WI-FI REPEATER BY RASPBERRY PI POWERED DRONE

Abstract

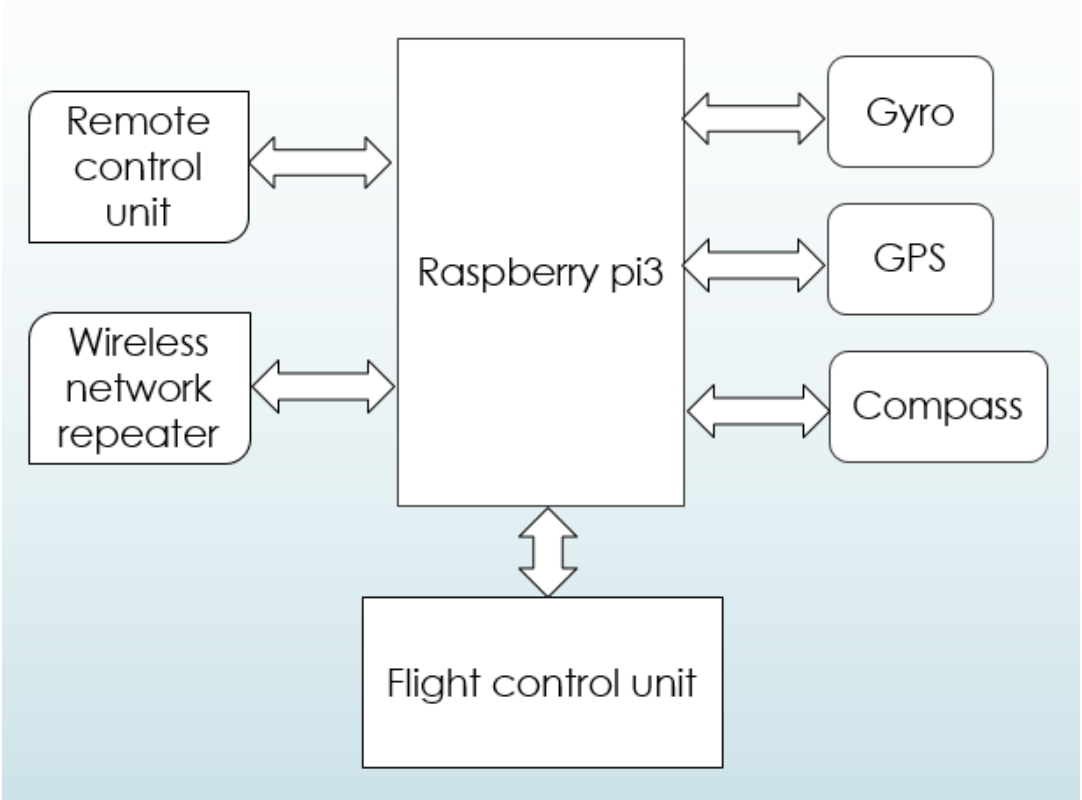
A quadcopter is a multicopter lifted and propelled by four rotors. It is widely used for capturing photos and videos from various altitudes in these days. In this project the aim is to design a quadcopter or drone which can provide a wireless local area network under emergency situations such as flood, earthquake and other natural disasters where there is communication blackouts occures. It can establish communication within a specific area via wireless uplink and downlink which makes it attractive and suitable to the worst situations where one cannot rely on a ground station like cell towers. A group of drones providing wireless network over a wide area prevent the total communication blackout and enables the affected people to call for help. A WiFi based network is easy to establish and cheap for this purpose. The drone simply act as an on air WiFi repeater to form an Ad-hoc network.

Features

The traditional quadcopter/drones are used for simple image/video recording purposes. There are so many windows for adding various applications to the drone like monitoring weather, wind speed, Gas sensor, etc. This drone comes with an integrated with a wireless communication protocol in order to create a temporary communication network over an area for emergency situations where traditional ground station based networks fails. It will be hooked up with a temporary base station in ground, most probably within a disaster management vehicle. In the same time it will be providing network to users in ground also to other nearby drones, forming an Ad-hoc network. So virtually there is no wired connection and no permanent base station. Each drone can act as a client as well as Access point server in same time. The quadcopter is powered by Raspberry pi 3 controller for establishing this wireless network and assisting in flight control and navigation of the drone. Pi 3 is used because it has an inbuilt Wi-Fi module so we only require one extra Wi-Fi antenna to establish the required network. If more range (coverage area) and speed is required, sophisticated Wi-Fi modules can be used. A USB Wi-Fi dongle is used in this drone for establishing the wireless network because of its small size, simple structure and easy to implement. So there are two Wi-Fi modules are available in Pi. One of them can be used as Wi-Fi client and the other as Wi-Fi server or AP. A set of sensors are required for the flight control and navigation purposes; like GPS, Gyro, accelerometer, Compass, etc. A set of motors and propellers required for flying the drone in air.

Conceptual diagram

Block diagram



Components and Tools required

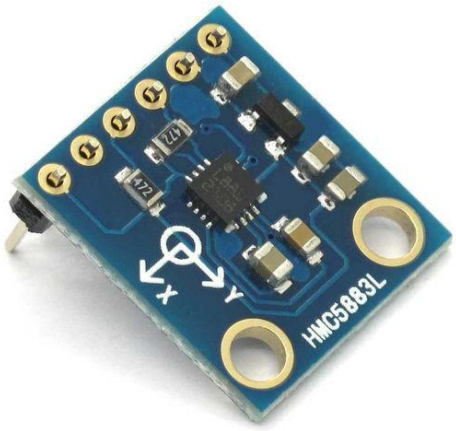
1. Raspberry Pi 3

The Raspberry Pi 3 is a $35 computer that is on the cusp of challenging the modern PC. The bump to the processing power of the latest machine has, according to its co-creator, elevated its performance to a point where it can comfortably be used as a desktop computer. All models feature a Broadcom system on a chip (SoC), which includes an ARM compatible central processing unit (CPU) and an on-chip graphics processing unit (GPU, a VideoCore IV). CPU speed ranges from 700 MHz to 1.2 GHz for the Pi 3 and on board memory range from 256 MB to 1 GB RAM. Secure Digital (SD) cards are used to store the operating system and program memory in either the SDHC or MicroSDHC sizes. Most boards have between one and four USB slots, HDMI and composite video output, and a 3.5 mm phone jack for audio. Lower level output is provided by a number of GPIO pins which support common protocols like I^2C. The B-models have an 8P8C Ethernet port and the Pi 3 and Pi Zero W have on board Wi-Fi 802.11n and Bluetooth. Since Pi has no inbuilt non volatile memory, The operating System and other working files are stored in an SD card. Peripheral devices like keyboard and mouse can be connected to Pi via USB port.

2. Wi-Fi Dongle

A USB Wi-Fi adapter or dongle plugs into one of your desktop or laptop's universal serial bus (USB) ports, allowing you to connect to a wireless network in the home, office, or a public place. You can use this connection to access shared files, devices, and documents, or to connect to the Internet. This can act as both Wi-Fi client and Wi-Fi server. Since Rasbperry Pi is our device, a linux supported USB dongle is required.

3. Compass

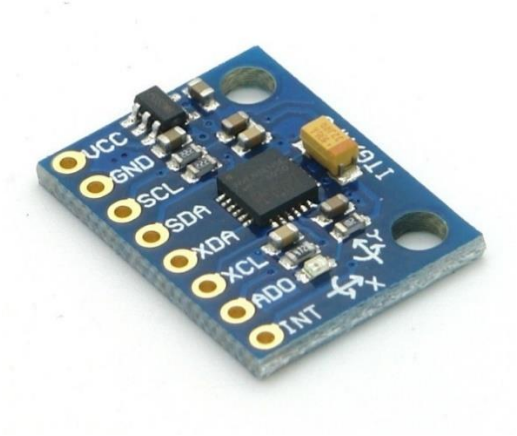


The Honeywell HMC5883L is a surface-mount, multi-chip module designed for low-field magnetic sensing with a digital interface for applications such as lowcost compassing and magnetometry. The HMC5883L includes our state-of-theart, high-resolution HMC118X series magneto-resistive sensors plus an ASIC containing amplification, automatic degaussing strap drivers, offset cancellation, and a 12-bit ADC that enables 1° to 2° compass heading accuracy. The I2C serial bus allows for easy interface. The HMC5883L is a 3.0x3.0x0.9mm surface mount 16-pin leadless chip carrier (LCC). Applications for the HMC5883L include Mobile Phones, Netbooks, Consumer Electronics, Auto Navigation Systems, and Personal Navigation Devices

4. GPS

A GPS navigation device or GPS receiver, and when used for vehicle navigation commonly referred to simply as a GPS, is a device that is capable of receiving information from GPS satellites and then to accurately calculate its geographical location. The Global Positioning System (GPS) is a global navigation satellite system (GNSS) made up of a network of a minimum of 24, but currently 30, satellites placed into orbit by the U.S. Department of Defense.

5. Gyroscope



A gyroscope is a device that uses Earth's gravity to help determine orientation. Its design consists of a freely-rotating disk called a rotor, mounted onto a spinning axis in the centre of a larger and more stable wheel.

Cost estimation

Raspberry pi 3 and power adapter: 3000/-

USB Wi-Fi dongle: 700/-

Sensors: 2200/-

Drone Frame and kit: 18000/-

Battery: 2500/-

Work plan

23/01/2017 - literature survey

01/02/2017 - Setting up Raspberry pi, interfacing sensors

15/02/2017 - Flight controller design, Filter design

20/02/2017 - Integrating Motors, propellers and Remote control unit

30/02/2017 - Testing and troubleshooting

02/03/2017 - Making Pi as a Ethernet to Wi-Fi router

10/03/2017 - Making Pi as a Wi-Fi to Wi-Fi repeater

20/03/2017 - Final testing and verification

Reports

Week 1: 23/01 - 29/01

Plan

Study about Quadcopter Physics
Study about Raspberry pi 3
Study about Languages used in PI3

Work Completed

Quadcopter physics learned

Week 2: 30/01 - 05/02

Plan

Study about Raspberry pi 3
Study about Languages used in PI3

Work Completed

Study about Raspberry pi 3
Study about Languages used in PI3

Week 3: 06/02 - 12/02

Plan

Installing OS into the Raspberry pi
Running some basic codes

Work Completed

Installing OS into the Raspberry pi
Running some basic codes

Week 4: 13/02 - 19/02

Plan

Interfacing sensors and their calibration
Study about various wireless network protocols

Work Completed

Interfacing sensors and their calibration
Study about various wireless network protocols

Week 5: 20/02 - 26/02

Plan

Study about Raspberry pi 3 network interfaces
Study about google and facebook drone projects

Work Completed

Study about Raspberry pi 3 network interfaces
Study about google and facebook drone projects

Week 6: 27/02 - 05/03

Plan

Challenges in Drone based networks
Make Raspberry pi3 has a simple Wi-Fi router from Ethernet
Finding codes and modification

Work Completed

Identified Challenges in Drone based networks
Required codes obtained

Week 7: 06/03 - 12/03

Plan

Make Raspberry pi3 has a simple Wi-Fi router from Ethernet

Work Completed

Successfully established Pi as Wi-Fi router from Ethernet

Week 8: 13/03 - 19/03

Plan

Purchase Wi-Fi dongle
Enabling Raspberry pi as a Wi-FI to Wi-Fi router
Modifying the codes

Work Completed

Code modified. Errors detected

Week 9: 20/03 - 26/03

Plan

Enabling Raspberry pi as a Wi-Fi to Wi-Fi router
Codes again modified

Work Completed

Successfully established Wi-Fi to Wi-Fi repeater on Raspberry Pi

Installing Raspbian OS on raspberry Pi 3

Introduction

Raspberry Pi is a credit card sized microprocessor available in different models with different processing speed starting from 700 MHz. Whether you have a model B or model B+, or the very old version, the installation process remains the same. People who have checked out the official Raspberry Pi website, might have seen them recommending the "NOOBS" or "NOOBS LITE" Operating System (aka "OS") for beginners. But using the Pi is very easy and from being a beginner, one will turn pro in no time. So, it's better to go with the more powerful and more efficient OS, the Raspbian. The main reason why Raspbian is extremely popular is that it has thousands of pre built libraries to perform many tasks and optimize the OS. This forms a huge advantage while building applications.

1 Downloading Raspbian and Image writer

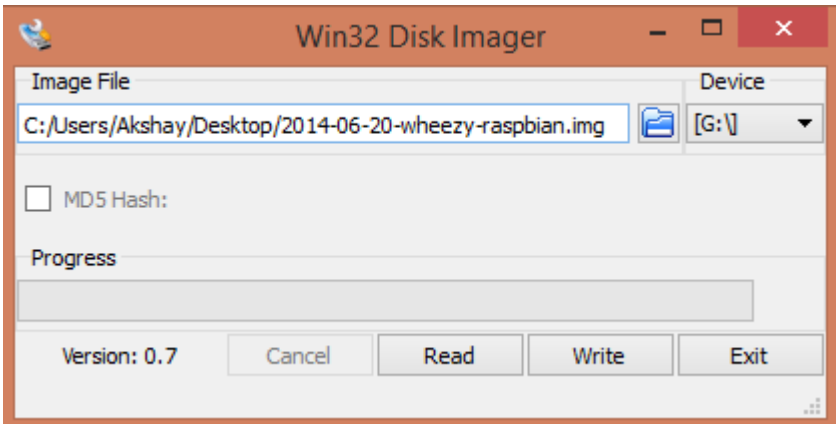
Download the latest version of Raspbian from here. You can download it directly or via the torrents.

You will be needing an image writer to write the downloaded OS into the SD card (micro SD card in case of Raspberry Pi B+ model). So download the "win32 disk imager" from here.

2 Writing the image

Insert the SD card into the laptop/pc and run the image writer. Once open, browse and select the downloaded Raspbian image file. Select the correct device, that is the drive representing the SD card. If the drive (or device) selected is different from the SD card then the other selected drive will become corrupted. SO be careful.

After that, click on the "Write" button in the bottom. As an example, see the image below, where the SD card (or micro SD) drive is represented by the letter "G:\"



Once the write is complete, eject the SD card and insert it into the Raspberry Pi and turn it on. It should start booting up.

