

树莓派环境搭建手册
AUTHOR:SHENHENGHENG

2017 年 12 月 09 日

目 录

第一章 系统准备	1
1.1 商品清单	1
1.1.1 树莓派 3 B+	1
1.1.2 无线网卡（树莓派使用）	2
1.1.3 树莓派摄像头模块（建议）	3
1.1.4 内存卡（不建议网上购买）	3
1.1.5 其他	3
1.2 烧录树莓派	4
1.2.1 软件准备	4
1.2.2 步骤	4
第二章 首次登录系统并进行必要的配置	7
2.1 必要的商品清单	7
2.1.1 hdmi 转 vga 转接头（可选）	7
2.1.2 其他	7
2.2 登录系统（不使用外接显示器）	7
2.3 额外的技巧	11
第三章 配置 AP ^[2]	12
3.1 配置流程	12
3.1.1 必需的包	12
3.1.2 配置网卡	12
3.1.3 配置 HOSTAPD 文件	12
3.1.4 配置 DNSMASQ	14
3.1.5 建立 IPV4 转发	15
3.1.6 我们已经完成了	15
3.2 注意事项	16

第四章 搭建 OpenCV3 环境 ^[3]	17
4.1 扩大文件系统	17
4.2 安装依赖	17
4.3 下载 OpenCV 源代码	18
4.4 Python2.7 还是 Python3.5	18
4.4.1 安装 Numpy	19
4.5 编译安装 OpenCV	19
第五章 如何初始化 HypriotOS	21
5.1 安装 HypriotOS	21
5.2 重要：如何进行 headless 配置	21
5.3 接下来	25
第六章 搭建机器学习与深度学习环境	26
6.1 搭建 TensorFlow 环境 ^[4]	26
6.2 安装 keras	26
参考文献	27

第一章 系统准备

1.1 商品清单

1.1.1 树莓派 3 B+

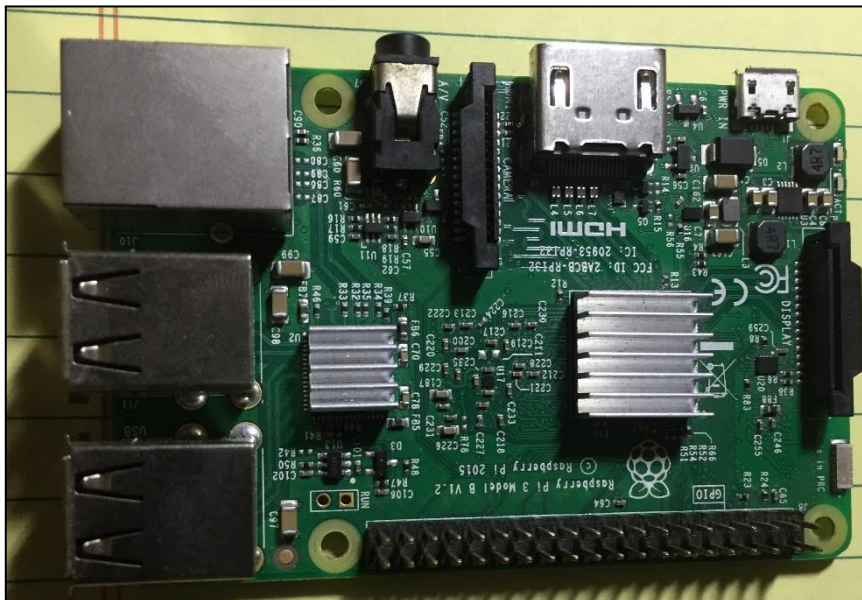


图 1-1 树莓派 3 B+正面



图 1-2 树莓派 3 B+侧面

推荐购买渠道（链接）：[淘宝](#)



图 1-3 购买页（裸机）

1.1.2 无线网卡（树莓派使用）



图 1-4 树莓派专用无线网卡



图 1-5 购买页（无线网卡）

推荐购买渠道（链接）：[淘宝](https://www.taobao.com)

1.1.3 树莓派摄像头模块（建议）

树莓派摄像头模块在项目中使用到，但后期发现树莓派摄像头模块可以用普通的 USB 摄像头代替。（可选，建议购买，帧稳定）



图 1-6 购买页（摄像头模块）

推荐购买渠道（链接）：[淘宝](#)

1.1.4 内存卡（不建议网上购买）



图 1-7 购买页（TF 小卡）

16G 够用。推荐电子商店购买，淘宝购买渠道（链接）：[淘宝](#)

1.1.5 其他

比如还需要准备的材料有数据线（USB 数据线 -> 安卓数据线），读卡器，电源（有充电宝用充电宝代替）。

1.2 烧录树莓派

1.2.1 软件准备

1) win32diskimager (开源软件): [下载地址](#)

