

20140706 Installing a Raspberry PI Distro using Windows

Installing a rPI distro to a SD card has gotten easier over the years, and there are a few tools to make it even easier. rPI came up with a small distro that will install one of the full featured distros called NOOBS, and this is what they recommend doing, though they say it will take a lot of time to do it.

This is a more “advanced” way, and in the end a time saver. (This should work with most of the distros, NOOBS is a bit different, and has a slightly easier method, but it’s really a usable distro either)

The 1st thing we need to do is download the rPI distro that we want to use, I choose Raspbian which seems to be one of the more popular distros.

<http://www.raspberrypi.org/downloads/> If the 6 distros official supported are not to your liking, google “raspberry pi distro” and you’ll get a list of unsupported distros, as said most of these distros will install the same way.

The downloads can be pretty big, from about 800mb to 2gb and can take a little while on a slow connection. After you get the distro you want, there are a couple of tools that will make life so much easier.

Win32 Disk Imager is a tool used for writing images to USB sticks or SD/CF cards and works amazing. <http://sourceforge.net/projects/win32diskimager/>

IF you choose NOOBS you may need the SD Formatter, what this does is remove some of the “junk” that windows puts on the card, and give you full access to the what is on the card. I’m not going to be using NOOBS but thought it worth mentioning just incase you get a failed install this tool can reformat your SD the “better” way, and might correct it.

https://www.sdcard.org/downloads/formatter_4/ There is also a SD tool for Mac OSX, gparted is the preferred way in Linux. Documentation on the format tool can be found here:

<http://www.raspberrypi.org/documentation/installation/noobs.md>

You’ll need to get yourself a SD card, 4GB or more would be recommended, but in a lot of cases a 2gb card would work, it just wouldn’t leave you with much room for anything else.

Next we need to extract the files from the zips, this is easy to do just right click on the download, and find “extract here”, a wizard will come up, click next a few times, and it is done you should have a new folder open with the extracted files. In the Win32 Disk Image folder double click Win32DiskImager. (This does not need to be installed)

<http://www.raspberrypi.org/documentation/installation/installing-images/windows.md>

SDFormatter does need to be installed, double click on the setup and follow the prompts (You only need to do this if you need to reformat your SD card or if you are using NOOBS)

Click on the folder button in Win32DiskImager, and find the Raspbian image you just extracted, make sure that the drive letter next to the folder is set for your SD card (Win32DiskImager will only show SD/CF/USB drive letters), Click on write, and wait, it might take a little while.

Once that is done, you should have a bootable SD card.

Now we have to setup the rPI. Adafruit has a wonderful 13 part tutorial on setup and using the rPI, step 2 can be found here, it’s a little out of date, but for the most part it’s correct.

<https://learn.adafruit.com/adafruits-raspberry-pi-lesson-2-first-time-configuration/overview>

pretty much the steps you will need to get started are Expand Filesystem - this will make the whole SD useable, When you write an image to a SD/USB drive it takes up as much as space as what the image is, in otherwords if you have a 1gb image, and use a 4gb card you will loose 3gbs. This tool brings the lost 3gbs back.

The next step will be to set a new password, I'm not sure what the default password is, so set one up.

lastly you'll want to change how it boots or rather what it boots too - With Raspbian you have a choice for a desktop gui (LXDE desktop), a console (default) display, or to boot into scratch. Once at least those three things are done, you can highlight finished, and reboot. You might also want to setup your timezone, and locale - thou strickly speaking this is not necessary to get into the system. (These settings can be found under Internationalisation Options)

If you need to make changes again, or what to explore some of the advanced things, open the LX Terminal and type "sudo raspi-config"

NOTE: IF you should chose to boot into Scratch and can't figure out how to get out of it, hold the CTRL-ALT-2 this will bring you to a terminal prompt, the user name is pi (unless you've changed it) and the password is whatever you set it to. From here you can run the configuration again, and change it. (CTRL-ALT-# is native to almost all linux distros, this one is no different luckily)

The current desktop comes with a few IDEs already pre-installed, so no need to install more (for now) Python IDLE 3, Scratch, though some other librarrys probably will need to be installed for most of these to be useful on the rPI.

It also has a WolframAlpha program installed by default:

Adafruit's 3rd tutorial goes into setup of a wifi network, not going into that too much, The biggest thing is make sure your USB adaptor doesn't use a lot of power, and that it's linux compatable, a lot of the cheapies that have RALink chips support linux with no problems. Adafruit recommends a RTL8192cu chipset.

You might want to know the IP address of the rPI thou: open the LXTerminal, and type sudo ifconfig, wired connections will be labeled eth0, wireless wlan0. your IP address will be inet addr.

Total optional step, but probably a good one to take at this point is to update the software on the rPI, open the LXterminal, and type.

