

树莓派 Raspberry Pi 使用简介

你需要以下这些东西：

一张 Micro SD 记忆卡

一个 SD 读卡器，用于将系统映像写入到 Micro SD 卡中供电来源。我们用的是一个旧的安卓手机充电器，你需要一个 5V 的 micro USB 接口充电器为它供电。

如果你是用的普通显示器而不是高清电视，你需要一条 HDMI 连接线与 HDMI-DVI 转换器。如果你的显示器支持 HDMI 或者你打算使用电视机进行连接，那么你就可以省去转换器了。

USB 接口的键盘和鼠标

一条以太网线

可选设备：

用于保护 Raspberry Pi 的盒子

除了键盘和鼠标之外，如果你打算要连接更多的 USB 设备，那么你需要一个 USB Hub 集线器。

我们在本文中使用的东西：

一个 Model B Raspberry Pi

一张 16GB Class 10 SanDisk Ultra SDHC 记忆卡 (传输率标为 300MB/s)

一个原本用于 HTC Inspire 的旧充电器

罗技 K260 无线键鼠套装 (两件只用一个接收器 , 因此只占用一个 USB 口)

一条 RJ-45 网线

设置

当你把上面的一切都准备好以后 , 我们就可以开始设置 Raspberry Pi 了

将 Raspbian 映像安装到 Micro SD 卡中

Raspberry Pi 出来已经有一段时间了 , 也正因如此 , 无论你是在用 Windows、Mac OS X 还是 Linux , 都有好几种方法将 Raspbian 的映像写到你的 Micro SD 卡中。然而为了让这篇文章不变成介绍各种映像写入软件的文章 , 我们将会把最常用的方法展示给大家看 : 在 OS X 与 Linux 中我们将会用系统自带的 dd 工具写入 , 而在 Windows 中我们将会使用一个叫 Win32DiskImager 的工具进行。

为了让过程变得简单 , 我们建议你在开始操作前 , 先把下载文件夹清空。然后 , 从这里下载 Raspbian.zip。在里面 , 选择最新的 Raspbian 来下载。当然你也可以选择使用 BT 来下载 , 为了能让其他人继续用 BT 下载 , 我们建议你在下载完成后继续上传。

把压缩文件解压以后 , 你将得到 Raspbian 的映像文件 , 这个时候打开终端窗口。如果你使用的是 OS X 10.7 或更高版本 , 你可以从应用

程序中的工具(10.7)或者其他(10.8)中找到终端窗口。打开后，使用“`cd ~/Downloads`”来进入下载文件夹。

如果你在开始前清空了你的下载文件夹，那么现在里面应该有两个文件，输入“`ls`”来确认是否分别有一个后缀与一个zip后缀的文件，如下图：

```
xenogears:Downloads john$ ls
2012-07-15-wheezy-raspbian.img  2012-07-15-wheezy-raspbian.zip
```

接下来就要开始将解压出来的映像文件写到 Micro SD 卡中了，很可惜这个操作并不是直接把文件拷到卡中那么简单。首先，我们在把 Micro SD 卡插到你的 Mac 之前，要把你电脑分区状态先搞清楚。如果你已经迫不及待把卡插了进去，那么先把它安全移除吧。

在终端窗口中输入“`df -h`”，你应该会看到类似这样的结果

```
xenogears:Downloads john$ df -h
Filesystem      Size  Used Avail Capacity  iused   ifree %iused  Mounted on
/dev/disk0s2    232Gi 122Gi 110Gi    53% 32108683 28731059   53%  /
devfs           180Ki 180Ki   0Bi   100%     624     0 100%  /dev
map -hosts      0Bi   0Bi   0Bi   100%     0     0 100%  /net
map auto_home   0Bi   0Bi   0Bi   100%     0     0 100%  /home
```

现在，插入你的 Micro SD 卡，再运行一次刚才的命令，你会发现多了一个“`/dev/disk1s1`”的设备。把它先记下来，等一会儿把它转换成原始设备名称时会用到的。

```
xenogears:Downloads john$ df -h
Filesystem      Size  Used Avail Capacity  iused   ifree %iused  Mounted on
/dev/disk0s2    232Gi 122Gi 110Gi    53% 32108705 28731037   53%  /
devfs           185Ki 185Ki   0Bi   100%     640     0 100%  /dev
map -hosts      0Bi   0Bi   0Bi   100%     0     0 100%  /net
map auto_home   0Bi   0Bi   0Bi   100%     0     0 100%  /home
/dev/disk1s1    74Mi  29Mi  46Mi   39%     0     0 100%  /Volumes/NO
NAME
```

