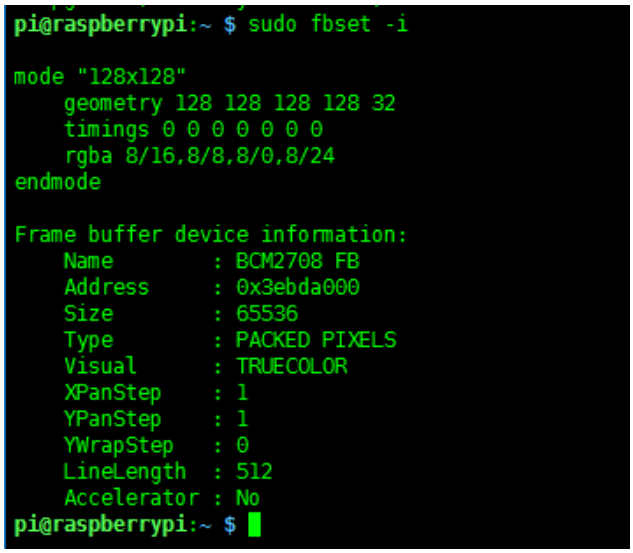


用 fbset 命令查看 fbtft 设备信息

```
sudo fbset -i
```

如果命令出错可以运行如下命令安装

```
sudo apt-get install fbset
```



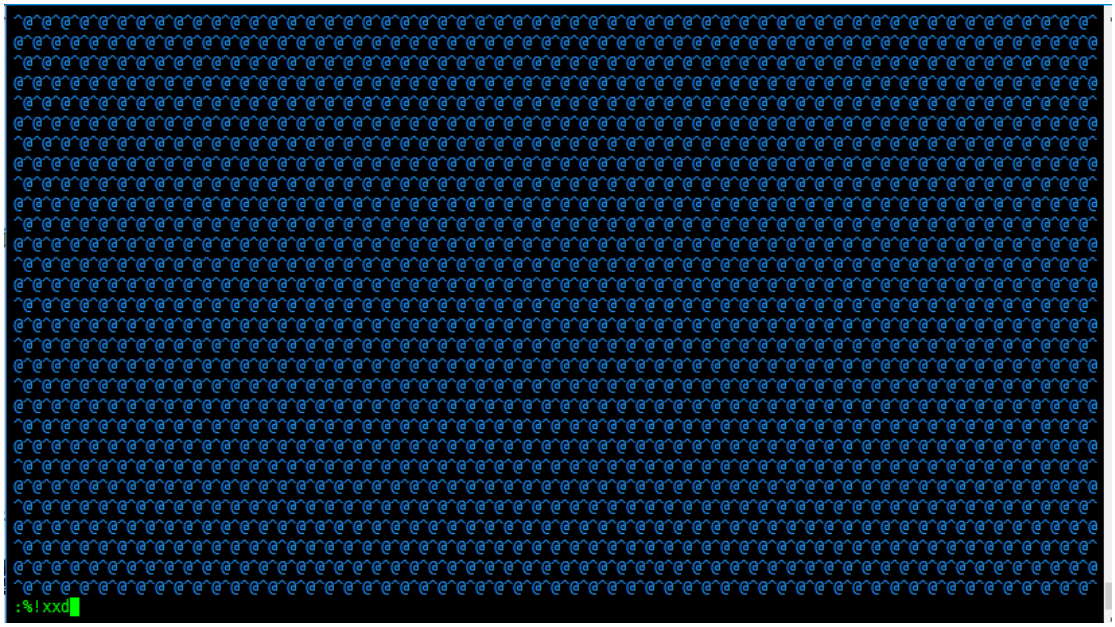
可以看到屏幕是 128x128,但是其他信息貌似是错误的，例如 Size 实测是 32768 而不是 65536.