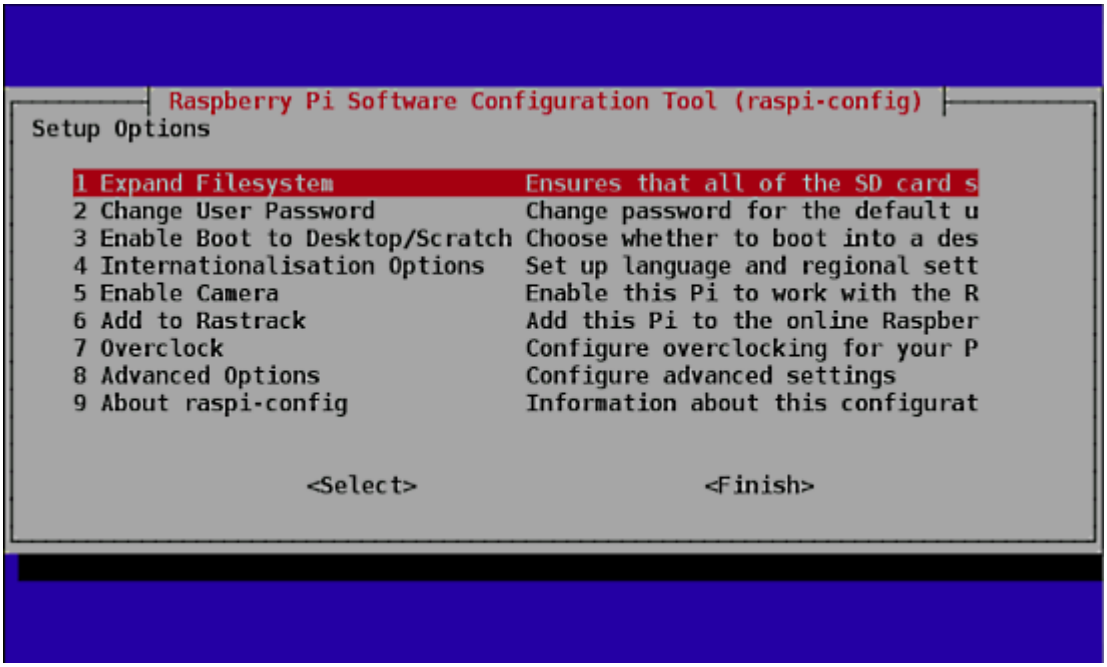
3 Setting up the Pi

Please remember that after booting the Pi, there might be situations when the user credentials like the "username" and password will be asked. Raspberry Pi comes with a default user name and password and so always use it whenever it is being asked. The credentials are:

login: pi

password: raspberry

When the Pi has been booted for the first time, a configuration screen called the "Setup Options" should appear and it will look like the image below.



If you have missed the "Setup Options" screen, its not a problem, you can always get it by typing the following command in the terminal.

sudo raspi-config

Once you execute this command the "Setup Options" screen will come up as shown in the image above.

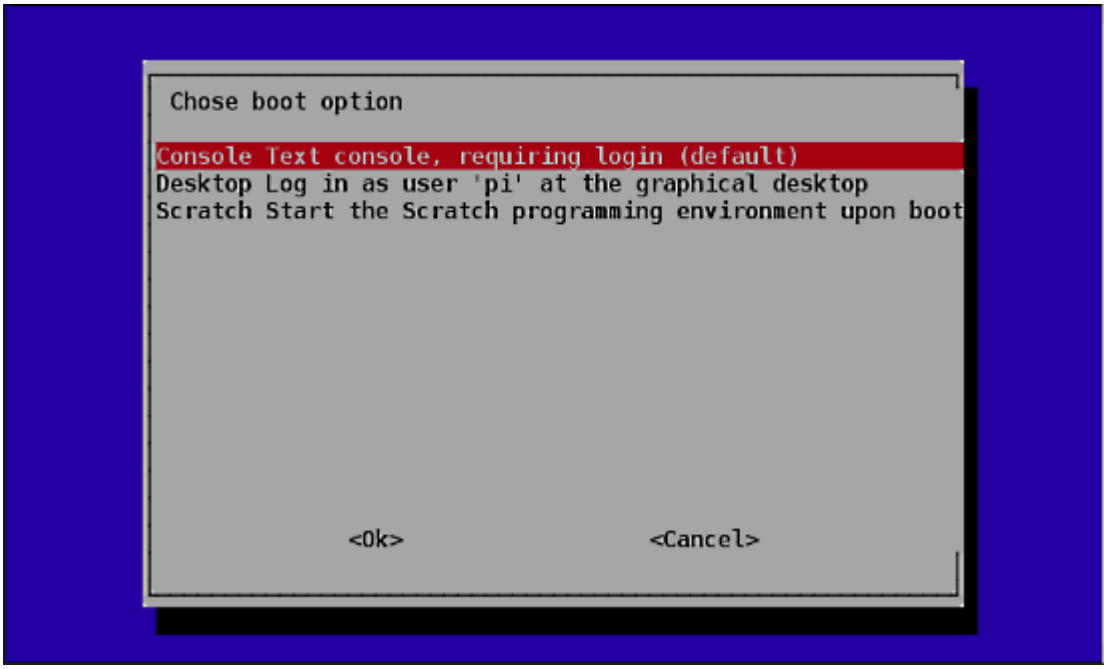
Now that the Setup Options window is up, we will have to set a few things. After completing each of the steps below, if it asks to reboot the Pi, please do so. After the reboot, if you don't get the "Setup Options" screen, then follow the command given above to get the screen/window.

- **The first thing to do:**

select the first option in the list of the setup options window, that is select the "Expand Filesystem" option and hit the enter key. We do this to make use of all the space present on the SD card as a full partition. All this does is, expand the OS to fit the whole space on the SD card which can then be used as the storage memory for the Pi.

- **The second thing to do:**

select the third option in the list of the setup options window, that is select the "Enable Boot To Desktop/Scratch" option and hit the enter key. It will take you to another window called the "choose boot option" window that looks like the image below.



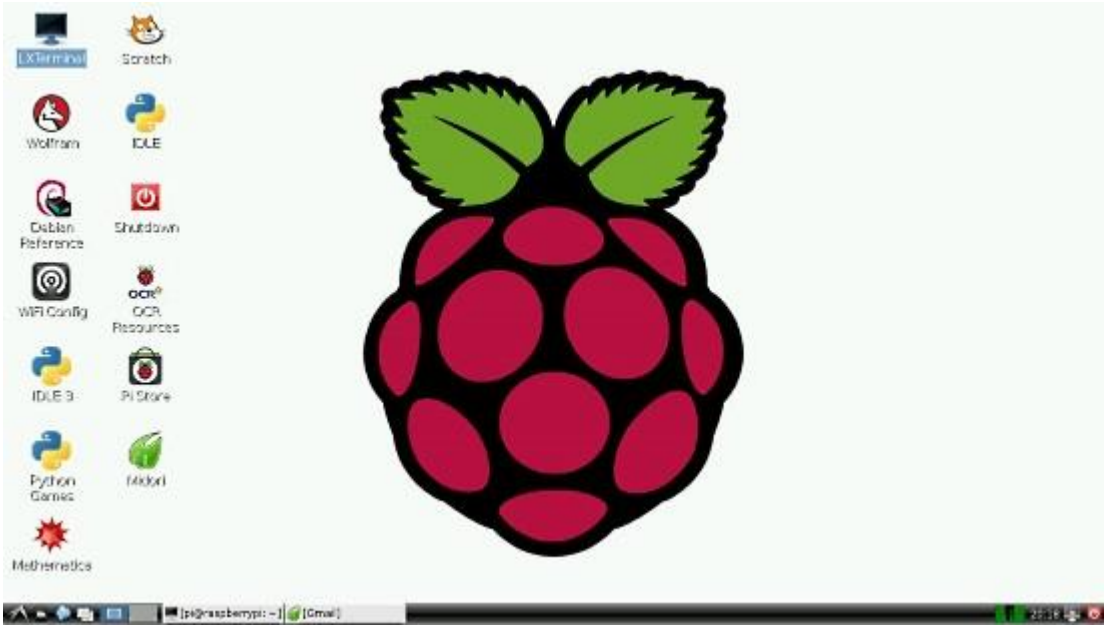
In the "choose boot option window", select the second option, that is, "Desktop Log in as user 'pi' at the graphical desktop" and hit the enter button. Once done you will be taken back to the "Setup Options" page, if not select the "OK" button at the bottom of this window and you will be taken back to the previous window. We do this because we want to boot into the desktop environment which we are familiar with. If we don't do this step, then the Raspberry Pi boots into a terminal each time with no GUI options.

Once, both the steps are done, select the "finish" button at the bottom of the page and it should reboot automatically. If it doesn't, then use the following command in the terminal to reboot.

sudo reboot

4 Updating the firmware

After the reboot from the previous step, if everything went right, then you will end up on the desktop which looks like the image below.



Once you are on the desktop, open a terminal and enter the following command to update the firmware of the Pi.

sudo rpi-update

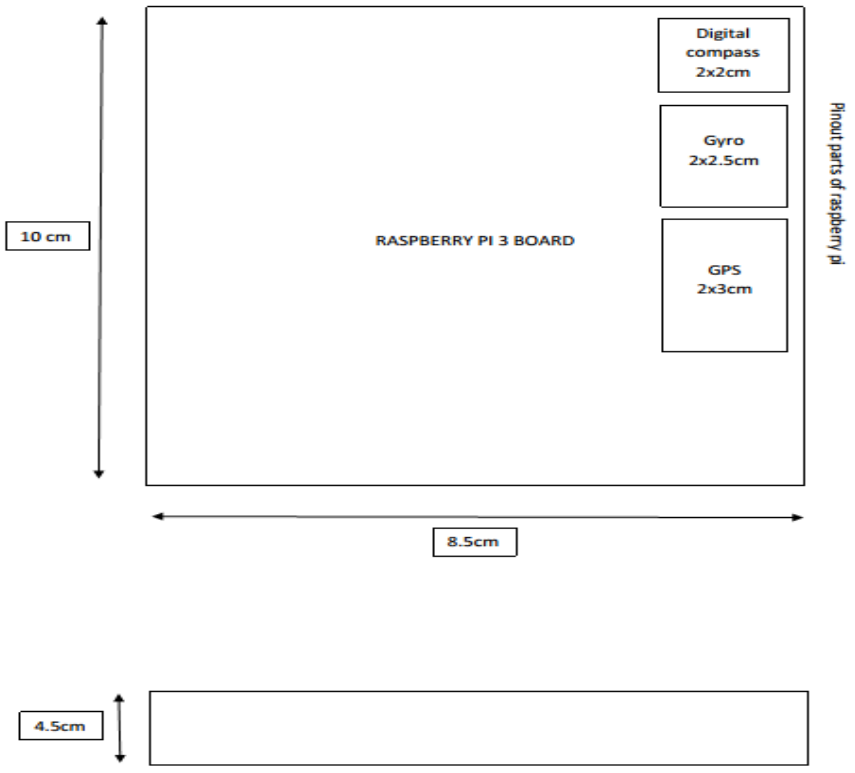
Updating the firmware is necessary because certain models of the Pi might not have all the required dependencies to run smoothly or it may have some bug. The latest firmware might have the fix to those bugs, thus its very important to update it in the beginning itself.

5 Conclusion

So, we have covered the steps to get the Pi up and running. This method works on all the different models of Raspberry Pi (model A, B, B+ and also RPi 2) as Raspbain was made to be supported on all models. However, while installing other software or libraries , the procedure might change a bit while installing depending on the model of the Pi or the version of Raspbian itself. The concept of Raspberry is to keep trying till you get the result or build that you want. This might involve a lot of trial and error but spending the time will be worth it. The actual usage doesn't end here. This is just the beginning. It is up to you to go ahead to build something amazing out of it.

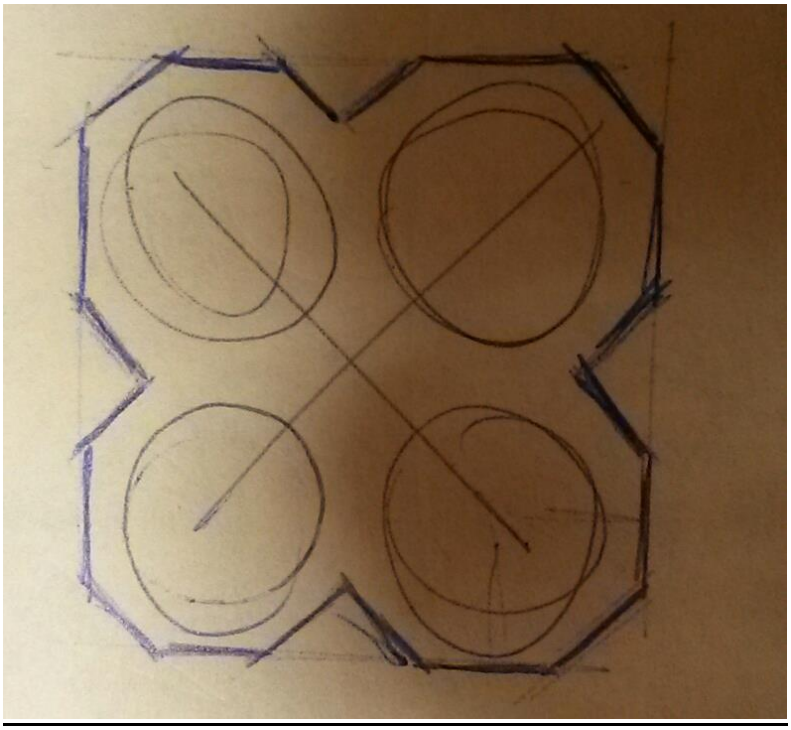
Step 3

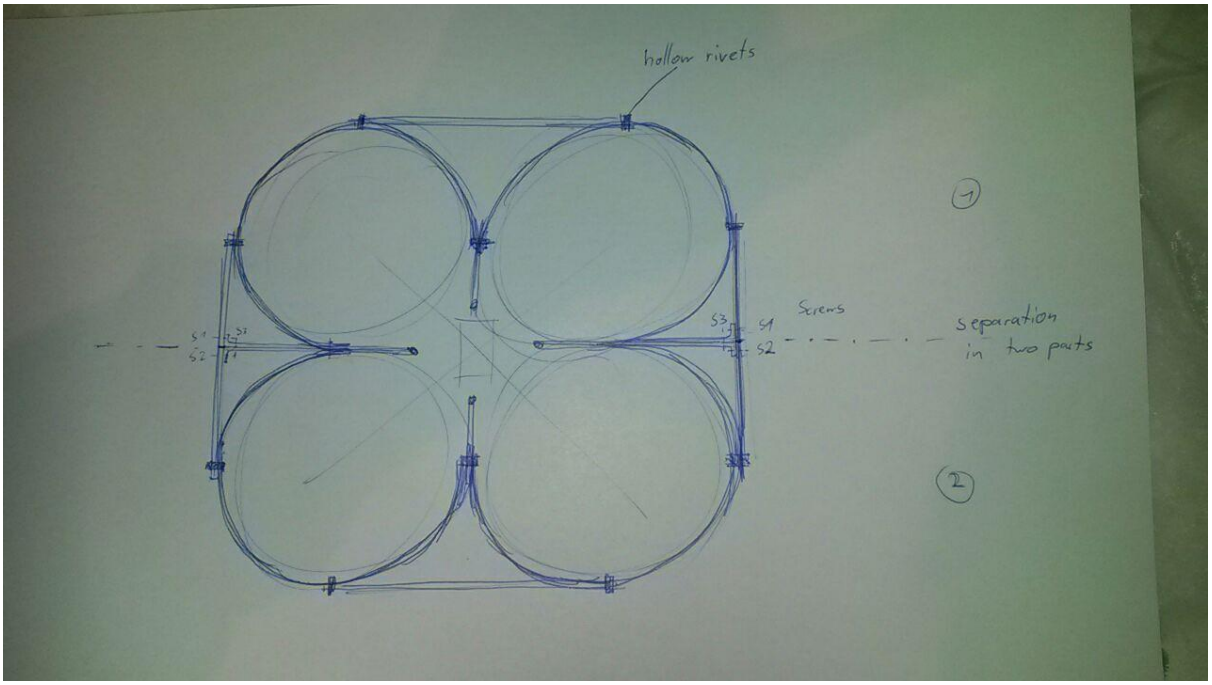
Find sizing of circuit board case on Drone frame

