

# Septentrio mosaic-go Internet Connection Sharing on Raspberry Pi Zero W

Hans van der Marel, Delft University of Technology, 14 March 2022.

## General Instructions

### Powering up and connecting the Raspberry Pi Zero W

The Raspberry Pi Zero W is an affordable yet powerful single board computer that is able to provide Internet access over Wifi to the mosaic-go.



The setup is simple

- Connect the USB cable to the USB port on the mosaic-go and the other end to an USB OTG cable connected to the USB port on the Raspberry Pi Zero W,
- Connect a power adapter to the USB PWR In port on the Raspberry Pi Zero W.

Both the Raspberry PI Zero W and mosaic-go will be powered through this single power adapter (or USB power bank).

## Connecting to the mosaic-go Web Interface

Once connected via USB, the mosaic-go can be reached from the Raspberry Pi using the default **Ethernet-over-USB** IP address `192.168.3.1`, or, when the Raspberry Pi is setup as a Wifi bridge, using the IP address `192.168.137.10` or name `mosaic`.

First, log in from another computer (on the local LAN) to the Raspberry Pi with ssh and port forwarding, using one of these commands assuming the hostname for the Pi is `raspberrypi`

```
ssh -L 8080:192.168.3.1:80 raspberrypi.local
ssh -L 8080:mosaic:80 raspberrypi.local
```

If you are using Putty, then go to options->ssh, add a new tunnel, and then save the configuration.

This will open a ssh session on the Raspberry Pi and forward local port 8080 to the mosaic-go Web Interface on the `usb0` interface.

Then `localhost:8080` in your web browser will connect you to the mosaic-go Web Interface.

Alternatively, if you have Raspberry Pi OS full installed, with the desktop environment, you can also start a browser on the Pi and connect to the mosaic Web Interface directly. In case you don't have a monitor and keyboard attached, start `vncserver` from the ssh command line, and then run VNC from another computer in the LAN.

## Connecting to the mosaic-go using RxControl

RxControl uses port 28784 to communicate with the mosaic-go.

First, log in from another computer (on the local LAN) to the Raspberry Pi with ssh and port forwarding, using one of these commands, assuming the hostname for the Pi is `raspberrypi`

```
ssh -L 8080:192.168.3.1:80 -L 28784:192.168.3.1:28784 raspberrypi.local
ssh -L 8080:mosaic:80 -L 28784:mosaic:28784 raspberrypi.local
```

This will forward `localhost:8080` to the Web Interface and `localhost:28784` can be used to connect to the mosaic-go using RxControl.

Unfortunately, RxControl and other tools from the RxTools suite cannot be run from the Raspberry Pi itself as they are not supported on arm processors. However, connecting to them using port forwarding works very well.

## Connecting to the mosaic-go using a terminal

Alternatively, if the receiver is connected via USB, COM1 or COM2, the receiver can be reached from a ssh session on the Raspberry Pi. If the receiver is connected by USB, the virtual COM ports are `/dev/ttyACM0` and `/dev/ttyACM1`.

Enter the following command to start a session on the USB2

```
minicom -D /dev/ttyACM1
```

To exit minicom enter `<ctrl-A> X` and for help `<ctrl-A> Z`. It may also help to enter `<ctrl-A> E` to turn local echo on for commands.

For example, to view the Internet address of the receiver do this

```
minicom -D /dev/ttyACM1
<ctrl-A> E
lif,IPParameters
<ctrl-A> X
```

To inspect NMEA messages on USB1

```
minicom -D /dev/ttyACM0
<ctrl-A> X
```

### Configuring Internet access on the Ethernet-over-USB

To connect from the mosaic-go to an NTRIP caster or use ftp-push it is necessary to configure Internet access over the Ethernet-over-USB interface.

By default, the receiver is **not** allowed to access the Internet over USB. This can be changed on the mosaic-go with the `setUSBInternetAccess` command, or, using the Web Interface, go to the communication menu, click on USB, and turn internet over USB on.