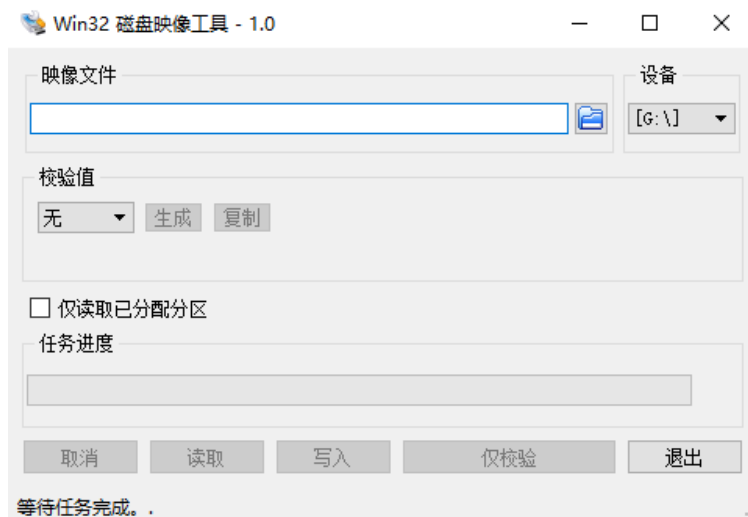


图 1-8 软件打开之后的样子

2) 树莓派系统 (很多, 这里选择 **RASPBIAN**): [下载地址](#)

1.2.2 步骤

1) 打开 win32diskimager -> 选中下载后的树莓派系统镜像文件 -> 设备选择你的读卡器 (盘符)

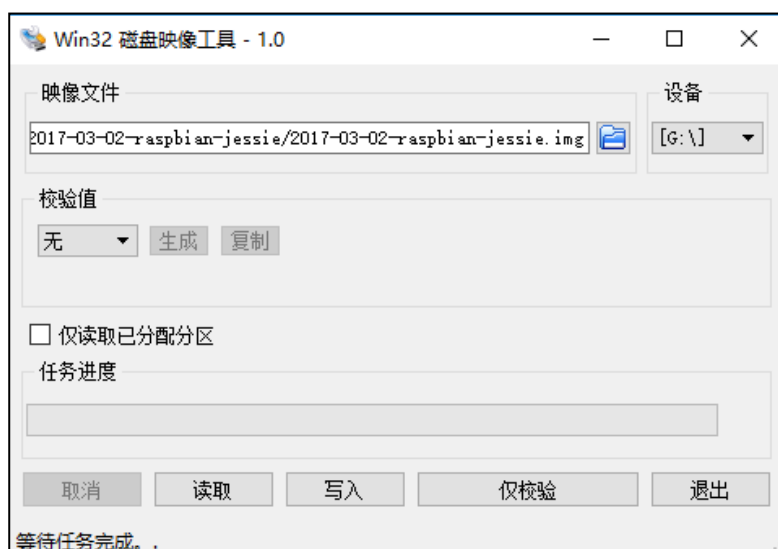


图 1-9 选中镜像和设备

2) 点击写入按钮, 将会对设备进行写入

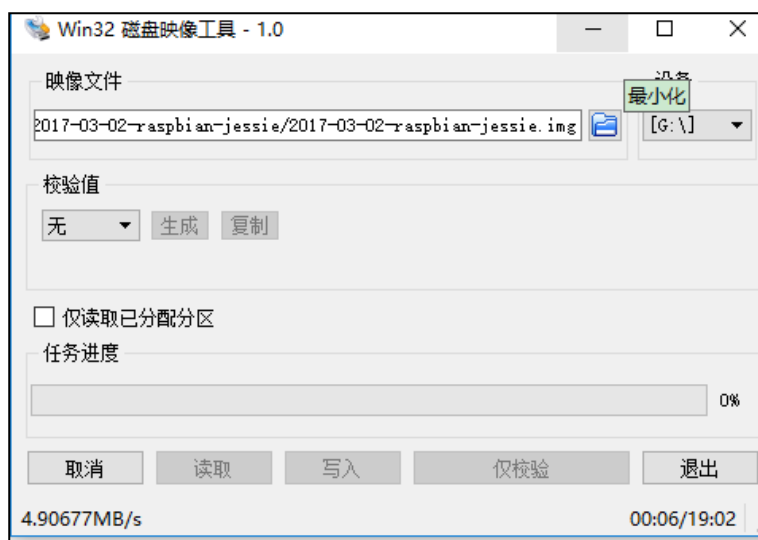


图 1-10 对设备进行写入操作

3) 写入完成后 -> 对刷入的系统进行测试 -> 将内存卡插入在树莓派上并试图点亮树莓派。(点亮即成功)

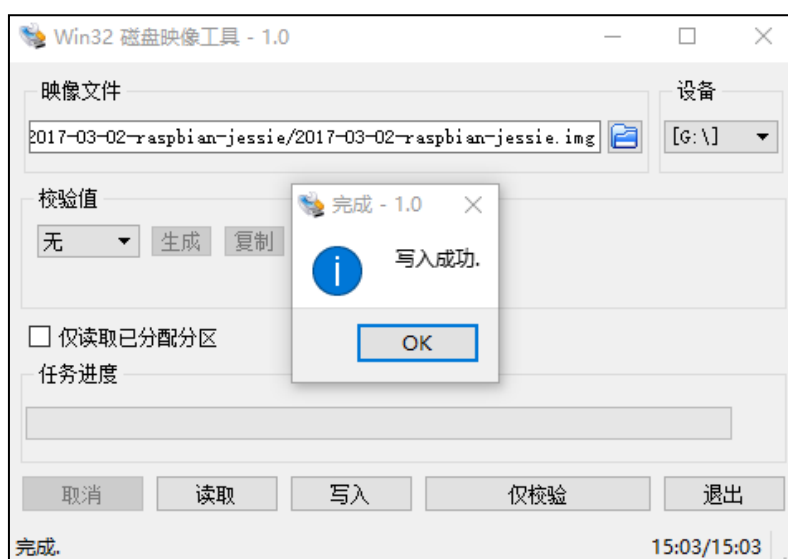


图 1-11 设备写入成功

通过图 1-12 所示，树莓派的连接方式如下图。正常的点亮方式是红灯信号灯常亮，绿色信号灯闪烁。

NOTE: 但最新的 **RASPBIAN** 系统默认刷入系统时不支持 SSH 服务，也就是说为了安全起见，树莓派不支持直接通过 SSH 登录，需要连接显示器来打开 SSH 服务，才能进行远程登录。



