GPS Antennas and their effect on LoRa Tracker power consumption

Stuart Robinson
March 2016

For some time I have wondered if you could improve the battery life of PICO High Altitude balloons tracker if you used the GPS's hot fix capability.

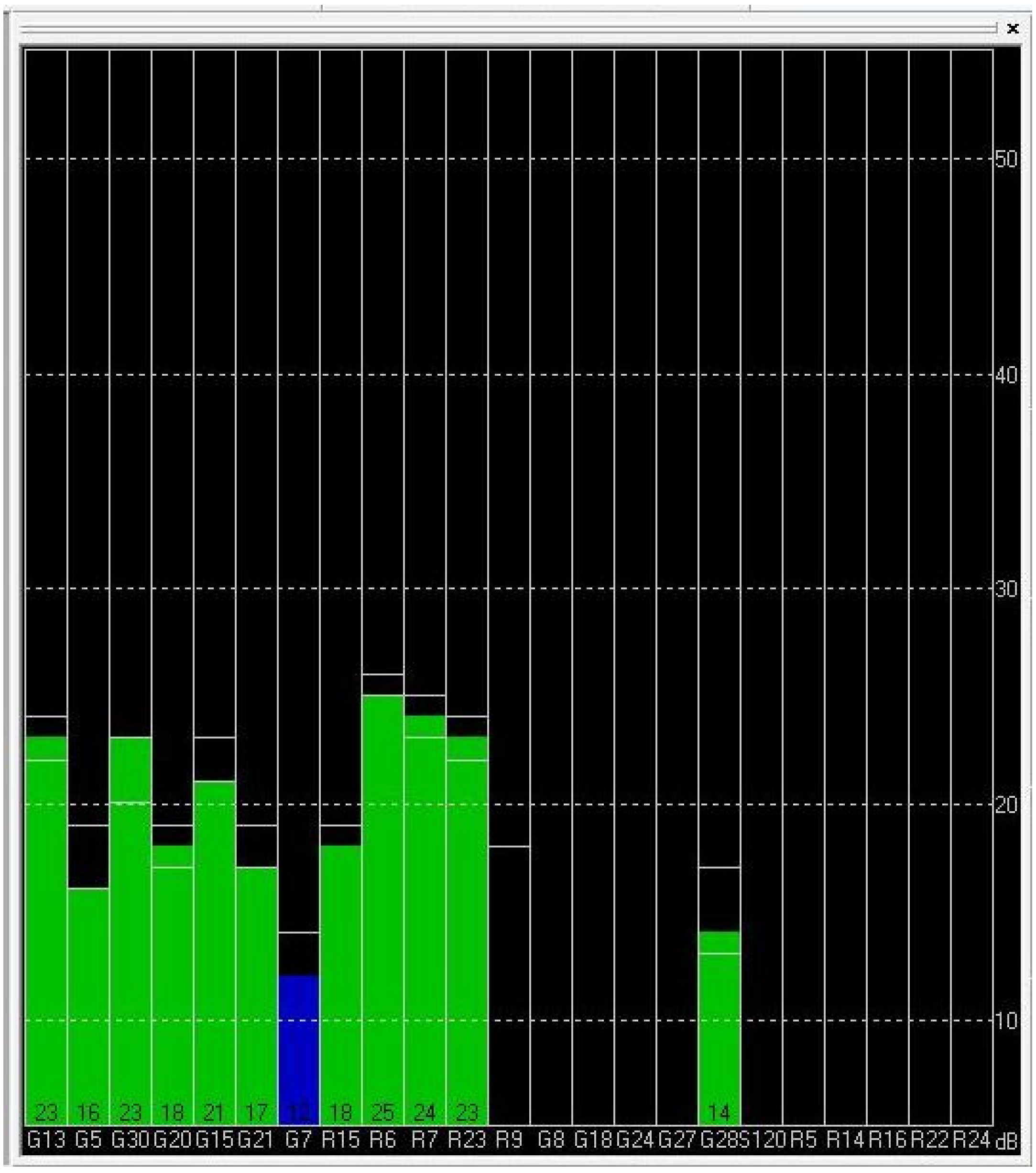
The common GPS for PICO HABing is the UBLOX MAX8 and if that is given a battery backup supply, the GPS will save the down loaded satellite data in battery backed up memory. It should then quickly acquire a new fix when its powered on. The potential for power saving is clear, if you only want a new GPS fix every minute or so, why have the GPS powered up and running all the time, even if its in power save mode?

