Make:
Sensors
Projects and Experiments to Measure the World with Arduino and Raspberry Pi
Tero Karvinen, Kimmo Karvinen & Ville Valtokari

4 Development Tools-Raspberry Pi

Contents

1 Ready to work

2 Raspberry Pi from Zero to First Boot

3 Troubleshooting Your Raspberry Pi Installation

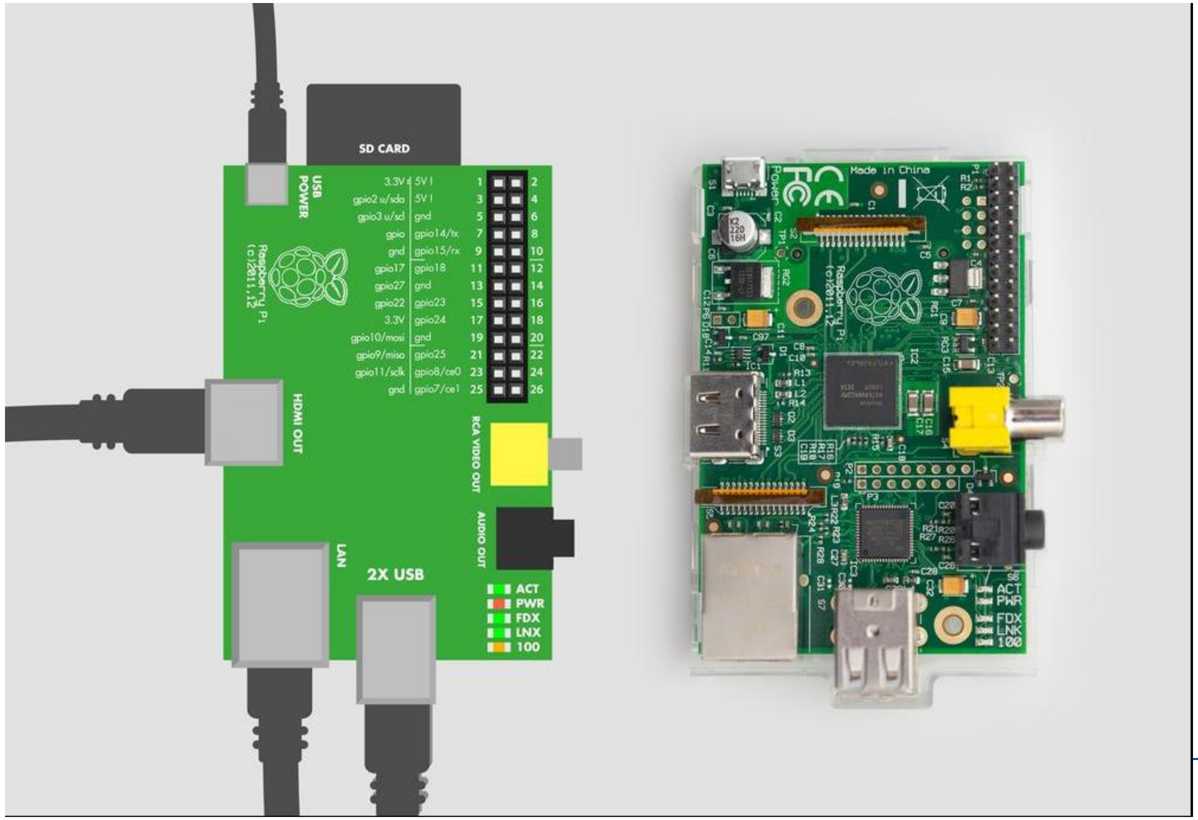
4 Feeling at Home in Linux

5 Connecting Electronics to Raspberry Pi Pins

6 GPIO in Python

Ready to work

- We recommend you start with the Raspberry Pi Model B, which includes wired Ethernet and enough USB ports for a mouse and keyboard.
- This makes it much easier to get started.



Ready to work

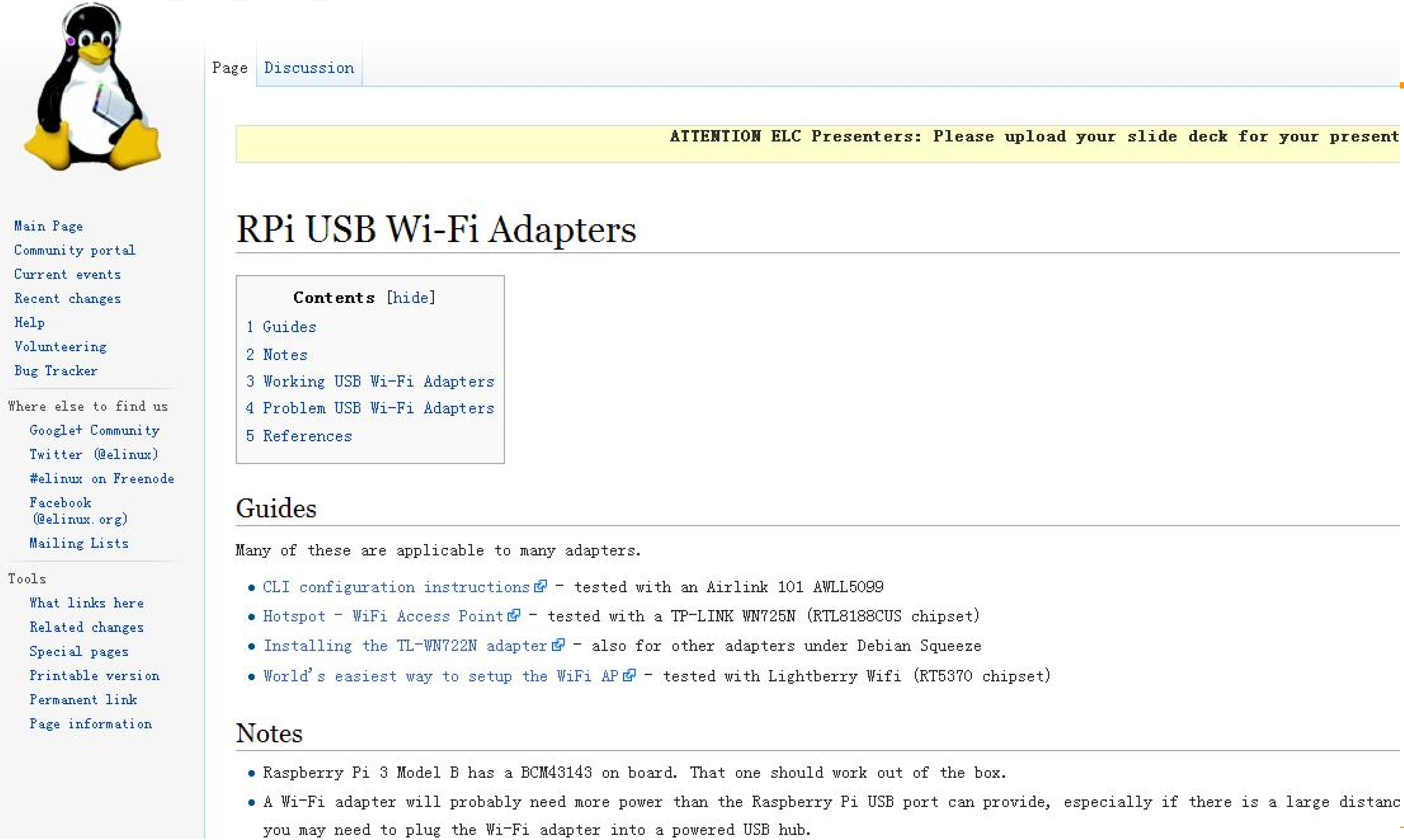
- Unless you buy your Raspberry Pi as part of a kit, it probably didn’t come with an enclosure
- A 4 GB SD memory card is big enough to fit the operating system.
- A bigger card may be less susceptible to wearing out over time (more storage to allocate to wear-leveling), so if you have an 8 GB or bigger card, even better.
- The Raspberry Pi can drive a full high-def display, and can even send sound over HDMI. Most likely, an HD television will work nicely as a display for your Pi.
- Having a keyboard and a mouse will make it easy to get started. Raspberry Pi Model B has exactly two USB ports, just enough for the mouse and the keyboard.

Ready to work

- If you want to add a USB WLAN adapter, you need a powered USB hub.
- After you install the operating system and boot to the graphical desktop environment, You'll be able to configure WiFi on your Pi by double-clicking the WiFi Config icon on the desktop

- See http://elinux.org/RPi_USB_Wi-Fi_Adapters for a list of WiFi adapters that are known to work with the Raspberry Pi.

http://elinux.org/RPi_USB_Wi-Fi_Adapters



Main Page
Community portal
Current events
Recent changes
Help
Volunteering
Bug Tracker

Where else to find us

Google+ Community
Twitter (@elinux)
#elinux on Freenode
Facebook (@elinux.org)
Mailing Lists

Tools

What links here
Related changes
Special pages
Printable version
Permanent link
Page information

ATTENTION ELC Presenters: Please upload your slide deck for your present

RPi USB Wi-Fi Adapters

Contents [hide]

