

# RADIO MAGIC

Robert C. Mazur, VA3ROM | [www.va3rom.com](http://www.va3rom.com) | [va3rom@gmail.com](mailto:va3rom@gmail.com)



*First published in the Jan-Feb 2021 issue of The Canadian Amateur*

## ISM Band “HiFER” Beacons: Part 2

*Special thanks to Robert “Bob” Confrey, WA1EDJ, John Davis, AE0CQ, and members of the Longwave Club of America (LWCA) for their invaluable help.*

### “The Best Laid Plans...”

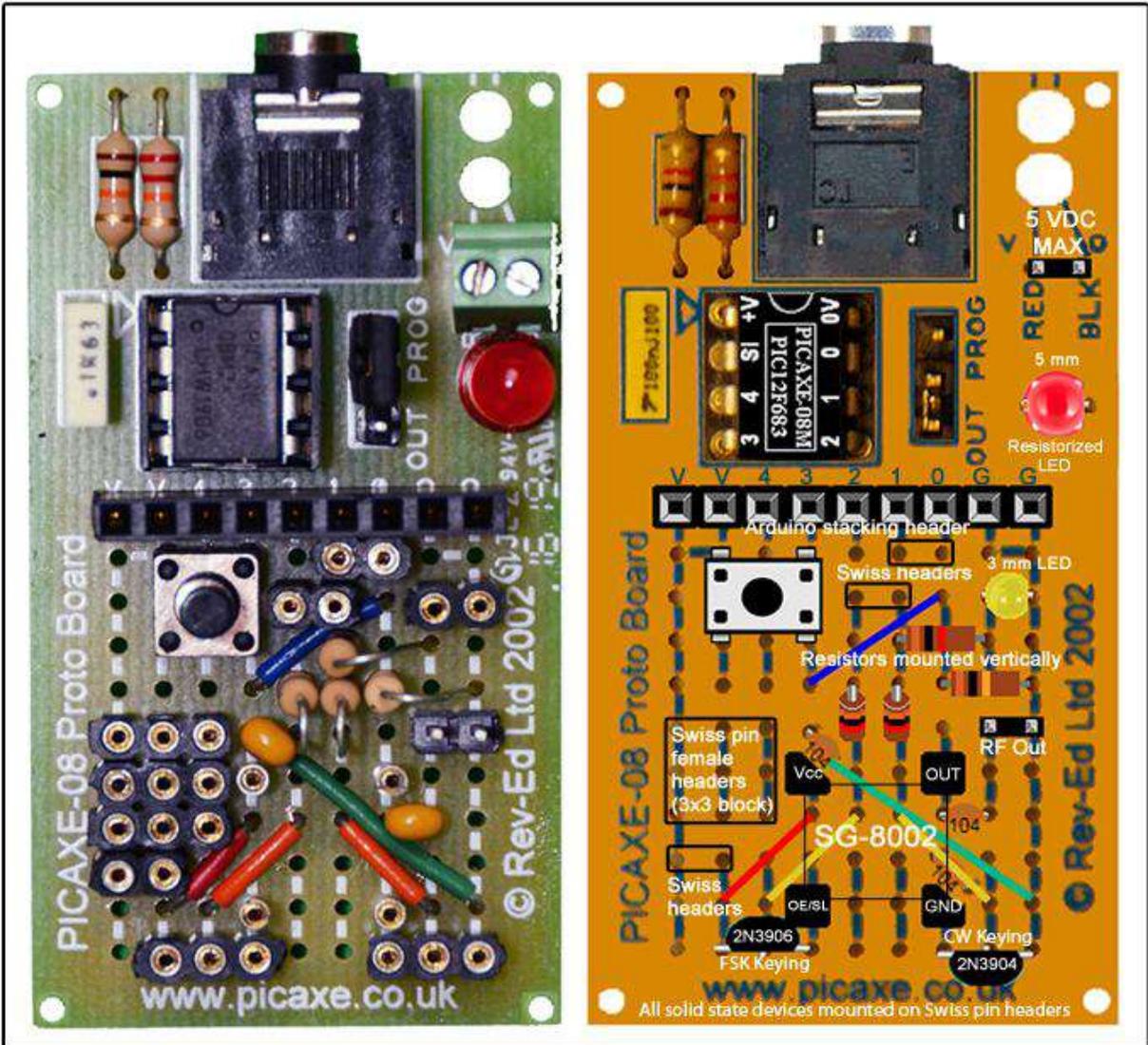
So, I designed and built my Arduino Uno  $\mu$ -controller, kitchen sink included, 22 metre (m) industrial, scientific and medical (ISM) band telemetry/propagation beacon deluxe gadget (see part 1). It worked great during home tests, transmitting its 4.6 milliwatt (mW) radio signal well over 100 metres (m) using a short piece of wire for an ersatz antenna. The Morse code continuous wave (CW) telemetry was copied and decoded by ear using a portable shortwave receiver. Great! All should work as well when it was up at my camp from where I could remotely monitor its sensor readings back at home. The high frequency experimental radio (HiFER) transmitter was installed inside a small shed behind my RV trailer and connected to a resonant, low-to-the-ground, inverted-V dipole. I also setup a separate SOTABeams WSPRlite 5 mW, 20 m weak signal propagation reporter (WSPR) transmitter connected to a ground mounted vertical antenna for side-by-side signal spotting and comparison. The Uno gadget transmits standard on/off keying (OOK) CW alongside the separate WSPR propagation beacon, technically “mode 2” for its 2-minute transmit time, which uses four frequency shift keying (4-FSK). Both ground wave signals were received and decoded during the bush road drive out to the main highway turnoff (3 km), and I assumed that they would still be both heard, albeit weakly in town, only 10 km as the crow flies.

