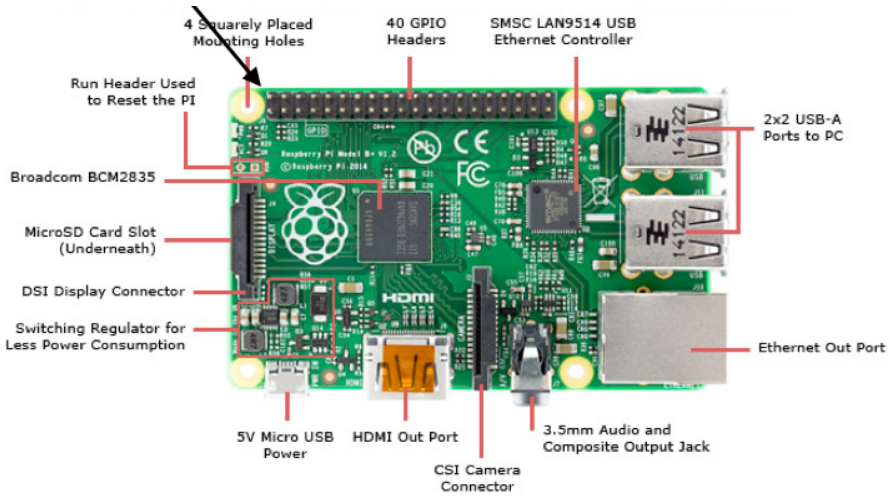
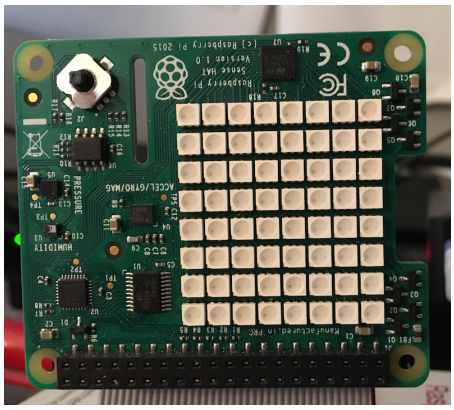
RASPBERRY PI2 DEMO

SenseHat

In this demonstration we will learn how to:
1) use a Raspberry Pi remotely
2) connect a “hat” (pluggable accessory hardware)
3) Write Python programs to make the hardware do something

This specific demo will make use of a “Sense Hat”, an inexpensive RasPi addon which has several different sensors and a 64 pixel RGB display. It was originally designed for the “AstroPi” mission (Pi + Hat were sent to the space station).





https://learn.adafruit.com/neopixels-on-raspberry-pi/overview

Step 1 - Plugging in the hardware

For this demo, there is nothing for you to do. The SenseHat has already been plugged in. Honestly, there are only two ways for it to go, so this wouldn’t be all that challenging anyway, but you can fry the SenseHat by plugging it backwards, so it’s already done for you. For interested parties, you can pick up both RasPis and SenseHats locally at Fry’s Electronics.

Step 2 - Talking to the Raspberry Pi

Normally when using a RasPi, you would connect its HDMI output port to a monitor, and plug in a USB keyboard and mouse. Since I don’t have that much equipment available, we will communicate with our RasPi’s over the network.

For this to work, you need to have an SSH client installed on your computer. On Mac/Linux this should be part of the operating system. On Windows, I suggest you install a free program called “Putty”.

1) RasPis are powered via USB. A standard USB port on a laptop or hub can deliver ~500 mA of power. The RasPi draws about 300 mA and the SenseHat uses about 50mA. So, it *should* be fine to power your RasPi from your laptop (I do it all the time), and almost all laptops have over-current protection so *if* you draw too much power it will just disable the USB port until you reboot. That said, if you are worried, I have a bunch of USB power supplies at the front, and you are welcome to plug

into one of these instead of your laptop. If something DOES happen to your laptop, I don't have any money to reimburse you for it! Since you are talking to the RasPi over WiFi, it doesn't matter if it's across the room. USB is not being used for communications, just power.

2) Each of the Raspberry Pis (RPI) we are using has a number on it. Make a note of this number, particularly if you decide to plug it in at the front of the room instead of using your laptop.
3) Power on your RPI (if you didn't already) by plugging the large USB connector to your computer and the small usb cable (microUSB) into the power port on the RPI.
4) You will see some blinking lights, and likely a brief white flash from the neopixels. Wait for 2 full minutes before attempting the next step.
5) Each RPI will act as its own WiFi access point. Note that you will lose access to the Internet while connected to it. Double check the number. Connect to the "pi#" wifi network, where # is the number of your RPI. It will prompt for a password. It is "introprog" (all lower case). If it fails, try entering the same password again.
6) The address of each RasPi will be 192.168.5.1 Now:
    1) (mac/linux) type: *ssh pi@192.168.5.1*
    2) (windows) launch Putty and connect to 192.168.5.1 with the username "pi"
7) The password is "progrpi06" (that's a zero, not a capital o)
8) You should now have a linux command prompt, like:

pi@raspberrypi:~ $

9) If you don't get this, please ask for help. Don't continue with the next step until you have this prompt.

Step 3 - Programming the SenseHat

With the SenseHat, there are a LOT of possibilities. This one device has:

- 8x8 RGB Pixels
- Mini-joystick
- Gyroscope
- Accelerometer
- Magnetometer
- Temperature
- Barometric pressure
- Humidity

Try this (WARNING, python2 not 3):

```
# sudo gives root permissions, which isn't strictly necessary for the sense hat
sudo ipython

from sense_hat import SenseHat
sense = SenseHat()
sense.clear()  # Blank the LED matrix
sense.set_pixel(0,0,128,128,128)
sense.set_pixel(1,1,255,0,0)

print sense.get_temperature_from_humidity()
help(sense)     # to get more ideas
```

Step 4 - Writing your own program

You are using the RasPi via a terminal session, so you cannot open graphical editor windows to write programs. However, there are some terminal based text editors you can use via SSH which still work pretty well. The easiest one to use is called 'nano'. If you type:

nano myprogram.py

It will open an editor where you can edit your program. Since this isn't mouse-based, you have to use control keys to exit the editor and perform other operations. The basic control keys are shown at the bottom of the editor window. If you see "^X" for example, this means hold down the control key and press X.

Try writing a program to do something interesting. Remember to run your program as:
sudo python myprogram.py

Step 5 - Shutting down

When you are done, you can simply unplug the USB power.