

Note: The serial port still has a few bugs, so it's not recommended to use for configuration. Raspbian can be hacked to get it to work but it's not covered by this guide. This Pi forum post talks about the serial port in more depth.

As a desktop, these materials are required:

- Pi 3 Starter kit -or- Pi 3/2 Accessory Kit and your own Pi
- USB Mouse
- USB Keyboard
- HDMI monitor/TV/adapted VGA

After configuration, 'headless' operation over Telnet/SSH requires:

- Pi 3 Starter kit -or- Pi 3/2 Accessory Kit and your own Pi
- 2nd computer connected via internet
- Telnet/ssh terminal software

You'll also need an internet connection to get resources! This link can be wired or wifi and must be available for the Pi. For wireless connections, check out our WiFi Adapter.

Suggested Reading and Viewing

You may want to check out the following tutorials and videos before continuing.

- Serial Terminal Basics
- Raspberry gPlo – Also linked-in later
- Pi Wedge Hookup Guide
- Getting Started With The Raspberry Pi – **video series**
 - Part 1
 - Part 2
 - Part 3

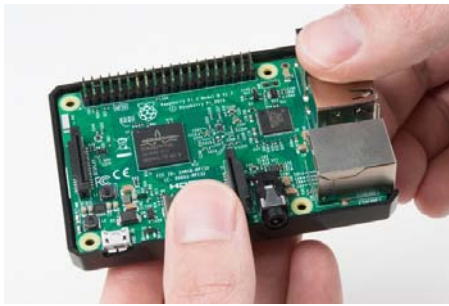
Assembly

The Pi is straight-forward and easy to put together, but, in the event that something doesn't seem right, this section will give you an idea of what it is supposed to look like.



Unbox and gather these components before beginning the assembly

1. Snap the Pi into the base of the 'tin', then snap the top into place.



*Fit the Pi into the base of the tin. Make sure the Pi is fully seated.
Check that the PCB is evenly recessed about the perimeter.*



Click the two halves together

2. Add the SD card



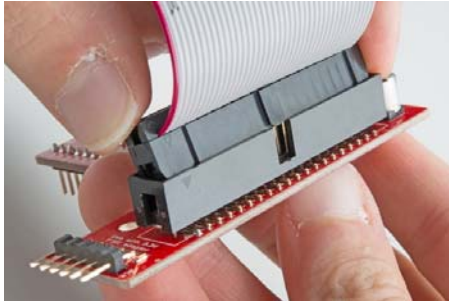
Installing an SD card – make sure the microSD card is flush with the side of the case when inserted properly. The Pi 3 microSD slot doesn't have a spring as the previous pis did, so if it's flush with the label outward, it is seated correctly.

3. Connect the ribbon cable to the Pi – notice that the pin 1 marking is very subtle. Orient the red stripe on the cable towards the SD card. Alternately, pin 1 can be identified by finding the missing/beveled corner of the header's silkscreen on the pi.



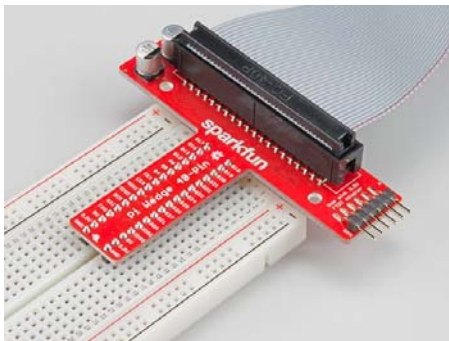
The pin 1 location and silkscreen is the same between the Pi B+ and Pi B. This image shows a partially inserted ribbon cable without the case in the way. The ribbon cable is oriented with the red "pin 1" marking pointing towards the SD card slot.

- Attach the ribbon cable to the wedge. Pin 1 is pointing towards the FTDI adapter.



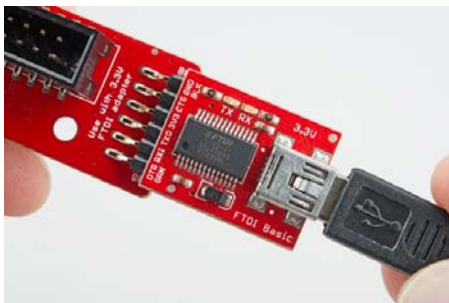
Socket the end of the ribbon cable into the Wedge. It is keyed, but each end of the cable is different. Make sure the ribbon extends away from the breadboard connection.