图 1-12 连接方式

第二章 首次登录系统并进行必要的配置

2.1 必要的商品清单

2.1.1 hdmi 转 vga 转接头（可选）



图 2-1 购买页（hdml 转 vga 转接头）

淘宝购买渠道（链接）：[淘宝](#)

2.1.2 其他

需要网线一根，台式电脑显示器（配有连接线）

2.2 登录系统（不使用外接显示器）

使用外接显示器相对比较简单，在这里省略不写。

从 11 月开始，SSH 服务被禁用。所以我们通过在 BOOT 磁盘（也就是你的内存卡）的 `/mnt/sdc1/` 文件夹中创建一个名为 `ssh` 的空文件来启用它。（即 `/mnt/sdc1/ssh`）。^[1]

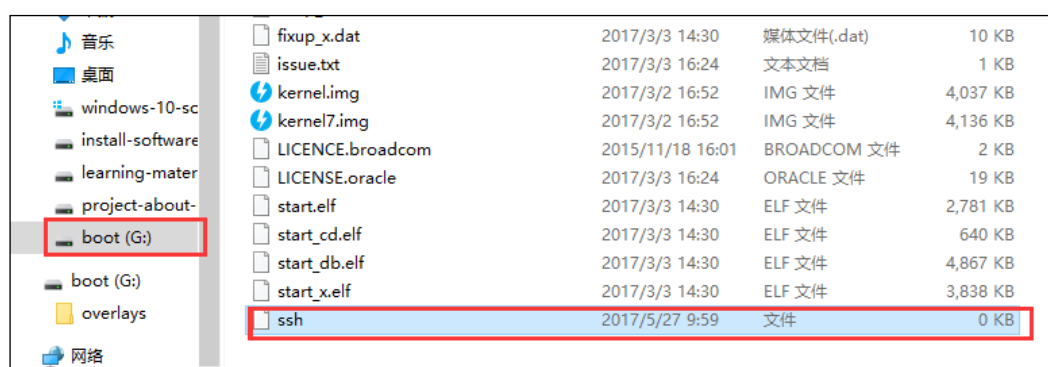


图 2-2 在内存卡的根目录下创建一个空的文件 ssh

这样的话，SSH 服务被启用了，节省了很多不必要的资源。通过网线连接到路由器上，

组成局域网，查看路由器的管理界面找到树莓派的 IP 地址。如图所示。

DHCP 客户端列表							
序号	IP 地址	MAC 地址	主机名	类型	状态	租约到期时间	操作
11	192.168.1.40	B4-0B-44-0D-C5-CC	Smartisan-T1	分配	离线	2017-05-28 07:54:15	🔍 ⏴
12	192.168.1.41	F4-8B-32-EB-32-D0	MI4LTE-xiaomish	分配	离线	2017-05-28 08:42:38	🔍 ⏴
13	192.168.1.44	B8-27-EB-FF-20-C3	raspberrypi	分配	在线	2017-05-28 10:02:45	🔍 ⏴
14	192.168.1.63	DC-2B-2A-B1-FF-EA	shinhiskiiPhone	分配	离线	2017-05-28 09:42:22	🔍 ⏴

