Program Notes - LoRa Tracker Test Program 180816

This program is a collection of routines that can be used to check that a tracker board is working correctly. Use this program before loading up the main tracker programs, it can be difficult to troubleshoot board problems with the full tracker program loaded.

As far as possible all appropriate functions on a board are tested and the tests can be run on a partially assembled board.

The test program needs the NewTone library installed;

https://bitbucket.org/teckel12/arduino-new-tone/wiki/Home

The standard Arduino tone library is very poor at producing the audio tones used by AFSK RTTY.

Ensure the appropriate board type is selected at the start of the program, such as;

#define PIHTracker3

The 'Tracker_definitions.h' file has some notes on how to identify a particular board type.

Also check that an appropriate frequency is selected;

#define Frequency1 434.400 //main frequency in MHz

Also ensure that the correct processor type is set, the test program will run on either ATMEGA328 or ATMEGA2560 boards.

#define ATMEGA328 //commment in if using ATMEGA328
//#define ATMEGA2560 //commment in if using ATMEGA2560

When the program is loaded switch to the Arduino IDE serial monitor at 38400baud. At start-up or reset you should see the program name and version number as defined at the beginning of the program, printed to the monitor;

#define programname "LoRaTracker Test Program"
#define programversion "V1.2"
#define dateproduced "03/08/2016"
#define aurthorname "Stuart Robinson"

The tests then proceed in a sequence that repeats, the test order is;

LED Flash

The board LED and the Pro Mini pin 13 LED should flash at close to once per second, if its faster or slower there is a problem with the Pro Mini.

CPU Temperature

The software uses some internal CPU registers to estimate the CPU temperature. This will likely be widely out, it needs to be calibrated for each Pro Mini to be even close to accurate. If you know the room temperature then you can adjust this program constant to give a more correct reading;

const float kelvin_offset = 312; //if temperature reports high, increase this number
//if low then reduce it

For better accuracy over a wider temperature range if you know the actual ADC reading at a particular temperature you can calibrate each pro Mini for significantly improved accuracy.

The test program reports the variable (wADC) that the temperature calculation routine reads from the CPU. It then uses this calculation to establish the temperature;

temp = (wADC - kelvin_offset) / (temp_conversion_slope);

If you collect the wADC readings from two different temperatures, say when the tracker is on the bench and after its been in the freezer a while, you can enter the results in the spreadsheet;

'ATMEGA328 Internal CPU Temperature and Voltage Measurements Calibration.xls'

And it will calculate from the actual wADC readings, what the slope should be. You can then manually adjust the field 'Manually Adjusted_kelvin_offset' (it's the yellow one) and the spreadsheet uses the provided wADC readings and temperatures with the same calculation as above to show what the new reported temperatures would be. Adjust the offset so that the temperatures reported closely match the two temperatures at which you recorded the wADC readings.

The spreadsheet contains an example taken from one of my own trackers, the high wADC reading at 26C was 354, and the low wADC reading at -21 was 305.

CPU VCC

The program uses the CPU itself to measure its own supply voltage, this may not be accurate. With a normal 3.3V Pro Mini you ought to get a reading of 3300mV, the reported value can be adjusted by varying this parameter for each individual board;

const long adc_constant = 1100000; //if voltage reports low increase this number
//if voltage reports high decrease this number

In the example in the above mentioned spreadsheet, I measured the actual VCC as 3290mV, whilst this was reported as 3204mV, so the constant was adjusted to;

(3290/3204) * 1100000 = 1129526

Supply Volts

This uses the resistor divider that is across the RAW input of the Pro Mini to read it's voltage, this is normally the battery voltage.

Check LoRa device

This sets up the SPI bus and does a write and read over the bus. If the value written is read back correctly then the routine will program the LoRa device and transmit an FM tone on the selected frequency. If you hear this tone on a UHF hand-held then the LoRa device is working, and the tone output of the CPU to DIO2 pin on the LoRa device is connected correctly.

After the tone there is a printout of all the LoRa devices registers.

If the LoRa device is not detected, the LEDs will flash very rapidly for a short while.

AFSK RTTY test

This sends the test string at the beginning of the program as AFSK RTTY audio out as it would for the HAB receiver. It is audio only and will not be heard on a UHF hand-held. This test is primarily designed for testing boards that are set-up as HAB receivers. You will need to have the resistor and capacitor fitted that are in series with the audio out pin, see the appropriate schematic.

Radio Control Pulse

For trackers being used as RC lost model trackers this test reads the RC servo pulse and reports the pulse width value which should be in the range of 1000uS-2000uS.

I2C Bus Scan

If no RC pulse is detected then the I2C bus is scanned for active devices. If a Ublox GPS is fitted a device should be detected on 0x42. If a BME280 or BMP280 sensor is fitted a device should be detected on 0x72.

Stuart Robinson
GW7HPW
August 2016