接下来，你需要先从系统中把卡卸载掉，以便“dd”这个工具可以将映像写入到 Micro SD 卡中。在终端窗口输入“sudo diskutil unmount /dev/disk1s1”，然后输入你的系统密码。

```
xenogears:Downloads john$ sudo diskutil unmount /dev/disk1s1
WARNING: Improper use of the sudo command could lead to data loss
or the deletion of important system files. Please double-check your
typing when using sudo. Type "man sudo" for more information.

To proceed, enter your password, or type Ctrl-C to abort.

Password:
Volume NO NAME on disk1s1 unmounted
```

如果操作成功的话，你会看到最后那一行字：“Volume XXX unmounted”中的 XXX 是你 Micro SD 卡的名称。现在我们就可以往卡上写入映像了。

记得刚才让你记下来的设备名么？现在就是用到它的时候了。将你的设备名（例如/dev/disk1s1）最后的 s1 去掉，然后在 disk 前面加上 r，变成“/dev/rdisk1”，这样你就得到 Micro SD 卡的原始设备名称了。

也就是说，“/dev/disk1s1” = “/dev/rdisk1”。懂了吗？

然后在终端中输入以下命令：“sudo ddb s=1m if=2012-06-15-wheezy-raspbian.img of=/dev/YOURDISKNAME”。将 YOURDISKNAME 改成你的原始设备名称，通常这个都是 /dev/rdisk1。

```
xenogears:Downloads john$ sudo dd bs=1m if=2012-07-15-wheezy-raspbian.img of=/dev/rdisk1
1850+0 records in
1850+0 records out
1939865600 bytes transferred in 106.766989 secs (18169152 bytes/sec)
```

写入的过程需要一点时间。当“dd”完成了它的工作以后，将桌面的 Micro SD 卡图标拖进回收站中以将 Micro SD 卡安全移除。

准备工作就完成了！直接到“初次启动你的 Pi”部分吧。

Our prep work is finished! Skip ahead to "Booting your Pi for the first time."

如果你使用的是 Linux

无论你使用的是哪一个发行版的 Linux，“dd”这个工具通常都是默认安装的。对于接下来的操作，我们假设：

- 1) 你的 sudo 程序已经设置妥当，而且你知道如何使用它
- 2) 你的 Linux 中已安装 fdisk 工具（大多数发行版都默认安装）

在这里我们使用的是最为流行的 Linux 发行版 Ubuntu，其中“dd”，“fdisk”已默认安装，并且“sudo”也已预先为用户配置好了。

首先，从这里下载 Raspbian 的最新版本压缩包。同样，你也可以选择使用 BT 下载。在你下载完成以后，第一个操作自然就是解压。

打开一个终端窗口，然后使用“`cd ~`”命令将工作目录变更到你的主目录中。输入“`unzip imagename.zip`”，将 `imagename` 替换为刚才你下载文件的名字。

如果你找不到刚才下载的压缩包，你可以尝试使用“`cd ~/Downloads/`”进入下载目录寻找。

```
john@buntubox:~$ unzip 2012-08-16-wheezy-raspbian.zip
Archive: 2012-08-16-wheezy-raspbian.zip
  inflating: 2012-08-16-wheezy-raspbian.img
```

接下来，我们要确定你的 Micro SD 卡在系统中的设备名称。插入你的 Micro SD 卡，然后在终端中输入“`sudo fdisk -l`”。

```
john@buntubox: ~
Disk /dev/sda: 250.1 GB, 250059350016 bytes
255 heads, 63 sectors/track, 30401 cylinders, total 488397168 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: 0x2df8d261

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           63      48006134   240003036   83  Linux
/dev/sda2                480006135  488392064    4192965    5  Extended
/dev/sda5                480006198  488392064    4192933+   82  Linux swap / Solaris

Disk /dev/sdb: 15.9 GB, 15931539456 bytes
64 heads, 32 sectors/track, 15193 cylinders, total 31116288 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: 0x000714e9

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1                8192      122879     57344    c   W95 FAT32 (LBA)
/dev/sdb2               122880    31116287   15496704   83  Linux
john@buntubox:~$
```

注意看列出来的设备：你运行命令后得出的结果跟上面显示的应该很相似

这里看到，/dev/sda 是一个大小 250.1GB 的设备，而/dev/sdb 是一个 15.9GB 的设备。很明显，/dev/sdb 就是我们的 Micro SD 卡。留意一下/dev/sdb 下面显示的分区：/dev/sdb1 与/dev/sdb2，我们要在写入映像之前将这两个分区先从系统中卸载掉：

```
sudo umount /dev/sdb1
```

```
sudo umount /dev/sdb2
```

```
Device Boot      Start         End      Blocks   Id  System
/dev/sdb1        8192        122879     57344    c   W95 FAT32 (LBA)
/dev/sdb2       122880     31116287    15496704  83   Linux
john@buntubox:~$ sudo umount /dev/sdb1
john@buntubox:~$ sudo umount /dev/sdb2
john@buntubox:~$
```