图 2-3 查看路由器管理，找到树莓派的 IP 地址

找到 IP 之后，就可以在电脑端打开 SSH 客户端 Putty, XShell, SecureCRT，以下按照 XShell 客户端测试。

1. 打开 XShell，并输入

```
ssh pi@192.168.1.44
```

格式为 ssh user@ip，并输入密码，默认密码：raspberry

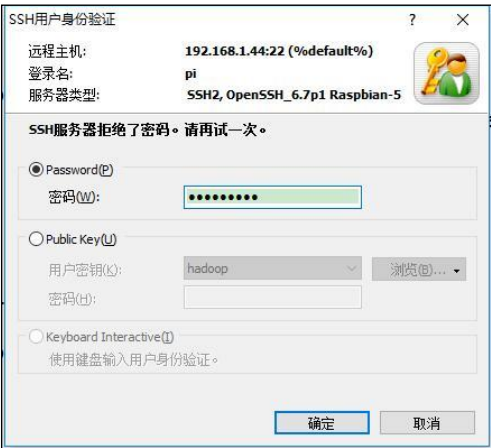
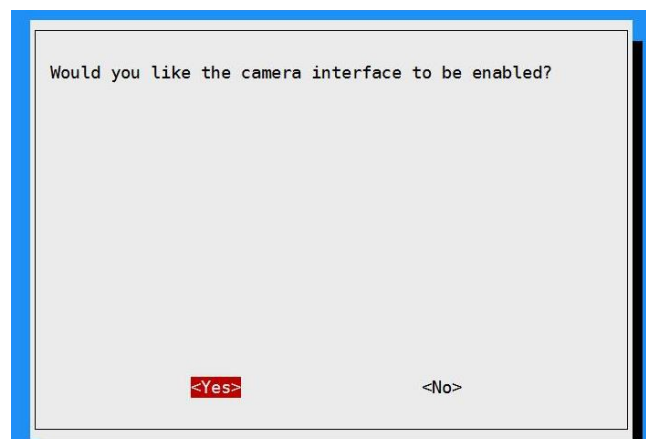
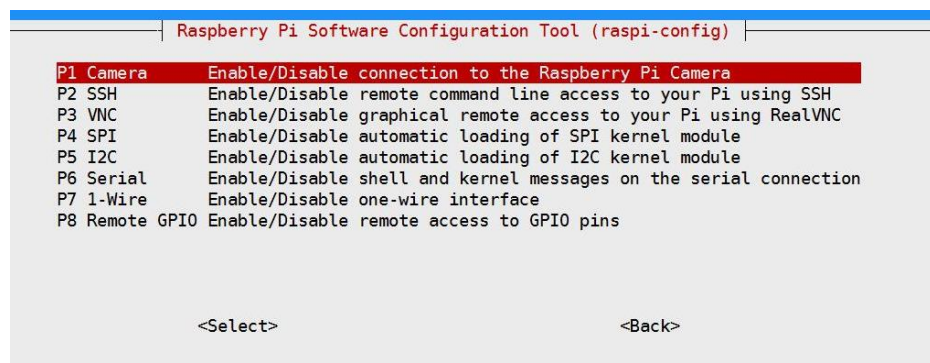
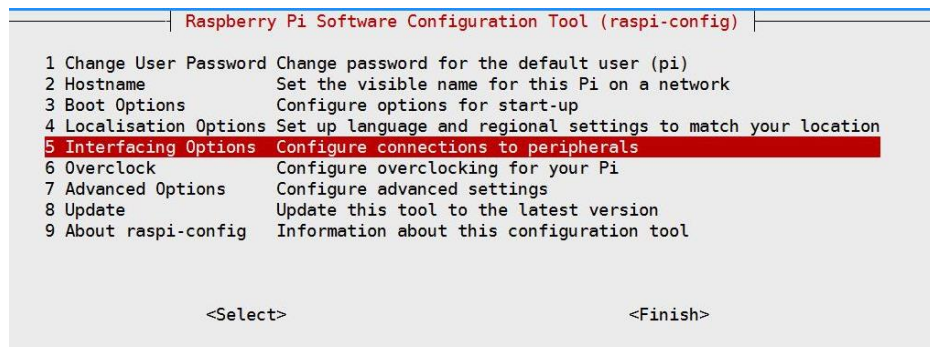
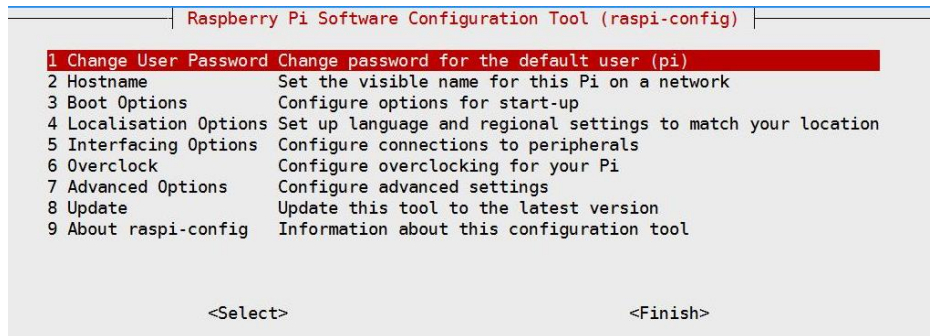
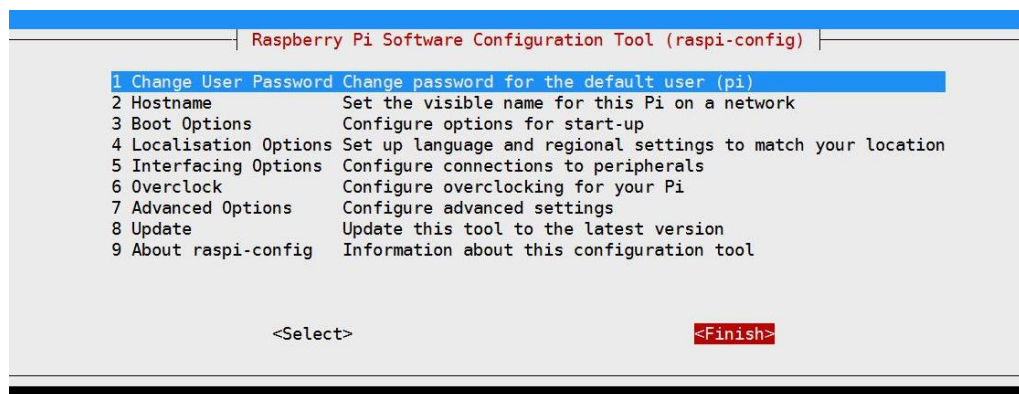
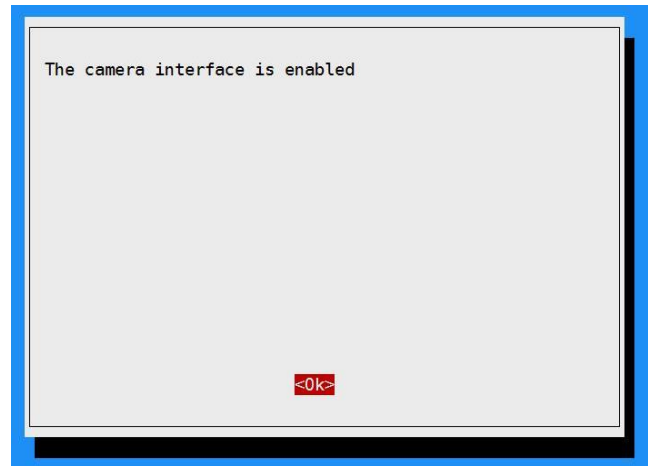


