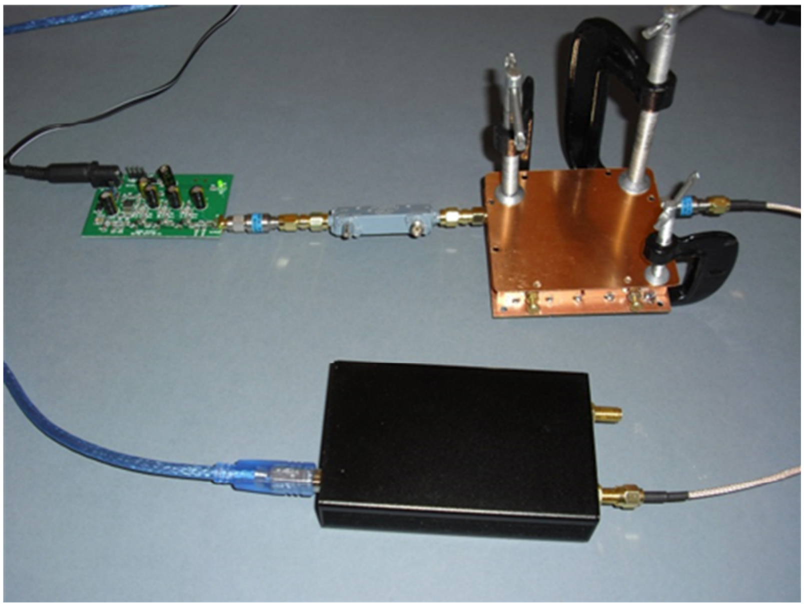
The RAGazine

Volume 2 Issue 3 February 2015



The NWT4 Spectrum Analyser and Tracking Source

BAA, Burlington House,
Picadilly,
London
(+44)207 734 4145
Reg Charity 210769

The RAGazine
British Astronomical Association
Radio Astronomy Group

BAA RAG Coordinator	**Paul Hyde**	g4csd@yahoo.co.uk
Ragazine Editor	**Jeff Lashley**	jeffl@spacecentre.co.uk
VLF reports	**John Cook**	jacook@jacook.plus.com

Web : http://www.britastro.org/radio

Notes for content submission

Content should be emailed to the Ragazine editor a minimum of fourteen days before the next publication date. Content submitted after that may not appear the next issue, but will be held for a later issue.

Observational reports are very welcome on topics where radio techniques are applied to observe astronomical objects, or geophysical events. Articles are welcome on topics of radio astronomy observational techniques, radio hardware and related technology, scientific programming, events, data processing, educational out-reach, book reviews, radio astronomy history etc.

The preferred format for submissions is Microsoft Word (.doc or .docx format). However I know that not everyone has access to Microsoft Office, note that the free office suite LibreOffice is available for Windows, Linux and Mac OS. Note that LibreOffice can save documents in Microsoft Word .docx format. If neither of these applications are available then plain text (.txt) is fine.

Images can be supplied embedded in the document or as separate files. Please include the author credits you require to be included in the article, and indicate whether you want your email address to be shown in the publication.

The BAA is not responsible for the opinions expressed by RAGazine contributors, and such published material does not necessarily express the views of BAA Council or RAG officers. Material without attribution is generally contributed by the editor.

2015 Publication dates and submission deadlines

Release Date	Submission deadline
11th May 2015	27th April 2015
10th August 2015	27th July 2015
9th November 2015	26th October 2015

RAG Coordinator report

By Paul Hyde

Welcome to the first edition of RAGazine for 2015 and, once again, my thanks to Jeff Lashley for pulling the material together. Please think whether you can offer any material for RAGazine, whether it's from personal work or progress on a club/society project. Your experience helps to inform and inspire others.

Solar eclipse VLF study

There will be a solar eclipse on March 20, though you will need to be a long way north (Faroe Islands, Svalbard/Artic Circle) to experience totality. This will be the last 'European' total eclipse for a good 10 years and a number of VLF observers will be paying particular attention to received signal strengths during this period and comparing results afterwards, with the aim of submitting reports to both the RSGB and the BAA. Of particular interest will be the NRK signal (37.5 kHz) from Grindavik, Iceland, though this will be too high a frequency for those using Spectrum Lab and standard sound cards which sample at 48 kS/s. However we are also interested in any results from other VLF (and LF) stations. Measurements should be submitted in a .csv format and with accurate time stamps, e.g. to a second or so. It is not unusual for the clocks of 'free-running' PCs/laptops to be out by several tens of seconds. The exact arrangements are still to be agreed but will probably involve measurements for the same period the day before and the day after the eclipse. Please contact me directly for more information if you would like to participate in this study.

Events and talks

Several BAA events are coming up in the next few months which will have a RAG presence: the Back to Basics meeting in Lincoln (March 14), the Winchester Weekend (April 10/11/12) and the Exhibition Meeting in Cardiff (June 27). In addition there are talks at Harrow (May 15), Wycombe (May 20), Heart of England (May 28) and York (June 19) Astronomical Societies. Visitors are welcome to all of these meetings though you will need to book for the Lincoln event and the Winchester meeting is already sold out. Local astronomical societies usually charge a small entrance fee for visitors. Further information can be found on the BAA website or that of the local society hosting the event. It is always good to meet those with an interest in radio astronomy, whether a beginner or seasoned observer, and we look forward to seeing you there.

RAG 2015

It's also time to think about the next RAG General Meeting. These events have been very popular to date - probably the best attended of any of the BAA Sections. The two reasons for this popularity are the number of us who are genuinely interested in radio astronomy, and that we have been able to provide a good selection of talks on the day. I have the offer of four talks so far, so I am half way there, but I would like to have as many as possible to choose from. Please let me know if you can help with this event, either as a speaker or if you have equipment, observations or poster material for display during the intervals.

Best wishes, Paul Hyde, g4csd@yahoo.co.uk

Solar activity last quarter 2014

Fig 1 shows activity levels since 2005. Sunspot numbers are courtesy of the BAA Solar Section. SID numbers since the last report are as follows:-

October = 72 (including 19 M and 4 X-class)

November = 46 (including 8 M and 1 X-class)

December = 44 (Provisional, including 5 M-class)

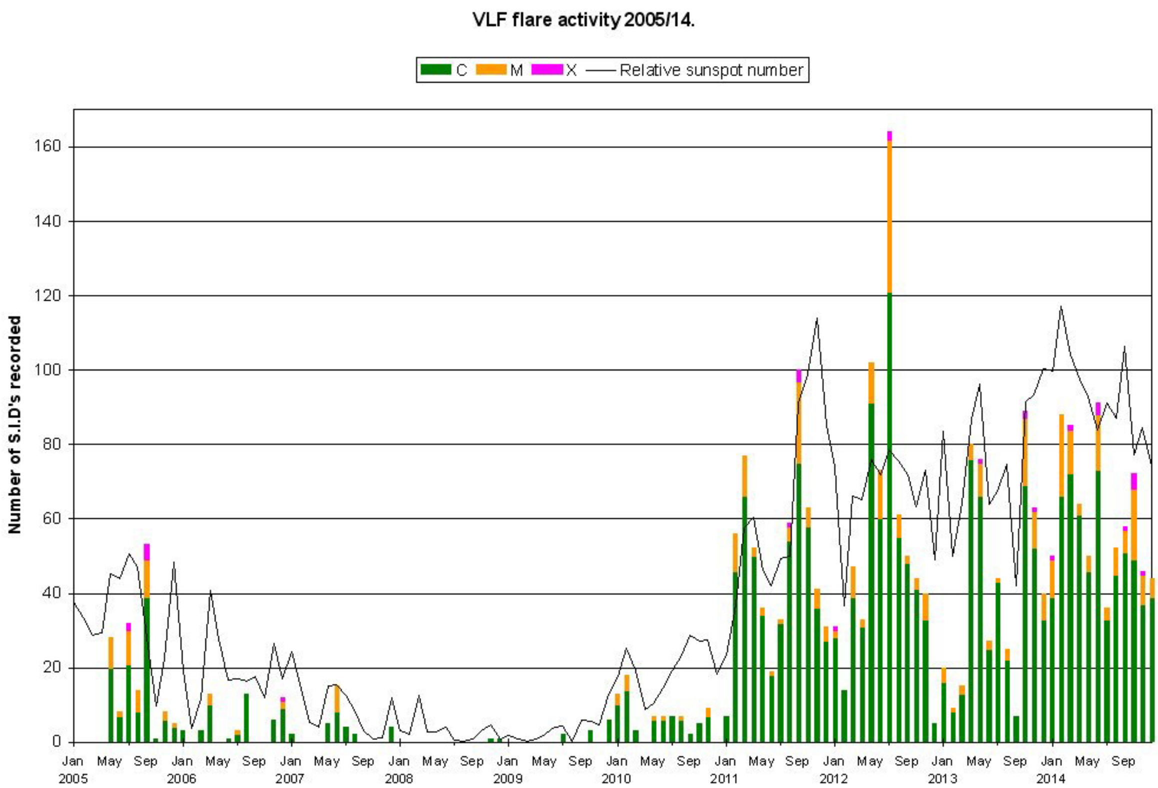


Fig 1: Activity chart.

Just as the short nights in summer make life difficult for visual astronomers, so the short day length in winter is a problem for VLF observers. The sun is very low in the sky in the UK at this time of year, and so it's UV and X-ray output has to penetrate a greater distance through the ionosphere to reach the D-region. The result is generally lower ionisation levels, and greater instability in signal strength of the closer VLF signals. The shorter day length means that we also have a shorter observing period, and will therefore record much less solar activity. The instability also makes it difficult to identify genuine SIDs among the random variations. The recorded SID numbers have fallen substantially since the autumn equinox partly due to this effect, but also due to a real fall in solar flare activity. There have been some large and complex sunspot groups over the last three months, but they have not all been particularly active despite their magnetic complexity.

The first of these complex groups, AR12192, made its appearance on October 17th, and over the next two weeks it produced six X-class flares. Four of these were observed as SIDs. Fig 2 shows activity on the 26th recorded by Paul Hyde. This was the busiest day of the month, with flares occurring in rapid succession. The X2.0 flare has produced a very significant disturbance at both 22.1kHz (red) and 23.4kHz (blue). Note that the SIDs appear inverted between the two signals due to the different path lengths involved. The background X-ray flux was above the C1 level for much of the time that AR12192 was visible, masking many much smaller flares.

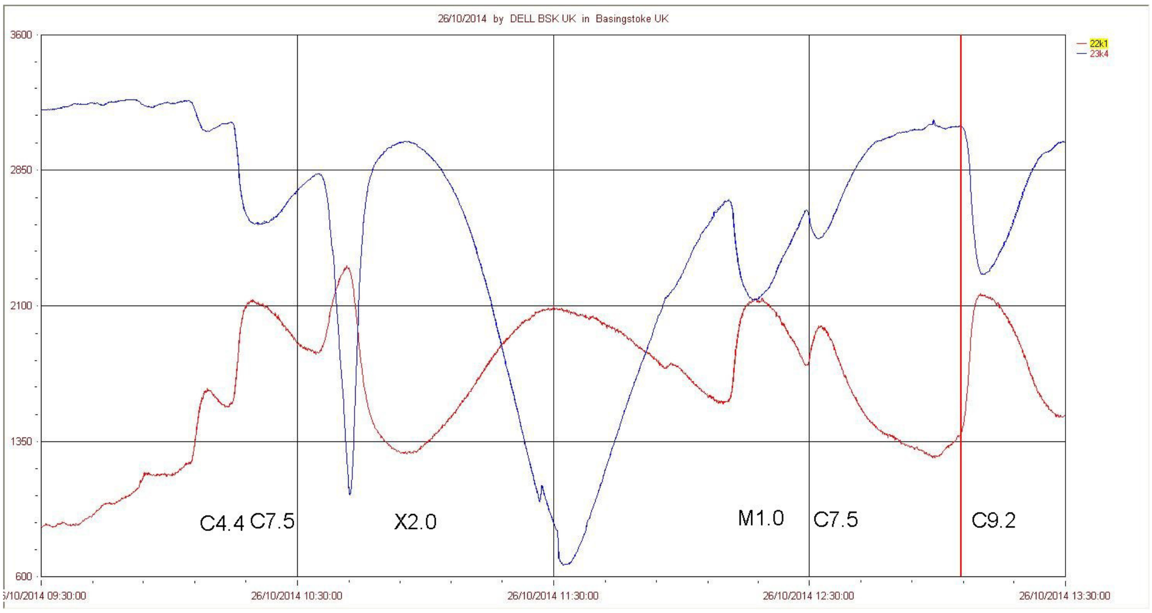


Fig 2: October 26th. Paul Hyde, Basingstoke.

Despite the high level of flare activity, very few CMEs were produced and there was little magnetic disturbance created. Most of that shown in the Bartels diagram was from coronal hole effects. A very small SFE (about 8nT) was recorded from an M4.3 flare on the 16th.

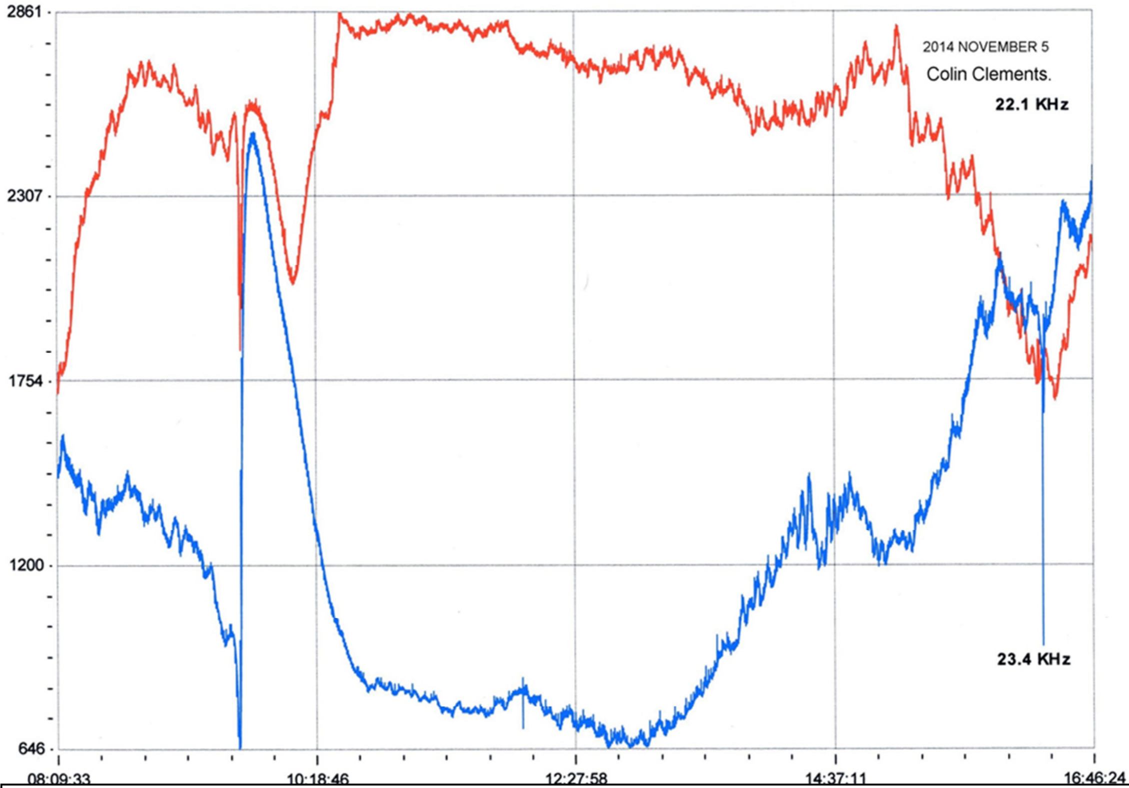


Fig 3: November 5th. Colin Clements, Lisburn.

Activity was considerably lower in November, with just a single X-class flare recorded. There was a nice firework to mark November 5th from an M7.9 flare. The SID is shown in Fig 3, recorded by Colin Clements. Peaking at 09:47UT, it has also produced a very significant disturbance at 23.4kHz (blue). The 7th was the busiest day in November, with a total of 9 SIDs recorded. Peter Meadows recorded some of this activity at 23.4kHz as shown in Fig 4. The X1.6 flare peaking at 17:18UT was too late in the afternoon to show on the 23.4kHz path eastwards to Germany, but was recorded by Mark Edwards at 24.0kHz on the westward trans-Atlantic path.

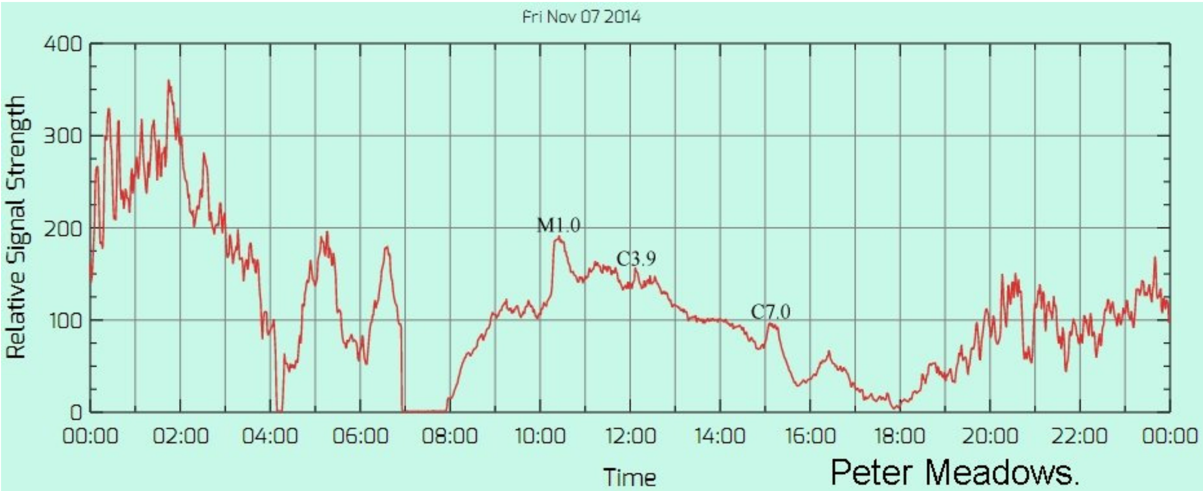


Fig 4: November 7th. Peter Meadows.

