HAB Tracker Receiver Program Notes

Stuart Robinson – August 2016

The receiver program is the companion for the LoRa HAB tracker transmitter, intended to receive the transmitted LoRa payloads from a balloon in flight. The receiver program will run on one of my tracker PCBs but it may be more convenient to use the larger (but still only 50mm x 50mm) shield PCB. To use the receiver program with the tracker transmitter PCB put a '#idefine PIHTRACKER3 line at the start of the program. To use the shield PCB put a '#define PIHSHIELD' line there instead.

The interface between the user and the receiver program is via a serial terminal program at 38400baud, 8bit, 2stop bit, no parity. I normally use Putty on PC and have all output logged to a time and date stamped file, but Teraterm is OK as well.

When a UKHAS style payload is received, it can optionally be sent via an AFSK RTTY audio uplink into HAB FLDIGI running on a PC (or other device) and thence onto Habitat for on-line tracking.

The tracker transmitter and receiver normally use LoRa at a bandwidth of 62.5Khz and as long as the transmitter and receiver frequency adjustments are set so that the frequency is within a 1khz or so at room temperature, no other frequency adjustments are needed and the receiver will be quite happy to operate unattended.

As explained in the transmitter software overview my HAB tracking software operates a command and control interface between transmitter. The transmissions to the tracker transmitter are at a lower power but longer range set of LoRa parameters so that when used with an approx 5dB gain vertical antenna at the receiving end the 10dBm ERP transmit limit is not exceeded.

To setup the receiver for a particular flight, copy the flight_settings.h file from the transmitters program folder into the receivers program folder and program up the receiver.

Frequency Calibration.

Both tracker transmitter can be setup to transmit a tone at startup, this lasts about 3 seconds and can be used to allow the transmit frequency to be measured, I use a small hand held frequency counter.

There are variations in the crystals used on the LoRa modules and at 434Mhz, a difference of 8Khz between two modules has been seen. At the LoRa bandwidth normally used in the tracker programs, 62.5Khz, the allowable variation is 15.625Khz so calibration adjustments are not normally needed. You can carry them out if you wish however.

Once you know how far out a transmitter or receiver is in frequency a compensation offset (plus or minus) can be added for each program. For the transmitter program look for this line in Flight-Settings.h;

const float const_TrTXFreqOffset = 0.0; //tracker transmitter frequency offset Mhz

For the receiver program look for this line in the receiver program;

const float const_RXFreqOffset = 0.0; //tracker receiver frequency offset Mhz

If you have an SDR that is accurate you can use that to measure the frequency of the tone being output on the software's display window, although if the transmitter or receiver board is nearby when being measured you will probably need to remove the aerial connection from the SDR in order to reduce the incoming signal level.

Memory Usage

The LoRa library routines are different between transmitter and receiver. The transmitter program uses a common buffer for transmit and receive packets. This is to reduce RAM usage and assist in reliability of the tracker. The receiver program can run on a Pro Mini with an ATMEGA328P processor, but when used on an ATMEGA2560 there are very large amounts of free program space and memory.

Using the Receiver

Program up the receiver, it starts up in flight receive mode ready to accept incoming UKHAS style payloads. You can leave it as is and received payloads are all logged to screen or optionally sent to HAB FLDIGI via an AFSK RTTY uplink.

Press ESC and the following menu is displayed, explanations of each of the options are explained below. The default start-up option of the tracker is FSK RTTY Enabled, Fence check disabled, Address Strip Enabled. If these setting are changed remotely they will survive resets of the remote tracker.

Note that the requests to change tracker options are queued and will automatically retry if not acknowledged. To save power the tracker transmitter only listen for incoming

commands for a very short period of 2 or 3 seconds. At the start of the listen period the tracker transmitter sends a CTS packet which the receiver with its queued command request is waiting for. The receiver then sends the command and waits for an ACK packet from the tracker transmitter. If no ACK packet is received the receiver program will continue to queue the command. A queued command can be cancelled by pressing a key on the terminal programs keyboard.

The receiver program is fully operational, the next update to the software will be to include an option to program the transmitter settings in full from the receiver.

AFSK Audio Uplink

Be sure the #define AFSKRTTYUPLOAD line is there at the top of the program

The fastest baud rate you can used will depend on the PC sound card and Operation system. I have reliably used 1200baud via a USB plug in sound adapter on a low specification notebook. The AFSKRTTY.h file has the numbers to use for different baud rates.

For the tracker transmitter boards (PIHTRACKER3) fit components C3 and R5 then connect the PC audio input to pin 8 on Con2 and ground to pin 6 on Con2

For the tracker transmitter boards (PIHSHIELD) fit components C5 and R1 then connect the PC audio input to pin 7 on ConB and ground to pin 6 on ConB

Stuart Robinson
GW7HPW
August 2016