Self Tracking with LoRa at 434Mhz

Stuart Robinson

LoRa has great potential for use in tracking. Using even the low UK transmit power limit (for license exempt devices) of 10mW distances of several hundred kilometres will be achieved where there is good line of sight between receiver and the remote tracker.

Reception at these ranges has always been possible using FSK RTTY, but its not really practical to receive these signals with very low cost portable equipment. In addition the FSK RTTY needs a lot of manual intervention to keep it on tune.

LoRa receivers can be very simple, a LoRa transceiver module (RFM98 for instance) with an Arduino Pro Mini or UNO (3.3V version) and Nokia 5110 LCD plus a few wires. This simple set-up when attached to an antenna will provide a similar performance in terms of distance to FSK RTTY. Side by side ground based tests indicate that LoRa when run at the same data rate as FSK RTTY provides around 5dB extra link margin.

LoRa provides a capability for the launcher to plot the path of the tracker themselves using very simple and portable equipment. If the tracker is at 8,000M altitude then the LoRa will be received on the ground up to around 250km away. A 30,000M high tracker will be received at around 450km away. Signal reception reports suggest longer distances would be possible, particularly at low LoRa data rates, but curvature of the Earth becomes the limiting factor.

Calculating where a tracker is located is not difficult, its sending out its Latitude and Longitude after all. Having the path of the tracker displayed on a map is a significant benefit, especially if you want to find the tracker after it has landed. There are ways of posting the trackers location to an on-line map but use of that method depend on having a portable Internet connection, that is clearly not always possible.

For the flight of my HABAXE2 balloon tracker in January 2015 I converted the received LoRa position data back into GPS NMEA sentences and sent the appropriately formatted data over a serial lead across to a Net book PC running the Memory Map application. This set-up used a PICAXE 28X2 based receiver. Memory Map on the PC then displayed the track of the balloon on top of a digital Ordnance Survey map. This worked well, but Memory Map does need output from the 'GPS' at least every 5 seconds or it times out. Memory Map is a paid for application but the same technique worked on a free application called YAAC.

As an alternative to carrying a portable PC around, if you have access to an Android phone or tablet, then a very simple and low cost set-up can be used to plot the path of the tracker. The LoRa receiver can also, if fitted with its own GPS, automatically calculate the distance and direction to the tracker. The direction can be used to aim a modest yagi gain antenna mounted on a stand or pole. Such a set-up does not need access to the Internet for tracking.

The Android Bluetooth Connection

I set-up my low cost Arduino Pro Mini based portable receiver, see picture on the right, to use a Bluetooth connection to the Android device. A HC06 Bluetooth module was connected directly to the terminal port on the Pro Mini module used to run the portable receiver. This portable receiver has its own GPS and will display the distance and direction to the remote tracker. It will run for a few days on a modest battery. You can plug it into a PC for upload into Habitat for HAB tracking. For tracking with LoRa this is the only receiver you need, it works just as well as a bench receiver.

The receiver program was set-up to send console output at 9600baud, the normal default speed for a HC06 Bluetooth adapter, so no set-up of the HC06 was required.

To use the Android device as a remotely connected serial terminal for the LoRa receiver install a Bluetooth application on the Android device called BlueTerm+, its free.

Turn on the Arduino LoRa receiver, pair the Bluetooth device (default name HC-06) in the settings options of the Android device. Then run Blueterm+. Goto settings and select 'Connect Bluetooth device'. Select the Bluetooth device you just paired.

Whatever is sent to the terminal port on the Arduino Pro Mini should now appear on the screen display of the Android device. This on its own is useful and the Arduino program in the receiver can be used to display useful information about the tracker and the received signals.

NMEA output format

The GPS on the tracker will be transmitting NMEA sentences such as this;

*$GPGGA,084357.000,5227.5551,N,00257.6081,W,1,4,3.25,104.6,M,49.5,M,,*41*
*$GPRMC,084357.000,A,5227.5551,N,00257.6081,W,1.10,104.12,010116,,,A*79*

So 'all' we need to do in the receiver program is convert the Latitude and Longitude information in the received LoRa signals, normally decimal degrees, into the format used by NMEA sentences shown above and send them to the terminal port on the Arduino Pro Mini.

