# All Things Digital

Amateur Radio for the 21st Century

028


**Robert C. Mazur, VA3ROM**

E: va3rom@gmail.com

W: http://www.va3rom.com



***First published in the March-April 2017 issue of The Canadian Amateur***


## QRSS CW—SENDING SLOW CONTINUOUS WAVE: PART 2

### INTRODUCTION

During my QRSS research, several web articles were found using PICAXE microcontroller units (MCUs) and AD9850 direct digital synthesis (DDS) modules to build simple QRSS and other data mode transmitters (beacons)—see Figure 1 A. While you can use other MCUs, most are “overkill” for building basic beacons the PICAXE is a simple but elegant platform is specifically designed for hobbyists who aren’t hardware specialists, RF designers, or expert computer programmers.

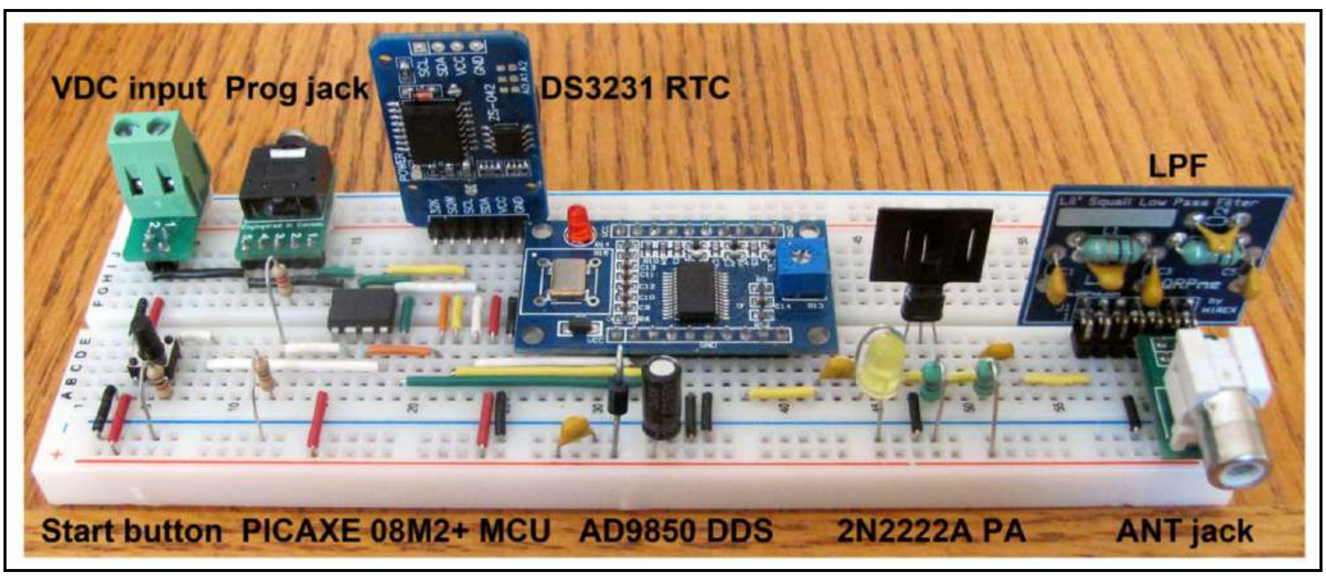



**FIGURE 1A: TRANSMITTER BREADBOARD LAYOUT**

## BUILD A PICAXE DDS CW TRANSMITTER

The PICAXE is a good choice for this specific task because it runs at one-quarter the clock speed of the Arduino Uno, uses one-eighth of the memory, and draws only one-fifth of the current. Figure 1A (see previous page) is my operational prototype QRSS and weak signal propagation reporter (WSPR) transmitter, Figure 1B is the simplified block diagram, and Figure 1C is the pictorial/schematic.

