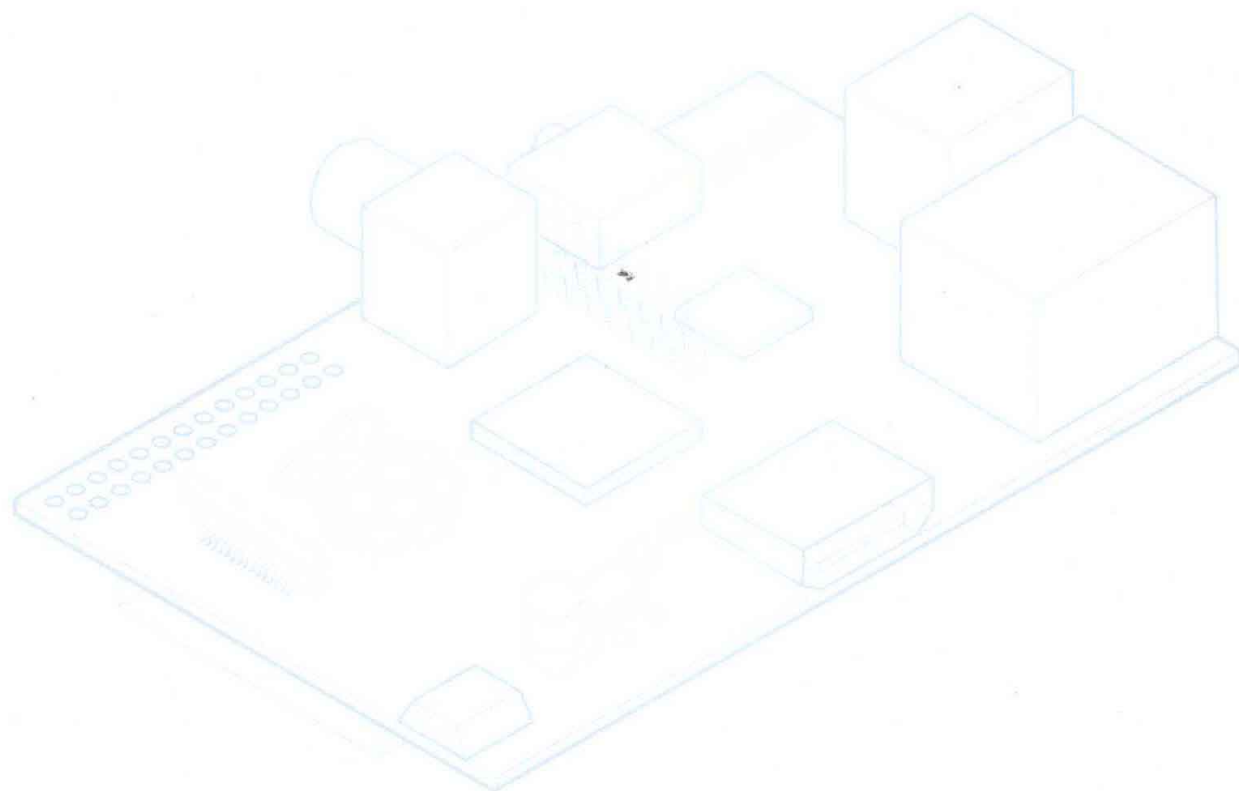
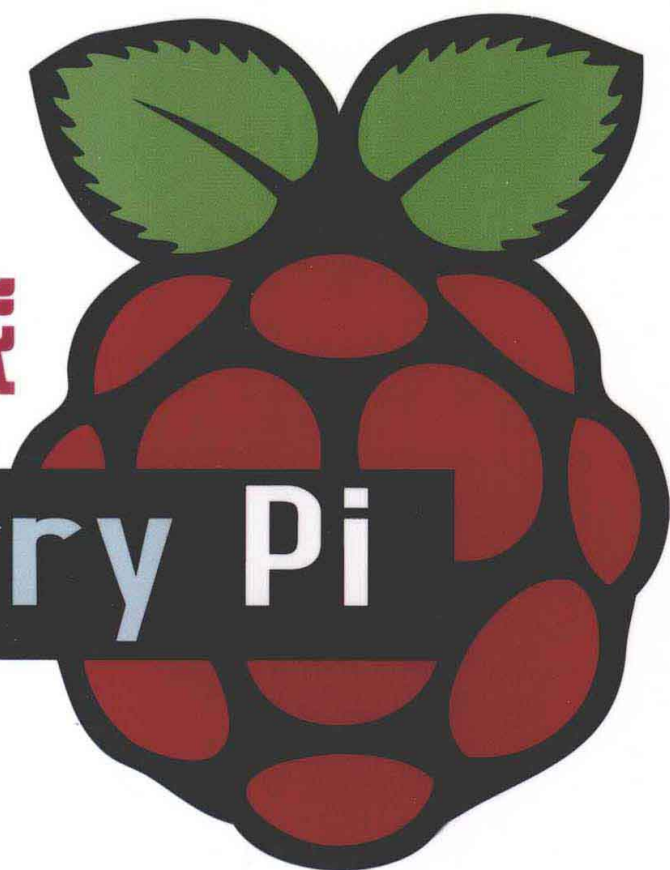




树莓派

Raspberry Pi

王江伟 刘青 编著



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

策划编辑：王静竞

封面设计：runsign 蓝正设计·赫健



树莓派是由树莓派基金会开发的超低价格的计算机，
旨在为发展中国家的学生提供一个相对便宜的计算机环境。
本书以树莓派的应用为主，介绍了在树莓派上快速安装与配置Linux服务器的
方法，以及如何通过硬件开发对树莓派进行扩展以完成更多的任务。
本书适用于希望通过树莓派来学习Linux系统、编程语言并使之实现各种
网络服务器功能的读者，也适用于那些希望用树莓派来学习硬件扩展或是硬
件开发的用户。



爱板网论坛：

易络盟官方指定的树莓派论坛，提供丰富评测、教程及实用项目文章，是大量树莓派爱好者进行技术心得交流的乐园。

上架建议：嵌入式系统

ISBN 978-7-5124-1262-0



定价：29.00元

玩转树莓派 Raspberry Pi

王江伟 刘青 编著

北京航空航天大学出版社

内 容 简 介

树莓派是由树莓派基金会开发的超低价格的计算机,它为发展中国家的学生提供了一个便宜的计算机编程环境。本书以树莓派的应用为主,介绍了使用树莓派完成主流服务器软件的方法,以及通过硬件开发对树莓派进行扩展以完成更多的任务。本书的前三章介绍了如何获得树莓派、如何选择它的周边设备,以及树莓派运行 Linux 所需要的知识;第4章介绍了如何将树莓派作为一个媒体中心与游戏机来使用;第5章是全书较为重要的一章,介绍了通过树莓派所使用的 Linux 系统来完成多个迷你服务器,如:SSH、WEB、论坛、数据库等功能;第6章则介绍了通过树莓派的扩展接口来连接的各种硬件。

本书适用于想通过树莓派来学习 Linux 系统、编程语言以及想通过树莓派来完成各种网络服务器功能的用户,也适用于那些想通过树莓派来学习硬件扩展或是硬件开发的用户。

图书在版编目(CIP)数据

玩转树莓派 Raspberry Pi / 王江伟,刘青编著. --
北京:北京航空航天大学出版社,2013.9
ISBN 978-7-5124-1262-0

I. ①玩… II. ①王… ②刘… III. ①操作系统—程序设计 IV. ①TP316

中国版本图书馆 CIP 数据核字(2013)第 221130 号

版权所有,侵权必究。

玩转树莓派 Raspberry Pi

王江伟 刘 青 编著
责任编辑 张冀青

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@gmail.com 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本:710×1 000 1/16 印张:11.5 字数:245 千字

2013 年 9 月第 1 版 2013 年 9 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-1262-0 定价:29.00 元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

前言

树莓派是一个非常了不起的产品，一个信用卡大小的电路板就涵盖了一台计算机所需要的全部硬件，用户可以以非常低的成本获得一个 Linux 学习环境或是一个包含硬件扩展接口的计算机系统。

但是，在中国介绍树莓派却是一件很麻烦的事情，因为在中国真正的树莓派用户都是一些单片机开发者或 Linux 专业用户。这些用户对树莓派的硬件、软件都非常了解，因此他们可以快速上手，而那些没有单片机或 Linux 使用经验的用户则很少会因为树莓派的某一功能（如硬件播放功能）来购买树莓派，因为在中国他们有更好的选择。对于那些专业或是准专业的用户来说，本书的定位实际上是一本快速手册。这本书主要介绍如何用最少的步骤来完成大部分树莓派所能实现的功能，如：BT 或 PT 下载、无线路由、个人网站建立、硬件扩展连接等。对于那些刚开始使用树莓派的用户来说，本书也可以作为一个不错的基础教程，前两章，用户可以系统地了解树莓派的历史以及学习 Linux 的基础操作，从第 3 章开始学习如何使用树莓派，并最终掌握树莓派系统的运行以及扩展设备的连接。

本书适用人群

书中大量使用了 UNIX/Linux 命令和工具，因此，阅读本书的读者需要对 UNIX/Linux 有一定的了解。如果用户没有 UNIX/Linux 的使用经验，也可以直接从本书的第 2 章开始学习 Linux 的基本操作以及如何操作编辑器等内容。

使用的平台、软件版本

书中所使用的命令如果没有特别说明，均是以 SSH 方式登录到树莓派后所输入的。

本书默认使用 root 用户登录（以“#”提示符显示），出于安全考虑，正式使用时如非必要请使用普通用户操作。

本书使用的操作系统为树莓派官方提供的 Debian 版本。

前言

本书使用的所有命令都在树莓派官方提供的 Debian(2012-12-16-wheezy-raspbian)版本中测试通过。

本书使用的网络配置参数(读者可以根据自己的需要进行设置)如下:

有线网络 IP 地址:172.16.3.4	无线网络 IP 地址:192.168.1.1
子网掩码:255.255.255.128	子网掩码:255.255.255.0
网关:172.16.3.1	网关:192.168.1.1
DNS:202.103.44.150	

本书的结构

第1章 什么是树莓派

介绍了什么是树莓派,以及树莓派各种版本的变迁、硬件接口类型;另外由于树莓派是英国设计并生产的,因此还为国内的用户介绍了如何获得树莓派,以及如何选择它的周边设备。

第2章 Linux 系统基础

树莓派所使用的系统是 Linux,所以本章为那些不太了解 Linux 的用户作了简单的介绍。其中包括 Linux 基础命令、文件系统、用户权限管理、安装/删除软件等。

第3章 使用树莓派

在这一章读者将了解到哪一种 Linux 发行版最适合自己的,并学习如何在各种平台上为树莓派安装系统镜像到 SD 卡上,以及进入树莓派系统中后应该如何设置一些系统参数,如:网络参数、中文环境等。另外,对于 Windows 的用户,还专门介绍了树莓派中的 X 窗口系统,希望能对读者有帮助。

第4章 媒体中心与游戏

由于树莓派内置了 H.264 等格式的高清视频的硬件解码,因此很多用户想将它作为一个家庭媒体中心来播放各种多媒体文件,事实上这是完全可行的。通过 Raspbmc 等专为媒体播放所移植的 Linux 发行版,用户可以轻松地完成这些工作,再加上 BCM2835 还不算弱的性能,使得它可以运行一些如模拟器、《雷神之锤》等游戏程序,让您在工作之余可以放松一下。

第5章 迷你服务器

树莓派最迷人的地方就是它通过一个只有信用卡大小的电路就可以实现一个低功耗电脑的全部功能,加上它使用的是 Linux 操作系统,更使得树莓派可以轻松地搭建一个迷你服务器平台。在这个平台中,无论是下载、WEB、NFS,它都可以轻松地胜任。不过由于 Linux 服务器的复杂性,本章中的例子都只是给出了一个能保证服务正常运行的最小化的配置。

第6章 扩展接口及应用

树莓派虽然是为了给青少年提供一个便宜的可练习编程环境的计算机,但是它通过 GPIO 等硬件接口具备了硬件扩展的能力。在这一章中,将会由浅入深地完成三个小例子:

- ①通过 GPIO 建立一个内在指示系统。
- ②连接 LM75 温度传感器获取当前温度。
- ③使用树莓派连接 GPS。

如果读者没有硬件设计的经验,那么可以从本章开始学习设计一个基础电路。

附录 A 是树莓派的显示参数,如果用户的 HDMI 显示输出不正确,可以在这里查找相应的显示参数进行设置;附录 B 是一些常见的硬件扩展板;附录 C 则介绍了如何在树莓派上重新安装一个新的 Debian 系统,它也适合一些小容量的 SD 卡作为启动盘来启动。

致 谢

本书在编写过程中得到了很多人的帮助,在这里我要感谢我的同事曹雪飞和李霞,他们不但承担了我的一些工作,使我有更多的时间完成本书,而且她们对本书提出了很多指导性的意见;另外,还要感谢于云,是她承担了本书的校对工作,以及感谢爱板网提供的帮助,是他们让我在第一时间获得了树莓派。

最后要感谢公司给我提供了一个宽松的工作环境,让我有足够的时间来完成这本书。

由于编者水平有限,书中难免有疏漏和错误之处,恳请广大读者批评指正。

作 者
2013 年 7 月

目 录

第 1 章 什么是树莓派	1
1.1 目前树莓派的版本	5
1.2 树莓派的硬件及接口	6
1.2.1 CPU 与内存	6
1.2.2 显示接口	6
1.2.3 USB 与 LAN 接口	7
1.2.4 其他接口	7
1.3 树莓派的硬件组装	8
1.4 如何获得树莓派	8
1.4.1 通过爱板网购买	9
1.4.2 通过淘宝购买	9
1.5 树莓派的周边设备	10
1.5.1 选择电源	10
1.5.2 无线网卡	10
1.5.3 USB 摄像头	11
1.5.4 SD 卡	11
1.5.5 键盘与鼠标	12
1.5.6 外 壳	12
1.5.7 HDMI 转 VGA 接口	14
1.6 获得帮助	14
第 2 章 Linux 系统基础	16
2.1 Linux 操作与布局	16
2.1.1 Linux 开机	16
2.1.2 Linux 关机	17
2.1.3 Linux 系统文件结构	17
2.1.4 Linux 的文件类型	19

目 录

2.1.5 常用命令	20
2.2 用户与权限管理	23
2.2.1 用户管理	23
2.2.2 权限管理	24
2.3 安装/删除软件	26
2.3.1 Debian 包管理	27
2.3.2 Arch 包管理	29
2.4 文本编辑器	29
2.4.1 nano	30
2.4.2 vi	30
2.5 使用 USB 存储设备	33
2.6 时间设置	34
2.6.1 手动设置系统时间	34
2.6.2 使用网络设置时间	34
第 3 章 使用树莓派	35
3.1 选择操作系统版本	35
3.1.1 Debian Linux	35
3.1.2 Arch Linux	36
3.1.3 Xbian	36
3.1.4 Raspbmc	37
3.1.5 其他系统	37
3.2 安装操作系统到 SD 卡上	39
3.2.1 校验镜像文件	39
3.2.2 安装操作系统	40
3.2.3 安装操作系统到 U 盘	42
3.2.4 红派所使用的系统	44
3.2.5 还原 SD 卡	45
3.2.6 更新系统	47
3.3 设置启动参数	52
3.3.1 使用 raspi-config 配置树莓派	52
3.3.2 分辨率设置	56
3.4 使用网络	56
3.4.1 设置有线网络	56
3.4.2 设置无线网络	59
3.5 中文环境	63

3.5.1	设置中文环境	63
3.5.2	安装中文字库	65
3.5.3	安装中文输入法	66
3.5.4	命令行中文支持	67
3.6	使用 X 窗口系统	68
3.6.1	使用云程序	68
3.6.2	使用 OpenOffice	71
3.6.3	使用 Gimp 编辑图片	72
第 4 章	媒体中心与游戏	74
4.1	媒体播放	74
4.1.1	XBMC 程序	74
4.1.2	Raspbmc	76
4.1.3	使用遥控器	85
4.1.4	音乐播放中心	87
4.2	游 戏	88
4.2.1	MineCraft	89
4.2.2	Quake 3	90
4.2.3	街机模拟器	91
第 5 章	迷你服务器	93
5.1	增加 SWAP 分区	93
5.1.1	建立 SWAP 分区	95
5.1.2	建立 SWAP 文件	96
5.2	WEB 服务与 WEB 脚本语言	97
5.2.1	Apache	97
5.2.2	Lighttpd	100
5.2.3	PHP	103
5.2.4	Perl	106
5.2.5	Python	108
5.3	数据库支持	110
5.3.1	MySQL	110
5.3.2	PostgreSQL	111
5.4	设论坛与 Wiki 网站	112
5.4.1	论 坛	112
5.4.2	建立 Wiki	116

目 录

5.5 共享文件	120
5.5.1 Samba	120
5.5.2 SFTP	123
5.6 无线路由器	125
5.6.1 手动配置无线路由	125
5.6.2 openwrt	127
5.7 远程管理	128
5.7.1 VNC 远程桌面	128
5.7.2 Xrdp 远程桌面	131
5.7.3 SSH	131
5.8 P2P 下载服务	134
5.8.1 Rtorrent	135
5.8.2 Transmission	136
5.8.3 Mldonkey	139
第 6 章 扩展接口及应用	143
6.1 使用 GPIO 建立内存警报指示灯	145
6.1.1 建立基本电路	146
6.1.2 通过 GPIO 控制 LED 灯	146
6.1.3 建立内存警告系统	147
6.2 通过 I ² C 接口连接温度传感器(LM75)	150
6.2.1 连接 LM75 到树莓派上	150
6.2.2 为树莓派启用 I ² C 功能	151
6.2.3 安装 i2cdetect 工具并检测温度传感器	151
6.2.4 使用 python 来访问温度传感器	152
6.3 树莓派上使用串口 GPS	156
6.3.1 为树莓派启用串口	157
6.3.2 连接 GPS 模块并启用控制台功能	158
6.3.3 连接 GPS 并获取位置信息与时间	159
6.3.4 使用 python 读取 GPS 信息	160
附录 A 树莓派显示模式表	162
附录 B 常见的树莓派扩展板	164
B.1 GPIO 串口扩展板	164
B.2 全功能扩展板	164

B.3	时钟扩展板	165
B.4	电源扩展板	165
B.5	LCD 按键扩展板	166
B.6	树莓派气象站扩展板	167
B.7	树莓派 ADC 扩展板	167
B.8	步进电机扩展板	168
附录 C	手动安装 Debian 系统	169

第 1 章

什么是树莓派

树莓派是一款只有信用卡大小的个人计算机,它是由英国的 Raspberry Pi 基金会开发的,最初的设计目的是以超低的硬件价格及开放的自由软件来为发展中国家的学生提供一个基本的计算机编程环境。但是随着树莓派计算机的推出,它已经成为众多计算机爱好者的新工具,通过树莓派这种超低成本迷你计算机,可以完成许多以前无法完成的事,例如将树莓派和摄像头一起放在探空气球中,记录天气情况或者是用它来完成智能家电的控制等。

事实上,早在 2006 年的时候,现在的树莓派基金会受托人埃·厄普顿(Eben Upton)还在英国剑桥大学的计算机实验室时就发现,每年申请读计算机专业的学生,编程能力都有下降。他们越来越依赖于图形界面,而不是像 20 世纪 90 年代那样每个来读计算机专业的人都有一定的编程能力。因此,埃·厄普顿与他的同事打算一起开发一个价格便宜的迷你计算机。在受到 Atmel 的 ATmega644 单片机的启发后,他们设计出了最初的树莓派计算机。从 2006 年起一直到 2008 年,埃·厄普顿设计了许多版本的树莓派原型,图 1-1 所示就是 2006 年树莓派的最初版本(原型),整个成本只需要 25 美元。它使用了 Atmel 公司的 ATmega644 微控制器,而时钟频率只有可怜的 22.1 MHz,并配有一个 512K 的 SRAM。

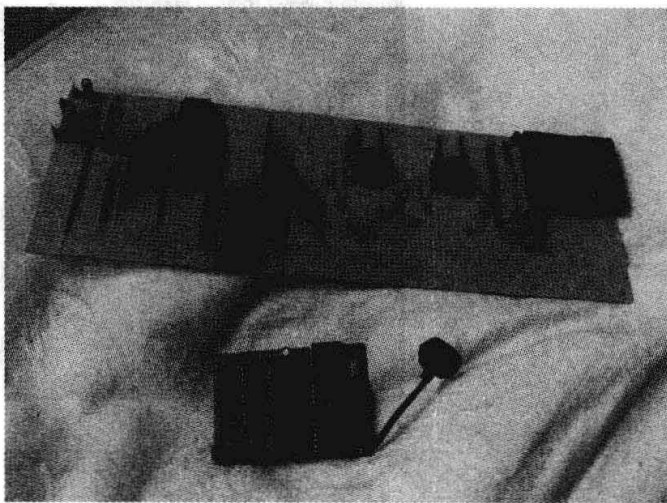


图 1-1 树莓派计算机最初的原型

到了 2008 年,由于技术的更新,专为如智能手机等移动设备而设计的处理器开始变得便宜了,同时也能提供更多的多媒体功能,于是埃·厄普顿和 Egypt、Rob、Jack 等人开始了树莓派项目并最终成立了树莓派基金会,于 2011 年设计并推出了第一个树莓派的阿尔法版。从图 1-2 可以看出,它已经和目前用户所使用的树莓派很像了。

这个原型机尺寸非常小,已经可以装在一个名片盒子中了,基本上现在用户所使用的树莓派接口(HDMI、USB、网络等接口)都有出现。

第1章 什么是树莓派

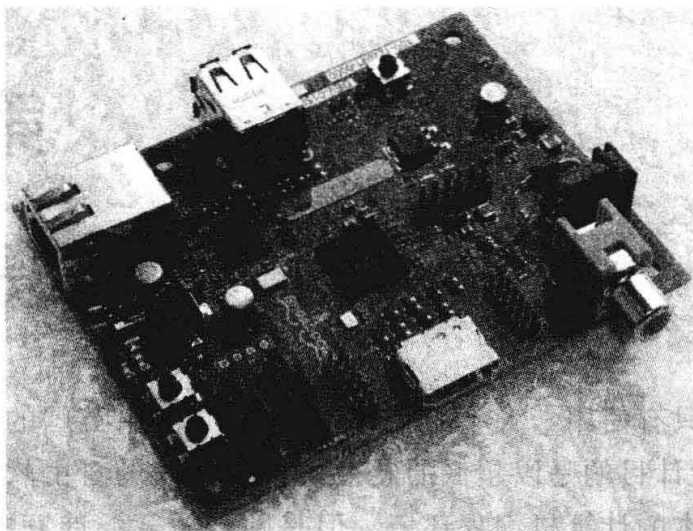


图 1-2 树莓派计算机的工程版

2012 年 1 月正式生产的树莓派,无论是外形还是质量都比之前的阿尔法版要好很多。在正式版本中使用了 700 MHz 的博通生产的 ARM 架构的 BCM2835 处理器,并配有 256 MB 内存,使用 SD 卡当做储存介质,并拥有一个有线网络接口、两个 USB 接口,以及 HDMI(支持声音输出)和 AV 端子输出。正式版分 A、B 两种型号,售价分别是 A 型 25 美元,B 型 35 美元。图 1-3 就是最终版投产的树莓派。

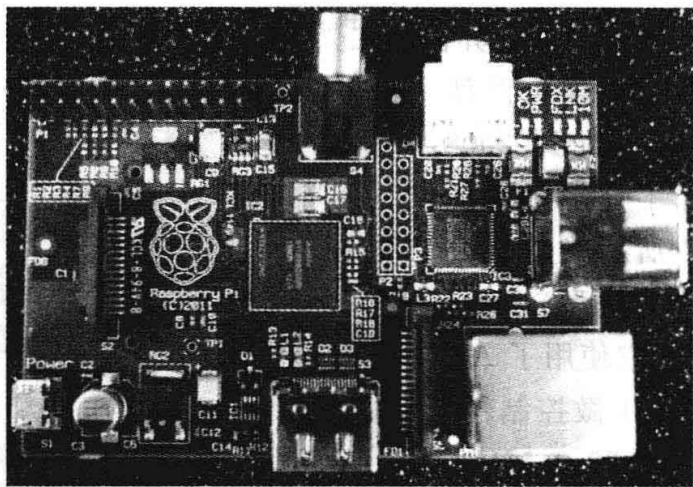


图 1-3 树莓派计算机的最终版

最终版的树莓派只有一张信用卡大小,而 CPU 的性能相当于 300 MHz、奔腾 II 的水平,图形处理能力相当于 Xbox 1 游戏机的水平。因此,它可以轻松运行像雷神之锤 III 竞技场和播放 1080p 电影。对于操作系统则是由树莓派基金会提供了 Debian 和 Arch Linux 的 ARM 发行版,通过自带的 Iceweasel、KOffice 等软件来满足基本的网络浏览、文字处理以及编程学习的需要。

图 1-4 是 2012 年 10 月推出的树莓派第 2 版,是最近一次改进后推出的。由于生产成本的下降,原有的内存从 256 MB 升级到了 512 MB,同时 PCB 主板也做了一

些修整,例如增加了定位孔、去除了 USB 供电的限流保险丝、USB HUB 芯片的电源控制功能,另外,GPIO 接口等也做了调整。

目前,国内用户所使用的树莓派大部分也是这个版本,本书也是以这个版本作为硬件平台来进行测试的。

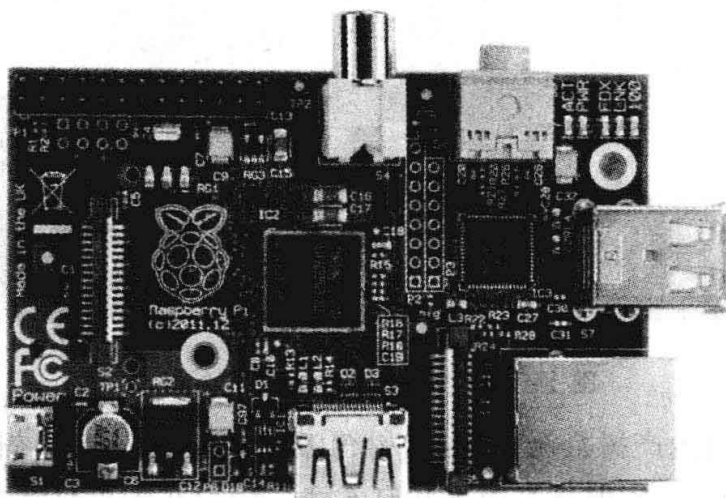


图 1-4 树莓派的第 2 版

红色中国版,2013 年 2 月 1 日由深圳市韵动电子有限公司获得授权生产,目前这个版本只在中国地区销售,为了与英国生产的树莓派区分,该版本使用红色 PCB 制造。其技术参数与目前生产的 B 型一样,都是 512 MB 的 REV2 版本。如图 1-5 所示,虽然整个 PCB 与芯片都与英国生产的一样,但目前还是有一些网友表示红色版树莓派的生产工艺没有英国生产的好。

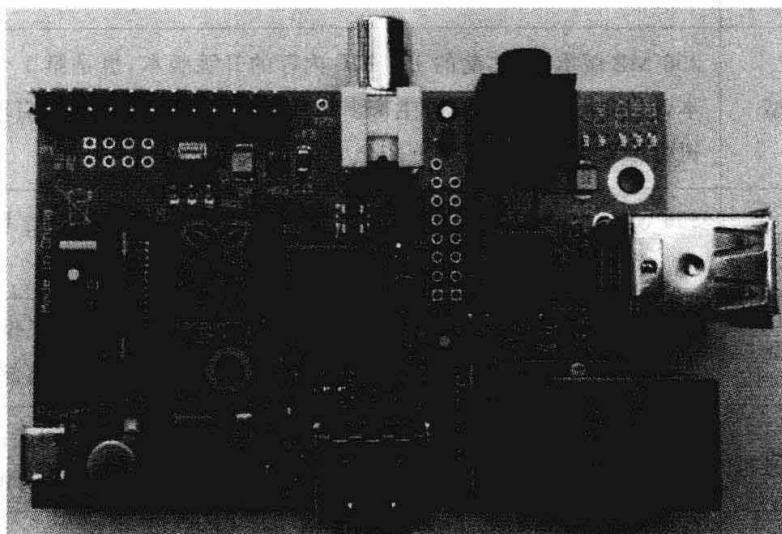


图 1-5 由中国生产的红色版树莓派

树莓派 A 型,这个版本是最近才开始生产的。从图 1-6 就可以看出没有网卡和 LAN9512 芯片,同时 PCB 板也做了与 REV2 B 型一样的调整。因为零件的减少使得售价降至 25 美元,不过目前只能在欧洲买到,其他地区的用户还得等待。

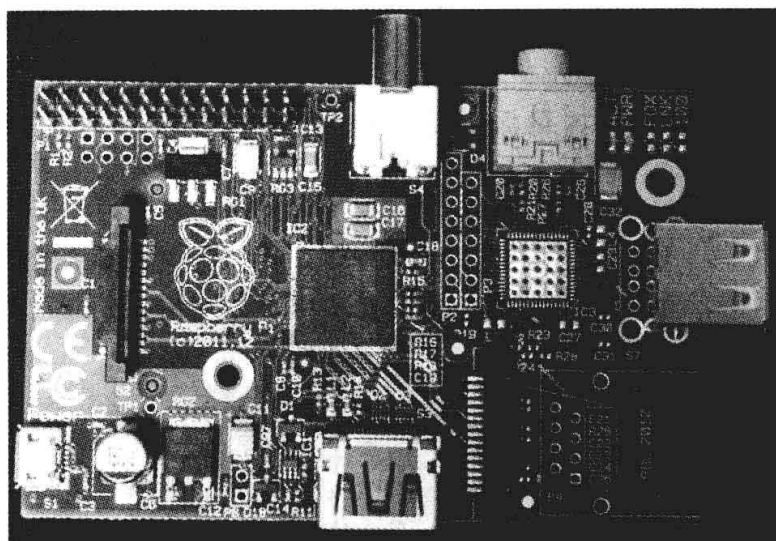


图 1-6 树莓派 A 型的第 2 个版本

从 2012 年开始生产树莓派到现在,树莓派的主板经历了复杂的演变,目前看来虽然只有 A、B 两个型号,但是其版本却有 6 个。用户在购买时需要注意它们之间的区别,总体来说,A 型比 B 型便宜,256 MB 内存的比 512 MB 内存的便宜。对于树莓派每个版本的详细说明可以参见表 1-1。

表 1-1 树莓派版本详细说明

序 号	型 号	说 明
1	A 型	128 MB 版本,这是最初设计的版本,当时由于成本考虑只有 128 MB 内存,没有网卡接口,但是后来正式生产时由于硬件成本的下降,因此这个版本演变成了 A 型 256 MB 版本
2	A 型	256 MB 版本,是之前的 128 MB 内存的升级版本,也是第 1 个正式生产的版本,但是这个版本并没有生产多久就被广大用户对 B 型的需求所淹没。于是树莓派基金会不得不停下来全力生产 B 型 256 MB 版本
3	B 型	256 MB 版本,与 A 型相比,多了网卡接口。这个版本也是 B 型的第 1 个版本。淘宝上还有少量的商家在卖这种型号
4	B 型	512 MB 版本,原有 B 型的升级版本,增加了定位孔,修改了供电电路,改动了 GPIO 接口位置等,目前能在国内买到的英国产的 B 型树莓派也是这个版本
5	B 型	红色中国版,2013 年 2 月 1 日由深圳市韵动电子有限公司获得授权生产只在中国地区销售的树莓派,并从 2013 年 2 月 20 号开始正式销售
6	A 型	2013 年 2 月 4 日开始生产的 256 MB 无网卡版本,是之前 A 型的升级版本,在主板上有定位孔等改动,因为零件的减少使得售价降至 25 美元,不过目前只能在欧洲买到,其他地区的用户还需等待

1.1 目前树莓派的版本

目前,树莓派共有 6 个版本。其中,5 个版本都是 A、B 两种型号的改进版本。除了最早的 128 MB 内存版本没有生产之外,以及刚取得授权的 B 型中国版(中国版才刚开始生产)以外,其他 4 个都曾大规模生产过。

目前国内市面上可以买到的树莓派主要是 256 MB 和 512 MB 的 B 型。不过当读者看到这本书的时候,红色版本的树莓派就已经大量上市了。树莓派的 A、B 型硬件参数如表 1-2 所列。

表 1-2 树莓派 A、B 型硬件参数表

项 目	A 型	B 型
上市价格	\$ 25	\$ 35
SOC	Broadcom BCM2835 (CPU, GPU DSP 和 SDRAM, USB)	
CPU	ARM1176JZF-S 核心 (ARM11 系列) 700 MHz	
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, 1080p 30h. 264/MPEG-4 AVC 高清解码器	
内存	256 MB (与 GPU 共享)	512 MB (与 GPU 共享)
USB 2.0	1 (支持 USB hub 扩展)	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 resolutions from 640×350 to 1920×1200 plus various PAL and NTSC standards	
音源输出	3.5 mm 插孔, HDMI	
板载存储	SD/MMC/SDIO 卡插槽	
网络接口	无,但可通过 USB 扩展	10/100 M 以太网接口 (RJ45 接口)
外设	8×GPIO/UART/I ² C, 带两个选择的 SPI 总线, +3.3 V, +5 V, ground (负极)	
额定功率	300 mA (1.5 W)	700 mA (3.5 W)
电源输入	5 V / 通过 MicroUSB 或 GPIO 口	
总体尺寸	85.60 mm×53.98 mm (3.370 in×2.125 in)	
质量	45 g	
操作系统	GNU/Linux (Debian, Arch Linux ARM, RISC OS, FreeBSD, Plan 9)	

1.2 树莓派的硬件及接口

1.2.1 CPU 与内存

出于成本的考虑,树莓派所使用的 CPU 是 ARM 架构的,与大多数用户平时所使用 X86 或是 AMD64 架构不同的是:ARM 是一款精简指令集(RISC)处理器;它的指令系统非常简单,只能让硬件执行最常用的且有限的指令。这使得 ARM 架构在一些专业用途的服务器中拥有很高的执行效率,例如 IBM 公司的 POWER PC 系列服务器就是使用的 RISC 架构的处理器。而它超低的功耗需求,则非常适合那些对电源消耗敏感的设备,如手机、嵌入式等系统。

在树莓派上,由于采用的是 CPU 与内存堆叠方式连接的,因此,我们只能看到堆叠在最上面的三星 DDR2 内存,而下面的 CPU 则是由 Broadcom 生产的 BCM2835 芯片。在这个芯片中,集成了现代计算机运行时所需要的大部分硬件,如中央处理器、图形处理芯片以及音频芯片等在内的大部分芯片。虽然芯片名称是 BCM2835,但它的内核其实是在 ARMv6 基础上实现的 ARM1176JZ,并且工作频率达到了 700 MHz。虽然频率不高,但是 ARM1176JZ 保证了在有限的功耗与面积内实现了超高性能。

ARM11 处理器的特点:

- ① 提供动态预测与静态预测两种预测方式,这使 ARM11 处理器能达到 85% 的预测正确性,并给每一个正确的预测减少 5 个时钟周期的等待时间。
- ② 增强的存储器访问使得数据可以更长时间地保存在 Cache 中。
- ③ 在 32 位的核心系统中使用了 64 位的数据通道,这样处理器每个周期可以读入两条指令或存放两个连续的数据,从而提高数据访问和处理的速度。
- ④ 用户可以在 ARM11 中加入浮点运算来达到更高的性能。

虽然以上特性看起来很好看,但不得不说的是,如果这些参数放在 2006 年以前,那一定是极好的,但是现在则有些落后了。如果只用它来运行一些简单的 Linux 系统的桌面程序或者使用命令行模式来操作,还是没有问题的,而那些想要追求较快运行速度的用户可能需要做好心理准备,因为树莓派的运行速度可能会让他们觉得时间过得太慢了。

1.2.2 显示接口

目前,树莓派提供了三种显示输出方式:HDMI(高清晰度多媒体接口)、AV 端子以及 DSI 输出。在这三种输出方式中,推荐使用 HDMI 输出,因为这是最简单的输出方式,而且显示清晰,没有信号的损失。

HDMI 接口的全称是高清晰度多媒体接口(High Definition Multimedia Inter-

face, HDM), 是一种全数字化的视频、音频、网络接口技术, 所有的数据都直接以数字方式传输, 非常适合用来传输数字图像。在树莓派提供的三种方式中, HDMI 输出使用是最方便的。如果用户的电视机或者显示器拥有一个 HDMI, 那就最好不过了, 只须使用 HDMI 线将显示设备与树莓派连接起来就可以了。

AV 端子就是我们常说的莲花插座、AV 接口, 目前几乎所有的电视机中都有这个接口。它除了可以传输视频外, 还可以输出两路(左、右)音频, 因此常见的 AV 接口都是三个, 但树莓派上没有 AV 音频接口, 所以只能使用一个 AV 接口来传输视频。

虽然使用 AV 端子会有一些色彩信号的损失, 但是对于那些没有支持 HDMI 接口的显示器的用户来说, 只能使用这种方式来连接电视机作为显示输出了。

DSI 是一种以串行传输方式来输出视频的方式。在树莓派的 SD 卡反面可以看到一个由塑料保护的接口, 这就是 DSI 接口, 通常用来连接平板电脑或者手机的显示屏。用户自己将 DSI 显示屏接口与主板上的 DSI 接口连接, 就可以使用了。但麻烦的是, 这种带 DSI 接口的显示屏在零售市场很少有, 而且显示的效果也不是很好, 因此并不推荐用户使用这些输出方式。

1.2.3 USB 与 LAN 接口

USB 与网络接口, 这两种接口大家都很熟悉。需要说明的是, 树莓派所集成的网络接口也是由 USB 转换出来的 10/100 MB 网卡, 它并不是由专门的网卡芯片来进行控制的, 因此也就不具备一些如网络唤醒、远程开机等功能; 而这个网卡对于那些使用千兆网络的用户来说可能还是一个灾难。在某些千兆无线路由器(如 WRT350Nv2)中, 树莓派由于不支持千兆网络, 所以根本无法连接到路由器(对于如 WRT350Nv2 这样的路由器, 可以通过刷 openwrt 的固件来解决这个问题)。

对于 USB 接口的使用, 两个 USB 口实在是太少了, 笔者建议使用一个有电源的 USB 扩展坞, 以保证 USB 设备的供电, 避免因供电不够而导致 USB 设备无法使用(这种情况是经常会出现的, 如果使用 USB 扩展接口, 请一定要使用带电源的)。

1.2.4 其他接口

由于树莓派使用 SD 卡作为硬盘, 因此整个操作系统的读写都是在 SD 卡上完成的。如果条件许可, 推荐用户使用高速的 SD 卡, 以获得最大的读写速度。使用时须注意 SD 卡的接口, 网络上已经有很多人反映 SD 接口部分很容易损坏, 因此请小心使用。如果安装好后发现系统无法启动, 可以尝试重新拔插 SD 卡, 以保证连接正常。

树莓派支持音频输出, 除了使用 HDMI 输入音频, 还可以使用自带的 3.5 mm 的音频输出来连接, 读者可以使用耳机或者扬声器进行播放。

1.3 树莓派的硬件组装

用户收到的树莓派通常是一块信用卡大小的主板,而树莓派的接口都是常用的接口,所以树莓派的硬件组装较为简单。每种设备接口的形状都是不同的,所以用户只需要大胆地将需要连接的设备插入可连接的接口上就行了。

对于连接方式不太了解的用户可以参考图 1-7。

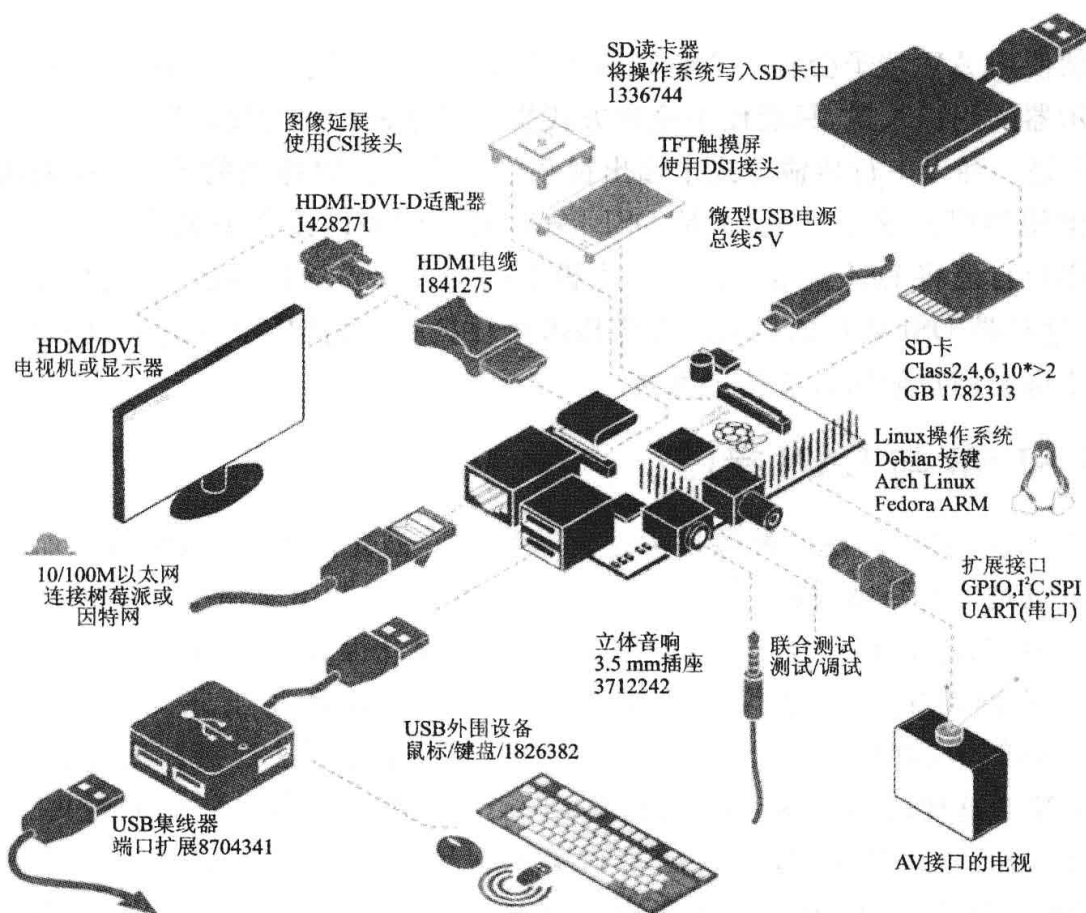


图 1-7 树莓派连接示意

1.4 如何获得树莓派

目前,树莓派是在英国生产的(早期有极少的版本是由中国生产的),如果国内用户想要直接购买树莓派会比较麻烦,目前可以通过三家公司(Element 14/Premier Farnell、RS Electronicsc 和 Allied Electronics)在全球范围内进行购买。如果读者想要购买英国生产的树莓派,可以选择以上三家公司的任何一家来购买,不过可以确定的是,由于树莓派的火爆热销,无论通过哪一家经销商购买都需要好几个星期才能收到货。

1.4.1 通过爱板网购买

爱板网是一个为广大电子工程师及相关专业老师、学生提供开发板选型帮助的专业网站。网站有各个品牌厂商开发板和评估板产品数据资料供下载,同时还提供详细的第三方评测报告,此外用户还能获得最新的开发板例程代码、视频、教程信息。爱板网一直致力于成为华语地区开发板和评估板选型的首选参考网站。

爱板商城,是爱板网旗下专门的开发评估板电子商务平台,除了销售树莓派主板及附件之外,还通过与易洛盟、Digilent 公司达成合作关系并销售 MCU 开发板、FPGA/CPLD 开发板和机器人开发模块等产品。此外,爱板商城还能提供商品的详细技术性能参数、100%实样数码照片和商品的相关技术资料。读者可以通过网站 <http://www.eeboard.com/shop/pi> 进行购买,如图 1-8 所示。



图 1-8 爱板网提供的英国生产的树莓派

1.4.2 通过淘宝购买

对于国内的用户来说,如果想找一样身边实体店没有的商品,最好的方式就是通过淘宝,目前在淘宝上已经可以买到包括树莓派及周边附件在内的大部分商品了。

使用淘宝购买有三个好处:第一,读者可以选择离自己最近的卖家,以便在最短的时间内收到货。这对于一些心急而不在乎费用的用户来说是不错的选择。第二个好处就是方便购买二手的树莓派。一些用户因为不熟悉树莓派所使用的 Linux 系统而选择通过淘宝来处理它,这时也可以淘到很便宜的树莓派。笔者的第一块树莓派就是通过淘宝购买的:256 MB 的 B 型、无线网卡以及 8G 的 Class10 SD 存储卡,整个加起来才 220 元,非常划算。第三,很多淘宝商家在卖树莓派的同时还卖树莓派的

第1章 什么是树莓派

周边附件,因此可以通过一家店买全所需要的全部商品。例如,已经安装好系统的SD卡、外壳等商品,从而节省了精力与邮费。

1.5 树莓派的周边设备

树莓派有丰富的设备接口,这也决定了它将会拥有丰富的周边附件。本节将介绍一些常用的附件,以及使用这些附件会带来问题,最大程度地避免用户因为周边设备的原因而无法使用树莓派。

1.5.1 选择电源

树莓派 A 型使用的电源是 5 V/500 mA 的,B 型用电量为 5 V/700 mA。虽然 A、B 两种型号的耗电量有一些差别,但它们之间的电源是可以通用的,加上 USB 设备,所以通常为树莓派提供的电源为 5 V/1 A。目前,几乎所有的智能手机都使用 5 V/500 mA、5 V/700 mA、5 V/1 A 这三种型号的充电器,所以在电源选择上用户可以较为轻松,只需要选择做工合格质量有保证的充电器即可。唯一需要注意的是,电源需要支持通过 micro-USB 接口进行连接。

如果用户的手机充电器不支持树莓派的电源规格或是接口无法使用,就要考虑重新购买一个 USB 电源了。这里推荐奔迈 PALM-USB 电源,支持输入电压为 100~240 V,输出 5 V/1 A 或者 5 V/2 A,加上可折叠与可拆卸的插头,特别适合与树莓派一起携带。

而连接树莓派的 micro-USB 接口的数据线实在是太容易获得了,即使用户没有这种数据线,也可以在自己所在城市的电子市场里轻松买到。

请一定购买做工合格的电源,在最新的 512 MB v2 版本的树莓派中已经取消了 USB 供电保险,因此如果使用了不合格的电源,则可能会直接烧掉整个树莓派主板。

1.5.2 无线网卡

如果想要通过无线网络来进行连接,那么就必须额外购买一个无线网卡。对于树莓派所运行的 Linux 系统来说,由于驱动程序的关系并不是所有的无线网卡都可以支持,所以要通过查看内核来了解 Linux 系统所支持的无线网卡芯片型号。

从图 1-9 可以看出,Linux 内核支持较多的无线网卡芯片,常用的 Broadcom 43xx 系列、Marvell 8xxx 系列或者 rtl81 等芯片都是支持的。用户在购买时须查看一下无线网卡的评测,或者查看包装上的说明是否支持 Linux 系统。

对于那些需要通过在树莓派上加装无线网卡来完成路由功能的用户来说则比较麻烦,因为用户所购买的无线网卡不但需要有相应的 Linux 系统驱动程序,还需无线网卡支持无线 AP 功能。由于 Linux 程序是通过 hostapd 程序来管理无线 AP 功能的,因此可查看 hostapd 程序的网站 <http://hostap.epitest.fi/hostapd/> 来获得 hos-

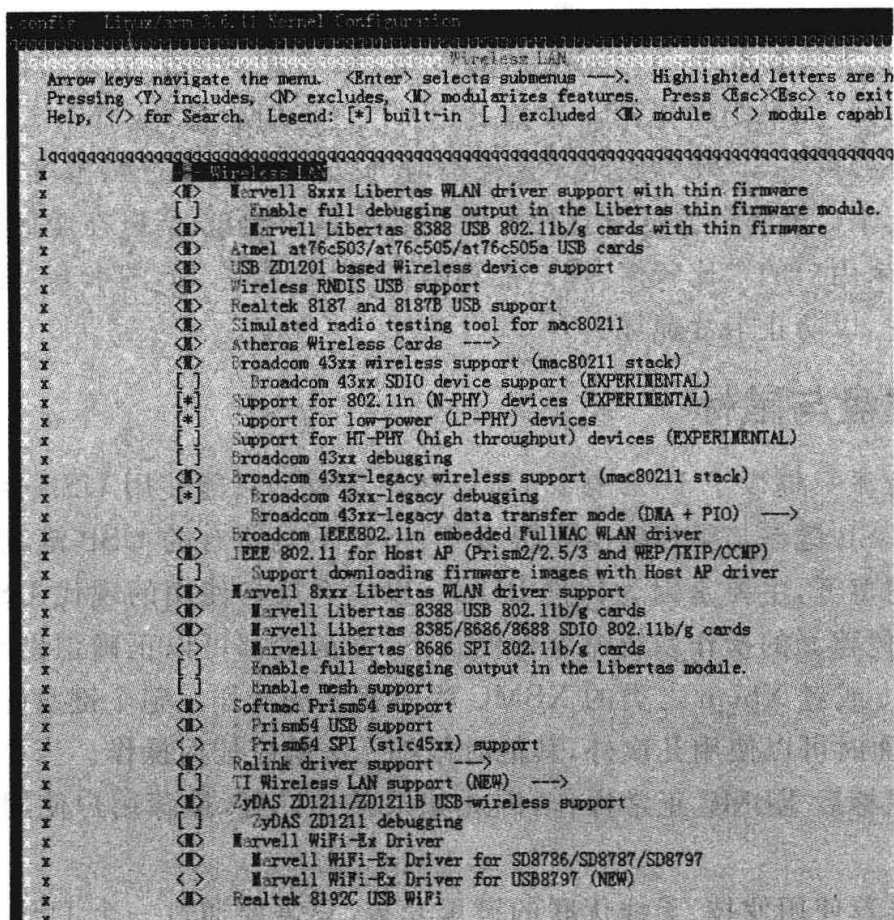


图 1-9 Linux 内核所支持的部分无线网卡

tapd 程序的相关信息。对于无线网卡所支持的硬件列表,可以参见 <http://wireless.kernel.org/en/users/Drivers> 页面,并根据硬件列表来选择无线网卡。

1.5.3 USB 摄像头

和无线网卡类似,在树莓派上使用 USB 摄像头也需要看型号,只有那些被内核支持的摄像头才可以使用。这里推荐使用 UVC 的 USB 摄像头。在 Linux 系统下,大部分的 UVC 摄像头都可以直接通过内核自带的驱动来使用,例如 uvc301。

1.5.4 SD 卡

树莓派的系统是存放在 SD 卡上的,因此 SD 卡的读写速度就决定了整个系统的读写速度。如果想要获得较高的运行速度,就需要采用高速的 SD 卡,例如使用 class10 的 SD 卡。不幸的是,SD 卡在树莓派上因为兼容性的原因而导致真实的读写速度并不能达到所标称的那么高。较为常见的读写速度一般为 10~20 MB。另外,由于兼容性问题,有一些 SD 卡无法在树莓派上使用,读者可以参考 http://elinux.org/RPi_SD_cards 页面。在这个页面上为树莓派测试了目前主流的 SD 卡的支持情况,读者可以查看后进行购买。

对于 SD 卡的容量,笔者推荐使用 4G 或 8G 的 SD 卡,因为无论是用 XBMC 还是 Debian 来做系统盘都不会占用太多的空间,哪怕是少量的个人数据也可以通过 SD 卡进行处理。如果要用树莓派来挂 BT、PT 之类的服务,或者个人数据量较大,则推荐使用外接的 USB 硬盘底座。这样不但可以扩展存储空间,也可以将以前的闲置硬盘利用起来。通常硬盘底座比硬盘盒更容易更换硬盘,所以在使用上也更方便。

另外,如果用户的卡比较多,那么推荐购买些 SD 或者 TF 的卡套,可用来保存平时不用的卡片,以防止卡片损坏。

1.5.5 键盘与鼠标

因为树莓派只提供了 USB 接口,所以在树莓派上只能使用 USB 接口的键盘与鼠标。虽然 USB 键盘与鼠标推出了很多年,但并不是所有的 USB 键盘和鼠标都能完美地支持树莓派,主要是因为软件 BUG。以笔者最初使用的现代 USB 有线键盘为例,在任何树莓派的操作系统命令行模式下,该键盘都可以正确识别与使用,一旦使用 Raspbmc 或者 Xbian 之类的 XBMC 程序就无法操作。整个键盘除了刚开机进入 XBMC 界面时可以使用几次外,其他时候都无法响应任何操作。当更换为无线套装的键盘和鼠标后 XBMC 正常操作,因此如果条件许可,推荐用户使用无线套装的键盘鼠标。

在这里推荐使用罗技、无线优联的键鼠套装,只需要通过一个 USB 接收器就可以同时完成键盘与鼠标的连接,而空出的另一个 USB 口可以用来连接无线网卡,实在是很方便。

1.5.6 外壳

在树莓派的论坛或 QQ 群里外壳总是一个话题,从最开始的使用名片盒自己动手做到塑料,再到目前最新的亚克力彩壳,总是能吸引很多的用户。其中,亚克力外壳受生产成本的影响而价格最高。一个树莓派所用的亚克力外壳,售价可以达到 30~50 元,虽然较贵,但目前较为流行的外壳还是由亚克力材料制作的,读者可以在淘宝上购买。

目前,可买到的亚克力外壳分为三种:一种是直接拼装的;另一种是由上、下两块外壳通过螺丝进行连接,较为有名的是狗骨头版外壳;第三种则是由多层亚克力板进行堆叠而成的,通过尼龙螺丝进行连接。

三种外壳中,直接拼装的基本上是全封闭的,能有效保护主板不受灰尘的影响,但是卡口较小,因此比较容易损伤到连接部位。除这个缺点以外,基本上还算是很不错的外壳,同时这种外壳也是淘宝上有最多商家在卖的外壳。图 1-10 就是直接拼装的外壳。

狗骨头版外壳由于只是上、下两块外壳进行包裹,因此很容易通过使用更长的连接柱来进行扩展,以方便加入扩展板。这对于那些经常需要通过 GPIO 连接或者使

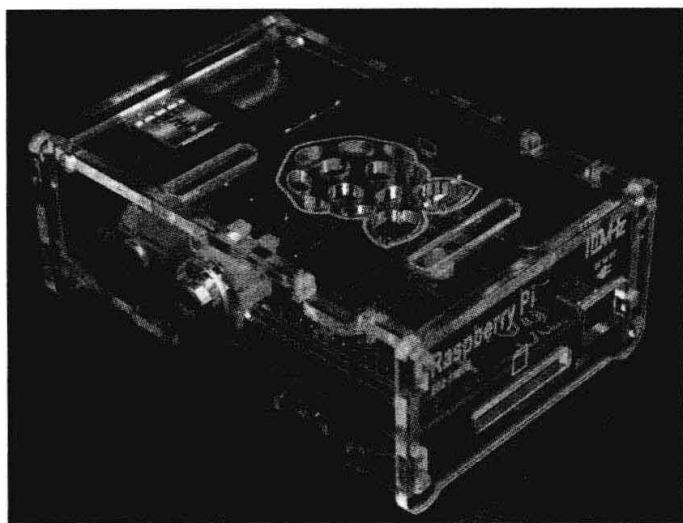


图 1-10 直接拼装的外壳

用扩展板的用户来说,是最合适的选择,同时这种外壳的散热性也是最好的(事实上,即使树莓派全速运行很长时间,也没有什么热量)。四个角的卡口可以有效地固定树莓派主板,而不像直接拼装的外壳那样会有轻微的晃动,其缺点是不能有效防尘。图 1-11 所示为狗骨头版外壳。



图 1-11 狗骨头版外壳

最后一种堆叠而成的,如图 1-12 所示,外形比较好看,目前原版外壳售价高达 160 元人民币,而国内相似的外壳售价也达到了 58 元。除了不方便接扩展板且价格超贵以外,基本上找不到什么缺点了。如果用户喜欢外观漂亮并且不打算使用扩展板则推荐使用这种外壳。