The output format of the NMEA sentences from the Arduino LoRa receiver that the mapping application accepts can be simplified as follows;

*$GPGGA,000000.000,5227.5220,N,00257.5340,W,1,4,3.16,101.0,M,53.3,M,,*4E*

*$GPRMC,000000.000,A,5227.5220,N,00257.5340,W,0.00,0.00,010101,,,A*78*

These are not completely consistent as standard NMEA sentences, the time is wrong and some fields have null values, but the mapping application does not seem to mind. Rather conveniently the Android set-up described here ignores all the LoRa receiver debugging output to the serial console such as this;

NP.NP.NP.NP.NP.NP.NP.NP.NP.NP.NP.NP.
Packet RX +11dB,-58dB,36
*S*1052.45877,-2.95894,101,15:10,1,0,*

Tlat 52.458698
Tlon -2.958900
Talt 101

*$GPGGA,000000.000,5227.5220,N,00257.5340,W,1,4,3.16,101.0,M,53.3,M,,*4E*
*$GPRMC,000000.000,A,5227.5220,N,00257.5340,W,0.00,0.00,010101,,,A*78*
NP.NP.NP.NP.NP.NP.NP.NP.NP.NP.NP.

This means that appart from the LoRa receiver sending out the NMEA sentences, no other receiver program modifications are needed.

External GPS Provider for Android

We now need an application on the Android device that will use the received Bluetooth output to simulate an external GPS. Install an application called “Bluetooth GPS Mouse” on the Android device. This can be tried for free, it costs £2 to buy.

Start “Bluetooth GPS Mouse” and select the 'Connect to Bluetooth Device' option at the top of the screen. Select the Bluetooth device on the Arduino LoRa receiver, its called HC-06 in my case. Make sure the service is started.

The display at the bottom half of the screen may display 'No GPS Solution is Computed' until it has received a couple of the GPGGA and GPRMC sentences, but should then show 'GPS Solution Available' and the display the details of the location of the remote tracker.

Once Bluetooth GPS Mouse is set-up, you can use an auto start application on the Android device to run it at start-up, it will then automatically connect to the Bluetooth adapter whenever its in range.

Android Mapping Application

The Android mapping application I have been using is 'AlpinQuest' which you can use for free, but the full version is only around £6.50 to buy. If you don’t have a suitable map already then you need to import a Map, see later for instructions on how to do this.

AlpinQuest is designed to use the GPS on the Android device, but has a function that allows an external GPS provider to be connected.

Install and start AlpinQuest.

Select the Position Icon at the bottom left of the screen. The Icon looks a bit like a snowman with a tie on.

Select 'Positioning details' and then the tools button at top right (it's a spanner and screwdriver).

For 'GPS Position' drop down the option list and select 'External Locations' and select 'OK'.

Within the 'External Locations' box on the main Alpinquest screen at top left you should see the number of fixes increasing, the altitude in the screen centre and the speed on the far right.

The position will by default display as something like 30U UC 48574 35774.

To change the location format to decimal degrees, select the main AlpinQuest menu (picture of two peaks) then;

Select 'Settings'

Select 'Location Format'

Select 'Lt/Long (Deg)

AlpinQuest should now be displaying the location of the remote tracker on the map display, as a blue dot, assuming you have the correct map loaded.

Maps for AlpinQuest

If an Internet connection is available you can select the map source and Alpinquest will load the maps it needs as the tracker changes location. These map tiles are stored so can be used when in offline mode. For some tracking you can browse to the area of choice (when in online mode) in advance and store the maps for the appropriate area. For tracking a balloon however you will not know the track in advance and you cannot of course always depend on an Internet connection.

AlpinQuest will use the old style QCT format maps of Memory Map, very handy if you have them. AlpinQuest does not appear to support the new QC3 format of Memory Map files, but you could try it and see with the free version of AlpinQuest. For some tracking a Road atlas may be sufficient, you can download a UK one in QCT format here;