FIGURE 1B: TRANSMITTER BLOCK DIAGRAM

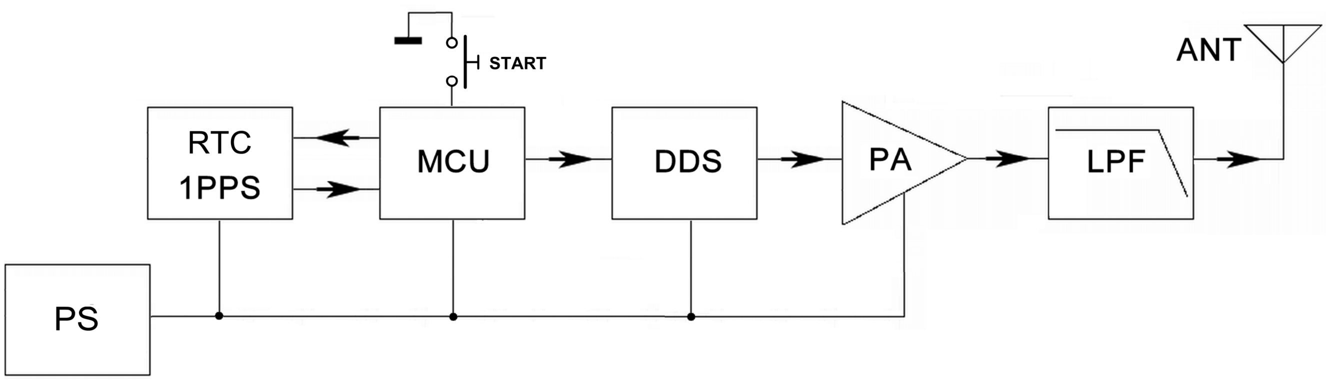
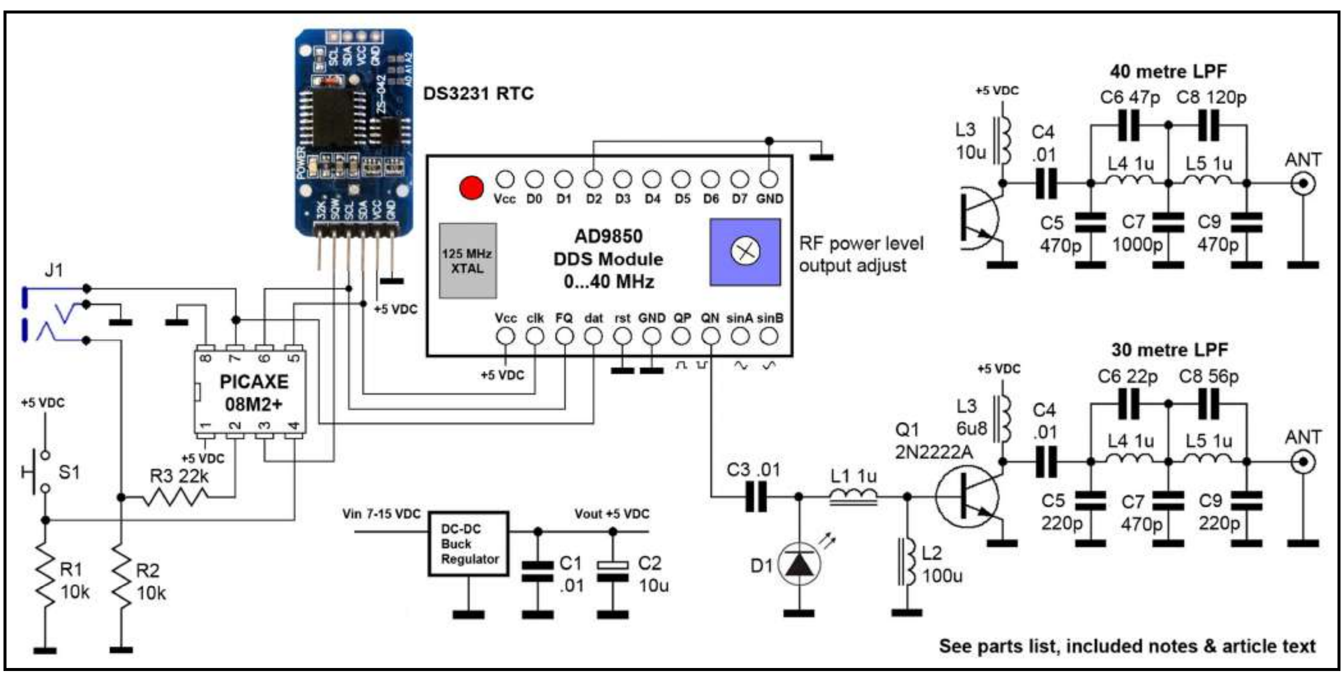