接下来，我们使用“dd”将映像文件写入到 Micro SD 卡中：“sudo dd bs=1M if=2012-08-16-wheezy-raspbian.img of=/dev/sdb”。跟在 Mac 中操作相同，这个过程需要一些时间来完成。

```
john@buntubox:~$ sudo dd bs=1M if=2012-08-16-wheezy-raspbian.img of=/dev/sdb
1850+0 records in
1850+0 records out
1939865600 bytes (1.9 GB) copied, 451.975 s, 4.3 MB/s
john@buntubox:~$
```

如果你看到终端显示跟上面类似的文字，那么写入已经完成了。现在，你可以跳到“初次启动你的 Pi”部分继续阅读。

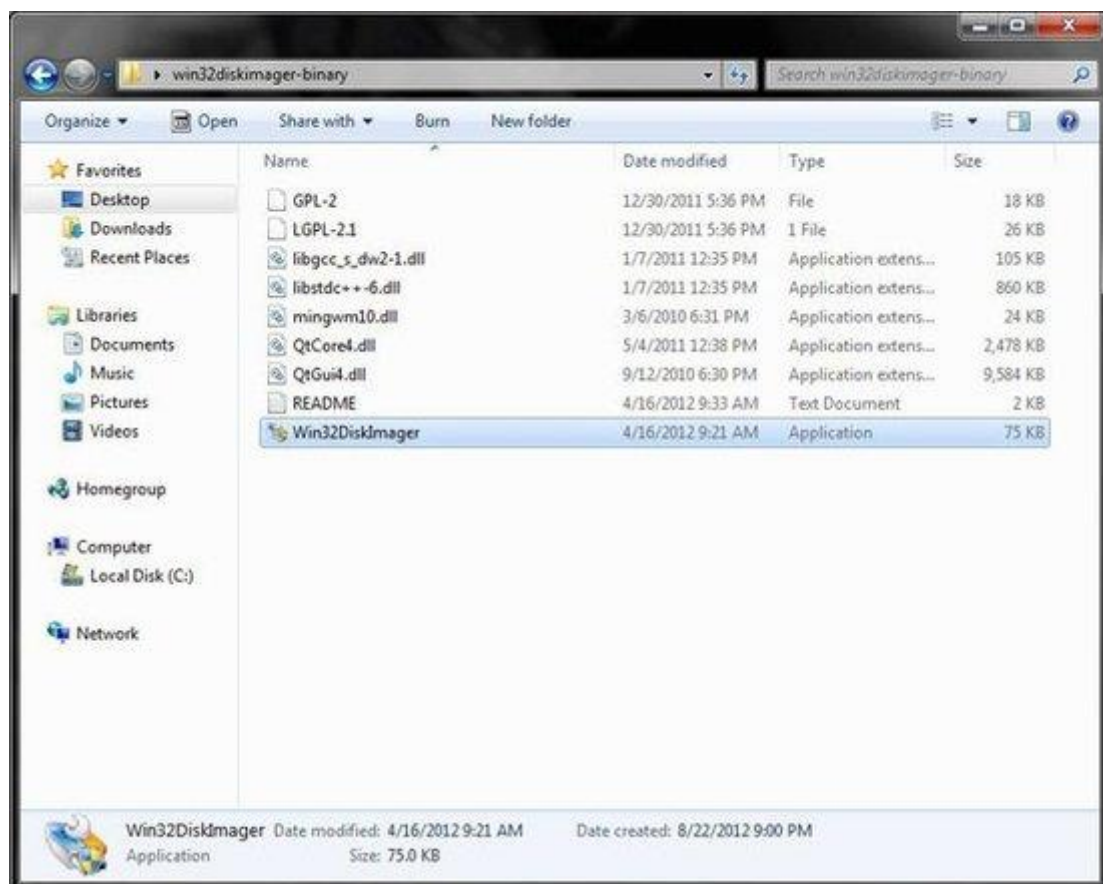
如果你正在使用 Windows

如果你正在使用的是 Windows 系统，有一个叫做 Win32DiskImager 的小工具可以帮你完成这项工作。把它下载下来以后，再把 Raspbian 的映像从这里下载下来。记得选择最新版本的 Raspbian。