The first thing to check is the effectiveness of the various antenna set-ups. For this I used the UBLOX U-Centre application which will display the signal strengths for each of the satellites the GPS is picking up signals for.

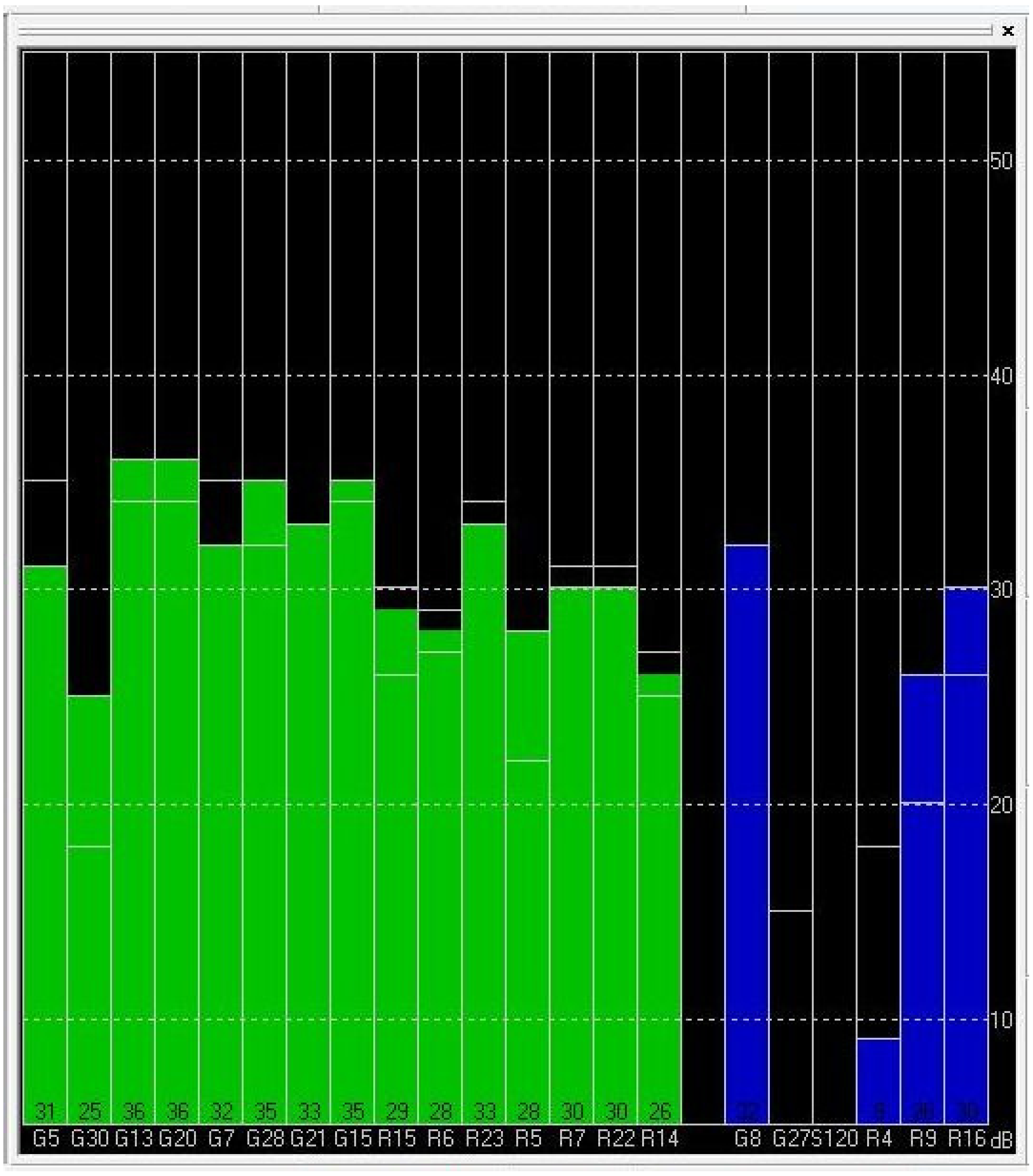
The GPS and tracker were position on a mast approximately roof top level in my garden, it's fairly flat where I live so the GPS antenna would have a reasonable view out to the horizon.