"sudo apt-get update"

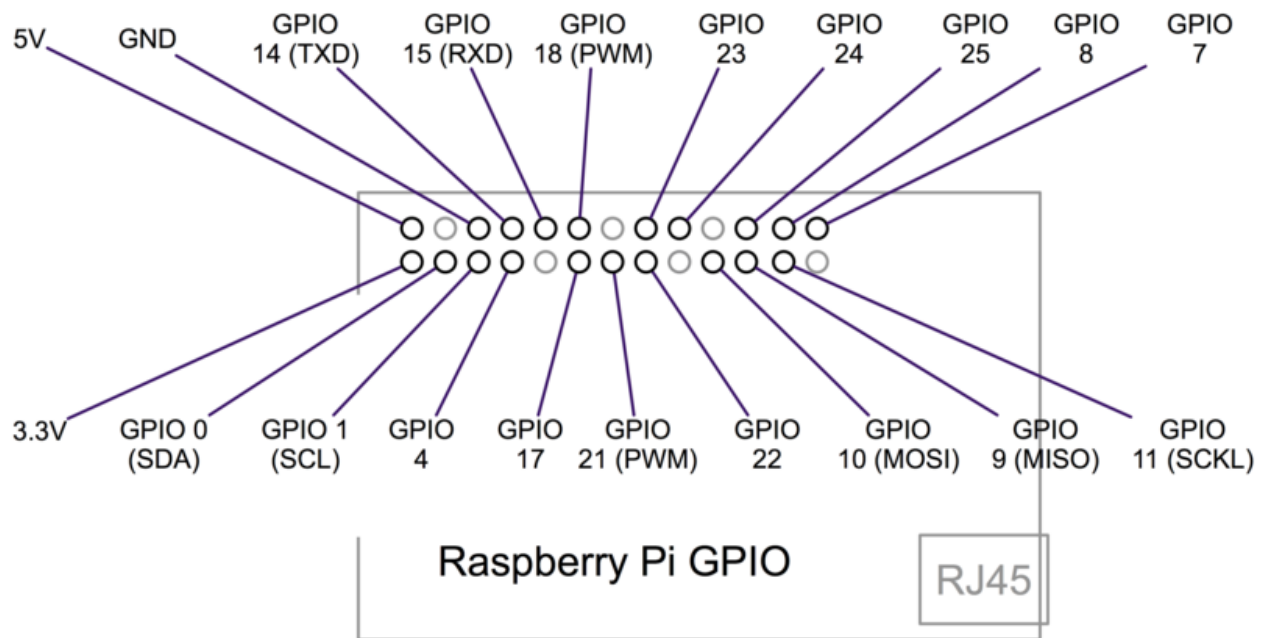
when it's done checking for updates, type

"sudo apt-get upgrade" and answer "Y" to the question, this will upgrade the software on the rPI and might take a little while to complete, depending on how many updates there are.

That is setting up your rPI, now to do something interesting (well sort of) with it.

Blinking an LED!

Adafruit Lesson 4: GPIO <https://learn.adafruit.com/adafruit-raspberry-pi-lesson-4-gpio-setup>



All of the pins on the rPI are 3.3v logic, so don't hook up anything that uses 5 volt logic unless you have a level shifter. But you can see the rPI has uarts, pwm, i2c, and SPI.

Again it seems Adafruits tutorial is a bit out of date as "GIT" is already installed on my Raspbian distro, but just in case:

To use Adafruits tutorial at this point you need to install "git", open LXTerminal, type "sudo apt-get update" this makes sure all the repositories are reachable, and checks to make sure there is no one software available.

After it's done: type "sudo apt-get install git" this tell the rPI to install a new piece of software called GIT, the apt-get package manager knows where in the repository to find this fetches it and installs it for you.

(once again the tutorial is incorrect, notice the addition of "s" on the http statement)

Now type:

```
git clone https://github.com/adafruit/Adafruit-Raspberry-Pi-Python-Code.git
```

then:

```
cd Adafruit-Raspberry-Pi-Python-Code
```

followed by:

```
ls
```

What this does is creates a working folder that has the code for the rest of the GPIO tutorials in it.

And what I want to do the github code will not be needed. I simply want to blink an LED on and off, but we will need the Python Development toolkit and the rPI.GPIO library

1st install the development toolkit:

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

next install the GPIO python library:
sudo apt-get install python-rpi.gpio

The tutorial goes into detail on how to setup I2C which is out of the scope of what I am doing. So I am going off and doing my own thing, the tutorial while they do contain some errors are very good, and easy to follow, and I would recommend them to anyone who wants to know more about the rPI, there are 13 tutorials total, and they start simple and go big.

<https://learn.adafruit.com/series/learn-raspberry-pi>

Maybe not as simple as what I am doing though: Flashing an LED using Python and the rPI

my python code looks something like this:

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GREEN_LED = 8
RED_LED = 7
GPIO.setmode(GREEN_LED, GPIO.OUT)
GPIO.setmode(RED_LED, GPIO.OUT)

While True:
    GPIO.setup(GREEN_LED, True)
    GPIO.setup(RED_LED, False)
    time.sleep(1)
    GPIO.setup(GREEN_LED, False)
    GPIO.setup(RED_LED, True)
    time.sleep(1)
```

General knowledge of linux and python is helpful, and google is your friend.

What is Scratch: *Scratch* is a free programming language and online community where you can create your own interactive stories, games, and animations

<http://scratch.mit.edu/> Want to learn Scratch? <http://learnscratch.org/>

What is Wolfram Alpha it is more than a search engine, it gives access to the world's facts and data and calculates answers across a range of topics.

<http://www.wolframalpha.com/>