Although old region AR12192 did return in November, re-numbered as AR12209, it produced mostly much weaker flares this time. With the low Sun angle already mentioned they did not create many distinctive SIDs. The winter-time instabilities often show as a very distinctive oscillation in the signal strength on the shorter paths. This was particularly clear on November 24th, as shown in Fig 5, recorded by Mark Edwards at 19.6kHz. There is so much instability that a C4.1 flare peaking at 11:07 is completely masked. Colin Clements recorded similar behaviour, both recordings showing a strong 4 minute period to these oscillations. They appear to be due to waves travelling along the lower boundary of the D-region in the same way as waves travel over the surface of water.

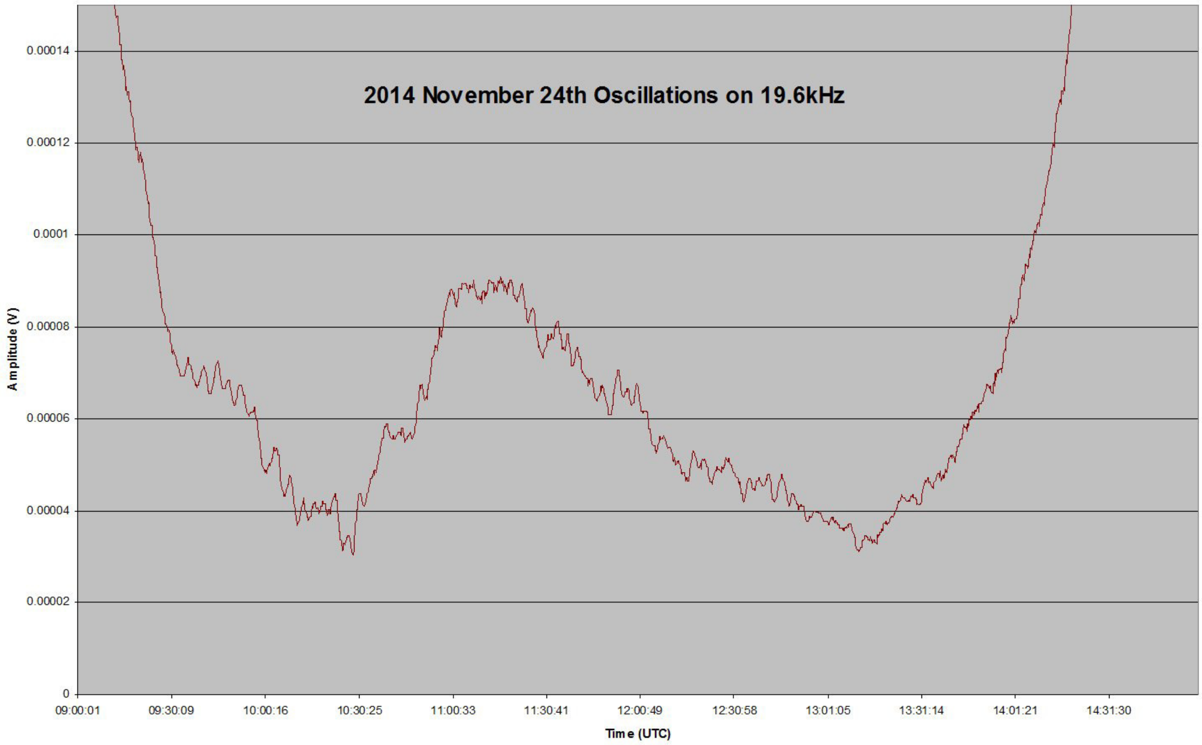


Fig 5: November 24th. Mark Edwards, Coventry.

There was just one major CME in November. This was associated with the X1.6 flare on the 7th. Fig 6 shows the magnetogram recorded by Roger Blackwell, showing a sudden storm commencement at 02:20UT on the 10th. The SSC is most visible in the Bx (blue) trace. Activity increases in the afternoon, lasting until midnight. Ignore the large spike just before 23:00, which is due to local interference. Using our SID timing of 17:18 on the 7th gives a CME transit time of 57 hours 12 minutes.

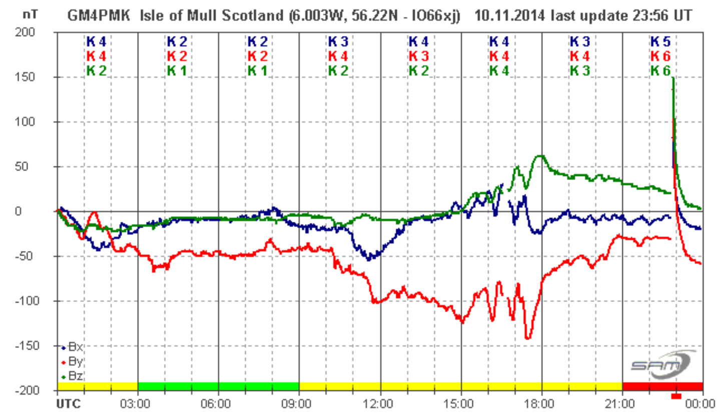


Fig 6: November 10th. Roger Blackwell, Mull.

December was also badly affected by oscillations, although there were also fewer flares to record. Roberto Battaiola in Milan suffered rather less of this instability, as shown in Fig 7 recorded on December 16th. Old region AR12192/209 made another brief return in December. It had significantly reduced in size and rapidly faded away to plage with little further flare activity.

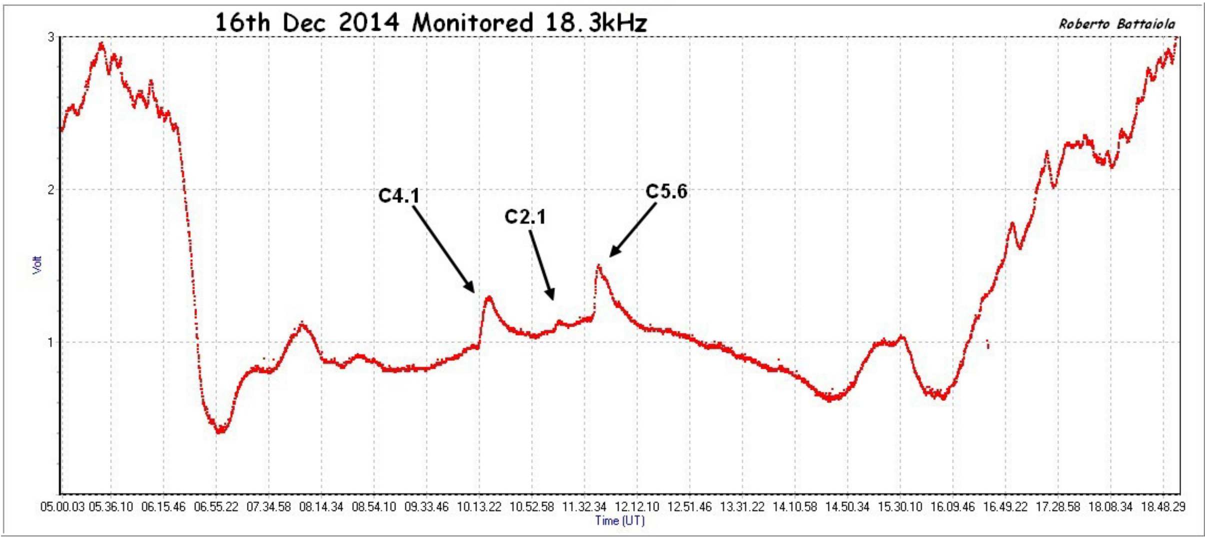


Fig 7: December 16th. Roberto Battaiola, Milan.

A series of CMEs produced some magnetic disturbance towards the end of December. The flares responsible all occurred around midnight and so were not recorded as SIDs. The last of these, an M6.9 flare at 21:58UT on the 21st, produced a strong SSC that I recorded at 11:14UT on the 23rd. The subsequent disturbance began around 21UT and lasted until 02:30 on the 24th. Further disturbances lasted through most of the holiday period.

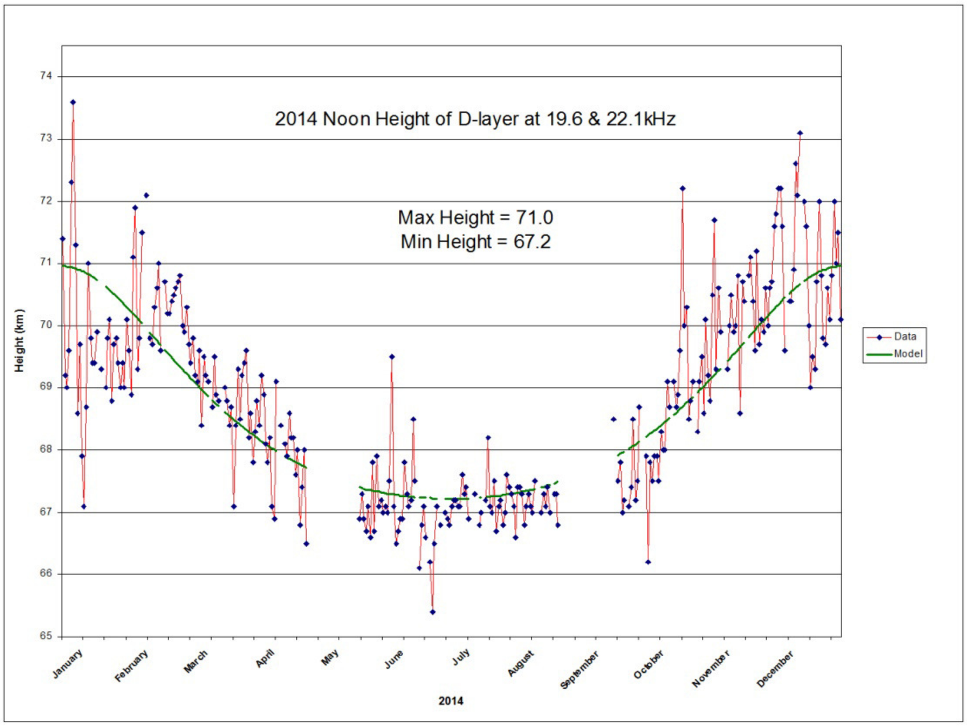


Fig 8: 2014 D-region height. Mark Edwards.

To round off the year, Mark Edwards has again provided a chart of D-region altitude modelled at 19.6 and 22.1kHz. The red lines are his measured data, and the green line is the modelled result. As can be seen, there is a lot of short-term variation in the measured data, particularly in the first few months. A similar chart for 2013 was published in Ragazine Vol. 1 issue 3. Charts for previous years were published in the monthly summaries, all available from the RAG web site.

Observers: Roberto Battaiola, Roger Blackwell, Colin Clements, Mark Edwards, John Elliott, Paul Hyde, Peter Meadows, Bob Middlefell, Steve Parkinson, Gonzalo Vargas, John Cook. My thanks to all contributors. If you would like to add your own observations, please contact jacook@jacook.plus.com.

2015 Solar Eclipse Notes

By John Cook

Solar observers are in for treat on March 20th with a Solar Eclipse. Greatest eclipse occurs over the sea between Iceland and the Hebrides, near the Faeroe Islands 64° 25'N, 6° 34'W). Throughout the UK there will be a good partial eclipse ranging from 85% in the south-east to 95% in the north of Scotland and Northern Ireland. This also presents an excellent opportunity for radio observers to record the event. Even better, the radio monitoring can be going on while eclipse progress is being watched (weather permitting!) through an eclipse viewer or projected by telescope. Do remember to take care however, remembering that even a 95% eclipse is too bright to watch with the unprotected eye. If in doubt about solar observing safety, contact the Solar Section for advice. Full details of eclipse magnitude throughout the UK can be found in an article by John Mason at www.britastro.org/article_render/6232.

Our normal VLF monitoring techniques should produce some interesting curves during the eclipse as the D-region begins to return to night-time conditions and back to day-time after maximum eclipse. First contact is about 07:40UT, maximum at 09:45, and final contact about 11:50UT (from Fred Espenak's fifty year eclipse cannon). Sunrise at the point of greatest eclipse is 05:37UT. Great circle paths from the UK to NAA, maine (24kHz) and NRK, Keflavic (37.5kHz) pass close to the path of totality to the west of maximum eclipse and should show a very strong response to the eclipse. Paths to Skelton and Anthorn are entirely within the partial eclipse area, as is the easterly path at 23.4kHz (but note that its usual morning break occurs during the period of interest). From northern Italy the eclipse will be about 70%, and should still produce a noticeable change in the diurnal curve.

The activity level of the sun at eclipse time is unknown of course, but a comparison of charts from the days before and after the 20th should provide a good base from which to judge its effect. Each of us has a different path for the various signals, and so as many observations as possible would be welcome for this event.

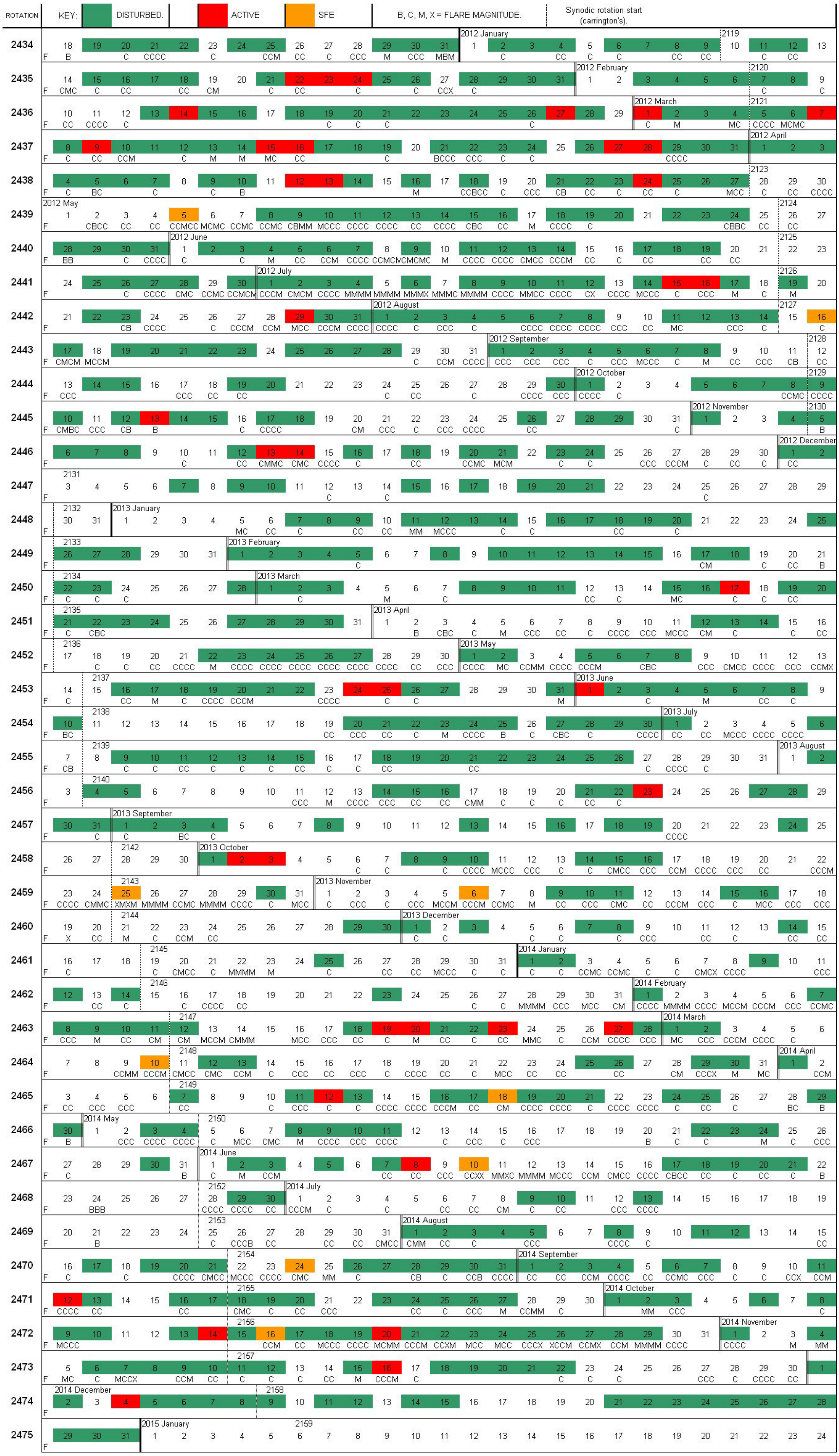


Fig 9: Bartels Diagram.

Meteor Forward Scatter
Paul Hyde

Unlike Meteor Backscatter, which utilises true radar techniques, Forward Scatter makes use of existing radio transmitters as the source for the reflections (scatter) from meteor trails. This makes the geometry more complex and you do not get the benefit of the ranging and true velocity information available from Backscatter systems. However Forward Scatter is a much more accessible technique as you do not need a high power transmitter or its associated OFCOM license. Both forward and backscatter techniques are used for professional meteor observing.

This year's Geminids provided a fine collection of meteor scatter signals with peaks of several hundred events per hour. Figure 1 shows activity as recorded by Paul Hyde (Basingstoke), Philip Norton (Lincoln) and John Wardle (Bridlington) whilst Figure 2 shows the same data normalized for comparison purposes. All three observers were monitoring the GRAVES radar station near Dijon, France which transmits at a frequency of 143.05 MHz. The activity rate is as recorded, without any correction to derive an equivalent to the Zenithal Hourly Rate figure used by optical observers.