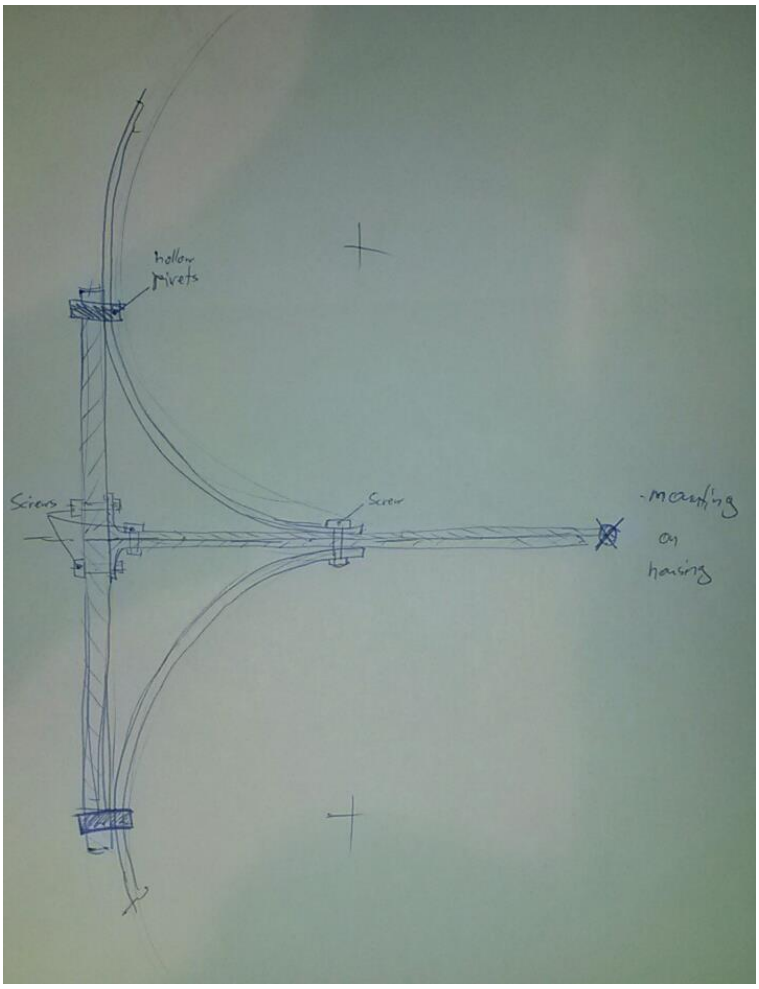


Step 4

Basic drone frame models on discussion and design



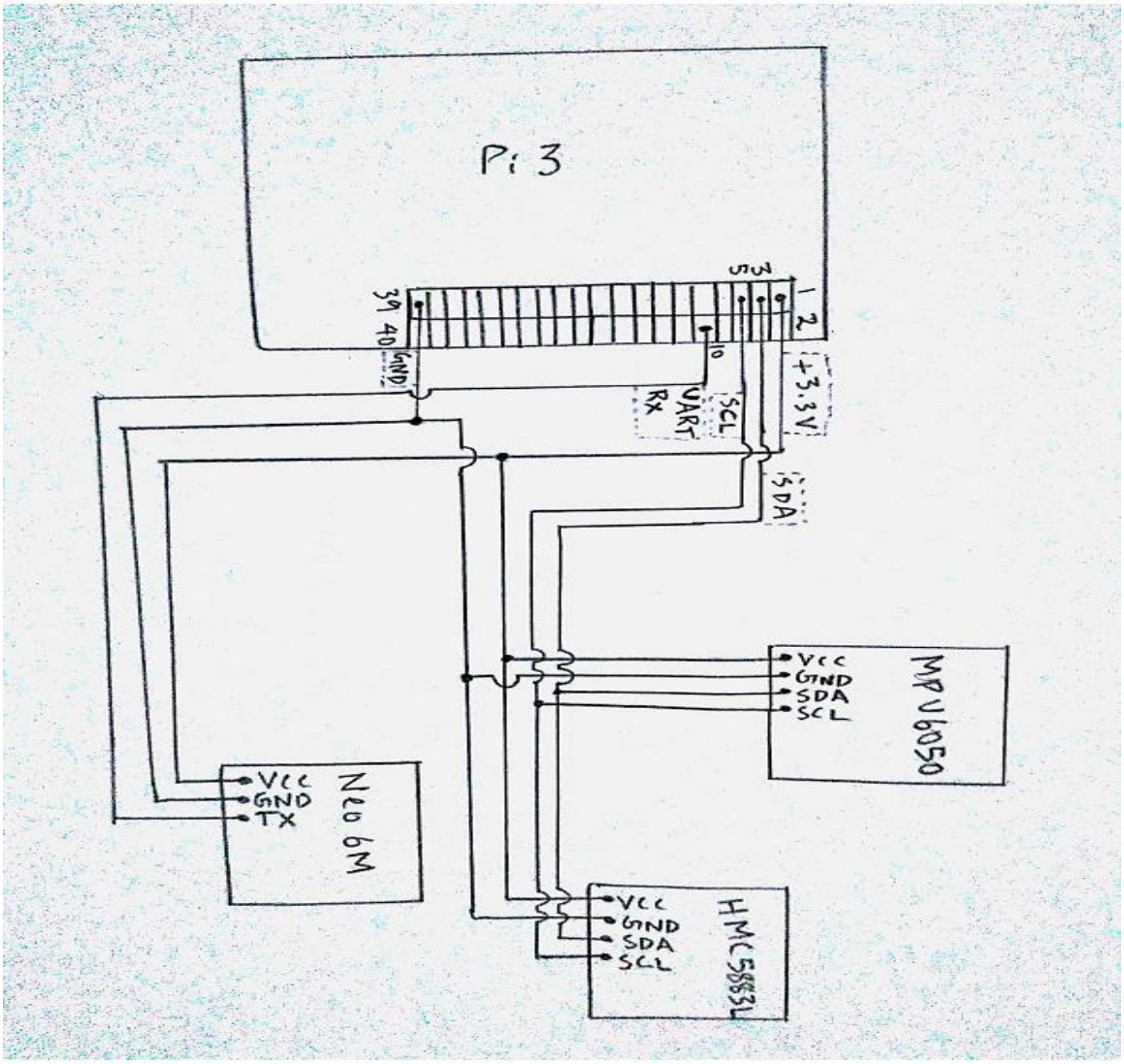

hollow rivets
1
S1
S2
S3
Screws
separation
in two parts
2

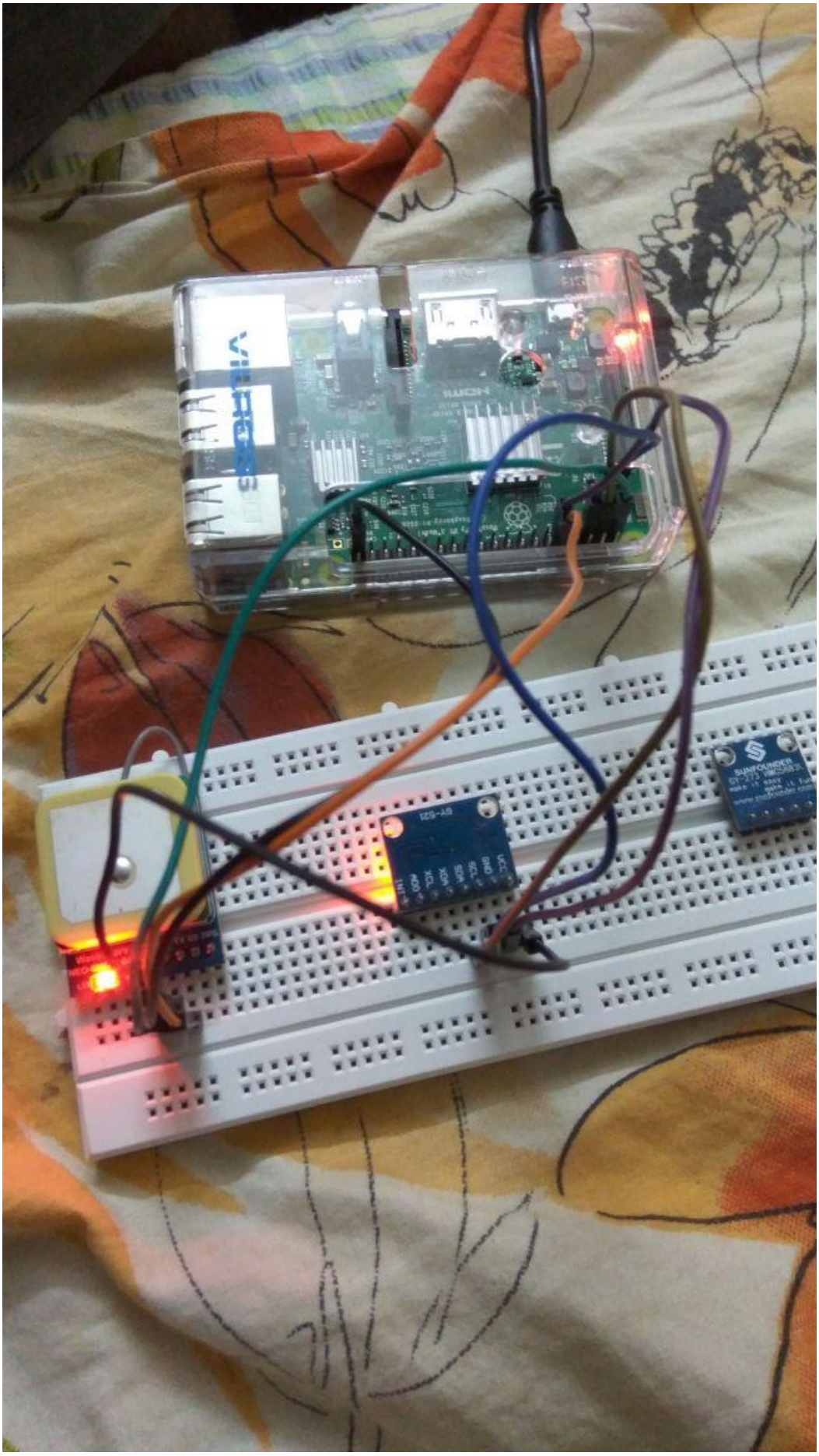

hollow
rivets
Screws
Screw
mounting
on
housing

Step 5

Basic Sensor / HW interfacing

Circuit Diagram




SUNFOUNDER

Procedure to Establish Wi-Fi to Wi-Fi repeater

Step 1: Preparation

This tutorial assumes you have your Pi mostly set up and ready to go. Please follow the tutorials in order to

- Install the OS onto your SD card
- Boot the Pi and configure

When done you should have a Pi that is booting Raspbian, you can connect to with a USB console cable and log into the Pi via the command line interface. Don't forget to expand the SD card, or you may run out of space.

