

74!

THE KNIGHTS QRSS WINTER 2018 COMPENDIUM 1st edition



- *BUILD A RASPBERRY PI GRABBER*
- *KNIGHTLY NEWS*
- *GOUPL ON LIFE, QRP LABS AND THE UNIVERSE*
- *SUPER DAVE'S ALL BAND GRABBER*
- *ZL2IK'S QRSS PRIMER*
- *AND MORE !*

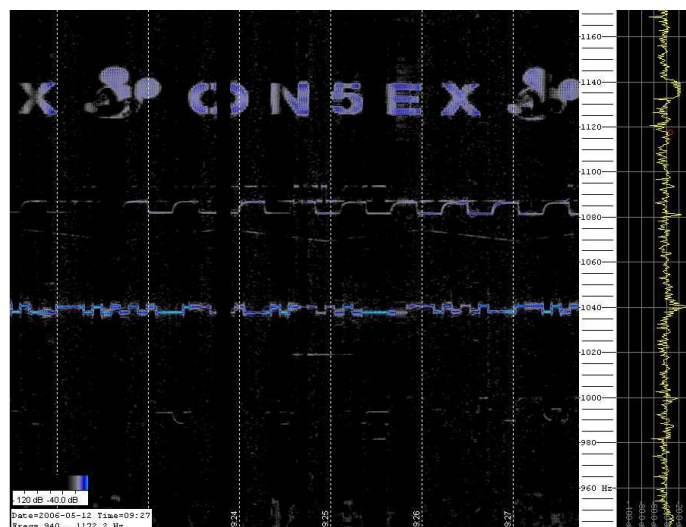


Knightrly News

Welcome to the first attempt at gathering together a collection of QRSS related articles and notes, as a way of marking out the existence of QRSS'ers and the Knights QRSS group. When this idea was born I was not quite sure what it would turn out to be. A QRSS Yearbook would never have enough material, and neither would a regular newsletter. So it is really a mini compendium. And then what could it be called ? I joked to myself that it could be called RST (Really Slow Today), a joke referencing the speed of our transmissions. But as luck would have it, there has been some gentle adoption of the numeric sign off code of 74, meaning "may your signals be bright" (on the grabbers. So now the world of radio has 72 (Good luck with your QRP), 73 (Best wishes), and now 74 for us QRSS'ers.

I am writing this as the year draws to a close in 2018. When I first became interested in QRSS about 6 years ago you were lucky to about 6-12 active grabbers at any one time on the compendiums. Some were only part time. Since I maintain several lists and grabber compendium websites, I see that this has grown to typically 70+ well known and generally reliable grabbers. This part of the hobby is certainly attracting the right kind of people. In a way I am glad that it still remains a niche art of the radio hobby. We find our brethren determined towards technical excellence, reliable transmitters and grabbers, website maintenance and gentlemanly conduct. Just the stuff that many of us looked forward to when we first began our amateur radio voyage. Long may it continue!

This year marks the sad loss of Johan ON5EX. Johan ran the first QRSS Knights mailing list with the old Mailman system. I have no doubt that QRSS would never have grown without that vital link keeping us all together. Thank you Johan.



And finally, as 2018 comes to a close, so does solar cycle 24. Zero sunspot numbers have become the norm and yet we all continue on. May I wish you all Knights group members a happy and healthy new year, and 74, may your traces be bright ! - G0FTD

THE BUILDING OF A QRSS GRABBER

David R. Hassall WA5DJJ

My QRSS life began in about 2007 when I heard Bill N2CQR talking about his little transistor transmitter putting out .01 watts sending a signal from Rome, Italy to Belgium. I marveled how he could do that and I wondered if I could do it too. The die had been cast. I was hooked. I learned to copy QRSS signals for a few days but that wasn't enough. I had to transmit them too. Thus, the journey began.



I went through everything I could read about QRSS. The techniques, the transmitters, the keying techniques and was off and running building transmitters, antennas and a few receivers. My antennas worked well and my transmitters were seen in diverse places in the world and in the early days the propagation was poor.

But I was still getting there. But my attempts to receive QRSS Signals usually involved my expensive Amateur Radio Receiver and my expensive PC huffing and puffing 24/7 to copy a single band. In those days, My computer cost me about \$1200 for a good fast one and my transceiver was close behind at \$900. So, for \$2100, you to could be a grabber station. The reliability of both units in those days running 24/7 was questionable, with the computer being the worse of the two. But it still was in the back of my mind that I needed to build a grabber. (To this day, I don't know why)

In 2012 I tried again but the total cost of building a multiband QRSS Grabber was still north of \$1000 a band and if you considered it being 10 bands that was the cost of a good car or a down payment on a home. Way above my meager means but it was then I started to experiment with all of the Amateur Kit receivers and transceivers to see if any of them would work. Later on the SDR receiver craze started but that required two computers so even though you had a small radio processing using and additional computer actually pushed the price up instead of down. (Most of the SDR receivers I tried were also essentially DEAF to QRSS signals, Others could get them to somewhat work, but I couldn't) So the quest for a good receiver continued. The urge to build an All Band QRSS Grabber increased with each technological advance.

When QRPLabs came out with the U3S, which was stable and transmitted on all bands, which could operate on multiple modes and do all sorts of wonderful things, The urge to build a All Band QRSS Grabber increased to a fever pitch. I could really study the Ionosphere variations and watch what happens to signals on multiple bands during a solar flare and a Geomagnetic storm in real time, if I only had a Multiband QRSS Grabber. I would actually be able to SEE the things that happens to signals on multiple bands as they are happening.

So, I started building every QRP transceiver and simple QRP receiver I could find but all of them has serious problems to stay on frequency and receive a signal reliably on a 24/7 basis. Trying to get 10 or 12 surplus PC's all running reliably 24/7 to process the receive signal audio was also no easy feat. (I kept running out of wall plugs and my home air conditioning system was groaning at the heat load), So that wasn't working out too well.



Then came 2017, Hans Summers built the QCX for the 2017 Youth on the Air conference. I heard about the QCX on the internet and spent 3 months trying to get one because everyone else wanted to do the same thing. I finally got one in July or August built it and tried it out and immediately ordered three more. My goal in 2017 was to have a 4 QRSS Grabber up and running for the QRSS New Year's Eve Party of 2017.

But I also didn't have enough computers to run that many instances of Spectrum Lab (my display choice for QRSS signal spectrum pages). I had seen that a number of the European QRSS grabbers were running something called LOPORA and they were on the internet with it and it's visual presentation wasn't quite as good as Spectrum Lab.

BUT, it could run on a little computer called a Raspberry Pi. Now, you have to know that me and computers have an adversarial confrontation just looking at each other. But the raspberry pi ran on just 5VDC at 320ma and was absolutely QUIET. No 400 Watt PC with noisy fans and rebooting Windows for "who knows what reason" to get it back up and copying the QRSS signals from the QRPLabs QCX. So, I ordered 5 Raspberry Pi model 2 B 's (One for each QCX receiver and a spare) It took about a month of learning how to boot one up and make a SD card that worked and buying the OTHER STUFF that you find out you need to make the little Raspberry Pi work, Like HDMI monitor, blue tooth keyboard and mouse, Powered USB HUBS, and so forth. But in the end, the QCX receivers were ready on 80M, 40M, 30M and 20M with a Raspberry Pi hooked to each one, A website was built on QSL.net, and I was ready to listen for signals by the middle of December, Just in time for the 2017 QRSS New Year's Eve Party.

The QRSS New Year's Eve Party of 2017 was a success even with a few glitches and a very poor thought out layout of the Multiband Grabber. It was almost impossible to work on if something broke without disturbing the whole grabber. But my original goal of 4 bands was accomplished but I had to have more. So work began on correcting the configuration and adding more bands. I want to be able to match the QRSS capability of the U3S. Thanks to PA2OHH Onno's hardwork and dedication, He has rewritten LOPORA to Verson 5 now and LOPORA is a high class QRSS spectrum Alalyzer with stacking, 8 hour and 24 hour displays and runs very well on a Raspberry Pi 3 B. I have QCX receivers built on the QRSS frequencies on 160M, 80M, 40M, 30M, 20M, 17M, 15M, 12M, and 10M bands. Now I should be able to watch the effects of a solar flare and/or a geomagnetic storm on the QRSS Signals on each band. I am presently engaged in that process NOW. My goal is to be on these bands reliably able to copy QRSS signals on a 24/7 basis for the 2018 QRSS New Year's Eve Party.

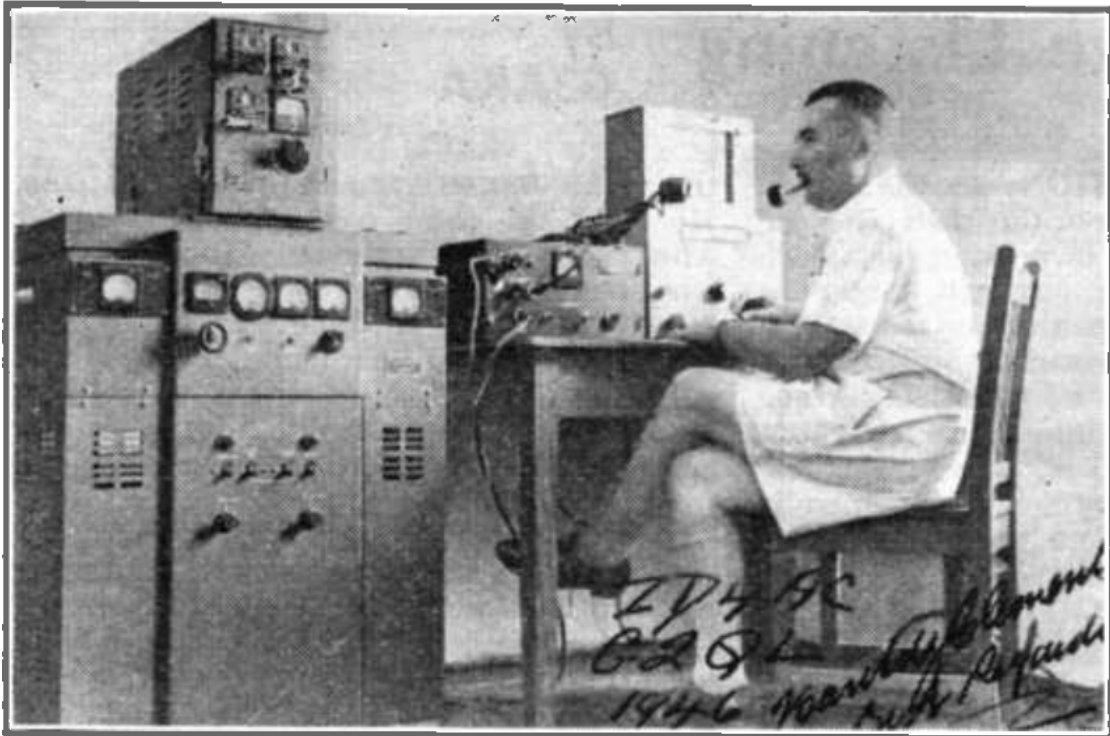
We will only know how that turns out after the 2018 QRSS New Year's Eve party. Hope to see your signal there. I will be listening on all bands with all the skill and sensitivity that I can muster. 73 de Dave.



The lengths people will go to !

It's hard to imagine where we have come from with our hobby. Here's an old photo from the 1940's. You needed a shack just for the radio. And then sent in by an anonymous QRSS'er. The description is of two of the QRP Labs U3 transmitters with modified LCD screens built into a common box.

The 1940's



The year 2018



THE ZL2IK QRSS PRIMER



QRSS - the Q Code word "QRS" - meaning "send slower", the extra "S" means "go to really slow"! This is a mode that has been gaining popularity in Amateur circles worldwide, for the ability of signals using the mode to be detected at extreme ranges, with very good accuracy. The mode has been organised into 200 Hz slots in each Amateur band, so space is tight.

For FSKCW the standard shift used, +4 Hz - allows 1 Hz spacing for the next occupant of the band every 5Hz, so space is even tighter! The lower frequency means "key up", and is the "quoted" frequency. The shifted frequency (+4Hz) means "Key down". Your transmitter should not drift more than +1.0 Hz either way! FSKCW6 is the standard used, Dots are 6 seconds long. The Farnsworth definition for Morse code says, dashes are 3 Dots long, an inter-element space is 1 Dot length, and inter-character spacing is 3 Dot lengths, and an inter-word space is 7 Dots (but not generally used). Another popular mode is DFCW10, each Morse element is assigned a frequency i.e. Dots are the rest (lower) frequency, and Dashes are the shifted frequency, shift remains as before at +4 Hz, Dots and Dashes are the same length - 10 seconds. Inter-element spacing is 2 seconds (no carrier) and inter-character spacing is 6 seconds (no carrier). DFCW10 is a great mode if your call sign has too many Dah's in it!



AN EXAMPLE OF DFCW
FROM (IK2) WVQ

Because things happen slowly on QRSS (some say it's like watching paint dry!) it is common to see propagation effects on Grabber images; the arrival of Coronal Mass Discharges (CME's) from the sun for instance. When the CME material physically collides with the ionosphere, it pushes it downwards at speed, which causes Doppler shifts on received HF signals, sometimes as high as +5 or +6 Hz (at 30 metres, but less on 40 metres), these are plainly visible on QRSS Grabs, and interesting to watch the aftermath, (it is literally "ripples in the ether") as the ionosphere bounces around settling back to normal.

From the Doppler shift and knowing the signal frequency you can easily calculate the speed at which the ionosphere was moving downwards - you may be quite surprised..... A shift of +5Hz at 10MHz corresponds to the ionosphere moving at 150m/s or 540km/hr downwards.

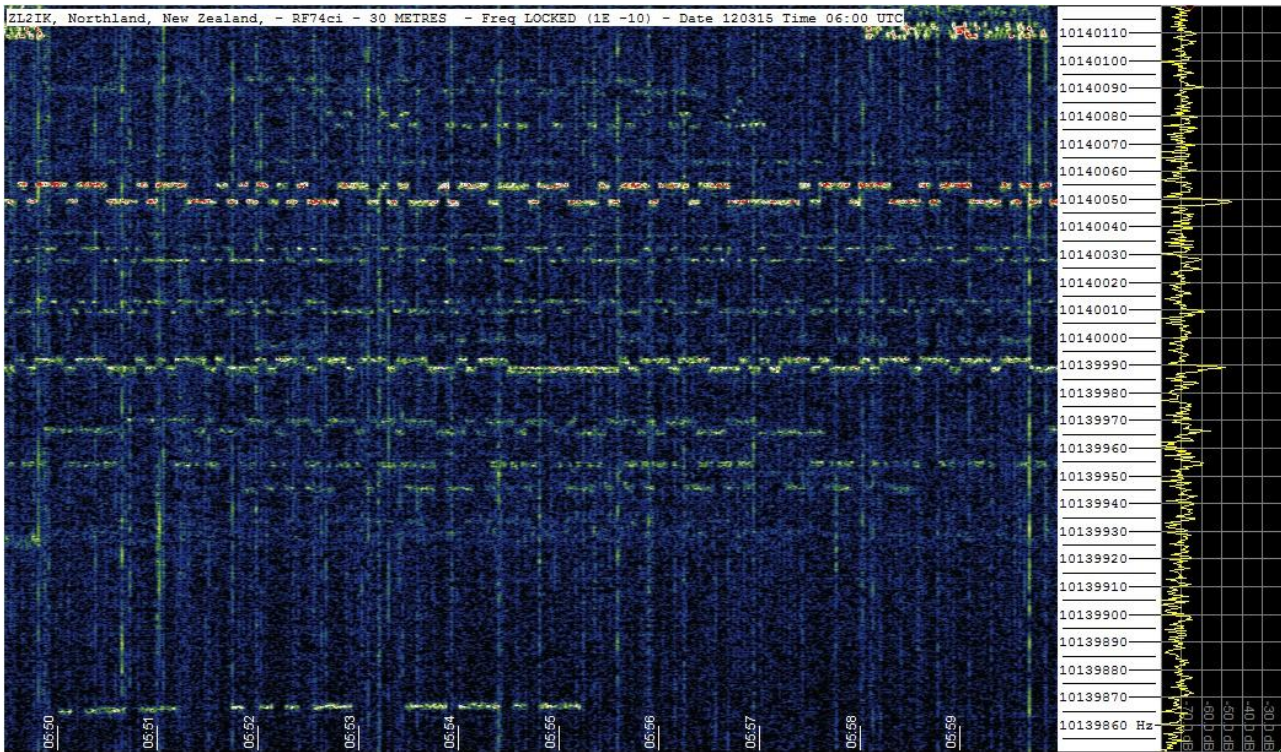
AN EXAMPLE OF FSKCW FROM G6NHU



Like all Amateur exchanges QRSS requires a transmitter, and a receiver. Transmitters are called MEPT's or Beacons. "Manned Experimental Propagation Transmitters", because in some countries (not in New Zealand however), it is a requirement that when the Beacon is on air the Station must be manned. It is common to have a 200mW QRSS beacon, on 30 metres, for instance, to be clearly identified at ranges exceeding 19,550 km's or 12,150 miles. This is the distance from ZL to Portugal - both ways - short and long path, although the short path can be tricky sometimes because of solar activity affecting the Northern Auroral Zone. The Receiving, or "Grabber" end is a little more complex, and requires a frequency stable Receiver, with reasonable selectivity, most modern Amateur HF rigs would cover this either in CW Mode, or even SSB mode, as long as it's IF filters have reasonably steep sides to keep adjacent channel QRM out of the receiver. The heart of a Grabber Station is a Computer, usually a PC, running audio spectrum analyser software. This looks at the spectrum of the narrow band audio coming from the receiver, via its sound card.

The Spectrum Analyser processes the audio using an “FFT” or Fast Fourier Transform method, to process the received audio into a Frequency versus Time graph, with colour and intensity changes, denoting the detected signal, as a horizontal “Waterfall”. The resultant detected signals look like this below:-

A TYPICAL WATERFALL IMAGE



There are a couple of ways that a Grabber Station can display results, one is to post his/her current result, every 10 minutes on a web-site, and post the accumulated results to a DropBox or similar site, for Transmitting Operators to refer to their longer term results. Some Grabbers post 2 pictures on the web, one taken at 10 minute intervals, and a second accumulated result which updates every 30 minutes but shows the last 4 to 8 hours on a single screen, these Grabs are great for assessing propagation and signal drift trends, over time. On my web site I post a 6 hour picture which is updated every 30 minutes.

