**Setting up a Raspberry Pi as a WiFi access point**

**Step 1: Setting up WiFi connection**

Start by booting the Raspberry Pi, connected to a display and a keyboard. Open up the terminal and edit the network interfaces file:

$ sudo nano /etc/network/interfaces

This file contains all known network interfaces, it'll probably have a line or two in there already.

Change the first line (or add it if it's not there) to:

auto wlan0

Then at the bottom of the file, add these lines telling the Raspberry Pi to allow wlan as a network connection method and use the/etc/wpa\_supplicant/wpa\_supplicant.conf as your configuration file.

allow-hotplug wlan0  
iface wlan0 inet dhcp  
wpa-conf /etc/wpa\_supplicant/wpa\_supplicant.conf  
iface default inet dhcp

*(ctrl-X, then type Y to quit and save)*

The next step is to create this configuration file.

**Step 2: Configuring WiFi connection**

Open up the wpa\_supplicant.conf file in the editor.

$ sudo nano /etc/wpa\_supplicant/wpa\_supplicant.conf

Again, some lines might already be present, just add the following.

network={  
ssid="mobo-2.5"  
psk="MDOppurtunity1@34"  
proto=RSN  
key\_mgmt=WPA-PSK  
pairwise=CCMP  
auth\_alg=OPEN  
}

The other parameters are network specific, I can't tell you what you need. If you boot Raspbian to desktop, you can launc the wpa\_gui (WiFi config) application and click 'Scan'. You'll find a list that has your network too with all flags you need. To do this on a RPi A you'll have to disconnect your keyboard and connect your dongle once the scanning list is open.

**Make sure it works**

Reboot the Raspberry Pi and it should connect to the wireless network. Use the bellow command for make shore.

$ ifconfig

The out put will be looks bellow but as wlan0

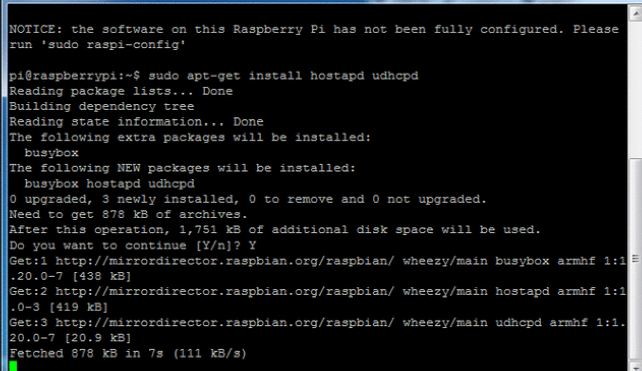


**Step 3: Install software**

Now we install the software 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**

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



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

(text above shows udhcpd but that doesnt work as well as isc-dhcp-server, still, the output should look similar)

**Step 4: 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.

Run this command to edit the file

sudo nano /etc/dhcp/dhcpd.conf

Find the lines that say

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;

Then scroll down to the bottom and add the following lines

subnet 192.168.42.0 netmask 255.255.255.0 {

range 192.168.42.10 192.168.42.50;

option broadcast-address 192.168.42.255;

option routers 192.168.42.1;

default-lease-time 600;

max-lease-time 7200;

option domain-name "local";

option domain-name-servers 8.8.8.8, 8.8.4.4;

}

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

Run

**sudo nano /etc/default/isc-dhcp-server**

and scroll down to **INTERFACES=""** and update it to say **INTERFACES="wlan0"**

close and save the file

**Step 5: Set up wlan0 for static IP**

If you happen to have wlan0 active because you set it up, run **sudo ifdown wlan0**

There's no harm in running it if you're not sure

Next we will set up the **wlan0** connection to be static and incoming. run **sudo nano**

**/etc/network/interfaces** to edit the file

Find the line **auto wlan0** and add a **#** in front of the line, and in front of every line afterwards. If you don't have that line, just make sure it looks like the screenshot below in the end! Basically just remove any old **wlan0** configuration settings, we'll be changing them up

Depending on your existing setup/distribution there might be more or less text and it may vary a little bit Add the lines

iface wlan0 inet static

address 192.168.42.1

netmask 255.255.255.0

After **allow-hotplug wlan0** - see below for an example of what it should look like. Any other lines afterwards should have a **#** in front to disable them

Save the file (Control-X Y <return>)

Assign a static IP address to the wifi adapter by running

sudo ifconfig wlan0 192.168.42.1

**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 broadcast name. The password can be changed with the text after **wpa\_passphrase=**

interface=wlan0

driver=rtl871xdrv

ssid=RPi\_WIFI

hw\_mode=g

channel=6

macaddr\_acl=0

auth\_algs=1

ignore\_broadcast\_ssid=0

wpa=2

wpa\_passphrase=Raspberry

wpa\_key\_mgmt=WPA-PSK

wpa\_pairwise=TKIP

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.

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. 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 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 Ethernet 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

**sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip\_forward**"

to activate it immediately

Run the following commands to create the network translation between the ethernet port **eth0** and the wifi port **wlan0**

sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT

sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT

You can check to see whats 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.ipv4.nat"**

sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

run **sudo nano /etc/network/interfaces** and add

up iptables-restore < /etc/iptables.ipv4.nat

to the very end Update hostapd

Before we can run the access point software, we have to update it to a version that supports the WiFi adapter.

First get the new version by typing in

**wget http://adafruit-download.s3.amazonaws.com/adafruit\_hostapd\_14128.zip**

to download the new version (check the next section for how to compile your own updated

**hostapd**) then

**unzip adafruit\_hostapd\_14128.zip**

to uncompress it. Move the old version out of the way with

**sudo mv /usr/sbin/hostapd /usr/sbin/hostapd.ORIG**

And move the new version back with

**sudo mv hostapd /usr/sbin**

set it up so its valid to run with

**sudo chmod 755 /usr/sbin/hostapd**

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.

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 Ethernet connection yet.

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

OK now that we know it works, time to set it up as a 'daemon' - a program that will start when the RPi boots.

Run the following commands

**sudo service hostapd start**

**sudo service isc-dhcp-server 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**

To start the daemon services. Verify that they both start successfully (no 'failure' or 'errors')

Then to make it so it runs every time on boot

**sudo update-rc.d hostapd enable**

**sudo update-rc.d isc-dhcp-server enable**

Then rebooting (**sudo reboot**)

Finished…..