Long Distance Tracking and Monitoring with LoRa

Stuart Robinson

18th April 2016

Part 1 – Introduction

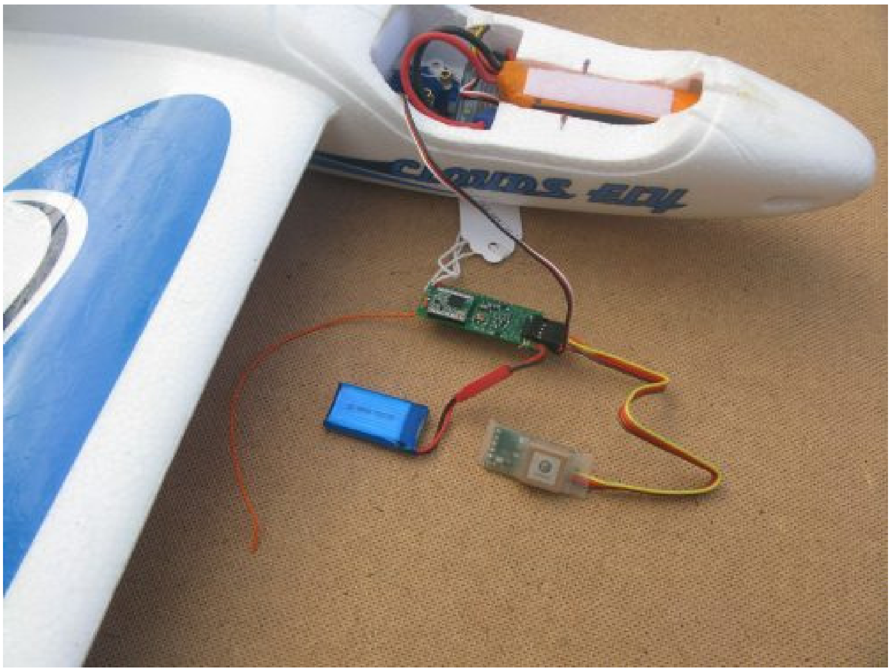
This first article in a series on using LoRa for long range tracking and monitoring provides some general notes on LoRa (**Lo**ng **Ra**nge) radio transceivers and GPS tracker design. Further articles will describes the building of a basic tracker transmitter, the building of receivers for the tracker, several enhancement options for the transmitter and receiver, the use of the board as a high altitude balloon tracker and finally the use of LoRa as a remote monitoring sensor.

The tracker has a small battery powered micro controller with a programme that reads the location information from a GPS module and converts that into digital data which is then transmitted with a LoRa device. A matching receiver translates the received data and displays it as position information either on a map, as Latitude and Longitude or as a distance and direction to the remote transmitter.

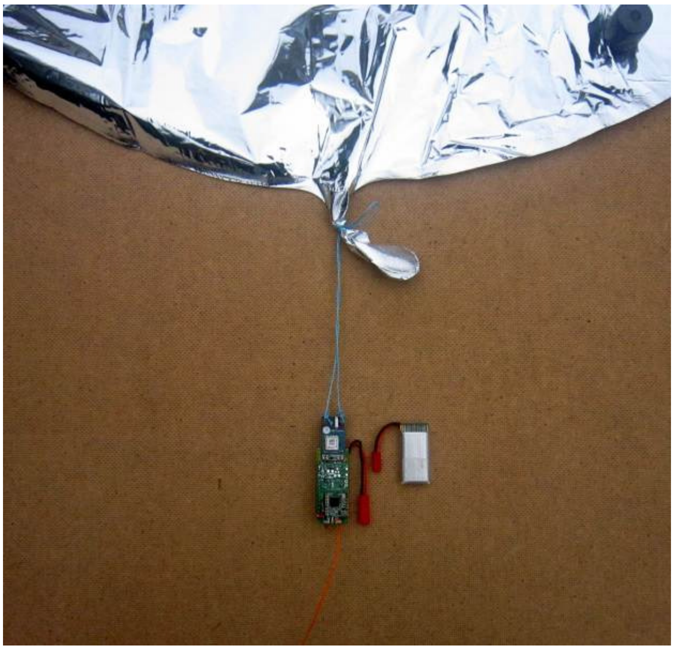
The transmitter and receiver printed circuit boards (PCBs) are small, 21mm x 50mm and designed to be low cost and very easy to build.