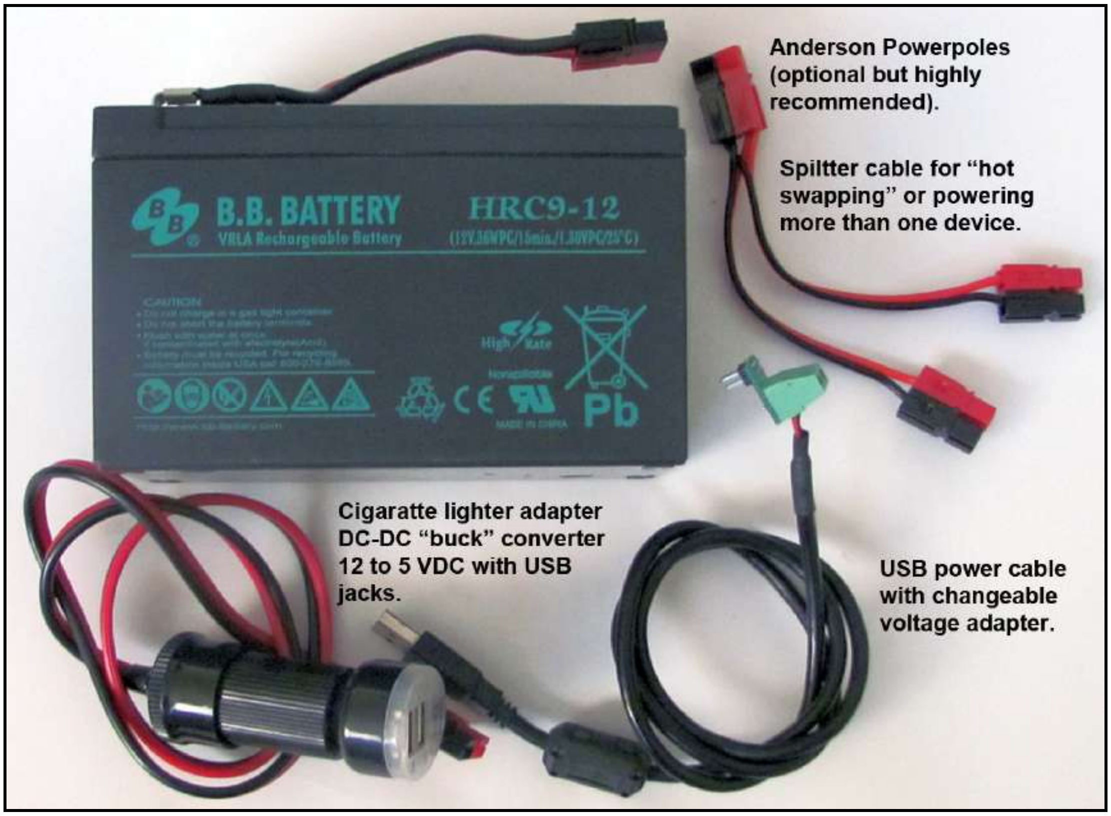

FIGURE 1C: TRANSMITTER PICTORIAL/SCHEMATIC



The building blocks consist of:

1. Direct current (DC) power supply (PS)
2. Real-time clock (RTC) module
3. MCU
4. DDS module
5. Very low power (QRPp) power amplifier (PA)
6. Lowpass filter (LPF)
7. Antenna system (ANT)

