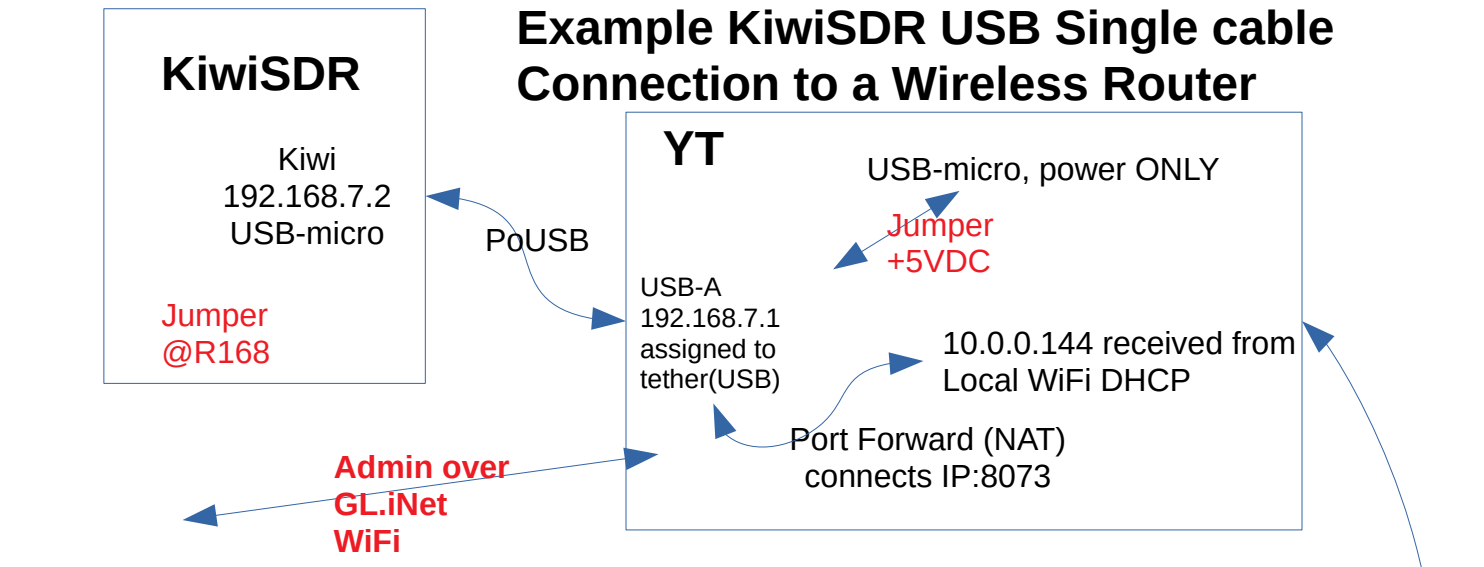
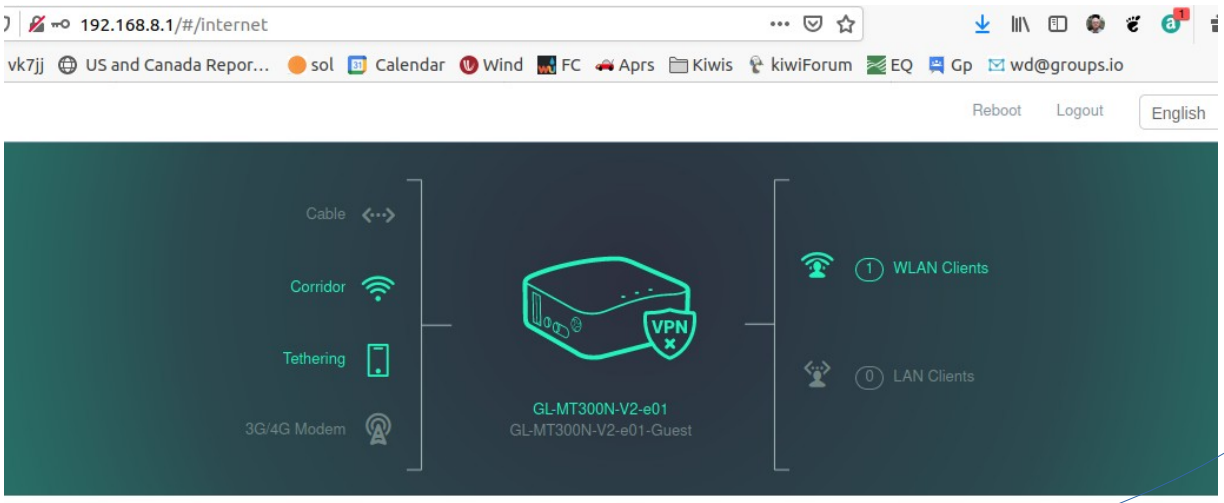
