

# Raspberry Pi Recipes

树莓派3食用指南

Yangxuan

Published  
with GitBook



---

# Table of Contents

引言	1.1
第零章-怎样阅读这本书	1.2
第一章-初见树莓派3	1.3
第二章-写入系统及登陆	1.4
第三章-网络配置	1.5
第四章-第一次启动后要做的事	1.6
第五章-中文化	1.7
第六章-接下来...	1.8
第七章-关于用途	1.9
第八章-更换为国内软件源	1.10
第九章-SSH	1.11
第十章-远程连接	1.12
第十一章-连接蓝牙键盘	1.13
第十二章-共享文件(Samba)	1.14
第十三章-备份	1.15
第十四章-软件篇	1.16
第十五章-Aria2下载(支持百度云)	1.17
第十六章-FFmpeg	1.18
第十七章-mpv	1.19
第十八章-命令行版网易云音乐	1.20
第十九章-其它命令行程序	1.21
第二十章-Nginx	1.22
第二十一章-Powerline	1.23
第二十二章-DNSPod	1.24
第二十三章-微信 远程遥控树莓派	1.25
第二十四章-写个爬虫(Scrapy)	1.26
第二十五章-硬件篇	1.27
第二十六章-连接显示器及LCD	1.28
第二十七章-关于外壳和温度	1.29
第二十八章-Lapdock	1.30

---

第二十九章-真的没有彩蛋	1.31
附录1-树莓派3可以用的系统	1.32
附录2-GPIO	1.33
附录3-购买相关	1.34
附录4-有用的链接	1.35

---

# 关于这本书

在线阅读: [GitBook](#)

下载: [PDF版](#), [Mobi版](#), [ePub版](#)

联系方式: [920196849@qq.com](mailto:920196849@qq.com)

这本gitbook的内容是关于树莓派3的使用，针对的是Raspbian系统，内容来源于网络。

使用树莓派是从B+开始，当时初次接触Linux相当多的东西都不了解，一开始也没有记录的习惯，经常是卡到一个地方去网上搜，等下次碰到一样的问题时候又忘记了，结果总是做很多重复的搜索，到后来决定把碰到的问题和解决办法写成一个文档，因为是和 *cache* 类似的感觉，就命名为 *rpi\_cache*，格式是 *markdown*；因为计算机相关的东西用英文搜索高效一些，而且英文相对简单明了，所以文档也就以就全用英文。版本管理采用的是 *git*，还专门尝试在树莓派上安装了Gitlab来管理这个文档。随着使用时间的积累，这个文档也越来越长，中间手里的 pi 也从B+换成了2B，等到了2016年3月 RPi 3发布的时候，发现 Raspbian 迁移到了 Jessie，很多东西都需要更新了,正好加了树莓派的一个QQ群，发现很多人有和我一开始上手时一样的困难，于是决定把之前的文档汉化更新，也就形成了现在这个文章的主要内容，当时是发在[作业部落](#)上，后来不断删改，文章越来越长，不再适应单个文档的形式，于是决定做成这本gitbook

最后说一下这本书的基本组成部分:

## PART 1

第一章到第十三章，讲的是树莓派的基本配置

## PART 2

第十四章到第十九章，树莓派软件相关

## PART 3

第二十章到第二十四章，其它一些相关资料

## PART 4

第二十五章到第二十八章，树莓派硬件相关

## PART 5

附录，一些树莓派的资料

## 第零章-怎样阅读这本书

这本书是使用 [Markdown](#)语法写成的

默认读者程度为树莓派新手，专门针对的硬件是3代(2代或zero大体通用，但个别部分不同)，系统为 `Raspbian`

下面列举一下会出现的格式

这是引用，一般是从维基或其它来源引用的文字，内容一般是对上下文的引申或相关冷知识

行内出现 `这样` 的部分，一般是专有名词或按键，这样的格式更容易辨识，可以提高可读性

```
uname -a
```

类似上面格式的内容有两种可能：

- 命令(一般是输入树莓派的)
- 需要删改的文本内容

如果是第二种，上一行应该会有提示，比如类似 `找到`，`添加`

其它没有特别提示的，一般是要输入树莓派终端的命令，这种可以直接复制到终端或自己对照输入，新手对 `Linux` 命令不熟悉的话容易输入错误，开始尽量多复制粘贴

其实在编写时两种内容格式并不一样

命令是这样：

```
```shell
df -h
```
```

而文本内容是总体缩进4个空格，不过渲染后看起来几乎一样

书中命令默认大家在树莓派上用的编辑器是 `nano`

编辑文件：

```
nano file_name
```

进入之后可以用键盘方向键移动

查找 `Ctrl + w`，在已经输入查找词语的情况下按 `Enter` 定位到下一个匹配的结果处

保存 `Ctrl + o`

退出 `Ctrl + x`

Windows 上建议大家不要用自带的记事本，因为很容易破坏格式，而且缩进这些都体现不出来，也不显示行号很低效

Windows 上文本编辑器推荐用 [Notepad ++](#)，功能强，体积小，而且免费

这里 Windows 上一般能看到的树莓派分区就一个，基本主要修改的文件也基本就一个：

`/boot/config.txt`

这是全路径，`/boot` 是启动分区，也就是在 Windows 上能看到的这个分区，格式是 `FAT`，树莓派要启动相关配置文件都在这里，所以很关键，不了解的话不要随便修改

`config.txt` 是配置的文件，包括修改分辨率，选择视频输出，音频输出，超频，具体各种参数的意义参照附录4给出的官方说明链接

## 第一章-初见树莓派3

刚接触树莓派3的人会发现它就是一块裸露的电路板，不像人们印象中的消费电子产品，而更接近于开发板，至于外壳这些都得另外选购

到手之后，要想启动你需要：

- 存储卡
- 电源
- 系统

（如果你属于进阶玩家的话可以尝试网络启动，这里有一篇[官方教程](#)）

关于存储卡，尽量选32G或之上，规格选Class10或以上，不同品牌读写的话可以参照这个表格，这个是超频后的SD卡在 pi3上的表现，我用的是三星 Evo+

| Card Make/Model         | hdparm buffered | dd write  | 4K rand read | 4K rand write |
|-------------------------|-----------------|-----------|--------------|---------------|
| Samsung Pro+ 32GB       | 39.93 MB/s      | 31.0 MB/s | 12.15 MB/s   | 1.84 MB/s     |
| Samsung Pro 16GB1       | 31.59 MB/s      | 32.8 MB/s | 11.20 MB/s   | 1.48 MB/s     |
| Samsung Evo+ 32GB       | 37.68 MB/s      | 20.0 MB/s | 12.20 MB/s   | 3.75 MB/s     |
| Samsung Evo Select 32GB | 24.56 MB/s      | 13.2 MB/s | 4.69 MB/s    | 0.82 MB/s     |
| Samsung Evo 16GB        | 32.47 MB/s      | 11.8 MB/s | 6.44 MB/s    | 1.25 MB/s     |
| SanDisk Extreme Pro 8GB | 40.52 MB/s      | 35.9 MB/s | 11.31 MB/s   | 1.28 MB/s     |
| SanDisk Extreme 16GB    | 40.88 MB/s      | 39.1 MB/s | 11.77 MB/s   | 2.36 MB/s     |
| SanDisk Ultra 16GB      | 37.41 MB/s      | 8.5 MB/s  | 6.71 MB/s    | 1.61 MB/s     |

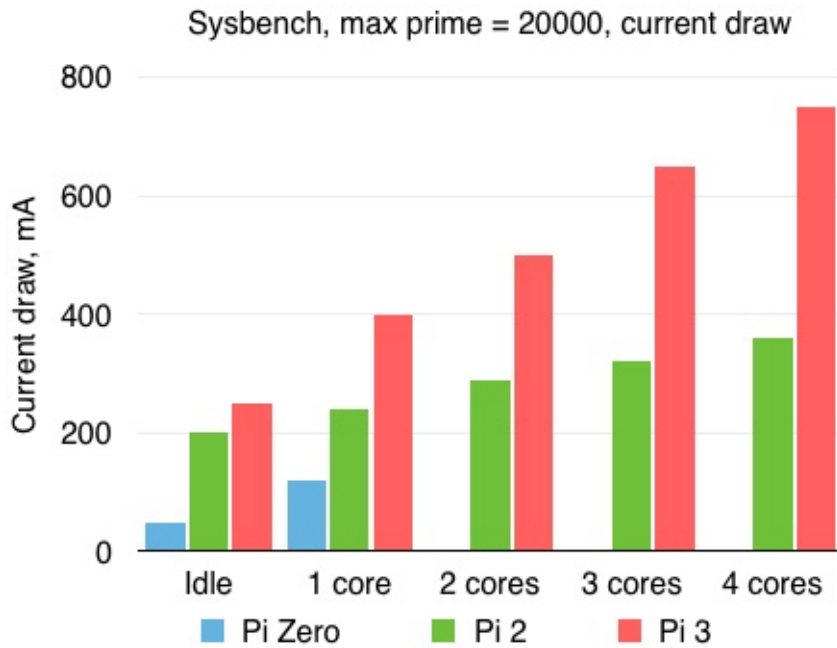
来源[pidramble](#)，版权归属原作者

电源一定要选择质量好的，推荐[官方](#)，或者其它规格为5V2A的电源

这是因为3代树莓派在加入蓝牙和Wi-Fi后耗电量大幅提升，



可以参照以下图片，来源 [pimoroni](#), 版权归属原作者



系统镜像 [Raspbian](#) 下载点 [这里](#)