Below are screen shots of the reported signal strengths of the received satellites for the antennas tested. For comparison the signal reports for two GPS modules with integral ceramic patch antennas are included.

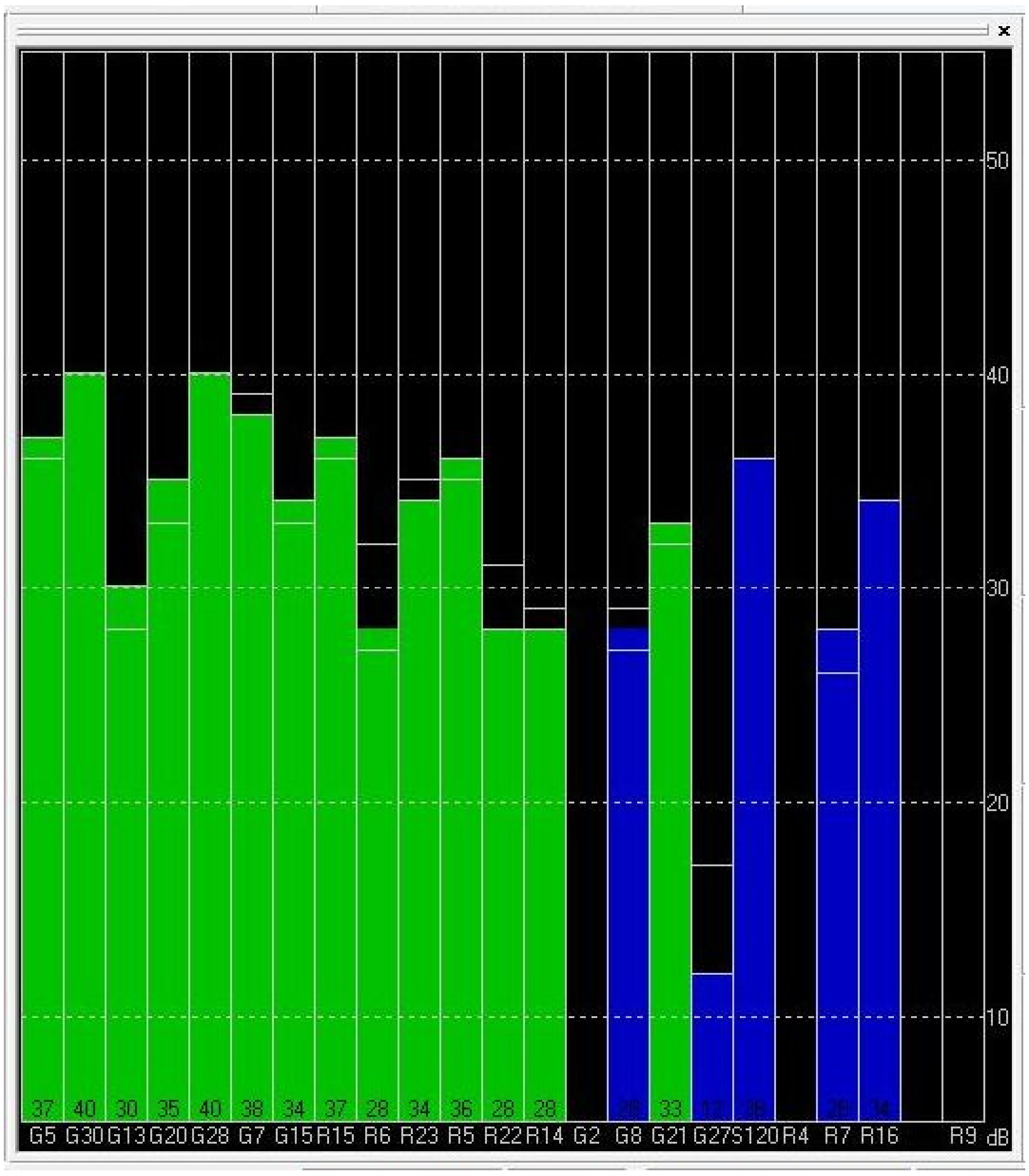
JTI 1575AT43A40 Ceramic Chip Antenna.



