LoRA-PI KISS Modem
User's Manual

RPC Electronics, LLC

Table of Contents

1. Introduction and Acknowledgments

The LoRA-Pi Modem is a KISS interfaced LoRA modem and 1 Watt transceiver all on a single board. The module is sized to fit on a Raspberry Pi 3/4/5 perfectly. There are three methods for connecting to the module: USB, GPIO UART and Bluetooth.

Thank you to the following people for their help with making this product a reality!

Remi Bilodeau VE2YAG Firmware author for the LoRA-Pi module
Mark Cohen Hardware and firmware field testing

2. Features and Dimensions

- LoRA modulation Modem
- KISS USB serial interface at 115200 baud
- 1 Watt 433 MHz transceiver on-board
- Sized to fit on a Raspberry Pi 3/4/5 (65 mm x 56 mm)

WARNING!! DO NOT power on modem without a proper antenna or 50 ohm load on the LoRA RF port. Failure to do so can result in permanent damage to the transceiver!

3. Front Panel

The front panel (with or without the 3D printed case) contains LED indicators and a single momentary button

LED's: Power, Transmit, Receive, Bluetooth and WIFI
Button: AP on Boot - Will start WIFI configuration interface on power-on when held in

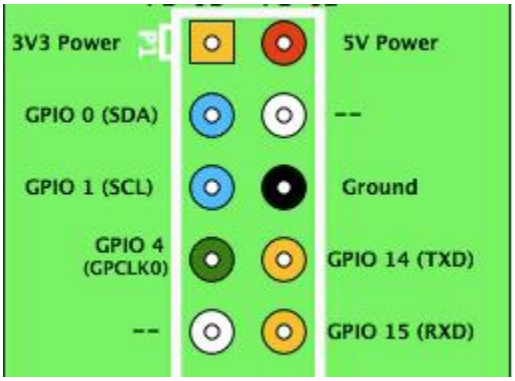
4. Connections

1. UHF LoRA RF - SMA Female
2. USB-C - Power, Configuration and KISS USB Interface
3. WIFI/Bluetooth - RP-SMA Female

WARNING!! DO NOT power on modem without a proper antenna or 50 ohm load on the LoRA RF port. Failure to do so can result in permanent damage to the transceiver!

5. Optional Raspberry Pi UART connector and stand-off kit

If the LoRA-Pi is being used with a Raspberry Pi 3/4/5, you have the option of connecting to it through the 2x20 pin header on the Raspberry Pi using the standard UART serial pins GPIO 14 and GPIO 15 (shown in the below diagram)

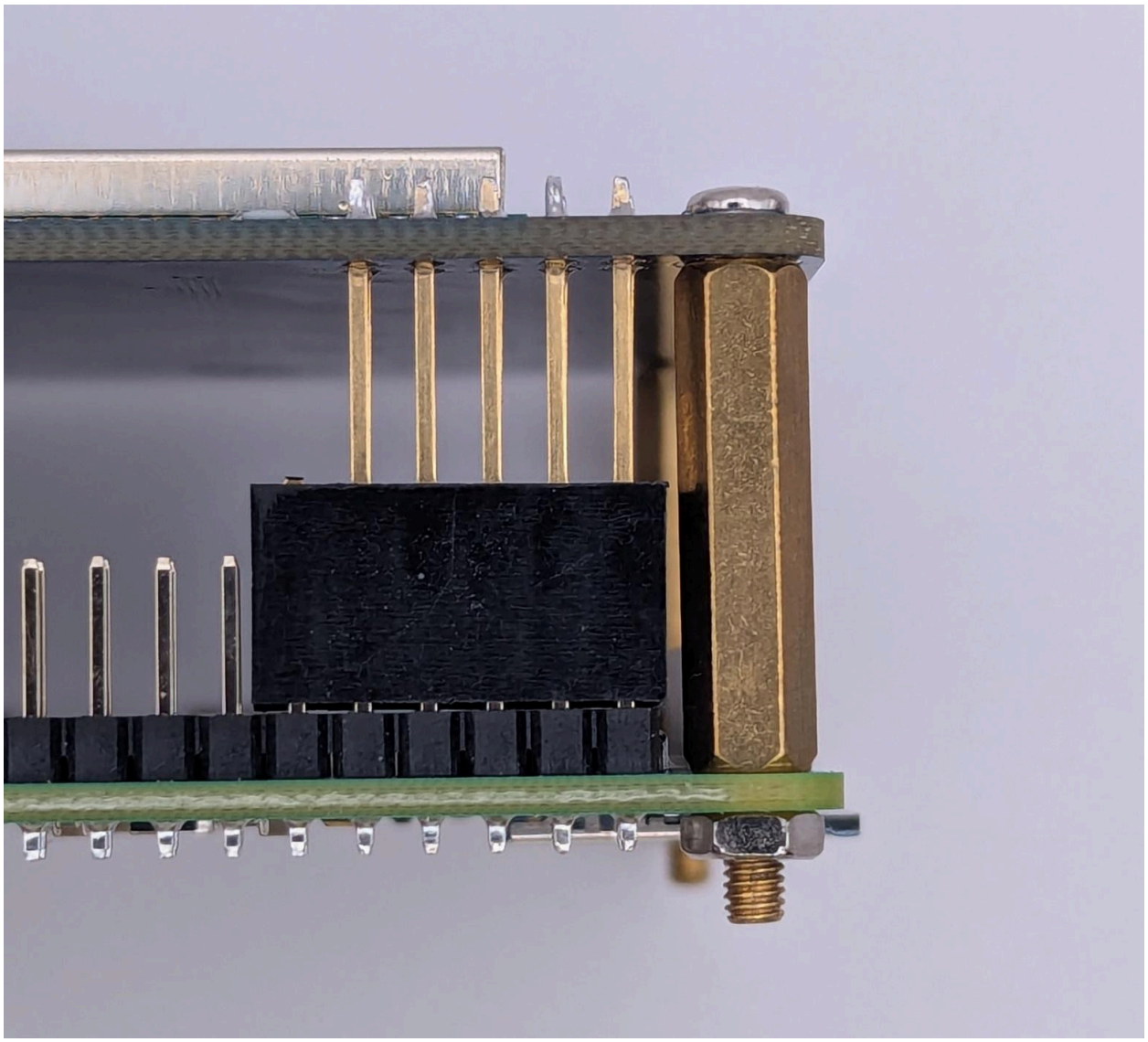


The LoRA-Pi will also get power and ground through the 40 pin header.

An optional kit can be purchased through RPC Electronics, LLC that includes the extended pin header sockets and stand-off hardware to install the LoRA-Pi directly on the Raspberry Pi as a “hat”.

Your optional kit will come with either two 1x5 header sockets or a single 2x5 header socket, depending on what is available at the time. The end user will need to solder these header sockets on. It's best to plug the sockets onto the Raspberry Pi first. Add the brass stand-offs to the Raspberry Pi using the threaded shaft end and nuts. Lastly, place the LoRA-Pi onto the tails of the header sockets, secure LoRA-Pi in place with screws and solder last. This will ensure the correct height and alignment.