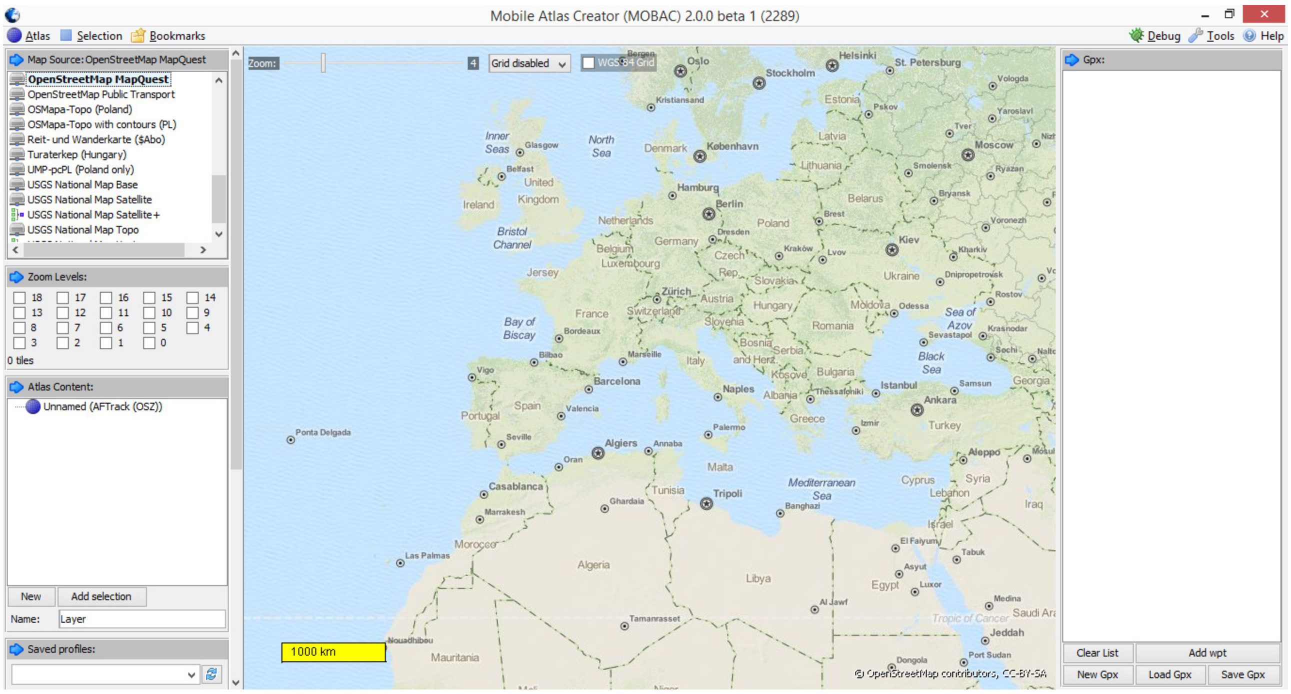
http://memory-map.co.uk/support/647849-Base-maps-and-Elevation-data-for-UKEurope

There is a process described below for using a Windows application called 'Mobile Atlas Creator' (free) to import one of the open source maps into a format used by AlpinQuest, these maps will then be available for offline use.

Importing Maps for AlpinQuest

These are instructions for a Windows PC. Download and unzip the latest copy of Mobile Atlas Creator (MOBAC), I used version 'Mobile Atlas Creator 2.0.0 beta 1'.

