The 75S-2B Receiver

Technical Description (TD)
Mike Bohn, KG7TR
22 February 2021

Introduction

The receiver shown in Figure 1 (all figures are at the end of this TD) is a homebrew creation that covers the classic HF ham bands, i.e., 80, 40, 20, and 15 meters, plus the first 500 kHz of 10 meters. It operates as a single conversion radio on 80 meters, and a dual conversion type for 40 meters and higher. It receives LSB, USB or AM on each band. The architecture is similar to a Drake 2-B up to the second mixer, and a Collins 75S-3 from there to the audio output. That's how I came up with the name "75S-2B".

Most of the active circuitry is implemented with 10 miniature vacuum tubes, using vintage 1960s hollow state circuitry. A touch of 21st century solid state flavor is added by two "Arduino" type microcontroller modules and ancillary hardware. These are used to implement the frequency counter/display and BFO functions. A LCD readout indicates frequency, band and emission mode. With the Arduino technology thrown in, you could say this is a sort of 20th/21st century hybrid.

In case you've never heard of it, Arduino technology is very popular these days in several hobbies, among them ham radio QRP. A concise explanation of it can be found on Wikipedia by doing an internet search on the keywords "arduino wiki". The basic building block is the Arduino microcontroller module. My introduction to using Arduinos for ham radio was via the excellent book "Arduino Projects for Amateur Radio" by Dr. Jack Purdum (W8TEE) and Dennis Kidder (W6DQ). Arduino modules and a whole lot of interfacing hardware are available all over ebay at very reasonable prices. The software is all open source and totally free. An Arduino module has a multitude of configurable analog and digital input/outputs (I/Os) available to perform various tasks. With Arduinos, you write and compile a program on your PC called a "sketch ", using the free Arduino application. Then you load the sketch into the microcontroller's memory over a USB connection. When power is applied to the module, the sketch executes as a continuous program loop.

After reading the first couple of chapters of Purdum and Kidder's book, I obtained some inexpensive modules of various types and began experimenting. Having never written a line of computer code in my life, I was concerned that I would never be able to get the modules to do what I wanted. But this turned out to be one of the easier parts of the project. It turns out there are lots of sample sketches available on the internet to use as starting points, as well as a multitude of modules and circuits to do a variety of radio related functions. With these basics at hand, I was able to find excellent reference material and examples to tailor my sketches and hardware configurations for this project. It only takes a few minutes to modify a sketch and reload it into the Arduino.

The final configuration of the radio described here is markedly different from the original vision. My initial approach was to use Arduino controlled RF signal generator type modules to replace the three oscillators used in the vacuum tube 75S-3, namely the crystal controlled heterodyne oscillator, the permeability tuned oscillator (PTO) and the BFO. Although I was eventually able to get the microcontrollers to behave the way I wanted, there were some learning

curve issues encountered with Arduino controlled RF modules that use Direct Digital Synthesis (DDS) microchips of the AD9850 type, or programmable clock generators of the Si5351 type. While these signal generator type modules are inexpensive, popular, and remarkable for what they can do, I discovered there are some caveats regarding spurious frequencies when used in sensitive receiver applications. In the end I had to abandon using these RF modules as substitutes for the first two oscillators in the receiver as originally envisioned. Instead, I had to revert back to good old hollow state technology for these functions. But the BFO and counter/display were still doable using Arduinos. There is a later section in this document describing the whole evolution of the project if you want to read about that journey and its associated "learning opportunities". For now I will just stick to the configuration as it ended up.

This is not intended to be a construction article. I have not put together a parts list, mainly because the RF coils I used would be difficult to find or duplicate these days, and the rest of the parts can be determined by inspection of the schematic or from supplemental information in the text. Most of the hollow state circuitry after the second mixer is unremarkable, being very similar to the Collins 75S-3. Areas that homebrewers and experimenters might find interesting include the preselector design process, the crystal heterodyne oscillator using cheap microprocessor crystals, or the Arduino hardware and software.