Note: the picture shows 1x6 header sockets with a single pin cut off. The kits with 1x5 header sockets have been professionally trimmed to remove the pin and unused plastic part of the connector.



The brass stand-offs in the kit each come with a single screw and nut for installation.

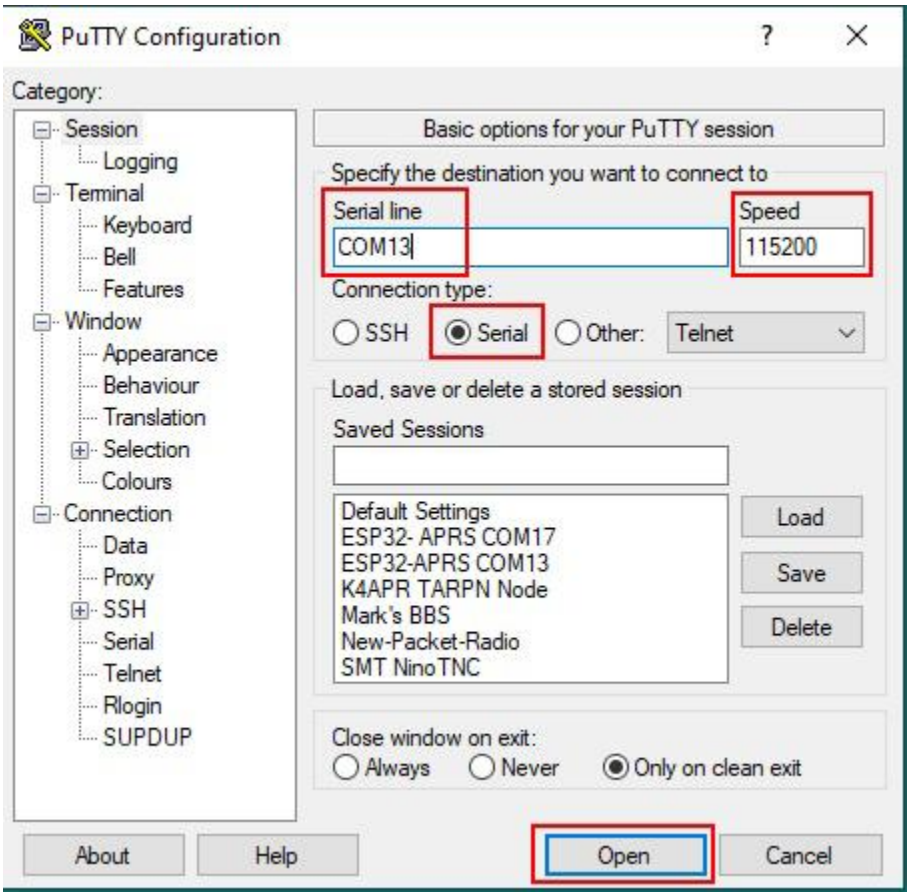
6. Configuration

- The LoRA-Pi module comes pre-configured for use as a KISS modem through the USB-C port with a serial baud rate of 115200.
- Most modern operating systems will automatically create a USB virtual com port when it detects the on-board FTDI interface.
    - Windows com ports will typically be: COM1, COM2, etc.
    - Linux com ports will typically be: ttyUSB0, ttyUSB1, etc.
    - Android will automatically detect the FTDI serial device connected
- The UART port also comes pre-configured to operate at 115200 baud.

6.1 Advanced Configuration Using PuTTY

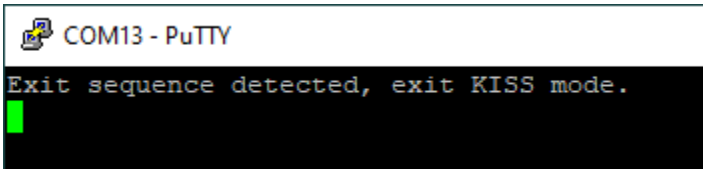
NOTE: This procedure will be replaced with the web configuration tool

1. Open PuTTY and use the following settings (Your serial port will likely not be the same as the one shown!)

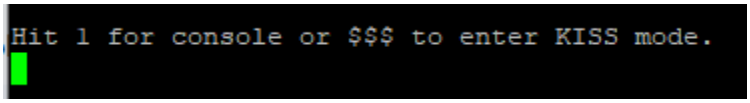


2. Clock the Open button to start the session

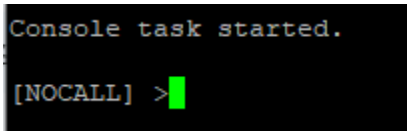
3. You will have a black, blank terminal open up
4. Click enter will not do anything, this is normal
5. Holding the shift key down, type the 4 key three times to send three $ dollar signs. This will break the modem out of the KISS interface. You should see this message:



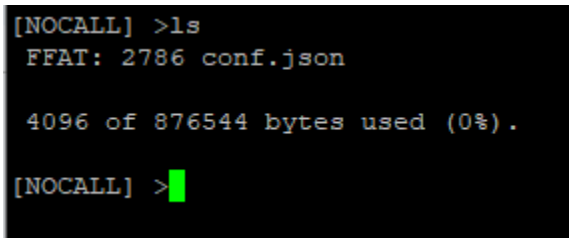
6. Hit the enter key and you should see this message:



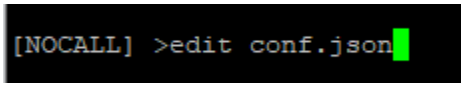
7. Press 1 and you should see this message:

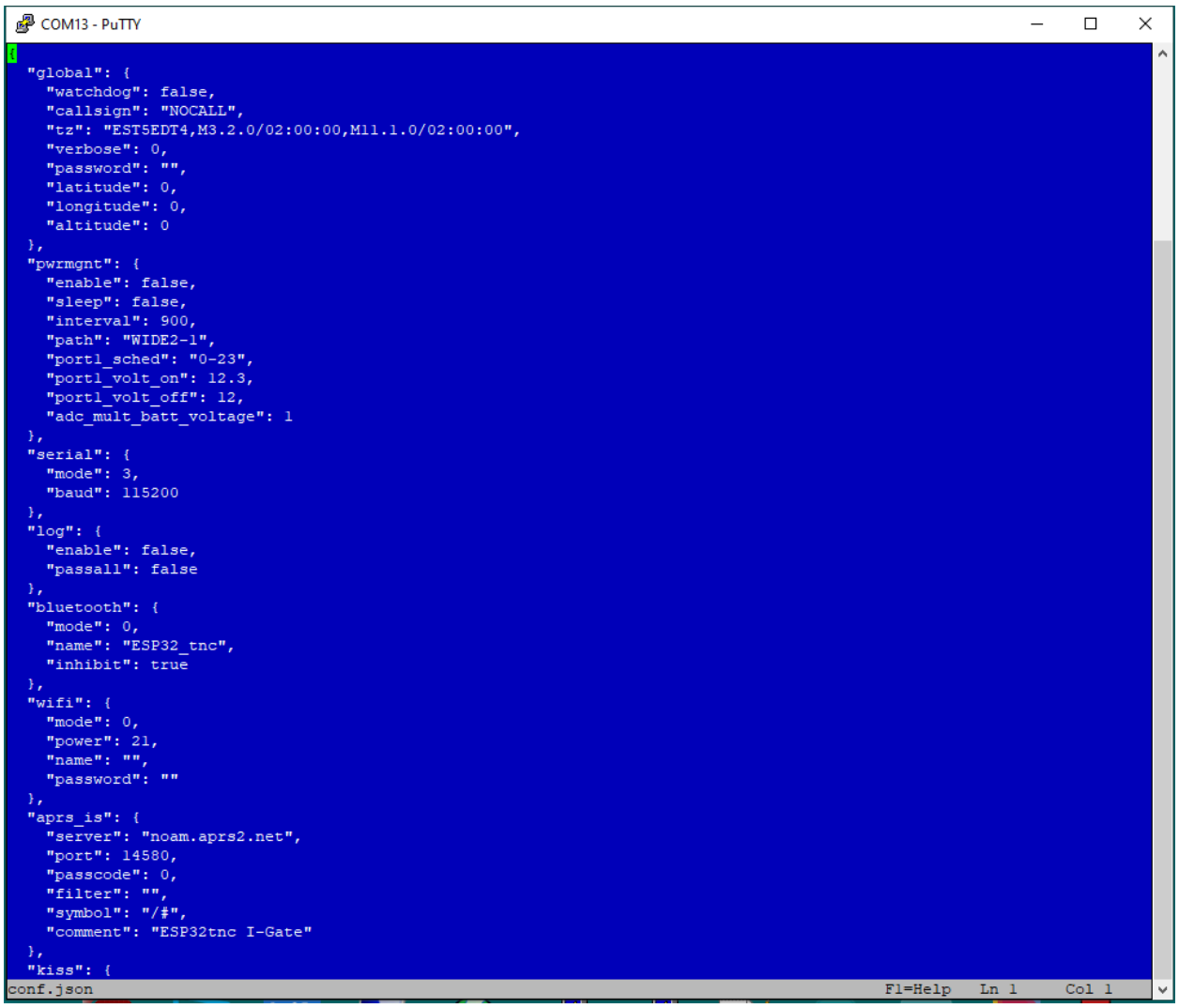


8. Type ls and hit enter:



9. Type edit conf.json and hit enter. This should open up a config editor:





There are only a few crucial settings sections that you may need to change with the LoRA-Pi:

- Serial
- WIFI (future functions)
- KISS
- Port (LoRA Transceiver Settings)

Serial

This sets the mode and baud rate of the UART port connection used with the Raspberry Pi 40 pin header

```
"serial": {
  "mode": 3,
  "baud": 115200
```

“Mode”: 3 enables KISS
“Baud”: Sets the baud rate

WIFI

This is currently not used at the time of writing this document, but will have future use

```
"wifi": {
  "mode": 0,
  "power": 21,
  "name": "",
  "password": ""
```

KISS

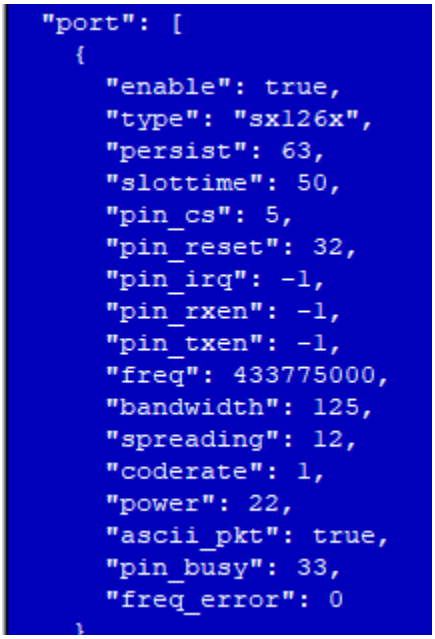
These settings affect how KISS is delivered to the host

```
"kiss": {
  "on_ip_port": 0,
  "gps_embed": false,
  "on_usb": true
```

“gps_embed” is not supported leave set to false
“on_usb” set to true will enable KISS communication through the USB port

Port (LoRA Transceiver Settings)

These settings tell the LoRA transceiver what mode to operate in and what frequency to operate on



Sometimes the frequency can look strange like this:

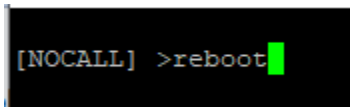
Don't be alarmed, you can enter the frequency in normally, using this format:

433775000

Make sure to enter in a valid frequency, out to nine (9) digits. No more, no less.

"ascii_pkt" should be set to true, in order for your LoRA-Pi to work with standard APRS LoRA packet data.

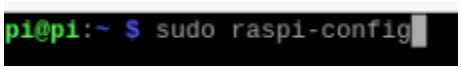
10. After all changes have been made. Use CTRL-S to save and CTRL-Q to quit.
11. Back at the command prompt issue a "reboot" command



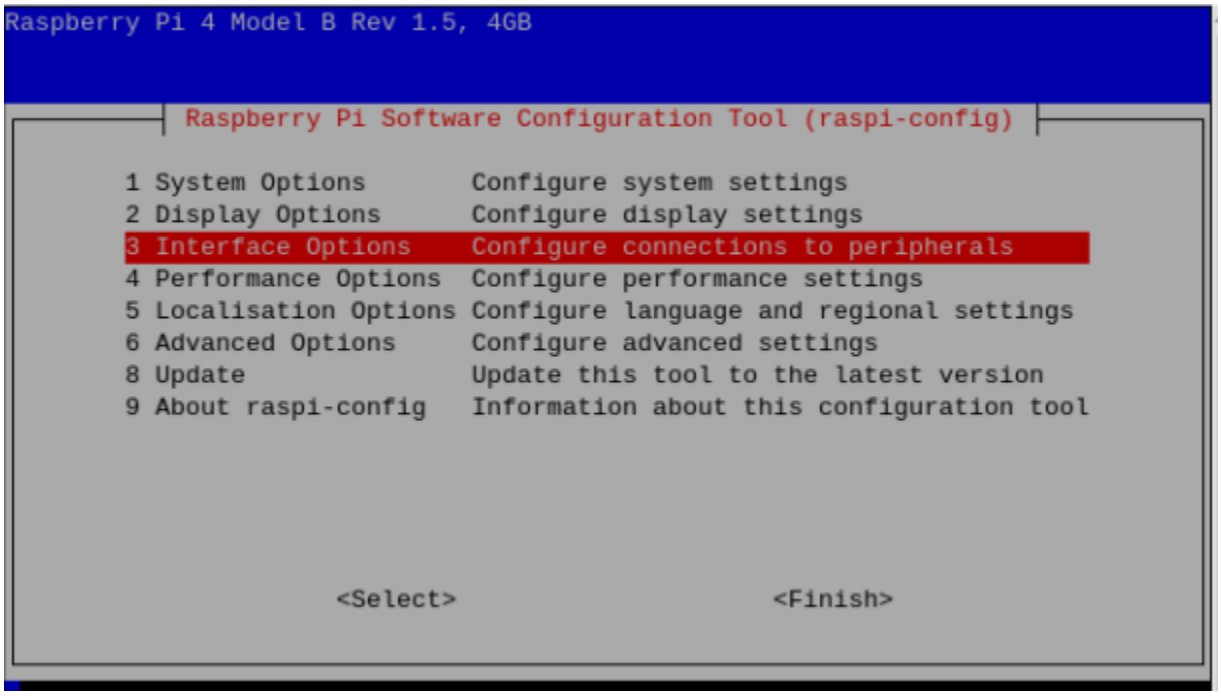
The LoRA-Pi modem should restart and fall back into it's KISS mode. It is now ready for normal operation either through the USB port or the UART pins port.