But Murphy reared his ugly head because both ground wave signals petered out somewhere along the highway drive home plus I had unknowingly picked the very congested ISM band frequency of 13.55560 megahertz (MHz). In this “neighbourhood” are a plethora of consumer and commercial ISM devices all merrily chirping back and forth so good luck trying to find my signal in the din even if it was there! That left me to somehow find the transmitter’s component CW skywave signal, which shot straight up to the heavens like an arrow fired into the air to land “I know not where...”

### **The Better Laid Plans**

First, the transmitter had to be moved to another 22 m ISM band frequency. Bob, WA1EDJ, suggested using the inexpensive, one-time programmable Epson SG-8002DC radio frequency (RF) oscillator integrated circuit (IC) and getting several with frequencies located within the 22 m HiFER beacon so-called “watering hole” segment (13.555300 to 13.555500 MHz). This region is used by radio hobbyists experimenting with QRSS (sending really, really slow) modes 3 or 6 (dit times in seconds), usually 2-FSK CW, OOK CW and 4-FSK WSPR2. I also reread my past “QRSS CW” articles to refresh my memory (Jan-Feb, Mar-Apr and May-Jun 2017 *TCA* refers). With QRSS6 FSK CW, the carrier is shifted up about 4 to 5 hertz (Hz) to transmit the “mark” and shifted back down to transmit the “space”. As an aside, Samuel Morse used these two terms to describe the output of the Morse register, which was a clock driven (ticker) paper tape inker/printer creating short/long marks/spaces of incoming landline Morse code characters. Bob also suggested informing the LWCA group about my beacon and its encoded telemetry and decoding methodology for feedback via the LWCA message board. *Note: Your receiver’s dial frequency must be tuned to 13.553900 MHz upper sideband (USB) for decoding signals using digital signal processing (DSP) software such as “Argo” for QRSS decoding and “WJST-X” for decoding WSPR.*