FIGURE 2: TRANSMITTER PORABLE POWER SUPPLY




## CIRCUIT DESCRIPTION (SIMPLIFIED)

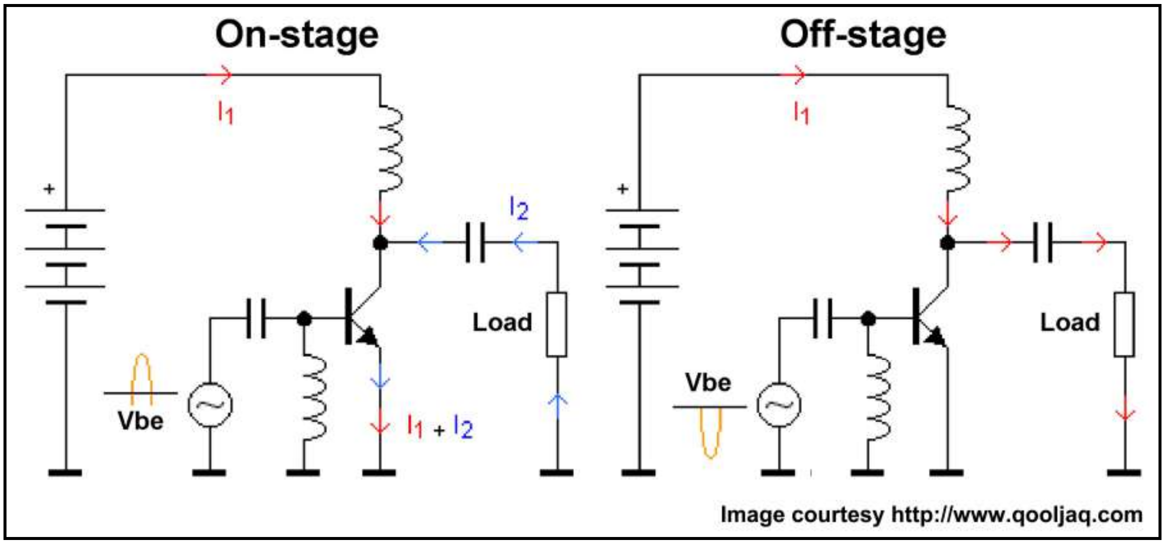
1. The entire circuit operates from 5 volts DC (VDC). For portable or field work, 12 VDC sealed lead acid (SLA) batteries are used with 12 to 5 volt DC-DC "buck" switching PS regulators (see Figure 2). These are physically small and light, and generate very little heat (lost energy) because they are extremely efficient (> 90%).