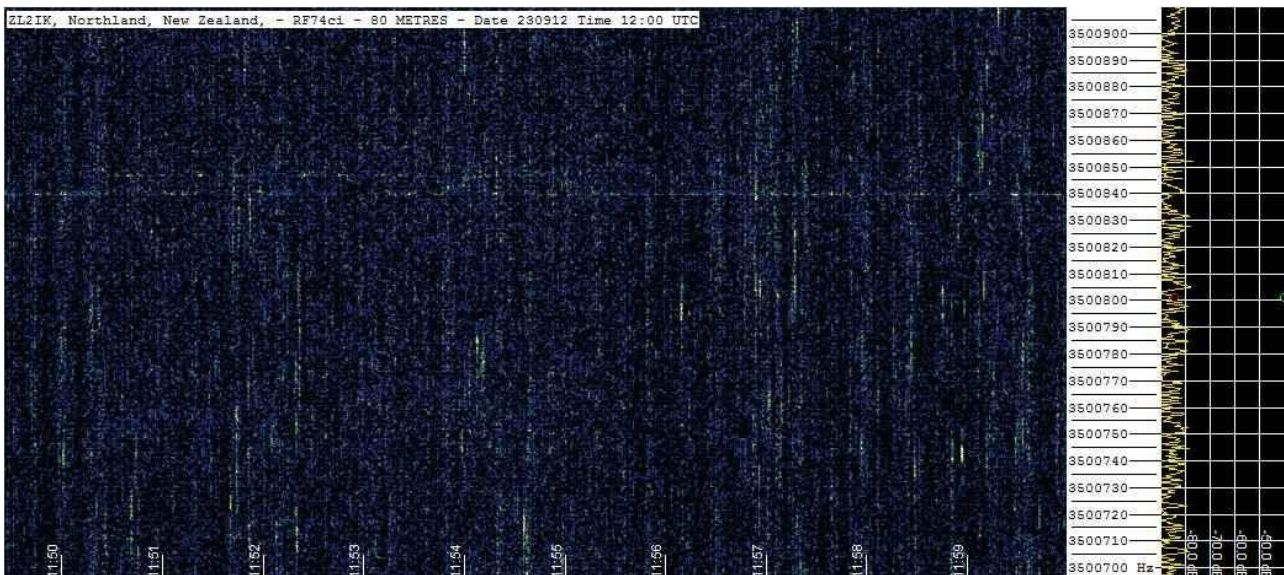
Stacking:

If one refers to the picture above, you will note that 4 stations have all started (and completed their call) in that frame, the rest seem to run rather randomly across the frame, and it is difficult to see where the call sign starts and ends. The ones that start and finish in the same frame are synchronised to the Grabber frame rate (10 minutes), this means that the call will appear in exactly the same place in every frame the station is received in. For international convenience all frames are started at exactly 10 minute intervals past the hour UTC. This allows a further enhancement of the received results, via an extra integration of the signals.

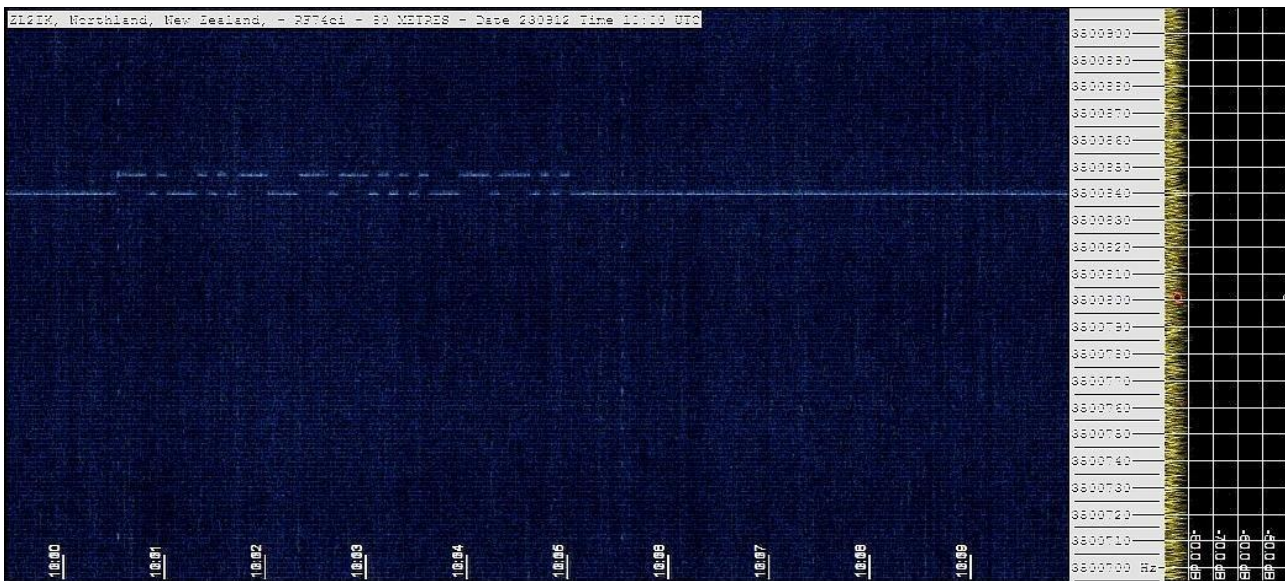
This is done by using a trick from Astrophotography, a tool which overlays pictures on each other and by adding the picture contents together, they build a new frame from integrating and averaging the accumulated data. The results are truly amazing! Noise, because it is random in nature and QRM (as long as it is not synchronised to 10 minutes) is magically cancelled out, and signals, because they are not random and in sync, add up and are enhanced! This is called Stacking.

(See the results below:) Note: The raw signal in a single frame is completely buried in the noise, and there is no real way of identifying the signal from an individual image, however integrating the data from 20 images and averaging the data produces the lower result..... There are a number of free picture processing programs that are capable of Stacking, however undoubtedly the best are from Astrophotography sources, I use Registax V6 2 **Below is NU7Z - the integration and averaging of 20 similar raw images taken over 3 hours and 20 minutes**

RAW UNSTACKED SINGLE IMAGE



A STACK OF IMAGES BRINGS OUT THE SIGNALS



If you are like me, transmitting as well as receiving, it would be nice to make stacks of my own signals from other Grabbers around the globe, and thanks to the internet and a nice little app you can do just that.

The app - "SeqDownload", enables you to pick a number of Grabber sites, and knowing a bit about the site, you can download their image - every time the image changes! The App allows you to do this with multiple sites, and to organise downloads effectively. The downloaded results are then available for you to make up your own stacks of your signals from Grabbers far and wide. There is a later development, from Scott Harden AJ4VD, who provides a website showing all the available Grabbers worldwide in real time.

Scott presents a Stack of the 6 previous images from each Grabber's posts to the internet; his web site has become a 'One Stop Shop' for Amateurs transmitting QRSS. However like always it depends on the availability of Grabbers on the band that you are transmitting on at the time. Fortunately the QRSS Group has an Email Forum in Groups.io - the "QRSS Knights", and this is the clearing house for cooperation between Transmitters and Receivers, and a place to chat and exchange ideas, the Knights are a friendly bunch.

To get on the air quickly, you will need a transmitter; most people these days build their own from a range of kits that are available on the internet. Principally they are available from QRP-Labs.com (U3s Kits and its derivatives).

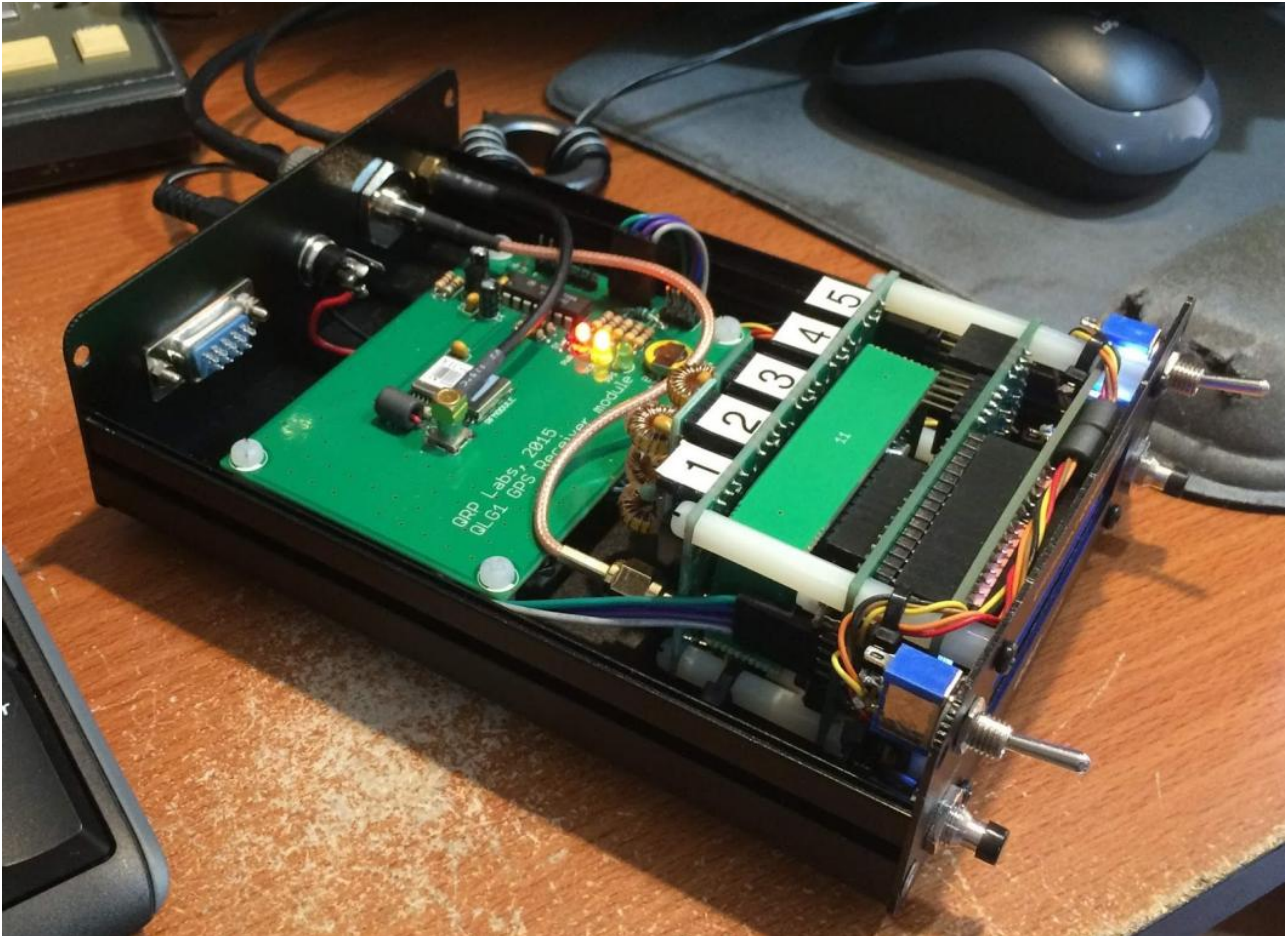
These kits are remarkably cheap and easy to build, most of the hard work is already done for you (mounted and pre-soldered SMD circuitry), and the rest is just following the comprehensive instructions until you complete the build. You can buy as much as you need or the whole caboodle, including a very nice extruded aluminium case with silk screened front and rear panel.

FRONT PANEL OF A QRP LABS QRSS TRANSMITTER



The kit comes with a GPS (for timing and frequency management) as well as a set of switchable low pass filters for band switching (all controlled on the fly by the built in micro-controller), it will cover the whole spectrum from 2200 metres to 2 metres. Output power as built is around 300mW, but you can increase that very simply by running the PA on a slightly higher Voltage. I get 1.0 Watt out of mine by running the PA on 9 Volts. If you purchase the kit and use the GPS in the Aluminium case you will also need an external Hockey Puck type external antenna for the GPS, the little square Ceramic antenna supplied will not work inside a metal case! The whole unit currently costs, all up, \$US134.00, for the complete U3s kit, with Si5351 Synthesiser Board, 6 band relay switched filter board, GPS Board, and 6 Low pass filter kits for the bands you want to operate on. The 3.3V Hockey Puck external antennas for the GPS are available in New Zealand from Nice Gear 6 , in Timaru, for \$18.00 each including GST.

INSIDE VIEW OF ZL2IK U3 TRANSMITTER



You can buy the OCXO module as an extra to the kit, this unit when properly setup has excellent frequency stability, and very seldom needs to be disciplined by the GPS. It plugs into the same socket as the standard Si5351 Synthesizer Board.

The U3s is quite versatile, it can run FSKCW, DFCW, QRSS (slow CW), Hellschreiber (slow and DX modes), CW (at ordinary speeds), CW ID, FSK (wideband FSKCW at normal speeds), WSPR, WSPR-15, Opera Modes, PI4, and JT9. For WSPR and JT9 modes the GPS is essential, as accurate time

keeping using these modes (within 1 second of UTC) is a must for success.

For those interested in setting up a grabber station, and there is plenty of room for one! I have the only full time HF Grabber Station running in the Southwest Pacific, and as far as I know the only full time station in the Southern Hemisphere. There are a few in Australia but they are definitely not full time stations and spend a lot of their time on LF rather than HF as I do.

You will need a receiver (of course) and some antennas for the bands of interest to you. You will also need a PC (with at least a stereo sound card), and some spare hard drive space. A web-site would be handy and you can get some space using your very own domain name, from Spark (NZ) Ltd. They web host for a very minimum monthly fee. Or simply use Dropbox or Onedrive for your web site. You will need some FFT software to convert your audio to Spectrum Grabs. I use SpectrumLab by Wolfgang Buscher DL4YHF, it's free.

It will run in Linux or WinXP, through to Win10. The program is available for download at <http://www.qsl.net/dl4yhf/spectra1.html> . The latest version is V2.90 b02. SpectrumLab can initially have a steep learning curve, but it is an excellent program for QRSS purposes. There are very extensive help files available for SpecLab at the same site. **For safety's sake it is best to run the audio from your receiver via a 600Ω:600Ω line transformer to isolate the computer sound card from the vagaries of the outside world!**

SpecLab comes with configurable DSP narrow band filters, Notch filters, Noise Blankers, Limiters, and even its own AGC system, to name just a few. Remember SpecLab is an Audio Spectrum Analyser; it looks at the audio output of your receiver, not the Signal directly. Signals are generally so weak you cannot hear them!

But you will see them amongst the noise on SpecLab's waterfall. You will need .usr files to get it up and running, I can provide those for you on request at peter@zl2ik.com if need be. These are plain text configuration files and automatically generated by SpecLab.

My .usr files are band dependent in that they show the received RF Frequency down the right hand side of the screen - but of course the actual input frequencies are audio from the receiver! They produce a Computer on screen presence that is 1247 x 699 pixels which will produce a screenshot of exactly 1100 x 640 pixels which is a good size for displaying on the World Wide Web. The screen shot only shows the

Waterfall and the right hand window showing frequency and amplitude, see the screen shots earlier in this article.

The screen shot does not show program controls seen on the computer monitor. SpectrumLab has the ability to be used as a standalone SDR on ELF/VLF, using the Computer Sound Card Microphone Input as the antenna port. It will quite happily receive signals from the lowest frequency your Sound Card can usefully “hear” (usually around 5 Hz or so), up to the highest frequency it can “hear” – on modern PC’s this is usually 96 kHz. This means with a half reasonable ELF antenna you can listen to the Schumann Resonances – now that’s another article in its own right! Don’t know what the Schumann Resonances are? Then – Wikipedia is your friend!

WSPR – pronounced “whisper”: WSPR is a low power mode invented by Professor Joe Taylor W1JT, who has over the years given the Amateur community a number of other narrow band, methods of communication, however by far the most popular one is WSPR, and in particular WSPR-15.

This sends a number of discrete signals at 1 second intervals which are slightly shifted from their neighbours, and each tone represents a symbol in the WSPR Code. A WSPR “frame” is 1 minute and 50 seconds long, and always starts on an even minute UTC, exactly.

There are 10 seconds left at the end of the Frame for the receiver to decode the received Frame and present it to firstly, the person receiving the signal (as a print out on a computer), and then send the frame to the wspnrt.org web site, which gives a visual representation on a map of the Transmitter location, the route the signal takes and the Receiver location.

The two QTH’s are joined by a line showing the great circle path between the two QTH’s. As you can imagine the Map gets rather “busy” at times! However wspnrt.org has built in filters you can use to just filter out your signals (with all the Stations that have received you over a selected time frame). There are other filters and a Database you can query to list all your activity over a selectable time and date.

The beauty of WSPR, unlike QRSS, is that you get instant and very visual (and textual) feedback on the results of your endeavours on any given band, at any given time.

To find out how well your signals are getting into a particular station you need to use the database, and a filter to download the receiver’s uploaded results from your frames. Like QRSS, WSPR is a weak signal

mode, (Transmit Powers rarely exceed 2 watts) and W1JT's decoding algorithm can decode signals that have S/N ratios down to -29 dB!

Quite remarkable, when you consider that a good human ear requires a S/N Ratio of about +5dB to decipher intelligible speech!

If you need to know more please contact me at my email address, I'm only too happy to answer your questions regarding these modes, and methods. I hope I have awakened a few keen budding aficionados.