The next few sections deal with the vacuum tube aspects of the 75S-2B. Later will follow some discussion of how I used, and attempted to use, Arduino technology to augment my hollow state design.

Mechanical Layout

The radio uses good old fashioned rack and panel construction. It is built on a 10 x 17 x 2 inch aluminum chassis mated to a 7 x 19 inch (size 4U) aluminum rack panel. The cabinet was originally an all-steel commercial unit from Hammond Industries (P/N RCDB1900713LG1). It was much bigger and heavier than needed for this application, but was the smallest I could find. In May, 2019 I finally got around to fabricating a custom cabinet from aluminum sheet metal as I had done for some of my earlier projects. It is seen in Figure 1.

Figures 2, 3 and 4 show top, rear and underside views of the chassis. At the rear is a vertically mounted 4 x 8 x 2 inch chassis used to house the frequency counter and BFO modules. The slug tuned coils in shielded cans were scavenged from a junk piece of test equipment labeled "Two Tone RF Source", bought years earlier at a hamfest for two dollars and probably built by Collins. These coils are plug-in type assemblies that can be pulled from their sockets after removing a screw from the bottom of the chassis. The two section variable capacitor used for the preselector is driven by a Millen 5:1 vernier reduction drive mounted directly to the front panel. A homebrew dial is mounted to the vernier to indicate band limits.

The main tuning capacitor in the front center of the chassis is a three section unit from a junk ARC-5 receiver. It is used to tune the VFO and the variable first IF circuits, and represents one of the major modifications to the original design. It is mounted vertically on standoffs because that's the only way it would fit at that point in the radio's construction without hacking up the front panel and chassis. A .125 inch thick aluminum plate is mounted under the variable to stiffen up things. If I were starting over from scratch, this variable would be mounted directly to the front panel using the four feet that are part of the capacitor's frame. I would also relocate the coils it tunes right next to it.

Shields mounted on the chassis bottom add extra rigidity, separate the various circuits, and provide mountings for rotary switch sections. Flexible couplings are used where needed to compensate for minor misalignments. Also visible in some places are extra holes in the chassis due to the various modifications. I despise these "monuments to poor planning", but it's a price I paid for not thoroughly researching the limitations of RF module performance during the breadboard phases.

The main tuning knob was carefully drilled and tapped to fit a homebrew spinner. It is mounted to a .25 inch stainless steel shaft riding in a panel bearing, and coupled to the main tuning capacitor thru a flex coupling. The old ARC-5 variable was thoroughly cleaned and lubricated prior to installation. One thing I have learned about these old variables is that you have to make sure the anti-backlash gears are not stuck together from decades of dried out oil. This is the gear assembly with the spring in it that is driven by the worm gear. This applies to ARC-5 transmitter variables as well. A thorough soaking with mineral spirits or WD-40 usually does the trick. After this cleaning, the whole tuning assembly is remarkably smooth, with virtually no backlash. As built, about 28 revolutions are required to cover the 500 kHz range of the VFO. The bandspread of the knob varies from about 19 kHz per turn with the variable at maximum capacitance to about 10 kHz per turn with it at minimum capacitance. This may be somewhat lower than the optimum tuning ratio for SSB, but it is what it is. Without any backlash present, I find it acceptable.

Details of the preselector and main tuning controls can be seen in Figure 5. As with my more recent homebrew radios, water slide decals were used to label the front panel and chassis, and for this radio, the preselector dial as well. The materials and processes used to do this are available on page 7 of the KG7TR Homebrewing Tips document. See the link at the bottom of the home web page at: http://www.kg7tr.com.

Block Diagram Description

A block diagram of the receiver is shown in Figure 6. V1 operates as an RF amplifier on all bands. Band switched preselector circuits in the grid and plate of this tube are tuned by a dual section variable capacitor to peak signal response at the desired frequency, and reject other signals.