Additionally, this feature requires allowing Internet sharing on your computer. The procedure to do so depends on your operating system.

The procedure to setup Internet access for the Raspberry Pi Zero W is explained in the next section.

Once the setup of the Raspberry Pi Zero W is complete you will have a fully working mosaic-go with access to the Internet through Wifi.

The mosaic-go cannot be reached anymore using the default **Ethernet-over-USB** IP address 192.168.3.1, but using the IP address 192.168.137.10 or name `mosaic`. This is also the address and name that should be used for ssh port forwarding from now on.

Now you can also configure the mosaic-go as an NTRIP client, server or caster to receive or send correction data, or push sbf and/or rinex files by ftp.

### Configuring NTRIP, ftp-push and data logging

To configure an NTRIP client, go to `Communication` and select `NTRIP`. In the NTRIP settings box, click on `New NTRIP client`. Enter the name or IP address and Port number of the Caster In most cases, you will also need to provide the username and password of your account. If the Caster ID is correct, a table should be available from which you can select a Mount point. Click OK to apply the changes. You should see the NTRIP link become active. In addition, the corrections icon in the status panel should show incoming corrections, and the position mode should now mention RTK fixed.

The ftp-push and rinex data logging can only be configured from RxControl. These options are not available from the Web Interface.

## Configuring the Raspberry Pi Zero W as Internet bridge

To configure the Raspberry Pi Zero W we assume there has been a fresh install of the Raspberry Pi OS Lite , with headless ssh access enabled and the wireless connection has been setup during the install. See <https://www.raspberrypi.com/software/> for the installer and instructions.

This done, it should be possible to start a `ssh` session on the Raspberry Pi.

### Install packages

Before we get started with installing and setting up our packages, we will first run an update on the Raspberry Pi by entering the following two commands into the terminal.

```
sudo apt update
sudo apt upgrade
```

Run the following command to install `dnsmasq` and `iptables`. Dnsmasq is a package that acts as both a local DHCP server and a local DNS server. We utilize this package so that we can assign IP addresses and process DNS requests through the Raspberry Pi itself and act like a router. Iptables is what we use to handle the firewall on our device. Minicom is what we will use to connect to the receiver using USB virtual COM ports.

```
sudo apt install dnsmasq iptables minicom
```

Inspect the network interfaces using `ifconfig`, `arp` and `route`. The `ifconfig` command should give you the names of the network interfaces `wlan0` and `usb0` . If you find other names make sure you update these for all the commands in this tutorial.

### Setup usb0 interface

We now proceed with setting up our `usb0` interface.

To do this we need to modify the `dhcpcd.conf` file by running the following command:

```
sudo nano /etc/dhcpcd.conf
```

Within this file we need to add the following lines

```
interface usb0
static ip_address=192.168.137.1/24
static routers=192.168.137.0
```

Now we can save and quit out of the file by pressing `CTRL + X` then pressing `Y` and then `ENTER`.

With our changes made to `dhcpcd` configuration we should now restart the service by running the following command:

```
sudo service dhcpd restart
```

Before we get started with modifying dnsmasq's configuration we will first make a backup of the original configuration by running the following command.

```
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
```

With the original configuration now backed up and moved out of the way we can now move on and create our new configuration file by typing the command below into the terminal.

```
sudo nano /etc/dnsmasq.conf
```

Now that we have our new file created we want to add the lines below.

These lines basically tell the dnsmasq package how to handle DNS and DHCP traffic.

```
interface=usb0          # Use interface usb0
listen-address=192.168.137.1 # Specify the address to listen on
bind-dynamic            # Bind to the interface
server=8.8.8.8          # Use Google DNS
domain-needed           # Don't forward short names
bogus-priv              # Drop the non-routed address spaces.
dhcp-range=192.168.137.50,192.168.137.53,12h # IP range and lease time
dhcp-host=56:1d:69:ea:93:f1,192.168.137.10,mosaic,infinite
```

The last line is an infinite lease coupled to the MAC address of the mosaic-go. To obtain the MAC address run the command `arp -a`, and use this as MAC address in the file. The mosaic-go will then always receive the IP address 192.168.137.10 with name mosaic. If you get the MAC address wrong, or forget the last line, the receiver will get an IP from the dhcp-range.