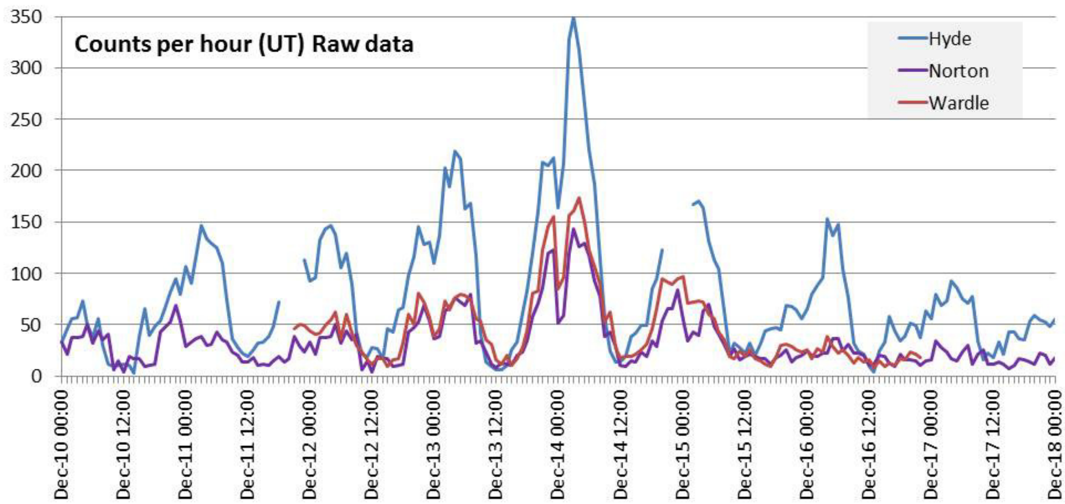


Figure 1 Hourly counts of meteor scatter reflections exceeding the trigger level

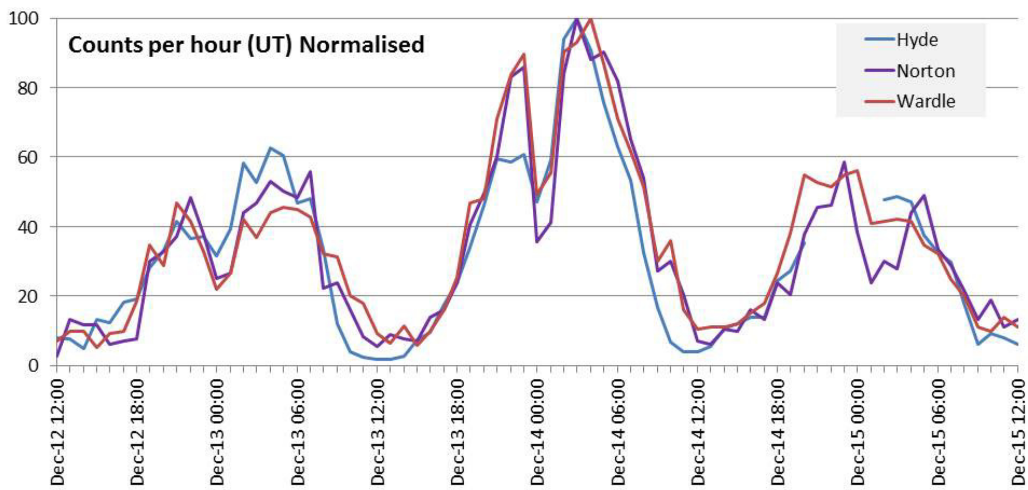


Figure 2 Data normalised to a peak of 100 and showing good agreement on the activity profile

The clear dip in the centre of the peak activity is not due to a change in the meteor flux density but is a consequence of the shower's Radiant moving into line with the receiver and transmitter locations. This is one of the factors that complicate meteor forward scatter, where you are dependent upon the reflection of energy from a source transmitter, whereas with optical observing the meteor is generating its own radiation. On the other hand, observing in the radio provides 24 hour/day coverage under all sky conditions, something impossible for optical astronomy.

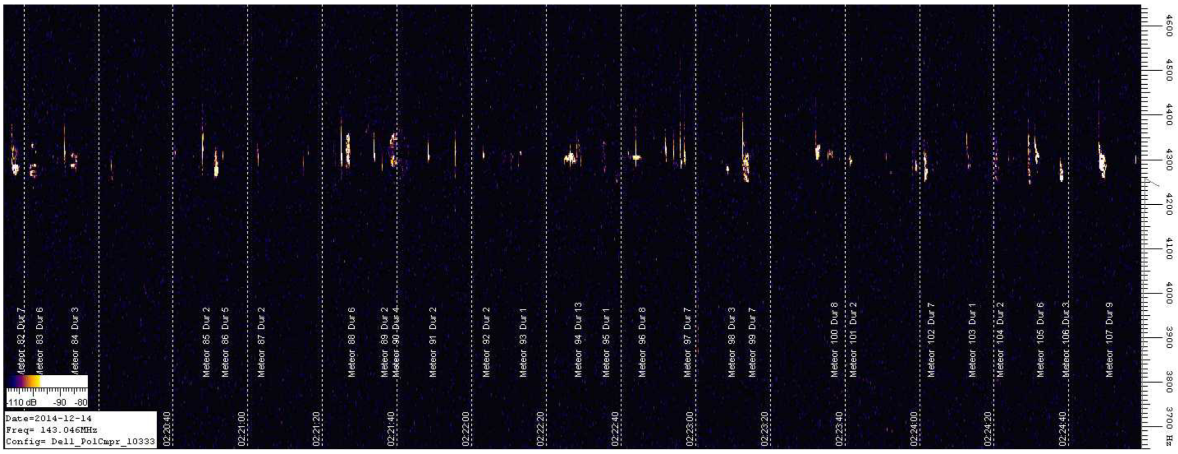


Figure 3 Individual meteor events with time markers at 20 second intervals and showing Doppler-shifted frequency on the vertical axis. A frequency of 4350 Hz equates to zero Line of Sight (LoS) velocity with reference to the observer. This 'wide-angle' view is used to check for interference and false triggers from ISS transits and from GRAVES signals reflected from the Moon.

Mass segregation in meteor streams

The automated system records the duration for which the reflected signal exceeds a pre-set trigger level and can provide some indication of particle size, on the basis that larger meteoroids produce more reflecting plasma. The amplitude of the received signal depends upon several other factors making this an imprecise measure, but given the large number of events it was decided to try comparing the percentage of longer duration events (duration counts of 4 and above, equating to 0.5 seconds) for the period of maximum observed activity. Figure 4 shows that the longer duration events comprised over 50% of the total for the morning of December 14 compared to just over 35% for December 13. This suggests that the average particle size was greater at the peak of the shower than during its initial stages. The (incomplete) data for December 15 also indicates a higher percentage of more massive particles for the latter stage of the event.

A possible reason for this is the 'mass segregation' process which forces lighter meteoroids towards the inner edge of the stream and which can be used to indicate the age of the meteor stream.

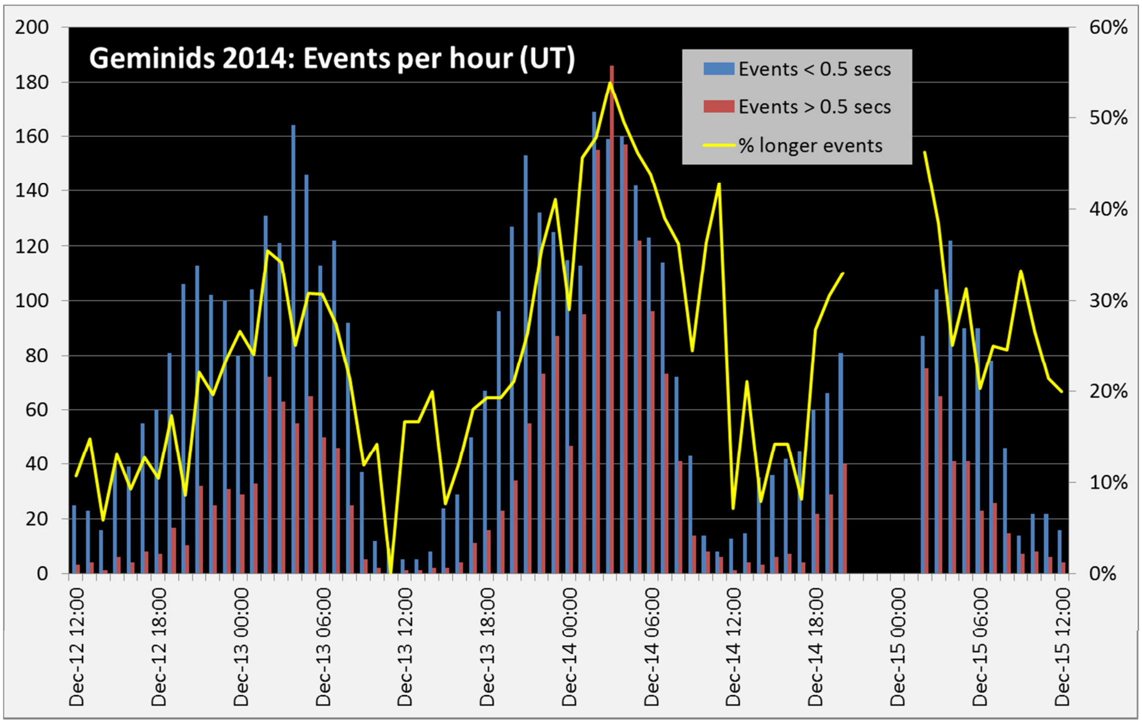


Figure 4 Distribution of longer versus shorter duration echoes for the period of peak activity. Normally the number of shorter events exceeds that of the longer duration ones but this was reversed for the peak of the Geminids.

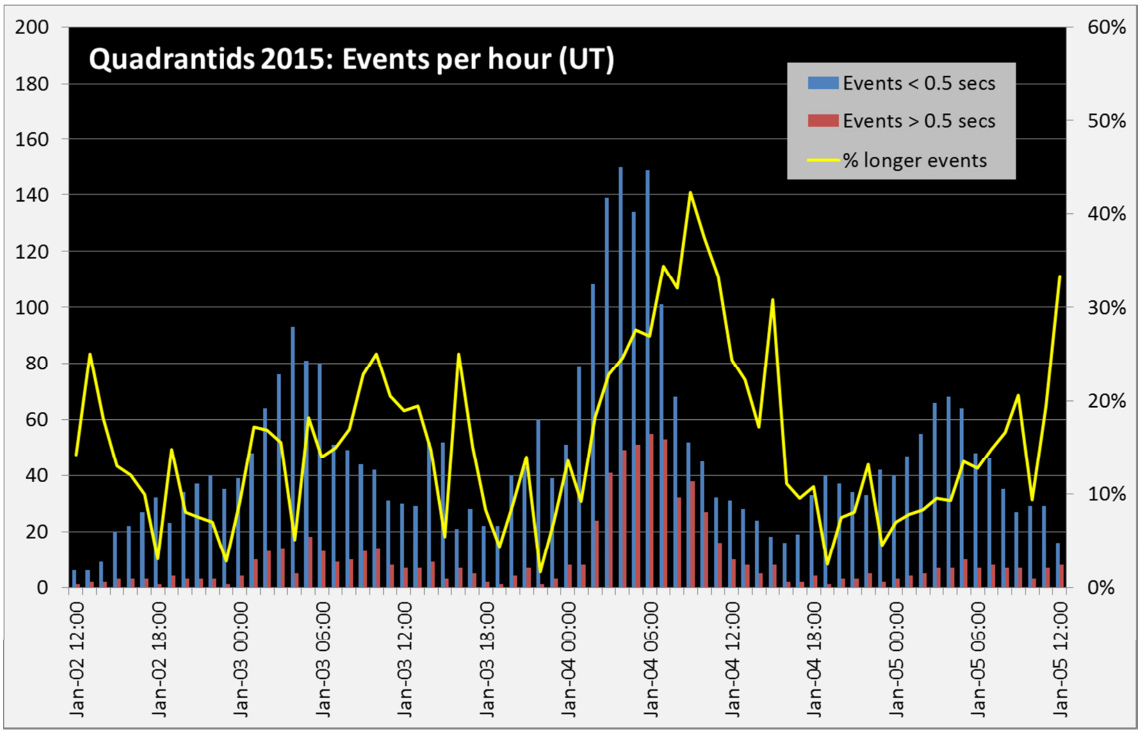


Figure 5 Equivalent ratios for the 2015 Quadrantid shower

Further observations of other meteor showers will be needed before anything more can be concluded, preferably observations received from a larger number of independent observers.

The effect of Sporadic meteors

The hourly meteor count is the sum of any shower activity overlaid upon that of background Sporadic meteors. Sporadic activity thus obscures the sort of analysis shown in the previous charts. Although sporadic activity varies throughout the year, it is relatively unchanging on a day-to-day basis and it is possible to derive an estimate based upon activity under non-shower conditions. This slow variation also offers a way of deriving both a profile for the sporadic activity and the Observability Function, a factor determined by the geometry and receiver characteristics for a specific observer/time and which modifies the true meteor flux density to produce the observed activity. be applied to a day's observations to produce an indication of the true meteor flux density. This is described in a paper by Steyaert, Brower and Verbelen.[1]

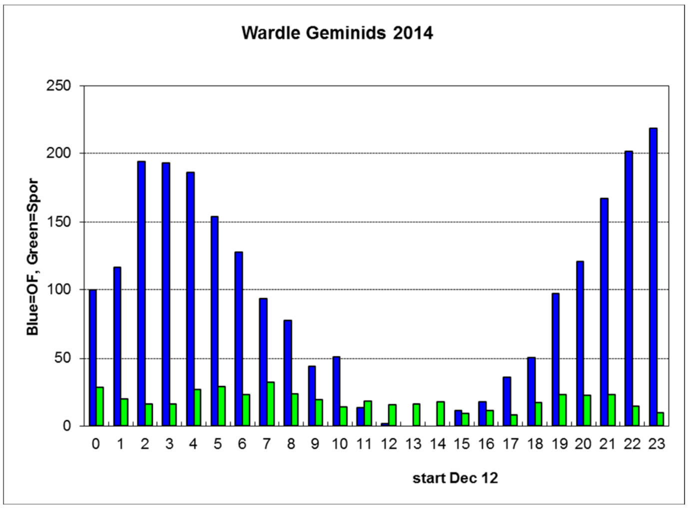


Figure 6 Derived Observability Function (Blue) and Sporadic profile (Green) obtained from John Wardle's observations (Courtesy Chris Stayaert, VVS /RMOB[2])

Figure 7 shows the agreement between the model and John's observations over the duration of the Geminids event with the Green curve showing the calculated activity and the Blue the actual observed activity.

Note that this method relies on both an unchanging sporadic profile over the period of observation and that the shower does not exhibit any outburst activity. In the case of the 2015 Geminids it produced an excellent match and it will be interesting to apply the technique to other shower events.

[1] A *numerical method* to aid in the combined determination of stream activity and Observability Function. Authors: *Steyaert*, Christian; Brower, Jeffrey; Verbelen, Felix ... WGN, Journal of the International *Meteor* Organization, vol. 34, no. 3, p.

[2] Association for Astronomy, Belgium/Radio Meteor Observing Bulletin

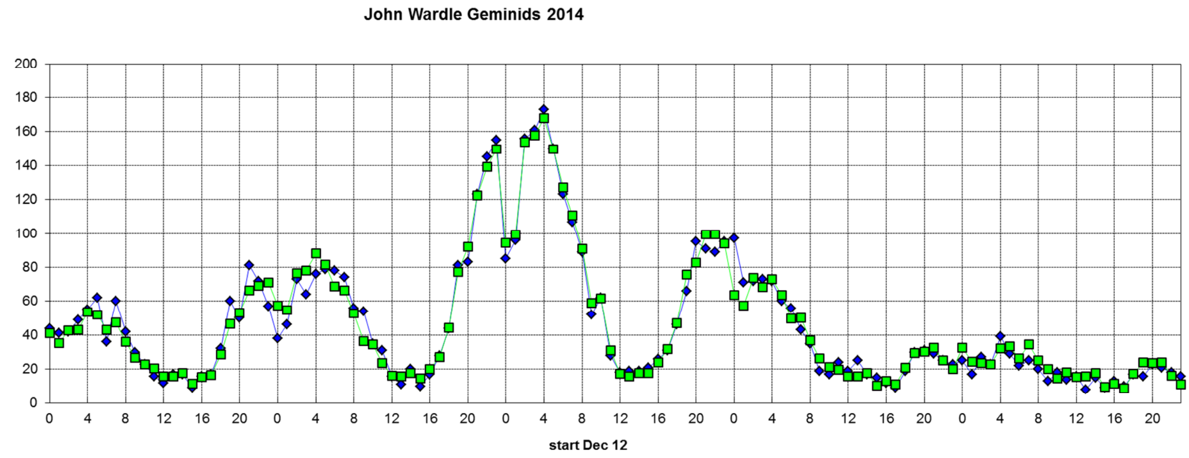
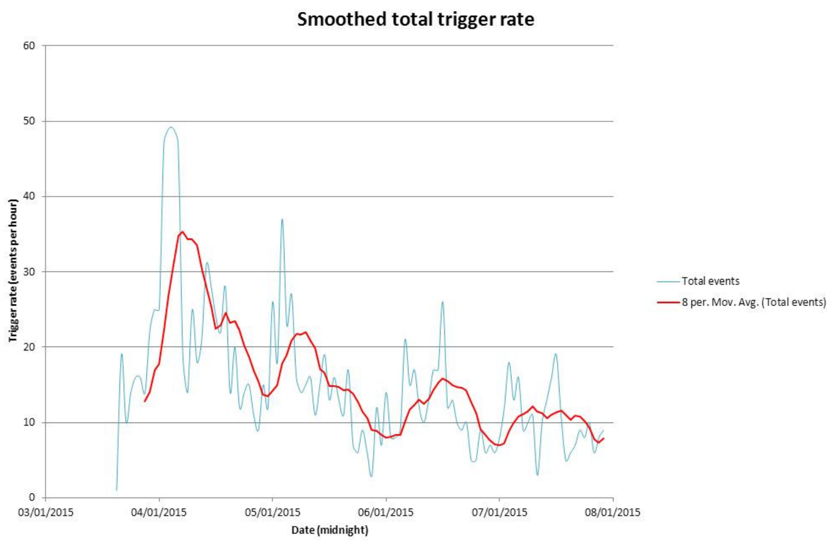


Figure 7 Modelled and Observed activity for 2014 Geminids (Courtesy Chris Stayaert, VVS /RMOB[3])

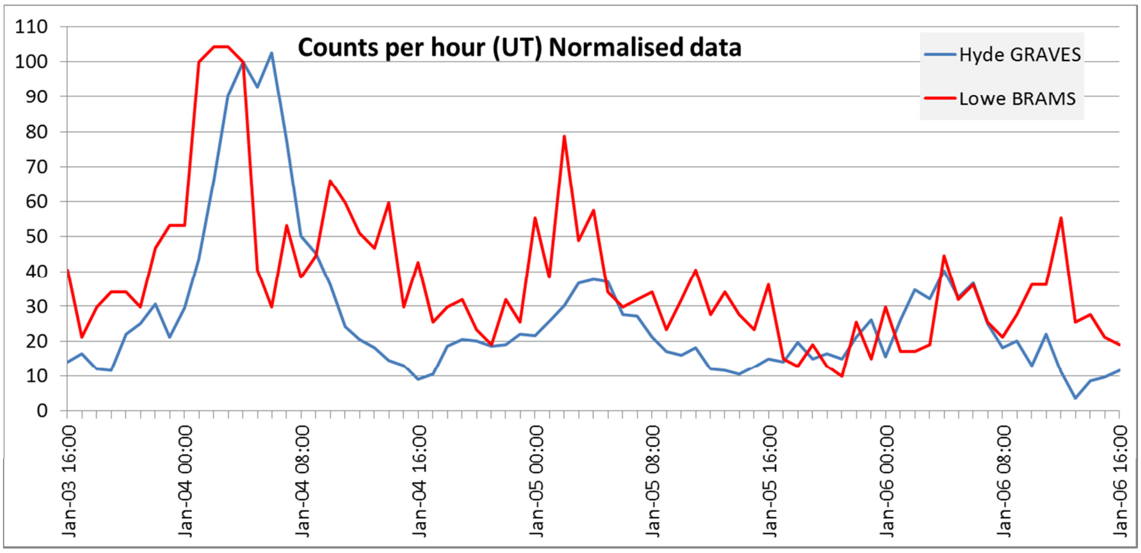
Observations using the BRAMS meteor beacon

All of the above observations were made using the GRAVES radar system. This produces a very strong 'searchlight' for illuminating meteors, but it also sweeps across the sky, complicating analysis of the results. The Belgium Radio Meteor Stations[4] system utilises a purpose-built beacon which provides a constant illumination pattern. It operates at 49.97 MHz, so requires a physically larger antenna than for GRAVES at 143.05 MHz. The lower frequency makes for a stronger reflection from meteor trails, but this is negated by the low power of the beacon, only 150 W compared to the estimated 10s of kW, if not higher, from GRAVES. This results in weak signals but Jim and Gerry Lowe have succeeded in producing results for the recent Quadrantids shower.