Output of the preselector is fed to the signal grid of V2A, the pentode section of the tube. V2A functions as the first mixer for 40 through 10 meters, and as a straight through amplifier on 80. Its triode section is used as a crystal controlled heterodyne oscillator for 40 through 10. The selected crystal frequency is 4.0 MHz higher than the lower end of the band in use. The crystals used are very inexpensive microprocessor crystals procured from Mouser Electronics. It just so happened that the frequencies of 11.0, 18.0, 25.0 and 32.0 MHz were available from multiple manufacturers for less than one dollar each. It was the availability of these cheap crystals on those exact frequencies that drove the final design to a Drake 2-B type front end. Cheap microprocessor crystals in the frequencies that would have been required to stay with a 75S-3 front end were not to be found from any source. On 40 through 10 meters the oscillator signal is injected into V2A's cathode to create the mixing action. On 80 meters the triode is section is turned off by grounding its grid and removing its plate voltage.

For all bands, the desired output of V2A is in the 3.5 to 4.0 MHz range, as peaked by two sections of the main tuning capacitor. This signal is fed to the signal grid (grid 3) of V4, the second mixer (actually the first mixer on 80 meters). The pentode section of V3 is the VFO, and

its frequency is tuned by the third section of the main tuning capacitor. This frequency is always 455 kHz higher than the variable IF. The triode section of V3 is a cathode follower for the VFO, and its output is injected into the oscillator grid (grid 1) of V4 to provide mixing action. The output of V4 is the final IF frequency of 455 kHz.

Thus, the radio is basically a single conversion 80 meter receiver, with crystal controlled oscillator injection providing dual conversion for 40 through 10 meters. In addition to the Drake 2-B, this popular architecture was used in the Mosley CM-1 and other receivers of the era. I even ran across it in some ARRL handbook designs using both tubes and solid state technology.

After V4, the radio looks very much like a Collins 75S-3. The 455 kHz output of V4 is routed through a 2.1 kHz Collins mechanical filter for LSB or USB, or an 8 kHz Murata ceramic filter for AM. This is followed by IF amplifiers V5 and V6. Output of V6 is applied to the grid of V8 product detector, and both diode plates of V7. In LSB or USB modes the audio output of V8 is routed through the AF gain control to the signal grid of V7, which operates as a voltage amplifier. In the AM mode, detected audio from one of the diodes of V7 goes through the AF gain control back into its grid. Output from V7 is fed to the audio output tube, V9, to drive the built-in speaker or external headphones.

The other diode in V7 generates an AGC voltage that is added to negative bias from the IF gain control to control gain of the receiver. The AGC voltage is applied to the signal grids of V1, V5 and V6. A switch adds capacitance to the AGC line to slow down release time, or disables the AGC function in the off position. The S meter is connected between the screen of V5 and the cathode of V6, with appropriate resistors in the circuit for scaling and zeroing the meter.

An Arduino controlled RF module applies a BFO injection signal to the cathode of V8 in LSB and USB modes. This signal is turned off for AM. Another Arduino module fed from the VFO output functions as a frequency counter and driver for the LCD display. Positions of the bandswitch and mode switch are sensed by the two Arduino modules as contact grounds. This data is used to: a) set the BFO frequency, b) apply an IF offset value in the counter for correct frequency display across the top line of the LCD, and c) display the selected band and mode across the bottom line of the LCD.

The power supply is all solid state except for the VR tube, and provides B+, filament and negative bias for the tube circuits. The +150 volt source is also fed to V10 to provide regulated B+ for V2B and V3 oscillators. A low voltage supply provides regulated voltages for the Arduino modules.

Vacuum Tube Circuit Details

The schematic diagram of the hollow state circuits is shown in Figure 7. It is also available at http://www.kg7tr.com/75s-2b-receiver.html as a .pdf that can be enlarged for easier reading. Starting at the top left, the antenna input is fed to a link on the preselector main input coil. Preselectors based on dual section variable capacitors work the same way in virtually every vacuum tube receiver I looked at. The one used here is very similar to that used in the Hallicrafters SX-117 and SX-146 receivers. Variable capacitor C1 tunes the preselector, and is a new unit procured from RF Parts. It has dual 365 pf sections that use a straight line frequency taper, such that the shaft is offset from the centerline of the plates. This means, for example, the capacitance at 50 percent rotation is about 107 pf instead of half of 365 pf. Trimmers across each section of C1 add about 20 pf when fully tightened.