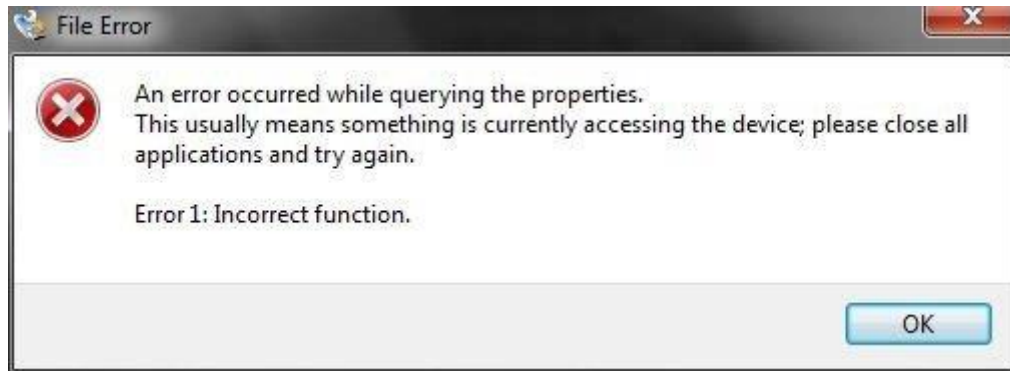
当你把 Win32DiskImager 和 Raspbian 映像文件都下载下来以后，把他们都解压出来。你可以把所有文件都解压到桌面的一个文件夹中，方便接下来的操作。