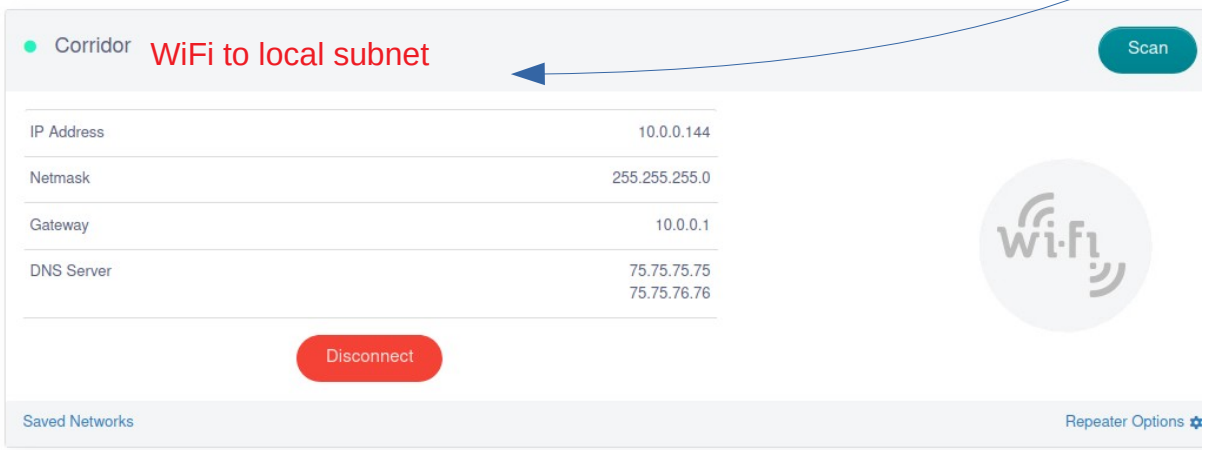
Example KiwiSDR USB Single cable Connection to a Wireless Router