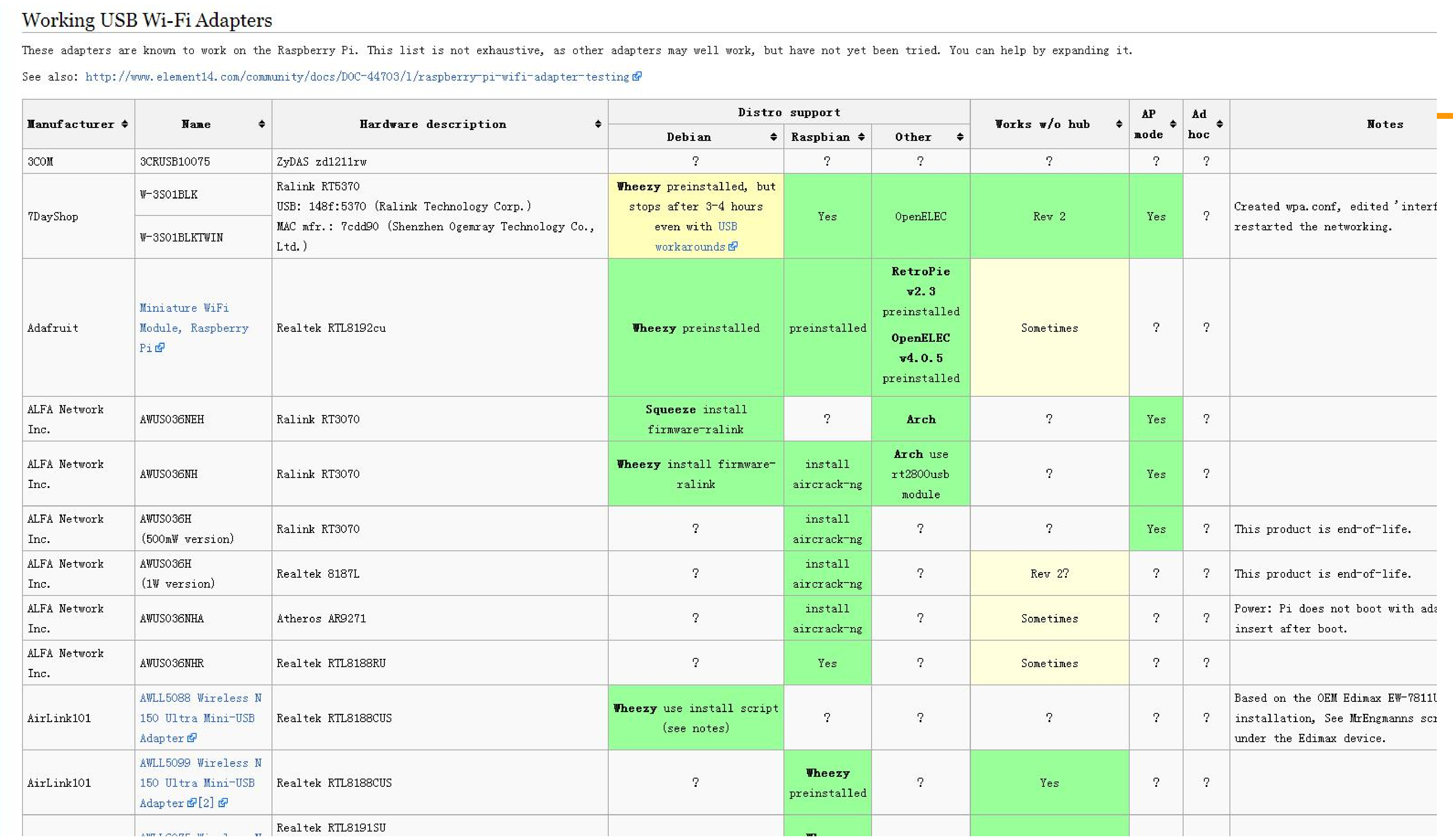
Guides

Many of these are applicable to many adapters.

- CLI configuration instructions - tested with an Airlink 101 AWLL5099
- Hotspot - WiFi Access Point - tested with a TP-LINK WN725N (RTL8188CUS chipset)
- Installing the TL-WN722N adapter - also for other adapters under Debian Squeeze
- World's easiest way to setup the WiFi AP - tested with Lightberry Wifi (RT5370 chipset)

Notes

- Raspberry Pi 3 Model B has a BCM43143 on board. That one should work out of the box.
- A Wi-Fi adapter will probably need more power than the Raspberry Pi USB port can provide, especially if there is a large distanc you may need to plug the Wi-Fi adapter into a powered USB hub.



Working USB Wi-Fi Adapters

These adapters are known to work on the Raspberry Pi. This list is not exhaustive, as other adapters may well work, but have not yet been tried. You can help by expanding it.

See also: http://www.element14.com/community/docs/DOC-44703/l/raspberry-pi-wifi-adapter-testing

Manufacturer	Name	Hardware description	Distro support			Works w/o hub	AP mode	Ad hoc	Notes
			Debian	Raspbian	Other				
3COM	3CRUSB10075	ZyDAS zd1211rw	?	?	?	?	?	?	
7DayShop	W-3S01BLK W-3S01BLKTWIN	Ralink RT5370 USB: 148f:5370 (Ralink Technology Corp.) MAC mfr.: 7cdd90 (Shenzhen Ogemray Technology Co., Ltd.)	**Wheezy** preinstalled, but stops after 3-4 hours even with USB workarounds	Yes	OpenELEC	Rev 2	Yes	?	Created wpa.conf, edited 'interf restarted the networking.
Adafruit	Miniature WiFi Module, Raspberry Pi	Realtek RTL8192cu	**Wheezy** preinstalled	preinstalled	**RetroPie v2.3** preinstalled **OpenELEC v4.0.5** preinstalled	Sometimes	?	?	
ALFA Network Inc.	AWUS036NEH	Ralink RT3070	**Squeeze** install firmware-ralink	?	**Arch**	?	Yes	?	
ALFA Network Inc.	AWUS036NH	Ralink RT3070	**Wheezy** install firmware-ralink	install aircrack-ng	**Arch** use rt2800usb module	?	Yes	?	
ALFA Network Inc.	AWUS036H (500mW version)	Ralink RT3070	?	install aircrack-ng	?	?	Yes	?	This product is end-of-life.
ALFA Network Inc.	AWUS036H (1W version)	Realtek 8187L	?	install aircrack-ng	?	Rev 2?	?	?	This product is end-of-life.
ALFA Network Inc.	AWUS036NHA	Atheros AR9271	?	install aircrack-ng	?	Sometimes	?	?	Power: Pi does not boot with ada insert after boot.
ALFA Network Inc.	AWUS036NHR	Realtek RTL8188RU	?	Yes	?	Sometimes	?	?	
AirLink101	AWLL5088 Wireless N 150 Ultra Mini-USB Adapter	Realtek RTL8188CUS	**Wheezy** use install script (see notes)	?	?	?	?	?	Based on the OEM Edimax EW-7811U installation, See MrEngmanns scr under the Edimax device.
AirLink101	AWLL5099 Wireless N 150 Ultra Mini-USB Adapter[2]	Realtek RTL8188CUS	?	**Wheezy** preinstalled	?	Yes	?	?	
		Realtek RTL8191SU							

Contents

1. Ready to work
2. Raspberry Pi from Zero to First Boot
3. Troubleshooting Your Raspberry Pi Installation
4. Feeling at Home in Linux
5. Connecting Electronics to Raspberry Pi Pins
6. GPIO in Python

Raspberry Pi from Zero to First Boot

- This chapter will get you up and running with the Raspberry Pi quickly. The first thing you need to do is to install Linux on the Raspberry Pi. It involves the following steps:
  - Download and extract the installer to a formatted SD card.
  - Insert the card into the Raspberry Pi and connect it to a keyboard, mouse, and monitor.
  - Turn it on, choose what to install, and wait.
- Once that's done, you are ready to boot the Pi into a graphical Linux desktop.

Raspberry Pi from Zero to First Boot

- You’ll need the following parts:
  - Raspberry Pi Model B
  - Micro USB cable and USB charger (or computer)
  - 4 GB SD card
  - Display with HDMI port
  - HDMI cable
  - USB mouse
  - USB keyboard

EXTRACT NOOBS*.ZIP

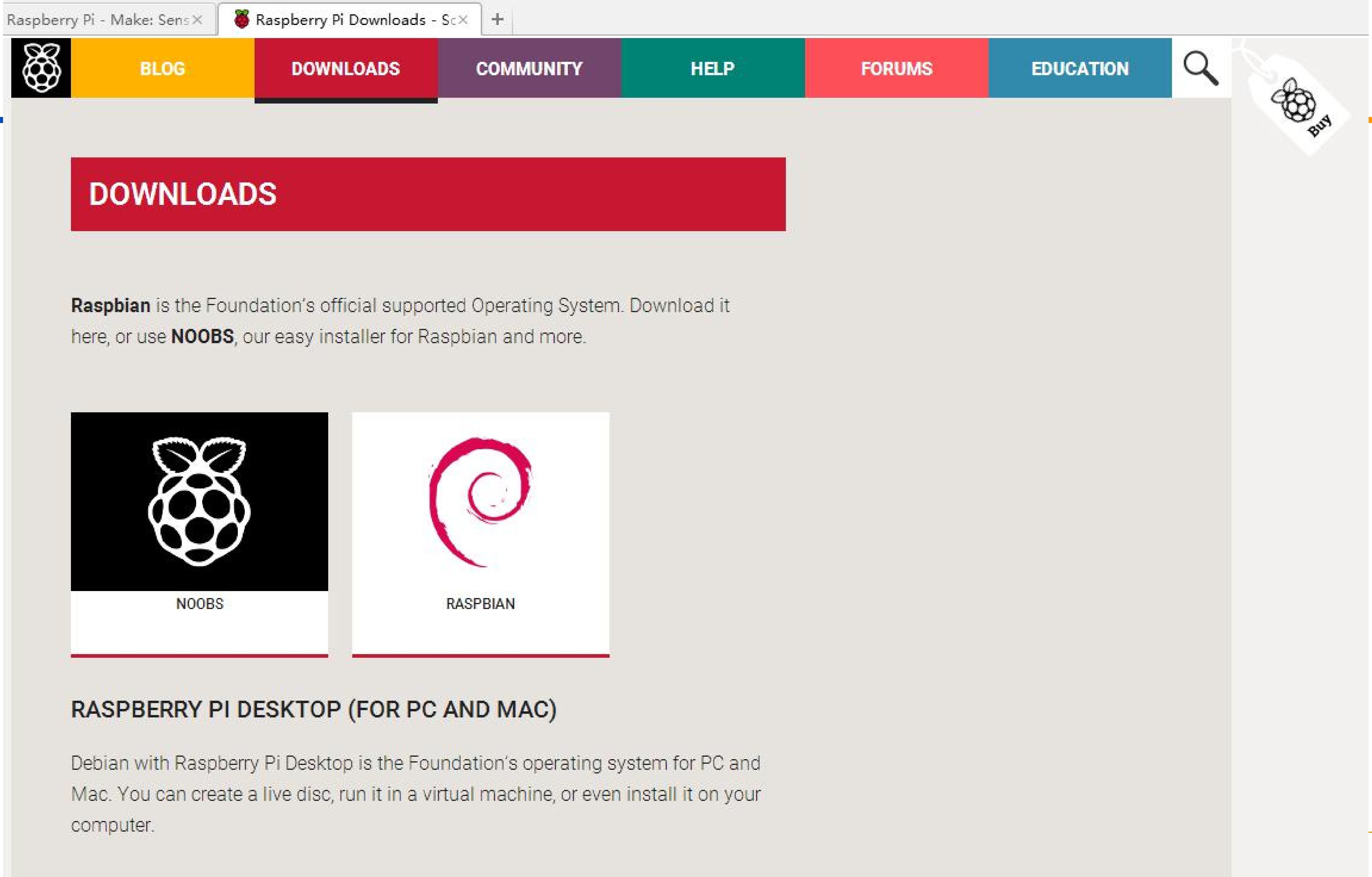
- Download *NOOBS_vX_Y_Z.zip* from http://raspberrypi.org/downloads.

下载 - 云加速由迅雷提供支持

NOOBS_v2_4_3.zip 1.46 GB - 0.00KB/S 00:00:00

- Insert the SD card into your computer. Most SD cards are FAT32 formatted at the factory, so unless you're using an SD card that you've formatted yourself, extracting the NOOBS zip to the SD card is enough.

- After you unzip the file, make sure that the bootcode.bin file is in the root (top-level) directory of the SD card.

