

Alachua ARES gears up for BITX HF Transceiver Projects

By Gordon Gibby KX4Z



BitX40 enclosed in thrift-shop cake-baking tin; SSB 7-12 watt transceiver can use electret mic or soundcard digital.

I've come upon an inexpensive HF single sideband radio that appears to be well-suited for advancing ARES emergency communications training for those who have been limited to VHF/UHF bands. Just as the Ubiquity systems are allowing us to develop microwave systems for under \$100, the BitX40 hits the \$70 price point, even with rapid DHL shipping from India.

This little “almost-finished-kit” has an interesting history. Trying to develop an inexpensive SSB transceiver, Ashar Farhan VU2ESE created the original BITX20 transceiver, producing 6 watts output with a simple analog design more than ten years ago. By 2007, 200 upgraded kits were put together by Henricks QRP Kits of California, called the BIT20A. The 7 MHz BitX40 came later, and used an analog VFO (until Dec. 2016) on a pre-constructed printed circuit board surface-mount kit that merely required the purchaser to solder connections to power supply, speaker, mic and such.

Then Farhan added a 2nd board: a digitally controlled VFO/display based on an Arduino Nano, controlling a Silicon Labs Si5355 chip (capable of generating up to *four* frequencies on demand, from 1-200 MHz). (<https://www.silabs.com/documents/public/data-sheets/Si5355.pdf>) The new digital VFO is known as the “Raduino” and has made this \$59 nearly-built “kit” into an exciting little 7-watts-and-up sports-car of less-than-high-power kits with thousands of homebrewers on the *groups.io* bitx20 forum. (See <http://www.hfsigs.com/> for purchase information.) This transceiver can *by itself* put out as much power as other kit+amplifier duo's – and as a *SSB transceiver*, it is capable of *many* different modes including voice. ,

The world-wide user group has already come up with freely available code (“sketch”) that is easily uploaded (via USB connector) into the Raduino, giving this stock-LSB transceiver both upper and lower sideband, two VFO's, receiver incremental tuning, innate CW, an improved calibration routine and scanning. By increasing the voltage powering the output MOSFET and beefing up the heatsink, output power can reach toward 20 watts.

For Alachua ARES, this represents an incredibly cheap HF transceiver suitable for

- teaching simple homebrew skills that make ARES members more resilient in emergencies
- developing a local voice ragchew/traffic net on either 80- or 40-meters, over a county area
- providing digital training/experience with NBEMS (FLDIGI) modes, and HF WINLINK
- teaching HF antenna skills
- demonstrate simple receiver and transceiver circuitry.

The two-board transceiver's intermediate frequency filter is four hand-chosen 12 MHz crystals in a row, with a crystal controlled Beat Frequency Oscillator (BFO) set to detect the lower sideband skirt of the IF. Although one could replace the crystal oscillator with a separate output from the Si5355 board, the most popular modification is to move the VFO from its 5MHz band (detecting lower sideband, adding to the 7 MHz received signal to produce the 12MHz IF frequency) to 19 MHz (detecting upper sideband by subtracting the 7 MHz input signal to reach 12MHz).

With easy software adjustment of the VFO, the rig can be moved to 80 meters by simply adding an output low pass filter to the transmitter (to quash the undesirable harmonics of the Class AB1 amplifier) and either an 80 meter bandpass or suitable lowpass filter to the receiver input.

This is an inexpensive transceiver ripe for modification!!

I've purchased two so far (both arrived quickly via DHL) and upgraded them with new software to give them all the new features. The difference is literally night and day. The original design is a bit quirky with lots of little "issues" that users of polished Icom/Kenwood/Yaesu transceivers aren't used to, but the software upgrades make the little rig a lot of fun even with some remaining pops and clicks. I've easily used it to make multiple digital WINLINK contacts, checked into a SSB net, and listened to all kinds of signals. PSK & RTTY are easy. Power level is moving up as I added a larger heatsink to the 99-cent PA mosfet IRF510. I've seen designs all the way to 28 watt output from this same switching enhancement mode MOSFET with adequate heatsink. A 12-volt supply (literally a gel-cell battery) supplies the system well, while the PA can be separately power from a higher voltage, and I'm currently using a thrift-store laptop 19V supply and getting 10+ watts output with a homemade heatsink from home depot pieces of carpenter's perforated metal.

In this first of two parts, I'll explain a bit how to get the system working somewhat "as built" and then in a second part describe more of raising the power and changing the band.

Kit Enclosure

The kit includes standoffs so the main circuit board can be installed on a cookie tin, thin plywood, cigar box, or cake baking pan. Mounting the Arduino VFO assembly is a bit more difficult as the mounting holes are quite close to circuitry. It is easier to create a mat of some soft pliable plastic that can be cut with a razor, and then use that to frame the digital display within a larger, jig-saw created opening. For on/off/volume, tuning, power and RF output, 1/4" 3/8" and 1/2" holes are required; in soft metal I used a 1/2" wood hole cutter successfully.

The printed circuit board has a generous ground plane built in, so shielding in a metal box isn't really necessary and many builders simply mount it all on a suitable flat board.

Power

The main board prefers no more than 12VDC, so a simple gell-cell works fine, or a 12VDC laptop charger could be used. Add in a fuse!

Software upgrade

Download the Integrated Development Environment (IDE) from the Arduino web site: <https://www.arduino.cc/en/Main/Software> Drop down the page until you find the one you can download for your operating system.....I found this *much easier than the online web version*. The Windows installer is here: https://www.arduino.cc/download_handler.php?f=/arduino-1.8.5-windows.exe . While connected to your Raduino with an USB cable, you'll choose the "type" of Arduino (choose *nano*); and you'll also choose the connection port on your computer; a little trial and error or experience with Device Manager helps.

There are several branches of free Raduino upgrade software, but the one I used was version 1.24 from here: <https://github.com/amunters/bitx40>. A recent offering is: https://github.com/amunters/bitx40/blob/master/raduino_v1.25.1.ino which adds multiple functions with one pushbutton normally-open switch (see Table below) from Raduino orange wire to ground. Instructions are available and also within the "sketch". Download the Sketch, (your software may envelop it into a "project") ; choose Compile and Upload and immediately your system is made much smarter! The code looks like standard "C" code so you can edit and add your call, etc.

Short or long presses of the function pushbutton change modes; here's the "short presses" table:

Short presses	Function chosen in 1.24 Sketch
1	Swap VFO A/B
2	Toggle RIT on/off
3	Toggle SPLIT on/off
4	Toggle USB/LSB
5	Start frequency scan

In order to get the 19MHz Raduino signal to have adequate strength, you'll need to carefully unsolder C91 and C92. (See page 1 of <http://www.petefleming.com/bitx-bfo.pdf>)

MORE TO COME

That's enough to get you going. In Part 2, I'll go over increasing the transmitter mic gain, increasing transmitter power, and other modifications.