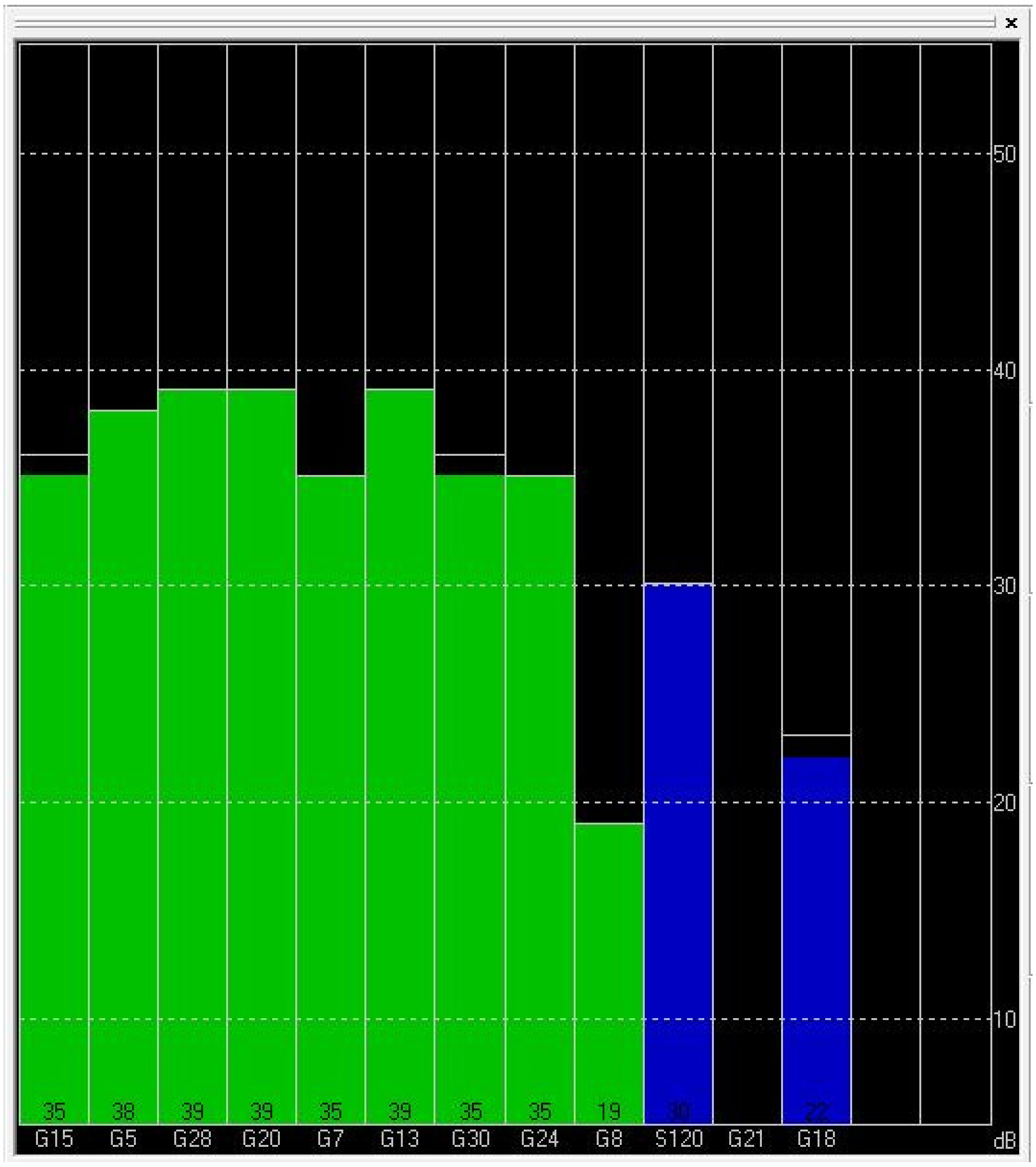
DIY 1/4 Wave vertical wire and 4 x 1/4 wave radials.
A GPS frequency of 1575Mhz was assumed.