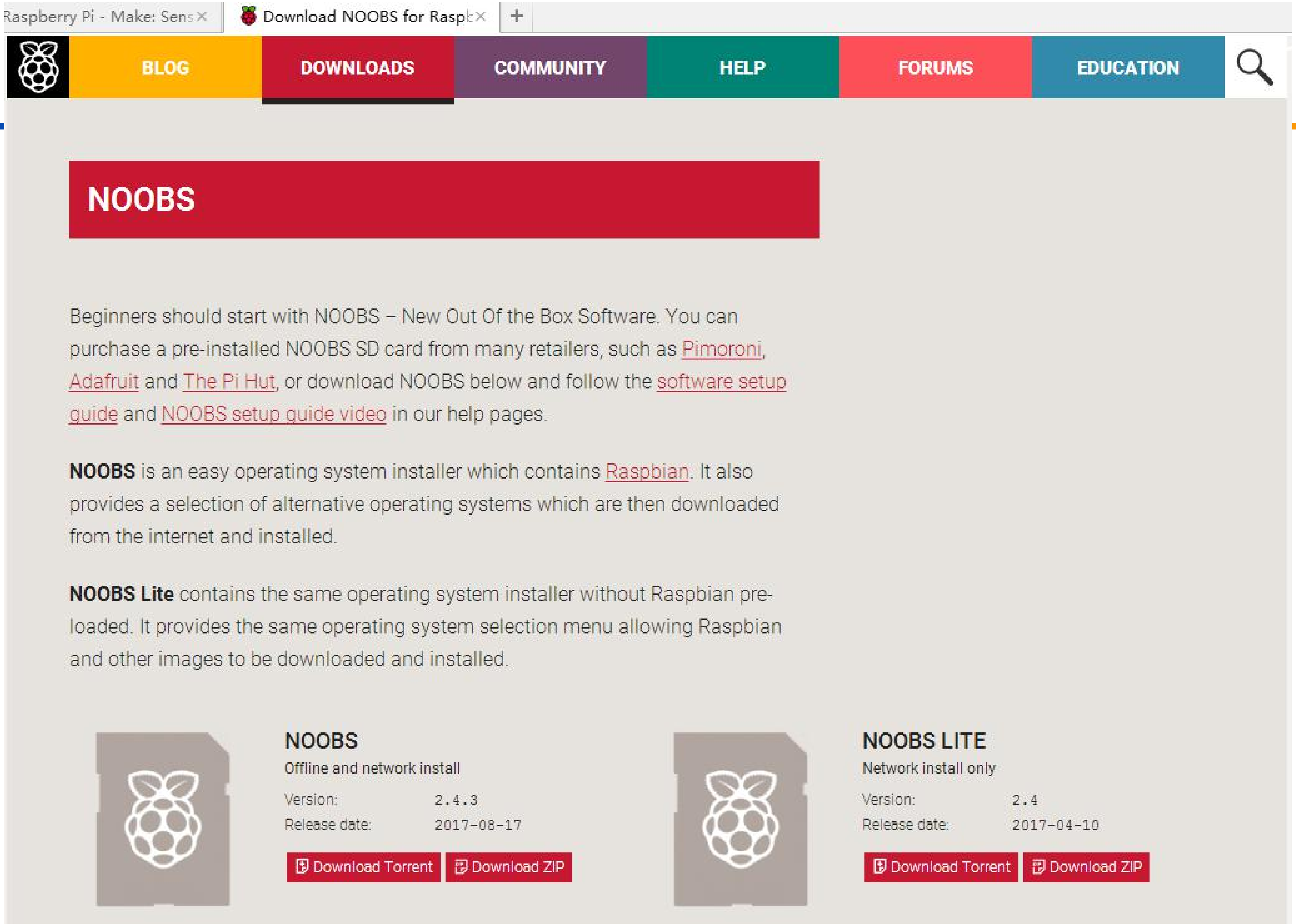


DOWNLOADS

Raspbian is the Foundation's official supported Operating System. Download it here, or use **NOOBS**, our easy installer for Raspbian and more.

RASPBERRY PI DESKTOP (FOR PC AND MAC)

Debian with Raspberry Pi Desktop is the Foundation's operating system for PC and Mac. You can create a live disc, run it in a virtual machine, or even install it on your computer.



NOOBS

Beginners should start with NOOBS – New Out Of the Box Software. You can purchase a pre-installed NOOBS SD card from many retailers, such as Pimoroni, Adafruit and The Pi Hut, or download NOOBS below and follow the software setup guide and NOOBS setup guide video in our help pages.

NOOBS is an easy operating system installer which contains Raspbian. It also provides a selection of alternative operating systems which are then downloaded from the internet and installed.

NOOBS Lite contains the same operating system installer without Raspbian pre-loaded. It provides the same operating system selection menu allowing Raspbian and other images to be downloaded and installed.

NOOBS

Offline and network install

Version: 2.4.3

Release date: 2017-08-17

NOOBS LITE

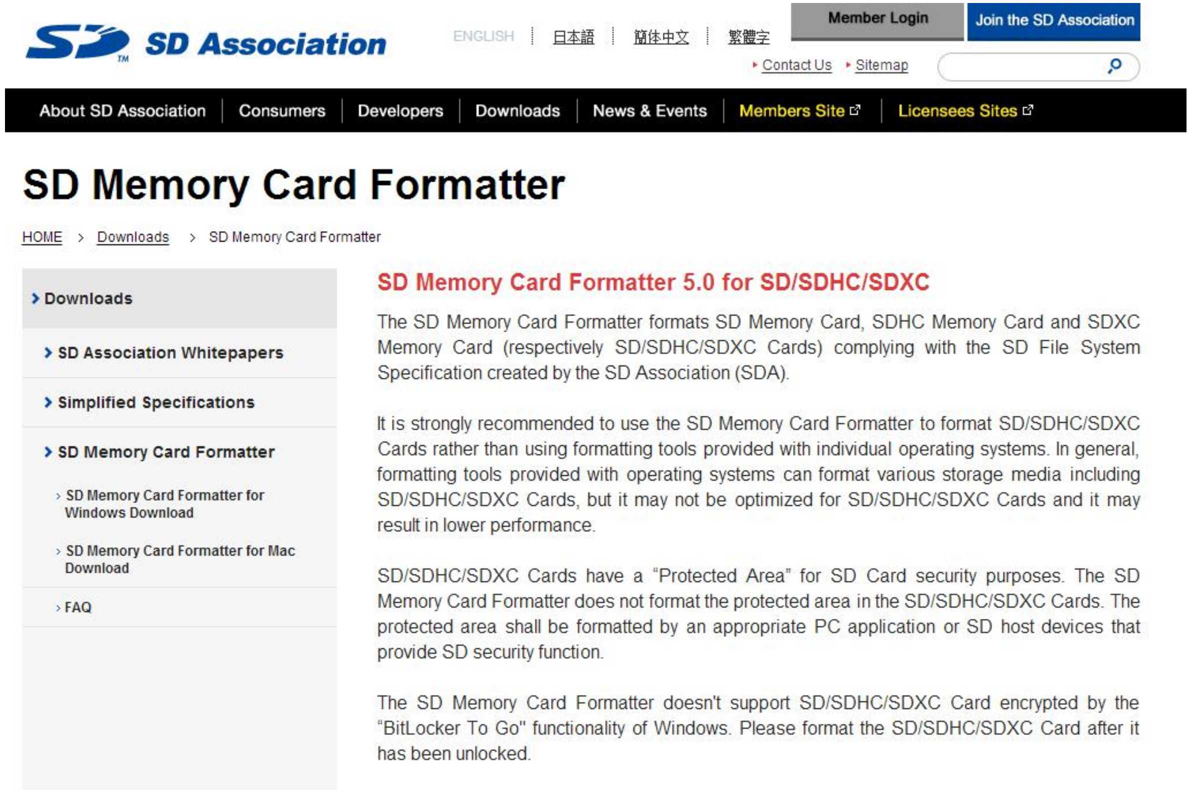
Network install only

Version: 2.4

Release date: 2017-04-10

EXTRACT NOOBS*.ZIP

- If you need to format the SD card, use the formatting tool from the SD Card Association. (https://www.sdcard.org/downloads/formatter_4/)


SD Association
ENGLISH | 日本語 | 簡体中文 | 繁體字
Member Login
Join the SD Association
Contact Us
Sitemap
About SD Association
Consumers
Developers
Downloads
News & Events
Members Site
Licensees Sites
HOME > Downloads > SD Memory Card Formatter
Downloads
SD Association Whitepapers
Simplified Specifications
SD Memory Card Formatter
SD Memory Card Formatter for Windows Download
SD Memory Card Formatter for Mac Download
FAQ

SD Memory Card Formatter

SD Memory Card Formatter 5.0 for SD/SDHC/SDXC

The SD Memory Card Formatter formats SD Memory Card, SDHC Memory Card and SDXC Memory Card (respectively SD/SDHC/SDXC Cards) complying with the SD File System Specification created by the SD Association (SDA).

It is strongly recommended to use the SD Memory Card Formatter to format SD/SDHC/SDXC Cards rather than using formatting tools provided with individual operating systems. In general, formatting tools provided with operating systems can format various storage media including SD/SDHC/SDXC Cards, but it may not be optimized for SD/SDHC/SDXC Cards and it may result in lower performance.

SD/SDHC/SDXC Cards have a "Protected Area" for SD Card security purposes. The SD Memory Card Formatter does not format the protected area in the SD/SDHC/SDXC Cards. The protected area shall be formatted by an appropriate PC application or SD host devices that provide SD security function.

The SD Memory Card Formatter doesn't support SD/SDHC/SDXC Card encrypted by the "BitLocker To Go" functionality of Windows. Please format the SD/SDHC/SDXC Card after it has been unlocked.

- In modern versions of Linux, Windows, and Mac you can just double-click or right-click the NOOBS zip file to extract it.
- For older versions of Windows, you can install 7zip to let you extract zip files.

CONNECT CABLES

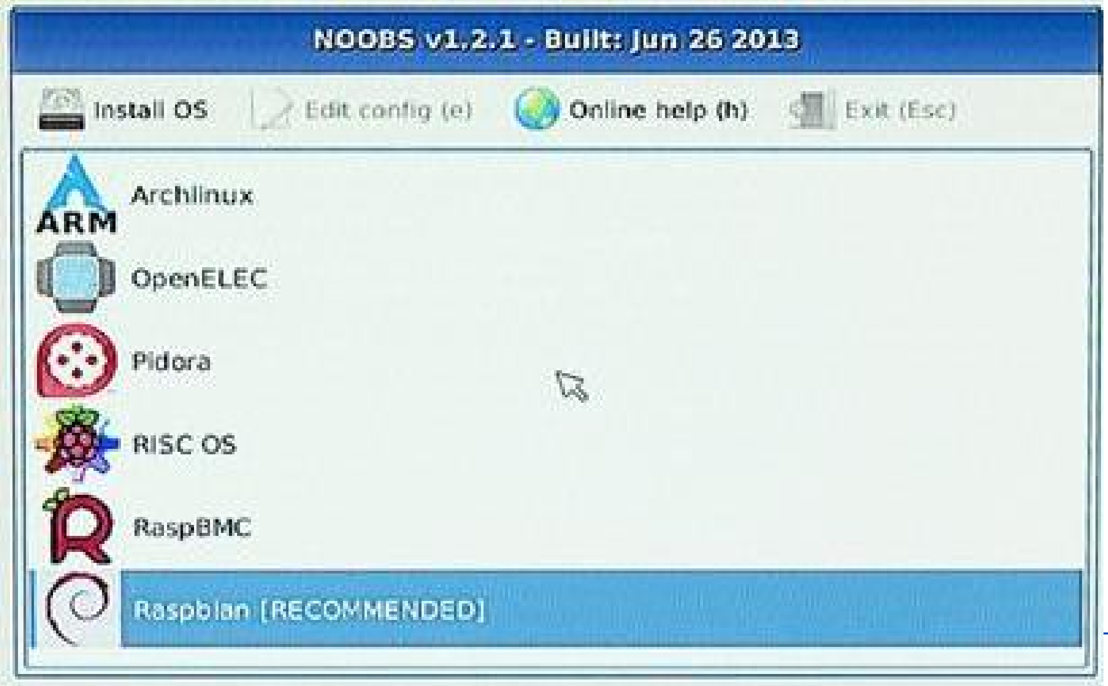
- Connecting the cables is easy, because each cable will fit only its correct socket. Plug the mouse and the keyboard into the Raspberry Pi’s USB ports. If you’re using an HDMI monitor, connect an HDMI cable between the monitor and Raspberry Pi. If you’re using an NTSC or PAL monitor, use a composite video cable to connect the yellow plug on the Raspberry Pi to the monitor.
- Next, connect the micro USB cable to Raspberry to supply power. Plug that cable into either a computer’s USB port or a 5 volt USB charger that provides at least 700 mA.