将你的 Micro SD 卡插入到你的读卡器中，留意一下读卡器在电脑中的盘符。将读卡器接到电脑以后，打开刚解压出来的

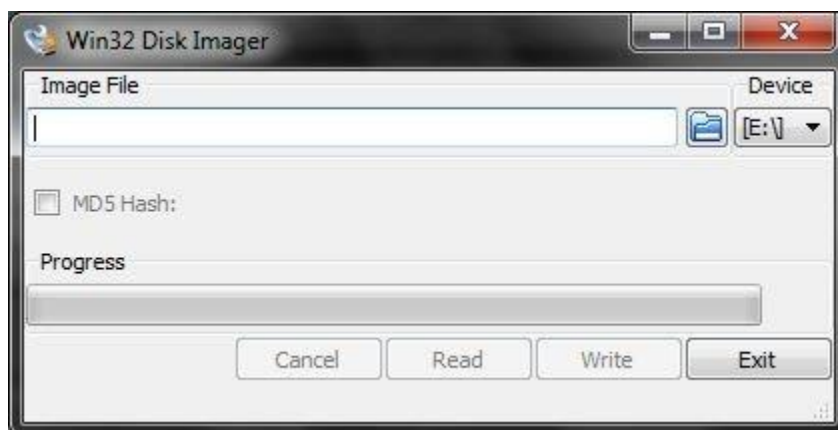
Win32DiskImager 程序：



在启动时，Win32DiskImager 可能会提示以下错误：（至少在 Windows 7 中是这样的）

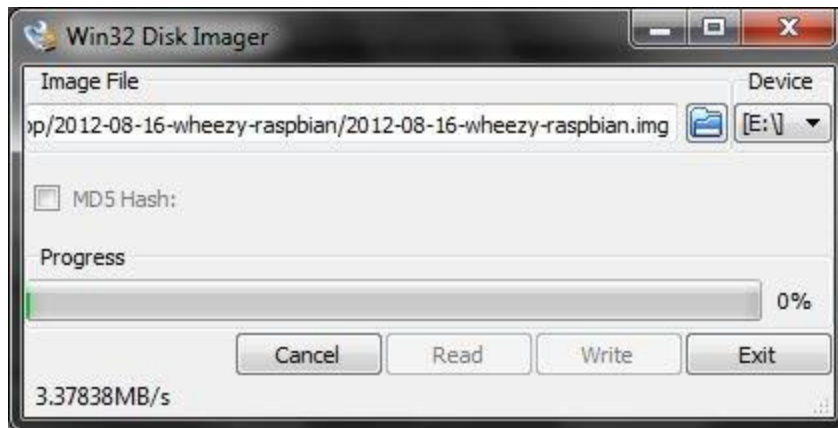


这个错误可以忽略掉，点 OK 继续



点击右边的蓝色文件夹图标，然后选中刚才解压出来的 Raspbian 映像文件。确保最右边 Device 中的盘符为你读卡器的盘符。

打开映像文件以后，点 Write，然后点一下 Yes 确定操作。整个写入过程需要一些时间来完成。



写入完毕以后，你会看到下面的信息，这时你就可以把你的 Micro SD 卡安全移除了。



初次启动你的 Pi

初次启动 Raspberry Pi 时，你会看到一个叫做 raspi-config 的配置工具。如果在日后使用过程中你需要更改这些设置，你可以通过在 Pi 的命令行中运行 raspi-config 来使用这个工具。在这里，你需要进行一些最基本的设置来继续使用你的 Pi。

```
Raspi-config

info          Information about this tool
expand_rootfs  Expand root partition to fill SD card
overscan      Change overscan
configure_keyboard  Set keyboard layout
change_pass    Change password for 'pi' user
change_locale  Set locale
change_timezone  Set timezone
memory_split   Change memory split
ssh           Enable or disable ssh server
boot_behaviour Start desktop on boot?
update        Try to upgrade raspi-config

                <Select>                <Finish>
```

首先，我们要选择 `expand_rootfs`。它的作用是将刚才写入到 Micro SD 卡中的映像文件大小扩展到整张 Micro SD 卡中。如果你使用的是一张较大的 Micro SD 卡（例如 16GB），那么你肯定像充分利用上面的空间。因为原本的映像只有大约 2GB 的大小，进行该操作就能将它扩展到与你的 Micro SD 卡同样的大小。

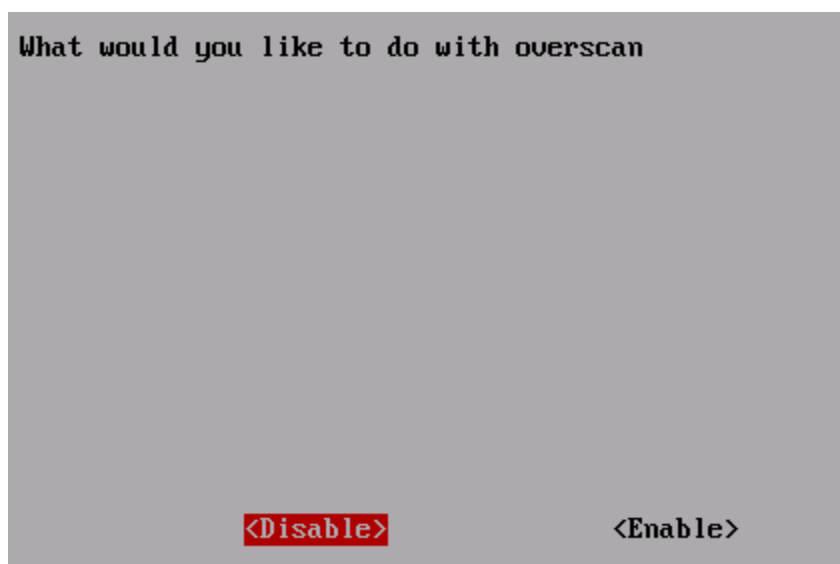
选中 `expand_rootfs` 选项，然后按下回车。你会看到如下提示，只需要再按一下回车就可以回到 `raspi-config` 的主菜单中。

```
Root partition has been resized.
The filesystem will be enlarged upon the next reboot

                <Ok>
```

接下来就是 overscan 选项。你会发现，屏幕显示的图像并没有完全占用你的显示器空间；所以，最后就是将 overscan 禁用掉，来让系统充分利用整个屏幕。但如果你的屏幕显示没有问题，那么你就可以跳过这个步骤。假设你需要禁用 overscan，那么将 overscan 选项选中后按下回车。

接下来的画面能够让你选择禁用(Disable)或者是启用(Enable) overscan。如果你在往后使用的过程中更换显示器或者电视机，你或许需要重新启用 overscan。