Raspbian 是基于 [Debian](#)，并专为树莓派进行优化的一套系统，不属于Raspberry Pi Foundation（树莓派基金会），但被列为官方支持的操作系统。

下一章介绍如何写入系统并登陆

## 第二章-写入系统及登陆

以 Windows 为例，写入系统可以用 [Win32DiskImager](#)

将上一章下载好的文件解压，然后打开软件之后点击右侧文件夹图标选择刚才解压好的 `.img` 镜像文件

确认右侧的 `Device` 下拉条选中在存储卡的卷标，不是的话自行选中

点击**Write**写入镜像；待写入完毕后将存储卡插入树莓派

备份的话也可以用这个软件，点击**Read**，选择镜像存放路径并输入镜像名称。

如果感觉上述过程不够详细，可以参见官方的[安装指导](#)。

写入完成后在 Windows上应该可以看到一个分区，进入可以看到一个很重要的文件 `/boot/config.txt`，建议选择合适的编辑器(关于Win上的编辑器推荐见第三章)打开，很多设置都是通过修改这个文件来完成的

写入之后会发现Micro SD卡的容量“变小”了，这只是表面现象：实际上写入之后卡上会存在两个分区，一个是启动分区，格式为FAT（也就是在Windows上可见的分区，容量约几十MB，`config.txt`就存放在这里），一个是系统分区，格式为EXT4（大小不到4G，在Windows上读写可以用安装在虚拟机里的Linux）

之后这张卡如果想挪作他用，可以选择[SD Card Formatter](#)来格式化

将存储卡插入树莓派并接上电源，如果有显示器的话在屏幕上应该可以看到启动画面了

一切正常的话你在十几秒后就能看到 Raspbian 的桌面

Raspbian的桌面是轻量级的[LXDE](#)，同时也做了一些调整，相对更加美观，官方在新版(2016-09-23)中将其新主题命名为 `PIXEL`，图标边框等都作了调整，并添加了开机闪屏，不过感觉闪屏不是特别实用，其它改动倒是都不错，比如UI终于比较现代一点

如果你没有接入显示器的话可能需要提供登陆信息

- 用户名: pi
- 密码: raspberry

这个账户加 `sudo` 应该基本可以完成所有操作，即使你添加了 `root` 账户也千万不要作日常使用，会有潜在的问题

进入系统后可以先浏览一下都有哪些东西，然后打开终端 `Terminal`，作为Linux的用户，和终端的交互是不可或缺的，很多设置都要通过它

如果不喜欢终端默认的配色可以考虑尝试下经典的Solarized Dark，方法很简单，终端的 `Edit -> Preferences`，把 `Background` 改为 `#002B36` 即可，其它不需要改动

先尝试一下，比如输入：

```
uname -a
```

查看存储可用空间

```
df -h
```

查看存储设备

```
lsblk
```

查看当前目录文件信息

```
ls -l
```

还有 `cd` 命令

```
cd dir
```

`dir` 是目录名，这条命令可用进入指定的目录

返回上一层目录是

```
cd ..
```

现在在终端内打开一个文件

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

大概浏览一下看它都有什么内容，不要修改

按 `Ctrl + x` 退出

上面命令里出现了 `nano`，它是系统自带的文本编辑器，通常用它来编辑文件

如果你想要一个更接近于平时使用的编辑器，可以选择预装的 `Geany`，你在菜单里可以找到它，配合插件可以很接近于 Windows 上的 Notepad++

(如果启动停留在彩虹画面,说明 **kernel.img** 启动失败,建议替换或重新写入系统, [信息来源](#) )

树莓派有两个指示灯,靠近卡槽的绿色指示灯指示读写状态,靠近充电接口的红色指示灯指示电源状态

正常电源接通后,红色指示灯常亮;如果闪烁说明供电不稳

开机后如果右上角有彩色小方块同样说明供电有问题,建议更换可靠的电源

有读写时绿色指示灯会闪烁

更多故障诊断参见 [elinux](#)

## 第三章-网络配置

- 有线连接 基本不需要额外配置，直接插入网线应该就可以
- Wi-Fi 两种办法，如果你有显示器和键盘直接点击图标配置即可；否则可以在第一个分区（也就是你在电脑上可见的那个分区）内新建一个 `wpa_supplicant.conf` 文件

输入以下内容

```
network={
    ssid="testing"
    psk="testingPassword"
}
```

两个引号内分别输入Wi-Fi名和密码

开机时系统发现这个文件会自动移到 `/etc/wpa_supplicant/` 目录下

Windows上的文本编辑器推荐 [Notepad++](#) 或者 [Atom](#)，千万别用 Windows自带的记事本，不然换行和缩进之类的全乱了，很多人启动卡壳都是因为 `/boot/config.txt` 的格式或内容有问题

## 第四章-第一次启动后要做的事

首次运行应该会自动扩展分区

进行下一步前建议大家先超频存储卡，此设置可以提高后续工作的效率；注意这里的操作没有任何危险，与CPU超频不是一个概念，进行这一步可以充分利用到 Class10的读写能力，官方默认锁频只是为了提高兼容性（主要是为了支持杂牌卡）

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

加入：

```
dtoverlay=sd_overclock=100
```

其中100可以为其他数值，比如有些卡不支持这么高可以设置为80（默认50）

重启后查看是否生效：

```
dmesg | grep "mmc0"
```

成功的话则会显示：

```
[ 2.596650] mmc0: overclocking to 100000000Hz
```

之后更新系统及软件到最新版本：

```
sudo apt-get update&&sudo apt-get upgrade
```

现在已经再不需要：

```
sudo rpi-update
```

只需 `apt-get update`

如果发现键盘输出有问题，则需要设置键盘：

```
sudo nano /etc/default/keyboard
```

将

```
XKBLAYOUT="gb"
```

改为：

```
XKBLAYOUT="us"
```

## 第五章-中文化

打开终端，输入：

```
sudo raspi-config
```

选择

```
5 Internationalisation Options
```

中文支持:

```
I1 Change Locale
```

方向键移动至列表末尾，用 `空格键` 选中以下项：

```
[*] zh_CN.UTF-8 UTF-8
```

之后按 `TAB` 键移动至 `Ok`，接下来会选择系统默认显示语言，按个人需求选择

修改时区：

```
I2 Change Timezone
```

进入后依次找到选中 **Asia**，城市选择 **Shanghai**

稍后系统会自动更新时间

中文字体：

新版(2016-09-23)不安装中文字体也可显示中文，所以可以直接安装输入法；不过鉴于之后有可能会变动，故保留这部分

~~新版系统 (2016-05-27) 似乎需要安装中文字体~~

```
sudo apt-get install fonts-wqy-zenhei
```

安装输入法（Fcitx，拼音）：



```
sudo apt-get install fcitx fcitx-pinyin
```

设置：

打开 **Menu -> Run**

输入：

```
im-config
```

选中 **fcitx**

可能会问是否改变系统设置，选择改动

重启后在右上方可以看到小键盘标志，右击选择 **Configure**，再点击 **+**，在打开的文本框内输入 'pinyin'，在上方列表中选中 **Pinyin**，点击 "OK"

测试输入法可在文本编辑器或浏览器内，按 **Ctrl + 空格键** 切换至拼音，按 **Shift** 可在中英文输入间切换

## 第六章-接下来...

到目前为止，大部分基本配置算是结束了，之后你可以用自己的方式去探索

不过这本gitbook还未完结，下面内容你可以选择自己感兴趣的来看

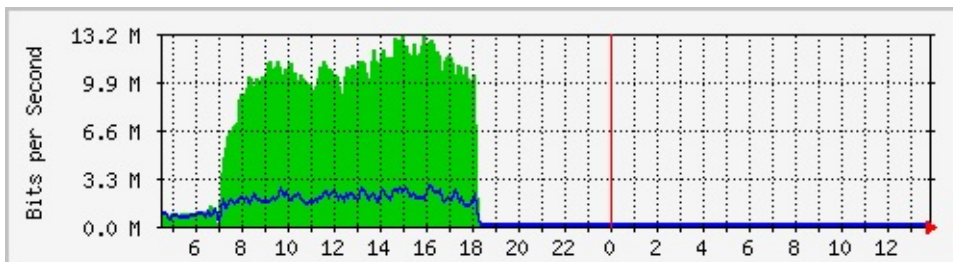
## 第七章-关于用途

继续树莓派的发现之旅

诚然树莓派的性能确实算不上强劲，甚至可能不如你的手机，但也没有想象的那么不堪

一个很好的例子就是一台树莓派3承担了官网 5% 的流量，图片来自官网博客，版权归属原作者Helen Lynn

下图是2月29日树莓派3发布当天这台RPi3的流量情况



在Linux电脑终端内输入：

```
HEAD -e https://www.raspberrypi.org/
```

如果最后返回结果有如下信息：

```
X-Served-By: Raspberry Pi 3
```

那你这次请求就是由这台树莓派3响应的

运行十几个小时后它宕机了，如果下一代内存加倍也许就能避免这种情况

但即使这样，当自己Blog的Server应当是足够的

这台树莓派还有一个自己的Twitter账号，内容是自动发布的

其它用途包括但不限于：[GitLab](#)（类似Github），[OwnCloud](#) (私有云)，[Docker...](#)

