## LABORATORY EXTRAS: OPENWRT OS AND APPLICATIONS

**Learning Outcomes**

By the end of this laboratory, student should be able to

* Write OPENWRT OS to SD Card
* Set USB interface for 5G RPI CPE
* Set up 5G RPI CPE as a router

**Activities**

* Write OPENWRT OS to SD Card
* Set USB interface for 5G RPI CPE
* Set up 5G RPI CPE as a router

**Equipment**

* Raspberry Pi 4B
* RM502Q-AE 5G Har
* SD Card
* Windows laptop

# Introduction

The OpenWrt Project is a Linux operating system targeting embedded devices. Instead of trying to create a single, static firmware, OpenWrt provides a fully writable filesystem with package management. This frees you from the application selection and configuration provided by the vendor and allows you to customize the device through the use of packages to suit any application. For developers, OpenWrt is the framework to build an application without having to build a complete firmware around it; for users this means the ability for full customization, to use the device in ways never envisioned.

People install OpenWrt because they believe it works better than the stock firmware from their vendor. They find it is more stable, offers more features, is more secure and has better support.

* **Extensibility:** OpenWrt provides many capabilities found only in high-end devices. Its 3000+ application packages are standardized, so you can easily replicate the same setup on any supported device, including two (or even five) year old routers. [More...](https://openwrt.org/reasons_to_use_openwrt#extensibility)
* **Security:** OpenWrt's standard installation is secure by default, with Wi-Fi disabled, no poor passwords or backdoors. OpenWrt's software components are kept up-to-date, so [vulnerabilities get closed shortly after they are discovered](https://openwrt.org/advisory/start). [More...](https://openwrt.org/reasons_to_use_openwrt#security)
* **Performance and Stability:** OpenWrt firmware is made of standardized modules used in all supported devices. This means each module will likely receive more testing and bug fixing than stock firmware which can be tweaked for each product line and never touched again. [More...](https://openwrt.org/reasons_to_use_openwrt#performance_stability)
* **Strong Community Support:** OpenWrt team members are regular participants on the [OpenWrt Forum](https://forum.openwrt.org/), [OpenWrt Developer](https://lists.openwrt.org/mailman/listinfo/openwrt-devel) and [OpenWrt Admin](https://lists.openwrt.org/mailman/listinfo/openwrt-adm) mailing lists, and [LEDE's IRC channels.](https://openwrt.org/contact#irc_channels) You can interact directly with developers, volunteers managing the software modules and with other long-time OpenWrt users, drastically increasing the chances you will solve the issue at hand. [More...](https://openwrt.org/contact)
* **Research:** Many teams use OpenWrt as a platform for their research into network performance. This means that the improvements of their successful experiments will be available in OpenWrt first, well before it gets incorporated into mainline, vendor firmware. [More...](https://openwrt.org/reasons_to_use_openwrt#research_platform)
* **Open Source/No additional cost:** OpenWrt is provided without any monetary cost. It has been entirely created by a team of volunteers: developers and maintainers, individuals and companies. If you enjoy using OpenWrt, consider contributing some effort to [help us improve it for others!](https://openwrt.org/start#openwrt_wants_you) All of the above is possible because OpenWrt is part of the Open Source community, and powered by Linux kernel. [Get the source code...](https://git.openwrt.org/)

# Write OpenWrt OS into SD Card

1. Download openwrt image file from this link: <https://downloads.openwrt.org/releases/23.05.3/targets/bcm27xx/bcm2711/>

A screenshot of a computer

Description automatically generated

1. Insert a 32GB SD card into the SD Card reader, then plug it to your laptop. Format if needed.
2. Open Raspberry Pi imager, do the same as Lab 2 but when choosing OS, select ‘custom’. Choose the image file you downloaded.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Click ‘Write’, it will take a few seconds to complete.
2. Once done, take out the SD Card and insert into 5G RPI CPE.
3. Connect monitor, keyboard and mouse to raspberry pi and power it on.
4. You should be able to see an OpenWrt OS flashing on the monitor.
5. Press “Enter” once there are no more messages appearing.

# Login to OpenWrt’s WebGui Page (Router’s Login Page)

1. Use ethernet cable to connect the raspberry pi 4b to a laptop.
2. Search 192.168.1.1(default) on browser. Login as root and there is no password.

A screenshot of a log in

Description automatically generated

1. After login, you will see the WebGui page which is similar to a router’s login page. Select System > Administration. You can set password if you want.