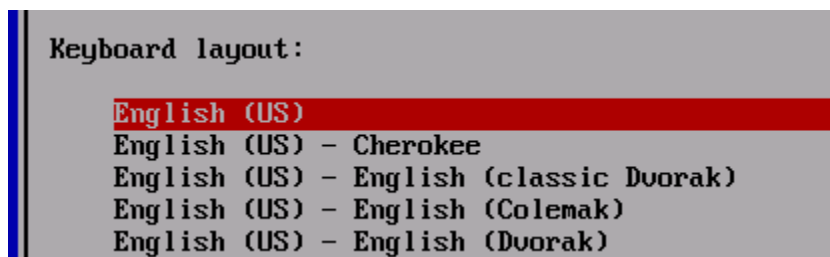
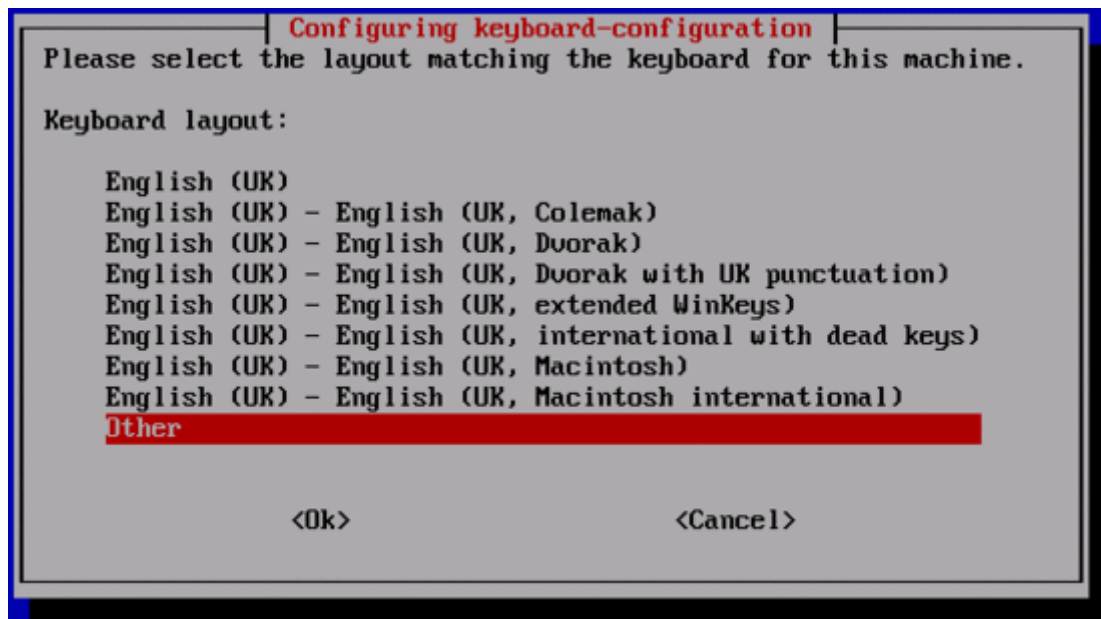


接下来，我们要确保我们的键盘正常工作。Raspbian 默认的是英国键盘布局，而我们在中国使用的键盘布局与美国的相同，因此我们要对它进行更改才能正常使用。

选中 `configure_keyboard`，然后按下回车。下面显示的画面中你会看到一个很长的列表，里面都是不同的键盘类型。你可以根据你的需要来选择，或者直接选择 `Generic 105-key (Intl) PC` 键盘。



在选择键盘类型以后，你需要为它选择一个键盘布局。刚开始显示的列表中，都只有英国的键盘布局，但是我们现在要选择美国的键盘布局，因此我们选中其他(Other)，然后再里面的列表选择 `English (US)`。



然后你会被问到关于辅助键的问题，选择默认的选项，并且在下一个画面中选择 No compose key 选项。同样，如果你需要更改这个设置，可以使用 raspi-config 来更改。

最后一个选项为是否打开 CTRL+ALT+BACKSPACE 的组合键。它可以在你的图形界面崩溃时，不需重启而将图形界面进程结束掉。

回到主菜单，下一步我们要设置一个用户密码。

```
Raspi-config

info          Information about this tool
expand_rootfs Expand root partition to fill SD card
overscan      Change overscan
configure_keyboard Set keyboard layout
change_pass   Change password for 'pi' user
change_locale Set locale
change_timezone Set timezone
memory_split  Change memory split
ssh           Enable or disable ssh server
boot_behaviour Start desktop on boot?
update        Try to upgrade raspi-config

<Select>                                <Finish>
```

选中 change_pass 然后按下回车。一个确认窗口弹出以后，在屏幕的底下会提示你输入一个新的 UNIX 用户密码。

```
Enter new UNIX password: _____
```