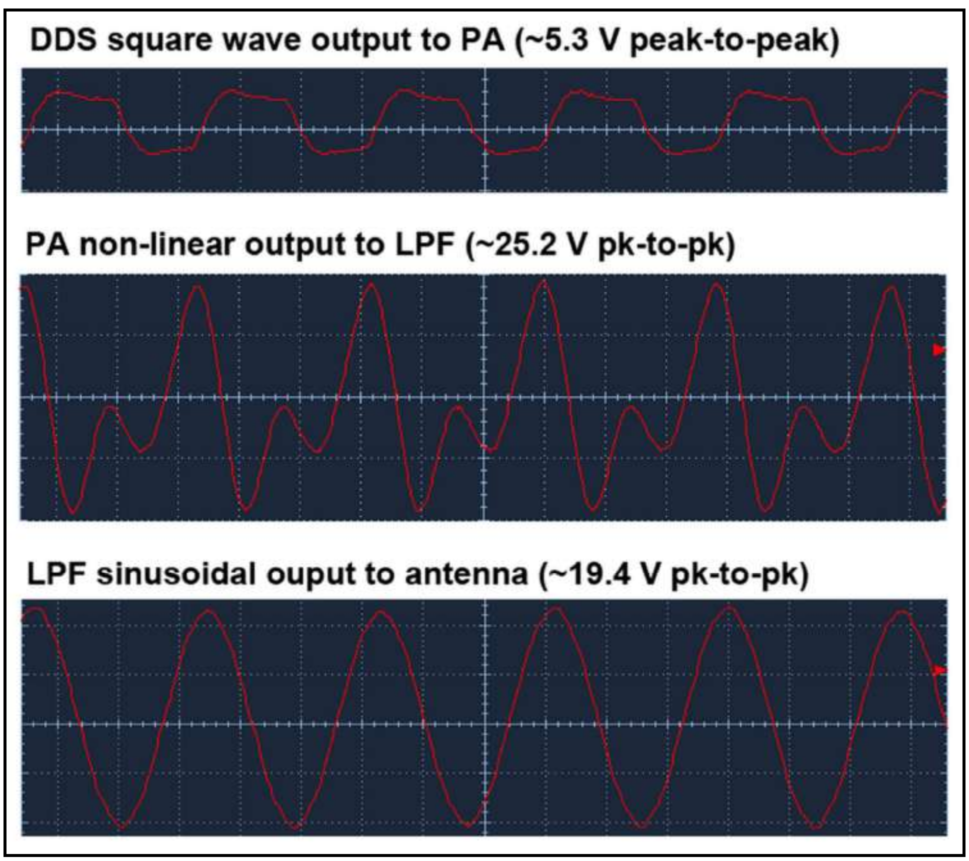
This portable PS provides enough “juice” for days of beaconing; adding a solar panel plus charge controller extends operating time almost indefinitely. *Note: If using a 6 VDC battery pack (AA cells or similar), insert a series power diode (1N4001 type) to reduce the voltage by 0.7 VDC.* ***NB: DO NOT EXCEED 5.5 VDC TO POWER THIS CIRCUIT (THE PIXAXE’S MAXIMUM OVER VOLTAGE LIMIT)!***

2. A DS3231 RTC module provides a 1 pulse-per-second (PPS) square wave used for accurate beacon timing and control. This is an easy to use, cheaper, and lower current consumption alternative over using a global satellite positioning system (GPS) device with an active (powered) patch antenna. A temperature controlled crystal oscillator (TCXO) gives the DS3231 good stability with a low parts-per-million (ppm) oscillator timing error (about 2 minutes per year).
3. A PICAXE MCU (08M2+) controls everything. For convenience, “on-the-fly” programming can be added and done in circuit, or you can use a separate breadboard programmer. *Note: A normally open (NO) pushbutton is used to start the beacon sequence on a whole minute (even or odd, as required).*
4. A DDS module (AD9850) generates both square and sinusoidal radio frequency (RF) waves, but we only use one of its square wave output to “hard key” the transmitter on and off. The AD9850’s onboard (blue) potentiometer is used as a variable transmitter drive control to produce nearly 500 milliwatts (mW) output from the PA (depending on the frequency).
5. With reference to Figures 3A and 3B (see next page): The PA (2N2222A) is designed as a non-linear, self-biased, class C amplifier. Non-linear means the output signal doesn’t faithfully represent the input signal (it’s deliberately distorted), but this is simpler and more efficient over using a linear type amplifier (required for voice or variable amplitude signals). Self-biasing means transistor Q1 only conducts (switches on) when the DDS RF square wave drive output exceeds 0.7 VDC (exceeding the diode barrier voltage of Q1), and more than half the DDS RF output appears to be “lost”, or so it seems. But when Q1 switches on and conducts current, inductor L3 builds up energy (like a flywheel) as current surges through it, and when Q1 switches off, L3 releases this energy as an inductive back current into the LPF ringing it like a bell, which keeps the RF signal going in the opposite direction!

6. A band specific LPF has an alternating RF current continuously flowing through it during a CW transmission. It sharply attenuates harmonics to produce a more sinusoidal RF signal output at the antenna.
7. The antenna system can be any type so as long as it presents a 50 ohm resistive load for the PA and an antenna tuner may be required to do this.