可以使用如下命令读取内存区。

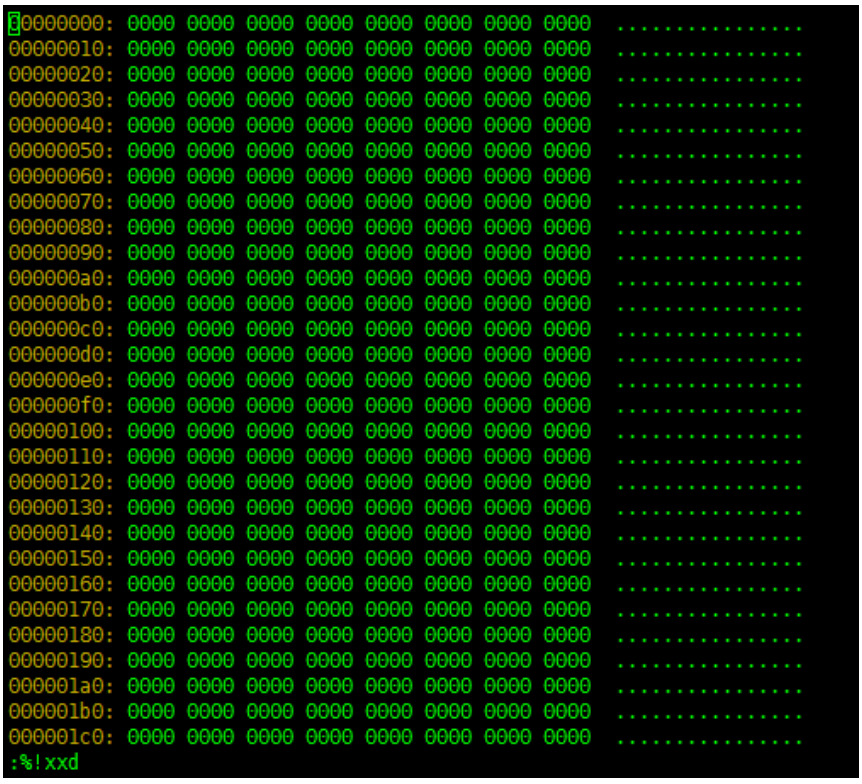
```
sudo cat /dev/fb1 > fb.fb
```

运行 ls -l 命令可以看到 fb.fb 文件的大小为 32768

```
vi -b fb.fb
```



输入底行命令将格式转换为 16 进制：:%!xxd



可以看到现在内存全部为 0，显示全黑。下面将最前面一行改为 FF.

编辑完成后不要忘记转换为二进制文件保存

```
:%!xxd -r

:wq
```

将修改后的内容显示到屏幕上。

```
sudo cat fb.fb > /dev/fb1
```

此时可以看到屏幕左下角有几个白点了。到这里可能已经明白 FrameBuffer 的作用了。FrameBuffer 就是用一块内存表示屏幕显示的内容，可以读取显示的内容，修改内存的内容可以改变屏幕显示的信息。

显示一张图片

下面我们来显示一张图片。首先安装 python 的 PIL 库

```
sudo apt-get install python-imaging
```

示例程序:

```
#!/usr/bin/env python2
import os
import struct
from PIL import Image

im = Image.open('time.bmp')
im = im.rotate(270)

w, h = im.size
print( "w:", w ,"h:", h)

with open('/dev/fb1', 'wb') as f:
    for j in range(0,h):
        for i in range(0,w):
            r,g,b =im.getpixel((i,j))
            rgb=struct.pack('H',((r >> 3) << 11)|((g >> 2) << 5)|(b >> 3))
            f.write(rgb);
```

程序读取一张图片，然后将图片转为内存数据并写入到/dev/fb1 设备中。这里需要注意的是屏幕是两个字节表示一个像素点，RGB565 的格式显示。

上面程序刷新的时候会看到屏幕是一行行显示的，刷新效果不是很好，下面将程序修改一下，先将要显示的内容保存到 fb 文件中，再用 cat 命令显示出来。

```
#!/usr/bin/env python2
import os
import struct
from PIL import Image

im = Image.open('time.bmp')
im = im.rotate(270)

w, h = im.size
print( "w:", w ,"h:", h)

with open('time.fb', 'wb') as f:
    for j in range(0,h):
        for i in range(0,w):
            r,g,b =im.getpixel((i,j))
            rgb=struct.pack('H',((r >> 3) << 11)|((g >> 2) << 5)|(b >> 3))
            f.write(rgb);
os.system('cat time.fb > /dev/fb1')
```

最后上传两个程序的压缩文件。

http://www.waveshare.net/w/upload/f/fd/Fbtft.tar.gz

显示用户界面

最后我们将用户界面到 1.44inch LCD HAT 上，虽然这个屏幕只有 128x128 分辨率，我们还是试一下将用户界面显示到屏幕上看下有什么样的效果。

显示用户界面只需将 fb0 上的内容直接拷贝到 fb1 上，fb0 和 fb1 同步。

首先安装一下工具软件

```
sudo apt-get install cmake git
```

使用 github 上的开源代码来实现这个功能，下载代码并安装。

```
cd ~
git clone https://github.com/tasanakorn/rpi-fbcp
cd rpi-fbcp/
mkdir build
cd build/
cmake ..
make
sudo install fbcp /usr/local/bin/fbcp
```

设置开机启动。在

```
sudo nano /etc/rc.local
```

设置开机启动。在 exit 0 前面添加 fbcp&.

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

fbcp&
exit 0
```

最后在/boot/config.txt 文件中设置用户界面显示尺寸。

```
sudo vi /boot/config.txt
```

在文件最后面添加上

```
hdmi_force_hotplug = 1
hdmi_cvt = 128 128 60 1 0 0 0
hdmi_group = 2
hdmi_mode = 1
hdmi_mode = 87
display_rotate = 1
```

启动树莓派后可以发现屏幕已经出现用户界面了。最后显示效果图。

显示屏幕常亮

打开 lightdm.conf

```
sudo vi /etc/lightdm/lightdm.conf
```

修改 lightdm.conf

找到[SeatDefaults]段下的'xserver-command',取消注释,修改为如下:

```
#xserver-command=X
```

修改为

```
xserver-command=X -s 0 -dpms
```

- -s # –设置屏幕保护不启用
- dpms 关闭电源节能管理

重启

```
sudo reboot
```

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