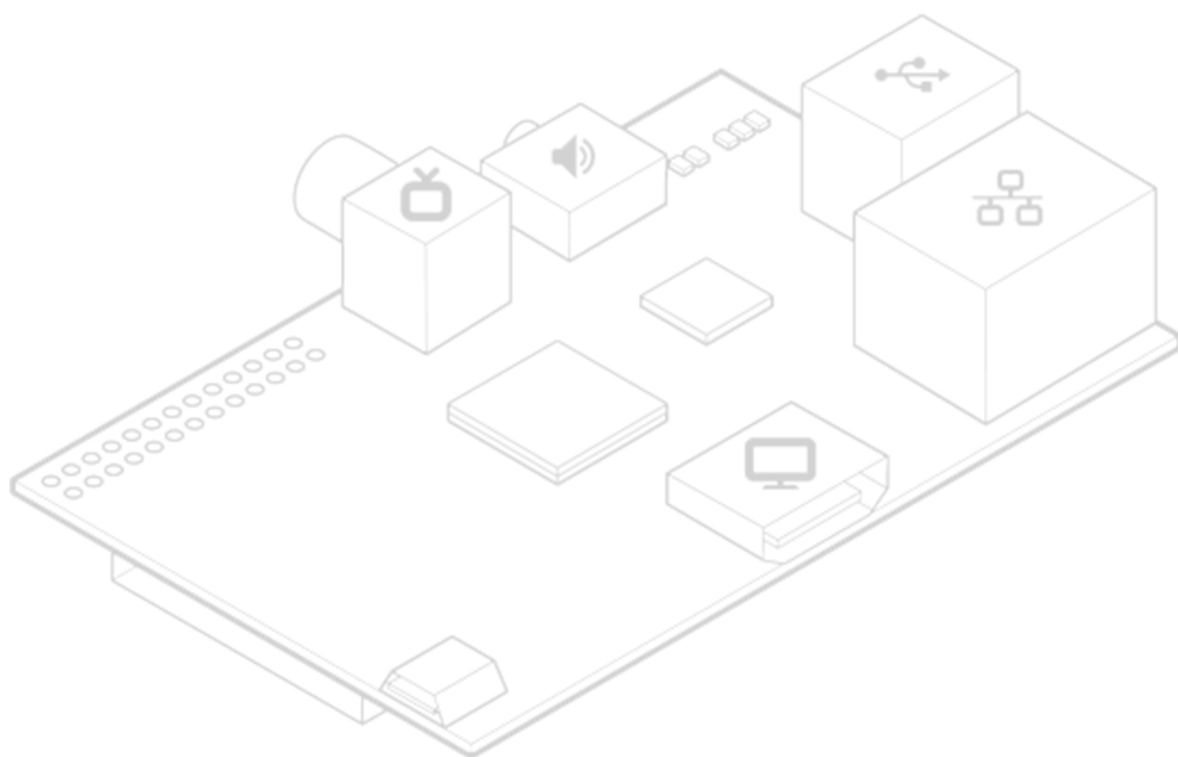
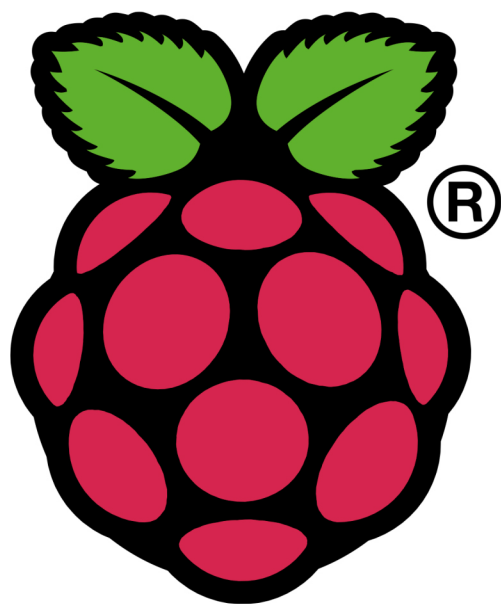


只有信用卡大小的卡片式电脑



树莓派



极客学院出版

前言

树莓派（Raspberry Pi）是尺寸仅有信用卡大小的一个小型电脑，您可以将树莓派连接电视、显示器、键盘鼠标等设备使用。目前，树莓派能替代日常桌面计算机的多种用途，包括文字处理、电子表格、媒体中心甚至是游戏。并且树莓派还可以播放高至 1080p 的高清视频。

本指南由浅至深的讲解了关于树莓派的常用功能及配置，能够帮助读者成功搭建基础开发环境，并对其中可能出现的问题做出详细讲解，帮助初学者解决实际问题。带领读者进行更多有趣的探索，以玩转树莓派。

适合人群

本指南覆盖了树莓派的许多主题，适合程序员、计算机软硬件爱好者，以及对树莓派感兴趣的读者阅读，也适合作为树莓派相关实践课程的基础教程。

预备知识

你需要对 Linux 及虚拟机有一定的了解。

目录

前言	1
第 1 章 树莓派介绍	3
第 3 章 树莓派配置	13
第 4 章 树莓派网络与更新配置	18
第 5 章 树莓派常用软件及服务	23
第 6 章 树莓派做 wifi 热点	29
第 7 章 树莓派做 web 服务器	36
第 8 章 树莓派内核编译与固件升级	41
第 9 章 树莓派 GPIO 控制	47
第 10 章 树莓派 spi 液晶屏支持	54



树莓派介绍



树莓派是什么？

Raspberry Pi(中文名为“树莓派”,简称为 RPi, 或者 RasPi/RPi)是为学生计算机编程教育而设计, 只有信用卡大小的卡片式电脑, 其系统基于 Linux。

树莓派由注册于英国的慈善组织“Raspberry Pi 基金会”开发, Eben · Upton /埃 · 厄普顿为项目带头人。2012 年 3 月, 英国剑桥大学埃本 · 阿普顿 (Eben Epton) 正式发售世界上最小的台式机, 又称卡片式电脑, 外形只有信用卡大小, 却具有电脑的所有基本功能, 这就是 Raspberry Pi 电脑板, 中文译名“树莓派”! 这一基金会以提升学校计算机科学及相关学科的教育, 让计算机变得有趣为宗旨。基金会期望这一款电脑无论是在发展中国家还是在发达国家, 会有更多的其它应用不断被开发出来, 并应用到更多领域。

一句话: 树莓派是一个卡片大小的开发板, 上面可以运行 Linux 系统, 我们可以用它开发我想要的设备。

树莓派的种类

- A 型: 1 个 USB、无有线网络接口、功率 2.5W, 500mA、256MB RAM (基本已经见不到了)。
- B 型: 2 个 USB、支持有线网络、功率 3.5W, 700mA、512MB RAM、26个GPIO (市售还有很多)。
- B+ 型: 4 个 USB 口、支持有线网络, 功耗 1W, 512M RAM 40个GPIO (2014 新出的, 推荐使用)。

购买请在淘宝上搜一下, 非常多: 200到--300元之间 (裸板, 不含 SD 卡、电源)

A 型

B 型

B+ 型

树莓派参数

B 型

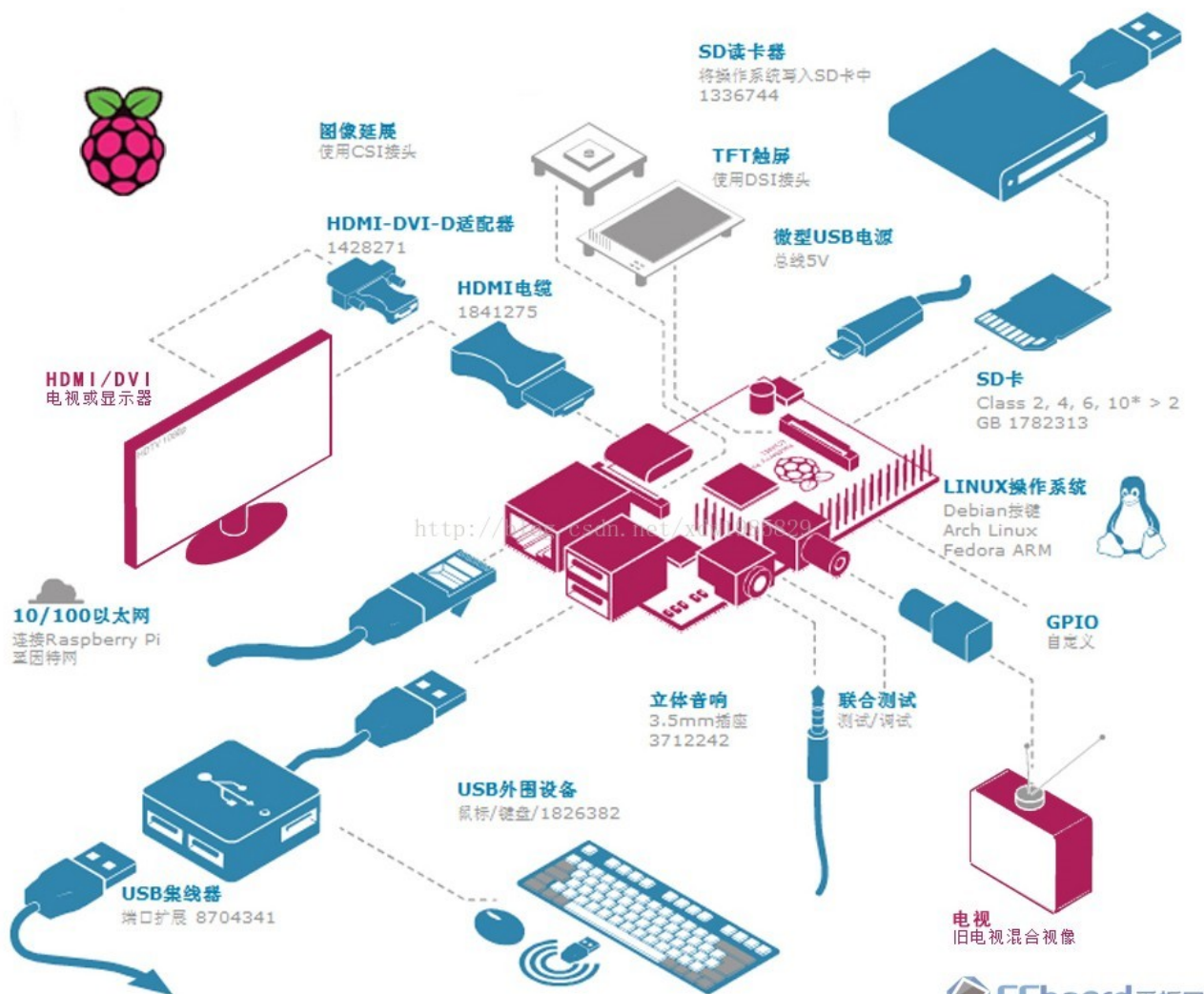
|简介|具体描述| |:----|:-----| |处理器| BroadcomBCM2835 (CPU, GPU, DSP和SDRAM, USB) ||C
PU|ARM1176JZF-S 核心 (ARM11 系列) 700MHz| GPU Broadcom VideoCore IV, OpenGL ES 2.0,10
80p 30 h.264/MPEG-4 AVC 高清解码器| |内存|512MByte| |USB 2.0|2 (支持 USB hub 扩展) | |影像输出
|Composite RCA (PAL & NTSC) , HDMI (rev 1.3 & 1.4) , raw LCD Panels via DSI 14 HDMI resoluti
on from 640x350 to 1920x1200 plus various PAL andNTSC standards| |音源输出| 3.5mm 插孔, HDMI|
|板载存储| SD/MMC/SDIO 卡插槽| |网络接口|10/100 以太网接口| |外设|8xGPIO、UART、I2C、带两个选择
的 SPI 总线, +3.3V, +5V, ground (负极) | |额定功率| 700mA (3.5W) | |电源输入|5V / 通过 MicroUSB
或 GPIO 头| |总体尺寸|85.60 x 53.98 mm (3.370 x 2.125 in) | |操作系统|Debian GNU/linux, Fedora, A
rch Linux ARM, RISC OS, XBMC|