别的地方也有它的身影：[MATLAB](#)，[Visual Studio Code](#)

西数还专为它做了硬盘：[PiDrive](#)

一些奇特的用途：[Morse Code\[?\]](#)，[Radiofax\[?\]](#)

本身一块开发板能有8百万 破千万的销量，就已经是很神奇的事情了

对Linux历史感兴趣的话可以看下纪录片The Code，最近整理了以前做的字幕，第一次当野生字幕组，已上传至B站：[The Code](#)

## 第八章-更换为国内软件源

你可能注意到了 `Raspbian` 的软件安装命令是 `apt-get`，软件是 `.deb` 格式

高级包装工具（英语：Advanced Packaging Tools，缩写为APT）是Debian及其派生发行版的软件包管理器。

`apt-get` 从软件源安装软件，而官方源相对连国内速度不是很快，不过国内有些院校和机构提供镜像，所以我们可以通过修改软件源的方式提高速度

（仅适用于连接不上官方源的用户，某些系统软件只能通过官方源安装或更新）

先测试最佳软件源：

安装 `netselect`

```
sudo apt-get install netselect
```

下载树莓派国内源检测脚本

```
git clone https://github.com/sjqlwy/ccrm.git
```

```
cd ./ccrm
```

运行：

```
sudo sh ccrm.sh
```

```
sudo apt-get update
```

测出连接速度最快的源之后，修改 `sources.list`，（假设结果为阿里源）

```
sudo nano /etc/apt/sources.list
```

加入：

```
deb http://mirrors.aliyun.com/raspbian/raspbian/ jessie main non-free contrib  
deb-src http://mirrors.aliyun.com/raspbian/raspbian/ jessie main non-free contrib
```

```
sudo apt-get update
```

（如果仍然遇到连接问题，则进行以下操作）

```
sudo nano /etc/apt/sources.list.d/raspi.list
```

在原来的内容前添加 **#** 号，注释掉原来的源

有时 `apt-get` 的key可能出问题，具体来说就算官方源也会报说来源未知

这时可以尝试运行：

```
sudo apt-key update
```

## 第九章-SSH

一般来说我们连接远程主机都是使用 SSH，比如在外面用手机访问家里的树莓派或是VPS，相比 [远程桌面](#) 这种服务更灵活，更自由

Secure Shell（缩写为SSH），由IETF的网络工作小组（Network Working Group）所制定；SSH为一项创建在应用层和传输层基础上的安全协议，为计算机上的Shell（壳层）提供安全的传输和使用环境。

因为 Windows 的 `CMD` 或 `powershell` 并没有提供 SSH，所以我们需要下载客户端 [putty](#) 或 [Xshell](#)

安卓推荐 [JuiceSSH](#)

如果你有安装 Github 的桌面客户端的话，也可以使用它的 Git Shell

在 Host Name 处填入 `raspberrypi` 或你的树莓派IP地址

提示输入用户名和密码的话，依次输入

- pi
- raspberry

之后你就可以远程操作你的树莓派了，当然不在一个网内的话可能需要其它配置

---

下面我们来配置免密码登陆，这样就不用每次输入密码了

先在树莓派上生成SSH Key:

```
ssh-keygen -t rsa -C YOURNAME@YOURDEVICE
```

```
cd ~
```

```
install -d -m 700 ~/.ssh
```

```
cat ~/.ssh/id_rsa.pub | ssh USERNAME@IP-ADDRESS 'cat >> .ssh/authorized_keys'
```

也可在putty或其它软件上生成

然后将生成的公钥 **pi.pub** 用Samba/SFTP/ZMODEM传送至树莓派

```
cat pi.pub >> .ssh/authorized_keys
```

如果你比较在意安全的话，上面设置成功后可以关闭密码登陆了

---

如果不愿意关闭密码登陆，可以通过开启两步认证来提高安全系数

```
sudo apt-get install ssh libpam-google-authenticator
```

配置：

```
google-authenticator
```

截屏(WIN: WIN + Prtscr , Android: Power Button + Volume - ) 存至安全的地方

所有选项均选yes

下载谷歌[Google Authenticator](#)，用手机扫描二维码

```
sudo nano /etc/pam.d/sshd
```

添加：

```
auth required pam_google_authenticator.so
```

然后

```
sudo nano /etc/ssh/sshd_config
```

找到 **ChallengeResponseAuthentication**

把 **no** 改成 **yes**

```
systemctl restart sshd.serviced
```

测试：

```
ssh pi@raspberrypi
```



经常通过 SSH 访问树莓派的话，你可能会发现有些地方不方便，比如正在进行的任务因为网络不稳定突然中断而前功尽弃，这时推荐你用 `screen` 或 `tmux`

安装:

```
sudo apt-get install screen
```

运行:

```
screen bash
```

Detach:

```
CTRL + A release, then press D
```

列出当前所有 screen:

```
screen -list
```

重新连接:

```
screen -r
```

结束: CTRL + D

重新连接 attached 的 session:

```
screen -D -r
```

## 第十章-远程连接

- VNC:

新版系统(2016-09-23)已经自带，可以直接配置

```
tightvncserver
```

```
vncserver :1 -geometry 1024x768 -depth 24
```

Client（PC，手机）上需要安装[Viewer](#)

- TeamViewer:

需在树莓派上安装 [Host](#)

然后还需要在PC或手机上安装 [TeamViewer](#)

（host显示有点问题，但不影响使用）

- XRDP:

```
sudo apt-get install xrdp
```

看到官方博客讲 XRDP 与 RealVNC 冲突，如想安装需先卸载自带的 RealVNC：

```
sudo apt-get purge realvnc-vnc-server
```

更推荐VNC，因为XRDP使用中曾经碰到过问题，而且当时未找到解决方案

## 第十一章-连接蓝牙键盘

```
bluetoothctl -a
```

```
power on
```

```
agent KeyboardOnly
```

```
default-agent
```

```
pairable on
```

```
scan on
```

```
pair 40:F4:07:4A:FC:E7
```

```
trust 40:F4:07:4A:FC:E7
```

```
connect 40:F4:07:4A:FC:E7
```

```
quit
```

## 第十二章-共享文件(Samba)

分两个部分讲，分别是

- 分享文件给 Windows
- 访问Windows的共享文件

先说怎么分享

在树莓派上:

```
sudo apt-get install samba samba-common-bin
```

```
sudo nano /etc/samba/smb.conf
```

找到 **##### Misc #####**

添加（可以提升文件传输速度，有线连接情况下约10MB/s）：

```
socket options = TCP_NODELAY IPTOS_LOWDELAY SO_RCVBUF=65536 SO_SNDBUF=65536
```

找到**===== Share Definitions =====**

把

```
read only = **yes**
```

改为:

```
read only = **no**
```

添加用户pi并设置密码

```
sudo smbpasswd -a pi
```

重启:

```
sudo systemctl restart smbd.service nmbd.service
```

Windows上 (win10) :

此电脑 > 映射网络驱动器 > 输入 `\\raspberrypi\pi`

---

然后是怎样访问Windows共享文件

临时的访问可以直接通过PCManFM(也就是文件管理器)访问

如果需要挂载的话参照下面的方法，这样可以实现一些实用的功能，比如用 `crontab` 定时将树莓派备份到电脑硬盘上

Windows端:

1. 添加用户:

用户名: pi

密码:raspberry

2. 打开网络共享

3. 确保防火墙允许文件和打印机共享

4. 创建文件夹 **share** 并右击其选择 高级共享,为用户 pi添加读写权限

树莓派上:

```
sudo apt-get install cifs-utils
```

```
sudo mkdir /mnt/share
```

```
sudo nano /etc/fstab
```

添加 :

```
//your.ip.he.re/share /mnt/share cifs username=pi,password=raspberry,iocharset=utf8,se  
c=ntlm 0 0
```

重启

```
sudo reboot
```



## 第十三章-备份

备份有几种方法

- `dd` 或 Win32DiskImager生成 `.img`
- `rsync`

用的比较多的是来自官方论坛的一个 [脚本](#)，版权归属原作者jinx，属于第一种方法，只不过它直接帮你压缩了镜像，更方便一些

可配合`crontab`定时执行备份

命令示例：

```
sudo sh bkup_rpimage.sh start -cz /mnt/share/$(uname -n)-$(date +%Y-%m-%d).img
```