73 DE

Peter Mulhare
ZL2iK/ZL4Ei
peter@zl2ik.com



SOME USEFUL WEB RESOURCES

QRSS Knights group - <https://groups.io/g/qrssknights>

SEQDOWNLOADER - <https://qsl.net/g0ftd/other/>

SEQDOWNLOADER - <http://www.nirsoft.net/utis/seqdownload.html>

Nice Gear - <https://nicegear.co.nz/>

Registax V6 <http://www.astronomie.be/registax/download.html>

QRSS Grabber Compendium - <http://www.swharden.com/qrss/plus/>

QRSS and stuff - <https://sites.google.com/site/andyg0ftd/test-page>

Animated QRSS Compendium - <http://yak.net:7899/index0.html>



I have been involved with Weak Signal modes such as QRSS for 15 years. There are not many people who have been continuously involved in this fascinating niche of amateur radio for 15 years. This article tells the story of my initial interest in the early days of QRSS experimenting, the various homebrew projects which I constructed to transmit and receive Weak Signal modes, and how that motivated me to create the first ever kit for QRSS, that evolved into the QRP Labs product line and the quite well-known company that QRP Labs is today. All because of QRSS!

My interest in electronics began more than 40 years ago when I was only around 1m high. Later, my teenage bedroom floor was always covered in dismantled televisions, radios, cassette records. In the early to mid-80's home computers became available and that opened up another fascinating new hobby to me. I finally got my amateur radio license G0UPL in 1994, but did not actually use it for another 8 years, having my first QSO in March 2002. All of my amateur radio equipment has always been homemade or kit built, I have no interest in buying and operating commercial amateur radio equipment. That's just my personal preference.

Around a year later as Winter 2003 arrived, and I don't remember exactly how, I came to know of QRSS. There was a group of primarily European experimenters calling themselves the "QRSS Knights", who got together on a mailing list hosted by a Belgian amateur. For some reason, who knows what, but I expect you know the feeling well too - QRSS and the other weak signal modes really grabbed my interest. Weak Signal mode experiments had of course taken place before 2003, but that's when I came to know of it so that's where I have to start my personal history.

Back in 2003, there were only two ways to send QRSS modes. One was with a computer, generating audio tones that could be sent to an SSB transceiver, and transmitted (with the transceiver turned down to low power output). But the idea of using all that expensive, power-hungry equipment to transmit a handful of milliwatts of power, just never appealed to me. The only other way to transmit QRSS was to build your own equipment. There were no commercial products dedicated to transmitting QRSS, nor any kits you could build.

Often people use the term “QRSS” to refer to the class of Weak Signal modes, but really “QRSS” refers to the simplest on/off keyed slow-Morse mode (see below).

Many people built simple transmitters with a few transistors and a crystal oscillator. Some used microcontrollers programmed to generate their callsign; others just transmitted simple patterns of on/off keyed carrier, or frequency shifted carrier.

Weak Signal advantages and disadvantages

The primary attraction of weak signal work is that by slowing down transmissions, to an extreme, the bandwidth of the transmission is greatly reduced, to under 1Hz in some cases. Modern PCs are used at the receiving station to run freeware Fast Fourier Transform (FFT) software that presents a visual view of a narrow slice of radio spectrum, each column of pixels corresponding to one narrow FFT bucket, usually under 1Hz. The colour of each pixel indicates the amplitude received in that bucket. Time flows horizontally across the screen so that over several minutes, a picture builds up.

The eye/brain combination is very good at spotting patterns, and ignoring gaps, fading and interference. The very narrow bandwidth means that the Signal to Noise Ratio (SNR) of a QRSS transmission is very much higher than conventional modes such as CW (Morse code) and SSB (phone). The high SNR means that very low powers can easily achieve worldwide communication on HF.

Some experimenters enjoyed pushing the power levels lower and lower and seeing just how low they could go, and still span the thousands of 10 thousand kilometres to the other station. Whilst ordinary QRSSers might be operating with around a hundred milliwatts, these hard-core weak weak signal experimenters pushed it down below a milliwatt, even down into the nanowatt regions!

The primary disadvantage of weak signal modes is that by transmitting information so slowly, it takes a very long time to send anything!

Types of Weak Signal modes

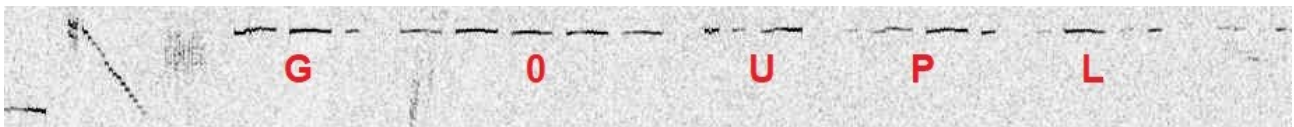
There were two main types of Weak Signal transmission used, in those days, QRSS and DFCW. Quickly FSKCW also became popular. Slow-Hellshreiber was a lot of fun but not so practical for communication purposes. Patterns were very useful for people building extremely simple equipment.

QRSS

This is just an on/off keyed carrier, exactly like CW (Morse Code) except very much slowed down. “QRS” is the standard telemetry Q-code for “Send More Slowly” so QRSS was adopted by the weak signal experimenters, with the extra S signifying EVEN more slowly. A lot more slowly, as it turns out.

A "standard" 12 words per minute CW transmission as a dit length of 0.1 seconds (dah is 0.3 seconds, symbol gap 0.1 seconds, character gap 0.3 seconds, word gap 0.7 seconds according to the standard timing). QRSS is around 100 times slower, with dit speeds of around 3, 5 or 10 seconds. Some experimenters on low frequencies where even higher SNR is desirable to overcome poor antenna efficiency, opt for even slower transmissions with dit speeds of 60 seconds or more.

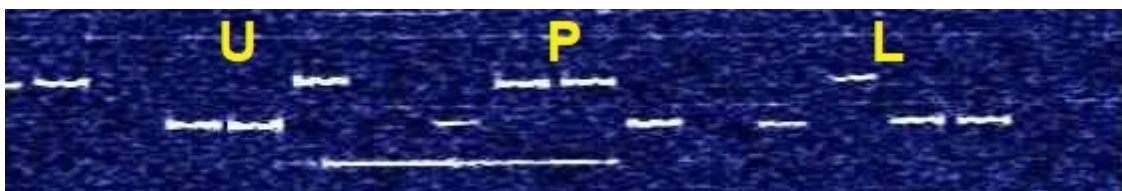
The image below shows the very first reception report I ever received for my QRSS signal, on 28-Mar-2004 at 12:03 UT from Peter DF3LP; the frequency was 10,143,900Hz (30m band). I have annotated this to show how my callsign is represented by the symbols.



DFCW

DFCW means Differential Frequency CW; instead of the 3:1 timing relationship between Morse dahs and dits, in DFCW a dit and a dah are transmitted for the same duration, but the dah is transmitted on a slightly higher frequency. Frequency shifts of a few Hz, say 3-5Hz, were common.

As an example, this is the second reception report I ever received in Weak Signal modes, this one on 28-Mar-2004 at 16:59 UT from Peter DL6NL. My frequency shift was 5Hz and the screen snapshot shows the second half of my callsign GOUPL (annotated below).

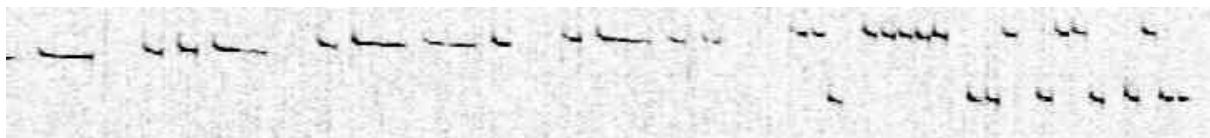


DFCW takes less time to send, because all the symbols are the same duration rather than dah's being 3x the length of a dit in QRSS. Since it is on two frequencies, it also takes more bandwidth; but it is easier to separate from what could be spurious carriers appearing on the band (you can see one such example at the bottom of the above screenshot). I found it a little harder to intuitively decode DFCW compared to QRSS, and often find myself humming the tones quietly in my head, in order to decode the Morse.

FSKCW

People quickly started using FSKCW (Frequency Shift Keyed CW). This is very similar to QRSS, except instead of a key-up (carrier off) period between Morse symbols, the carrier is just shifted to a frequency a few Hz lower.

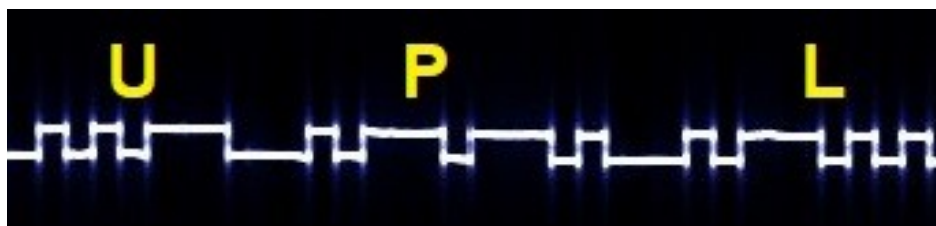
That might seem pointless but in fact it has several advantages. From a practical perspective, when homebrewing very simple equipment with a crystal oscillator driving an amplifier to produce a hundred milliwatts for example, it was found to be difficult to avoid frequency chirp. On key down, for reasons such as power supply drop, slight crystal heating, whatever – you’d often have a frequency drift that was hard to eliminate. Here’s an example of one of my early transmissions, showing “UPL” in QRSS followed by “G0UPL” in DFCW.



You can see that at key-down there is a slight drift down in frequency. It’s a Hz or less. But on these narrowband weak signal modes, that is highly noticeable as you can see. It’s ugly and it can hinder decoding when signals are weak (DX or poor conditions etc).

Keeping the oscillator running all the time was one solution but even then, many people (me included) found it very hard to get rid of this chirp. FSKCW solved this problem completely, because the oscillator and the amplifier were running all the time. Everything was always warmed up and the current draw was constant.

Another advantage of FSKCW was that under very weak signal conditions, one might easily mistake a few whiffs of faint carrier



fading in and out, for a QRSS signal. FSKCW provides a far more definitive signal which can more easily be separated from random interfering carriers.

Slow Hellschreiber

Hellschreiber was a mode developed in Germany before the 2nd World War. It is an early predecessor to Fax, a similar concept. An on/off keyed signal scans lines of an image of text, and transmits them sequentially over a radio link; the receiving station prints these lines onto paper controlled by the incoming signal.

There’s a weak-signals equivalent, which we call slow-Hellschreiber. In this case the transmission carrier is shifted in such a way that the FFT program (as used for other QRSS modes) draws an image of text on the screen. It is both on/off keyed and frequency shifted.

This is my first slow-Hellschreiber transmission, as received by Peter DL6NL in 2004. The “font” used in this example was not the original Hellschreiber font; in fact it was the character font used by the 1982 Sinclair ZX Spectrum home computer!



Slow-Hellschreiber is fun, it is fascinating to see text appear on the screen; but it is even slower to send information than other QRSS modes, and has a worse Signal to Noise Ratio advantage; therefore it requires more power to span a particular communications distance.

Patterns

Sending a unique pattern was popular because it was very easy to do with a simple homebuilt transmitter. A simple transistor multivibrator can produce timing, and you could play with the way the frequency shift looks using resistor and capacitor networks.



This screenshot (right) was my transmission in March 2006 as received by Peter ZL4OL on the opposite side of the planet. The power output at my end was about 300mW to an indoor wire in the attic. The pattern is a simple up/down shift, each cycle took about 25 seconds. The slowing rising edge makes it quite distinctive, like a “signature”.

QRSS Frequencies

In the early days, the 30m band was by far the most popular for QRSS work. The conventional frequency was 10.140MHz, with a narrow QRSS band covering 10,140,000 to 10,140,100Hz. This is where QRSS monitoring stations looked for signals. Sometimes activity was also on other bands. Many Low Frequency distance records were set using Weak Signal techniques.

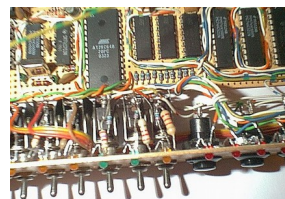
Signal Reports

In the early days of QRSS, people monitored the bands and sent reports by email, or by the QRSS Knights Reflector. It was a very personal and friendly niche of the hobby (and still is).

Later, people set up “grabber” websites to continuously and automatically display on a web page, exactly what their radio receiver and computer was decoding from the QRSS sub-band of the band they were monitoring. This made it easier to see if your signal was being copied on those remote “grabber” stations, without the receiving station operator necessarily having to actively be sorting through all his screengrabs.

Winter 2003: My first QRSS modes transmitter

As I mentioned before, I have been involved with QRSS for 15 years. I built my first weak signal modes transmitter in winter 2003. I was on air with it during Christmas 2003. As I said, I didn't like the idea of a computer involved to generate the audio tones, and anyway I didn't own a SSB transceiver to feed the audio to. What I ended up designing was a very complex beacon controller which had no microcontroller, but did use a lot of Integrated Circuits (ICs, a.k.a. chips) to produce the message encoding.



The message encoding was stored on the 8KByte EEPROM chip (Electrically Erasable Programmable Read Only Memory), so that the message was not lost when the power was removed. A series of counters stepped through the frequency shift commands stored in the memory. The message encoding was entered using a number of toggle switches and push-buttons, with LEDs showing the contents of the current memory byte, and the memory address. Here's a photo of part of the front panel:



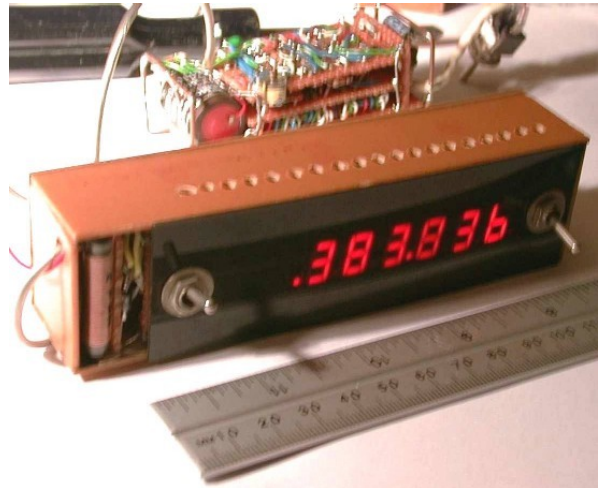
The whole thing was built using point-to-point wiring on a piece of perf-board, installed in an enclosure made from single-sided unetched PCB material. This controller and transmitter had a power output of 200mW and could send any of the QRSS modes (QRSS, DFCW, FSKCW, Patterns, Slow-Hellschreiber) because each symbol sent was programmed into one byte of memory, specifying 16 levels of frequency shift, four possible durations, and whether the transmitter was on or off. A crystal oscillator was used, using the popular 10.140MHz frequency. Frequency shift was applied using a 5mm red LED as varicap (also known as "varactor") diode, to pull the crystal slightly. I had been using LEDs as varicaps since 1994, when I built the frequency counter (see below). I have written a large web page about unofficial varicaps here: <http://hanssummers.com/varicap>. LED "varicaps" became my favourite method of shifting crystal oscillator frequency and featured in many subsequent projects.

Programming this system with the buttons and toggle switches was a huge task! I remember sitting through most of the night on one occasion, taking 6 continuous hours of toggle switching and button pressing, to enter a carefully prepared message which contained a combination of QRSS, DFCW and Slow Hellschreiber modes. Insane!

By the way - you might be wondering about the poor quality of the photos! That was because I was using an old digital camera, with a resolution of only

640 x 480 pixels. For some years, this camera had been used by George G3RJV for editing the G-QRP Club magazine SPRAT, but when the club upgraded to a better camera, he gave it to me! In 2003, mobile phones didn't have cameras. It wasn't like today where your smartphone has a 12 megapixel camera on the back, and other camera on the front. Or even more cameras!

After all that work designing and building such a complex weak signals beacon controller, I was highly disappointed that not one single person heard my beacon, during that Christmas 2003 operating period! It was time for a rethink, and to learn one of the most important lessons about Weak Signal work, which still hits people even today: frequency accuracy! Unless your transmission frequency is accurately calibrated, it will not be in the narrow 100Hz sub-band where QRSS monitoring stations are listening. You'll be transmitting somewhere else and nobody will hear you.

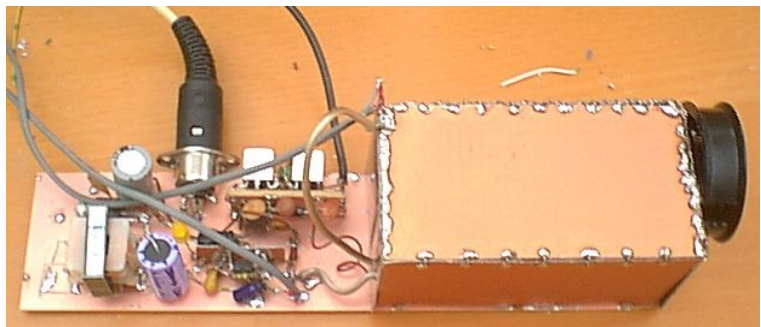


Quartz Crystals make stable oscillators, but the frequency output is never exactly what is written on the can. You might be a few kHz too high, or too low. So, the chances are, that you will miss the QRSS sub-band. Hence it is necessary to adjust the oscillator to get it on the right frequency, by adjusting the capacitance in the crystal circuit. I had used a homemade frequency counter <http://www.hanssummers.com/radiofreqcounter.html> to make this adjustment. Of course, I had forgotten that the timebase for this frequency counter is also a crystal, which would also require adjustment by measuring a known accurate frequency. Which I had NOT done.

Therefore, my QRSS transmissions that Christmas had been outside the QRSS sub-band, and nobody had been listening where I was transmitting. It was time to try to calibrate my frequency counter, then my transmitter. To do this, I had to build yet another project, a 30m Receiver! More info on the beacon controller: <http://hanssummers.com/qrss>

Jan 2004: 30m Receiver

This simple receiver was a direct conversion receiver using an SA602 mixer/oscillator chip, a VFO with a 74LS04 inverter chip (in the shielded compartment shown), and an op-amp as audio amplifier.