BOOT AND INSTALL RASPBIAN

- As soon as you connect power to the Raspberry Pi, it boots. No power switch is needed.
- If nothing appears on the screen, you may need to select the right output mode for the Raspberry Pi. The default output mode is HDMI, but if you are connected via HDMI and see nothing, try pressing 2 on the keyboard connected to your Raspberry Pi to select HDMI Safe Mode. If you are connected via the composite (yellow) connector, press 3 for a PAL monitor or television, or 4 for an NTSC monitor or television.

BOOT AND INSTALL RASPBIAN

- You are greeted with a graphical menu of different operating systems as well as language and keyboard type.
- Choose “Raspbian [RECOMMENDED]” (Figure 1-2) and select your language and type of keyboard you’ll be using.



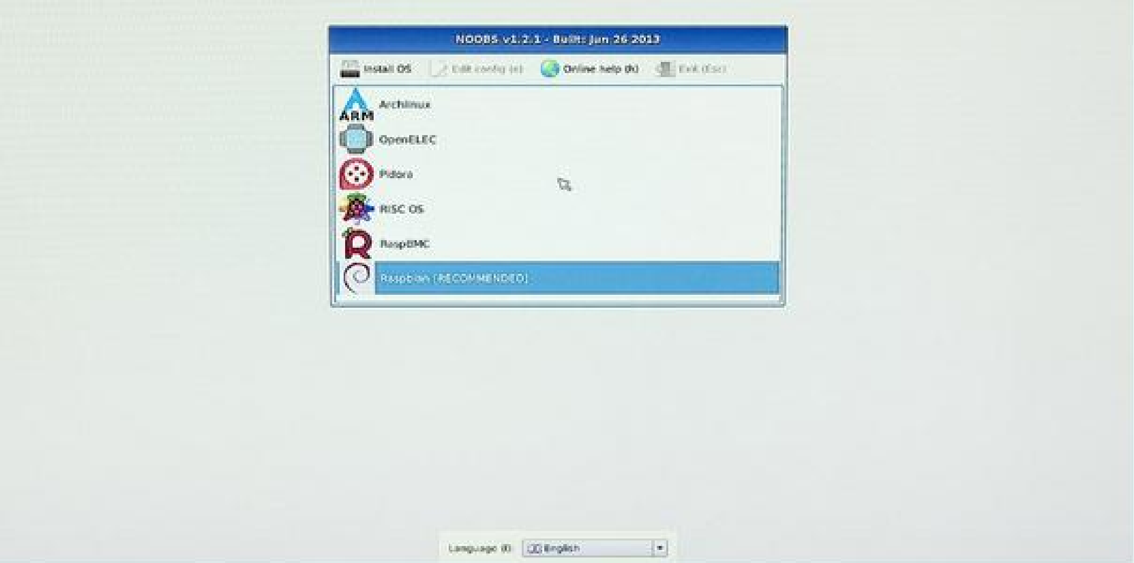
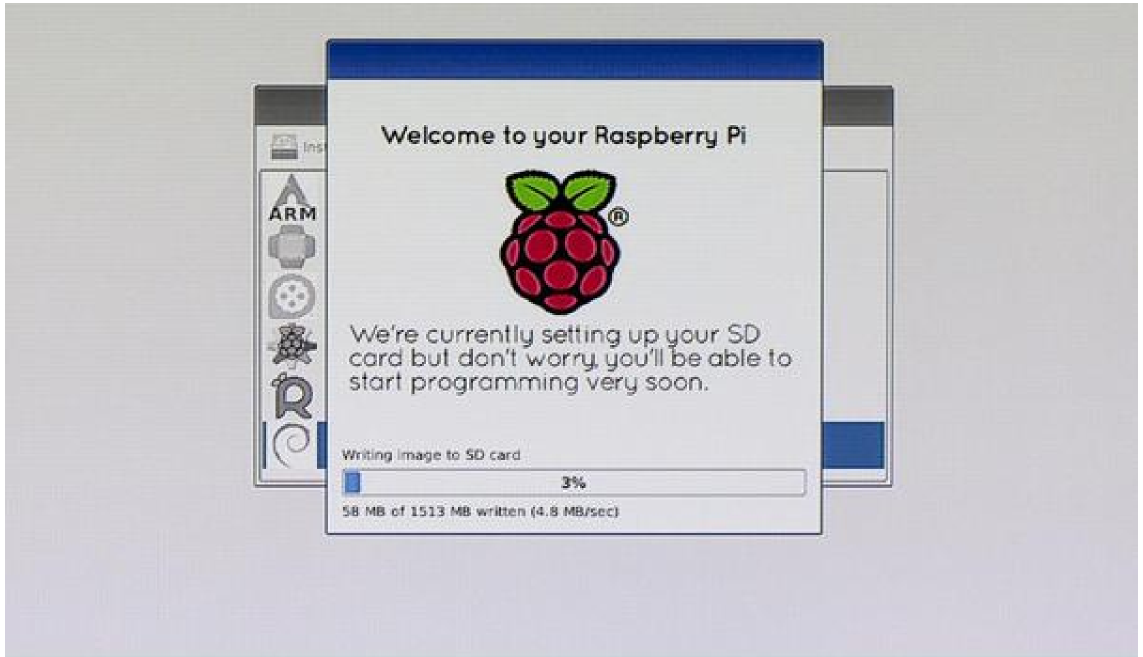


Figure 1-2. Choosing an operating system

BOOT AND INSTALL RASPBIAN

- If you know any Debian, Mint or Ubuntu, you will feel at home with this choice;
- Raspbian takes a few minutes to finish installing (Figure 1-3).
- After the installer completes, it will indicate that it installed the operating system successfully.
- Press Enter or click OK to reboot.



Welcome to your Raspberry Pi

We're currently setting up your SD card but don't worry, you'll be able to start programming very soon.

Writing image to SD card

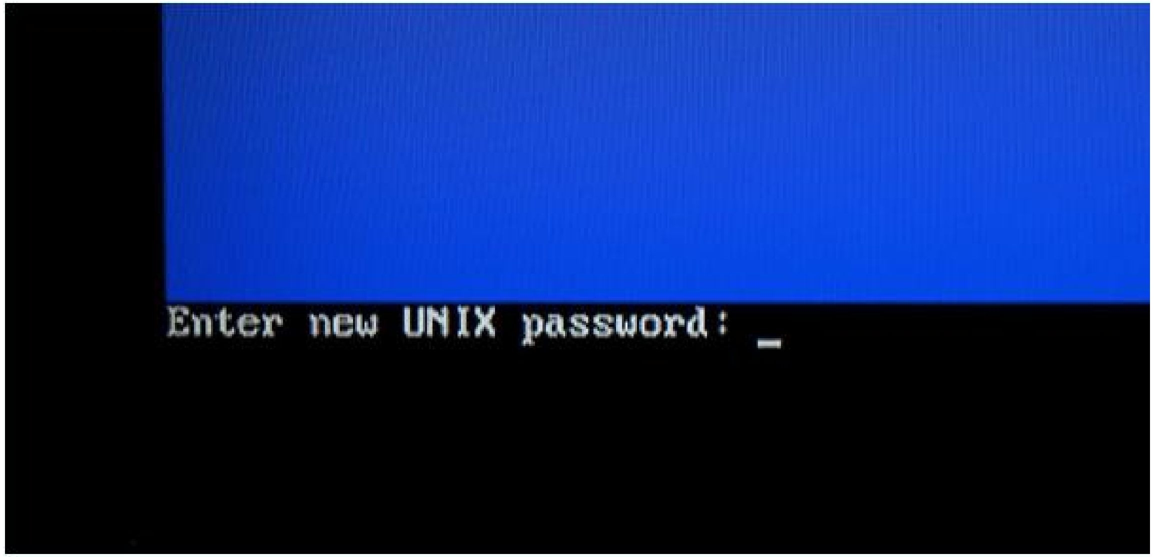
3%

58 MB of 1513 MB written (4.8 MB/sec)

Figure 1-3. Raspbian installs

BOOT AND INSTALL RASPBIAN

- The Raspberry Pi configuration utility opens. Use arrow keys and Tab to navigate, and press Enter/Return to select an option, as shown in Figure 1-4.
- You’ll want to enable the Boot to Desktop option. When you have finished changing settings, use Tab to select Finish and reboot when asked.
- After the Raspberry Pi reboots, it will start up in a graphical desktop and will log you in automatically.



```
Enter new UNIX password: _
```

Figure 1-4. Changing your password

BOOT AND INSTALL RASPBIAN

- If you have chosen not to enable Boot to Desktop, you'll always start in the command-line interface. Log in as "raspberry" with password "pi" (unless you changed the password).
- After you log in, type startx to start the X Window System, which is the graphical desktop.
- Welcome to Linux! You have now installed Raspbian on Raspberry Pi (Figure 1-5).

BOOT AND INSTALL RASPBIAN

- ➢ Welcome to Linux!   You have now installed Raspbian on Raspberry Pi (Figure 1-5).

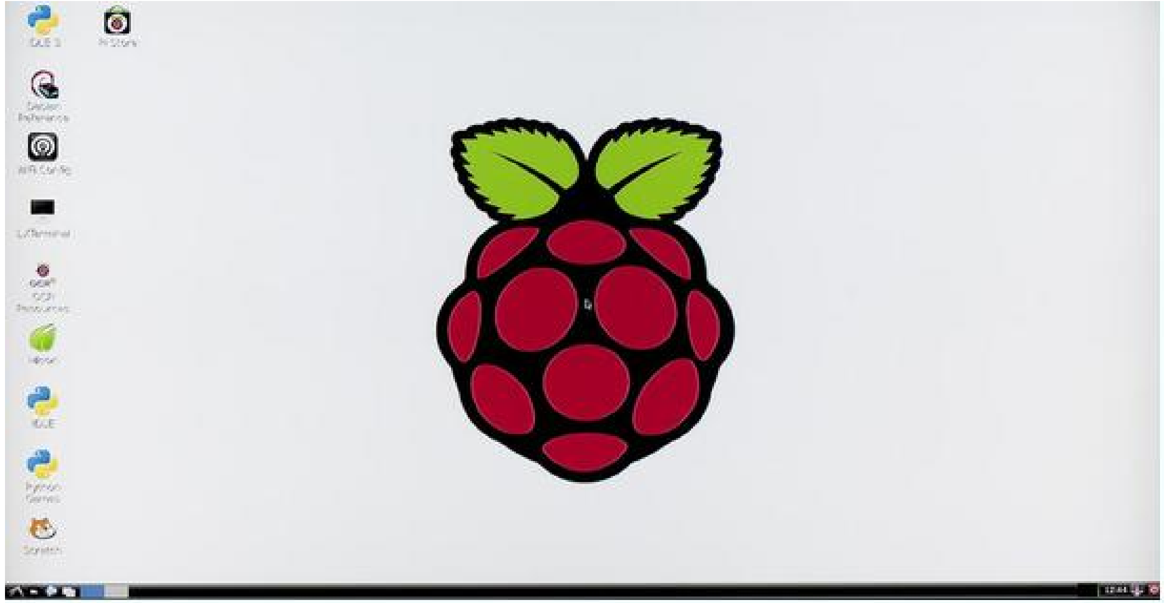


Figure 1-5. Welcome to Linux

- ➢ To turn off your Raspberry Pi, double-click the Shutdown icon on the desktop. After it finishes the power down process, you should unplug the power.