目前能买到的这种亚克力彩壳是蓝色的,其彩虹壳也即将上市,相信读者看到本书时,已经可以在 pibox.taobao.com 上购买到这两种彩壳了。

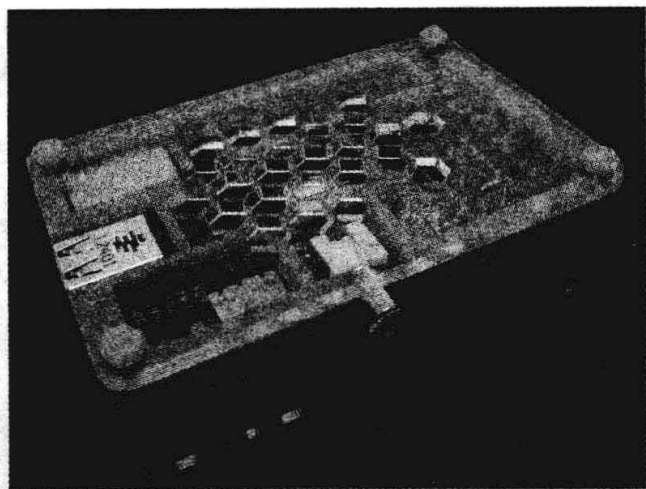


图 1-12 亚克力彩壳

1.5.7 HDMI 转 VGA 接口

如果用户的显示设备没有 HDMI 接口,那么可以选择 HDMI 转 VGA 接口,因为它可以将树莓派的 HDMI 输出转换成标准的 VGA 接口。这种转换接口可以在电子市场或淘宝上买到。用户需要注意的是,HDMI 的默认输出分辨率为 $1920\text{ ppi} \times 1080\text{ ppi}$,而不支持 HDMI 输入的显示设备通常都不支持这么高的分辨率。这样即使用户通过 HDMI 转 VGA 接口连接了树莓派与显示器,也会因为无法支持这么高的分辨率而无法显示。这种情况下,用户可以通过修改启动参数来进行调整。读者可参考第 3 章中“分辨率设置”来进行设置。

HDMI 转 VGA 接口有两种,一种需要接电源,另一种则无需接电源,推荐使用需要接电源的这种。目前市场上 HDMI 转 VGA 接口有很多种,各种转换芯片混杂其中,用户在购买时需要仔细查看。

1.6 获得帮助

对于树莓派所运行的 Debian 与 Arch Linux 来说,自带的 Man 文档与 info 帮助页面涵盖了 Linux 系统几乎所有的内容;对于那些有一定英文阅读能力的用户来说,这是一个学习的好助手。但是,对于那些不熟悉 Linux 又想使用树莓派的用户来说,通过网络来寻求帮助是最好的方式。用户可以通过以下论坛进行求助:

① 树莓派官方论坛 <http://www.raspberrypi.org/phpBB3/>

涵盖了树莓派的方方面面,从使用到编程,从教育到各种项目,无所不包,是最权威的树莓派论坛。不过目前还没有中文讨论区,因此只能使用英文进行交流。

② 树莓派杂志 <http://www.themagpi.com/>

每个月更新一次的免费树莓派杂志,除了 2012 年 6 月有一期中文外,其余都是

用英文编写的。每一期都会介绍一些树莓派的最新消息与应用。推荐树莓派用户定期查看。

③ 爱板网树莓派版块 <http://www.eeboard.com/pi>

爱板网树莓派讨论区,是 element14 官方指定的技术支持,为广大树莓派爱好者提供良好的交流平台。讨论区汇集众多树莓派爱好者,拥有大量评测文章、教程资料、开发项目、心得分享等,用户可以在此咨询疑难问题、交流经验、分享项目,共同学习。

④ 树莓派论坛 <http://www.shumeipai.net/>

国内最早的树莓派论坛,几乎所有国内第一批的树莓派用户都是在这里进行交流的。板划分得较为细致,并且有一个核心教程区,用户可以在这里找到大部分的基础教程。

第 2 章

Linux 系统基础

Linux 是一套免费的类似于 UNIX 的操作系统, Linus Torvalds 最初设计它的时候只是为了方便自己从大学的服务器上下载邮件与新闻组, 但是后来加入了磁盘驱动程序和文件系统, 渐渐地这个软件也成了 Linux 第一个内核的雏形。由于 GPL 的通用性授权允许用户销售、复制并且改动程序, 而且必须免费公开修改后的代码, 这使得越来越多的人开始加入到 Linux 内核的开发工作中来。

单从 Linux 这个词来说, 只表示操作系统内核本身, 但通常采用“Linux 内核”来表达这个意思, 而 Linux 一词则用来指基于 Linux 内核的完整操作系统。这个系统中包含了 GUI(图形界面)组件和许多其他实用工具, 很多组件与工具都是由 GNU 计划提供的。

由于 Linux 系统具有开放源代码的特性, 所以它很容易被移植到各种硬件设置上。目前, 全球有大量的设备运行着 Linux 系统, 从手机到路由器, 从航天飞机到战略核潜艇, 几乎需要操作系统的地方就有 Linux 的身影。

2.1 Linux 操作与布局

2.1.1 Linux 开机

由于树莓派并不像计算机一样拥有电源开关, 所以只要为树莓派接上电源, Linux 系统就会开始加载内核并显示很多启动信息, 到“login as:”提示符出现为止, 如下所示:

```
login as: pi
root @172.16.3.4's password:
Linux raspberrypi 3.6.11+ #371 PREEMPT Thu Feb 7 16:31:35 GMT 2013 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.
```

```
Last login: Fri Mar 8 03:10:06 2013 from 172.16.3.119
```

```
pi@raspberrypi ~ $
```

用户可以在登录提示符下输入用户名与密码,树莓派默认的用户名是 pi,密码为 raspberry;输入并确认后可以看到返回了一个以 \$ 符号为结尾的提示符,这说明 pi 用户的权限为普通用户。这时就可以对树莓派进行各种操作了。

如果需要以 root 用户登录,那么需要先通过 pi 用户来修改 root 的密码,如下所示:

```
$ sudo passwd root
```

```
Enter new UNIX password:
```

```
Retype new UNIX password:
```

```
passwd: password updated successfully
```

输入两次密码并确认后,系统会提示密码已经安全更改了,这时就可以使用 root 用户来登录了。与普通用户不同的是,root 用户的提示符是“#”号,以区分普通用户。

如果需要进入树莓派自带的图形界面,也很简单,在命令提示符下输入 startx 并回车,即可开启图形界面。

```
$ startx
```

2.1.2 Linux 关机

Linux 系统的关机是不能直接拔掉电源线的,因为 Linux 系统会将内存作为暂存区,如果直接拔掉电源线会使一些在内存中的数据没有来得及写入 SD 卡中,从而造成数据的丢失或是损坏 SD 卡上的数据,造成系统无法启动。

Linux 系统常用的关机命令有两个,一个是 shutdown 命令,另一个是 halt 命令。无论哪一个命令都可以关闭 Linux,只不过 shutdown 的功能更多,可以支持定时关机、重启等功能。例如,20 min 后关闭系统可以输入以下命令:

```
# shutdown -h 20
```

指定 15:00 重新启动则输入以下命令:

```
# shutdown -r 15:00
```

无论使用哪一种命令来关闭系统都需要有 root 用户的权限,如果用户使用的是像 pi 这样的普通用户,可以使用 sudo 命令来暂时获得 root 权限,如下所示:

```
$ sudo halt
```

2.1.3 Linux 系统文件结构

在 Windows 中,每一个分区都是一个树形结构,因此有多少个分区就有多少个

树形结构,而在 Linux 中则只有一个树形结构,所有的文件、分区都是存在于一个树形结构中。在这个结构中,最上层的是根目录,其他所有的目录、文件、分区都是在根目录下建立的。读者可以通过 `ls -l /` 命令来查看整个根目录的文档,如图 2-1 所示。

```

raspberrypi:~# ls -Fl /
total 84
drwxr-xr-x  2 root root  4096 2012/08/08 22:47:21 bin/
drwxr-xr-x  2 root root 16384 1970/01/01 00:00:00 boot/
drwxr-xr-x 12 root root  3240 2013/03/07 06:51:04 dev/
drwxr-xr-x 99 root root  4096 2013/03/07 07:55:10 etc/
drwxr-xr-x  3 root root  4096 2012/08/08 21:50:30 home/
drwxr-xr-x 13 root root  4096 2012/08/08 22:45:21 lib/
drwx----- 2 root root 16384 2012/08/08 21:42:52 lost+found/
drwxr-xr-x  4 root root  4096 2013/03/07 06:43:59 media/
drwxr-xr-x  4 root root  4096 2013/03/07 07:04:23 mnt/
drwxr-xr-x  3 root root  4096 2012/08/08 21:53:18 opt/
dr-xr-xr-x 81 root root    0 1970/01/01 00:00:00 proc/
drwx----- 3 root root  4096 2013/03/07 07:55:10 root/
drwxr-xr-x 12 root root   420 2013/03/07 07:26:25 run/
drwxr-xr-x  2 root root  4096 2012/08/08 22:47:22 sbin/
drwxr-xr-x  2 root root  4096 2012/06/10 07:08:09 selinux/
drwxr-xr-x  3 root root  4096 2013/03/03 09:21:32 srv/
drwxr-xr-x 12 root root    0 1970/01/01 00:00:04 sys/
drwxrwxrwt  4 root root   80 2013/03/07 07:37:03 tmp/
drwxr-xr-x 10 root root  4096 2012/08/08 21:44:32 usr/
drwxr-xr-x 11 root root  4096 2012/08/08 23:00:24 var/

```

图 2-1 Linux 系统根目录

图 2-1 就是在树莓派上运行的 Debian 系统的根目录情况。在这个根目录下有以下目录:

- ① `/bin/` 用于存放基本系统的可执行文件,如 `ls`、`cp`、`rm`、`mkdir` 等程序。
- ② `/boot/` 引导目录,用于存放系统引导程序如 Linux 内核以及启动配置文件。
- ③ `/dev/` 设备目录。在 Linux 系统中,所有的设备都被视为文件,而在这个目录中存放了所有的设备,例如第一个 SATA 硬盘或 U 盘会被识别为 `sda` 文件,而 SATA 硬盘或 U 盘的第一个分区会被识别为 `sda1` 文件。
- ④ `/etc/` 系统配置文件存放目录。这是一个十分重要的目录,在这个目录中存放了几乎所有的 Linux 系统软件所需要的配置文件。如果读者需要对这个目录中的文本进行修改,那么最好是先将需要修改的文件进行备份,以保证在修改之后还可以回到原来的状态。
- ⑤ `/home/` 用户目录。除了 `root` 用户外,其他所有的使用者的数据都存放在这个目录下。在树莓派的系统中,读者可以看到 `home` 目录中已经有一个 `pi` 的子目录,这个就是 `pi` 用户的默认目录。
- ⑥ `/lib/` 基本系统的动态链接库存放位置。在这个目录中存放了可以维持一个基本系统启动所需要的库文件。如果没有这个目录,系统程序根本就无法工作。
- ⑦ `/lost+found/` 分区系统的目录。系统非正常关机而产生的文件通常都存放在这里,另外由 `fsck` 等程序进行硬盘修复后产生的文件也存放在这里。
- ⑧ `/media/` 媒体目录。许多自动挂载程序都会将如 U 盘这样的外部设置挂

载到这个目录中。

⑨ /mnt/ 传统的外部设备挂载点。在早期的时候,除了系统分区外的其他分区,如U盘等设备,都会被挂载到这个目录下以供用户读写。不过现在已经被/media/取代。

⑩ /opt/ 额外的程序目录,可以用来安装一些额外的程序,另外也有系统管理员将服务器软件安装到这个目录中。

⑪ /proc/ proc是一个虚拟文件系统。这个目录是存放在内存中的,因此不会占用硬盘空间,在Linux系统中系统会将每个process ID赋予一个文件名,系统或用户通过读取这些设备来了解它们的信息。例如,读者想知道当前系统CPU的信息,可以使用cat /proc/cpuinfo命令来查看CPU信息。

⑫ /root/ 超级用户目录。

⑬ /run/ 用于存放系统启动时描述系统信息的文件。这个目录最初是在/var/目录下的,但是现在被提升到根目录下。

⑭ /sbin/ 与bin一样,存放可执行文件,只不过这个目录下的可执行文件需要较高的执行权限。通常这个目录的文件只有root用户有权限执行。

⑮ /selinux/ 用于存放selinux的配置文件,SELinux (Security-Enhanced Linux)是美国国家安全局(NSA)对于强制访问控制的实现,目的是提高Linux的安全等级,但是在树莓派中默认是没有使用的。

⑯ /srv/ 服务器数据位置。这个目录是被单独提取出来的,用于存放如MySQL等程序或数据的位置。如果你的系统没有这些应用就无须担心。

⑰ /sys/ 与/proc目录一样,也是一个虚拟目录,是由内核中的sysfs系统来实现的,其作用与proc有些类似,但除了与proc具有相同的查看和设定内核参数功能之外,还有为Linux统一设备模型作为管理之用。

⑱ /tmp/ 临时目录。由程序所产生的临时文件都会存放在该目录下,不用担心这个目录会占用太多的空间,因为每次系统启动都会清除这个目录的内容;同时这也是系统为数不多的拥有所有用户可读写属性的目录。

⑲ /usr/ Linux系统所安装的程序都是存放在该目录中的,除了可执行程序外还有帮助手册、语言文件等内容。如非必要请不要修改这个目录的内容。

⑳ /var/ 与/usr目录类似,只不过这个目录中的文件经常发生变动,例如/var/log目录存放的是系统服务运行时产生的日志,/var/mail存放的是本机用户的邮件。

在整个Linux系统的目录,用户通常只需要对/etc目录下的配置文件进行修改,而其他的目录则无须作任何操作,系统会很好地处理新安装软件的存放位置。

2.1.4 Linux 的文件类型

Linux的文件类型分为以下几种:

① 普通文件:这种文件是一般性文件,通常是配置文件、文本或是用户数据,在终端显示中默认为白色。

② 目录:在终端的显示中通常以蓝色表示,并且在文件名的最后加了一个“/”来表示目录。

③ 可执行文件:对于这种文件,系统默认显示为绿色,并在文件名的最后加了一个星号“*”来表示这是一个可执行文件。

④ 压缩文件:通常这类文件以 .zip、.gz、.bz2 等扩展名为结尾,文件的显示颜色为红色,在终端中很容易分辨出来。

⑤ 链接文件:在 Linux 系统中为了避免重复文件的出现而设计出的一种文件。例如,在 /lib/ 目录下有一个 lib6. so 文件,而在 /usr/lib 目录下也需要一个 lib6. so 的文件时就可以通过生成一个链接文件来连接到 /lib/lib6. so,从而节省了硬盘空间。链接文件在终端以青色显示,并在文件名称后面加一个“@”符号。

⑥ 设备文件:以黄色表示,这种文件通常只存在于 /dev 目录下。

关于更多的示例可以查看图 2-2。

```
root@raspberrypi:/lib# ls -F
arm-linux-gnueabi/      libnih-dbus.so.1@
cpp@                    libnih-dbus.so.1.0.0
firmware/              libnih.so.1@
init/                  libnih.so.1.0.0
klibc-s06SiFHCdmbhHGtm0ylyHu6vb0.so* libxtables.so.7@
ld-linux-armhf.so.3@   libxtables.so.7.0.0
libip4tc.so.0@         linux.tar
libip4tc.so.0.1.0      lib/
libip6tc.so.0@         modprobe.d/
libip6tc.so.0.1.0      systemd/
libipq.so.0@           terminfo/
libipq.so.0.0.0        udev/
libiptc.so.0@          xtables/
libiptc.so.0.0.0
```

图 2-2 各种 Linux 系统文件标志

2.1.5 常用命令

由于树莓派在第一次启动时并没有开启图形界面,因此对于那些习惯 Windows 的图形界面操作的用户来说,第一次使用树莓派的 Linux 系统时面对命令提示符时往往不知道如何操作。不过没有关系,对于需要图形界面的用户,可以使用 startx 命令来开启图形界面,但是在 Linux 系统中命令行往往会比图形界面更有效率。以最简单的更改文件名为例,用户需要为 100 个文件按大小编号,如果在图形界面中,需要花费半天的时间点鼠标进行修改,而在命令行下,只需要一条命令就可以完成这一工作。因此如果是打算长期使用 Linux,那么学习如何在命令行下操作是一件非常必要的事情。

下面介绍一些 Linux 系统中最基本的操作命令,如目录操作命令、查看磁盘空间命令、解压操作命令等。

1. 目录操作命令

命令名称:cd

作用:切换当前工作目录。cd 命令是最常用的 Linux 命令之一,用于切换用户当前的工作目录。如果需要快速回到登录用户的家目录,可以直接使用不带参数的 cd 命令来返回。

语法:cd 目标目录路径

示例:

cd .. 返回上一层目录

cd /home 进入/home 目录

cd 或 cd ~/ 返回用户目录

命令名称:ls

作用:显示目录与文件。

语法:ls <目标目录路径>

命令名称:mkdir

作用:建立目录。

语法:mkdir 目录名称

命令名称:cp

作用:如果需要复制子目录可以加上-r 参数。

语法:cp 源文件或目录 目标文件或目录

命令名称:mv

作用:命令格式与 cp 类似,都是先指定源文件或目录位置,然后指定目标文件或目录位置。

语法:mv 源文件或目录 目标文件或目录

命令名称:rm

作用:删除目录或文件。使用此命令时默认只删除文件,如果需要删除目录,则需要加上-r 参数。

语法:rm 目录名称

2. 文件操作命令

命令名称:cat

作用:常用的显示文件内容命令。如果文件很长,则可以使用“cat | more”命令来翻页查看。

语法:cat 文件名称

命令名称:ln

作用:生成链接文件,使用格式与 cp 命令类似,默认生成硬链接,如果需要生成软链接,则可以使用-s 参数。

语法:ln 源文件或目录 目标文件或目录

3. 查看磁盘空间命令

命令名称:du

作用:查看目录使用空间,如果需要显示的单位更容易查看,则可以使用-h 参数。

语法:du 要查看目录路径

命令名称:df

作用:查看硬盘使用空间,如果需要显示的单位更容易查看,则可以使用-h 参数。

语法:df

4. 解压操作命令

命令名称:tar

作用:建立或解压 tar 文件,解压使用-xf 参数,如果要建立压缩文件,则可以使用-c 参数。

语法:tar 文件名

示例:

tar xf linux.tar

解压 tar 文件

tar cf linux.tar linux/

将 linux 目录打包成 linux.tar 文件

命令名称:unzip

作用:解压 zip 文件。

语法:unzip zip 文件名

命令名称:bzip2

作用:建立或解压 bz 文件,解压使用-d 参数,如果要建立压缩文件,则可以使用-c 参数。

语法:bzip2 -d 文件名

示例:

bzip2 xf linux.bz2

解压 bz2 文件

bzip2 cf linux.tar.bz2 linux.tar 将 linux.tar 文件打包成 linux.tar.bz2 文件

5. 帮助命令

命令名称:man

作用:查看程序的帮助手册。

语法:man 指令名

命令名称:info

作用:与 man 一样可以查看程序的帮助手册,只不过在查看时可以上下翻页。

语法:info 指令名

2.2 用户与权限管理

由于 Linux 系统是一个可以供多人同时使用的系统,因此为了区别每个用户而产生了用户管理与权限管理机制。这种机制可以很好地区分每一个用户,并针对这些用户进行管理与服务,最大限度地保证用户的数据安全。

2.2.1 用户管理

在树莓派安装完成后,整个系统会有两个账号,一个是默认的系统管理账号“root”,另一个则是普通用户账号“pi”。如果你的系统有更多的使用者需要连接到树莓派上,最好的办法就是为他们建立单独的账号。这样做的好处就是方便管理,同时也可以将每个用户的数据进行独立管理,从而最大限度地保证用户的信息不被外泄。

使用 adduser 命令建立新用户较为简单,只需要在命令行中输入 adduser 加上新增加的用户即可。在建立的过程中,会提示输入新用户的密码以及一些如手机号码、用户全名等在内的个人信息,如果不想输入这些信息也可以直接回车跳过。

建立一个 jonsk 的用户名:

```
# adduser jonsk
Adding user 'jonsk' ...
Adding new group 'jonsk' (1000) ...
Adding new user 'jonsk' (1000) with group 'jonsk' ...
Creating home directory '/home/jonsk' ...
Copying files from '/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for jonsk
Enter the new value, or press ENTER for the default
    Full Name []: jonsk.echo
    Room Number []: 612
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
```

删除,用户使用 deluser 命令即可,该命令默认只是删除用户在系统内的用户记

录,而不会删除/home 目录下的用户目录。系统管理员确认被删除用户已经备份了自己在/home/下的数据内容后,通过 root 账号对这些目录进行手动删除。

```
# deluser jonsk
Removing user 'jonsk' ...
Warning: group 'jonsk' has no more members.
Done.
```

2.2.2 权限管理

在 Linux 系统中,如果输入 `ls -l` 命令则会显示出文档的详细信息。在这些信息中,最重要的就是文件权限信息。由于 Linux 系统是可以供多人同时使用的系统,因此如何区别每个人的数据以及每个人的权限是一个十分重要的问题。下面介绍通过 `ls -l` 命令查看文件权限。如果用户使用 `ls -l` 命令,而看到的文件建立时间与上面不一样,请不要担心,因为为了方便查看,笔者重新定义了时间的显示方式。如果您也想这样显示,可以将下面的命令加入到配置文件中或直接在命令行下输入:

```
export TIME_STYLE = '+ %Y/%m/%d %H:%M:%S'
```

然后使用 `ls -l` 命令来查看文件信息,就和下面的显示一样。

```
# ls / -l
total 60
```

文件属性	文件数	拥有者	所属组	文件大小	建立日期	建立时间	文件名
drwxr-xr-x	2	root	root	4096	1970/01/01	00:11:22	bin
drwxr-xr-x	2	root	root	4096	2012/12/26	05:12:30	boot
drwxr-xr-x	11	root	root	3020	2013/03/10	06:34:06	dev
drwxr-xr-x	57	root	root	4096	2013/03/11	07:26:59	etc
drwxr-xr-x	2	root	root	4096	2013/03/09	02:40:31	home
drwxr-xr-x	11	root	root	4096	2013/03/11	03:42:18	lib
drwxr-xr-x	2	root	root	4096	2013/03/08	03:42:47	media
drwxr-xr-x	22	root	root	4096	2013/02/09	00:55:35	mnt
drwxr-xr-x	2	root	root	4096	2013/03/08	03:42:47	opt
dr-xr-xr-x	73	root	root	0	1970/01/01	00:00:00	proc
drwx-----	3	root	root	4096	2013/03/11	07:27:38	root
drwxr-xr-x	10	root	root	340	1970/01/01	00:21:31	run
drwxr-xr-x	2	root	root	4096	2013/03/08	03:57:51	sbin
drwxr-xr-x	2	root	root	4096	2012/06/20	09:23:29	selinux
drwxr-xr-x	2	root	root	4096	2013/03/08	03:42:47	srv
dr-xr-xr-x	12	root	root	0	1970/01/01	00:00:07	sys
drwxrwxrwt	2	root	root	4096	2013/03/10	13:20:45	tmp

drwxr-xr-x	10	root	root	4096	2013/03/08	03:42:47	usr
drwxr-xr-x	11	root	root	4096	2013/03/08	03:42:47	var

在上面的例子中,显示了 Linux 根目录下的详细文件信息。在这些文件信息中,最重要的就是第一列,它详细描述了文件和目录的权限,而第三与第四列则显示了这个文件和目录属于哪一个用户或组。

Linux 的文件属性可以分为三种:只读(r)、写(w)和可执行(x)。但是上面的文件属性却分为 10 小格,这是因为除了第一格显示目录外,另外三组每组三格分别表示文件所有者权限、同一组内的权限以及其他用户权限:

dlc	rwX	rwX	rwX
文件类型	用户	组	其他用户

第一栏中如果显示 d,则表示这是一个目录;如果是链接文件,则在这里显示 l;如果是设备文件,则显示 c。

第一个 rwx 栏位: -rwx----- 表示文件拥有者所拥有的权限。

第二个 rwx 栏位: ---rwx--- 表示同一工作组内用户权限。

第三个 rwx 栏位: -----rwx 表示其他用户权限。

例如:

-rwx rwx rwx 表示无论哪一个用户都可以对这个文件进行读写与执行。

-rw----- 只有文件拥有者可以对这个文件进行读写,但是没有执行权。

-rw-rw-rw- 表示所有的用户都有读写权。

对于 Linux 来说,如果需要变更这些权限,可使用 chmod 命令,但是在介绍这个命令之前先介绍一下如何在 Linux 上通过英文字母法或者数字法来表示这些权限。

(1) 英文字母法

前面介绍过用户权限被分为 10 小格,除了第一格外,其他都是 3 格一组,共 3 组:第一组 User 用 u 表示,第二组 Group 用 g 表示,第三组 Other 用 o 表示。即:

第一个 rwx 栏位: -rwx----- 属于拥有者权限用 u 表示。

第二个 rwx 栏位: ---rwx--- 属于组使用权限用 g 表示。

第三个 rwx 栏位: -----rwx 属于其他使用者权限用 o 表示。

除了用 u、g、o 表示以外,还可以使用 a 表示所有人,而每一种使用者权限都可以直接使用 r、w、x 来对应表示读、写、可执行,最后再配上“+”或者“-”,将它们进行连接就可以设置用户权限了。如果需要同时设置多个用户的权限可以使用“,”来区分,例如:

```
# ls -lh linux.tar
-rw-r--r-- 1 root root 0 2013/03/11 03:15:58 linux.tar
# chmod u+rw,g-r,o-r linux.tar
# ls -lh linux.tar
```



```
-rw----- 1 root root 0 2013/03/11 03:15:58 linux.tar
```

在上面的例子中,为 linux.tar 这个文件删除了同组和其他用户的读权,并为自己设置读写权。

(2) 数字法

如果觉得英文字母表示法很麻烦,那么可以尝试用数字表示法。在数字表示法中,只读权限用 4 表示,写入权限用 2 表示,可执行权限用 1 表示,即:r=4,w=2,x=1。

如果要为所有的用户设置所有权限,可以使用 777,即“读+写+执行”(4+2+1=7)。为所有用户授予所有权限时要考虑是否要这样做,因为这很可能造成一个安全隐患。

```
# chmod 777 linux.tar
# ls linux.tar
-rwxrwxrwx 1 root root 0 2013/03/14 08:27:57 linux.tar *
```

取消除自己以外的所有用户的可执行权限:

```
# chmod 766 linux.tar
# ls linux.tar
-rwxrw-rw- 1 root root 0 2013/03/14 08:27:57 linux.tar
```

前面介绍了如何管理用户的权限,但是还有一个问题:如何将一个文件的所有权进行移交?这种权限使用 chown 指令来进行,如果用户不想使用 chgrp 命令改变用户组,也可以在 chown 中直接指定用户名与用户组来进行变更。例如,下面的例子是将 root 用户的 booklist.txt 文件变更为 jonsk 用户与 jonsk 用户组:

```
# ls -lh
-rw-r--r-- 1 root root 0 2013/03/14 08:34:20 booklist.txt
# chown jonsk:jonsk booklist.txt
# ls -lh booklist.txt
-rw-r--r-- 1 jonsk jonsk 0 2013/03/14 08:34:20 booklist.txt
```

变更文件所有者是一项非常有用的命令,这也是 Linux 权限管理中一个很重要的内容,但是在每次变更之前须再三确认,以保证数据的安全以及用户的隐私。

2.3 安装/删除软件

Linux 与 Windows 不同的是,它的软件安装基本上没有什么安装界面,自然也就没有让用户选择安装参数等信息了。因此在 Linux 上安装软件通常有两种方法:一种是通过编译好的二进制包安装,另一种则是通过源代码安装。这两种方法中,二进制安装最省时,只要安装好以后软件就可以使用,而源代码安装则较为麻烦,首先需要安装编译工具,然后还要下载代码进行编译。在树莓派上,一般只需要使用系统

自带的软件包管理工具进行二进制安装即可。

2.3.1 Debian 包管理

Debian 的包管理系统是一个非常优秀的包管理系统,通过 aptitude 工具可以方便地查找、安装软件包,并且会记住哪些包是用户已经安装的,哪些是为了满足依赖关系而安装的。在不被已安装包需要的情况下,aptitude 会自动卸载它们。

在使用 aptitude 工具之前有一件事需要做,那就是设置软件仓库位置,以及更新软件包仓库信息(关于软件仓库源的设置可以参考 3.2.6 小节的更新系统),只有更新了信息我们才可以使用最新的软件包。

使用 apt-get 安装 vim 的文档软件包:

```
# apt-get install vim-doc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  vim-doc
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 1659 kB of archives.
After this operation, 9050 kB of additional disk space will be used.
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main vim-doc all 2:7.3.547-7 [1659 kB]
Fetched 1659 kB in 2s (591 kB/s)
Selecting previously unselected package vim-doc.
(Reading database ... 22351 files and directories currently installed.)
Unpacking vim-doc (from .../vim-doc_2%3a7.3.547-7_all.deb) ...
Setting up vim-doc (2:7.3.547-7) ...
```

移除 vim 的文档软件包并清除配置文件:

```
# apt-get remove --purge vim-doc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  vim-doc *
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 9050 kB disk space will be freed.
Do you want to continue [Y/n]? Y
(Reading database ... 22498 files and directories currently installed.)
Removing vim-doc ...
```

apt 常用指令：

apt-get upgrade	更新系统中所有的包到最新版；
apt-get install(软件包名)	安装软件包；
apt-get --reinstall install	重新安装软件包；
apt-get remove(软件包名)	卸载软件包；
apt-get --purge remove	卸载软件包并清除它产生的配置文件；
apt-get clean	清除已下载的软件包文件；
apt-cache(软件包名)	加上不同的子命令和参数的使用可以实现查找、显示软件、包信息及包信赖关系等功能；
apt-cache stats(软件包名)	显示当前系统所有使用的 Debain 数据源的统计信息；
apt-cache search(软件包名)	可以查找相关的软件包；
apt-cache show(软件包名)	可以显示指定软件包的详细信息；
apt-cache depends(软件包名)	可以查找软件包的依赖关系。

使用 apt 程序安装软件的最大好处就是可以通过网络下载最新的软件包，并能自动解决依赖关系。如果是通过手动下载软件包并且想通过 apt 来安装就比较麻烦，不过 Debian 还提供了另一个程序 dpkg。它虽然不能自动解决依赖关系，但是可以支持安装单独的软件包。例如，已经下载了一个软件包，名为 vim-doc，然后通过 dpkg 程序来手动安装它：

```
# dpkg -i vim-doc_2%3a7.3.547-7_all.deb
(Reading database ... 22498 files and directories currently installed.)
Preparing to replace vim-doc 2:7.3.547-7 (using vim-doc_2%3a7.3.547-7_all.deb) ...
Unpacking replacement vim-doc ...
Setting up vim-doc (2:7.3.547-7) ...
```

同样，也可以使用 dpkg 来卸载软件包，如下所示：

```
# dpkg -r vim-doc
(Reading database ... 22498 files and directories currently installed.)
Removing vim-doc ..
```

dpkg 常用参数：

-i	手动安装指定的软件包；
-s	查询已安装的包的详细信息；
-L	查询系统中已安装的软件包所安装的位置；
-S	软件属于哪个软件包；
-I	查询 deb 包的详细信息；
-r	卸载软件包，但保留配置文件；

- P 全部卸载；
- c 列出的内容；
- reconfigure 重新配置。

Debian 的包管理系统是一个自动化程度很高的程序，用户只需要设置好软件仓库的源后就可以方便地安装各种软件，而无需自己再花时间编译安装。

2.3.2 Arch 包管理

包管理器也是一款非常优秀的软件包管理工具，它将二进制包格式和易用的构建系统结合起来，使用户可以方便地通过 Arch 库更新或者由用户来创建软件包。与 Debian 的包管理系统一样，Pacman 包管理器也可以自动解决依赖关系，并通过命令轻松地下载和安装最新的软件包。读者可以使用下面给出的指令来进行更新与安装 Arch 软件包。

Arch 的常用指令：

- pacman -Sy 源信息更新；
- pacman -Syu 软件随源升级；
- pacman -Ss(软件名) 从源中查找软件；
- pacman -Sw(软件名) 从源中下载软件；
- pacman -S(软件名) 从源中安装软件；
- pacman -R(软件名) 删除软件(包括该软件依赖但是未被其他软件依赖的包)；
- pacman -Sc 删除/var/cache/pacman/pkg/* 中未安装包；
- pacman -Scc 删除/var/cache/pacman/pkg/* 中所有包；
- pacman -Q 查询所有已安装软件；
- pacman -Q(软件名) 查询单个软件版本信息；
- pacman -Qi(软件名) 查看单个软件包内容；
- pacman -Sg(软件名) 从源中查询软件组包含软件。

2.4 文本编辑器

目前，在任何 Linux 系统中基本上都会默认安装两种文本编辑器，分别是 nano 与 vi。前者是为那些痛恨 vi 这样的需要输入指令来进行文本编辑的用户准备的，与 vi 不同的是，nano 很像早期 DOS 系统下的 edit.exe，通过快捷键用户可以很容易地使用 nano 来编辑、保存文本。如果读者很少编辑 Linux 系统上的文件，那么可以花一点时间学习一下 nano；如果打算成为如 Linux 系统管理员，或是打算长期使用 Linux 系统，还是选择 vi 吧。虽然刚开始使用很痛苦，但是习惯后真的很好用。

2.4.1 nano

GNU nano 是一个小巧的文本编辑器, nano 提供许多额外的特性, 例如: 交互式的查找和替换、定位到指定的行列、自动缩进、特性切换、国际化支持以及文件名标记完成。用户直接在命令提示符中输入 nano 并回车就可以打开 nano 编辑器, 如果需要打开文件则可以在 nano 后面加上需要打开的文件名即可, 如图 2-3 所示。

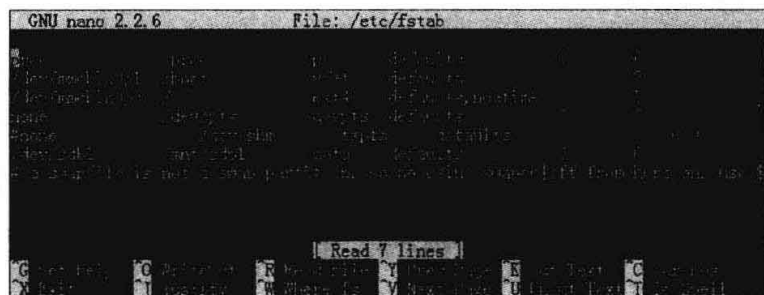


图 2-3 nano 编辑界面

进入 nano 后, 在屏幕下方会出现提示行, 介绍各种操作所需要使用的组合键, 例如:

Ctrl+O 表示可以把文件内容写入。

Ctrl+X 表示可以退出 nano。

对新用户来说, 这两个组合键一定要记住。这也是使用较多的组合键。

Ctrl+W 搜索功能, 按下后输入需要搜索的关键字, 然后回车确定。nano 会搜索到第一个匹配的文本, 如果还要接着向下搜索, 可以用 Alt+W。

Alt+6 复制一整行。

Ctrl+K 剪切一整行。

Ctrl+U 粘贴。

如果需要复制/剪切多行或者一行中的一部分, 先将光标移动到需要复制/剪切的文本的开头, 按 Ctrl+6 (或者 Alt+A) 做标记, 然后移动光标到待复制/剪切的文本末尾, 这时选定的文本会反白, 按 Alt+6 键复制, 按 Ctrl+K 键剪切。如果想在选择过程中取消, 可以按 Ctrl+6 键。

另外, 还有 Ctrl+Y 和 Ctrl+V 键是往前翻页和往后翻页, 如果需要了解更多的组合键, 可以使用 Ctrl+G 键来获得帮助功能。

2.4.2 vi

vi 是所有 Linux/Unix 系统都会提供的屏幕编辑器, 虽然初学者都会觉得它很难使用, 但习惯以后就会觉得没有比这个更方便的编辑器了。在树莓派上提供的 vi 是一个最基本的编辑器, 但是笔者更习惯使用 vim 这一强大的 vi 版本, 如图 2-4 所示。读者可以使用 apt 来安装:

```
# apt - get install vim
```

安装完成后直接使用 vi 命令就可以开始使用了,如果需要调整默认的 vim 配置,可以通过修改/etc/vim/vimrc 文件来完成,如语法加亮、自动对齐等功能。

```
proc                /proc                proc                defaults            0            0
/dev/mmcblk0p1      /boot                vfat                defaults            0            1
/dev/mmcblk0p2      /                    ext4                defaults,noatime    0            1
none                /dev/pts             devpts              defaults            0            0
#none               /dev/shm             tmpfs               defaults            0            0    0 0
/dev/sdb1            /mnt/sdb1            auto                defaults            0            0
# a swapfile is not a swap partition, so no using swapon/off from here on, use
dphys-swapfile swap[on|off] for that

"/etc/fstab" 7 lines, 512 characters
```

图 2-4 vi 编辑界面

1. vi 的基本概念

vi 编辑器有多种模式,最常用的只有三种,分别是命令模式(command mode)、插入模式(insert mode)和底行模式(last line mode)。不过我们经常将这三种模式简化为两个模式:命令模式和插入模式。

(1) 命令模式(command mode)

当用户启动 vi 或者通过 vi 打开某个文件后的默认模式,在这个模式中可以控制屏幕光标的移动,字符、行的删除、移动,或是复制某区段及切换进入其他两种模式。如果用户不太明白自己当前是在什么模式下,可以按 Esc 键回到命令模式。

(2) 插入模式(insert mode)

只有在 insert mode 下,才可以进行文字输入。读者可以在命令行模式下按下小写的 i、a、o 键进入插入模式。进入插入模式后,vi 会在最下面显示—install—字样,表示当前正在插入模式中,读者可以随意增加、删除字符;如果需要退出插入模式,按 Esc 键可退回到命令模式。

(3) 底行模式(last line mode)

进入输入 vi 的各种指令,也可以设置编辑环境、寻找字符串、跳转到指定行号等。进入底行模式,需要在命令行模式中输入“:”字符。同样,如果要退回到命令模式,可以按 Esc 键。

除了以上三种模式外,还有可视模式、选择模式、Ex 模式等,不过对于初学者来说并不需要了解这些模式,只要能使用命令、插入、底行三种模式就可以完成基本的文本编辑工作了。

2. 基本操作

其实在用户了解了如何切换 vi 的各种模式后,就可以尝试通过 vi 来编辑文件了,不过光编辑是没有用的,最终还需要保存并退出 vi。下面介绍 vi 的一些组合键

第2章 Linux 系统基础

以及如何使用 vi 来保存并退出 vi。

首先介绍的就是保存与退出。用户在完成了文本编辑功能后,最重要的操作就是保存文档。这时需要先退回到命令模式,然后按下“:”字符进入命令模式,输入 w 进行保存,如果需要保存并退出 vi 则可以输入:wq。关于保存和退出的命令可以参考下面的指令:

```
:w    保存;
:wq   保存并退出;
:q!   不保存并强行退出;
:x    退出,如果文件更改则保存;
ZZ    退出,如果文件更改则保存(大写的 Z)。
```

在 vi 中操作时有一个很大的优点就是双手无须离开键盘的字母区,只需要通过各种命令的组合就可以完成各种操作。例如,使用 h、j、k、l 可以进行上、下、左、右的移植,使用 Ctrl+f、Ctrl+b 可进行上、下翻一页。其他基本操作如下所示:

```
dd    删除光标所在行;
dw    删除一个字(word);
x     删除当前字符;
X     删除前一个字符;
D     删除当前光标后面的全部字符;
yy    复制一行,此命令前可跟数字,表示复制多行;
p     将粘贴板的内容粘贴到当前行的下面;
P     将粘贴板的内容粘贴到当前行的上面;
/     向下查找;
?     向上查找;
>     块右移;
<     块左移;
y     复制块;
d     删除块;
~     切换块中内容的大小写。
```

事实上,vi 编辑器功能实在是太强大了,它的功能如果全部介绍出来可以单独再写一本书了,而这里只是教会读者使用最基本的功能,因此如果需要更多的帮助,可以参考 vi 提供的内置帮助功能。通过它的帮助及自带的教程,相信一定会让读者满意的。

```
:help      打开帮助首页。
:help cmd  查找 normal mode 命令,比如 help dd。
:help i_cmd 查找 insert mode 命令,比如 help i_Ctrl-y。
:help :cmd  查找 command-line 命令,比如 help :s。
```

:help 'option 查找选项,比如 help 'tabstop。

2.5 使用 USB 存储设备

之前介绍过 Linux 系统是将所有的设备当成文件来进行处理的,因此要使用 USB 存储设备,就必须先知道 USB 存储设备的文件名。在命令提示符下输入 fdisk -l 命令,可查看当前系统的存储设备,如下所示:

```
# fdisk -l

Disk /dev/mmcblk0: 7969 MB, 7969177600 bytes
4 heads, 16 sectors/track, 243200 cylinders, total 15564800 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00014d34

    Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1      8192        122879     57344     c    W95 FAT32 (LBA)
/dev/mmcblk0p2     122880       3788799    1832960    83    Linux

Disk /dev/sda: 1019 MB, 1019215872 bytes
4 heads, 16 sectors/track, 31104 cylinders, total 1990656 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

    Device Boot      Start         End      Blocks   Id  System
/dev/sda1          2048        1990655    994304     83    Linux
```

在上面的例子中,共有两个设备。其中,/dev/mmcblk0 是我们安装了系统的 SD 卡。在这个卡上有一个 FAT32 分区和一个 Linux 分区。另一个设备就是 U 盘,容量为 1 GB,其设备名是/dev/sda。在这个设备上只有一个 Linux 分区:/dev/sda1。至此,通过 fdisk -l 命令知道了 U 盘的设备名称,接下来需要将这个设备加载到/mnt 目录中:

```
# mount /dev/sda1 /mnt
```

现在 U 盘已经被加载到/mnt 目录中,下面通过对/mnt 目录的操作来实现对 U 盘等设备的访问。例如,通过 ls 命令查看/mnt 的内容:

```
# ls /mnt
linux-2.3.6.tar.gz  raspberry.tar
```

如果要拔出 U 盘,须先使用 umount 命令取消对 U 盘的挂载,例如:

```
# umount /mnt
```

如果一切正常,命令执行后会返回到提示符状态,这时就可以拔出 U 盘了。

2.6 时间设置

之所以单独列出时间设置是因为 Linux 系统对时间的要求非常高。例如 GCC 程序,如果时间设置不正确会产生无法编译的问题,而解压程序在时间设置不正确的情况下也会在解压时报错。基于成本的考虑,树莓派上并没有硬件实时时钟系统,因此在每次重新启动系统后都需要通过网络或者手动设置时间,以保证系统时间的正确性。

2.6.1 手动设置系统时间

如果只是查看当前时间,可以使用系统自带的 `date` 命令,而手动设置时间则需要 root 用户权限执行 `date` 命令。例如,设置当前时间为 2013 年 2 月 3 日 15 点 54 分 30 秒:

```
# date -s "15:54:30 2013-2-3"
```

在树莓派上,每次启动都设置一次时间是很麻烦的事,但如果用户的树莓派无法连接到网络上获得时间,那么也只有使用这种方法了。

2.6.2 使用网络设置时间

利用网络对时是最方便的,树莓派自带了网络对时程序 `ntp`,并在每次开机时会自动启动 `ntp` 程序联网对时。默认的对时服务器是由 Debian 提供的 `0.debian.pool.ntp.org` 服务器,如果想要对时程序更快一些,可以修改 `ntp` 的配置文件,将对时服务器设置到国内的时间服务器上,即修改 `/etc/ntp.conf` 文件,将 `0.debian.pool.ntp.org` 替换为 `210.72.145.44` (这是中国国家授时中心服务器的 IP 地址)即可。

第 3 章

使用树莓派

本章主要介绍如何选择树莓派可以使用的操作系统,如何将这些操作系统安装到树莓派上,并且还会介绍一些关于操作系统的设置,以便更好地操作树莓派。

3.1 选择操作系统版本

树莓派除了目前官方网站所提供的四个 Linux 版本(Debian 7、Soft-float Debian 7、Arch、RIS0)外,还有其他的 Linux 系统可提供对树莓派的支持,而这些系统都有自己的设计目标。例如,Xbian 和 Raspbmc 是专为视频播放而设计的,openwrt 则是为方便将树莓派配置成无线路由而移植过来的,Puppy 是为轻量级桌面而设计的。因此,读者需要首先确认自己使用树莓派做什么,然后才能决定使用哪一个系统。当然如果有足够多的 SD 卡,可以在每一张卡上安装一个系统,需要哪个系统时可以直接换卡切换。

3.1.1 Debian Linux

Debian 系统是一个老牌的 Linux 发行版本。它支持非常多的平台,从 X86、AMD64、MIPS 到 ARM、PowerPC 等,几乎涵盖了目前所有的计算机平台。由于 Debian 系统的大部分基础工作都来自于 GNU 工程(GNU 目标是设计一套免费好用的系统),因此 Debian 又被称为 GNU/Linux。现在 Debian 系统附带了超过 29 000 个软件包,其中移植到树莓派上可用的软件包约有 9 534 个,已经足够满足用户在大部分工作方面的需求。

目前,Debian 有三个发行版本:稳定版、测试版和开发版,但是对树莓派的用户来说,并不需要考虑太多,因为官方提供的就是最新的测试版 Debian 7.0(代号:wheezy),读者可以直接在树莓派的官方网站(<http://www.raspberrypi.org/downloads>)上下载。在这个版本中,集成了超轻量级的桌面系统 LXDE、Midori 浏览器,以及开发包、文档等软件,可以满足用户对树莓派的日常使用和系统开发的需要。图 3-1 就是 Debian 的默认桌面。

在树莓派网站上提供了两个版本的 Debian 7 的下载。一个是 Debian 7,这个版本能完美地支持 armhf,能充分利用树莓派的 FPU 来工作。Soft-float Debian 7 则

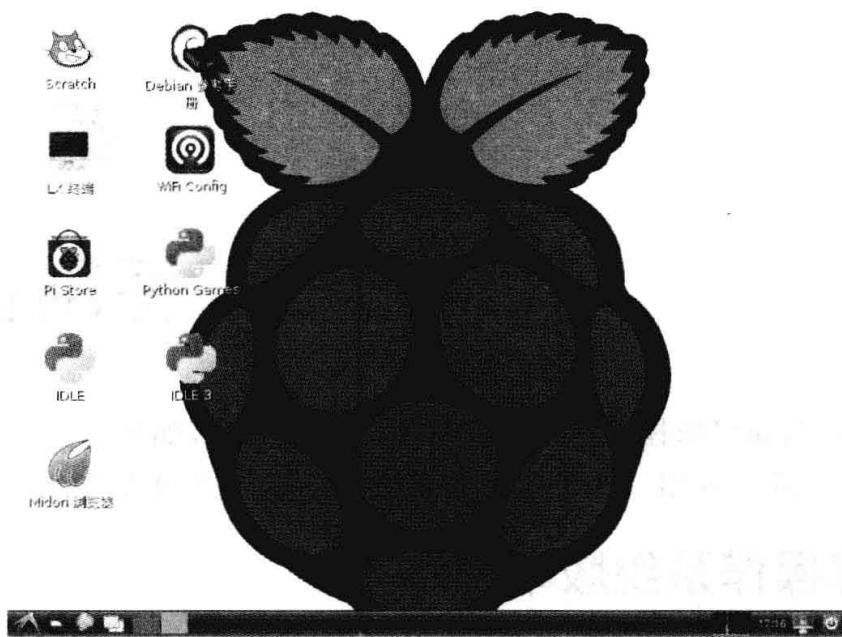


图 3-1 Debian 默认的 LXDE 桌面

是为 Java 准备的,由于 Oracle JVM 只能支持到 ARM7,而树莓派的 CPU ARM6 是不被支持的,因此只能使用 Soft-float 版本的 Debian 7。

3.1.2 Arch Linux

Arch Linux 是一个采用滚动升级模式的通用 GNU/Linux 发行版,与 Debian、gentoo 等发行版本最大的不同是它更注重结构精简与简洁,只需要一次安装后通过滚动升级就可以维护系统的升级工作。另外,它的包管理系统借鉴 Debian 与 gentoo 的优点,所使用的软件包安装程序 pacman 能自动处理软件包的依赖关系,而保留类 ports 系统,可以让用户最大程度地对软件进行定制。Arch Linux 最大的缺点也是太过自由与精简,由于 Arch 安装完成后只有一个基本的系统,因此用户除了需要按照自己的需求进行其他软件的安装外,还需要手动定制整个系统的环境。因为官方没有提供图形界面的配置工具,因此大部分的配置都需要通过命令行或文本编辑器来完成。

读者可以在树莓派的官方网站 <http://www.raspberrypi.org/downloads> 下载到 Arch Linux 的最新版本。

3.1.3 Xbian

Xbian 是一个专为树莓派优化设计的媒体中心,是 XBMC 与 Raspbian 结合的一个发行版本。它的任务就是完成媒体中心的功能,因为树莓派的低性能,针对性地做了一些优化,例如直接使用 HDMI 输出、使用硬件解码、使用遥控器控制等。在迷你播放器方面,它除了支持本地文件播放以外,还支持网络视频点播、图片查看等功能。除此之外,由于是专门为媒体中心设计的,所以支持更多的多媒体文件格式、外置字

幕等功能。

3.1.4 Raspbmc

Raspbmc 与 Xbian 一样,也是一个专为树莓派设计的媒体中心。它与 Xbian 的区别在于,Xbian 的定制与优化更深,占用 SD 卡空间更少,而 Raspbmc 的使用人数多一些,网络上的资源多一些。但是,无论是 Raspbmc 还是 Xbian,本质上都是使用 XBMC 软件进行媒体播放的。使用 Raspbmc 的另一个好处是,它可以自动更新,并且这种更新还受到树莓派官方的支持。

3.1.5 其他系统

除了前面介绍的 Linux 系统外,目前在树莓派上还可以安装其他的操作系统,但是这些操作系统很多都是以测试并证明 Linux 的可移植性而存在的,或者是因为缺少开发人员维护,软件数量有限而并不太好使用。对于这些系统,其本身并没有太多的使用价值,不过将来也许使用的人多了,可能会吸引更多的人来开发它们。

1. Puppy

Puppy,超轻量的 Linux 发行版,由于所有软件只带有基本程序、库等必需的文件,因此,它体积小巧,结构简单,启动速度超快,通常一个很小的 U 盘就能安装下整个系统,并带有丰富的驱动程序和图形界面程序,能让用户完成网络浏览,文档、图片查看与处理等功能。如果安装中文包,还可以方便地实现系统的中文化工作,其桌面如图 3-2 所示。

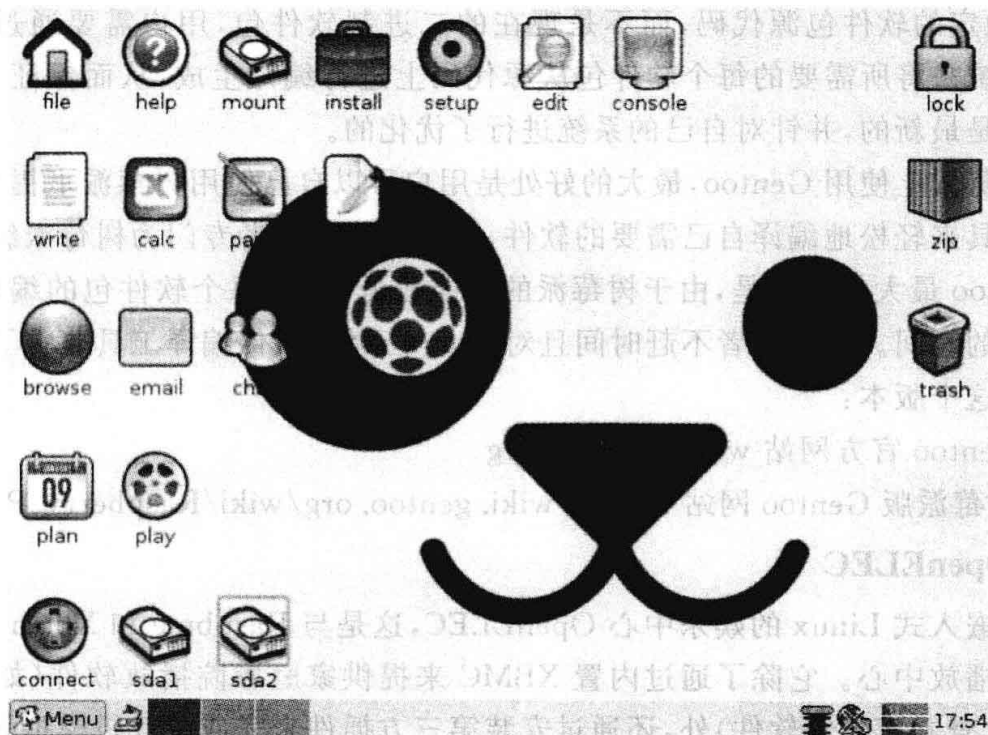


图 3-2 Puppy 桌面

读者可以通过网址 <http://bkhome.org/blog/?viewDetailed=02910> 找到 Puppy 的树莓派版本的安装文档。

如果需要对 Puppy 进行中文化,则可以参考:

- Puppy Linux 中文项目(<http://code.google.com/p/chinesepuppylinux/>)
- Puppy Linux 中文项目论坛(<http://e-pup.5d6d.net/>)

2. Openwrt

专为路由器设计的一款嵌入式 Linux 系统。最初,它是基于 Linksys 提供的 GPL 源代码而建立的,但后来它几乎重写了所有的代码,并使用 Linux 2.4.30 作为工作核心,加入了大量的工具,并保证文件系统的可写入。这使得 Openwrt 成为一个 mini 的运行 Linux 系统的计算机。现在,Openwrt 已经发展到可以支持 ARM、PowerPC 或 MIPS 在内的多种平台,所附带的 3 000 多个软件包可以完成大部分用户的需求。

目前,在树莓派上运行的 Openwrt 并未获 Openwrt 官方的支持,而是个人行为,其目的在于让用户能方便地将树莓派变成一个 mini 无线路由器,而无须进行各种复杂的设置。但是,目前看来,整个项目还需要更多的时间来完成,读者可以在网站(<https://code.google.com/p/raspberrypi-openwrt/>)上查看它的信息。

3. Gentoo

Gentoo,非常有名的基于 Portage 包管理系统的 Linux 发行版本,与 Debian 和 Redhat 等系统使用的包管理系统不同。Gentoo 使用的包管理系统默认提供的是最新的、最稳定的软件包源代码,而不是现在的二进制软件包,用户需要通过 Portage 包管理系统来将所需要的每个软件包从源代码上进行编译生成,从而保证用户所使用的系统是最新的,并针对自己的系统进行了优化的。

在树莓派上使用 Gentoo,最大的好处是用户可以自己使用树莓派上提供的完整的编译工具来轻松地编译自己需要的软件,无须等待开发者专门为树莓派编译移植。使用 Gentoo 最大的缺点是,由于树莓派的性能较低,几乎每个软件包的编译都需要花费很长的时间。如果读者不赶时间且对 Linux 系统、软件编译工具非常了解,可以尝试以下这个版本:

- Gentoo 官方网站 www.gentoo.org
- 树莓派版 Gentoo 网站 http://wiki.gentoo.org/wiki/Raspberry_Pi

4. OpenELEC

开放嵌入式 Linux 的娱乐中心 OpenELEC,这是与 Raspbmc 和 Xbian 类似的一个多媒体播放中心。它除了通过内置 XBMC 来提供家庭影院播放软件(如 DVD 播放器、图片查看工具等软件)外,还通过安装第三方插件来完成游戏机模拟等功能,并且还能自动更新整个系统,无需用户手动操作,如图 3-3 所示。读者可以在 OpenELEC 官方网站 <http://openelec.tv/> 上下载它的最新版本。

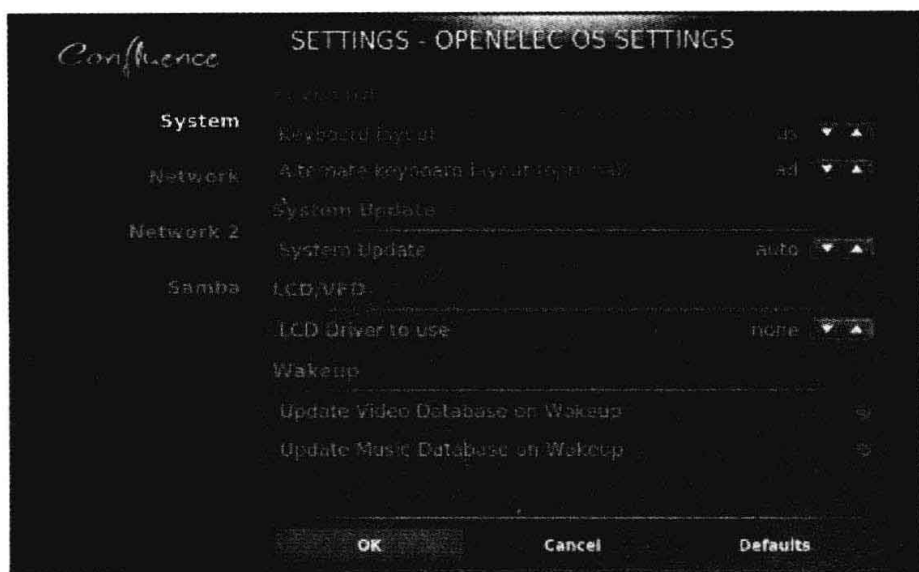


图 3-3 OpenELEC 设置界面

目前,OpenELEC 的稳定版本是 2.99,而最新的测试版 3.0 则已经进入了 RC 状态。也许本书上市时读者就可以使用稳定的 3.0 版本了,读者可以在 <http://openelec.tv/get-openelec> 网站上下载到 OpenELEC 的最新版本。

3.2 安装操作系统到 SD 卡上

3.2.1 校验镜像文件

在安装操作系统镜像到 SD 卡之前,还有一件事要做,就是校验 SD 卡的完整性。目前无论是哪一个树莓派的操作系统,都会在下载页面上提供镜像的校验码。例如由 Raspberry Pi 官网提供的四种 Linux 发行版本,都会在下载链接的下面提供一个 SHA-1 的校验码,如图 3-4 所示。

读者可以通过 SHA-1 哈希值对下载的镜像文件进行校验:

在 Linux 下校验 SHA-1 哈希值时需要使用 `shasum` 程序,它是由 `coreutils` 软件包提供的,`shasum` 也是目前每一个 Linux 版本都默认提供的程序。

```
# shasum archlinux-hf-2012-09-18.zip
```

```
56e043f3c629f6c408759bcc157d2d635ce0b6af archlinux-hf-2012-09-18.zip
```

校验出的 SHA 为 56e043f3c629f6c408759bcc157d2d635ce0b6af,与官网提供的 SHA 校验码一样。如果校验码与网站提供的不一样,则表明文件有错误,需要重新下载镜像文件。

在 Windows 下则需要下载专用的校验程序,目前 Windows 校验 SHA-1 哈希值的程序有很多,读者可以挑选自己喜欢的校验程序,如:

Arch Linux ARM

Arch Linux ARM is based on Arch Linux, which aims for simplicity and full control to the end user. Note that this distribution may not be suitable for beginners. The latest version of this image **uses the hard-float ABI**, and boots to a command prompt in around ten seconds.