7. Configure Raspberry Pi for UART Connection

1. Start with a micro SD card flashed with the latest Raspberry Pi OS 64 bit. (as of writing this, Trixie is the latest version)
2. The LoRA-Pi should already be installed using the UART pins on the 40 pin header
3. Boot the Raspberry Pi and either SSH into it or use VNC remote desktop to access the Raspberry Pi.
4. Run the command "sudo raspi-config"

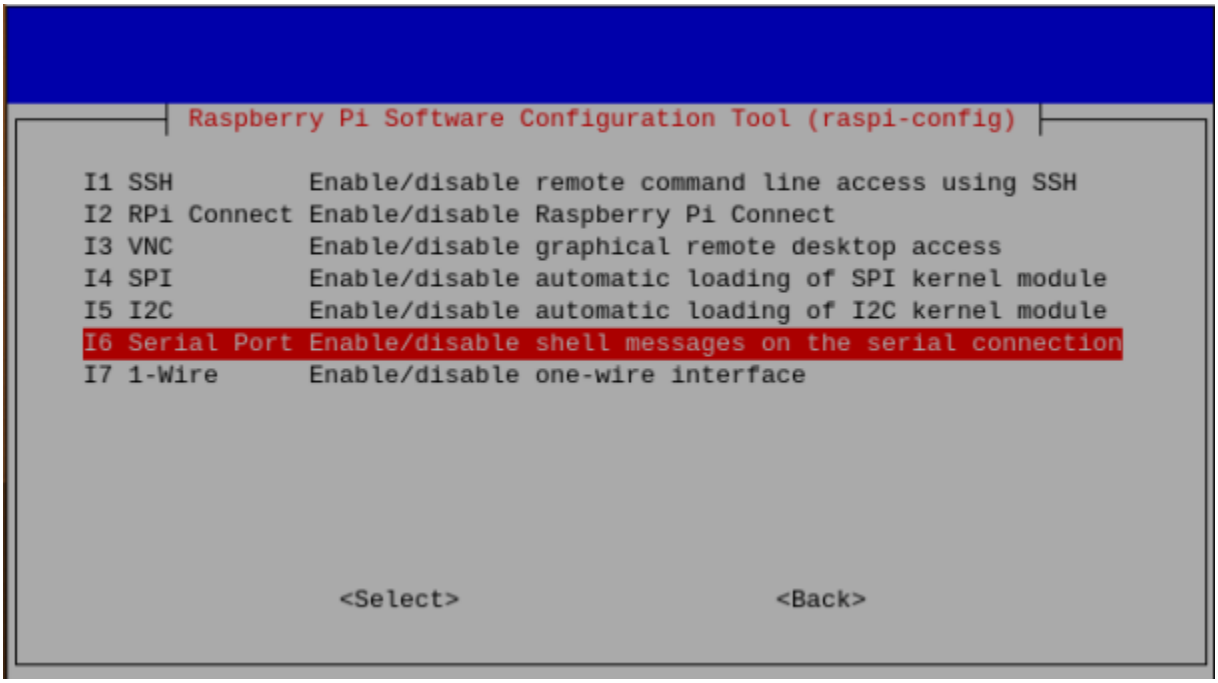


5. This should open a menu. Arrow down to select "Interface Options"

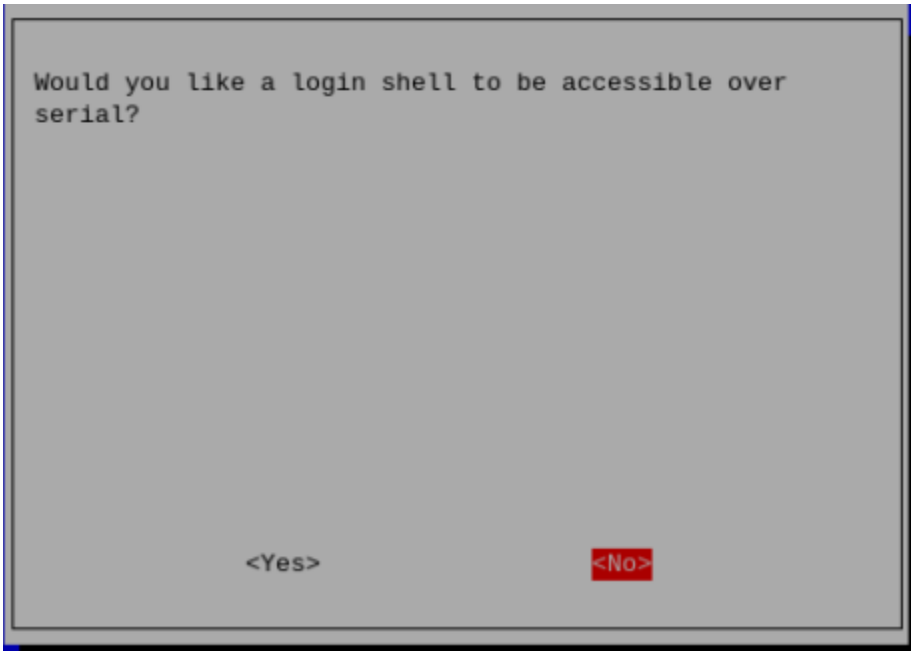