Contents

1. Ready to work
2. Raspberry Pi from Zero to First Boot
3. Troubleshooting Your Raspberry Pi Installation
4. Feeling at Home in Linux
5. Connecting Electronics to Raspberry Pi Pins
6. GPIO in Python

TROUBLESHOOTING YOUR RASPBERRY PI INSTALLATION

- Is your card FAT32 formatted?
- Red power LED (PWR) not lit?
- Black screen (but red PWR LED lit)?
- Four colored boxes on the screen?
- Boot fails with some error messages?
- Messed up your operating system?
- No Internet?

TROUBLESHOOTING YOUR RASPBERRY PI INSTALLATION

- Is your card FAT32 formatted?
  - 如果你无法使用SD卡启动系统，有可能是因为SD卡未被正确格式化。
  - 在Linux系统中可以使用内置的图形化分区工具来格式化SD卡；
  - 在Windows和MAC系统中，可以使用SD卡协会的格式化工具，进行格式化。

TROUBLESHOOTING YOUR RASPBERRY PI INSTALLATION

- Is your card FAT32 formatted?
- Red power LED (PWR) not lit?
  - □ 红色电源指示灯不亮？或者闪烁？或者变暗？或者亮起片刻之后又立即熄灭？
  - □ 这有可能是树莓派没有获得足够大的电源。
  - □ 需要提供5V并超过1A电流的USB电源适配器。
  - □ 有可能笔记本电脑的USB供电不足，可以尝试换到台式机，后者电力充足的PAD也可以。

TROUBLESHOOTING YOUR RASPBERRY PI INSTALLATION

- Is your card FAT32 formatted?
- Red power LED (PWR) not lit?
- Black screen (but red PWR LED lit)?
  - 这可能是由于树莓派无法读取SD卡中的启动程序造成的。
  - 把电源关闭，然后重新插入SD卡，确保接触良好。
  - 确认启动程序中的第一个文件bootcode.bin在SD卡的根目录下。
  - 如果还是黑屏，那就重新格式化SD卡，解压NOOBS压缩包，重新执行安装过程。
  - 或者换SD卡。

TROUBLESHOOTING YOUR RASPBERRY PI INSTALLATION

- Is your card FAT32 formatted?
- Red power LED (PWR) not lit?
- Black screen (but red PWR LED lit)?
- Four colored boxes on the screen?
  - 这是因为启动程序正常工作了，读取了SD卡，但是操作系统的Kernel.img却无法正常启动。
  - 再次重新格式化SD卡，解压NOOBS压缩包，重新执行安装过程。
  - 或者换SD卡。

TROUBLESHOOTING YOUR RASPBERRY PI INSTALLATION

- Is your card FAT32 formatted?
- Red power LED (PWR) not lit?
- Black screen (but red PWR LED lit)?
- Four colored boxes on the screen?
- Boot fails with some error messages?
  - 尝试拔出所有的USB设备，如键盘、鼠标和WiFi适配器，仅保留SD卡，显示器和电源。
  - 拔出SD卡再插入，确保接触良好。
  - 如果问题依旧，那么重新格式化SD卡，解压NOOBS压缩包，重新执行安装过程。

TROUBLESHOOTING YOUR RASPBERRY PI INSTALLATION

- Is your card FAT32 formatted?
- Red power LED (PWR) not lit?
- Black screen (but red PWR LED lit)?
- Four colored boxes on the screen?
- Boot fails with some error messages?
- Messed up your operating system?
  - 如果正常的命令无法运行，显示器上都是乱七八糟的信息，或者树莓派突然停止运行。那么，开机时按住shift键，选择重新安装Raspbian即可。这种操作高效，但是会删除SD卡中所有数据。如果还是有问题，那么重新格式化SD卡，解压NOOBS压缩包，重新执行安装过程。

TROUBLESHOOTING YOUR RASPBERRY PI INSTALLATION

- Is your card FAT32 formatted?
- Red power LED (PWR) not lit?
- Black screen (but red PWR LED lit)?
- Four colored boxes on the screen?
- Boot fails with some error messages?
- Messed up your operating system?
- No Internet?
  - 如果在启动前已经连接了以太网，它应该像标准的网络一样正常工作。
  - 否则，就按照大家熟悉的方式来排除网络连接问题（Ping，查看指示灯，检查接入端等）

Contents

1. Ready to work
2. Raspberry Pi from Zero to First Boot
3. Troubleshooting Your Raspberry Pi Installation
4. Feeling at Home in Linux
5. Connecting Electronics to Raspberry Pi Pins
6. GPIO in Python

Feeling at Home in Linux

- Raspberry Pi *is* Linux. Well, it’s built on Linux.
- The Linux operating system is composed of the kernel as well as thousands of utilities and applications from various sources.
- Raspberry Pi is not a workstation-class device. In terms of computing power, it’s more comparable to an entry-level portable tablet or a mobile phone.
- So even if you boot to its graphical desktop, don’t expect to dump your laptop or desktop computer just yet.
- Extremely low computing power combined with little memory means that applications like LibreOffice and Mozilla Firefox won’t be usable.

Command-Line Interface （CLI） is Everywhere, Forever

- Most of the commands you’ll use with Raspberry Pi are the same you’d use on a Mac or Linux computer, and are similar even to the command-line tools on Windows.
- As you may know, most of the world’s servers run Linux. Google, Facebook, Amazon, and most supercomputers run Linux.
- Web servers don’t run a graphical user interface, so that’s why most programmers and system administrators must know how to use the command-line interface (CLI).
- You can use these commands on a Linux desktop or laptop, too.

LOOKING AROUND

- After you boot your Raspberry Pi into the graphical user interface, you can start up the command-line interface by double-clicking the LXTerminal icon on your desktop.
- Besides being great for browsing the Web, the Midori web browser is useful if you want to copy and paste some sample code from http://botbook.com.
- The prompt $ means Linux is waiting for your command.

LOOKING AROUND

- **Type pwd** to print your current working directory. Linux answers with a path, such as /home/pi/.
- To list the files in the working directory, **type ls** and press Return or Enter.
- Whenever you get an error like *No such file or directory*, just use pwd to see where you are and ls to see what files are in the working directory.

LOOKING AROUND

- To edit (or create) a file called foo.txt, use **nano** foo.txt. Type some text, then press Control-X to save it (when prompted whether to save, type y. When asked for the filename, just press Enter or Return).
- To edit the file some more, type the command nano foo.txt again.

Text Files for Configuration

- In Linux, most things are text files. With just the commands ***nano***, along with the sudo command, a skillful hacker can change many system settings.
- All configuration in Linux is stored in text files. System-wide configuration files are in the **/etc/** directory and per user configuration is in the user‘s home directory, **/home/pi/**. Even the Pi’s input and output pins can be manipulated by editing text files under the **/sys/** directory.
- Later in this lesson, you will learn to connect sensors and LEDs to Raspberry using GPIO pins.

SUDO MAKE ME A SANDWICH

- Linux is well known for its robust security model. The separation of user privileges is one of its key features. Normal users can make changes only to files that affect their working environment. This means that they can modify files only in their home directory (/home/pi/) and in temporary working directories such as /tmp/.
- The super user, or root user, is all-powerful and can change any file on the system. To use root' s privileges, put sudo in front of a command. Putting sudo in front of the command runs that commands under the privileges of the root user.

SUDO MAKE ME A SANDWICH

- **For example**, Raspbian‘s package manager（apt） makes it very easy to install additional software, but you need to use root’s privileges to install anything.
- Before you install new software, you need to update the list of what‘s available with ***sudo apt-get update***. This requires a network connection because all the software packages are on a file server.
- Many Linux and Unix systems (such as OS X) are configured such that you need to type your user password when you use sudo. This is an extra safety step. By default, Raspberry Pi’s Raspbian operating system does not ask for this. Be careful using sudo, because you can easily make mistakes that would render the operating system unbootable.

SUDO MAKE ME A SANDWICH

- You can install any program from a repository by specifying its package name.
- Eg. To install ipython use ***sudo apt-get -y install ipython***. The package manager (apt) does everything for you.
    - □ The -y parameter tells the package manager to assume a “yes” answer to any questions it asks.)
- After a moment, you can run the newly installed package by typing ipython. Any python command will work here, but exit() will get you back to the command prompt (“$”), where you can type shell commands.

SUDO MAKE ME A SANDWICH

- Different prompts are one way of indicating which program you‘re talking to.
- When you see the shell, or command prompt (**$**), you can type the name of built-in shell commands as well as programs installed on your Raspberry Pi.
- When you see another prompt, such as the ipython prompt (**In [1]:**), you can type Python commands.

SUDO MAKE ME A SANDWICH

- Installing daemons (also known as servers) is just as easy.
- Try installing the most popular web server in the planet, Apache, with this command: ***sudo apt-get -y install apache2***.
- When it finishes, to browse your Apache web server from another computer, determine your IP address with the output of the ifconfig command and type that address into a web browser on another computer connected to your network.
- You will see at least two adapters listed in the output of ifconfig. Use the Ethernet adapter’s (eth0) address, or if you‘re using WiFi, use that adapter’s address.

Contents

1. Ready to work
2. Raspberry Pi from Zero to First Boot
3. Troubleshooting Your Raspberry Pi Installation
4. Feeling at Home in Linux
5. Connecting Electronics to Raspberry Pi Pins
6. GPIO in Python

Connecting Electronics to Raspberry Pi Pins

- The Pi’s GPIO (general-purpose input and output) pins let you connect electronic components directly to the Raspberry Pi.
- These pins are called general purpose because you can decide what purpose they serve, and you can even configure the same pin to be an input or an output at different times.