Design of the preselector was an interesting exercise. I used a spreadsheet based on measured values of C1 at each five percent of rotation from maximum to minimum capacitance, and inductance values selected in four ranges (but five bands) by bandswitch sections S1A and S1C. Formulae in the spreadsheet calculate the resonant frequency in each range at each five percent of C1 rotation. A copy of the spreadsheet is available at http://www.kg7tr.com/75s-2b-receiver.html. The objective was to end up with a tuner that covers 3.5 to 30 MHz in four ranges, using a single control (which in this case is C1). An additional nice to have is to ensure that none of the ham band segments result in settings of C1 that overlap each other. Not essential, but it does make the preselector dial simpler and less confusing.

So it became a matter of using the spreadsheet to go back and forth between the bands and adjust values of capacitance and inductance to meet the design objectives. The starting point was 40 meters using main coils L1 and L4, each tuned by a parallel section of C1. These components are always in the circuit, and there are no additional capacitors or inductors in the circuit for 40 meters. The desired resonance peak on 40 meters is at about the center of C1's rotation. For 80 meters, additional 220 pf capacitors were added in parallel with C1A and C1B to cover 3.5 to 4.0 MHz. For 20 through 10 meters, no additional capacitance was used. Instead, additional inductances were connected in parallel with L1 and L4 to decrease total tank inductance, thereby increasing the resonant frequency. For 20 meters the additional inductances are L2 in series with L3, and L5 in series with L6. For 15 and 10 meters the additional inductances are simply L3 and L6. 15 and 10 meters are covered using the same coils, as indicated by the connections on S1A and S1C. The commercial receivers I looked at used extra coils for 15 meters, but I found they weren't needed. As Figure 5 shows, all the classic ham bands are covered without any overlap on the preselector dial.

So by using inductances in parallel, preselectors are able to do the magic they do. It's also worth mentioning that main coils L1 and L4 have link windings to couple the antenna input and V1 output signals into the tuned circuits of the preselector. Because the additional inductors used for 20 through 10 meter coverage are in parallel with L1 and L4, the same voltages appear across them as well. Therefore, no additional coupling links are required.

A 6DC6 is used as the RF amplifier at V1. I figured if this tube was good enough for the 75S series, it's good enough for this radio. I suppose a 6BZ6 as used in the Drake 2-B would have been about the same, but I didn't have one. AGC is applied to V1's grid through a network identical to the 75S-3B. The capacitor connected to the mute line was added in the "B" version of the Collins receiver. I think it's supposed to help the receiver recover faster when it's unmuted.

As in the Drake 2-B, V2 is a 6U8 whose pentode section is used as the first mixer/amplifier, with the triode section serving as a crystal oscillator. This tube has both its cathodes connected together and run through a 1k resistor to ground. This common connection above RF ground allows mixing action to take place in the pentode section on 40 through 10 meters. The triode section operates as a crystal oscillator on those bands. Each band has its own coil and capacitor combination switched into the plate circuit. A close up of the crystals and coils mounted on the bandswitch is shown in Figure 8. The oscillator output is taken from the triode's cathode as described earlier. On 80 meters the triode grid is grounded and the plate is open circuit, effectively disabling the oscillator. The pentode section then operates as a straight through amplifier with low gain as a result of the unbypassed 1k resistor connected to its cathode.

The output of V2A is coupled to the variable bandpass IF. The location of C2 precluded short leads from its stators to L11 and L12. Therefore, these connections are run through

RG-174/U coax as seen in Figure 2. Each section of C2 is about 390 pf, which is way too much capacitance for the bandpass needed. Therefore, 91 pf padder capacitors are connected in series with C2A and C2B. If I had the variable from the 3 to 6 MHz ARC-5 receiver this situation might have been better, since that capacitor is only 150 pf per section.