图 2-4 登录界面

2. 启用 Picamera 模块

```
sudo raspi-config
```





完成后，测试摄像头是否成功启用。（拍摄一张照片和拍摄一段视频）

```
raspistill -t 1 -o img.jpg
```

```
raspivid -o video.h264
```

3. 手动连接 wifi

```
sudo vi /etc/wpa_supplicant/wpa_supplicant.conf
```

文件的内容如下：

```
country=GB

ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev

update_config=1

network={

    ssid="Netcore"

    key_mgmt=WPA-PSK

    psk="327327327"

}
```

然后重新启用 wlan0

```
sudo ifdown wlan0
```

```
sudo ifup wlan0
```

此时查看 wlan0 分配的 IP 地址：

```
ifconfig
```

```
wlan0      Link encap:Ethernet  HWaddr e8:4e:06:28:fd:91
            inet addr:192.168.1.23  Bcast:192.168.1.255  Mask:255.255.255.0
            inet6 addr: fe80::ff21:bd31:f343:335e/64  Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:11 errors:0 dropped:42 overruns:0 frame:0
            TX packets:29 errors:0 dropped:1 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:2308 (2.2 KiB)  TX bytes:5617 (5.4 KiB)
```

此时就可以拔掉网线，就可以用无线网卡连路由进行通信。

4. 更新和升级软件

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

2.3 额外的技巧

一般情况下，为了好记，我经常利用 DNS 解析服务将我的域名解析为我的树莓派的 ip 地址，十分方便。

第三章 配置 AP^[2]

3.1 配置流程

3.1.1 必需的包

- 1) hostapd
- 2) dnsmasq

```
sudo apt-get install dnsmasq hostapd
```

3.1.2 配置网卡

将 wlan1（购买的网卡）配置一个静态 IP 地址
首先告诉树莓派 DHCP 获取 IP 时忽略 WLAN1

```
sudo nano /etc/dhcpd.conf
```

在文件的最后添加：

```
denyinterfaces wlan1
```

配置静态 IP

```
sudo nano /etc/network/interfaces
```

文件内容如下：

```
allow-hotplug wlan1

iface wlan1 inet static
    address 172.24.1.1
    netmask 255.255.255.0
    network 172.24.1.0
    broadcast 172.24.1.255

#    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

重启 DHCP 服务。

```
sudo apt-get install dhcpd5 #如果没有这个服务
```

```
sudo service dhcpd restart
```

然后重启 WLAN1 设备以适应配置

```
sudo ifdown wlan1; sudo ifup wlan1
```

3.1.3 配置 HOSTAPD 文件

```
sudo nano /etc/hostapd/hostapd.conf
```

内容是：

```
# This is the name of the WiFi interface we configured above
```

```
interface=wlan1
```

```
# Use the nl80211 driver with the brcmfmac driver
```

```
driver=nl80211
```

```
# This is the name of the network
```

```
ssid=Pi3-AP
```

```
# Use the 2.4GHz band
```

```
hw_mode=g
```

```
# Use channel 6
```

```
channel=6
```

```
# Enable 802.11n
```

```
ieee80211n=1
```

```
# Enable WMM
```

```
wmm_enabled=1
```

```
# Enable 40MHz channels with 20ns guard interval
```

```
ht_capab=[HT40][SHORT-GI-20][DSSS_CCK-40]
```

```
# Accept all MAC addresses
```

```
macaddr_acl=0
```



```
# Use WPA authentication

auth_algs=1


# Require clients to know the network name

ignore_broadcast_ssid=0


# Use WPA2

wpa=2


# Use a pre-shared key

wpa_key_mgmt=WPA-PSK


# The network passphrase

wpa_passphrase=123456789


# Use AES, instead of TKIP

rsn_pairwise=CCMP
```

检查配置是否起作用：

```
sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
```

如果成功，则会在你的移动设备上（智能手机）上检测到 Pi3-AP 的路由。如果你尝试连接，则不会成功分配地址，因为还没有做完！

因为我们还需要告诉 **hostapd** 在启动时启动时查找配置文件。使用

```
sudo nano /etc/default/hostapd
```

打开默认配置文件，找到该行

```
#DAEMON_CONF=""
```

并且用下面这行替换掉

```
DAEMON_CONF="/etc/hostapd/hostapd.conf".
```

3.1.4 配置 DNSMASQ

```
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
```

```
sudo nano /etc/dnsmasq.conf
```

将下面的内容粘贴上去：

```
interface=wlan1      # Use interface wlan0

listen-address=172.24.1.1 # Explicitly specify the address to listen on

bind-interfaces      # Bind to the interface to make sure we aren't sending things elsewhere

server=8.8.8.8       # Forward DNS requests to Google DNS

domain-needed        # Don't forward short names

bogus-priv           # Never forward addresses in the non-routed address spaces.