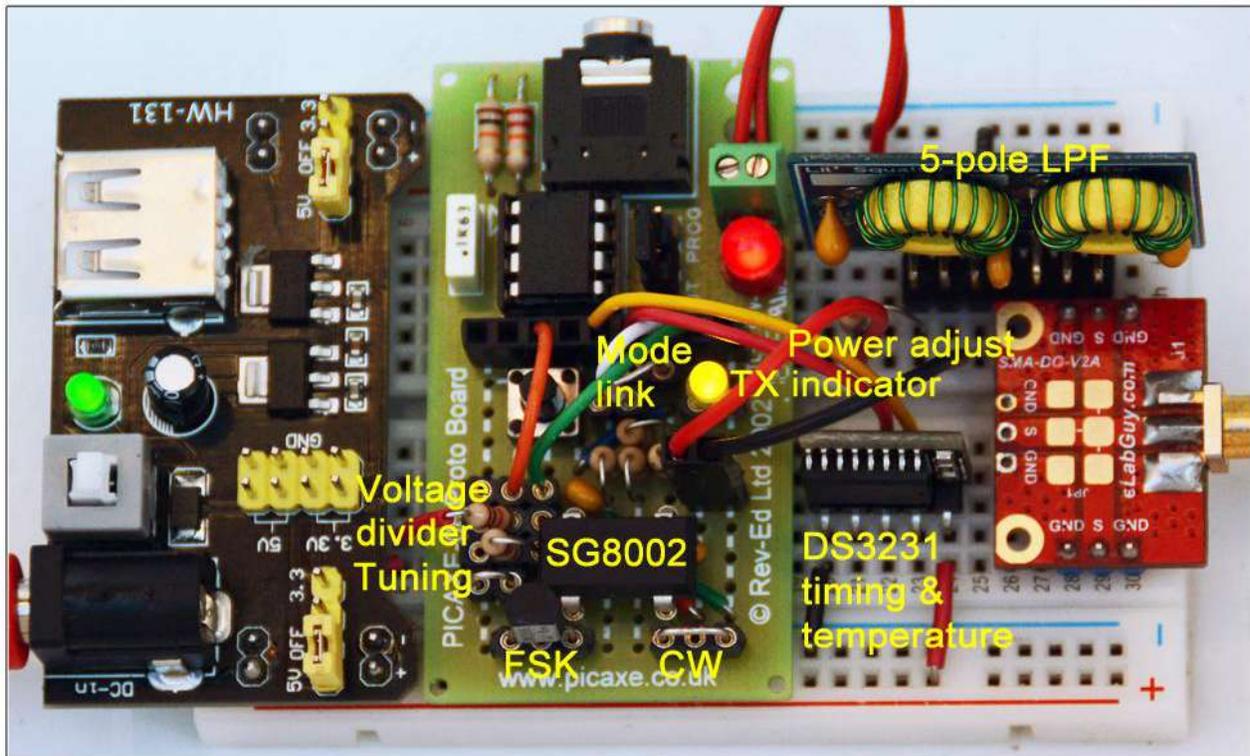
After perusing my old QRSS CW articles, I decided to redesign the transmitter, and also take a more minimalistic approach using a PICAXE-08M2+  $\mu$ -controller built on the ingeniously designed Revolution Education protoboard (see Figure 1A, next page).



**Figure 1A: PICAXE Protoboard Transmitter Layout**

Left is a picture of my PICAXE protoboard HiFER transmitter (version 2) before installing the solid-state devices and jumpers. Right image is the protoboard layout created with the PICAXE electronic bread board layout emulator ("PEBBLE"). All wiring is on top, and Swiss machine pin female headers enable quick swapping in/out of components.

Figure 1B (next page) depicts the solderless breadboard working prototype. This was used for the "proof of concept" testing and tweaking.