**FIGURES 3A AND 3B: CLASS-C RF AMPLIFIER AND LPF OPERATION**



## SOME DDS CAVEATS

Frequency stability is an absolute must when using narrow-band data modes so any DDS "jitter" (tiny, unwanted deviations) must be minimized. To do this, the AD9850 master clock oscillator (MCO) crystal requires a small heatsink seated on top, but it doesn't have to be metal because even a small plastic poke chip works just as well. (Tip courtesy of Hans Summers, G0UPL). The electronics should be housed inside a metal enclosure to provide RFI shielding, and also trap heat produced to keep the ambient air temperature around the DDS's MCO constant. Once circuit components reach thermal equilibrium (after about 30 minutes) the frequency should stabilize to within +/- 1 hertz (Hz).

Using a well-regulated, low noise DC power supply is very important because every circuit stage is connected to and dependent upon it, so we don't want any voltage variation during the lengthy QRSS transmissions (several minutes long). Additionally, any alternating current (AC) ripple (noise) mixing in with the DC voltage isn't a good thing.

Finally, crystal controlled oscillators are never spot-on frequency because of ppm errors caused by the individual quality and cut of the crystal ("rock"), but most DDS module RF uncorrected errors are usually within +/- 100 Hz. However, even this is too much because the QRSS/WSPR sub-bands are only 200 Hz wide, and large frequency errors can easily shift your beacon well above or below their respective sub-bands.

## PROGRAM DESCRIPTION (SIMPLE)

Revolution Education, the company behind the PICAXE, created a new and free object oriented programming platform called "Blockly", which novice/student programmers can use to easily and quickly create PICAXE programs (see Figure 4, next page). Blockly has a huge advantage over other MCU platforms because it supports internet "cloud" programming using the Chrome browser, and it also runs on Google Android and Apple iOS smartphones and tablets providing cross-platform computer support (Windows, Linux, and Apple). Once you have a working program, Blockly can convert the blocks to standard PICAXE BASIC language code that you "flash" (load) to the MCU. Old-school programmers (like me) prefer to use the PICAXE "Editor 6" (Windows only) or "Axepad" (cross-platform) independent development environments (IDEs).