6. Tab to "<Select>" and hit the enter key to select that menu option

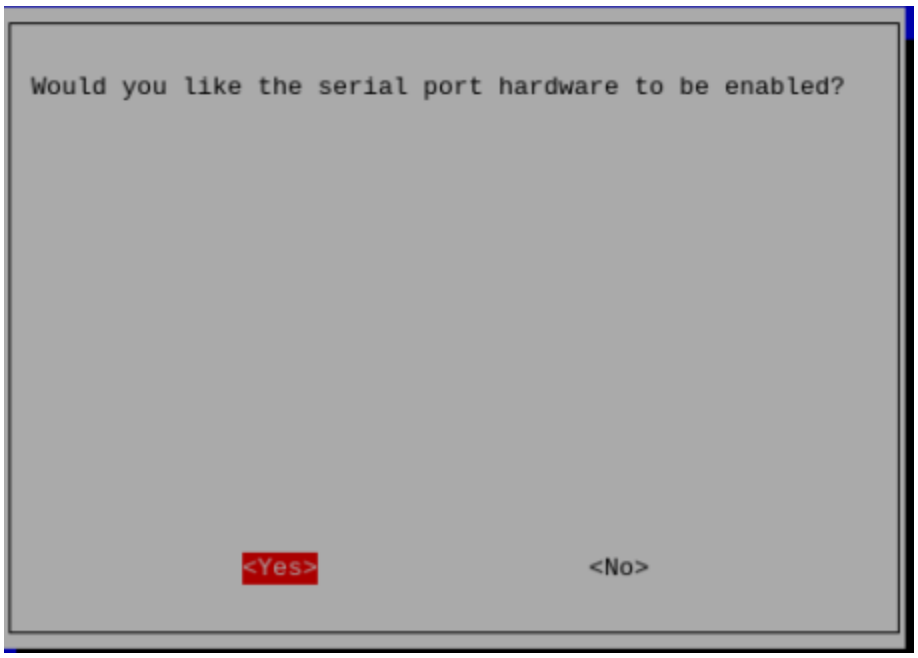
7. This will open another menu. Arrow down and select “Serial Port”



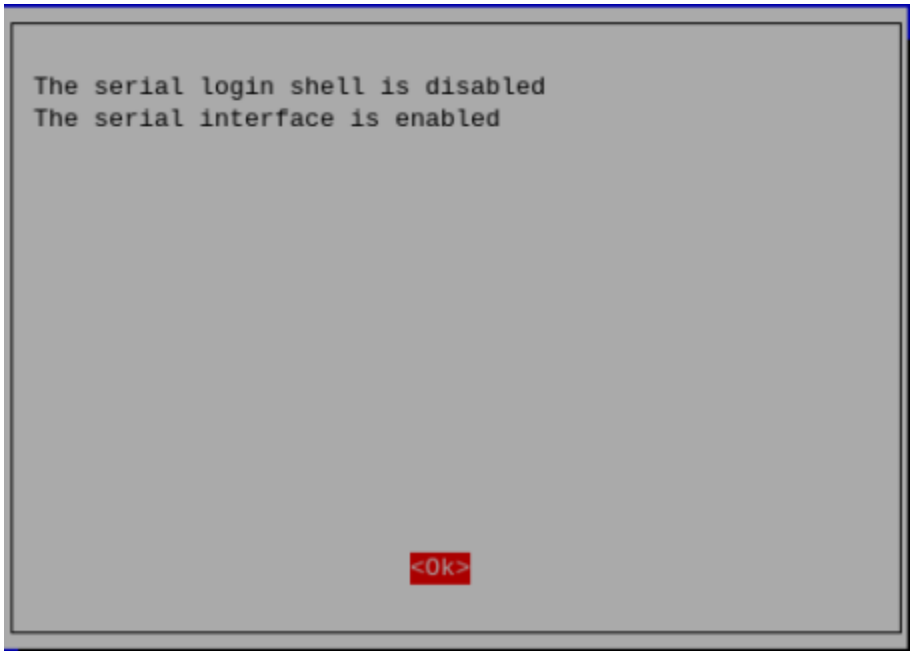
8. Tab to “<Select>” and hit the enter key to select that menu option
9. The next window will ask if you want the login shell accessible over serial. Select “<No>”



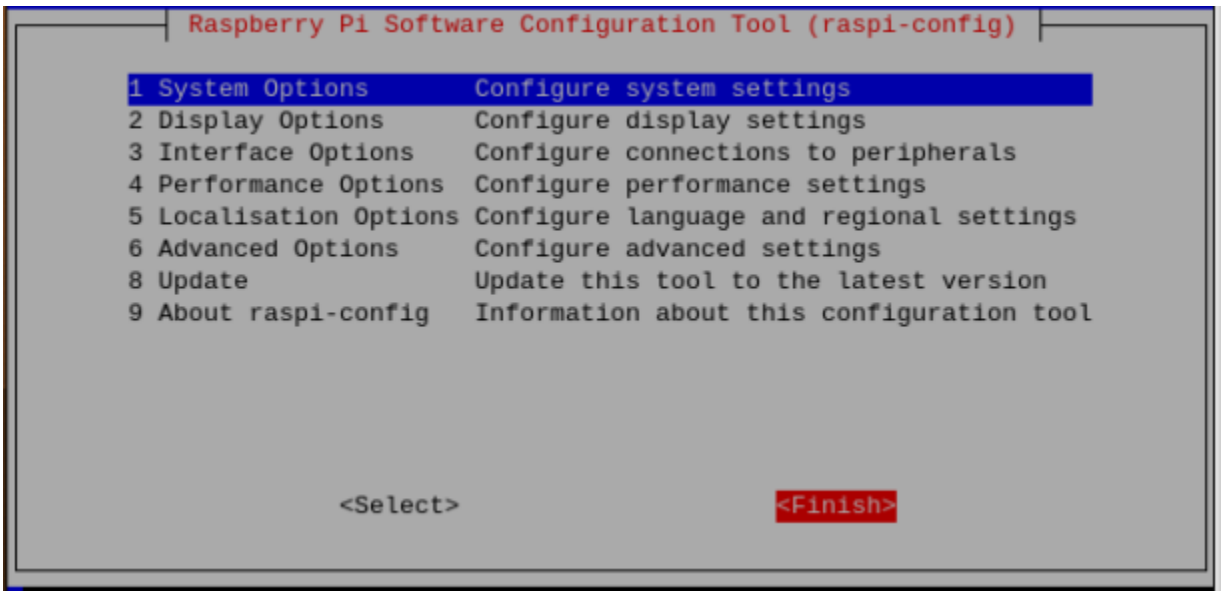
10. The next window will ask if you would like the serial port hardware enabled. Select “<Yes>”