Connecting Electronics to Raspberry Pi Pins

- ➢ Throughout this course, you will learn the following:
  - □ Digital output (turn an LED on and off)
  - □ Digital resistance (detect whether a button is pressed or a sensor is active)
  - □ Digital input for very short pulses (used by sensors such as a distance sensor)
  - □ Analog resistance (analog resistance sensors for pressure, light, temperature)
  - □ Industry standard protocols, such as I2C and SPI (used by the Wii Nunchuk and analog-to-digital converters)

Connecting Electronics to Raspberry Pi Pins

- Unlike most other tutorials, we‘ll teach you how to use digital input and output without having to invoke root privileges all the time. This provides security and stability benefits.
- For digital input, you’ll learn to use the Pi‘s internal pull-up resistor, so that your circuit uses a minimum number of components.
- For measuring analog resistance, you’ll use an external analog-digital converter chip.
- Many components in everyday products communicate over industry standard protocols, such as I2C and SPI. You will see examples of both protocols later in the lesson.
- But first, let‘s show you how to use the most basic form of digital output.

Hello GPIO, Blink an LED

- In this “Hello GPIO World” example, you’ll attach a new LED to Raspberry Pi and blink it.
- We start all of our projects with a “Hello World” on any platform, on any language. So whenever you are about to build something more complicated, it’s a good idea to build this “Hello GPIO World” first.
- This lets you confirm that the hardware and software are functioning at the most basic level.
- If your “Hello World” example doesn’t work, you need to fix it before trying something more complicated.

Hello GPIO, Blink an LED

- Parts needed:
  - □ Raspberry Pi
  - □ Female-to-male jumper wires, black and green or yellow color
  - □ Solderless breadboard
  - □ 470 Ohm resistor (yellow-violet-brown stripes)
  - □ An LED

Hello GPIO, Blink an LED

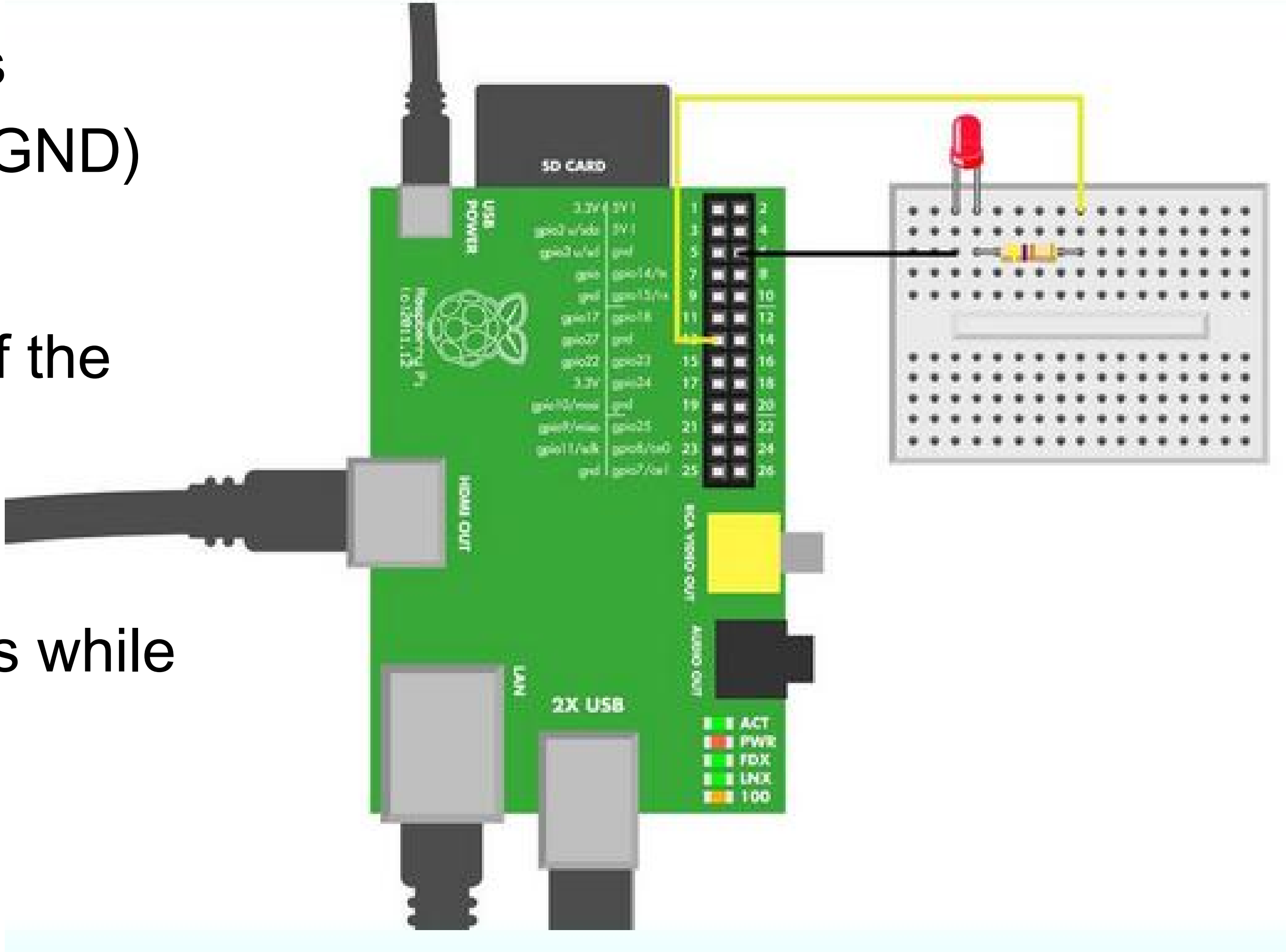
- GPIO pins are not protected against overcurrent (Figure below).
- Unlike Arduino, Raspberry Pi is not forgiving on user mistakes.
- Data pins can take only 3.3 V. Connecting a +5 V pin to a data pin can easily break your Raspberry Pi, or at a minimum, render that pin unusable.
- Double-check anything that you build on a breadboard, and be very careful where you place any test probes if you use a multimeter with the GPIO pins.

Building the Circuit

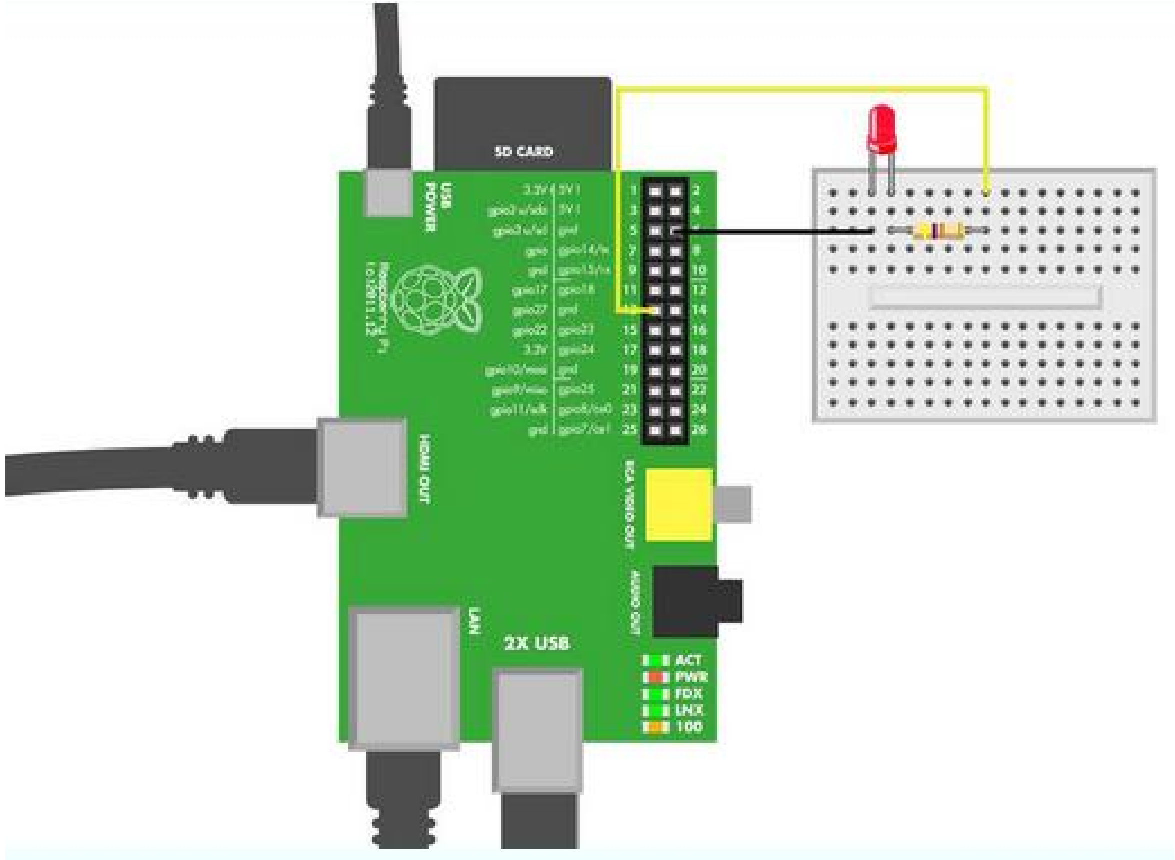
- The circuit is simply an LED with a current limiting resistor, connected in series between GPIO pin 27 and ground (GND) (Figure right).

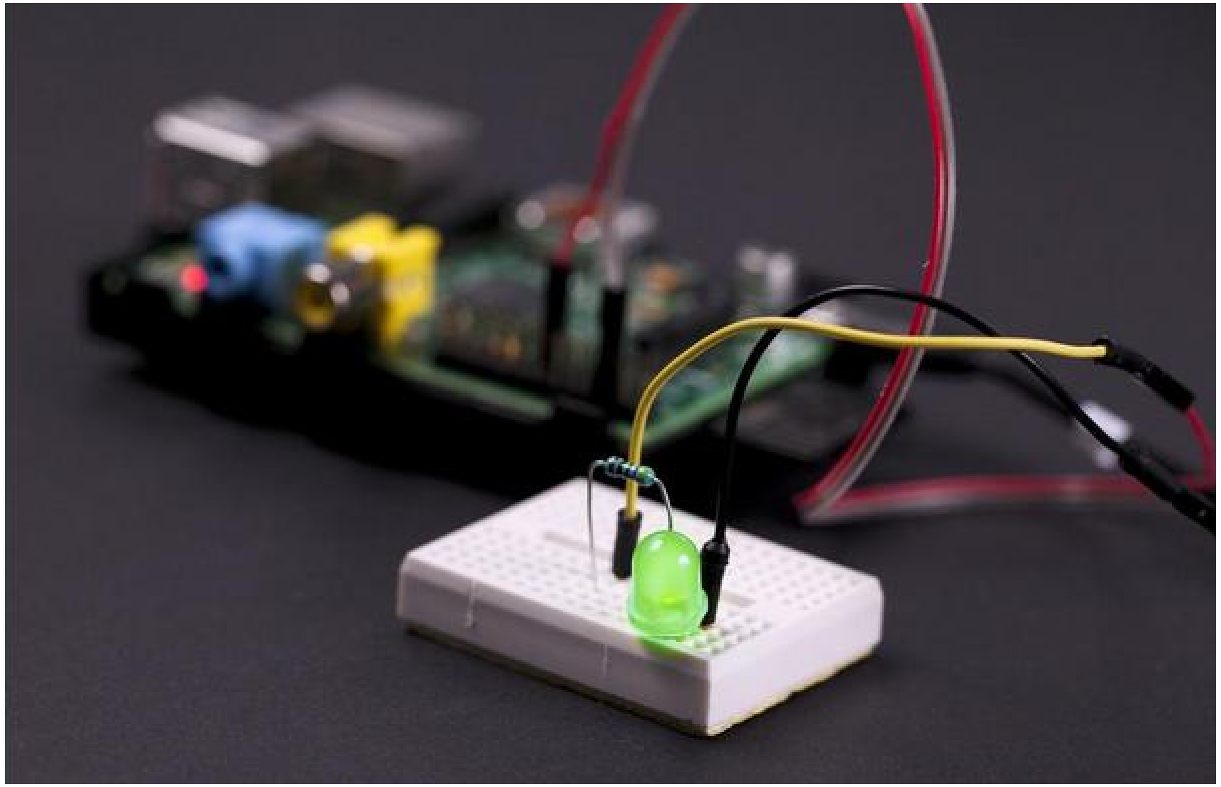
- Connect the short (negative) lead of the LED to the black wire, and the long (positive) lead to the resistor.