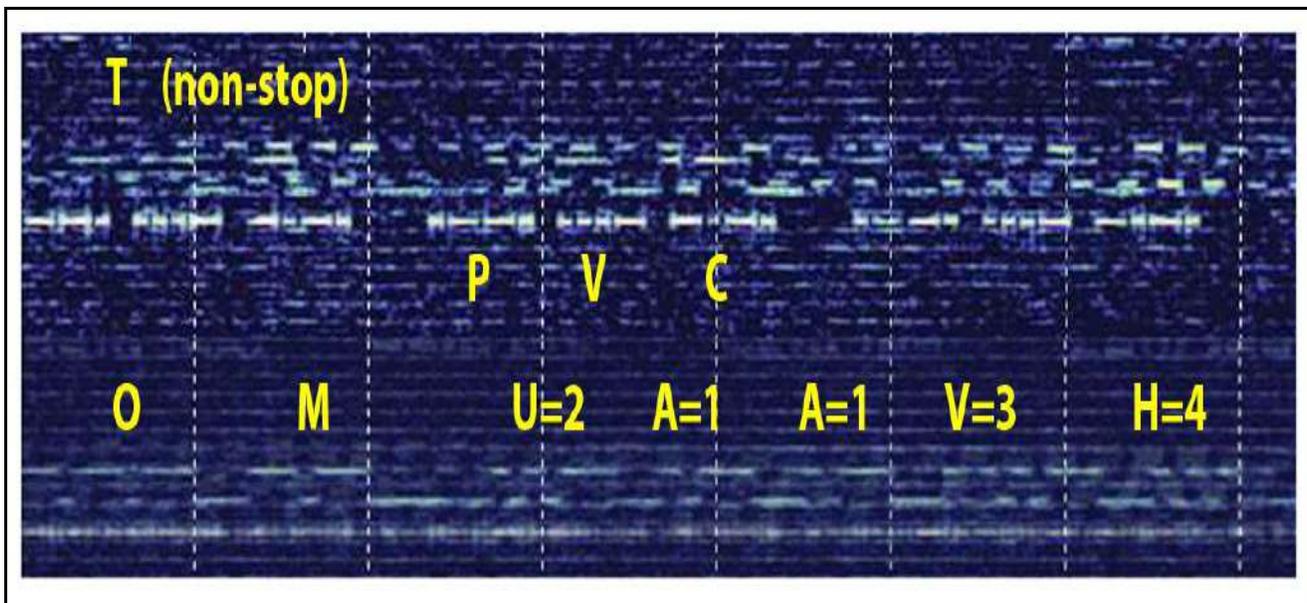
**Figure 1B: PICAXE HiFER Transmitter**

My HiFER telemetry transmitter prototype built on half-size solderless breadboard. It's designed to be field programmable and frequency flexible by just changing the LFP and SG-8002DC RF oscillator.

The PICAXE-08M2+ is a small 8-pin  $\mu$ -controller, powerful, self-contained, plus it has support code ("libraries") for commonly used external sensors/devices inside the chip. It also allows for user selectable processor speeds (to 32 MHz), with up to four separate (parallel processing) programs running simultaneously in memory, plus it works with direct current voltages (VDC) as low as 1.8 VDC.

Because local monitoring of my beacon wasn't possible, and the nearest internet remote receivers were over 600 km away, a conventional speed, very low power (QRPP) CW signal would be buried in the background atmospheric noise and very difficult to copy by ear even if I could find it by randomly selecting remote receivers. It was like looking for a microscopic RF needle somewhere in an infinite RF haystack!

And so I changed the mode to QRSS6 using 2-FSK CW for its far superior weak-signal readability (literally). Doing this dramatically *decreases* its signal bandwidth and *increases* its power spectral density (PSD) in watts-per-hertz (W/Hz) to the equivalence of a WSPR2 signal, which is about 5000 times stronger than an SSB signal (2500 Hz reference bandwidth). The telemetry packet was also reduced to just the bare essentials to fit inside the standard 10-minute QRSS transmission window or frame: ISM call ("ROM"), ambient air temperature and solar/battery voltage levels. This alternates once an hour with my generic grid square (EN58) to identify the signal's origin for others monitoring it for propagation purposes. Transmission time is increased (on average) to about 8 minutes as compared to around 30 seconds using 12 words-per-minute (wpm) OOK CW. But this is the quid pro quo the universe and Shannon-Hartley theorem demands we pay—the data rate must slow down. However, unlike WSPR and other "all or nothing" weak-signal data modes, QRSS users can get some "change" back by using an astrophotography technique called frame "stacking" to remove more noise, which Shannon defines as anything that interferes with the desired signal (including other signals). But that's a technique beyond the scope of this article. See Figure 2.

