# LoRa Tracking

Although RTTY works extremely well, it does require some fairly expensive hardware at the receive end (an SDR costing £150 plus a PC, or a “real” radio at £300+ new with a PC or tablet). It also requires some skill in setting up.

For some years there have been low-cost transceiver modules on the market, that combine a transmitter (needed at the balloon end) and receiver (needed on the ground) in a module costing about £5. However the range has been poor and they’ve not been good enough to use for HAB. This all changed with the introduction of LoRa (Long Range Radio) modules, which offer very similar range to RTTY whilst also offering the possibility of higher data rates (perfect for sending down images). LoRa modules interface via SPI and can be used with any processor that has an SPI port (e.g. Arduino, Raspberry Pi). Given that the first thing we want to do with any received data is upload it to the internet, the Pi is an ideal option here.

## 

## 

## Transmitting LoRa

The standard PITS software includes support for LoRa, and just needs to be enabled.

The PITS LoRa board has 2 positions available - CE0 and CE1 - and these can be configured separately. It’s possible to use both modules but in most cases you would only use one.

As usual, configuration is in /boot/pisky.txt. The default contents disable LoRa but you just need to un-comment some lines to enable a LoRa module:

LORA\_Frequency\_0=434.450  
LORA\_Payload\_0=PI\_SKY\_LORA  
LORA\_Mode\_0=1  
LORA\_Binary\_0=N

Frequency should be obvious by now :-). Payload is the ID sent, and is best set to 6 characters or less if you are sending images (the maximum image payload name is 6 characters). Mode should be 1, and binary N. Any other LoRa settings in the file can be left commented out.

The above are for CE0; to configure CE1 change all the “\_0” parts to “\_1”.

With the above changes saved, restart the tracker. You can either:

1. Restart the Pi with “sudo restart”
2. Restart the tracker program with “sudo killall tracker” followed by “cd ~/pits/tracker” and then “sudo ./tracker”

**Receiving LoRa**

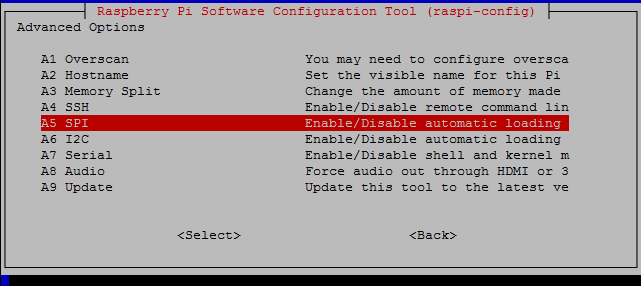
Physically install is very simple - just push on to the Pi model A+ or B+, using a standard pin header extender (supplied). Or an pin header with extended pins can be used if you want to stack another board on top.

Next, burn an operating system onto a suitable SD card. For this purpose anything from 4GB should be fine. The following instructions are for Raspbian, and no other operating systems have been tested.

First, run raspi-config:

sudo raspi-config

Then expand the filesystem, as you would normally do, then choose Advanced Options --> SPI and enable SPI.



It's also worthwhile to change the hostname (Advanced Options --> Hostname). Finally, close the program and choose the reboot option.

Once rebooted, login again. We now have some software to install. First, install wiringPi, which is used for the SPI library and to read the status of the LoRa module via 2 of its INT pins:

cd ~  
git clone git://git.drogon.net/wiringPi  
cd wiringPi  
./build

Next, install SSDV which is used to decode download images:

cd ~  
git clone https://github.com/fsphil/ssdv.git  
cd ssdv  
sudo make install

The gateway software uses the curl library for internet access (uploading telemetry data and/or image data), so install that:

sudo apt-get install libcurl4-openssl-dev

and the ncurses library used for the screen display:

sudo apt-get install libncurses5-dev

Finally, install the gateway software itself:

cd ~  
git clone https://github.com/PiInTheSky/lora-gateway.git  
cd lora-gateway  
make

That completes the installation, so now for the configuration. The main settings are in a file gateway.txt in the above folder (/home/pi/lora-gateway). Here's a simple example:

tracker=MYCALLSIGN  
  
frequency\_0=434.451  
mode\_0=2  
  
#frequency\_1=434.450  
#mode\_1=0

This firstly sets your callsign, which if you are a radio amateur would normally be your radio callsign, but it can be something else.