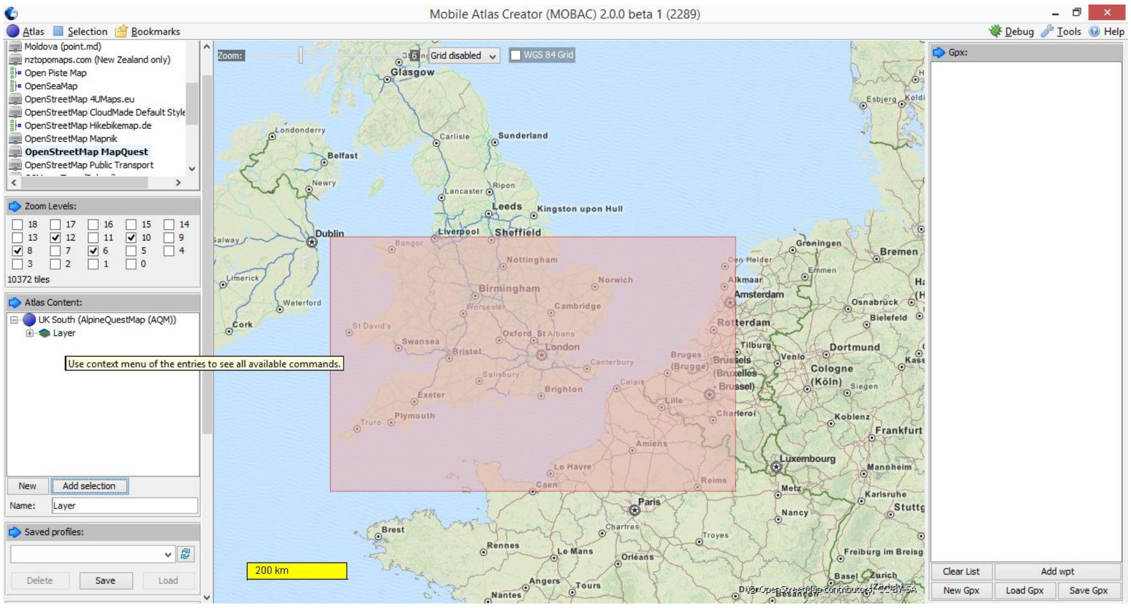
No need to install it just run 'Mobile Atlas Creator.Exe' from the unzipped folder. This is a typical opening screen.



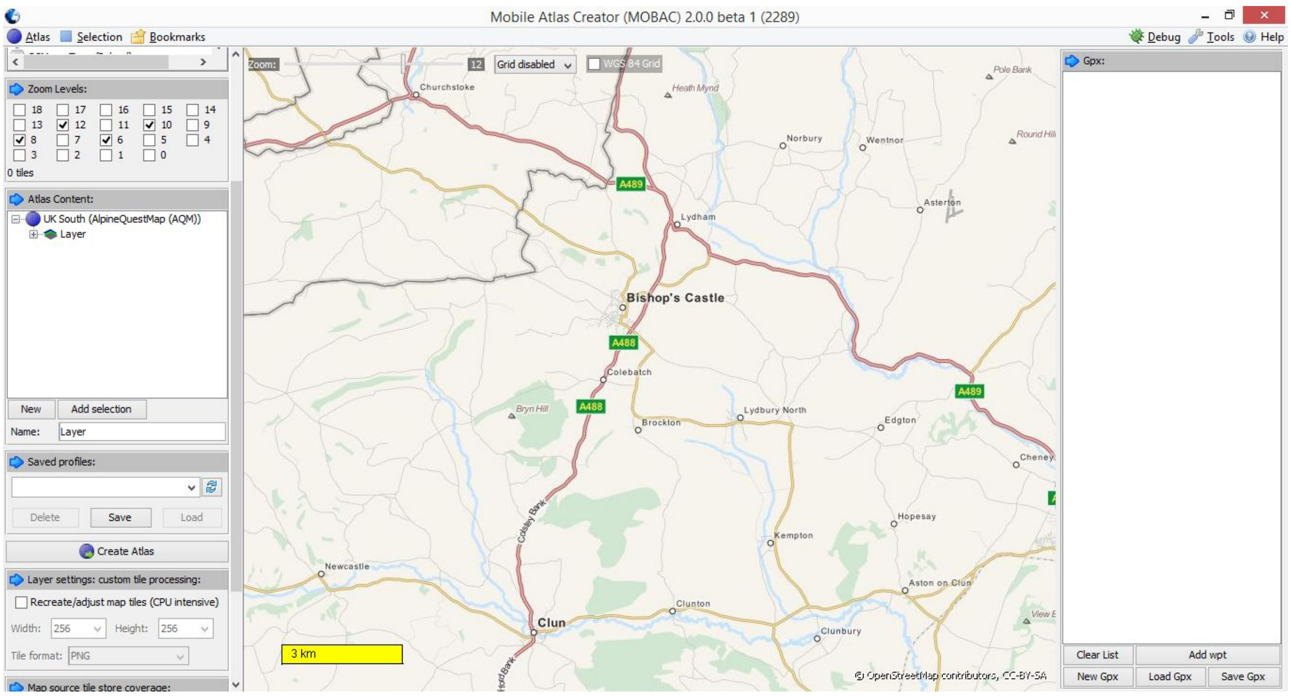
Using the Map Source panel on the top left select 'OpenStreetMap Mapquest'