A screenshot of a computer program

Description automatically generated

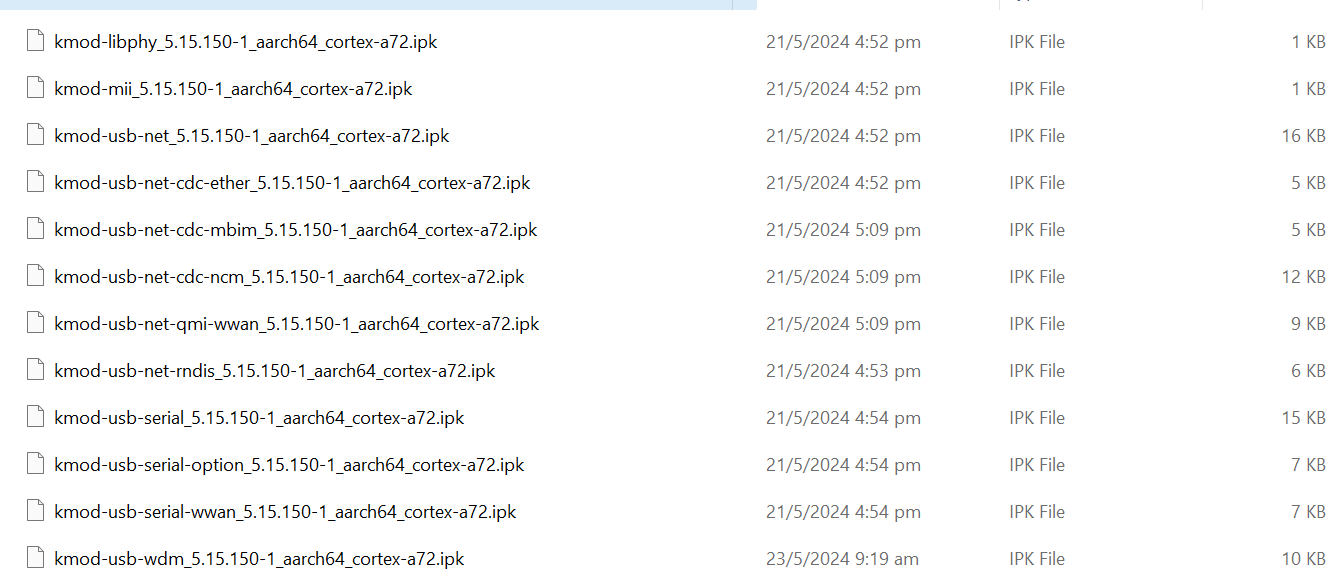
# Set up OpenWrt Web Gui

1. Go to ‘Network’ > ‘Interface’. Click ‘Add new interface’. Is there a interface called usb0 or wwan0? If not, that means that you will need to install all the necessary drivers needed to support Quectel RM502Q-AE. This can be done by installing all the necessary kmod packages which you can find from this link: <https://downloads.openwrt.org/releases/23.05.3/targets/bcm27xx/bcm2711/kmods/5.15.150-1-bd9c06a169c49b7613087c1d65b97108/>

A screenshot of a computer

Description automatically generated

1. You will need to download the following kmod packages. Tip: **Ctrl+F** to find the kmod packages that you need.



1. After you downloaded on your laptop, you will need to transfer to the raspberry pi. Download WinSCP from this link: <https://winscp.net/eng/download.php>