- You should make these connections while the Raspberry Pi is shut down and unplugged.



Building the Circuit



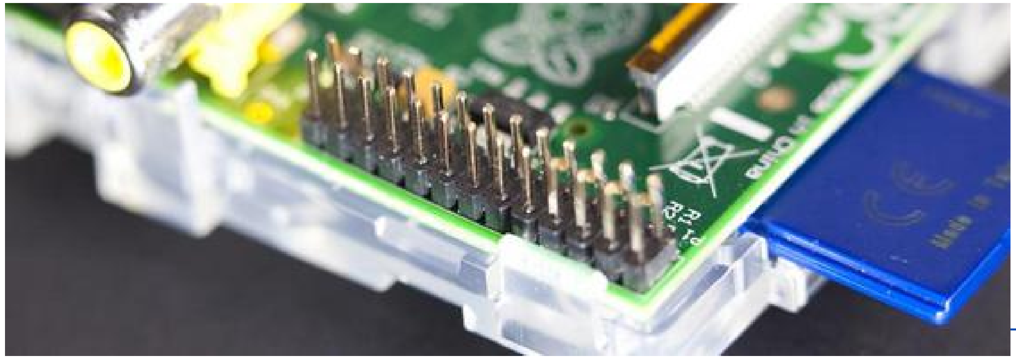


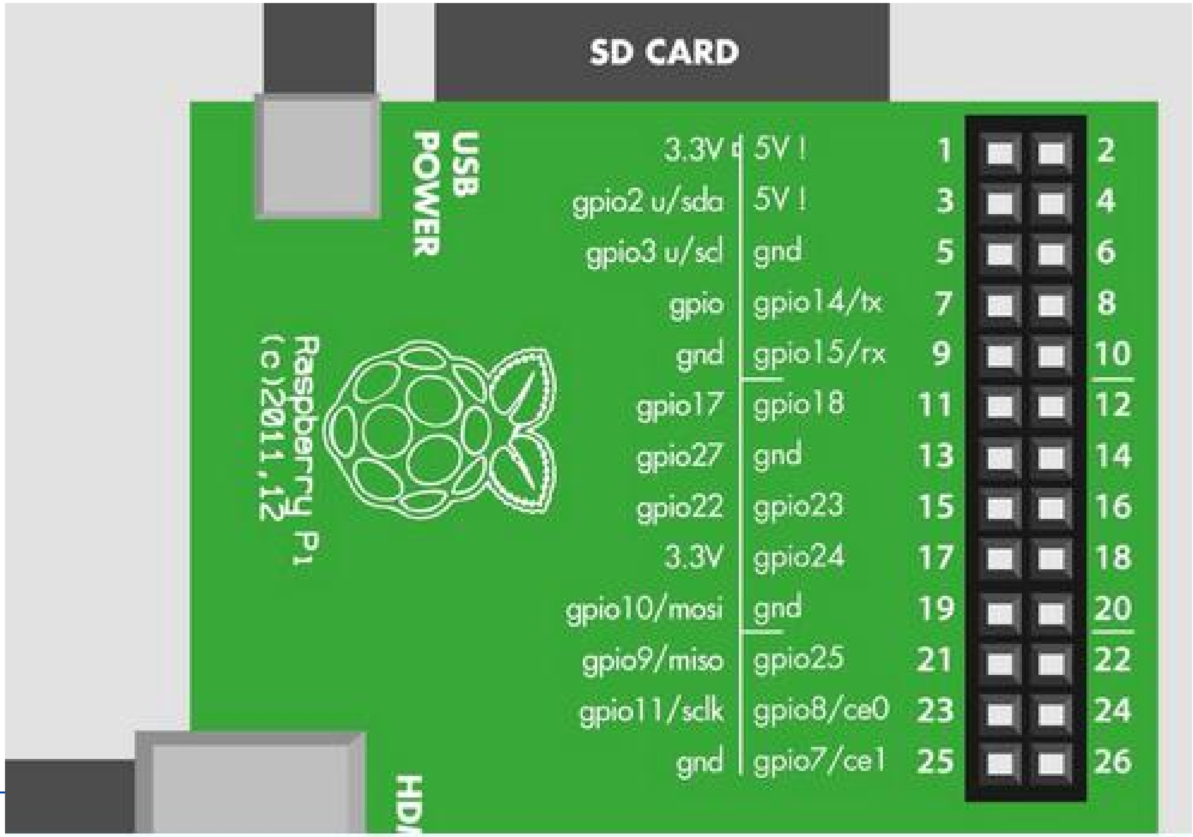
- ➢ Build the circuit on the breadboard (Figure top). Double-check all connections to avoid breaking your Raspberry Pi. After you're sure you've connected the wires correctly, you can power up your Pi.

Two Numbering Systems: Purpose and Location

- To learn about how the Raspberry Pi's pins are numbered.
- Each GPIO pin has two numbers: purpose and physical location.
- To find the correct pin in the GPIO header, you should learn to convert between the two numbering systems.

GPIO pin (BCM, used in code)	Physical location (Board)
GPIO 27	13
GND	6




SD CARD
USB POWER
Raspberry Pi
(c)2011,12
HDMI
3.3V | 5V ! | 1 | 2
gpio2 u/sda | 5V ! | 3 | 4
gpio3 u/scl | gnd | 5 | 6
gpio | gpio14/tx | 7 | 8
gnd | gpio15/rx | 9 | 10
gpio17 | gpio18 | 11 | 12
gpio27 | gnd | 13 | 14
gpio22 | gpio23 | 15 | 16
3.3V | gpio24 | 17 | 18
gpio10/mosi | gnd | 19 | 20
gpio9/miso | gpio25 | 21 | 22
gpio11/sclk | gpio8/ce0 | 23 | 24
gnd | gpio7/ce1 | 25 | 26

Controlling GPIO Pins from the CLI

- Let’s see how to use the command-line interface to control the GPIO pins we just connected.
- First you’ll try it out as root, then advance to using it without needing to invoke sudo privileges each time.
- Text files control everything in Linux. The kernel GPIO driver (a piece of software that controls how Linux talks to GPIO pins) makes the GPIO pins available to you through the virtual /sys/ file system.
- To control GPIO, you simply edit or otherwise make changes to these text files.

Controlling GPIO Pins from the CLI

- You don’t need a graphical user interface at this point, so we can do everything from the LXTerminal command-line interface. Double-click its icon to launch it.
- To turn on the pin
  - □ you first export it, configure it for “out” mode,
  - □ and write the number “1” to it.
- All this is done by editing text files.

Controlling GPIO Pins from the CLI

- How to display text

```
$ echo "Hello BotBook"

$ echo "Hello BotBook" > foo.txt
```

- You can see what’s inside that file with the cat command:

```
$ cat foo.txt

Hello BotBook
```

use sudo

- We'll use sudo the first time through lighting an LED.
- Does it feel wrong to use root for non-administrative tasks?
  - □ If not, it should; if you mistype a command with sudo, you can potentially render your operating system unusable and will need to reinstall it.
- Don't worry, you'll use sudo just to initially try it. Later, you'll fix Linux's file permissions and interact with the files that control GPIOs as a normal user.

use sudo

- Type ***sudo -i*** to get a root shell.
- Use root shell only as long as required by this task, and type exit when you're done.
- You'll notice that your prompt changes to a hash mark, **#**.
- Be careful what you type as root, as mistakes can break your operating system.

Light Up the LED

- This creates the new virtual file you’ll use to blink the LED.

```
# echo "27">/sys/class/gpio/export
```

- Next, set pin 27 to out mode, so that you can turn it on and off.

```
# echo "out" > /sys/class/gpio/gpio27/direction
```

- Now turn on the pin:

```
# echo "1" > /sys/class/gpio/gpio27/value
```

- Your LED should light up now. Once you have enjoyed the light for a while, turn it off:

```
# echo "0" > /sys/class/gpio/gpio27/value
```

C语言编写的GPIO控制例程，实现LED的每隔一秒闪烁一次

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
int main(void)
{       FILE *p=NULL;
        int i=0;
        p = fopen("/sys/class/gpio/export","w");
        fprintf(p,"%d",38);
        fclose(p);
        p = fopen("/sys/class/gpio/gpio38/direction","w");
        fprintf(p,"out");
        fclose(p);
```

C语言编写的GPIO控制例程，实现LED的每隔一秒闪烁一次

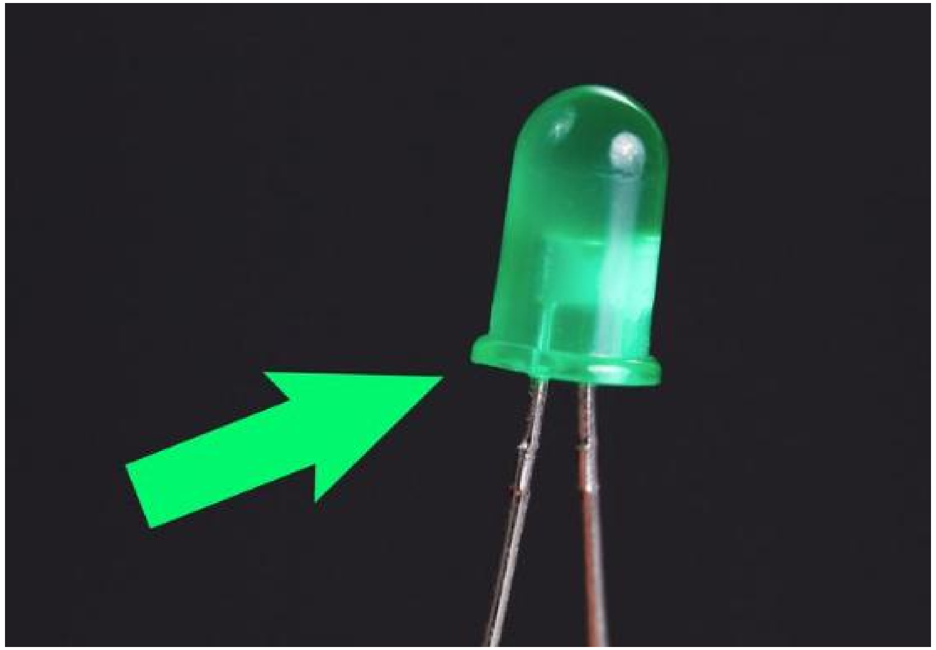
```
for(i=0;i<100;i++)
    {      p = fopen("/sys/class/gpio/gpio38/value","w");
           fprintf(p,"%d",1);
           sleep(1);
           fclose(p);
           p = fopen("/sys/class/gpio/gpio38/value","w");
           fprintf(p,"%d",0);
           sleep(1);
           fclose(p);
    }
    p = fopen("/sys/class/gpio/unexport","w");
    fprintf(p,"%d",38);
    fclose(p);     return 0;
}
```

Troubleshooting

- Lacking a 470 Ohm resistor?
  - Use any resistor in the hundreds of ohms range to take only a minor risk with your board. The resistor is just used for limiting current through the LED, to avoid frying the LED or the pins on the Raspberry Pi. If you don’t mind the LED being a bit too dim or slightly overworked, you can use any resistor value between 100 Ohm and just under 1k Ohm. (Such resistors will have a brown third stripe;)

Troubleshooting

- My LED does not light
  - Did you insert it the right way? LEDs have polarity, and if you insert them backward, they won't light up.
  - The long positive leg of the LED goes to GPIO 27 (through the resistor).
  - The negative side of the LED has a flattened area in the plastic body of the lamp, and a shorter lead than the positive lead.



GPIO Without Root

- Avoiding root privileges will make the system more secure and more stable.
- In modern versions of Linux, devices attached to your system are controlled by udev.
- Udev is a rule-based system that can run scripts when devices are plugged in.
- If you have developed apps for Android under Linux, you may have created a udev rule to modify permissions when your cell phone is plugged into the computer.
- If you have developed with Arduino on Linux, you have probably added yourself to the dialout group to get access to serial over USB.

GPIO Without Root