B+ 改进

- 更多的 GPIO: B+ 将通用输入输出引脚增加到了 40 个, 而 Model B 则只有 26 个;
- 更多的 USB: B+ 提供了 4 个 USB 端口, 对热插拔有着更好的兼容性(Model B 只有 2 个); -支持 micro SD: 旧款的 SD 卡插槽, 已经被换成了更漂亮的推入式microSD 卡槽;
- 更低的功耗: 将线性式稳压器换成了开关式, 功耗降低了 0.5W 到 1W;
- 更好的音频: 音频电路部分采用了专用的低噪供电;
- 简洁的外形: USB 接口被推到了主板的一边, 复合视频移到了 3.5mm 音频口的位置, 此外还增加了 4 个独立的安装孔。

树莓派分布图

B 型, B+ 类似, 请自行对比参考





T



树莓派启动方式及支持的系统

树莓派开发板没有配置板载 FLASH，因为它支持 SD 卡启动，所有我们需要下载相应镜像，并将其烧写在 SD 上，启动系统即可（这个镜像里包含了我们通常所说的 bootloader、kernel、文件系统）树莓派由于其开源特性，支持非常多的系统类型（指的文件系统）：

Raspbian、Arch Linux ARM、Debian Squeeze、Firefox OS、Gentoo Linux Google Chrome OS、Raspberry Pi Fedora Remix、Slackware ARM QtonPi、Slackware ARM、WebOS、RISC OS、FreeBSD NetBSD、Android 4.0(Ice Cream Sandwich)

详细介绍请看维基百科：<http://zh.wikipedia.org/wiki/%E6%A0%91%E8%8E%93%E6%B4%BE>

树莓派镜像下载地址

<http://www.raspberrypi.org/>

官网（速度较慢）

<http://pan.baidu.com/share/home?uk=671504480#category/type=0>

树莓派论坛提供的下载地址（百度网盘速度较快）

建议下载：wheezy-raspbian 笔者更新时，最新版本为：2014.6.20(B+ 版建议使用此版本，笔者将旧版烧入 B+ 版中貌似网卡及 USB 驱动不能识别)

镜像烧写

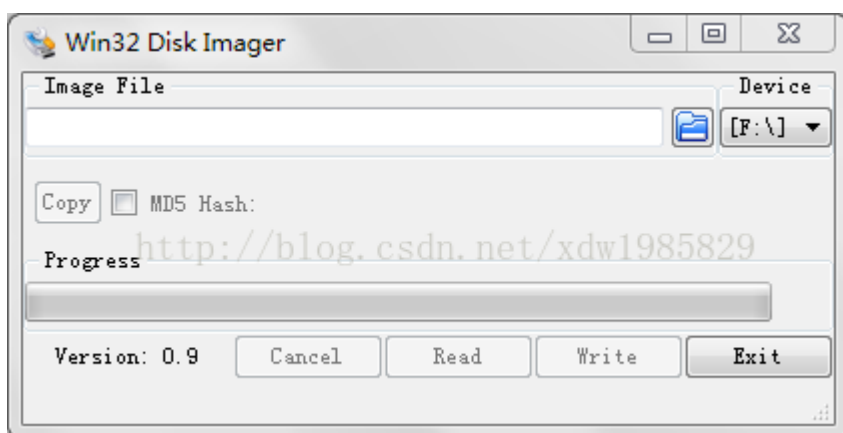
准备工作

1. 一张 2G 以上的 SD 卡及读卡器，最好是高速卡，推荐 Class4 以上的卡，卡的速度直接影响树莓派的运行速度 笔者建议最好 4G 以上，否则后续开发会使用经常不够用
2. WinXP 和 Win7 下安装镜像的工具：Win32DiskImager.zip 同样在上一步的网址下载
3. 上一步下载好的镜像

安装

1. 解压下载的系统压缩文件，得到 img 镜像文件
2. 将 SD 使用卡托或者读卡器后，连上电脑

3. 解压并运行 win32diskimager 工具

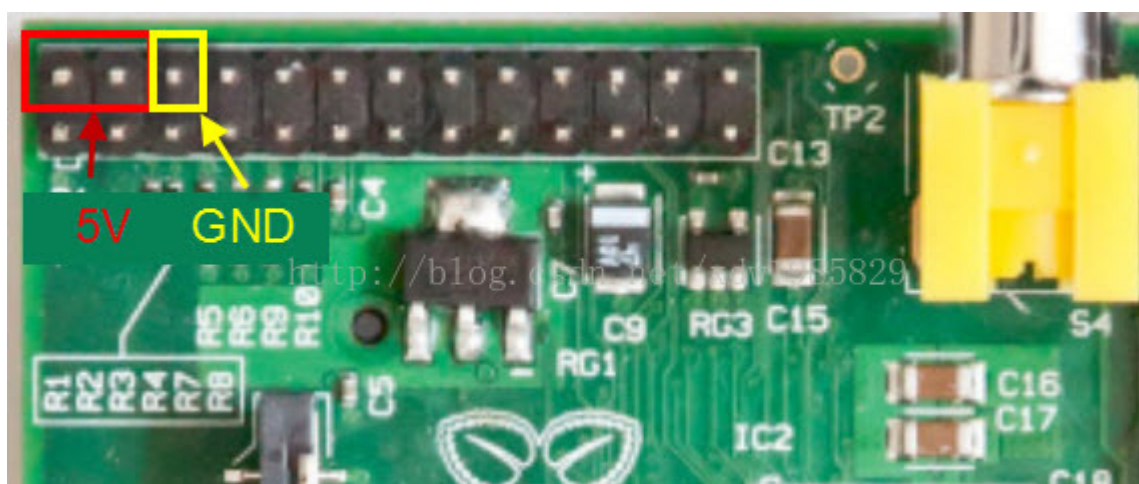


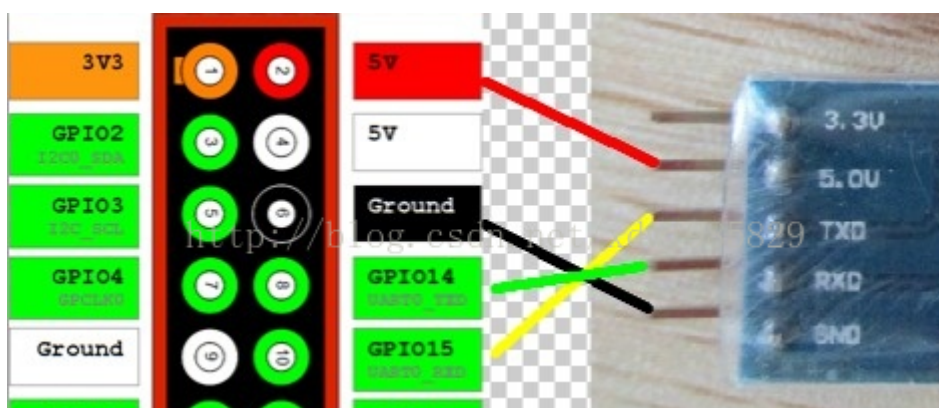
1. 在软件中选择 img 文件，“Device”下选择 SD 的盘符，然后选择“Write”然后就开始安装系统了，根据你的 SD 速度，安装过程有快有慢。
2. 安装结束后会弹出完成对话框，说明安装就完成了，如果不成功，请关闭防火墙一类的软件，重新插入 SD 进行安装，请注意安装完，win 系统下看到 SD 只有 74MB 了，这是正常现象，因为 linux 下的分区 win 下是看不到的！

开发板供电

树莓派开发板有两种供电方式：

- 使用 micro usb 供电：所有 android 手机的充电线都可以给他供电
- 外接电源直接供电：B 型（B+型）GPIO 左上角的 2、4 管角为电源正极，6 管角为地