The processor used is an ATMEGA328 in the form of a 3.3V Arduino Pro Mini, so there are a large number of program libraries, examples and tutorials available for those that want to explore their own applications.

LoRa tracker as RC 'lost' model finder



Board as a PICO high altitude balloon tracker

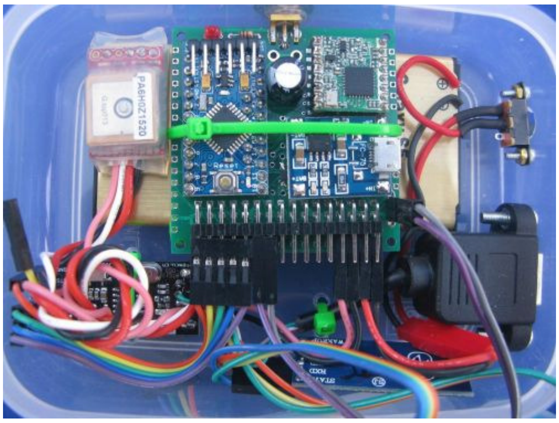


A remote environment sensor

The board would normally be powered from a single Lithium ion or polymer battery which can be charged from a small solar panel or other external 5V supply. A 3V battery can also be used. The boards can implement a high level of power saving; with AA Lithiums as a power source, the board could transmit sensor data every 5 minutes for up to 5 years.

A larger PCB (50mm x 50mm) has been developed, this can be used as a tracker or receiver and has most of the ATMEGA input output pins available on an expansion header. The board is easier to use for a more capable receiver and can either operate standalone with an Arduino Pro Mini fitted or as a LoRa and Arduino (3.3V!) shield.

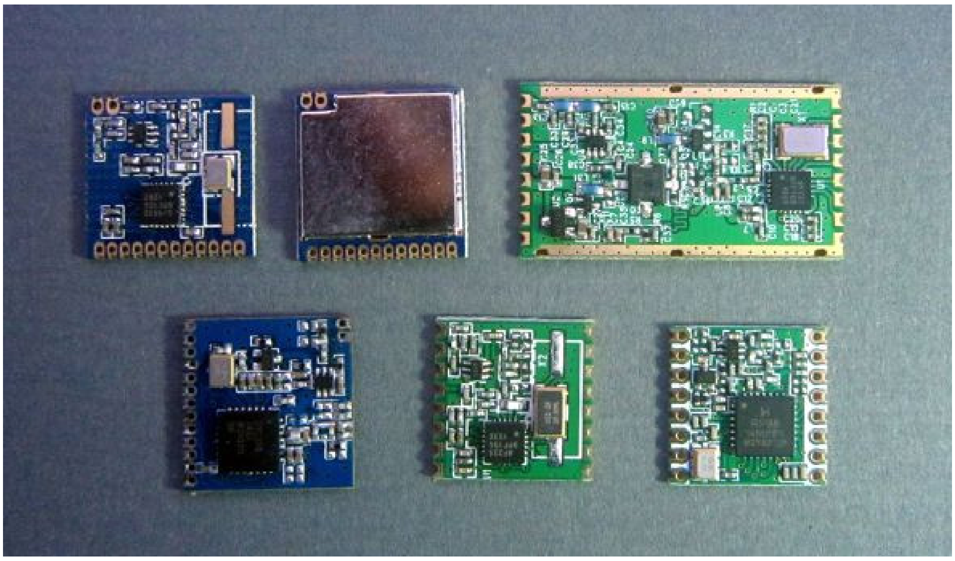
The receiver below is small and self contained, ideal for portable use and traipsing around the countryside looking for your 'lost' model or balloon. It has its own GPS, and will calculate the distance and direction to the 'lost' item, or you can use the Bluetooth link to an Android mapping application. The receiver plugs into a USB port for battery charging and terminal access. The receiver also has a FRAM memory device fitted and it will remember the last received location payload.