[3] Association for Astronomy, Belgium/Radio Meteor Observing Bulletin
[4] See http://brams.aeronomie.be/

Figure 8 shows Jim and Gerry's BRAMS observations and Figure 9 the results from both GRAVES and BRAMS, normalised for comparison. The BRAMS observations peaked at 49 events per hour compared to 204 events per hour for the GRAVES observations.



The BRAMS system is designed for use by observing stations in Belgium and neighbouring countries and path lengths for the UK make it a challenging source to use but it is better suited for serious meteor observing. VVS (the Belgium equivalent of the BAA) operates a similar beacon at 49.99 MHz and a long term objective of the BAA Radio Astronomy Group is to develop the resources and techniques to support a similar beacon system in the UK.

My thanks to the following observers for providing their data for inclusion in this report:

- **John Wardle** (Bridlington) uses a 4-element Yagi antenna and a Yaesu FT-817 communications receiver with Spectrum Lab software.
- **Philip Norton** (Lincoln) uses a 4-element Yagi antenna together with a FUNcube Dongle receiver and Spectrum Lab software.
- **Jim and Gerry Lowe** (Great Missenden) use a horizontally polarised 3-element Yagi (for 49.97 MHz) and a FUNcube Dongle Pro Plus receiver with Spectrum Lab software.
- **Paul Hyde** (Basingstoke) uses a vertically polarised 3-element Yagi antenna mounted at 3 metres above ground level, with a FUNcube Dongle Pro Plus receiver and Spectrum Lab software. Full details of the construction and setting up of the station can be found in the Sky at Night magazine article downloadable from the RAG website at www.britastro.org/radio/

Thanks also to Chris Stayaert for providing the modelling of the Geminid shower activity.

Paul Hyde

News Notes

A problem with meteor forward scatter is that you don't get any information on the trajectory of the meteor. The optical guys use this to help distinguish between Sporadic meteors and Shower meteors. Without this knowledge the sporadics tend to "pollute" any detailed analysis of the shower activity.

A group of French radio amateurs (Sylvain AZARIAN, F4GKR, Jean-Jacques MAINTOUX, F1EHN, Jeremy MAINTOUX and Frederic RIBLE, F1OAT) have been working on using passive radar techniques with FM broadcast transmitters as the sources. There are plenty of these transmitters around and they have relatively high output powers. By observing and correlating meteor reflections from different FM transmitters they have shown that it is possible to determine the change in the distance of the trail. By combining the results from 3 or more transmitters it should be possible to plot a 3D path for the meteor.

This method gets over the dependence upon GRAVES and avoids the cost of setting up beacons as used by the BRAMS system. It should also be a lot more sensitive given the transmitter powers used. As well as being able to sort the sporadics from shower events the authors note "This measurement could help to assess the meteor dispersion on atmosphere and moreover to reveal speed and direction of wind and probably more data interesting for the scientific people."

For more information on this work see RETRAM | REcognition and TRAjectory of Meteors and the list of publications contained on that site. Some fascinating and inspiring material!

Paul Hyde

Raspberry Pi Model 2 is out

A new version of the Raspberry Pi has just been released, known as the Model 2. It appears very much like the B+ model with the same extended GPIO interface, and four USB ports. However under the hood it sports a 4 core microprocessor, running a little faster at 900MHz, and has twice the RAM at 1GB.

Reports suggest it is up to six times faster than its predecessor. As soon as I saw them earlier this month I had to have one, and immediately ordered one from CPC for under £30. While I have not had much chance to fully evaluate it yet, it should offer significant improvements for some real time processing. To take full advantage of the performance gains, it will be necessary to use parallel processing techniques. For example it is capable of running 4 concurrent threads. I will report results in a future issue.

There may be a short term delay in getting hold of one, but they are not expecting waiting lists of months as was seen on its first release.

Jeff Lashley

Using the NWT4 Spectrum Analyser and Tracking Source

Paul Hyde

Peter East's website[5] carries a paper describing how to use the NWT4 RF Digital Analyser and Tracking Source for aligning and testing antennas and filters for Hydrogen Line reception. These are available for around £55, including shipping, from a number of suppliers advertising on e-bay. There are different versions of the analyser available, some being sold without the Tracking Source, so be sure to order the right one! [6]

Despite the illustration provided on the internet, neither Peter nor I received a 12V DC power supply for the Tracking Source, though the USB cable and two SMA test cables for the Analyser were include. Delivery took about two weeks.

The User Guide provided on the accompanying CD-ROM is one of the most unreadable documents I've ever come across and is just about useless. A typical example is the phrase "Translation as a result of access, to Chinese prevail" which I think is intended to mean that the original Chinese text takes precedence over the translation.

I found that the software drivers loaded automatically under Windows 7, though others have reported having to use the versions on the CD-ROM. However I was not able to run the main installer application until I had copied it across onto my Desktop and, as Peter notes, deleting any Chinese characters in the file names, along with any other illegal characters such as commas – thanks to Victoria Penrice for spotting that one!

I also purchased a second-hand three-port Directional Coupler via e-bay for around £30, including postage. This was manufactured by Narda and rated for 1 to 2 GHz operation. Similar devices appear regularly and prices seem to be predominantly driven by the specified frequency range. It's not worth paying for coverage that you don't intend to use or for one that is designed to handle a lot of RF power.

I used these components to align interdigital filters and a 22-element Yagi which I had previously

[5] See http://y1pwe.co.uk/RAProgs/index.html

[6] Search on 138M-4.4G SMA signal source, generator, simple spectrum analyzer +Tracking source

constructed using the design calculators and techniques described papers by Peter and which are also available on his website. I was lucky enough to have the temporary use of a Tektronix 2712 Spectrum Analyser and HP 8657B Signal Generator for comparison purposes. Professional test equipment of this type costs several thousand pounds to purchase as second-hand equipment and well into five figures as new, so you have to make some allowances given the £55 purchase cost!

Noise Source into Tektronix 2712 Spectrum Analyser

The 'Tracking Source' is really a wideband noise source which can be used for testing filters and antennas. The main limitation is in the low signal levels available, particularly at the higher frequencies.

The Tektronix 2712 Spectrum Analyser was first used to measure the output levels using a 10 dB pad between the Noise Source and the Analyser input. Measurements were at a Bandwidth setting of 300 kHz. The following results take account of the attenuator.

Frequency (MHz)	Noise Source O/P (dBm)
0	-10
10	-14
50	-15
100	-17
200	-20
500	-20
1000	-24
1500	-26
1800	-32

NWT4 Signal Generator into Tektronix 2712 Spectrum Analyser

You can use the NWT4 kit to generate a signal, but only as an alternative to the spectrum analyser function.

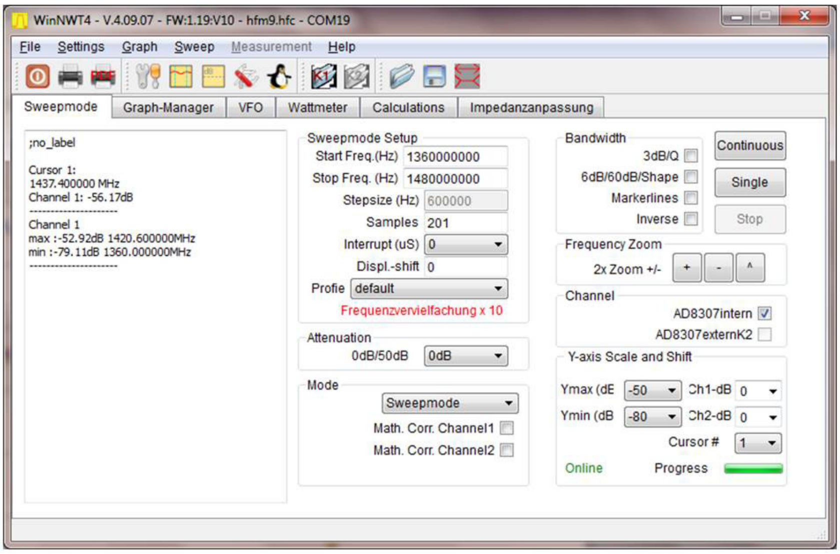
Entered Freq (MHz)	Measured Freq (MHz)	Measured Level (dBm)
130	130.0	-13.0
140	139.2	-12.7
145	144.0	-12.4
200	200.0	-12.7
1200	1198.0	-13.5
1800	1800.0	-15.4

First harmonics (2 x O/P frequency) were better than -50 dBm but second harmonics (3 x O/P frequency) were at -22 to -23 dBm, i.e. only about 10 dB below the Fundamental.

Setting up 1420 MHz filters

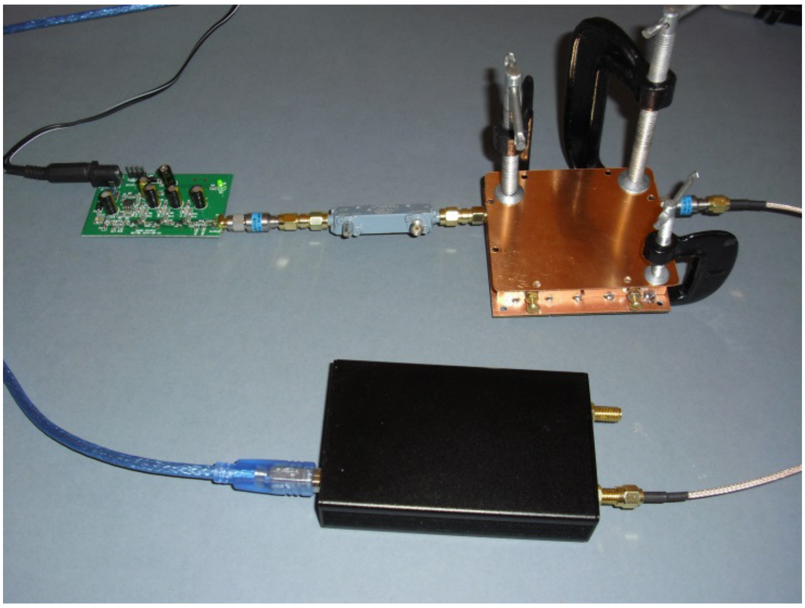
So far I have built and aligned four filters for Hydrogen Line work. The first followed Peter's design using 0.9 mm copper plate. The problem with this material is soldering the resonant elements (rods) to the casing as the copper sheet conducts away the heat too quickly. I had hoped that a 100W Weller solder gun would suffice, but this proved inadequate. A mini blow torch was powerful enough but it was impossible to control the solder flow and, frankly, dangerous to try. A culinary-type blow torch worked, but the best solution proved to be a gas-powered soldering iron (Maplin N95AP) which came with a blow torch attachment. This provided a nicely controllable pencil flame which enabled me to fit and adjust the rods and to solder on brass nuts to hold the adjustment screws.

The other three filters were constructed from double sided printed circuit board which is a lot easier to handle and which can be constructed using a standard soldering iron.

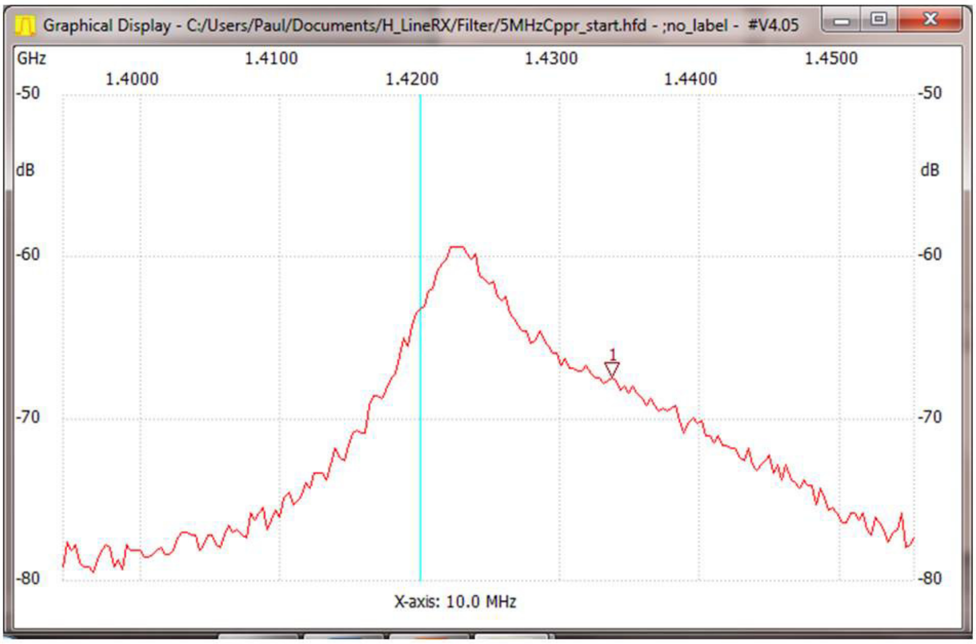


Configuring the NWT4 Spectrum Analyser. Note the need to use the x10 frequency setting (under Settings and then Options) for working at 1420 MHz

For setting up the filters I used the test configuration described in Peter's note but also inserted the Directional Coupler to measure the amount of signal reflected back from the filter. The lid was clamped on for initial testing, to allow the length of the rods to be modified if needed. Once the performance has been verified the lid can be soldered on prior to rechecking and tweaking the adjustment screws if needed.

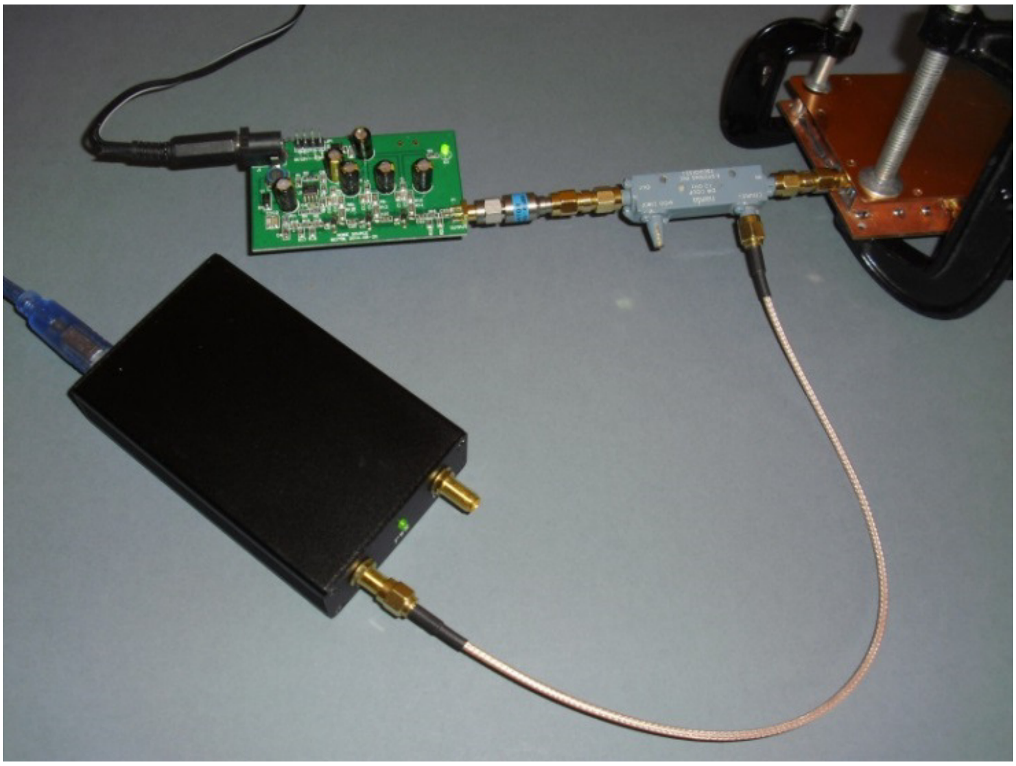


The first measurement was made with the adjustment screws removed. The length of the centre rod was adjusted for a resonant peak at around 1430 MHz. If the peak was too high the rod was lengthened by applying heat to the solder joint and gently pulling the rod through. For a low frequency peak the rod was shortened using a needle file. In both cases you only need to vary the length by a fraction of a millimetre at a time.