Torrent	archlinux-hf-2012-09-18.zip.torrent
Direct download	archlinux-hf-2012-09-18.zip
SHA-1	56e043f3c629f6c408759bcc157d2d635ce0b6af
Default login	Username: root Password: root

图 3-4 Arch Linux SHA-1 校验码

HashMyFiles <http://dl.pconline.com.cn/download/90027.html>

Hash <http://keir.net/software.html>

fciv(微软文件校验和完整性验证)

[http://www.microsoft.com/en-us/download/details.aspx?](http://www.microsoft.com/en-us/download/details.aspx?id=11533)

id=11533

HashCalc <http://www.onlinedown.net/soft/38329.htm>

推荐使用 Hash 1.04, 如图 3-5 所示, 将需要校验的文件拖到软件界面上就可以了, 读取一次文件就可以完成 MD5、SHA1 与 CRC32 的校验。

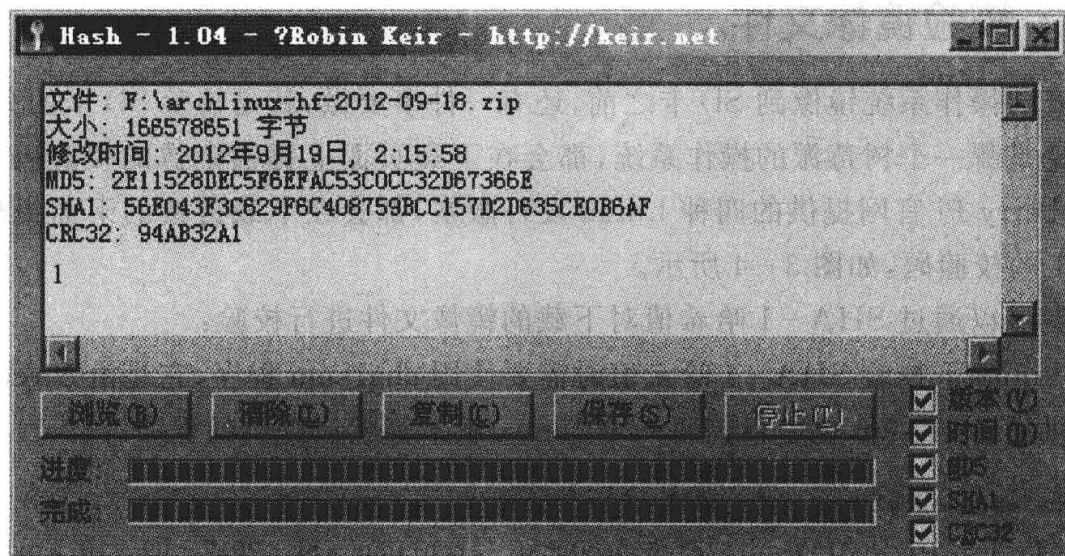


图 3-5 Hash 1.04 校验程序

3.2.2 安装操作系统

Linux 下安装镜像文件需要使用到 dd 工具。这个工具是默认提供的, 而解压工

具 unzip 则需要另外安装。如果用户的系统中没有 unzip, 可以使用 apt - get install unzip 命令进行安装。

对下载的镜像文件进行解压, 如下所示:

```
# unzip 2012 - 12 - 16 - wheezy - raspbian. zip
Archive: 2012 - 12 - 16 - wheezy - raspbian. zip
  inflating: 2012 - 12 - 16 - wheezy - raspbian. img
```

解压完成后会得到一个 2G 大小的名为 2012-12-16-wheezy-raspbian. img 的文件。这个就是我们需要写入 SD 卡中的镜像文件。将 SD 卡放入到 USB 读卡器中, 并插到计算机上, 系统会显示检测到的硬盘设备, 这时就可以使用 fdisk 来查看 SD 卡的设备名称:

```
# fdisk -l

Disk /dev/mmcblk0: 7969 MB, 7969177600 bytes
4 heads, 16 sectors/track, 243200 cylinders, total 15564800 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00017b69

Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1    8192       122879       57344    c   W95 FAT32 (LBA)
/dev/mmcblk0p2   122880     15564799      7720960  83   Linux

Disk /dev/sda: 3904 MB, 3904897024 bytes
4 heads, 32 sectors/track, 59584 cylinders, total 7626752 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0001f327

Device Boot      Start         End      Blocks   Id  System
/dev/sda1        4096       147455       71680    c   W95 FAT32 (LBA)
```

结果显示 SD 卡的设备名称是 /dev/sda, 在这个设备上有一个分区名为 sda1。当确认了 SD 卡的设备名称后就可以使用 dd 工具来写入镜像文件, 使用下面的命令:

```
# dd? bs = 1M if = 2012 - 12 - 16 - wheezy - raspbian. img of = /dev/sda
记录了 1850 + 0 的读入
记录了 1850 + 0 的写出
1939865600 字节(1.9 GB)已复制, 411.346 秒, 4.7 MB/秒
```

等上几分钟后, 如果显示上面类似的信息则表示安装成功; 如果给出的信息不

同,则有可能是 SD 卡出现了问题,读者可能要换一块 SD 卡试试。

Windows 下安装镜像文件到 SD 卡上也需要专用的工具,树莓派推荐的工具为 Win32 Disk Imager。这是一个镜像读写工具,可以将 SD 卡或 U 盘的内容写入到一个镜像文件中,也可以将指定的镜像文件写入到 SD 卡或 U 盘中。读者可以在树莓派的下载页面中找到它:

① 运行 Win32 Disk Imager 后,通过蓝色的打开图标选择要写入的镜像文件,在设备(Device)栏中选择 SD 卡所在的分区,然后依次单击 Write 和 Yes,即可对 SD 卡进行镜像写入,如图 3-6 所示。

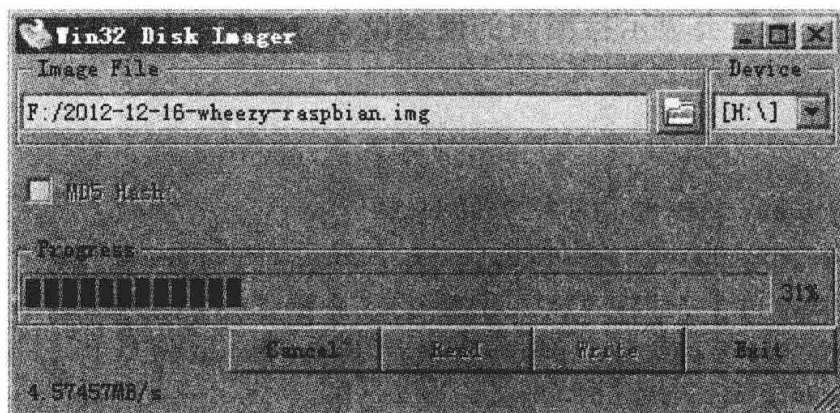


图 3-6 使用 Win32 Disk Imager 安装镜像

② 等上一会儿,在镜像写入完成后会弹出提示框,显示 Write Successful,表示镜像安全写入。这时读者可以弹出 SD 卡,将它插到树莓派上进行系统引导了。

除了 Win32 Disk Imager 以外,还有很多镜像写入工具,比较有名的就是 USB Image Tool。它比 Win32 Disk Imager 更先进的地方是,Win32 Disk Imager 通过盘符确认 SD 卡的位置,而 USB Image Tool 则是通过设备来进行确认的。因此,无论 SD 卡被写入了什么数据,都可以通过 USB Image Tool 写回来。

USB Image Tool 下载地址是 <http://dl.pconline.com.cn/download/58458.html>。

3.2.3 安装操作系统到 U 盘

由于树莓派所有的系统文件和虚拟内存(SWAP)都是在 SD 卡上,因此 SD 卡的速度对性能的影响可能会很大。从目前的情况来看,SD 卡的速度越快,系统读写的速度也就越快。如果使用的是高速 SD 卡,最直接的效果就是用户在运行程序时速度有明显的提升。但是因为 SD 卡的兼容性问题,很多标称 Class 10 规范的卡在树莓派上经常只能获得低于 20 MB/s 的读取速度,而写入速度则更慢。因此如果想要获得更快的速度,可以考虑将系统文件安装到 U 盘上使用。目前 U 盘的读写速度通常都可以达到 20 MB/s 左右,这样很容易获得较高的运行速度,而且 U 盘极少存在兼容性的问题,价格便宜,不容易损坏。

由于树莓派默认是通过 SD 卡上的系统 FAT 分区来进行引导的,因此即便将整

个系统安装到 U 盘上,也还是需要通过一个 SD 卡来引导树莓派启动的;所以 SD 卡仍然是必需的,但用户可以使用一个较小的 SD 卡(SD 卡容量至少有 100 MB)作为启动盘。

1. 安装系统到 U 盘

安装系统到 U 盘有两种方法:一种是直接使用镜像写入工具,将系统镜像写入到 U 盘中;另一种方法则是在 Linux 系统下,重新为 U 盘分区并格式化,然后再将镜像内的文件系统复制到 U 盘中。这两种方法的区别在于第二种方法可以多使用约 60 MB 的空间。

2. 安装引导程序到 SD 卡

首先格式化 SD 卡,然后在 http://www.kuaipan.cn/file/id_5787571510593695.htm 上下载 SD 卡的引导分区镜像文件,然后使用 Win32 Disk Imager 之类的工具将镜像文件写入 SD 卡中即可。

3. 修改配置文件,使用 U 盘进行引导

在 SD 卡上有一个名为 cmdline.txt 的文件,其内容如下:

```
dwc_otg.lpm_enable = 0 console = ttyAMA0,115200 kgdboc = ttyAMA0,115200 console = tty1  
root = /dev/sda2 rootfstype = ext4 elevator = deadline rootwait
```

这个文件的作用就是为系统指定引导参数,其中最重要的就是通过 root 参数指定系统根分区的位置。在默认的配置中,root 的位置是/dev/mmcblk0p2,即 SD 卡的第二个分区,如果用户需要使用 U 盘启动则需要修改 root=/dev/mmcblk0p2 为 root=/dev/sda2。

修改完成后就可以将 SD 卡与 U 盘插入树莓派中,重新启动了。如果一切顺利,用户就可以使用 U 盘来启动系统;如果启动失败则有可能是系统盘位置出错,读者可以将/dev/sda2 修改成/dev/sdb2,或者是/dev/sdc2、/dev/sdd2,再试试(由于设备的兼容性问题,只有很少的情况下才会出现这种情况)。

如果读者想追求更完美的方式,可以接着修改/etc 目录下的 fstab,将/dev/mmcblk0p2 修改为/dev/sda2。这样用户在使用 df 或者是 mount 命令时,显示的根分区就是 U 盘上的分区。

proc	/proc	proc	defaults	0	0
/dev/mmcblk0p1	/boot	vfat	defaults	0	2
/dev/sda2	/	ext4	defaults,noatime	0	1

最终的情况可以参考图 3-7。

修改 fstab 重启后,df 与 mount 的显示加载的根文件系统虽然为/dev/目录下的 root 设备,但是通过查看/dev/root 文件可以看出它是链接到 sda2 设备上的。


```

root@raspberrypi ~# cat /boot/cmdline.txt
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=
ttyL root=/dev/sda2 rootfstype=ext4 elevator=deadline rootwait
root@raspberrypi ~# cat /etc/fstab
proc /proc defaults 0 0
/dev/mmcblk0p1 /boot vfat defaults 0 0
/dev/sda2 / ext4 defaults,noatime 0 1
root@raspberrypi ~# df -h
文件系统 容量 已用 可用 已用% 挂载点
rootfs 3.6G 2.6G 842M 76% /
/dev/root 3.6G 2.6G 842M 76% /
devtmpfs 243M 0 243M 0% /dev
tmpfs 42M 204K 42M 1% /run
tmpfs 5.0M 0 5.0M 0% /run/lock
tmpfs 33M 0 33M 0% /run/shm
/dev/mmcblk0p1 56M 17M 40M 30% /boot
root@raspberrypi ~# mount
/dev/root on / type ext4 (rw,noatime,user_xattr,barrier=1,data=ordered)
devtmpfs on /dev type devtmpfs (rw,relatime,size=243320k,nr_inodes=62205,mode=755)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=43730k,mode=755)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run/shm type tmpfs (rw,nosuid,nodev,noexec,relatime,size=29540k)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620)
/dev/mmcblk0p1 on /boot type vfat (rw,relatime,fsck=0,uid=0,gid=0,codepage=cp437,iocharset=ascii,shortname=mixed,errors=remount-ro)
root@raspberrypi ~# ls /dev/root -lh
lrwxrwxrwx 1 root root 4 1月 30 14:35 /dev/root -> sda2
root@raspberrypi ~#

```

图 3-7 安装到 U 盘后的各种参数

3.2.4 红派所使用的系统

对于一些红派的用户来说,在使用旧的系统时可能会显示电源指示灯一直亮,但是 ACT 指示灯只闪烁一下后就没有响应的情况了,而系统也无法启动。这是由于之前树莓派均采用了三星 DDR2 内存。但是由于三星内存供货不足,因此树莓派基金会兼容使用了 Hynix 的 DDR2 内存(以后在英国生产的树莓派也可能会使用 Hynix 的 DDR2 内存)。对于这些使用 Hynix 的 DDR2 内存的树莓派主板来说,需要更新到最新的系统或是 Firmware,才能正常工作。目前国内生产的红派由于使用的是 Hynix 的 DDR2 内存,而不是原先的三星 DDR2 内存,因此需要使用最新的系统。

目前,树莓派官方网站只有 Raspbian“wheezy”和 Arch Linux ARM 两种系统的更新以支持 Hynix 内存,用户可以直接到 <http://www.raspberrypi.org/downloads> 页面上下载 2013 年 2 月以后释放出来的系统镜像文件。如果需要使用其他版本的操作系统则需要做一点点修改,首先在 <http://pan.baidu.com/share/link?shareid=397442&uk=1929742841> 上下载 RPi firmware for Hynix or Nanya LPDDR2.rar 文件,然后将压缩后获得的三个文件 (bootcode.bin、fixup.dat、start.elf) 替换 SD 卡中 FAT16 分区中的同名文件即可。

事实上,无论是 Hynix 还是三星的 DDR2 内存,在树莓派上的表现是一样的,并不会因为使用了不同厂商的内存而导致性能出现差异或不稳定的情况,而系统更新以及 Firmware 的问题会随着时间的推移而解决。当用户看到本书的时候,无论是哪一种系统都应该可以完美地支持 Hynix 和三星 DDR2 内存了。

3.2.5 还原 SD 卡

由于 Linux 系统使用的分区与 Windows 分区是不同的,而微软公司似乎也没有打算支持 Linux 系统的分区格式,因此读者如果想要将在树莓派上使用的 SD 卡重新在 Windows 系统上使用,就需要重新对 SD 卡进行分区处理,并格式化成 Windows 能识别的 FAT32、exFAT 或者 NTFS 格式。对于 SD 卡还原最方便的还是使用 Linux 系统,通过内置的程序就可以完成这项工作:

① 对 SD 卡进行重新分区,将原有的 FAT32 分区与 Linux 分区删除,然后将 SD 卡重新分为一个主分区即可,输入:

```
# fdisk /dev/sda
```

② 输入 p 查看当前 SD 卡的分区情况,以确认 sda 是否是 SD 卡。

```
Command (m for help): p
```

```
Disk /dev/sda: 3904 MB, 3904897024 bytes
```

```
121 heads, 62 sectors/track, 1016 cylinders, total 7626752 sectors
```

```
Units = sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk identifier: 0x00017b69
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1		8192	122879	57344	c	W95 FAT32 (LBA)
/dev/sda2		122880	3788799	1832960	83	Linux

可以看出,在 sda 设备上有两个分区:一个 sda1,是 FAT32 分区引导系统用的;另一个 sda2 分区,是 Linux 分区用于存放系统使用的。

③ 确认分区情况后开始删除分区,首先输入 d,回车,然后输入要删除的名称号码。由于有两个分区,因此输入 1 表示删除第一个分区,回车后再次输入 d,系统会自动删除剩下的第二个分区。

```
Command (m for help): d
```

```
Partition number (1-4): 1
```

```
Command (m for help): d
```

```
Selected partition 2
```

④ 输入 p,回车,查看磁盘分区情况:

```
Command (m for help): p
```

```
Disk /dev/sda: 3904 MB, 3904897024 bytes
```

```
121 heads, 62 sectors/track, 1016 cylinders, total 7626752 sectors
```

```
Units = sectors of 1 * 512 = 512 bytes
```

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00017b69

Device	Boot	Start	End	Blocks	Id	System
--------	------	-------	-----	--------	----	--------

可以看到之前的分区都被删除了,整个磁盘很干净。

⑤ 重新建立 SD 卡分区,在命令提示符中输入 n,可建立新的分区:

Command (m for help): n

Partition type:

p primary (0 primary, 0 extended, 4 free)

e extended

系统会提示建立的分区类型,输入 p 以主分区形式建立分区,然后连续回车三次确认使用默认的分區号、起始位置。

Select (default p): p

Partition number (1-4, default 1): 1

First sector (2048 - 7626751, default 2048):

Using default value 2048

Last sector, + sectors or + size{K,M,G} (2048 - 7626751, default 7626751):

Using default value 7626751

⑥ 由于 fdisk 程序默认建立的分区都是 Linux 格式,因此需要修改分区格式为 Windows 所支持的格式。输入 t 确认后再输入 c,表示将新的分区设置为 FAT32 格式。

Command (m for help): t

Selected partition 1

Hex code (type L to list codes): c

Changed system type of partition 1 to c (W95 FAT32 (LBA))

⑦ 再次输入 p 查看分区,这时可以看到已经建立了一个使用 SD 卡全部空间的 FAT32 分区。

Command (m for help): p

Disk /dev/sda: 3904 MB, 3904897024 bytes

121 heads, 62 sectors/track, 1016 cylinders, total 7626752 sectors

Units = sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00017b69

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1		2048	7626751	3812352	c	W95 FAT32 (LBA)

⑧ 在“Command (m for help):”提示符下使用 `wq` 命令,保存修改并退出 `fdisk` 程序。

```
Command (m for help): wq
```

```
The partition table has been altered!
```

```
Calling ioctl() to re-read partition table.
```

```
WARNING: If you have created or modified any DOS 6.x
partitions, please see the fdisk manual page for additional
information.
```

```
Syncing disks.
```

⑨ 保存退出后,可以直接将 SD 卡拿到安装有 Windows 系统的计算机上进行格式化,即可还原 SD 卡到 Windows 系统下允许的状态。

如果是在 Windows 系统下对 SD 卡进行还原,可能要麻烦一些。由于 Windows 自带的磁盘管理工具并不支持对 SD 卡或 U 盘之类的设备进行分区,因此需要使用一些第三方的工具。

U 大师 <http://www.udashi.com/>

硬盘分区魔术师 <http://www.onlinedown.net/soft/261883.htm>

DiskGenius <http://www.onlinedown.net/soft/9570.htm>

推荐使用 U 大师,这个由数码之家出品的 U 盘引导工具自带了 U 盘空间回收功能,只需要插上 U 盘,启动 U 大师并点击还原 U 盘,十多秒后即可重新在 Windows 系统下使用 SD 卡了,如图 3-8 所示。

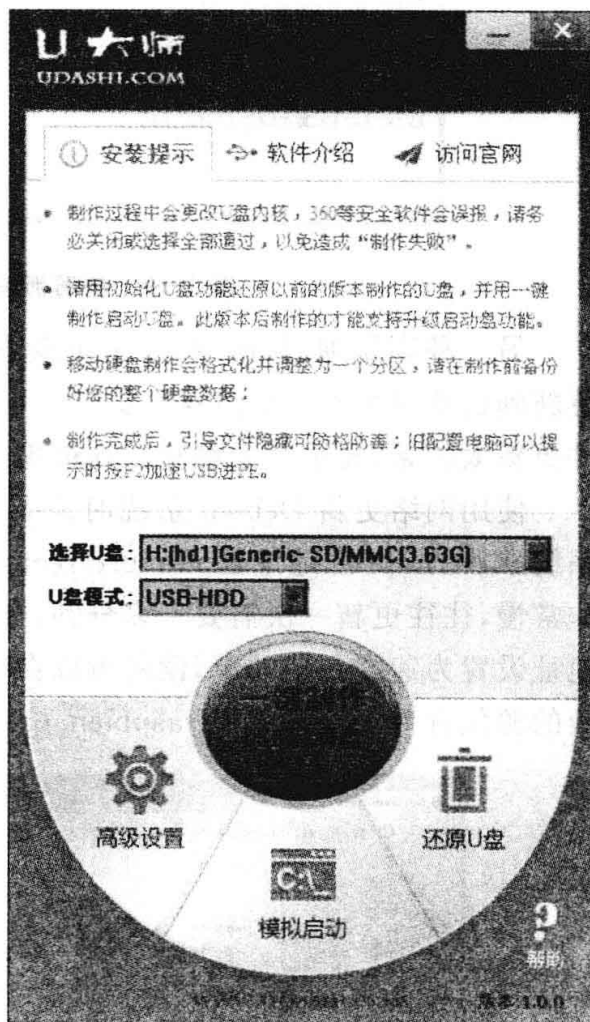


图 3-8 U 大师界面

3.2.6 更新系统

树莓派提供的 Debian 系统目前有两种更新方式。一种方式是通过树莓派官方网站上提供的镜像来更新系统。虽然每次更新都需要重新安装所需要的软件并且设置所有的配置,但为了方便,那些不能使用网络更新的用户,目前的树莓派官方网站几乎每个月都会放出一个 Debian 的安装包。读者可以通过 <http://downloads.raspberrypi.org/images/raspbian/> 网站下载最新的安装镜像。

第3章 使用树莓派

图 3-9 为树莓派官方网站提供的镜像文件,在这个目录中包含所有的历史镜像与最新的镜像文件。

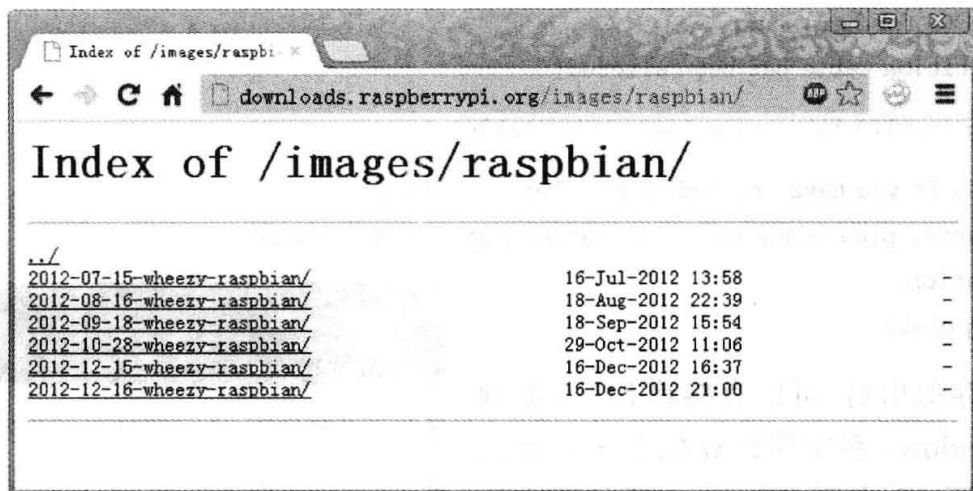


图 3-9 树莓派官方网站提交的镜像文件

另一种则是通过 apt 程序来更新系统。使用 apt 程序,用户可以轻松地将系统更新到最新的版本。虽然这种更新非常容易,但是它非常依赖网络,而且树莓派的软件更新太频繁,几乎每天都有软件被更新。

使用网络更新 Debian 系统时需要使用到 Debian 软件包服务器。由于 Debian 系统默认的 apt 源地址是 mirrordirector.raspbian.org,而该网站在国内访问的速度非常慢,往往更新一次需要十多分钟。因此为了提高同步速度,需要将默认的 apt 源地址设置为速度更快的源,读者可以在树莓派的网站上查看目前提供 Debian 树莓派的源位置 <http://www.raspbian.org/RaspbianMirrors>,如图 3-10 所示。

Region	Country	Organization	URL
America		University of Mato Grosso do Sul UFMS	rsync://200.129.202.132/Raspbian
Australia	Australia	AARNet	@http://mirror.aarnet.edu.au/pub/raspbian/raspbian
Australia	Australia	Over The Wire	@http://mirror.overtewire.com.au/pub/raspbian/raspbian
Australia	Australia	SkyMesh Pty Ltd	@http://mirror.skymesh.net.au/pub/raspbian/raspbian
Australia	Australia	Internode	(http ftp)://mirror.internode.on.net/pub/raspbian/raspbian/
Asia	Singapore	NUS School of Computing SigLabs	@http://mirror.nus.edu.sg/raspbian/raspbian
Asia	China	Tsinghua University Network Administrators	@http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/
Asia	South Korea	dnetwork	@http://mirror.dnwinet.kr/raspbian/raspbian/
Asia	South Korea	Neowiz Games corp	@http://ftp.neowiz.com/raspbian/raspbian
Asia	South Korea	KAIST	(http ftp rsync)://ftp.kaist.ac.kr/raspbian/
Asia	Indonesia	DATAUTAMA-NET-ID	(http rsync)://kartolo.sby.datautama.net.id/raspbian/raspbian

图 3-10 Debian 的源镜像地址

目前,只有清华大学提供了可供树莓派使用的镜像。如果用户在使用默认镜像

访问时感觉速度较慢,可以使用清华大学提供的镜像网站来提高更新速度。

编辑/etc/apt/sources.list 文件,将文件中的 `http://mirrordirector.raspbian.org/raspbian/` 部分更换为 `http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/` 即可。

```
# cat /etc/apt/sources.list
deb http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/ wheezy main contrib non-free rpi
```

修改完成后保存文件并退出,之后需要通过 apt 程序重新更新软件包仓库信息:

```
# apt - get update
命中 http://mirrors.tuna.tsinghua.edu.cn wheezy InRelease
命中 http://mirrors.tuna.tsinghua.edu.cn wheezy/main armhf Packages
命中 http://mirrors.tuna.tsinghua.edu.cn wheezy/contrib armhf Packages
命中 http://mirrors.tuna.tsinghua.edu.cn wheezy/non-free armhf Packages
命中 http://mirrors.tuna.tsinghua.edu.cn wheezy/rpi armhf Packages
命中 http://archive.raspberrypi.org wheezy InRelease
命中 http://archive.raspberrypi.org wheezy/main armhf Packages
忽略 http://mirrors.tuna.tsinghua.edu.cn wheezy/contrib Translation-zh_CN
忽略 http://mirrors.tuna.tsinghua.edu.cn wheezy/contrib Translation-zh
忽略 http://mirrors.tuna.tsinghua.edu.cn wheezy/contrib Translation-en
忽略 http://mirrors.tuna.tsinghua.edu.cn wheezy/main Translation-zh_CN
忽略 http://mirrors.tuna.tsinghua.edu.cn wheezy/main Translation-zh
忽略 http://mirrors.tuna.tsinghua.edu.cn wheezy/main Translation-en
忽略 http://mirrors.tuna.tsinghua.edu.cn wheezy/non-free Translation-zh_CN
忽略 http://mirrors.tuna.tsinghua.edu.cn wheezy/non-free Translation-zh
忽略 http://mirrors.tuna.tsinghua.edu.cn wheezy/non-free Translation-en
忽略 http://mirrors.tuna.tsinghua.edu.cn wheezy/rpi Translation-zh_CN
忽略 http://mirrors.tuna.tsinghua.edu.cn wheezy/rpi Translation-zh
忽略 http://mirrors.tuna.tsinghua.edu.cn wheezy/rpi Translation-en
忽略 http://archive.raspberrypi.org wheezy/main Translation-zh_CN
忽略 http://archive.raspberrypi.org wheezy/main Translation-zh
忽略 http://archive.raspberrypi.org wheezy/main Translation-en
正在读取软件包列表.....完成
```

在仓库信息更新完成后,用户可以通过 apt 程序来对整个系统进行更新。在 Debian 系统中,更新系统常用的方法是通过 apt-get 加上 upgrade 参数或 dist-upgrade 参数来更新。upgrade 参数只是将目前的系统进行升级,并不会处理新安装软件的依赖关系。如果有软件在更新时被这种依赖关系影响,upgrade 参数会默认保留旧的软件,例如:

```
# apt - get upgrade
```


正在读取软件包列表.....完成

正在分析软件包的依赖关系树

正在读取状态信息.....完成

下列软件包的版本将保持不变:

```
gnome-themes-standard libqt4-dbus libqt4-network libqt4-svg libqt4-xml
libqtcore4
libqtgui4 qdbus
```

下列软件包将被升级:

```
alsa-utils apt apt-utils base-files bash bsdutils console-setup
console-setup-linux cpio cpp-4.6 cups-bsd cups-client cups-common debconf
debconf-i18n debconf-utils firmware-atheros firmware-brcm80211 firmware-libertas
firmware-ralink firmware-realtek fontconfig fontconfig-config fuse g++-4.6
gcc-4.6 gcc-4.6-base gcc-4.7-base ghostscript gnupg gpgv gvfs-gvfs-backends
gvfs-common gvfs-daemons gvfs-fuse gvfs-libs idle-python2.7
keyboard-configuration libapt-inst1.5 libapt-pkg-dev libapt-pkg4.12 libavcodec53
libavutil51 libblkid1 libcups2 libcupsimage2 libdrm2 libfontconfig1 libfreetype6
libfuse2 libgail-3-0 libgcc1 libgfortran3 libgomp1 libgs9 libgs9-common
libgtk-3-0 libgtk-3-bin libgtk-3-common libmikmod2 libmount1 libnfnetwork0
libnih-dbus1 libnih1 libparted0debian1 libpci3 libpolkit-agent-1-0
libpolkit-backend-1-0 libpolkit-gobject-1-0 libportmidi0 libpython2.7
libreadline5 libreadline6 libsmclient libstdc++6 libstdc++6-4.6-dev libsysfs2
libsystemd-daemon0 libsystemd-login0 libtiff4 libuuid1 libwbclient0 libxapian-dev
libxapian22 linux-libc-dev midori mount parted pciutils perl perl-base
perl-modules pistore policykit-1 python-pygame python2.7 python2.7-minimal
readline-common samba-common smbclient tar tzdata util-linux xarchiver xkb-data
```

升级了 106 个软件包,新安装了 0 个软件包,要卸载 0 个软件包,有 8 个软件包未被升级。

需要下载 79.7 MB 的软件包。

解压缩后会消耗掉 301 KB 的额外空间。

您希望继续执行吗? [Y/n]

在上面的例子中,upgrade 程序只升级了 106 个软件,另外有 8 个软件因为依赖关系,而被保留了旧的版本。

如果遇到上面这种问题,可以使用 dist-upgrade 参数来升级软件,dist-upgrade 参数会通过安装或删除软件自行解决软件包的依赖关系。整个过程完全自动化,不需要用户操心。

```
# apt-get dist-upgrade
```

正在读取软件包列表.....完成

正在分析软件包的依赖关系树

正在读取状态信息.....完成

正在对升级进行计算.....完成

下列【新】软件包将被安装:

```
gnome-accessibility-themes gnome-themes-standard-data libqtdbus4
```

下列软件包将被升级：

```
alsa-utils apt apt-utils base-files bash bsdutils console-setup
console-setup-linux cpio cpp-4.6 cups-bsd cups-client cups-common debconf
debconf-i18n debconf-utils firmware-atheros firmware-brcm80211 firmware-libertas
firmware-ralink firmware-realtek fontconfig fontconfig-config fuse g++-4.6
gcc-4.6 gcc-4.6-base gcc-4.7-base ghostscript gnome-themes-standard gnupg gpgv
gvfs gvfs-backends gvfs-common gvfs-daemons gvfs-fuse gvfs-libs idle-python2.7
keyboard-configuration libapt-inst1.5 libapt-pkg-dev libapt-pkg4.12 libavcodec53
libavutil51 libblkid1 libcups2 libcupsimage2 libdrm2 libfontconfig1 libfreetype6
libfuse2 libgail-3-0 libgcc1 libgfortran3 libgomp1 libgs9 libgs9-common
libgtk-3-0 libgtk-3-bin libgtk-3-common libmikmod2 libmount1 libnfnetwork0
libnih-dbus1 libnih1 libparted0debian1 libpci3 libpolkit-agent-1-0
libpolkit-backend-1-0 libpolkit-gobject-1-0 libportmidi0 libpython2.7
```

```
libqt4-dbus
```

```
libqt4-network libqt4-svg libqt4-xml libqtcore4 libqtgui4 libreadline5
libreadline6 libsmbclient libstdc++6 libstdc++6-4.6-dev libsysfs2
libsystemd-daemon0 libsystemd-login0 libtiff4 libuuid1 libwbclient0 libxapian-dev
libxapian22 linux-libc-dev midori mount parted pciutils perl perl-base
perl-modules pistore policykit-1 python-pygame python2.7 python2.7-minimal qdbus
readline-common samba-common smbclient tar tzdata util-linux xarchiver xkb-data
```

升级了 114 个软件包，新安装了 3 个软件包，要卸载 0 个软件包，有 0 个软件包未被升级。

需要下载 89.1 MB 的软件包。

解压缩后会消耗掉 164 KB 的额外空间。

您希望继续执行吗？[Y/n]

需要说明的是，`apt-get upgrade` 和 `dist-upgrade` 两个参数其实没有区别，都是升级软件包。它们的区别在于，`dist-upgrade` 能识别出当依赖关系改变时的情形并做出处理，而 `upgrade` 对此情形不处理。

当用户在 `apt-get` 程序中确认“y”后，将会开始整个升级过程。由于树莓派的 CPU 性能较低，因此整个更新过程需要很长时间。当更新完成后，整个系统会成为最新的 Debian 7，同时还需要再次更新 Debian 7 的 `apt` 包信息，并使用 `apt-get` 程序清理那些不需要的软件包：

```
# apt-get update
# apt-get autoremove --purge
```

如果用户是从较早的版本升级，可能会在升级完成后再次安装软件时出现以下错误：

```
W: GPG error: http://mirrors.163.com lenny-proposed-updates Release: The following
signatures couldn't be verified because the public key is not available: NO _
PUBKEY AED4B06F473041FA
```

第3章 使用树莓派

W: GPG error: http://mirrors.163.com lenny/updates Release: The following signatures couldn't be verified because the public key is not available: NO_PUBKEY AED4B06F473041FA

W: You may want to run apt - get update to correct these problems

这时可以通过更新 apt 程序公钥来解决这个问题:

```
# aptitude install debian - archive - keyring
```

下列“新”软件包将被安装。

```
debian - archive - keyring
```

0 个软件包被升级,新安装 1 个,0 个将被删除,同时 0 个将不升级。

需要获取 30.4 KB 的存档。解包后将使用 84.0 KB。

警告:您将安装以下软件包的不可信版本!

不可信的软件可能会危害您的系统安全。

只有当非常清楚地了解这就是您所要执行的操作时,再进行安装操作。

```
debian - archive - keyring
```

您想忽略这条警告信息并继续执行吗?

要继续,请键入“Shi”;要中止,请键入“Fou”;shi

更新完成后即可解决无法获得正确的密钥的问题。

3.3 设置启动参数

比起早期树莓派系统,现在的系统无论是可用性还是稳定性都要成熟很多,这得益于整个 Debian 系统的不断修复和新版本的演进。以 raspi-config 为例,在早期的版本中,raspi-config 的某些功能还不能很好地工作,但是现在,raspi-config 程序已经可以很好地配置整个树莓派系统了。这包括显存使用数量、分辨率设置及超频等功能,而不再像以前一样去修改配置文件。

3.3.1 使用 raspi-config 配置树莓派

在 Debian 系统上,第一次启动树莓派并登录成功后会启动 raspi-config 程序,以便让用户配置自己的树莓派系统。如果用户在以后的使用中想要修改一些配置,也可以通过 raspi-config 命令来重新启动这个程序,如图 3-11 所示。

raspi-config 程序以菜单驱动的方式进行选择,读者可以通过上、下光标键选择需要配置的项目,而左、右光标键可通过点击 Select 来对项目进行确认,最后点击 Finish 按钮退出 raspi-config 程序。

第一项是 info,介绍了 raspi-config 程序的作用。

第二项是 expand_rootfs,作用是扩展根系统到整个 SD 卡上。如果用户使用的是较大的 SD 卡,由于默认镜像只有 2 GB,因此写入后也只会占用 2 GB 的空间,SD

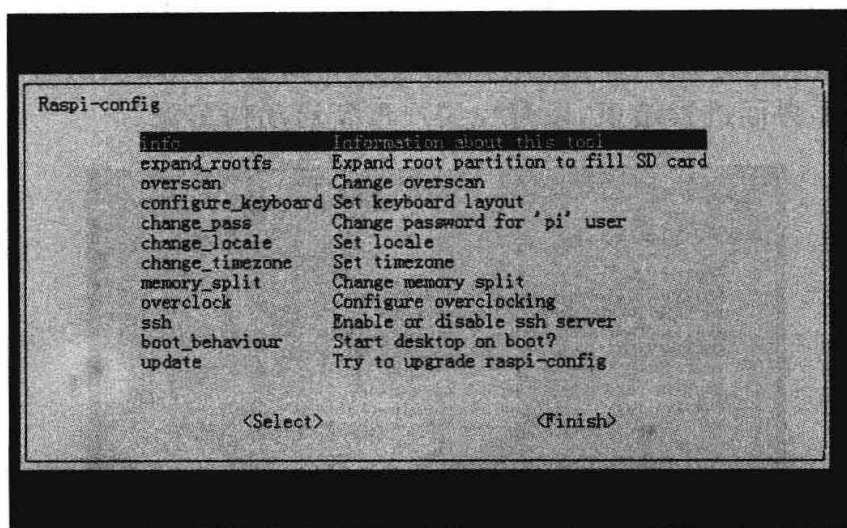


图 3-11 raspi-config 程序

卡上其他的空间则不会用到,而 `expand_rootfs` 选项可以让用户对系统分区进行扩展,可扩展到使用整个 SD 卡的空间。选中 `expand_rootfs` 选项,确认后等上几秒,会看到成功扩展根分区到整个 SD 卡上的提示,重新启动后会生效。

在某些显示设备上,树莓派默认的显示并不会完全占满整个显示器空间,而 `overscan` 选项可以通过 `overscan` 开关使屏幕显示占满显示窗口,如图 3-12 所示。

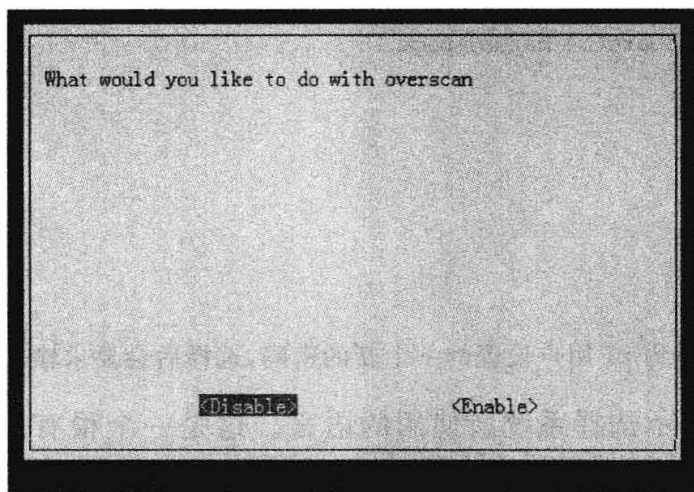


图 3-12 禁用 overscan 以占满显示器

`configure_keyboard` 选项可以配置键盘。由于树莓派的产地是英国,因此默认的键盘布局也是英式的。这对于国内使用美式键盘布局的用户来说是非常不方便的。

在 `configure_keyboard` 选项中需要首先选择键盘类型,对于国内大多数用户来说,直接选择 `Generic 105-key (Intl) PC` 键盘即可。在随后的键盘布局中,默认的全是英式键盘,因此需要选择 `Other` 选项,如图 3-13 所示,并在 `Other` 中确认选择 `English (US)` 选项。接下来会提到辅助键的问题,这时可以使用默认,并在下一个界

第3章 使用树莓派

面中选择 No compose key 选项。最后一个选择是,确认是否打开 Ctrl+Alt+Backspace 组合键。它可以在图形界面无响应的时候不需重启而通过组合键 Ctrl+Alt+Backspace 将图形界面进程结束,以避免用户重新启动树莓派。

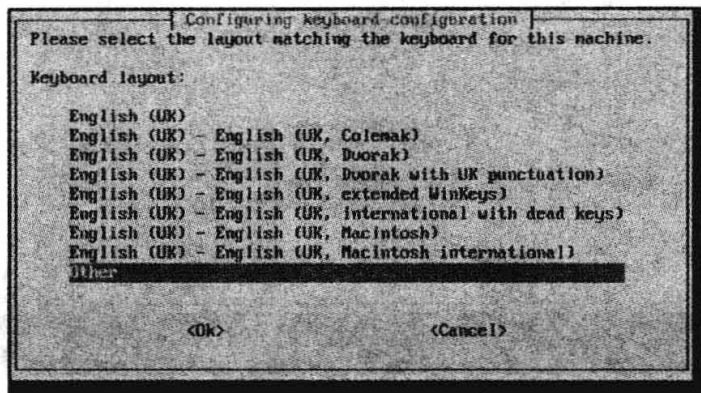


图 3-13 选择其他地区的键盘配置

事实上,最方便的配置键盘的方法还是直接修改键盘的配置文件,如/etc/default/keyboard,将 XKBLAYOUT="gb"中的 gb 改为 us,即可将键盘布局由英式改为美式。

```
# cat/etc/default/keyboard
# KEYBOARD CONFIGURATION FILE

# Consult the keyboard(5) manual page.

XKBMODEL = "pc105"
XKBLAYOUT = "us"
XKBVARIANT = ""
XKBOPTIONS = ""

BACKSPACE = "guess"
```

change_pass 选项是为 pi 用户设置的一个新的密码,选择后会要求输入两次新的密码。

change_locale 表示选择系统所使用的语言。这是一个很有用的选项,如果用户需要使用中文,可以在这个选项中选择以 zh_CN 开头的语言编码,如图 3-14 所示。

用户可以通过空格键来选择多个语言编码,但是在系统中只有一个语言编码会起作用,当选择好语言并确认后可以看到图 3-15。

这里需要用户选择一个默认的语言编码,一般默认使用 zh_CN.UTF-8 编码就可以了。

change_timezone 选项的作用是设置时区,和键盘设置一样。树莓派默认的时区也是英国的,用户可以在 change_timezone 选项中选择 Asia(亚洲)、Shanghai(上海)来设置中国的时区。在时区设置中是没有北京的,中国区只有上海、重庆等地区。

memory_split 选项的作用是设置显存。如果用户并不需要通过树莓派来播放视频或是使用桌面系统,可以将内置 GPU 的显存设置为最小的 16 MB,以节省出更

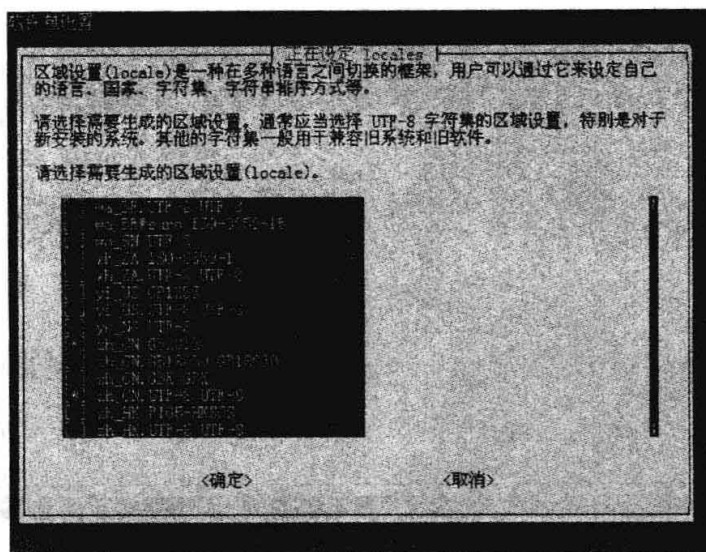


图 3-14 选择语言编码

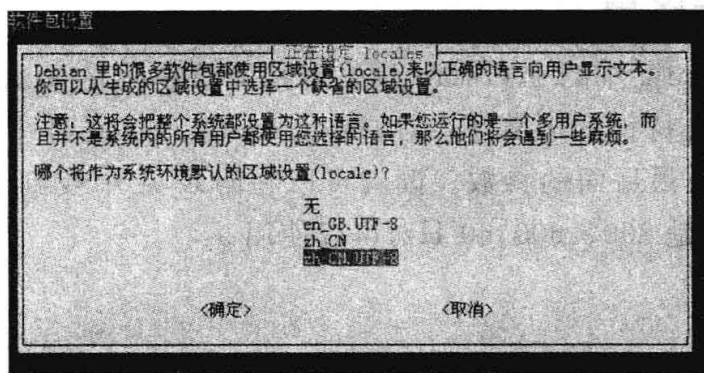


图 3-15 选择默认的编码

多的内存给系统使用。

Overclock(超频),你没有看错,由于树莓派默认 CPU 的性能较低,所以树莓派允许适当的超频,但是这可能会缩短树莓派的寿命,并有可能烧毁掉树莓派系统,因此并不推荐使用,如图 3-16 所示。

默认的频率是 700 MHz,用户可以选择将树莓派超频到 800~1 000 MHz 来使用。

ssh 是默认的远程管理工具。这个选项是选择是否将 ssh 作为默认服务在开机后启动,强烈推荐使用开机后启动,因为通过 ssh 可以远程管理树莓派的一切。

选项 boot_behaviour 为设置是否在开机后直接进入桌面,而不是进入到命令行状态,默认选项表示开机不进入桌面。如果用户偏爱桌面可以设置为 YES。

选项 update 为更新系统软件,用户可以通过 3.2.4 小节介绍的更新系统方法来手动更新。

当所有的设置完成后选择 Finish,系统会提示用户重新启动树莓派,以便让新的配置生效。

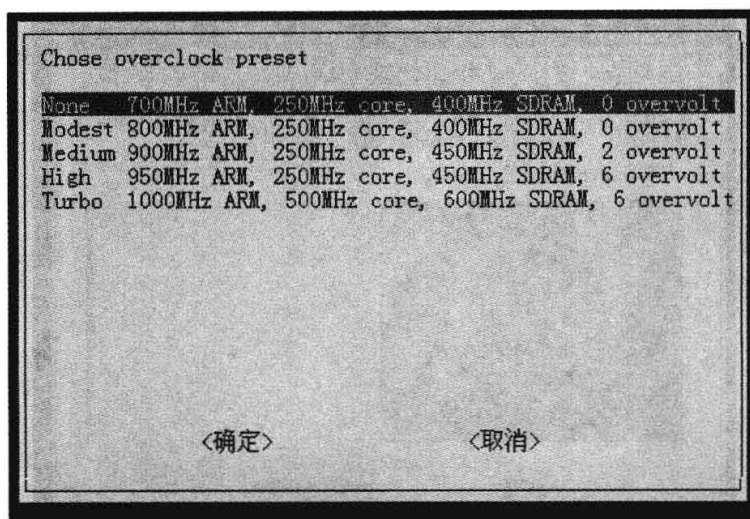


图 3-16 超频设置

3.3.2 分辨率设置

如果在 raspi-config 中的 overscan 开关还是无法将你的显示屏幕占满,那么需要设置启动参数来支持你的屏幕了。在 /boot/config.txt 文件中,删除 hdmi_mode 之前的“#”号并修改最后面的参数。读者可以在附录 A 中找到合适的参数来匹配显示器,例如显示器是 800×600、60 Hz,可以使用 9:

```
hdmi_mode = 9
```

如果你的显示器所使用的分辨率在附录 A 中都找不到,那么只能使用手动方式来指定显示器的长宽尺寸,以便于系统进行控制;另外,修改 boot/config.txt 文件,将以下两行的注释取消,然后在参数部分输入显示器所支持的长宽尺寸:

```
framebuffer_width = 1280
framebuffer_height = 720
```

修改完成后需要重新启动树莓派以便于设置生效。

3.4 使用网络

对于树莓派所使用的 Linux 系统来说,如果没有了网络,那么真的是没有什么使用的乐趣,你无法更新系统,无法通过网络安装新的软件,只能使用树莓派自身的功能。因此请尽可能在树莓派上使用网络。

3.4.1 设置有线网络

在树莓派默认的系统里,有线网络是使用 dhcp 来获得 IP 地址的,因此如果网络中有路由器或是 dhcp 服务器,树莓派可以在开机后自动获得相应的网络设置。用户

可以通过查看路由器的 dhcp 记录来得知树莓派的 IP 地址,如图 3-17 所示。

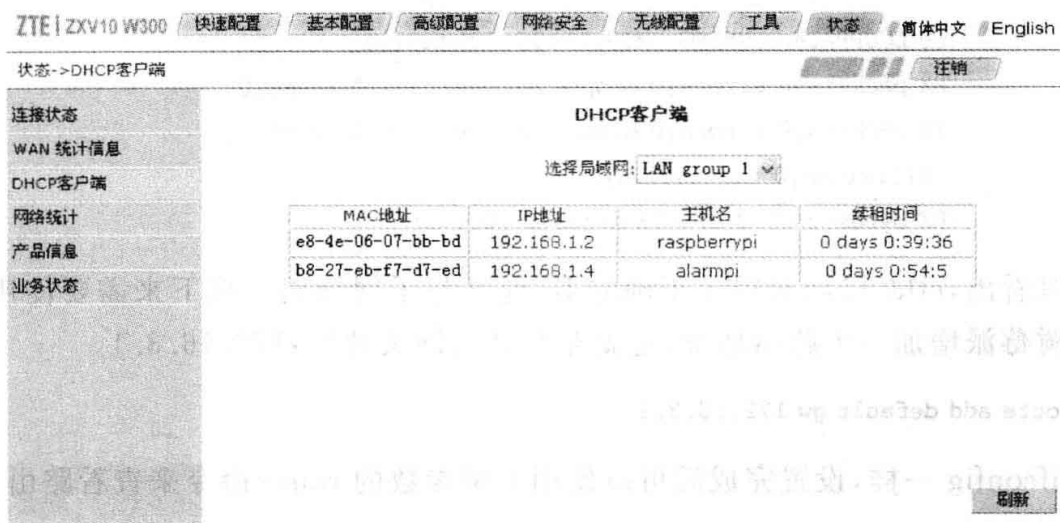


图 3-17 使用路由器查看 dhcp 信息

如果 dhcp 服务器配置无误,则树莓派在获得了网络参数后即可连接到网络上。对于那些心急上网的用户也可以使用以下三条命令来配置网络,只不过这种配置并不会写入到配置文件中,因此重启树莓派后就会丢失网络配置:

```
# Ifconfig eth0 172.16.3.4 netmask 255.255.255.128
# route add default gw 172.16.3.1
# echo nameserver 202.103.44.150 > /etc/resolv.conf
```

手动设置树莓派的网络需要通过 ifconfig 与 route 两个命令。其中,ifconfig 用于配置网卡的 IP 与子网地址,而 route 则是配置路由地址。最后一条命令是写入 DNS 地址到/etc/resolv.conf 文件中。下面详细介绍这三条命令的用法。

```
# Ifconfig eth0 172.16.3.4 netmask 255.255.255.128
```

其中,eth0 是指树莓派的网卡,如果还有其他的有线网卡则可能是 eth1;172.16.3.4 是要为树莓派指定的 IP 地址;netmask 是为树莓派指定一个子网掩码 255.255.255.128。读者可以将 IP 地址与子网掩码换成自己所在网络的 IP 地址和子网掩码。

设置完成后可以使用不加任何参数的 ifconfig 命令来查看网络配置情况:

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:f7:d7:ed
          inet addr:172.16.3.4  Bcast:172.16.3.127  Mask:255.255.255.128
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:109978 errors:0 dropped:304 overruns:0 frame:0
          TX packets:14872 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:38051599 (36.2 MiB)  TX bytes:2049795 (1.9 MiB)
```

第3章 使用树莓派

```
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            UP LOOPBACK RUNNING  MTU:16436  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

可以看出,eth0 已经获得了正确的 IP 地址与子网掩码。接下来需要使用 route 命令为树莓派增加一个路由地址,也就是默认的网关地址:172.16.3.1。

```
# route add default gw 172.16.3.1
```

与 ifconfig 一样,设置完成后可以使用不带参数的 route 命令来查看路由表:

```
# route
Kernel IP routing table

```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
default	172.16.3.1	0.0.0.0	UG	0	0	0	eth0
172.16.3.0	*	255.255.255.128	U	0	0	0	eth0

可以看出,已经新增加了一个默认的缺省网关 172.16.3.1。

最后的 echo 命令是将 nameserver 202.103.44.150 写入到/etc/resolv.conf 文件中,以达到更新 DNS 记录的目的。

但更多的时候,我们都会为树莓派指定一个固定的 IP 地址并写入到配置文件中。这样做的好处是,每次树莓派启动后 IP 地址是固定的,而不需要通过 dhcp 服务器来查看树莓派的 IP 地址,或是每次启动后都输入命令来配置网络。在 Debian 中,配置网络的文件主要是由/etc/network/目录下的 interfaces 文件和/etc 目录中 resolv.conf 文件控制。前者是设置网卡参数,后者是设置 DNS 参数。

/etc/network/ interfaces 文件很简单,下面介绍每一行的含义:

auto lo eth0	开机后自动启动网络回环和网卡;
iface lo inet loopback	设置 lo 为网络回环;
iface eth0 inet static	设置 eth0 使用静态地址;
address 172.16.3.4	设置 eth0 的 IP 地址为 172.16.3.4;
netmask 255.255.255.128	设置 eth0 的子网掩码为 255.255.255.128;
gateway 172.16.3.1	设置 eth0 的网关为 172.16.3.1。

如果想让树莓派在每次启动时自动获得 IP 地址,可以使用将 # iface eth0 inet dhcp 一行前面的“#”号删除,并删除 iface eth0 inet static 行及后面的内容,这样每次开机时树莓派会使用 dhcp 方式启动网卡来获得 IP 地址。

至此,已经完成大部分的树莓派网络参数的设置,只剩下最后一项——设置 DNS 参数。DNS 参数需要在 `/etc/resolv.conf` 文件中通过 `nameserver` 指令来指定。例如:`nameserver 8.8.8.8`,读者可以使用熟悉的编辑器修改 `/etc/resolv.conf` 文件,将 `nameserver` 后面的参数改为自己网络中 DNS 的参数,如下所示:

```
# cat /etc/resolv.conf
nameserver 202.103.44.150
nameserver 8.8.8.8
```

读者也可以在 `resolv.conf` 文件中像上面一样指定多个 DNS 服务器的 IP 地址。配置完成后需要重新启动网络进程,让新的网络配置生效:

```
# /etc/init.d/networking restart
[warn] Running /etc/init.d/networking restart is deprecated because it may not re-ena-
ble some interfaces ... (warning).
[ ok ] Reconfiguring network interfaces...done.
```

现在整个网络的设置完成了,下面通过最简单的 `ping` 程序来调试网络是否连通:

```
# ping www.baidu.com -c 5
PING www.a.shifen.com (115.239.210.26) 56(84) bytes of data.
64 bytes from 115.239.210.26: icmp_req=1 ttl=55 time=21.7 ms
64 bytes from 115.239.210.26: icmp_req=2 ttl=55 time=22.8 ms
64 bytes from 115.239.210.26: icmp_req=3 ttl=55 time=23.8 ms
64 bytes from 115.239.210.26: icmp_req=4 ttl=55 time=22.8 ms
64 bytes from 115.239.210.26: icmp_req=5 ttl=55 time=21.8 ms
--- www.a.shifen.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 21.795/22.663/23.895/0.799 ms
```

在上例中,通过 `ping` 程序发送了 5 个 `ping` 信号,调试了是否能连接到百度网站,从返回的消息来看,连接正常,百度网站返回了正确的消息。

3.4.2 设置无线网络

在使用无线网络之前,需要先确认无线网卡的驱动是否已经加载了。如果用户的无线网卡被内核支持,则内核会自动加载它的驱动模块。例如,笔者使用的是 EP-N8508GS 的无线网卡,而与这个网卡对应的驱动模块为 `8192cu`,那么通过 `lsmod` 命令可以查看到这个模块已经被自动加载了:

```
# lsmod
Module                Size      Used    by
snd_bcm2835           12808      0
```

snd_pcm	74834	1	snd_bcm2835
snd_seq	52536	0	
snd_timer	19698	2	snd_seq,snd_pcm
snd_seq_device	6300	1	snd_seq
snd	52489	5	snd_seq_device,snd_timer,snd_seq,snd_pcm,snd_bcm2835
snd_page_alloc	4951	1	snd_pcm
8192cu	485042	0	

如果用户的无线网卡没有被支持,那么可能需要去网卡生产厂商的网站上下载它的 Linux 驱动程序;如果厂商没有提供驱动程序,那么树莓派则无法使用这个无线网卡。用户可能需要重新购买一个树莓派所支持的无线网卡。

与配置有线网络一样,无线网络的配置也需要使用到/etc/network/interfaces 文件。读者可以在 interfaces 文件的最后面加入以下内容:

```
allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

由于无线路由器通常都会为客户端分配 IP 地址,因此在这个例子里我们在配置文件中的最后一行设置了 dhcp 为缺省的 IP 获得方式。

修改完以后,还需要编辑/etc/wpa_supplicant/目录下的 wpa_supplicant.conf 文件,并写入以下内容:

```
ctrl_interface = DIR = /var/run/wpa_supplicant GROUP = netdev
update_config = 1

network = {
    ssid = "需要连接的无线信号的 SSID"
    psk = "无线网络密码"
    proto = RSN
    key_mgmt = WPA-PSK
    pairwise = CCMP
    auth_alg = OPEN
}
```

保存退出后就可以使用 ifup 来启动无线网络了。

```
# ifup wlan0
```

等上一会儿,使用 iwconfig wlan0 命令可以查看无线网络是否已经连接上了。

```
# iwconfig wlan0
```

```
wlan0 IEEE 802.11bg ESSID:"o621" Nickname:"<WIFI@REALTEK>"
```



```

Mode:Managed Frequency:2.437 GHz Access Point: 00:15:EB:56:39:19
Bit Rate:54 Mb/s Sensitivity:0/0
Retry:off RTS thr:off Fragment thr:off
Encryption key: ****-****-****-****-****-****-****-****
Security mode:open
Power Management:off
Link Quality=100/100 Signal level=72/100 Noise level=0/100
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0

```

使用 ping 程序也可以联通到无线路由器的网关 192.168.1.1 上,如下所示:

```

# ping 192.168.1.1 -c 4
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_req=1 ttl=255 time=2.72 ms
64 bytes from 192.168.1.1: icmp_req=2 ttl=255 time=3.91 ms
64 bytes from 192.168.1.1: icmp_req=3 ttl=255 time=2.70 ms
64 bytes from 192.168.1.1: icmp_req=4 ttl=255 time=2.77 ms

--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 2.703/3.028/3.912/0.514 ms

```

如果要查看当前有哪些无线网络,可以使用 iwlist 程序。下例中通过 iwlist 命令找到了三个无线网络,分别是 o621、CMCC 和 CMCC-AUTO。

```

# iwlist wlan0 scan
wlan0 Scan completed :
    Cell 01 - Address: 00:15:EB:56:39:19
        ESSID:"o621"
        Protocol:IEEE 802.11bg
        Mode:Master
        Frequency:2.437 GHz (Channel 6)
        Encryption key:on
        Bit Rates:54 Mb/s
        Extra:rsn_ie=30140100000fac040100000fac040100000fac020000
        IE: IEEE 802.11i/WPA2 Version 1
            Group Cipher : CCMP
            Pairwise Ciphers (1) : CCMP
            Authentication Suites (1) : PSK
        Quality=100/100 Signal level=70/100
    Cell 02 - Address: 06:08:D2:E0:62:4F
        ESSID:"CMCC"
        Protocol:IEEE 802.11bg
        Mode:Master
        Frequency:2.412 GHz (Channel 1)

```

```

Encryption key:off
Bit Rates:54 Mb/s
Quality = 101/100   Signal level = 47/100
Cell 03 - Address: 0A:08:D2:E0:62:4F
ESSID:"CMCC - AUTO"
Protocol:IEEE 802.11bg
Mode:Master
Frequency:2.412 GHz (Channel 1)
Encryption key:on
Bit Rates:54 Mb/s

Extra:wpa_ie = dd160050f20101000050f20401000050f20401000050f201
IE: WPA Version 1
    Group Cipher : CCMP
    Pairwise Ciphers (1) : CCMP
    Authentication Suites (1) : 802.1x
Extra:rsn_ie = 30140100000fac040100000fac040100000fac010100
IE: IEEE 802.11i/WPA2 Version 1
    Group Cipher : CCMP
    Pairwise Ciphers (1) : CCMP
    Authentication Suites (1) : 802.1x
    Preauthentication Supported
Quality = 101/100   Signal level = 46/100

```

对于图形界面的用户而言,如果想要配置无线网络,可以使用 wpa_gui 程序来进行配置,也可以使用 Scan 来搜索无线连接,使用 Connect 连接到无线网络上,非常方便,如图 3-18 所示。

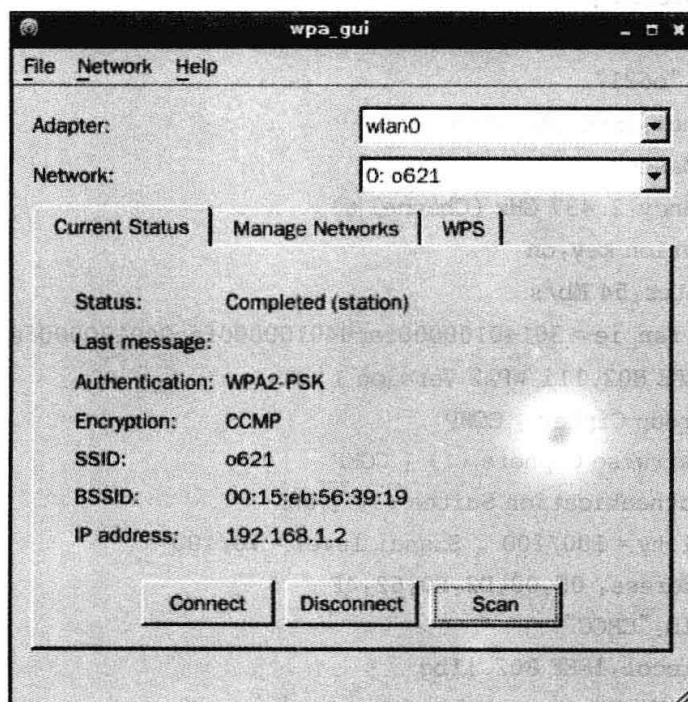


图 3-18 使用图形界面设置无线参数

3.5 中文环境

中文环境在任何一个 Linux 系统的使用中都是一个大问题,如何中文化也是在网络上被问到较多的问题之一。经过多年的发展,现在 Linux 系统已经可以较为完整地支持中文的显示、输入与输出,而在树莓派上使用中文需要完成以下的设置:

- ① 设置中文环境,让系统知道用户所使用的语言环境是中文。
- ② 安装字库,只有安装了中文字库,系统才能正确地显示出中文。
- ③ 窗口环境中安装输入法以支持中文输入。
- ④ 在命令行环境使用中文需要安装 zhcon 的中文平台。

只有完成了以上设置,用户才能在窗口或命令行下显示和输入中文。

3.5.1 设置中文环境

设置中文环境最简单的方法就是通过 raspi-config 程序来设置。读者可以查看 3.3.1 小节中的 change_locale 选项部分来选择一个中文编码,但是这种方法并没有安装语言支持包,因此最好还是通过手动来设置中文环境。

首先安装 locales-all 语言包。这个语言包包含了 Linux 系统所支持的所有语言编码,虽然软件包本身只有 3 MB 左右,但是解压后需要占用约 114 MB 的空间,因此请保证 SD 卡上有足够的空间。

```
# apt - get install locales - all
```

安装完成后用户可以通过 locale-a 命令来查看当前系统所支持的所有语言编码。在显示的最后会有很多 zh_CN 的中文支持,说明系统已经做好了支持中文的准备:

```
# locale - a
... .. 以上删除很多.....
yo_NG
zh_CN
zh_CN.gb18030
zh_CN.gbk
zh_CN.utf8
zh_HK
zh_HK.utf8
zh_SG
zh_SG.gbk
zh_SG.utf8
zh_TW
zh_TW.euctw
```

```
zh_TW.utf8
zu_ZA
zu_ZA.utf8
```

现在来查看一下当前系统的语言编码,通过不加参数的 `locale` 命令就可以看到当前系统所使用的语言为英式的 UTF-8 编码:

```
# locale
LANG = en_GB.UTF-8
LANGUAGE =
LC_CTYPE = "en_GB.UTF-8"
LC_NUMERIC = "en_GB.UTF-8"
LC_TIME = "en_GB.UTF-8"
LC_COLLATE = "en_GB.UTF-8"
LC_MONETARY = "en_GB.UTF-8"
LC_MESSAGES = "en_GB.UTF-8"
LC_PAPER = "en_GB.UTF-8"
LC_NAME = "en_GB.UTF-8"
LC_ADDRESS = "en_GB.UTF-8"
LC_TELEPHONE = "en_GB.UTF-8"
LC_MEASUREMENT = "en_GB.UTF-8"
LC_IDENTIFICATION = "en_GB.UTF-8"
LC_ALL =
```

`export` 命令用于设置系统使用中文语言,编码则使用 `utf8`。

```
# export LANG = zh_CN.utf8
# export LC_ALL = zh_CN.utf8
```

再次查看当前系统语言,可以看到整个系统都已经使用了 `zh_CN` 的中文语言:

```
# locale
LANG = zh_CN.utf8
LANGUAGE =
LC_CTYPE = "zh_CN.utf8"
LC_NUMERIC = "zh_CN.utf8"
LC_TIME = "zh_CN.utf8"
LC_COLLATE = "zh_CN.utf8"
LC_MONETARY = "zh_CN.utf8"
LC_MESSAGES = "zh_CN.utf8"
LC_PAPER = "zh_CN.utf8"
LC_NAME = "zh_CN.utf8"
LC_ADDRESS = "zh_CN.utf8"
LC_TELEPHONE = "zh_CN.utf8"
LC_MEASUREMENT = "zh_CN.utf8"
LC_IDENTIFICATION = "zh_CN.utf8"
```

```
LC_ALL = zh_CN.utf8
```

在/etc/profile 中的最后增加以下两行,以便让设置在重启后也能生效:

```
export LANG = zh_CN.utf8
export LC_ALL = zh_CN.utf8
```

至此,整个中文环境设置完成,用户只需要安装中文字体就可以在 Linux 中显示中文了。

3.5.2 安装中文字库

目前,在 Linux 系统下最好的字库就是文泉驿所提供的文泉驿字体,具体效果可以查看图 3-19。树莓派的软件仓库也收录了三种文泉驿字体,分别是文泉驿点阵、文泉驿正黑体以及文泉驿微米黑。用户可以通过以下命令来安装这三种字体:

```
# apt - get install ttf - wqy - microhei ttf - wqy - zenhei xfonts - wqy
```



图 3-19 文泉驿字体使用效果

除了文泉驿字库外,还有台湾文鼎科技提供的宋体和楷体等多种字体,读者可以通过 apt-get 程序来安装:

```
fonts - arphic - bkai00mp  繁体的楷体字库
fonts - arphic - bsmi00lp - "AR PL Mingti2L Big5" Chinese TrueType font by Arphic Technology
fonts - arphic - gbsn00lp  简体中文宋体字库
fonts - arphic - gkai00mp  简体中文楷体字库
fonts - arphic - ukai - "AR PL UKai" Chinese Unicode TrueType font collection Kaiti style
fonts - arphic - uming - "AR PL Uming" Chinese Unicode TrueType font collection Mingti style
latex - cjk - chinese - arphic - bkai00mp  latex 使用的简体中文楷体 CJK 字库
latex - cjk - chinese - arphic - bsmi00lp - traditional Chinese KaiTi fonts for CJK
latex - cjk - chinese - arphic - gbsn00lp - traditional Chinese KaiTi fonts for CJK
latex - cjk - chinese - arphic - gkai00mp  latex 使用的繁体中文楷体 CJK 字库
xfonts - intl - chinese  xorg 所带的简体中文字库
xfonts - intl - chinese - big  xorg 所带的繁体中文字库
fonts - moe - standard - kai  中文 TrueType 楷体字库(non-free)
fonts - moe - standard - song  中文 TrueType 宋体字库(non-free)
```

3.5.3 安装中文输入法

目前,在 Linux 系统下,有三种输入法可以选择,分别是 fcitx(小企鹅中文输入法)、scim 和 ibus2 输入法。其中,fcitx 是一套由国人开发的中文输入法,其编写目的是为桌面提供一个灵活的输入方案,而不像 scim 与 ibus2 那样复杂和繁琐。目前,它包括了三个内置的输入法拼音、区位和基于码表的输入法。

现在国内使用最多的输入法就是拼音与五笔,而在 fcitx 下使用这两种输入法,只需要通过以下命令来进行安装:

```
# apt - get install fcitx - pinyin fcitx - table - wbpinyin - y
```

apt 程序会自动解决所有的依赖关系,用户只需要等待安装完成后重新启动桌面即可使用到 fcitx 输入法,如图 3-20 所示。

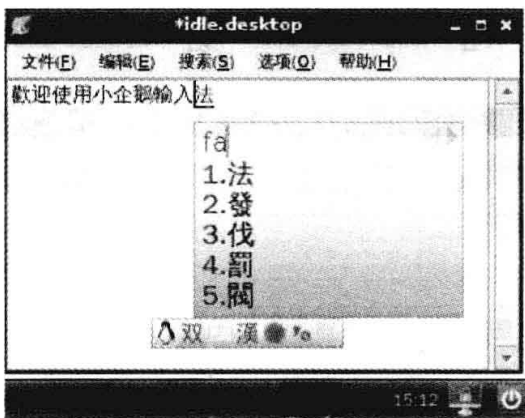


图 3-20 fcitx 输入法

除了 fcitx 以外,用户还可以使用 scim 和 ibus2。事实上,这两个软件并不是输入法,只是输入引擎。通过这些输入引擎可以使用多种输入法来输入,例如安装 scim 时再加上 scim-pinyin 包即可得到拼音输入法的支持:

```
# apt - get install scim scim - pinyin
```

安装完成后重新启动桌面系统,在右下角可以看到 scim 的使用图标,用户可以像在 Windows 系统下一样来使用 scim 输入汉字,如图 3-21 所示。

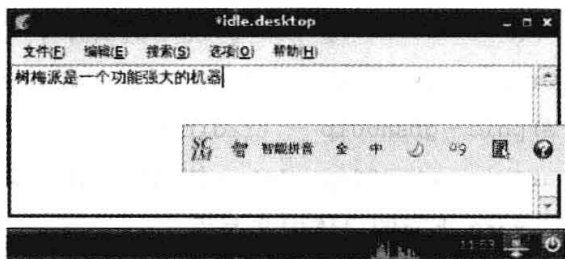


图 3-21 scim 输入法

如果想使用 ibus, 可以使用下面的命令来安装:

```
# apt - get install ibus - table - wubi ibus - pinyin
```

如果读者想尝试一下这三种输入法后再决定使用哪一种输入法, 可以先将这三种输入法都安装到系统中, 然后使用 Debian 提供的输入法配置工具来进行配置, 选择当前系统中使用哪一个输入法作为默认的输入法来使用。

在命令行中输入 `im-config` 来选择一个输入法, 如图 3-22 所示。系统提示是使用自动模式还是手动模式。如果要自行选择输入法则需要使用手动模式, 如下所示:

```
# im-config
```

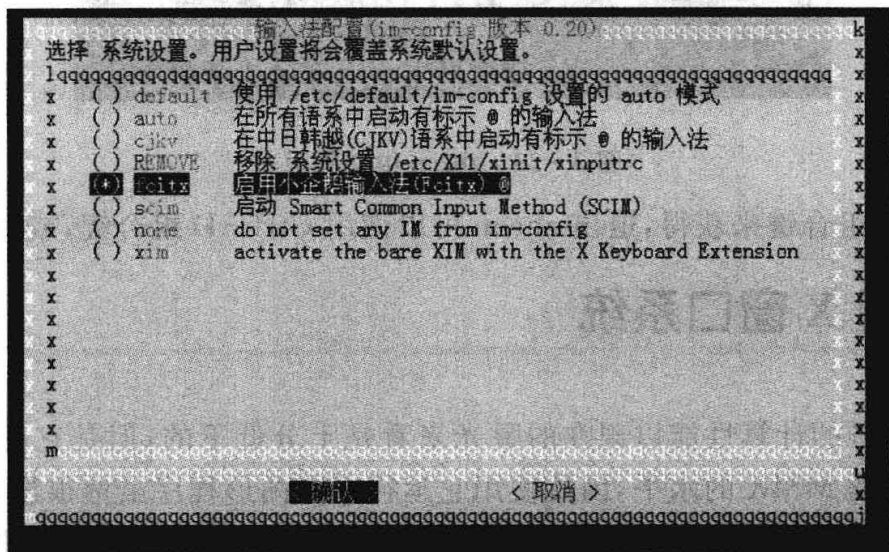


图 3-22 输入法配置工具

选择一个喜欢的输入法作为默认的输入法, 这里选择 fcitx 作为默认的输入法。

如果选择的输入法还需要安装其他组件, 则 Im-config 程序会提示读者, 读者可以自行选择安装这些组件。

3.5.4 命令行中文支持

在命令行输入中文需要安装 zhcon。这是一个老牌的中文平台, 支持在 Linux 控制台下的多内码中文平台, 能在命令行下显示简繁体中文、日文、韩文等双字节字符。characters. Supported 语言编码包括 GB2312、GBK、BIG5、JIS 和 KSC, 还支持 UTF-8。图 3-23 就是 zhcon 的运行界面。

```
# apt - get install zhcon
```

安装完成后通过 zhcon 命令启动 zhcon。

进入 zhcon 后读者可以在控制台显示和输入中文, 如果需要使用帮助可以通过

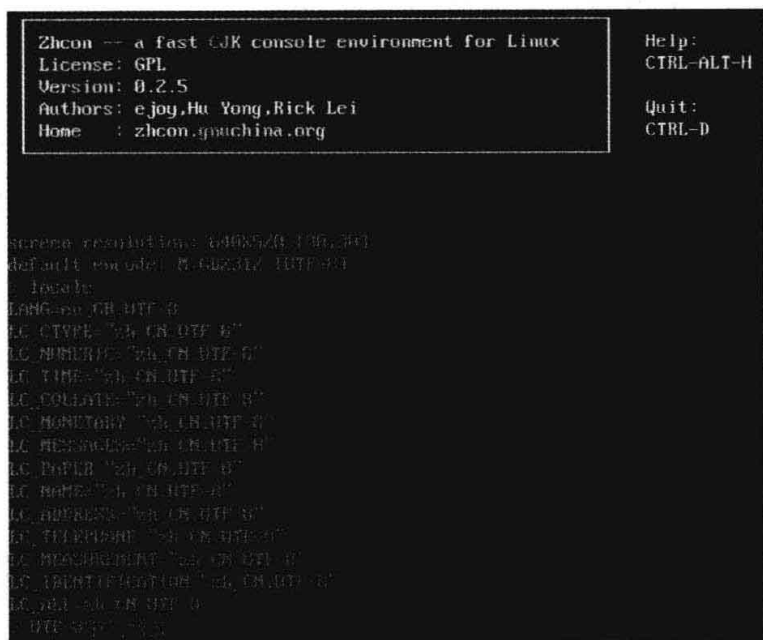


图 3-23 zhcon 中文界面

Ctrl+Alt+H 组合键来获得,退出 zhcon 可以使用 Ctrl+D 组合键,或者 exit 命令。

3.6 使用 X 窗口系统

虽然树莓派的计算性能以现在的眼光来看是十分低下的,但是它的实现性能仍然可以达到奔腾 MMX 的水平,因此使用它运行一些图形程序虽然很慢,但还是可以用的。不过遗憾的是,树莓派网站所提供的 Debian 并没有包含太多的图形程序,一些如文字处理、电子表格或是图像编辑工具都需要用户手动通过 apt 程序来安装。当然用户也可以选择一些云程序,这样只需要一个浏览器即可完成包含文字处理等在内的的工作。

另外,如果用户打算长期使用图形界面,那么可以肯定的是,需要为集成的显卡分配更多的显存,以获得更快的速度。推荐为树莓派分配 128 MB 的显存,并启用更多的 SWAP 空间。

3.6.1 使用云程序

使用云程序最大的好处就是用户只需有一个浏览器就可以在任何时候、任何平台上打开程序,而无须将程序安装到本地计算机上,也不需要考虑程序是否兼容自己所使用的平台的问题。但云程序也不是完美的,首先它需要一个稳定高速的互联网连接,其次云程序所提供的功能也不是最新的。由于技术等原因,它们通常只提供一些刚刚满足用户需求的功能。以云文字处理为例,国内目前提供云文字处理的有金山快写,国外则有 thinkfree、Google 等厂商。在浏览器中运行的金山快写,云并不支持新版本的 docx 格式,而且还需要 Flash 的支持才能运行。虽然在 X86 平台上这不

是什么问题,但是在 ARM 平台上则较为麻烦,而国外的 thinkfree 云在编辑文档或表格时需要 Java 的支持,也是较为麻烦的,如图 3-24 所示。

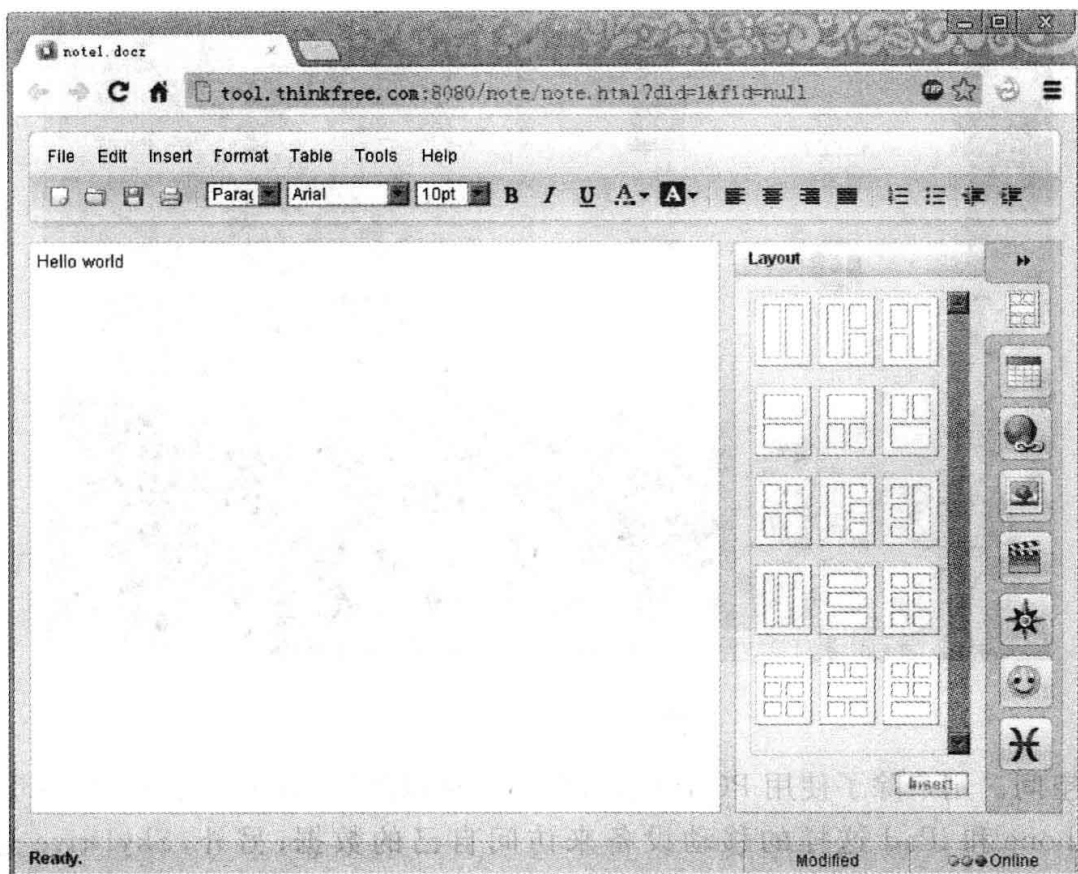


图 3-24 使用 thinkfree 云字处理程序编写 HTML 文档

对树莓派系统来说,很多的云程序都需要一个比 Debian 系统自带的 Midori 或 Dillo 更为强大的浏览器,因此用户需要安装一个浏览器以支持任何的云程序。这里推荐使用 Chromium 浏览器,它是由 Google 推出的一个开源项目,已经能很好地在树莓派下运行。

安装 chromium 浏览器,如果想要中文支持,还可以一起安装 chromium 浏览器的 chromium-110n 多语言支持包。安装命令如下:

```
# apt - get install chromium - browser chromium - 110n
```

安装完成后可以使用 chromium 浏览器来查看云程序,国内常用的云平台有 web.qq.com,如图 3-25 所示。

腾讯提供的 web.qq.com 是一个基于 WEB 页面的云平台。在这个平台中,用户不但可以登录 QQ 与好友聊天,还可以完成如文档的云保存、天气查看、新闻浏览等工作。通过它集成的云市场,用户还可以增加自己想要的应用程序,目前腾讯的云市场中涵盖了娱乐、生活、新闻、社交、办公等在内的约 5 138 个应用,足以满足用户的大部分需求。

在所有的云存储中,微软公司的 SkyDrive 是做得非常好的一个,默认就拥有 7G

第3章 使用树莓派

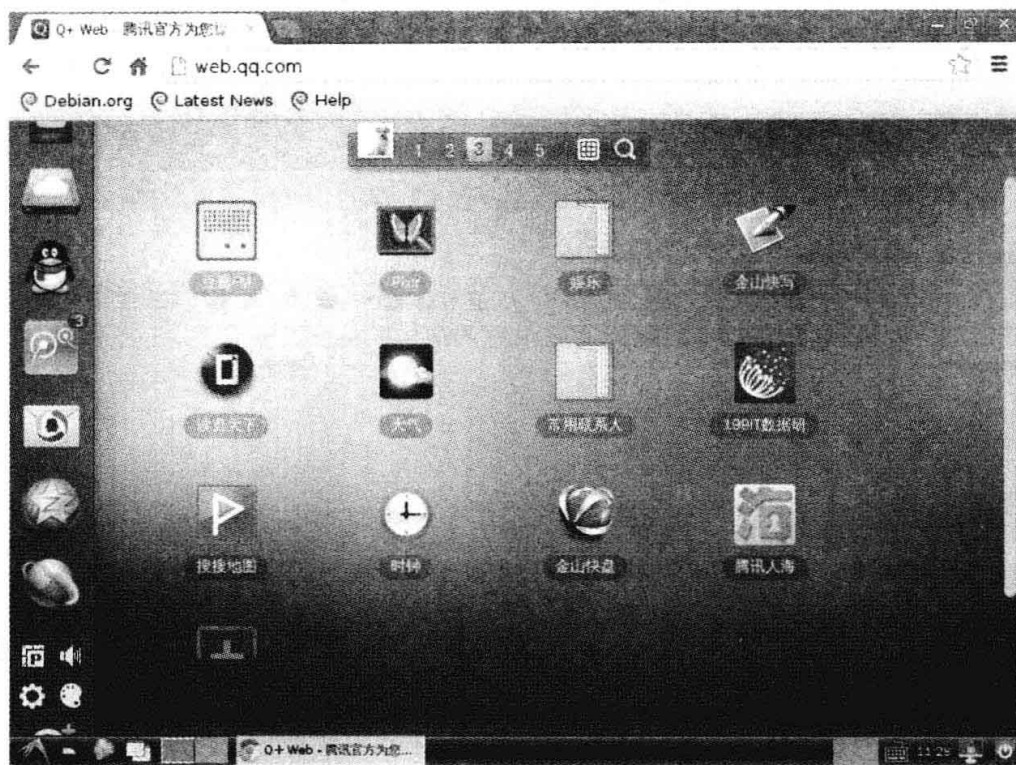


图 3-25 QQ 云界面

的存储空间。用户除了使用 PC 访问存储外,还可以使用 Android、Windows Phone,或者 iPhone 和 iPad 这样的移动设备来访问自己的数据;另外, SkyDrive 还支持 Word、Excel、PowerPoint 和 OneNote 文档编辑与查看,通过共享功能还可以使多个用户同时处理同一个文档,读者可以通过 <https://skydrive.live.com/> 进行访问,如图 3-26 所示。

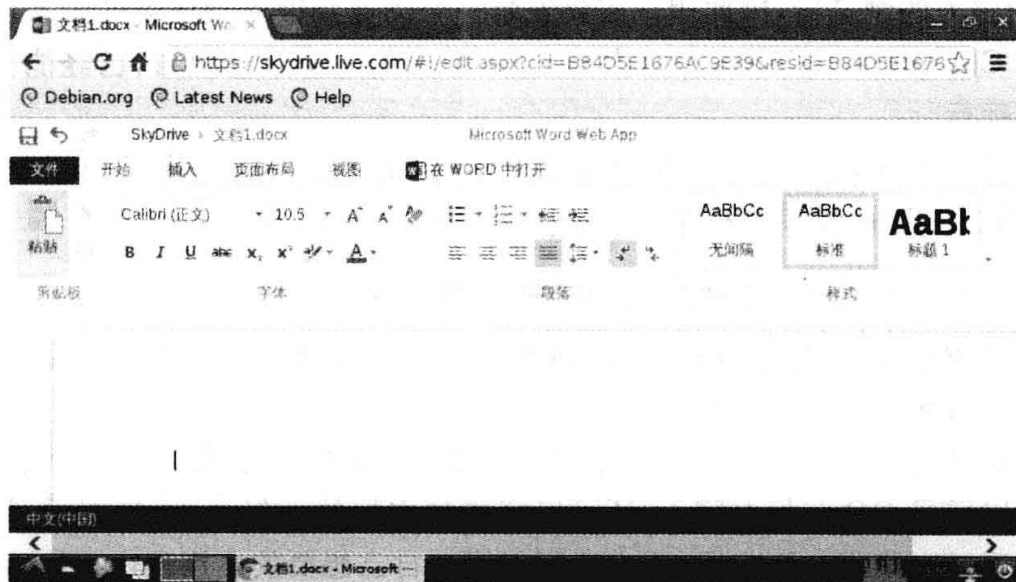


图 3-26 使用 SkyDrive 编辑文档

如果用户的要求更高,则可以使用微软公司的 365Office。它是一个付费的云平

台,提供与微软 Office 类似的桌面套件功能,以及如即时通信、日历、网站、文档共享等功能。用户可以从几乎任何一种移动设备上通过客户端程序或是 <http://office365.microsoft.com> 来访问 365Office。

3.6.2 使用 OpenOffice

如果用户长时间都无法接入互联网,那么可能就需要放弃使用云程序,转而使用本地安装的 Office 程序来完成文字或表格处理的工作。OpenOffice 是一个开源的办公软件,由于开源的特性使它能在 Windows、Linux、MacOS 等操作系统上运行。

与其他的商业 Office 办公软件相比,OpenOffice 由于使用了 GPL 版权,所以易于传播,也很容易获得源代码程序而无须担心版权问题。在用户界面和功能方面,与市场上的其他同类产品很相似,所以也有很多公司通过修改 OpenOffice 来推出自己的办公套件。例如国内的红旗中文贰仟软件技术有限公司推出的 RedOffice,就是基于 OpenOffice 修改后推出的。

目前,在树莓派上运行的 OpenOffice 其实是 OpenOffice 的 3.3 版本的一个分支,虽然软件包的名称还是 OpenOffice,但它的真实名称是 LibreOffice。用户如果觉得麻烦,可以将 LibreOffice 与 OpenOffice 理解成一个软件(事实上它们确实是同一个软件)。

在 LibreOffice 中,包含了完整的 OpenOffice 软件中的六大套件和一个 LibreOffice 主程序:

① LibreOffice: LibreOffice 的主程序,通过它可以打开或新建 LibreOffice 所支持的各种文档。

② LibreOffice Base: LibreOffice 的数据库程序,可以建立类似于微软公司的 Access 一样的数据库文档,只是它支持更多的数据库类型。

③ LibreOffice Calc: LibreOffice 的表格处理程序,类似于微软公司的 Excel,可以支持多种表格文件。

④ LibreOffice Draw: LibreOffice 的绘图程序,可以通过它来建立矢量等格式在内的图片文件。

⑤ LibreOffice Impress: LibreOffice 演示文稿,适用于制作演示时使用的幻灯片文件。

⑥ LibreOffice Writer: LibreOffice 文字处理程序,也是使用最多的程序,它支持多种格式,并能直接输出 PDF 文档。

⑦ LibreOffice math: LibreOffice 公式编辑器,可以很容易地创建和编辑科学公式、方程,并插入到其他的 LibreOffice 文档中。

在树莓派上使用 LibreOffice 时需要使用一个较大的 SD 卡,因为 LibreOffice 加上 LibreOffice 的中文包,共需要约 350 MB 的空间。这还不包含它的帮助程序、字体及推荐工具。读者可以使用 apt 程序来安装 LibreOffice 套件以及它的中文包:

第3章 使用树莓派

```
# apt - get install openoffice.org openoffice.org - l10n - zh - cn
```

安装完成后最好重新启动系统,让 OpenOffice 图标能在桌面菜单中显示出来。读者可以在左下角的菜单中的办公中找到除公式编辑器外的所有组件,或者直接在“其他”菜单下查看所有的 OpenOffice 组件;也可以点击“办公”菜单下的 LibreOffice 图标,或者直接打开 LibreOffice 的主程序。在这个程序中可以打开或建立 LibreOffice 所支持的各种文档。

由此可以看出,Writer 的界面与微软公司或金山的 Office 软件没有太大的区别,用户只需要尝试一下便可很快上手操作,如图 3-27 所示。

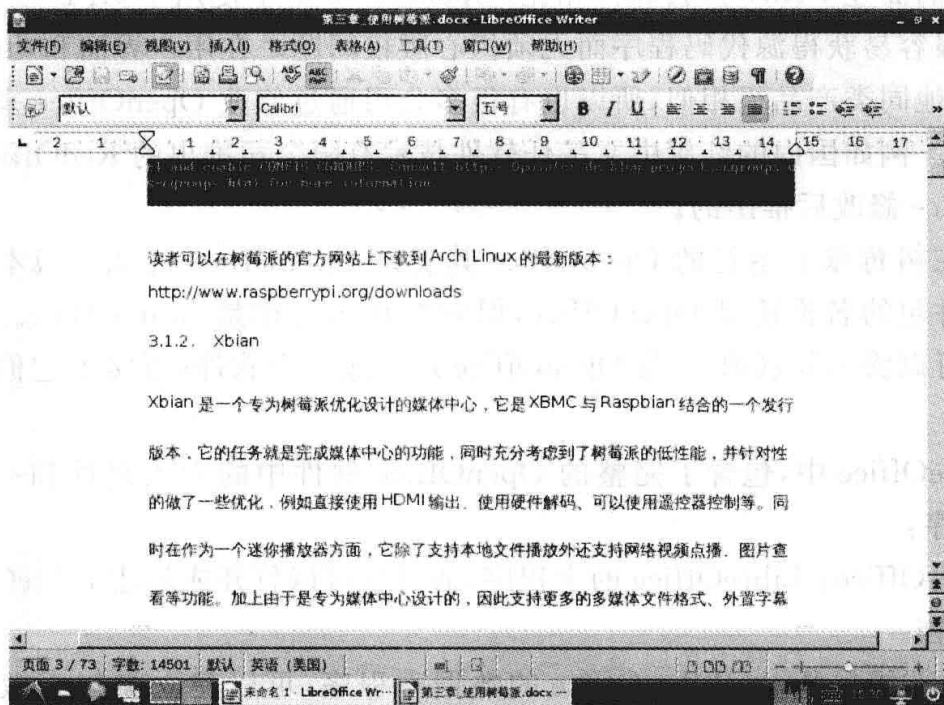


图 3-27 使用 LibreOffice Writer 程序编写文档

3.6.3 使用 Gimp 编辑图片

Gimp 是 GNU 图像处理程序(GNU Image Manipulation Program)的缩写,是一个功能超级强大的图像处理程序。在它刚刚推出还是测试版的时候,就已经被广泛应用了,例如 1997 年的电影《泰坦尼克》就使用了安装有 Gimp 程序的 Linux 工作站来完成图片处理工作。现在,Gimp 早已经推出了稳定版本,目前最新的版本是 2.8.2-2。它提供了各种图像的处理工具,而以前一些传统的功能,如快速制作网页按钮以及网站 Logo,则被更加强大的滤镜以及组件模块所取代,同时还可以直接使用绝大部分的 Photoshop 插件。这一切的功能,使图像艺术家可以尽情地发挥自己的创意,而不再受到工具软件的限制。普通用户也可以轻松地完成图像的处理工作。

由于在 Debian 发行版中默认的是没有安装 Gimp 的,所以需要通过下面的命令来安装 Gimp:


```
# apt - get install gimp gimp - help - common
```

整个安装需要约 80 MB 的空间,故首先应保证 SD 卡上有足够的空间来安装。

Gimp 的界面与通常所使用的 Photoshop 有些区别。它分为三个主界面,分别是左侧的工具箱,中间的图片编辑区,右边的通道、图层、画笔窗口。如果用户对 Gimp 不是太了解或没有接触过类似的软件,可以在 Gimp 界面中按 F1 键,可获得帮助。如果用户在安装 Gimp 时没有安装“帮助”,也没有关系,Gimp 会自动通过网络来提供一个在线的用户手册,供用户学习和了解 Gimp,如图 3-28 所示。

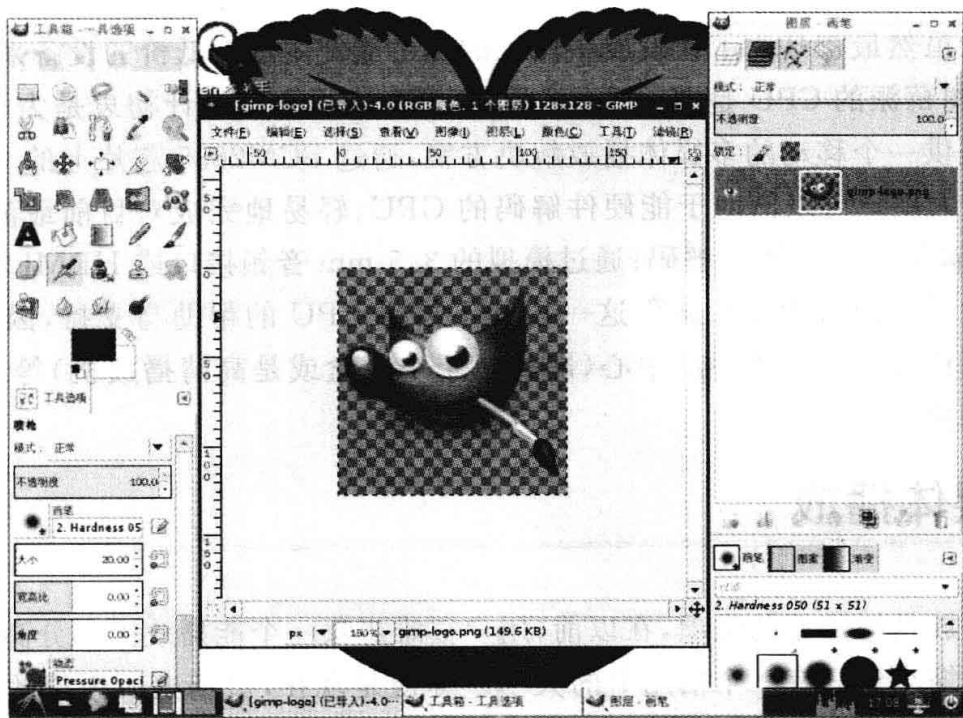


图 3-28 Gimp 工作界面

在 Gimp 中保存文件时,都会存为 Gimp 的专用 XCF 格式,因此如果需要上传到 Internet 或以其他方式与人分享图片,则需要使用 Gimp 的导出功能,将它导出为 jpg 或其他格式。

另外,由于图片编辑的复杂性,使 Gimp 在编辑图片时需要占用更多的内存,因此请尽可能地为系统分配更多的内存。

第 4 章

媒体中心与游戏

树莓派虽然最初是为了让发展中国家的学生能使用这样低价的设备来学习编程知识,但是树莓派的 CPU 是 Broadcom 的 BCM2835。它的设计初衷是为了在 HTC 的手机上提供一个移动的多媒体播放解决方案,通过 BCM2835 芯片上的 VideoCore IV 模块,广大用户可以借助于能硬件解码的 GPU,轻易地完成对目前最流行的 H.264 格式的高清视频的硬件解码;通过模拟的 3.5 mm 音频接口或 HDMI 输出,还可以播放各种类型的音频文件。而这一切都不需要 CPU 的帮助与支持,因此更多的人将树莓派作为一个媒体播放中心(即类似于小米盒或是高清播放机)等 HTPC 这样的设备来使用。

4.1 媒体播放

媒体播放是树莓派的强项,在以前,用户很难想象一个能播放大部分高清格式的播放机可以缩小到只有一个信用卡的大小。即使是现在,市场上所销售的大部分高清播放机或机顶盒之类的设备至少有 A5 纸张大小(A5 纸张的尺寸为 A4 的一半)。而配置上,专为树莓派开发与移植的媒体播放中心系统,让树莓派的用户可以获得与普通高清播放机一样的操作界面与效果。

目前,可以直接将树莓派作为家庭影院的操作系统有 Raspbmc、Xbian 和 Open-ELEC。虽然有这三种系统可以选择,但它们的核心都是一样的,均使用了 XBMC 作为媒体中心。因此本书只介绍了 Raspbmc,作为媒体中心来使用。

对于这三个系统,Raspbmc 是树莓派官方推荐的系统,功能也比另外两个系统要多一些,例如支持 LIRC、XBMC - PVR,自动调整分区,更快速的缓存和显示电影缩略图等。Xbian 则更新频繁,不过在某些软件的使用上违反了 GPL 版权;Open-ELEC 以小巧、速度快著称,一张 1G 的 SD 卡就可以安装整个系统,并能通过 sh 脚本自动更新。在 2012 年 8 月后的 OpenELEC 版本解决了大部分的速度问题,已经成为一个优秀的媒体播放系统。

4.1.1 XBMC 程序

如果希望将树莓派作为一个媒体中心来播放各种媒体文件,就不得不使用到

XBMC。XBMC 是一个非常棒的开源媒体中心软件,它最初是为 Xbox 游戏机开发的(不过目前最新版本已经不支持 Xbox 了),目前 XBMC 系统可以运行在 Linux、MacOS、Windows 等系统中。

XBMC 支持多种播放方式,通过 XBMC 可以播放本地的音频和视频文件,还可以直接播放网络媒体,加上它对各种网络媒体协议的支持,使得用户可以将自己的媒体库放在家庭网络中或者互联网上。因此,只需要指定播放位置或协议后,XBMC 会自动扫描你的媒体文件并自动建立媒体资料库。如果只有媒体文件,没有封面图片、内容介绍和海报剧照,也不要紧,通过 XBMC 的扩展插件,还可以自动为你的媒体文件下载这些介绍。

如果用户想要收看如 PPS、土豆、新浪视频等视频网站的内容,也可以通过网络的支持或者使用 Python 语言来对 XBMC 进行扩展,用户可以轻松地通过 XBMC 来实现这些网站的网络媒体的播放。而内置的天气预报功能与许多音频视觉效果,加上列表和幻灯片的图片播放功能,可以让树莓派完美地实现你的 mini HTPC 梦想。

之前介绍过的 XBMC 可以运行在多种平台上,如果用户想要先尝试一下 XBMC 系统后,再决定是否在树莓派上运行它,那么可以通过以下网站来下载 XBMC 的其他平台版本:

- XBMC 官方网站 <http://xbmc.org/>
- XBMC 下载地址 <http://xbmc.org/download/>

XBMC 在各个平台上的默认界面都是一样的,用户可以快速上手并使用它。XBMC 目前可以播放以下格式的媒体与协议:

① 支持物理媒体: Blu-ray Disc (unencrypted)、CDs、DVDs、DVD - Video、Video CDs (包括 VCD/SVCD/XVCD)、Audio - CD (CDDA)、USB Flash Drives、local Hard Disk Drives。

② 支持网络客户协议: AirPlay/AirTunes、UPnP / DLNA (Digital Living Network Alliance)、SMB/SAMBA/CIFS、AFP、DAAP、Zeroconf/Avahi/Bonjour、NFS、HTTP、HTTPS、FTP、RTSP (RTSPU、RTSPT)、MMS (MMSU、MMST)、Podcasting、TCP、UDP、SFTP、RTP、RTMP (包括 RTMP、RTMPT、RTMPE、RTMPTE、RTMPS)、DHCP、NTP。

③ 支持网络服务器协议: JSON - RPC Server、D - Bus Server、Web Server、FTP Server、UPnP AV media Server、a multi - protocol Event Server。

④ 流媒体格式: AVI、MPEG、WMV、ASF、FLV、MKV/MKA (Matroska)、QuickTime、MP4、M4A、AAC、NUT、Ogg、OGM、RealMedia RAM/RM/RV/RA/RMVB、3gp、VIVO、PVA、NUV、NSV、NSA、FLI、FLC、DVR - MSand WTV。

⑤ 视频格式: MPEG - 1、MPEG - 2、H. 263、MPEG - 4 SP/ASP、MPEG - 4 AVC (H. 264)、HuffYUV、Indeo、MJPEG、RealVideo、RMVB、Sorenson、WMV、Cinepak。

第4章 媒体中心与游戏

⑥ 音频格式: MIDI、AIFF、WAV/WAVE、AIFF、MP2、MP3、AAC、AACplus (AAC+)、Vorbis、AC3、DTS、ALAC、AMR、FLAC、Monkey's Audio (APE)、Real-Audio、SHN、WavPack、MPC/Musepack/Mpeg+、Shorten、Speex、WMA、IT、S3M、MOD (Amiga Module)、XM、NSF (NES Sound Format)、SPC (SNES)、GYM (Genesis)、SID (Commodore 64)、Adlib、YM (Atari ST)、ADPCM (Nintendo GameCube)、CD-DA。

⑦ 数字图像格式: RAW Image formats、BMP、JPEG、GIF、PNG、TIFF、MNG、ICO、PCX、Targa/TGA。

⑧ 字幕格式: AQTitle、ASS/SSA、CC、JACOsub、MicroDVD、MPsub、OGM、PJS、RT、SMI、SRT、SUB、VOBsub、VPlayer。

⑨ 元数据标签: APEv1、APEv2、ID3 (ID3v1 和 ID3v2)、ID666、Vorbis comments for audio file formats、Exif、IPTC (包括 GeoTagging) for image file formats。

4.1.2 Raspbmc

之前说过,在树莓派上使用的媒体播放中心都是 XBMC 软件,Raspbmc 也不例外。Raspbmc 是树莓派官方推荐的媒体播放中心系统,能完成大部分的多媒体播放,可以在 Raspbmc 的网站 www.raspbmc.com 上下载它的最新版本。

1. 安装 Raspbmc

Raspbmc 提供了多种安装方式,其中,光镜像文件就提供了两个:基础镜像和完整镜像。基础镜像文件包含了 Raspbmc 的基本程序,用户需要在第一次启动 Raspbmc 后联通网络,进行下载更新;否则无法使用。其优点是镜像文件小,容易下载,但是在第一次开机后会通过网络下载最新的系统文件。由于更新服务器在国外,整个更新需要下载约 180M 的数据,因此在网络条件不好的情况下,更新速度非常慢。

完整镜像包含了 Raspbmc 的整个内容,无需连网更新系统,优点是安装完后就可以使用,缺点是它不是最新的版本,但下载后只需要安装到 SD 卡中就可以使用。

这两个版本可以在 Raspbmc 的 <http://www.raspbmc.com/download/> 页面上下载。选择下载 Network Image 与 Standalone Image 镜像即可。对于下载的镜像文件,解压后都是扩展名为 img 的镜像文件,读者可以参阅 3.2 节。

除了使用镜像文件进行安装以外,Raspbmc 还提供了基于 Windows 系统的安装程序:Raspbmc Installer。读者可以在 <http://www.raspbmc.com/wiki/user/windows-installation/> 上下载到它的最新版本。

将下载的文件解压后会得到一个 installer.exe 的可执行文件,这就是 Raspbmc 的安装程序。每次运行它时都需要点击 Accept 按钮来接受警告与授权,最后读者可以看到如图 4-1 所示的安装界面。

通过读卡器将 SD 卡连接到计算机的 USB 口以后,窗口中将会显示出设备的 ID、卷名及设备名称。如果没有出现相应的设备,可以点击边上绿色的刷新按钮。

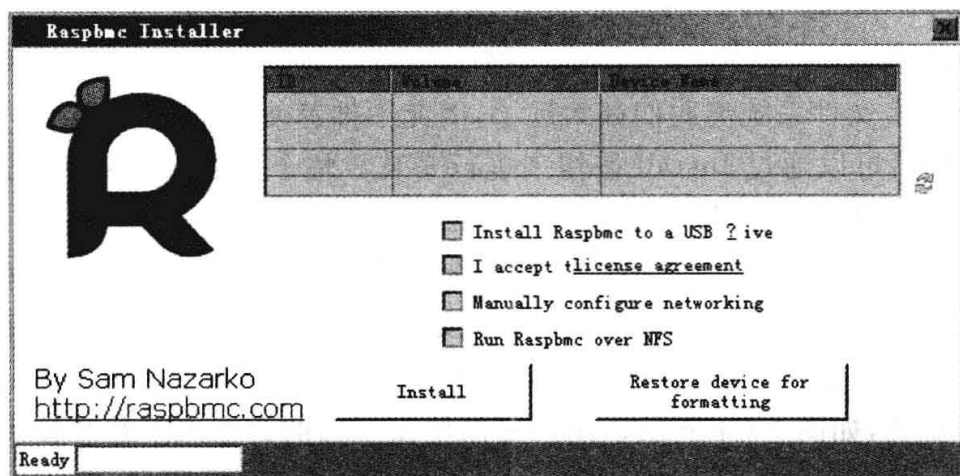


图 4-1 Raspbmc Installer 界面

在 Raspbmc Installer 程序中有四个选项,分别是:

① Install Raspbmc to a USB Live:安装 Raspbmc 到 U 盘这样的 USB 设备中使用,但启动还是需要 SD 卡的。

② I accept tLicense agreement:是否接受授权,如果不接受就无法安装 Raspbmc。

③ Manually configure networking:配置网络,与使用镜像安装不同的是 Raspbmc Installer。程序可以自定义 Raspbmc 系统的网络配置,这是一个非常有用的选项。读者只要选择 Manually configure networking 选项就可以打开网络配置窗口,读者可以在图 4-2 与图 4-3 中配置网络。

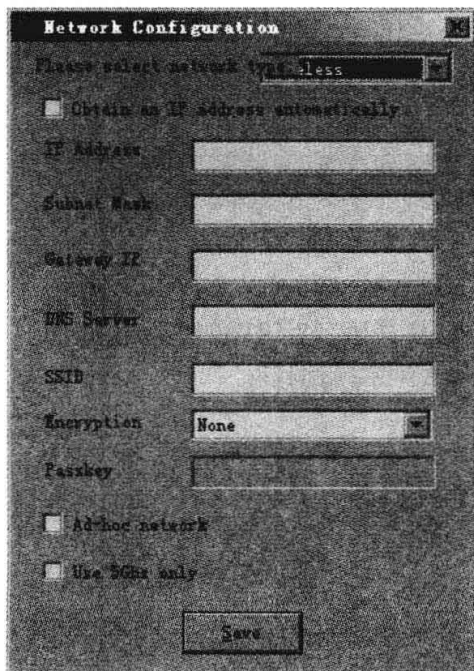


图 4-2 有线网络配置界面

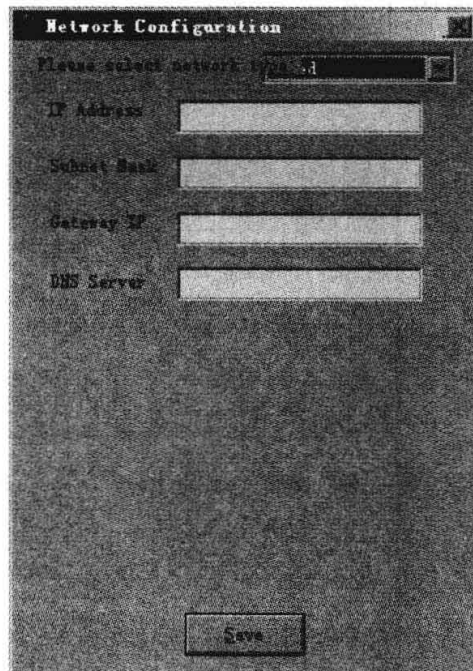


图 4-3 无线网络配置界面

在网络配置中,可以通过选择网络接口类型来选择配置有线网络接口与无线网络配置。

第4章 媒体中心与游戏

④ Run Raspbmc over NFS: 是否在 Raspbmc 上运行 NFS(网络文件)系统。NFS 是 Linux 系统中一个共享文件的网络文件系统,对于大多数的国内用户来说,NFS 格式的网络文件系统离我们都太远了,因此不推荐使用。

选择完成后可以通过 Install 按钮来进行安装。整个安装过程视网络速度而定,对于国内用户来说,这可能是一个很长的过程。

如果用户想要恢复 U 盘的空间到 Windows 系统下使用,也可以通过 Raspbmc Installer 程序中的 Restore device for formatting 按钮来格式化 U 盘为 FAT32 格式。

对于 Linux 和 MAC 系统的用户而言,Raspbmc 提供了一个 Python 脚本用于 Raspbmc 的安装,如图 4-4 所示。用户可以使用下面的命令来下载和使用:

(1) Linux

下载安装脚本:

```
# wget http://svn.stmlabs.com/svn/raspbmc/testing/installers/python/install.py
```

将脚本设置成可执行程序并运行它:

```
# chmod +x install.py && install.py
```

(2) OSX

下载安装脚本:

```
curl -O http://svn.stmlabs.com/svn/raspbmc/testing/installers/python/install.py
```