树莓派访问方式

初始启动

外接 HDMI 或者 AV 显示器，USB 鼠标、键盘

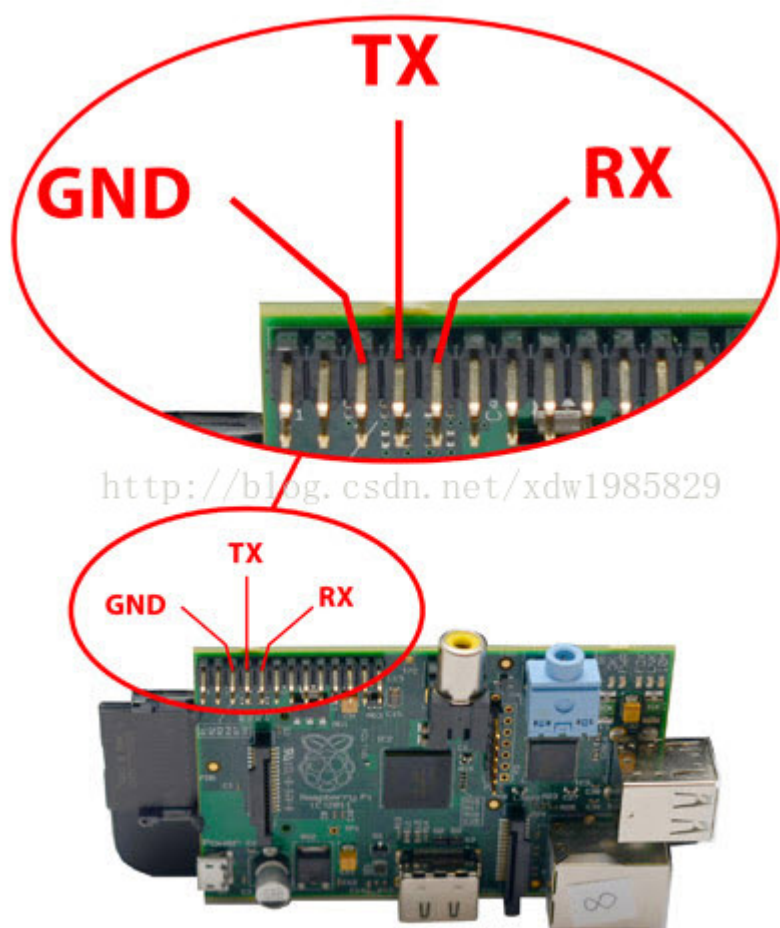
把树莓派当作一个小电脑，可以用外接键盘、鼠标操作树莓派，树莓派启动界面会显示在 HDMI 或者 AV 显示器上。笔者不太建议这种启动方式——抱着个显示、鼠标键盘太不方便了，而且带 HDMI 的显示器不好找。（虽然可以用 HDMI 转 VGA 接在电脑显示器上代替）



使用 USB 转 ttl 串口线通过串口访问树莓派

树莓派默认支持串口启动，我们只需要一根：usb 转 ttl 线（淘宝上很多，10 几块钱一根）连接树莓派与电脑，即可通过串口控制调试树莓派（电脑上使用超级终端或者其它串口调试工具，笔者建议使用 Xshell4.0）

树莓派串口连接口请见下图所示



在终端中会显示类似启动信息，如下：

中间会提示输出用户名和密码：

默认用户名：pi

默认密码：raspberrypi

```
octopi login: pi
Password:
Last login: Fri Jun 20 09:32:38 UTC 2014 on ttyAMA0
Linux octopi 3.12.22+ #691 PREEMPT Wed Jun 18 18:29:58 BST 2014 armv6l
```

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the

```
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.
```

```
NOTICE: the software on this Raspberry Pi has not been fully configured. Please run 'sudo raspi-config'  
pi@octopi:~$
```

其它访问方式

- SSH: 由于树莓派只有一个串口, 当我们开发中需要占用这个串口或者我们希望进行远程开发 (通过一个根网线或无线连接到树莓派) 可以借助 Linux 系统特有的一种服务 SSH 来完成 (请见后面的文章)。
- VNC: VNC 是 linux 下面常用的远程桌面, 用它可以在 Windows 或者 unix 主机上方便的通过网络操作远程主机而不需要一个额外的显示器, 非常实用 (请见后面的文章)。
- 远程桌面: 远程桌面是 Windows 自带的一个远程桌面登陆系统, 很好用 (请见后面的文章)。

以上三种方式都是通过网络对树莓派进行控制, 可以说只要知道了树莓派的 ip 地址, 一切控制 so easy。



树莓派配置



经过前面两步我们的树莓派已经正常的工作起来了，但是在真正用它开发之前还需要进行一系列的配置以及软件的安装，这样开发起来才会得心应手。

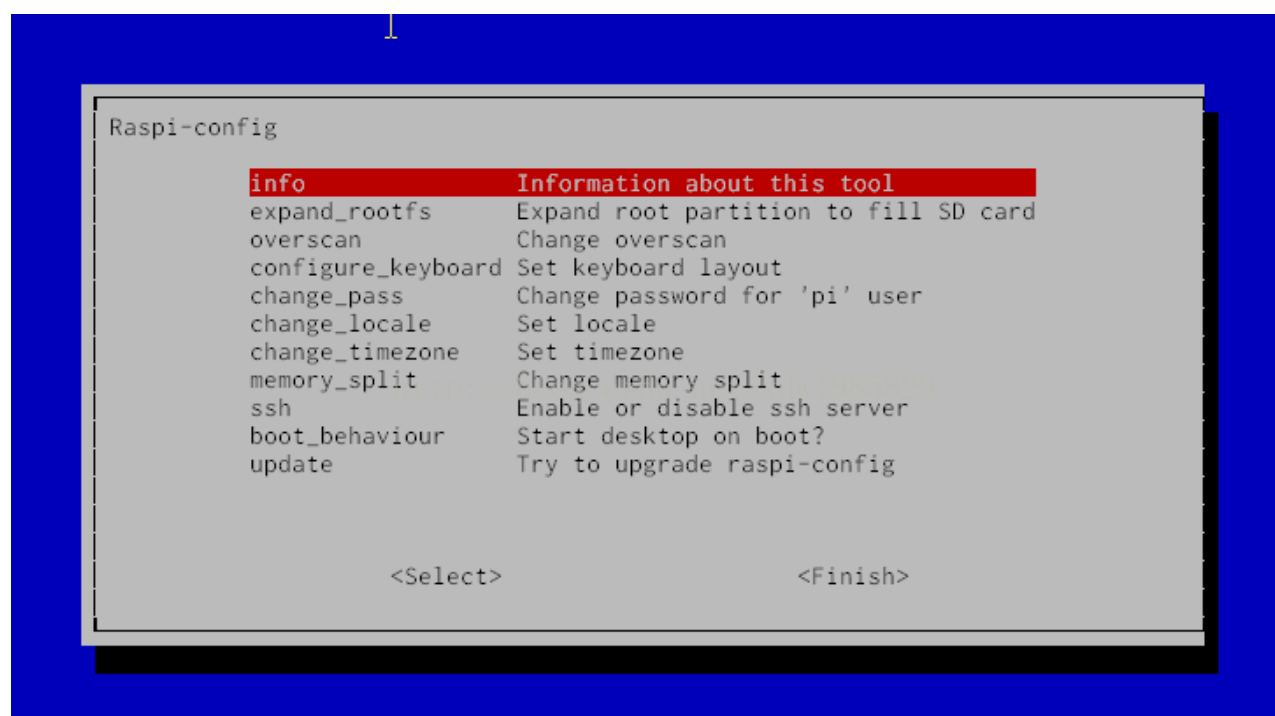
配置选项

树莓派第一次使用的时候需要进行一个简单的配置，在命令行模式下运行以下命令：

```
$sudo raspi-config
```

新旧版本的配置界面不太一样，下面列举两种比较常见的：

旧版本



- `expand_rootfs` - 将根分区扩展到整张 SD 卡（树莓派默认不使用 SD 卡的全部空间，有一部分保留，建议选中）
- `overscan` - 可以扩充或缩小屏幕（旧版不能自适应屏幕，新版没有这个选项，貌似可以自适应，没仔细研究）
- `configure_keyboard` - 键盘配置界面
- `change_pass` - 默认的用户名是 `pi`，密码是 `raspberrypi`，用 `ssh` 远程连接或串口登录时要用到这个用户名和密码，这里可以更改密码。

- `change_locale` - 更改语言设置。在 Locales to be generated: 中, 选择 `en_US.UTF-8` 和 `zh_CN.UTF-8`。在 Default locale for the `-system environment`: 中, 选择 `en_US.UTF-8` (等启动完机器, 装完中文字体, 再改回 `zh_CN.UTF-8`, 否则第一次启动会出现方块)。
- `change_timezone` - 因为树莓派没有内部时钟, 是通过网络获取的时间, 选择 Asia - Shanghai。
- `memory_split` - 配置给桌面显示的显存。
- `ssh` - 是否激活 `sshd` 服务。
- `boot_behaviour` - 设置启动时启动图形界面, 正常肯定是 Yes。

新版本 (比较新的镜像大部分是这个界面, 做了不少改变)

```

----- | Raspberry Pi Software Configuration Tool (raspi-config) | -----
Setup Options

1 Expand Filesystem           Ensures that all of the SD card s
2 Change User Password        Change password for the default u
3 Enable Boot to Desktop/Scratch Choose whether to boot into a des
4 Internationalisation Options Set up language and regional sett
5 Enable Camera               Enable this Pi to work with the R
6 Add to Rastrack             Add this Pi to the online Raspber
7 Overclock                   Configure overclocking for your P
8 Advanced Options            Configure advanced settings
9 About raspi-config          Information about this configurat

                                <Select>                                <Finish>
-----

```

1. Expand Filesystem 扩展文件系统 (同旧版)。

2. Change User Password 改变默认 pi 用户的密码, 按回车后输入 pi 用户的新密码。

3. Enable Boot to Desktop/Scratch 启动时进入的环境选择。

Console Text console, requiring login(default)

启动时进入字符控制台, 需要进行登录 (默认项)。

Desktop log in as user 'pi' at the graphical desktop

启动时进入 LXDE 图形界面的桌面。

Scratch Start the Scratch programming environment upon boot

启动时进入 Scratch 编程环境。

4. Internationalisation Options 国际化选项, 可以更改默认语言

Change Locale: 语言和区域设置, 建议不要改, 默认英文就好。想改中文, 最好选安装了中文字体再进行这一步, 安装中文字体的方法:


```
sudo apt-get update
sudo apt-get install ttf-wqy-zenhei ttf-wqy-microhei
```

移动到屏幕底部，用空格键选中 zh-CN GB2312,zh-CN GB18030,zh-CN UTF-8,然后按回车，然后默认语言选中 zh-cn 然后回车。

Change Timezone: 设置时区，如果不进行设置，PI 的时间就显示不正常。选择 Asia（亚洲）再选择 Chong qing（重庆）即可。

Change Keyboard Layout: 改变键盘布局

5.Enable Camera

启动 PI 的摄像头模块，如果想启用，选择 Enable，禁用选择 Disable就行了。

6.Add to Rastrack

把你的 PI 的地理位置添加到一个全世界开启此选项的地图，建议还是不要开了，免得被跟踪。

7.Overclock

SN	描述
None	不超频，运行在 700Mhz，核心频率 250Mhz，内存频率 400Mhz，不增加电压
Modest	适度超频，运行在 800Mhz，核心频率 250Mhz，内存频率 400Mhz，不增加电压
Medium	中度超频，运行在 900Mhz，核心频率 250Mhz，内存频率 450Mhz，增加电压 2
High	高度超频，运行在 950Mhz，核心频率 250Mhz，内存频率 450Mhz，增加电压 6
Turbo	终极超频，运行在 1000Mhz，核心频率 500Mhz，内存频率 600Mhz，增加电压 6