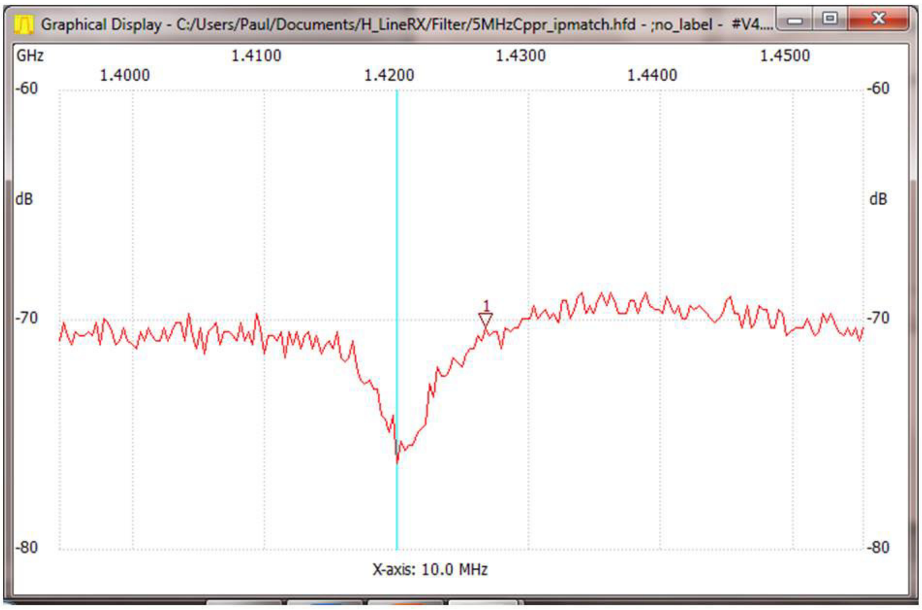


The centre tuning screw was then inserted and the response centred on 1420 MHz.

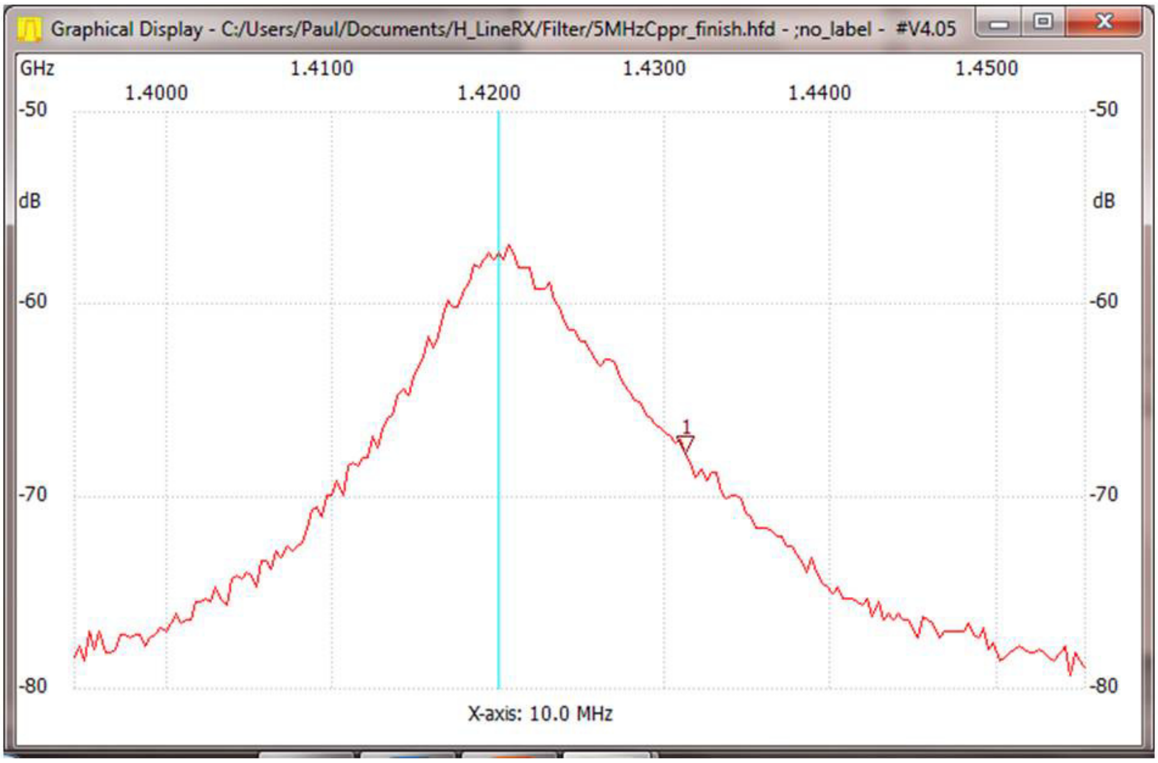
I then looked at the power being reflected back from the input port of the filter. To do this you need to connect the Noise Source to the output of the Directional Coupler and the filter to the input. The output of the filter should be terminated with a 50 ohm load.



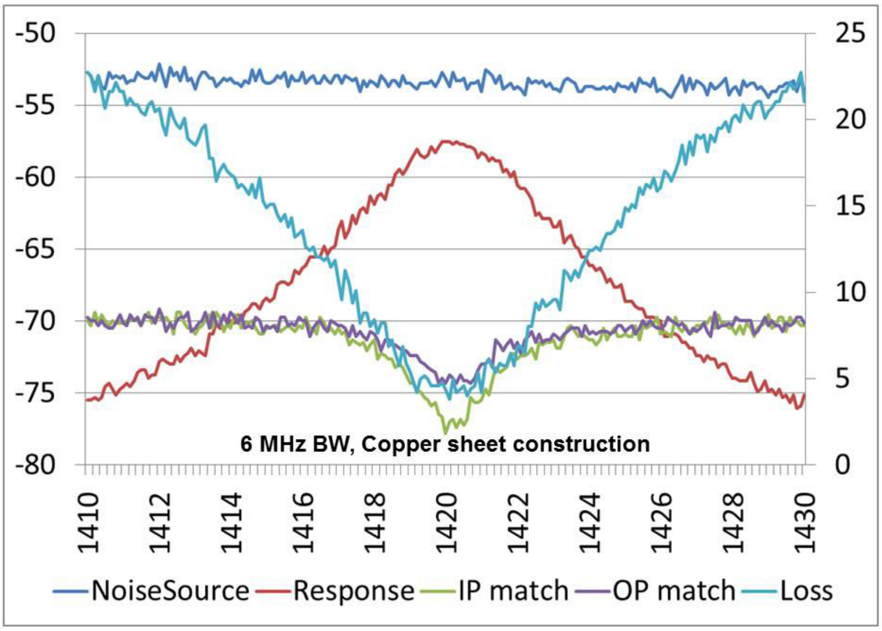
This time the adjustment screw on the input rod was used to minimize the amount of power being reflected back from the filter input port. The filter was then flipped around and the process repeated for the output port - with the input port now being terminated.



The filter was returned to the original test configuration and all three adjustment screws 'tweaked' to provide the best overall response.

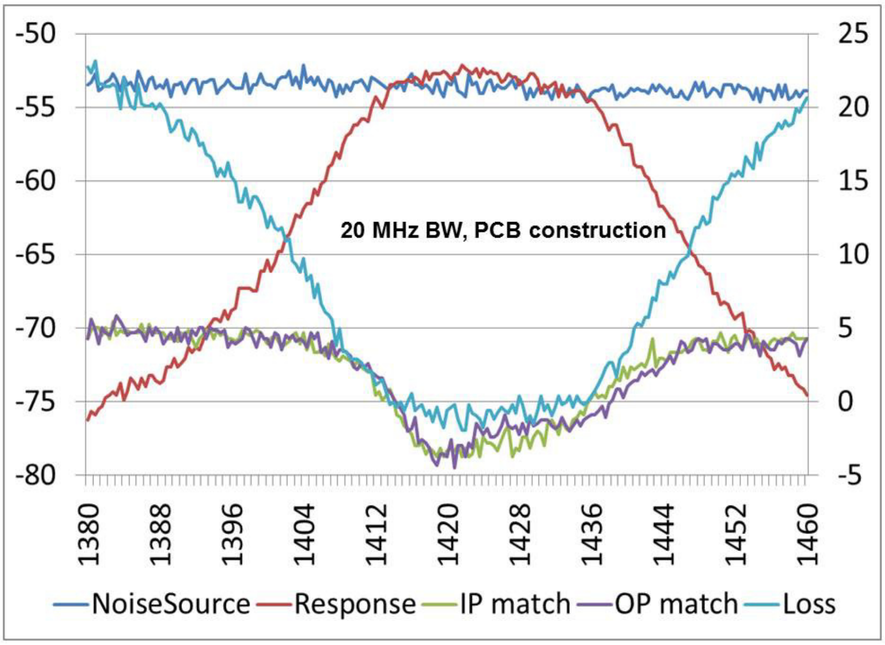


Finally the overall performance of the filter was checked by replacing it with a barrel connector and measuring the output of the test generator. The measured loss for the filter was 5 dB, compared to a theoretical value of 2.5 dB.



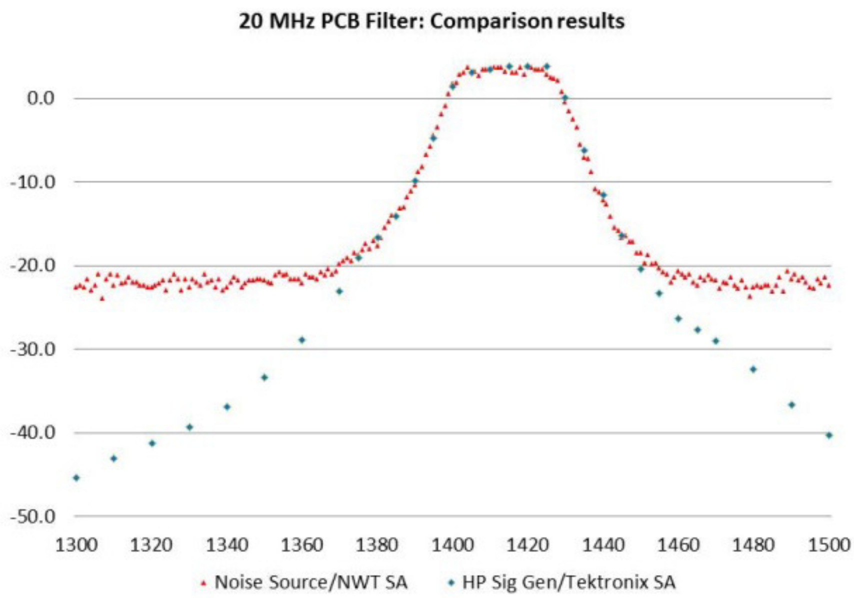
N.B. Loss is measured using the right hand axis

I then repeated the process with three filters manufactured using printed circuit board. The first was designed for a 20 MHz bandwidth with the intention of using it in place of the two expensive 100 MHz Mini-Circuits filters used in the second version of Peter's receiver.

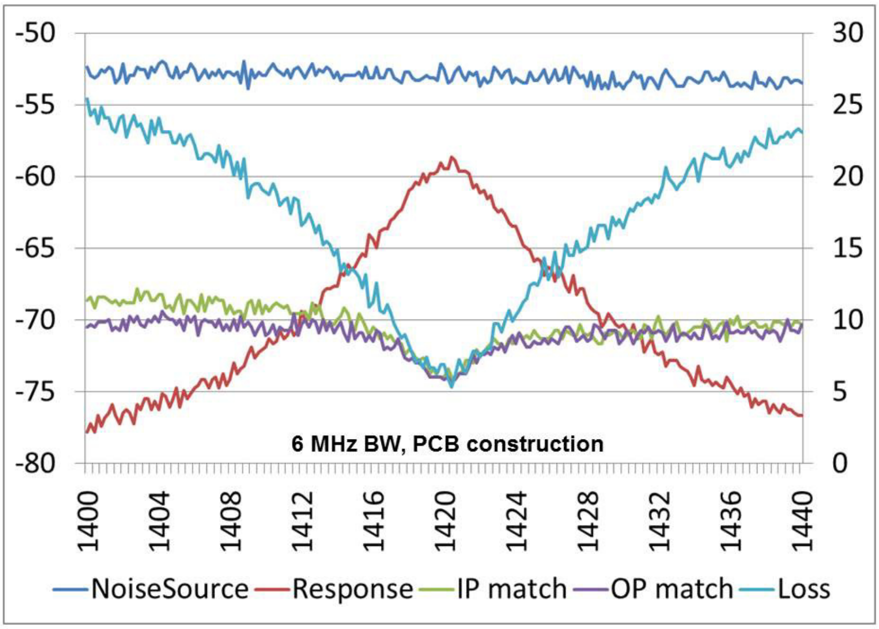


The results suggest that this filter has some gain! This is plainly impossible so the filter either loads the Noise Generator, despite the 6dB attenuators on either side, or the Spectrum Analyser measurement is depressed for a wideband noise signal.

For comparison the HP signal generator and Tektronix spectrum analyser were used to measure the response of this filter. Note the benefit of the much higher output from the professional equipment. The NWT4 measurements are limited by the noise floor.



Finally a 6MHz bandwidth version of the PCB filter was aligned and found to have a loss of 6.5 dB, as compared to the 5dB for the copper plate version. A post mortem revealed that the filter was a millimetre out on the critical internal dimension of 52.8 mm (a quarter wavelength at 21 cm) and a second PCB version achieved the same 5dB loss as the copper plate filter.



Conclusions

The NWT4 Spectrum Analyser and associated Noise Source is a handy tool for £60 and the combination is quite adequate for setting up simple filters. You don't really need the Directional Coupler and I tuned-up my final attempt by adjusting the centre rod to get the peak in the right place and then adjusting the input and output screws for maximum signal. However a Directional Coupler does enable you to see where the problems might be and, importantly, can also be used to measure the match of simple Yagi antennas.

Paul Hyde

Low Cost Hydrogen Line Radio Telescope using the RTL SDR - Phase 3

Abstract
This paper describes the phase 3 upgrade, further improving the receiver. A dual channel switched receiver based on the pseudo-correlation continuous comparison receiver[1,2] is described that seeks to minimise the effects of gain variations. Early results show excellent repeatable performance.

Introduction

In the basic receiver paper[3] , it was noticed that receiver gain (60dB RF and 49dB for the RTL SDR) drifted between measurements affecting both FFT average and ratio/temperature determination. In this upgrade a switched correlation receiver approach is adopted to minimise the effect of gain drift. Receiver units were boxed in aluminium to further reduce external interference. Two parallel channels, as well as requiring more hardware, unfortunately require two SDR dongles and computers capable of synchronised recording. Ideally the dongles should be matched, but innate frequency tuning accuracy can be compensated.

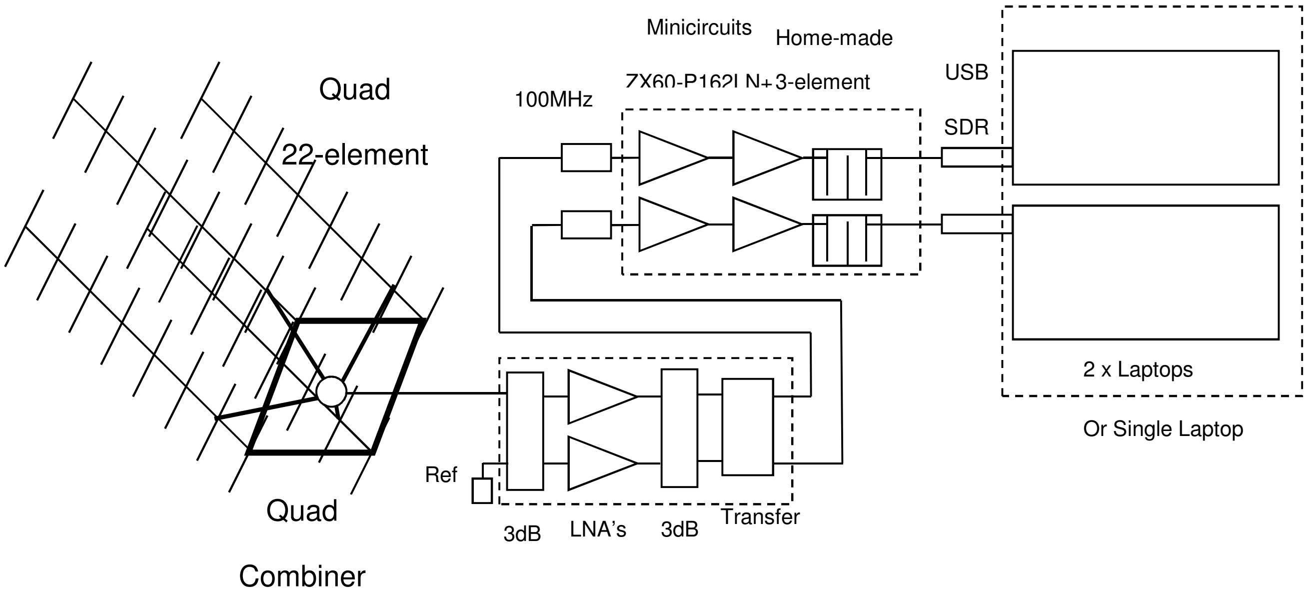


Figure 1. Simple Radio Telescope Upgrade 3

Figure 1 shows the upgrade schematic. The receiver is now boxed in three units, the switched preamplifier closely attached to the antenna and the remaining twin channels, placed remotely near the SDR/Laptop processors. If the PC/Laptop computer has two separate USB ports then it is possible to run two SDR dongles on one computer, otherwise two synchronised computers are necessary. Synchronising software is identical for either option (see Appendix 2).