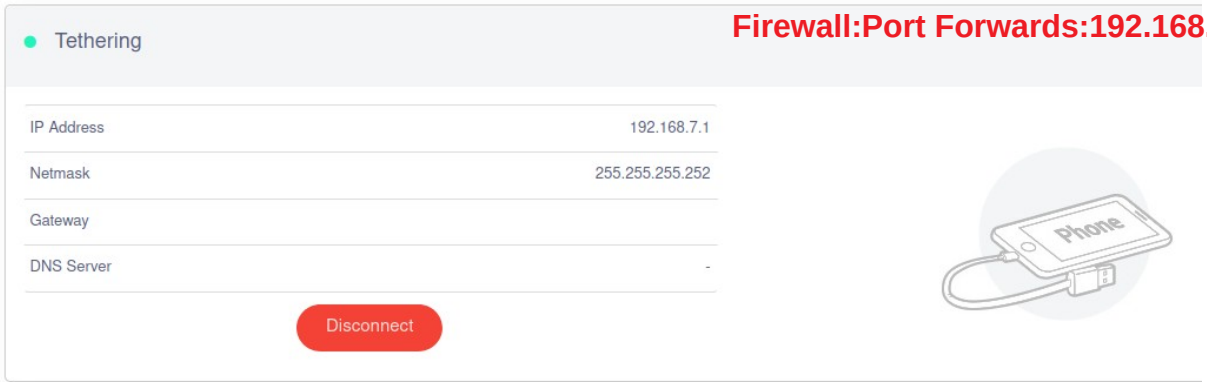


Access over Local WiFi

Kiwi accessible as 10.0.0.144:8073



Firewall:Port Forwards:192.168.7.2:8073



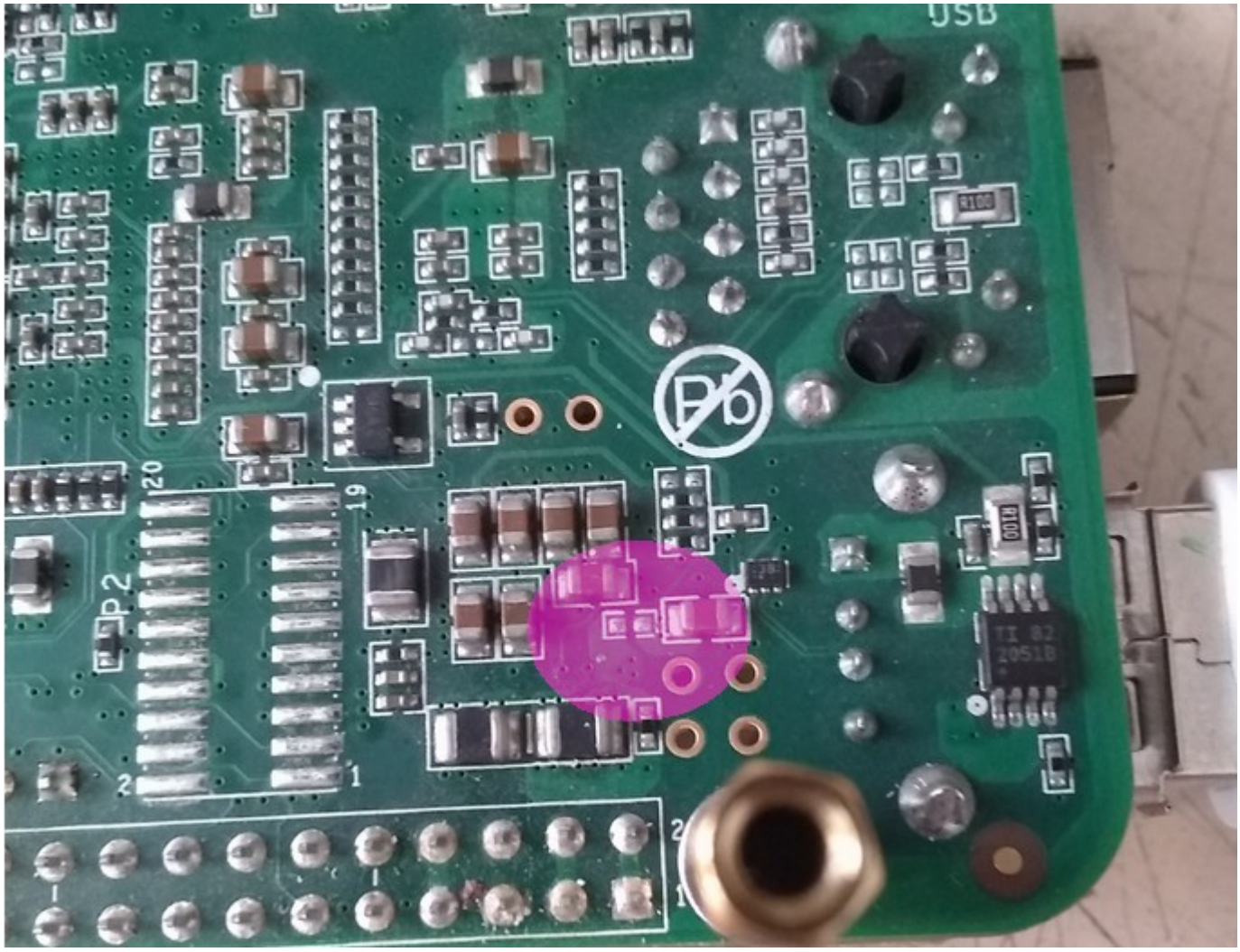
[Updated 11 March 2021 to include mechanical strain relief and additional Kiwi IP address information]

Here's what I think it takes to make a portable Kiwi System work better. This includes hardware mods to allow powering a YT ("Yellow Thing") wireless router over its single data connection from the Kiwi as well as YT set-up to access the Kiwi. Doing this can remove wired-LAN and power supply common mode current paths that raise the KiwiSDR's noise floor. This effect is somewhat insidious because it can appear when an antenna feedline is attached and may lead the user to think the problem is coming in via the antenna operating in a conventional manner.