Radio Transceiver Modules

There are a number of radio frequency (RF) transceivers that could be used, common choice are devices or modules that operate in the Industrial Scientific and Medical (ISM) bands, 434Mhz or 868Mhz in the UK and other parts of the World, 915Mhz in the Americas. Operation in the ISM band does not (normally) require a licence as long as local transmit power, band spacing and duty cycle restrictions are adhered to.

Some manufactures produce small modules that contain the supporting circuitry a typical RF integrated circuit (IC) needs. Examples are the Hope RFM22B based on the Silicon Labs Si4432 device and the Dorji DRF1278F or Hope RFM98 module based on the Semtech LoRa SX1278 device. These modules are small, low cost and ideal for integration into micro controller based projects, note the relatively large number of tiny surface mount components these modules contain;



The older style ISM modules such as the RFM22B (middle of bottom row) can be persuaded to cover long distances under ideal conditions and they have been used in two small Earth orbiting satellites, $50SAT and T-LogoQube. However, the range of these modules under typical ground based urban conditions is only around 100M or less at the UK ISM limit of 10mW.

A search range that could be only 100M or less in built up areas is not that useful for a GPS tracker or locator, but the introduction of LoRa technology by Semtech in late 2013 changed all that.

During 2014 I was able to carry out a series of side by side comparison of the RFM22B module and two LoRa modules; the Dorji DRF1278F and Hope RFM98. The tests showed that at the same data rate and power, LoRa technology had approximately a 20dB signal gain over the FSK based RFM22B. A signal gain of 20dB represents a range improvement of a factor of 10. This extends a not so useful range of 100M to a far more useful 1KM.

The distances LoRa telemetry can cover under ideal conditions are truly remarkable. Some of the best examples come from using LoRa for tracking High Altitude balloons, particularly in the UK. With a reasonable vertical omni directional receiving antenna, telemetry from a high altitude balloon has been received at more than 400km using only 10mW of transmit power. This range is achieved with very simple and low cost receivers, no expensive and bulky low noise amplifiers are required.

What is LoRa?

LoRa is a type of chirp spread spectrum transmission and has been implemented by Semtech for use in low cost ISM band devices.

For a normal FSK style transceiver (such as the Hope RFM22B) the receiver needs to see signal that is strong enough to be greater than the local RF noise level. Noise levels in the ISM bands are a performance limiter and each device in use adds to the local RF noise level. The total noise a receiver sees is a combination of local RF noise and noise generated by the electronics inside the receiver.

Assume a typical FSK transceiver needs to see a signal that is 5dB above the receiver's noise level in order to decode it. In comparison LoRa technology allows for signals to be received that are as much as 20dB below noise level, thus the LoRa signal can be up to 25dB (5dB + 20dB) weaker than the FSK one and still be received. If LoRa can decode a signal that is 25dB weaker, then the transmitter can be further away than in the FSK example. If you double the distance that a transmitter is away from the receiver then the receiver will see signals that are 6dB weaker, thus the 25dB difference equates to a distance difference of approximately 17 times.

So due in the main to the below noise level performance of LoRa it is able to receive signals that are much weaker and thus much further away than with some previous ISM band modules

LoRa modules are straight forward to drive and very flexible, the internal firmware takes care of all internal adjustments and optimisations. The parameters that need to be configured for LoRa are, Spreading Factor, Coding Rate, Bandwidth and transmit power. These parameters can be easily changed so that you can set-up for low data rate long range or high data rate shorter range.

LoRa is also a two way technology, the tracker transmitter is a receiver too, so if you can pick-up signals from your tracker you can send it commands or exchange data with it also.

Bandwidth

We first need to consider the impact of the bandwidth setting. The LoRa bandwidth is the amount of frequency spread the signal occupies. The bandwidths LoRa can be set to are;

7.8Khz, 10.4Khz, 15.6Khz, 20.8Khz, 31.25Khz, 41.7Khz, 62.5Khz, 125Khz, 250Khz, 500Khz.