- Socket the Wedge into your breadboard



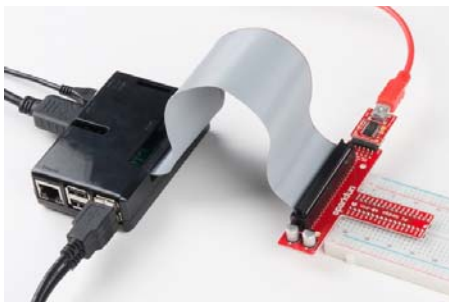
Wedge inserted in breadboard.

- Attach the FTDI connector matching "GRN" to "GRN" and "BLK" to "BLK" between the boards.



The FTDI serial adapter is connected matching GRN and BLK connections

- Attach desired consumer computer equipment.



The fully assembled kit. Additional to the kit, user supplied monitor, mouse, and keyboard are shown. This Pi is now a desktop computer.

Getting an OS

Note: At the time this was written, the kit comes with a 16GB card loaded with the Noobs OS installation image, version 1.9.0. This card should be ready to boot right out of the box.

Getting a New Image

If something didn't work, or the installation has been corrupted (messing around in the file system were you?), getting a new copy is easy.

- Obtain the Noobs OS from raspberrypi.org.
- Format the card to erase all the files.
- Unzip the contents of the Noobs zip file to the empty, formatted microSD card.

That's it! You're ready to go. For other imaging, check out this tutorial on sd cards and writing images. You can also check out this guide from the Raspberry Pi foundation.

The First Boot

Before you apply power for the first time, run through this pre-flight checklist.

- Is the microSD card installed and seated firmly?
- Is the HDMI attached and the monitor powered?
- Are the mouse and keyboard plugged in?
- Are the mouse and keyboard standard USB, not wireless? – Some wireless keyboards have trouble enumerating, so use one you trust.
- Is the whole setup secure on your desk and not liable to jump onto the floor at the first touch?

Ok, you're ready to apply the power to the Pi.

Power Adapter Requirement! Make sure the power being supplied is from a the included 5.24V, 2.4A power supply and not your USB connection. The USB most likely won't have enough current supplying capacity and will result in a brown-out of the Pi that can damage it, and it will likely mess up the files on your SD card.

First, you should see a color chart appear on your screen that indicates the Pi has power and is doing something but doesn't have software loaded yet.

Next, Noobs asks if the raspbian distro should be installed. Check the box to select it, and choose your language/keyboard layout here (can also be changed later).

Noobs will take a few minutes to manage the partitions and install the OS.

When it's done, it should report that the OS was installed successfully. Click OK, the Pi will reboot into a graphical interface.

The Noobs default configuration is to auto-log in as user `pi`, password `raspberry`.

Performing a Full System Upgrade

Once connected to the Internet (see the Configuring the Pi section), it can be a good idea update all the packages that are currently installed. Usually, new packages replace old ones that are faulty, but sometimes new packages have bugs of their own. If your current system is stable and all

the functions are good, maybe don't upgrade. If you're starting a fresh project though, get everything up-to-date *before* you start putting in the work.

Enter the following commands in the shell (process takes about 10 minutes).

- `sudo apt-get update` – go fetch the latest package information.
- `sudo apt-get upgrade` – And answer Y. Upgrades all the packages. This stage will take a while.
- `sudo shutdown -r now` – Reboot the machine.

Methods of Working with the Pi

Once the Pi is configured, there are a few methods of using it depending on if you want to use it like a desktop or manage it remotely.

This section covers using the Pi in the following ways

- Using HDMI-out – Operate your Pi like a desktop computer.
- Using the serial terminal – some functionality but buggy in the latest Raspbian release.
- Using SSH – Operate through a network linked to another computer.

Using HDMI-out

Noobs 1.8.0, with Raspbian, automatically logs in and starts a graphical user interface.

If configured to boot to shell, log in with `pi` as the user name and `raspberry` as the password. Then, enter `startx` to enter a graphical environment where you are presented with a desktop-type menu-driven operating system.

If you need to get back to text land, you can either

- log off through the Task Bar Menu – drops back to the shell, closing down the X window system
- open `xterm` from the Task Bar – opens a shell in a graphical window
- Use CTRL-ALT-1 through CTRL-ALT-8 – gives you a number of shells, with 7 being the graphic environment (if loaded).