Hardware mods:

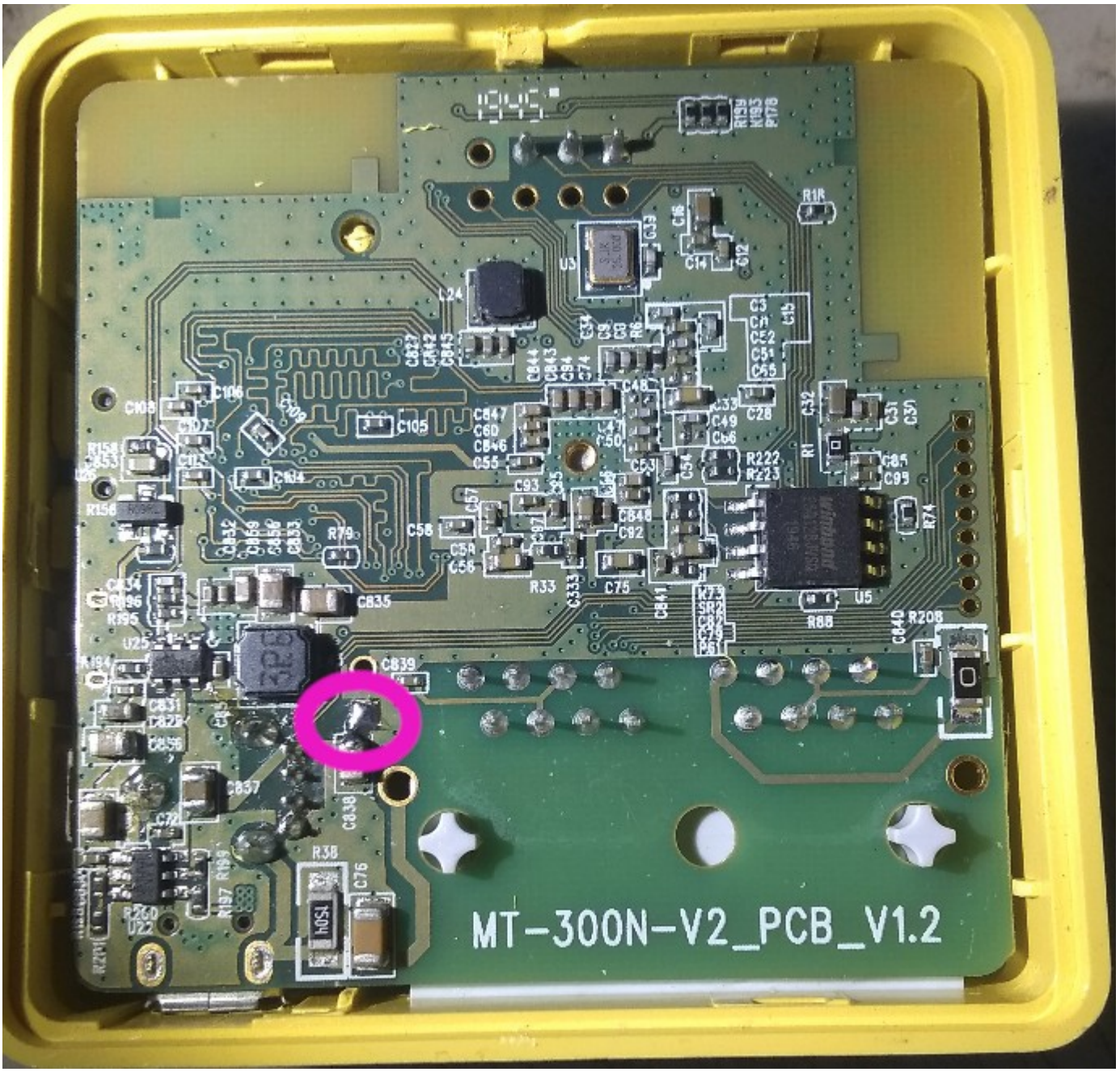
The hardware modifications allow powering the wireless router over the micro-USB Client connection from the Kiwi. In this way, with Power-over-USB only a single cable connection between the Kiwi and the router is necessary, removing the danger of common-mode noise ingress to the Kiwi – this is a demonstrated vulnerability of the KiwiSDR.

