UART and USB Issues

This section covers common pitfalls encountered when using the USB and UART ports. Please note that for -H hardware, the auxiliary port is either USB or UART, but not both.

UART Troubleshooting

The UART port is described in more detail in the Serial Interface Guide available for download in the Doodle Labs Technical Library. The guide includes information about how to manually configure the UART port over the CLI. Troubleshooting the serial interface generally requires the user to be able to SSH into the radio.

The general steps to debug the UART port are

1. Make sure you can get data from a PC to the UART port. You can SSH into the Mesh Rider Radio and run the application `picocom` as described in the Serial Interface Guide. Make sure you disable `socat` before running `picocom`.
   a. In order to eliminate the user's host machine as the source of the problem, you can connect the UART port in loop-back mode but directly wiring the UART TX to the RX. Any message you type in `picocom` should be echoed back to you.
   b. Assuming loop-back works, then the problem could be a mismatch in the signalling levels, or UART settings (baudrate, parity etc).
2. Assuming that serial data can be sent and received over the UART interface you can debug the networking side. Close `picocom` and re-enable and re-start the serial interface bridge using the GUI. Then in the Mesh Rider Radio CLI, you can
   a. Check that `socat` and `ser2net` are running (they will both be running)

```
root@smartradio:/# ps | grep -E "socat|ser2net" | grep -v
"socat|ser2net"
3133 root      1012 S    /usr/sbin/ser2net -c /var/run/ser2net.conf
3370 root      1072 S    /usr/bin/socat tcp:127.0.0.1:65534 UDP4-
LISTEN:2000
```

   b. Make sure that the firewall is open if you are using server mode

```
root@smartradio:/# iptables -L | grep 2000
ACCEPT     udp  --  anywhere             anywhere             udp
dpt:2000 /* !fw3: Allow-Socat */
ACCEPT     tcp  --  anywhere             anywhere             tcp
dpt:2000 /* !fw3: Allow-Socat */
```

c. Check the status of the socket. In TCP mode, the socket should be either in a `LISTEN` state or an `ESTABLISHED` state. In UDP mode, the state will be either empty or `ESTABLISHED`

```
root@smartradio:/# netstat -tuapn | grep 2000
tcp        0      0 0.0.0.0:2000            0.0.0.0:*
LISTEN      20711/socat
```

d. If the connection state is not `ESTABLISHED`, then it means that the application trying to connect to the Mesh Rider Radio is unable to connect. This is normally a problem with the configuration. For example, the IP address, network port or firewall could be configured wrongly.

USB Troubleshooting

Different hardware variants have different types of USB ports. Please see the first section for details. USB uses a master-slave communications protocol where a single host can support up to 128 slaves. In general, a USB port is either a USB device or a USB host; USB devices cannot communicate with other USB devices and USB hosts cannot communicate with other hosts. Please check what type of USB port your hardware has before proceeding.

Some devices have USB OTG ports which can switch between host and device mode. Typically they use the 5-V VBUS line as an indicator for which mode they should be in. If an OTG port detects power on the VBUS line, then it will switch to device mode.

Mesh Rider Radio USB Device Ports

The USB Device port is the main data interface on the Wearable and Hex-Band models. It only supports IP networking over USB and does not support any other USB protocol such as HID. This port can be directly plugged into a laptop, or certain Smart Devices (typically tablets and not phones). Please see the note on Android below.

Mesh Rider Radio USB Host Ports

The USB Host port is setup by default to support IP networking over USB. It can be made to support certain USB devices, depending on the type of driver required. It cannot be connected directly with a laptop, but it can be connected to certain Smart Devices (typically phones). See the note on Android below.

Android

Android phones and tablets usually have USB OTG ports, and may support USB tethering or USB reverse tethering.

- USB Tethering is when the Android device is sharing it's internet connection with another device. The Android device's OTG port switches to device mode (it needs a 5-V input to do so). The Android device also starts a DHCP server and routes traffic from the connected device to it's internet connection. Only one such Android device can be on the subnet. USB Tethering is normally supported on Android phones as they are expected to have a dedicated internet connection. This mode works with the Mesh Rider Radio's USB Host port.
- USB Reverse Tethering is when the Android device gets it's internet connection from another device. The Android device's OTG port switches to host mode and it raises the VBUS line to 5-V. The Android device switches to DHCP client mode by default, but it can also be setup with a static IP address. Many of such devices can be on the same subnet. USB Recerse Tethering may be supported on Android tablets as they do not have a dedicated internet connection. This mode works with the Mesh Rider Radio's USB Device port (Wearable, Hex-Band).

Signal Integrity and Power

USB lines run at high speed and are very sensitive to both differential and common-mode disturbances. Make sure that all four lines (VBUS, USB+, USB-, GND) are tightly twisted together. Ideally the lines should be kept short for best signal integrity.

Some Mesh Rider Radio models supply a 5-V output with the USB. This output is rated to a maximum of 1-A. If the connected device is power hungry, then you should supply the power externally.

Debugging

You can check the connectivity of the USB port if you SSH into the Mesh Rider Radio. If you are using the USB Host port, unplug and then plug in the USB device, and then run

```
root@smartradio:/# dmesg | tail
```

This prints out the latest kernel messages and you should see some notifications about activity on the USB port. You can also type `ifconfig usb0` to make sure that the USB port is up and bridged to the network.

If you are using the USB Device port, you should check for connectivity information on the host machine itself rather than the Mesh Rider Radio (for example, notifications that a new USB device was attached). From the Mesh Rider Radio, you can attempt to directly ping the host machine, and should resolve any IP conflicts (see Radio Access Issues in the Troubleshooting Guide).