A screenshot of a computer error

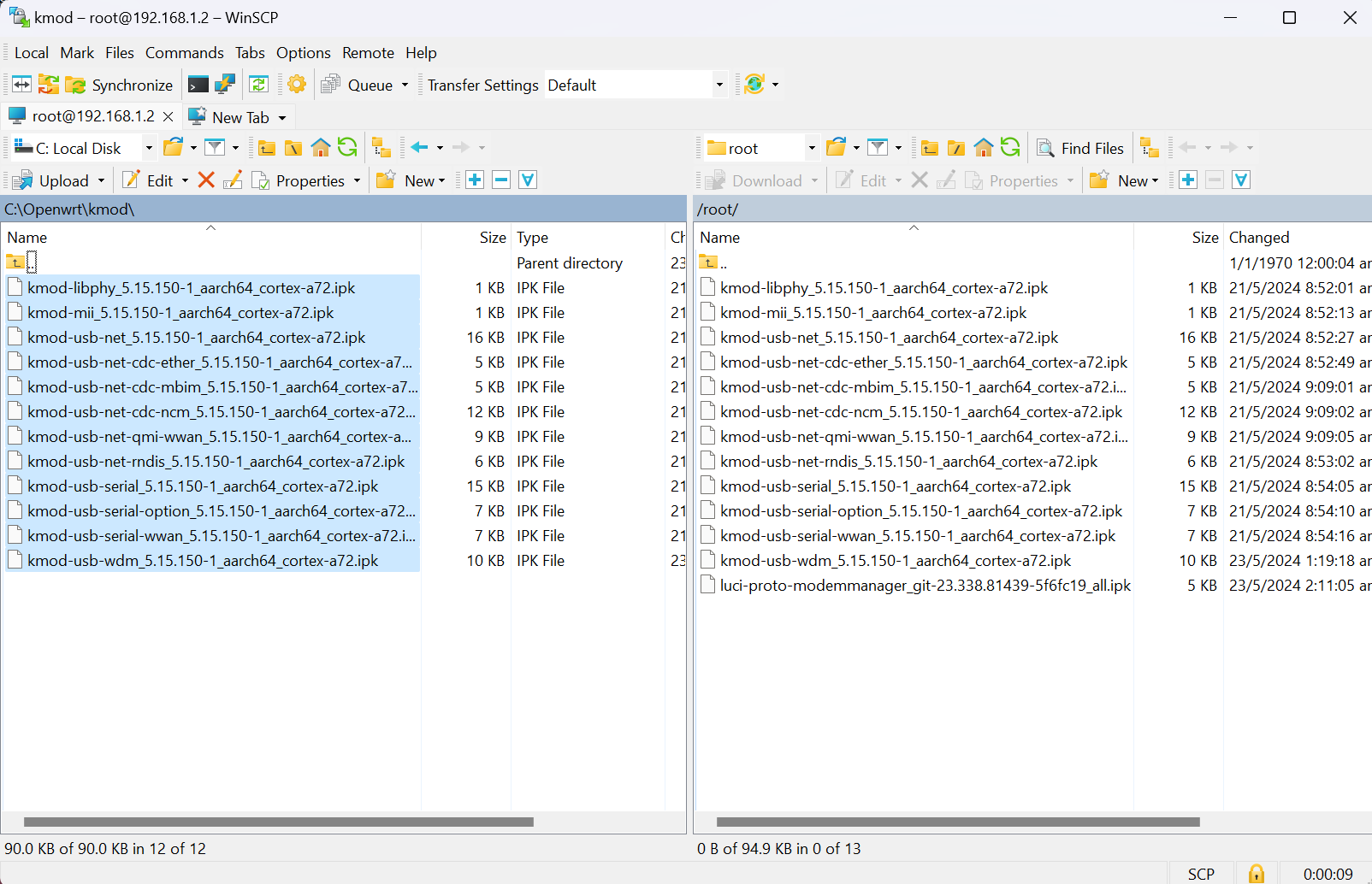
Description automatically generated

1. Click “WinSCP”, select ‘New Site’. Choose ‘File protocol’ 🡪 SCP, ‘Host name’🡪 192.168.1.1, ’Port number’ 🡪 22, ‘Username’🡪 root, ’Password’🡪 your password. Then, click ‘Login’.

A screenshot of a computer

Description automatically generated

1. Drag all the kmod packages from your laptop to the root directory of the raspberry pi.



1. SSH into the terminal of the raspberry pi running openwrt via VScode or PuTTY. Once you are inside the terminal, use ‘opkg install’ to install each kmod packages e.g. opkg install kmod-libphy\_5.15.150-1\_aarch64\_cortex-a72.ipk. Before you try to install, you must ensure that the kmod packages are already inside the raspberry pi root directory. Sometimes, some kmod packages will fail to install because you need to download other kmod packages first. Simply ignore the failed kmod packages and move to the next one. Then, install the failed kmod packages again once every other kmod packages are installed.