将脚本设置成可执行程序并运行它:

```
# chmod +x install.py && install.py
```

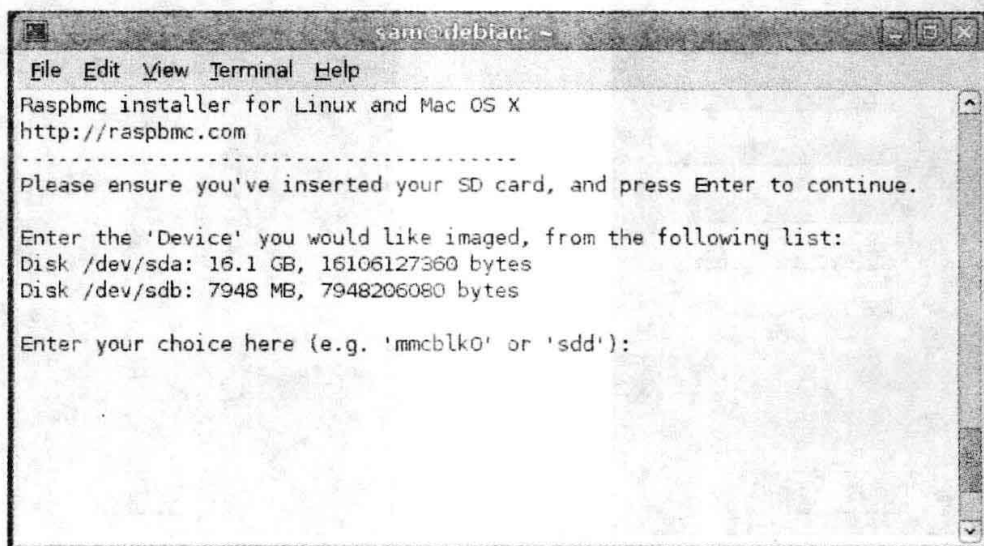


图 4-4 Linux 与 MAC 脚本安装

Install.py 脚本提示用户选择哪一个设备来安装 raspbmc 系统,并下载最新的镜像文件来安装。

2. 配置 Raspbmc

初次启动 Raspbmc 时,系统会自动完成一些配置工作,如:调整分区大小为整个 SD 卡大小、连接网络更新系统,等等。在这期间,Raspbmc 还会因为配置变更而重新启动一次。整个更新过程需要一些时间。当第一次进入到 Raspbmc 的 XBMC 界面中时一般会提供选择一种语言,用户只要选择简体中文即可看到整个 Raspbmc 的界面变成中文。

如果语言选择错误或是不小心按 Esc 键跳过了选择,也没有关系,通过“系统”→“设置”→“用户界面”→“界面语言”→“语言选项”也可以将语言设置为中文,而对于简体中文字幕则可以通过指定字符集为 GBK 编码来正确显示。

Raspbmc 的配置可以分成两个部分。一个是 Raspbmc 系统的设置,它是由菜单中的程序选项下属的 Raspbmc Settings 选项来进行设置的。在这个菜单中,可以对 Raspbmc 的配置进行设置,整个设置被分成四大部分,分别是网络设置、每天晚上更新、系统设置、红外遥控器设置,如图 4-5 所示。

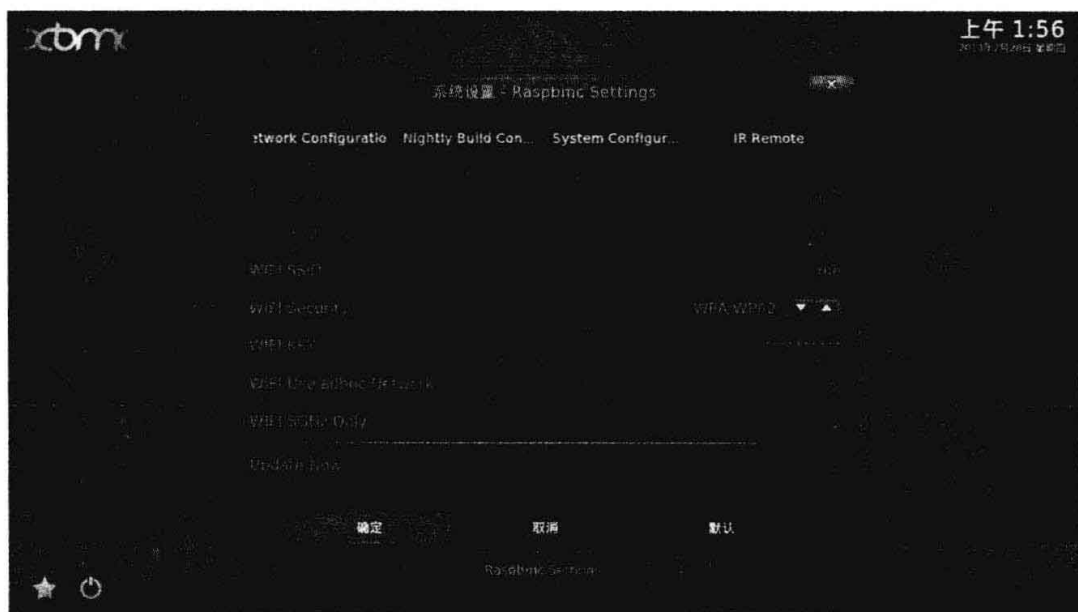


图 4-5 XBMC Raspbmc 配置

在这四个部分中,最常用的就是网络设置(network configuration)。用户只需在树莓派上插上所支持的无线网卡,并且在无线配置中设置好 WIFI SSID 与密码,Raspbmc 就会自动进行连接并获得 IP 地址等信息。

在系统配置中,可进行密码、系统更新、超频、服务管理、高级系统等方面的设置:

Password Management

密码管理

Enter password

输入密码

Confirm password

再次输入密码

Set password

设置密码

System Upgrades

系统更新

Keep Raspbmc updated	保持 Raspbmc 更新
Remove UI res limit (read FAQ)	移除用户界面限制
Enable Audio Engine	启用音频引擎
(be aware of very rough edges)	启用音效
Disable unsafe shut down warning	关闭不安全的警告
Advanced system settings	高级系统设置
MPEG2 codec license	输入 MPEG2 编解码器许可
VC1 codes key	输入 VC1 编解码器许可
Disable overscan	禁用分辨率扫描
System Performance Profile	系统性能配置
Advanced overclocking	高级超频
CPU overclock	CPU 超频
Core overclock	核心超频
Image Sensor Pipe overclock	图像传感器管超频
GPU overclock	GPU 超频
RAM overclock	RAM 超频
Over Voltage	电压设置
Additional config. txt settings	增加额外设置到 config. txt 中
Service Management	服务管理
FTP Server	FTP 服务器
SSH Server	SSH 服务器
Samba Server	Samba 服务器
Remote Control	遥控器设置
Cronjob scheduler	Cron 设置
TVHeadend Server	TVHeadend 服务器

事实上,在整个系统设置中,Raspbmc 使用默认的设置就已经可以工作得很好。Raspbmc 会自动更新系统,并超频到 800 Hz,而服务器软件方面默认开放了 SSH、FTP、SAMBA。当然,如果用户不需要传输文件则可关闭 FTP 服务器,如果不通过 Windows 网络来查看文件也可以关掉 Samba 服务器。如果用户购买了 VC-1 或是 MPGE2 的授权码,还可以在高级系统设置中 VC-1 或是 MPGE2 的位置输入授权码,以获得硬件解码这两种格式的能力。

每天晚上更新(Nightly build configuration)可以 Install XBMC cightly(手动安装更新最新的系统)、Switch etween versions of XBMC(切换 XBMC 系统的版本)或是 Factory Reset(复出厂设置)。对于红外遥控器的设置及大多数用户来说,保持默认的设置就可以了。

另一个配置则是 XBMC 本身的设置,用户可以通过“程序”→“系统设置”来进行

设置,如图4-6所示。



图 4-6 XBMC 系统设置

在前面的 Raspbmc 设置中,主要针对的是系统方面的设置,而在 XBMC 中的这个设置则主要针对 XBMC 系统的一些细节。例如在视频设置中,用户可以修改视频资料库的信息、字幕的大小、播放比例、是否自动播放下一节等内容。如果需要使用不同的皮肤,也可以在用户界面中设置。

3. 播放多媒体

对于图片、视频、音乐这三种常见的媒体,读者可以直接使用相应的菜单功能来进行播放,唯一需要注意的是这些媒体文件存放的位置。如果使用的是 U 盘或是移动硬盘之类的本地设备,Raspbmc 会自动加载它们,并在浏览目录的窗口中显示出来。对于局域网上存放的媒体,读者可以通过 UPNP 和 Windows 网络进行访问。

对于大多数的视频文件来说,通过树莓派来播放都不是问题,因为它支持硬件解码,所以包括 H264 等在内的格式的高清文件是完全没有问题的。图 4-7 就是使用树莓派来播放高清电影的效果。如果购买了 VC-1 和 MPEG2 的硬件解码功能并启动了它们,那么对于一些 WMV 或者 DVD 之类的视频格式来说也不是问题。用户通过视频菜单就可以播放,但是对于一些像 RMVB 之类的完全靠 CPU 解码的视频格式就非常困难了。

音频解码方面,常见的格式都没有问题。唯一有卡顿现象的就是 DTS 和 DTS-HD。目前这两种格式还在依靠软件解码,因此十分消耗 CPU 资源。如果用户播放带 DTS 音频的视频的时候出现卡顿现象,可以考虑对 CPU 进行小幅度超频,如超频到 900 MHz。目前官方已经允许适当超频了。如果超频无法稳定工作,可以考虑增加输入电源或者适当提高 CPU 的核心电压;但是超频不能将频率提升得太高,否则会烧毁整个 CPU。

网络上的媒体文件分为两种。一种是由本地网络提供的,这些媒体文件通常存放在如 Windows 网络、UPnP、NFS 甚至是 SSH 服务器上。对于这些媒体,在

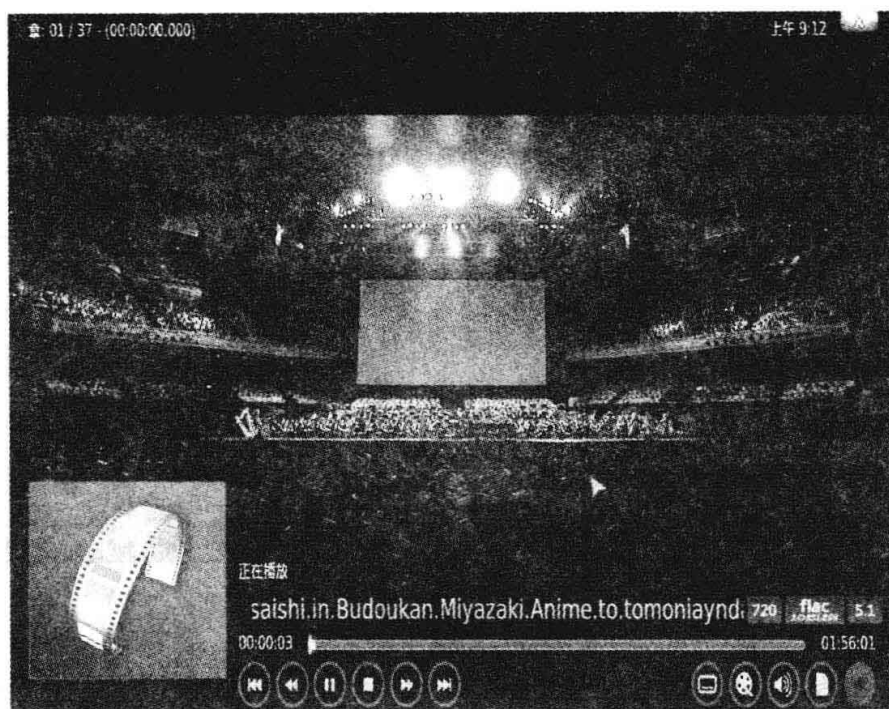


图 4-7 使用 XBMC 播放高清电影《久石让在武道馆》

Raspbmc 中可以直接选择相应的网络协议来进行处理,如图 4-8 所示,例如在添加视频、目录等选项中选择“添加网上邻居”选项(推荐使用“添加网上邻居”选项来进行添加,因为在这个选项中可以通过协议来选择列表所没有列出的网络协议)。在出现的窗口中,选择“协议”来检索需要使用的协议类型。

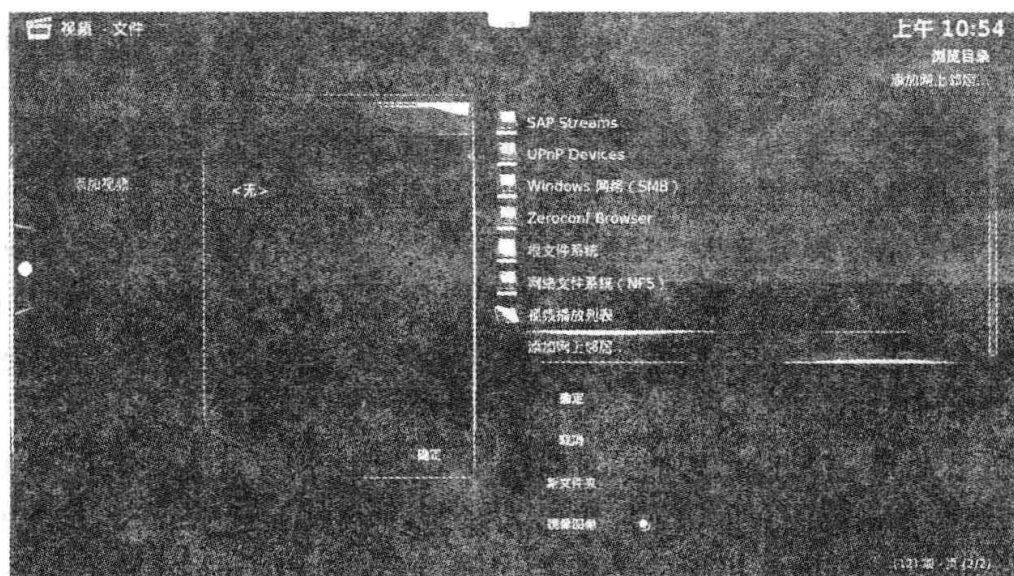


图 4-8 添加网络媒体

目前在网上邻居中所支持的协议有:

- Windows 网络(SMB)
- Apple 文件协议(AFP)

- 安装壳(SSH/SFTP)
- 网络文件系统(NFS)
- RSS 信源
- UPnP 媒体服务器
- iTunes 音乐共享(DAAP)
- FTP 服务器
- WebDAV 服务器(HTTP)
- WebDAV 服务器(HTTPS)
- Web 服务器目录(HTTP)
- Web 服务器目录(HTTPS)
- TuxBox 客户端
- MythTV 客户端
- VDR Streamdev 客户端
- HST Tvheadend 客户端

另一种媒体则是由网站提供的,例如新浪视频、土豆网、迅雷看看、优酷及酷我音乐盒等。如果要查看这些媒体,则需要安装相应的网站插件。用户可以在 <http://code.google.com/p/xbmc-addons-chinese/downloads/list> 上下载 repository.googlecode.com/xbmc-addons-chinese-edn.zip 这个插件合集。在这个合集中包含了以下插件。

① 天气插件:

weather, weatherchina 中国天气(Weather China)

② 视频插件:

plugin.video.kankan 迅雷看看(KanKan)

plugin.video.pptv PPTV 视频

plugin.video.kugoumv 酷狗 MV(KugouMV)

plugin.video.soptv SOP 电视(soptv)

plugin.video.letv 乐视网(LeTV)

plugin.video.tv189 天翼视讯(TV189)

plugin.video.netitv 天翼高清(NETITV)

plugin.video.tv365 TV365

plugin.video.youku 优酷视频(YouKu)

plugin.video.ppstream PPS 网络电视(PPStream)

plugin.video.qvod 快播视频(Qvod)

plugin.video.cntv 中国网络电视台点播(CNTV)

plugin.video.cntv-live 中国网络电视台直播(CNTV Live)

plugin.video.sohuvideo 搜狐视频(SoHu)

plugin.video.somovie 搜索电影(SoMovie)
 plugin.video.yinyuetai 音悦台 MV(YinYueTai)
 plugin.video.qiyi 奇艺视频(QIYI)
 plugin.video.openv 天线高清(OPENV)
 plugin.video.tudou 土豆视频(Tudou)
 plugin.video.tencent 腾讯视频(Tecent)
 plugin.video.ku6 酷6云中剧场(Ku6)
 plugin.video.wzipad 热点影院(wzipad)
 plugin.video.sina 新浪视频(Sina)
 plugin.video.5ivdo 5ivdo

③ 程序插件:

plugin.program.mystocks MyStocks/中国股市行情插件
 plugin.program.utorrent uTorrent 下载管理插件
 RSSreader Rss 阅读插件(暂未加入扩展库)

④ 音频插件:

plugin.audio.kuwobox KuwoBox/酷我音乐盒插件
 plugin.audio.baiduradio BaiduRadio/百度电台联盟插件
 plugin.audio.top100 TOP100/巨鲸音乐网插件

⑤ 脚本:

script.lrclyrics-LRC Lyrics LRC 同步歌词脚本
 script.shootersub shooterSub 射手字幕脚本

安装插件有很多方法,可以通过 U 盘下载后安装,也可以使用 SSH 方式下载。由于 Raspbmc 默认就开启了 SSH 服务,因此最简单的方法就是通过 SSH(SSh 连接方式可以参见第 3 章 SSH 部分)客户端直接连接到 Raspbmc 中(Raspbmc 默认的用户名是 pi,密码为 raspberry),然后使用 wget 命令下载 repository.googlecode.xbmc-addons-chinese-edn.zip 文件。最后在扩展管理中使用 zip 安装方式进行安装:

```
# cd /home/pi
# wget https://xbmc-addons-chinese.googlecode.com/files/repository.googlecode.xbmc-addons-chinese-edn.zip
```

转到 Raspbmc 中,选择“系统设置”→“扩展功能”→“从 zip 文件安装”,选择安装的位置为/home/pi 目录,可以看到 repository.googlecode.xbmc-addons-chinese-edn.zip 文件,选择它,并单击“确定”按钮,即可开始安装,如图 4-9 所示。

目前在 zip 安装过程中是没有进度显示的,安装时间并不长,随后在“视频”→“视频扩展功能”→“获取更多”等选项中可以看到非常多的视频网站支持。图 4-10 是在扩展功能中安装新浪视频插件。

选择自己要看的网站,后安装它。随后在相应的扩展功能中就可以通过该网站

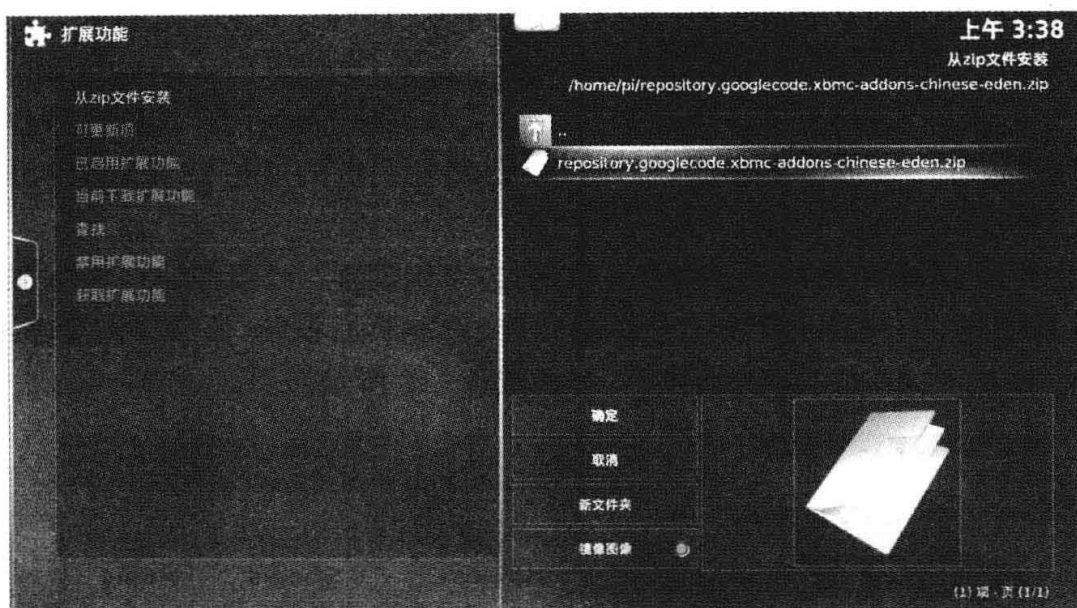


图 4-9 选择本地安装文件

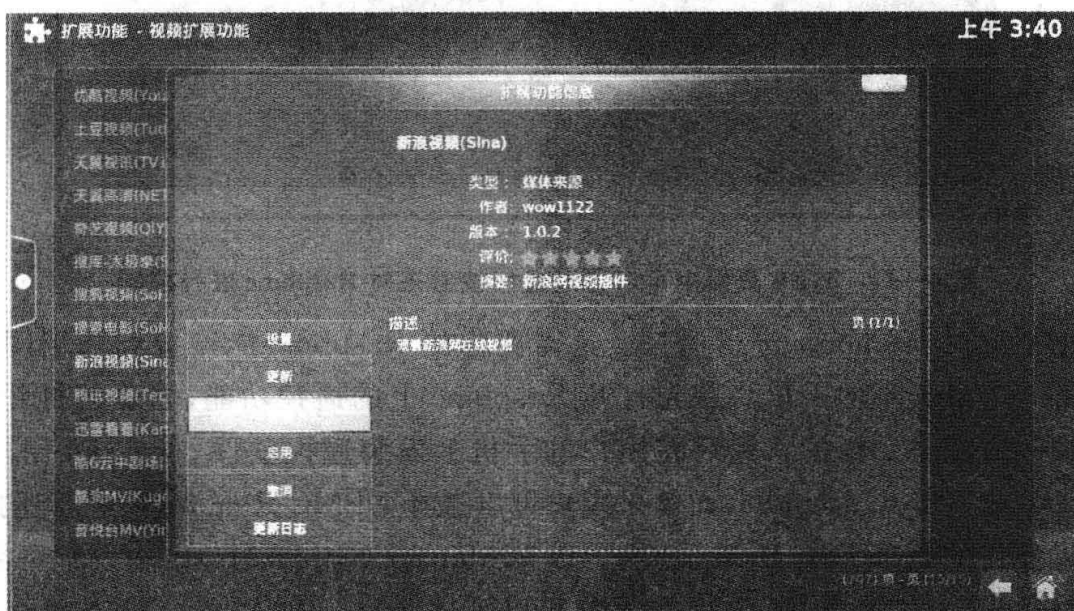


图 4-10 安装新浪视频插件

来查看视频,对于音乐、图片也是如此,并可在相应的扩展功能中进行查看。

4.1.3 使用遥控器

在 Raspbmc 中使用的遥控器分为四种:①通过电视机自带的遥控器进行遥控;②通过手机进行遥控;③使用 MCE 遥控器;④通过 WEB 页面进行控制。

使用电视机遥控器对 Raspbmc 进行遥控。Raspbmc 在 2012 年 6 月以后的版本都加入了 HDMI-CEC 控制功能,读者通过 HDMI 线将遥控器的信号从电视传到 Raspbmc 上并控制它。这需要你的电视机能支持 HDMI-CEC 控制,并且使用的 HDMI 线也支持遥控(不可以是较低版本的 HDMI 线或是偷工减料的线)。

第4章 媒体中心与游戏

在 Raspbmc 中, HDMI - CEC 控制功能是默认打开的, 如果没有打开, 可以选择 System→Settings→Services→Remote control, 然后选择 'Allow programs on this system to control XBMC' 来打开遥控功能。另外, 电视机有的 HDMI 控制选项也要设置成打开状态。一切设置完成后, 即可以使用电视机的遥控器对 Raspbmc 进行控制。

通过 WEB 页面进行控制。用户可以通过“系统设置”→“服务”→“Web 服务器”→“允许通过 HTTP 控制 XBMC”选项来开启 WEB 页面控制, 然后在浏览器中输入 Raspbmc 的 IP 地址, 即可打开遥控页面, 如图 4-11 所示。

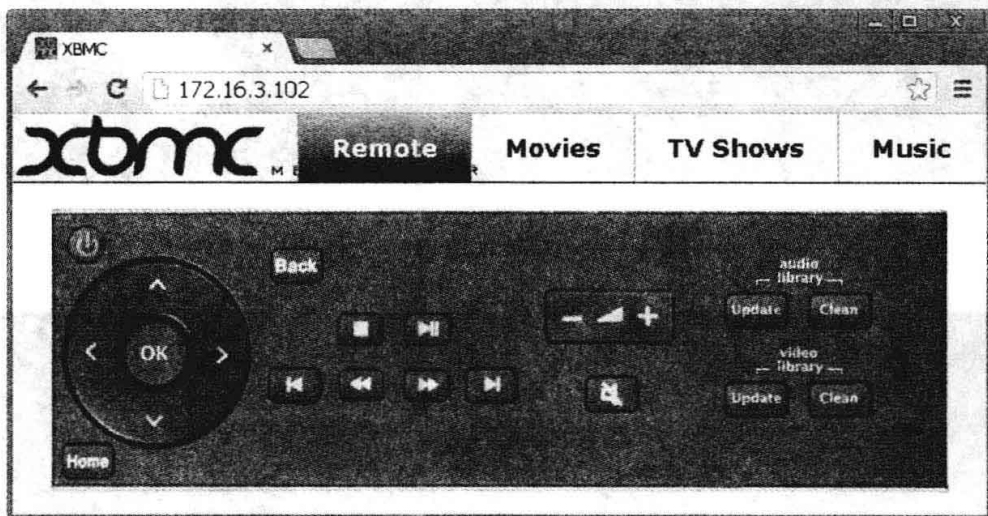


图 4-11 WEB 页面中可以通过虚拟按键来对 Raspbmc 进行控制

使用手机对 Raspbmc 进行遥控。这需要你的 Raspbmc 能连接到局域网中, 并且有一部苹果或安卓的手机。首先打开 Raspbmc 上的遥控功能, 方法与打开 WEB 页面遥控的方法是一样的。然后在用户的手机上安装遥控软件, Android 用户可以选择 Yatse 或 XBMC Remote, 而苹果用户可以选择 XBMC Remote 来进行遥控。

如果使用的是 XBMC Remote, 则可以在 Manage XBMC Hosts 中设置所需要连接的 Raspbmc 主机信息, 通常只需要填写以下三个信息即可以连接:

- Display name: 显示的连接名称。
- Host IP: Raspbmc 主机的 IP 地址。
- Api Port: 连接的端口号, 默认为 8080, 但是 XBMC 通常使用的是 80 号端口。

设置完成后, 读者通过手机上显示的遥控器界面进行操作, 如图 4-12 所示。

如果使用的是 Yatse, 则 Yatse 会自动搜索 XBMC 主机并连接; 如果 Yatse 没有搜索到, 也可以使用 MEDIA CENTERS 来设置需要连接的主机信息。

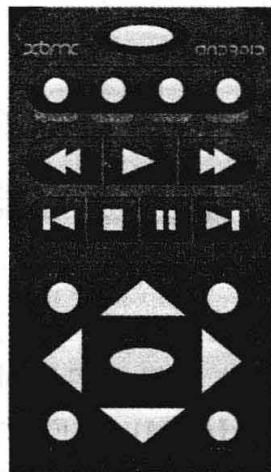


图 4-12 XBMC Remote 遥控器截图

整个设置所需要填写的信息与 XBMC Remote 类似。

使用 MCE 遥控器进行控制时需要额外购买一个 MCE 遥控器,目前 Raspbmc 支持大部分的 MCE 遥控器,用户只需要将 MCE 的接收器与 USB 连接就可以正常使用了。而 MCE 遥控器推荐使用遥酷 yaocoo 的 MCE 遥控器。

这四种方法中,电视机自带的遥控器进行遥控最为简单,只要电视和 HDMI 线没有问题就可以成功操作;手机遥控则需要网络支持,并且还得是智能手机;MCE 遥控器虽然接上就可以用,但是需要额外购买;WEB 页面控制,可以让用户通过网络来远距离控制 XBMC。

4.1.4 音乐播放中心

前面介绍的 Raspbmc 程序可以完成一个媒体中心所需要的所有功能,但是对于那些无法使用图形界面的用户来说,由于长时间工作在命令行模式下,自然也就无法使用图形界面的音乐播放工具了,而 MOC(Music On Console)就是为了解决命令行模式下的音乐播放问题而专门设计的命令行音乐播放器。

遵循 Linux 软件设计的思想, MOC 整个软件非常简洁、小巧, 却能完成 mp3、ogg、flac、musepack、speex、wave、aiff、au 等音乐格式的播放, 甚至用户还可以使用 MOC 来播放 Internet 上的流媒体格式。

安装与运行 MOC:

```
# apt - get install moc
# mocp
```

与其他程序一样,MOC 也在 Debian 软件仓库中,用户可以使用 apt-get 程序来进行安装;启动 MOC 则只需要在命令行中输入 mocp,然后回车即可,如图 4-13 所示。

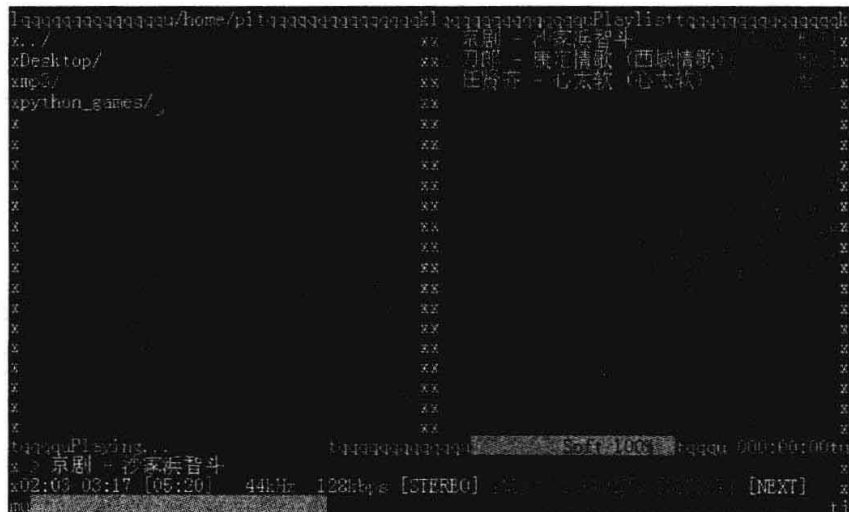


图 4-13 使用 MOCP 来播放音乐

整个界面分为左、右两部分,左边是目录窗口,可以通过上、下光标来进行移动并通过回车进行确认或是播放,而右边则是播放列表。用户在左边的目录上按下“a”键,则目录中所有的音乐文件都添加到播放列表中;如果要删除播放列表,则使用大写的“C”键;如果要在命令行中进行其他操作,还可以按下小写的“q”键,可切换回命令行界面,而无须担心音乐的播放中断,更多的组合键说明可以参见表4-1。

表4-1 mcp 组合键说明

组合键	说 明	组合键	说 明
q	离开 mcpd 播放器,但并不中断音乐播放	R	切换重复
ENTER	播放文件或进入目录	X	打开/关闭下一首选项切换
DOWN	向下移动	TAB	TAB 切换播放列表和文件列表
UP	向上移动	l	显示播放列表
PAGE_DOWN	移动到下个页面	a	添加文件或目录到播放列表中
PAGE_UP	移动到上个页面	C	清除播放列表
HOME	移动到菜单中的第一项	A	归递添加文件或目录到播放列表中
END	移动到菜单中的最后一项	Y	删除播放列表中不存在的媒体文件记录
Q	退出	<	1%的减少量
s	停止	>	1%的增加量
n	播放下一个文件	,	减少量 5%
b	播放上一个文件	.	由 5%提高音量
SPACE	暂停	RIGHT	快进
f	切换 ReadTags 选项	LEFT	后退
S	开启随机播放		

4.2 游 戏

得益于 GPU 的硬件性能以及 Java 强大的可移植性,再加上 BCM2835 还不算弱的性能,使得树莓派在游戏上的表现犹如一个哑铃一样,一边可以通过硬件 GPU 解码来运行如 Quake 3(《雷神之锤 III》)这样的大型第一人称射击游戏,而另一边则通过 CPU 运算来运行一些 Java 游戏,如 Minecraft(《我的世界》)这样的轻量级游戏。而这个哑铃中间的部分则表现得差强人意,真正勉强可玩的也就是一些街机模拟器游戏。因此建议用户不要对游戏要求太高。

本节介绍的游戏除了《我的世界》没有在 Debian 软件仓库中提供外,其他两个游

戏 Quake 3 与街机模拟器都可以直接通过 apt-get 程序来进行安装,但是这两个软件包都需要额外的设置。因此,下面介绍的 Minecraft 和 Quake 3 两个游戏都是通过网络上提供的安装包来进行安装的,用户只需解压后就可以直接使用。

4.2.1 Minecraft

Minecraft 是一款 3D 的第一人称沙盘游戏,整个游戏所有场景都是以方块方式显示的,因此它追求的并不是华丽的画面与特效,而是游戏的可玩性。玩家在游戏中只有两件事可以做:建设、破坏。玩家可以花上几分钟建起一个小屋,也可以用数天时间来建造一个属于你自己的壮丽的城堡;或者使用破坏工具,在地下破坏出一个地下宫殿。只要玩家能想到的,都可以在《我的世界》中实现,而这一切都需要大量的时间,因此这这也是一个消磨时间的好游戏。读者可以在图 4-14 中看到运行效果。



图 4-14 《我的世界》运行效果

与原版的 Minecraft 有一些不同的是,树莓派版本的 Minecraft 有一些小小的修改,并且功能上也有一些增减。不过没有关系,这些并不会影响到玩家的热情。关于 Minecraft 的最新版本与消息可以通过 Minecraft 的树莓派版网站 <http://pi.minecraft.net/> 来获得。

Minecraft 的安装十分简单,下载 Minecraft,然后解压、运行,就可以开始玩了。

① 下载 Minecraft:

```
# wget https://s3.amazonaws.com/assets.minecraft.net/pi/minecraft - pi - 0.1.1.tar.gz
```

② 解压:

```
# tar xf minecraft - pi - 0.1.1.tar.gz
```

③ 进入到 Minecraft 目录并运行它:

```
# cd mcpi && ./ minecraft - pi
```

第4章 媒体中心与游戏

这款游戏让您体验建立一个属于自己的世界,游戏的教程可以在《我的世界》官网上查找。

4.2.2 Quake 3

相信每一个玩计算机游戏的人都应该听过《雷神之锤Ⅲ》吧。这款经典的多人连线第一人称射击游戏(FPS)是由 id Software 采用 id Tech3 引擎制作的,当年它的出现可以说是红遍全球,而 id Tech3 引擎从诞生起一直到 6 年后还被一些主流游戏所采用。虽然好游戏不断出现,但是在今天看来它还是具有非常好的可玩性,如图 4-15 所示。

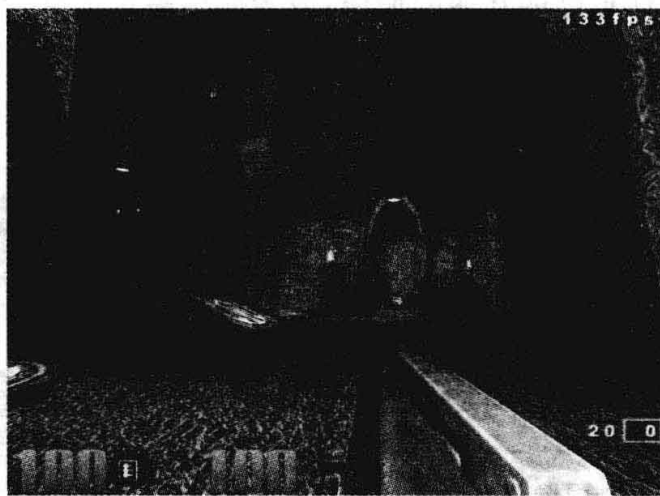


图 4-15 Quake 3 运行效果

《雷神之锤Ⅲ》几乎可以在目前所有的 Linux 平台上运行,树莓派也不例外,通过硬件 GPU 加速功能,树莓派可以轻松地运行《雷神之锤Ⅲ》。相信那些喜欢第一人称射击游戏的树莓派用户不会失望了。

在树莓派上运行《雷神之锤Ⅲ》有多种安装方法,除了 Debian 系统软件仓库内的 Quake 3 软件包安装外,还可以通过源代码自行编译 Quake 3,但是无论使用哪一种方法,都只能安装好程序。对于 Quake 3,运行所需要的数据包是无法安装的,只能通过 Quake 3 的安装 CD 进行复制。不过网络上已经有用户共享了自己的 Quake 3 软件包,除了在树莓派上运行所需要的程序外,还包含了 Quake 3 数据包。用户只要下载、解压后就可以使用了。

① 下载 Quake 3:http://www.kuaipan.cn/file/id_5787571510594308.htm。

② 解压 Quake 3 程序包:

```
# tar xf Quake 3_of_raspberry.tar.bz2
```

③ 进入 Quake 3 目录并运行 Quake 3 程序:

```
# cd Quake 3 & ./ start.sh
```

4.2.3 街机模拟器

在树莓派上能使用的街机模拟器有很多种,不过最著名的还是 MAME(Multiple Arcade Machine Emulator),自 1998 年第一个版本的推出到现在经历了多次完善,其功能也涵盖了多种街机主板,并且支持在不同的平台上运行,针对不同的 CPU 进行优化。由于开放源代码的特性,使得 MAME 有非常多的衍生版,例如“MAME Plus!”以及后来的 mame32c 等。事实上,这个程序几乎统治了整个 Windows 下的街机模拟器,而在 Linux 平台上,主流的 MAME 版本则是 advmame。本书也将介绍 advmame 的用法。

① 在 http://www.kuaipan.cn/file/id_5787571510594311.htm 页面上下载 MAME。

② 解压并第一次运行 MAME:

```
# tar xf mame_for_raspberry.tar.bz2
```

```
# cd mame
```

```
# ./bin/advmame
```

```
Creating a standard configuration file...
```

```
Configuration file '/home/pi/.advance/advmame.rc' created with all the default options.
```

```
The default rom search path is
```

```
 '/home/pi/.advance/rom:/home/pi/mame//share/advance/rom'. You can change it using the 'dir_rom' option in the configuration file.
```

```
./bin/advmame dino
```

```
AdvanceMAME - Copyright (C) 1999 - 2003 by Andrea Mazzoleni
```

```
MAME - Copyright (C) 1997 - 2003 by Nicola Salmoria and the MAME Team
```

第一次运行完成后会提示在用户目录下建立配置文件,用户可以在 /. Advance 目录下找到 advmame 的配置文件,而 rom 的存放目录则是 /mame//share/advance/rom。

修改配置文件 advmame.rc,找到 input_idleexit 0 一行,将它下面的所有的 input_map 开关的行全部删除,输入以下内容来定义按键:

```
input_hotkey yes
```

```
input_idleexit 0
```

```
input_map[coin1] keyboard[0,f1]
```

```
input_map[start1] keyboard[0,f2]
```

```
input_map[p1_up] keyboard[0,w]
```

```
input_map[p1_down] keyboard[0,s]
```

```
input_map[p1_left] keyboard[0,a]
```

```
input_map[p1_right] keyboard[0,d]
```

```
input_map[p1_button1] keyboard[0,u]
```

第4章 媒体中心与游戏

```
input_map[p1_button2] keyboard[0,i]
input_map[p1_button3] keyboard[0,o]
input_map[p1_button4] keyboard[0,j]
input_map[p1_button5] keyboard[0,k]
input_map[p1_button6] keyboard[0,l]
```

上面的例子定义了 F1 为投币, F2 为开始, W、S、A、D 分别定义为上、下、左、右, U、I、O、J、K、L 对应 6 个按键。修改完成后“保存”退出。完成 advmame 的配置, 接下来将下载的 rom 文件存放在 /mame//share/advance/rom 目录中, 就可以开始使用 advmame 来玩游戏了。

首先开启一个终端窗口, 输入以下命令:

```
# ./mame/bin/advmame dino
```

其中, dino 是街机游戏 rom 包的名称, 整个游戏的运行画面如图 4-16 所示。

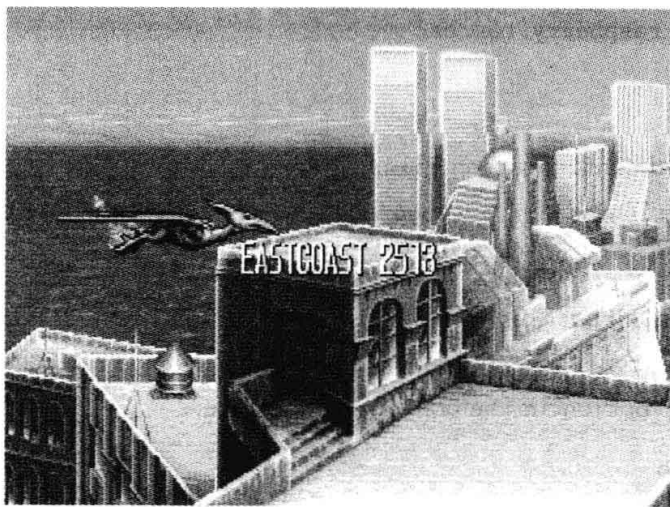


图 4-16 Advance 模拟恐龙大格斗的效果

整个模拟器在默认频率运行时非常卡, 基本上不具备可玩性, 只有当超频达到 900 Hz 以上时, 才勉强可以玩。

第 5 章

迷你服务器

对于很多树莓派来说,很多人都认为它是一种单片机,在使用上也是将它作为单片机来使用,例如用它做机器人、电动小车、智能家电的控制芯片,等等。事实上,这对于树莓派来说是不公平的,由于树莓派是运行完整的 Linux 系统,因此树莓派能做的事情比大家常用的单片机要多得多。本章为大家介绍基于 Linux 系统下的树莓派在网络服务器方面的应用,读者可以通过强大的 Linux 服务器软件将树莓派扩展成为一个超迷你的 Linux 服务器,来完成如 WEB 服务、高清下载机及数据备份中心等功能。

由于 Linux 服务器软件的复杂性,本章所介绍的服务器软件都可以单独出一本书进行介绍,因此这里只介绍服务器软件最常用的配置,以保证用户能在最短的时间内完成服务器的配置,并能保证服务器软件正确工作。

5.1 增加 SWAP 分区

在树莓派上使用服务器软件,首先需要为树莓派增加一个 SWAP 分区(如果读者使用的是 Debian 系统,可以跳过本节,Debian 默认使用了一个 SWAP 文件作为 SWAP 分区)。虽然这不是必需的,但是,这里仍然推荐建立一个 SWAP 分区。SWAP 分区即交换分区,其作用是当系统的物理内存不够使用的时候可以将 SWAP 分区作为内存来使用。虽然在树莓派这种超轻量级的服务器上是不可能有什么大负载的服务,而在实际运行中使用到 SWAP 的机会更是少得可怜,但是也不排除物理内存耗光而导致服务器崩溃的情况,因此十分有必要为树莓派建立一个 SWAP 分区。

图 5-1 为树莓派的内存使用情况。当树莓派运行 20 天后(主要运行 Samba 与 Apache),可以从图 5-1 中看出,内存中由于缓存了大量的数据,已经开始使用了 SWAP 分区中约 24 KB 的空间。

由于 Linux 系统在运行时会缓存大量的数据,所以内存的使用量总是显示很高。但这其实并不要紧,这些缓存的数据在有程序需要使用到内存时,系统会自动释放一些内存空间出来,以保证新进程的运行。如果用户觉得内存不够用,也可以手动释放内存。



图 5-1 树莓派运行状态

首先,将内存中的数据与硬盘进行同步,使用三次 sync 命令:

```
# sync & sync & sync
```

然后通过 proc 文件系统对缓存进行清理:

```
# echo 3 > /proc/sys/vm/drop_caches
```

这个命令会释放缓存、目录以及 inodes。图 5-2 显示的是对缓存进行清理后的效果,其中系统使用了约 37 MB 的内存,而剩余内存为 410 MB,显得非常富裕。

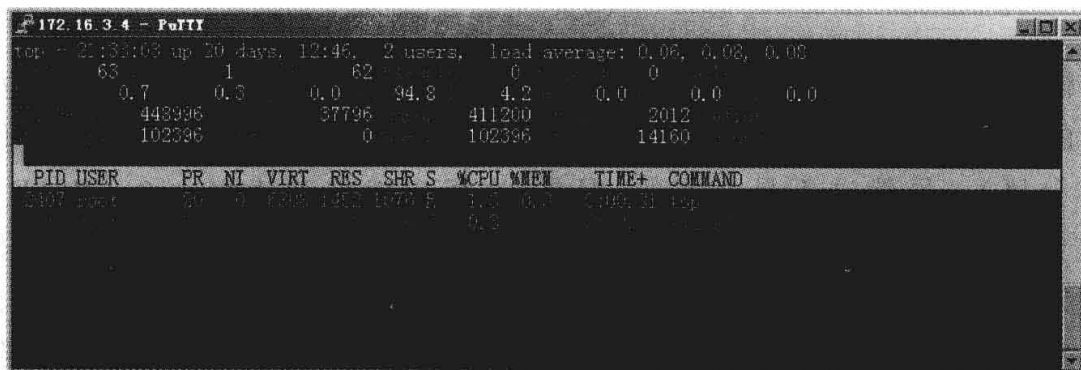


图 5-2 清理后的效果

如果想让内存在使用时不再缓存各种数据,可以在/etc/sysctl.conf 文件中增加如下参数,这样内存就不会持续增加了,但是重复读写文件的性能将会有所下降。

```
vm.dirty_ratio = 1
vm.dirty_background_ratio = 1
vm.dirty_writeback_centisecs = 2
vm.dirty_expire_centisecs = 3
vm.drop_caches = 3
vm.Swapiness = 100
```

```
vm.vfs_cache_pressure = 163
vm.overcommit_memory = 2
vm.lowmem_reserve_ratio = 32 32 8
kern.maxvnodes = 3
```

对于树莓派而言,512 MB 的内存是绝对够用的,因此对于新建立的 SWAP 来说,其实并不需要太大,通常 100 MB 的 SWAP 就足够了。当然,如果有足够多的磁盘空间,也可以为它分出 1 GB 的 SWAP 空间,但如果分得太多就纯粹是浪费了,因为系统根本就用了不太多的 SWAP 空间。

5.1.1 建立 SWAP 分区

建立 SWAP 分区需要使用到 fdisk 或 cfdisk 之类的分区软件。其中,fdisk 是一个历史悠久的分区工具,可以完成多种文件系统的分区,如:linux、fat32、ntfs、linux swap 等。本节将使用 fdisk 来进行分区。

使用 fdisk 等分区工具会对磁盘分区表进行变更,因此在此操作之前请先备份你的数据。

fdisk 命令格式为 fdisk /dev/设备名。例如,fdisk /dev/sda 是指主机内的第一个 SATA 硬盘,sdb 则是第二块 SATA 硬盘。

```
# fdisk /dev/sdb

Command (m for help): n                //输入 n,进入新建分区状态
Command action
  e   extended
  p   primary partition (1-4)
  p                //输入 p 选择建立主分区,需要注意的是主分区最多只能建立四个
Partition number (1-4): 1              //输入分区号
First cylinder (1-2088, default 1):    //直接回车
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-2088, default 2088): +100M
//输入分区大小前面使用“+”表示新增,单位可以使用 K、M、G 来代表 KB、MB、GB
```

完成上面的内容后,一个新的 100 MB 的主分区就完成了。下面开始指定分区的类型:

```
Command (m for help): t                //输入 t 来为分区指定类型
Selected partition 1
Hex code (type L to list codes): 82    //82 是指 SWAP 分区的类型
Changed system type of partition 1 to 82 (Linux swap / Solaris)

Command (m for help): wq                //输入 wq,保存分区表并退出 fdisk 程序
The partition table has been altered!

Calling ioctl() to re-read partition table.
```

Syncing disks.

完成分区之后,需要对新的 SWAP 分区进行格式化,以便于系统使用,如下所示:

```
# mkSWAP /dev/sdb1
```

Setting up SWAPspace version 1, size = 112416 KiB

no label, UUID = 4be6b26c - 0d53 - 46bf - b9b4 - ae2ebe6b25e0

通过 SWAPON 启用新的 SWAP 分区:

```
# SWAPON /dev/sdb1
```

现在可以通过 top 与 free 来查看 SWAP 分区是否启动:

```
# free -m
```

	total	used	free	shared	buffers	cached
Mem:	438	426	12	0	5	364
- / + buffers/cache:		56	381			
SWAP:	99	0	99			

最后需要编辑/etc/fstab 文件,使 Linux 交换分区在引导时自动加载,修改/etc/fstab 文件,在文件的最后面增加以下内容:

```
/dev/sdb1          SWAP                SWAP default        0          0
```

至此,SWAP 分区结束,读者可以重新启动树莓派来查看新的 SWAP 是否启用。

5.1.2 建立 SWAP 文件

如果你的 SD 卡已经因为完成了分区而无法再为 SWAP 建立单独的分区,那么可以使用 SWAP 文件来建立 SWAP。这样的好处是不需要再次调整分区结构,并且可以控制 SWAP 的大小。(注:树莓派所运行的 Debian 系统就是采用这种方法来提供 SWAP 空间的。)

① 创建 SWAP 文件

```
# dd if = /dev/zero of = /mnt/usb/file.swp bs = 1000k count = 100
```

记录了 100 + 0 的读入

记录了 100 + 0 的写出

102400000 字节(102 MB)已复制,4.95895 s,20.6 MB/s

上面的例子中,在/mnt/usb/目录下创建了一个 102 MB 大小的 file.swp 文件,这个文件会将作为 SWAP 来使用。如果读者需要更大的 SWAP 文件,可以通过调整 count 参数来进行控制。

② 格式化并激活 SWAP 文件。在 SWAP 文件使用之前,需要使用 mkswap 命令对它进行格式化,否则 swapon 程序将无法识别 SWAP 文件。

```
# mkswap /mnt/usb/file.swp
# swapon /mnt/usb/file.swp
```

通过 swapon 命令激活的 file.swp 文件已经开始为 SWAP 交换分区工作了。读者可以通过 top 或 free 命令来进行查看。

③ 在 /etc/fstab 文件中增加以下黑体内容, 以方便重启后系统能自动加载 SWAP 文件。

```
# cat /etc/fstab
```

Proc	/proc	proc	defaults	0	0
/dev/mmcb1k0p1	/boot	vfat	defaults	0	2
/dev/mmcb1k0p2	/	ext4	defaults,noatime	0	1
/mnt/usb/file.swp	none	SWAP	defaults	0	0

如果需要停用与删除 SWAP 分区, 可以先使用 swapoff 命令, 以停止对 SWAP 文件的使用, 然后使用 rm 命令删除文件, 对空间进行释放。

```
# SWAPOff/mnt/usb/file.swp
# rm /mnt/usb/file.swp - f
```

5.2 WEB 服务与 WEB 脚本语言

WEB 服务是基础的互联网服务之一, 也是目前服务器上使用最多的服务。本节主要介绍在树莓派上安装与使用两种常用的 WEB 服务器, 分别是 Apache 与 Lighttpd。前者的发展历史几乎就是互联网的 WEB 服务器的历史, 同时 Apache 也是目前使用最多的 WEB 服务器。如果读者是一个 Linux 系统管理员, 则会毫无难度地完成 Apache 的安装与配置。后者则是新出现的 WEB 服务器, 与 Apache 相比, 虽然功能不如 Apache 丰富, 但是胜在小巧, 对于轻量级的应用以及静态 Html 页面, 有巨大的优势。

虽然 WEB 服务器提供了强大的功能, 但它只能完成 WEB 页面的服务提供, 而不能完成 WEB 页面的内容提供, 因此还需要安装一些 WEB 脚本语言来为用户提供一个强大的编程环境。这样才能让 WEB 页面呈现出更多的效果, 完成更多的功能。

5.2.1 Apache

Apache 是互联网上使用最多的 WEB 服务器, 也是历史最悠久、功能最强大的 WEB 程序。本小节主要介绍 Apache 在树莓派上的安装, 便于以后安装基于 WEB 页面的应用程序, 如论坛、Wiki 程序。目前, 树莓派版 Debian 提供的是 Apache2.2.22 版本。此版本是在 1.3 的基础上进行重新设计、编写的, 速度快, 支持多种多路处理模块、多种 WEB 编程语言支持以及更好的安全性。

第5章 迷你服务器

使用 Apache 的另一个好处是, Apache 所支持的功能非常多, 几乎所有的主流 WEB 语言、数据库系统都提供了 Apache 的支持模块。这些模块无论是安装还是配置、使用, 都非常方便, 几乎不需要用户操心其配置。因此推荐读者使用 Apache 作为 WEB 服务器。

1. 安装 Apache

由于 Debian 提供了 Apache 的安装包, 因此可以直接使用 apt 程序进行安装:

```
# apt - get install apache2
```

```
正在读取软件包列表... 完成
```

```
正在分析软件包的依赖关系树
```

```
正在读取状态信息... 完成
```

```
将会安装下列额外的软件包:
```

```
apache2 - mpm - worker apache2 - utils apache2.2 - bin apache2.2 - common libapr1 libaprutil1 libaprutil1 - dbd - sqlite3 libaprutil1 - ldap
ssl - cert
```

建议安装的软件包:

```
apache2 - doc apache2 - suexec apache2 - suexec - custom openssl - blacklist
```

下列【新】软件包将被安装:

```
apache2 apache2 - mpm - worker apache2 - utils apache2.2 - bin apache2.2 - common libapr1 libaprutil1 libaprutil1 - dbd - sqlite3
libaprutil1 - ldap ssl - cert
```

升级了 0 个软件包, 新安装了 10 个软件包, 要卸载 0 个软件包, 有 62 个软件包未被升级。

需要下载 1 348 KB 的软件包。

解压缩后会消耗掉 4 990 KB 的额外空间。

您希望继续执行吗? [Y/N]Y

```
.....
```

```
Enabling module authn_file.
```

```
Enabling module authz_host.
```

```
Enabling module reqtimeout.
```

```
正在设置 apache2 - mpm - worker (2.2.22 - 12) ...
```

```
[....] Starting web server: apache2apache2: Could not reliably determine the server's
fully qualified domain name, using 127.0.1.1 for ServerName
```

```
. ok
```

```
正在设置 apache2 (2.2.22 - 12) ...
```

```
正在设置 ssl - cert (1.0.32) ...
```

安装完成后, Apache 服务启动时会警告没有合适的域名, 不过没有关系, 下一节将会修改这个参数, 并重新启动 Apache。

2. 配置与管理 Apache

原有的 Apache 配置是由 /etc/apache/httpd.conf 文件提供的, 但是由于功能越

来越多,所以 httpd.conf 被拆分为多个配置文件,详情如下:

- httpd.conf: Apache 的主配置文件。它定义了 Apache 的一些基本配置以及对其他配置文件的引用,是我们经常修改的文件。
- conf.d 目录:用于存放一些杂项的设置。
- mods-available 目录:定义各种模块配置的目录。此目录并不会直接使用,而是通过 mods-enabled 目录进行调用。
- mods-enabled 目录:调用各种模块配置的目录。如果需要使用某个模块,则通过连接 mods-available 目录下的配置文件来实现引用。
- sites-available 目录:一些常规的设置与 mods-available 目录一样,不会直接引用。
- sites-enabled 目录:引用 sites-available 目录下的内容来确认哪些配置需要使用。

对于大多数的读者来说,只需要增加一个 ServerName 参数来标识一下服务器名称就可以了,而其他的参数可以使用 Apache 默认的设置即可让 Apache 工作得很好。

使用熟悉的编辑器修改/etc/apache2/目录下的 apache2.conf 文件,在全局设置(Global configuration)部分增加一个 ServerName 参数,即:

ServerName 主机名或 IP 地址

图 5-3 所示为修改 apache2.conf 配置文件。

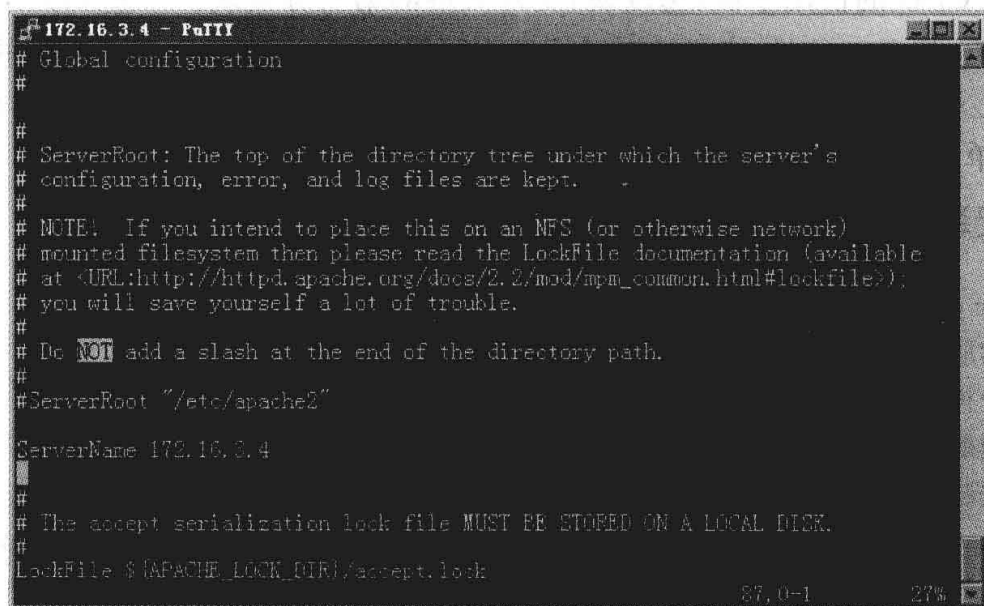


图 5-3 修改 apache2.conf 配置文件

ServerName 命令的作用是设置 Apache 的主机名或主机的 IP 地址,以便让用户进行访问。在修改完成后可以使用下面的命令重新启动 Apache 服务器,以便让新的配置生效。

```
# /etc/init.d/apache2 restart
```

```
[ ok ] Restarting web server: apache2 ... waiting .
```

使用新的地址进行访问,如果看到图 5-4 所示的页面则 Apache 启动成功。

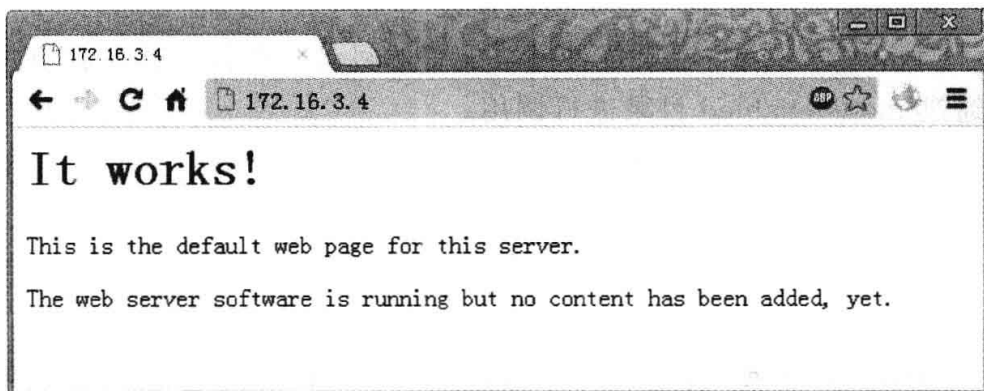


图 5-4 Apache 默认的运行页面

到目前为止,Apache 服务器已完成安装,在 Apache 中默认的页面文件存放在 /var/www 目录下,读者可以将自己的 html 文件放在此目录下,以方便通过 WEB 浏览器进行访问。

5.2.2 Lighttpd

Lighttpd 是一个轻量级的 WEB 服务器软件,其设计目标就是内存开销小,兼容性好,并且 CPU 的占用率低,速度快,支持大量的第三方模块。一些 Apache 所提供的重要功能,如 FastCGI、访问控制、URL 重写等功能,Lighttpd 也有相应的实现;同时,其配置文件类似于 Apache 的配置文件,用户可以快速上手。因此,如果用户需要一个简单、轻量级的 WEB 服务器软件,可以尝试一下 Lighttpd。

1. 安装 Lighttpd

```
# apt-get install lighttpd
```

```
正在读取软件包列表... 完成
```

```
正在分析软件包的依赖关系树
```

```
正在读取状态信息... 完成
```

```
将会安装下列额外的软件包:
```

```
libfam0 libterm-readkey-perl libterm-readline-perl-perl spawn-fcgi
```

```
建议安装的软件包:
```

```
fam rrdtool
```

```
下列【新】软件包将被安装:
```

```
libfam0 libterm-readkey-perl libterm-readline-perl-perl lighttpd spawn-fcgi
```

```
升级了 0 个软件包,新安装了 5 个软件包,要卸载 0 个软件包,有 89 个软件包未被升级。
```

```
需要下载 409 KB 的软件包。
```

```
解压缩后会消耗掉 1 232 KB 的额外空间。
```

您希望继续执行吗? [Y/n]Y

.....