11. The next window will just confirm the two choice you made.

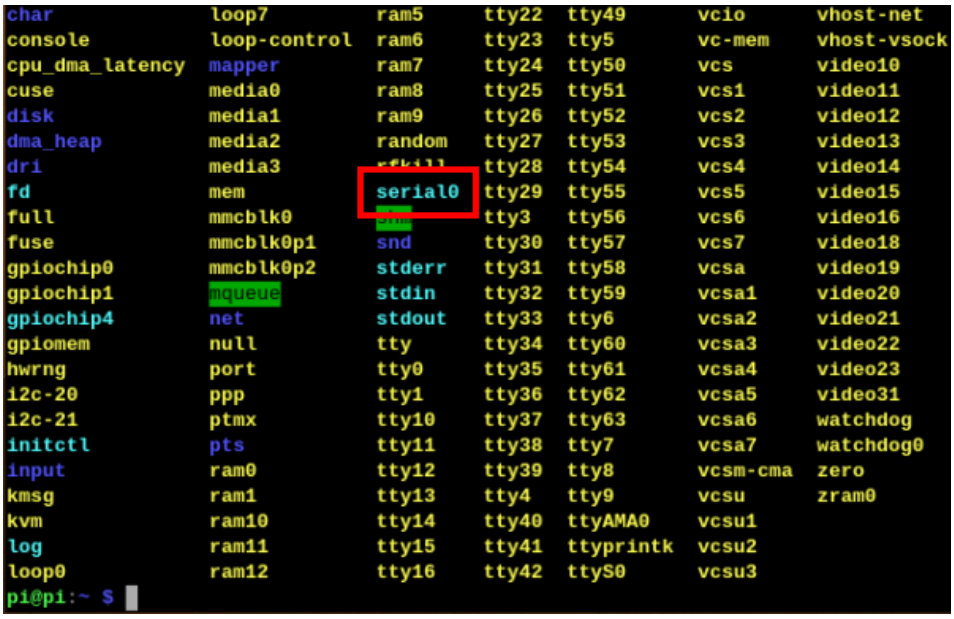


12. You will be returned to the previous menu. Tab to “<Finish>” and hit the Enter button.



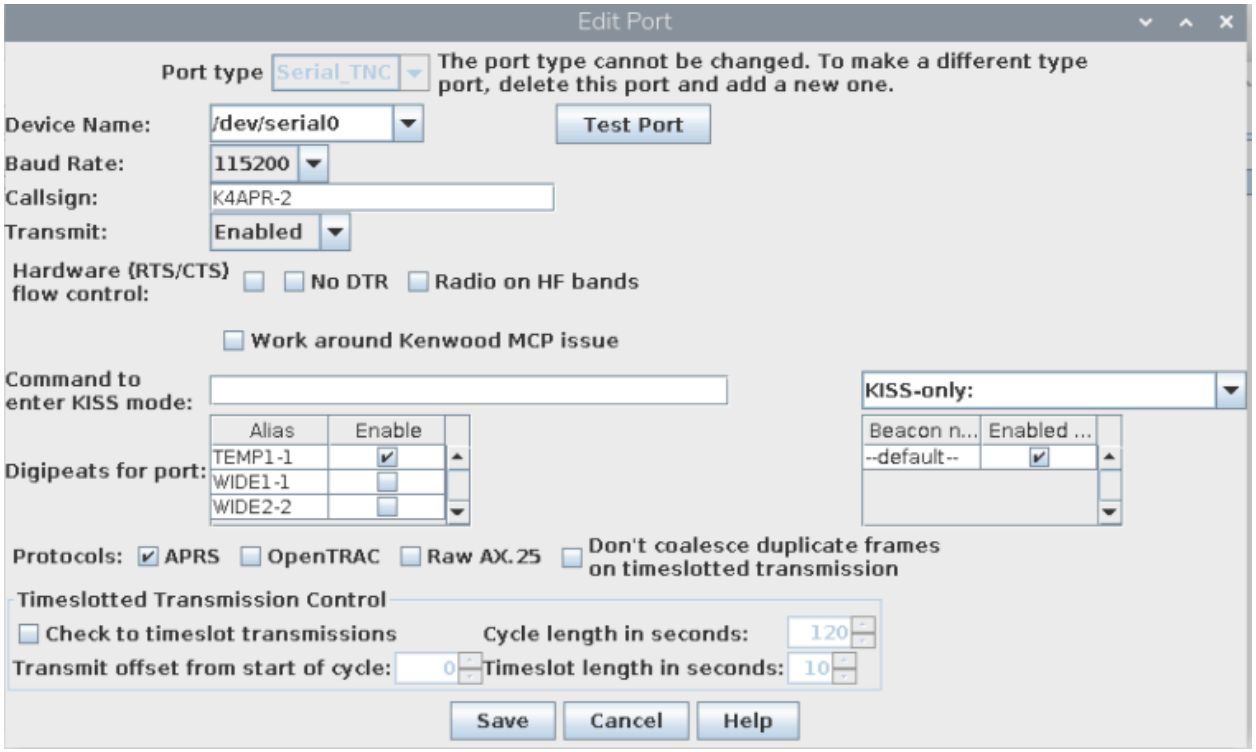
13. You will be returned back to the black Linux terminal window.
14. To confirm your serial port now is enabled, run the following command:

15. Look for “serial0” in the list:



16. From here on, you'll use "/dev/serial0" to reference this port in any software that needs to access the LoRA-Pi modem.