其中 `/mnt/share` 为备份目录，根据个人情况设置，可设置为NAS或其它位置 `-c` 为新建备份，`-z` 为启用压缩，会生成 `img.gz` 压缩文件，节约空间，其它选项可参看脚本

`rsync`的方法很简单，就一行命令

```
sudo rsync -aXv --exclude={"/dev/*","/proc/*","/sys/*","/tmp/*","/run/*","/mnt/*","/media/*","/lost+found"} / /path/to/backup/folder
```

## 第十四章-软件篇

接下来几章将列几个软件的安装



## 第十五章-Aria2下载(支持百度云)

本章内容参考自这篇文章

安装:

```
sudo apt-get install aria2
```

```
mkdir ~/.aria2
```

```
touch ~/.aria2/aria2.session
```

```
nano ~/.aria2/aria2.conf
```

添加:

```
# 基本配置
# 下载目录
dir=/home/pi
# 下载从这个文件中找到的urls, 需自己建立这个文件
# touch /home/pi/.aria2/aria2.session
input-file=/home/pi/.aria2/aria2.session
# 最大同时下载任务数, 默认 5
#max-concurrent-downloads=5
# 断点续传, 只适用于 HTTP(S)/FTP
continue=true

# HTTP/FTP 配置
# 关闭连接如果下载速度等于或低于这个值, 默认 0
#lowest-speed-limit=0
# 对于每个下载在同一个服务器上的连接数, 默认 1
max-connection-per-server=5
# 每个文件最小分片大小, 例如文件 20M, 设置 size 为 10M, 则用2个连接下载, 默认 20M
#min-split-size=10M
# 下载一个文件的连接数, 默认 5
#split=5

# BT 特殊配置
# 启用本地节点查找, 默认 false
bt-enable-lpd=true
# 指定最大文件数对于每个 bt 下载, 默认 100
#bt-max-open-files=100
# 单种子最大连接数, 默认 55
```

```
#bt-max-peers=55
# 设置最低的加密级别, 可选全连接加密 arc4, 默认是头加密 plain
#bt-min-crypto-level=plain
# 总是使用 obfuscation handshake, 防迅雷必备, 默认 false
bt-require-crypto=true
# 如果下载的是种子文件则自动解析并下载, 默认 true
#follow-torrent=true
# 为 BT 下载设置 TCP 端口号, 确保开放这些端口, 默认 6881-6999
listen-port=65298
#Set UDP listening port used by DHT(IPv4, IPv6) and UDP tracker
dht-listen-port=65298
# 整体上传速度限制, 0 表示不限制, 默认 0
#max-overall-upload-limit=0
# 每个下载上传速度限制, 默认 0
#max-upload-limit=0
# 种子分享率大于1, 则停止做种, 默认 1.0
#seed-ratio=1
# 做种时间大于2小时, 则停止做种
seed-time=120

# RPC 配置
# 开启 JSON-RPC/XML-RPC 服务, 默认 false
enable-rpc=true
# 允许所有来源, web 界面跨域权限需要, 默认 false
rpc-allow-origin-all=true
# 允许外部访问, 默认 false
rpc-listen-all=true
# rpc 端口, 默认 6800
rpc-listen-port=6800
# 设置最大的 JSON-RPC/XML-RPC 请求大小, 默认 2M
#rpc-max-request-size=2M
# rpc 密码, 可不设置
#rpc-passwd=raspberry
# rpc 用户名, 可不设置
#rpc-user=aria2pi

# 高级配置
# This is useful if you have to use broken DNS and
# want to avoid terribly slow AAAA record lookup.
# 默认 false
disable-ipv6=true
# 指定文件分配方法, 预分配能有效降低文件碎片, 提高磁盘性能, 缺点是预分配时间稍长
# 如果使用新的文件系统, 例如 ext4 (with extents support), btrfs, xfs or NTFS(MinGW build only), falloff 是最好的选择
# 如果设置为 none, 那么不预先分配文件空间, 默认 prealloc
file-allocation=none
# 整体下载速度限制, 默认 0
#max-overall-download-limit=0
# 每个下载下载速度限制, 默认 0
#max-download-limit=0
# 保存错误或者未完成的下载到这个文件
# 和基本配置中的 input-file 一起使用, 那么重启后仍可继续下载
save-session=/home/pi/.aria2/aria2.session
```

```
# 每5分钟自动保存错误或未完成的下载，如果为 0，只有 aria2 正常退出才回保存，默认 0
save-session-interval=300

# 若要用于 PT 下载，需另外的配置，这里没写
```

新建 aria2c 文件

```
sudo nano /etc/init.d/aria2c
```

添加:

```
#!/bin/sh
# /etc/init.d/aria2c

#### BEGIN INIT INFO
# Provides: aria2c
# Required-Start:    $network $local_fs $remote_fs
# Required-Stop:     $network $local_fs $remote_fs
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: aria2c RPC init script.
# Description: Starts and stops aria2 RPC services.
#### END INIT INFO

#VAR
RUN="/usr/bin/aria2c"
ARIA_PID=$(ps ux | awk '/aria2c --daemon=true --enable-rpc/ && !/awk/ {print $2}')
```

```
# Carry out specific functions when asked to by the system
case "$1" in
  start)
    echo "Starting script aria2c "
    if [ -z "$ARIA_PID" ]; then
      $RUN --daemon=true --enable-rpc=true -D --conf-path=/home/pi/.aria2/aria2.conf
      echo "Started"
    else
      echo "aria2c already started"
    fi
    ;;
  stop)
    echo "Stopping script aria2c"
    if [ ! -z "$ARIA_PID" ]; then
      kill $ARIA_PID
    fi
    echo "OK"
    ;;
  restart)
    echo "Restarting script aria2c"
    if [ ! -z "$ARIA_PID" ]; then
      kill $ARIA_PID
```

```

fi
sleep 3    # TODO:Maybe need to be adjust
$RUN --daemon=true --enable-rpc=true -D --conf-path=/home/pi/.aria2/aria2.conf
echo "OK"
;;
status)
if [ ! -z "$ARIA_PID" ]; then
    echo "The aria2c is running with PID = "$ARIA_PID
else
    echo "No process found for aria2c RPC"
fi
;;
*)
echo "Usage: /etc/init.d/aria2c {start|stop|restart|status}"
exit 1
;;
esac
exit 0

```

添加可执行权限:

```
sudo chmod +x /etc/init.d/aria2c
```

```
sudo update-rc.d aria2c defaults
```

```
sudo service aria2c restart
```

aria2 本身没有图形界面，需要另外安装

推荐使用 [Chrome](#)插件 [YAAW for Chrome](#)

安装完需要配置

- 点击 YAAW进入Web UI界面，点击右上方小扳手图标，在 `JSON-RPC Path` 栏填入 `http://raspberrypi:6800/jsonrpc`，然后点击 `Save` 保存
- 右击 YAAW的图标，点击 `选项`，在 `JSON-RPC` 的 `RPC Path` 栏填入 `http://raspberrypi:6800/jsonrpc`，然后点击 `Save` 保存

安装完成之后在右键菜单就能看到 `ARIA2 RPC` 选项，`Chrome`内右击需要下载的连接就可以添加到树莓派上进行下载，需要看下载进度就直接点击 YAAW的图标进 `Web UI`进行查看，对于BT下载，直接右键torrent的连接添加，下载完torrent会自动开始下载任务

至于百度云的下載需要[BaiduExporter](#)这个插件，安装配置请参考其项目主页；之后按上边那样配置RPC路径；安装完通过Chrome进入百度云，在 `上传` 这一栏出现 `导出下载` 按钮，选择 `ARIA2 RPC` 即可将选中的文件通过 aria2下载下来

这种方法下载百度云不限速，也可以同时下载多个文件，但不能下载文件夹

另外图形界面还有其它选择

[webui-aria2](#)

[YAAW](#)

[AriaNg](#)

对于以上几个项目都是先 `git clone` 下来，然后放到自己的网页目录下，最后设置 `RPC` 的路径

另外如果你没有公网IP的话，可以考虑用我写的这个[python脚本](#)，利用微信做远程下载

## 第十六章-FFmpeg

这里的安装方法启用了 `mmal`，可以提升性能

```
sudo apt-get update
```

```
sudo apt-get -y install autoconf automake build-essential libass-dev libfreetype6-dev \
    libSDL1.2-dev libtheora-dev libtool libva-dev libvdpau-dev libvorbis-dev libxcb-shm0
-dev \
    libxcb1-dev libxcb-shm0-dev libxcb-xfixes0-dev pkg-config texinfo zlib1g-dev
```

```
mkdir ~/ffmpeg_sources
```