On the map display select the area you want, I will download a map for the South of the

UK, the area is highlighted in the picture below;



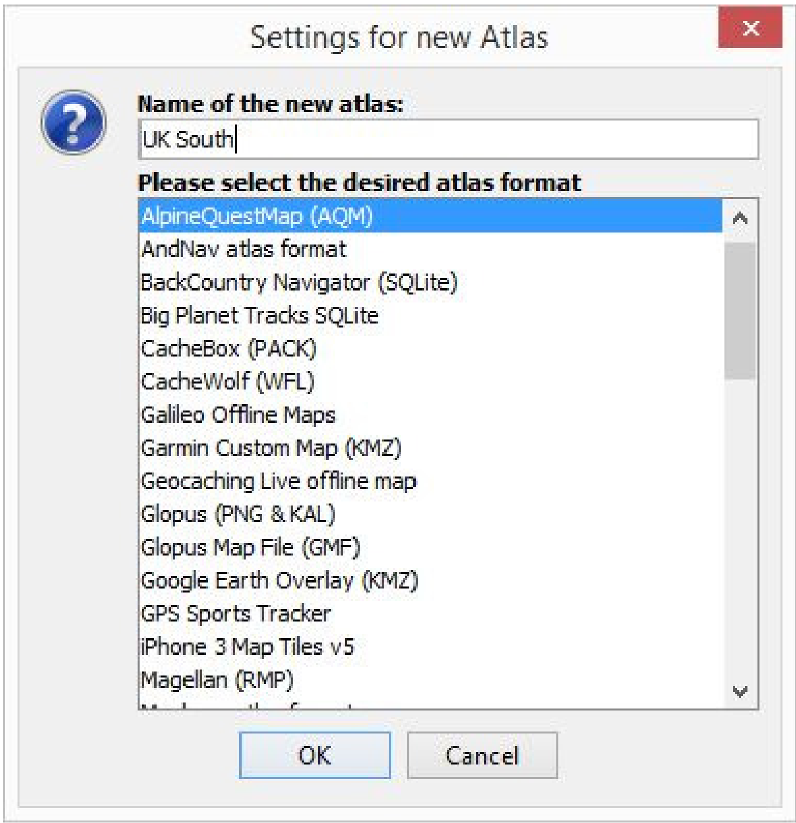
The map is displayed in the picture at zoom level 6, and in the zoom levels selection panel on the left I have selected zoom levels 12,10,8 and 6. Zoom level 12 is the highest resolution and this is the level of detail we eventually get;



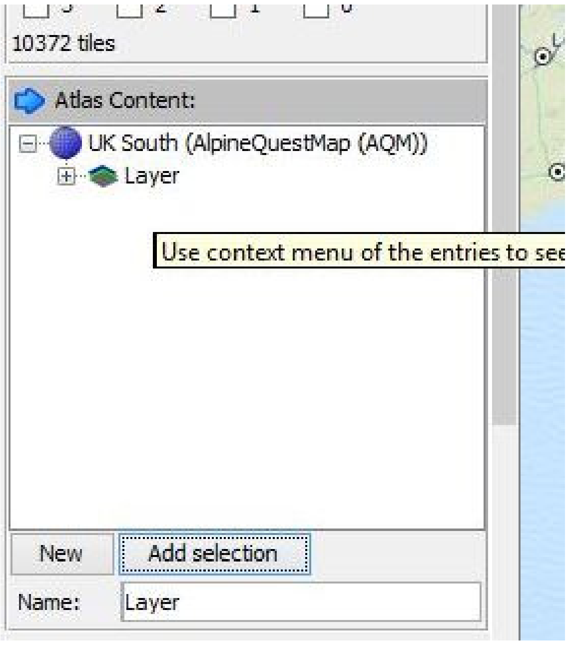
To start the process, in the 'Map Source' box, top left of the scrollable boxes on the left select 'OpenStreetMap MapQuest' as the source.

In the 'Zoom Levels:' box set the zoom levels you want. This box will display the number of tiles to be downloaded, more than 10,000 can take a while.

In the 'Atlas Content' box select 'New', you will see the 'Settings for new Atlas' window. Name your new Atlas (map) and select 'AlpinQuestMap (AQM)' for the desired format. Click OK.