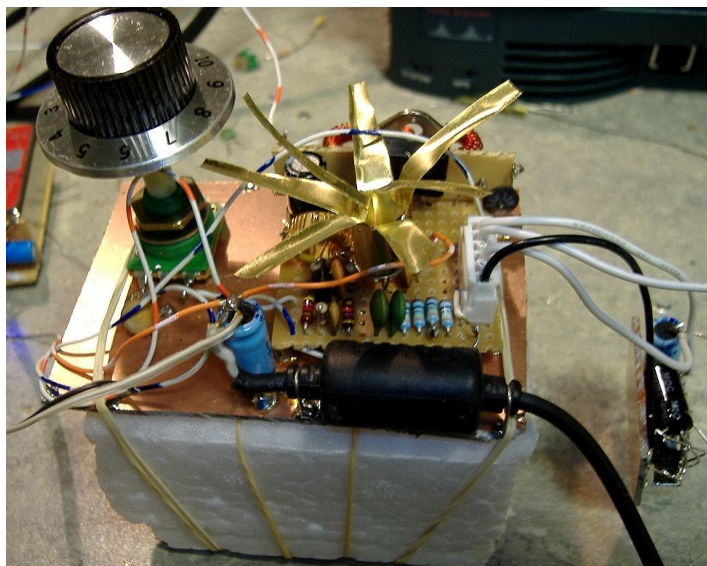


This simple arrangement was enough to receive the Russian time signal station RWM on exactly 9996kHz. By tuning the VFO to zero-beat the Russian station, I

was able to measure the VFO on my frequency counter, and work out the error amount of the frequency counter. I could then go back to my QRSS Beacon transmitter, and adjust its oscillator to get it into the QRSS sub-band.

Once I had performed this critical calibration of my frequency counter, and adjustment of my transmitter, the reception reports started flowing in and it was all great fun. 30m Receiver: <http://www.hanssummers.com/30m.html>

Mar 2006: A simple Junkbox QRSS transmitter

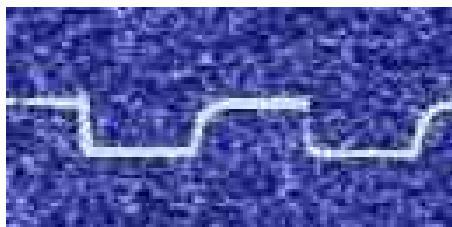


The next project I built was put into operation in March 2006. This was built entirely out of junkbox parts, and just consisted of a transistor crystal oscillator and an amplifier with around 350mW power output; a two-transistor multivibrator generated an up//down pattern shift.

I installed this thing on a spare ceramic floor tile in the attic (not that I was nervous about it catching fire or anything) and powered it from an old laptop computer power supply. The antenna was a wire strung along

the attic, not far from pipes and wires etc.

The simple up/down pattern had a duration of about 25 seconds and the distinctive shape became well recognized among QRSS operators. The beacon remained in uninterrupted operation from February to August 2006. There were reception reports from around the world: Australia, New Zealand, America and many countries around Europe. In total I counted 45 unique stations that copied the beacon; many of whom copied it multiple times of course.



First to report the beacon was David VK6DI on 28-Feb-2006. Some months later, in July 2006, I visited David VK6DI at his home in Perth for several weeks. I recall well, after a 24-hour journey from London, meeting David VK6DI for

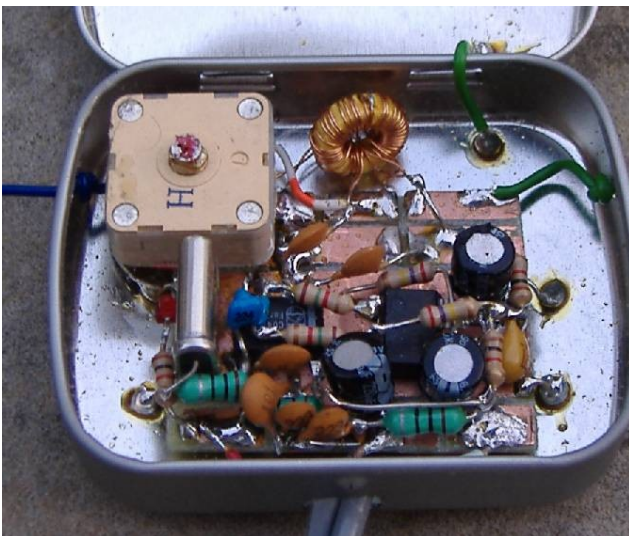
the first time face to face, some time after midnight local time in Western Australia. We drove out to his home in the hills. When we arrived we crept up to his radio room, he switched on his computer monitor and well what did we see? The beacon from my attic, across the other side of the world, booming in loud and clear. It was a truly unforgettable moment! Junkbox beacon: <http://hanssummers.com/qrssjb>

Apr 2006: A 30m QRSS Receiver

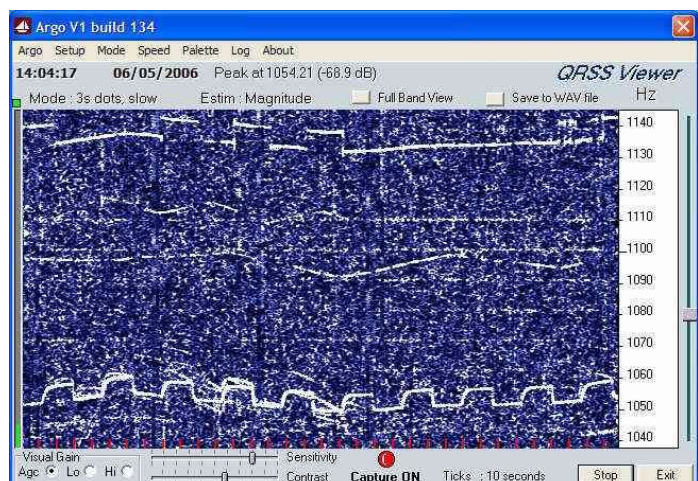
Around April 2006, I wanted to try my hand at receiving QRSS too, not just transmitting as I had to date. I built a simple and tiny direct conversion receiver using a 10.140MHz crystal, SA602 oscillator/mixer chip, and an op-amp. The output could be fed to a laptop soundcard and into one of the spectrum analysis (FFT) software packages such as Argo. It was built into a tiny mint tin – much smaller than the common “Altoids” mint tin that is popular with radio constructors. Two halves of a 30m dipole antenna were connected directly to the receiver, so it could be hung between two trees; with 5V USB power coming from the laptop and audio fed back via shielded cables.

It worked very well. In May 2006, Andrea IW0HK was visiting London, and I met up with him, with my tiny QRSS receiver and my laptop, and we did what we jokingly called a mini-DXpedition, to Hampstead Heath in North London. A nice hill to sit on, looking across London to my junkbox beacon transmitter just South of London.

Below, here’s the receiver in its tin (left), with Andrea IW0HK and I, at Hampstead Heath. I’m on the left. A bit younger than today :-/



Particularly interesting to observe, is the ghost images above and below my beacon transmission. The frequency drift that is evident is just the temperature-induced crystal frequency drift on the receiver quartz crystal, as it hung flapping in the wind between two trees on Hampstead Heath.



Between our reception location, and the transmitter on the other side of London, planes descend across central London to land at the world's busiest international airport, London Heathrow to the West of London. The ghost images are caused by reflections of the signal off the aircraft. As the aircraft approach, the velocity relative to the signal path causes the up-frequency shift. When the aircraft is right between the transmitter and receiver, there is no motion relative to the reflected signal path so the doppler shift is zero. In some of the traces you can see the doppler shift out the other side, as the aircraft starts to travel AWAY (as it goes down to land) and doppler shift is in the other direction.

That sunny day in May we also were able to copy QRSS transmissions of Paolo IZ1KXQ, Andrea IK4IDP and Larry WB3ANQ. Time was limited to a half hour or so, but it was great fun. I wrote an article about this receiver for the July 2007 edition of Electronics Weekly's "Gadget Freak" feature. (I know this weekly, monthly, makes no sense, and I don't have an explanation). The magazine even sent a photographer to my apartment to make me look as insane as possible. 30m QRSS receiver: <http://www.hanssummers.com/qrssrx.html>

An QRSS transmitter powered by a homemade battery!

Around late 2007 the G-QRP Club's annual challenge was to design a transmitter powered by natural sources. For me, lemon batteries immediately came to mind! Remember the school experiment with a lemon, and two dissimilar metals, powering an LCD clock? Since I knew the power available would be low, this would result in low transmitter power... which is what QRSS is perfect for!

I designed a simple circuit using just three BC107 transistors. One as oscillator on 7.014MHz (with no power amplification stage); two more as multivibrator to produce a simple frequency shift pattern to identify the signal as mine. The crystal was later changed to 7.000MHz (a gift from prominent QRSS enthusiast Peter DL6NL).



It was all built on a base which was a piece of tin can. Literally, a piece of tin can cut from the wall of a food tin... but surprisingly useful and easy to solder to. The red LED that you see in the photo, is my usual method of causing a frequency shift in a crystal oscillator circuit, using the LED as a varicap.

This design was notable in its ability to operate down to very low voltages. It still worked down to as low as 0.7V supply voltage. By the time the supply voltage was 1.5V the circuit could produce a massive 2mW of RF power output!

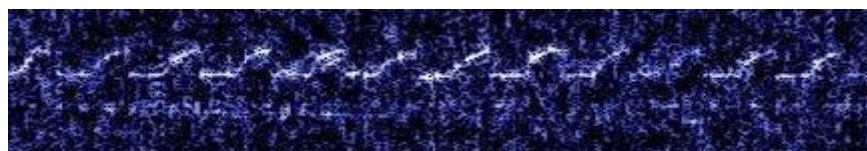
It has a quite high efficiency, not much power is wasted. These features were crucial to the desire of powering the transmitter with a homemade battery.

Surfing the internet, you can find lots of projects which power LCD clocks with lemons, potatoes etc. But they never measured how much power they could get and for how long. It works qualitatively but there is no quantitative detail; so this was the first question to resolve. A kg of lemons and lots of experiments later, I was able to estimate how many thousands of lemons it would take to produce a couple of mA for powering my tiny transmitter - around 30,000 actually - even after



lots of optimization I could get it down to 140 lemons but now the construction was complex, it did not make it a very feasible prospect. On a tip from someone (long forgotten, sorry), I decided to try other electrolytes, I experimented with vinegar and bleach. The final “battery” was made from 4 cells, each cell consisting of a small orange juice bottle, wound with copper wire, placed in a Heinz baked bean can tin and filled with bleach. It could produce 0.7V into a 330-ohm load, about 2mA. The four cells were connected in series.

Powering my simple 3-transistor QRSS transmitter, this arrangement produced 3mW of RF power output, even rising to over 6mW after a few days. I had a short piece of wire thrown out of the window of our small 2nd floor London apartment, which is about the most awful antenna you could imagine. But still sufficient to get quite a few signal reports from around Europe. This one is from Pierre ON5SL on 04-Feb-2008, and shows the distinctive shape of the signal:



After five days the power output had dropped to 1mW, and the bleach batteries were exhibiting severe corrosion, to the point of bleach leaking out of the tins; the stench in the apartment had become unbearable; the experiment was deemed a tremendous success and terminated in the interests of domestic harmony. Homemade batteries:

<http://www.hanssummers.com/homebrew/homebrewbattery.html>

Very low power QRSS transmitter: <http://www.hanssummers.com/qrsslp.html>

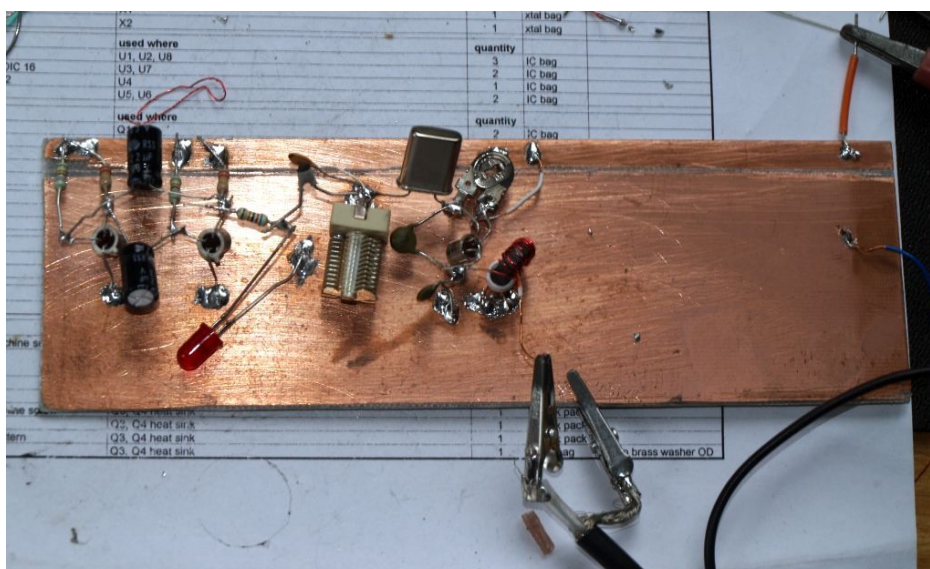
Jul 2008: An ode to QRSS (Steve G0XAR)

My good friend Steve G0XAR, also a fellow QRSS enthusiast, built his own copy of the above low power QRSS transmitter. He wrote the following “Ode to QRSS”, with all due apologies to the late great William McGonagall.

For over a decade, I have wondered exactly WHY it would be necessary to apologize to William McGonagall's ghost, who even IS this McGonagall person anyway, but I was always too ashamed to ask. Now I feel this history may be incomplete if this question is left unresolved, and since Google is my friend, the website <http://www.mcgonagall-online.org.uk/> at least provides the following introduction:

"William Topaz McGonagall, poet and tragedian of Dundee, has been widely hailed as the writer of the worst poetry in the English language. A self-educated hand loom weaver of Irish descent, he discovered his discordant muse in 1877 and embarked upon a 25 year career as a working poet, delighting and appalling audiences across Scotland and beyond."

Anyway, without further ado, and with reference to Steve's transmitter (pictured), here's Steve's ode:



*It was on the eleventh of July two thousand and eight,
In the GOXAR shack a well remembered date,
When GOUPL and YL came to stay
from London which is far away.*

*The XAR shack was a terrible mess
But construction had started on a beacon for QRSS
There were capacitors, resistors and transistors three
Soldered to a strip of PCB
Plus a crystal for the the thirty meter band
All badly soldered by GOXAR's hand*

*Also, in the course of construction
Because of a power supply disruption
the output transistor, my last BC107
let out its smoke and went to heaven*

*"No worries" said G0UPL "What you should do"
"Is use a 2n2222!"*

*We soldered in the shiny new part,
The power supply we then did start
No output current did we see
On the meter, which frustrated me
G0UPL then said "Upon inspection,
I see you have reversed the base and collector connections,
that will have let out the smoke"
I did not see the funny side of the joke,
but another 2n2222 was fitted,
and this time power was emitted!*

*The output signal sounded very rough
And finding the problem was exceedingly tough*

*The G0XAR Oscilloscope
had lost its trigger, so there was no hope
As there was no signal to be seen
on its lovely fluorescent green screen.*

*However G0UPL's massive brain
put us on the right track again
as we reduced the output power
by turning the potentiometer
the roughness it did go away
Oh happy, happy, happy day.*

*We connected the beacon to the doublet
via an ATU to match the beacon with it
we had to guess at the various settings
but managed to avoid any more smoke outletting
Then we mailed our fellow QRSS knights
that the beacon would be on that night
and using Argo it might be in sight.*

*The output power was measured at 10 milliwatts
which most people think is not a lot.
But was sufficient to be received by G6AVK
in Essex which must be 80 miles away.
Then a grabber screen we did see
from I2NTD*

*In Bergamo, which is in the beautiful country of Italy,
clearly showing the G0XAR beacon for all to see!.*

*From Norwich to Bergamo is over nine hundred miles,
For our signals to get there gave 'XAR and 'UPL many smiles
Next i will use the Si570
connected to my computer using USB*

*to give me a better choice of frequencies.
How far will my qrss signals get? Well we shall see!*

2008, a QRSS ocean buoy project

Around this time, I started a project I called “Voyager”. This was a project, which you can read some notes on here <http://www.hanssummers.com/voyager>, which never got near to being made operational. The project was named after the famous Voyager 1 and Voyager 2 space probes. Though obviously a much more humble concept, the idea of exploring and sending back data was a similarity. I well remember walking to work along the river Thames in London, staring thoughtfully at the dark waters flowing by, and imagining all the details of the project.

This was a plan to put a radio transmitter in a piece of plastic drain pipe, with some batteries, and as many different simple low cost instruments as possible, then send the data back using QRSS. I got as far as building a prototype 40m QRSS transmitter and a companion high performance 40m QRSS receiver – but that was as far as it meant.

I mention this just now for two reasons. Firstly, because it sets the background for, and shows my tendency to have an interest in such things, as the balloon experiments that I was to become involved in several years later; secondly because this project was later one of the inspirations Bob ZL1RS cites for his Ocean Floater project.

At the risk of messing up the chronological order of this narrative, Bob really did build an ocean buoy, using several of the QRP Labs kits, and set it adrift in the South Pacific Ocean in May 2016. Powered by 18 alkaline D-cell batteries, the location of the buoy was reliably tracked throughout 407 days at sea, until the batteries were depleted, using WSPR and JT9 weak signal modes. Bob’s very interesting web page on this project:



<https://www.gsl.net/zl1rs/oceanfloater.html>

Several other individuals/teams were interested in also preparing an ocean buoy using a weak signal modes transmitter for tracking and telemetry; however, as far as I know none of these have been taken through to completion.

Jan 2009, Si570 QRSS transmitter

My friend Jan G0BBL runs <https://www.sdr-kits.net/> and he kindly gave me a module he was producing at the time, containing a Si570 synthesizer and microcontroller to configure it. The Si570 is produced by a US semiconductor called SiLabs and this was my first introduction to their product line. It's important because later I came to be very familiar with the Si5351A, as I will discuss further on below.

The Si570 is a frequency synthesizer chip that includes its reference crystal in the package. It has a configurable output frequency but this needs a microcontroller to set up. Jan G0BBL's board had a USB socket and could be configured by a program on a PC. There was a mode which allowed you to configure the frequency, and use one of the pins of the onboard microcontroller to produce a controlled amount of frequency shift. Ah-ha! Perfect for FSKCW mode transmission!