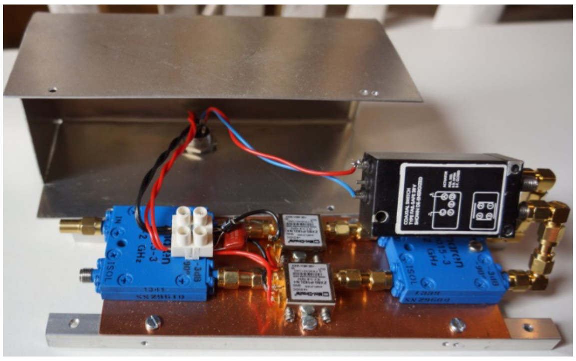


Figure 2a Preamplifier/Switch Unit

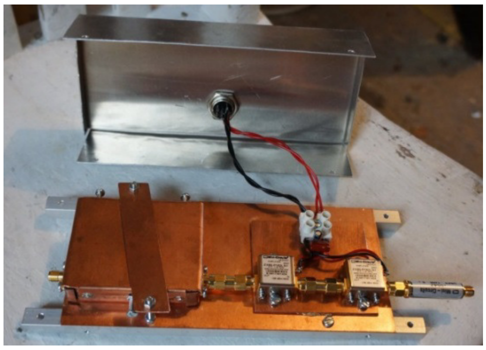


Figure 2b Post-amplifier/Filter Unit

Receiver Description

The receiver architecture follows the basic form of the earlier receivers and similarly applies the ratiometric Dicke-Fix[5] processing described previously.

The preamplifier/correlation receiver (Figure 2a) is now comprised of two 3dB couplers from Anaren (1-2GHz type: 10015-3) and two low noise amplifiers (Minicircuits ZX60-P162LN+) with a transfer switch (Sivers type PM7553) at the output. The couplers may be 90° or 180° types (see Appendix). The first coupler splits the input signals equally between the two similar amplifiers, whilst the output coupler recombines the amplified signals and splits them to the output ports such that the amplified input signals are isolated and exit different ports. Effectively, both input signals pass through both amplifiers and so are equally affected by any amplifier gain changes so that any subsequent comparison preserves that of the two inputs. Detailed analysis is given below.

The correlation receiver outputs are amplified and filtered in the secondary twin channel units shown in Figures 1 and 2b that feed a corresponding pair of RTL2832U SDR dongles. Operation of the transfer switch re-directs the preamplifier output signals to the opposite amplifier chain and SDR dongle. This enables compensation of the two channel amplifier/dongle gain differences.

In addition, it may be necessary to adjust spectrum measurements or dongle frequency settings to match different SDR frequency measurement accuracies.

Receiver Operation Analysis

Figure 3 shows a basic schematic of the total receiver system.

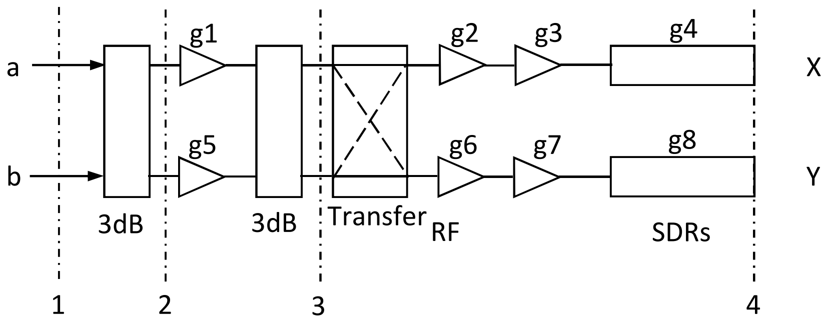


Figure 3 Twin Channel Switched Receiver Chain

The top channel is denoted 'X' with a generalised input signal 'a' and the lower, second channel 'Y' with an input signal 'b'. 180° couplers are assumed so that the voltage outputs of the input coupler, at position '2' after the first coupler in Figure 3, are $\frac{a}{\sqrt{2}}+\frac{b}{\sqrt{2}}$ in channel 'X' and $\frac{a}{\sqrt{2}}-\frac{b}{\sqrt{2}}$ in Channel 'Y', as shown in Table 1.

Channel	1	2	3	4
X	a	$\frac{a}{\sqrt{2}}+\frac{b}{\sqrt{2}}$	$\left(\frac{a+b}{2}\right)g1+\left(\frac{a-b}{2}\right)g2$	$\left(\left(\frac{a+b}{2}\right)g1+\left(\frac{a-b}{2}\right)g5\right)g2g3g4$
Y	b	$\frac{a}{\sqrt{2}}-\frac{b}{\sqrt{2}}$	$\left(\frac{a+b}{2}\right)g1-\left(\frac{a-b}{2}\right)g2$	$\left(\left(\frac{a+b}{2}\right)g1-\left(\frac{a-b}{2}\right)g5\right)g6g7g8$
X_{Sw}	a	$\frac{a}{\sqrt{2}}+\frac{b}{\sqrt{2}}$	$\left(\frac{a+b}{2}\right)g1+\left(\frac{a-b}{2}\right)g2$	$\left(\left(\frac{a+b}{2}\right)g1-\left(\frac{a-b}{2}\right)g5\right)g2g3g4$
Y_{Sw}	b	$\frac{a}{\sqrt{2}}-\frac{b}{\sqrt{2}}$	$\left(\frac{a+b}{2}\right)g1-\left(\frac{a-b}{2}\right)g2$	$\left(\left(\frac{a+b}{2}\right)g1+\left(\frac{a-b}{2}\right)g5\right)g6g7g8$

Table 1 Channel voltages - 180° Couplers

In Table 1, X and Y are the channel outputs with the transfer switch in the straight through position. XSw and YSw are the outputs when the transfer switch is in the crossover position. The amplifier gains gn are voltage gains and gn^2 are the corresponding power gains. For proper combination at the output coupler it is essential to ensure that the through phase of both channels is near identical.

Channel	Power
X	$\left(\frac{a^2}{4}(g1+g5)^2+\frac{b^2}{4}(g1-g5)^2\right)g2^2g3^2g4^2$
Y	$\left(\frac{a^2}{4}(g1-g5)^2+\frac{b^2}{4}(g1+g5)^2\right)g6^2g7^2g8^2$
X_{Sw}	$\left(\frac{a^2}{4}(g1-g5)^2+\frac{b^2}{4}(g1+g5)^2\right)g2^2g3^2g4^2$
Y_{Sw}	$\left(\frac{a^2}{4}(g1+g5)^2+\frac{b^2}{4}(g1-g5)^2\right)g6^2g7^2g8^2$

Table 2 Power Output

Table two shows that if the gains g1 and g5 are not equal then there is signal leakage between the outputs and therefore the ratio measurement will be distorted. However, further analysis shows that if the gain difference is less than about 1dB the error/leakage term can be ignored.

The two measurement ratios are,

$$R1 = \frac{Y_{Sw}}{Y} \approx \frac{a^2}{b^2} \frac{(g1+g5)^2 g6^2 g7^2 g8^2}{(g1+g5)^2 g6^2 g7^2 g8^2}$$

$$R2 = \frac{X}{X_{Sw}} \approx \frac{a^2}{b^2} \frac{(g1+g5)^2 g2^2 g3^2 g4^2}{(g1+g5)^2 g2^2 g3^2 g4^2}$$

Taking the square root of the product of these ratios, we get,

$$R = \sqrt{R1R2} = \frac{a^2}{b^2}$$

Demonstrating that this result produces a virtually gain-invariant measure of the input powers or temperature.

For completeness it can be shown that the error in R, dR due to non-equal preamp section gains is,

$$dR = c^2 \frac{1-R^2}{1+Rc^2}, \ \ where, \ c = \frac{1-g5/g1}{1+g5/g1}$$

For all values of R, between 0 and 1, the error in R is less than 0.012 if, 1.25>g5/g1>0.8, implying the preamplifier gains need to match to better than about 2dB.

Receiver Operation – Software Control

With the transfer switch in the straight-through position and the antenna directed at the wanted source, both SDRs are initiated simultaneously to record digitised data for a set time. Similarly, with the transfer switch in the crossover position, simultaneous operation of the SDRs for the same duration, generates a second pair of data files.

To achieve simultaneous recording, two terminal windows, one for each SDR, run a copy of the control software that is set with the desired recording parameters.

Typical control software commands to run two SDRs (-d 0 and –d 1) on a single computer for MS WINDOWS are,

>RN_RTLAT "./rtl_sdr dat0.bin –f 1420e6 –d 0 –g 49 –n 100e6" 256 14 24 00
>RN_RTLAT "./rtl_sdr dat1.bin –f 1420e6 –d 1 –g 49 –n 100e6" 256 14 24 00

These are run in two Command windows set to the working directory, which also contains the Osmocom rtl_sdr tools. Any measured frequency offsets of the SDRs can be compensated in the '–f ' frequency command figure so that both output spectra align.

These commands start recording data from both SDRs at 14hr 24m 0s using the Osmocom rtl_sdr program. In this example, on collection of 100 million samples, the data is analysed in 256 point blocks using a FFT algorithm and outputs averaged spectrum text files dat0.txt and dat1.txt.

Operating the transfer switch and resetting the software commands as required, a second set of files dat3.txt and dat4.txt can be generated.

Inputting these four data text files into Excel or mathcad software the temperature ratio graphs can be produced indicating H-Line characteristics.

Receiver Test Measurements

The antenna measurements recorded and presented in this section used a single 22-element Yagi tuned to 1420MHz. The aim of the measurements was to confirm any improved receiver stability and to calibrate the ratio plots.

Two sets of measurements were made. The first set involved no antenna and the antenna port was left open-circuit, simulating a zero antenna temperature (Tant = 0). Note that the open circuit input could also change the low-noise amplifier noise figure and so may not provide an accurate measure of the system noise temperature, but will give a rough guide.

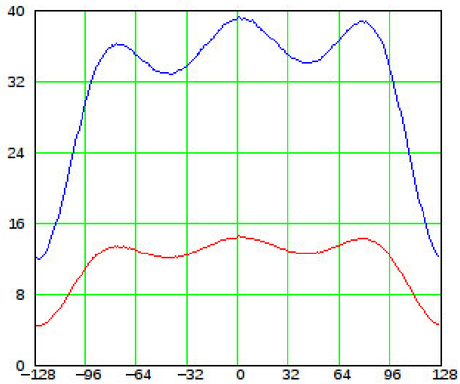


Figure 5a Channel X Spectra

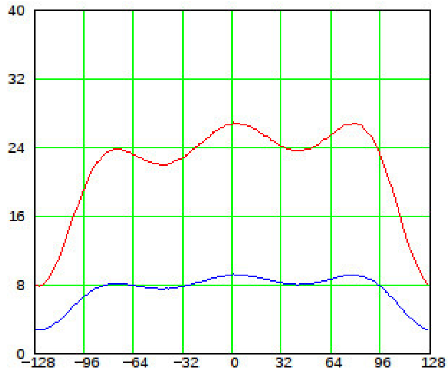


Figure 5b Channel Y Spectra

Figure 5 shows the spectrum plots for the two transfer switch states (Red - switch state 1; Blue – switch state 2). It is immediately evident that there is a significant gain difference between the two channels of some 4dB, and the band profiles do not track exactly.

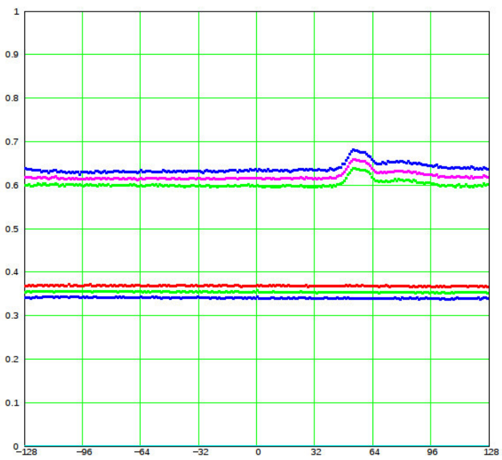


Figure 6 Spectrum Ratios: lower plots – open-circuit input; upper plots – antenna input

The reference/cold ratio appears to be approximately 4.5dB – this can be more accurately determined by forming the ratio as shown in the lower half of Figure 6 (Red – R1, Y channel ratio; Blue – R2, X channel ratio; Green plot – R, the square root of the product of the two channel ratios).

Referring to the lower plots in Figure 6 corresponding to the receiver with an open-circuit at the normal antenna input; the red line is the spectrum ratio from channel X and the blue the ratio of the channel Y measurements (Figure 5). The green line corresponds to the square-root of the product of the X and Y channel measurements.

We can re-calibrate the ratio ordinate by noting that, when T_{ant} = 0, from Figure 6, R = 0.35

With these substitutions in the ratio equation $R = .\dfrac{T_{ant} + T_{sys}}{T_{ref} + T_{sys}}$, together with T_{ref} = 290°K, we find,

T_{sys} = 156°K (equivalent to 1.9dB noise figure).

This seems rather high, since the RF amplifiers used have a quoted noise figure of 0.6dB, equivalent to 43°K. The 3dB coupler may have a loss of around 0.5dB (~35)°K.

The assumption that with an open or short circuit amplifier input, it still maintains its quoted noise figure may be suspect. There is also the possibility that coupler isolation could affect the measurement accuracy.

Figure 6 can now be calibrated, since R=0.35 is equivalent to 156°K and from the ratio definition above, ratio R = 1, corresponds to a temperature of 290°K (T_{ref}) then, for any other value of R,

$$T_{ant} = 446R - 156$$

The upper plots in Figure 6 were measured with the Yagi connected to the preamplifier input (Blue – R1, Y channel ratio; Green – R2, X channel ratio; Magenta plot – R, the square root of the product of the two channel ratios).

The ratio corresponding to the magenta plot baseline is approximately 0.615 and the ratio at the hydrogen line peak is 0.66. Using the equation above, the antenna temperatures are calculated as 118.3°K and 138.3°K, respectively. The first is a measure of the antenna sidelobe power and resistive losses. The difference 20°K is the indicated hydrogen line temperature. The antenna beamwidth is about 26° and, assuming the hydrogen source is mainly constrained to a band 5° centred on zero galactic latitude, then the measure power represents about a fifth of the true hydrogen temperature, now indicating 100°K which is near to that expected.

Assuming the antenna achieves its designed gain (24dB) and the antenna + system temperature is as calculated above (156+118.3 = 274.3 ≅ 24dB) making the G/T of the system, 0dB.

Conclusions

Gain measurements show that in the correlation receiver section, the amplifiers in both channels match well in both gain and phase. Isolation is better than 20dB. Tests similarly combining the twin amplifier channels with 3dB couplers were inconclusive and appeared not to completely separate the input signals.

However the present receiver architecture did demonstrate better stability between measurements than the simpler earlier versions.

More information is available at http://www.y1pwe.co.uk/RAProgs/index.html

References

1. Harris et al, "Design Considerations for Correlation Radiometers", http://www.astro.umd.edu/~harris/kaband/gbt_memo254.pdf
2. Mennella et al, "Advanced pseudo-correlation radiometers for the Planck-LFI instrument", http://www.deepspace.ucsb.edu/wp-content/uploads/2013/02/Advanced-Pseudo-Correlation-Radiometers-for-the-Planck-LFI-Instrument-Proceedings-20031.pdf
3. http://www.y1pwe.co.uk/RAProgs/HLRrtl.pdf
4. http://www.y1pwe.co.uk/RAProgs/HLRrtl2U.pdf
5. http://www.y1pwe.co.uk/RAProgs/Ratiometric Dicke Radiometer.doc
6. http://sdr.osmocom.org/trac/wiki/rtl-sdr
7. http://www.y1pwe.co.uk/RAProgs/dos/AMP_STS.EXE
8. http://www.y1pwe.co.uk/RAProgs/linux/amp_sts64.exe
9. http://www.y1pwe.co.uk/RAProgs/linux/amp_stats
10. http://www.y1pwe.co.uk/RAProgs/dos/BIN_TXT.EXE
11. http://www.y1pwe.co.uk/RAProgs/linux/bin_txt64.exe
12. http://www.y1pwe.co.uk/RAProgs/linux/bin_txt
13. http://www.y1pwe.co.uk/RAProgs/dos/RAFFT.EXE
14. http://www.y1pwe.co.uk/RAProgs/linux/r64fft.exe
15. http://www.y1pwe.co.uk/RAProgs/linux/rafft
16. http://www.y1pwe.co.uk/RAProgs/dos/RAFFT2.EXE
17. http://www.y1pwe.co.uk/RAProgs/linux/r64fft2.exe
18. http://www.y1pwe.co.uk/RAProgs/linux/rafft2
19. http://www.y1pwe.co.uk/RAProgs/dos/RN_PRGAT.EXE
20. http://www.y1pwe.co.uk/RAProgs/linux/r64_prg_at.exe
21. http://www.y1pwe.co.uk/RAProgs/linux/run_prg_at
22. http://www.y1pwe.co.uk/RAProgs/dos/RUN_RTL.EXE
23. http://www.y1pwe.co.uk/RAProgs/linux/r64_rtl.exe
24. http://www.y1pwe.co.uk/RAProgs/linux/run_rtl
25. http://www.y1pwe.co.uk/RAProgs/dos/RN_RTLAT.EXE
26. http://www.y1pwe.co.uk/RAProgs/linux/r64_rtl_at.exe
27. http://www.y1pwe.co.uk/RAProgs/linux/run_rtl_at

Appendix

1. RF Gain Variation Tolerance Analysis

Two gain varying processes can be considered. The first involves gain variation during the measurement dwell period, and the second, gain variation between measurement periods.

Case 1 Single channel switched between the antenna and a reference load[3,4].

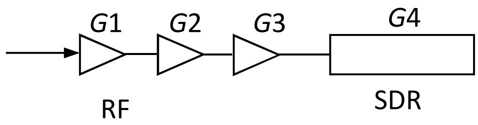


Figure A1 Single Channel Receiver chain

The power gain variation with time can be assumed to be a general function of the form,

$$G(t) = (G1G2G3G4).$$

When switched to the antenna, the output data is proportional to, $(T_{ant} + T_{sys})G(t)$, where, T_{ant} is the antenna + sky temperature, T_{sys} is the receiver system noise temperature and T_{ref} is the reference load temperature.