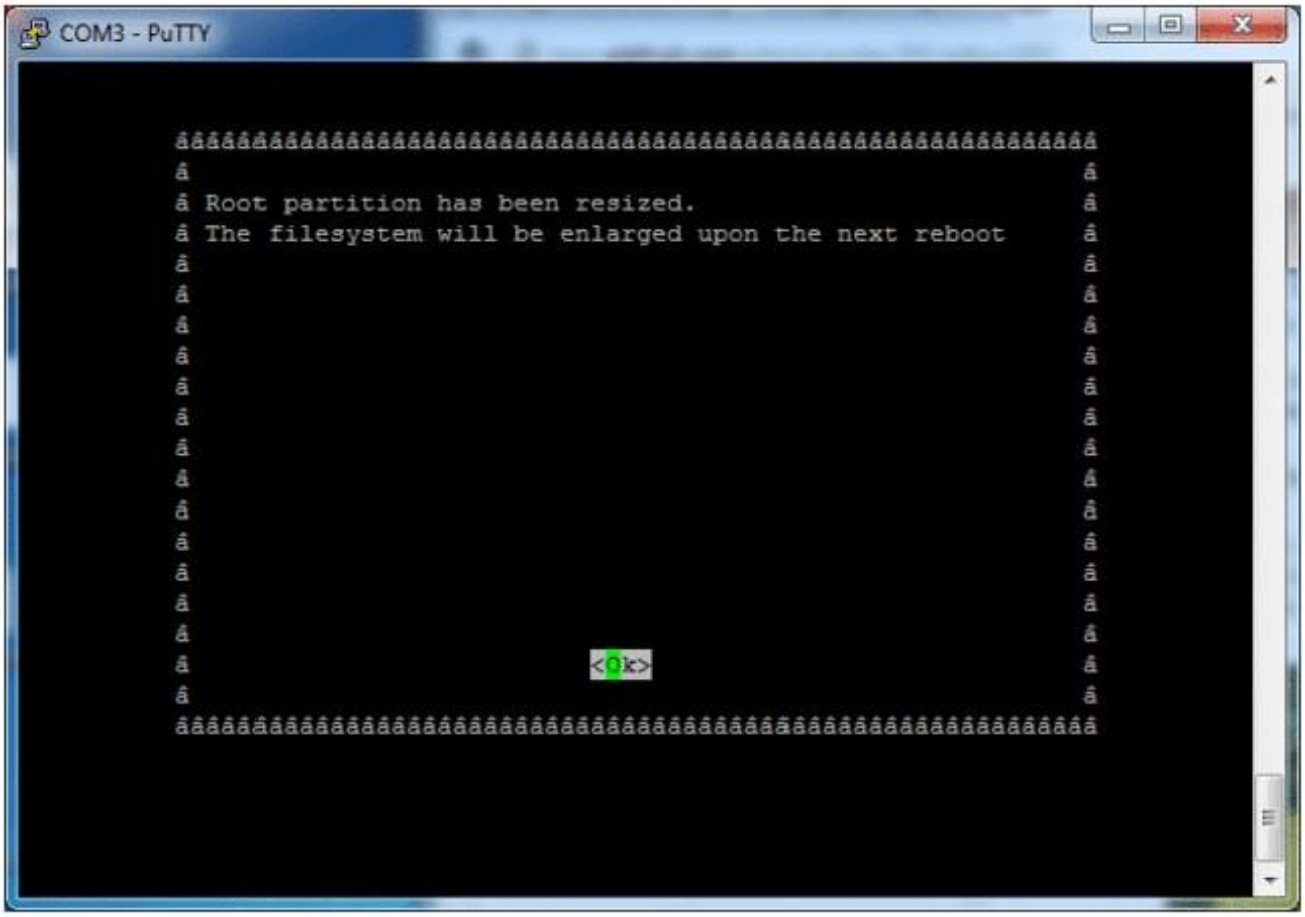
Step 2: Requirements

Hardware Requirement:

- Raspberry Pi 3
- Power adapter
- SD Card
- USB Wi-Fi Dongle (AP supported)
- Ethernet with internet access
- Display, Keyboard, Mouse

Software Requirement:

- Raspbian Jessie OS for Pi 3
- Hostapd package: Enables Pi to create an access point
- Dnsmasq package: This is a combined DHCP and DNS server
- Isc-dhcp server package: For setting Static IP address to the modules
- Iptables package: For routing IP packets between wi-fi modules



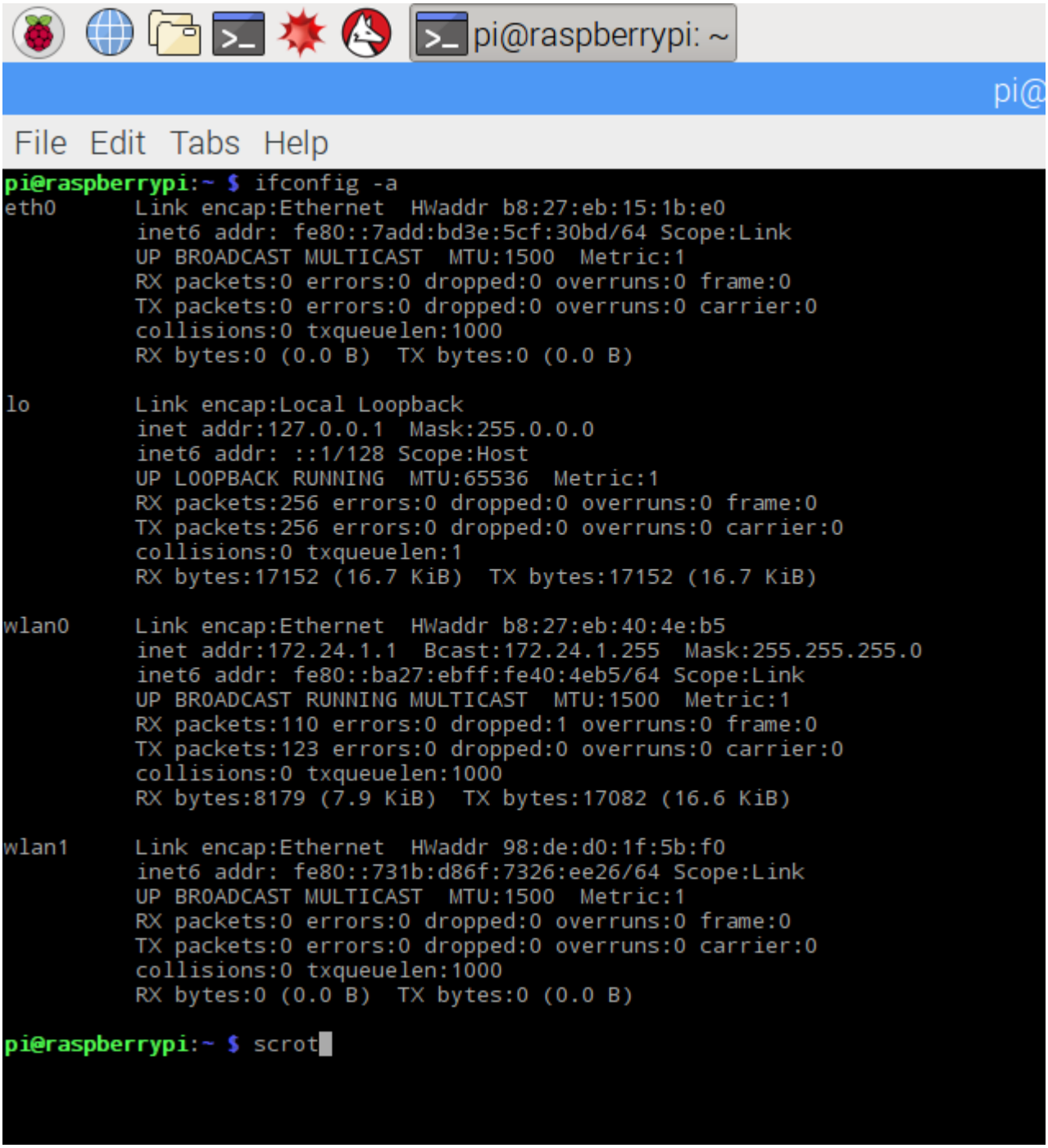
Step 2: Check Network configuration status

You will also want to set up your WiFi dongle. Run

sudo shutdown –h

now and then plug in the WiFi module when the Pi is off so you don't cause a power surge. If you have a Pi 3, or any other Pi with built in WiFi, an external WiFi adapter is not required but you can use one if you need a bigger/external antenna. When it comes back up check with ***ifconfig -a*** that you see wlan0 - the inbuilt WiFi module, wlan1 – the external WiFi dongle and eth0 as Ethernet module.

*****Scrot*** command is used for taking screenshots.



If this is shown then we can assure that the modules are intact and we can go for further steps.

Step 3: Install software

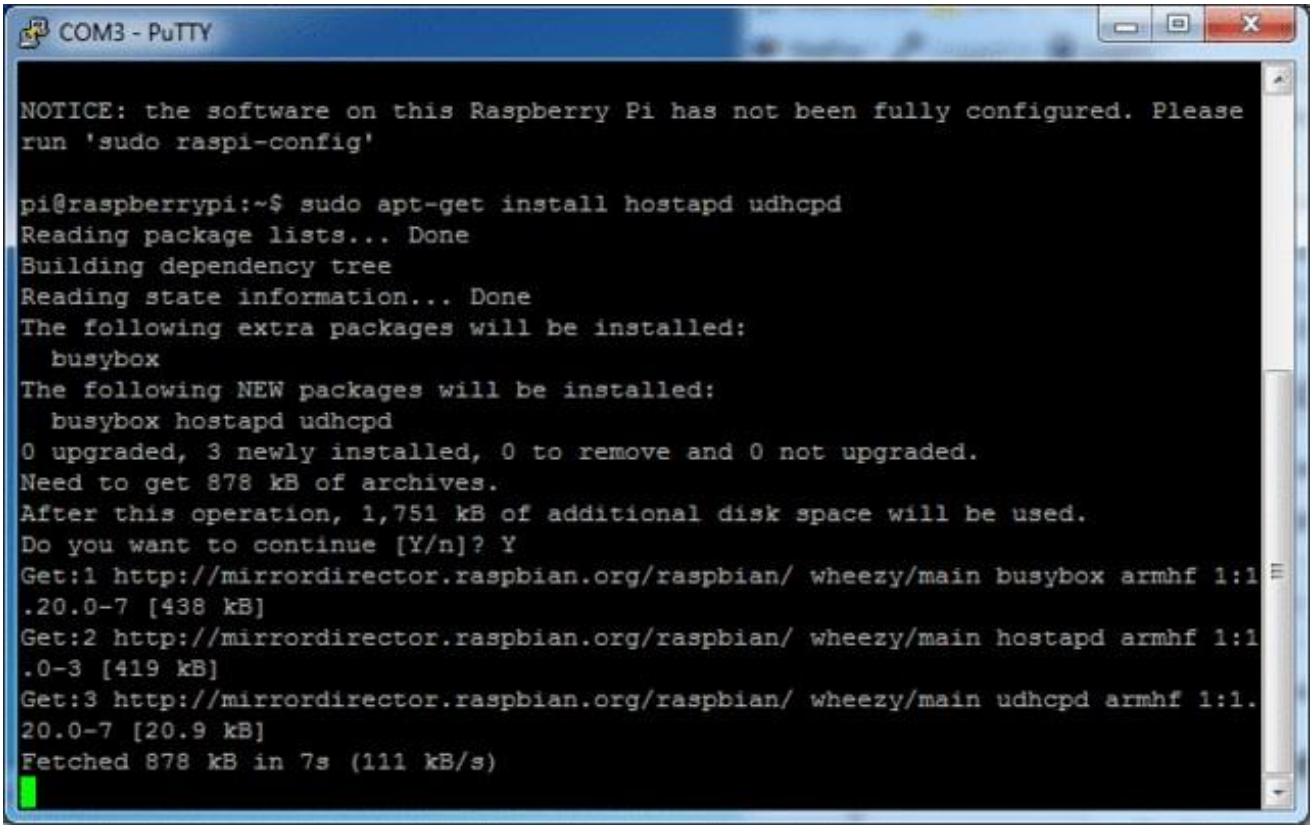
Next up we install the software packages required for setting up the WiFi repeater onto the Pi that will act as the 'hostap' (host access point) You need internet access for this step so make sure that Ethernet connection is up.

sudo apt-get update ***For updating all drivers to latest version

sudo apt-get install hostapd isc-dhcp-server ***Install isc-dhcp and hostapd

sudo apt-get install dnsmasq hostapd ***Install dnsmasq

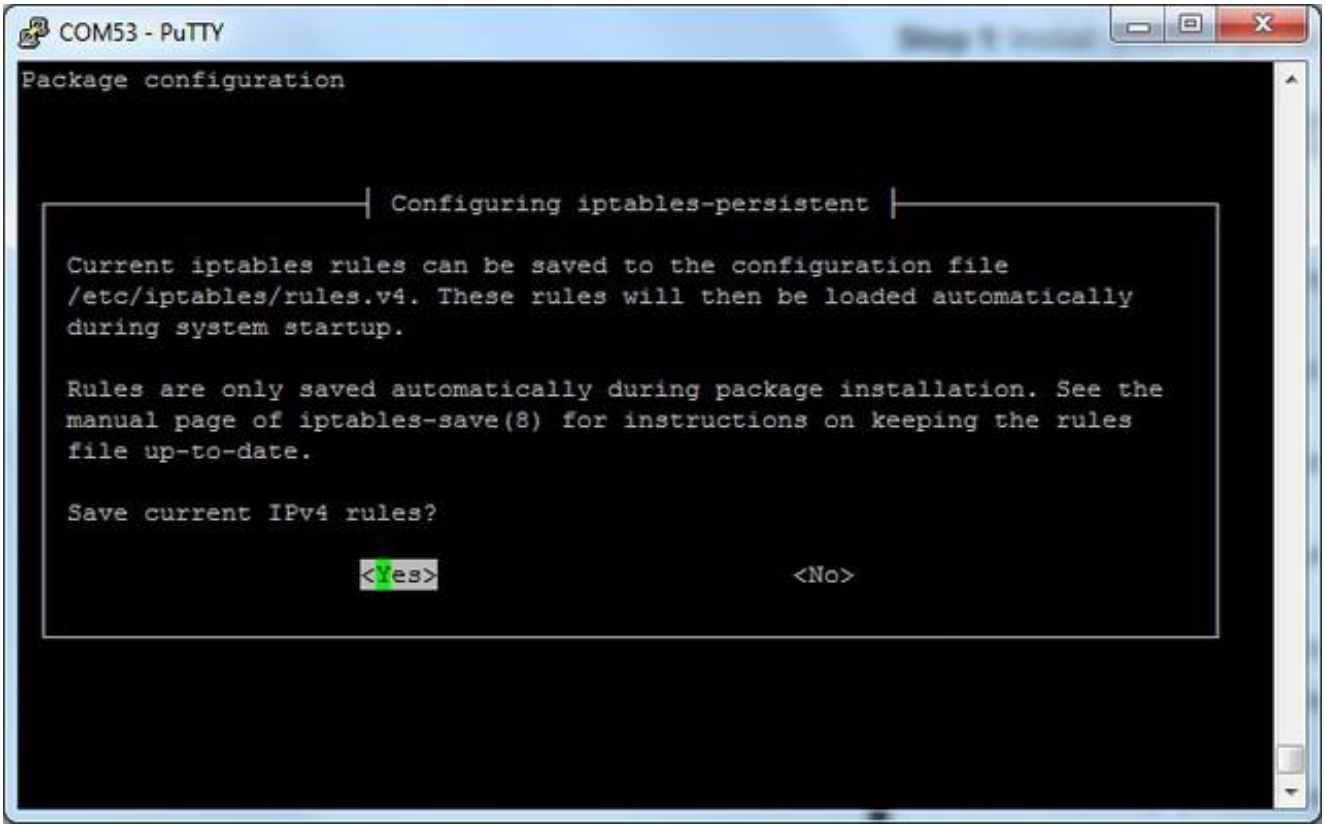
(You may need to sudo apt-get update if the Pi can't seem to get to the apt-get repositories)



Also install a nice **iptables** manager with

sudo apt-get install iptables-persistent

You'll get two 'config' screens, say Yes to both to save IPV4 rules.