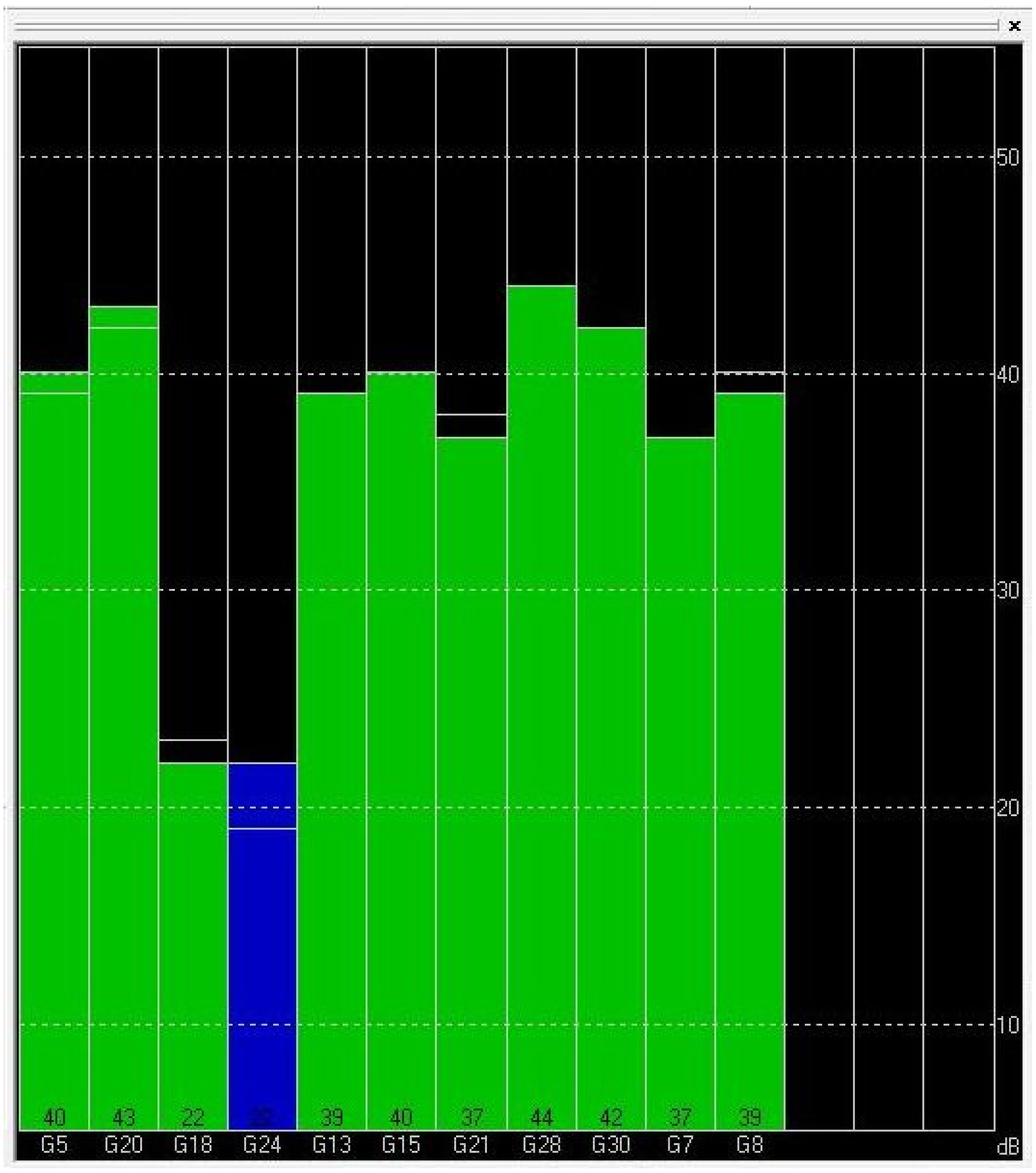
DIY 3/4 Wave vertical wire and 4 x 1/4 wave radials.
A GPS frequency of 1575Mhz was assumed.