I put together a transmitter with an amplifier, Low Pass Filter, and using an ATtiny13 microcontroller programmed with my callsign to key the Si570 board (see below, for more on this ATtiny13 keyer).



This transmitter was very successful despite my very limited London urban apartment short-wire antenna. See for example this report on 30m by Claudio I2NDT on 02-Jan-

2009. Si570 transmitter: <http://hanssummers.com/qrss570.html>



Jul 2009: Tropical QRSS (a.k.a. Pirate of the Caribbean)

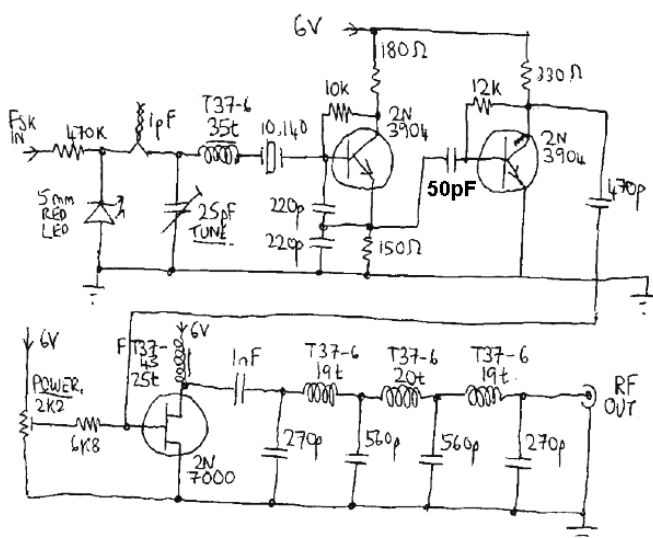
2009, clearly a busy year. By mid-July XYL and I were about to set off for a 2-week holiday on the Caribbean island of Grenada. Packing suitcases with XYL has always been a rather slow process, which as the night before departure wears on, appears to get slower and slower until the point at which time almost appears to stand still, and you wonder if it might stop altogether. On this occasion we had to wake up at 5am the following morning and set off for London's Heathrow Airport for our flight to Grenada.

At 3am (t minus 2 hours), everything was packed and it was time to go to bed. They say that lack of sleep is like alcohol intoxication and this might have been a factor in the rather rash, but as it turned out, PIVOTAL decision which I made on the spur of the moment. Goodnight darling, I'm just going to the radio room to work on something for a bit...

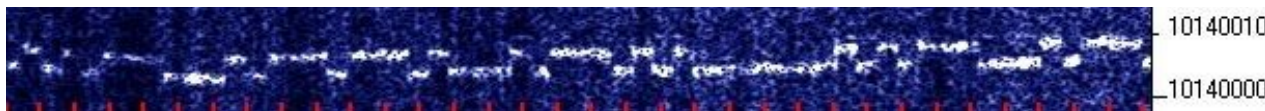
So, what had I thought of, at 3am in the morning, two hours before it was time to wake up and go to the airport? Of course! A QRSS transmitter to take with me!

I had actually already developed a C program for an ATtiny13 microcontroller that encoded my callsign, see <http://hanssummers.com/qrsskeyer>, which I also used in the Si570 QRSS transmitter project. The Atmel ATtiny13 is an 8-pin chip that incorporates an AVR processor and several peripherals, with just a 1KByte program space and 64 bytes of RAM. Plenty, however, for encoding a callsign into FSKCW. I'd also already found on the bench a 30m Low Pass Filter built on a piece of perf-board. The rest of the transmitter was hastily thrown together using whatever parts were nearest at hand, installed in a tiny "Ozone" mint tin (the same type as used in the 30m QRSS Receiver project I described previously), and the whole thing tested.

By 5am, it was all ready and working, along with a wire antenna (30m dipole, FT37-43 toroid balun, and 5m of RG59 coax). Packed neatly into an empty ice cream box. Off we went to Grenada! See <http://hanssummers.com/qrstropical> for all the details of the transmitter and the operation.



The following day I set up the antenna strung over the roof, plugged the transmitter into a 5V mobile phone charger for power, and off it went. The transmission encoded was FSKCW "UPL" - note, not the full callsign. Within a short time, reception reports began to arrive from Europe and US. First to report was Mike G8NXD.



As for Pirate of the Caribbean? Well you've seen the movie(s). Or if you haven't, you should have. Sitting there in the Caribbean, and with unlicensed AM broadcast radio operators in the UK being known as "Pirate Radio stations" in the period covering the '60s to '80s; the opportunity for a little humour seemed too good to pass up.

There was also the small technicality that it was actually ACCURATE... I had no idea whether the GOUPL radio license would be applicable automatically in Grenada. I had neither inclination nor time, to try to find out, what kind of special permission might probably be required, and spend a day in the Grenadian capital trying to explain QRSS to some official or another. Furthermore, having been married at that time for still just under a year, it did seem to me it might be rather early to rock the boat of domestic bliss. And it was only 100mW, after all, who would notice...

This joking reference to the movie and talk of piracy wasn't all too popular among some QRSS enthusiasts and initiated a bit of a debate. There are some who take great interest in examining every detailed facet of the legal terminology in their amateur radio license. Others who really don't care too much, mostly though taking a responsible attitude and trying not to cause any inconvenience to anyone else... and it IS only a hundred milliwatts. Of course, a whole spectrum of opinions between the two extremes. My political incorrectness has got me into trouble on a number of occasions. The debate is probably long since forgotten, but illustrates another recurring issue which does keep coming up in this niche of the hobby. More on the tropical QRSS transmitter: <http://hanssummers.com/grsstropical>

Later operation from Turkey: <http://hanssummers.com/grssmed>

Oct 2009: QRSS talk to the G-QRP Club convention, and the keyer chip

At the G-QRP Club's annual gathering, called the "G-QRP Club Mini-convention" which was then at Halifax, Yorkshire, UK, I gave one of the day's presentations. This was I think, the first presentation I ever gave,

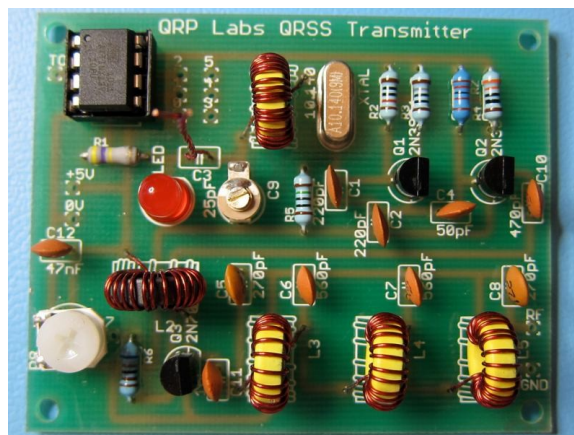


dedicated solely to QRSS modes. To many of the audience who had never heard of QRSS, the topic and technical details were absolutely fascinating. I also offered to program and send anyone who wanted to experiment with it, one of my tiny keyer chips programmed with their callsign (at a cost of £2 to cover the chip cost and UK postage). It was quite a popular thing for a while, I forget how many chips I sent out to people but certainly many new people must have been introduced to the QRSS niche of our hobby. Photos from GOUPL talks: <http://hanssummers.com/talks.html>

May 2010: Dayton 2010 presentation and the first QRP Labs kit

The following May I travelled once again to Dayton for the FDIM conference and Dayton hamvention. I had been invited to give a speech on QRSS and weak signal modes.

My friend Steve G0XAR and I got talking about it and somehow the idea of creating a QRSS transmitter was born. I forget how exactly, but I seem to remember Steve initiated this interesting plan.



There were still some months to go, enough time, but not excessively so, to get prepared. Time was short and we decided to use the simple QRSS transmitter which I had designed in two hours in the middle of the night, in July 2009 before going on our Caribbean vacation. Steve and I agreed that I would design a PCB and write the documentation; Steve would order the PCBs and components, and pack them into 100 little bags. Not long before the Dayton trip it was all ready, Steve and I met for lunch as I recall, and Steve handed over a box containing the 100 QRSS kits.

This was the first kit that had ever been designed for QRSS weak signal modes; prior to this, as I mentioned before, you either had to use a PC generating audio for an SSB transmitter, or you had to homebrew your own QRSS equipment from scratch. Making a kit available made it significantly easier to try out this interesting aspect of amateur radio, and making something easier meant more people had the opportunity to do it. It was also a landmark because it was the very beginning of QRP Labs, the very first QRP Labs kit.

The Dayton FDIM presentation was very well received. At the end of the presentation I announced that I had brought with me (what I thought would be) plenty of kits, of the Caribbean project, which I would be selling for \$15 that evening at the FDIM Vendors' evening. Everybody thought \$15 was very cheap for a kit like that. Including lots of the other kit vendors, some of whom were quite pointed in their criticism of the pricing - a theme which continues routinely to this day, but which I have never let bother me or change my path.

A coffee break was scheduled after my presentation and before I had even stepped down from the podium a little crowd had gathered around me, of people who wanted to buy the kit. I started writing a list of names. Since the source code for every one of those 8-pin AVR microcontrollers had to be individually altered to encode the operator's callsign - a job taking only a couple of minutes, but all those "couple of minutes" would add up fast - it was nice to be able to get a head start on some of the chip programming before vendor's evening.

Evening came and George Dobbs G3RJV, founder of the G-QRP club, kindly offered to let me sit on his G-QRP club table. I had not sold kits before and didn't really know what it meant to be a "vendor". Steve and I had had no idea whether all those 100 small bags of parts could be sold. We needn't have worried! There was a long line snaking around the room, my own little pile-up.



People waited patiently to buy their kit. 100 of them were sold in several hours.

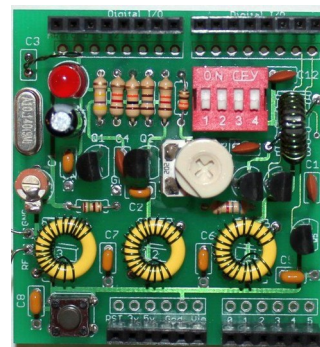
After Dayton, Steve and I decided we'd continue producing this kit. Over the next 2 years around 700 of the QRSS kits were sold. The 100 kits for Dayton FDIM 2010 had been produced for 30m, which is the most popular QRSS band; but later we also produced them in 40, 80 and 160m versions. The 10.140MHz crystals for the 30m band version had to be specially custom-ordered in a batch of 500 crystals, since this frequency is not a commonly available one.

The first QRP Labs kit, a QRSS transmitter: <http://www.qrp-labs.com/qrsskit.html>

Dec 2011: "Ham Radio for Arduino and Picaxe" ARRL book and the Arduino Shield kit

I was invited to write some chapters for this ARRL book <http://hamradioprojects.com> and eventually contributed three chapters:

- QRSS Beacon
- Multimode Transmitter Shield
- High Voltage, High Frequency and High Temperature Data Logger



The first two of these are QRSS projects. They used the Arduino microcontroller platform <https://www.arduino.cc/> which uses the ATmega328 microcontroller, one of the Atmel AVR processors that I was already familiar with, but this was my first introduction to the Arduino.

One of the agreements with the editors was to produce an Arduino “shield” kit for the Multimode Transmitter Shield chapter. In the Arduino world, a “shield” is a board that plugs on top of the Arduino board’s expansion headers, and provides additional functionality. Thousands of different boards are now available around the world.

The QRP Labs QRSS shield kit <http://www.qrp-labs.com/qrssarduino.html> was essentially an Arduino shield implementation of the same QRSS kit described above, but instead of the 8-pin microcontroller producing the frequency shift keyed (FSK) modulation, the Arduino was used. The book chapter provided a detailed explanation of the necessary source code to produce FSKCW from the kit. The kit could produce on/off keying as well as FSK.

May 2012: Ultimate QRSS transmitter kit

Things carried on in this way for another year, or perhaps year and a half. Then there was an unfortunate cloning incident, of the QRSS kit. At the time it was quite annoying to Steve G0XAR and I (to say the least), but this is all ancient and unpleasant history now so we don’t need to dig into it or embarrass anyone. In the end, we remembered what the Americans say: “Don’t get mad, get even”... and this was the initial inspiration to design a new QRSS transmitter kit which would be 10 times better than the first, but only marginally more expensive; it would therefore render the original QRSS kit obsolete, at least for anyone that knew about it.

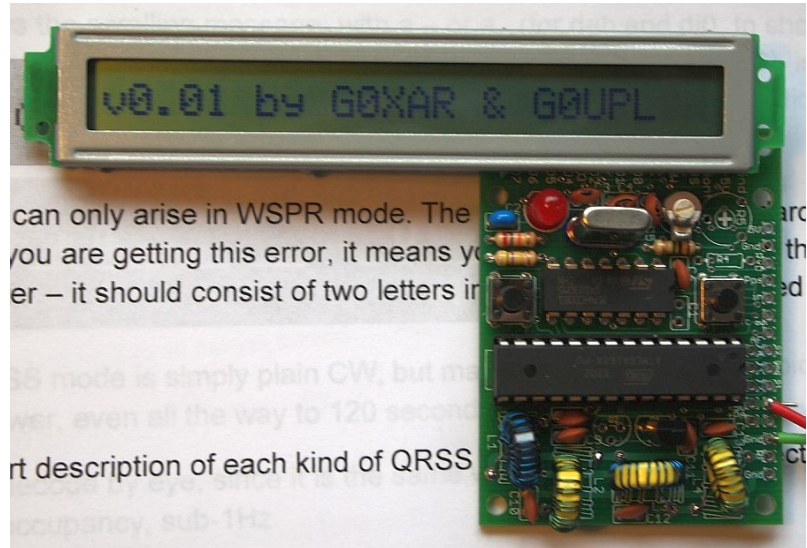
The next QRP Labs kit was born, called the “Ultimate QRSS transmitter kit”. For the first time, this kit was configurable by the operator. It meant that Steve G0XAR no longer had to individually compile each kit builder’s microcontroller source code, to encode their callsign. Now with the new kit, the operator entered their own callsign and other details, in the menu.

The kit had an unusual 24-character Liquid Crystal Display, which (perhaps due to being unusual, not the standard HD44780-compatible module) was sold very cheaply by a US surplus electronics store, <http://allelectronics.com>. Even now, six years later, they still sell this display, according to their website they have a “huge” quantity and they sell them for two for \$1. Actually, this is many times cheaper than what we paid in 2012 - but the display was certainly very much cheaper than any other LCD modules available at that time.

The display had an awkward set of connector pins at 0.05-inch pitch (1.27mm), that was not easy to solder to. In the kit, the display was soldered directly to the PCB; though many constructors opted to use wires between the LCD module and PCB so that they could mount the kit in an enclosure.

The transmitter was conceptually similar to the original QRSS kit, still using a crystal oscillator (but this time using a 74HC00 logic IC) and MOSFET amplifier stage with the 2N7000 device. The familiar red LED “varicap” was still present, to pull the crystal frequency slightly to create the frequency shift keying; it was still necessary to trim the oscillator frequency to get it on frequency. The amount of FSK frequency shift was now adjusted by a configuration menu parameter.

A more powerful Atmel AVR microcontroller was employed, the ATmega168. With the LCD and two tiny push-buttons on the board, it was possible to scroll through a number of menu items and configure the operation of the transmitter. The kit featured a GPS interface; serial data was decoded into time and location data to convert to Maidenhead locator and set the realtime clock in the firmware, and the 1 pulse per second (pps) signal used to discipline the frequency. During calibration, it was necessary to coarse-tune the frequency using the trimmer capacitor; thereafter the GPS (if connected) could discipline the frequency, eliminating any long-term temperature-induced frequency drift.



The new kit also could encode WSPR on-chip. Typically transmitting it required a PC generating audio tones sent to an SSB transmitter. Many people didn't like tying up a large expensive power-hungry PC and a large expensive power-hungry SSB transceiver, for long periods to send WSPR. There were a couple of homebrew projects using microcontrollers where people had used a utility in the PC WSPR Software package to generate a list of the WSPR tone symbols, then copy/pasted them into microcontroller source code, compiled it and installed it on a microcontroller. But this was hardly easy or convenient, for many people. Wait! What is WSPR? For those that don't know, WSPR stands for Weak Signal Propagation Reporter, it was developed by Joe Taylor K1JT and introduced in 2008, via a software application for PCs. A WSPR transmission digitally encodes your Callsign, Maidenhead grid square, and power level into a sequence of tones. There are 162 tone symbols in the transmission, each is one of four different tones (frequencies) separated by 1.46Hz. The bandwidth of the signal is about 6Hz and the transmission lasts 1 minute and 52 seconds. The transmissions must always begin on the 2nd second of even minutes past the hour, plus or minus a few seconds; a maximum drift of 4Hz during the transmission can be tolerated by the decoding software. The data encoding uses sophisticated error correction techniques so decoding is possible even if parts of the message are lost or corrupted by interference. Here's what WSPR transmissions look like on an FFT (such as the popular Argo software).

They are quite distinctive but cannot be decoded by eye; a PC running the WSPR decoding software is necessary.

WSPR decoding receiving stations are normally connected to the internet, and WSPR decoding "spots" are updated to an internet database at <http://wspnet.org>; on this website you can view download all your "reception reports", or see them plotted on a map.



The WSPR software packages were (and still are) Open Source but highly incomprehensible; it is a mixture of Fortran and C, but trying to understand what it does is quite challenging. Fortunately, Andy G4JNT wrote a concise description of the WSPR encoding process http://www.g4jnt.com/wspr_coding_process.pdf that enabled me to write my WSPR encoder for the new kit.