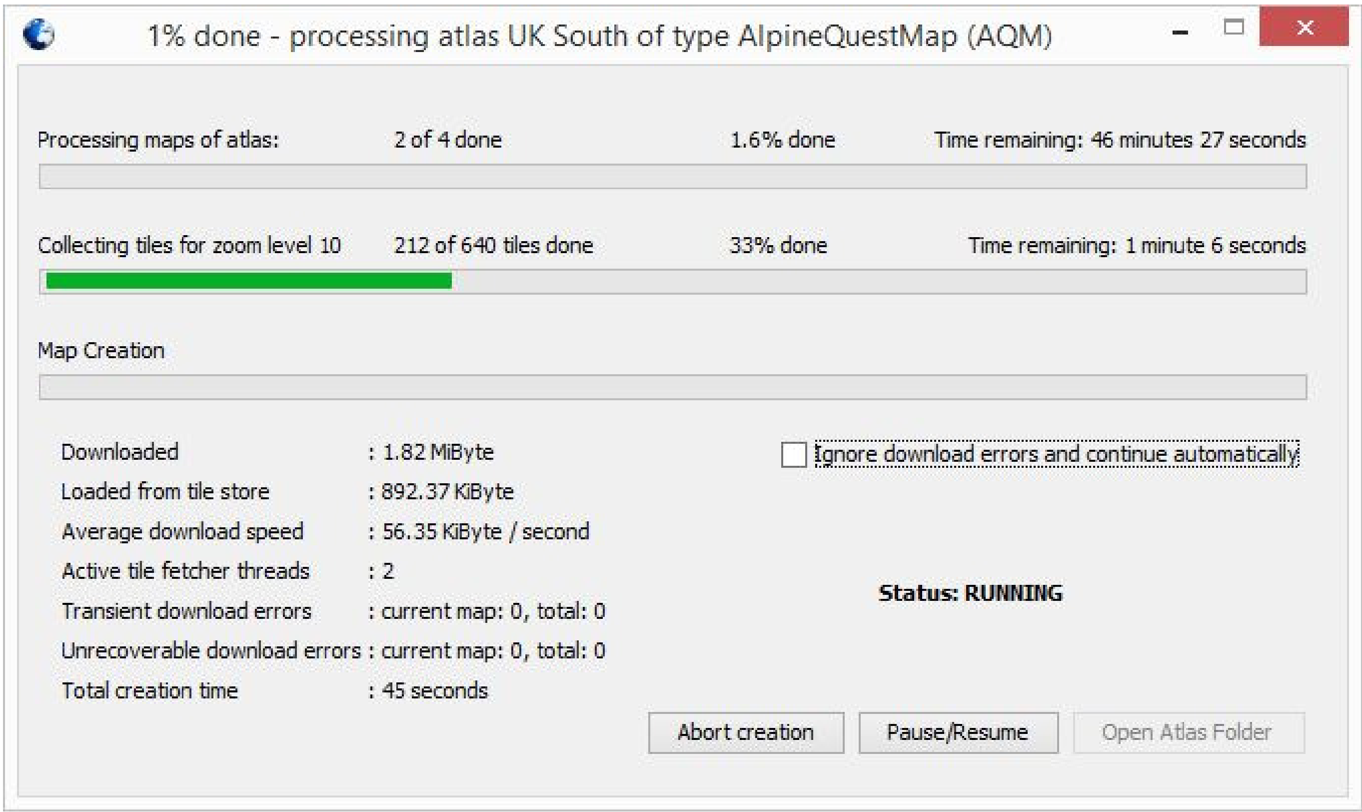


Zoom out the map display on the right and use the mouse and left button to select the area for you map. See one of the previous pictures.



In the 'Atlas Content' box, select 'Add Selection' and use the Mouse and left button to select the area you want. The Atlas Content box should now show a layer.

Go down to the Create Atlas button on the left and select it, the map generation should run;



In the case of the map generation above, the map files will be created in;

C:\Mobile Atlas Creator 2.0.0 beta 1\atlases\UK South_2016-01-01_141349

Copy the file with the AQM extension onto a Micro SD card, normally to a folder at the root of the SD card called \Alpinquest\Maps, and you can then load the map in AlpinQuest.

Stuart Robinson
GW7HPW
January 2016