Here is an example of setting up the LoRA-Pi modem in YAAC

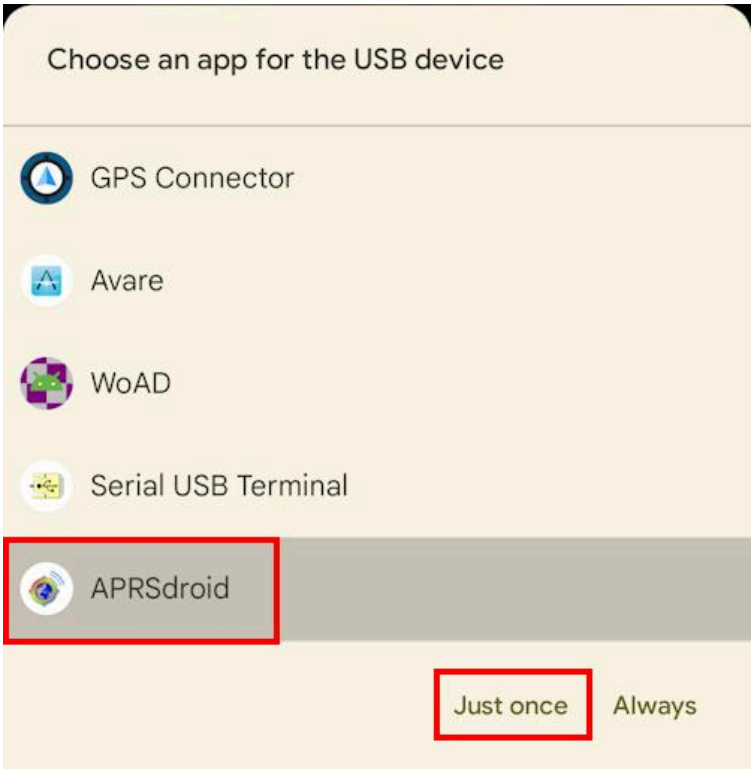


8. Configure APRSDroid

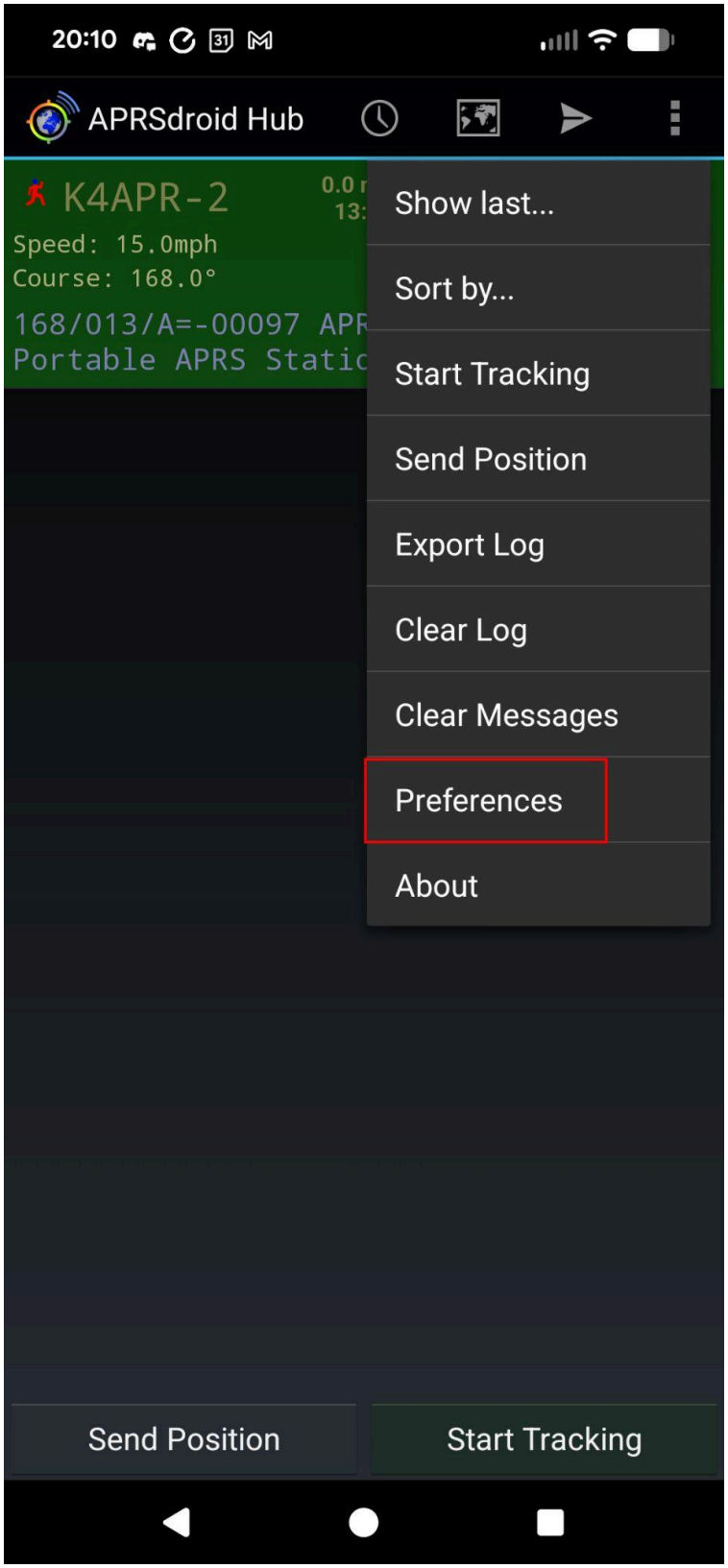
If you have an Android device, a USB-C to USB-C cable is all you need to run mobile/portable LoRA APRS with the LoRA-Pi modem. The FTDI USB interface drivers are built into Android OS.

Continued on the next page…

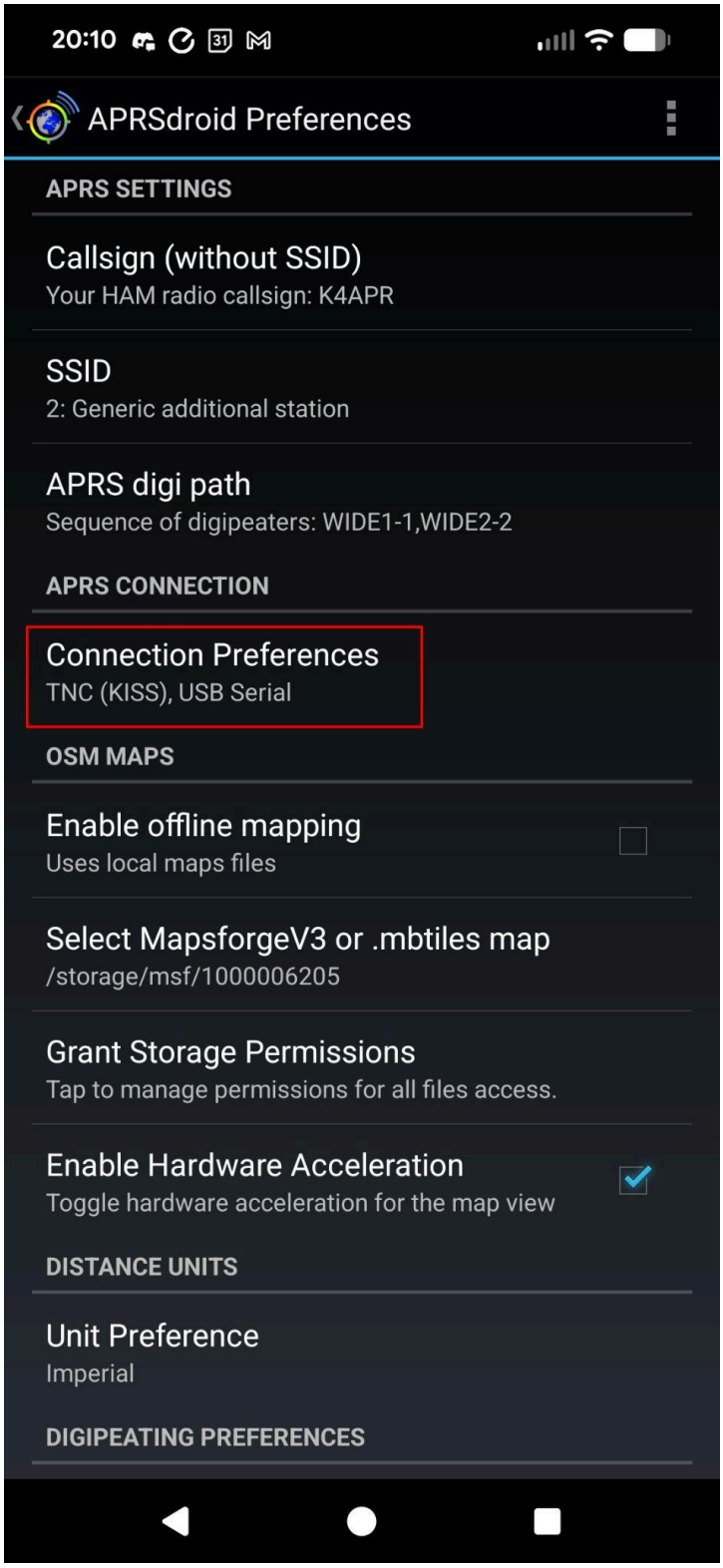
Plug the LoRA-Pi into your Android device and you'll likely get a pop-up like the one below asking which app you would like to use it with. You can also make the decision to open that app automatically when the LoRA-Pi is connected. If you don't want that to happen, choose the "Just Once" option.