8.Advanced Options 高级设置

SN	标准	描述
A1	Overscan	是否让屏幕内容全屏显示
A2	Hostname	在网上邻居或者路由器能看到的主机名称
A3	Memory Split	内存分配，选择给 GPU 多少内存
A4	SSH	是否运行 SSH 登录，建议开户此选项，以后操作 PI 方便，有网络就行，不用开屏幕了
A5	SPI	是否默认启动 SPI 内核驱动，新手就不用管了
A6	Audio	选择声音默认输出到模拟口还是 HDMI 口
A7	Update	把 raspi-config 这个工具自动升级到最新版本

- A6
- 0 Auto 自动选择
- 1 Force 3.5mm ('headphone') jack强制输出到3.5mm模拟口
- 2 Force HDMI 强制输出到HDMI

9.About raspi-config

关于 raspi-config 的信息。



4

树莓派网络与更新配置



有线网络：

自动获取 IP：树莓派默认有线网卡是使能的，只需将网线插入树莓派网卡，即可自动获得 IP(要求在局域网内)。

手动设定 IP：如果是电脑与树莓派直连，不能自动获得IP，可以使用：ifconfig eth0 192.168.1.123 设定 ip(下次重启就没了)。

设置静态 IP：如果担心在同网络情况下 ip 或者不固定，可以讲电脑设置为静态 ip，方法如下：

在终端中打开一下文件：

```
sudo vi /etc/network/interfaces
```

(需要学习 VI 基本操作，树莓派自带的 vi 不是很好用，请先参考下面文档更新一下 vi)

```
auto lo
iface lo inet loopback
iface eth0 inet dhcp
```

将以上内容改为：

```
auto lo
iface lo inet loopback

iface eth0 inet static
address 192.168.1.1
netmask 255.255.255.0
gateway 192.168.1.1
```

无线网络

有线网络需要受到网线的限制，现在无线网络已经非常的普及，通过一个无线网卡与无线路由器链接起来岂不是更加无拘无束，下面介绍一下无线网络的配置。

无线网卡驱动的确认真

树莓派内置了很多无线网卡的驱动，大家可以在这个网站查找所支持的型号 http://elinux.org/RPi_VerifiedPeripherals#USB_Wi-Fi_Adapters 笔者手里面有一个：8188CUS(网卡芯片)的验证支持。

验证方法：

将 USB 无线网卡插入树莓派 USB 接口（旧版系统会自动重启，新版不会），敲入：

```
$lsusb:
lsusb
```

提示

```
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 004: ID 0bda:8176 Realtek Semiconductor Corp. RTL8188CUS 802.11n WLAN Adapter
```

或者敲入：

```
$ifconfig:
```

提示

```
wlan0 Link encap:Ethernet HWaddr 00:0b:81:87:e5:f9
      UP BROADCAST MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:3 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

说明树莓派支持你的无线网卡，可进行下一步设置。

无线网卡的配置

使用 vi 打开以下文件进行修改：

```
auto lo
iface lo inet loopback
iface eth0 inet dhcp
auto wlan0
allow-hotplug wlan0
iface wlan0 inet dhcp
wpa-ssid "jikeWiFi"
wpa-psk "helloworld"
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
```

重启即可连入：网络名为 jikeWiFi，密码为 helloworld 的网络，如果想要设置为静态 ip，类似有线网卡，修改相应代码即可。

更新源测试

有线或者无线网络连通过，我们后面对软件更新是，需要首先进行更新列表更新，执行以下命令即可：

```
sudo apt-get update
```

显示示例：

```
Get:1 http://raspberrypi.collabora.com wheezy Release.gpg [836 B]
Get:2 http://archive.raspberrypi.org wheezy Release.gpg [490 B]
Get:3 http://raspberrypi.collabora.com wheezy Release [7,532 B]
Get:4 http://mirrordirector.raspbian.org wheezy Release.gpg [490 B]
Get:5 http://mirrordirector.raspbian.org wheezy Release [14.4 kB]
Get:6 http://raspberrypi.collabora.com wheezy/rpi armhf Packages [2,214 B]
Get:7 http://archive.raspberrypi.org wheezy Release [7,263 B]
....
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en
Fetched 6,992 kB in 1min 12s (96.6 kB/s)
Reading package lists... Done
```

改变更新源

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

将默认的用 # 号屏蔽，改为

```
deb http://mirrors.ustc.edu.cn/raspbian/raspbian/ wheezy main contrib non-free rpi
```

或者