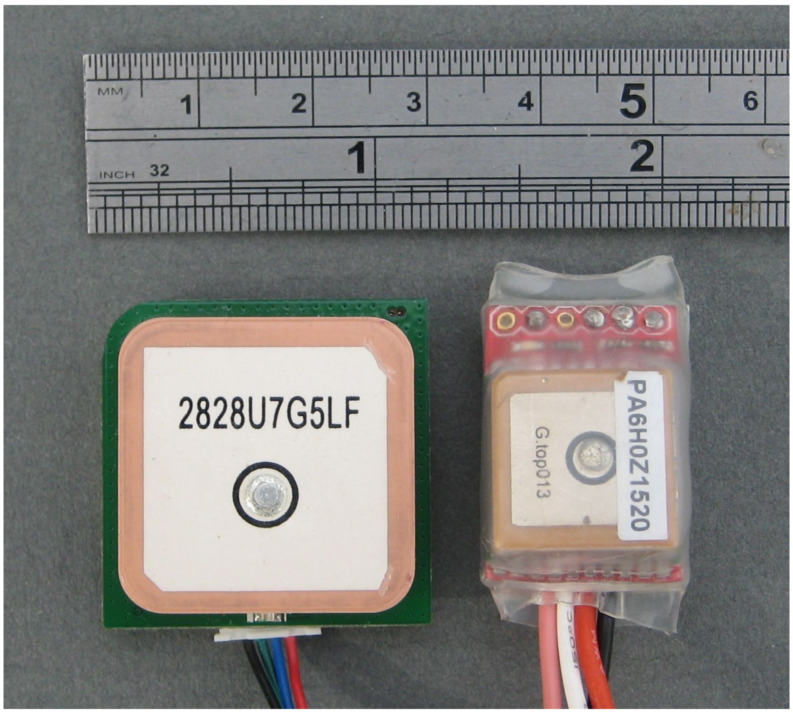
PA6H Small Ceramic Patch, 15mm x 15mm



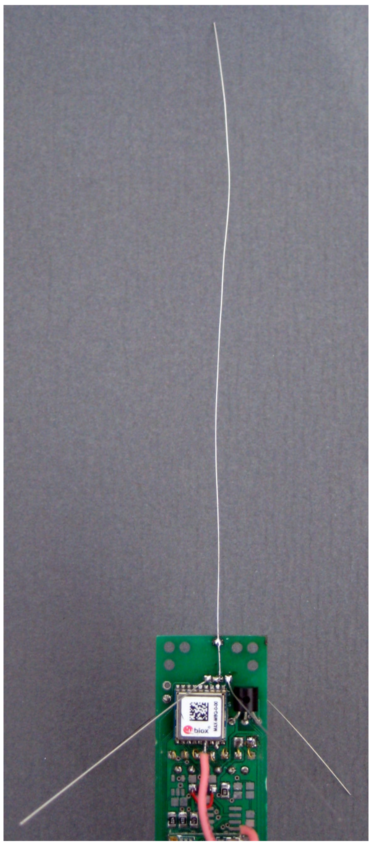
UBLOX 7 Large Ceramic Patch, 24mm x 24mm.