Step 4: Taking Back ups

In this tutorial we are going edit the original network configuration files and DHCP of the Pi 3 from its default behaviour. So it is better to take backup of all files before making permanent changes to them. Run the following codes to take backups.

sudo cp /etc/dhcp/dhcpd.conf /etc/dhcp.conf.backup ***DHCP

sudo cp /etc/network/interface /etc/network/interfaces.backup ***Network interface

sudo cp /etc/sysctl.conf /etc/sysctl.conf.backup ***IPV4

sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.backup ***Dnsmasq

Step 5: Set up DHCP server

Next we will edit /etc/dhcp/dhcpd.conf, a file that sets up our DHCP server - this allows wifi connections to automatically get IP addresses, DNS, etc. Also make wlan0 from assigning automatic IP by DHCP. For this edit dhcpd file.

sudo nano /etc/dhcp/dhcpd.conf

Enter the following line at bottom:

denyinterfaces waln0

find the line saying:

option domain-name "example.org";

option domain-name-servers ns1.example.org, ns2.example.org;

and change them to add a # in the beginning so they say:

#option domain-name "example.org";

#option domain-name-servers ns1.example.org, ns2.example.org;

Find the lines that say:

 # If this DHCP server is the official DHCP server for the local

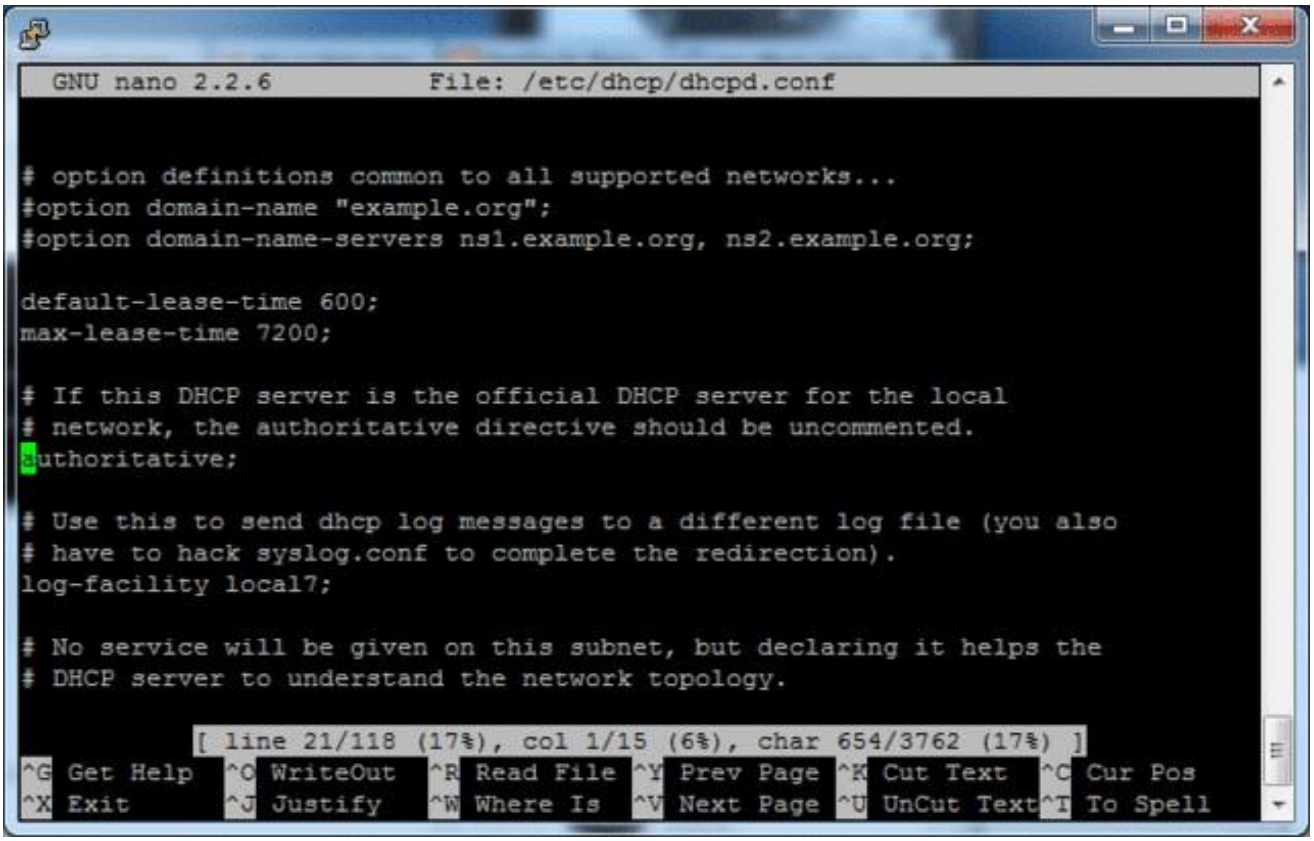
 # network, the authoritative directive should be uncommented.

#authoritative;

and remove the # so it says:

 # If this DHCP server is the official DHCP server for the local

 # network, the authoritative directive should be uncommented. authoritative;



Save the file by typing in Control-X then Y then return.

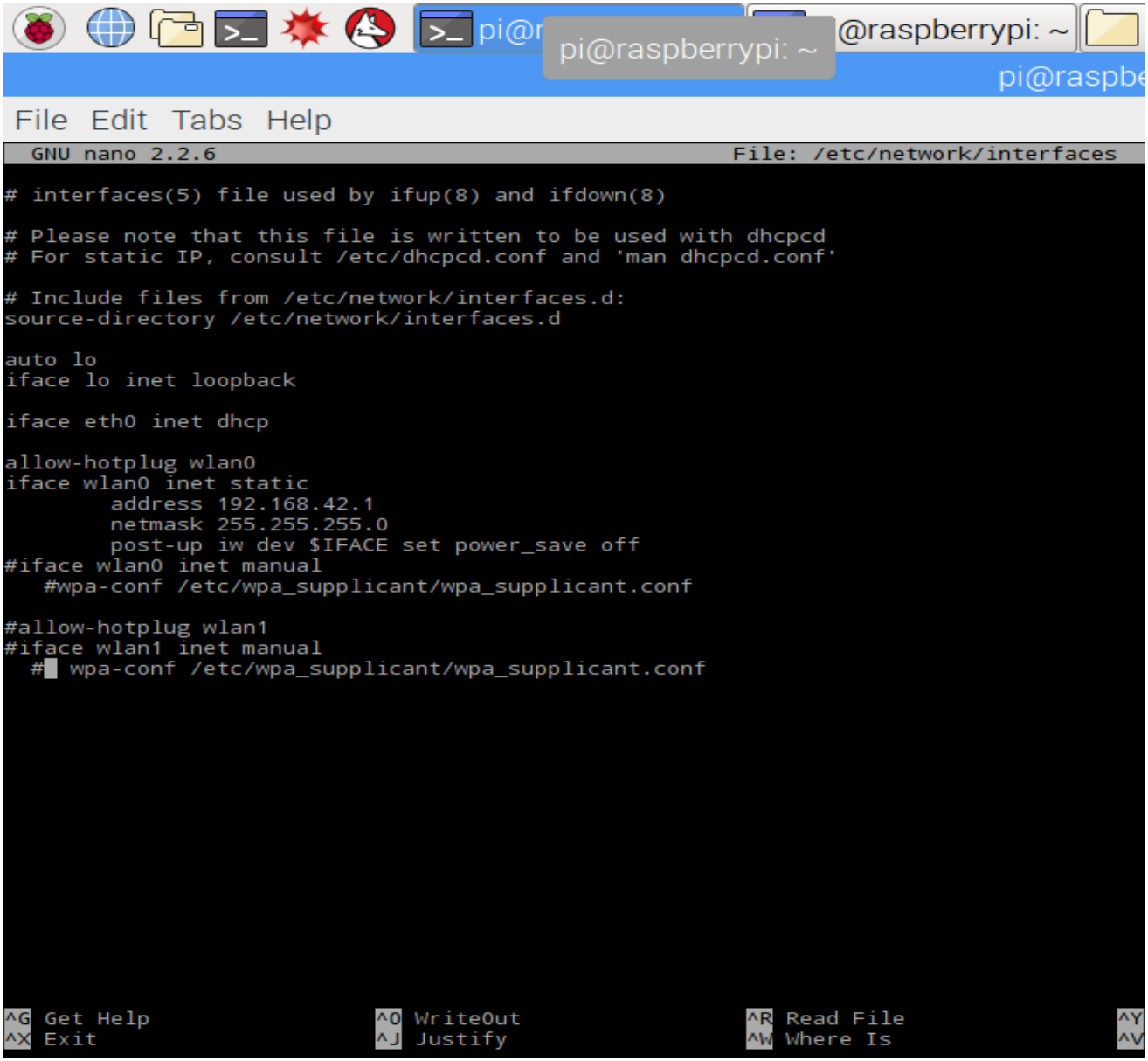
Now we need to edit the network interface files and add network address, mask, etc. for that run

sudo nano /etc/network/interfaces

Add the following lines below “allow – hotplug wlan0”:

allow-hotplug wlan0
iface wlan0 inet static
address 172.24.1.1
netmask 255.255.255.0
network 172.24.1.0
broadcast 172.24.1.255
#wpa-conf
/etc/wpa-supplicant/wpa-supplicant.conf

Save it.



Now restart dhcpcd service. Run;

sudo service dhcpcd restart

Reload the configuration:

Sudo ifdown wlan0; sudo ifup wlan0

Now we succusfully disabled raspberry pi's Wi-Fi from acting as client

Step 6: Configure Access Point

Now we can configure the access point details. We will set up a password-protected network so only people with the password can connect.

Create a new file by running

sudo nano /etc/hostapd/hostapd.conf

Paste the following in, you can change the text after ssid = to another name, that will be the network broad cast name. The password can be changed with the text after wpa_passphrase=

interface=wlan0 ****** module name***

driver=rtl871xdrv ****** Driver hardware name of the module***

ssid=PiAp ****** AP name***

hw_mode=g ****** use 2.4 GHZ***

channel=6 ******use channel 6***

macaddr_acl=0 ****** accept all MAC addresses***

auth_algs=1 ****** use wpa authentication***

ignore_broadcast_ssid=0 ****** Require client to know the network name***

wpa=2

wpa_passphrase=qwerty123 ****** Password***

wpa_key_mgmt=WPA-PSK ****** use preshared key***

wpa_pairwise=CCMP

ieee80211n=1 ******enable 802.11n***

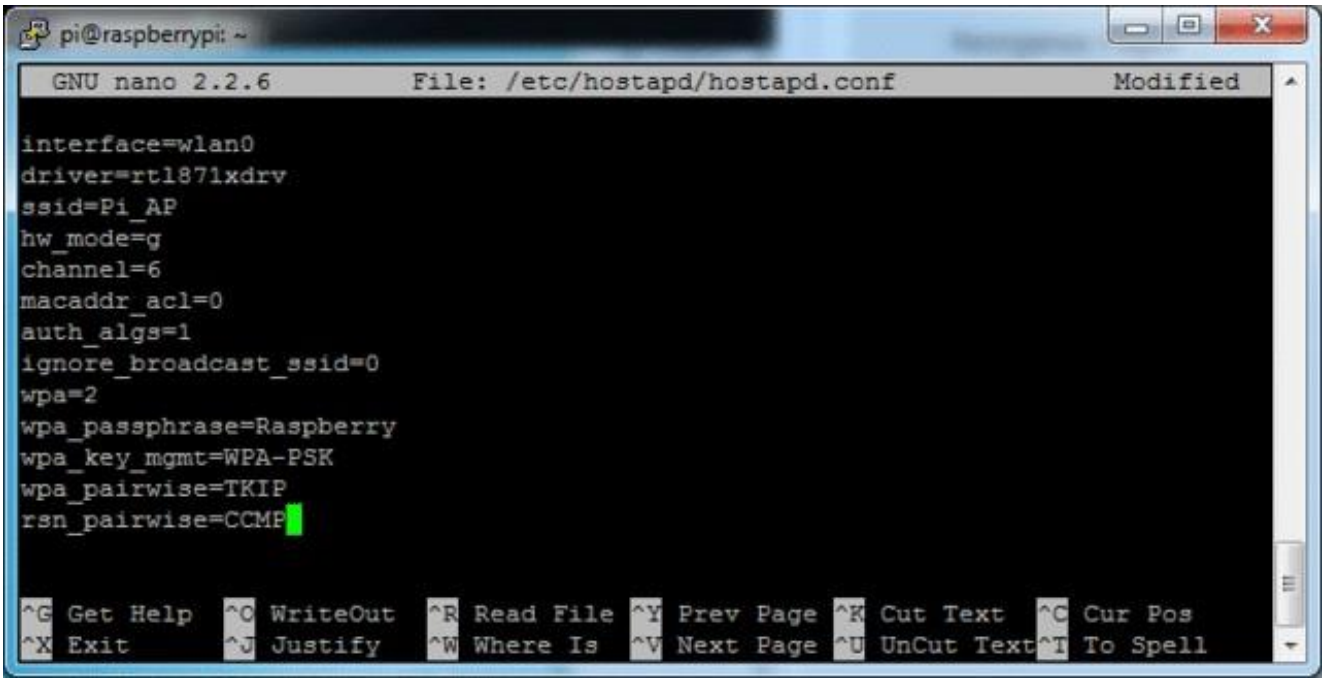
wmm_enabled=1

ht-capab = [HT40][SHORT-GI-20][DSS-CCK-40]