到这里就差不多完成了。接下来设置你的“区域”，这个主要会决定系统使用的字符集和语言。同样，如果你不想使用英国英语，那么你

可以在这里更改这些设置。在演示中我们选择的是美国英语，因此我们选择 en_US.UTF-8，并下翻列表将 en_GB 选项剔除

```
Locales to be generated:
[ ] de_LU.UTF-8 UTF-8
[ ] de_LU@euro ISO-8859-15
[ ] do_MV UTF-8
[ ] dz_BT UTF-8
[ ] el_CY ISO-8859-7
[ ] el_CY.UTF-8 UTF-8
[ ] el_GR ISO-8859-7
[ ] el_GR.UTF-8 UTF-8
[ ] en_AG UTF-8
[ ] en_AU ISO-8859-1
[ ] en_AU.UTF-8 UTF-8
[ ] en_BW ISO-8859-1
[ ] en_BW.UTF-8 UTF-8
[ ] en_CA ISO-8859-1
[ ] en_CA.UTF-8 UTF-8
[ ] en_DK ISO-8859-1
[ ] en_DK.ISO-8859-15 ISO-8859-15
[ ] en_DK.UTF-8 UTF-8
[ ] en_GB ISO-8859-1
[ ] en_GB.ISO-8859-15 ISO-8859-15
[ ] en_GB.UTF-8 UTF-8
[ ] en_HK ISO-8859-1
[ ] en_HK.UTF-8 UTF-8
[ ] en_IE ISO-8859-1
[ ] en_IE.UTF-8 UTF-8
[ ] en_IE@euro ISO-8859-15
[ ] en_IN UTF-8
[ ] en_NG UTF-8
[ ] en_NZ ISO-8859-1
[ ] en_NZ.UTF-8 UTF-8
[ ] en_PH ISO-8859-1
[ ] en_PH.UTF-8 UTF-8
[ ] en_SG ISO-8859-1
[ ] en_SG.UTF-8 UTF-8
[ ] en_US ISO-8859-1
[ ] en_US.ISO-8859-15 ISO-8859-15
[ *] en_US.UTF-8 UTF-8
[ ] en_ZA ISO-8859-1
[ ] en_ZA.UTF-8 UTF-8
[ ] en_ZM UTF-8
```

再下一个对话框中会让你选择默认的区域，选择你刚才选的区域然后按回车继续。

回到 raspi-config 的主菜单，设置适当的时区。选中 change_timezone 选项然后按下回车。你会先看到一个地区列表。



选择一个地区，然后下一个菜单中会显示该区域的具体位置。



再次回到主菜单，剩下还没选过的选项我们可以忽略，直接点击完成 (Finish)。系统会提示你，一些变更需要重启才能生效。重启以后，你会看到一个登录界面，如下：

```
My IP address is 192.168.11.22
Debian GNU/Linux wheezy/sid raspberrypi tty1
raspberrypi login: _
```

在这里，用户名为“pi”，密码就是你刚才设置的 UNIX 用户密码。

使用你的 Raspberry Pi

现在，你已经登录到了你之前设置的 Raspberry Pi 中了。你要做的第一件事，就是在命令行中输入“startx”来进入图形界面，以下我们会把这个界面称之为“窗口管理器”。

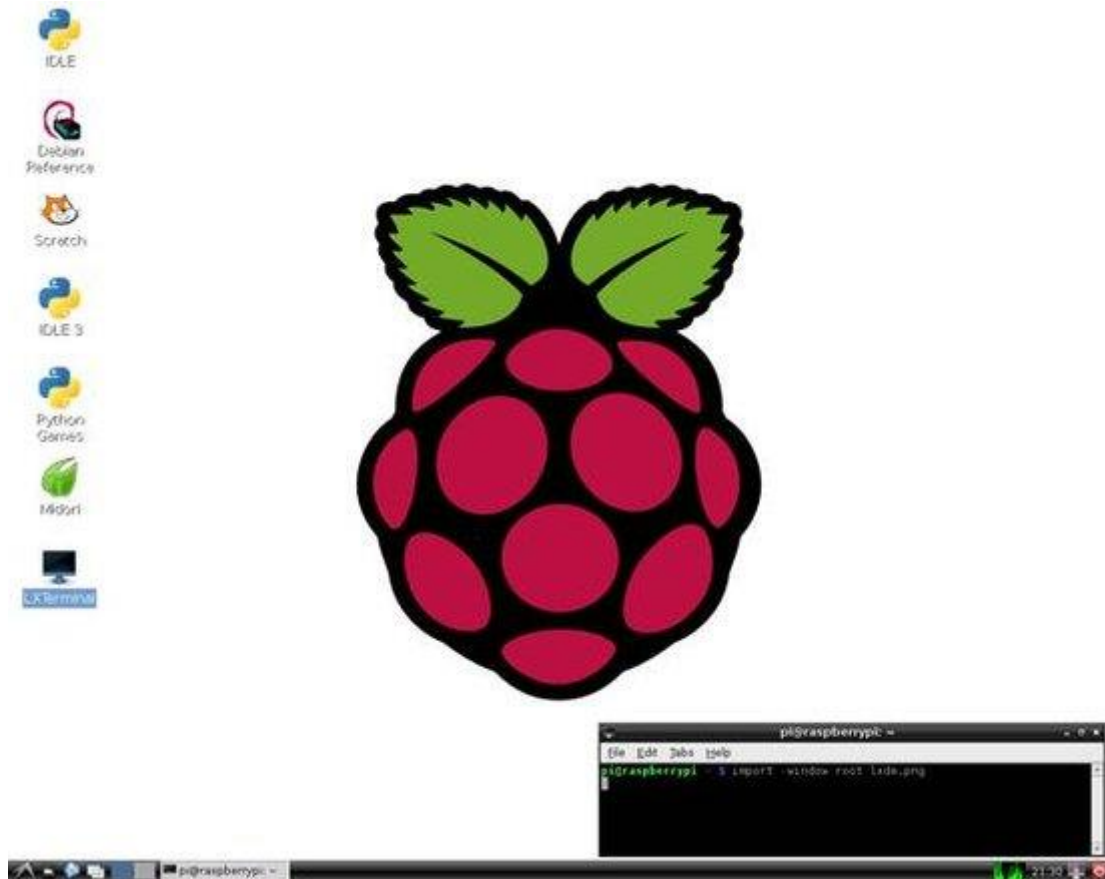
```
Debian GNU/Linux wheezy/sid raspberrypi tty1
raspberrypi login: pi
Password:
Last login: Tue Aug 21 21:24:50 EDT 2012 on tty1
Linux raspberrypi 3.1.9+ #168 PREEMPT Sat Jul 14 18:56:31 BST 2012 armv6l

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.

Type 'startx' to launch a graphical session
pi@raspberrypi ~ $
```