Yasm:

```
sudo apt-get install yasm
```

libx264:

```
sudo apt-get install libx264-dev
```

libfdk-aac:

```
cd ~/ffmpeg_sources
```

```
wget -O fdk-aac.tar.gz https://github.com/mstorsjo/fdk-aac/tarball/master
```

```
tar xzvf fdk-aac.tar.gz
```

```
cd mstorsjo-fdk-aac*
```

```
autoreconf -fiv
```

```
./configure --prefix="$HOME/ffmpeg_build" --disable-shared
```

```
make -j4
```

```
make install
```

```
make distclean
```

libmp3lame:

```
sudo apt-get install libmp3lame-dev
```

libopus:

```
sudo apt-get install libopus-dev
```

ffmpeg:

```
sudo ldconfig
```

```
cd ~/ffmpeg_sources
```

```
wget http://ffmpeg.org/releases/ffmpeg-snapshot.tar.bz2
```

```
tar xjvf ffmpeg-snapshot.tar.bz2
```

```
cd ffmpeg
```

configure:

```
PATH="$HOME/bin:$PATH" PKG_CONFIG_PATH="$HOME/ffmpeg_build/lib/pkgconfig" \  
./configure --arch=armv7h --target-os=linux --enable-gpl --enable-libx264 --enable-ope  
ng1 \  
--enable-x11grab --enable-mmal --enable-nonfree --enable-libass \  
--enable-libfdk-aac --enable-libmp3lame --enable-libopus --bindir="/usr/bin"
```

make:

```
make -j4
```

```
sudo make install
```



## 第十七章-mpv

直接安装应该也行，不行的话再按下面的方法来

```
sudo apt-get install git devscripts equivs
```

```
git clone https://github.com/mpv-player/mpv-build.git
```

```
cd mpv-build
```

```
./update
```

```
rm -f mpv-build-deps_*.deb
```

```
mk-build-deps -s sudo -i
```

build package:

```
dpkg-buildpackage -uc -us -b -j4
```

安装:

```
sudo dpkg -i ../mpv*.deb
```

## 第十八章-命令行版网易云音乐

```
sudo apt-get install python-dev mpg123
```

```
sudo pip install Netease-MusicBox
```

如果报pip相关的错误，可能需要安装 `pip`：

```
sudo apt-get install python-pip
```

使用软件：

```
musicbox
```

上下移动：`J`、`K`

确认：`Enter`

播放/暂停：`Space`

其它操作参加其[Github页面](#)

## 第十九章-其它命令行程序

- 邮件: Mutt

安装:

```
sudo apt-get install mutt
```

安装中如果被询问 postfix 的配置，全选默认即可

配置:

```
sudo mkdir -p ~/.mutt/cache/headers
```

```
sudo mkdir ~/.mutt/cache/bodies
```

```
sudo touch ~/.mutt/certificates
```

```
sudo touch ~/.mutt/muttrc
```

**Tips:** 如果在Gmail帐户启用了“两步认证”，则需为Mutt单独生成一个密码，在下面添加帐户信息的时候就填这个专用密码

```
sudo nano ~/.mutt/muttrc
```

添加:

```
set ssl_starttls=yes
set ssl_force_tls=yes
set imap_user = '
change_this_user_name@gmail.com'
set imap_pass = 'PASSWORD'
set from='
change_this_user_name@gmail.com'
set realname='Your Name'
set folder = imaps://imap.gmail.com/
set spoolfile = imaps://imap.gmail.com/INBOX
set postponed="imaps://imap.gmail.com/[Gmail]/Drafts"
set header_cache = "~/.mutt/cache/headers"
set message_cachedir = "~/.mutt/cache/bodies"
set certificate_file = "~/.mutt/certificates"
set smtp_url = 'smtps://
change_this_user_name@gmail.com:
PASSWORD@smtp.gmail.com:465/'
set move = no
set imap_keeplive = 900
set record=""
```

发送邮件:

```
mutt -s "Subject" somejoeorjane@someserver.com
```

有帮助的链接: [Arch wiki](#)

---

- RSS: Newsbeuter

```
sudo apt-get install newsbeuter
```

```
mkdir ~/.newsbeuter
```

```
sudo nano ~/.newsbeuter/urls
```

添加RSS源:

<https://www.raspberrypi.org/feed/>

保存退出; 首次运行:

```
newsbeuter
```

按 **R** 重新加载

退出按 **q**

---

- BBS: Telnet

```
sudo apt-get install telnet
```

```
luit -encoding gbk telnet newsmth.net
```

## 第二十章-Nginx

安装:

```
sudo apt-get install nginx
```

启动:

```
sudo /etc/init.d/nginx start
```

此时你在浏览器内访问 `http://raspberrypi/` (或IP地址)可以看到默认的主页

假设你想配置 Aria2的Web UI，并已完成第十五章的准备工作

那么先下载 `webui-aria2` :

```
cd ~
```

```
git clone --depth=1 https://github.com/ziahamza/webui-aria2
```

`depth=1` 这个选项是只 `clone` 最新版本，适合 `commits` 比较多的repo

然后复制到 Nginx 默认的目录(Raspbian下是 `/var/www/html` )下

```
sudo cp -R /usr/share/nginx/html/webui-aria2/ /var/www/html/webui-aria2
```

此时在浏览器输入 `http://raspberrypi/webui-aria2` 即可访问 `webui-aria2`

使用以下命令查看当前默认的目录

```
grep www /etc/nginx/sites-available/*
```

如果你的主页是 `index.php`，参照官方的这篇[文档](#)进行设置

## 第二十一章-Powerline

基本就是美化终端的作用吧，有可能拖慢打开终端速度

```
sudo apt-get install python-pip
```

```
sudo pip install git+git://github.com/Lokaltog/powerline
```

```
wget https://github.com/powerline/powerline/raw/develop/font/PowerlineSymbols.otf
```

```
wget https://github.com/powerline/powerline/raw/develop/font/10-powerline-symbols.conf
```

```
mv PowerlineSymbols.otf /usr/share/fonts/
```

```
fc-cache -vf /usr/share/fonts/
```

```
mv 10-powerline-symbols.conf /etc/fonts/conf.d/
```

找 powerline位置:

```
pip show powerline-status
```

bash的配置:

```
nano ~/.bashrc
```

添加:

```
export TERM=xterm-256color
```

```
powerline-daemon -q
```

```
POWERLINE_BASH_CONTINUATION=1
```

```
POWERLINE_BASH_SELECT=1
```

```
. /usr/local/lib/python2.7/dist-packages/powerline/bindings/bash/powerline.sh
```

```
mkdir ~/.config/powerline
```

```
cp /usr/local/lib/python2.7/dist-packages/powerline/config_files/config.json ~/.config/powerline/config.json
```

```
nano ~/.config/powerline/config.json
```

定位到:

```
    "shell": {  
        ...  
        "theme": "default",
```

改为:

```
        "theme": "default_leftonly",
```

vim的配置:

```
sudo apt-get install vim-nox
```

```
nano ~/.vimrc
```

添加:

```
set rtp+=/usr/local/lib/python2.7/dist-packages/powerline/bindings/vim/  
set laststatus=2  
set t_Co=256
```

tmux的配置:

```
sudo apt-get install tmux
```

```
nano ~/.tmux.conf
```

```
source /usr/local/lib/python2.7/dist-packages/powerline/bindings/tmux/powerline.conf  
set-option -g default-terminal "xterm-256color"
```





## 第二十二章-DNSPod

需注册DNSPod的账号，如果做Server的树莓派IP总是变动的话，可以用这个实时更新域名的IP

脚本来自[github](#)，版权归属原作者[chuangbo](#)，以下说明引自README

替换上你的Email，密码，域名ID，记录ID等参数，就可以运行了。会在后台一直运行，每隔30秒检查一遍IP，如果修改了就更新IP。

获得domain\_id可以用curl curl -k <https://dnsapi.cn/Domain.List> -d "login\_email=xxx&login\_password=xxx"

获得record\_id类似 curl -k <https://dnsapi.cn/Record.List> -d "login\_email=xxx&login\_password=xxx&domain\_id=xxx"

下载脚本

```
wget https://gist.githubusercontent.com/chuangbo/833369/raw/d9d66084abeaa6cacaee88066fa50bad2f98e422a/pypod.py
```

开机自运行：

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

在

```
exit 0
```

前添加：

```
python /usr/bin/pypod.py
```

不过似乎自启和Gitlab有冲突

## 第二十三章-微信远程遥控树莓派

自己写的一个简单脚本，基于 [littlecoder](#) 的 [ItChat](#) 框架，能实现基本的控制，大部分命令行的命令应该都可用，比较适合无公网IP的情况下作远程控制，欢迎到[Github页面](#)反馈报错

脚本下载：

```
git clone https://github.com/littlecodersh/ItChat
```

```
cd ItChat
```

```
wget https://raw.githubusercontent.com/yangxuan8282/wechat-remote/master/wechat_remote.py
```

运行：

```
python wechat_remote.py
```

具体功能的安装及使用见项目的说明：<https://github.com/yangxuan8282/wechat-remote/blob/master/README-zh.md>

基于安全考虑，默认只接受自己发送的指令（也就是从手机微信上发送），因为包括 `sudo` 命令在内的指令也可用；部分账号无法给自己发送信息，需用其它账号发送，并注释掉相关代码（在项目说明 [Issues](#) 部分有具体说明）

比较适合远程下载，也可以播放在线视频

特别说明一下 `apt-get` 命令也可以执行，但不建议这样做，因为返回结果较慢，而 `subprocess` 需要 "Wait for process to terminate."