Ceramic Patch GPSs



Tracker with 3_4 Wave vertical antenna and 4 x 1_4 Wave Radials



Comparison of Hot Fix times

I set-up one of my GPS PICO trackers to power the GPS up, measure how long it takes to get a new fix, report the fix time and then turn off the power again. I set the software to power up the GPS for a fix approximately once a minute.

Its clear from the signal strength screen shots shown earlier that the DIY wire antenna is far more effective than the common ceramic chip antenna, so how does the difference affect the time take for a hot fix?

JTI 1575AT43A40 Ceramic Chip Antenna.

First I ran the UBLOX GPS with the small ceramic antenna most often used for a PICO HAB.

The Hot fix times over a 15 minute period were;

24964mS, 36959mS, 13959mS, 31961mS, 16945mS, 34978mS, 18982mS, 59996mS, 31975mS, 51994mS, 80005mS, 25969mS.

That is an average of 35.72 seconds per fix. Hardly worth the bother of using hot fix, as the GPS is in acquisition mode during hot fix so you may well use more power than a GPS left running in cyclic power saving mode.

DIY 1/4 Wave vertical wire and 4 x 1/4 wave radials.

Constructing the antenna requires that you have holes to fit the lightweight wire and ground plane radials, but it is easy to build these into a PCB design. The wire used is Ernie Ball Custom Gauge 13. Easy to solder, be careful to bend the sharp cut ends over to avoid stabbing injuries.

The Hot fix times over a 15minute period were;

7259mS, 7768mS, 7253mS, 7254mS, 7281mS, 7280mS, 7285mS, 7286mS, 7282mS, 6777mS, 7288mS, 7286mS, 7271mS, 7283mS.

An average of 7.27 seconds per fix.

DIY 3/4 Wave vertical wire and 4 x 1/4 wave radials.

The Hot fix times over a 15minute period were;

1954mS, 977mS, 1945mS, 1958mS, 968mS, 1953mS, 1962mS, 1958mS, 1951mS, 955mS, 1950mS, 1944mS, 1915mS, 1940mS.

An average of 1.74 seconds per fix.

I sent the results to Michael Kirkhart, a colleague on the $50SAT project, he is a design engineer working for NeverLost GPS systems, he said;

“Based on the C/N0 numbers you captured, I am not surprised by the long hot fix times you reported for the ceramic antenna. The 3 pieces of information the GPS receiver needs to quickly produce an initial position solution (assuming, it is able to quickly acquire and track at least 4 or more satellites) is the approximate position of the receiver, the ephemerides and clock correction for all tracked satellites, and time. Under hot start conditions, how quickly it can re-synchronize with GPS time is what will drive the time to first fix (TTFF, in GPS jargon). To do this, it has to be able to achieve sub-frame sync with the navigation message from at least one of the satellites. The C/N0 threshold for reliable decoding of the navigation message is about 26 dB-Hz, at least for the SiRF GPS receivers I have worked with. It appears the uBlox receivers have about the same threshold. Since the strongest signal from the ceramic antenna is about 25 dB-Hz, it is below the threshold, and the probability of bit decode errors increases dramatically. Under these conditions, the receiver has no choice but to try again and again until it can perform a reliable decode. This can take a long time, as you have shown”

Battery run time estimates.

During the initial fix acquisition the GPS consumes 26mA. Once the fix is acquired GPS current in cyclic mode varied between 19mA and 24mA. If the hot start was used then the current consumption was around the same figure during to 2 seconds or so that it took to acquire a new fix.

If we choose to power down the GPS, and wake it up to get a fix once a minute, it also makes sense to power down the micro controller too, this is straight forward in the Arduino environment and sleep current goes down to 25uA or so.