正在设置 libterm - readkey - perl (2.30 - 4) ...

正在设置 libterm - readline - perl - perl (1.0303 - 1) ...

正在设置 lighttpd (1.4.31 - 3) ...

[ok] Starting web server: lighttpd.

正在设置 spawn - fcgi (1.6.3 - 1) ...

update - alternatives: using /usr/bin/spawn - fcgi. standalone to provide /usr/bin/spawn - fcgi (spawn - fcgi) in 自动模式

update - alternatives: 警告: not replacing /usr/share/man/man1/spawn - fcgi.1.gz with a link

安装到最后,会出现一个关于 Lighttpd 帮助手册的错误警告,无视即可。安装完成后,Lighttpd 会自动启动,并在开机时作为服务启动。

2. 配置与管理 Lighttpd

Lighttpd 的配置文件存放在 /etc/lighttpd/ 目录下,其中,lighttpd.conf 文件是最主要的控制文件,conf-available 是具体功能的配置存放目录,conf-enabled 目录则是配置引用存放目录。

```
# ls /etc/lighttpd/ * - F
/etc/lighttpd/lighttpd.conf

/etc/lighttpd/conf-available:
05 - auth.conf          10 - fastcgi.conf      10 - ssl.conf
10 - accesslog.conf     10 - flv-streaming.conf 10 - status.conf
10 - cgi.conf           10 - no-www.conf       10 - userdir.conf
10 - dir-listing.conf   10 - proxy.conf        10 - usertrack.conf
10 - evasive.conf       10 - rrdtool.conf      15 - fastcgi-php.conf
10 - evhost.conf        10 - simple-vhost.conf 90 - debian-doc.conf
10 - expire.conf        10 - ssi.conf          README

/etc/lighttpd/conf-enabled:
```

lighttpd.conf 配置文件非常短,默认的配置文件的只有 28 行,主要定义了 html 文件、日志等存放位置、索引文件类型、服务器启动用户、模块、端口号等内容,如下所示:

```
server.modules = (
    "mod_access",
    "mod_alias",
    "mod_compress",
    "mod_redirect",
    # "mod_rewrite",
```

```

)

server.name = "172.16.3.4"
server.document-root = "/var/www"
server.upload-dirs = ( "/var/cache/lighttpd/uploads" )
server.errorlog = "/var/log/lighttpd/error.log"
server.pid-file = "/var/run/lighttpd.pid"
server.username = "www-data"
server.groupname = "www-data"
server.port = 80

index-file.names = ( "index.php", "index.html", "index.lighttpd.html" )
url.access-deny = ( "~", ".inc" )
static-file.exclude-extensions = ( ".php", ".pl", ".fcgi" )
compress.cache-dir = "/var/cache/lighttpd/compress/"
compress.filetype = ( "application/javascript", "text/css", "text/html", "text/
plain" )

# default listening port for IPv6 falls back to the IPv4 port
include_shell "/usr/share/lighttpd/use-ipv6.pl " + server.port
include_shell "/usr/share/lighttpd/create-mime.assign.pl"
include_shell "/usr/share/lighttpd/include-conf-enabled.pl"

```

默认的配置就可以很好地工作,笔者只是为它增加了一个 `server.name` 参数来标识服务器。如果读者需要更多的功能,可以通过 `ln` 命令链接 `/etc/lighttpd/conf-available` 目录下的文件到 `/etc/lighttpd/conf-enabled` 目录中,即可启用相应的功能。例如启用代理模块,如下所示:

```

# ln -sf /etc/lighttpd/conf-available/10-proxy.conf /etc/lighttpd/conf-enabled/
10-proxy.conf

# ls /etc/lighttpd/conf-enabled -l
总用量 0
lrwxrwxrwx 1 root root 42 2013/01/04 15:55:18 10-proxy.conf -> /etc/lighttpd/conf-
available/10-proxy.conf

```

可以看到, `/etc/lighttpd/conf-enabled` 目录下已经有一个 `10-proxy.conf` 的文件被链接到 `/etc/lighttpd/conf-available/10-proxy.conf` 目录下,现在使用下面的命令重新启动 Lighttpd 服务器,以便让新的配置生效:

```

# /etc/init.d/lighttpd restart
[ ok ] Stopping web server: lighttpd.
[ ok ] Starting web server: lighttpd.

```

现在可以在浏览器中输入服务器 IP 地址或主机名来进行访问,如图 5-5 所示。页面提示这个 Lighttpd 服务器已经开始工作了,但用户还没有放上 html 页面,

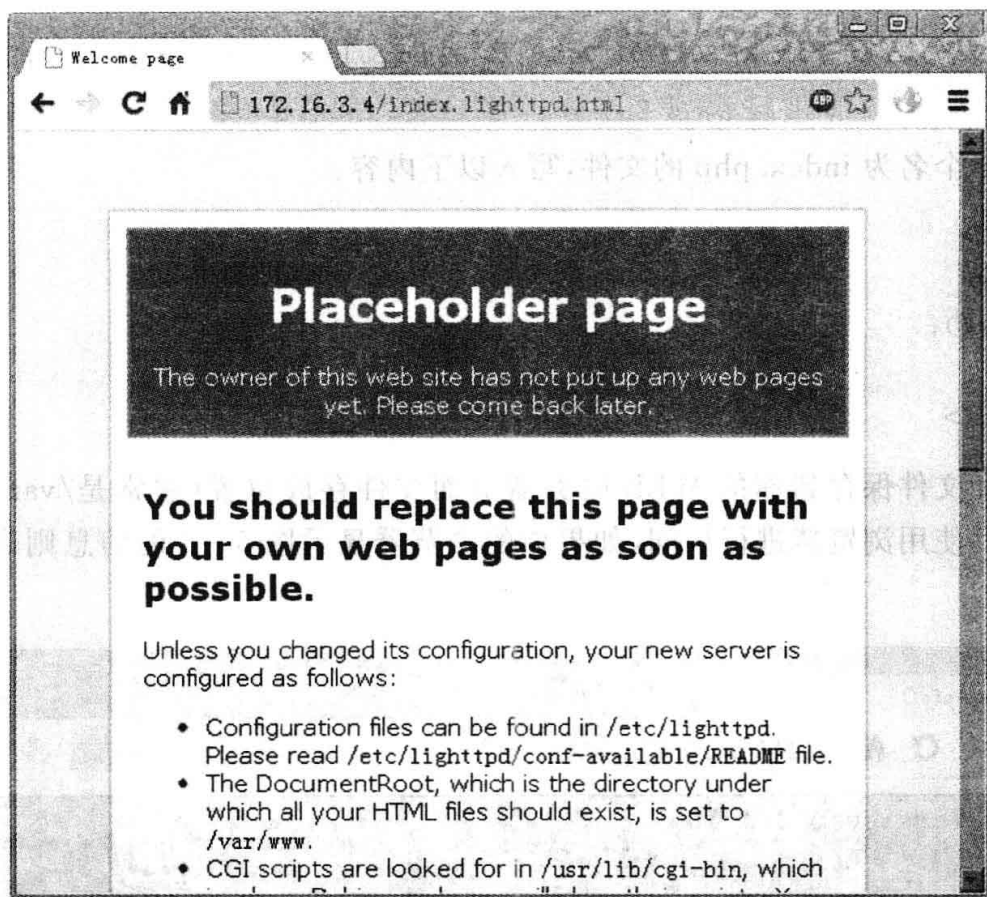


图 5-5 Lighttpd 默认启动页面

因此建议用户尽快放上自己的页面文件。

5.2.3 PHP

PHP 是一个强大的 WEB 编程语言,使用 C 语言开发,拥有运行速度快,能够嵌入到 HTML 语言等优点,并且 PHP 混合了 C、Java、Perl 等程序的语法,因此相对于其他编程语言更容易上手,实用性更强,特别适合初学者使用。

目前,Debian 提供的是 PHP5。在这个版本中,使用了 Zend II 引擎,并加入了新的功能,如 PHP Data Objects(PDO),同时还支持小容量的 RDBMS - SQLite 数据库,以及直接 XML 数据访问等功能。

① 安装 PHP5:

```
# apt - get install php5
```

② 配置 PHP5。由于在安装 PHP5 时 PHP 程序会自动安装到 Apache 中,因此对于 Apache 来说,只需保证/etc/apach/mods-enabled 目录中有 PHP5 的配置链接即可。

```
root@raspberrypi /etc/apache2/mods-enabled # ls php* -l
lrwxrwxrwx 1 root root 27 2013/01/04 18:10:33 php5.conf -> ../mods-available/php5.
```



```
conf
```

```
lrwxrwxrwx 1 root root 27 2013/01/04 18:10:33 php5.load -> ../mods-available/php5.
```

```
load
```

建立一个名为 index.php 的文件,写入以下内容:

```
<html>
<? php
phpinfo();
? >
</html>
```

将这个文件保存到你的 WEB 服务器页面文件存放位置(通常是/var/www 目录下),然后使用浏览器进行访问,如果你的浏览器显示图 5-6 的信息则表示 PHP 安装成功。

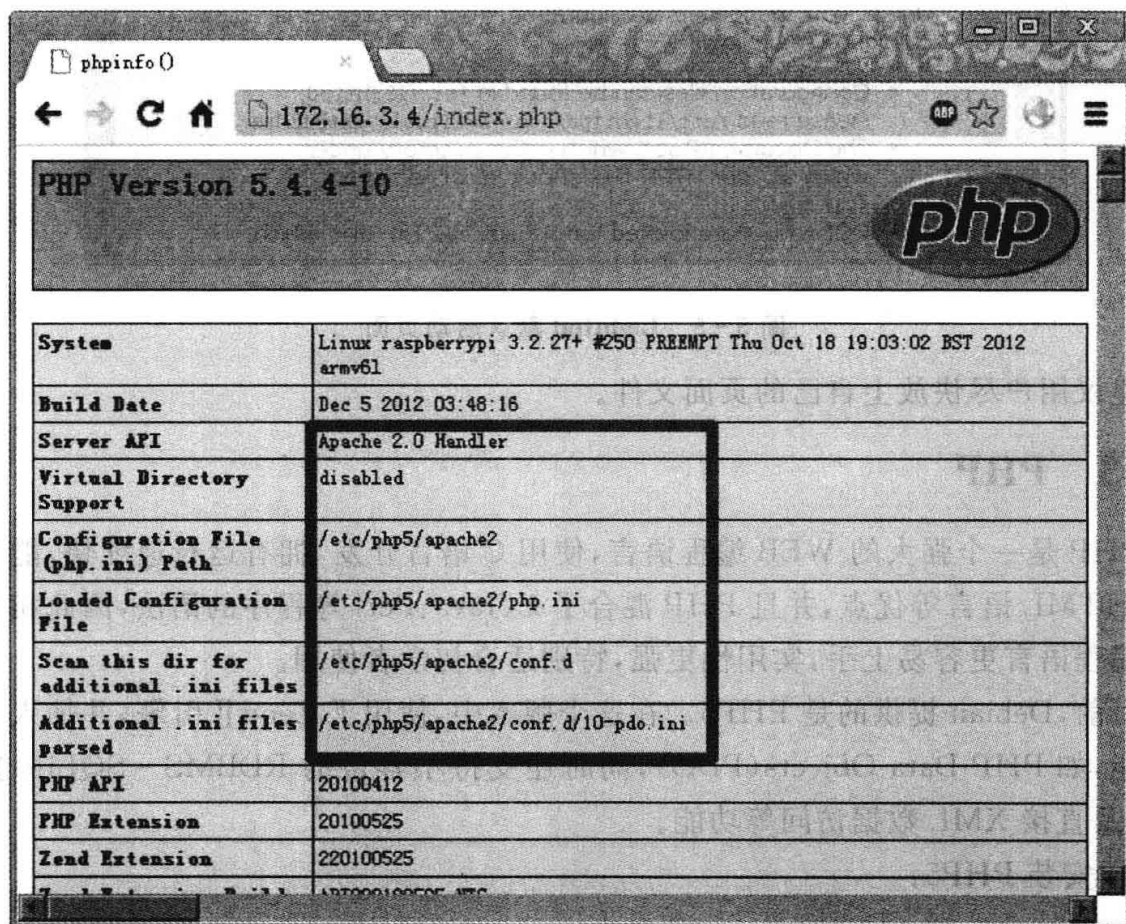


图 5-6 Apache 的 phpinfo 信息

对于 Lighttpd 来说则要麻烦一些,Lighttpd 使用 FastCGI 方式来调用 PHP 程序,因此不能像 Apache 那样通过加载模块来实现 PHP 的支持,而是首先需要安装 PHP5 的 CGI 程序:

```
# apt-get install php5-cgi
```

配置 Lighttpd 的 FastCGI 模块在 `/etc/lighttpd/conf-available/10-fastcgi.conf` 文件后追加以下内容,然后保存退出。

```
fastcgi.server = (".php" =>
    ("localhost" =>
        (
            "socket" => "/tmp/php-fastcgi.socket",
            "bin-path" => "/usr/bin/php-cgi",
            "max-procs" => 2,
            "bin-environment" =>
                (
                    "PHP_FCGI_CHILDREN" => "100",
                    "PHP_FCGI_MAX_REQUESTS" => "1000"
                ),
            "broken-scriptfilename" => "enable"
        )
    )
)
```

在 `conf-enabled` 目录下为 Lighttpd 建立一个配置链接:

```
# ln -sf /etc/lighttpd/conf-available/10-fastcgi.conf /etc/lighttpd/conf-enabled/10-fastcgi.conf
# ls /etc/lighttpd/conf-enabled/
10-fastcgi.conf  10-proxy.conf
```

修改完成后可以重新启动 Lighttpd 程序:

```
# /etc/init.d/lighttpd restart
[ ok ] Stopping web server: lighttpd.
[ ok ] Starting web server: lighttpd.
```

与测试 Apache 下的 PHP 程序一样,在 `/var/www` 目录下建立一个名为 `index.php` 的文件,写入以下内容:

```
<html>
<? php
phpinfo();
? >
</html>
```

使用浏览器进行访问,如果你的浏览器显示出图 5-7 的信息,则表示 Lighttpd 成功地使用了 FastCGI 方式调用 PHP。

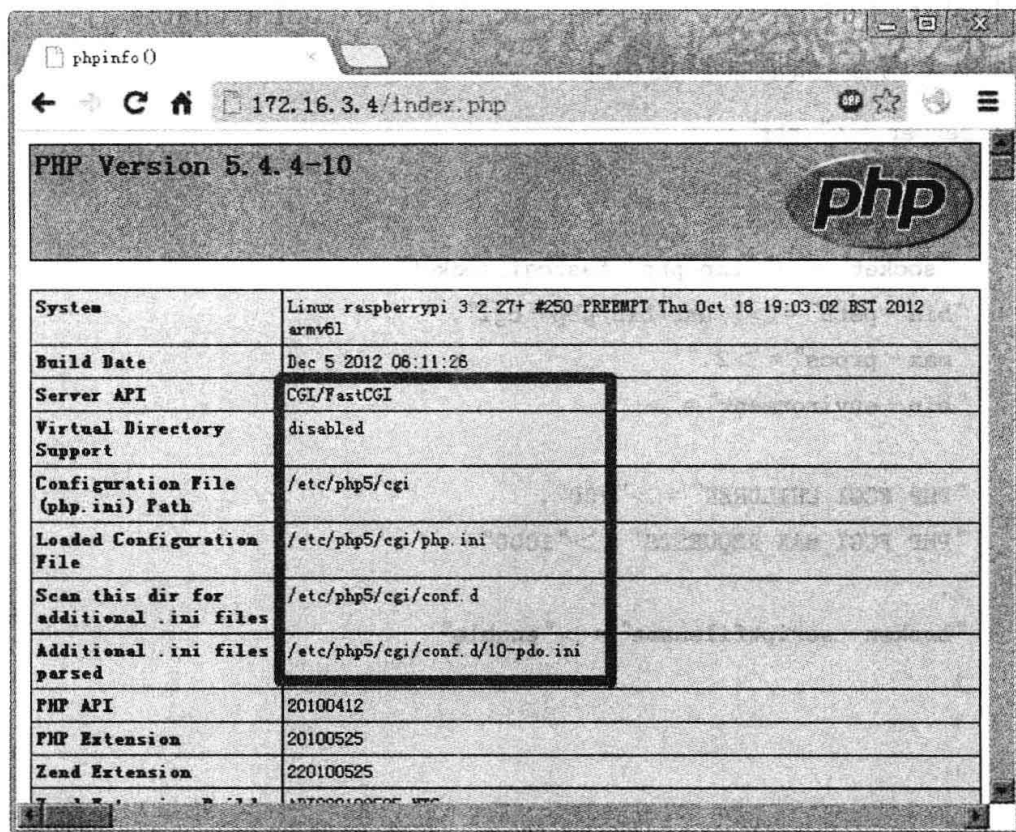


图 5-7 Lighttpd 的 phpinfo 信息

5.2.4 Perl

在树莓派的 Debian 版本中默认安装了 Perl 程序。由于 Perl 的广泛使用,所以大多数 Linux 系统默认安装它,但是在 Debian 上安装的 Perl 并不是最新的版本。如果想要在 WEB 服务器上使用 Perl,则需要更新 Perl 并安装相应的 Perl 模块。

由于 Apache 提供了 Perl 的模块包,因此只需要安装 libapache2-mod-perl2 就可以完成 apache 对 Perl 的模块化支持(除了以模块方式运行 Perl 外,Apache 还支持以 CGI 方式运行 Perl,但是效率会低一些)。

为 apache 安装 libapache2-mod-perl2 包:

```
# apt-get install libapache2-mod-perl2 -y
```

安装完成后系统会自动重启 Apache,以便于加载新的模块,并完成对 Perl 的模块支持。

现在用户可以在 html 页面中调用 Perl 程序或直接使用 Perl 的脚本程序,如果需要确认 mod_perl 是否启用,可以将下面的代码加入到 httpd.conf 中,以开启 mod_perl 自带的 Status 模块来查看 Perl 模块信息:

```
<Location /perl-status>
SetHandler modperl
```

```
PerlOptions + GlobalRequest
PerlResponseHandler Apache2::Status
</Location>
```

通过 `http://hostname/perl-status` 进行访问,如图 5-8 所示。

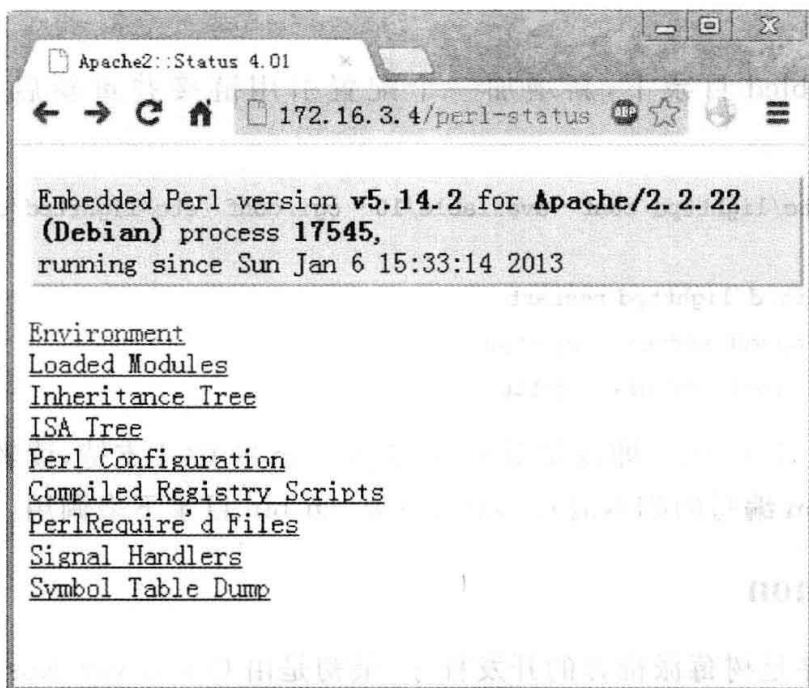


图 5-8 Perl 状态页面

如果能看到这个页面,说明 `mod_perl` 模块成功启动并正在运行中;如果页面显示没有 `Devel::Symdump` 模块,则需要通过 `cpan` 来安装 Perl 的 `Symdump` 模块:

```
# cpan
cpan[1]> install Devel::Symdump
cpan[1]> exit
```

安装完成后重新刷新页面就可以看到详细内容了。

Perl 对于 `Lighttpd` 的支持则要麻烦一些,由于没有现成的 Perl 模块供 `Lighttpd` 程序使用,因此需要使用 CGI 方式调用 Perl,修改 `/etc/lighttpd/conf-available/` 目录下的 `10-cgi.conf` 文件,取消对 Perl 的注释(也可以顺便取消对 `python` 的注释,这样做可以直接启动对 `python` 的 CGI 支持)。

```
# /usr/share/doc/lighttpd/cgi.txt
server.modules += ( "mod_cgi" )

$ HTTP["url"] = ~ "^/cgi-bin/" {
    cgi.assign = ( "" => "" )
}
```

```
## Warning this represents a security risk, as it allow to execute any file
```

第5章 迷你服务器

```
## with a .pl/.py even outside of /usr/lib/cgi-bin.
#
cgi.assign = (
    ".pl" => "/usr/bin/perl",
    ".py" => "/usr/bin/python",
)
```

在 conf-enabled 目录下,新增加一个配置引用链接并重新启动 Lighttpd 服务器:

```
# ln -sf /etc/lighttpd/conf-available/10-cgi.conf /etc/lighttpd/conf-enabled/10-cgi.conf
# #/etc/init.d/lighttpd restart
[ ok ] Stopping web server: lighttpd.
[ ok ] Starting web server: lighttpd.
```

启动完成后,Lighttpd 即可获得 Perl 与 python 的 CGI 支持,读者可以将自己使用 Perl 或 python 编写的脚本放在 /var/www/cgi-bin 目录下调用。

5.2.5 Python

Python 程序是树莓派推荐的开发程序,最初是由 Guido van Rossum 于 1989 年底发明的,1991 年公开发行了它的第一个版本。Python 语法简洁而清晰,不像 Perl 难以阅读;具有统一的代码风格,简单的语法,使得 Python 易读、易维护,并且是被大量用户所欢迎的、用途广泛的语言,还拥有丰富和强大的类库。因此可以轻松地把用其他语言制作的各種模块(C/C++)联结在一起。

在 Apache 服务器中,只需要安装 libapache2-mod-python 包即可使 Apache 完成对 python 的支持。

```
# apt-get install libapache2-mod-python
```

安装完成后会自动重新启动 Apache 进程。现在重新启动 Apache 会开始支持 python 模块,用户可以直接在 html 文件中调用 python 的程序。

如果要查看 python 模块的运行状态,可以通过 Apache 自带的 info 信息模块来查看,首先启动 info 模块,在 /etc/apache2/mods-enabled/ 目录中增加 info 模块的引用链接。

```
# ln -sf /etc/apache2/mods-available/info.load /etc/apache2/mods-enabled/info.load
# ln -sf /etc/apache2/mods-available/info.conf /etc/apache2/mods-enabled/info.conf
```

修改 /etc/apache2/mods-available/ 目录下的 info.conf 文件,将 Allow from 127.0.0.1 修改为你的 IP 地址或是 all,允许用户查看服务器信息。


```

<IfModule mod_info.c>
#
# Allow remote server configuration reports, with the URL of
# http://servername/server-info (requires that mod_info.c be loaded).
# Uncomment and change the "192.0.2.0/24" to allow access from other hosts.
#
<Location /server-info>
    SetHandler server-info
    Order deny,allow
    Deny from all
    Allow from all
#    Allow from 192.0.2.0/24
</Location>
</IfModule>

```

修改完成后需要重新启动 Apache, 并通过 `http://hostname/server-info` 来查看服务器信息, 如图 5-9 所示。



图 5-9 Apache 的服务器信息

警告:在服务器正式运行后一定要修改 Allow from 参数, 以禁止用户获得服务器运行信息。如果条件许可最好删除 `/etc/apache2/mods-enabled/` 目录下的 `info.load` 与 `info.conf` 文件, 以禁止使用 info 模块:

```
# rm -f /etc/apache2/mods-enabled/info.*
```

关于 Lighttpd 的 python 支持可以查看 5.2.4 小节中 Lighttpd 对 Perl 的支持。

5.3 数据库支持

5.3.1 MySQL

MySQL 是目前网络上使用最广泛的数据库系统之一,由于免费并开放源代码的特点,一般中小型网站的开发都会选择 MySQL 作为网站的数据库,与 PHP/Perl 配合 Apache 可以形成优秀的 WEB 服务器平台。

```
# apt - get install mysql - server
```

安装过程中会要求设置 MySQL 的 root 用户密码(请一定记好你的密码),如图 5-10 所示,重复输入两次即可。安装完成后会启动 MySQL 服务。

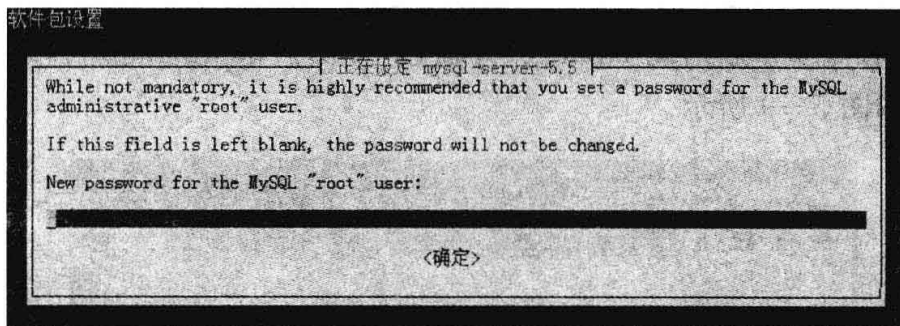


图 5-10 设置 MySQL 密码

使用 ps 来查看当前系统进程,可以看到 MySQL 服务已经启动了。

```
# ps ax | grep mysql
3699 pts/0    S+   0:00 grep mysql
21774?      S     0:00 /bin/sh /usr/bin/mysqld_safe
22100?      Sl    11:13 /usr/sbin/mysqld -- basedir = /usr -- datadir = /var/
lib/mysql -- plugin - dir = /usr/lib/mysql/plugin -- user = mysql -- pid - file = /var/run/
mysqld/mysqld.pid -- socket = /var/run/mysqld/mysqld.sock -- port = 3306
22101 ?      S     0:00 logger -t mysqld -p daemon.error
```

为了方便对 MySQL 的管理,一般使用 phpMyAdmin 对 MySQL 进行管理。读者可在 http://www.phpmyadmin.net/home_page/ 页面上下载,国内也有很多网站提供了 phpMyAdmin 的下载。

使用 wget 下载 phpMyAdmin 并解压它:

```
# wget http://nchc.dl.sourceforge.net/project/phpmyadmin/phpMyAdmin/3.5.5/phpMyAd-
min-3.5.5-all-languages.tar.bz2
# tar xf phpMyAdmin-3.5.5-all-languages.tar.bz2
```

解压完成后会出现一个名为 phpMyAdmin-3.5.5-all-languages 的目录,我们使

用下面的命令将它修改成 phpMyAdmin, 以便在浏览器登录时输入。

```
# mv phpMyAdmin - 3.5.5 - all - languages phpMyAdmin
```

最后安装一些 phpMyAdmin 运行时所需要的依赖包软件, 例如 PHP5 的 MySQL 驱动库等。如果没有安装这些依赖包软件, 那么会在浏览 phpMyAdmin 管理页面时出现图 5-11 的错误。

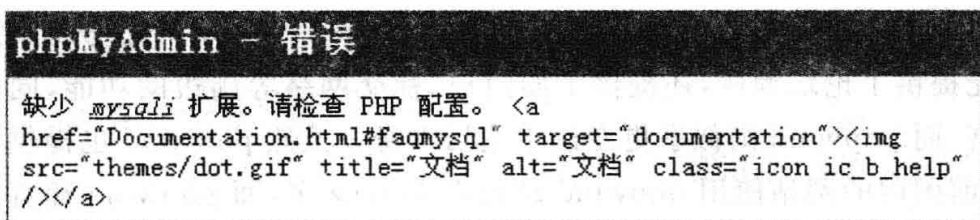


图 5-11 缺少 MySQL 扩展错误

使用 apt-get 安装依赖包:

```
# apt - get install php5 - mysql php5 - mcrypt.
```

安装完成后重新启动 Apache 服务器, 现在读者可以使用 http://服务器 ip/phpMyAdmin 的格式来访问 phpMyAdmin 页面了。用户是 root, 密码则是之前在安装 MySQL 时设置的密码。

5.3.2 PostgreSQL

PostgreSQL 是一个优秀的数据库系统, 是以加州大学伯克利分校计算机系开发的 POSTGRES 为基础开发的对象关系型数据库管理系统, 有着灵活的许可证模式, 任何人都可以以任何目的免费使用、修改, 并分发 PostgreSQL 程序。目前, PostgreSQL 的最新版本是 9.2。此版本在性能方面有很大提升, 并支持一些新的 SQL 特性以及复制等方面。

PostgreSQL 使用 apt-get 来进行安装:

```
# apt - get install postgresql
```

安装完成后 PostgreSQL 服务会自启动, 并开始工作, 如果需要配置 PostgreSQL, 可以修改/etc/postgresql/9.1/main/目录下的文件, 通常只需要修改以下两个文件即可:

① pg_hba.conf 文件负责访问控制, 如果想让所有用户都可以不使用密码直接访问, 可以追加以下内容:

```
host    all          all          0.0.0.0/0      trust
```

② postgresql.conf 是 PostgreSQL 的主配置文件, 包含了大部分的配置, 例如连接端口等内容。通常保持默认设置就可以了。

现在 PostgreSQL 已经正常工作了,用户已经可以开始使用 PostgreSQL 了。

5.4 设论坛与 Wiki 网站

5.4.1 论坛

目前,国内使用最多的论坛程序有两种:Discuz 和 phpwind。从使用上来说,Discuz 不光提供了论坛程序,还提供了如门户、社交网络等周边的功能,同时界面风格更为丰富,而 phpwind 则似乎更专注于论坛本身。虽然 phpwind 也提供了门户等功能,但目前国内的网站使用 phpwind 却没有 Discuz 多,而 phpwind 却有比 Discuz 更快的反应速度,以及大量的内置插件,使用户在论坛管理时更为方便与快捷。

如果用户第一次架设论坛系统,不妨自己将这两种论坛分别安装一下,看看哪一种更适合自己。

1. Discuz

Discuz 是目前国内使用得最多的通用社区论坛软件系统,从 2001 年发布至今已经拥有 11 年的应用历史,能够提供论坛(BBS)、社交网络(SNS)、门户(Portal)、群组(Group)、开放平台(Open Platform)等多种应用,是目前国内成熟度最高、使用最广泛的论坛软件系统。

由于 Discuz 论坛是依赖于 PHP 与 MySQL 程序的,因此安装 Discuz 论坛首先需要保证树莓派上已经安装好 WEB 服务、PHP 与 MySQL 程序,具体安装可以参考 5.2 和 5.3 节。

首先需要在 <http://www.comsenz.com/downloads/install/discuzx> 上下载最新版本的 Discuz 程序。目前最新的 Discuz 为 2.5 版,对于不同的语言编码会有不同的版本,因此读者需要仔细选择,但可以保证的是这些版本都支持中文。

先进入到 WEB 服务器存放 WEB 文件的 /var/www 目录下,然后使用 wget 下载 Discuz 程序。

```
# cd /var/www/  
# wget http://download.comsenz.com/DiscuzX/2.5/Discuz_X2.5_SC_UTF8.zip
```

完成后解压下载的 Discuz_X2.5_SC_UTF8.zip 程序包。

```
# unzip Discuz_X2.5_SC_UTF8.zip
```

解压完成后得到三个目录:

① Readme:说明目录,用于存放说明文件、新版本及历史更新记录、授权文件等内容,推荐查看一下。

② Utility:目录为论坛附带工具,包括升级程序等,读者如果是全新安装则可以

跳过。

③ Upload: Discuz 论坛程序的目录。这个目录中有一些需要有写入的权限,否则在安装时会因为无权限运行而无法安装。

在安装之前需要为以下文件和目录设置可执行权限,读者可以通过 `chmod 777-R` 命令来为目录赋权。

```
./config/config_global.php
./config/config_ucenter.php
./config
./data
./data/avatar
./data/plugindata
./data/template
./data/threadcache
./data/attachment
./data/attachment/album
./data/attachment/forum
./data/attachment/group
./data/log
./uc_client/data/cache
./uc_server/data/
./uc_server/data/cache
./uc_server/data/avatar
./uc_server/data/backup
./uc_server/data/logs
./uc_server/data/tmp
./uc_server/data/view
```

修改 Discuz 目录名为 `bbs`, 这样以后可以直接使用 `bbs` 的目录名来访问论坛:

```
# mv upload bbs
```

当以上所有准备工作完成后就可以通过浏览器开始安装 Discuz 论坛程序了。通过 `http://服务器 IP 地址/bbs` 格式进行访问后会直接进入安装向导,如图 5-12 所示。

单击“同意”后 Discuz 程序会开始自动检查安装环境,如图 5-13 所示,以确认服务器系统、文件权限是否满足要求。如果条件全都符合则直接单击“下一步”按钮。

在设置运行环境中,需要选择是全新安装还是只安装 Discuz,全新安装会安装 UCenter(用户管理中心),而仅安装 Discuz 则需要手动指定已安装 UCenter 的位置,推荐使用全新安装。

开始配置数据库安装参数,如图 5-14 所示。这一步大部分参数都可以使用默认,真正需要修改的有数据库密码(推荐将数据库名由 `ultrax` 改成其他名称)以及管

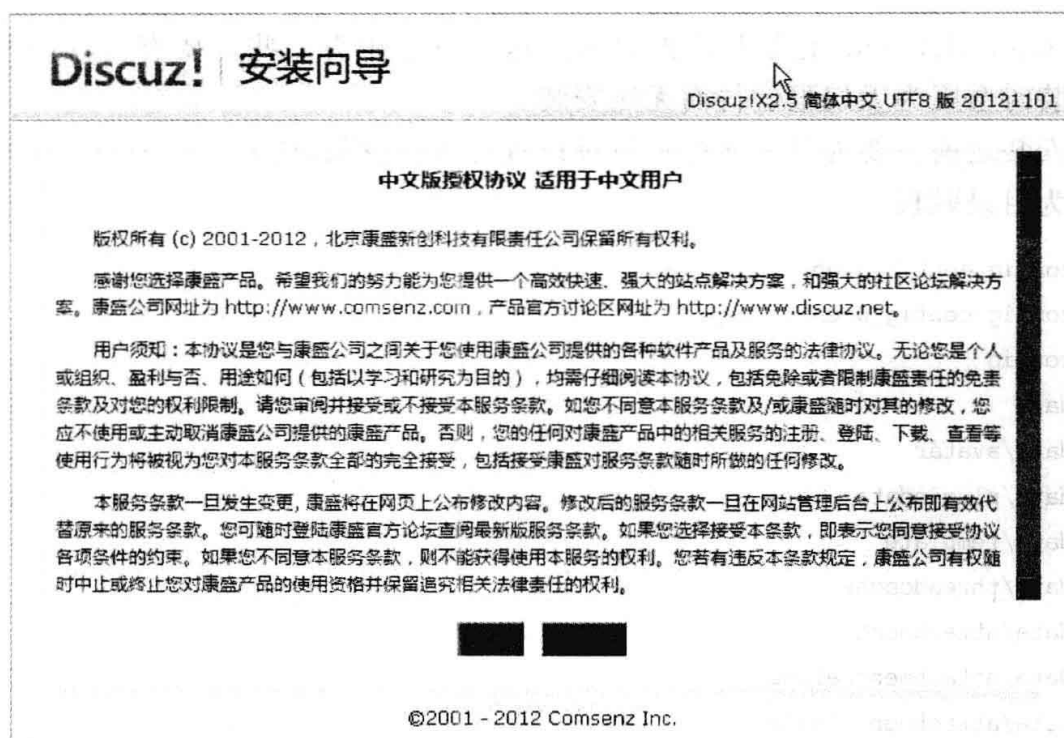


图 5-12 Discuz 安装向导



图 5-13 检查安装环境

理员账号和密码。

修改完成后单击“下一步”按钮，进入数据库的正式安装。整个过程需要安装

Discuz! 安装向导

Discuz!X2.5 简体中文 UTF8 版-20121101

3. 安装数据库

正在执行数据库安装

检查安装环境 设置运行环境 **创建数据库** 安装

填写数据库信息

数据库服务器: 数据库服务器地址, 一般为 localhost

数据库名:

数据库用户名:

数据库密码:

数据库前缀: 同一数据库运行多个论坛时, 请修改前缀

系统信箱 Email: 用于发送程序错误报告

填写管理员信息

管理员账号:

管理员密码: 管理员密码不能为空

重复密码:

管理员 Email:

©2001 - 2012 Comsenz Inc.

图 5-14 设置数据库信息

500 多个数据表,在树莓派上需要 10 min 左右,如果一切顺利,在数据库安装完成后会通知用户安装成功,随后用户可以使用刚才的地址访问自己的论坛了。

2. phpwind

phpwind 是基本 PHP 与 MySQL 程序的论坛程序,也是国内使用最多的通用论坛程序之一。与 Discuz 不同的是,它能提供更简洁的界面,更快的响应速度,以及官方集成的大量的各种用途的插件,因此如果用户只关注论坛本身,而不是论坛周边,则推荐使用 phpwind 程序。

首先,在 phpwind 的官方网站(<http://www.phpwind.net/>)上下载最新版本的 phpwind 程序:

```
# wget http://www.phpwind.com/downloads/forums/phpwind_v9.0_utf8.zip
```

解压 phpwind 程序:

```
# unzip phpwind_v9.0_utf8.zip
```

与 Discuz 一样,解压后会获得类似的目录结构:

① readme.txt:说明文件,用于介绍 phpwind。

第5章 迷你服务器

② utility: 目录为论坛附带工具。

③ upload: phpwind 论坛程序的目录。

修改 phpwind 默认的文件夹名称为 bbs:

```
# mv upload bbs
```

在浏览器中打开“http://服务器 ip/bbs”目录, 然后开始安装 phpwind。

整个过程与 Discuz 一样: 单击“接受”协议, 随后 phpwind 会开始检查系统是否符合安装需求。通常解压后的默认权限即可满足安装要求。接下来单击“下一步”按钮, 开始输入数据库参数, 与 Discuz 一样需要输入 MySQL 数据库的 root 用户密码以及建立的 phpwind 数据库名称和管理员信息, 如图 5-15 所示。

图 5-15 设置数据库信息

所有参数输入完成后单击“创建数据”按钮, phpwind 开始建立数据表, 完成安装。当安装完成后就可以使用之前的网址(http://服务器 ip/bbs)重新访问 phpwind 论坛了。

5.4.2 建立 Wiki

MediaWiki 是一套以 GPL 授权发行的 Wiki 引擎, 也是一套强大的知识分享系

统。在这个系统上,每个人都可以分享自己的知识,基于这个特性,Wiki也可以成为一种记录工具。例如,使用它记录每天的工作日志,建立自己的知识库,甚至是离线的维基百科系统。

目前的 MediaWiki 具有在线编辑、内容回溯、版本管理、模块等功能,也是目前维基百科及其他维基媒体基金会的计划所采用的官方 Wiki 网站版本。由于使用了 PHP 语言编写,并以 MySQL 作为数据库,这使得它很容易在 LAMP 平台上进行安装与管理。

安装需求:

① PHP:安装时需要 PHP 5.3.2 以上版本。

② 数据库:如果使用 MySQL 作为数据库则要求使用 5.0.2 以上版本;如果使用的是除 MySQL 数据库以外的数据库,那么还可以选择 PostgreSQL 或 SQLite 等作为存储数据库。

③ 缩略图功能支持:如果需要使用到图片,那么还需要安装 ImageMagick 或 GD 图形库。

④ 数学公式支持:如果需要使用数学工具,那么还要安装 TeX 软件包。

⑤ 安装 MediaWiki 需要在 <http://www.mediawiki.org/wiki/MediaWiki> 上下载 MediaWiki 的源代码,目前最新版本是 1.20.2,用户可以通过以下命令来下载、解压 MediaWiki 程序:

```
# cd /var/www
# wget http://dumps.wikimedia.org/mediawiki/1.20/mediawiki-1.20.2.tar.gz
# tar xf mediawiki-1.20.2.tar.gz
# mv mediawiki-1.20.2 mediawiki
```

完成以上修改后使用浏览器打开 MediaWiki 的页面,MediaWiki 会自动显示出安装页面。读者单击“下一步”按钮后会在选择语言后进行系统检查,如图 5-16 所示,如果需要缩略图功能支持,可以使用 `apt-get install imagemagick` 命令来进行安装。

在本例中,由于访问量很小,因此没有安装 xcache 等加速器以及解决字符问题。单击“下一步”按钮后会开始配置数据库参数,如图 5-17 所示。

数据库设置方面,除非用户有特别的要求,否则使用默认配置即可,如图 5-18 所示。

单击“继续”按钮后会开始配置 Wiki 的名称以及管理员账号等信息,如图 5-19 所示。

单击“继续”按钮后,会有选择是否通过多问我一些问题来设置,如用户权限、电子邮件、版权信息、图像控制、对象缓存等方面的参数,用户也可以直接跳过这些参数的设置。

最后单击“继续”按钮开始正式安装。整个过程需要约 10 min 的时间,在安装完

MediaWiki 1.20.2配置

欢迎使用MediaWiki!

环境检查

对当前环境是否适合安装MediaWiki作基本的检查。如果您在安装过程中需要帮助, 请提供这些检查的结果。

- PHP 5.4.4-10已安装。
 - 警告: 找不到APC ☐、XCache ☐或WinCache ☐, 无法启用对象缓存。
Object caching is not enabled.
 - 已找到ImageMagick: /usr/bin/convert。如果你启用了上传功能, 缩略图功能也将被启用。
 - 使用服务器名“http://172.16.3.4”。
 - 使用服务器URL“http://172.16.3.4/mediawiki”。
 - 警告: intl PECL扩展 ☐无法处理Unicode正常化, 故只能退而采用运行较慢的纯PHP实现的方法。如果您运行着一个高流量的站点, 请参阅Unicode正常化 ☐一文。
- 环境检查已经完成。您可以安装MediaWiki。

- 语言
- 已有wiki
- 欢迎使用
MediaWiki!
- 连接到数据库
- 升级当前配置
- 数据库设置
- 名称
- 选项
- 安装
- 完成!
- 重新开始安装

图 5-16 MediaWiki 环境检查

数据库类型:

☒ MySQL

MySQL设置

数据库主机:

☐ 帮助

localhost

标识本wiki

数据库名称:

☐ 帮助

my_wiki

数据库表前缀:

☐ 帮助

用于安装的用户账号

数据库用户名:

☐ 帮助

root

数据库密码:

☐ 帮助

图 5-17 设置数据库信息

成后会提示安装成功, 并下载 LocalSettings.php 文件重新上传到/var/www/mediawiki 目录中, 随后即可访问你的 Wiki 系统了。

MediaWiki 1.20.2配置

数据库设置

供网页访问使用的数据库账号

☒ 使用 and 安装程序相同的账号

存储引擎:

☒ InnoDB

☐ MyISAM

[帮助](#)

数据库字符集:

☐ 二进制

☒ UTF-8

[帮助](#)

[上一步](#) [下一步](#)

- 语言
- 已有wiki
- 欢迎使用MediaWiki!
- 连接到数据库
- 升级当前配置
- **数据库设置**
- 名称
- 选项
- 安装
- 完成!
- 重新开始安装

图 5-18 设置数据库

名称

Wiki 的名称:

[帮助](#)

项目名字空间:

[帮助](#)

☒ 与wiki名称相同:

☐ 项目

☐ 其他 (自定义)

管理员账号

您的名字:

[帮助](#)

密码:

确认密码:

电子邮件地址:

[帮助](#)

[帮助](#)

☒ 订阅发行公告邮件列表。

- 语言
- 已有wiki
- 欢迎使用MediaWiki!
- 连接到数据库
- 升级当前配置
- 数据库设置
- **名称**
- 选项
- 安装
- 完成!
- 重新开始安装

图 5-19 名称等信息设置

5.5 共享文件

5.5.1 Samba

Samba 是一个网络共享工具,可以在 Linux 上实现 SMB (Server Message Block) 协议(也被称为 NETBIOS/LanManager 协议)。SMB 协议是 Windows 系统之间共享磁盘与打印机的协议,因此在 Linux 系统上安装 Samba 后可以实现与 Windows 系统之间的文件与打印机的共享。如果用户使用了 Windows 域或活动目录,也无须担心,因为 Samba 除了支持域以外,还可以支持 share、user、server 三种共享文件。

```
# apt-get install samba
```

使用以上命令安装完成后,Samba 会自动启动。

Samba 的配置文件是 smb.conf,存放在/etc/samba 目录下。smb.conf 配置文件分为两个部分:一个是以[global]标记的全局参数,在这里的参数会作用于整个 Samba 配置;另一个则是共享标记,这个标记会在后面介绍。

```
[global]                                # 全局标记
workgroup = CREATER                     # 设置 Samba 所在的工作组
server string = tys
dns proxy = no                           # 是否使用 dns 代理
hosts allow = 172.16.3.119               # 允许访问 Samba 的主机
log file = /var/log/samba/log. %m        # 日志文件位置
max log size = 1000                      # 最大日志尺寸
syslog = 0                              # 日志级别
panic action = /usr/share/samba/panic-action %d
security = user                          # 安全等级
encrypt passwords = true
passdb backend = tdbsam                  # 密码存储格式
obey pam restrictions = yes
unix password sync = yes                 # 是否同步系统密码
passwd program = /usr/bin/passwd %u      # 密码设置程序
passwd chat = *Enter\snew\s*\spassword:* %n\n *Retype\snew\s*\spassword:*
          %n\n *password\supdated\ssuccessfully* .
pam password change = yes                # 是否使用 pam 模块
map to guest = bad user
usershare allow guests = yes
```

对 Samba 服务器来说,最难的配置是安全方面。Samba 服务器支持以下三种安

全等级。

① user:默认的安全等级,当使用 user 安全级时会要求输入用户名和密码才能访问。

② share:共享级别的安全等级,使用这一级别时不需要输入账号和密码就可以访问 Samba 上的共享。

③ server:如果使用了 server 等级则会通过 smb 服务器来验证用户名和密码。

如果用户想要在最短的时间内完成共享,可以使用 share 级别配合 hosts allow 参数来进行控制。在这个级别中,只会允许 hosts allow 列表中所允许的客户端进行访问,而这些访问是不需要用户名与密码的。如果想让所有用户都直接访问,可以使用 hosts allow all 参数进行设置,例如:

```
security = share
hosts allow = all
```

在 smb.conf 文件中,默认的安全级别是 user 模式,即需要输入用户名与密码进行验证,只有通过后才可以访问。这将运用到 smb.conf 配置文件中的共享标记,例如在 smb.conf 文件的最底部新增加一个名为 jonsk 的共享:

```
[jonsk]    # 设置共享名
path = /mnt/usb
browseable = yes
writable = yes
create mask = 0664
directory mask = 0775
valid users = jonsk
guest ok = no
write list = jonsk
```

在上面的例子中增加了一个共享标记,即[jonsk]。这个标记就是共享的名称,也是用户能在 Windows 的网上邻居中所看到的共享名,如图 5-20 所示。



图 5-20 在 Windows 中查看 Samba 共享

如果需要设置多个共享,可以通过设置多组共享标记来实现,例如:

```
# [profiles]
#   comment = Users profiles
#   path = /home/samba/profiles
#   guest ok = no
#   browseable = no
#   create mask = 0600
#   directory mask = 0700

# [printers]
#   comment = All Printers
#   browseable = no
#   path = /var/spool/samba
#   printable = yes
#   guest ok = no
#   read only = yes
#   create mask = 0700
```

在每个共享标记中,通常有以下一些参数:

path = /mnt/usb	# 用于设置共享的路径
browseable = yes	# 是否允许客户浏览查看
writable = yes	# 是否允许客户写入
create mask = 0664	# 用户新建的文件默认权限
directory mask = 0775	# 用户新建的目录默认权限
valid users = jonsk	# 设置可访问的用户
guest ok = no	# 是否允许来宾查看
write list = jonsk	# 可写入用户

由于使用了 user 安全等级,并在 jonsk 的共享中指定了可访问用户为 jonsk,因此需要新增加一个 jonsk 用户并为他设置密码。在 Debian 所提供的 Samba 中默认使用了 tdbsam 数据库作为用户管理数据库,因此需要使用 pdbedit 程序来管理 tdb-sam 数据库。

pdbedit 参数:

- ① pdbedit -a username:新建 Samba 账户。
- ② pdbedit -x username:删除 Samba 账户。
- ③ pdbedit -L:列出 Samba 用户列表,读取 passdb.tdb 数据库文件。
- ④ pdbedit -Lv:列出 Samba 用户列表详细信息。
- ⑤ pdbedit -c "[D]" -u username:暂停该 Samba 用户账号。
- ⑥ pdbedit -c "[]" -u username:恢复该 Samba 用户账号。

新增加 jonsk 用户,并设置密码:

```
# pdbedit -a jonsk
new password:
```

```
retype new password:
```

```
Failed to add entry for user jonsk
```

设置完成后重新启动 Samba 程序,以便让新的配置生效:

```
# /etc/init.d/samba restart
```

```
[ ok ] Stopping Samba daemons: nmbd smbd.
```

```
[ ok ] Starting Samba daemons: nmbd smbd.
```

现在可以通过 Windows 的网上邻居对 Samba 服务器进行访问,Samba 会提示输入用户名与密码进行校验,如图 5-21 所示。输入正确的用户名和密码后通过验证可以看到 jonsk 共享里的内容。

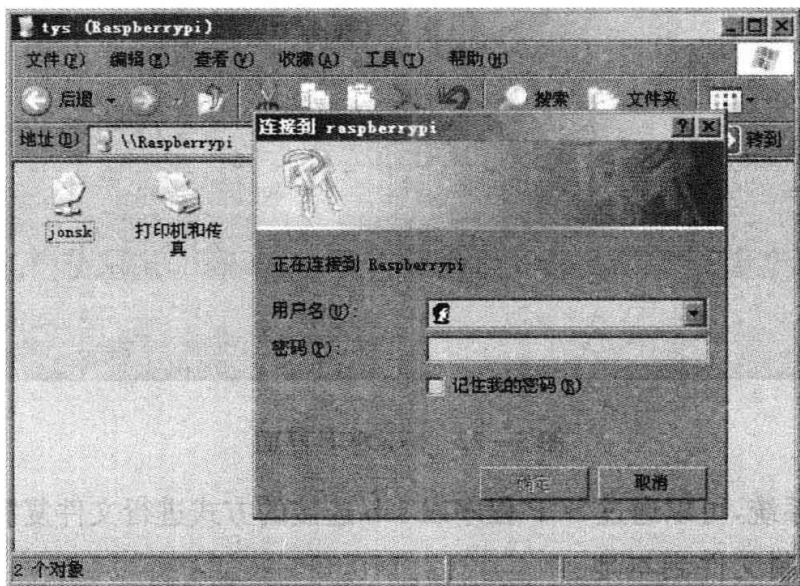


图 5-21 Samba 用户校验

5.5.2 SFTP

SFTP 是指由 openssh 程序自带的一个安全的 FTP 程序,用户不需要任何设置就可以通过 SFTP 传输文件或是完成对文件的改名、移动、删除等工作。加上整个文件传输的过程是通过 ssh 进行加密的,因此这对于那些讨厌配置 Samba 服务器的人来说是一个最好的文件共享方式。

在 Windows 下使用 SFTP 程序需要用到 WinSCP 客户端,读者可以在 <http://winscp.net/> 上下载到它的安装程序。由于 WinSCP 使用很广泛,在国内的很多网站上也可以下载到它。

第一次使用 WinSCP 客户端时,读者需要输入 ssh 主机名,也就是树莓派主机的 IP 地址,以及用户名和密码,并通过登录进行访问。如果想要以后访问时不再输入这些信息,可以选择保存成“会话”,这样下次可以直接单击“会话”进入。