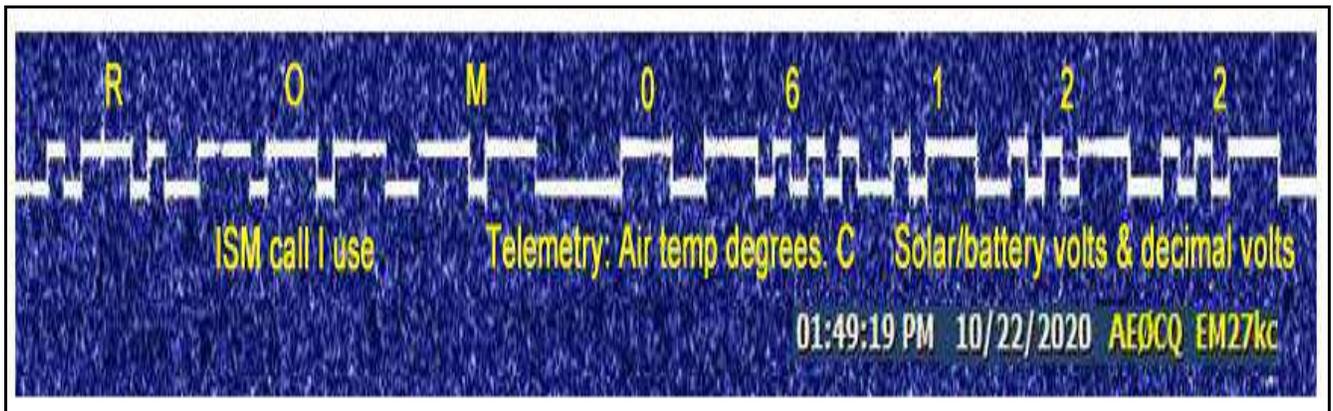


**Figure 2: Frame Stacking Example**

I unknowingly picked a popular HiFER frequency area and collided with "T" and "PVC" signals (top image). But because my transmissions use 10 minute transmission time synchronized frames and the others don't, stacking several successive frames ("grabs") reinforced my signal and reduced them as "noise" (bottom image). *Note: Morse code "cut" numbers used for my telemetry transmissions.*