屏幕快速闪烁几次后，你应该会看到如下画面：（除了那个终端窗口以外）



欢迎使用 LXDE 窗口管理器。LXDE 是一个非常轻量级，并且功能齐全的图形界面，它能够很好地运行在 Raspberry Pi 中。如果你从来未使用过 LXDE，别担心，它用起来跟 Windows 非常相似。点击最左下角的图标，你会看到一个应用程序列表。

现在能干嘛呢？

当你把 Pi 配置完成，并且打开了 LXDE 窗口管理器以后，你就得到了一台运行完整 Debian 的 Linux 机器了。在这个时候，你可以使用 Midori 来浏览网页，架设一个网页服务器，或者是进行一些平常的工作。

但首先，我们要谈的是 Raspberry Pi 本身最注重的方面：教育。特别是，软件开发教育

在 Raspberry Pi 上撰写你的第一个程序

Raspbian 中预装 Python，它是 Raspberry Pi 的官方编程语言，还有 IDLE 3，一个 Python 的集成开发环境(IDE)。我们将会教你如何使用 IDLE 在 Raspberry Pi 上写一个简单的程序。

在学习一门新的编程语言时，通常第一件事都是写一个“Hello World!” 的小程序。接下来我们要教你的跟这个也差不多，不过要比它稍微花哨一些。这是一个会问你的名字，然后作自我介绍的 Python 程序。

首先，直接双击 LXDE 桌面上的图标来打开 IDLE 3



```
File Edit Shell Debug Options Windows Help
Python 3.2.3 (default, Jul 6 2012, 13:39:51)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
>>>
```

点击文件 > 新窗口，就会出现一个可以让你输入文字的空白窗口。

输入以下内容到你的新窗口中：

```
#my first Python program
```

```
username = input("Hello, I'm Raspberry Pi! What is your name?
```

```
")
```

```
print ('Nice to meet you, ' + username + ' have a nice day!')
```

如下图：

A screenshot of a Python IDE window. The title bar contains the menu items: File, Edit, Format, Run, Options, Windows, Help. The main text area contains the following Python code:

```
#my first Python Program  
username = input("Hello, I'm Raspberry Pi! What is your name? ")  
print ('Nice to meet you ' + username + ' have a nice day!')
```

The status bar at the bottom right shows "Ln: 1 Col: 0".

现在，点击文件 > 另存为，然后将你的文件保存为“hello”，点击保存。

现在就可以运行你刚才写的程序了！

点击运行 > 运行模块，或者直接按 F5 键。

当 IDLE 3 窗口中提示你“ Hello, I' m Raspberry Pi! What is your name?” 的时候，输入你的名字然后按下回车。你会看到 Raspberry Pi 对你做出回应。



```
File Edit Shell Debug Options Windows Help
Python 3.2.3 (default, Jul 6 2012, 13:39:51)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Hello, I'm Raspberry Pi! What is your name? John
Nice to meet you John have a nice day!
>>>
```

Ln: 8 Col: 4