A low bandwidth signal, such as 7.8Khz will give the best (longest distance) performance, but with the low cost components used in the Hope and Dorji modules a transmitter and receiver may not talk to each other at this bandwidth. The receiver has a capture range, this is the percentage difference allowable in relation to the bandwidth that the transmitter and receiver can be apart in frequency in order for the receiver to pick up the transmitters signal. This percentage is typical 25% of the bandwidth.

Lets assume the bandwidth in use is 10Khz, the allowable variation is 25% or 2.5Khz and the transmitter is on 434.000Mhz. For the link to work the receiver needs to be on or between 433.9975Mhz and 434.0025Mhz. This is an exacting requirement for a low cost crystal, even if transmitter and receiver are at the same temperature.

Low bandwidths of 20.8Khz have been used for tracking and they do give better performance, but care is needed as devices used at this bandwidth may not talk to each other unless a calibration factor is measured and applied.

A bandwidth of 62.5KHz has been found to be reliable and allows for differences in manufacturing tolerance and temperature between transmitter and receiver devices.

Spreading Factor

This controls the amount of signal processing gain. The spreading factor can be set between 6 and 12. A setting of 6 can only be used with fixed length packets, so has not been used in my applications.

The spreading factor defines the below noise performance. SF6 gives a -7.5dB below noise performance and SF12 a -20dB below noise performance.

The larger spreading factors therefore give more range, the trade off is that higher spreading factor packets take longer to transmit.

Coding Rate

Additional data can be added to the packet to allow for error correction. This can result in small signal gains at the limit of reception. The coding rate can be varied between 4:5 and 4:8, the higher 4:8 rate will result in longer packets.

A fuller description of LoRa will be found in this document;

http://www.semtech.com/images/datasheet/an1200.22.pdf

How the LoRa settings affect range

Having established a practical LoRa bandwidth, 62.5KHZ for a tracker, we can use this to look at how the spreading factor affects range and the time to transmit a typical 20 byte GPS location payload.

The lower spreading factors will give the highest data rates and the shortest packets in time terms, in turn this is more battery efficient. The lowest spreading factor we will consider is 7.

Now, assume we are using a bandwidth of 62.5KHZ and a spreading factor of 7, that will give an equivalent data rate of 2734bps, according Semtech's 'Lora Calculator' tool. The time taken to transmit our 20 bytes is 98mS. Assume that is our base and that in a typical urban area the limit of reception is 500m. To find a 'lost' model you would need to explore an area on a 1km grid, so that you are always within 500m of the 'lost' model.

If we switch to spreading factor 12 then the data rate drops to 146bps and the transmit time goes up to 2.2 seconds so this is a slow packet. The benefit of the higher spreading factor (and longer transmit time) is significant, we now have an approximate 14dB of extra signal gain. This is equivalent to 5 times further range, so our 500M now extends to 2.5km and the search grid is now strips 5km apart.

Now 5 times longer distance may not seem a lot, but appreciate how much difference it can make to the transmit power required to cover the extra distance. If we were transmitting at 10mW and that gave us the 500m range, to cover 5 times further range just by increasing transmit power would require a transmitter putting out 25 times the 10mW (250mW) and a very much larger battery.

Packet Addressing

LoRa is a two way technology, each device is a transceiver. Most often on a particular frequency there will be one transmitter and one receiver. However, this may not always be the case and there could be several trackers in use on the same frequency.

In order to keep the software simple and allow for the receipt of signals from multiple receivers or directed commands to particular tracker, a basic addressing scheme has been implemented. The key to this is that regardless of the data content of the actual payload each payload is preceded in the transmitted packet with 3 control bytes. In general the control bytes have been restricted to ASCII printable characters so that they can be shown directly on a terminal\monitor, the 3 bytes are;

Packet type. This either describes the content of the packet, which could be a GPS location payload or is a command to do something and there is no payload.

Packet Destination. The node number that the packet is destined for.

Packet Source. The node number that the packet was sent from.

The destination and source packet bytes mean that node ‘2’ (could be your base station receiver) can send a command that only station ‘3’ will respond to. This command could be a reset command, a request to turn on and off certain transmission types etc. Node ‘3’ can be set-up so that it only accepts commands from a particular node.