Remember, shutdown with the menu item or enter `sudo shutdown -h now` from a shell, and wait for the system to halt before removing power.

Using the Serial Terminal without a Monitor

Connect the FTDI to the mini-usb cable and plug into a usb port on your computer.

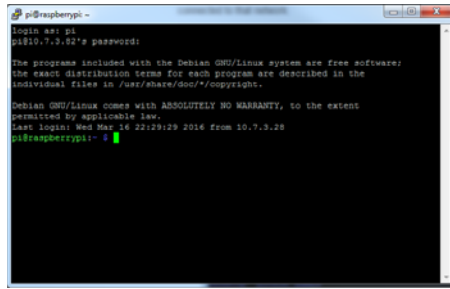
Set the serial settings to **72000** baud, 8 bit, no parity, 1 stop and no flow control. This is an odd rate that is 1.6 times slower than before. Also, CPU frequency effects the symbol rate so be careful if using this interface, and plan for administering by a different method.

Using SSH

A good way to operate a Pi is to attach it to the local network somewhere, then manage it from another computer connected to that same network.

Note: This method relies on a local Internet connection. Work through the configuration process with the monitor/mouse/keyboard or serial terminal (described below), then come back here.

To do this, download PuTTY or a similar SSH terminal for your system and connect to the Internet port used by the pi. The TTY interface gives you a serial-like interface but with colors that make it a little nicer to use.



A PuTTY terminal looks nicer than serial and can be used over networks

To obtain your IP address, get to a terminal, and use the command `ifconfig`. Alternately, hover the mouse pointer over the network icon on the task bar.

With Ethernet and wireless attached, `ifconfig` returns something like the following:

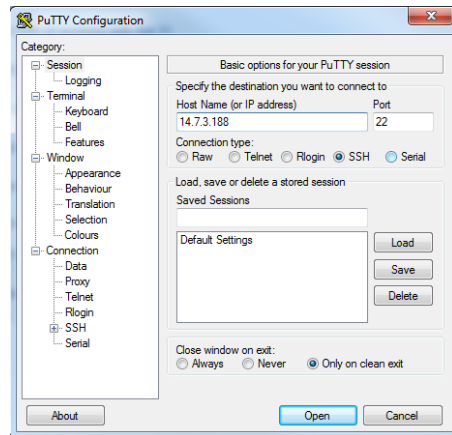
```
pi@raspberrypi:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:a8:3b:56:1a:f7
          inet addr:14.7.3.188  Bcast:14.7.3.255  Mask:255.255.2
55.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3026 errors:0 dropped:0 overruns:0 frame:0
          TX packets:462 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:229516 (224.1 KiB)  TX bytes:60888 (59.4 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1104 (1.0 KiB)  TX bytes:1104 (1.0 KiB)

wlan0     Link encap:Ethernet  HWaddr 74:df:21:5b:a3:9c
          inet addr:32.8.0.142  Bcast:32.8.0.255  Mask:255.255.2
55.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:120 errors:0 dropped:40 overruns:0 frame:0
          TX packets:12 errors:0 dropped:4 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:20955 (20.4 KiB)  TX bytes:9956 (9.7 KiB)
```

If it looks similar but the IP addresses aren't present, that network link hasn't been established.

In this example, we know our Ethernet is on IP 14.7.3.188 and our wireless on 32.8.0.142. These can be entered into the PuTTY configuration window to begin the session. From here, it just works like the serial link!



Configuring PuTTY

Configuring the Pi

This section goes over configuring the keyboard, wired, and wireless internet.

Configuring the Keyboard Layout

The Raspbian distribution comes defaulted to European hardware. With US keyboards, the “ (quotation mark) symbol is replaced by @ (commercial at) and our number sign # is replaced by the European pound sign £. This can make it aggravating when trying to #define things. Invoke the config tool with "sudo raspi-config," and take the following actions:

Set the default locale for the system

- Select "Internationalisation Options"
- Select "Change Locale"
- Deselect en_GB.UTF-8 UTF-8
- Select en_US.UTF-8 UTF-8, (Ok)
- Set default to en_US.UTF-8, (Ok)

Change the keyboard layout – from the Internationalisation Options menu,

- Change Keyboard Layout
- Leave set as: Generic 105-key (Intl) PC (Ok)
- Select Other (Ok)
- Select English (US) (Ok)
- Select English (US) (Ok)
- Select default (Ok)
- Select No compose key (Ok)
- Set Ctrl+Alt+Bksp function (Ok)

Finish with the dialog and get back to the shell

Try the “ and # keys at the prompt. It may be necessary to restart the pi at this point.

