# RTTY Tracking

RTTY is old. It’s almost as old as the hills that the radio signal won’t pass through.

RTTY is also very simple. It’s basically a radio version of good old RS232, with the ones and zeroes being represented by 2 slightly different radio frequencies. The “1” (Mark) is typically the higher frequency, and “0” (Space) the lower frequency. Each byte begins with a start bit (space), followed by the byte itself (usually 7 or 8 bits), then an optional parity bit, and finally a stop bit. Any gap between bytes simply extends the stop bit.

It’s vital that the transmitter and receiver to completely agree about the above format, because any difference will mean garbage being received. So all settings (timing - baud rate, number of data bits, etc.) must agree 100%.

## Transmitting RTTY

So to create an RTTY-encoded radio signal from the Pi, all we need to do is connect the Pi serial port to a suitable radio device. We want the “1” and “0” values (which are on the Pi voltages of 3.3V and 0V) converted to two different radio frequencies. In other words, we want to modulate the frequency of the radio signal from a voltage. This is Frequency Modulation (FM) as we discussed earlier, and we want an FM transmitter that accepts a voltage input to modulate the output frequency.

The Radiometrix NTX2B is such a device.

If we send the full Pi voltage (a swing of 3.3V) to the NTX2B, the output will change by many kilohertz. This is too much, as we will see later, so we reduce it to a few hundred Hertz using a potential divider circuit that reduces the 3.3V swing to around 0.5V.

## Receiving RTTY

This is rather more complex, and needs more kit:

* Radio Receiver - needs to be able to receive between 430MHz and 440MHz (also known as the 70cm band (as the wavelength is 70cms long)) and also be able to receive SSB (single side band), it is important to note that this isn't FM, many radios only receive FM so look for SSB (which may be described as USB and LSB). Examples of radios used by present UKHAS members include the Yaesu FT-790R - old but amazing, can sometimes be found on ebay for around £100, Yaesu FT-817(ND) - probably now the favourite, very wide range of receive and transmit and is portable due to a battery. approx £450, AOR AOR8000 - Wideband Receiver (doesn't do Transmitting), Yupiteru MVT-7100 - Wideband receiver (doesn't do Transmitting), ICOM IC-R10 - Wideband receiver (doesn't do Transmitting)
* Or an SDR (Software Defined Radio). These are increasingly popular, starting at the cheap but deaf end with TV USB dongles, and working up to devices such as Hack RF that can also transmit. For our purposes though, a Funcube Dongle Pro+, AirSpy or SDRPlay is perfect.
* Suitable aerial - e.g. 2m/70cm magmount for the car; 2m/70cm colinear for a base station, or 70cm Yagi to help pick up a weak signal after landing.
* Computer with a soundcard and an audio in/mic in running Windows, Mac OS X or Linux. Or an Android tablet of phone.
* Cable to connect the radio to the computer - usually a 3.5mm jack (plug) to 3.5mm jack (plug) (should be 2-pole (mono) connectors but you can sometimes successfully utilize easier to find ready made stereo (3-pole) cables.
* Internet connection, anything from mobile broadband to normal broadband will do
* For a PC, dl-fldigi - an adapted version of the free fldigi soundcard decoding software, this takes the audio that your radio outputs and decodes the balloons signal, it then sends the telemetry it's found over the internet to our server which plots the payloads position on to a map. For Android “HAB Tracker” does the same thing.

### Tuning In

The following assumes use of a real radio and a PC. First, switch the radio on and set the frequency to whatever you expect the transmitter to be using (e.g. the frequency setting in /boot/pisky.txt). Make sure the receiver is set to USB mode, that Squelch is turned off (when it’s on it cuts the audio output if there is no or little signal), and that the volume is set to a comfortable level.

Assuming that the transmitter is on you should now hear the signal; if not then turn the frequency dial up/down a bit.

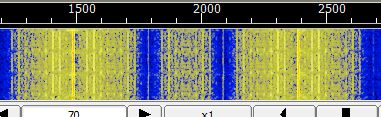
If the transmitter is close then it’s possible to miss-tune but still hear the signal. It will sound OK but will not decode (because the high/low tones are swapped). So, when you can hear the signal, you need to run a quick test. Turn the frequency down a bit and the signal should go up in frequency. If instead it goes down, then continue turning the dial (in the same direction) until the signal disappears and then reappears.