In my provided software the 3 control bytes are automatically added to each outgoing packet and stripped from each received packet.

An example of the 3 control bytes from a tracker would be;

G*2

Which means there is a GPS error (G) its been sent as a broadcast (*) and its from node 2.

This simple protocol can be used to queue requests to intermittent receivers. For instance the high altitude balloon version of my tracker software assumes that the in flight tracker does not listen for commands continuously; it only listens for incoming packets for short periods in order to minimise power consumption. At the beginning of a listen period (which may only be a few seconds) the tracker sends a ready (clear to send) packet. The receiver is normally not so power constrained so is left in permanent listen mode. The receiver has been given a queued request (could be to release a payload command for instance) and is waiting for the ready packet from the tracker. When the ready packet is received, the receiver sends the queued command and listens for an acknowledge packet. If the receiver does not receive an acknowledge it will keep queuing the request.

For the lost model locator application the packet addressing bytes are mainly used to allow the receiver to identify the type of data that is in the payload and take appropriate action.

The high altitude balloon tracking software uses the packet addressing to implement a command and control interface between the ground receiver and the remote balloon.

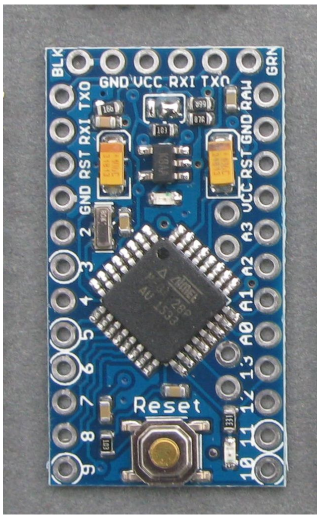
Design Considerations for a tracker

Considerable thought has gone into simplifying the design and build of this tracker and receiver. The first LoRa trackers I designed were based on surface mount technology (SMT) components and assembly. Using SMT is the easiest choice in most cases and whilst SMT is really designed for automated assembly, DIY assembly at home is possible. After several (successful) iterations of SMT based designs it became obvious that a much simpler design using only header pins in holes (PIH) and a few through hole wired components would be practical. It is hoped that the availability of such an easy build design will encourage the DIY enthusiast to build their own tracker.

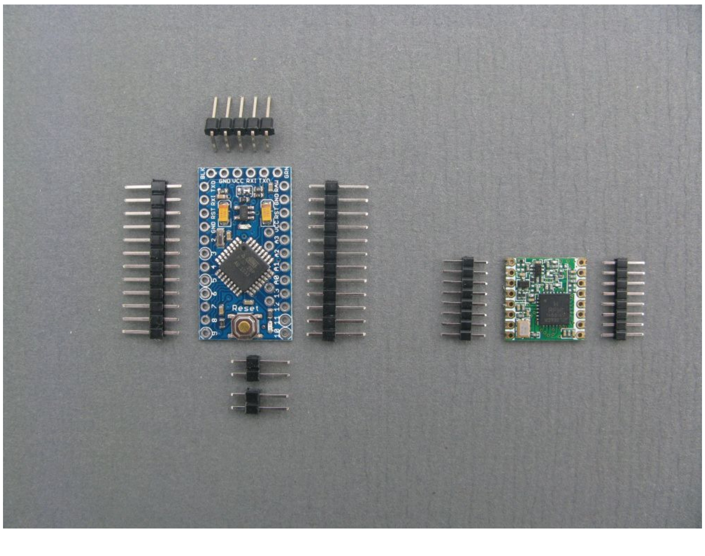
A lot of modern small electronic components are only available in SMT format so it is fortunate that the Arduino Pro Mini module is small, light, cheap and comes ready assembled with the SMT components such as the regulator, capacitors and other components already fitted, see picture;

Arduino Pro Mini – be sure to use the 3.3V

8Mhz version!