## 第二十四章-写个爬虫(Scrapy)

跑爬虫比较接近电影里对Hacker印象，输入几行代码，程序就可以运行很久

刚好树莓派已经预装了Python2，可以在树莓派上体验一下，另外爬虫确实很实用，可以高效搜集很多信息，比复制粘贴强了很多

而且Scrapy门槛很低，几个小时就能上手，得到你想要的数据库

安装：

```
sudo apt-get install libffi-dev
```

```
sudo apt-get install libxml2-dev
```

```
sudo apt-get install libxslt1-dev
```

```
sudo apt-get install python-dev
```

```
sudo apt-get install libssl-dev
```

安装配置virtualenv：

```
sudo pip install virtualenv
```

安装virtualenvwrapper：

```
sudo mkdir ~/scrapy_spider
```

```
sudo pip install --user virtualenvwrapper
```

```
export WORKON_HOME=~/.Envs
```

```
mkdir -p $WORKON_HOME
```

```
source /usr/local/bin/virtualenvwrapper.sh
```

创建名为scrapy的virtualenv

```
mkvirtualenv scrapy
```

配置完之后会终端行首会显示 `(scrapy)$`，说明处于名为scrapy的virtualenv

```
ls $WORKON_HOME
```

使workon在重启后依然有效：

```
sudo nano ~/.bashrc
```

添加以下内容至末尾：

```
export WORKON_HOME=$HOME/Envs
source /usr/local/bin/virtualenvwrapper.sh
```

```
source ~/.bashrc
```

virtualenv内安装scrapy：

```
(scrapy)$ sudo pip install scrapy
```

进入名为scrapy的virtualenv：

```
workon scrapy
```

退出virtualenv：

```
deactivate
```

简单爬虫示例：

单个的爬虫文件最关键是 `start_urls`，以及返回的内容

比如你想抓取北京所有的7-11门店名称及位置

先登录官网[店铺信息](#)页面，一共有8个区有店，鼠标悬停观察每个区的url，这样 `start_urls` 就确定了

需要先获取每个门店的url，用Chrome浏览器右键 [审查](#) 门店链接，弹出窗口内右击相应代码位置 `Copy > Copy XPath`

得到：

```
...tr[1]/th/a
```

所以url表达式为：

```
response.xpath('..//tr/th/a/@href')
```

补全即可得到完整url：

```
response.urljoin(href.extract())
```

然后进入单个门店url查看想获取信息的XPath

门店名称为：

```
//*[@id="mainContentsBox"]/h4
```

门店位置（百度地图坐标）在js代码内：

```
//*[@id="mainContentsBox"]/script
```

用Shell测试是否能抓取店名及位置（百度地图坐标）：

```
scrapy shell http://www.7-11bj.com.cn/?post/cq/223/id/270.html --nolog
```

输入：

```
response.xpath('//div[@id="mainContentsBox"]/h4/text()').extract()[0]
```

能够返回店名

百度地图坐标藏在js代码内，需用[正则表达](#)提取

```
response.xpath('//*[@id="mainContentsBox"]/script').re(r'BMap.Point\(([0-9.]+),([0-9.]
+)\)\')
```

能够返回位置（百度地图坐标），7-11北京官网的一部分坐标似乎本身有问题，并不是爬虫的关系，直接去url内查看源码也是一样的结果

创建单个爬虫：

```
cd ~/scrapy_spider
```

```
workon scrapy
```

```
sudo nano ~/scrpay_spider/711bj.py
```

### 711bj.py

```
# -*- coding: utf-8 -*-
import scrapy

class StackOverflowSpider(scrapy.Spider):
    name = '711bj'
    start_urls = ['http://www.7-11bj.com.cn/?chengqu/cq/223.html'] + \
        ['http://www.7-11bj.com.cn/?chengqu/cq/224.html'] + \
        ['http://www.7-11bj.com.cn/?chengqu/cq/225.html'] + \
        ['http://www.7-11bj.com.cn/?chengqu/cq/226.html'] + \
        ['http://www.7-11bj.com.cn/?chengqu/cq/227.html'] + \
        ['http://www.7-11bj.com.cn/?chengqu/cq/228.html'] + \
        ['http://www.7-11bj.com.cn/?chengqu/cq/242.html'] + \
        ['http://www.7-11bj.com.cn/?chengqu/cq/243.html']

    def parse(self, response):
        for href in response.xpath('..tr/th/a/@href'):
            full_url = response.urljoin(href.extract())
            yield scrapy.Request(full_url, callback=self.parse_store)

    def parse_store(self, response):
        yield {
            'title': response.xpath('//div[@id="mainContentsBox"]/h4/text()').extract(
)[0],
            'BMap.Point': response.xpath('//*[@id="mainContentsBox"]/script').re(r'BMa
p.Point\(([0-9.]+),([0-9.]+)\)'),
            'link': response.url,
        }
```

保存退出 `CTRL + X` ,提示是否保存按 `y`

运行爬虫：

```
scrapy runspider 711bj.py -o test711bj.csv
```

查看结果：

```
cat ~/scrapy_spider/test711bj.csv
```

这当然是非常简单的一个爬虫，推荐去看[官方文档](#)，比如这个[例子](#),你看懂这个例子后几分钟就能写出自己的爬虫，适用于大部分网站。官方文档写的非常完善，很多时候你搜索Scrapy的问题都会被链接到这里

另外推荐一个[视频](#)，这是[Eloy Zuniga Jr.](#)一次讲座的音频修复版，时长约一个多小时，完全看懂的话，写爬虫基本就没什么问题了

爬虫可配合 `crontab` 定时爬取，比如想每月爬一次7-11在北京的店铺信息

```
crontab -e
```

首次编辑 `crontab` ，会被要求选择编辑器，一般默认（Nano）即可，在文件末加入：

```
0 0 1 * * env -i bash -c 'export WORKON_HOME=~/.Envs && source /usr/local/bin/virtualenvwrapper.sh && source ~/.Envs/scrapy/bin/activate && cd ~/scrapy_spider && scrapy runspider 711bj.py -o 711bj_$(date +%Y%m%d).csv && deactivate'
```

如果爬虫被封，需要用IP代理或者加入Header

另外发现 `crontab` 调用要比直接运行爬虫速度快，可能是不用再终端显示的原因



## 第二十五章-硬件篇

下几章都是硬件相关

## 第二十六章-连接显示器及LCD

可选：HDMI输出，3.5mm视频输出，其它转接方式，以及[官方显示屏](#)

- HDMI最方便，直接连接即可

需要注意的是，开机时如果没连接显示器便不会有视频输出，即使开机后连接了显示器也是一样的结果；如果想强制视频输出的话，需要修改 `config.txt` 文件，假设你的显示器是HDMI接口1366x768分辨率

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

添加：

```
hdmi_force_hotplug=1
hdmi_group=2
hdmi_mode=16
hdmi_drive=2
```

这里 `hdmi_force_hotplug=1` 是强制输出到HDMI，即使未检测到HDMI设备

- 3.5mm视频输出

可以连接CRT旧电视，接口集成在音频口，所以直接在音频口接耳机会发现底噪很大，不过社区正在[改进](#)这点

可以直接淘宝购买Zune的RCA Cable，或者DIY接口定义一致的线材，定义参考下图，图片版权归属原作者[MATT](#)：

# Raspberry Pi Model B+ 3.5mm Audio/Video Socket

| Device      | Sleeve | Ring 2 | Ring 1 | Tip  | OK? |
|-------------|--------|--------|--------|------|-----|
|             |        |        |        |      |     |
|             | 4      | 3      | 2      | 1    |     |
| Model B+    | Video  | Ground | Right  | Left | ✓   |
| Apple       | Video  | Ground | Right  | Left | ✓   |
| Zune        | Video  | Ground | Right  | Left | ✓   |
| Camcorders  | Right  | Ground | Video  | Left | ⚠   |
| MP3 Players | Ground | Video  | Right  | Left | ✗   |

www.raspberrypi-spy.co.uk

另外需要修改 config.txt （PAL制式电视）：

（新版的Jessie不用修改也能显示，如果没声音的话再进行以下操作）

树莓派上：

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

或Windows上打开 config.txt

将

```
#sdtv_mode=2
```

前面的#删掉，变为：

```
sdtv_mode=2
```

配合老电视和 Xbox 360手柄或8位堂完全仿红白机的FC30手柄怀旧FC是相当不错的选择，系统方面可以选择[Lakka](#) , [RetroPie](#) , [recalbox-os](#)

- HDMI转接VGA

但[多篇文章](#)都建议一定用有源的HDMI转VGA，也就是可以连接电源的，否则会烧毁树莓派

修改 `config.txt` 文件，添加：

```
hdmi_force_hotplug=1
hdmi_group=2
hdmi_mode=16
hdmi_drive=2
```

`hdmi_mode=16`为设置1024x768 60Hz 其它分辨率可参考官方[config.txt说明](#)