dhcp-range=172.24.1.50,172.24.1.150,12h # Assign IP addresses between 172.24.1.50 and 172.24.1.150
with a 12 hour lease time
```

3.1.5 建立 IPV4 转发

```
sudo nano /etc/sysctl.conf
```

去掉

```
#net.ipv4.ip_forward=1
```

前面的#

```
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

上面这条命令是立即激活转发！

还需要通过在 `wlan0` 接口和 `wlan1` 接口之间配置 NAT 来分享 Pi 的互联网连接到我们通过 WiFi 连接的设备。可以使用以下命令来执行此操作：

```
sudo iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE

sudo iptables -A FORWARD -i wlan0 -o wlan1 -m state --state RELATED,ESTABLISHED -j ACCEPT

sudo iptables -A FORWARD -i wlan1 -o wlan0 -j ACCEPT
```

但是，我们需要在每次重新启动 Pi 时应用这些规则，运行下面的命令

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

为了避免每次重启都需要运行该命令，打开

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

在 `exit 0` 前添加一行

```
iptables-restore < /etc/iptables.ipv4.nat
```

3.1.6 我们已经完成了

```
sudo service hostapd start
```

```
sudo service dnsmasq start
```

3.2 注意事项

1. 树莓派 3 和树莓派 2 的配置过程是不一样的，上面的配置过程仅适用于树莓派 3.
2. 上面的配置是 WLAN1 做 AP，WLAN0 将数据流转发给 WLAN1 的，那么转发也可以做以太网和 wlan0 转发，以太网和 wlan1 转发。过程类似。
3. 为了激活，必须重启！

第四章 搭建 OpenCV3 环境^[3]

4.1 扩大文件系统

由于编译安装 OpenCV 需要的空间很大，所以应该做的第一件事是扩展文件系统，以包括 micro-SD 卡上的所有可用空间：

```
sudo raspi-config
```

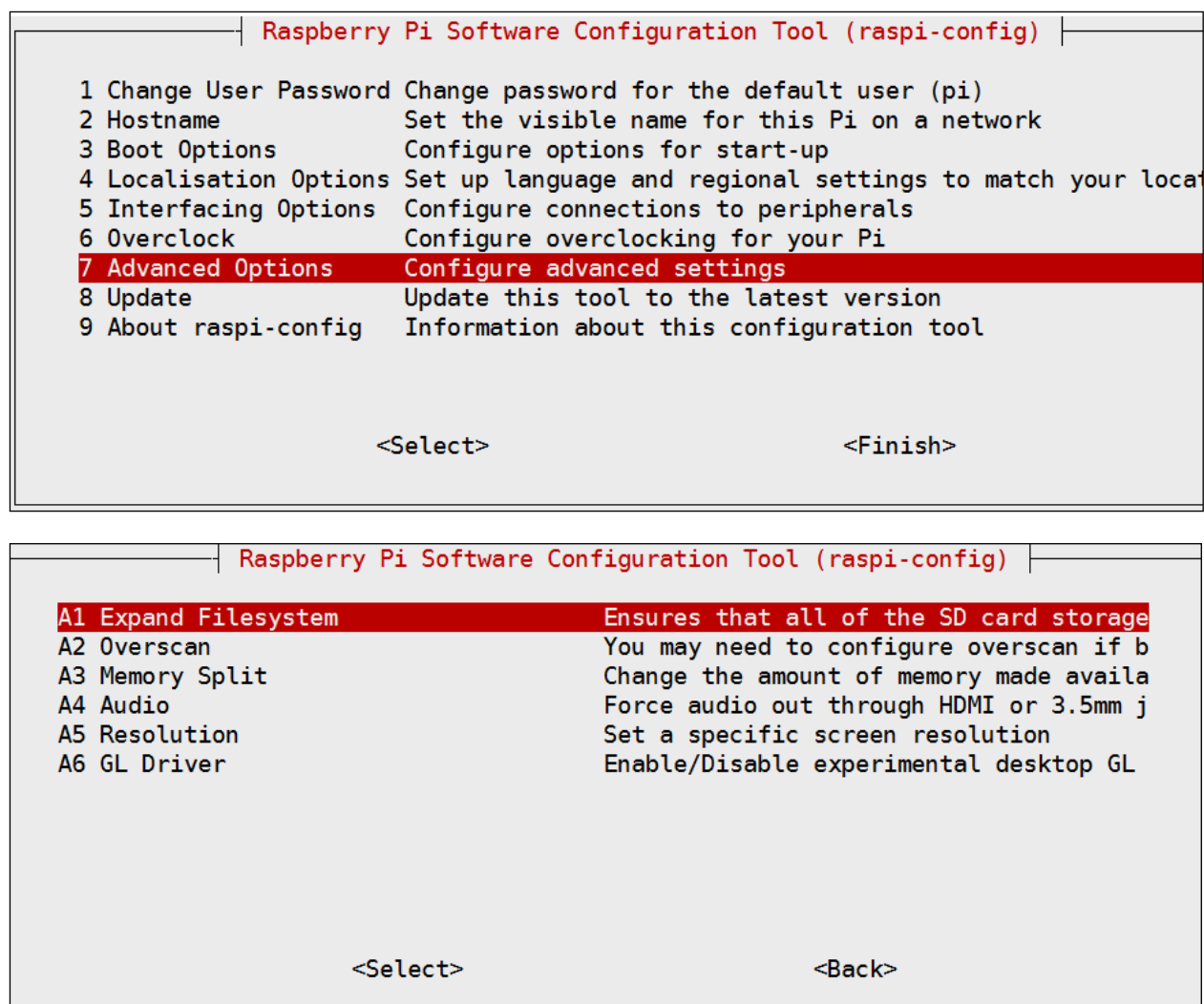


图 4-1 扩展文件系统

4.2 安装依赖

```
sudo apt-get install build-essential cmake pkg-config -y
```

```
sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev -y
```

```
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev -y

sudo apt-get install libxvidcore-dev libx264-dev -y

sudo apt-get install libgtk2.0-dev -y

sudo apt-get install libatlas-base-dev gfortran -y

sudo apt-get install python2.7-dev python3-dev -y
```