WSPR has grown and grown in popularity, it now far outstrips interest in the original QRSS modes. Perhaps that is because transmission, reception and reporting is automatic. Like the original QRSS kit, the new "Ultimate" was available in versions for 160, 80, 40 and 30m, the latter being most popular. The kit could be configured to operate quite a list of Weak Signal modes:

- Plain QRSS (on/off keyed slow CW)
- FSK/CW mode (frequency shifted slow CW)
- DFCW (Dual frequency CW)
- WSPR
- Slow-Hellschreiber
- Original speed on/off keyed Hellschreiber
- Normal CW
- Patterns

The latter feature, user definable patterns, was a nice feature that was carried over throughout the subsequent evolution of the kit. It allowed the operator to configure a sequence of shifts, to trace out whatever frequency shift pattern they wanted. In this example the configuration message *01ACDFGIJJKKKJJIGFDCA87542110001124578* produces a slow frequency-shifted sinewave trace:

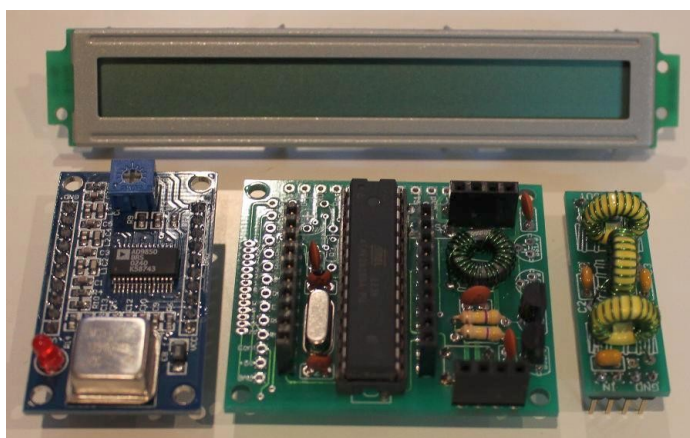


The Ultimate QRSS kit was sold for 50% more than the original QRSS kit, a modest increase for the huge additional list of features: LCD, user-configured operation, GPS interface, and all the modes, including WSPR. It immediately

rendered the original QRSS kit obsolete (though they continued to sell occasionally, until stocks were exhausted); almost 400 units of the Ultimate kit were sold in about 6 months. Ultimate QRSS kit information: <http://www.qrp-labs.com/qrsskitmm.html>

Mar 2013: Ultimate2 QRSS kit

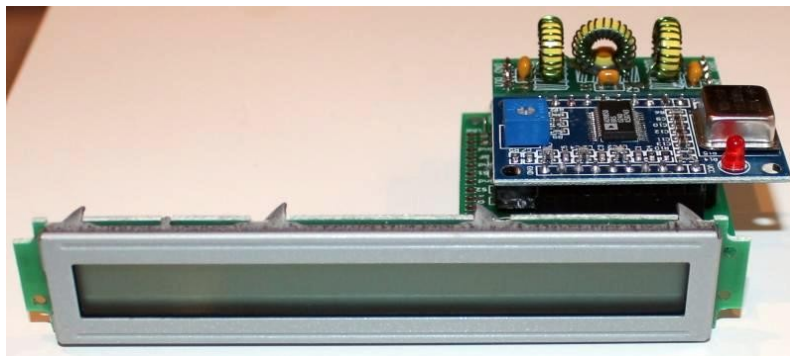
Around the end of 2012, Steve G0XAR and I amicably stopped working on the QRP Labs kits together. In early 2013 I began work on a successor to the Ultimate kit and launched the "Ultimate2" kit in March 2013. From this time onwards, QRP Labs became very serious for me, and I started putting a lot of time into it during evenings and weekends. QRP Labs grew quite fast, after this. The Ultimate2 advanced on the Ultimate kit, by using a Direct Digital Synthesis (DDS) chip as the frequency source. It made very precise frequency control possible. DDS chips are quite expensive, they were available from Analog Devices in one-off free sample quantities, which many hobbyists made good use of, but which for a kit producer wasn't useful. DDS chips are also tiny, only available in Surface Mount format with very close pin-spacing, making them quite difficult to deal with for the average amateur radio constructor. Well, eBay saved the day, a cheap Chinese module was available which integrated the AD9850 DDS (0 to 40MHz) and 125MHz reference oscillator and all other necessary components, onto a small ready-assembled blue-coloured PCB. At the time, these were available for about \$5-\$6. This is about one third the price of an AD9850 DDS chip from distributors such as DigiKey and there were a lot of theories about how this could be possible, including that the chips are fake. I've dealt with the Chinese electronic components market quite a lot over the last 5 years and my own theory about these low-cost DDS modules is that the chips were non-RoHS (the lead-free component regulations), and when the RoHS rules came into effect, some manufacturer in China got left with a big pile of non-RoHS chips, which they couldn't use in their product, and they got dumped into the black market. There, they found their way into this DDS module, which I believe was actually originally a design for a DDS development board to demonstrate the product.



The Ultimate2 kit was created to use this AD9850 DDS module. The same AllElectronics 24 x 1 LCD module was used. This time the Low Pass Filter (LPF) was also made on a separate PCB, that was plugged into the main board. This made it easy to change the band of the kit, just by plugging in a new filter. These LPF kits became the 2nd kit in the QRP Labs kit product line, as LPF kits are a useful module to any amateur radio experimenter building his own transmitters. They were initially available for bands 160 to 10m, later extended down to 630m and 2200m bands, and upwards to include 6m,4m,2m,222MHz.

They continue to be very popular to this day, unchanged since the original kit launched in 2013.

The quality of the Chinese AD9850 DDS modules was not particularly high nor particularly consistent, though I did notice it improved considerably over the next two years. There was a blue trimmer potentiometer which set the comparator threshold for the chip to produce its squarewave output that was used by the Ultimate2 kit. I



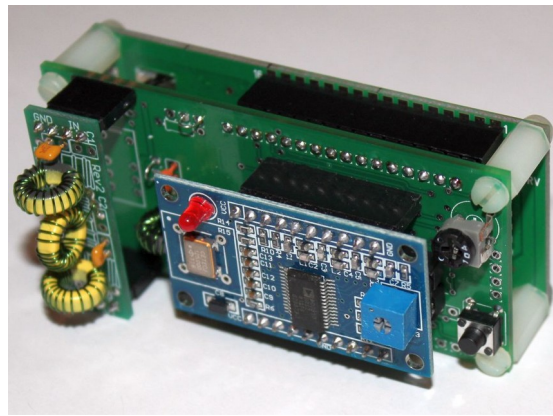
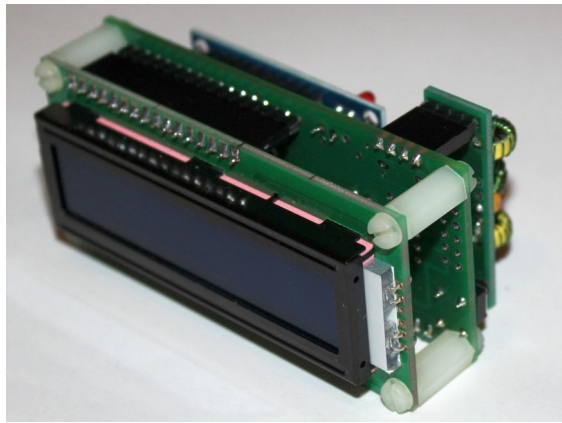
I had to individually test and align every AD9850 DDS module, to make sure that it worked (for a good start!), and that the frequency stability was good enough for QRSS modes, and to align the blue trimmer potentiometer. The yield of good modules was around 90-95% to start with, but they did improve over the following year and reached more like a 99% pass rate. The testing/alignment was quite time-consuming and as the hundreds rolled by into thousands, it did get a bit tedious!

The Ultimate2 functionality was initially inherited from its predecessor the Ultimate, but using the DDS as the frequency generator. A continuous stream of owner requests meant the firmware kept evolving. Owners could update the firmware themselves if they had an AVR programmer. The easy band swap and frequency synthesis meant that the kit was made available for all nine bands from 160m to 10m; later 60m was also added (the band allocation becoming available in an increasing number of countries). The Ultimate2 kit, or “U2” as it became affectionately known, was very popular and sold about 400 in 6 months. Ultimate2 kit information: <http://www.qrp-labs.com/qrsskitmm.html>

An interesting digression, about this LCD module, which not many people know: I could never figure out how to display anything in the 24th character, at the right-hand edge of the display. Every time I wrote a character in that 24th position, the code froze. My solution, which was not really a solution at all, but an admission of defeat, was to NOT write anything into the 24th character. I wonder how many Ultimate and Ultimate2 kit owners ever realized that the 24th character was always empty?

Nov 2013: Ultimate3 QRSS/WSPR kit

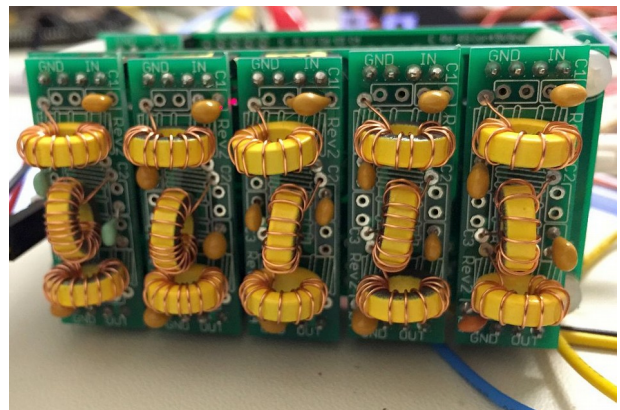
The popularity of the Ultimate2 kit and the growing interest in QRP Labs products inspired the next iteration of the “Ultimate” series: the Ultimate3. The big improvement here, was the move to the more standard HD44780-compatible LCD module, with 16 characters per row, on two rows, and nice blue backlight.



It made the surplus AllElectronics display (with no backlight) look ancient and obsolete! The new modules were getting cheaper and cheaper, and began to rival the price of the AllElectronics module (no surprise then, that the AllElectronics LCDs have got so cheap now!).

The use of this LCD module brought two advantages other than the much nicer, more professional look. The first of these two big improvements, was the physical assembly of the modules. The Ultimate and Ultimate2 kits had been rather awkwardly arranged because of the size and shape of the LCD module, and the location of its connection pins. The new 1602 LCD modules are 80 x 36mm, with M3 mounting holes in the corners. By good fortune, I had made the Low Pass Filter kit modules 1.5 x 0.5 inches – the longer dimension is 37mm in metric terms, which is almost the same as the LCD height. This made it possible to create the “stack” or “module sandwich” shown in these photographs, held together by hex spacers. It produces a stable and “cute” stack of boards that is very appealing, as well as highly functional.

The second important improvement was that whereas the AllElectronics module required 8-bit data addressing, taking 8 I/O signals from the microcontroller, the 1602 HD44780-modules can be run in 4-bit data mode. Additionally, they require one less control signal. This freed up enough control signals that I could create one more new QRP Labs kit, the 6-band relay-switched filter board! (Pictured here, a kit built by Kevin KK4YEL).



Now the Ultimate3 kit was able to transmit sequences across up to 6 different bands, all automatically, and highly configurable by the operator. It brought a whole new dimension of flexibility to the kits and inspired a lot of experiments. A smart black anodized aluminium enclosure kit was made available. Though many inventive constructors produce their own enclosure solutions it can't be denied that many of us (myself included) are not very adept at metal bashing; we are lucky if we can manage soldering to any degree of competency. For all

such people, the nice drilled, cut, silk-screen-printed aluminium case and its accessory kit (buttons, connectors, hardware, feet) was a Godsend.

Ultimate3 QRSS kit information: <http://www.qrp-labs.com/ultimate3/u3.html>



QRSS in Japan

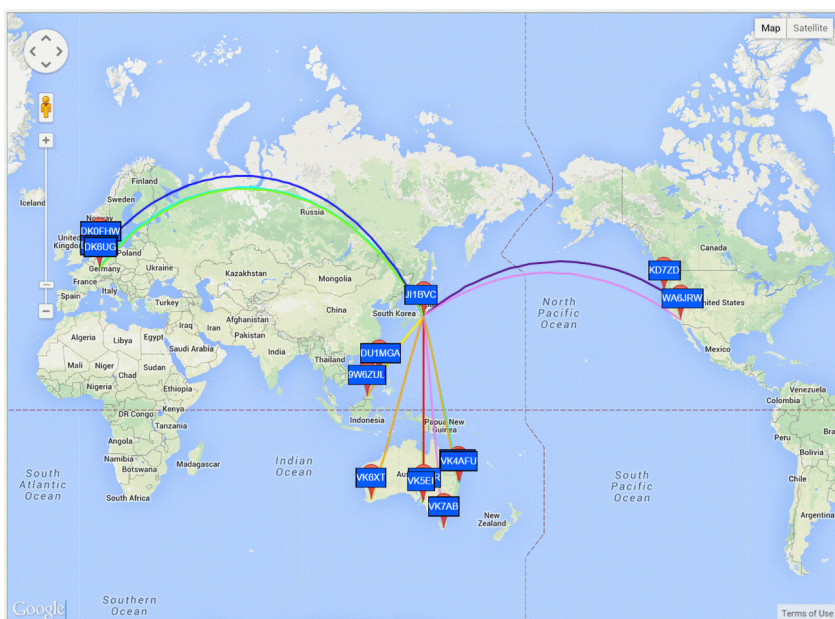
I had the opportunity to spend an amazing 5.5 years in Tokyo Japan from February 2011 to June 2016 so of course, this was where a lot of the QRP Labs kit development took place. Japan is not a CEPT signatory country nor does it have a reciprocal licensing arrangement with UK. My G0UPL license was therefore not usable in Japan. If you are Japanese, you can pass the written examination, and the required Morse test (which is Japanese Morse, additional to the standard international Morse). This is tantamount to an impossibility for non-native Japanese.

The solution taken by many foreigners in Japan, was to take the US license, examination sessions are held in Tokyo several times per year. I took the Technician, Intermediate and Extra papers all in one sitting, and used the reciprocal licensing system to get my Japanese station license J11BVC.

I actually rarely used the license, and even then, only on Weak Signal modes. I had a very poor 30m dipole up on the roof, only about 1m from metal railings around the perimeter of the flat roof, with the elevator machinery and air conditioning machinery all nearby. But I was able to have some fun with it. Japanese accommodation is famously small, and my radio room / shack / lab even smaller... I installed myself in the bedroom closet, just 5 x 4 feet (1.5 x 1.2m); but big enough to hold a small desk and chair, and all my equipment and parts. As long as you did not stay inside with the door closed for more than a few hours, the oxygen levels didn't fall too low. This was the headquarters of QRP Labs for 5.5 years! Here's where the Ultimate2, Ultimate3 and Ultimate3S kits and many others were developed and produced!



This map shows the results of some hours of WSPR operation in March 2015.



Jan 2015: Ultimate3S QRSS/WSPR kit

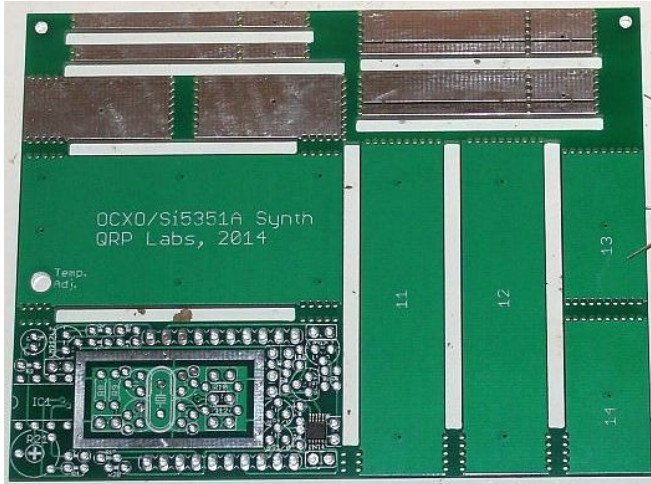
Unfortunately, or perhaps fortunately depending on how you look at it, all those black-market (my theory) AD9850 DDS modules started getting used up. In total I estimate I had purchased 2,500 of these modules, each one individually tested and aligned by me. The price was starting to rise, by the end of 2014 they were costing \$8 per

module. Today (October 2018) they are around \$13 or \$14. Even at \$8, these modules were squeezing my capability to continue to offer the Ultimate3 kit at its famously low price.

An even worse problem was that whereas I used to be able to telephone a factory and say “I want 500 AD9850 DDS modules” and they’d be manufactured to meet my demand and sent over; now it was getting impossible to find the chips to manufacture into the modules. It was necessary for the suppliers to go around the markets in Shenzhen grabbing whatever AD9850 chip stock they could get from here and there, smaller quantities at a time. My entire QRP Labs operation was now based around this module and so this was getting to be a big worry. I was chatting to my good friend Jan GOBBL, who runs <https://www.sdr-kits.net/> and he mentioned the new Si5351A synthesizer chip. Unlike the expensive Si570 which many hams were using in their projects (see earlier), the Si5351A was very much cheaper, not much more than \$1. It did require a 25MHz or 27MHz crystal (whereas the Si570 had one built in). Hardly a problem, considering the factor of 10 (or more) difference in price! Additionally, the Si5351A chip runs down to lower frequencies (down to 3.5kHz) and can produce three different frequency.



I was chatting to my good friend Jan GOBBL, who runs <https://www.sdr-kits.net/> and he mentioned the new Si5351A synthesizer chip. Unlike the expensive Si570 which many hams were using in their projects (see earlier), the Si5351A was very much cheaper, not much more than \$1. It did require a 25MHz or 27MHz crystal (whereas the Si570 had one built in). Hardly a problem, considering the factor of 10 (or more) difference in price! Additionally, the Si5351A chip runs down to lower frequencies (down to 3.5kHz) and can produce three different frequency outputs simultaneously. This chip is a real bonus to the amateur radio world and now used in many places, even in the Elecraft KX2 transceiver. I designed two new kits around this chip, to solve my



AD9850 DDS problem. Both of these kits are designed to have the same 2x10-pin header connectors as the AD9850 DDS module, with the supply, control signals and outputs on the same pins. The Si5351A chip is rather small so I had the PCB factory solder them to the PCB, not requiring the kit constructor to do it. It is supposed to be a plug-in replacement for the AD9850 DDS module, my intention was to make it possible to plug into an Ultimate3 kit so that I would only need to manufacturer these synthesizer kits instead of buying the AD9850 DDS modules. The places, even in the Elecraft KX2 transceiver. I designed two new

kits around this chip, to solve my AD9850 DDS problem. Both of these kits are designed to have the same 2x10-pin header connectors as the AD9850 DDS module, with the supply, control signals and outputs on the same pins. The Si5351A chip is rather small so I had the PCB factory solder them to the PCB, not requiring the kit constructor to do it. It is supposed to be a plug-in replacement for the AD9850 DDS module, my intention was to make it possible to plug into an Ultimate3 kit so that I would only need to manufacturer these synthesizer The places, even in the Elecraft KX2 transceiver. I designed two new kits around this chip, to solve my AD9850 DDS problem. Both of these kits are designed to have the same 2x10-pin header connectors as the AD9850 DDS module, with the supply, control signals and outputs on the same pins. The Si5351A chip is rather small so I had the PCB factory solder them to the PCB, not requiring the kit constructor to do it. It is supposed to be a plug-in replacement for the AD9850 DDS module, my intention was to make it possible to plug into an Ultimate3 kit so that I would only need to manufacturer these synthesizer way the Si5351A chip is configured is totally different to the AD9850 DDS, but the firmware was changed so that it could operate both chips, and even decide at power-up time which chip module was plugged in, and adapt to that. One of the kits is a basic synthesizer kit, the other is an Oven Controlled Crystal Oscillator (OCXO) version: the only kit-built OCXO in the world! Pieces of PCB are soldered together to make an internal oven chamber for the 27MHz crystal. It's fiddly to assemble, but works well for occasions where you need (or want) very high frequency stability.

