

Parts:

Breadboard / sensor / jumper / cables / DHT22

WiFi adaptor for Pi

iPhone

Mac OSX laptop with ETHERNET CONNECTOR, power supply

Cat 5 cable

Hair dryer

Entire GSOC pi box (you may need the monitor/mouse/keyboard) don't forget to put everything in there!

DETERMINING AVAILABLE WIFI NETWORKS FROM THE PI:

Ideally I've found this to work WHEN YOU ARE CONNECTED DIRECTLY TO THE PI and additionally have the Pi's WiFi adaptor plugged in and on. Utilize the command:

```
raspberry.pi$ iwlist wlan0 scan
```

This will tell you the SSID's that you can connect to BUT YOU WILL STILL NEED THE PASSWORD. After getting the SSID / password, follow the steps below:

FINDING THE IP ADDRESS OF THE RASPBERRY PI ON A NETWORK:

```
raspberry.pi$ ifconfig
```

CONNECTING THE PI TO A NEW WIFI NETWORK:

You need to be able to INITIALLY ACCESS the raspberrry pi, here are ways that can be done:

- 1) Monitor and keyboard, use ifconfig from the Raspberry Pi's UI Terminal application.
- 2) Set up Brad's iPhone hotspot (172.20.10.3 is the raspberrry pi's address if it's the first connection) NOTE: repeated this after cold booting both the phone and the pi seems to be persistent at this IP, though note that the pi was the first device to connect to the iPhone's hotspot after it was activated in each tested instance.
Note: if you wish to SSH into the iPhone via the hotspot, remember that your computer also has to be on the iPhone's hotspot network.
- 3) Use a direct connection with your laptop. From MacOSx, System Preferences, Sharing. Within the Sharing dialog, check Internet Sharing and Thunderbolt Ethernet. Plug the Cat5 cable directly between the Mac and Pi, then use SSH from the Mac to connect to the pi as follows: BRAD\$ ssh pi@192.168.2.2

Configuring the raspberrry pi for a new wifi network:

- 1) Access the pi as per above (i.e. either directly with monitor + keyboard or ssh)
- 2) From the pi, cd /etc/wpa_supplicant
- 3) sudo nano wpa_supplicant

```
network={  
    ssid="The_ESSID_from_earlier"  
    psk="Your_wifi_password"  
}
```

4) save the file and reboot the pi (sudo reboot)

Get back onto the pi as you had before (i.e. monitor+keyboard or prior SSH method), and then use ifconfig to examine wlan0's IP address. This is the raspberry pi's address on the new wifi network.

After knowing this, you SSH into the pi from another computer on this same network using:

```
> ssh pi@IPADDRESS
```

SETTING UP THE [lighttpd web server](#) (pronounced "lighty" web server) on the Pi:

Installing lighttpd

To install the lighttpd web server issue the command:

```
sudo apt-get install lighttpd
```

This will install the web server and also pull in any other packages (called dependencies) that are required. The server will be automatically started and set to start by default after a reboot, e.g.:

```
[ ok ] Starting web server: lighttpd.
```

Set permissions on the web directory /var/www/

It is useful to change the permissions on the www directory to allow your user to update the webpages without needing to be a root permissioned user.

Change the directory owner and group:

```
sudo chown www-data:www-data /var/www
```

Allow the group to write to the directory:

```
sudo chmod 775 /var/www
```

Add the pi user to the www-data group:

```
sudo usermod -a -G www-data pi
```

You should reboot (sudo reboot) or logout and back in - to pick up group permissions, or if running Mac OSX you can just start a new terminal session.

USING THE WEB SERVER

You also need to be on this same Wifi network in order to access the website hosted on the Pi. Using the iPhone Wifi example above, after joining the iPhone's Wifi network, you could navigate your browser to 172.20.10.2:8051 and observe the temp / humidity page.

NOTE:

To “invoke” the page, from the Pi you need to run the WSGI python script authored for this purpose. This script reads the DHT22 sensor and fills in a very simple HTML template (contained within the script) with the sensor's values:

```
> cd /var/www
```

```
> sudo python host_temp_humidity.py <IP ADDRESS OF RASPBERRY PI WEB SERVER>
```

The “invoking” step above assumes that the host_temp_humidity.py is using the 172.20.10.2 network for hosting. If you are changing networks (e.g. the Discovery Cube's) then you need to edit (use sudo NANO on the pi) the host_temp_humidity.py file and change the IP of the WSGI hosted site.