Now we can save and quit out of the file by pressing `CTRL + X` then pressing `Y` and then `ENTER`.

### Enable IP forwarding

We need to configure the Raspberry Pi so that it will forward all traffic from our `usb0` connection over to our `wlan0` connection.

To do this we must first enable ipv4p IP Forwarding through the `sysctl.conf` configuration file, so let's begin editing it with the following command:

```
sudo nano /etc/sysctl.conf
```

Within this file you need to find the following line, and remove the `#` from the beginning of it, so that it reads

```
net.ipv4.ip_forward=1
```

Now we can save and quit out of the file by pressing `CTRL + X` then pressing `Y` and then `ENTER`.

Now since we don't want to have to wait until the next reboot before the configuration is loaded in, we can run the following command to enable it immediately.

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

## Setup iptables

Now that IPv4 Forwarding is enabled we can reconfigure our firewall so that traffic is forwarded from our `usb0` interface over to our `wlan0` connection.

Run the following commands to add our new rules to the iptable:

```
sudo iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
sudo iptables -A FORWARD -i wlan0 -o usb0 -m state --state \
RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i usb0 -o wlan0 -j ACCEPT
```

Iptables are flushed on every boot of the Raspberry Pi so we will need to save our new rules somewhere so they are loaded back in on every boot. To save our new set of rules run the following command.

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

Now with our new rules safely saved somewhere we need to make this file be loaded back in on every reboot.

The most simple way to handle this is to modify the `rc.local` file. Run the following command to begin editing the file.

```
sudo nano /etc/rc.local
```

We need to add the line

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

Make sure this line appears above `“exit 0”`. This line basically reads the settings out of our `iptables.ipv4.nat` file and loads them into the iptables.

Now we can save and quit out of the file by pressing `CTRL + X` then pressing `Y` and then `ENTER`.

## Final steps

Finally all we need to do is start our `dnsmasq` service. To do this, all you need to do is run the following command:

```
sudo service dnsmasq start
```

Now you should finally have a fully operational Raspberry Pi WiFi Bridge, you can ensure this is working by plugging any device into its Ethernet port, the bridge should provide an internet connection to the device you plugged it into.

To ensure everything will run smoothly it's best to try rebooting now.

This will ensure that everything will successfully re-enable when the Raspberry Pi is started back up.

Disconnect the receiver. Run the following command to reboot the Raspberry Pi

```
sudo reboot
```

and reconnect the receiver.

### **Setup port forwarding (not recommended)**

Another method is to setup network address translation for port 80 of the Raspberry Pi to port 80 on usb0 (the mosaic Web Interface). This can be done by adding the following iptables commands to the setup:

```
sudo iptables -A FORWARD -d 192.168.137.10 -p tcp --dport 80 -j ACCEPT
sudo iptables -A FORWARD -s 192.168.137.10 -j ACCEPT
sudo iptables -t nat -A PREROUTING -d 192.168.178.11 -p tcp --dport 80 -j
DNAT --to-destination 192.168.137.10
```

The same can be done for RxControl port 28784.

This approach is okay on your own LAN network, but it is not save over the Internet. Besides, the tables have to be setup using the IP addresses of the wlan0 and usb0 devices, which may change.

We recommend strongly to use port forwarding with ssh and not to expose these ports to the outside.

## **References**

[1] Mosaic-go Getting Started, Septentrio, 2021.

[2] Mosaic-X5 Reference Guide, Septentrio, 2021.

[3] Gus, Easy Raspberry Pi WiFi Bridge, Pi My Life Up, <https://pimylifeup.com/raspberry-pi-wifi-bridge/> . Viewed on March 12 2022.

[4] dnsmasq man pages.

[5] H. van der Marel, Septentrio mosaic-go Internet Connection Sharing on Windows-10, Delft University of Technology, March 2022.