```
deb http://mirror.nus.edu.sg/raspbian/raspbian wheezy main contrib non-free rpi
```

或者用以下地址代替上面的地址栏

中山大学 [Raspbian http://mirror.sysu.edu.cn/raspbian/raspbian/](http://mirror.sysu.edu.cn/raspbian/raspbian/)

中国科学技术大学 [Raspbian http://mirrors.ustc.edu.cn/raspbian/raspbian/](http://mirrors.ustc.edu.cn/raspbian/raspbian/)

清华大学 [Raspbian http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/](http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/)

华中科技大学 [Raspbian http://mirrors.hustunique.com/raspbian/raspbian/](http://mirrors.hustunique.com/raspbian/raspbian/)

Arch Linux ARM <http://mirrors.hustunique.com/archlinuxarm/>

大连东软信息学院源（北方用户）

Raspbian <http://mirrors.neusoft.edu.cn/raspbian/raspbian/>



5

树莓派常用软件及服务



更新 vim

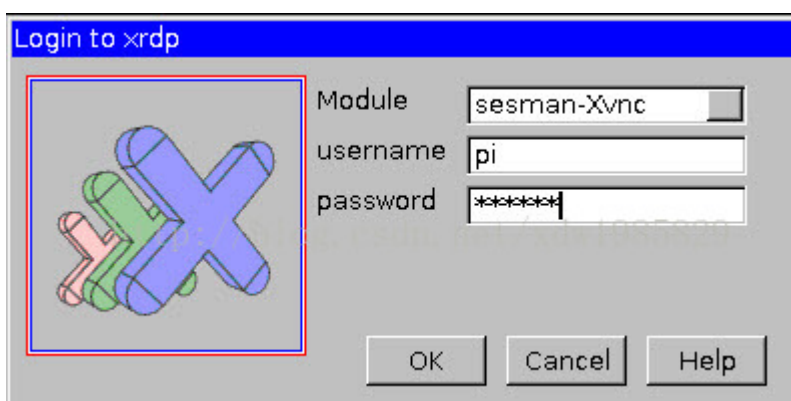
树莓派自带的 vim 不是很好用，比如默认不支持小键盘及方向键等，我们可以更新一下

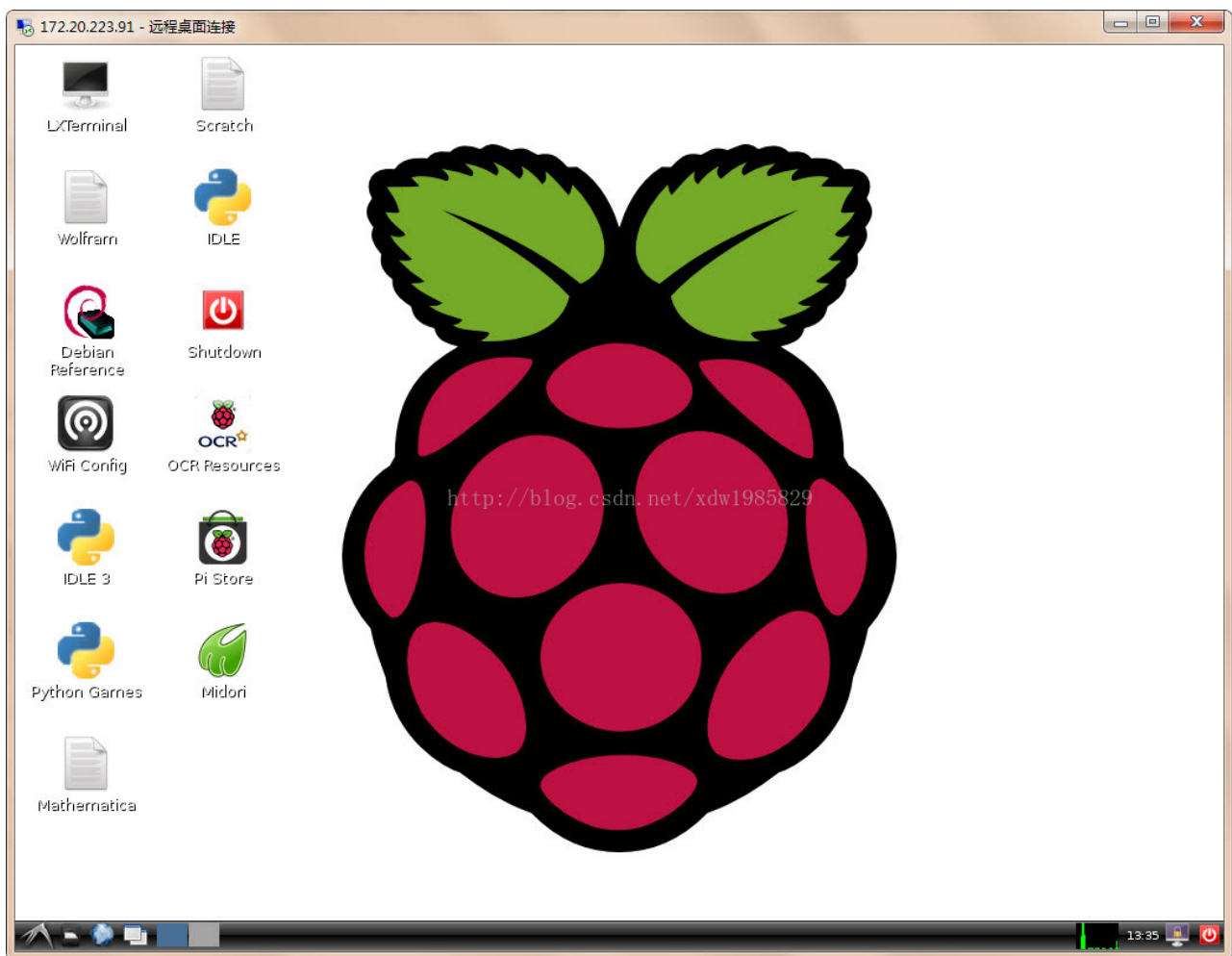
```
sudo apt-get install vim
```

使用远程桌面

通过 Windows 自带的远程桌面软件，可以远程访问树莓派桌面应用

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



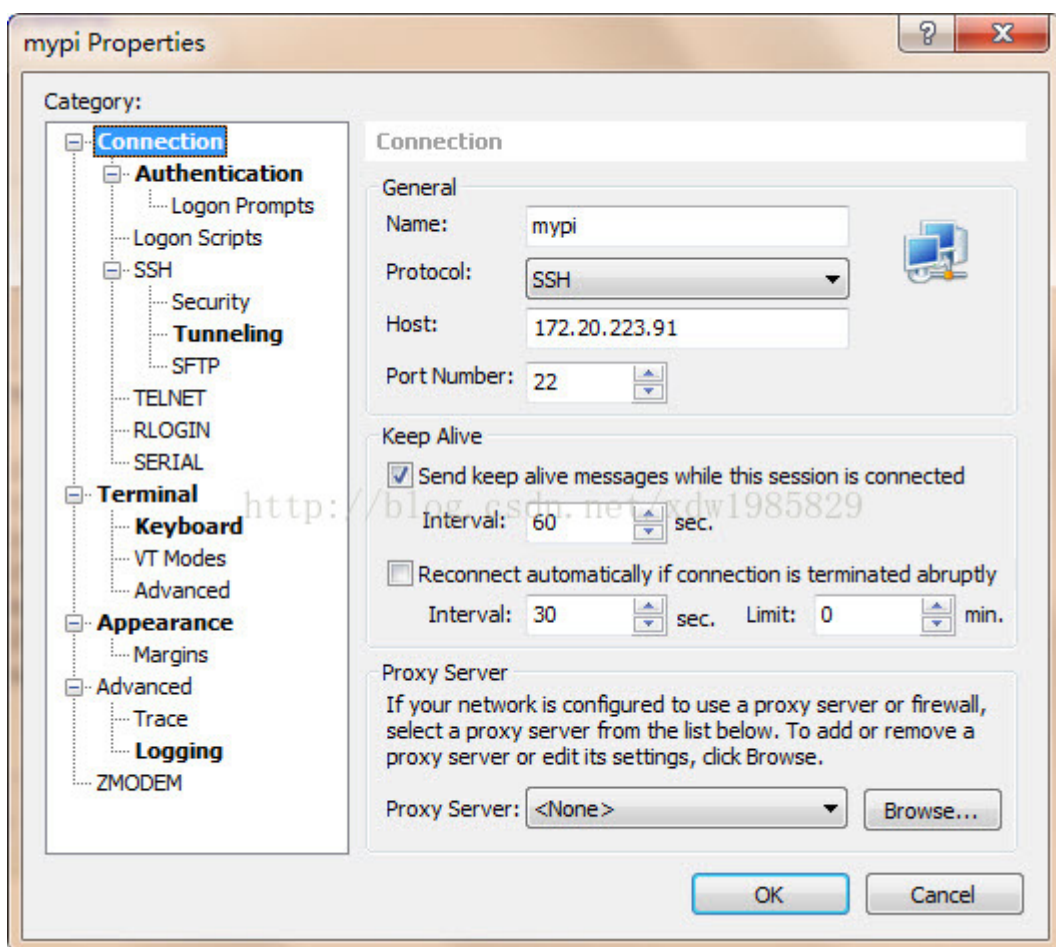


ssh

ssh 服务默认已经安装完毕，只需要安装一个 ssh 客户端即可，笔者推荐 xshell4.0(Xmanage4.0 的一个组件)。

host 填入:开发板 ip 即可。

用户名、密码同串口登陆一致（默认：pi、raspberrypi）



使用 samba

通过 samba 服务，可以自由、访问修改树莓派家目录下的文件、代码。

首先安装 samba `sudo apt-get install samba samba-common-bin`

然后修改配置文件 `sudo vi /etc/samba/smb.conf`

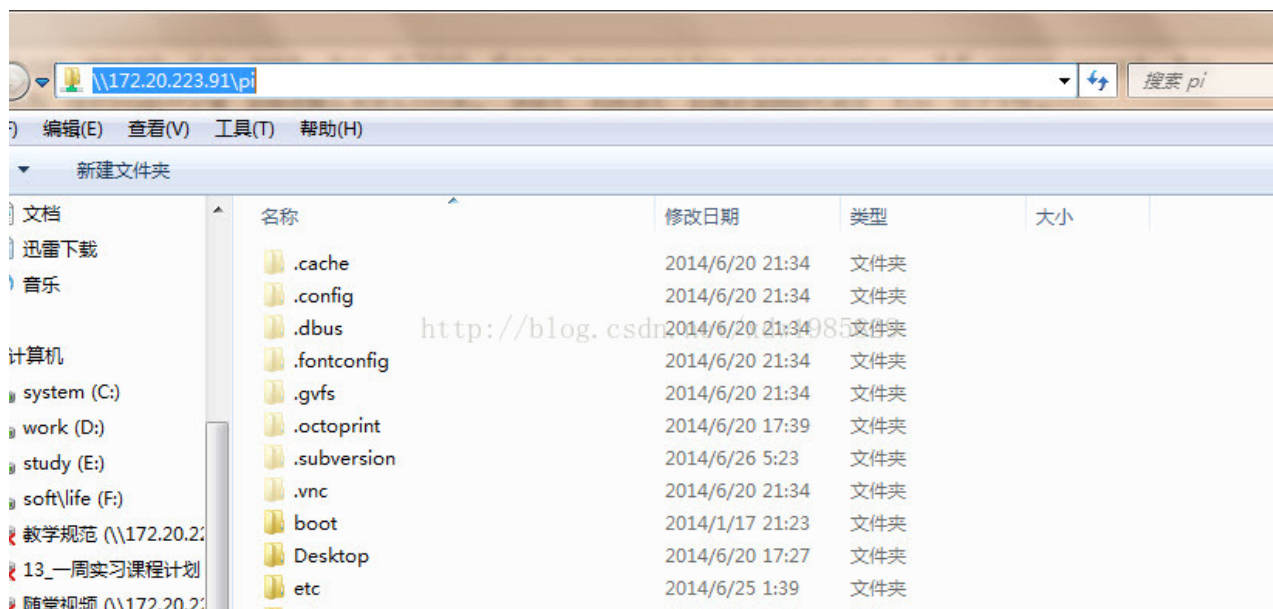
下面的配置是让用户可以访问自己的 home 目录。

- 开启用户认证，找到 `##### Authentication #####`，将 `# security = user` 的 `#` 号去掉。
- 配置用户可以读写自己的 home 目录，在 `[homes]` 节中，把 `read only = yes` 改为 `read only = no`。
- 找到 `browseable=no` 改为 `yes`，否则等会访问时此文件夹会隐藏。

重启 samba 服务 `sudo /etc/init.d/samba restart`

把系统默认用户 pi 添加到 samba `sudo smbpasswd -a pi` 会提示你输入密码：建议跟串口登陆的密码一致，防止密码太多弄混了。

windows 访问 任意打开一个文件浏览器，输入 ip 地址: \\172.20.223.91\pi 现在就可以从其它机器上访问 pi 这个用户的 home 目录了。



挂载 U 盘

当需要大容量外接存储设备时，可使用U盘挂载 通常我们在 /mnt 或 /media 目录下新建一个目录作为挂载点。比如：

```
sudo mkdir /mnt/udisk
```

手动挂挂载

挂载命令

```
sudo mount -o uid=pi,gid=pi /dev/sda1 /mnt/udisk
```

用完之后卸载

```
sudo umount /mnt/1GB_USB_flash
```

注意： sda1 是取决于你的实际情况，a 表示第一个硬盘，1 表示第一个分区。FAT 格式 U 盘 mount 本身就能支持，但如果你的 U 盘或移动硬盘使用的是 exFAT 格式，mount 会说不支持。没关系，安装 exfat-fuse 软件之后 mount 就支持了。 `sudo apt-get install exfat-fuse`

开机挂载

如果想开机自动挂载，而不是每次手工执行，可以编辑 `/etc/fstab` 文件。在末尾添加一行：

```
/dev/sda1 /mnt/udisk vfat rw,defaults 0 0
```

每次开机就会自动挂载。

热插挂载

需要希望想电脑一样，插上自动识别挂载在某一目录下，拔下自动 `umount`，请按一下操作 `sudo vi /etc/udev/rules.d/10-usbstorage.rules`（此文件默认没有，需要新建）赋值以下内容即可，会自动在 `/mnt/udisk` 目录下挂载 U 盘。

```
KERNEL!="sd*", GOTO="media_by_label_auto_mount_end"
SUBSYSTEM!="block",GOTO="media_by_label_auto_mount_end"
IMPORT{program}="/sbin/blkid -o udev -p %N"
ENV{ID_FS_TYPE}=="", GOTO="media_by_label_auto_mount_end"
ENV{ID_FS_LABEL}!="", ENV{dir_name}="%E{ID_FS_LABEL}"
ENV{ID_FS_LABEL}=="", ENV{dir_name}="Untitled-%k"
ACTION=="add", ENV{mount_options}="relatime,sync"
ACTION=="add", ENV{ID_FS_TYPE}=="vfat", ENV{mount_options}="iocharset=utf8,umaskk=000"
ACTION=="add", ENV{ID_FS_TYPE}=="ntfs", ENV{mount_options}="iocharset=utf8,umaskk=000"
ACTION=="add", RUN+="/bin/mkdir -p /mnt/udisk/", RUN+="/bin/mount -o $env{mount_options} /dev/%k /mnt/udisk/"