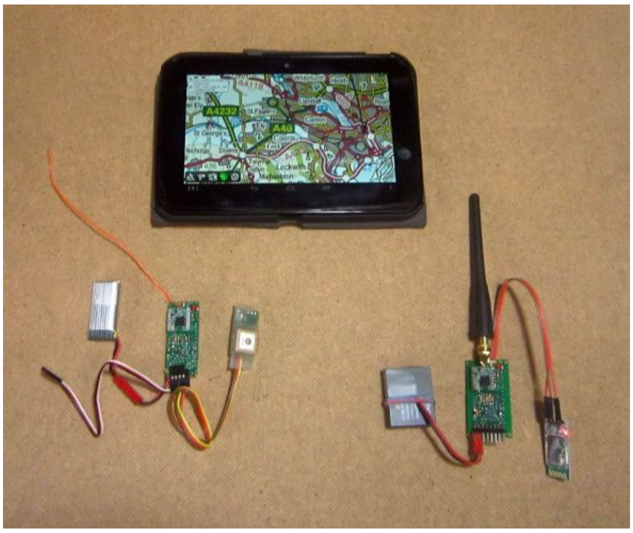


The connections for the Pro Mini are a series of 0.1” spaced holes along the edges and this allows the Pro Mini to be mounted with the header pins in the PCB holes. The same is true of the chosen LoRa device the RFM98 (on the right below) it also comes with holes for pin headers, 2.0mm spacing in this case, see picture;



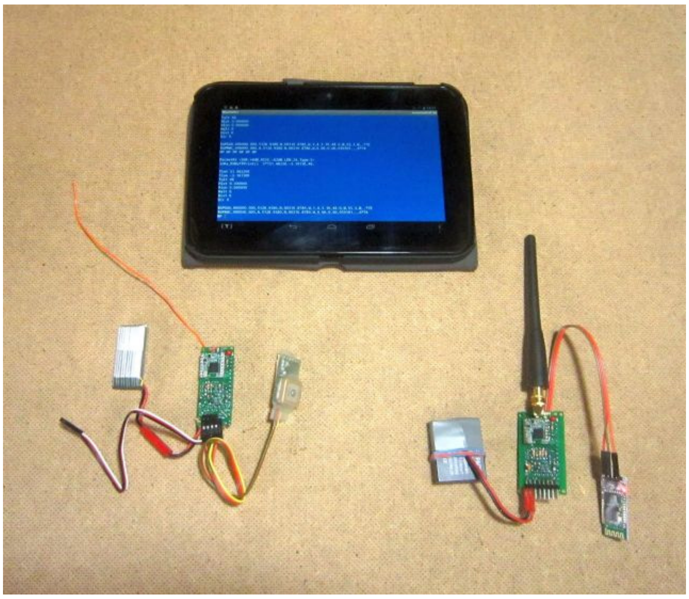
The same PCB design is used as a receiver and the location or GPS co-ordinates of the transmitter can be displayed in several ways.

First, on a map using an Android phone/tablet mapping application via a Bluetooth link.



Tracker, receiver with Bluetooth adapter and Android mapping application

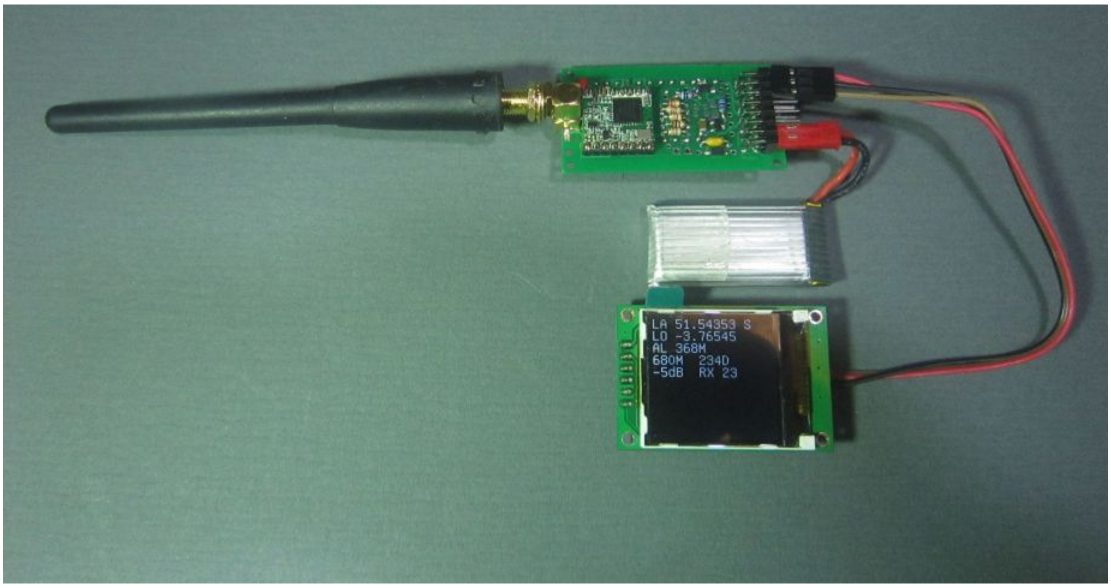
On an Android phone/tablet Bluetooth serial terminal as Latitude, Longitude, distance and direction.