```
$ sudoedit /etc/udev/rules.d/88-gpio-without-root.rules
```

Example 1-1. 88-gpio-without-root.rules

```
SUBSYSTEM=="gpio", RUN+="/bin/chown -R root.dialout /sys/class/gpio/“
SUBSYSTEM=="gpio", RUN+="/bin/chown -R root.dialout /sys/devices/virtual/gpio/"
SUBSYSTEM=="gpio", RUN+="/bin/chmod g+s /sys/class/gpio/“
SUBSYSTEM=="gpio", RUN+="/bin/chmod g+s /sys/devices/virtual/gpio/"
SUBSYSTEM=="gpio", RUN+="/bin/chmod -R ug+rw /sys/class/gpio/“
SUBSYSTEM=="gpio", RUN+="/bin/chmod -R ug+rw /sys/devices/virtual/gpio/"
```

GPIO Without Root

- Sets the owner of the two directories to be root, and the group to be dialout.

```
SUBSYSTEM=="gpio", RUN+="/bin/chown -R root.dialout /sys/class/gpio/“
SUBSYSTEM=="gpio", RUN+="/bin/chown -R root.dialout /sys/devices/virtual/gpio/"
```

- Sets the sticky bit flag on these two directories.

```
SUBSYSTEM=="gpio", RUN+="/bin/chmod g+s /sys/class/gpio/“
SUBSYSTEM=="gpio", RUN+="/bin/chmod g+s /sys/devices/virtual/gpio/"
```

- Configures the permissions on the directories to give members of the dialout group read and write permission.

```
SUBSYSTEM=="gpio", RUN+="/bin/chmod -R ug+rw /sys/class/gpio/“
SUBSYSTEM=="gpio", RUN+="/bin/chmod -R ug+rw /sys/devices/virtual/gpio/“
```

GPIO Without Root

- 创建好了新的规则文件之后，使用如下命令重启udev服务程序，触发并使用新的规则：
  - □ $ sudo service udev restart
  - □ $ sudo udevadm trigger --subsystem-match=gpip

GPIO Without Root

- 下一步，检查所有关系是否正确：
  - □ $ ls -lR /sys/class/gpio/
  - □ ls命令输出的信息会多次提及“dialout”用户组。
  - □ 参数 -l 表示累出文件的详细信息（包括拥有者、用户组、权限）
  - □ 参数 -R表示同时也列出所有子目录的详细信息。

GPIO Without Root

- 至此，我们做完了所有使用非root权限操作设备GPIO的准备，下面就尝试一下：
  - □ $ echo “27” > /sys/class/gpio/unexport #先取消导出，下一句就不报错
  - □ $ echo “27” > /sys/class/gpio/export #导出GPIO 27
  - □ $ echo “out” > /sys/class/gpio/gpio27/direction #设置该I/0口为输出
  - □ $ echo “1” > /sys/class/gpio/gpio27/value #设置该I/O口为高电平，灯亮
  - □ $ echo “0” > /sys/class/gpio/gpio27/value #设置该I/O口为低电平，灯灭
- 以上命令提示符是“$”，表示当前为普通用户
- 这样几乎所有的编程语言都能使用GPIO了。

GPIO in Python

- You can use GPIO from Python by just writing and reading files in /sys/.
- This is the same method you used earlier with the shell.

Hello Python

- As always, start with a “Hello World” to test your environment.
- Using a text editor, create the file:

```
$ nano hello.py
```

If you can’t save the file, you might have used the cd command to navigate to part of the file system where you don’t have permissions. Type `cd ~` to return to your home directory and try it again.

- The file needs only one line:

```
print "Hello world!“
```

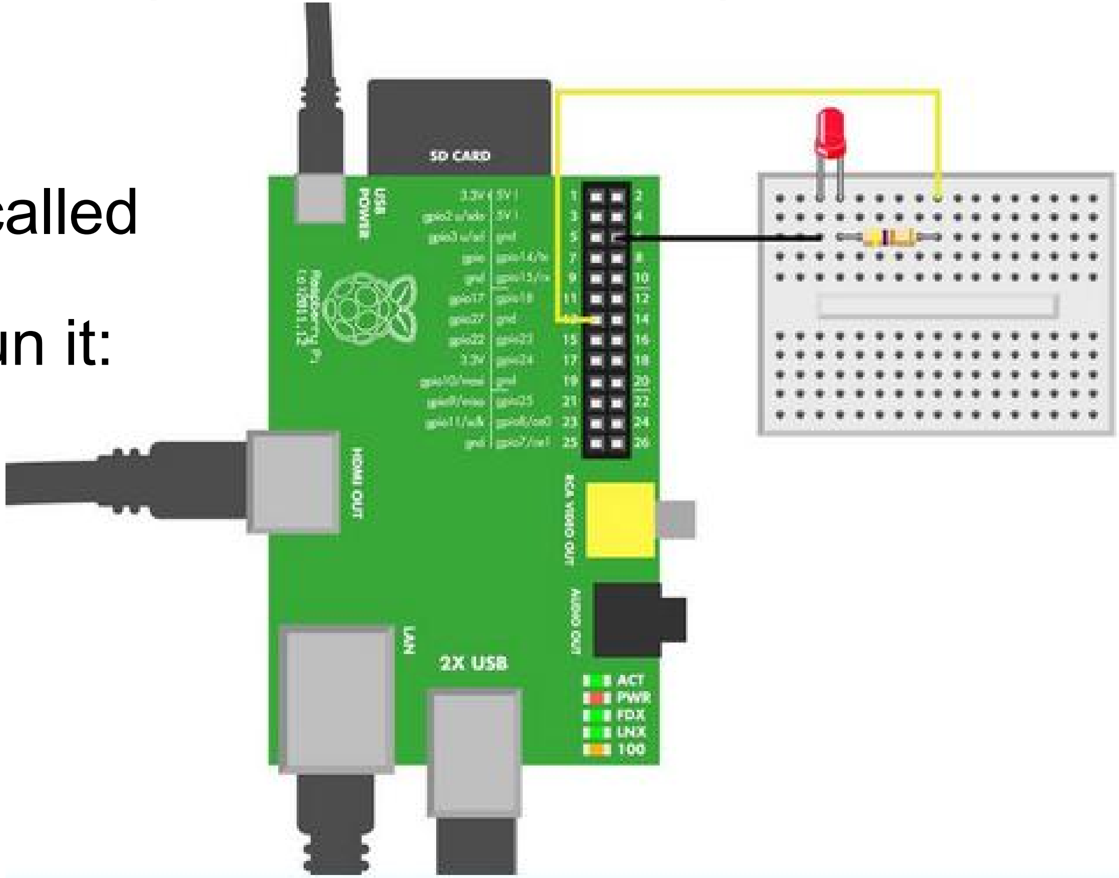
- Save the file (in nano, you save with Control-X).Now run your program:

```
$ python hello.py
Hello world!
```

Python GPIO

- ➢ Let’s blink the LED connected to GPIO pin 27.
- ➢ Wire up the LED if you don’t still have it set up from the earlier experiment (Figure 1-7).
- ➢ Save the code in Example 1-2 to a file called led_hello.py on your Raspberry Pi and run it:

```
$ python led_hello.py
Blinking LED on GPIO 27 once...
```



GPIO in Python

Example 1-2. led_hello.py

```
# led_hello.py - light a LED using Raspberry Pi GPIO
# (c) BotBook.com - Karvinen, Karvinen, Valtokari
import time
import os
def writeFile(filename, contents):
    with open(filename, 'w') as f:
        f.write(contents)
```

GPIO in Python

```
# main
print "Blinking LED on GPIO 27 once..."
if not os.path.isfile("/sys/class/gpio/gpio27/direction"):
    writeFile("/sys/class/gpio/export", "27")
time.sleep(0.1)
writeFile("/sys/class/gpio/gpio27/direction", "out")
writeFile("/sys/class/gpio/gpio27/value", "1")
time.sleep(2)   # seconds
writeFile("/sys/class/gpio/gpio27/value", "0")
```

Troubleshooting

- You receive a permission denied error
  - If you get an error like IOError: [Errno 13] Permission denied or IOError: [Errno 2] No such file, you can live dangerously and try running as root:
  - $ sudo python led_hello.py # for testing only
  - If it works correctly as root, good! You can now fix the permissions on the GPIO virtual files (GPIO Without Root). If you still have the same problem, reboot Raspberry Pi by shutting down, unplugging power, and plugging it back in. If all goes well, you can run it the correct way, as a normal user:
  - $ python led_hello.py

Troubleshooting

- LED does not light, but the program doesn’ t give any errors
  - Check the LED polarity and connections.
  - Check that you have connected the jumper wires to the correct pins on the GPIO header (Figure 1-9).
  - If that doesn’ t help, you can use a multimeter to verify that you used the correct resistor (e.g., 470 Ohm).
  - You can test the LED with a circuit that just has a battery, the resistor, and the LED in series. This will tell you whether the LED works.

Summary

- You have combined the power of Linux and electronics.
- The system administration techniques you have been practicing are leading you in the right direction. It′ s always good to use minimum (non-root) privileges whenever possible.
- You can apply your new system administration and GPIO skills with all the sensors in this book. Raspberry Pi usually shines with sensors that use more advanced protocols. This advantage is clear with I2C sensors such as Wii Nunchuk, MPU 6050 accelerometer-gyro, and the GY65 atmospheric pressure sensor.Welcome to embedded Linux!