# Setting up the Pi – General Use

Greetings! This is the first of three tutorials designed to get you started in the Raspberry Pi world. Topics include general setup, setting up the Pi as a webserver, and using the GPIO.

In this walkthrough, you will take your Pi from out-of-the-box to ready-and-raring – we cover setting up the SD card, internet, and a static IP address.

If you encounter problems at any point, attempt troubleshooting yourself and then ask in the Facebook group. I used these approaches recently, so they should all be up-to-date, but there’s always room for something to go wrong!

**Materials**

1. 8GB MicroSD Card (or higher)
2. Raspberry Pi
3. Monitor
4. HDMI cable
5. USB Keyboard
6. A personal computer – your laptop or desktop
7. SD Card Reader
8. 1.5A microUSB Power Supply
9. Ethernet cable

## Step 1: Prepare the SD Card

The SD card holds the operating system – when looking for tutorials, you may often find authors using the terms “Linux box” or “Linux machine,” because rPi is essentially a Linux computer. I am going to walk you through installing Raspbian, the official Raspberry Pi OS, but on the download page you’ll find several operating systems to choose from.

**Materials**

1. 8GB MicroSD Card (or higher)
2. A personal computer – your laptop or desktop
3. SD Card Reader

Note: This probably comes as part of your computer. If you don’t have one built in, a USB adapter like in Figure 1 will suffice. My computer has an SD slot, so I am using an adapter as shown in Figure 2. Simply put: Do whatever necessary to get the MicroSD into your computer)

|  |  |
| --- | --- |
| Figure 1 | Figure |

**Process**

1. **Get the Software**

Download Raspbian as a zip file from <https://www.raspberrypi.org/downloads/raspbian/>. Consider going for a walk, because this may take a few minutes. When it is finished, extract the zip to get a disk image.

1. **Format the SD Card**

Insert the SD card into your personal computer using the SD card reader. For Mac and Windows computers, download the SDFormatter from the SD Association at <https://www.sdcard.org>, and use it with the settings specific in the appropriate section below. If you are a Linux user, see the Linux section (obviously). Now, I’ll note that formatting isn’t strictly necessary since we are flashing an image – both erase the previous data entirely. I like to format it anyway because it gives me a sense of having a “clean slate.” Formatting *is* necessary if you download NOOBs, which I describe in the next section.

* 1. **Mac**

Use SDFormatter and choose “Overwrite Format.” As always when dealing disks, be sure that you have the right one or unfortunate deletion/corruption will ruin your day. If it says something about being used in another process, it probably didn’t like that the card was mounted. In the process of complaining it unmounts the disk, so just press format again and all should progress swimmingly.

* 1. **Windows**

Use SDFormatter and set "FORMAT SIZE ADJUSTMENT" to ON. As always when dealing disks, be sure that you have selected the correct drive (a.k.a. the one that your SD card is inserted into; you can see it in the “Computer” tab of the file explorer) or unfortunate deletion/corruption will ruin your day. Full disclosure, I’ve never done this process on a Windows, so I can’t provide support in case of errors.

* 1. **Linux**  
     Did you know that using and building on previous achievements is a trait unique to humans? While most people use computers without knowing how to make their own, other tool-using species have to learn how to make their own insect-sticker by watching others – no one will teach them how or hand theirs down. So, in the spirit of *sapiens sapiens* pride, I guide you to <https://learn.sparkfun.com/tutorials/sd-cards-and-writing-images>. Scroll to the Linux section of “Formatting your card.”   
       
     This tutorial has write-ups for Mac and Windows too, but uses built-in tools that the rPi community has encountered problems with (at least in the case of Macs), so I recommend using SDFormatter.

1. **Put Raspbian on the SD Card**

To flash the Raspbian image onto the SD card, follow “M-Short’s” instructions at <https://learn.sparkfun.com/tutorials/sd-cards-and-writing-images#downloading-and-installing-the-image>. He has guidelines for all three systems. Go for a much longer walk than earlier. This part takes a long time (a few hours for me; it was done when I got back from dinner).