A screenshot of a computer

Description automatically generated

1. After all the kmod packages are installed, go back to ‘Network’ > ‘Interface’ > ‘Add new interface’. When you select ‘Device’, there should be either a ‘usb0’ or ‘wwan0’ interface option. If you saw ‘wwan0’ interface only, disconnect the 5G Hat from the Raspberry Pi, connect the 5G Hat to your laptop. Use QCOM or pyserial script to switch to ECM mode. Input these 2 AT commands:

**AT+QCFG=”usbnet”,1**

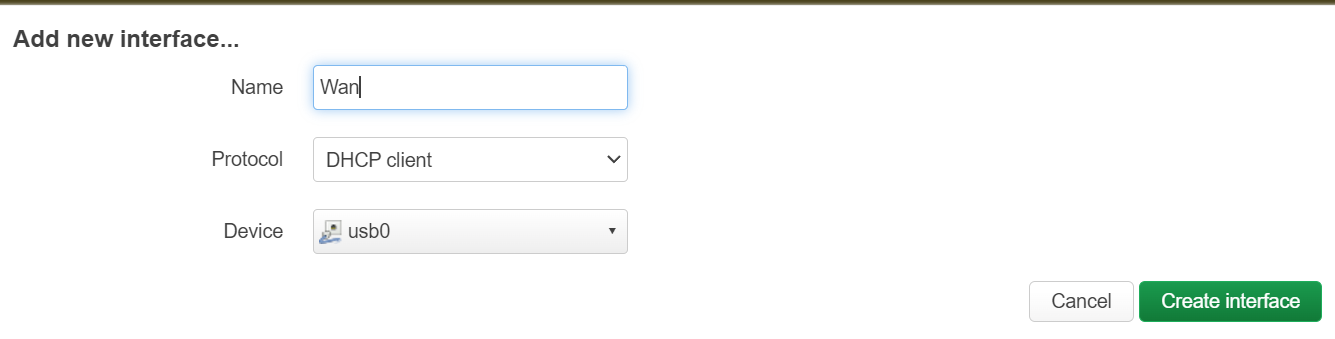
**AT+CFUN=1,1**

After it is done, connect back and go to the WebGUI again to see ‘usb0’ interface. If there is still no ‘usb0’ interface, that means you did not install the kmod packages for ECM driver which is **kmod-usb-net-cdc-ether\_5.15.150-1\_aarch64\_cortex-a72.ipk**