Configuring the Internet Interfaces

Automatic Configuration

Raspbian does a good job of configuring wireless networks automatically. By default, DHCP is configured so that the Pi will receive an IP address when a network cable is plugged it to the Ethernet port or when a wireless network is connected.

To use the graphical network tool, right click on the icon on the right side of the task bar, and click "WiFi Networks (dhcpcd) Settings". Then, select the interface desired (wlan0 or eth0) to disable the DHCP and set your own IP, if necessary.

To connect to a wireless network, click on the icon, select the desired network, and enter the password.

Hovering over the icon will bring up a status of wlan0 and eth0 that also shows the IP address.

Manual (text-based) Configuration

At this time, the network configuration works out of the box so there's really nothing to configure, but in case something goes awry, here's the basics of what can be played with and a known-working configuration to compare with.

A configuration file, `interfaces`, configures both wired and wireless devices. Enter the following command into a terminal to edit the `interfaces` file.

```
sudo nano /etc/network/interfaces
```

Replace "nano" with "leafpad" if you prefer graphics. Here's what our Pi 3 is using:

```
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

iface eth0 inet manual

allow-hotplug wlan0
iface wlan0 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

allow-hotplug wlan1
iface wlan1 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

By default, this file is set up to get its configuration from `/etc/wpa_supplicant/wpa_supplicant.conf`, which is really the proper place for wifi information. Here are the contents:

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

network={
    ssid="myNetworkSSID"
    psk="mySuperSecurePassword"
    key_mgmt=WPA-PSK
}
```

A note on network configuration: Try the gui tool first, and only modify the files as a last resort. If you find the tool doesn't work, save

your configuration files as a backup, and don't be afraid to try your hand at a manual configuration.

Applying the Changes

Sometimes the link should be cycled for a new configuration to take. Rather than shutting down the pi and restarting, use 'ifdown' and 'ifup' to bring the link down and back up.

For **wireless** connections, use

```
sudo ifdown wlan0
```

and

```
sudo ifup wlan0
```

For **wired** connections, use

```
sudo ifdown eth0
```

and

```
sudo ifup eth0
```

Reading Inputs and Toggling the LEDs

This section contains instructions for getting the software necessary to compile programs that use the GPIO, then redirects to our **Raspberry gPIo tutorial**.

Getting WiringPi for Using C++

The WiringPi library is required to operate the GPIO with C++. There are two methods to get it, mentioned below. Either way, it will have to be built before use.

Get From the WiringPi Website

The first way to obtain it is to download from a direct link to the WiringPi website: <http://wiringpi.com/download-and-install/>.

Get Using Git (Preferred)

Another way to obtain it is using apt-get to obtain git, then by using git to get the repository.

- Install Git – Enter `sudo apt-get install git-core` (may be installed by default)
- Git the wiringPi repo – Enter `sudo git clone git://git.drogon.net/wiringPi`

This will make a folder in your current directory called WiringPi.

Build WiringPi

However it was obtained, the first step is to build the WiringPi library. The source comes with a script to do all this for you.

- Go to the WiringPi directory – Enter `cd wiringPi`
- build the WiringPi resources – Enter `sudo ./build`

This builds the helper files, modifies some paths in Linux and gets WiringPi ready to rock.

At this point, the library should work. Try it out

- Enter `gpio readall`

An Ascii table of pins and states should appear.

Getting Python

Raspbian comes with Python pre-installed. Continue to the gPiO tutorial to find out how to use it.

Using the GPIO.

This excellent tutorial covers all the INs, OUTs, and PWMs of GPIO with the Pi platform.



Raspberry gPiO

OCTOBER 29, 2015

How to use either Python or C++ to drive the I/O lines on a Raspberry Pi.

Resources and Going Further

With the Pi 3 reading buttons and toggling LEDs, you're sure to be the coolest kid on the block, but what else can it do? Let us know how you use your Pi 3 in the comments!

Here are some links to our other Pi resources.

- The gPiO tutorial
- The SPI and I2C tutorial

Need some inspiration?

- Raspberry Pi Twitter Monitor
- Hackers in Residence: Sound visualizer
 - Part 1
 - Part 2
- Raspberry Pi Time-lapse Part 1
- Raspberry Pi Time-lapse Part 2