如图 5-22 所示,在操作界面中,左边是本地驱动器,右边是用户目录,读者可以

使用“..”目录返回到上一层目录,并对文件进行拖动,以实现文件的上传与下载。

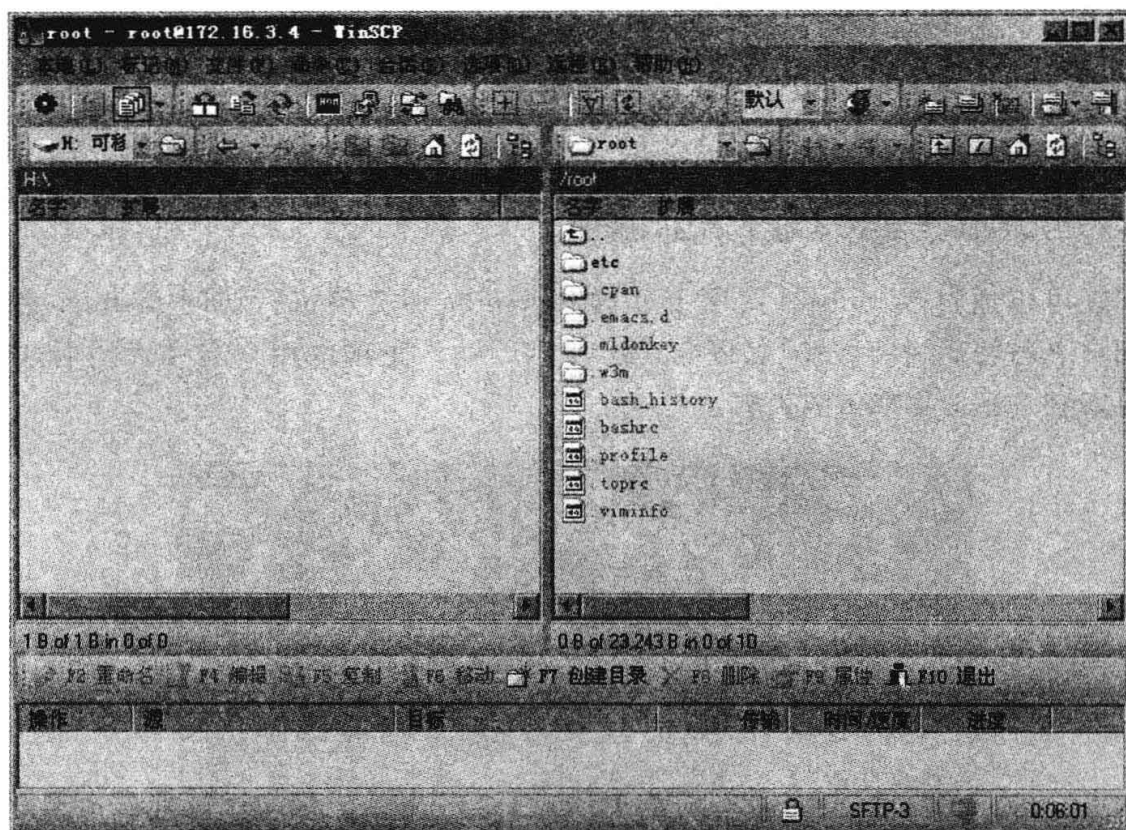


图 5-22 WinSCP 界面

对于 Linux 系统,可以通过 SCP 程序以 ssh 连接的方式进行文件复制,其格式如下:

从服务器复制文件到本地:

scp 服务器用户名@IP 地址:文件名 本地路径

从本地复制文件到服务器:

scp 本地路径 服务器用户名@IP 地址:文件名

从服务器复制文件到服务器:

scp 源服务器用户名@IP 地址:文件名 目标服务器用户名@IP 地址:文件名

例如,从服务器复制到本机:

```
# scp root@172.16.3.4:/mnt/usb/cdrom_img/debian-6.0.4-i386-businesscard.iso /
put
```

```
The authenticity of host '172.16.3.4 (172.16.3.4)' can't be established.
ECDSA key fingerprint is ef:7c:2f:81:46:35:d0:84:7a:f9:3c:6a:71:be:b1:e2.
```

```
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added '172.16.3.4' (ECDSA) to the list of known hosts.
```

```
root@172.16.3.4's password:
```

```
debian-6.0.4-i386-businesscard.iso          100 %   46MB   2.6MB/s   00:18
```

如果是第一次使用 ssh 或是 scp 程序连接到 ssh 服务器,则系统会提示是否将密码加入到本机中,这时输入 yes 即可,然后输入 root 用户的密码。

5.6 无线路由器

在无线路由器 WIFI 已经普及的今天,很多人都可以通过身边的 WIFI 网络连接到 Internet。在这种 WIFI 普及的浪潮中,无线路由器是功不可没的,仅在笔者的家里就可以通过笔记本电脑找到超过 25 个无线 WIFI 信号,而这还只是公开了 SSID 信号的,如图 5-23 所示。

如果你还没有无线路由器,那么可以在树莓派上安装一个无线网卡,将它作为一个无线路由器来使用。

5.6.1 手动配置无线路由

使用树莓派来做无线路由,首先需要确认的是无线网卡已经安装了正确的驱动并能够工作,其次所使用的无线网卡需要支持 AP 功能,否则即使正确配置了无线路,由软件也无法正常工作。

需要安装的软件有 hostapd 与 isc-dhcp-server,其中 hostapd 是一个用于访问点以及认证服务器的服务程序。通过后台程序来控制和管理认证,以实现 IEEE 802.11 接入的管理功能,目前版本的 hostapd 支持 Linux 上 Host AP,而用户可以使用以下命令来安装 hostapd 与 isc-dhcp-server 软件:

```
# apt - get install hostapd isc - dhcp - server
```

安装完成后开始配置无线路由器。首先修改网络配置,通过修改/etc/network/interfaces 文件,在文件最后增加以下内容来为无线网卡指定 IP 地址以及子网地址:

```
iface wlan0 inet static
address 192.168.2.1
netmask 255.255.255.0
```

修改/etc/default/hostapd 文件,取消“# DAEMON_CONF = ""”最前面的注释号“#”,并增加一个 hostapd 的配置文件,其位置为/etc/hostapd/hostapd.conf:

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

配置 hostapd 服务器,建立/etc/hostapd/hostapd.conf 文件并输入以下内容(以“#”号开头的行为注释行,可以不用输入):

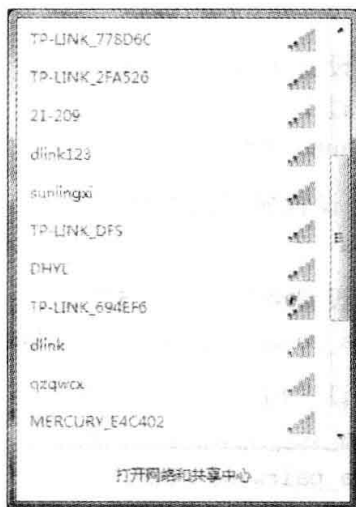


图 5-23 Windows 找到的无线信号

```

# 设置默认的接入点为无线网卡 wlan0
interface = wlan0

# 设置驱动程序为 nl80211
driver = nl80211

# 设置网卡工作模式为 802.11G
hw_mode = g

# 设置无线网络 SSID 为 RaspberryPi
ssid = RaspberryPi

# 设置无线通道为 5,如果发现连接速度慢或有干扰,也可以设置为其他数值,但最好不要超过 12
channel = 5

# 设置加密模式为 WPA2
wpa = 2

# 设置密码为 12345678aa
wpa_passphrase = 12345678aa

# 设置加密方式为 CCMP
wpa_key_mgmt = WPA-PSK
wpa_pairwise = CCMP
rsn_pairwise = CCMP
beacon_int = 100
auth_algs = 3
wmm_enabled = 1

```

修改完成后保存退出,并重新启动 hostapd 服务器:

```
# /etc/init.d/hostapd restart
```

至此,hostapd 服务器配置完成,接下来配置 isc - dhcp - server 服务器。isc - dhcp - server 是一个轻量级的 dhcp 服务器,主要是负责为通过 hostapd 服务器认证的客户端分配 IP 地址。

修改/etc/dhcp/目录下的 dhcpd.conf 文件,首先注释掉以 option domain - name 为开头的两行,以取消对域名的设置。

```

# option domain-name "example.org";
# option domain-name-servers ns1.example.org, ns2.example.org;

```

随后在文档的最后面增加以下内容:

```

subnet 192.168.2.0 netmask 255.255.255.0 {
    range 192.168.2.10 192.168.2.100;
    option routers 192.168.2.1;
    option broadcast-address 192.168.2.127;
    option domain-name-servers 8.8.8.8,8.8.4.4;
    default-lease-time 600;
    max-lease-time 7200;
}

```

修改完成后重新启动 dhcpd 服务：

```
# /etc/init.d/isc-dhcp-server restart
```

重新启动服务后,就可以通过客户端来连接无线路由了,但这时只能通过无线连接到树莓派上,而且还不能上网。其原因是还没有设置数据的转发功能,而在 Linux 系统中我们通常使用防火墙来实现这一功能。

首先清除 filter 表与 mangle 表中已使用的规则,如果读者有定义防火墙规则,那么需要检查防火墙的设置,以保证与后面的转发规则没有冲突:

```
# iptables -F
# iptables -X
```

设置转发规则:

```
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

将转发规则保存到防火墙配置中:

```
# iptables -save > /etc/iptables.up.rules
```

开启防火墙:

```
# /etc/init.d/iptables start
```

最后开启内核转发功能:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

或直接将上面的命令加入到 /etc/init.d/networking 脚本的最后面即可。

5.6.2 openwrt

目前的 openwrt 系统并不是一个正式的版本,它还有很多的不确定性,因此只推荐那些有 openwrt 或 Linux 系统使用经验的用户使用。

首先,用户需要在 <https://code.google.com/p/raspberrypi-openwrt/> 上获得最新版本的 openwrt,随后用镜像将它写入到 SD 卡中,最后使用 SD 卡开机即可。在 openwrt 中,默认的 LAN 口的 IP 地址是 192.168.1.1,子网地址为 255.255.255.0,因此读者需要将自己的计算机的网络接口设置到与 LAN 同一网段,才能通过浏览器对 openwrt 进行管理。

配置完成后即可通过浏览器对 openwrt 进行访问,如图 5-24 所示,访问 openwrt 时默认的用户名是 root 而密码则是空的,登录后会提示修改密码。

如果用户看到的界面是英文的,没有关系,可以在“系统”→“软件包管理”中安装 luci 的中文包,并在系统中选择使用中文即可。

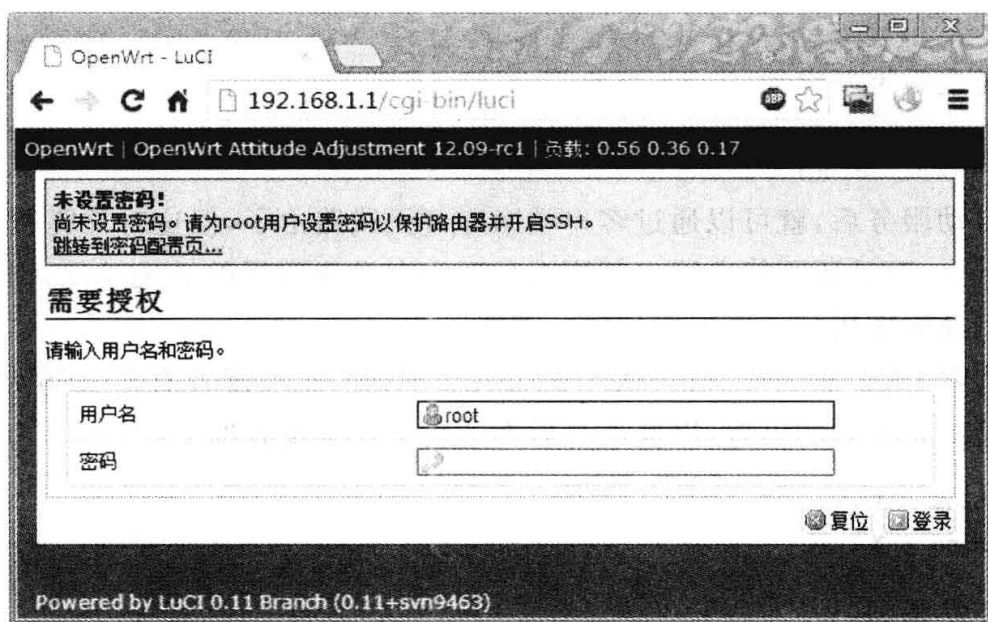


图 5-24 openwrt 登录页面

另外,也有网友(恩山 wifi 论坛,acamar 用户)提供了针对树莓派自行编译的 openwrt 版本,读者可以通过以下链接查看。

openwrt 介绍 <http://www.right.com.cn/forum/thread-111992-1-1.html>

下载连接 <http://pan.baidu.com/share/link?shareid=166150&uk=1059722366>

5.7 远程管理

对于大部分的服务器来说,远程管理是管理这些服务器的唯一方法;而对于树莓派来说,如果不将它作为桌面系统来使用,那么最方便的控制方法就是通过远程管理来操作树莓派。本节将介绍三种远程管理软件:VNC、Xrdp 以及 SSH。其中 VNC 与 Xrdp 可以远程控制桌面程序,而 SSH 则只能通过字符界面来进行控制。

5.7.1 VNC 远程桌面

VNC (Virtual Network Computing)是虚拟网络计算机的缩写,是由 AT&T 的欧洲研究实验室开发的基于 UNIX 和 Linux 操作系统的开源远程控制软件。与之前广泛使用的 Xmanager 之类的远程管理软件不同的是,VNC 由两部分组成,一部分是客户端应用程序,负责将客户指令发送到服务器端;另一部分是服务器端应用程序,负责将客户指令转换成相应的操作。在以远程方式操作目标主机时,即使操作过程中客户机与目标主机之间的网络断开,也不会影响到远程主机的运行。现在 VNC 已经移植到包括 Windows 在内的多种操作系统中,用户已经可以通过 VNC 完成跨平台的远程管理。在新版本的 VNC 中甚至可以通过 VNC 服务器内建的 WEB 接口来使用浏览器进行远程的管理工作。

目前,在 Debian 中 VNC 的服务器端包裹名称是 `tightvncserver`,而不是 VNC,用户可以使用 `apt-get install tightvncserver` 命令来安装 VNC 的服务器端:

```
# apt-get install tightvncserver
```

由于 VNC 默认是通过命令行启动的,并没有提供一个启动脚本,因此如果想让 VNC 作为服务在开机时启动,就需要建立一个脚本文件,并在相应的启动级别上建立链接文件。

使用熟悉的文本编辑器,在 `/etc/init.d/` 目录下建立一个名为 `tightvncserver` 的文件,并将以下内容输入到文件中,并保存起来。

```
#!/bin/sh
# /etc/init.d/tightvncserver
# Customised by raspicndotcom
# http://www.penguintutor.com/linux/tightvnc
# Set the VNCUSER variable to the name of the user to start tightvncserver under
VNCUSER='pi'
eval cd ~ $ VNCUSER
case "$1" in
start)
    su $ VNCUSER -c '/usr/bin/tightvncserver :1 - geometry 800x600 - depth 16 - pix-
elformat rgb565'
    echo "Starting TightVNC server for $ VNCUSER "
    ;;
stop)
    pkill Xtightvnc
    echo "Tightvncserver stopped"
    ;;
*)
    echo "Usage: /etc/init.d/tightvncserver {start|stop}"
    exit 1
    ;;
esac
exit 0
```

在上面脚本中,通过 `VNCUSER` 这个环境变量来指定一个 VNC 的用户。本例中使用的是树莓派的默认账号“pi”,如果读者想换成其他账号,也可以修改 `VNCUSER` 的参数;如果想修改 VNC 的分辨率为 1024×768 ,可以将 `su $ VNCUSER-c '/usr/bin/tightvncserver :1 - geometry 800x600 - depth 16 - pixelformat rgb565'` 中的 800×600 改为 1024×768 ;如需其他分辨率,则可以以此类推。

脚本建立完成后需要为新建的 `tightvncserver` 脚本文件赋予可执行权限,并通过 `update-rc.d` 更新 `tightvncserver` 服务为默认启动状态。

```
# chmod 755 /etc/init.d/tightvncserver
# update-rc.d tightvncserver defaults
```

最后,通过 `tightvncserver` 脚本来启动 VNC 服务:

```
#/etc/init.d/tightvncserver start
```

第一次启动需要输入 VNC 的连接密码。

在 Windows 系统中使用 VNC 需要安装 VNC 客户端,用户可以在 VNC 网站上下载。国内各大软件站点也有提供下载。

安装完成后可以通过 VNC Viewer 程序来连接树莓派。需要说明的是,在对话框中的 Server 文本框中除了填写目标主机 IP 地址外,还需要加上 VNC 服务器的端口号 5901,否则是无法连接成功的,如图 5-25 所示。

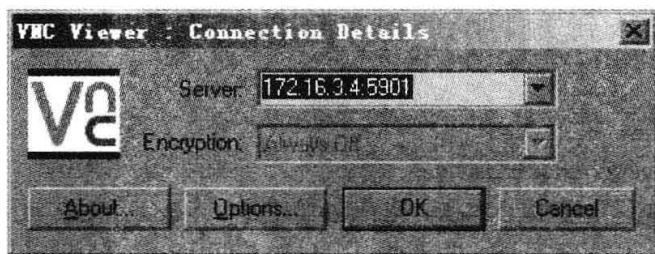


图 5-25 VNC Viewer 登录设置

单击 OK 按钮后会提示输入密码,如果 VNC 的 Username 一栏无法填写,则说明 VNC 指定了用户名,用户需要输入相应的密码才能连接成功。在树莓派的 VNC 服务器中,我们指定 pi 用户作为 VNC 账号使用,因此这里也需要输入 pi 用户的密码,如图 5-26 所示。

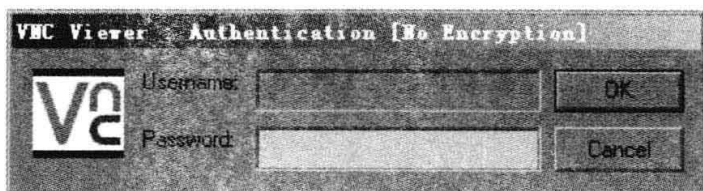


图 5-26 VNC 如果指定了用户则只需要输入密码

成功连接后等上一会儿,会出现树莓派的桌面环境,如图 5-27 所示,而这个环境则是属于 pi 用户的。

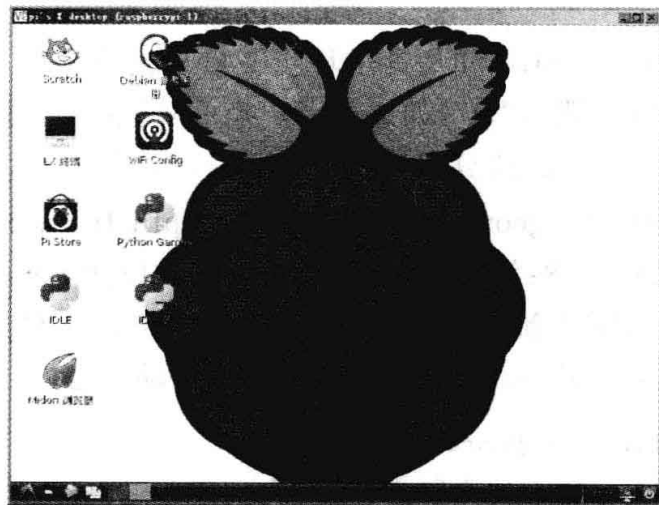


图 5-27 VNC 登录界面

5.7.2 Xrdp 远程桌面

如果用户每天都使用的是微软的系统,那么肯定会对 Xrdp 更有兴趣。事实上,与 VNC 相比起来,我更喜欢使用 Xrdp 的远程管理系统,因为它使用的是微软的 Xrdp 协议,客户端上也不需要重新安装一个连接程序,而是直接使用 Windows 提供的远程桌面连接工具就可以完成跨平台的远程管理工作。不过也正是因为这些优点,使得 Xrdp 更适合在 Windows 的客户端中使用,而不适宜在 Linux/Unix 或 MAC 中推广。

与前面的程序一样,还是通过 apt 程序来安装 Xrdp:

```
# apt - get install xrdp
```

安装完成后 Xrdp 程序会自动启动,这时就可以回到 Windows 系统中使用“远程桌面连接”程序进行远程管理了。与 VNC 不同的是,微软的“远程桌面连接”工具可以直接在连接时通过显示选项卡选择分辨率,而不需要在服务器上设置,如图 5-28 所示。

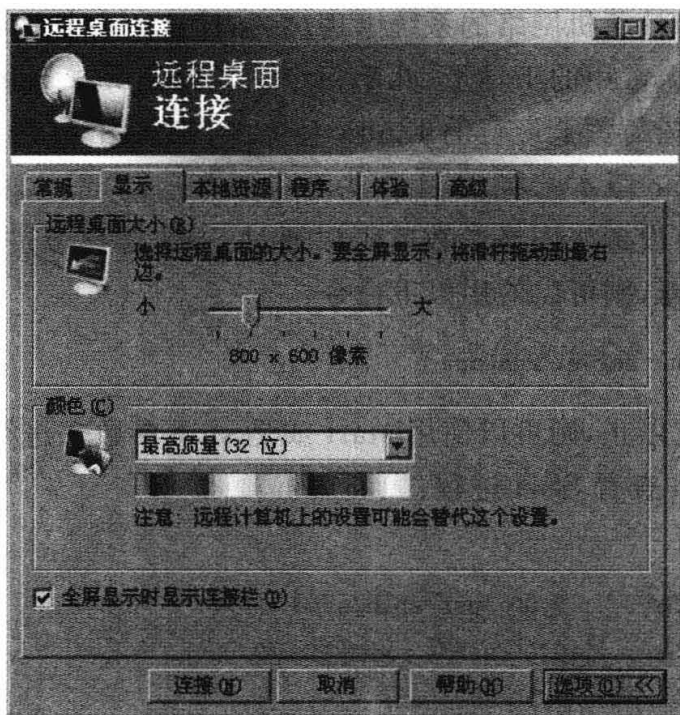


图 5-28 Windows 远程桌面连接设置

在连接窗口中输入用户名和密码后即可进入相应用户的桌面环境,如图 5-29 所示。

5.7.3 SSH

SSH 是 Secure Shell 的缩写,是目前使用最广泛的安全的远程连接工具。最初,它只是在 UNIX 系统上应用,但是强大的安全性及开放的源代码使 SSH 在极短的时间内得到了迅猛的发展,几乎所有的 UNIX/Linux、Windows、MAC 等系统上都出现了 SSH 的版本。

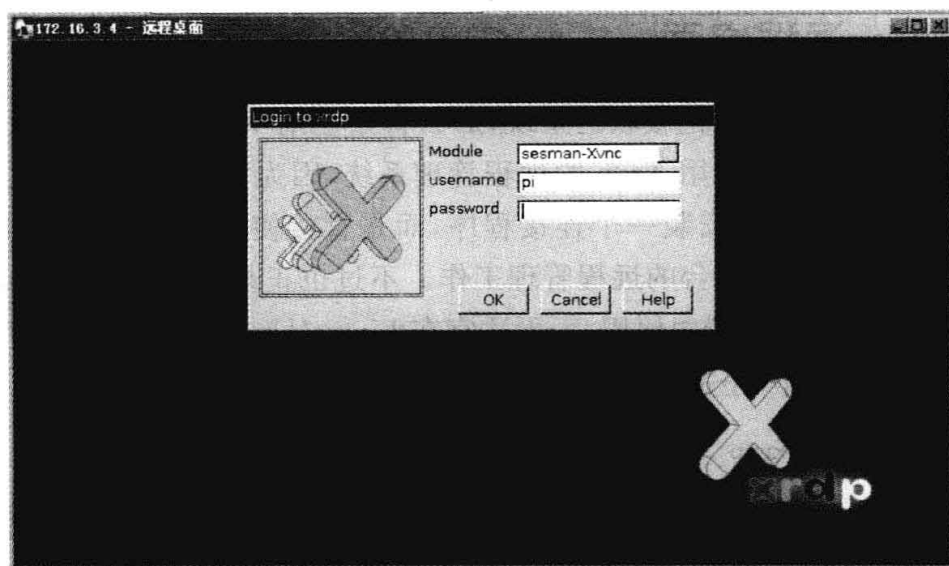


图 5-29 Xrdp 登录

如果读者是一个计算机网络的系统管理员,估计这一小节是可以直接跳过的,因为 SSH 实在是一个太常用的工具了,几乎每个使用过 Linux 系统的人都会用过它。目前,几乎所有能在树莓派上运行的 Linux 系统中,都将 SSH 服务器作为默认的安装,而树莓派的 Debian 版本则将 SSH 作为开机服务启动,以方便用户进行管理,在树莓派通电后只需要设置好网络参数就可以通过 SSH 客户端进行连接。如果需要重新启动 SSH 服务器,则可以使用以下命令:

```
# /etc/init.d/ssh - server restart
```

如果需要启动或停止,则可以使用 start 或是 stop 参数来进行,启动完成后可以使用 `ps ax | grep ssh` 查看 SSH 进程:

```
# ps ax | grep ssh
782 ?      Ss      0:00 /usr/sbin/sshd
810 ?      Ss      0:00 sshd: root@pts/0
999 pts/0  S+      0:00 grep ssh
```

可以看到 SSH 服务正在运行中,并且有一个 root 用户通过 SSH 连接到树莓派上。

SSH 的客户端有很多,在 Windows 系统下,笔者推荐使用 PuTTY。PuTTY 是一套免费的跨平台的远程登录工具,整个软件只有 400 多 K,非常小巧,而且不需要安装就可以使用。除了支持 SSH1、SSH2 以外,还可以提供 SSH 通道、多种终端支持,快速复制、粘贴等功能,绝对是每个系统管理员都会必备的工具。除了 PuTTY 程序之外,这套工具还包含了以下组件:

- PuTTY Telnet 和 SSH 客户端。
- PSCP 命令行模式的 SCP 的客户端,实现文件复制。

- PSFTP 命令行模式的 SFTP 客户端,实现加密的 FTP 文件传输。
- PuTTYtel 实现 Telnet 客户端。
- Plink 命令行工具,执行远程服务器上的命令。
- Pageant PuTTY、PSCP、Plink 的 SSH 认证代理。
- PuTTYgen RSA 和 DSA 密钥生成工具。

用户可以在以下网站下载到它的最新版本:

- PuTTY 官网 <http://www.putty.org/>
- PuTTY 下载页面 <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

国内各大网站也提供 PuTTY 程序的下载,但需要注意的是,如果不是在官方网站上下载的 PuTTY 程序,则需要校验一下 md5 或是 sha1 编码,以保证程序没有被改动过。

PuTTY 内置中文支持,只需要在 PuTTY 中选择 Windows 下的 Appearance 选项并修改 Font settings(字体设置),将当前字体修改为中文字体。读者可以使用仿宋、新宋、黑体等字体,如图 5-30 所示。

133

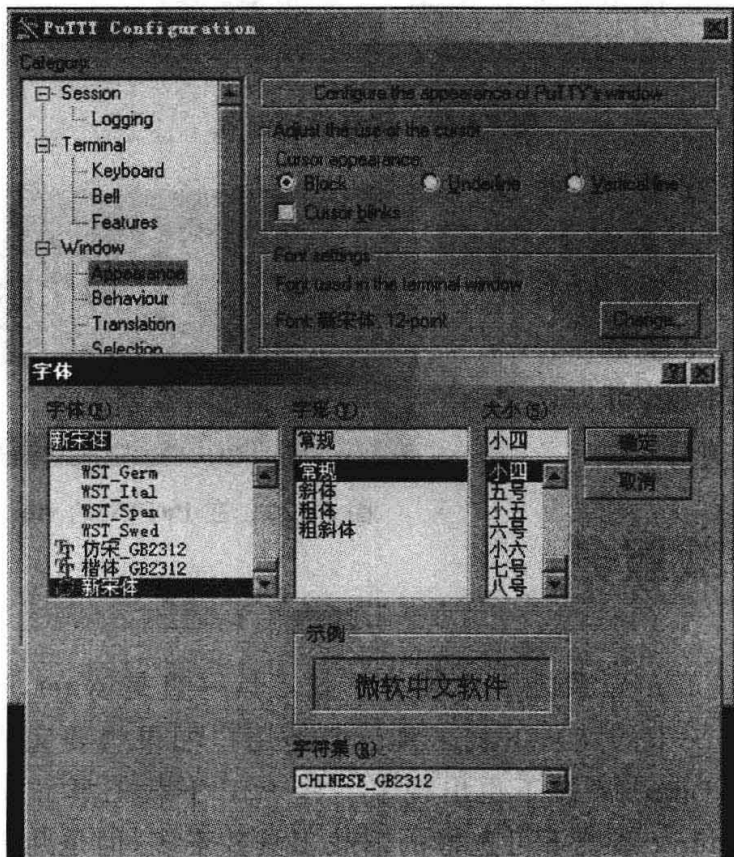


图 5-30 PuTTY 中文设置

修改 Translation 中的语言编码,改为 UTF-8(如图 5-31 所示),以加入中文支持。

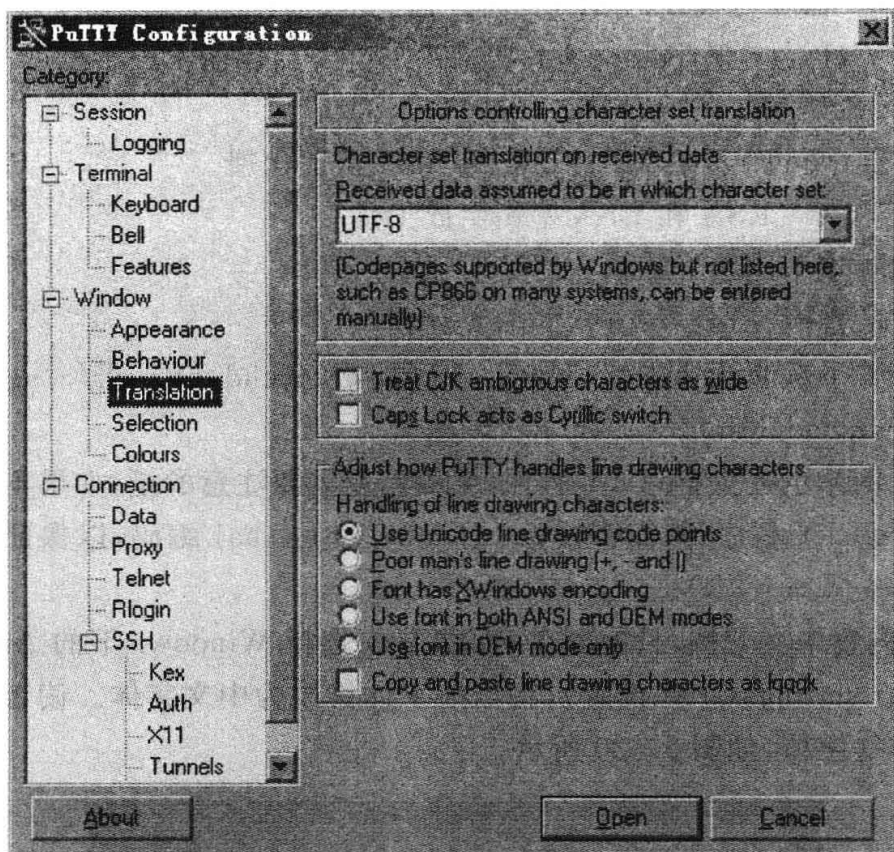


图 5-31 语言代码设置

设置完成后就可以通过 Session 选项连接到树莓派上了。图 5-32 是在 PuTTY 窗口中打开 vim 这类的编辑器,通过 Windows 上的输入法来输入中文,可以看出所有的中文都能完美显示。

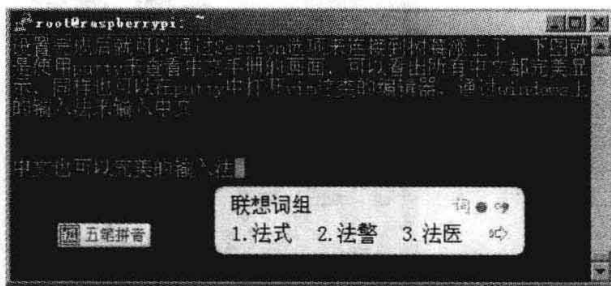


图 5-32 在 PuTTY 的 vim 中输入中文

5.8 P2P 下载服务

这么多年来,在 Linux 系统下传统的下载工具一直是 wget,速度快,支持协议多,使得它成为 Linux 下的默认下载工具。但是用户如果想要完成 P2P 下载,wget 则无能为力。因此,Linux 系统下也出现了一些 P2P 下载工具。本着不重复发明轮子的设计思想,在 Linux 系统下可选择的下载工具并不多,但是几乎每一个工具都有大量用户,每种都有自己的特色。例如:Rtorrent 可以在终端下以文本窗口运行,而 Transmission 可以支持 bt 与 pt 下载,Mldonkey 则可以完成电驴的下载。读者可以根据自己的需要选择工具。

5.8.1 Rtorrent

Rtorrent 是一款简洁、轻量级的 BT 客户程序,它使用 ncurses 库和 C++ 编写,因此可以完美地在运行终端中以文本窗口模式运行。这样就可以最大限度地节省系统资源。配合 screen 这样的多重视窗管理程序可以轻松地在多个平台的终端中进行管理。

安装并启动 Rtorrent:

```
# apt-get install rtorrent
# rtorrent
```

启动完成后可以看到 Rtorrent 界面,如图 5-33 所示。不要奇怪,之前就说过 Rtorrent 是基本文本窗口,所以读者看到的 Rtorrent 的主界面都是以文本字符组成的。由于是由文本字符组成的界面,因此也无法使用鼠标来进行控制,在 Rtorrent 中,所有的操作都是通过键盘来完成的。

Rtorrent 常用组合键如下:

开始下载: Ctrl + S

暂停下载: Ctrl + D

停止下载: Ctrl + K

清除 Rtorrent 下载列表: Ctrl + D

置最大上传速度: [增加] a / s / d, [减少] z / x / c

校验下载的文件: Ctrl + R

退出 Rtorrent 程序: Ctrl + q



图 5-33 Rtorrent 界面

如果需要增加下载链接,可以直接在 Rtorrent 界面中输入回车,在 load. normal> 提示符下输入 bt 种子文件的路径即可:

```
load. normal> /root/xlrq.torrent
```

如果需要查看下载的详细信息,可以使用上、下光标键来选择下载的任务,选中的任务会在最前方出现“*”号字符,然后通过左、右光标键来查看任务的详细信息,如图 5-34 所示。

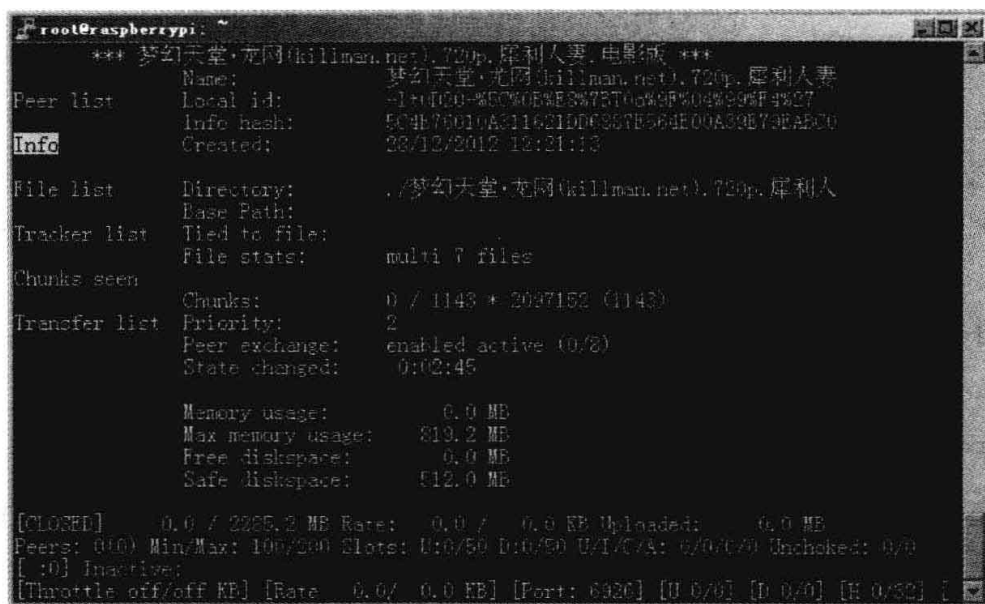


图 5-34 Rtorrent 查看下载信息

5.8.2 Transmission

Transmission 是一个 bt 与 pt 的客户端,整个系统由 C 语言开发而成,拥有内存消耗少、速度快、界面精简等特点,同时还支持以命令行、WEB 页面、GTK+ 等多种界面进行控制。对于树莓派而言,推荐使用 WEB 页面控制的方式来运行,如图 5-35 所示,以最大限度节省硬件资源。

Transmission 的主要特点:

- ① 支持磁性链接;
- ② 支持点对点的加密;
- ③ 内置 WEB 服务器可以通过 WEB 页面远程控制;
- ④ 支持带宽限制;
- ⑤ 支持微型传输协议 μ TP。

由于 Transmission 支持多种控制方式,每种控制方式都有自己的专用安装包,因此在安装时除了需要安装公共软件包外,还需要选择相应的控制界面的安装包。

以下五种模式安装其中一种即可:

- | | |
|-------------------------|-----------------------------|
| transmission-cli | 命令行界面的安装包; |
| transmission-daemon | 以服务模式运行的安装包,支持以 WEB 页面进行管理; |
| transmission-gtk | GTK 界面的安装包; |
| transmission-qt | QT 界面的安装包; |
| transmission-remote-cli | 命令行模式下的字符菜单界面。 |

如果用户使用的是图形界面,则可以选择安装 GTK 或 QT 的安装包(图形界面需要在 X Windows 环境下运行);如果是通过远程运行,则推荐使用 transmission -

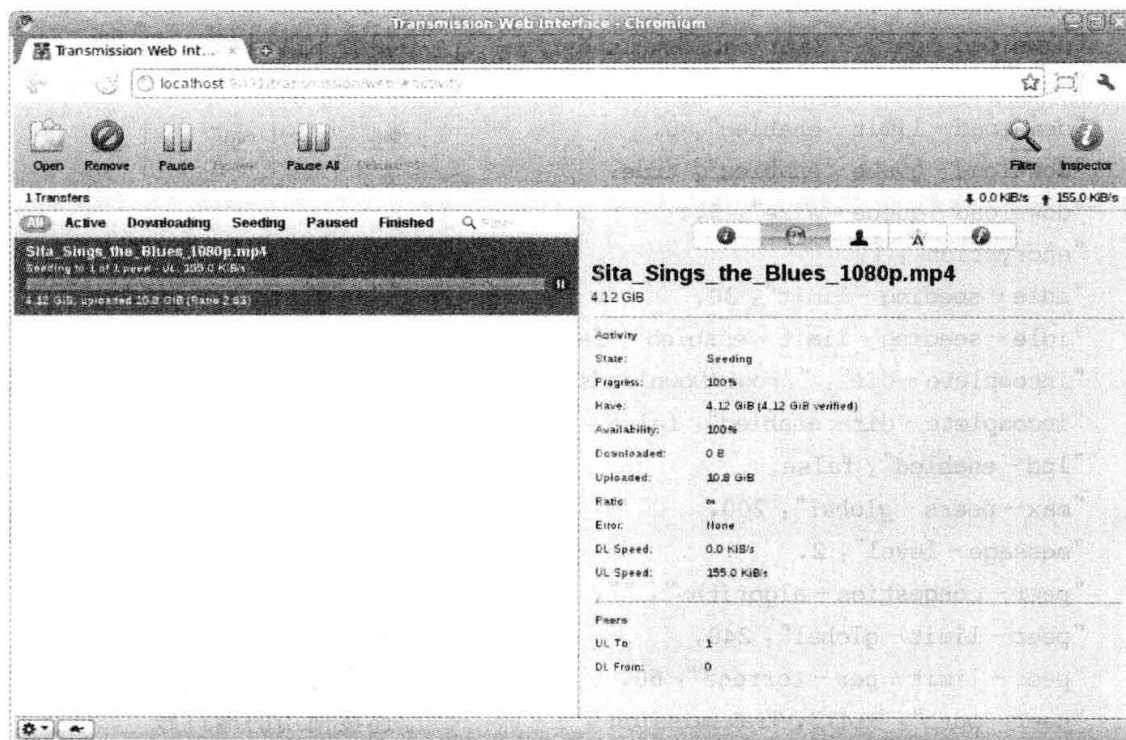


图 5-35 Transmission 的 WEB 页面控制界面

daemon 包,这样可以通过 WEB 页面进行管理。

```
# apt-get install transmission-daemon
```

安装完成后 Transmission 会以服务的模式启动。

由于 Transmission 服务每次关停时都会更新它的配置文件 settings.json,因此对 settings.json 文件修改时需要先将 Transmission 服务停止:

```
# /etc/init.d/transmission-daemon stop
```

```
[ ok ] Stopping bittorrent daemon: transmission-daemon.
```

修改/etc/transmission 目录下的 settings.json 文件:

```
{
  "alt-speed-down": 50,
  "alt-speed-enabled": false,
  "alt-speed-time-begin": 540,
  "alt-speed-time-day": 127,
  "alt-speed-time-enabled": false,
  "alt-speed-time-end": 1020,
  "alt-speed-up": 50,
  "bind-address-ipv4": "0.0.0.0",
  "bind-address-ipv6": ":::",
  "blocklist-enabled": false,
  "blocklist-url": "http://www.example.com/blocklist",
  "cache-size-mb": 4,
  //启动黑名单支持
}
```

```

"dht-enabled": true,
"download-dir": "/mnt/usb/downloads/", //设置下载时储存的位置
"download-limit": 100,
"download-limit-enabled": 0,
"download-queue-enabled": true,
"download-queue-size": 5,
"encryption": 1,
"idle-seeding-limit": 30,
"idle-seeding-limit-enabled": false,
"incomplete-dir": "/root/Downloads", //设置下载时未完成时的储存位置
"incomplete-dir-enabled": false,
"lpd-enabled": false,
"max-peers-global": 200,
"message-level": 2,
"peer-congestion-algorithm": "",
"peer-limit-global": 240, //最大连接数
"peer-limit-per-torrent": 60, //单个种子连接数
"peer-port": 51413, Transmission //连接使用的端口号
"peer-port-random-high": 65535,
"peer-port-random-low": 49152,
"peer-port-random-on-start": false,
"peer-socket-tos": "default",
"pex-enabled": true,
"port-forwarding-enabled": false,
"preallocation": 1,
"prefetch-enabled": 1,
"queue-stalled-enabled": true,
"queue-stalled-minutes": 30,
"ratio-limit": 2,
"ratio-limit-enabled": false,
"rename-partial-files": true,
"rpc-authentication-required": true, //用户验证,改成 false 可以关闭
"rpc-bind-address": "0.0.0.0",
"rpc-enabled": true,
"rpc-password": "12345678", //WEB 登录密码,设置后重新启动
//服务不会再以明文显示
"rpc-port": 9091, //设置 WEB 页面管理端口
"rpc-url": "/transmission/",
"rpc-username": "jonsk", //WEB 登录用户名
"rpc-whitelist": "0.0.0.0", //可访问 WEB 的客户端 IP 地址,
//0.0.0.0 表示允许所有地址访问
"rpc-whitelist-enabled": false, //白名单过滤,改成 false 可以关闭
"scrape-paused-torrents-enabled": true,
"script-torrent-done-enabled": false,
"script-torrent-done-filename": "",

```



```

"seed-queue-enabled": false,
"seed-queue-size": 10,
"speed-limit-down": 100,                //设置最大下载速度
"speed-limit-down-enabled": false,       //启用限制下载速度
"speed-limit-up": 100,                   //限制上传速度
"speed-limit-up-enabled": false,         //启用限制上传速度
"start-added-torrents": true,
"trash-original-torrent-files": false,
"umask": 18,
"upload-limit": 100,
"upload-limit-enabled": 0,
"upload-slots-per-torrent": 14,         //每个种子的上传连接数量
"utp-enabled": true
}

```

修改最多的主要是 WEB 登录用户与密码以及可访问 WEB 的客户端 IP 地址, 如果读者想要省事, 则可以直接关闭白名单过滤与用户验证, 这样修改完成后启动 Transmission 即可正常访问。

```

# /etc/init.d/transmission-daemon start
[ ok ] Starting bittorrent daemon: transmission-daemon.

```

现在可以在 WEB 页面上输入 `http://Transmission` 服务器所在的 IP+端口号 (9091) 进行管理与下载了, 下面的例子笔者输入的是 `http://172.16.3.4:9091`。用户名与密码就是之前在 `settings.json` 文件中所定义的 `rpc-password` 与 `rpc-username`。

用户可以点击 open Torrent 按钮增加新的 bt 或是 pt 种子进行下载, 在下载任务上双击可以查看具体信息。如果用户喜欢以程序方式进行管理, 也可以在下面的地址下载 Transmission Remote 软件管理:

<http://code.google.com/p/transmission-remote-dotnet/w/list>

如果下载任务中显示 Tracker returned an error: You already are downloading the same torrent. You may only leech from one location at a time. 则无须担心, 一段时间后 Transmission 会自己解决这个问题(可能是几小时, 也可能是十几个小时)。

5.8.3 Mldonkey

之前介绍的 Transmission 只能完成 bt 与 pt 的下载, 如果读者需要支持其他如 Ed2k 之类的下载就只能选择 Mldonkey。除了 bt 外, Mldonkey 还支持 eDonkey、Kad、HTTP/FTP 等下载方式。但对于国内用户来说, 除了 eDonkey 以外, 其他的下载支持都没有什么用, 与 Transmission 一样, Mldonkey 除了支持 html 页面进行管理外, 还支持 Telnet 接口进行管理。另外, 还有很多以 Telnet 为基础开发的图形管理工具, 如 KMLdonkey 和 Sancho, 都可以方便地对 Mldonkey 进行管理。与其他的

Ed2k 客户端相比, Mldonkey 最大的特点就是支持同时连接多服务器, 从而大大提高了可用源的数量, 因此它的速度比其他的 ed2k 客户端要快得多。

安装 Mldonkey。由于 Mldonkey 是由 ocaml 语言编写的, 因此系统会自动安装上 ocaml 语言包。

```
# apt-get install mldonkey-server
```

在安装过程中会提示是否需要在开机时自动启动 Mldonkey, 读者可以根据自己的需要选择。如果打算长期下载, 那么推荐在开机时启动 Mldonkey, 如图 5-36 所示。

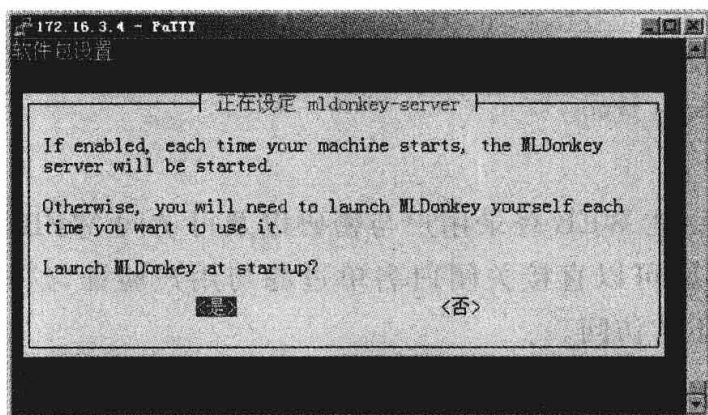


图 5-36 Mldonkey 启动设置

安装完成后 Mldonkey 会作为服务自动启动, 用户可以使用 `ps ax | grep mlnet` 命令进行查看:

```
# ps ax | grep mlnet
13397 ?    Sl  0:45 /usr/bin/mlnet -log_file /var/log/mldonkey/mlnet.log -pid /var/
run/mldonkey
13402 pts/0  S+   0:00 grep mlnet
```

若需要使用 Telnet 程序对 Mldonkey 进行设置, 可以允许其他 IP 地址的用户对 Mldonkey 的 WEB 进行访问与控制。如果系统没有 Telnet 程序可以通过下面的命令进行安装。

```
# /etc/init.d/mldonkey-server start
```

```
Starting MLDonkey: mlnet.
```

```
# apt-get install telnet
```

Mldonkey 可以通过 Telnet 对 4000 号端口进行管理。

```
# telnet 127.0.0.1 4000
```

```
Trying 127.0.0.1...
```

```
Connected to 127.0.0.1.
```

```
Escape character is '^['.
```

```
Welcome to MLDonkey 2.8.5
```

```

Welcome on mldonkey command - line

Use ? for help

Mldonkey command - line:
> auth admin ""    允许访问
Full access enabled

Mldonkey command - line:
> passwd newpasswd  设置密码
Password of user admin changed

Mldonkey command - line:
> set allowed_ips "授权客户端 ip 地址"
> quit            退出

```

设置完成后可以在浏览器里输入 `http://ip 地址:4080` 进行管理,如图 5-37 所示,用户可以通过空白栏边上的 input 按钮增加新的下载链接。

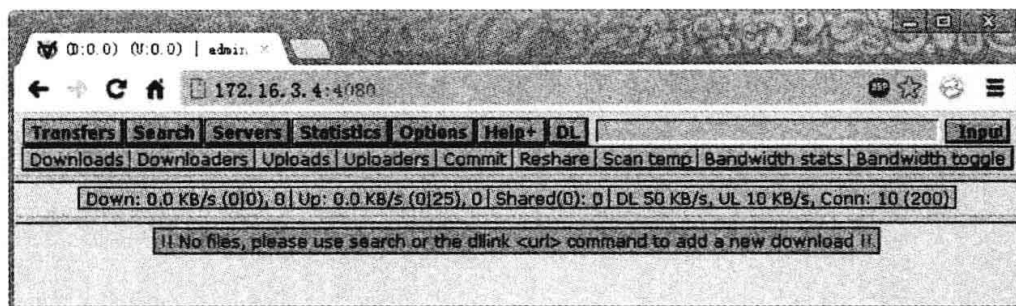


图 5-37 Mldonkey 的 WEB 界面

除了在空白栏中增加新的链接外,还可以在这里导入各种服务器列表,例如 `server_met` 文件位置:

```
servers http://eserver.googlecode.com/svn/server.met
```

在树莓派中,如果 Mldonkey 以服务模式运行,则它的配置文件会存放在 `/var/lib/mldonkey` 目录中,而不像用户模式,在用户目录下生成 `mldonkey` 文件来存放配置文件:

```
# ls /var/lib/mldonkey
```

<code>downloads.ini</code>	储存了 Mldonkey 全局性的设置,这个配置文件十分重要。
<code>searches.ini</code>	储存自定义的搜索问题。
<code>servers.ini</code>	储存你能连接的服务器列表。
<code>friends.ini</code>	储存你好友的名字和对应的 IP。
<code>stats.ini</code>	储存上传下载统计信息。
<code>file_sources.ini</code>	储存能连接到的源。
<code>files.ini</code>	当前下载的文件列表及其 md4 - hash。

shared_files_new.ini 存储共享的文件列表及其 md4 - hash。

对于每一种 P2P 协议都有各自的配置文件：

donkey.ini 储存 eDonkey、Overnet 的网络设置。

bittorrent.ini 储存 bt 的网络设置。

directconnect.ini 储存 DirectConnect 的网络设置。

fasttrack.ini 储存 Fasttrack 的网络设置。

fileTP.ini 储存 FTP/HTTP 的网络设置。

如果读者不喜欢手动修改配置文件,也可以使用 Mldonkey 的管理页面中的 Options 选择来进行设置,所以配置修改后立刻生效,不像手动修改配置,还需要重启 Mldonkey,因此推荐使用此方法。

第 6 章

扩展接口及应用

树莓派除了提供类似 PC 一样的功能外,还提供了扩展接口 P1,它位于树莓派的左上角,由 26 个 2.54 mm 间距的双排插针组成,方便了用户的使用和扩展,如图 6-1 所示。

树莓派扩展接口 P1

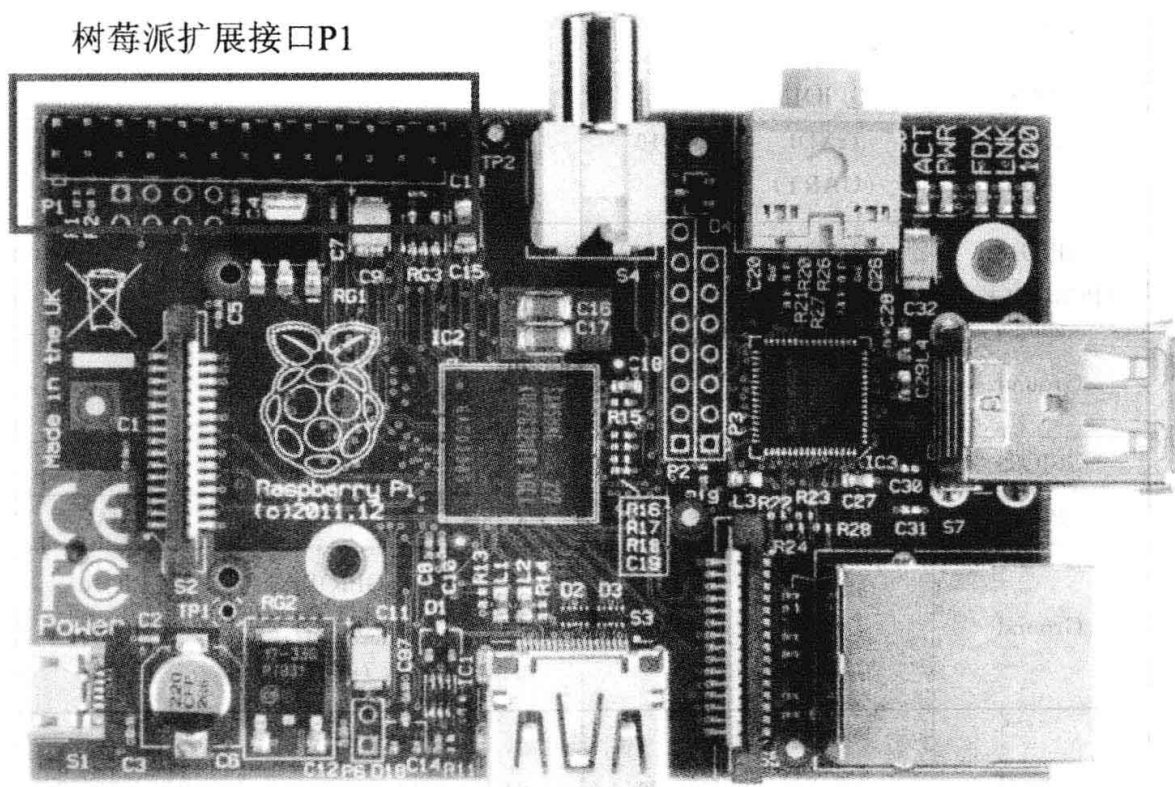


图 6-1 树莓派扩展接口 P1

关于这 26 个针脚的作用,读者可参考表 6-1。需要注意的是,第一版和第二版的树莓派在针脚接口上是有区别的。在第一版中,针脚接口 4、9、14、17、20 和 25 是保留未使用的,如果同时使用 GPIO 的 3.3 V 和 5 V 接口进行供电,针脚接口 1 的输出是固定输出 1~50 mA 电流,但是针脚接口 2 的输出是基于 USB 电源的输入电流。如果用户使用的是 1 A 的电源,则针脚接口 2 的输出为 300 mA,因为 B 型树莓派自身需要 700 mA。

表 6-1 树莓派 GPIO 针脚说明

第二版	第一版	针脚接口编号		第一版	第二版
5 V	5 V	2	1	3.3 V	3.3 V
5 V	—	4	3	GPIO0 (SDA) (I ² C)	GPIO2 (SDA) (I ² C)
Ground	Ground	6	5	GPIO1 (SCL) (I ² C)	GPIO3 (SCL) (I ² C)
GPIO14 (TXD) (UART)	GPIO14 (TXD) (UART)	8	7	GPIO4 (GPCLK0)	GPIO4 (GPCLK0)
GPIO15 (rXD) (UART)	GPIO15 (rXD) (UART)	10	9	—	Ground
GPIO18 (PCM_CLK)	GPIO18 (PCM_CLK)	12	11	GPIO17	GPIO17
Ground	—	14	13	GPIO21 (PCM_DOUT)	GPIO27
GPIO23	GPIO23	16	15	GPIO22	GPIO22
GPIO24	GPIO24	18	17		3.3 V
Ground	—	20	19	GPIO10 (MOSI) (SPI)	GPIO10 (MOSI) (SPI)
GPIO25	GPIO25	22	21	GPIO9 (MISO) (SPI)	GPIO9 (MOSI) (SPI)
GPIO8 (CE0) (SPI)	GPIO8 (CE0) (SPI)	24	23	GPIO11 (SCLK) (SPI)	GPIO11 (MOSI) (SPI)
GPIO7 (CE1) (SPI)	GPIO7 (CE1) (SPI)	26	25	—	Ground

在表 6-1 中,这 26 个针脚中其实并不只有 GPIO 接口,它还包含了 I²C、

UART、SPI 等接口。通过这些接口,用户可以配合相应的软件来完成各种电子项目或直接连接各种扩展设备。

本节将由浅入深地介绍并完成三个小例子:

① 使用 GPIO 接口建立一个可以通过命令进行手动控制的 LED 灯电路,然后对这个电路进行扩展,将它变成一个内存指示灯系统,并通过 Linux 系统下最基本的 bash 脚本来完成对树莓派系统内存使用情况的监控。

② 通过 I²C 接口连接一个 LM75 温度传感器,来获得当前温度的信息。例中将会使用树莓派所推荐的 python 语言来获得 LM75 温度传感器的信息。

③ 树莓派连接一个高科技设备——GPS 系统,除了使用 GPS 设备获得当前的地理位置与时间外,还会介绍在 python 的程序中如何读取 GPS 信息。

在开始这三个小例子之前,先了解一下这三个例子所使用接口的具体信息:

① GPIO 是 General Purpose Input/Output (通用输入/输出)的缩写,它通过读取接口所输出的高低电平状态来获得相应的设备信息。因此用它来作为控制和监视某些简单的设备极为方便。

② I²C 是 Inter-Integrated Circuit(内部整合电路)的缩写,是飞利浦公司推出的一种两线制串行通信总线,简单地说就是,连接到总线的器件间传递信息的通道。除此之外,I²C 总线还具有连接简单、信号线少、可以挂载多个从设备等优点,因此 I²C 接口被广泛应用于各种传感器芯片上。在数据采集应用中,主机通过 I²C 接口在总线上挂载多个传感器,从而可以获取多个传感器的值,对收集到的数据进行统计和分析。

③ UART 是一种简单的串口通信标准,英文全称为 Universal Asynchronous Receiver/Transmitter(通用串行异步收/发接口),经常用于主机与辅助设备之间实现全双工信息传输和接收。

6.1 使用 GPIO 建立内存警报指示灯

如果说初学计算机编程,最简单的入门程序是在屏幕上显示“Hello World!”,那么学习控制 GPIO 接口,最简单的入门实验就是通过 GPIO 控制发光二极管(LED)闪烁发光。LED 灯是一个小型电子元件,只需要很低的电流和电压,就能如同灯泡般发光。虽然这个例子很简单,不过它并没有太多的用处,因此没有多少用户会经常通过手动输入命令来控制 LED 灯的显示。下面这个实例将会更进一步,完成一个内存警报指示灯,这样我们就可以通过 LED 灯来获得内存的使用信息了。

完成这个实验需要以下工具:

- ① 一个面包板或是电路板;
- ② 三个不同颜色的 LED 发光二极管(本书使用的是红、黄、绿三种颜色);
- ③ 三个 220 Ω 左右的电阻;

第6章 扩展接口及应用

④ 四根跳线。

以上材料在一般的电子商店里都可以很容易地买到,如果实在无法买到,可以在 ukonline2000 网站(<http://ukonline2000.taobao.com/>)上进行购买。ukonline2000 提供了一个供树莓派用户使用的新手套装,里面包含了这个实验所需的电子零件。

6.1.1 建立基本电路

准备好以后,首先通过面包板建立一个基本的电路,可以参考图 6-2 进行连接。需要注意,LED 灯只能让电流从一个方向经过,如果接反了,LED 灯就不会亮了。

如果连接没有问题,那么就启动树莓派吧。你会看到 LED 灯亮起来,下面要做的事就是通过 GPIO 来控制 LED 灯的亮与灭的状态。

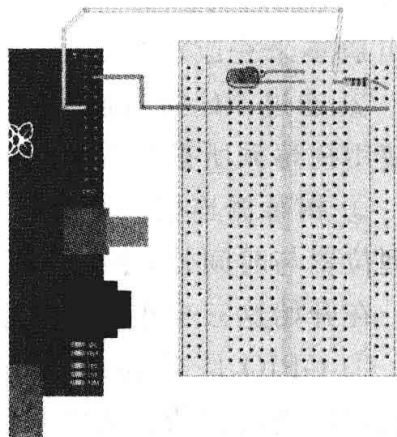


图 6-2 电路连接示意图

6.1.2 通过 GPIO 控制 LED 灯

按照传统的方法直接控制硬件是一件非常麻烦的事情,因为这需要掌握大量的软、硬件知识,不过在树莓派上使用 GPIO 却没有这么麻烦,因为通过 WiringPi 这个开源项目,用户可以不用编写任何代码就可以很容易地控制 GPIO。由于 WiringPi 并没有做成 Debian 的软件包,并且作者也没有提供打包好的压缩包下载,因此想要使用它需要通过 git 程序来获得。下面的命令可以安装 git 程序,并获得 WiringPi 的源代码:

```
# apt - get install git - core
# apt - get update
# apt - get upgrade
# git clone git://git.drogon.net/wiringPi
```

当下载过程完成后,读者可以看到在当前目录下有一个 WiringPi 目录,接下来使用以下命令对 WiringPi 进行编译安装:

```
# cd wiringPi
# ./build
```

如果一切顺利,安装完成后 GPIO 等程序会自动安装到你的系统中。用户可以通过 gpio-v 命令来测试一下,如果能看到以下信息则表示 GPIO 安装成功。

```
# gpio - v
gpio version: 2.07
Copyright (c) 2012 - 2013 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio - warranty
```

This Raspberry Pi is a revision 2 board.

由于 WiringPi 的端口定义与树莓派的端口定义有一些区别, WiringPi 为了简化端口的定义, 将树莓派不连续的 GPIO 端口号按照 0~16 的顺序一一对应编号, 因此用户在树莓派上使用 GPIO 时, 需要按照 WiringPi 的定义方式来使用 GPIO 接口。具体的定义方式可以通过图 6-3 来查找相应的端口编号。

wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin
-	-	3.3v	1 2	5v	-	-
8	R1:0/R2:2	SDA0	3 4	5v	-	-
9	R1:1/R2:3	SCL0	5 6	0v	-	-
7	4	GPIO7	7 8		14	15
-	-	0v	9 10		15	16
0	17	GPIO0	11 12	GPIO1	18	1
2	R1:21/R2:27	GPIO2	13 14	0v	-	-
3	22	GPIO3	15 16	GPIO4	23	4
-	-	3.3v	17 18	GPIO5	24	5
12	10	MOSI	19 20	0v	-	-
13	9	MISO	21 22	GPIO6	25	6
14	11	SCLK	23 24	CE0	8	10
-	-	0v	25 26	CE1	7	11
wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin

注: R1 代表老版树莓派, R2 代表新版树莓派

图 6-3 WiringPi 与树莓派的 GPIO 端口对照表

下面可以尝试使用 gpio 命令来控制 LED 灯。

① 使用以下命令配置 GPIO 为输出模式:

```
# gpio mode out
```

② 使用以下命令点亮 LED 灯:

```
# gpio write 0 1
```

③ 使用以下命令关闭 LED 灯:

```
# gpio write 0 0
```

④ 通过以下命令读取 GPIO 的状态:

```
# gpio read 0
```

0

6.1.3 建立内存警告系统

前面已经完成一个简单的 LED 灯的电路, 并成功地通过输入命令来控制它的状

第6章 扩展接口及应用

态。不过光是通过输入命令来控制 LED 是没有什么乐趣的,因此还要对这个电路进行扩展,为它再增加两组 LED 灯,最终的完成图如图 6-4 所示。

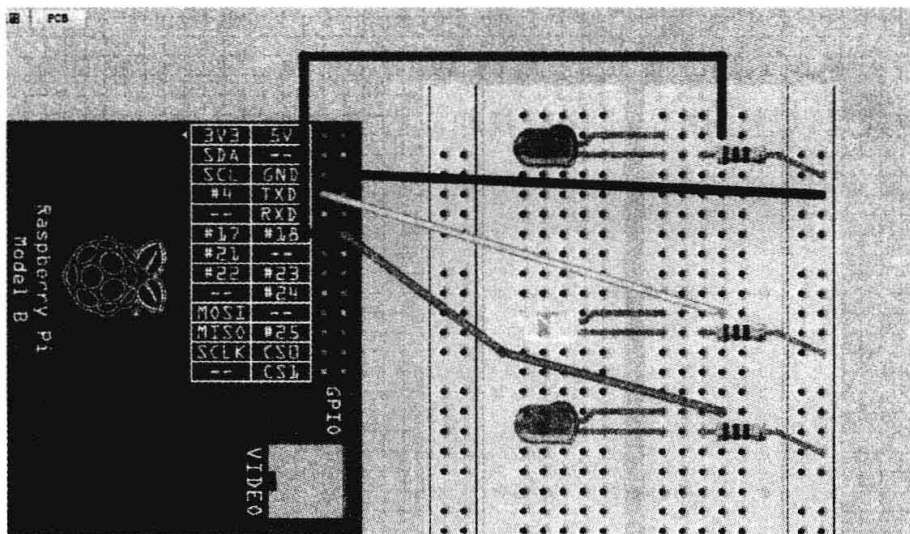


图 6-4 三组 LED 灯电路连接示意图

当用户按照上面的连接完成以后,我们可以使用 gpio 命令来对 LED 灯进行测试,查看是否连接成功。输入下面的命令分别测试不同颜色的 LED 的状态。

点亮和熄灭 LED2:

```
# gpio write 1 1
# gpio write 1 0
```

点亮和熄灭 LED3:

```
# gpio write 7 1
# gpio write 7 0
```

现在的面包板上有三组 LED 灯,我们通过输入命令对这三组 LED 灯进行分别控制,红色的 LED 灯表示内存不足,警告;绿色 LED 灯表示内存充足;黄色 LED 灯表示一般状态。

下面要编写一个获得内存状态的程序,是内存充足还是内存不足,又或是内存正常? 解决的办法是使用 free 程序,它可以读取出当前内存的使用情况,包括总内存数量、已使用内存、剩余内存等信息。

```
# free
```

	total	used	free	shared	buffers	cached
Mem:	448776	175332	273444	0	35300	117720
- / + buffers/cache:	22312	426464				
SWAP:	0	0	0			

现在已经完成了对 LED 的控制,并且已经有办法获得内存的信息,最后只需要编写一个程序来判断 free 命令所获得的内存情况并控制相应的 LED 灯。


```

#!/bin/bash
# Check if current user is a root
sudoret='sudo -n whoami'
sudoexitval=$?
if [ $sudoexitval != 0 ] || [ $sudoret != 'root' ]; then
    echo "Error: You're required to be sudoer or root to run this script. " >&2
    exit 1;
fi

scriptdir='dirname $BASH_SOURCE'
cd $scriptdir

gpio mode 0 out    # red led
gpio mode 1 out    # green led
gpio mode 7 out    # yellow led
gpio write 0 0
gpio write 1 0
gpio write 7 0

while true; do
    declare -i memall='free -m | grep Mem | awk '{print($2)}''
    declare -i memused='free -m | grep Mem | awk '{print($3)}''
    declare -i memusage='expr 100 \* $memused / $memall '
    if [ "$memusage" -ge "80" ]; then
        gpio write 0 1    # red led (low free memory)
        gpio write 1 0
        gpio write 7 0
    elif [ "$memusage" -lt "80" -o "$memusage" -ge "50" ]; then
        gpio write 0 0
        gpio write 1 0
        gpio write 7 1    # yellow led (normal free memory)
    else
        gpio write 0 0
        gpio write 1 1    # green led (enough free memory)
        gpio write 7 0
    fi
    sleep 1
done

```

在上面的程序中,首先对用户环境进行判断,如运行用户是否是 root 等。从第 11 行开始,设置 GPIO 接口为输出模式,并读取了所使用的三个 GPIO 接口的信息。接下来是一个简单的循环判断,先通过 free 命令读取出内存的信息,接着开始条件判断,如果内存使用量超过 80% 则显示红色 LED 灯,如果是 50%~80% 则显示黄灯,其他情况则显示绿灯。最后,sleep 1 语句会控制程序每秒钟检查一次。如果用

第6章 扩展接口及应用

户不想检查得这么频繁,则可以将它设置为 5 或 10。

读者可以用编辑器将上面的代码输入到 memch.sh 文件中,并通过下面的命令为它赋予可执行权限,并执行它:

```
# chmod 700 memch.sh
# ./memch.sh
```

现在,如果一切顺利,用户可以看到面包板上的 LED 灯开始亮起。如果显示红色表示内存不足,如果是绿色表示内存充足,如果是黄色表示内存使用到 50%~80%。不过由于 Linux 会充分利用内存的特性,因此大部分的用户可能很难看到绿灯亮起,不过没有关系,通过下面的命令我们可以手动释放内存:

```
# echo 3 > /proc/sys/vm/drop_caches
```

释放内存完毕后用户就会看到绿亮亮起,表示当前内存充足。至此,用户完成了一个内存警报系统,它可以实时报告当前系统的内存使用情况。如果用户离树莓派系统比较远,也可以通过在 WEB 页面中调用 memch.sh 脚本,这样就可以通过网页来查看内存的使用情况了。

6.2 通过 I²C 接口连接温度传感器(LM75)

本节通过为树莓派连接使用 LM75 温度传感器来获得当前温度。选用 LM75 除了它是一款常用的温度传感器外,还因为 Linux 的内核直接包含了它的驱动程序。这就省去了再额外安装驱动程序的过程。

在这个例子中,我们还将为树莓派启用 I²C 的支持模块,并为它安装 I²C 工具来访问所安装的 I²C 设备。

6.2.1 连接 LM75 到树莓派上

LM75 是一款内置带隙温度传感器和 $\Sigma-\Delta$ 模/数转换功能的 I²C 温度传感器。它也是温度检测器,可以提供检测的温度精度为 0.5℃,环境温度范围是 -55~+125℃,并且支持过热告警输出功能。由于它使用两线的串行 I²C 总线接口与控制器通信,因此在连接上十分简单,加上 LM75 芯片的电源范围是 2.8~5.5 V,因此无须电平转换就可以直接和树莓派的 I²C 接口连接,按照图 6-5 所示将 LM75 模块和树莓派连接起来。

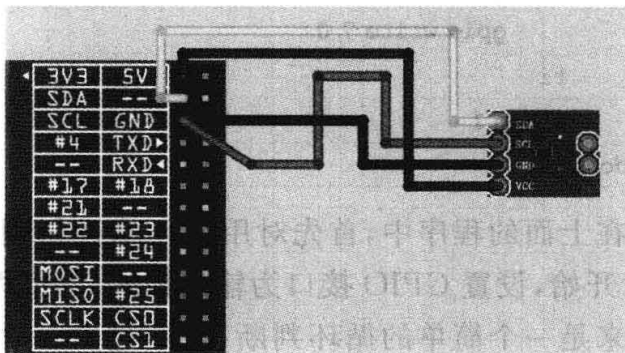


图 6-5 电路连接示意图

整个过程很简单,直接使用跳线将 LM75 与树莓派连接起来就行了。

6.2.2 为树莓派启用 I²C 功能

树莓派官方提供的 Raspbian 系统中并没有能使用的 I²C 总线接口,因此需要手动添加 I²C 模块。添加的过程很简单,使用你所熟悉的编辑器在/etc/目录下的 modules 文件的最后面添加以下两行内容,保存并退出就可以了。

```
i2c-bcm2708
i2c-dev
```

完成配置后,需要重新启动树莓派并通过执行 ls 命令,查看/dev 目录下是否已经有 I²C 设备文件的存在;用户也可以使用 lsmod 命令查看是否加载了 i2c-bcm2708 与 i2c-dev 模块。

```
# ls /dev/i2c *
/dev/i2c-0  /dev/i2c-1

# lsmod |grep i2c
i2c_dev          5620  0
i2c_bcm2708      3759  0
```

上面的两个命令可以看到/dev 目录下已经有了 i2c-0 与 i2c-1 这两个设备,同时 lsmod 命令也可以看到 i2c-bcm2708 与 i2c-dev 模块都已经加载了。

说明:老版 256M 树莓派使用的 I²C 设备名为 i2c-0,而新版 512M 树莓派则使用 i2c-1 作为设备名。

6.2.3 安装 i2cdetect 工具并检测温度传感器

完成了之前的工作后,当启动树莓派时,温度传感器就已经可以开始工作了,但是还需要一个工具来获得温度传感器采集到的信息,而这个工具就是 i2cdetect。

使用下面的命令来安装 i2cdetect:

```
# apt-get install i2cdetect
```

安装完成后就可以通过 i2cdetect 下面的命令来检测芯片是否正确连接(如果芯片和驱动程序都已经正常工作,那么就会顺利得到检测结果):

```
# i2cdetect -y -a 1
 0 1 2 3 4 5 6 7 8 9 a b c d e f
00: -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- --
```

第6章 扩展接口及应用

```
50: -----
60: -----
70: -----
```

如果用户使用的是第一版的 256M 树莓派, 请将上面命令最后面的 1 改为 0; 否则将无法检测出芯片。在上面的例子中, “48” 就是检测找到的温度传感器 LM75 的地址: 0x48。

LM75 温度传感器常用的寄存器如表 6-2 所列。

表 6-2 LM75 的常用寄存器列表

寄存器名称	寄存器地址	操作方式	描 述
温度寄存器(Temp)	00h	只读	温度值, 如 0x15, 代表 21℃
配置寄存器(Config)	01h	可读可写	配置芯片的各种功能, 智能或者关闭等
温度阈值寄存器(Tos)	03h	可读可写	配置温度告警 Tos 输出的阈值

注: 芯片的其他详情请参考芯片厂家的芯片手册。

根据寄存器列表, 我们知道温度寄存器地址为 0x00, 可以通过 i2cget 命令来读取温度:

```
# i2cget -f -y 1 0x48 0x00 b
0x14
```

i2cget 程序从温度寄存器中读取了当前温度, 不过这个温度是以十六进制的形式表现出来的, 因此需要将它转换成十进制。

在 Linux 系统下, 十六进制转十进制可使用 echo 命令, 读者可以将 i2cget 命令读取出来的数据放到下面命令的括号中进行转换:

```
# echo $((0x14))
20
```

“0x14” 为十六进制的温度值, 转换为十进制就是 20℃。

6.2.4 使用 python 来访问温度传感器

除了使用 i2c-tools 工具来读取温度外, 还可以使用 python 程序来读取温度, LM75 的代码分成三个 python 程序, Raspi_I2C.py、Raspi_LM75.py 和 lm75_example.py。其中, Raspi_I2C.py 是通用的 I²C 驱动程序, Raspi_LM75.py 是 LM75 的驱动程序, 而 lm75_example.py 是 LM75 的测试程序, 三个程序需要放在同一目录下。

Raspi_I2C.py 源代码:

```
#!/usr/bin/python
import smbus
```

```

=====
# Raspi_I2C Base Class
=====
class Raspi_I2C :

    def __init__(self, address, bus = smbus.SMBus(1), debug = False):
        self.address = address
        self.bus = bus
        self.debug = debug

    def reverseByteOrder(self, data):
        "Reverses the byte order of an int (16 - bit) or long (32 - bit) value"
        # Courtesy Vishal Sapre
        dstr = hex(data)[2:].replace("L","")
        byteCount = len(dstr[:2])
        val = 0
        for i, n in enumerate(range(byteCount)):
            d = data & 0xFF
            val |= (d << (8 * (byteCount - i - 1)))
            data >>= 8
        return val

    def write8(self, reg, value):
        "Writes an 8 - bit value to the specified register/address"
        try:
            self.bus.write_byte_data(self.address, reg, value)
            if (self.debug):
                print "I2C: Wrote 0x% 02X to register 0x% 02X" % (value, reg)
        except IOError, err:
            print "Error accessing 0x% 02X; Check your I2C address" % self.address
            return -1

    def writeList(self, reg, list):
        "Writes an array of bytes using I2C format"
        try:
            if (self.debug):
                print "I2C: Writing list to register 0x% 02X:" % reg
                print list
            self.bus.write_i2c_block_data(self.address, reg, list)
        except IOError, err:
            print "Error accessing 0x% 02X; Check your I2C address" % self.address
            return -1

    def readList(self, reg, length):
        "Read a list of bytes from the I2C device"

```



```

results = []
try:
    results = self.bus.read_i2c_block_data(self.address, reg, length)
    if (self.debug):
        print "I2C: Device 0x%02X returned the following from reg 0x%02X" % (self.
address, reg)
        print results
    return results
except IOError, err:
    print "Error accessing 0x%02X: Check your I2C address" % self.address
    return -1

def readU8(self, reg):
    "Read an unsigned byte from the I2C device"
    try:
        result = self.bus.read_byte_data(self.address, reg)
        if (self.debug):
            print "I2C: Device 0x%02X returned 0x%02X from reg 0x%02X" % (self.ad-
dress, result & 0xFF, reg)
        return result
    except IOError, err:
        print "Error accessing 0x%02X: Check your I2C address" % self.address
        return -1

def readS8(self, reg):
    "Reads a signed byte from the I2C device"
    try:
        result = self.bus.read_byte_data(self.address, reg)
        if (self.debug):
            print "I2C: Device 0x%02X returned 0x%02X from reg 0x%02X" % (self.ad-
dress, result & 0xFF, reg)
        if (result > 127):
            return result - 256
        else:
            return result
    except IOError, err:
        print "Error accessing 0x%02X: Check your I2C address" % self.address
        return -1

def readU16(self, reg):
    "Reads an unsigned 16-bit value from the I2C device"
    try:
        hiByte = self.bus.read_byte_data(self.address, reg)
        result = (hiByte << 8) + self.bus.read_byte_data(self.address, reg + 1)

```

```

        if (self.debug):
            print "I2C: Device 0x% 02X returned 0x% 04X from reg 0x% 02X" % (self.ad-
dress, result & 0xFFFF, reg)
            return result
        except IOError, err:
            print "Error accessing 0x% 02X: Check your I2C address" % self.address
            return -1

def readS16(self, reg):
    "Reads a signed 16 - bit value from the I2C device"
    try:
        hibyte = self.bus.read_byte_data(self.address, reg)
        if (hibyte > 127):
            hibyte -= 256
        result = (hibyte << 8) + self.bus.read_byte_data(self.address, reg + 1)
        if (self.debug):
            print "I2C: Device 0x% 02X returned 0x% 04X from reg 0x% 02X" % (self.ad-
dress, result & 0xFFFF, reg)
        return result
    except IOError, err:
        print "Error accessing 0x% 02X: Check your I2C address" % self.address
        return -1

```

说明:新版树莓派为 `bus=smbus.SMBus(1)`,老版树莓派请改为 `bus=smbus.SMBus(0)`。

Raspi_LM75.py 源代码:

```

#!/usr/bin/python

import time
from Raspi_I2C import Raspi_I2C

=====
# LM75 Class
=====

class LM75:
    i2c = None

    # IC Identifiers
    __IC_LM75 = 0x00

    # Register
    __LM75_REG_TEMP = 0x00
    __LM75_REG_CONFIG = 0x01

```

```

# Constructor
def __init__(self, address = 0x49, ic = __IC_LM75, debug = False):
    self.i2c = Raspi_I2C(address)
    self.address = address
    self.debug = debug

    # Make sure the IC specified is valid
    if (ic < self.__IC_LM75):
        if (self.debug):
            print "Invalid IC. Using the LM75 by default"
        self.ic = __IC_LM75
    else:
        self.ic = ic

def readTemperature(self):
    # Read the Temperature results
    result = self.i2c.readU8(self.__LM75_REG_TEMP)
    if (self.ic == self.__IC_LM75):
        # Return value for the LM75
        return ( result )

```

lm75_example.py 源代码:

```

#!/usr/bin/python

from Raspi_LM75 import LM75

=====
# Example Code
=====

# Initialise the Temperature using the default mode (IC = LM75, default address)
tmp = LM75()

# Read Temperature
result = tmp.readTemperature( )
print "Temperature = %d C" % (result)

```

将三个文件放在同一目录下后,执行 lm75_example.py 程序:

```

# python lm75_example.py
Temperature = 23 C

```

程序在最后面显示出了当前检测到的温度值是 23℃。

6.3 树莓派上使用串口 GPS

GPS(全球定位系统)可以为用户提供当前的位置信息和准确的时间,所以 GPS

模块经常用于各种导航设备和时钟校准设备中,而 GPS 模块通常使用串口来进行连接,因此用户可以方便地通过树莓派所提供的串口来连接这些设备。在本例中介绍一些树莓派上使用串口 GPS 模块的方法,方便玩家使用树莓派定位和读取当前时间。

6.3.1 为树莓派启用串口

树莓派的扩展接口中也提供一种简化的 UART 接口,有三个接口:GND、TX、RX,如图 6-6 所示。需要说明的是,树莓派提供的 UART 接口是 TTL 电平的串口,并非普通 PC 通常使用的 RS-232 电平的串口。由于 TTL 电平是 5 V 的,而 RS-232 是负逻辑电平,定义 +5~+12 V 为低电平, -12~-5 V 为高电平;因此,在使用树莓派的串口时,需要注意互连设备的串口电平方式,不能盲目地把树莓派 TTL 电平的串口直接与 PC 机 RS-232 电平的串口直接连接。这样会导致芯片的接口损坏。正确的方法是使用一个可以进行电平转换的芯片(常用的芯片有 MAX232 和 PL2303)或转换线进行转换。

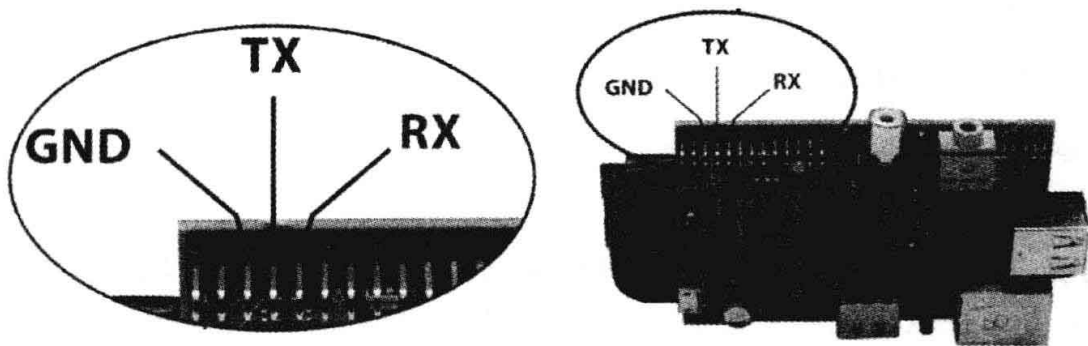


图 6-6 树莓派 UART 接口

可以在终端中输入 ls 命令查看相应的串口设备:

```
pi@raspberrypi:~$ ls /dev/ttyA*
/dev/ttyAMA0
```

其中“/dev/ttyAMA0”就是树莓派提供给大家的串口。在树莓派官方提供的系统镜像中,串口默认是作为系统的 console 控制台使用的,用户可以使用一根 USB 转 TTL 的数据线将 PC 和树莓派的串口连接,也可以将 PC 的 RS-232 口通过 RS-232 转 TTL 的转换器与树莓派的串口连接。连接后,PC 就能通过 console 控制台登录树莓派,并执行各种操作,其运行效果与 SSH 连接后的效果是一样的。

树莓派的串口除了用做系统 console 控制台外,还可以用来和其他串口设备进行通信,不过在使用前需要在系统中将默认的 console 控制功能去除,按下面的步骤执行去除操作。

① 编辑/boot 目录下的 cmdline.txt 文件,修改文件内容:

```
# nano /boot/cmdline.txt
```

将下面文件的内容：

```
dwc_otg.lpm_enable = 0 console = ttyAMA0,115200 kgdboc = ttyAMA0,115200
console = tty1 root = /dev/mmcblk0p2 rootfstype = ext4 elevator = deadline
rootwait
```

改为

```
dwc_otg.lpm_enable = 0 console = tty1 root = /dev/mmcblk0p2 rootfstype = ext4
elevator = deadline rootwait
```

② 编辑 `inittab` 初始化文件,修改串口相关的配置:

```
# nano /etc/inittab
```

找到以下内容:

```
# Spawn a getty on Raspberry Pi serial line
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

将下面两行注释掉:

```
# Spawn a getty on Raspberry Pi serial line
# T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

说明: `T0` 为该行 ID,一般表示该行的用处; `23` 指该行的运行级别是 2 级、3 级; `respawn` 表示进程在无论什么时候终止,都会自动重启; `/sbin/getty -L ttyAMA0 115200 vt100` 表示具体的命令,该命令通过 `/sbin/getty` 程序打开串口 `/dev/ttyAMA0` (树莓派串口),波特率设置为 115 200 bps,终端模式为 `vt100`。

6.3.2 连接 GPS 模块并启用控制台功能

准备一个串口为 3.3 V 电平的 GPS 模块,注意 GPS 模块的串口电平必须是 3.3 V 的,和树莓派的串口电平一致,然后按照图 6-7 和图 6-8 的连接方式将 GPS 模块串口的收(GPS_RX)和树莓派串口的发(Raspi_TX)相连,GPS 模块串口的发(GPS_TXD)和树莓派串口的收(Raspi_RXD)相连。

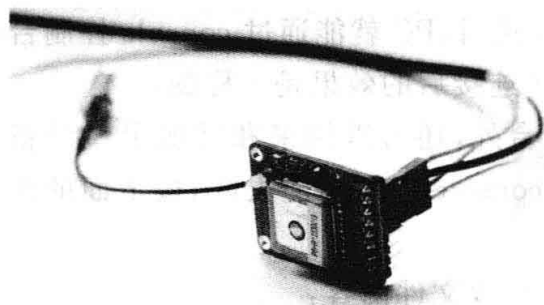


图 6-7 GPS 模块实物图

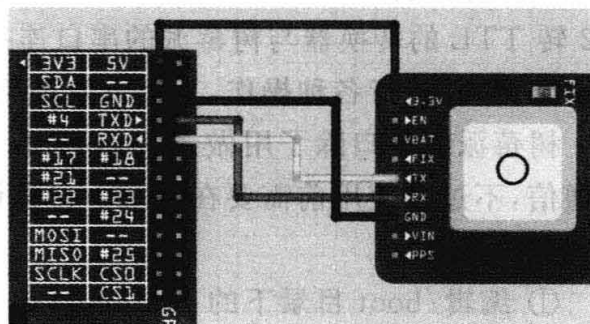


图 6-8 GPS 模块连接示意图

6.3.3 连接 GPS 并获取位置信息与时间

电路连接好以后,确保串口也已经启动,这时就可以在树莓派上安装 GPS 相关的软件来对 GPS 进行测试了:

```
# apt - get install gpssd gpssd - clients python - gps
```

使用以下命令来配置 GPS 的端口使用串口设备:

```
# gpssd /dev/ttyAMA0 - F /var/run/gpssd.sock
```

现在可以将 GPS 模块尽量放置户外或者窗外,方便 GPS 模块能接收 GPS 信号定位,然后执行 cgps 命令,查看 GPS 模块的定位情况:

```
# cgps - s
```

当模块成功后,会如图 6-9 所示显示时间、位置等信息。

Time:	2013-01-24T08:56:30.000Z	PRN:	Elev:	Azim:	SNR:	Used:
Latitude:		11	80	287	37	Y
Longitude:		1	59	288	26	Y
Altitude:	215.6 ft	32	53	207	29	Y
Speed:	0.0 mph	19	52	153	24	Y
Heading:	127.3 deg (true)	14	34	076	45	Y
Climb:	0.0 ft/min	39	29	150	30	Y
Status:	3D FIX (7 secs)					

图 6-9 GPS 模块定位信息显示

如果发现长时间没有显示定位信息,请检查一下 GPS 模块上方是否有遮挡物,另外天气对 GPS 定位也有一定的影响,因此最好是选择晴朗天气。如果以上条件都满足但还是没有定位成功,用户可以尝试将原有 gpssd 命令杀死,然后再次配置 gpssd 程序,重启用 cgps 来定位:

```
# killall gpssd
```

```
# gpssd /dev/ttyAMA0 - F /var/run/gpssd.sock
```

```
# cgps - s
```

GPS 模块除了定位外,还可以为树莓派提供准确的时钟。下面介绍如何通过 ntp(网络对时软件)将 GPS 模块的精确时钟作为树莓派的系统时钟。如果用户的系统中没有安装 ntp 软件,则可以使用以下命令来进行安装:

```
# apt - get install ntp
```

然后编辑 ntp.conf 文件:

```
# nano /etc/ntp.conf
```


在文件最后添加 GPS 时钟的内容：

```
# gps ntp
server 127.127.28.0 minpoll 4
fudge 127.127.28.0 time1 0.183 refid NMEA
server 127.127.28.1 minpoll 4 prefer
fudge 127.127.28.1 refid PPS
```

最后重新启动 ntp 服务：

```
# service ntp restart
```

等待片刻(如图 6-10 所示),查看时间系统时间是否同步上 GPS 时间：

```
# ntpq -p
```

```
pi@raspberrypi:~$ ntpq -p
remote          refid          st t when poll reach  delay  offset  jitter
=====
*ns1.lums.net.uk 33.117.170.50  2 u  54  64   7  65.454   2.666   5.800
+resntp-b-vip.lo 127.151.91.34   3 u  45  64  17  55.704  -5.169   8.482
+bart.nexellent. 194.242.34.149  2 u  17  64  17  76.585  -4.271  57.595
+v01.s01.be.it2g 193.190.230.65  2 u  20  64  37  86.464  -2.374 228.460
xSHM(0)          .NMEA.         0 l  11  16 377   0.000 144.714   3.026
SHM(1)          .PPS.         0 l   -  16  0   0.000   0.000   0.000
```

图 6-10 同步网络时间

如果长时间同步不上,用户可以尝试将原有 gpsd 命令杀死,然后再次配置 gpsd 程序,并启用 cgps 来获取时间:

```
# killall gpsd
# gpsd /dev/ttyAMA0 -F /var/run/gpsd.sock
# service ntp restart
```

6.3.4 使用 python 读取 GPS 信息

除了使用 cgps 软件查看 GPS 信息外,还可以用 python 的 GPS 库进行编程。下面是一个简单的 python 程序,读取串口 GPS 模块的时间信息,并打印出来。

gpstest.py 源代码:

```
import gps

# Listen on port 2947 (gpsd) of localhost
session = gps.gps("localhost", "2947")
session.stream(gps.WATCH_ENABLE | gps.WATCH_NEWSTYLE)

while True:
    try:
```

```

report = session.next()
# Wait for a 'TPV' report and display the current time
# To see all report data, uncomment the line below
# print report
if report['class'] == 'TPV':
    if hasattr(report, 'time'):
        print report.time
except KeyError:
    pass
except KeyboardInterrupt:
    quit()
except StopIteration:
    session = None
    print "GPSD has terminated"

```

如图 6-11 所示, 执行 gpstest.py 程序:

```
pi@raspberrypi:~$ sudo python gpstest.py
```

如果 GPS 定位成功, 就会打印出当前的 GPS 时间。



```

~Cwebide@raspberrypi:~$ sudo python gpstest.py
2013-01-24T10:39:27.000Z
2013-01-24T10:39:28.000Z
2013-01-24T10:39:29.000Z
2013-01-24T10:39:30.000Z
2013-01-24T10:39:31.000Z

```

图 6-11 输出 GPS 时间

更多 python-gps 库提供的函数使用和例程请查看官网 http://gpsd.berlios.de/client-howto.html#_python_examples。

附录 A

树莓派显示模式表

表 A-1 树莓派显示模式表

模 式	分辨率	频率/Hz	模 式	分辨率	频率/Hz
hdmi_mode=1	640×350	85	hdmi_mode=44	1400×1050	85
hdmi_mode=2	640×400	85	hdmi_mode=45	1400×1050	120
hdmi_mode=3	720×400	85	hdmi_mode=46	1440×900	—
hdmi_mode=4	640×480	60	hdmi_mode=47	1440×900	60
hdmi_mode=5	640×480	72	hdmi_mode=48	1440×900	75
hdmi_mode=6	640×480	75	hdmi_mode=49	1440×900	85
hdmi_mode=7	640×480	85	hdmi_mode=50	1440×900	120
hdmi_mode=8	800×600	56	hdmi_mode=51	1600×1200	60
hdmi_mode=9	800×600	60	hdmi_mode=52	1600×1200	65
hdmi_mode=10	800×600	72	hdmi_mode=53	1600×1200	70
hdmi_mode=11	800×600	75	hdmi_mode=54	1600×1200	75
hdmi_mode=12	800×600	85	hdmi_mode=55	1600×1200	85
hdmi_mode=13	800×600	120	hdmi_mode=56	1600×1200	120
hdmi_mode=14	848×480	60	hdmi_mode=57	1680×1050	—
hdmi_mode=15	1024×768	43	hdmi_mode=58	1680×1050	60
hdmi_mode=16	1024×768	60	hdmi_mode=59	1680×1050	75
hdmi_mode=17	1024×768	70	hdmi_mode=60	1680×1050	85
hdmi_mode=18	1024×768	75	hdmi_mode=61	1680×1050	120
hdmi_mode=19	1024×768	85	hdmi_mode=62	1792×1344	60
hdmi_mode=20	1024×768	120	hdmi_mode=63	1792×1344	75
hdmi_mode=21	1152×864	75	hdmi_mode=64	1792×1344	120
hdmi_mode=22	1280×768	—	hdmi_mode=65	1856×1392	60
hdmi_mode=23	1280×768	60	hdmi_mode=66	1856×1392	75
hdmi_mode=24	1280×768	75	hdmi_mode=67	1856×1392	120
hdmi_mode=25	1280×768	85	hdmi_mode=68	1920×1200	—

续表 A-1

模 式	分辨率	频率/Hz	模 式	分辨率	频率/Hz
hdmi_mode=26	1 280×768	120	hdmi_mode=69	1 920×1 200	60
hdmi_mode=27	1 280×800	—	hdmi_mode=70	1 920×1 200	75
hdmi_mode=28	1 280×800	60	hdmi_mode=71	1 920×1 200	85
hdmi_mode=29	1 280×800	75	hdmi_mode=72	1 920×1 200	120
hdmi_mode=30	1 280×800	85	hdmi_mode=73	1 920×1 440	60
hdmi_mode=31	1 280×800	120	hdmi_mode=74	1 920×1 440	75
hdmi_mode=32	1 280×960	60	hdmi_mode=75	1 920×1 440	120
hdmi_mode=33	1 280×960	85	hdmi_mode=76	2 560×1 600	—
hdmi_mode=34	1 280×960	120	hdmi_mode=77	2 560×1 600	60
hdmi_mode=35	1 280×1 024	60	hdmi_mode=78	2 560×1 600	75
hdmi_mode=36	1 280×1 024	75	hdmi_mode=79	2 560×1 600	85
hdmi_mode=37	1 280×1 024	85	hdmi_mode=80	2 560×1 600	120
hdmi_mode=38	1 280×1 024	120	hdmi_mode=81	1 366×768	60
hdmi_mode=39	1 360×768	60	hdmi_mode=82	1 920×1 080	60
hdmi_mode=40	1 360×768	120	hdmi_mode=83	1 600×900	—
hdmi_mode=41	1 400×1 050	—	hdmi_mode=84	2 048×1 152	—
hdmi_mode=42	1 400×1 050	60	hdmi_mode=85	1 280×720	60
hdmi_mode=43	1 400×1 050	75	hdmi_mode=86	1 366×768	—

附录 B

常见的树莓派扩展板

树莓派的扩展接口 P1 提供了多种扩展接口,为了能让用户方便地使用这些扩展接口,因此就出现了各种各样基于树莓派的扩展板。在这些树莓派的扩展板中通常会集成一种或几种功能,用户可以直接使用这些扩展板来完成某些功能。目前,市面上树莓派的扩展板比较多,功能也各种各样,这里介绍几款国内外比较受欢迎的扩展板。

B.1 GPIO 串口扩展板

GPIO 串口扩展板是基于树莓派的串口和 GPIO 接口的扩展功能板卡(见图 B-1),主要是将树莓派上的 GPIO 和串口扩展出来。其中,每个树莓派的 GPIO 接口都参考 arduino 的设计方式,采用 3 引脚式(GND、VCC 和 GPIO)接口方便用户可以接各种外设,并将树莓派 TTL 电平的 UART 串口转换为 PC 机通用的 RS-232 电平的串口,方便用户通过 PC 机直接调试并控制树莓派,并且扩展板还将 GPIO 串口插座的形式扩展出来。

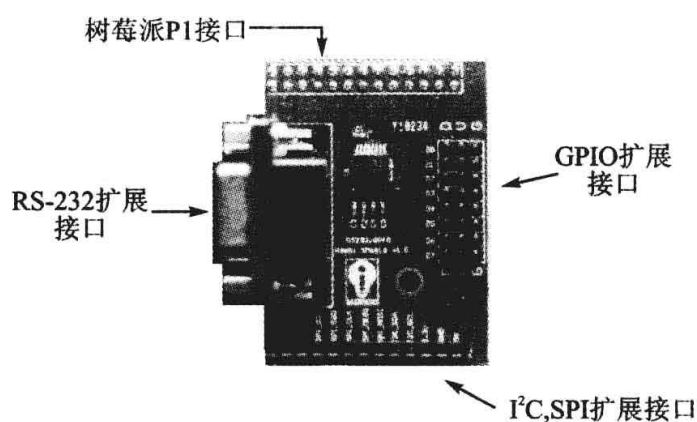


图 B-1 串口扩展板接口定义图

B.2 全功能扩展板

全功能扩展板是基于树莓派的 GPIO 接口和 I²C/SPI 接口的扩展功能板卡,见图 B-2,可以将树莓派上的 GPIO 的输出扩展出来,支持高电压大电流的输出驱动,

能将树莓派的 3.3 V 电平的 I²C 和 SPI 接口转换为 5 V 电平,并通过 ULN2803 芯片直接驱动电机、步进电机舵机、LED 等外设;还可以配合“GPIO 串口扩展板”一起使用,进一步增强扩展功能。

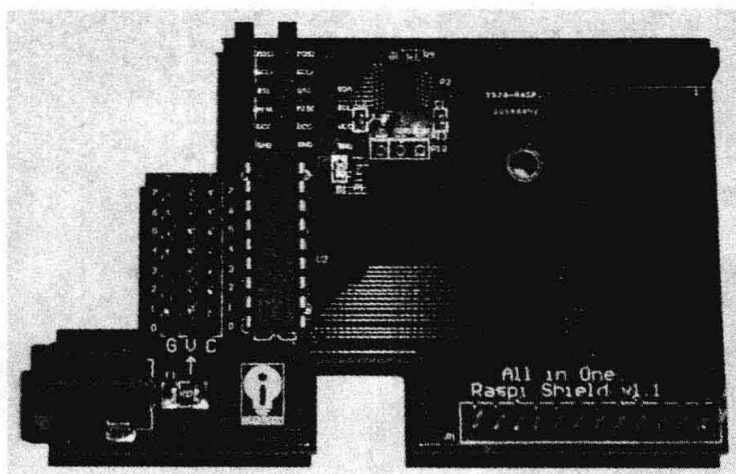


图 B-2 全功能扩展板接口定义图

B.3 时钟扩展板

时钟扩展板是基于树莓派的 I²C 接口的扩展功能板卡,见图 B-3,主要是通过扩展板上的 RTC(I²C 接口)来为树莓派提供不会掉电丢失的精确时钟,并通过树莓派上的 I²C 接口来控制 4 位数码管显示时钟。扩展板还集成了一个蜂鸣器,可以通过编程使其成为闹铃或者告警等用途。

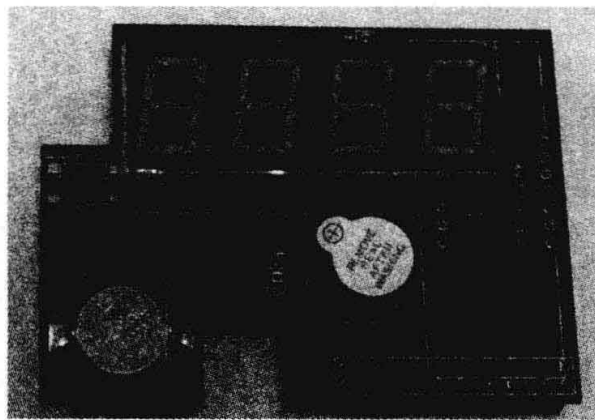


图 B-3 时钟扩展板实物图

B.4 电源扩展板

电源扩展板是基于树莓派的多功能扩展板卡,见图 B-4,主要是通过扩展接口

附录 B 常见的树莓派扩展板

为树莓派提供稳定的电源,支持宽电压输入并且可以通过开关来开启关闭电源,而内置的数码管还可以显示电压与电流,具有 5 V 电平的 I²C 接口,以及 RS-232 串口。

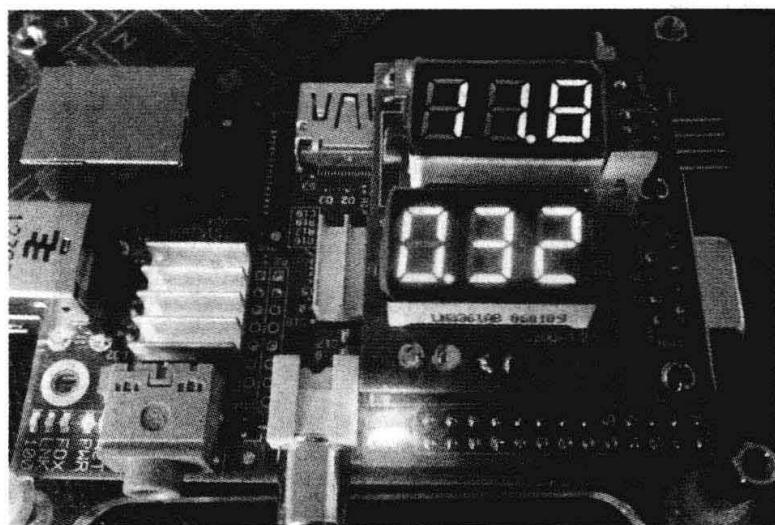


图 B-4 电源扩展板实物图

B.5 LCD 按键扩展板

LCD 按键扩展板是一个集成了 LCD1602 字符液晶屏和 5 个按键的多功能扩展板,见图 B-5,使用树莓派的 I²C 接口,通过扩展板上的 MCP23017 芯片(I²C 接口的 16 位 I/O 扩展芯片)来控制扩展板的显示和按键操作。另外,扩展板可以通过程序控制切换 LCD 屏的背光颜色,支持显示 16×2 字符,可调整灰度。详细介绍请前往网址 www.adafruit.com/products/1109。

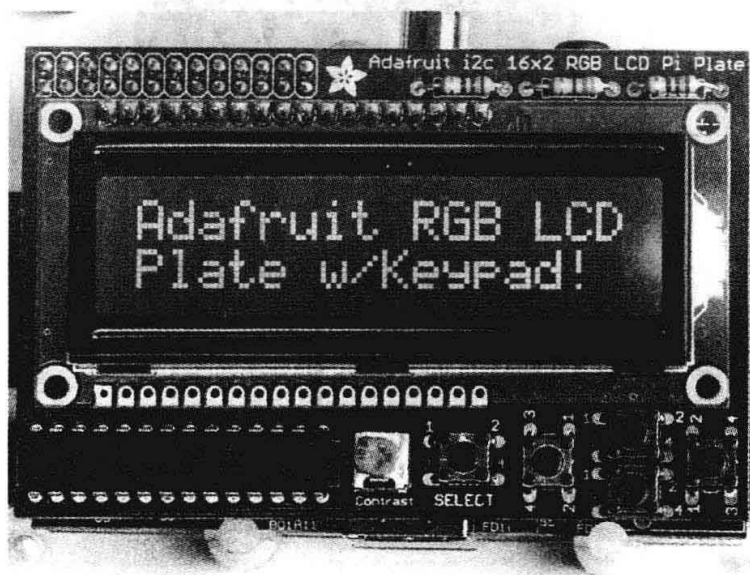


图 B-5 LCD 按键扩展板实物图

B.6 树莓派气象站扩展板

树莓派气象站扩展板是一个集成了温度、湿度、气压等多种传感器和实时时钟的多功能扩展板,见图 B-6。此扩展板可以完成一个小型的气象监测站的功能,其中温度传感器测量范围为 $0\sim 50\text{ }^{\circ}\text{C}$,气压传感器测量范围为 $50\sim 150\text{ kPa}$,湿度传感器测量范围为 $20\%\sim 90\%$;另外,扩展板还提供了 GPIO 控制接口,可用来控制继电器一类的外设。详细介绍请前往网址 <http://www.web4robot.com/PiWeather.html>。

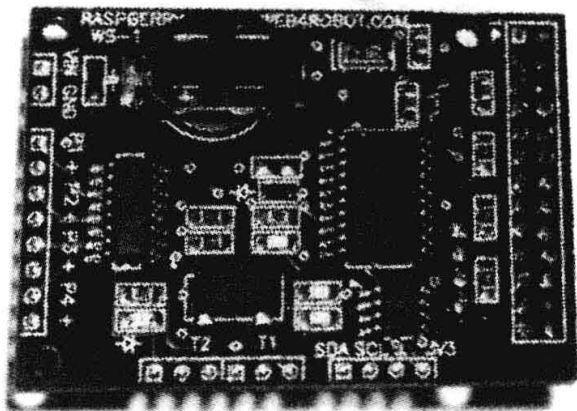


图 B-6 气象站扩展板实物图

B.7 树莓派 ADC 扩展板

树莓派 ADC 扩展板是一个集成了 8 通道 17 位的模/数转换器的多功能扩展板,见图 B-7。此扩展板可以完成 A/D 转换、树莓派的 I²C 接口配置和读取转换器的各通道值。

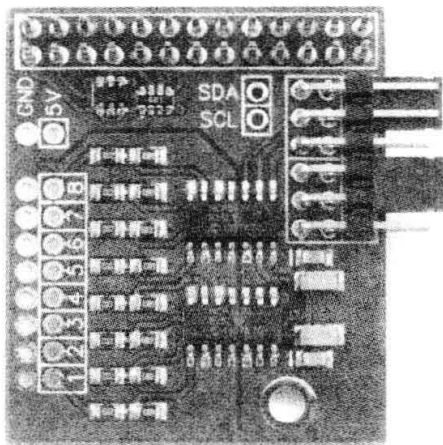


图 B-7 ADC 扩展板实物图

详细介绍请前往产品网址 <http://www.abelectronics.co.uk/products/3/Raspberry-Pi/17/ADC-Pi-V2---Raspberry-Pi-Analogue-to-Digital-converter>。

B.8 步进电机扩展板

步进电机扩展板是集成了 2 个步进电机的驱动接口的扩展板,见图 B-8。此扩展板可以同时驱动 2 路五线四相式的步进电机,并有 LED 显示。

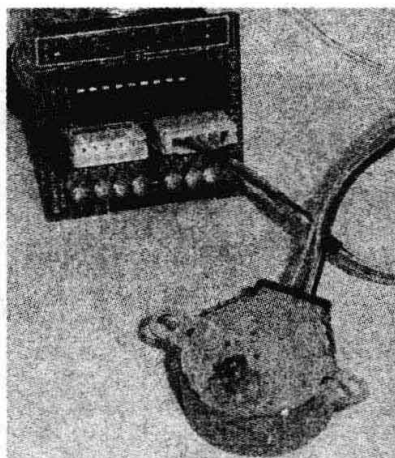


图 B-8 步进电机扩展板实物图

详细介绍请前往产品网址 <http://mypishop.com/Stepper.html>。

更多扩展板的信息可以登录网址 http://elinux.org/RPi_Expansion_Boards 了解详情。

附录 C

手动安装 Debian 系统

由于树莓派并没有 BIOS 系统,因此不可能像在 PC 上安装系统一样为树莓派安装操作系统,而树莓派官方网站上虽然提供了 Debian 系统下载,但它只是一个已经安装好的镜像文件。在这个文件中包含了一些用户可能不需要的程序或是文档,例如当用户需要用树莓派来作为服务器时,可能就不需要桌面系统的程序了。而且由于镜像文件的大小是 2 GB 的,因此它也无法安装在小于 2 GB 的 SD 卡上。因此读者如果想要一个纯净的 Debian 系统,就只能选择手动来安装了。

安装 Debian,首先需要准备两个 SD 卡。一个要大于 2 GB,这个 SD 卡会作为临时的系统来使用。当你手动安装完成 Debian 后,就可以将这个 SD 卡作为其他用途了。另一个 SD 卡则要求至少 512 MB 以上的空间。如果你所安装的程序不多,推荐使用一个 1 GB 的 SD 卡和一个读卡器就可以了。如果你已经准备好了这些,那么就可以开始生成一个纯净的 Debian 系统了。

首先,为树莓派安装系统,你需要将树莓派官方提供的 Debian 镜像写入到一个大于 2 GB 的 SD 卡中,然后作为系统盘启动,具体方法可以参考 3.2 节。

将 1 GB 的 SD 卡放入读卡器中并插入到树莓派的 USB 接口上,用户需要将 1 GB 的 SD 卡分为一个 50 MB 的 Vfat 分区,而剩下的空间则分为 ext4 分区。关于分区方法,可以参见前面章节所介绍的 fdisk 使用方法。

分区完成后,使用格式化工具格式化新分区,并将新建的 ext4 挂载到 /mnt 目录下,然后在 /mnt 目录下建立一个 boot 目录,用来挂载刚才建立的 vfat 分区。

```
# mkfs.msdos /dev/sdb1
# mkfs.ext4 /dev/sdb2
# mount /dev/sdb2 /mnt
# mkdir /mnt/boot
# mount /dev/sdb1 /mnt/boot
```

安装 debootstrap 程序

debootstrap 程序是 Debian 系统提供的一个安装程序,它可以在指定的目录内安装一个最小化的 Debian 系统(等同于使用 Debian 光盘安装系统):

```
# apt-get install debootstrap
```

附录 C 手动安装 Debian 系统

安装完成后就可以使用以下命令安装 Debian 系统：

```
# debootstrap --arch armhf wheezy /mnt http://mirrordirector.raspbian.org/raspbian/
```

在上面的命令中，`--arch` 参数是指定我们所需要的安装平台。由于树莓派是基于 ARM 平台的，所以我们选择 `armhf` 参数（这个参数表示我们所安装的系统是支持硬件浮点运算的）。`wheezy` 表示 Debian 的版本；`/mnt` 是指安装路径，最后的连接是告诉 `debootstrap` 程序从哪里下载系统程序。

在树莓派上运行 `debootstrap` 程序需要很长时间，至少 3 个小时，因此需要耐心等待。

当安装完成后就有了一个要供启动的原生 Debian 系统了，不过在使用它之前还需要做一些设置。例如，为它设置一个 root 密码，更新它的 `fstab` 文件等。

更新系统文件配置：

```
# cp /etc/fstab /mnt/etc/
```

更新软件包信息：

```
# cp /etc/apt/source /mnt/etc/apt
```

更新网卡、DNS 配置文件：

```
# cp /etc/network/interfaces /mnt/etc/network/interfaces
```

```
# cp /etc/resolv.conf /mnt/etc/resolv.conf
```

复制内核以及内核模块与固件到新系统中：

```
# cp -ar /boot/* /mnt/boot/
```

```
# cp -ar /lib/modules /mnt/lib/
```

```
# cp -ar /lib/firmware/ /mnt/lib/firmware/
```

通过 `chroot` 程序进入到原生系统中，这一步很重要。之前我们所做的工作都是在宿主系统中完成的，现在需要通过 `chroot` 命令进入到新的系统中，而在这个新的系统中，我们所做的一切操作等同于使用新系统启动后的操作：

```
# chroot /mnt
```

更新 apt 软件仓库信息：

```
# apt-get update
```

设置新系统的 root 用户密码：

```
# passwd root
```

安装常用软件（如 `openssh` `vim` 等）：

```
# apt-get install openssh-server vim ntp
```

当所需要的软件安装以及各种配置工作完成后,我们就可以使用 `exit` 命令退出 `chroot` 环境了:

```
# exit
```

最后要卸载掉 `mnt` 目录下的 `boot` 分区与根目录下的 `mnt` 分区:

```
# cd ..
```

```
# umount /mnt/boot
```

```
# umount /mnt
```

至此,我们就完成了所有的工作。用户可以关闭现在的系统,换上新的 SD 卡来启动。

Images have been losslessly embedded. Information about the original file can be found in PDF attachments. Some stats (more in the PDF attachments):

```
{
  "filename": "44CK546p6L2s5qCR6I6T5rS+PVJBU1BCRVJSWVBJ44CLXzEzNDYxMzQyLnppcA==",
  "filename_decoded": "\u300a\u73a9\u8f6c\u6811\u8393\u6d3e=RASPBERRYPI\u300b_13461342.zip",
  "filesize": 43008900,
  "md5": "71b8a8a5d939962e9bb4952f7e6bd019",
  "header_md5": "f39fcd777b7969743452eec5c913c381",
  "sha1": "ab0e1fba5b466d788f998c84359cf190b12ebc12",
  "sha256": "fea568b2bad47f040f45fc8daf6a46ddcce3183b573c9b9bb0427a2ef61cb0d2",
  "crc32": 1170939302,
  "zip_password": "",
  "uncompressed_size": 48588194,
  "pdg_dir_name":
"\u00ed\u2562\u2550\u00b5\u256b\u00ac\u2569\u2248\u258c\u00ab\u253c\u2554=RASPBERRYPI\u00ed\u2556_13461342",
  "pdg_main_pages_found": 171,
  "pdg_main_pages_max": 171,
  "total_pages": 183,
  "total_pixels": 991375840,
  "pdf_generation_missing_pages": false
}
```