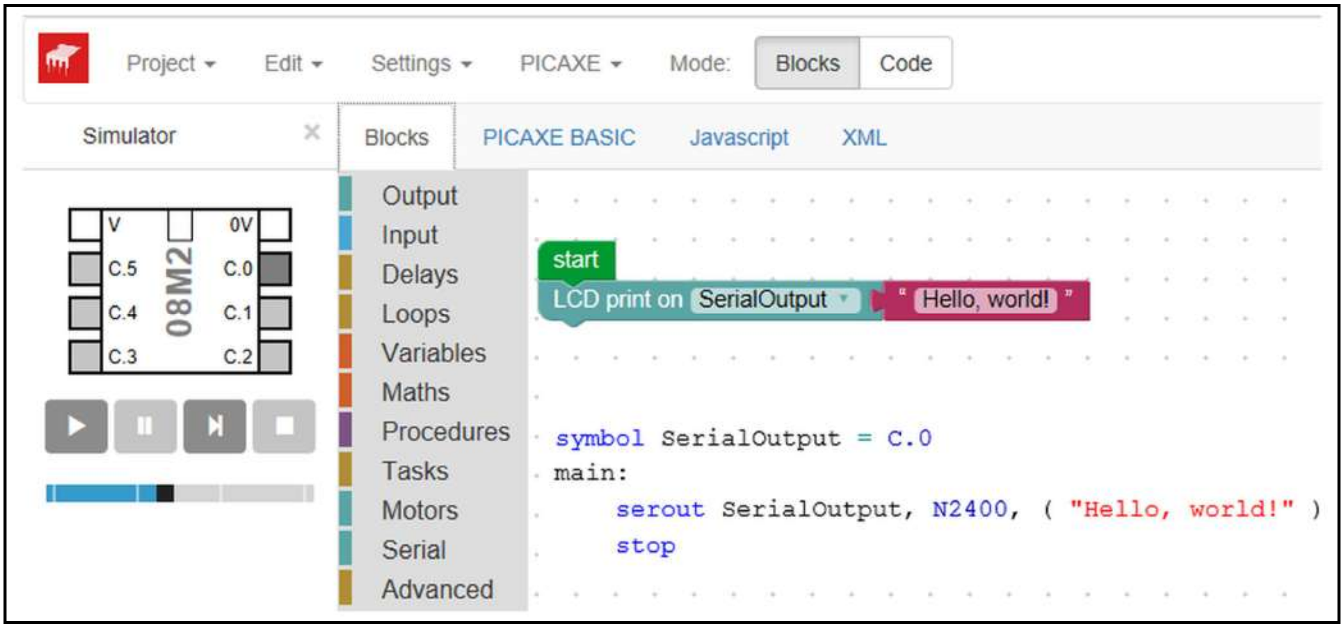



**FIGURE 4: BLOCKLY PROGRAMMING EXAMPLE**

Traditionally, the first computer program you write on any new platform is always "Hello, world!" In this case, it's by using Blockly via internet cloud programming with Google Chrome. For comparison, the equivalent PICAXE BASIC language code is pasted below the blocks.

The supplied QRSS/WSPR beacon source code (with six different CW modes) is fully commented, so I'll only mention a few major points. If you are an experienced programmer you'll have no problems following the code, and should be able to enhance and/or convert it over to another MCU platform. The major areas to watch out for are how an interrupt service routine (ISR) is implemented, and how the electrical programmable read only memory (EEPROM) is accessed by other particular MCUs.

For QRSS use, you program the beacon's callsign and desired transmission mode (only one at a time can be used). I've included sample QRSS and WSPR frequencies for both the 30 and 40 metre (m) bands, but the actual transmitter frequency will vary depending on your specific DDS module's ppm error (no two are the same). The QRSS beacon is transmitted first, followed by the WSPR beacon (make sure it starts on an even minute), but you can change the order and timing, or transmit only one mode (usually WSPR). The supplied WSPRMSG by W3PM program creates the required 162-byte WSPR beacon symbol table based on the callsign, Maidenhead grid square, and transmitter power in decibels per millivolt (dBm). This is saved to a text file, and you copy/paste these values into the appropriate WSPR code section of the QRSS program.

The QRSS and WSPR beacon fixed frequency (hexadecimal) byte values are determined using Analog Devices (ADI) ADIsimDDS website calculator. These byte values are loaded into AD9850 in on-the-fly to create the required DDS RF output. The ADI website calculator also shows any and all harmonics, images, and spurs generated by the interaction of the AD9850's MCO and the DDS generation process.


**FIGURE 5: QRP LABS QRSS/WSPR QRPp TRANSMITTER**

This transmitter (kit built) was carried onboard the MV POLAR SEA aka the "CANADA C3" expedition celebrating Canada's 125th birthday (2017). It sailed around the entire North American continent continuously transmitting WSPR beacons on three Amateur Radio bands (40, 30 and 20 m) 24/7 for nearly five months.