On the Kiwi, pads for the missing R168 are to be found on the BBG above P9, pins 1&2 here:



Either solder in a zero ohm 0603 SMD part or jumper with a bare wire. This connects the Kiwi system 5V, conditioned by filtering at the Kiwi's barrel power connector, to both USB-A (Host) and USB-micro (Client) connectors on the Kiwi's BeagleBones board. This supply then connects via a single USB-A <==> USB-micro cable and can supply power for the router, thus avoiding an additional power supply path and potential noise current loop.

On the wireless router, here called "YT" for "Yellow Thing", the two USB connector 5V lines have adjacent pads, USB-A 5V below the USB-micro(power) 5 in the photo, that can be accessed and tied together:



I've actually used both a Schottky diodes across these pads as well as the jumper present in the picture above. This diode is 'backwards' across the IC that normally provides 5V to the USB -A connector. It appears that shorting it rather than using a diode is OK.

The PoUSB system has run for extended periods with no issue and so far no local noise/ingress at all. The Kiwi can even be powered from a SMPS 5V Buck converter with no evidence of the switching frequency anywhere in the Kiwi.

With this arrangement, for the first time an isolation transformer at the Kiwi SMA makes no improvement and also for the first time one can actually short the SMA shield (Kiwi PCB ground plane at the SMA) to the Kiwi enclosure with no visible effect at HF.

Since getting the PoUSB to work I've discovered that the GPS QRN at 1575 MHz is lower but still present. It seems that without big LAN&PS CM noise current sources the KiwiSDR is better behaved. I find that a GPS antenna mounted right on the cover of the Kiwi works if I ground the SMAs directly to the aluminum enclosure. This is something I wasn't able to do previously as it added a loop that LAN/PS CM coupled into.

These latter results seem to me important, certainly for portable operation and may indicate things are more under control than they were. It may also be worthwhile to use PoUSB to help avoid problems on stationary systems and not only the portable system for which PoUSB was originally intended.

Mechanical mods:

To operate the KiwiSDR with PoUSB in this way means using the very fragile micro-USB connector on the Kiwi. The surface mount connector very easily breaks or comes off the PCB. It is very important to protect this connector and so a mechanical strain relief is vital. If the connector breaks, it may take PCB traces with it and make the BeagleBones Green PCB unusable.

I’ve designed and 3D printed a simple plastic clamp that mounts on the end of the Kiwi to clamp the right angle USB cable so that the connector on the Kiwi PCB won’t break (Fig. 1).

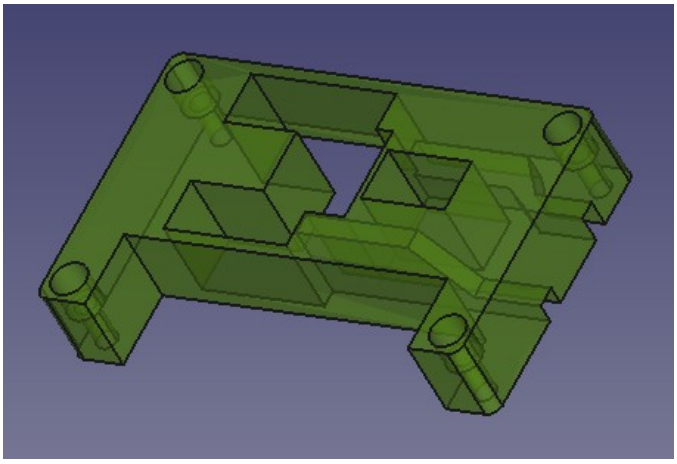


Figure 1: Plastic clamp to protect the Kiwi micro-USB connector

Mount the clamp over the existing enclosure end plate with four 3x16mm socket head screws that replace the four original on the Kiwi enclosure. Use a right angle micro-USB <==> USB-A cable and a right angle power connector/cable so that you can route those cables under the clamp and out the right side of the Kiwi.

Write me for a 3D printable .STL file and print the clamp in PLA with the Kiwi side up and supports enabled. It should take a couple of hours to print.