V3 VFO is a Colpitts circuit similar to those used in all my other homebrew projects. Tank coil L13 is a fixed inductor wound on a threaded ceramic core and enclosed in a shielded can. Values of the various fixed capacitors in the tank circuit were determined partly by calculation, and partly by experimenting with capacitors on hand. They are all dipped silver mica with the exception of a single 47 pf, N750 ceramic capacitor used for temperature compensation. The RF voltage at the cathode of the pentode section is coupled the grid of the triode section, which acts as a cathode follower.

Originally, RFC3 was connected directly between the cathode of the triode section and ground. After building my HB-75 transceiver (described on my website), I discovered that this configuration resulted in a lot of excess plate current in the triode section with little increase in output voltage. This translated into extra plate dissipation and heat in the tube. So I added the 2.2k resistor, bypassed by a .01µf capacitor, to alleviate this problem. The plate current went from about 11ma to 2ma, with still plenty of output to drive mixer V4. This decrease in load current also allowed me to increase the resistance of the dropping resistor in series with the V10 regulator tube. As it now stands, worst case load current for the V2 and V3 oscillators is about 5.2ma. The additional current into the regulator tube runs at about 17ma, which is a nominal value.

The output of the V3 cathode follower is then coupled to mixer V4, as well as a preamplifier used to drive the Arduino counter module. Unlike the 75S-3, no sideband shift is incorporated into the VFO. This would have required some kind of circuit to shift the VFO frequency by exactly the right amount for the opposite sideband, at every frequency across its range. I was not up to that task. As it stands, the VFO frequency does not shift, but offsets are incorporated into the counter software that ensure the correct frequency is always displayed. Even the center frequency of the AM filter is accounted for in the AM mode.

After a 10 to 15 minute warm up the VFO is very stable, with no objectionable drift noted. It is also mechanically stable, especially with the radio mounted in its cabinet. Note the 91k resistors connected to the grids of V3. These are .25 watt resistors found at many locations in the radio. All of these could have been 100k in the .25 watt size if only I had some. But I do have a slew of 91k resistors in that size, so that's what I used.

To finish up the Drake 2-B cloning, a 6BE6 at V4 is used as the first/second mixer to generate the 455 kHz IF. From this point on the radio is very similar to the Collins 75S-3. V4's output is switched through FL1 for SSB or FL2 for AM. The fixed capacitors across FL1 were selected in test to resonate the input and output of the filter. For the AM filter, tuned matching transformers were necessary to maintain the same signal levels and S meter readings as SSB. The 2.2k load resistors are based on Murata data sheet information.

V5 and V6 are IF amplifiers using conventional circuitry. A fixed and variable capacitor were used to resonate fixed inductance L16 for the simple reason I didn't have any more slug tuned coils with enough inductance for 455 kHz. AGC is applied to the grids of both tubes. The S meter circuits connect the meter between the screen of V5 and the cathode of V6. As AGC voltage becomes more negative with stronger signals, the screen voltage on V5 increases and the cathode voltage on V6 decreases. This causes the meter to deflect upward. R1 balances the

circuit for no deflection when there is no AGC voltage. The sensitivity set by the resistors used yields a nominal S9 reading with 50 microvolts present at the antenna input.

Note that the homebrew S meter dial shown in Figure 1 starts at S4. The meter scale was calibrated using my HP 8640B signal generator and modern S unit standards described on Wikipedia and other internet sources. These standards set S9 at 50 microvolts into 50 ohms at the antenna input, and back down from there at 6 db per S unit. This means you get values of .2, .4 and .8 microvolts for S1, S2 and S3 respectively. Well, the truth is the S meter can't even deflect until the signal exceeds the AGC threshold, which for this receiver is about 1.2 microvolts. This is a typical value for a tube receiver like this. The AGC action is "delayed" until the signal reaches the threshold to provide adequate gain and signal to noise ratio. A modern S4 happens to work out to 1.6 microvolts, where my meter has just started to deflect upward. So that's where the scale starts. The meter is accurate at S9/50 microvolts to within an S unit on each band and all frequencies within each band. Radios you see with S meter scales showing markings below S3 or S4, but with typical AGC thresholds of perhaps one to two microvolts at the antenna, are more likely based on about 3 to 4 db per S unit criteria at the low end of the scale.