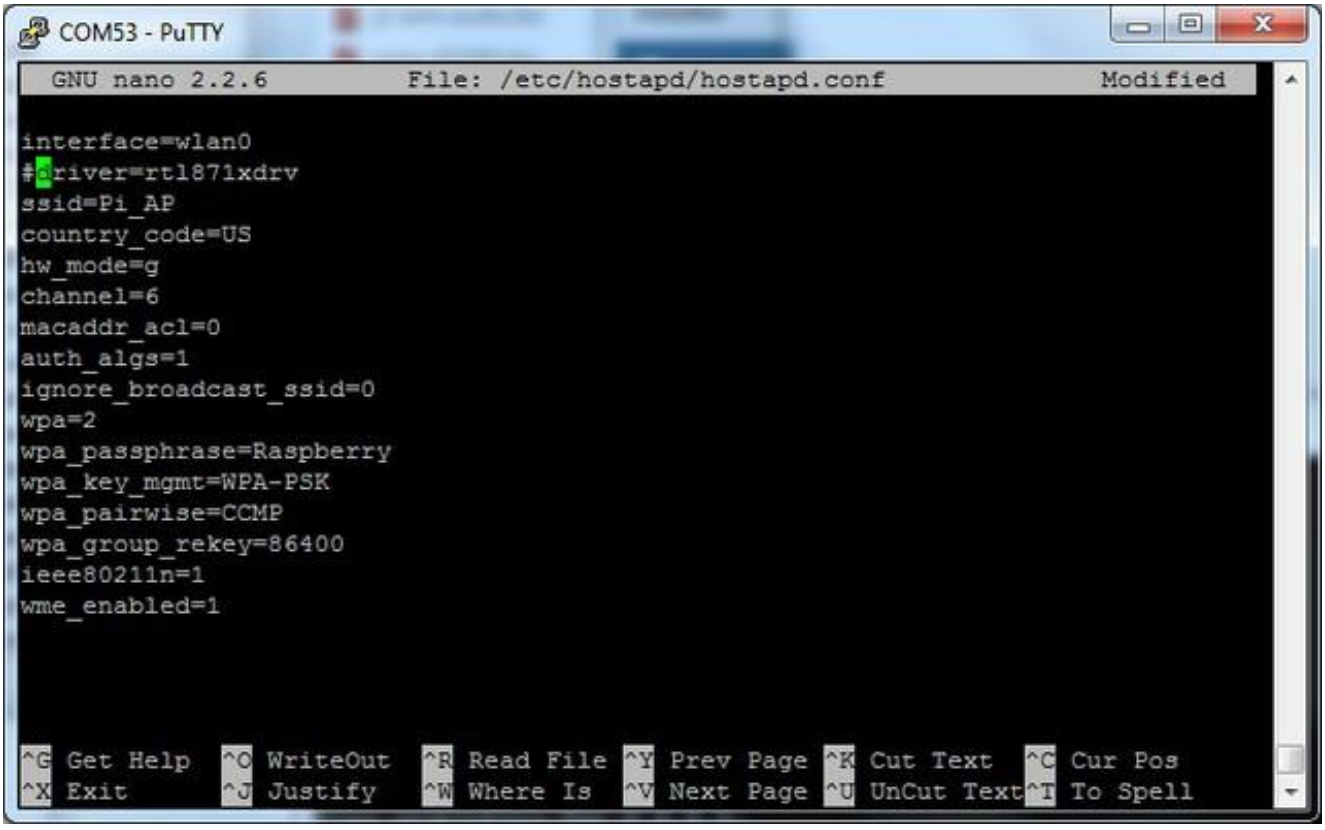
******enable 40HZ channel with 20ns guards***

rsn-pairwise=CCMP

If you are not using the Adafruit wifi adapters, you may have to change the driver=rtl871xdrv to say driver=nl80211 or something



If you are using the Raspberry Pi 3's internal WiFi adapter, comment out the driver=rtl871xdrv line altogether:



Save as usual. Make sure each line has no extra spaces or tabs at the end or beginning. This file is pretty picky

Now we will tell the Pi where to find this configuration file for starting automatically on each boot. Run

sudo nano /etc/default/hostapd

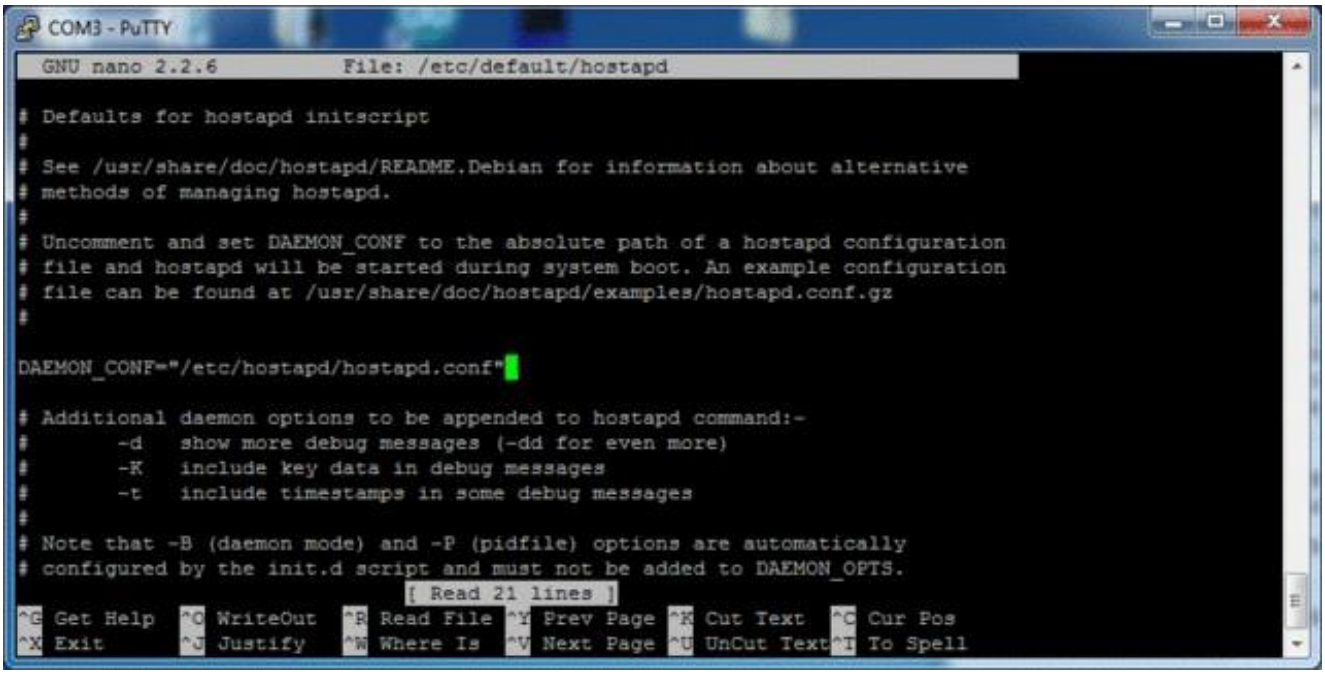
Find the line

#DAEMON_CONF=""

And edit it so it says

DAEMON_CONF="/etc/hostapd/hostapd.conf"

Don't forget to remove the# in front to activate it. Then save the file



Step 7: First test

Finally we can test the access point host. Run

sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf

To manually run hostapd with our configuration file. You should see it set up and use wlan0 then you can check with another wifi computer that you see your SSID show up. If so, you have successfully set up the access point.

If you get this warning

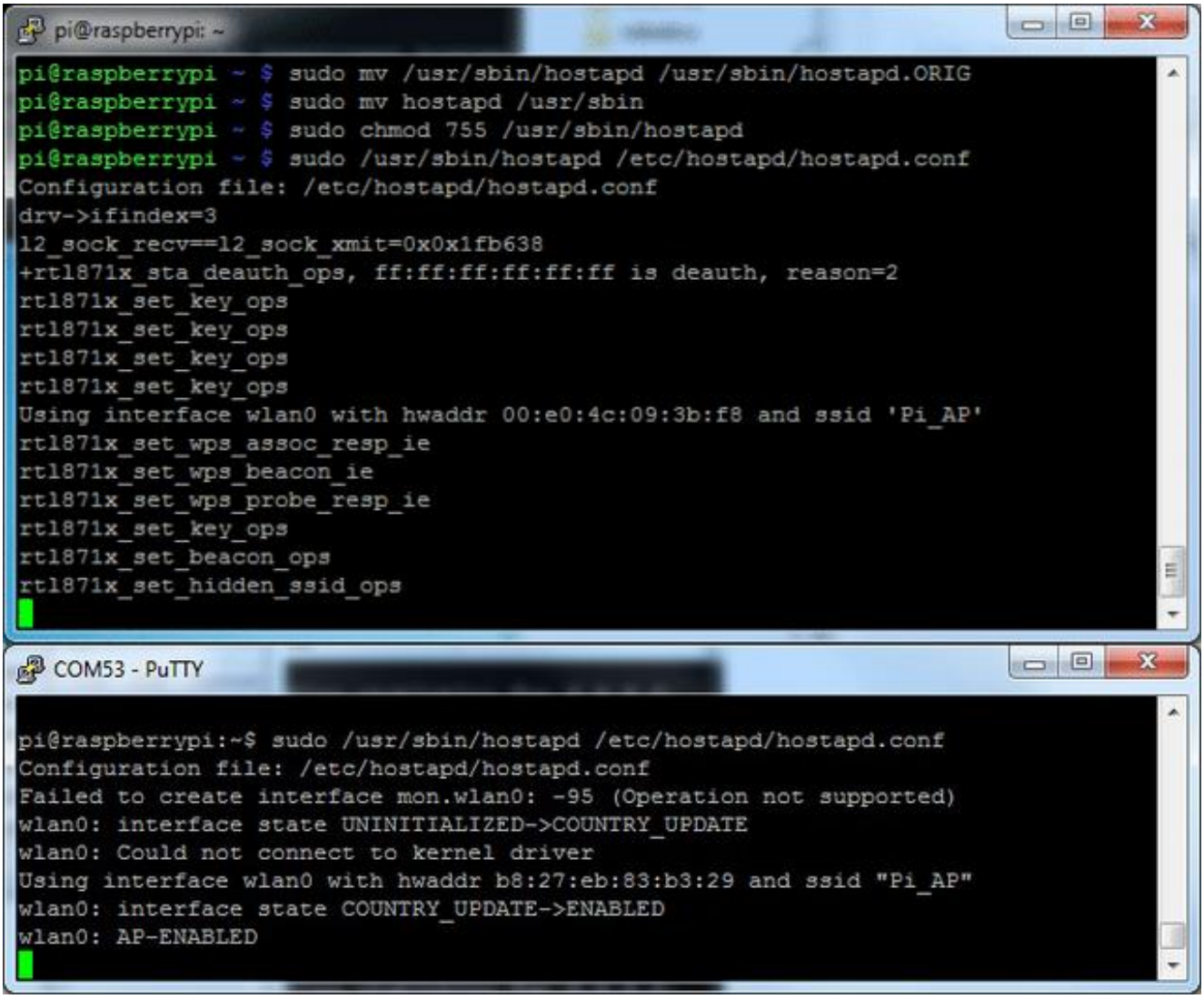
Configuration file: /etc/hostapd/hostapd.conf Line 2: invalid/unknown driver 'rtl871xdrv'

1 errors found in configuration file '/etc/hostapd/hostapd.conf'

Failed to set up interface with /etc/hostapd/hostapd.conf Failed to initialize interface

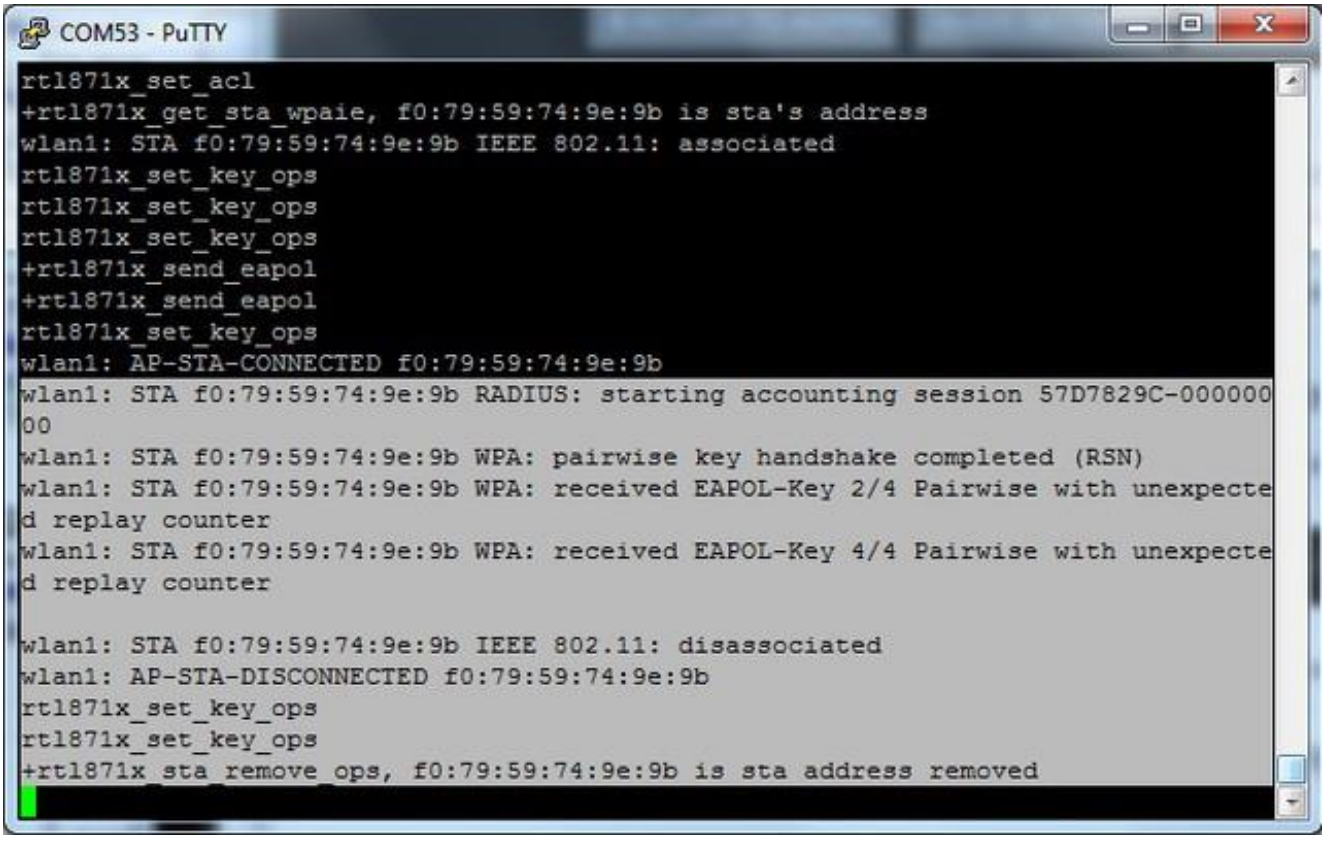
It could mean that either you are not using a RTL871Xdrv WiFi adapter (e.g. Pi 3 internal wifi) and should comment out the driver=rtl871xdrv line in the hostapd config OR you *are* using that chipset and you need to download our recompiled hostapd binary

If it does work, you should get something like this:



And see a new access point created:

You can try connecting and disconnecting from the Pi_AP with the password you set before (probably Raspberry if you copied our hostapd config), debug text will display on the Pi console but you won't be able to connect through to the Wi-Fi dongle yet, IP forwarding is not yet enabled



Cancel the test by typing Control-C in the Pi console to get back to the Pi command line

Step 8: Configure DNSMASQ

Now we have to configure the DNSMASQ for broadcasting our static IP and to enable others to connect to it.Run:

sudo nano /etc/dnsmasq.conf

add the following lines to the file

interface= wlan0

listen-address=172.24.1.1.band-interface

server=8.8.8.8 *** forward DNS request to google DNS

domain-needed ***No short names allowed for domains

bogus-priv ***Never forward address in the non routed address space

dhcp-range=172.24.1.50,172.24.1.150 ***IP address within this range

Step 9: Configure Network Address Translation

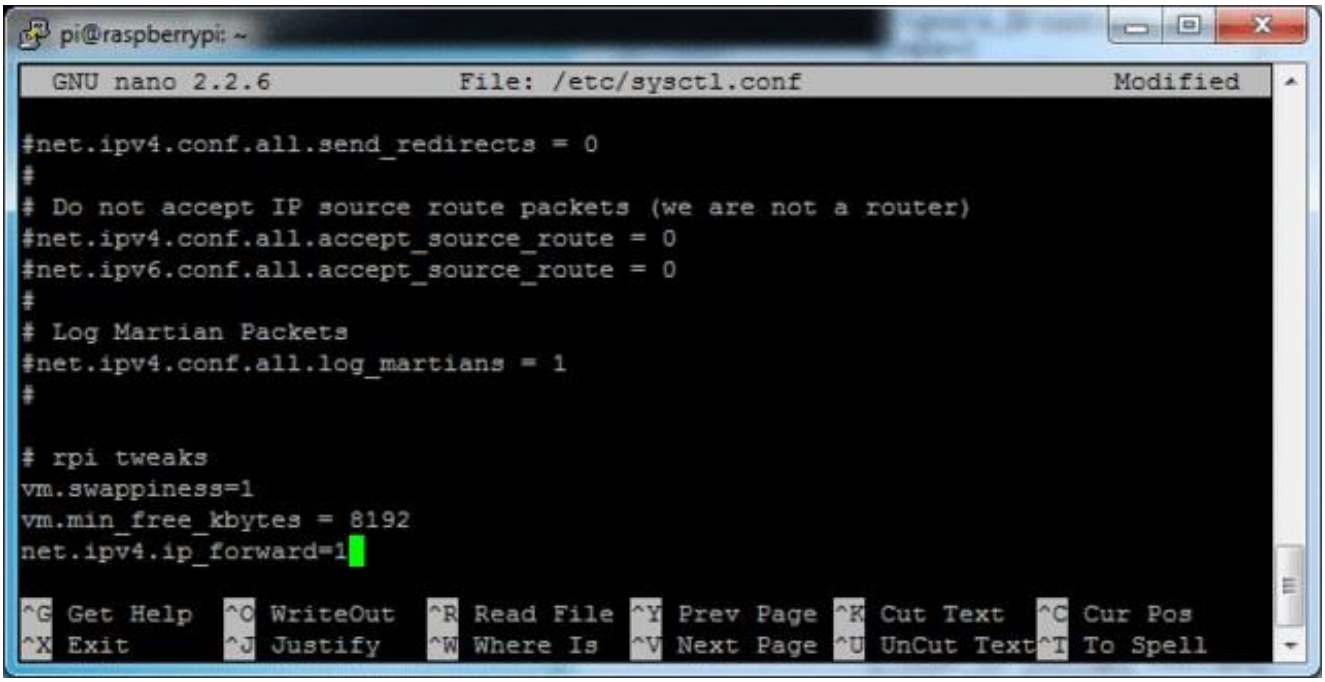
Setting up NAT will allow multiple clients to connect to the WiFi and have all the data 'tunneled' through the single WI-FI IP. (But you should do it even if only one client is going to connect). Run

sudo nano /etc/sysctl.conf

Scroll to the bottom and add

net.ipv4.ip_forward=1

on a new line. Save the file. This will start IP forwarding on boot up



Also run the following code to activate the IP forwarding immediately.

sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward "

Run the following commands to create the network translation between the Wi-FI dongle port
Wlan1 and the onboard wifi port wlan0

sudo iptables -t nat -A POSTROUTING -o wlan1 -j MASQUERADE

sudo iptables -A FORWARD -i wlan1 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT

sudo iptables -A FORWARD -i wlan0 -o wlan1 -j ACCEPT

You can check to see what's in the tables with

sudo iptables -t nat -S sudo iptables -S

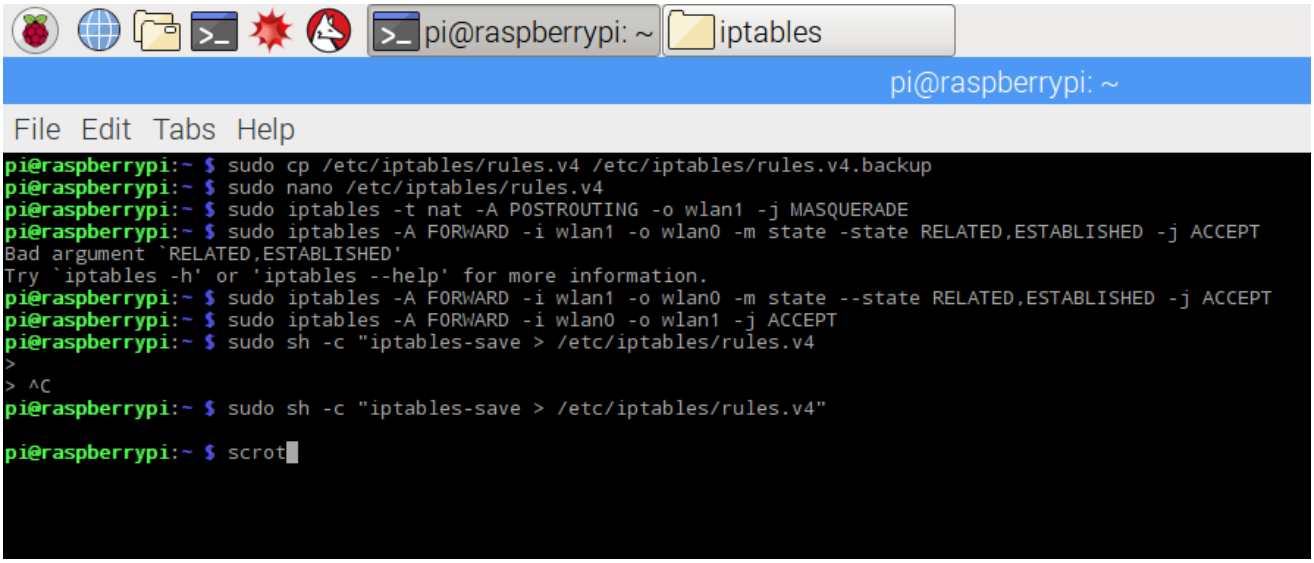
To make this happen on reboot (so you don't have to type it every time) run

sudo sh -c "iptables-save > /etc/iptables/rules.v4"

sudo nano /etc/rc.local

add the following lines

iptables-restore < /etc/iptables/rules.v4



The iptables-persistent tool you installed at the beginning will automagically reload the configuration on boot for you.

Step 10: Finishing up

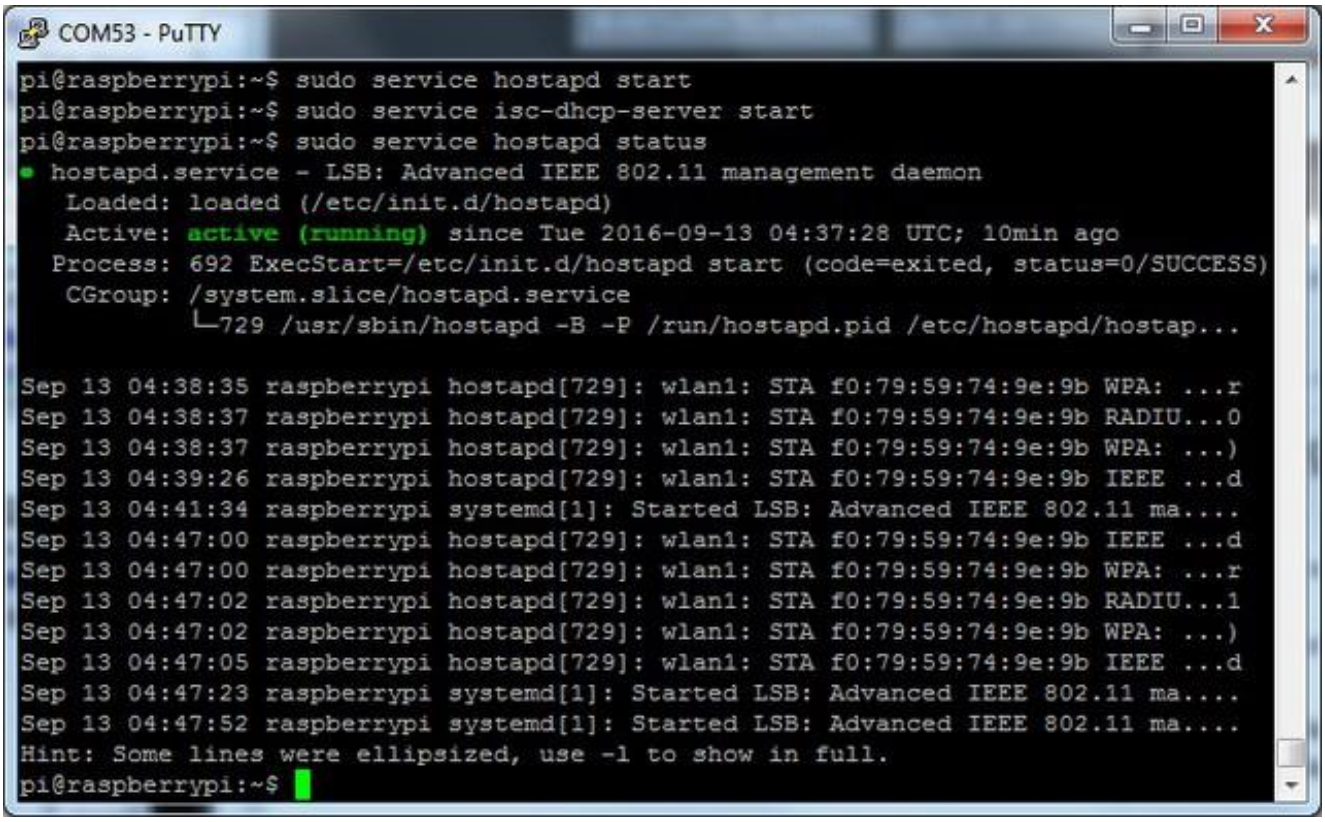
Now that we know it works, time to set it up as a 'daemon' - a program that will start when the Pi boots. Run the following commands

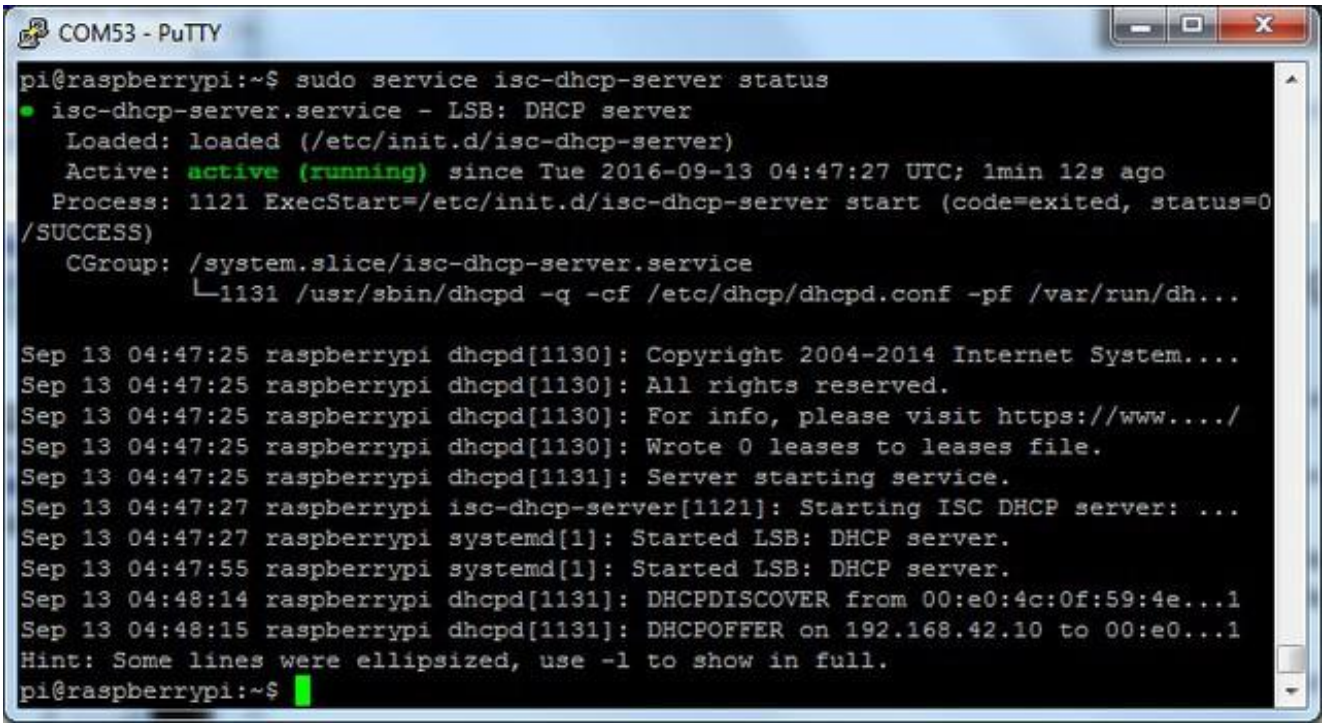
sudo service hostapd start
sudo service isc-dhcp-server start
sudo service dnsmasq start

you can always check the status of the host AP server and the DHCP server with

sudo service hostapd status

sudo service isc-dhcp-server status


COM53 - PuTTY
pi@raspberrypi:~$ sudo service hostapd start
pi@raspberrypi:~$ sudo service isc-dhcp-server start
pi@raspberrypi:~$ sudo service hostapd status
● hostapd.service - LSB: Advanced IEEE 802.11 management daemon
Loaded: loaded (/etc/init.d/hostapd)
Active: active (running) since Tue 2016-09-13 04:37:28 UTC; 10min ago
Process: 692 ExecStart=/etc/init.d/hostapd start (code=exited, status=0/SUCCESS)
CGroup: /system.slice/hostapd.service
└729 /usr/sbin/hostapd -B -P /run/hostapd.pid /etc/hostapd/hostap...
Sep 13 04:38:35 raspberrypi hostapd[729]: wlan1: STA f0:79:59:74:9e:9b WPA: ...r
Sep 13 04:38:37 raspberrypi hostapd[729]: wlan1: STA f0:79:59:74:9e:9b RADIU...0
Sep 13 04:38:37 raspberrypi hostapd[729]: wlan1: STA f0:79:59:74:9e:9b WPA: ...)
Sep 13 04:39:26 raspberrypi hostapd[729]: wlan1: STA f0:79:59:74:9e:9b IEEE ...d
Sep 13 04:41:34 raspberrypi systemd[1]: Started LSB: Advanced IEEE 802.11 ma....
Sep 13 04:47:00 raspberrypi hostapd[729]: wlan1: STA f0:79:59:74:9e:9b IEEE ...d
Sep 13 04:47:00 raspberrypi hostapd[729]: wlan1: STA f0:79:59:74:9e:9b WPA: ...r
Sep 13 04:47:02 raspberrypi hostapd[729]: wlan1: STA f0:79:59:74:9e:9b RADIU...1
Sep 13 04:47:02 raspberrypi hostapd[729]: wlan1: STA f0:79:59:74:9e:9b WPA: ...)
Sep 13 04:47:05 raspberrypi hostapd[729]: wlan1: STA f0:79:59:74:9e:9b IEEE ...d
Sep 13 04:47:23 raspberrypi systemd[1]: Started LSB: Advanced IEEE 802.11 ma....
Sep 13 04:47:52 raspberrypi systemd[1]: Started LSB: Advanced IEEE 802.11 ma....
Hint: Some lines were ellipsized, use -l to show in full.
pi@raspberrypi:~$