4.3 下载 OpenCV 源代码

```
cd ~

wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.1.0.zip

unzip opencv.zip
```

OpenCV 3.1.0 地址: https://github.com/Itseez/opencv_contrib/archive/3.1.0.zip

```
wget -O opencv_contrib.zip https://github.com/Itseez/opencv_contrib/archive/3.1.0.zip

unzip opencv_contrib.zip
```

4.4 Python2.7 还是 Python3.5

1) pip 包管理器

```
wget https://bootstrap.pypa.io/get-pip.py

sudo python get-pip.py


sudo pip install virtualenv virtualenvwrapper

sudo rm -rf ~/.cache/pip


# virtualenv and virtualenvwrapper

export WORKON_HOME=$HOME/.virtualenvs

source /usr/local/bin/virtualenvwrapper.sh


echo -e "\n# virtualenv and virtualenvwrapper" >> ~/.profile

echo "export WORKON_HOME=$HOME/.virtualenvs" >> ~/.profile

echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.profile


source ~/.profile
```

4.4.1 安装 Numpy

```
pip install numpy
```

如果正在为 Python 2.7 编译 OpenCV 3, 请确保 Python 2 部分包含 **Interpreter** , **Libraries** , **numpy** 和 **packages path** :

4.5 编译安装 OpenCV

```
cd ~/opencv-3.1.0/

mkdir build

cd build

cmake -D CMAKE_BUILD_TYPE=RELEASE \
      -D CMAKE_INSTALL_PREFIX=/usr/local \
      -D INSTALL_PYTHON_EXAMPLES=ON \
      -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-3.1.0/modules \
      -D BUILD_EXAMPLES=ON ..

make -j4
```

很容易出错, 因为树莓派不一定 CPU 的资源不一定够用, 使用

```
make clean

make
```

```
sudo make install

sudo ldconfig
```

然后检查是否存在层 cv2.so 或者 cv2.cpython-34m.so:

对于 Python27 说,

```
ls -l /usr/local/lib/python2.7/site-packages/
```

如果存在, 则说明在 python2 的环境下能够使用 OpenCV。

对于 Python3.4 说:

```
ls -l /usr/local/lib/python3.4/site-packages/
```

如果存在, 则表明在 Python3 的环境下能够使用 OpenCV

但是对于 Python3 说需要将 cv2.cpython-34m.so 重命名为 cv2.so 才能使用。

安装完成后，需要将编译生成的文件和解压后的文件删除。

```
rm -rf opencv-3.1.0 opencv_contrib-3.1.0
```

第五章 如何初始化 HypriotOS

5.1 安装 HypriotOS

这个就不用再说了，非常简单的，可以参考我的之前文章，但是最近我在部落格里面发现，安装方法比较多，我在这里抛砖引玉，有以下安装方法：

- ☺ Windows 下安装方法（目前就发现一种方法和一个小工具）
 - ☺ Win32 磁盘映像工具 <https://sourceforge.net/projects/win32diskimager>
 - ☺ 工具： <https://www.raspberrypi.org/forums/viewtopic.php?t=195939>
- ☺ Linux 下的安装方法
 - ☺ flash tools <http://www.jianshu.com/p/3e2a901e66b7>
 - ☺ space.sh 看起来很牛 x <https://rpi.sh/post/setup/>

5.2 重要：如何进行 headless 配置

先放出一堆参考链接：

推荐 1：这个适用于 RaspOS

<https://raspberrypi.stackexchange.com/questions/10251/prepare-sd-card-for-wifi-on-headless-pi>

推荐 2：虽然这个工单已经关闭了，但是还是有帮助的

<https://github.com/hypriot/blog/issues/60>

推荐 3：官方救命稻草

<https://github.com/hypriot/image-builder-rpi/releases>

-----☺☺☺☺☺☺-----

对于 headless 的配置，简化讲，就是一般情况下，我们安装好树莓派镜像后，开机十分麻烦，再说如果手头上没有显示器，那就惨了。

- ☺ 在根目录(其实就是网上说的/boot 分区)，首先添加一个空文件 ssh 代表启用 ssh 服务，找个路由器，找根网线连到路由器上，然后进入路由器的管理界面，查看增加那些 ip，这个 ip 很可能就是你的树莓派（一般情况下，都会说明主机名！）
- ☺ 上面的陈述还是较为简单的，但是如果这几样你如果少一样，都无法完成开机，那么 headless 就是我们想要的一种结果