### “Oh where, Oh where...”

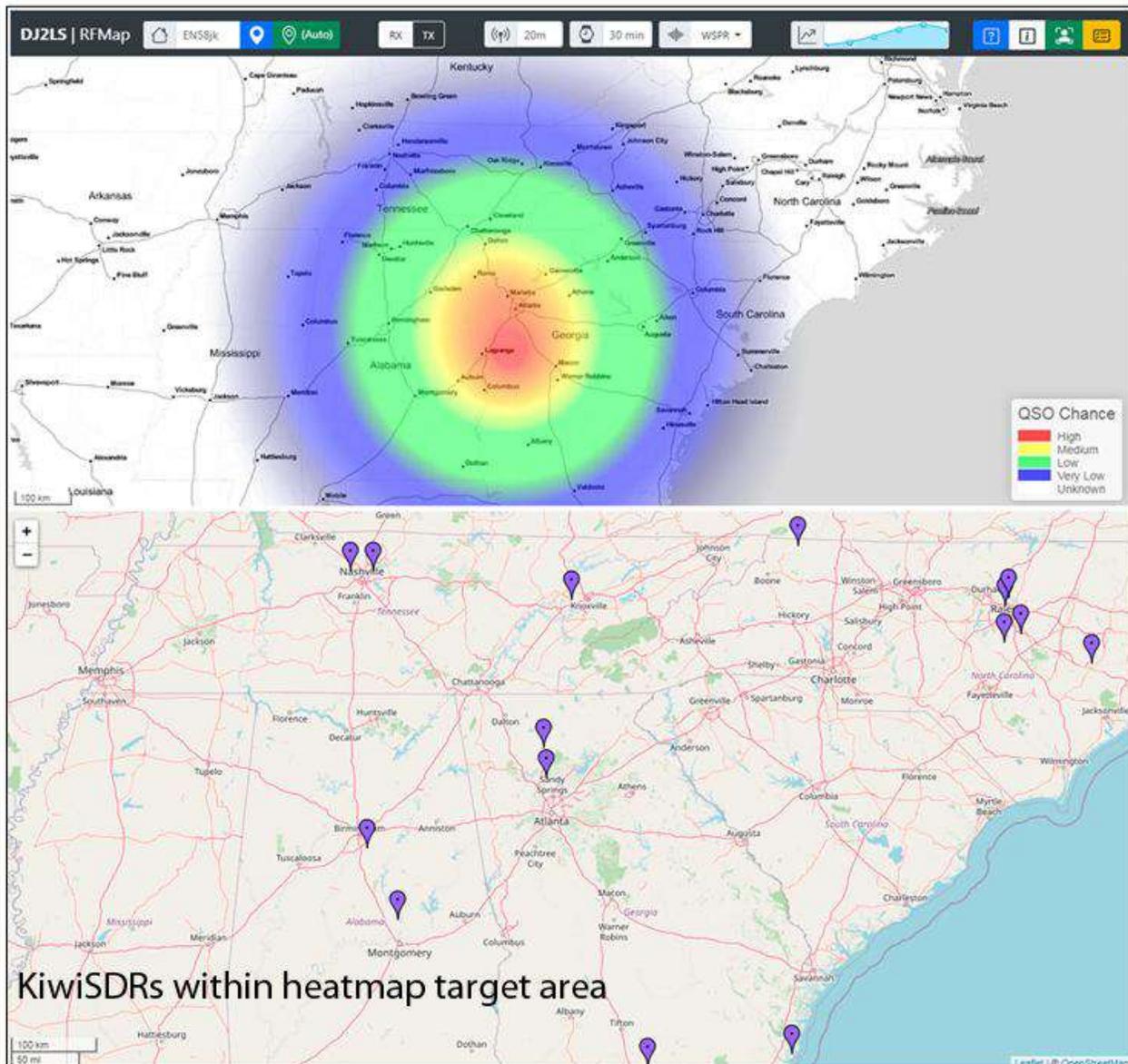
Okay, I have to stop here and backtrack a bit because I’ve put the radio cart before the horse. I have to tell you exactly how I found the various remote receivers to use in the first place. While the LWCA members might spot my QRSS6 signal and pass on the telemetry via the message board or direct email (see Figure 3), or I could use Scott Harden, AJ4VD’s QRSS Plus “grabber” website, I still wanted to find and decode my own dog darn data whenever during the day and not always have to rely on someone else.



**Figure 3: Someone Else Can Hear Me!**

LWCA member John Davis, AE0CQ (1358 km south southwest from me) posted this 22 m screen grab of my telemetry beacon using Argo. Credit: AE0CQ.

Serendipitously, last summer, I found Simon Lang, DJ2LS’ internet based data modes RF “heat” map. See Figure 4, next page. Simon’s system displays data mode transmitted and received signals within Maidenhead grid square locators on a global map showing their relative signal strength “heat” or intensities. This awesome software tool helps narrow down the area(s) where my WSPR “bird dog” signals were landing by using different colours for their different intensities, which, in turn, makes it easier to find the nearest remote receiver to use—in my specific case—using the KiwiSDR (software defined radio) receivers map. I do it this way because QRSS transmitted signals aren’t spotted on Simon’s heat map, but I can use my 20 m WSPR beacon spots data extracted from the global WSPRnet web server on the heat map to select any available internet remote public access KiwiSDR in or near the “hot” WSPR spot areas. I can also use my home station’s received WSPR spots of other stations for the same purpose.