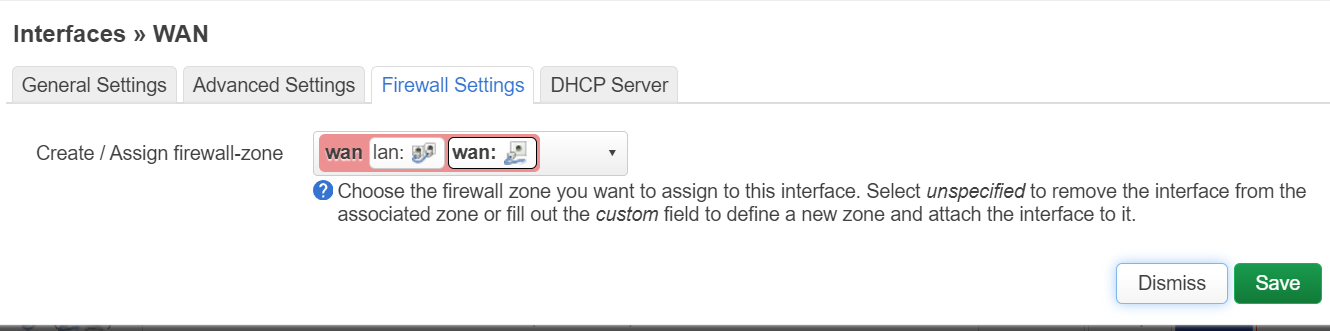
A screenshot of a computer

Description automatically generated

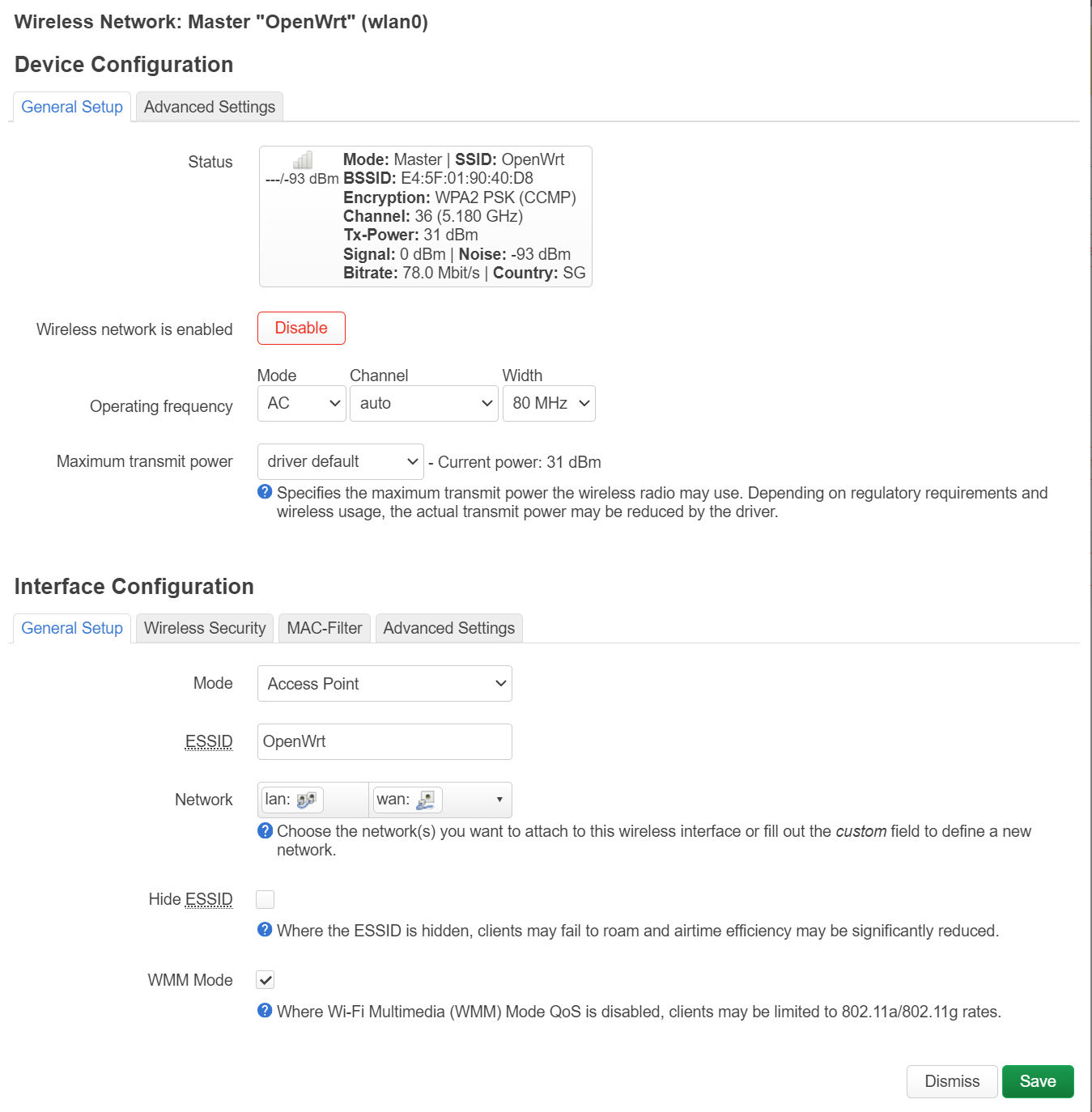
1. Click ‘Add in new interface’ to create a new interface. Choose ‘Name’🡪Wan, ‘Protocol’🡪 DHCP, ‘Device’🡪 usb0.



1. After the ‘Wan’ interface is created, edit that interface. Under ‘Firewall Settings’, assign wan interface to firewall zone.



1. Go to ‘Network’> ‘Wireless’. You may set a password for ‘Openwrt’ Wi-Fi if you want. Go to ‘Wireless Security’ and select ‘Encryption’🡪WPA2-PSK, ‘key’🡪your password. After you are done editing the interface, save and apply.