COM53 - PuTTY
pi@raspberrypi:~$ sudo service isc-dhcp-server status
● isc-dhcp-server.service - LSB: DHCP server
Loaded: loaded (/etc/init.d/isc-dhcp-server)
Active: active (running) since Tue 2016-09-13 04:47:27 UTC; 1min 12s ago
Process: 1121 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, status=0
/SUCCESS)
CGroup: /system.slice/isc-dhcp-server.service
└1131 /usr/sbin/dhcpd -q -cf /etc/dhcp/dhcpd.conf -pf /var/run/dh...
Sep 13 04:47:25 raspberrypi dhcpd[1130]: Copyright 2004-2014 Internet System....
Sep 13 04:47:25 raspberrypi dhcpd[1130]: All rights reserved.
Sep 13 04:47:25 raspberrypi dhcpd[1130]: For info, please visit https://www..../
Sep 13 04:47:25 raspberrypi dhcpd[1130]: Wrote 0 leases to leases file.
Sep 13 04:47:25 raspberrypi dhcpd[1131]: Server starting service.
Sep 13 04:47:27 raspberrypi isc-dhcp-server[1121]: Starting ISC DHCP server: ...
Sep 13 04:47:27 raspberrypi systemd[1]: Started LSB: DHCP server.
Sep 13 04:47:55 raspberrypi systemd[1]: Started LSB: DHCP server.
Sep 13 04:48:14 raspberrypi dhcpd[1131]: DHCPDISCOVER from 00:e0:4c:0f:59:4e...1
Sep 13 04:48:15 raspberrypi dhcpd[1131]: DHCPOFFER on 192.168.42.10 to 00:e0...1
Hint: Some lines were ellipsized, use -l to show in full.
pi@raspberrypi:~$

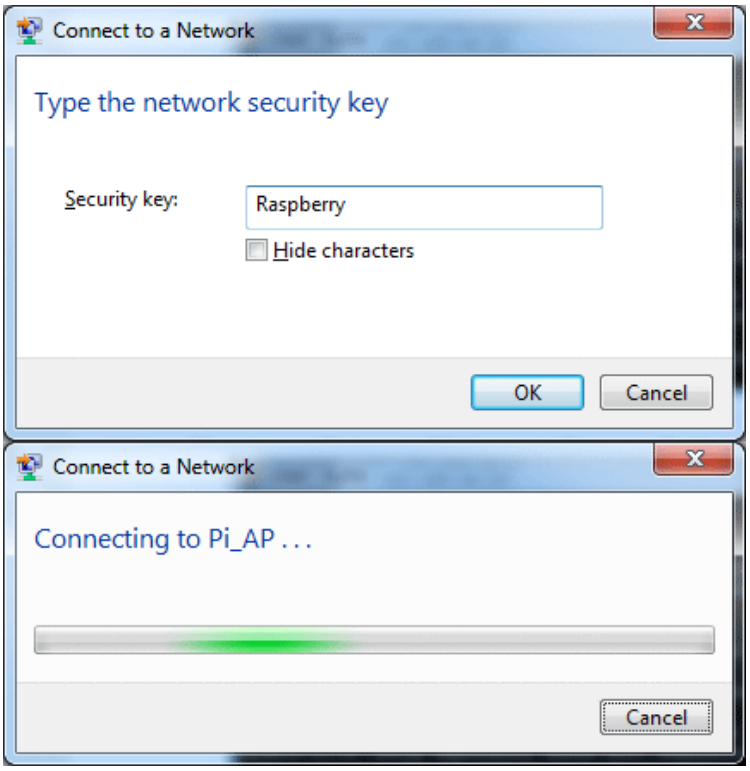
Step 10: Connect and Test

Now that we have the software installed on a Pi, it's time to connect to it and test the connection. I'm using a Windows computer but any kind should work fine. On the Pi, run the command

tail -f /var/log/syslog

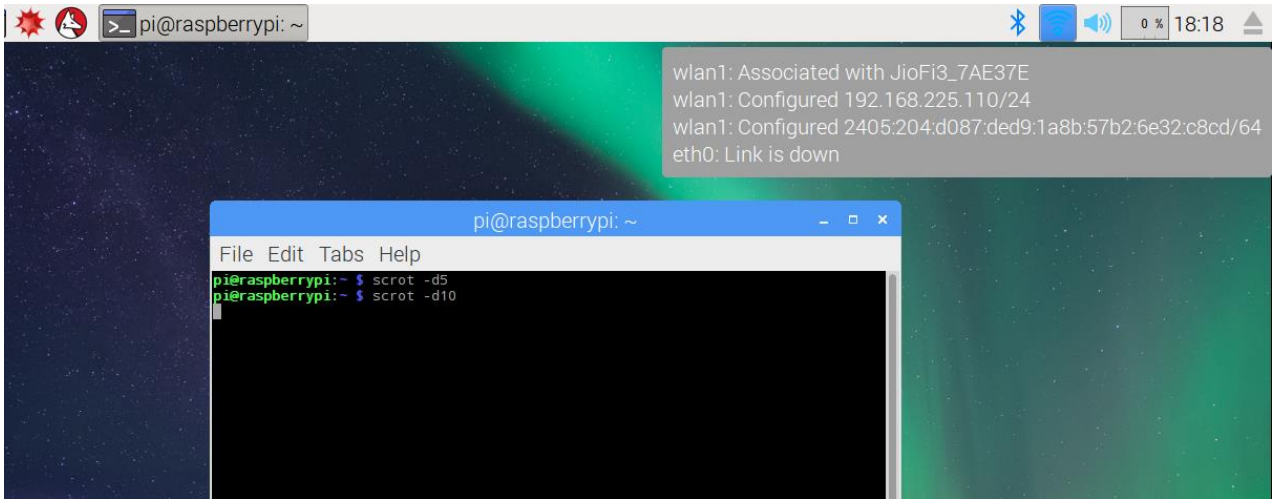
to watch the system log data, handy for checking and debugging whats going on. connect with another computer to the AP you made in the previous step

Enter the WPA key you specified in the previous step



In the Pi syslog you should see stuff like this! It indicates that a client connected, at what time and what IP address was given to them. If you can't connect at all, something is wrong with hostapd.

```
pi@raspb... ✖  pi@raspb... ✖
 or directory.
pi@raspberrypi:~ $ sudo service dhcpcd restart
pi@raspberrypi:~ $ sudo ifdown wlan0;sudo ifup wlan0
pi@raspberrypi:~ $ sudo nano /etc/hostapd/hostapd.conf
pi@raspberrypi:~ $ sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
Configuration file: /etc/hostapd/hostapd.conf
Line 2: invalid/unknown driver 'n180211'
1 errors found in configuration file '/etc/hostapd/hostapd.conf'
Failed to set up interface with /etc/hostapd/hostapd.conf
Failed to initialize interface
pi@raspberrypi:~ $ sudo nano /etc/hostapd/hostapd.conf
pi@raspberrypi:~ $ sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
Configuration file: /etc/hostapd/hostapd.conf
Failed to create interface mon.wlan0: -95 (Operation not supported)
wlan0: Could not connect to kernel driver
Using interface wlan0 with hwaddr b8:27:eb:40:4e:b5 and ssid "pi3wifi"
pi@raspberrypi:~ $ ate UNINITIALIZED->ENABLED
pi@raspberrypi:~ $ sudo nano /etc/sysctl.conf
pi@raspberrypi:~ $ sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
pi@raspberrypi:~ $ sudo iptables -t nat -A POSTROUTING -o et0
pi@raspberrypi:~ $ sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
pi@raspberrypi:~ $ sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT
pi@raspberrypi:~ $ sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
pi@raspberrypi:~ $ sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
pi@raspberrypi:~ $ sudo nano /etc/rc.locl
pi@raspberrypi:~ $ sudo nano /etc/rc.local
pi@raspberrypi:~ $ scrot
pi@raspberrypi:~ $ sudo service hostapd start
pi@raspberrypi:~ $ sudo service dnsmasq start
pi@raspberrypi:~ $ sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
Configuration file: /etc/hostapd/hostapd.conf
Failed to create interface mon.wlan0: -95 (Operation not supported)
wlan0: Could not connect to kernel driver
Using interface wlan0 with hwaddr b8:27:eb:40:4e:b5 and ssid "pi3wifi"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
wlan0: STA fc:64:ba:1b:5d:0f IEEE 802.11: associated
wlan0: AP-STA-CONNECTED fc:64:ba:1b:5d:0f
wlan0: STA fc:64:ba:1b:5d:0f RADIUS: starting accounting session 58CE4433-00000000
wlan0: STA fc:64:ba:1b:5d:0f WPA: pairwise key handshake completed (RSN)
```



If everything runs fine, you are able to connect to the Raspberry pi Wi-Fi access point. Here the USB dongle (wlan1) act as client and onboard Wi-Fi chip (wlan0) act as AP. If we need to increase the range of network we can use sophisticated Wi-Fi dongles and route the data in between as explained the above steps.

CONCLUSION

In this project it is seen that how to implement a cheap, reliable emergency network with minimum complexity and set up time. This temporary network established in environments where natural disasters like earthquake, flood or fire blackouts the current communication systems in that area will have a huge impact on rescue processes and number of survivors. Instead of implementing a stationary ground based communication station, a floating base station is introduced for establishing emergency wireless network. Floating in the sense that the network providing station is not grounded but airborne so that it is more manoeuvrable and can be used over harsh terrains where it is difficult or practically impossible to reach and establish ground station based communication. So for this particular purpose quadcopter or drone are introduced.

A quadcopter is used as the floating base station in the air which is integrated with readily available and widely used Wi-Fi communication protocol so that devices/people in the ground can easily and quickly connect to it and establish communication and without any fuss. Also there is no need for special radio receivers if other military or satellite radio links are used. The drone used can balance itself while flying and equipped with GPS based navigation system. So a group of drones providing wireless network over a wide area prevent the total communication blackout and enables the affected people to call for help. It is much more flexible in a way that they can also focus on specific area according to the requirements in the field. Raspberry pi is used to establish the Wi-Fi network which act as both client and access point. The client connects to far away non affected ground station and the access point enables other devices and drones to connect to it forming an Ad-Hoc network.

This system can also be used to provide networks for military personal involved in tactical mission in the battlefield, providing communication in rural areas, in dense forest and

harsh environments where scientific expedition and research occurs, etc. This system has very good potential in the sense that it can be modified according to specific requirements in the field. Adding more features like human detection, thermal sensors, obstacle avoidance autonomous navigation and flight can yield a sophisticated system for emergency rescue missions. That is instead of manually operating the drone, If the location co-ordinates of the affected area is updated to the drone via satellite or any other means, it will automatically travels to that location and establish communication and take part in rescue. If solar power source is used, the drone can stay in air for longer time.

REFERENCE

[1] Pengcheng Wang, Zhihong Man, Zhenwel Cao, Yong Zhao, “Dynamics modelling and linear control of quadcopter” , 2016 international conference on advanced mechatronic system (ICAMechs), Pages 498-503 , 2016.

[2] Leandro Lima Gomes, Lucas Pires Leal, Tiago Roux Oliveira, Jose Paulo Vilela, “Unmanned Quadcopter Control using a motion capture system”, IEEE Latin America Transactions, Volume 14, issue 8, Pages 3606-3613, 2016

[3] Joao Gutemberg B, Farias Filho,Carlos E. T. Dorea, Wallace M. Bessa, “Modelling, Test Benches and Identification of a Quadcopter” , IEEE conference publications (LARS/SBR), Pages 49- 54, Year 2016.

[4] He Luo, Yanqiu Niu, Zhengzhenng Liang, Xiang Fang, “Quadcopter autonomous control system based on image recognition” 12th World congress on intelligent control and automation (WCICA), Pages 2237-2240, Year 2016

[5] [Online].Available: https://github.com/

[6] [Online].Available: https://en.m.Wikipedia.org/

[7] [Online].Available:https://learn.adafruit.com/setting-up-a-raspberry-pi-as-a-wifi-accesspoint

[8] [Online].Available:https://frillip.com/using-your-raspberry-pi3-as-a-wifi-access-point-with-hostapd

[9] [Online].Available: https://www.raspberrypi.org/documentation

[10] [Online].Available: https://github.com/

[11][Online].Available:https://www.dronetrest.com/t/beginners-guide-to-drone-autopilots-flight-controllers-and-how-they-work