Finally. set the frequency dial so that the signal is fairly low in tone (not screechy) and now you are ready.

### Decoding

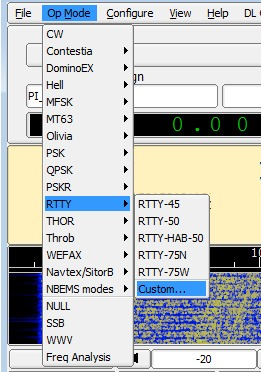
Run dl-fldigi. If this is the first time you’ve run it on this PC you’ll need to enter some settings, such as your callsign (your name or school name for example), latitude and longitude, and which audio device you are using for input (Line In).

With dl-fidigi running, connect the radio receiver output jack to the PC line in jack using a standard 3.5mm audio cable (mono or stereo). You should then see the received signal as a “waterfall” in dl-fldigi.

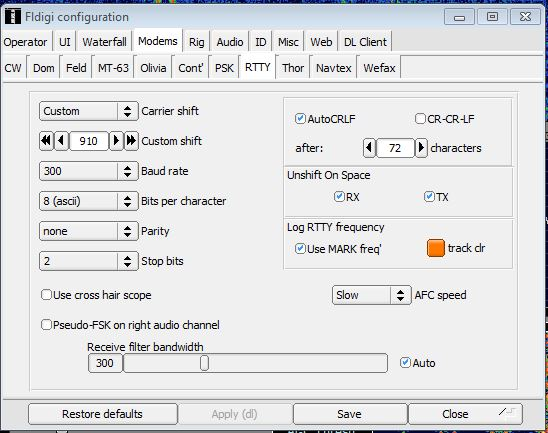


The waterfall shows audio tones from low on the left to medium on the right. High notes are not shown at all which is why the above advised you to set the receiver frequency so that the signal is not too high. With the receiver frequency set correctly you should see 2 vertical lines (yellow normally, red if the volume is too high) in the waterfall; adjust the frequency if you only see one of them.

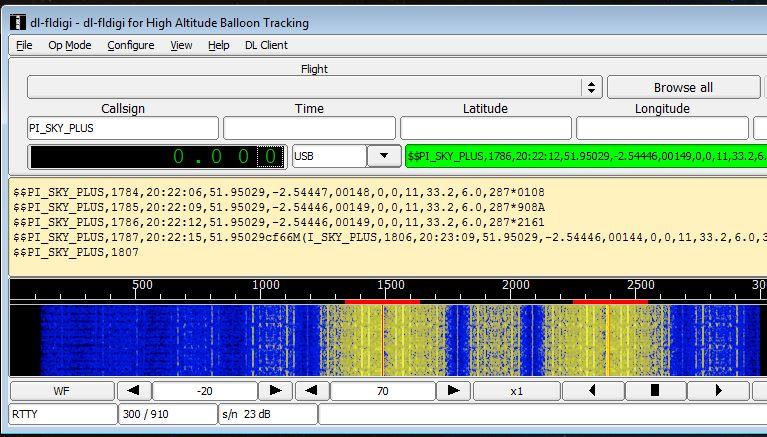
Now that you have the signal in dl-fldigi, you should set dl-fldigi to the correct RTTY mode. In the menu, choose Op Mode → RTTY → Custom:



and then set 300 baud, 8 data bits, no parity, 2 stop bits, audio shift 910Hz (these are all PITS defaults).



You will then see 2 vertical red cursor lines in the waterfall. Click once halfway between the 2 yellow signal lines, to align the red cursor lines with the yellow signal lines. You should then start to see decoded data in the window above the waterfall.



I suggest you do this with the camera disconnected or disabled, so you only have text in that window, which makes it easy to see if it's working or not. You can see above that the text window has telemetry from the tracker - GPS location etc.

If you do send images, you don't have to do any more setup - dl-fldigi will decode the image data automatically. You can view those images using the menu option View --> SSDV.