A screenshot of a computer

Description automatically generated

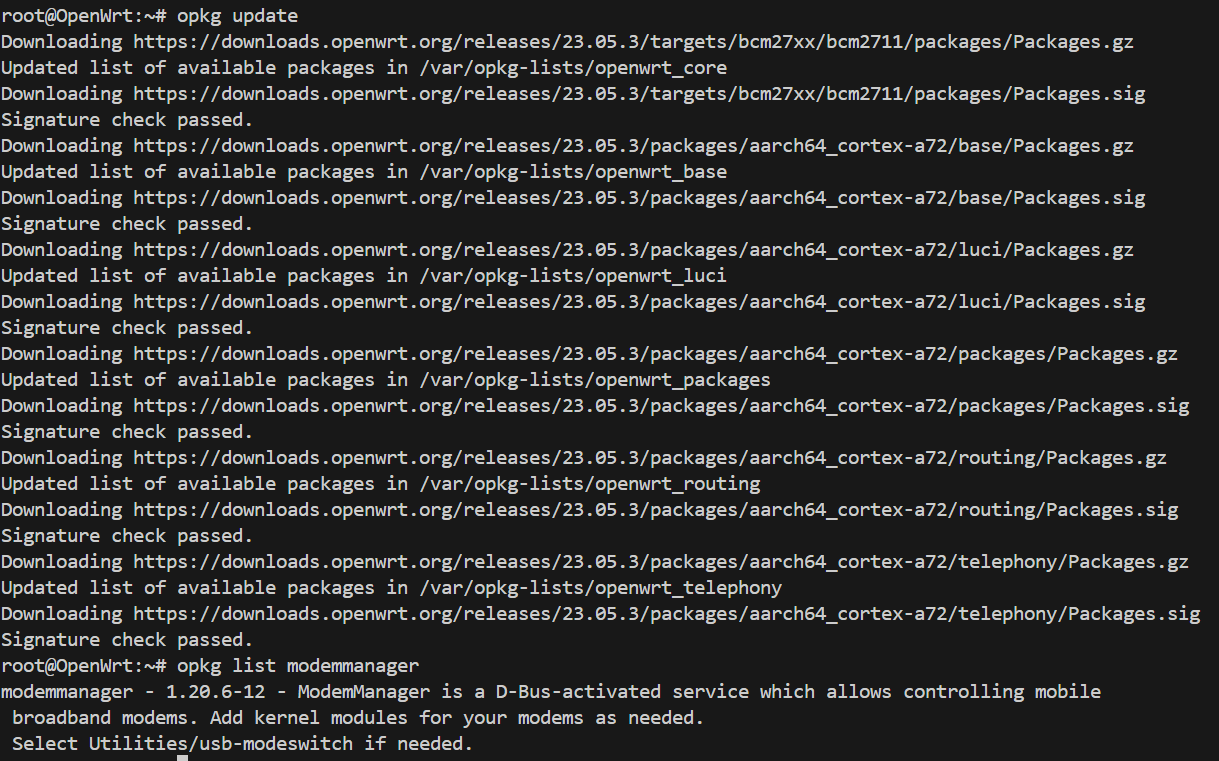
1. Do a hard reboot (turn off/on power raspberry pi). If successful, your ‘Wan’ interface should be getting some traffic (RX/TX MB). You should be able to connect multiple devices to the ‘OpenWrt’ Wi-Fi Network and browse the internet.

A screenshot of a computer

Description automatically generated

1. In the picture above, you can that I created another interface called ‘MBIM’. This interface is an alternative to ‘Wan’ and uses the MBIM protocol instead of ECM to connect to 5G. To use MBIM, you must install the necessary kmod packages for MBIM using ‘opkg install’.
2. Furthermore, you must install the modem manager first by using these commands:
3. **opkg update**
4. **opkg install usbutils kmod-usb-net-qmi-wwan kmod-usb-serial-option luci-proto-modemmanager uhubctl socat coreutils-timeout iptables-mod-ipopt pservice**
5. **reboot**

After the modem manager is installed, switch to MBIM mode using AT commands but instead of disconnecting, you can also install minicom on OpenWrt OS now that you are connected to the 5G cellular network. To install minicom, use this command ‘**opkg install minicom**’, you may have to do a ‘opkg update’ again.



1. Once the AT commands are sent to switch to MBIM mode and the modem manager is already installed, go to ‘Network’🡪’Interface’🡪’Add new interface’. Select ‘Name’🡪mbim, ‘Protocol’🡪ModemManager. Click ‘Create interface’.

A screenshot of a computer

Description automatically generated

1. Edit ‘mbim’ interface, select ‘Modem device’🡪Quectel – RM502Q-AE, ‘APN’🡪e-ideas, ‘Allowed network technology’🡪5g, ‘IP Type’🡪IPv4/IPv6. Click ‘Save’ and hard reboot the raspberry pi. If the Modem device is empty, hard reboot the raspberry pi until you can see the Modem device.