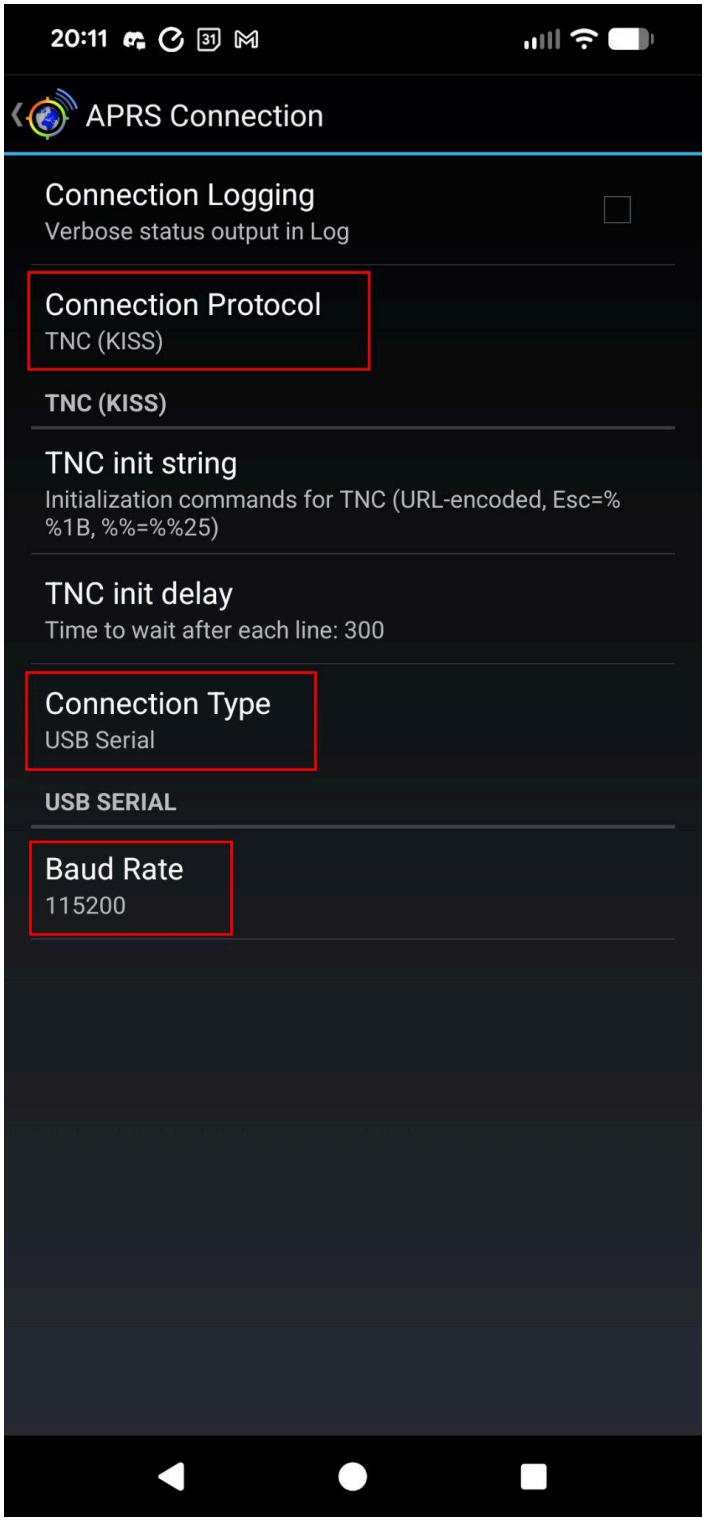
Open APRSDroid and go to the preferences using the top right menu icon



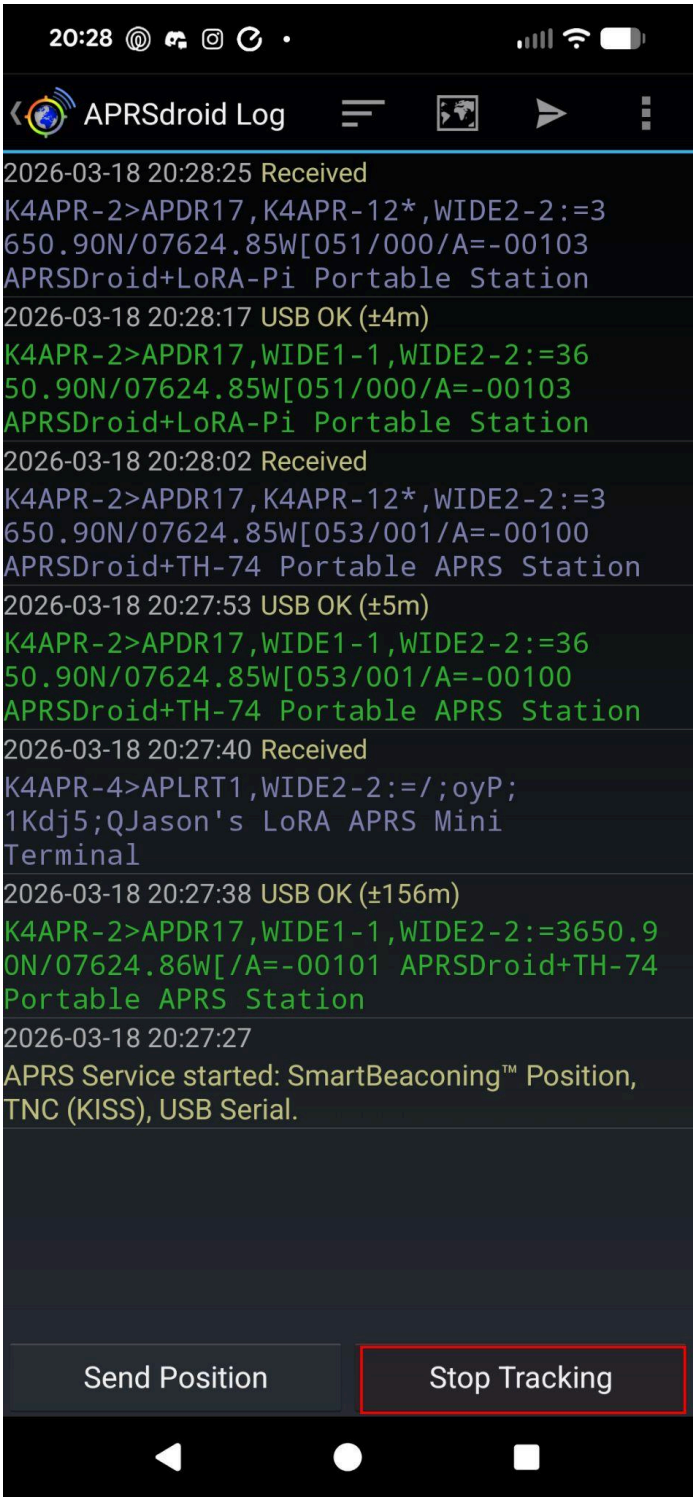
Select “Connection Preferences”



Under the APRS Connection menu, set Connection Protocol to TNC (KISS), Connection Type to USB Serial (this might already be selected automatically) and Baud Rate to 115200



Return to the main screen and press the “Start Tracking” button in the bottom right. It will change to “Stop Tracking”. On this screen you can observe live packet data being sent and received



Use the Station List screen to see all stations that have been received over RF