Si5351A Synth: <http://www.qrp-labs.com/synth.html>
 OCXO/Si5351A Synth: <http://www.qrp-labs.com/ocxokit.html>

Note that in both these cases, there is no physical adjustment of the 27MHz reference crystal oscillation frequency. It's a neat trick: the frequency does not need to be spot on 27.000000MHz - the microcontroller can accurately measure it using the 1pps signal from a GPS,



then compensate for the error when calculating the configuration of the Si5351A. Si5351A source code examples are provided <http://www.qrp-labs.com/synth/si5351ademo.html>

A problem with the plan to make it plug-in compatible was that the AD9850 DDS produced a 5V squarewave output, whereas the Si5351A module output was only a 3.3V squarewave. Unfortunately, that drove the power amplifier transistor less hard, and the result was about a 50% loss of transmitter power output. Technically 50% doesn't matter much but to many people, and to many perceptions, it does... so this necessitated an upgrade to the Ultimate3 kit, which was named the Ultimate3S. Ultimate3S: <http://www.qrp-labs.com/ultimate3/u3s.html>

The main change for the Ultimate3S ("U3S") was adding a coupling capacitor and The Ultimate3, particularly on the higher frequency bands such as 10m band - which had always been challenging on the Ultimate3. DC bias circuit to the power amplifier. With this change, the power output of the Ultimate3S comfortably exceeded it.

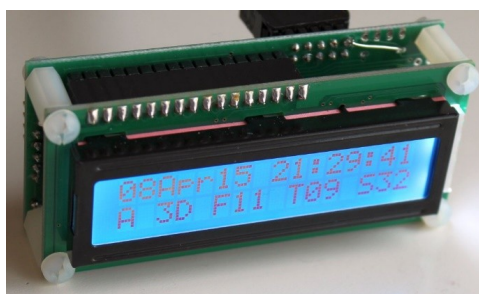


The switch to the Si5351A also let us climb up higher in frequency, since the chip will officially go up to 200MHz. So, the Ultimate3S can operate on 6m, 4m, 2m and even the US 222MHz band (violating the Si5351A specification a little). On the lower frequencies, it operates on 630m and 2200m bands.

The Ultimate3S kit continues to be popular to this day. To date, almost 6,000 of the Ultimate Series kits have been sold (Ultimate, U2, U3 and U3S).

The unfortunate design error for the Ultimate3S (the loss of half the output power due to the lower Si5351A voltage output than the AD9850) had an important side-effect. I'd just manufactured 640 of the Ultimate3 boards, which were now useless. By the way, why 640? Because at that time, the PCB factory actually produce the boards on a massive 3-square-meter piece of unetched FR4 board; 640 was the number of Ultimate3 PCBs required to totally fill up one 3m² board which gave the optimum pricing!

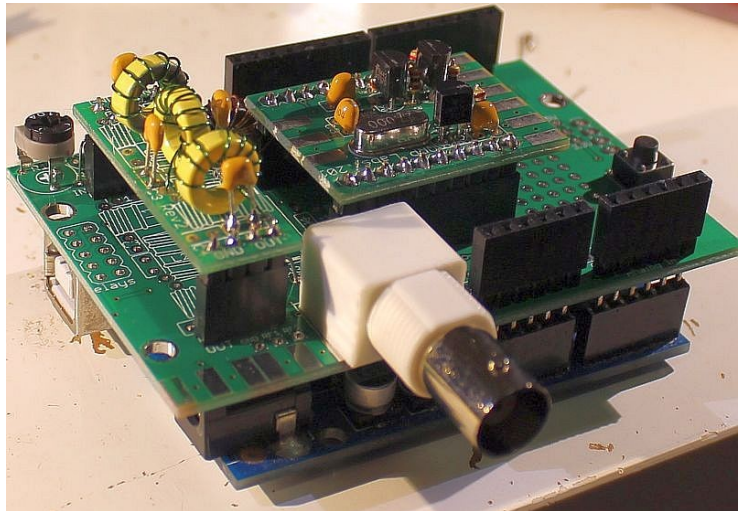
What to do with this big box of useless PCBs? The solution came in the form of two more spin-off kits, designed during 2015, which would use up those old "wasted" PCBs. A Clock kit (left) and an Si5351A VFO/Signal Generator kit.



The VFO kit has rotary encoder tuning, variable and fixed frequency outputs from 3.5kHz to 200MHz, quadrature LO output mode, frequency memories, IF offset, and band-switching relay outputs. Both the Clock and the VFO have optional GPS discipline (time, in the case of the clock; and frequency, in the case of the VFO/SigGen kit. Clock kit: <http://www.qrp-labs.com/clockn.html>
VFO kit: <http://www.qrp-labs.com/vfo.html>

Both kits became very successful in their own right as important projects or modules for hobbyists, and the wasted boards were all used up; nowadays the kits use the same boards (and components pack) as the Ultimate3S. So, the mix-up with the AD9850 DDS vs Si5351A output voltages which had me despairing in Jan 2015 ended up beneficial. Always try to remember: “every cloud has a silver lining!” – not always true, but often it is so.

Apr 2015: New Arduino Shield kit
There are always those who want to do their own programming and since the cloning incident, and for other reasons, QRP Labs code hasn't been Open Source. For all such people the Arduino shield that was made available at the same time as the ARRL Book (see above) was a great solution. But it needed updating now that the rest of the QRP Labs kits had moved on. Hence the new Arduino shield kit, designed to replicate the functionality of the Ultimate3S kit.



Pictured here plugged on to an actual Arduino, the new Arduino shield kit has sockets for the LPF kit and Si5351A kit. It also has a space for the 6-band relay-switched LPF kit, and it can accommodate the OCXO/Si5351A Synth kit if required. It has the same power amplifier as the Ultimate3S kit. There's space for a header on the board to connect a GPS for GPS discipline. And even a little prototyping area, a 0.1-inch matrix of holes/pads in the spare board space, where you could add your own components if you need changes. This shield is compatible with all the other kits so can do everything an Ultimate3S can do, and more – if you want to write the code for it. New Arduino shield kit: <http://www.qrp-labs.com/uarduino.html>

Jun 2015: QRSS to the skies!

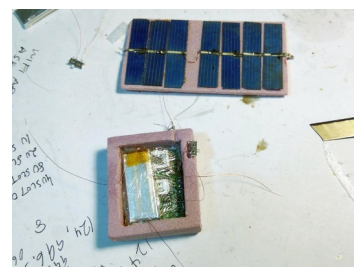
In 2015 David VE3KCL contacted me about flying a lightweight version of the Ultimate3S transmitter kit on a High Altitude Balloon. I'd read about high altitude balloon flights before using latex weather balloons, and later came to know of Andy VK3YT's "pico-balloon" flights from Australia. They used inelastic mylar film "party" balloons which don't expand as they go up; if you get the gas inflation volume correct, they reach an altitude where everything balances in equilibrium and the balloon doesn't burst, then they can get carried along by the jet stream for days, weeks or even months. They are much less costly than weather balloon flights and a different type of outcome: long distance rather than pictures from the edge of space. Typical altitudes are 9,000 to 12,000m; once above 10,000m the chances of staying away from high altitude bad weather are pretty good.

An awful lot of people write to me about an awful lot of different things and for the most part, one has to accept that there are only 24 hours in a day, and most of these fun projects have to be passed by. But not so with Dave's high altitude U3S talk. That bit me right there where the previous "Voyager" plans had left off, the sense of exploration and adventure, tracked by low power radio, I couldn't escape.

The challenges of this kind of flight are that your balloon could fly literally, around the world. It can go anywhere, if it lasts long enough (doesn't get taken down by high altitude weather systems, or the helium/hydrogen gas leak out). The communications distances are therefore vast. The huge distances imply HF communications on 20m or 30m bands. Such DX would normally require some reasonable amount of power to ensure a good chance of successful communications - but unfortunately these "party balloon" flights can only carry a very light payload. This totally rules out the use of a "reasonable amount of power" because you can't carry the equipment to produce it.

Weak signal modes, particularly WSPR, are the ideal solution because they have very narrow bandwidth as discussed earlier, therefore very high SNR, and small powers can span great distances.

Hence it came to pass that the balloon adventures began; Dave launched his first U3S flight on 03-Jun-2015 and hasn't stopped ever since. Other balloon enthusiasts have joined in and also launched U3S balloon flights, notably Jim N2NXZ and Dan DL6OW, but also some others Bob ZL1RS and my apologies if I have neglected anyone else. WSPR, FSKCW, QRSS modes and others have all been operated from the skies from U3S flights.



The first flights used U3S firmware installed on "Arduino Nano" boards which happen to use the same processor as the Ultimate3S. Later flights used purpose-built hardware and firmware. From the short first flight lasting only a few hours, Dave has improved the process and achieved several circumnavigation flights. It has been a long and enjoyable collaboration

between Dave VE3KCL and I, and I hope to soon produce a balloon tracker kit so that more people can enjoy flying weak signals.

More information: <http://www.qrp-labs.com/flights.html>

Updated 29-Mar-2018 01:28Z; Loc=, Duration=17d 15h 28m, Distance=51,492km
Alt=10,600m, Speed=62knots, Batt=3.76V, Temp=-10C, GPS=1, Sat=1



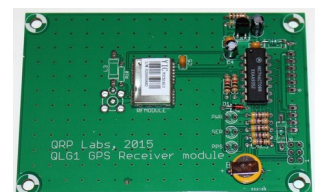
Jun 2016: Eating spam: the QLG1 GPS kit

We all suffer from spam emails and it doesn't get any better, when you have a website presence. Normally these emails are just a big fat pain in the posterior. I get a lot of emails from China advertising PCB manufacture and assembly services and all sorts of products. Now the ONLY time I ever responded to one of these spam emails, led to this new kit, the QLG1 GPS Receiver module kit!

QRP Labs kits had been compatible with GPS receiver modules for some years, because as you know, the Weak Signal modes require a lot of frequency stability and some, such as WSPR, also require precise timing. GPS modules however, caused a lot of problems.

Various different modules had different interfacing requirements, usually involving pull-up resistors to get the voltage levels right: most GPS modules have 2.8V output logic but the kit microcontrollers run on 5V. Many GPS modules also require 3.3V supply and so additional voltage regulators were needed. To resolve all the confusion many people asked me to stock GPS modules that I could guarantee would work with the kits. For some time I'd been doing just that, going through a succession of different module types: VK16E, SKM52, SKM61 as supply dictated, and improved modules became available. Interfacing requirements still arose but at least they could be described and documented, and anyone buying these modules would know how to use them.

So back to the spam email... this one was from a GPS module manufacturer, and led me to a wild train of thought – what if I could design my own GPS module that would be just PERFECT for use with all the QRP Labs kits, not involving any compromises at all. So for the first time in my life, I answered the spam email and the project was underway.

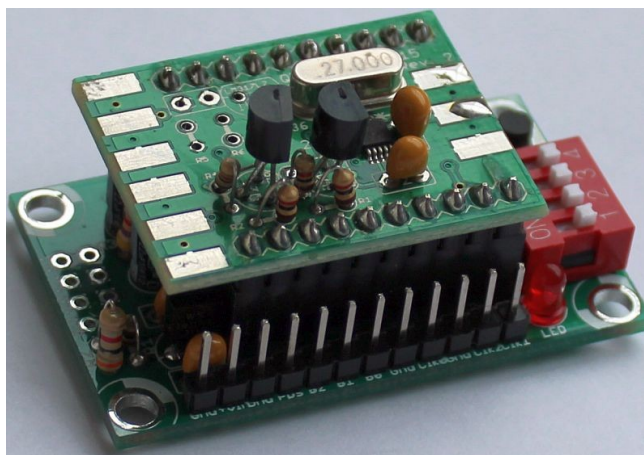


The module has the GPS RF module pre-soldered at the factory. There's a custom-tuned patch antenna on the rear side which together with the relatively huge groundplane (large PCB size) results in an incredibly sensitive GPS receiver section which usually works fine indoors etc.

The kit runs from 5V supply (it includes its own voltage regulator); it has three LEDs red, yellow and green which indicate power, serial data and 1pps activity respectively. And very importantly, a 74ACT08 chip acts as an accurate and fast voltage level converter from 2.8V to 5V logic, so no dodgy pull-up resistors (with their compromises on noise margin) are needed. Eating spam can sometimes be good for you! QLG1 GPS Receiver kit: <http://qrp-labs.com/qlg1>

Apr 2016: Simple Si5351A controller, "ProgRock"

In April 2016 this new kit was released which is a simple and low-cost way to configure the Si5351A. The name is a humorous reference to the rock genre but also supposed to mean "programmable crystal" since quartz crystals in amateur radio constructions were always regarded as very stable oscillators and referred to as rocks.



The ProgRock kit has a small AVR microcontroller that configures the Si5351A Synth kit plugged in; frequencies can be configured by 4-DIP switches and a button, or by a serial port interface. It includes a unique power supply noise rejection circuit designed by Alan G8LCO. It's a great way to do RF synthesis cheaply. ProgRock kit: <http://qrp-labs.com/progrock>

Jun 2016: Big LIFE changes!

2016 was a big year, a monumental year, really... the end of my 22-year career in investment banking IT (software development and management). No more corporate politics, stress, corruption. 2016 was the year the Weak Signals obsession took over my life completely. QRP Labs became my full-time occupation. It had grown too big and could no longer fit in my evenings and weekends as a side business! Now it had to consume my DAYS too.

Aug 2016: A Weak Signals Receiver

For quite a few year's I'd wanted to design a receiver to complement the QRSS transmitters. I'd already had one attempt at a kit for this, a failure which never made it to production. I still have a box of 200 PCBs hanging around here. They are the victim of one too-optimistic designer, one iteration of prototype boards

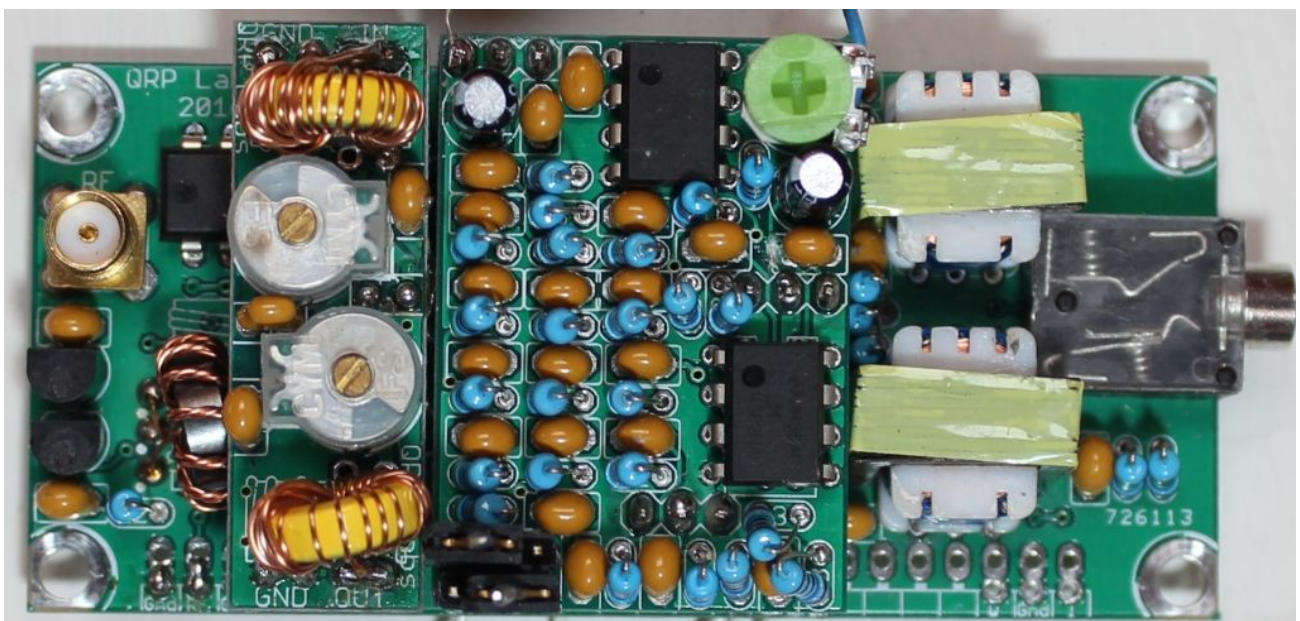
too few... they never worked, and the project, dating back to perhaps 2010 or so, never made it out of the lab.

2016 was the time to right that wrong. The result was a receiver module with excellent performance, including built-in 600:600-ohm audio isolation transformers, for feeding audio to a PC sound card without creating ground loops (50/60Hz hum problems).

The receiver is direct conversion using a Quadrature Sampling Detector (QSD) in a high-performance double-balanced configuration. High-performance low-noise operational amplifiers in instrumentation amplifier configuration result in really excellent performance (dynamic range, IP3 and common mode rejection). The output is a 3.5mm stereo socket suitable for feeding to a PC for demodulation. The output is I and Q channels suitable for a Software Defined Radio package to demodulate.