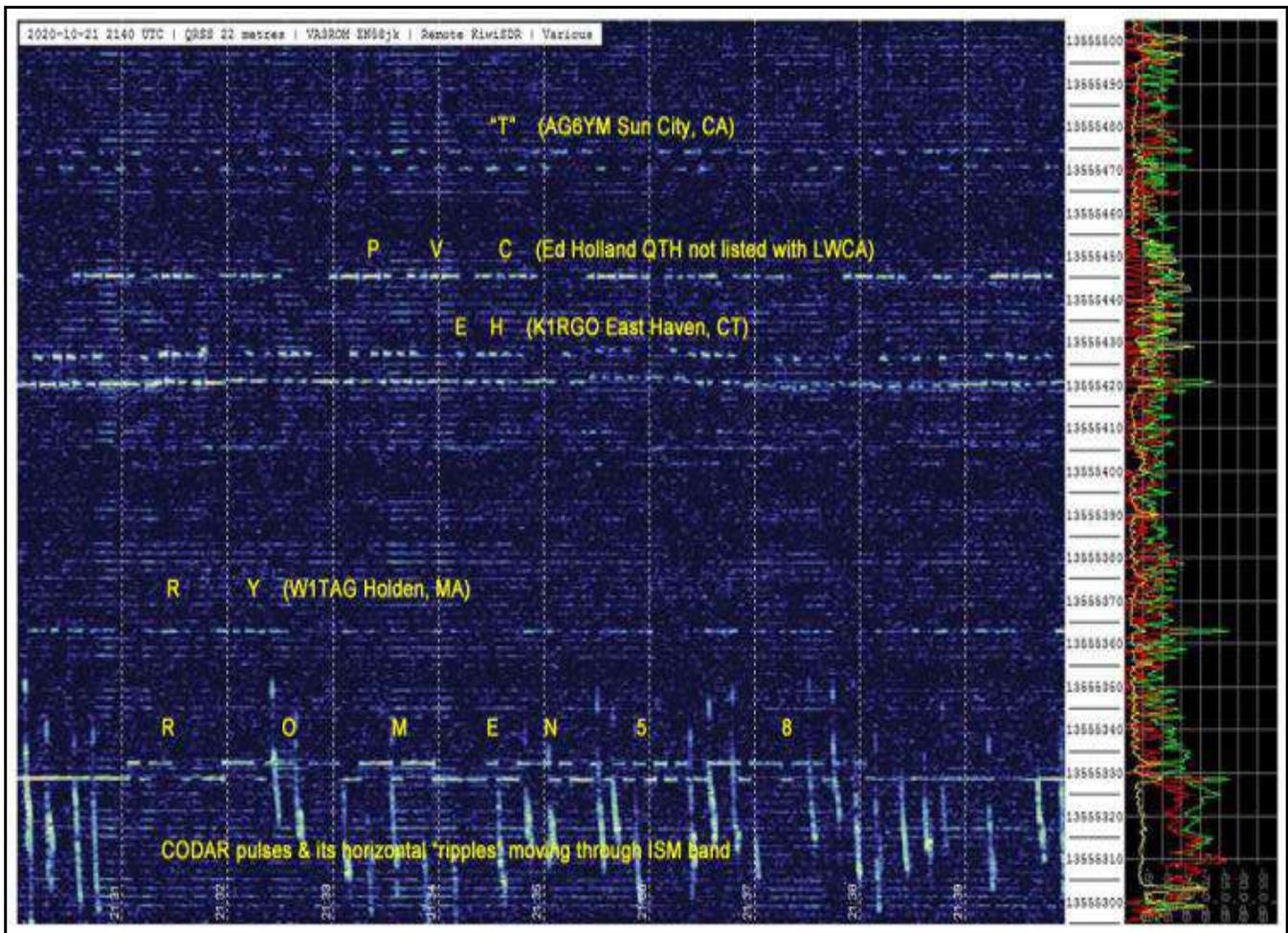


**Figure 4: DJ2LS Heat Map and KiwiSDR Maps**  
 Composite of DJ2LS' heat map spots of my station's received WSPR signals and the public KiwiSDR site map.

**Observations**

I'm very pleased with the results so far. I've learned a lot about this strange very low power world where things happens just because it can! And it's so very cool monitoring a "flea power" beacon you built from scratch that can shoot radio arrow signals across the continent and perhaps Atlantic and Pacific by summer 2021—hopefully—as if by magic.