ACTION=="remove", ENV{dir_name}!="", RUN+="/bin/umount -l /mnt/udisk/}", RUN+="//
bin/rmdir /mnt/udisk/"
LABEL="media_by_label_auto_mount_end"
```



T



6

树莓派做 wifi 热点



原理：Pi 使用有线连入网络，然后接 USB 无线网卡作为热点，提供 Wifi 接入。

USB 无线网卡驱动

如果接上 USB 无线网卡，使用 `ifconfig` 命令，能直接看到 `wlan0`，那么恭喜你，可以直接跳过这一步。如果没有请查询一下树莓派支持的 USB 无线网卡型号，可参考以下网址：http://elinux.org/RPi_VerifiedPeripherals#USB_Wi-Fi_Adapters

修改 wlan0 为静态 IP

相当于设置路由器 wlan 口 IP,即我们访问路由器通常使用的: 192.168.1.1

```
sudo vim /etc/network/interfaces
```

把原来关于 wlan0 的注释掉：（可能跟这个不一样，跟 wlan0 有关的注释掉即可）。

```
#auto wlan0
#iface wlan0 inet dhcp
#wpa-ssid "360WiFi-II"
#wpa-psk "xiaolizi"
```

添加下面的

```
iface wlan0 inet static
address 192.168.0.1
netmask 255.255.255.0
gateway 192.168.0.1
```

完成之后需要重启

安装 hostapd

官方的 `hostapd` 不支持 8188CUS，后面需要重新卸载安装新的，测试时这里必须先装旧的，然后后面卸了装新的，否则也不能用。

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

编辑 hostapd 默认配置文件

```
sudo vim /etc/default/hostapd
```

找到 #DAEMON_CONF= "", 修改为:

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

然后编辑 `sudo vim /etc/hostapd/hostapd.conf`

增加以下代码:

```
[ruby] view plaincopy
# Basic configuration

interface=wlan0
ssid=raspberrypi
channel=1
#bridge=br0

# WPA and WPA2 configuration

macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=3
wpa_passphrase=12345678
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP

# Hardware configuration

driver=rtl871xdrv
ieee80211n=1
hw_mode=g
device_name=RTL8192CU
manufacturer=Realtek
```

修改WiFi 名和密码

```
ssid=raspberrypi
wpa_passphrase=12345678
```


保存退出，然后重启服务

```
sudo service hostapd restart
```

或者执行以下命令生效

```
sudo hostapd -dd /etc/hostapd/hostapd.conf
```

如果你使用的网卡提示一下信息

```
Configuration file: /etc/hostapd/hostapd.conf
nl80211: 'nl80211' generic netlink not found
Failed to initialize driver 'nl80211'
rmdir[ctrl_interface]: No such file or directory
```

那么，还是要使用第三方的 hostapd。

安装新的 hostapd

删除原来的 hostapd

```
sudo apt-get autoremove hostapd
```

下载第三方驱动并安装

```
wget https://github.com/jenssegers/RTL8188-hostapd/archive/v1.1.tar.gz
tar -zxvf v1.1.tar.gz
```

编译

```
cd RTL8188-hostapd-1.1/hostapd
sudo make
sudo make install
```

然后再重启服务，应该提示成功

```
$ sudo service hostapd restart
[ ok ] Stopping advanced IEEE 802.11 management: hostapd.
[ ok ] Starting advanced IEEE 802.11 management: hostapd.
```

将 hostapd 加入开机自启动

```
sudo service hostapd start
sudo update-rc.d hostapd enable
```

如果这里提示的是失败，重启后网络建立成功，用手机可以搜到这个网络

安装 DHCP 服务

以上步骤建立起了 WiFi 热点，但是无法自动获取 ip，需要以下步骤

```
sudo apt-get install udhcpd
```

编辑配置文件

```
sudo vim /etc/udhcpd.conf //修改以下信息，start和end是重点，注意跟第一步的静态ip在一个网段
start 192.168.0.20
end 192.168.0.200
interface wlan0
```

接下来编辑 `/etc/default/udhcpd` 并且将下面这行注释掉，以使DHCP Server正常工作

```
#DHCPD_ENABLED="no"
```

启动 dhcp 服务器

```
sudo service udhcpd start
sudo update-rc.d udhcpd enable
```

经过此步手机已经可以接入 WiFi 网络，并且自动获取 ip。

配置路由转发

理论上是经过这一步，手机可以通过共享树莓派的无线网络上网了，但是可能还存在一些路由转发问题。

设置路由映射规则

```
sudo iptables -F
sudo iptables -X
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
sudo bash -c iptables-save > /etc/iptables.up.rules
```

编辑 `sudo vim /etc/network/if-pre-up.d/iptables`

添加下面两行代码：

```
#!/bin/bash
/sbin/iptables-restore < /etc/iptables.up.rules
```

保存退出，然后修改 iptables 权限：

```
sudo chmod 755 /etc/network/if-pre-up.d/iptables
```

开起内核转发

```
sudo vim /etc/sysctl.conf
```

找到下面两行：

```
#Uncomment the next line to enable packet forwarding for IPv4
#net.ipv4.ip_forward=1
```

把 `net.ipv4.ip_forward` 前面的 `#` 去掉，保存退出。

然后：

```
sudo sysctl -p
```

其它问题

最近经常发现无线网卡配置的 DHCP 不能发挥作用，经过排查发现给无线网卡指定的静态 IP 失败了，也就是说无线网卡没有 IP 导致 DHCP 无法工作，将 `/etc/default/ifplugd` 的内容修改配置如下：

```
INTERFACES="eth0"
HOTPLUG_INTERFACES="eth0"
ARGS="-q -f -u0 -d10 -w -l"
SUSPEND_ACTION="stop"
```



7

树莓派做 web 服务器



一想到 Linux Web 服务器，我们首先想到的是：Apache + MySql + Php。

- Apache 是世界使用排名第一的 Web 服务器软件。可以运行在几乎所有广泛使用的计算机平台上，由于其跨平台和安全性被广泛使用，是最流行的 Web 服务器端软件。
- MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发。是最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的 RDBMS(关系数据库管理系统)应用软件之一。
- PHP（外文名: Hypertext Preprocessor，中文名：“超文本预处理器”）是一种通用开源脚本语言。语法吸收了 C 语言、Java 和 Perl 的特点，易于学习，使用广泛，主要适用于 Web 开发领域。

树莓派可以安装这个 LAMP 系列，但 Apache 和 MySql 对于树莓派这个小小的机器，太重了，主要是消耗内存多 \ 速度慢 \ 占用磁盘大(约 200M)，所以可以选择安装一个轻量级的 Web 服务器：nginx + php + sqlite

- nginx 是个轻量级的 Web 服务器，是一款轻量级的 Web 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器，上 nginx 的并发能力确实在同类型的网页服务器中表现较好。
- SQLite 是一款轻型的数据库，是遵守 ACID 的关系型数据库管理系统，它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百 K 的内存就够了。

Apache + MySql + Php 安装

安装 Apache

Apache 可以用下面的命令来安装 `sudo apt-get install apache2`

Apache 默认路径是 `/var/www/`

其配置文件路径为 `/etc/apache2/`

可以通过 `sudo vi /etc/apache2/ports.conf` 修改监听端口号

重启服务生效 `sudo service apache2 restart`

安装 mysql

`sudo apt-get install mysql-server` 安装过程中，会出现一个提示符让你输入一个密码。这个密码是 mysql root 用户的密码。

安装 PHP

输入下面的命令，就可以安装 PHP 5,以及 PHP 访问 mysql 数据库所需要的库。

```
sudo apt-get install php5  
sudo apt-get install php5-mysql
```

测试

安装完成后，可以在浏览器中输入你路由器的 IP 或域名，就可以访问你的网站了。你应该能看到一个页面显示 “It works”，但是没有其它内容。

创建一个/var/www/index.php

```
<?php  
    print <<< EOT  
<!doctype html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<title>Test successful</title>  
</head>  
<body>  
<h1>Test successful</h1>  
<p>Congratulations.</p>  
<p>Your webserver and PHP are working.</p>  
</body>  
</html>  
EOT;  
  
?>
```

nginx + php + sqlite 安装

安装 nginx web服务器 (约6MB)

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

启动 nginx

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

nginx 的 www 根目录默认在 `/usr/share/nginx/www` 中

修改 nginx 的配置文件

```
sudo vi /etc/nginx/sites-available/default
```

以下几个选项注意一下

`listen 8080;## listen for ipv4; this line is default and implied` – 监听的端口号，如果与其它软件冲突，可以在这里更改。

`root /usr/share/nginx/www;` – nginx 默认路径 html 所在路径。`index index.html index.htm index.php;` – nginx 默认寻找的网页类型，我们可以增加一个 `index.php`。

PHP 脚本支持

找到 php 的定义段，将这些行的注释去掉，修改后内容如下

```
location ~ .php$ {
    fastcgi_pass unix:/var/run/php5-fpm.sock;
    fastcgi_index index.php;
    include fastcgi_params;
}
```

php 段中有一些其它定义，不要去动它，比如

```
# fastcgi_split_path_info ...
# fastcgi_pass 127.0.0.1:9000
```

安装 php 和 sqlite (约 3MB)

```
sudo apt-get install php5-fpm php5-sqlite
```

重新加载 nginx 的配置

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

测试 html

通过主机的 IE 访问树莓派，可以看到主页(表示 Web 服务器已正常启动)

测试 php

在树莓派中生成一个 `php` 文件

```
sudo vi /usr/share/nginx/www/index.php
```


在文件中输入以下内容

```
<?php
    print <<< EOT
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Test successful</title>
</head>
<body>
<h1>Test successful</h1>
<p>Congratulations.</p>
<p>Your webserver and PHP are working.</p>
</body>
</html>
EOT;

