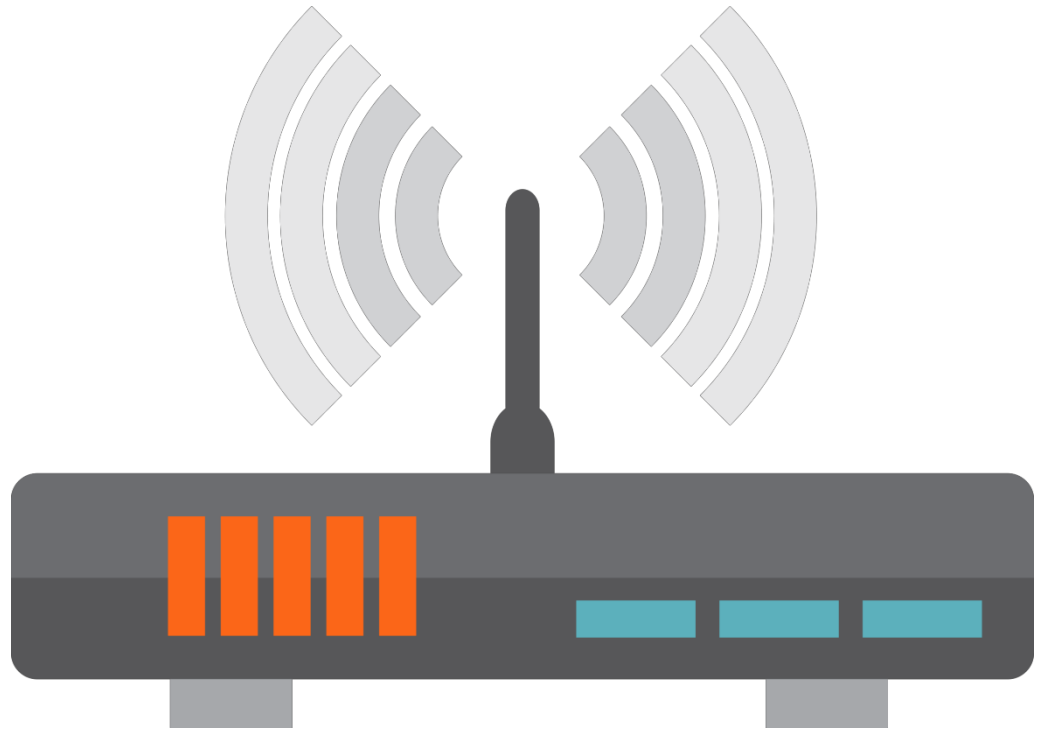
Project: Building a Wireless Access Point

Module Overview

- Real Life Project for your DIY Shed
- Raspberry Pi Wireless Access Point
- Raspberry Pi Model B or B+
- Ralink Technology, Corp. RT5370 Wireless Adapter

WLAN
192.168.1.0/24



Corporate Backbone
192.168.0.0/24

wlan0
192.168.1.1/24

eth0
DHCP

Wireless NIC

- The NIC requires support nl80211driver to work without modification to the hostapd service (WAP)
- The chipset **RT5370** is supported
- Prefer external antenna

```
$ lsusb
Bus 001 Device 002: ID 0424:9512 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 004: ID 148f:5370 Ralink Technology, Corp. RT5370 Wireless Adapter
```

Discovering your Wireless Card

The command **lsusb** can list the content of the USB Bus

```
$ sudo vi /etc/network/interfaces
auto wlan0
iface wlan0 inet static
 address 192.168.1.1
 netmask 255.255.255.0
```

Configuring a Static IP Address for wlan0

Network configuration is stored on a Debian based system in /etc/network/interfaces

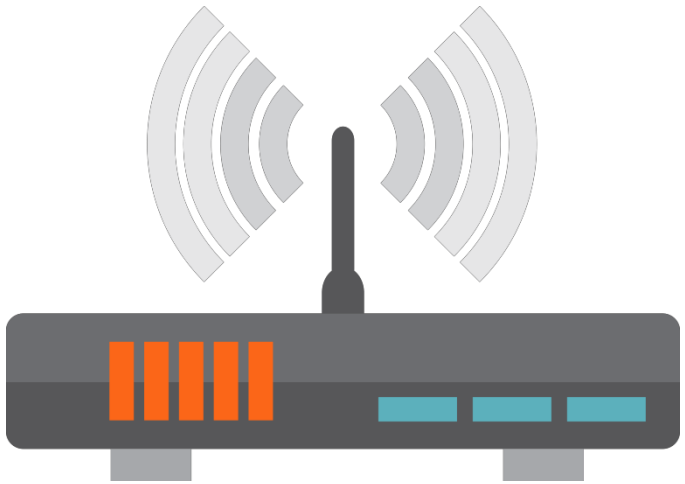
Lines starting with auto will bring the interface up at boot time or ifup -a

The line for wpa-roam can be deleted or commented as it is not required for WAP

```
$ sudo ifdown wlan0
$ sudo ifup wlan0
$ ip address show wlan0
```

Restart the wlan0 interface

You may also like to reboot the Pi to ensure the interface is brought up correctly at boot time



Demo: Static Address wlan0

```
$ sudo vi /etc/sysctl.conf
# Uncomment the next line to enable packet forwarding for
IPv4
net.ipv4.ip_forward=1
$ sudo sysctl -p
```