因此，我们需要在开机之前，把所有的事全部搞定（启动 ssh 服务，启用网卡，连上无线），

因此关于 HypriotOS 的 headless 配置这篇文章来说，我们在这里以它为例（特殊）！不同于官方镜像，他的 headless 配置相比较简单些。

New Features

- Bash completion for `docker-compose` and `docker-machine` cli commands
- cloud-init with a default configuration `/boot/user-data` file
 - hostname: black-pearl
 - username: pirate
 - password: hypriot
- cloud-init, examples included in default `user-data` file
 - WiFi configuration
 - set system locale
 - set timezone
 - update/upgrade APT packages
 - install additional APT packages
 - run additional commands at first boot only

Breaking change

- No more device-init, please convert your `/boot/device-init.yaml` to `/boot/user-data`

图 6-1 官方说明

上图说的是，他们已经大大的简化了初始配置步骤，一个 `user-data` 文件就可以搞定！但是注意，你的镜像版本是否为 HypriotOS v1.7.1

下面放出我的 `/boot/user-data` 文件

```
#cloud-config
# vim: syntax=yaml
#

# The current version of cloud-init in the Hypriot rpi-64 is 0.7.9
# When dealing with cloud-init, it is SUPER important to know the version
# I have wasted many hours creating servers to find out the module I was trying to use wasn't in the cloud-
init version I had

# Documentation: http://cloudinit.readthedocs.io/en/0.7.9/index.html

# Set your hostname here, the manage_etc_hosts will update the hosts file entries as well
hostname: black-pearl
```

```

manage_etc_hosts: true

# You could modify this for your own user information
users:
  - name: pirate
    gecos: "Hypriot Pirate"
    sudo: ALL=(ALL) NOPASSWD:ALL
    shell: /bin/bash
    groups: users,docker,video
    plain_text_passwd: hypriot
    lock_passwd: false
    ssh_pwauth: true
    chpasswd: { expire: false }

# # Set the locale of the system
# locale: "en_US.UTF-8"

# # Set the timezone
# # Value of 'timezone' must exist in /usr/share/zoneinfo
# timezone: "America/Los_Angeles"

# # Update apt packages on first boot
# package_update: true
# package_upgrade: true
# package_reboot_if_required: true
package_upgrade: false

# # Install any additional apt packages you need here
# packages:

```

```

# - ntp

# WiFi connect to HotSpot

# - use `wpa_passphrase SSID PASSWORD` to encrypt the psk
write_files:
  - content: |

    allow-hotplug wlan0

    iface wlan0 inet dhcp

    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

    iface default inet dhcp

    path: /etc/network/interfaces.d/wlan0

  - content: |

    ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev

    update_config=1

    network={

      ssid="Netcore"

      psk="327327327"

      proto=RSN

      key_mgmt=WPA-PSK

      pairwise=CCMP

      auth_alg=OPEN

    }

    path: /etc/wpa_supplicant/wpa_supplicant.conf

# These commands will be ran once on first boot only

runcmd:

  # Pickup the hostname changes

  - 'systemctl restart avahi-daemon'

```

```
# # Activate WiFi interface  
- 'ifup wlan0'
```

5.3 接下来

接下来就等待连接你的 Hotpot，连接成功后，打开你的 shell，连接即可：

☺ hostname: black-pearl

☺ username: pirate

☺ password: hypriot

第六章 搭建机器学习与深度学习环境

6.1 搭建 TensorFlow 环境^[4]

1) 下载并安装

```
# For Python 2.7

wget https://github.com/samjabrahams/tensorflow-on-raspberry-pi/releases/download/v1.1.0/tensorflow-1.1.0-cp27-none-linux\_armv7l.whl

sudo pip install tensorflow-1.1.0-cp27-none-linux_armv7l.whl


# For Python 3.4

wget https://github.com/samjabrahams/tensorflow-on-raspberry-pi/releases/download/v1.1.0/tensorflow-1.1.0-cp34-cp34m-linux\_armv7l.whl

sudo pip3 install tensorflow-1.1.0-cp34-cp34m-linux_armv7l.whl
```

由于国内的网络原因，因此会出现 403。为了避免这种事情的发生，我提前下载好了并放在了七牛云上。

python2: http://olxgfquub.bkt.clouddn.com/tensorflow-1.1.0-cp27-none-linux_armv7l.whl
python3: http://olxgfquub.bkt.clouddn.com/tensorflow-1.1.0-cp34-cp34m-linux_armv7l.whl

2) 重装 mock 库

```
# For Python 2.7

sudo pip uninstall mock

sudo pip install mock


# For Python 3.3+

sudo pip3 uninstall mock

sudo pip3 install mock
```

6.2 安装 keras

```
sudo pip install keras
```

参考文献

- [1] <https://caffinc.github.io/2016/12/raspberry-pi-3-headless/>[Z].: 2017.
- [2] Frillip. <https://frillip.com/using-your-raspberry-pi-3-as-a-wifi-access-point-with-hostapd/>[Z]. 2017: 2017.
- [3] Pyimagesearch. www.pyimagesearch.com/2016/04/18/install-guide-raspberry-pi-3-raspbian-jessie-opencv-3/[Z]. 2017: 2017.
- [4] Samjabrahams. <https://github.com/samjabrahams/tensorflow-on-raspberry-pi>[Z]. 2017: 2017.
- [5] <https://raspberrypi.stackexchange.com/questions/10251/prepare-sd-card-for-wifi-on-headless-pi>[Z]. 2017: 2017.
- [6] <https://github.com/hypriot/blog/issues/60>[Z]. 2017: 2017.
- [7] <https://github.com/hypriot/image-builder-rpi/releases>[Z]. 2017: 2017.