Along the data record, blocks 'n' spectra are derived using the digital Fast Fourier algorithm and averaged.

The antenna data average can be written, $\overline{G_{nA}(t)}(T_{ant} + T_{sys})$

Switched to the reference load, the reference load data average is, $\overline{G_{nR}(t)}(T_{ref} + T_{sys})$

The processed ratio[5] is,

$$R_{proc} = \frac{\overline{G_{nA}(t)}}{\overline{G_{nR}(t)}} \cdot \frac{T_{ant} + T_{sys}}{T_{ref} + T_{sys}} = \left(1 + \frac{\delta G}{\overline{G_{nR}(t)}}\right) . R_{true}$$

where δG is the mean gain change between antenna and reference measurements ($= \overline{G_{nR}} - \overline{G_{nA}}$).

With a knowledge of the reference load and system temperatures, this data ratio allows estimation of the sky noise or target temperature.

For four cascaded amplifiers, the gain variation is approximately, $\frac{\delta G}{\overline{G_n(t)}} \approx \frac{\delta G_1}{G_1} + \frac{\delta G_2}{G_2} + \frac{\delta G_3}{G_3} + \frac{\delta G_4}{G_4}$

Showing a proportional measurement error dependant on the sum of the gain variations of each gain stage.

Case 2 Twin Channels with post-preamplifier switching

To assess the effect of gain changes between switch states, we rewrite the ratio equations, identifying the switched gains, ie,

$$R1 = \frac{Y_{Sw}}{Y} \approx \frac{a^2}{b^2}\frac{(g1_s + g5_s)^2 g6_s{}^2 g7_s{}^2 g8_s{}^2}{(g1+g5)^2 g6^2 g7^2 g8^2}$$

$$R2 = \frac{X}{X_{Sw}} \approx \frac{a^2}{b^2}\frac{(g1+g5)^2 g2^2 g3^2 g4^2}{(g1_s + g5_s)^2 g2_s{}^2 g_s3^2 g4_s{}^2}$$

Replacing $g2_s{}^2$ by $\overline{G2}\left(1+\frac{\delta G2}{\overline{G2}}\right)$ and similarly for the other switched position gain terms and substituting for these, the wanted ratio R becomes,

$$R = \sqrt{R1R2} \approx \frac{a^2}{b^2}\left(1+\frac{\delta G6}{2\overline{G6}}+\frac{\delta G7}{2\overline{G7}}+\frac{\delta G8}{2\overline{G8}}-\frac{\delta G2}{2\overline{G2}}-\frac{\delta G3}{2\overline{G3}}-\frac{\delta G4}{2\overline{G4}}\right)$$

G1 and G5 (power gains) are not represented as these cancel perfectly on multiplication. The variance terms are halved due to the square root function. With temperature consistent mean gain fluctuations, it is feasible that some gain variation can now cancel leading to a more stable value of the ratio *R*.

It is possible to incorporate the amplifiers g_2, g_3, and g_6, g_7 within the preamplifier couplers if size and weight allow. This has the advantage that temperature gain drift of these amplifiers is now cancelled and only the SDR error terms of amplifiers g_4 and g_8 remain.

2. Correlation Receiver RF Analysis 90° 3dB Couplers

Figure A1 shows the correlation receiver with 90° 3dB couplers and the corresponding stage signal voltages. The complex maths symbol 'j' represents 90° phase shift and 'j^2' 180° or signal negation (since, j^2= -1).

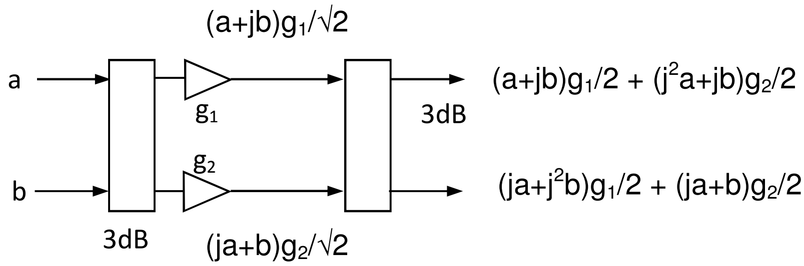


Figure A2 Correlation Receiver using 90° 3dB Couplers

The output signal components simplify to $a(g_1-g_2)/2 + jb(g_1+g_2)/2$, and $ja(g_1+g_2)/2 + b(g_2-g_1)/2$.

This shows that the signal 'a' appears at the lower output port amplified by the sum of the two amplifier gains with a 90° phase shift and the 'b' signal exits the upper output port similarly phase shifted.

As for the 180° coupler version analysed in the main body, there are similar error components should the amplifier gains be unequal.

2. Software Tools

For MS Windows/DOS – 32bit, (64bit) and Linux versions in [] brackets.

Program	Function	Command Line (Win32)
AMP_STS.EXE[7] (amp_sts64.exe[8]) [amp_stats[9]]	Takes rtl_sdr.exe[6] binary file and outputs a text file of the SDR ADC data distribution.	AMP_STS <infile> <outfile>
BIN_TXT.EXE[10] (bin_txt64.exe[11]) [bin_txt[12]]	Takes rtl_sdr.exe[6] binary file and outputs a text file of the SDR ADC data.	BIN_TXT <infile> <outfile> <number of 8192 sample data blocks>
RAFFT.EXE[13] (r64fft.exe[14]) [rafft[15]]	Takes rtl_sdr.exe[6] binary file applies the FFT algorithm to data blocks, and averages these over the input data length and outputs a text file	RAFFT <infile> <outfile> <Number of FFT points>
RAFFT2.EXE[16] r64fft2.exe[17]) [rafft2[18]]	As above but data offset to reduce the zero frequency spike.	RAFFT2 <infile> <outfile> <Number of FFT points>
RN_PRGAT.EXE[19] (r64_prg_at.exe[20]) [run_prg_at[21]]	Uses PC clock to run any program at a set time, on one or more PCs simultaneously.	RN_PRGAT <"prog.exe"><hr min sec>
RUN_RTL.EXE[22] (r64_rtl.exe[23]) [run_rtl[24]]	Runs the rtl_sdr[6] program and generates a bin file which is then processed with the FFT algorithm to output spectrum averaged text file.	RUN_RTL <"rtl_sdr data.bin + command line">< Number of FFT points>
RN_RTLAT.EXE[25] (r64_rtl_at.exe[26]) [run_rtl_at[27]]	Uses PC clock to run the rtl_sdr[6] program at a set time, generates a bin file which is then processed with the FFT algorithm to output spectrum averaged text file.	RN_RTLAT <"rtl_sdr data.bin + command line">< Number of FFT points><hr min sec>

Example 1: Running the MS Windows 32 Tools

1. Copy 'command.com' from 'Windows/system32' directory, RAFFT2.exe from the link – Reference (16) and your Osmocom 'rtl_sdr' recorded .bin files to your working directory.
2. Open 'command.com', change the directory to your working directory if required and type in:-
   RAFFT2 <infile> <outfile> 256
   where, 'infile' is your recorded .bin file 'outfile' is any name you choose and could end in .txt as it is a text file. The final command line entry '256' is the number of FFT points you choose. This must be a power of 2.
3. Open it in 'notepad', 'select all', 'copy' and 'paste' in 'Excel' and follow Excel instructions to produce a graphic display.

Correcting for Ground and Background Noise

By Brian Coleman, Gordon Dennis

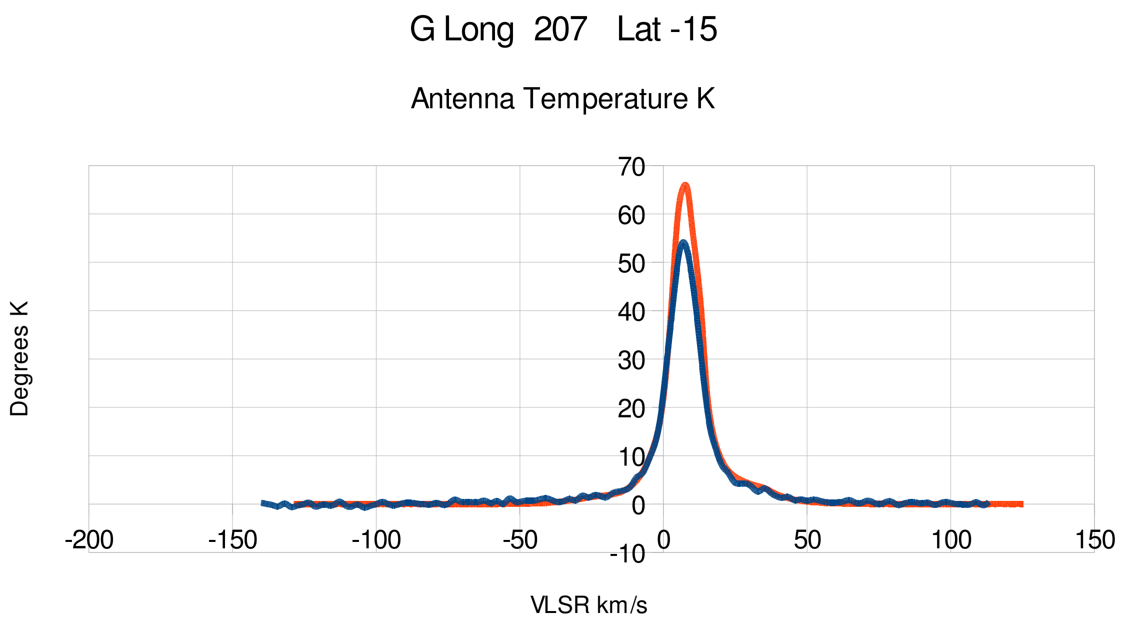
For the HLOG mapping project we have been making observations of Hydrogen Spectra along the galactic equator in 2 degree steps of longitude, and a few degrees above and below the equator also in 2 degree steps, which is usually enough to capture more distant / non local structure we are interested in. So far we have made our observations from Longitude 174 to 30 but getting closer to the Galactic centre is difficult and ultimately impossible from our location in the Northern Hemisphere (51.25 degrees N).

We have a high horizon all around the observatory, due to trees and other obstructions, so at the low elevation required to observe longitudes of less than 30, ground noise becomes a problem. For our mapping project we do not need great precision in our brightness measurements (Antenna temperature in degrees Kelvin) but as the ground noise begins to swamp the target of the observations we run the risk of missing useful structure.

Another source of error, as we approach the galactic centre, is the higher background temperature across the whole of the spectrum that we are interested in.

Recently I have started to use the background data provided by RadioEyes. By selecting the 1.4GHz Bonn data, and pointing to the required celestial coordinates on the map, the L band background temperature is displayed. This can then be added to T Sys to provide improved accuracy for our antenna temperature calculations. An allowance for Ground noise can be added in the same way.

To test the efficacy of this approach I made some observations of Calibration Point S 8 (l = 207 b -15) whilst this was relatively close to the horizon. Normally I try to make observations no closer than 15 degrees to the nearest obstruction / horizon to minimize ground noise but in this case the clearance was rather less.



February 2015

Fig 1

Fig 1 Shows our data plotted in Blue and the LAB data corrected for a telescope with the same beam width in Red. But here no corrections have been applied.

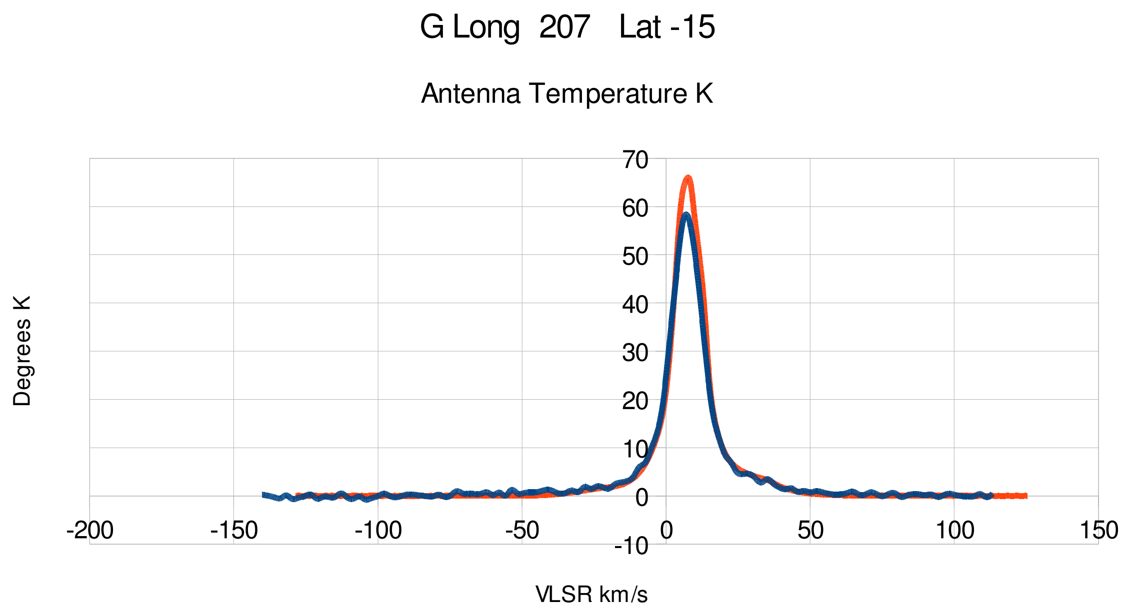


Fig 2

Fig 2 shows the same plot with the background temperature of 3.6K added to T Sys.

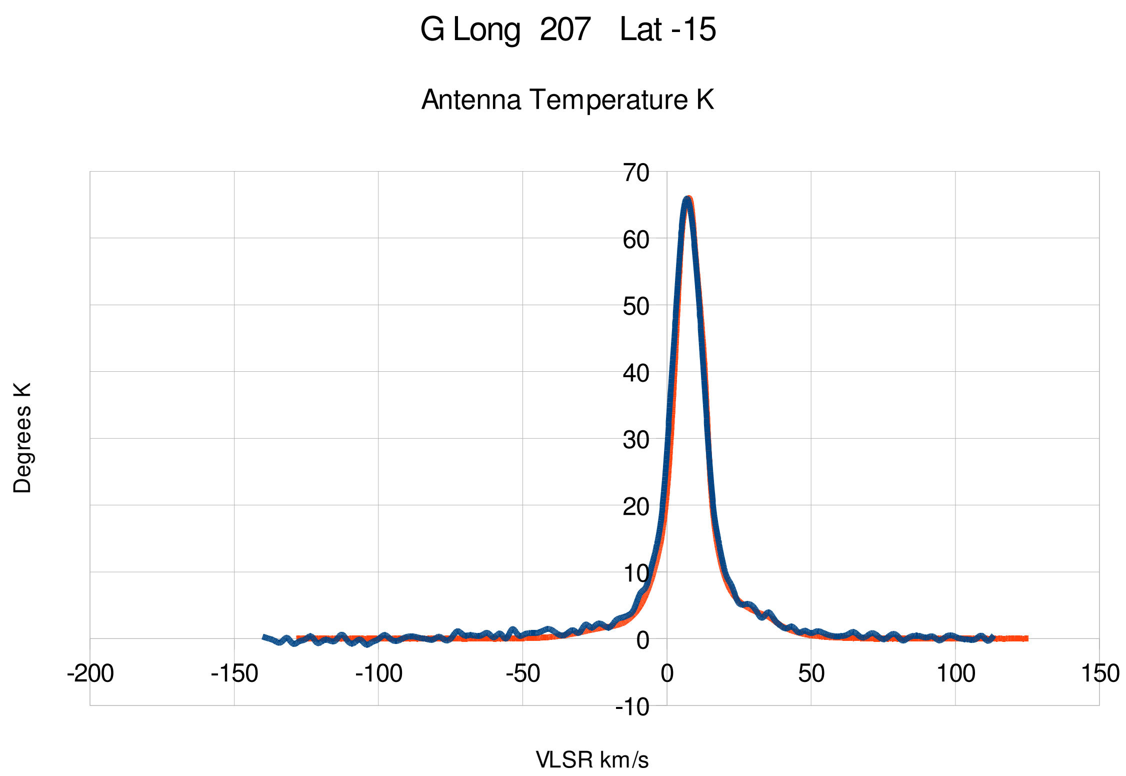


Fig 3

Fig 3 shows the same data with the background correction and an allowance of 6.4K added for ground noise resulting from the proximity of this observation to the local horizon.

Applying these kinds of corrections we have, so far, extended our observations to longitude 24 with reasonable results, see fig 4.

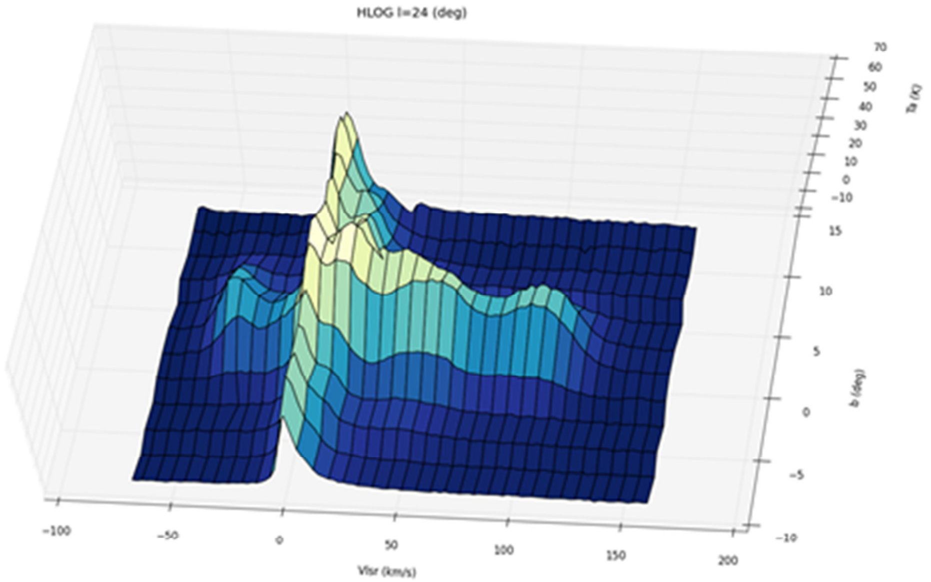


Fig 4

Although far from perfect, when plotted in 3d the structure is reasonably clear and can be used in our mapping project. The peaks at close to zero VLSR, which extend beyond the observed latitudes, are from "local" hydrogen mostly within 5Kly, which we are not attempting to map at this stage.