Software mods:

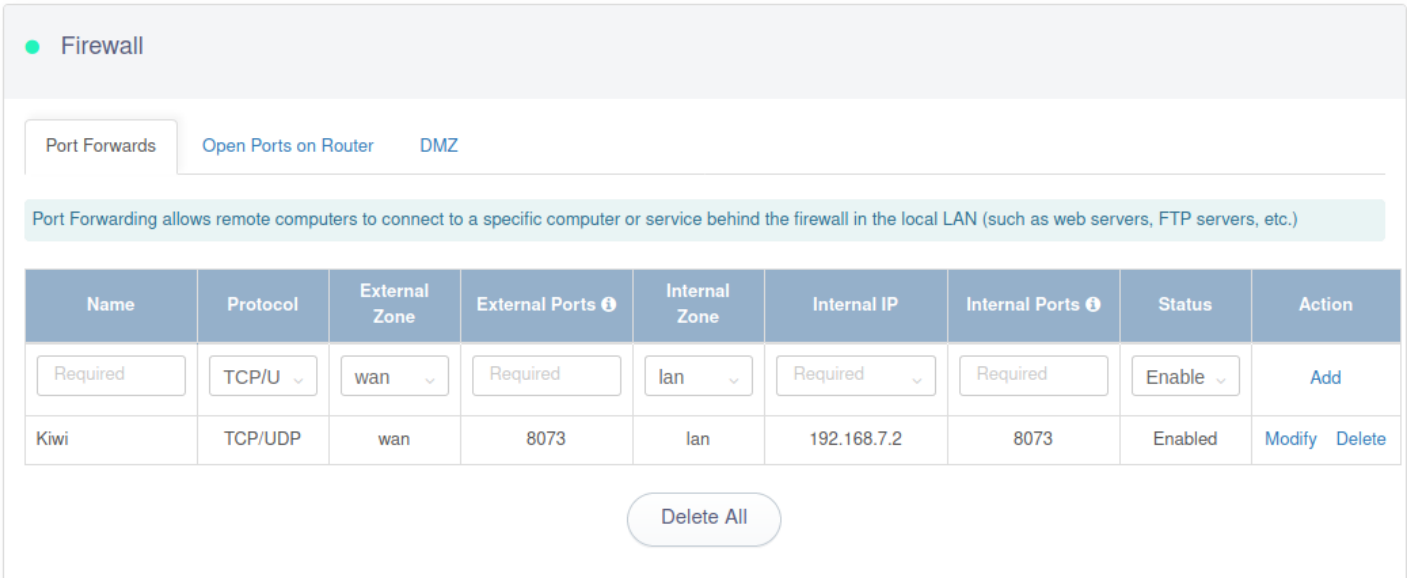
The Kiwi seems to always assign 192.168.6.2/30 or 192.168.7.2/30 to it's client USB port. I haven't determined how that choice is made. You can log into the Kiwi as root, perform an 'ifconfig' to see what address the port has received. Use that address to determine port forwarding, below. Other than waiting for it to become available after power up nothing needs to be done. The YT discovers the router OK once it has come up and **may** assign YT an address via DHCP, at least sometimes. A better way to handle this is to set the YT port address statically to 192.168.6.1 or 192.168.7.1 depending upon what the Kiwi did (above). That setting is made under "Advanced Settings: Interface" in the router's administration tool. Do it prior to setting the port forwarding.

Beyond this physical and power and port configuration, all that is required is to create a port forward for the Kiwi so that YT can present that port to the outside world as though it were coming from itself. To do this, just log into YT via the WiFi admin interface at SSID GL-MT300N-V2-* (or something similar) at its factory default IP address 192.168.8.1. On a new YT the password is "goodlife" or maybe "Goodlife", I've changed mine but it says in the YT instructions which it is. It might be a good idea to update the YT firmware at this point. Once in and updated, go to the "Firewall" tab and map the Kiwi's ports to YT by setting up the tethering interface.