?>
```



8

树莓派内核编译与固件升级



前面我们已经使用官方给的镜像搭建好了树莓派的开发环境，可以说已经可以上手应用了，但是如果官方提供的内核有些功能不能满足我们的需要或者需要对内核进行部分裁剪，那么就需要对树莓派的内核进行裁剪，这里我们教一下大家如何更新树莓派内核。

获取升级所需源码

下载地址

官方网址: <https://github.com/raspberrypi>

上面列出了树莓派所有的开源软件

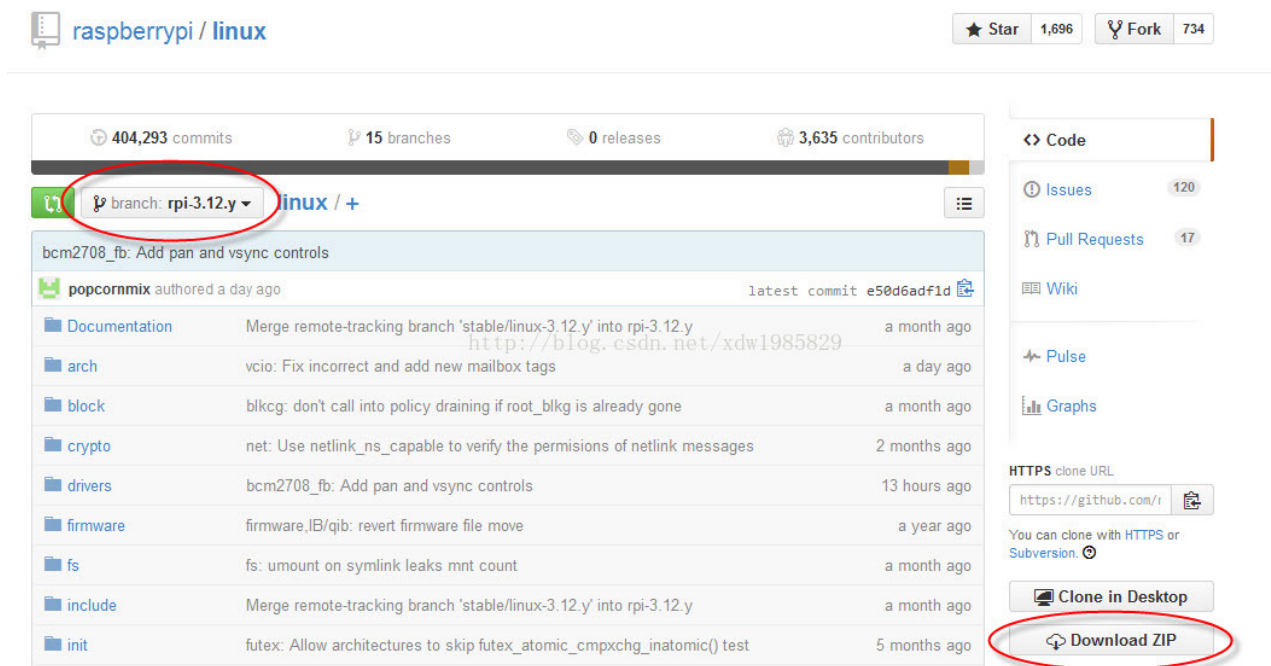
- firmware: 树莓派的交叉编译好的二进制内核、模块、库、bootloader
- linux: 内核源码
- tools: 编译内核和其他源码所需的工具——交叉编译器等

我们只需要以上三个文件即可，下面的工程可以了解一下

- documentation: 树莓派离线帮助文档，教你如何使用、部署树莓派（树莓派官方使用教程）
- userland: arm 端用户空间的一些应用库的源码——vc 视频硬浮点、EGL、mmal、openVG 等
- hats: Hardware Attached on Top，树莓派 B+ 型板子的扩展板资料
- maynard: 一个 gtk 写成的桌面环境
- scratch: 一个简易、可视化编程环境
- noobs: 一个树莓派镜像管理工具，他可以让你在一个树莓派上部署多个镜像
- weston: 一个应用程序
- target_fs: 树莓派最小文件系统，使用 busybox 制作
- quake3: 雷神之锤 3 有线开发源码 firmwareb

下载方法

网页直接下载



到所需要下载的工程，左上角选版本，右方有一个 download ZIP 按钮可直接下载（笔者下载完成后，在 linux 中解压提示出错，Windows 又非常慢切内核建议不要在 Windows 环境解压，所以笔者不建议使用这种方法）。

使用 git 下载

```
$ mkdir raspberriypi_src
$ cd raspberriypi_src
$ git clone git://github.com/raspberriypi/firmware.git
$ git clone git://github.com/raspberriypi/linux.git
$ git clone git://github.com/raspberriypi/tools.git
```

会得到三个文件夹：firmware linux tools

编译、提取内核及其模块

获得内核配置文件

在运行的树莓派中运行

```
$ls /proc/
```

可看到一个叫 config.gz 的文件，他是当前的树莓派配置选项记录文件，我们将他拷出，放入我们的内核源码目录树下。

```
$cp /proc/config /home/pi
```

我们这里使用前面交过的 samba 拷出并拷入内核源码目录下，不熟悉的人可参考前面文章。

在 linux 内核源码下执行

```
$zcat config.gz > .config
```

```
.          COPYING          .gitignore  lib          modules.order
..         CREDITS          include    .mailmap     Module.symvers
arch       crypto           init       MAINTAINERS  net
block     Documentation    ipc       Makefile     README
.config   drivers          Kbuild    .missing-syscalls.d REPORTING-BUGS
config.gz firmware          Kconfig   mm           samples
.config.old fs                kernel    modules.builtin scripts
```

配置、编译内核

修改内核源码 makefile ARCH 类型和编译器路径 `$vi Makefile +195`，找到以上类似代码，改为如图所示。

```
# Note: Some architectures assign CROSS_COMPILE in their arch/*/Makefile
ARCH ?= arm
CROSS_COMPILE ?= ../tools/arm-bcm2708/arm-bcm2708hardfp-linux-gnueabi/bin/arm-bcm2708hardfp-linux-gnueabi-
```

查看、修改配置选项 `$make menuconfig`，可出现以下界面。

```

Linux/arm 3.12.26 Kernel Configuration
the menu. <Enter> selects submenus ---> (or empty submenus ----). Hi
> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc>
[*] built-in [ ] excluded <M> module < > module capable

[ ] Patch physical to virtual translations at runtime
  General setup --->
  [*] Enable loadable module support --->
  [*] Enable the block layer --->
    System Type --->
    Bus support --->
    Kernel Features --->
    Boot options --->
    CPU Power Management --->
    Floating point emulation --->
    Userspace binary formats --->
    Power management options --->
  [*] Networking support --->
    Device Drivers --->
    File systems --->
    Kernel hacking --->
    Security options --->
  -* Cryptographic API --->
    Library routines --->
  [ ] Virtualization ----

<Select> < Exit > < Help > < Save > < Load >

```

如果不做修改，直接选中 exit 即可（注意使用键盘操作）。

编译内核镜像 \$make 在 arch/arm/boot 目录下可以看到一个叫 zImage 的文件，就是我们新的内核，但是树莓派需要另外一种格式的镜像，需要进行处理一下，执行以下命令

```

$cd tools/mkimage

$./imagetool-uncompressed.py ../../linux/arch/arm/boot/zImage

```

即可在当前文件夹下看到一个叫 kernel.img 的文件，就是我们需要的内核了。

提取 modules 上一步其实不但编译出来了内核的源码，一些模块文件也编译出来了，这里我们提取一下，

```

$cd raspberrypi_src
$mkdir modules
$cd linux
$ make modules_install INSTALL_MOD_PATH=../modules

```

即可在 modules 得到我们需要的模块文件。

升级 RPi 的 kernel、Firmware、lib

将SD卡拔下插在电脑上（可使用读卡器）

1. 升级内核，将新编好的内核拷入 SD 卡，改名为 `kernel_new.img`。打开 boot 目录下找到 `config.txt` 文件，加入 `: kernel=kernel_new.img` 这一行。
2. 升级 boot，将 `firmware/boot/` 目录下文件拷入 SD 卡 boot 目录：`fbootcode.bin` `fixup.dat` `fixup_cd.dat` `start.elf`
3. 更新 vc 库及内核 modules，将编译出来的 `modules/lib/modules` 拷入树莓派文件系统 `/lib` 下



9

树莓派 GPIO 控制



常用开源工程简介

树莓派内核中已经编译自带了 gpio 的驱动，我们常通过一些第三方写好的库函数来完成具体的操作，比较常见的操作库函数有：

python GPIO

- 开发语言——python
- 简单介绍——树莓派官方资料中推荐且容易上手。python GPIO 是一个小型的 python 库，可以帮助用户完成 raspberry 相关 IO 口操作，但是 python GPIO 库还没有支持 SPI、I2C 或者 1-wire 等总线接口。
- 官方网站——<https://code.google.com/p/raspberry-gpio-python/>

wiringPi

- 开发语言——C 语言
- 简单介绍——wiringPi 适合那些具有 C 语言基础，在接触树莓派之前已经接触过单片机或者嵌入式开发的人群。wiringPi 的 API 函数和 arduino 非常相似，这也使得它广受欢迎。作者给出了大量的说明和示例代码，这些示例代码也包括 UART 设备，I2C 设备和 SPI 设备等。
- 官方网站——<http://wiringpi.com/>

BCM2835 C Library

- 开发语言——C 语言
- 简单介绍——BCM2835 C Library 可以理解为使用C语言实现的相关底层驱动，BCM2835 C Library 的驱动库包括 GPIO、SPI 和 UART 等，可以通过学习 BCM2835 C Library 熟悉 BCM2835 相关的寄存器操作。如果有机会开发树莓派上的 linux 驱动，或自主开发 python 或 PHP 扩展驱动，可以从 BCM2835 C Library 找到不少的“灵感”。
- 官方网站——<http://www.airspayce.com/mikem/bcm2835/>

树莓派GPIO编号方式

1. 功能物理引脚：从左到右，从上到下：左边基数，右边偶数：1-40
2. BCM：编号侧重 CPU 寄存器，根据 BCM2835 的 GPIO 寄存器编号。
3. wpi：编号侧重实现逻辑，把扩展 GPIO 端口从 0 开始编号，这种编号方便编程。正如下图 WiringPi 一栏。

BCM 编码方式	wpi 编码方式	功能名	物理接口				功能名	wpi 编码方式	BCM 编码方式		
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
		3.3v			1	2		5v			
2	8	SDA.1	ALT0	1	3	4		5V			
3	9	SCL.1	ALT0	1	5	6		0v			
4	7	GPIO. 7	IN	1	7	8	0	ALT0	TxD	15	14
		0v			9	10	1	ALT0	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	0	IN	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	1	21	22	1	OUT	GPIO. 6	6	25
11	14	SCLK	ALT0	1	23	24	1	ALT0	CE0	10	8
		0v			25	26	1	ALT0	CE1	11	7
0	30	SDA.0	ALT0	1	27	28	1	ALT0	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
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	

python GPIO

先安装 python-dev，输入以下指令。

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

安装 RPi.GPIO，依次输入以下指令。

- 下载：\$ wget http://raspberrypi-gpio-python.googlecode.com/files/RPi.GPIO-0.5.3a.tar.gz
- 解压缩：\$ tar xvfz RPi.GPIO-0.5.3a.tar.gz
- 进入解压之后的目录：\$ cd RPi.GPIO-0.5.3a
- 启动安装：\$ sudo python setup.py install

例子：