Bluetooth connection to Android tablet as a serial terminal.

Or with an LCD display to show Latitude, Longitude, distance and direction. A standard Digole serial display module is pictured. The addressing protocol used by the receiver code is common to a wide variety of the Digole displays. Since the PCB expansion header has I2C connections on it displays that use this type of serial connection can be used as well, assuming suitable code is written

.



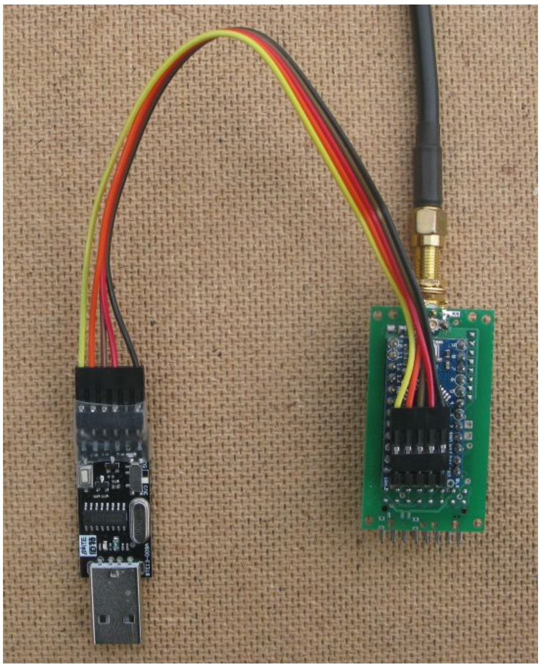
```
LA 51.54353 S
LO -3.76545
AL 368M
680M  234D
-5dB  RX 23
```

A simple receiver – tells you where the tracker is!

The use of a Nokia 5110 LCD is also described in a later article, this uses a custom serial backpack of my own design that turns the 5110 into a Digole style display. The 5110 is only 84×48 pixels so cannot display large amounts of text, however the displays have high contrast and work extremely well out of doors whilst most LCD\TFT\OLED displays are unreadable in bright sunlight.

Alternatively the receiver can be left connected to the programming header and used with a PC Serial terminal application such as Putty, TeraTerm or just use the serial monitor in the Arduino programming environment;