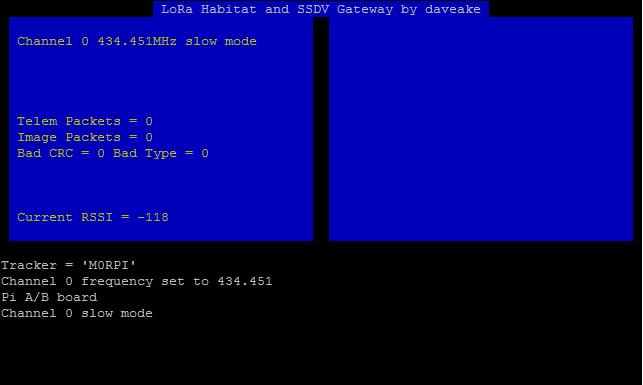
The next part sets the frequency and mode for the first LoRa device (the one in position "1"). Frequency is in MHz and should match the frequency of the tracker that you intend to receive. "Mode" is described below, where the other (optional) settings are also described. The final lines are commented out, to disable the second LoRa module.

tracker=<callsign>. This is whatever callsign you want to appear as on the tracking map and/or SSDV page.  
  
EnableHabitat=<Yes/No>. Enables/disables telemetry upload to habitat.  
  
EnableSSDV=<Yes/No>. Enables/disables image upload to the SSDV server.  
  
frequency\_<n>=<freq in MHz>. This sets the frequency for LoRa module <n> (0 for first, 1 for second). e.g. frequency\_0=434.450  
  
mode\_<n>=<mode>. Sets the "mode" for the selected LoRa module. This offers a simple way of setting the various  
 LoRa parameters (SF etc.) in one go. The modes are:  
  
 0 = (normal for telemetry) Explicit mode, Error coding 4:8, Bandwidth 20.8kHz, SF 11, Low data rate optimize on  
 1 = (normal for SSDV) Implicit mode, Error coding 4:5, Bandwidth 20.8kHz, SF 6, Low data rate optimize off  
 2 = (normal for repeater) Explicit mode, Error coding 4:8, Bandwidth 62.5kHz, SF 8, Low data rate optimize off  
 3 = (normal for fast SSDV) Explicit mode, Error coding 4:6, Bandwidth 250kHz, SF 7, Low data rate optimize off  
  
SF\_<n>=<Spreading Factor> e.g. SF\_0=7  
  
Bandwidth\_<n>=<Bandwidth>. e.g. Bandwidth\_0=41K7. Options are 7K8, 10K4, 15K6, 20K8, 31K25, 41K7, 62K5, 125K, 250K, 500K  
  
Implicit\_<n>=<Y/N>. e.g. Implicit\_0=Y  
  
Coding\_<n>=<error\_coding>. e.g. Coding\_0=5 (4:5)

To run, just type

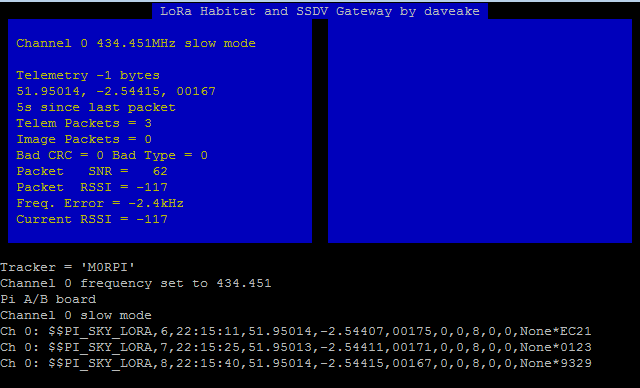
sudo ./gateway

and you will see a screen like this:



In this example, channel 0 (the LoRa module in position 1 on the board) is enabled, at 434.451MHz, and mode 0 which is a shorthand for a mode suitable for continuous long-range telemetry transmission.

The status display updates as packets are received:



Finally, to exit the program, press Q or CTRL+C.