一些有源HDMI转VGA的选择：

[白色款1](#)，[白色款2](#)，[黑色款1](#)，[黑色款2](#)，[Moto](#)，（似乎这款最便宜，需要另外配一个Micro HDMI母对HDMI公，比如[这个](#)）

- Nokia 5510/3310 LCD

这块屏幕的尺寸约1.5寸，分辨率为84x48，不论字体采用8x8还是6x12显示的内容都很有限，而且因为是拆机屏，质量难以保证 诺基亚5510/3310的液晶模块controller一般为PCD8544，接口为 [SPI](#)

连线：

| LCD | Driver gpio name | Raspberry Pi | 备注           |
|-----|------------------|--------------|--------------|
| GND |                  | GND          |              |
| VCC |                  | 3.3V         |              |
| CLK |                  | SCLK         |              |
| DIN |                  | MOSI         |              |
| D/C | dc               | GPIO24       | Command/data |
| CS  |                  | CE0          | Chip Select  |
| RST | reset            | GPIO25       |              |
| LED | led              | GPIO23       | led          |

然后通过：

```
gpio readall
```

查阅附录2查看树莓派上对应的端口

对照 `BCM` 及 `Physical` 栏，依次找到连线表内的对应端口

下面是一种接法：

| LCD | Raspberry Pi(Physical) |
|-----|------------------------|
| GND | 9                      |
| BL  | 16                     |
| VCC | 1                      |
| CLK | 23                     |
| Din | 19                     |
| DC  | 18                     |
| CE  | 24                     |
| RST | 22                     |

启用 `SPI`：

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

加入：

```
dtparam=spi=on
```

重启生效

这块单色LCD可以通过 `FBTFT` 进行console显示，这样在终端里能做的事用它都可以做到  
由于Linux kernel staging tree已经加入FBTFT，所以不需额外安装

终端显示：

```
sudo modprobe fbtft_device name=nokia3310
```

`dmesg` 会显示：

```

.....
[ 52.092954] fbtf: module is from the staging directory, the quality is unknown, you have been warned.
[ 52.098978] fbtf_device: module is from the staging directory, the quality is unknown, you have been warned.
[ 52.101017] fbtf_device: SPI devices registered:
[ 52.101040] fbtf_device: spidev spi0.0 500kHz 8 bits mode=0x00
[ 52.101053] fbtf_device: spidev spi0.1 500kHz 8 bits mode=0x00
[ 52.101062] fbtf_device: 'fb' Platform devices registered:
[ 52.101092] fbtf_device: soc:fb id=-1 pdata? no
[ 52.101132] fbtf_device: Deleting spi0.0
[ 52.101429] spi spi0.0: setting up native-CS0 as GPIO 8
[ 52.101544] fbtf_device: GPIOs used by 'nokia3310':
[ 52.101555] fbtf_device: 'reset' = GPIO25
[ 52.101564] fbtf_device: 'dc' = GPIO24
[ 52.101572] fbtf_device: 'led' = GPIO23
[ 52.101580] fbtf_device: SPI devices registered:
[ 52.101592] fbtf_device: spidev spi0.1 500kHz 8 bits mode=0x00
[ 52.101603] fbtf_device: fb_pcd8544 spi0.0 400kHz 8 bits mode=0x00
[ 52.113572] fb_pcd8544: module is from the staging directory, the quality is unknown, you have been warned.
[ 52.253490] graphics fb1: fb_pcd8544 frame buffer, 84x48, 7 KiB video memory, 0 KiB DMA buffer memory, fps=20, spi0.0 at 0 MHz

```

*lsmod* 会显示：

| Module      | Size  | Used by                  |
|-------------|-------|--------------------------|
| fb_pcd8544  | 3048  | 0                        |
| fbtf_device | 27354 | 0                        |
| fbtf        | 27484 | 2 fb_pcd8544,fbtf_device |
| syscopyarea | 2773  | 1 fbtf                   |
| sysfillrect | 3299  | 1 fbtf                   |
| sysimgblt   | 1921  | 1 fbtf                   |
| fb_sys_fops | 1157  | 1 fbtf                   |

启用终端：

```
con2fbmap 1 1
```

(如果遇到屏幕正面看是black screen，在别的角度可以看到文字的话，这和屏幕本身连接方式有关，其液晶与电路板是用导电胶连接的，可能没有充分接触，稍微压一下边框应该可以正常显示；如果LED闪得厉害，多半是屏幕有问题，建议找卖家换一个)

## 第二十七章-关于外壳和温度

外壳推荐[官方](#)，因为3代相比前代取消了Micro SD卡的弹出式设计，这样需要在卡的位置预留更大空间才能方便取出，取消的原因据说是部分用户反映弹出式可能把卡弹飞

另外有[用户](#)说3代的树莓派要比前代热了很多

不过在日常使用中并没有那么严重，目前实际使用中大部分时间在50°以下，除非超频或者持续满负荷运作，一般来讲加装散热片足够，风扇可选但并不必须

散热片最好选铜材料，因为常用导热材料里它的导热系数最高，铝质也可，尺寸14mm\*14mm,另外如果你很想尝试新材料的话可以考虑Versarien的这款[泡沫结构散热块](#)，散热效果未知

查看树莓派温度：

```
/opt/vc/bin/vcgenclmd measure_temp
```

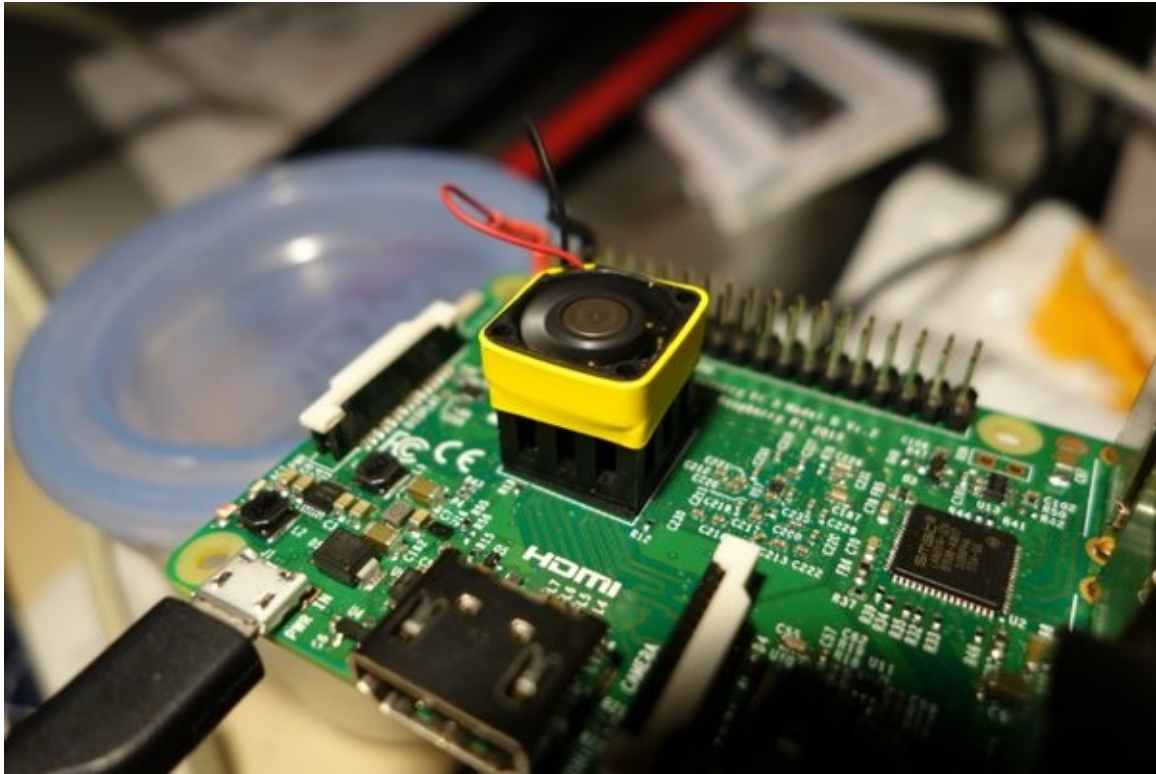
如果想持续检测温度：

```
watch /opt/vc/bin/vcgenclmd measure_temp
```

退出： `CTRL + c`

如果你的树莓派大部分时间都很烫的话，比如总是在右上角出现红色小方块(这是系统的温度报警，说明已经超过85度)，可以考虑风扇加散热片

以下是一个不错的方案，产品来自[RaspberryPi4u](#)，图片版权归属原作者[Gareth Halfacree](#)



可惜这位卖家已经不做子(已经重操旧业，不过价格翻了好几倍，标99美元)，想要类似产品的  
话可以DIY，风扇选购[SEPA 1505](#)，这是一款15mm\*15mm的风扇，可以接3.3V或5V，参照  
[GPIO定义](#)，红正黑负，红色接5V输出，黑色接GND

芯片与散热块之间用 Laird 的相变硅脂连接，粘度还可以，这个产品特点是高温(65度以上)散  
热效果比较好，比散热块自带的导热胶多降5度左右；首次使用得先加热到60度以上2小时左  
右才能看出散热效果，