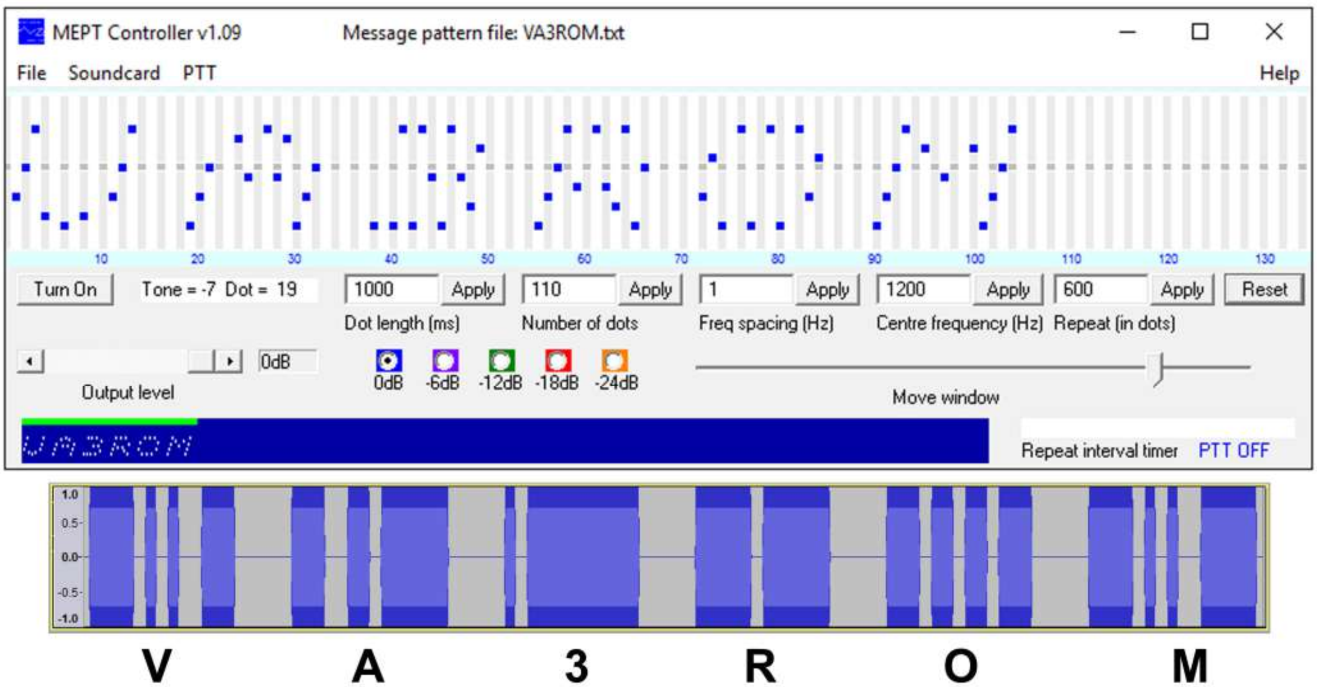
## SOME ALTERNATIVES

If you prefer to build kits, check out QRP Labs' line of QRSS and other neat products and ancillary add-ons. The "Ultimate 3S" or "U3S" (see Figure 5) is built around an Arduino MCU and Si5351 DDS module. It has many functions and features, including 16 different QRSS modes, and a GPS receiver module for precise beacon timing.

A special, smaller and lighter version of the U3S is routinely flown by pico-balloons (party foil type), and several have already circumnavigated the world (some several times!), continuously transmitting their positions and telemetry back to Earth using various QRSS and other data modes. Artistic Amateurs can also “pixel paint” custom QRSS beacon audio code usually conventional or unconventional glyphs, shapes, letters, etc., and then transmit them as audio files using a single sideband (SSB) transmitter. This is made possible by digital “wizard” Murray Greenman, ZLPBU’s, “MEPT Controller” tools (see Figure 6).

**Figure 6: MEPT Design Tool**

Some planning is needed to design shapes because only one dot or “dit” can be turned on in any one column (top). The generated audio files are easy to edit using a free program called “Audacity”.




Android smartphone users have two awesome QRSS and WSPR beacon apps created by Andrea Salvatore, IU4APC. Smartphones and related apps are untapped resources that can perhaps bring a younger generation into the 21st century Amateur Radio Service.

**MY FINAL**

In part 3, we’ll look at some specific uses of transmitted and received QRSS CW signals, besides the obvious. –73

## REFERENCES AND RESOURCES

**ADIsimDDS Calculator**
http://tinyurl.com/gpvjzus

***A "Whisper" for Canada C3***
https://wp.rac.ca/a-whisper-for-canada-c3/

**Android Smartphone WSPR and QRSS Software**
http://tinyurl.com/m5ylc7a
http://tinyurl.com/mjdzsoz

**Kits**
QRP Labs http://qrp-labs.com
Nano DDS VFO http://tinyurl.com/gqty46y

**PICAXE (Programming & Projects)**
http://www.picaxe.com
http://www.picaxecloud.com
http://tinyurl.com/z5na4fs
http://tinyurl.com/jpv7zt6
http://tinyurl.com/z3b6udm
http://tinyurl.com/hhqzucf

**W3PM**
http://www.knology.net/~gmarcus

**ZL1BPU**
http://tinyurl.com/zqpt6or