- a. From thousands of decodes using various remote KiwiSDRs, I'm very confident that the QRSS6 and WSPR2 FSK modes have equal propagation behaviours and performances when transmitted in the same time frame, with equal transmitter power, similar antennas and wavelengths. If I can find the one, I can find the other in the same general area(s). Only the decoding method is different.
- b. Avoid using coaxial ("coax") cable feed lines with insufficient shielding and cheap connectors (think Belden and Amphenol). And keep the feed as short as possible.
- c. Avoid using loading coils, traps and/or antenna tuners because they add losses that we can't afford at the sub-5 mW RF power level.
- d. A common-mode current choke prevents RF current from flowing back down the *outside* of the coax shield (I use four snap-on ferrite chokes). We want our miniscule signal radiating from the antenna not from the coax!
- e. Avoid 22 m band ISM frequencies anywhere near 13.55600 MHz!
- f. Use the 22 m ISM watering hole, the LWCA message board and DJ2LS' heat map.
- g. Set up a separate WSPR receiving or transmitting station to help locate your QRSS signals. You'll also provide data via the WSPRnet for others to use.
- h. The bane of coastal radio hobbyists is the high frequency (HF) Coastal Ocean Dynamics Application Radar (CODAR) system, which is used for near-surface ocean wave monitoring (wave height, speed/direction, tsunami alerts, etc.). It transmits frequency modulated (FM) USB rapid CW kilowatt pulses. One well-used CODAR frequency is 13.500000 MHz. Ouch! (Figure 5, next page).
- i. From what I've been observing, even before the current solar cycle began its rebirth, 22 m QRPP FSK CW radio waves seem to be able to enter in and travel great distances through the lower ionosphere's plasma jet stream like "corks" caught in a magical and majestic sky "river" bobbing along whichever way the current carries it.



**Figure 5: Signals Decoded Using Remote KiwiSDR**

Example of decoded 22 m ISM band multiple QRSS signals using VE7GL's remote KiwiSDR also showing CODAR pulses on top of my miniscule signal, which holds its own against the kilowatt pulses! Station identification information is courtesy of the LWCA. I'm using Spectrum Laboratory, which is a powerful and complex DSP software package. Note: Morse "cut" numbers are used to so encoded telemetry fits inside a 10-minute transmission frame except when transmitting my grid square (EN58).

## My Final

Part 3 looks at the final transmitter design, battery/solar power supply, external sensors, PICAXE control code, and the setup and use of remote KiwiSDRs along with using the free digital signal processing (DSP) soundcard decoding software.—73

## References and Resources

Longwave Club of America (LWCA) [www.lcwa.net](http://www.lcwa.net)

Argo <http://www.weaksignals.com>

WJST-X <https://www.physics.princeton.edu/pulsar/k1jt/wsjsx.html>

Spectrum Laboratory <https://www.qsl.net/dl4yhf/spectra1.html>

Shannon-Hartley theorem [https://en.wikipedia.org/wiki/Shannon-Hartley\\_theorem](https://en.wikipedia.org/wiki/Shannon-Hartley_theorem)

QRSS Plus <https://swharden.com/qrss/plus>

DJ2LS RF "heat" map <https://maps.dj2ls.de>

KiwiSDR receivers map <http://rx.linkfanel.net>

WSPRnet <http://www.WSPRnet.org>

### Parts Sources

RobotShop <https://www.robotshop.com/ca> is a great source for PICAXE products. If you're just getting started order the PICAXE-08M USB Starter Pack. It's a great deal especially for the HiFER transmitter project. The Revolution Education PICAXE store <https://www.picaxestore.com> provides a wider variety of PICAXE products not normally stocked by RobotShop.

An excellent source for magnet wire, toroids is Kits and Parts <https://kitsandparts.com>. Additional components can be obtained from Universal Solder <https://www.universal-solder.ca>.

Pre-programmed SG-8002DC's are available from Digikey <https://www.digikey.ca>. These are SGR-8002DC-SHB-ND XTAL OSC PROG XO CMOS 5V 50PPM <https://www.digikey.ca/en/products/detail/epson/SGR-8002DC-SHB/755257>. You add each one individually to your shopping cart with a note telling Digikey what frequency you wanted programmed from 1 MHz to 125 MHz.

### **Free Design and Programming Tools**

PICAXE Programmer v6.0 and PEBBLE <https://picaxe.com/software>.

LTSpiceVII <https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html#>

Iowa Hills RF Designer <http://www.iowahills.com/index.html>

Elsie <http://tonnesoftware.com/elsie.html>