#### Sidebar – NOOBS

An alternative path to obtaining Raspbian is to put NOOBS on the SD card instead of installing Raspbian directly. Beginners often consider NOOBS easier because it doesn’t require flashing the system image. It’s more of a drag and drop sort of deal.

If you want to go the NOOBS route, here you go:

1. Download the most recent version of NOOBS (which is probably no longer 1.9.2 like in Figure 3) from raspberrypi.org, and extract the folder.

Figure 1

1. Format the SD card like in the instructions above.
2. Copy all of the files into your SD card – don’t put the folder that is extracted, go one directory further and copy the files inside the folder, as shown in Figure 3.
3. Put the SD card into your Pi, plug in the USB keyboard, and hook up the monitor using the HDMI cable. Power the Pi, and it should boot into the OS selection panel.
4. Select your desired OS (probably Raspbian, but if you know what you’re doing then be adventurous and try another), and click “Install.”
5. Take a break. NOOBS will take care of the rest; come back when the Pi reboots into Raspbian. Voila!

NOOBS does some interesting stuff to streamline the process for users. One is that it partitions the SD card – that is, breaks the 8GB into smaller pieces – which then appear when you put the card back into your computer for any reason. This is why you reformat it with complete overwrite: to put it back in one piece. To learn more about this partitioning, read this article:

<https://github.com/raspberrypi/noobs/wiki/NOOBS-partitioning-explained>.

## Step 2: Configure Raspbian

In this step, you will update your Pi, get on the internet, and configure it to have a static IP address.

**Materials**

1. 8GB MicroSD Card (or higher) – all nice and formatted
2. Raspberry Pi
3. Monitor
4. HDMI cable
5. USB Keyboard
6. 1.5A microUSB Power Supply
7. Ethernet cable

**Process**

1. **Get plugged in**

Insert the newly-flashed SD card into the Pi. Attach the USB keyboard and plug in the monitor with the HDMI cable. Plug in the power supply, and wait to boot into the GUI. Plug in the Ethernet cable with the other end attached to your router. If you don’t have access to your router, skip to the internet section and return here to configure.

1. **Update**

Find the terminal in the toolbar along the top of the screen. We are going to use the keyboard exclusively, like real nerds (mice are for chumps), so hit the windows key in the lower left corner of your keyboard to open the Menu. Use arrow keys to navigate to Accessories, and select Terminal. When the terminal opens, run the following four commands. These download files, which is why we need an Ethernet connection – we will set up Wi-Fi next. You will have to enter ‘y’ now and then to give it permission to continue.

|  |
| --- |
| Terminal  Sudo apt-get update  Sudo apt-get upgrade  Sudo apt-get dist-upgrade  Sudo rpi-update |

1. **Raspi-Config**

Run sudo raspi-config in the terminal, and this cute little display comes up. You can do a lot here – we are merely going to enable a SSH. Go to **Advanced Options > SSH**, and click enter on ‘yes’. If you want you can change the password and set the Pi’s name, but be careful changing stuff that you don’t understand. When you need these capabilities for a particular project, learn about and turn them on then. Leave it alone for now. :)

### Internet

Setting up internet is very easy if you have a mouse – as you would on any other computer, go to the top right corner. Click on the picture of two computers, and a list of available networks should appear.

Now, without a mouse, you have to mess with a file. But it’s still easy. Execute sudo nano /etc/wpa\_supplicant/wpa\_supplicant.conf to open a file, and scroll to the bottom. Add these lines:

|  |
| --- |
| /etc/wpa\_supplicant/wpa\_supplicant.conf  network={  ssid=”[network name]”  psk=”[password]”  } |

If you’re network doesn’t have a password, replace the psk line with “key\_mgmt=NONE”. If you are on an enterprise network, such as school or work, add an identity line: identity=”[username]” before psk. Your username is the part of your school email before the @ symbol.

Reboot wlan0 by executing sudo ifdown wlan0, then sudo ifup wlan0. Finally, execute ping [www.google.com](http://www.google.com) to see if you have connection (if it’s unsuccessful, try again after a minute, sometimes the Pi takes a little while to connect).

Now run ifconfig in the terminal, and try not to be intimidated by the deluge of information. Find the section about wlan0, and within it, something that looks like inet addr:192.168.#.##. This is the IP address that your router provided you when you came online. In the next section, we will make it static.