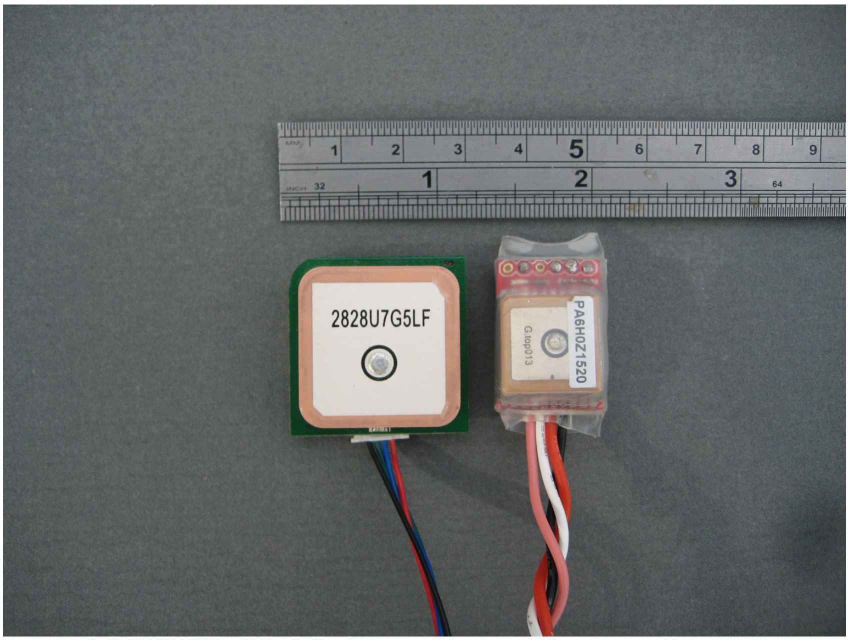
Easy connection to PC Terminal



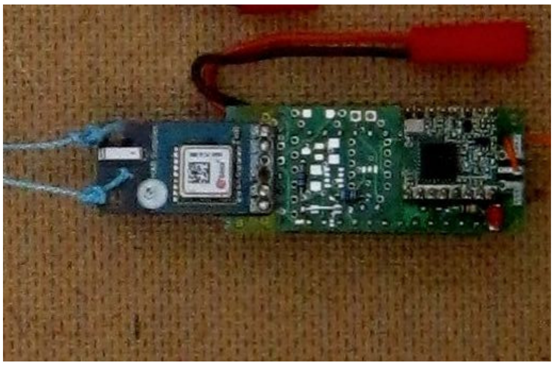
GPS Modules

Different applications may require different types of GPS. For a lost model locator you would normally use a lead connected GPS that has an integral ceramic patch antenna. This type of GPS module is available with both small and large ceramic patch antennas, the bigger ones are heavier but will normally have better reception which may be needed in difficult circumstances; these types of GPS are shown in the picture below.

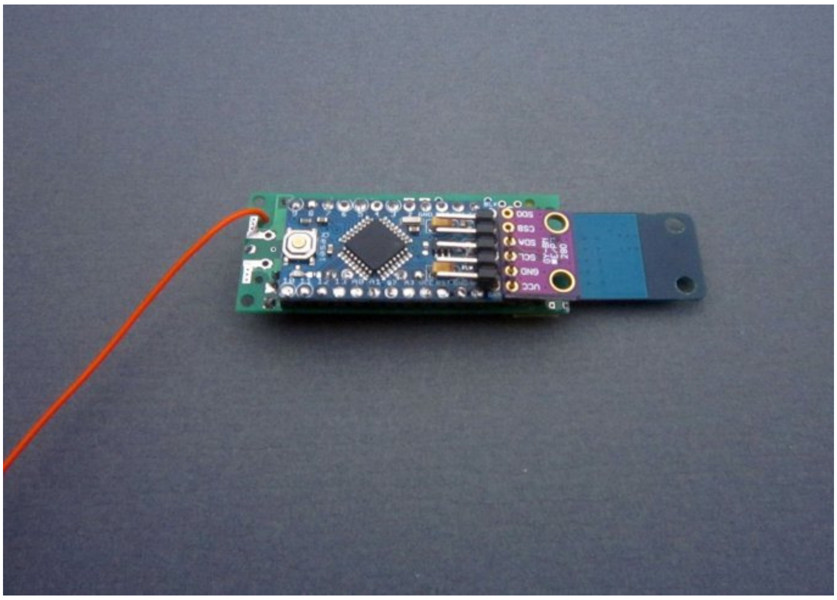
Large and small GPS with integral patch antenna



The PCB can also be fitted with a ready made lightweight GPS breakout board that allows it to be used as a PICO high altitude balloon (HAB) tracker. How this attaches to the tracker board is shown in the picture.



The PCB has an expansion header of 2 rows of 8 pins. This has dual power connections (battery and 5V) serial connections for a GPS and the I2C connections from the processor available. The tracker can be used for connecting I2C devices such as displays, sensors or GPSs, here is the tracker is fitted with a BME280 module a temperature, humidity and pressure sensor.



In the next article, building a tracker.

Stuart Robinson

GW7HPW

March 2016