A screenshot of a computer

Description automatically generated

1. Under the ‘Interfaces’, you can that the ‘mbim’ interface is now getting some traffic (RX/TX MB) while the ‘Wan’ interface is no longer getting any traffic. This means that the ‘OpenWrt’ Wi-Fi Network is now using MBIM to connect to 5G.

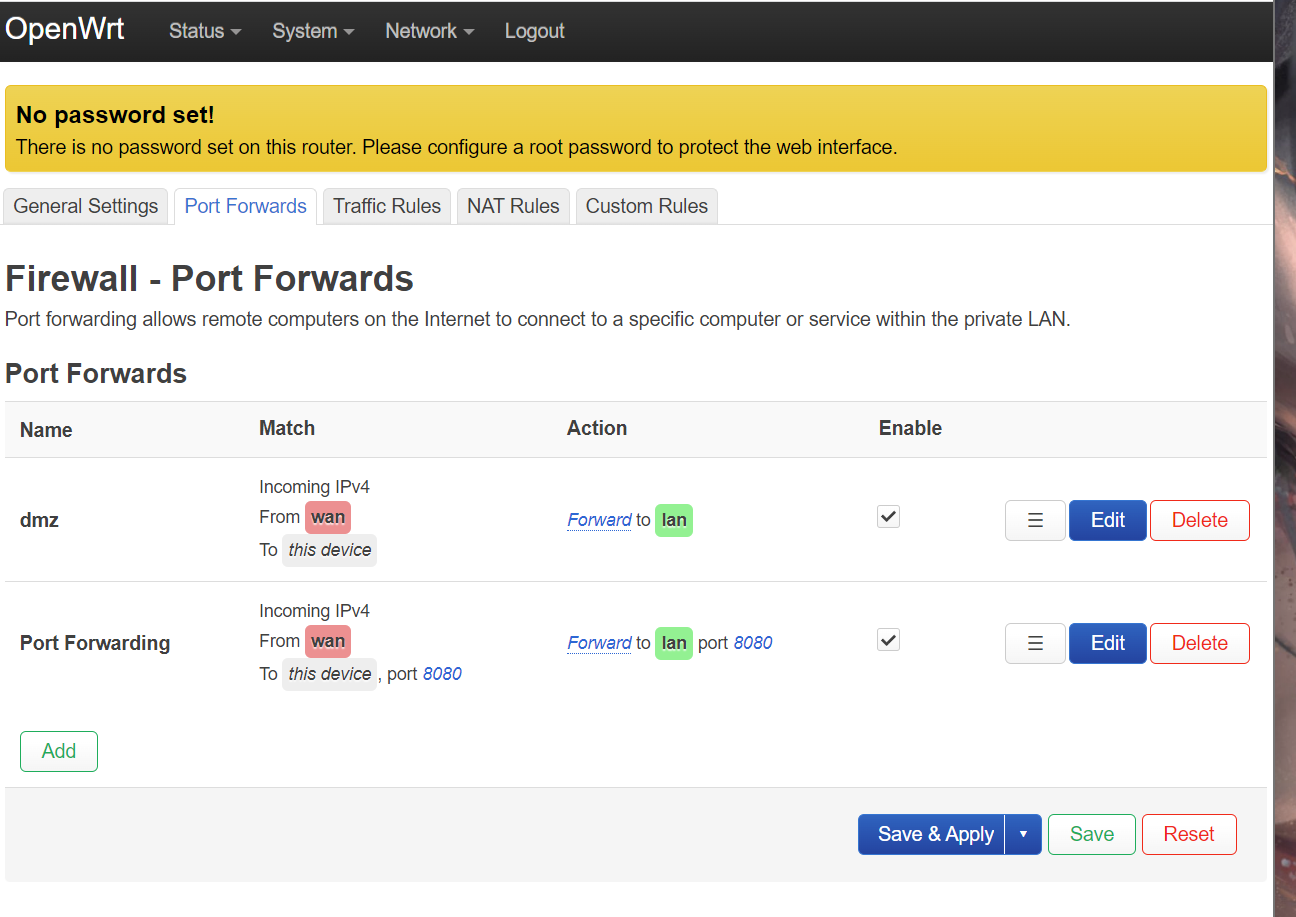
A screenshot of a computer

Description automatically generated

# OpenWrt Applications - Port Forwarding

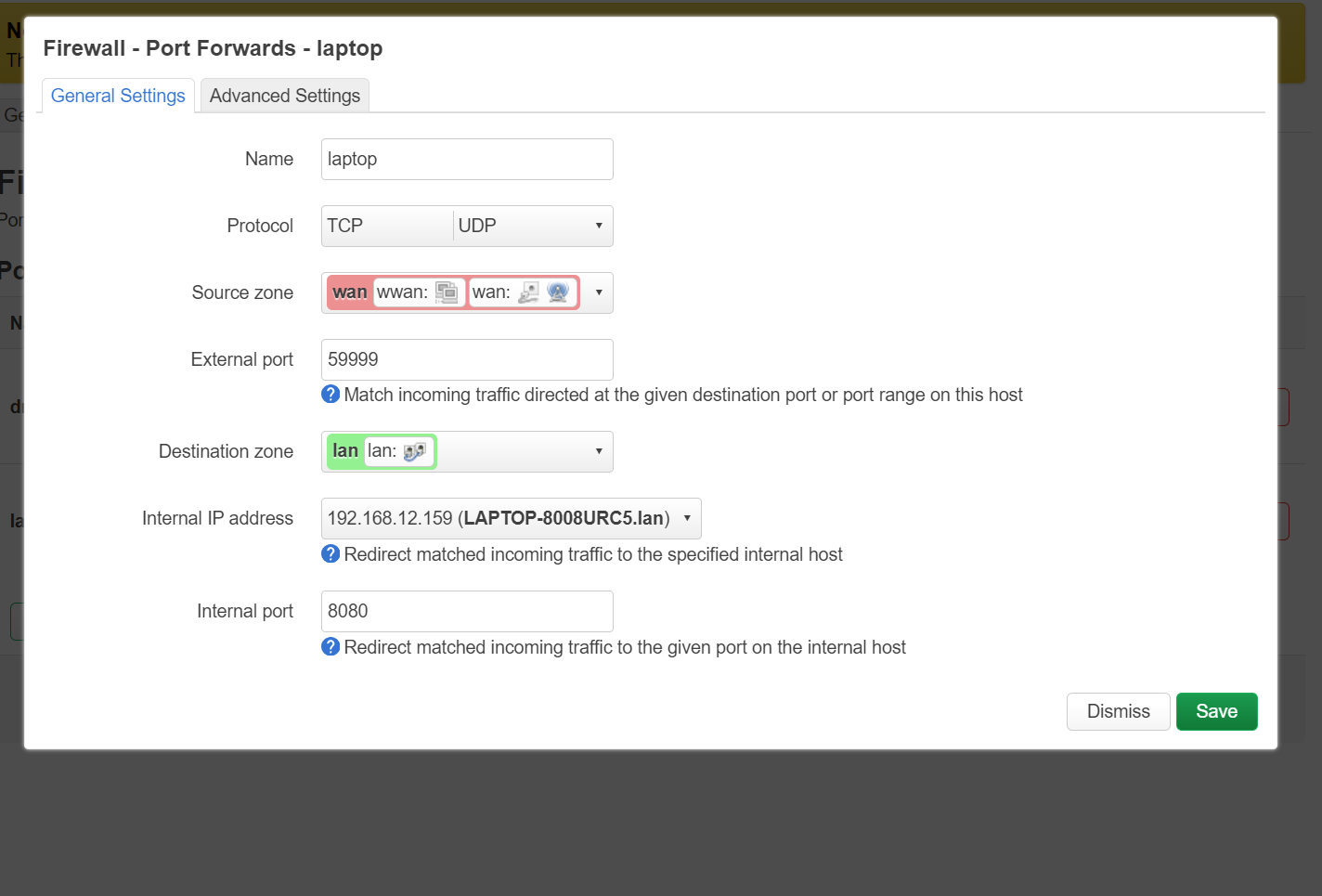
Port Forwarding allows computers or services in private networks (LAN) to connect over the internet with other public or private computers or services. Computers from outside are not able to communicate with computers inside a private network by default. Hence, port forwarding is one way to resolve this issue. The router can easily identify incoming outside traffic for who by looking at the port number and it will route the traffic accordingly.

1. On the LuCi Web GUI, go to Network> Firewall > Port Forwards.



1. Click ‘Add’, add a port forwarding rules.

The port number depends on applications (usually 8080) and the internal IP address depends on specific use case.



# OpenWrt Applications - DMZ

Demilitarized Zones (DMZ) in routers works the same as port forwarding but is much more “open” than port forwarding. Configuring DMZs without care is a **huge security issue**.