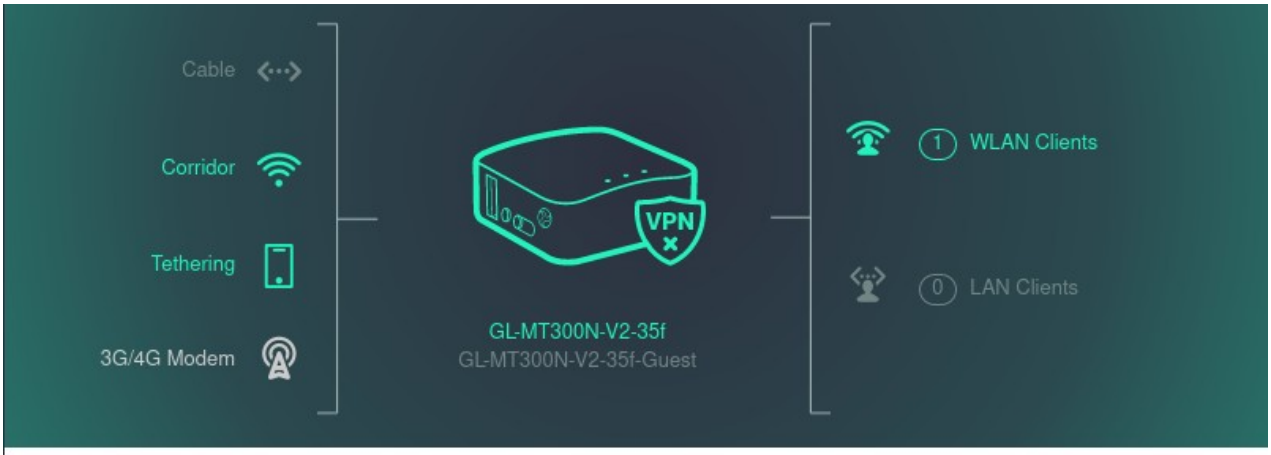
Assign a static rather than DHCP address to YT's USB(Tethering) port. The tethering port comes from the factory set to DHCP and should be changed to static, 192.168.7.1 (or 192.168.6.1) to match the hard coded network that the Kiwi demands and remove the necessity for DHCP from the Kiwi. Both YT and Kiwi USB will then have static addresses.

YT's USB port is set to static as 192.168.8.1 under :More Settings: Advanced: Network:Interfaces:Tethering:Edit. Give it the static address and a subnet mask. It only communicates with the Kiwi but 255.255.255.0 is an OK mask.

Here's an example that works:



YT should now both have connection to a booted-up Kiwi and should pass that on to the outside world. To get a WiFi connection to the outside, you'll need to have scanned for your home WiFi network, entered the password and connected. This should give you a picture on YT's main page of this form (in this example the WiFi network is SSID "Corridor"):



Corridor

Scan

IP Address	10.0.0.162
Netmask	255.255.255.0
Gateway	10.0.0.1
DNS Server	75.75.75.75 75.75.76.76

Disconnect

Saved Networks

Repeater Options

Tethering

IP Address	192.168.7.1
Netmask	255.255.255.252
Gateway	
DNS Server	-

Disconnect

The above is showing the Tethering/USB connection to the Kiwi, the WiFi connection to the local WiFi network and in this case the connection to the administration WiFi connection shown under WLAN Clients.

Using the IP address found for the home WiFi network, 10.0.0.162 in this example, the Kiwi should be accessible via WiFi as 10.0.0.162:8073 assuming the default Kiwi port of 8073 was set up on the Kiwi administration page was mapped with no change as shown above.

You will need to have previously provided an administrative password to the Kiwi since if you don't if you try to get to Kiwi admin via something like 10.0.0.162:8073/admin as you normally would it will complain that you aren't on its 192.168.7.x USB subnet and won't let you in. If you have reason to want to SSH directly into the Kiwi/BBG I suppose you'll have to add an additional port forward or else temporarily use wired LAN to connect to the Kiwi.

I think that adding this single-cable PoUSB connection may be pretty worthwhile for portable Kiwi systems. With no connections to the router (YT) there is no longer a CM current ingress path from LAN/PS lines, through the Kiwi and out the antenna or even PS-to-LAN current through the Kiwi ground plane. I think this is why I'm no longer seeing network of any SMPS noise when powering the Kiwi from a 12V via a 5V buck converter switching at ~960 kHz. Even CM current from the antenna coax has nowhere to go so the Kiwi doesn't see any QRM now. The system can function by powering through YT's micro-USB power input instead of the Kiwi's barrel but this would be a bad idea since it would add a noise current path from the external power supply.

There is still an issue of the Kiwi leaking internal 1575 MHz QRM and causing problems with Kiwi-GPS use. But since the other CM current paths are mitigated it is now finally possible to short the Antenna and GPS SMAs on the KiwiSDR to the Kiwi enclosure, I used aluminum HVAC tape to test this, and so to have the Kiwi's GPS antenna sit directly on top of the enclosure. Previously there was so much QRM that the KiwiGPS couldn't acquire good satellite signals.

There is no doubt still much to learn here but hopefully others of you will try this and add to our collective knowledge.

Glenn N6GN

24 Feb 2021

[Updated 11 March 2021]