Enabling routing on the Pi

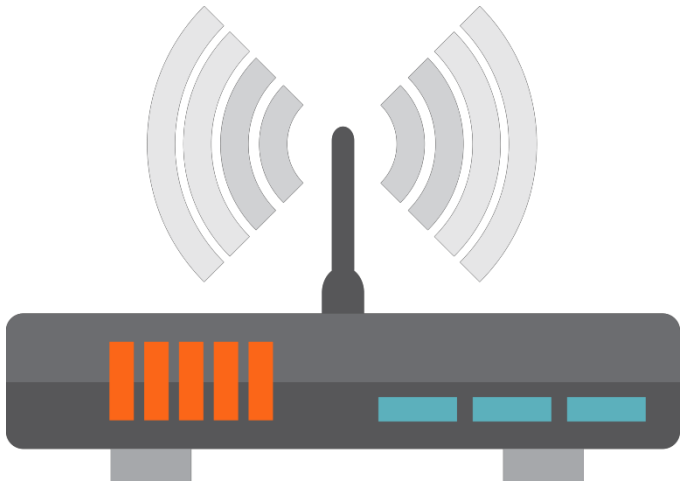
To enable routing uncomment the line that sets **ip_forward**

Load those setting with **sysctl**

```
$ sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
$ sudo bash –c “iptables-save > /etc/iptables.nat”
$ sudo vi /etc/network/interfaces
up iptables-restore < /etc/iptables.nat
```

Enable NAT on the Pi

Network Address Translation will allow access to network and internet resources from the private address range of the WiFi



Demo: Configure Routing and NAT

```
$ sudo apt-get install isc-dhcp-server
$ sudo vi /etc/default/isc-dhcp-server
INTERFACES="wlan0“
$ sudo sed -i.bak ‘/^#/d;/^$/d’ /etc/dhcp/dhcpd.conf
```

Install the DHCP Server on the Pi

The DHCP Server will issue IP Addresses from the WAP

Make sure the **INTERFACES** directive points to **wlan0**

Backup and clean the dhcpd.conf

/etc/dhcp/dhcpd.conf

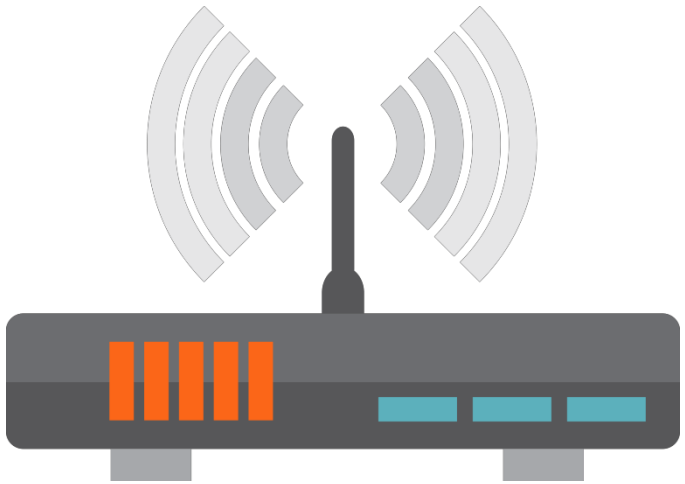
```
$ sudo vi /etc/dhcp/dhcpd.conf
ddns-update-style none;
option domain-name “example.com";
option domain-name-servers 192.168.0.3, 8.8.8.8;
default-lease-time 604800;
max-lease-time 604800;
log-facility local7;
subnet 192.168.1.0 netmask 255.255.255.0 {
        range 192.168.1.10 192.168.1.254;
        option broadcast-address 192.168.1.255;
        option routers 192.168.1.1;
}
```

```
$ sudo service isc-dhcp-server start
$ sudo update-rc.d isc-dhcp-server defaults
```

Start the DHCP Server

We can now start the DHCP service

Additionally we will set DHCP to auto start



Demo: Configure DHCP

```
$ sudo apt-get install hostapd
$ sudo vi /etc/init.d/hostapd
DAEMON_CONF=/etc/hostapd/hostapd.conf
```

Installing the WAP Service

We install the hostapd service which will act as the WAP

Within the service init script we need to point to the main configuration file

Configure the WAP

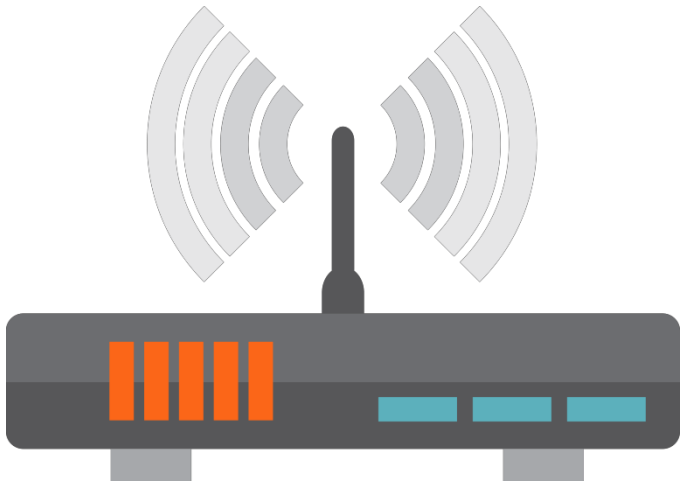
```
$ sudo vi /etc/hostapd/hostapd.conf
interface=wlan0
ssid=WIFI_ACCESS
channel=6
hw_mode=g
wpa=2
wpa_passphrase=Password1
wpa_key_mgmt=WPA-PSK WPA-EAP WPA-PSK-SHA256 WPA-EAP-SHA256
```

```
$ sudo service hostapd start
$ sudo update-rc.d hostapd defaults
$ sudo hostapd /etc/hostapd/hostapd.conf
```

Starting the WAP

We can now start the WAP and set for auto-start

If it does not start troubleshoot by starting the service manually



Demo: Configure WAP

Summary

- Configure Raspberry Pi WAP
- DHCP Server
- IP Routing and NAT
- WAP service from hostapd