Using the improved ¾ wave wire antenna for the GPS there are two possibilities I will consider;

1. The normal routine of powering the processor and GPS all the time, checking the fix once a minute and sending the LoRa telemetry payload once a minute also.

2. Putting the micro controller and GPS to sleep for a minute, waking up, getting a new GPS fix, sending the LoRa telemetry payload, then going back to sleep.

Consider the first situation, with the GPS and processor powered all the time the current consumption is going to be fairly constant at around 19mA. Each minute will consume, allowing for the 33mA for 490mS of transmit current an average of;

19 + ((0.490/60)*33) = 19.27mA.

So a 400mAhr Lithium battery (approx 10g) would last;

400mahr/19.27ma = 20.75 hours, less than a single day.

For the second situation it's easiest to model in a spreadsheet. Using the power consumption and timing figures below;

GPS and processor hot fix current average 26mA.
Processor idle current 4mA
Transmit current 33mA.
Transmit time 0.49 seconds.
Sleep current 25uA.

It turns out our 400mAhr battery will now last 370 hours or 15.5 days. However note that we also have the flexibility to vary the battery life to suit, we could stretch it to 30 days by checking the GPSs and transmitting on a 2 minutes cycle.

Also of note is that if the tracker were powered by a pair of AAA Lithiums (14g) then for a 1 minute GPS\TX cycle the battery life would be 46 days and 90 days for a 2 minute cycle.

Using the hot start capability, I found little difference if the GPS was set up in cyclic power saving mode.

The tests and comparisons above were all carried out within a few hours of each other in order to reduce the effect of variations in GPS reception . Over the last year or so I have carried out a number of GPS tests in the same location, i.e. my back garden. I have noticed that there is a variation in lock times from day to day, sometimes the GPS goes to first fix fairly quickly, 45 seconds or so, but on occasion it can take several minutes. Why there should be such variation (at ground level) I am not sure.

I have also built and tested a LoRa tracker for terrestrial tracking purposes, based on a Arduino Pro Mini and a Ublox GPS with a ceramic patch antenna (see above). The requirement was to transmit the location of the tracker every 10 minutes or so. The Ublox GPS used has built in battery backup, achieves very short hot fix times and run from a pair of AA Lithiums the battery life would be 2-5years.

Software Backup Mode

In the tests above the power down of the GPS was achieved by using a MOSFET to switch the power supply going to the GPS. This is a universal method that can be used with most GPS that provide for battery backed up last position and almanac. This usually requires a backup supply to be applied to a specific pin in the GPS, the backup supply can either be a small lithium battery, supercap or simply a voltage source provided by the controlling microcontroller.

UBLOX GPS provide methods by which particular parts of the GPS receiver can be turned off via software in order to reduce power consumption. One of the easiest mechanisms to use with the a UBLOX GPS is the software backup mode. By sending an appropriate UBX command, the GPS powers off (mostly) within a couple of seconds of receiving the command. The GPS can be woken from sleep by any activity on its serial input pin, a single pulse is sufficient.

In software backup mode the power consumption of the GPS falls to circa 500uA, so its not really suitable if the GPS off times are significantly extended in order to get multi month or year endurance from small batteries. Powering the GPS off completely using a MOSFET as described above is a better method in such circumstances.

Having said that, the Software Backup method is suitable for high altitude balloon trackers and it removes the need for the additional components to switch off the (Ublox) GPS.

Measuring actual consumed milliamp hours

Most of my trackers can be powered via the 3.3V VCC supply provided by the USB serial programming adapter. There are devices designed to measure the current and power used by USB devices which can be used to measure the long term current consumption of an attached USB device. One such current monitor is shown below;

It keeps a running maHr used of the attached device. The programming adapter itself consumes around 21mA, so its reasonable to assume that any power used above that is used by the tracker. Thus we have a relatively simple and low cost way of testing different transmission and GPS use profiles of a tracker.

Stuart Robinson
GW7HPW