风扇与散热块用热缩管固定，截取一段热缩管套在散热块与风扇连接处，用打火机加热后热  
缩管因其特性会收缩；



## 第二十八章-Lapdock

Lapdock是Moto出的一款硬件，类似笔记本的外观，配有电池，显示器，键盘，触摸板，和树莓派非常搭

我是2016年5月淘宝购入Lapdock，400接近全新(也可能是翻新)，不过用一个月坏掉了，免费换了一个使用至今没有任何问题，后来电源用的是青米出的多口充电器(双U口可输出20V 3A，需配个5.5转4.0x1.7的转接头)，树莓派同样由它供电，还可以另外带一个9V的路由器

树莓派连接Lapdock需要的两条转接线会在下面给出链接(不是广告，同时卖这两个线的店铺淘宝好像就一家，Lapdock能稳定供货的好像也就一家，现在不知道有没有了)

需要一个Micro HDMI转HDMI 母转公用作视频输出，比如[这个](#)，一个USB 2.0 A公对MICRO B母作数据传输(Lapdock的键盘及USB口设备)，比如[这个](#)

如果用Moto Lapdock的话，可能发现没有声音，需要做以下改动：

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

找到：

```
hdmi_drive=2
```

去掉前面的#使其生效

另外Lapdock的音量键需配置才能在 Raspbian 上使用

```
sudo nano ~/.config/openbox/lxde-pi-rc.xml
```

使用 `CTRL + w` 搜索 `keyboard`

定位到：

```
<keyboard>
.....
```

添加：

```
<keybind key="XF86AudioLowerVolume">
  <action name="Execute">
    <command>amixer set PCM 250-</command>
  </action>
</keybind>
<keybind key="XF86AudioRaiseVolume">
  <action name="Execute">
    <command>amixer set PCM 250+</command>
  </action>
</keybind>
```

保存 **CTRL + o**

退出 **CTRL + x**

之后在Pi的终端内运行：

```
openbox --reconfigure
```

## 第二十九章-真的没有彩蛋

```
telnet towel.blinkenlights.nl
```

## 附录1-树莓派3可以用的系统

树莓派没有BIOS，所以未插卡接显示器不会有任何显示(网络启动的话不用卡也行)

由于有活跃的社区支持，使得可选系统很丰富，最新资料请参照[官网](#)

桌面系统

OS	下载	附加信息
RASPBIAN JESSIE	<a href="#">镜像</a> ； <a href="#">安装说明</a>	官方系统，无论性能还是支持方面都很突出
Kali Linux	<a href="#">镜像</a>	支持内置Wi-Fi
Fedora	<a href="#">下载页面</a> ； <a href="#">安装</a> 参照 <a href="#">wiki</a>	XFCE；LXDE；MATE可以运行
KANO OS	<a href="#">NOOBS</a>	儿童教学向
Xubuntu	<a href="#">镜像</a>	16.04；支持内置Wi-Fi、蓝牙
Lubuntu	<a href="#">镜像</a>	16.04
Ubuntu Mate	<a href="#">下载</a>	16.04；稍显臃肿，运行速度较Raspbian逊色很多；支持蓝牙及内置Wi-Fi
Deepin15	<a href="#">镜像</a> ； <a href="#">来源</a>	用户名密码均为pi；解压后写入镜像即可
RaspEX	<a href="#">官网</a>	基于Debian，Ubuntu
Chromium OS	<a href="#">2GB版本</a> ； <a href="#">16GB版本</a>	访问Google站点还不错（包括Youtube）
（Manjaro）	<a href="#">官网</a> ； <a href="#">安装说明</a>	建议用archlinux；Base版本提供XFCE；支持内置Wi-Fi；软件升级后默认主题显示有问题，建议安装 <a href="#">主题</a>
（OpenSuse）	<a href="#">下载</a> ； <a href="#">安装说明</a>	未能成功启动，而且反馈来看不是个别现象，问题已经存在数年之久，不建议安装；如想尝试需自备USB转TTL设备

媒体中心

OS	下载	附加信息
OSMC	<a href="#">镜像</a>	界面很漂亮，支持部分电视遥控器
OpenELEC	<a href="#">下载页面</a>	
XBian	<a href="#">sourceforge</a>	
RuneAudio	<a href="#">官网</a>	音乐播放

## 游戏中心

OS	下载	附加信息
recalbox	<a href="#">下载页面</a>	<a href="#">安装说明</a>
Lakka	<a href="#">下载</a>	XMB界面；支持内置Wi-Fi

如果你有一台配备GTX 600/700/900系列显卡的PC，可以使用[Moonlight](#)将画面串流到树莓派3上

## Server

OS	下载	附加信息
RASPBIAN JESSIE LITE	<a href="#">镜像</a>	
MINIBIAN	<a href="#">sourceforge</a>	各方面接近Jessie Lite
Dietpi	<a href="#">官网</a>	<a href="#">DietPi vs Raspbian Lite</a>
Ubuntu Server	<a href="#">standard</a> 、 <a href="#">minimal</a>	16.04
Arch Linux	<a href="#">镜像</a>	安装参照 <a href="#">Installation</a>
CentOS	<a href="#">下载</a> ； <a href="#">官方说明</a>	
Manjaro	<a href="#">官网</a>	SERVER EDITION
OwnCloud	<a href="#">下载</a>	基于Ubuntu Core； <a href="#">安装说明</a> （参照 'Where are we now' 部分最后一句）

多系统启动工具：[Berryboot](#)

更多选择：[http://elinux.org/RPi\\_Distributions](http://elinux.org/RPi_Distributions)

以上可用系统均支持树莓派3且仍在更新，至于树莓派2可能有更多选择

虽然树莓派3有一颗64位的处理器，但是系统目前仍为32bit，官网论坛的xylhao提供了64位Debian[镜像](#)

64位的一些进展：[FreeBSD](#)（需用串口通信），[U-Boot](#)，[官网论坛](#)，[Gentoo论坛](#)

另外关于 Win10 IoT，如果不是做硬件开发的话不建议安装，因为和普通的Win10完全是两个概念，程序需要自己开发，确实有GUI，但和你想的那种完全不同

## 附录2-GPIO

-----Pi 3-----												
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM		
		3.3v			1	2		5v				
2	8	SDA.1	IN	1	3	4		5V				
3	9	SCL.1	IN	1	5	6		0v				
4	7	GPIO. 7	IN	1	7	8	1	ALT5	TxD	15	14	
		0v			9	10	1	ALT5	RxD	16	15	
17	0	GPIO. 0	IN	0	11	12	0	IN	GPIO. 1	1	18	
27	2	GPIO. 2	IN	0	13	14		0v				
22	3	GPIO. 3	IN	0	15	16	1	OUT	GPIO. 4	4	23	
		3.3v			17	18	1	OUT	GPIO. 5	5	24	
10	12	MOSI	ALT0	0	19	20		0v				
9	13	MISO	ALT0	0	21	22	1	OUT	GPIO. 6	6	25	
11	14	SCLK	ALT0	0	23	24	1	OUT	CE0	10	8	
		0v			25	26	1	OUT	CE1	11	7	
0	30	SDA.0	IN	1	27	28	1	IN	SCL.0	31	1	
5	21	GPIO.21	IN	1	29	30		0v				
6	22	GPIO.22	IN	1	31	32	0	IN	GPIO.26	26	12	
13	23	GPIO.23	IN	0	33	34		0v				
19	24	GPIO.24	IN	0	35	36	0	IN	GPIO.27	27	16	
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28	20	
		0v			39	40	0	IN	GPIO.29	29	21	
-----Pi 3-----												
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM		

命令行查看(需安装[WiringPi](#)):

```
gpio readall
```

## 附录3-购买相关

至于购买应该渠道挺多的

淘宝，[e络盟](#)，[欧时](#)都有

[e络盟](#)和[欧时](#)两个版本应该区别不大，之前几代两家都用过，质量都不错

基本选便宜的购买就好，因为至今没听说有假货，冒牌的倒是有，不过差太多，另外淘宝的话不建议买套餐，配的壳子和电源都是很差的那种，最好自己挑；加原装壳差不多一共220，230

需要注意的是3代和2B壳不完全通用，主要是指指示灯的位置换了

电源的话不一定要另外买，靠谱的手机充电器就可以；官方那个性价比不太高

其它很多配件都没用过，具体怎样也不清楚



## 附录4-有用的链接

这本gitbook的很多信息都是来自下面的链接

- 官方文档：[www.raspberrypi.org/documentation](http://www.raspberrypi.org/documentation)
- 官方论坛：[www.raspberrypi.org/forums](http://www.raspberrypi.org/forums)
- 故障排除：[R-Pi\\_Troubleshooting](#)
- 官方config.txt说明：[config-txt.md](#)
- [elinux.org](http://elinux.org)
- [wiki.archlinux.org](http://wiki.archlinux.org)
- [www.reddit.com/r/raspberry\\_pi](http://www.reddit.com/r/raspberry_pi)