If there is anyone out there with some artistic or graphic skills who would like to use our data to create an impression of the Galaxy, I would be delighted to hear from them.

Thanks to group member Dave Hardwick we now have some excellent tools which can plot the rotation curve for the galaxy and use the rotation data to help identify which peak belongs to which part of the galaxy. The observed rotation curve differs from the predicted motion which gives us some insight into the discovery of unexplained mass in the galaxy, dark matter.

EAARO UPDATE - FEBRUARY 2015

By Jason Williams

As the Managing Director of the East Anglian Amateur Radio Observatory I would like to personally welcome you to our first news update. I hope this will become a regular feature of your Ragazine and I and my team look forward to keeping you up to date on our latest news and projects in progress.

Firstly, following a recent recruitment drive we are pleased to announce some new members to the EAARO team who have offered their kind support. Dr Mark Galloway, Observatory Manager at Bayfordbury Observatory (University of Hertfordshire) and Dr Megan Argo from Jodrell Bank Observatory (University of Manchester) are both very keen to contribute their knowledge and skills.

Recently I was interviewed by BBC Radio Orkney about our plans for construction of our Meteor Radar Project due to be based in Orkney, North Scotland. EAARO is successfully building relationships with Orkney Council, Orkney Tourist Information Centre, and key people from the local community as we discuss our plans to bring space science and discovery to the Northern Isles. This has generated much interest and excitement so it appears that we will have a warm welcome when we visit Orkney later in the year with assistance from Matt Baumer (freelance videographer), Mark Worrall (freelance presenter & researcher) and David Bodenham (freelance producer) who collectively will form our media production division and assist us in producing our first video documentary as we survey some of the sites we have chosen for our radar outstations.

There have been some changes to our management structure this year and we are delighted to announce that Jeff Lashley has been appointed on to the board as our technical director.

Plans are afoot in 2015 to find accommodation for our Space Operation Centre. This will act as the central hub for all of EAARO's scientific operations and will combine our Command & Control Centre with Administration, and Meeting facilities. A number of meetings have already taken place with our local college and we are hoping to reach a mutually beneficial agreement to secure some campus space in return for our contribution to their existing STEM (Science, Technology, Engineering, and Mathematics) facilities. Space allowing we hope to include a viewing gallery so students and members of the public can visit, see how we operate, and enjoy the experience. This allows us to better engage with the wider community without the need to arrange visits to our research stations such as our main observatory site in rural Hertfordshire and our North Ronaldsay Station on the Orkneys.

Towards the end of last year we were very pleased to partner with our first secondary school. Students at the Priory School in Hitchin, Hertfordshire will soon be amongst the first to join in our plans to immerse young people directly in our research by assisting us in the antenna design and testing for our Meteor Radar Project. This will be achieved through a number of workshops and field days that we hope will inspire and educate through our passion for Radio Astronomy.

I hope you have found this article interesting and look forward to bringing readers some exciting news in the next edition as EARRO continues to evolve.

Jason B Williams MInstSCE M0YJW 8th February 2015

Introduction to Python

By Jeff Lashley

This is the first of a series of articles introducing programming languages useful for technical programming. Python is not only mature and well supported, it is easy to learn and use. What makes Python powerful is the wide range of extension packages available to do just about anything.

Programming languages can be either compiled, interpreted or somewhere in between. A compiled language such as C is converted by the compiler to a machine code application in order to obtain best speed at runtime. An interpreted language on the other hand is translated by a runtime environment when the user runs the code, so can be considerably slower than a fully compiled language. Although Python is interpreted, all is not that simple. When using object oriented methods Python is implicitly compiled to a bytecode version to improve performance. In addition many of the add on packages used within Python are in fact fully compiled (often written in C) so that Python applications can also be high performers. For example the NumPy package provides Python with a high performance matrix handling system, and since it is compiled code is very efficient indeed.

These articles assume you are using a Linux operating system, such as Ubuntu or Debian based one. I recommend Lubuntu for a PC and Raspbian for the Raspberry Pi. If you only own a Windows PC, Python can be installed on it, though the examples have not been tested under Windows.

You have three options:

1. Install Lubuntu on another PC, even quite old ones will usually work well enough.
2. Buy a Raspberry Pi which are very cheap
3. Or install Oracle Virtual Box, and run Lubuntu as a virtual machine.

Installation of Linux is of course another story beyond the scope of this article. Python comes ready installed in Lubuntu and Rasbian, however there are two versions, Python2 and Python3. Code written for Python2 (currently version 2.7.9) is not necessarily compatible with Python3 (currently version 3.4.2) the syntax does vary. Many scientific programmers still use Python2 even now mainly because many of the best packages are/were slow to migrate to Python3. So for the examples provided here Python 2.7 will be assumed. In order to check your system open a terminal (lxterminal in Lubuntu) and type python then press enter. The Python interpreter will start and a text header will print the version.

While you can use any text editor to write Python programs you get no feedback on the accuracy of your code syntax. That is where an integrated development environment (IDE) comes into its own. The best IDE for Linux (works on Windows too) is eclipse, the latest being eclipse Luna. It does not however come with python support so a plugin called pydev needs to be added. Since eclipse is a complicated high end IDE, it is not best suited to absolute beginners so I will not cover its use any further. Another useful simple tool is geany, but what I really recommend for python coding is spyder. Install it in Lubuntu or Raspbian by:

sudo apt-get –y install spyder

This process adds a lot of other packages suited to scientific programming including numpy for array handling, scipi for data analysis, and matplotlib for graphing. Spyder was designed for scientific programming. It consists of three main windows, a code editor which includes code completion, a console window for viewing output from the code, and a configurable window which can show help and other information.

It is not the intention of these short articles to teach you programming, but it is intended to provide a taster to encourage you to learn more about it. At the end I include a list of tutorials and useful reading. The following example code demonstrates three programming styles, retro 'serial' programming where the program essentially runs from line 1 to the end, with the odd short loop. Functional programming where the code is broken up into separate callable functions (subroutines), making the code a little easier to read and understand. Finally the application is converted to a fully object oriented one. This is the most powerful form of coding, and has more than one benefit. Firstly objects can be readily re-used, adapted and maintained. But more importantly objects, in this case Python classes are more efficient, because on first use the Python interpreter semi compiles them to .pyc bytecode files which run faster.

The example then is an application to download weather data from an internet url. The site is http://openweathermap.org. In order to use their API for free you should register and get an API key for yourself. These examples do not include an API key but they will work for testing at least. In order to keep the data handling simpler, the weather data downloaded in the application is the latest weather report for the site chosen, although with a minor change a 5 day or 16 day forecast can be downloaded. I will leave it to you to explore adapting it to process forecasts.

The serial code example:

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

import json
import urllib
import datetime


celcius = u'\N{DEGREE SIGN}'+'C'
degree = u'\N{DEGREE SIGN}'
url = "http://api.openweathermap.org/data/2.5/weather?q=Melton Mowbray,uk"
debug = False

response = urllib.urlopen(url);
data = json.loads(response.read())

if debug:
    print 'Raw Data'
    print data
    print
    print 'coord      :', data['coord']
    print 'sys        :', data['sys']
    print 'main       :', data['main']
    print 'wind       :', data['wind']
    print 'clouds     :', data['clouds']
    print 'weather    :', data['weather']

dt = str(datetime.datetime.fromtimestamp(int(data['dt'])))
date, time = dt.split(' ')
sys = data['sys']
rising = sys['sunrise']
setting = sys['sunset']
```

```
d,sunrise = str(datetime.datetime.fromtimestamp(rising)).split(' ')
d,sunset = str(datetime.datetime.fromtimestamp(setting)).split(' ')

print
print 'Main data'
main = data['main']
print 'Date           :', date
print 'Time           :', time
print 'Sunrise        :', sunrise
print 'Sunset         :', sunset
print 'Pressure       :', main['pressure'], 'mb'
print 'Temp           :', main['temp']-273.15, celcius
print 'Humidity       :', main['humidity'],'%'
print
cloud_cover = data['clouds']
print 'Cloud Cover    :', cloud_cover['all'], '%'
w = data['weather']
for i in range(0, len(w)):
    x =  w[i]['description']
    print 'Weather        :', x
    wind = data['wind']
print 'Wind m/s       :', wind['speed'], 'm/s'
print 'Wind direction :', wind['deg'], degree
```

The first two lines start with a #, this is a single line comment and are ignored by the Python interpreter, so strictly speaking they are not required but these lines have meaning to the Linux operating system if this file was to be made executable directly.

The first code lines starting with import are used to load extension packages (other programming languages can refer to these as libraries). The weather data is supplied in a special format known as a JSON, we also need the urllib to access the internet, and the datatime package to handle dates and times more easily.

Strings of characters are identified by either single quotes such as ‘ xxx’ or double quotes “xxx” there is no difference but they must be in matched pairs. The next 4 lines define some variables which will be used later, the u before a String tells Python to use Unicode character set. I added the debug variable so that you can examine the raw data more easily. By making debug True, the raw data is printed along with the elements of interest. This was a useful feature while developing.
Note the definition of the url can be easily changed to your location, by changing the town name from Melton Mowbray to your own town. Also to get the forecast data instead of weather report change the part weather? to forecast?

The response variable as the name suggests hold the response from opening the url. The data can be accessed by using the read() function as shown in the next line. The json.loads function takes the JSON object and converts it into a fairly complicated Python data structure made up of a combination of Python lists and Python dictionaries. Lists are designated by square brackets and comma separated elements eg: z = [x,y]. The elements are indexed numerically starting from 0. So that z[0] = x. Dictionaries are bound by braces such as z = {‘element1’: x, ‘element2’: y} and are accessed by z[‘element1’] which results in x.

If statements, are where tests and decisions can be made, the end of the test statement is always a semicolon. Note the indentation of the following lines is very important. It is the block of indented code which is executed if the test condition is true. The code which follows must return to the previous level of indentation. This indentation is used elsewhere in other forms of loop, definitions of functions, class definitions etc. A decent code editor such a Spyder will handle those for you. You just need to remember to un-indent when that block ends.

The rest of the lines are fairly self-explanatory, but note the two lines starting with d, sunrise and d, sunset. The sunrise and sunset elements of the JSON file are supplied in unix time, which is the number of seconds passed since 00:00 UT on 1st January 1970. The datetime package has a function to convert this to a text string containing the date followed by the time, with a space in between. The split(' ') tool then breaks this String into two parts (around the space character) creating a two element Python list. The d, sunrise and d, sunset is a short cut way of accessing those list elements. For example:

```
test = “jeff knows”
j,k = test.split(‘ ‘)
print j,k
```

is equivalent to:

```
test = “jeff knows”
result = test.split(‘ ‘)
print result[0], result[1]
```

The same technique is useful when handling output from functions. Using the Python keyword return followed by more than one variable can return multiple values easily. For example:

```
Def  routine():
      x=1, y=3, z=8
      return x,y,z

j,k,l = routine()
print j,k,l
```

will result in the output

```
1 3 8
```

The Functional Version

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

import json
import urllib
import datetime
celcius = u'\N{DEGREE SIGN}'+'C'
degree = u'\N{DEGREE SIGN}'
debug = False

def getWeather():
    url = "http://api.openweathermap.org/data/2.5/weather?q=Melton Mowbray,uk"
    response = urllib.urlopen(url);
    data = json.loads(response.read())
```

```
    return data

def unixtimeToUT(unixtime):
    date,time = str(datetime.datetime.fromtimestamp(unixtime)).split(' ')
    return date,time

def decodeWeather(data):
    main = data['main']
    pressure = main['pressure']
    temperature = main['temp']
    humidity = main['humidity']
    date, datatime = unixtimeToUT(data['dt'])
    sys = data['sys']
    rising = sys['sunrise']
    setting = sys['sunset']
    date,sunrise = unixtimeToUT(rising)
    date,sunset = unixtimeToUT(setting)
    w = data['weather']
    cloud_cover = data['clouds']
    for i in range(0, len(w)):
        weather =  w[i]['description']
    wind = data['wind']
    return
date,pressure,temperature,humidity,datatime,sunrise,sunset,cloud_cover,weather,wind

if __name__ == '__main__':
    data = getWeather()

date,pressure,temperature,humidity,datatime,sunrise,sunset,cloud_cover,weather,wind
= decodeWeather(data)
    if debug:
        print 'Raw Data'
        print data
        print
        print 'coord      :', data['coord']
        print 'sys        :', data['sys']
        print 'main       :', data['main']
        print 'wind       :', data['wind']
        print 'clouds     :', data['clouds']
        print 'weather    :', data['weather']

    print
    print 'Weather Now'
    print
    print 'Date           :', date
    print 'Time           :', datatime
    print 'Sunrise        :', sunrise
    print 'Sunset         :', sunset
    print 'Pressure       :', pressure, 'mb'
    print 'Temp           :', temperature-273.15, celcius
    print 'Humidity       :', humidity,'%'
    print 'Cloud Cover    :', cloud_cover['all'], '%'
    print 'Weather        :', weather
    print 'Wind m/s       :', wind['speed'], 'm/s'
    print 'Wind direction :', wind['deg'], degree
```

This code is basically the same as before, except tasks have been split into functions. It makes the code easier to read, and maintain. The def (short for define) here defines the functions, and values are returned at the end, including multiple returns as described above. Functions only have a local scope for variables, they do not have direct access to variables created elsewhere in the application. Any variables that need to be accessed can be passed by value and the variable name appearing inside the brackets of the function name. For example consider this:

```
Def square(x):
```

```
        x = x*x
        print x
        return x

x = 6
y = Square(x)

print x,y
```

If you are not careful you can get really mixed up and make mistakes. Note only the value of x is passed to the square function, the x variable within the function is distinctly different from the x variable outside, and as in this case hold different values. It could be less confusing in some cases to write it like this:

```
Def square(val):
        val = val*val
        print val
        return val

x = 6
y = Square(x)

print x,y
```

The line:

```
if __name__ == '__main__':
```

needs some interpretation. It defines the start point for the application. The code following this line will be executed when the file is run standalone. However if the code is imported into another application as a module, the code after this line will be ignored, but the imported functions will still be accessible.

The object oriented version

This consists of two files, the Class and the application which instantiates the Class. In simple cases such as this the Class could also be declared within a single source code file.

Filename weatherlog_class.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

import json
import urllib
import datetime

class weather():
    def getWeather(self):
        url = "http://api.openweathermap.org/data/2.5/weather?q=Melton Mowbray,uk"
        response = urllib.urlopen(url);
        data = json.loads(response.read())
        return data
    def unixtimeToUT(self,unixtime):
        date,time = str(datetime.datetime.fromtimestamp(unixtime)).split(' ')
```

```
        return date,time

    def decodeWeather(self,data):
        main = data['main']
        pressure = main['pressure']
        temperature = main['temp']
        humidity = main['humidity']
        date, datatime = self.unixtimeToUT(data['dt'])
        sys = data['sys']
        rising = sys['sunrise']
        setting = sys['sunset']
        date,sunrise = self.unixtimeToUT(rising)
        date,sunset = self.unixtimeToUT(setting)
        w = data['weather']
        cloud_cover = data['clouds']['all']
        for i in range(0, len(w)):
            weather =  w[i]['description']
        windspeed = data['wind']['speed']
        winddir = data['wind']['deg']
        d = {'date': date, 'pressure': pressure, 'temperature': temperature,
'humidity': humidity, 'datatime': datatime, 'sunrise': sunrise, 'sunset': sunset,
'cloud_cover': cloud_cover, 'weather': weather,'windspeed': windspeed, 'winddir':
winddir}
        return d
```

As before variable scope must be considered in the functions, but if one function needs to refer to another within the same class, the term self is needed, and should automatically be passed into the member functions.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

import weatherlog_class as w
import json
debug = False
celcius = u'\N{DEGREE SIGN}'+'C'
degree = u'\N{DEGREE SIGN}'


if __name__ == '__main__':
    w_data = w.weather()
    data = w_data.getWeather()
    d = w_data.decodeWeather(data)
    if debug:
        print 'Raw Data'
        print data
        print
        print 'coord      :', data['coord']
        print 'sys        :', data['sys']
        print 'main       :', data['main']
        print 'wind       :', data['wind']
        print 'clouds     :', data['clouds']
        print 'weather    :', data['weather']

    print
    print 'Weather Now'
    print
    print 'Date           :', d['date']
    print 'Time           :', d['datatime']
    print 'Sunrise        :', d['sunrise']
    print 'Sunset         :', d['sunset']
    print 'Pressure       :', d['pressure'], 'mb'
    print 'Temp           :', d['temperature']-273.15, celcius
    print 'Humidity       :', d['humidity'],'%'
```

```
    print 'Cloud Cover    :', d['cloud_cover'], '%'
    print 'Weather        :', d['weather']
    print 'Wind spees     :', d['windspeed'], 'm/s'
    print 'Wind direction :', d['winddir'], degree

    file = open('weather.log', 'a')
    txt = json.dumps(d)
    file.write(txt)
    file.write('\n')
    file.close()
```

The only change here from the previous version is the grouping of the functions into a class, or object saved under the name weatherlog_class.py. This file is imported into the application like any package in this case it is given an alias name of w to simplify accessing later. When importing a python source file, the .py is not typed.

In order to use the class, we need to create an instance of it as we do on the line w_data = w.weather(), and then to access our functions we use the syntax, data = w_data.getWeather() and that's it.

At the end a file is opened, if the file does not exist it will be created, if it does exist, the 'a' means append to its current content. Equally a 'r' would be used to only read a file, and a 'w' to only write to a file. The function json.dumps(d) converts the data structure to a text object for writing to the file.

I do hope this inspires you to learn more

Useful reading:

Websites:
http://www.python-course.eu/course.php
https://docs.python.org/2/tutorial/
https://docs.python.org/2/download.html

Books:
Beginning Python novice to professional by Magnus Lie Hetland, Apress
Python essential reference by David M. Beazley