```
# -*- coding: utf-8 -*-
import RPi.GPIO as GPIO
import time
# BOARD编号方式，基于插座引脚编号
```

```
GPIO.setmode(GPIO.BOARD)
# 输出模式
GPIO.setup(11, GPIO.OUT)

while True:
    GPIO.output(11, GPIO.HIGH)
    time.sleep(1)
    GPIO.output(11, GPIO.LOW)
    time.sleep(1)
```

执行：

```
sudo python led.py
```

说明：

- GPIO.setmode(GPIO.BOARD)，采用插座引脚编号方式。
- 由于采用插座引脚编号方式，此处的 11 脚相当于 BCM2835 寄存器编号方式的引脚 11。

WiringPi GPIO

说明

WiringPi 是应用于树莓派平台的 GPIO 控制库函数，WiringPi 遵守 GUN Lv3。wiringPi 使用 C 或者 C++ 开发并且可以被其他语言包转，例如 Python、ruby 或者 PHP 等。wiringPi 包括一套 gpio 控制命令，使用 gpio 命令可以控制树莓派 GPIO 管脚。用户可以利用 gpio 命令通过 shell 脚本控制或查询 GPIO 管脚。wiringPi 是可以扩展的，可以利用 wiringPi 的内部模块扩展模拟量输入芯片，可以使用 MCP23x17/MCP23x08（I2C 或者 SPI）扩展 GPIO 接口。另外可通过树莓派上的串口和 Atmega（例如 arduino 等）扩展更多的 GPIO 功能。另外，用户可以自己编写扩展模块并把自定义的扩展模块集成到 wiringPi 中。WiringPi 支持模拟量的读取和设置功能，不过在树莓派上并没有模拟量设备。但是使用 WiringPi 中的软件模块却可以轻松地应用 AD 或 DA 芯片。

wiringPi 安装

- 方案 A——使用 GIT 工具，通过 GIT 获得 wiringPi 的源代码。

```
git clone git://git.drogon.net/wiringPi
cd wiringPi
./build
```

build 脚本会帮助你编译和安装 wiringPi。

- 方案 B——直接下载，我们可以在<https://git.drogon.net/?p=wiringPi;a=summary> 网站上直接下载最新版本编译使用。

```
tar xzf wiringPi-xx.tar.gz
cd wiringPi-xx
./build
```

测试

wiringPi 包括一套 gpio 命令，使用 gpio 命令可以控制树莓派上的各种接口，通过以下指令可以测试wiringPi 是否安装成功。

```
$gpio -v
$gpio readall
```

即可出现上面的 io 图。

例子

[cpp] view plaincopy 在 CODE 上查看代码片派生到我的代码片

```
#include <wiringPi.h>
int main(void)
{
    wiringPiSetup();
    pinMode(0, OUTPUT);
    for(;;)
    {
        digitalWrite(0, HIGH); delay(500);
        digitalWrite(0, LOW); delay(500);
    }
}
```

编译运行

在树莓派上：

```
gcc -Wall -o test test.c -lwiringPi
sudo ./test
```

在虚拟机中：

```
am-linux-gcc -Wall -o test test.c -lwiringPi
sudo ./test
```

注意事项:

- IO 的编号方式略有不同, 采用 wiring 编码方式。
- `-lwiringPi` 表示动态加载 wiringPi 共享库。

BCM2835 C Library

- 下载: `$ wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.35.tar.gz`
- 解压缩: `$tar xvzf bcm2835-1.35.tar.gz`
- 进入压缩之后的目录: `$cd bcm2835-1.35`
- 配置: `$./configure`
- 从源代码生成安装包: `$make`
- 执行检查: `$sudo make check`
- 安装 bcm2835库: `$sudo make install`
- 例子[cpp] view plaincopy 在 CODE 上查看代码片派生到我的代码片

```
#include <bcm2835.h>

// P1插座第11脚
#define PIN RPI_GPIO_P1_11

int main(int argc, char **argv)
{
    if (!bcm2835_init())
        return 1;

    // 输出方式
    bcm2835_gpio_fsel(PIN, BCM2835_GPIO_FSEL_OUTP);

    while (1)
    {
        bcm2835_gpio_write(PIN, HIGH);
        bcm2835_delay(100);

        bcm2835_gpio_write(PIN, LOW);
```

```
bcm2835_delay(100);  
}  
bcm2835_close();  
return 0;  
}
```

注意事项：

- IO 的编号方式略有不同，采用 wiring 编码方式。
- -lwiringPi 表示动态加载 wiringPi 共享库。



10

树莓派 spi 液晶屏支持



树莓派官方支持 av 及 HDMI 输出，板上预留了一个 csi 接口的液晶显示屏，但是一直没有相应的模组出现。在很多应用场合我们需要一些小型的液晶屏显示一些基本的信息，所以小屏驱动很是必要。

在 github 上有一个开源工程：notro/fbtf，完整的实现了 framebuffer 驱动，让树莓派完美支持 tft 液晶，下面对移植过程进行一个简单说明。

官网地址

工程首页：<https://github.com/notro>

fbtf 源码：<https://github.com/notro/fbtf>

编译好的固件（基于 3.12.25+）：<https://github.com/notro/rpi-firmware>

使用说明(wiki)：<https://github.com/notro/fbtf/wiki>

使用编译好的固件(3.12.25+)

环境：树莓派

<https://github.com/notro/rpi-firmware>

打开 SPI

树莓派默认 spi 是关掉的，我们需要打开

```
sudo vi /etc/modprobe.d/raspi-blacklist.conf
```

把下面这句话前面的 # 号删掉

```
blacklist spi-bcm2708
```

下载

- 以模块的形式编译进内核（需要手动或脚本加载模块） 3.12.25+
sudo REPO_URI=https://github.com/notro/rpi-firmware rpi-update
- 直接编译进内核
sudo REPO_URI=https://github.com/notro/rpi-firmware BRANCH=builtin rpi-update

- 以模块的形式编译进内核

```
sudo REPO_URI=https://github.com/notro/rpi-firmware BRANCH=latest rpi-update
```

- 直接下载压缩包，手动安装

```
http://tronnes.org/downloads/2014-06-20-wheezy-raspbian-2014-07-25-fbtf-master-firmware.zip
```

配置

手动加载模块

```
sudo modprobe fbtf_device name=adafruit22
```

name 后面的名字，要跟相应的液晶驱动芯片移植，使用的液晶芯片为：fb_ra8875，所以这里写的是：er_tftm050_2。其它芯片请查阅：https://github.com/notro/fbtf/blob/master/fbtf_device.c 文件。正常会提示以下信息。

```
fbtf_device: SPI devices registered:
fbtf_device:  spidev spi0.0 500kHz 8 bits mode=0x00
fbtf_device:  spidev spi0.1 500kHz 8 bits mode=0x00
fbtf_device: 'fb' Platform devices registered:
fbtf_device:  bcm2708_fb id=-1 pdata? no
fbtf_device: Deleting spi0.0
fbtf_device: GPIOs used by 'adafruit22':
fbtf_device:  'reset' = GPIO25
fbtf_device:  'led' = GPIO23
fbtf_device: SPI devices registered:
fbtf_device:  spidev spi0.1 500kHz 8 bits mode=0x00
fbtf_device:  fb_hx8340bn spi0.0 32000kHz 8 bits mode=0x00
graphics fb1: fb_hx8340bn frame buffer, 176x220, 75 KiB video memory, 16 KiB buffer memory, fps=20, spi0.0 at 32 MHz
```

在/dev/目录下出现: /dev/fb1 设备

自动加载模块

```
sudo vi /etc/modules
```

加入以下语句，既可以在启动时自动加载模块。

```
spi-bcm2708
fbtf_device name=er_tftm050_2 speed=28000000 fps=25 verbose=0
```

使用

手动启动 x11 和控制台到新的液晶屏

X Windows 显示在 fb1 上:

```
$FRAMEBUFFER=/dev/fb1 startx
```

Console 显示在 fb1 上:

```
$con2fbmap 1 1
```

自动登陆 x11

```
sudo vi /etc/inittab
#1:2345:respawn:/sbin/getty --noclear 38400 tty1
1:2345:respawn:/bin/login -f pi tty1 </dev/tty1 >/dev/tty1 2>&1

sudo vi /etc/rc.local
su -l pi -c "env FRAMEBUFFER=/dev/fb1 startx &"
```

测试

将 fb0 上的内容直接拷贝到 fb1 上,fb0 和 fb1 同步

<https://github.com/notro/fbtf/wiki/Framebuffer-use#framebuffer-mirroring>

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

启动: fbcp &

关闭fbcp: killall fbcp

启动时使用 fb1

```
$sudo apt-get install xserver-xorg-video-fbdev

$sudo vi /usr/share/X11/xorg.conf.d/99-fbdev.conf
```

加入以下语句:

```
Section "Device"
    Identifier "myfb"
```

```
Driver "fbdev"
Option "fbdev" "/dev/fb1"
EndSection
```

启动: startx

测试:

```
apt-get -y install fbi
fbi -d /dev/fb1 -T 1 -noverbose -a test.jpg
```

由内核及源码编译

下载、编译内核源码

请见 [《树莓派内核编译与固件升级》\(\)](#)

下载、编译 fbtft 源码

```
$cd linux(进入下载好的内核源码目录)
$cd drivers/video
$git clone https://github.com/notro/fbtft.git
```

修改内核源码的 Kconfig 及 Makefile

```
Add to drivers/video/Kconfig: source "drivers/video/fbtft/Kconfig"
Add to drivers/video/Makefile: obj-y += fbtft/
$make menuconfig(在配置界面加入所选用液晶的驱动支持)
```

```
Device Drivers --->
Graphics support --->
<M> Support for small TFT LCD display modules --->

<M> FB driver for the HX8353D LCD Controller
<M> FB driver for the ILI9320 LCD Controller
<M> FB driver for the ILI9325 LCD Controller
<M> FB driver for the ILI9340 LCD Controller
<M> FB driver for the ILI9341 LCD Controller
< > FB driver for the ILI9481 LCD Controller
<M> FB driver for the ILI9486 LCD Controller
<M> FB driver for the PCD8544 LCD Controller
<M> FB driver for the RA8875 LCD Controller
```

极客学院

jikexueyuan.com

中国最大的IT职业在线教育平台



更多信息请访问 

<http://wiki.jikexueyuan.com/project/raspberry-pi/>