### Static IP address + SSH

Why do you need a static IP address? Well, this keyboard-monitor access is really only necessary for configuration. Most of the time you will ssh – a way of securely accessing devices remotely by executing ssh [account address] in a terminal – into your Pi and use it from your personal computer’s terminal, which is called “headless.” If your device is assigned an IP address each time you boot ssh-ing is impossible, so we assign a static address that will remain, well, static across reboots.

**Useful Note!** In Raspberry Pi 3, there is an alternative. Setting a static IP is risky on networks that you don’t fully control because you could connect with your assigned IP while that IP is being used by someone else. Instead, on these networks use your hostname. Go into raspi-config like earlier, and find the option to change your hostname. It is a mask for your IP address which stays static even when your IP doesn’t. Set it to something unique, and ssh in with ssh [pi@[hostname].local](mailto:pi@[hostname].local). This is the method I encourage for enterprise networks like UCSP-PROTECTED because the danger of duplicate IPs is high. However, the following instructions for setting a static IP is useful for you home network.

Open the file /etc/dhcpcd.conf, and add/edit these lines.

|  |
| --- |
| /etc/dhcpcd.conf  SSID [network name]  Inform 192.168.#.## |

The restriction is that the 3rd section, the single hash, has to be the same as whatever your router provides you. Usually it’s 1. For example, my router assigned me 192.168.1.9 and I set my static address to 192.168.1.70.

Reboot by running sudo shutdown –r now. When your device comes back online, run ifconfig again. Now the inet address should be what you set in the file!

This should work for Ethernet connections too. That is, if you connect to your router with an Ethernet cable, it will successfully connect using your static IP. If you want to have a different IP address for Ethernet, add these lines.

|  |
| --- |
| /etc/dhcpcd.conf  interface eth0  static ip\_address=192.168.#.## |

Reboot, and voila! Test it by plugging in an Ethernet connection before booting, viewing the IP address in eth0 with ifconfig, and then unplugging and running ifconfig again. You should at first see your set IP address under eth0 with none set for wlan0, and then the one set for wlan0 appear when you unplug. Ping [www.google.com](http://www.google.com) at each step to make sure you don’t lose connectivity.

At this point, we no longer need auxiliary parts – we can connect with ssh, and only need to carry around the pi itself and a power supply for it! Go ahead and pack up the hdmi cable, monitor, and keyboard.

## Step 3: Hello World

**Materials**

1. 8GB MicroSD Card (or higher) – all nice and formatted
2. Raspberry Pi
3. 1.5A microUSB Power Supply

Now go to your personal computer, ensure that it is connected to the same network that your Pi is on, and open a terminal (on Windows you have to use PuTTY. Full disclosure, I have never had to do that, so I can’t confirm which tutorials are correct – poke around the internet until you find out how to ssh on a Windows computer). From your computer, execute ssh [pi@192.168.#.##](mailto:pi@192.168.#.), where the IP address is the one you set in the previous section. It will ask for permission to connect (grant it, duh), and the password. Unless you changed the password in raspi-config earlier, it is “raspberry”.

After a moment, it will log you into the Pi! You may now utilize UNIX commands as you normally do. For example, create a simple program to greet your Pi – you two are ready to go on great adventures together.

|  |
| --- |
| helloworld.cpp  **#include** **<iostream>**  using namespace std;  int main() {  cout << **"Hello World\n"**;  return 0;  } |

Compile by executing g++ helloworld.cpp –o hello, and run it with ./hello.



## Conclusion

Next, visit “Setting Up the Pi – LAMP” to configure your pi as a webserver. This tutorial will give you an overview of what it means to be a webserver, steps to becoming one, and an example site.

## Resources

Wi-Fi

<http://www.makeuseof.com/tag/setup-wi-fi-bluetooth-raspberry-pi-3/>

NOOBS partitioning

<https://github.com/raspberrypi/noobs/wiki/NOOBS-partitioning-explained>

SD cards and writing images

<https://learn.sparkfun.com/tutorials/sd-cards-and-writing-images>