The kit comes with a plug-in Band Pass Filter (BPF) kit to attenuate unwanted out-of-band signals, reducing intermodulation products. The BPF kit is also available separately, and can be used with the relay-switched filter kit; making these modules popular with homebrewers for use in their own projects.

Optionally a Polyphase audio phase shift network kit can be plugged onto the receiver module, resulting in single sideband demodulation (rejection/cancelation of the unwanted sideband). The rejection is optimized around 1500Hz which is optimal for WSPR reception and can also be used for other QRSS mode reception. Pictured here is the receiver module with BPF and Polyphase modules plugged in.

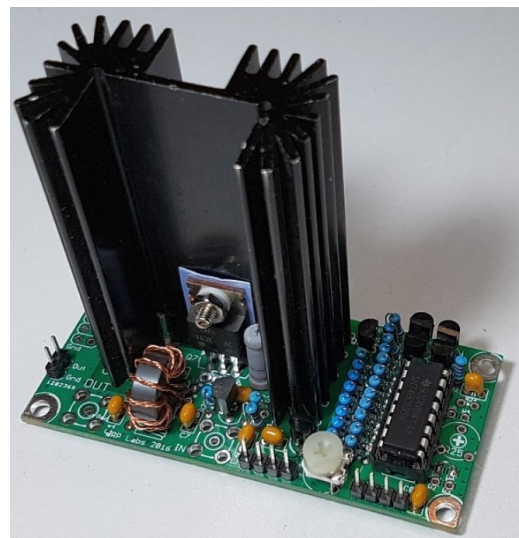


The board is 80 x 37mm, the same size as the Ultimate3S kit and VFO/Signal Generator kit. The receiver kit can be paired with these kits. It requires a local oscillator at 4x the reception frequency, which is divided by four with a 74AC74 chip to produce quadrature for the QSD.

The receiver has really excellent performance, the QSD has really high dynamic range and IP3, high sensitivity due to the low-noise op-amps, and a low level of intermodulation products due to the narrow BPF. I lost count of the number of people who emailed me to tell me they get more WSPR spots with this receiver than they have ever been able to get with their Kenwood/Yaesu/Icom/Elecraft etc. But of course those radios do a lot of other things too, they aren't just dedicated to QRSS reception ☺ Band Pass Filter kit: <http://grp-labs.com/bpfkit> Receiver kit: <http://grp-labs.com/receiver> Polyphase module kit: <http://grp-labs.com/polyphase>

Jan 2017: When milliwatts aren't enough...

It's controversial... how much power do you need for weak signal modes? Argue as you wish but, it can't be denied for very low frequency work on 630m or 2200m bands, antennas are inefficient and a bit of extra power is welcomed; further on 10m band higher power is very useful. In between, we can debate. For all those who wanted higher power output from the Ultimate3S, and a module to use in their own products, this Class C amplifier (non-linear) was launched in Jan 2017. It uniquely includes a power modulator circuit controlled by an 8-bit Digital to Analogue converter, that can be used by the Ultimate3S kit to create precise raised cosine keying envelopes (no key clicks).

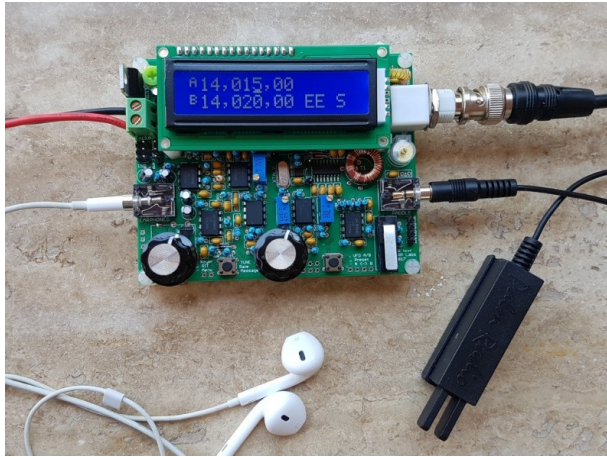


For standalone use, an Atmel AVR microcontroller is available that I call the "RC1 Key-shaping chip" that produces the necessary raised cosine envelope shape with this amplifier. 5W HF PA kit: <http://grp-labs.com/pa>

Aug 2017: QCX CW transceiver kit (with WSPR mode)

In February 2017 I received an invitation to propose kits for the 2017 Youngsters On The Air summercamp buildathon to be hosted by the RSGB in UK in August 2017. I proposed a radical CW transceiver which was planned to have extremely high performance, built-in test performance, tons of features, and all at a low enough cost to meet the tight budget for the buildathon part of the summercamp event. The proposal was accepted and the "QCX" CW transceiver kit project got underway. This is a CW transceiver primarily and I won't describe all the features here which may be out of scope for this article.

But QRP Labs, and I, have a history of Weak Signal obsession so I included a GPS interface and WSPR beacon function in the QCX kit, to stay true to our roots!



I hope you have enjoyed this foray into my personal experience of Weak Signal modes starting with QRSS and its derivatives, which dominate my amateur radio experience; and how it led to the formation of QRP Labs, the QRP kit company; finally, how the Weak Signals projects and QRP Labs grew to dominate my life, now it IS my working and hobby life. I owe an indescribable debt to all the fantastic people I have met through the QRSS side of the hobby; for their help with kits, design advice, all the education I have received from the wise, for their personal friendship. For me, QRSS has been a truly life-changing experience!

**Long live the
MILLIWATTS!**

~~~~~  
If you are not already a member of the Knights QRSS group, please have a look here -

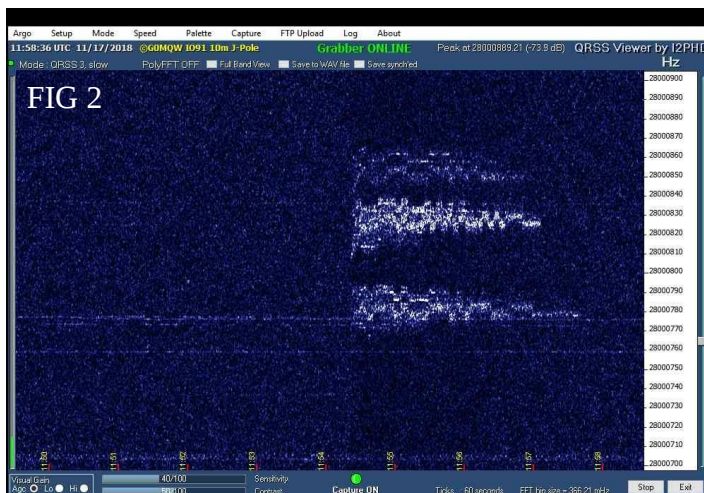
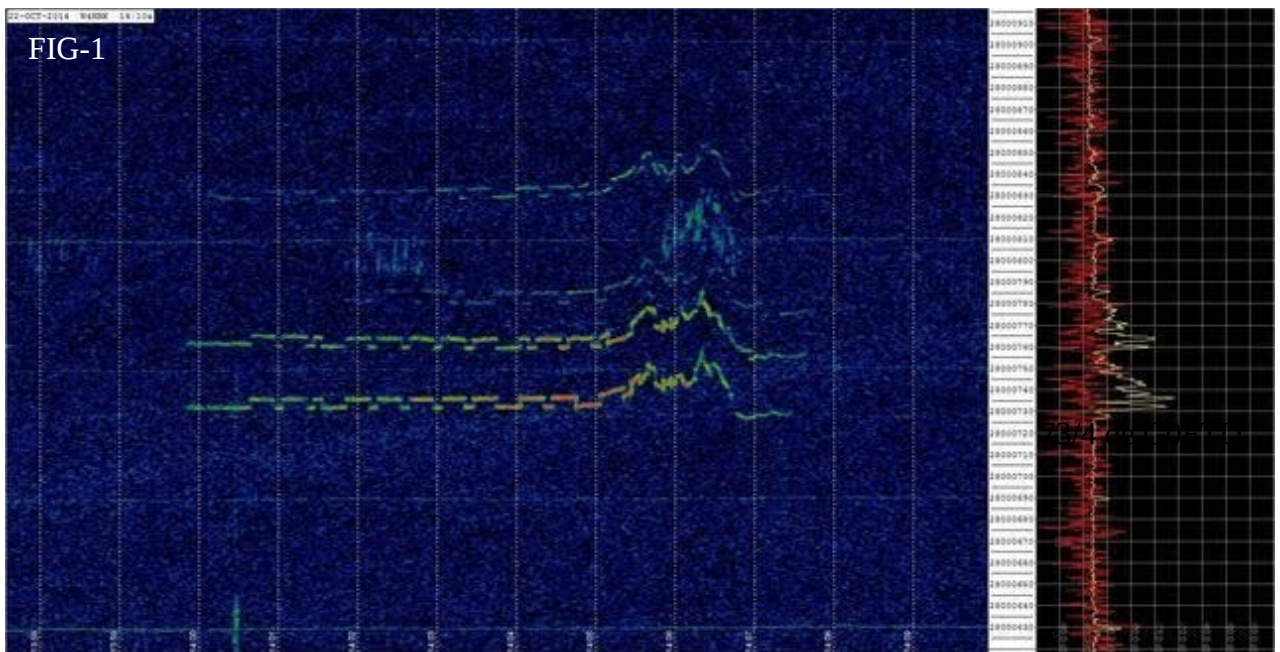
<https://groups.io/g/qrssknights>

~~~~~

My Favourite Grabs.

I tried to decide what were my favourite grabs from all the grabbers that I have seen. It was a difficult choice. But I decided upon two grabs to illustrate the power of QRSS. And that power is being able to demonstrate that not only is a propagation path available, but a visual indication of what has made the ionosphere what it is.

In FIG-1 we see a grab from Bill, W4HBK in Florida viewing UK signals. What a perfect example of showing what a large solar flare can do to 28Mhz propagation back in 2014. As the ionosphere expands, and the bottom layer expands towards Earth, the Doppler shift goes upwards. Image copyright of W4HBK.



And to the left, FIG-2 is a fantastic grab from G0Mqw. It shows a meteor scatter burst that ionised the ionosphere for several minutes in 2018. Note the **VERY** sudden onset of signals from three stations. They were (top to bottom), G0FTD, G6NHU and G0PKT in southern England. All stations were typically 80 miles away from the G0Mqw receiver grabber. All transmitter powers were typically 500mw (half a watt). - Image copyright G0Mqw. - Andy G0FTD

Building a Raspberry

Pi QRSS

Grabber – by Andy GOFIO



I suppose I was late the party when it came to getting a Raspberry Pi mini computer. But once I had one I was pretty amazed. They are certainly NOT toys, but instead they have real power. For me it was easy to get started. A UK supplier had a complete Raspberry Pi 3B+ starter kit for a mere £50. It was dead easy to set up, because there was no set up, if that makes sense. The Raspbian operating system is already on an ready made and supplied SD Card and ready to go.

So what was next ? Well I was curious to see if I could make it into a QRSS grabber system. For this I would need to get a program called QRSSPIG, which is a grabber program for QRSS running on it. Now with Linux based systems, unless the software is already available in a repository to be installed without user intervention, then you have to build from source code. It sounds scary. But once you know how it is easy. Or there is another alternative. I'll describe them both here.

The easy way. Grab a pre-compiled operating system image, such as mine and install it to your SD Card. Install does not mean copy, it means install ! The SD Card needs to have a piece of code in it to instruct the Raspberry Pi that it has to boot from the SD Card, and how to load the operating system. An SD Card needs to be 32GB maximum, and formatted to FAT32 only. I prefer to use a program called Etcher to burn the image to the SD Card.

There are straight binaries, or Java based versions of Etcher, for all operating systems. I recommend NOT using the Java ones, because more often than not, you find that the Java runtime engine on you PC ends up being a different version than what is expected for the software. Once you have burnt the SD Card image, insert the card in your RPi and switch it on. You will see some desktop shortcuts with an example to run QRSSPIG. You need to edit a QRSSPIG config file to suit your operating band, and ftp passwords etc.

There are a few different types of config file. The choice depends upon whether you are using a simple sound card dongle, a cheap SDR dongle in direct sampling mode, or an SDR dongle / receiver in "normal" mode. You can make various shortcuts to be saved on your desktop to make life easier.

Now in my case, I wanted to make a minimalist Raspberry Pi system, so that I could use it like a Swiss Army Knife computer system. To do this, I decided to add a very basic on board screen for it. There are various addon screens available from Ebay and Amazon. I chose the cheapest at £10. It simply clips onto the PCB and can mirror what you see on an external monitor. The standard Rpi case would not quite fit the 2.2 inch TFT display properly because there is an infra red LED fitted to it ! There are two options here. Remove the LED, because you don't need it. Or, I found a nice case that has enough room. It is called the PiBow Ninja and it is made of layers of Perspex. The cost was about £15.

If I am out portable then I don't want to carry a keyboard with me. I try to make my system work as much with a mouse only as I possibly can. But there are times when you might get caught out and need a keyboard. So my Pi image included an on screen keyboard too. For a PSU, I use one of those battery packs intended for use with cellphones. See FIG1.



The RPI only takes about 500ma from a 5 volt source, even with a small TFT screen added. So it should be very easy to obtain a long operating time from one of these common cellphone battery packs. Adding an SDR dongle only adds about 150ma. Less if you use direct sample mode. It is a good idea to obtain short USB cables for your interconnections to the PSU as well as any external dongles.

This is because space is tight around the four USB sockets on a Pi 3B+ model. Trying to connect a mouse and an SDR dongle or soundcard dongle usually results in poor connections or PCB damage. You have been warned. The Rpi has WIFI, and for portable use it is possible to use your cellphone as a gateway system. All cellphones should have this option available (consult your handbook). By enabling this, your Rpi connects via WIFI to the cellphone and relays you to the cellular network. That way you can upload grabs. Dead easy.

RIGHT – A CHEAP £2 USB SOUND CARD DONGLE SUITABLE FOR A GRABBER



One day you might need to compile QRSSPIG from source. Here are the instructions on how to do it. They are current as of December 2018. Learning this skill will allow you to become competent in building other software available for Linux in the future, since the procedure usually stays the same when you use cmake etc. The usual procedure is to clone a project from somewhere like Github. Find out what dependencies the project may require for it to work and install them. Then make the program using cmake compiler of similar. You may need to install these.

HOW TO BUILD A PROGRAM FROM SOURCE CODE.

1. So you have downloaded this set of files and hopefully you have put them in
home/pi/qrsspig-master

2. From the terminal window use the following commands -

sudo apt-get install cmake

sudo apt-get install libboost-program-options-dev

sudo apt-get install libyaml-cpp-dev**sudo apt-get install libfftw3-dev**

sudo apt-get install libgd-dev

sudo apt-get install librtlsdr-dev

sudo apt-get install libhackrf-dev

sudo apt-get install libasound2-dev

sudo apt-get install libssh-dev

sudo apt-get install libcurl4-openssl-dev

sudo apt-get install libliquid-dev

sudo apt-get install librtfilter-dev

(The last two depend upon how up to date your OS is, but try them anyway, it doesn't hurt). Time to build the program.. From the command line you need to navigate to where you unzipped the source code, which should be home/pi/qrsspig-master, so you type cd home/pi/qrsspig-master to get there.

Then type -

mkdir build

cd build

cmake ..

make

sudo make install

Hopefully you should get no error messages and your copy of QRSSPIG has now been turned into a binary file and installed for you.

I recommend that you create in your home/pi/ directory that you create a two folders called qrsspigconfigs and qrsspigjpgs and note that they are case sensitive. This is so that you can easily edit your config files without permission errors, and view the grabs that it produces. Plug a sound card dongle into any USB socket and QRSSPIG will probably see this is hw:1 device. Use the command qrsspig -l(list) to check what it can see. Edit the sample config file with your passwords for ftp'ing your grabs to your website and have fun !

To use QRSSPIG, type the following -

```
qrsspig -c /home/pi/qrsspigconfigs/yourconfigfile.yaml <enter>
```

If all goes well, you just see some text about QRSSPIG configuring itself and then running. You also see an Ascii audio level meter, adjust your audio to a suitable level. At the end of every 10 min segment, 0-10 10-20 20-30 etc you will see a message that a file called hf1.jpg has been pushed and uploaded to your website. That's it. You do NOT see anything else. There is no screen build up as the signals are received. You just see each 10 minute grab created. Be aware that a yaml file needs EXACT formatting. Spaces have to be exact.

The ALSA base frequency is your USB dial frequency. The usual way to set up a config is that the ALSA base frequency is 1.5Khz lower than the centre of the QRSS band that you want to monitor. Easy when you know how. I must admit that I struggled at first, but thanks to Onno PA2OHH and Martin (Mr QRSSPIG), and their excellent advice on compiling, I was able to get it all working, Thanks guys ! Get stuck, just send a message to the Knights QRSS group. **GENERAL NOTE - NOT RPi related. If you use a desktop Linux like Debian or Linux Mint, then QRSSPIG is available for automated install via the software manager or through the command line "sudo apt-get install qrsspig"**.

Link below for a pre-built Raspberry Pi image containing QRSSPIG and other radio based software tailored for the radio amateur. It is called YAPI (Yet Another Pi Image).

https://drive.google.com/drive/folders/1ZBY-K8tzq961PldXG_sQ7YTUvRFrhiak

The tail end



Well folks I hope you enjoyed this little compilation. Maybe print it out and give to a friend? Despite writing many articles, this is only the second time I have tried to put together a compilation. The first time was in about 1979 and cost 10p a copy. Let us look forward to 2019. I predict more animated grabbers, and more grabbers generally. We might need a little more TX power in the next 12-24 months but generally we should be OK. ***We are not here to DX, but to prove if a path exists with appropriate power. We have at least a 20db advantage over SSB, so to prove the same point we only need 100x less power. Please remember that.*** Once the solar cycle starts again about 250mw will be PLENTY. I also wonder if some clever person will be able to record 10 min samples from online SDR's and then quickly process them into grabs ? That would increase our span too. Hopefully the 80m band split will sort itself out too. And please remember that all images and text in this publication are copyright of their respective owners. And with that, I now bring to a close the Knights QRSS Winter Compendium 2018. May your traces be bright ! - 74 de Andy GOFTD