The output of V6 is coupled through L17 and L18 to V7 and V8. These coils are identical, and happened to have a low impedance link winding. So I decided to couple them together to make a double tuned transformer. This allowed using the tuned coil on L17 to feed the plate of V6, and the low end of the tuned coil on L18 to feed the AGC circuit. By experimentation it was determined that the .001µf capacitor between the links is the value that provides critical coupling.

The IF signal at the pin 5 diode plate of V7 is rectified to generate AGC voltage. The circuit works the same in SSB and AM modes. When positive peaks of the signal exceed the bias on the pin 2 cathode, the diode conducts and charges up the capacitors at both ends of the 4.7k resistor on the low end of L18. These capacitors provide a necessary RF path to ground for AGC and AM detection in V7, and SSB detection in V8. For AGC detection we are left with a negative DC voltage across the .01µf capacitor that depends on the signal amplitude. The bias on V7's cathode provides the delayed AGC action. No rectification occurs until the signal exceeds this threshold. When AGC switch S3 is in the slow or fast positions, the negative voltage across the .01µf capacitor is applied to the AGC bus to control the gain of V1, V5 and V6. When the switch is off the AGC voltage is disconnected from the ACG bus.

The AGC bus is also connected to a circuit consisting of a 22k resistor and two capacitors that determine the fast attack, slow decay characteristics needed for SSB. For slow AGC, a .22µf capacitor is connected in the circuit to increase the decay time. For fast AGC, a .047µf is used instead. The selected capacitor is charged through the 22k resistor to decrease the attack time somewhat to avoid short duration pulses like static pops from swamping the AGC. The selected capacitor is discharged through the 3.3meg resistor to provide slow decay. The other end of the AGC circuit is connected to the wiper of the R3 IF gain control, which is part of a voltage divider connected to the -75 volt source. When the receiver is unmuted, the voltage at the wiper of R3 can be varied from about -1 to -35 volts. This voltage is applied the AGC bus to manually control the gain of the receiver. When S3 is off, the AGC network is shorted out and the bias from the R3 wiper is applied directly to the AGC bus. With the switch in slow or fast the bias is applied to the AGC bus through the network. In this condition any AGC voltage present will be added to the manual gain bias. When the receiver is muted, a 12k resistor is inserted into the low end of the bias network, thereby increasing the voltage on the AGC bus to a value that cuts off the controlled tubes for any setting of R3.

When the IF gain is backed off from maximum, the grid bias on the controlled tubes goes more negative and the gain of the receiver is reduced. This means it takes a stronger signal at the antenna to activate the AGC. The increased bias also causes the steady state S meter reading to increase. The net effect is that the IF gain can be set so the S meter only deflects on SSB voice peaks, but still gives a true signal strength indication. This is a technique we are all familiar with on most SSB receivers even into the solid state era. With slow enough AGC decay we get reduced noise between peaks and words during a transmission. On AM reception there is less benefit to be realized since the carrier level is constant.

For AM reception, we use the IF signal applied to the pin 6 diode plate of V7 through a 100 pf capacitor. Rectification takes place, and the 100 pf capacitor filters most of the RF. The resulting envelope of the IF signal appears across the voltage divider formed by two 91k resistors when S2G is in the AM position. Further RF filtering occurs in the 470 pf capacitor on the wiper of S2G. The resulting AM audio is applied to the high side of the AF gain control R2. From here the signal comes off the wiper of R2 and goes to the control grid of V7. The triode section of V7 amplifies the signal and applies it to audio output tube V9, which drives the speaker or headphones through T1. A 220 ohm resistor in series with the headphones greatly reduces annoying 60 Hz hum while still providing plenty of volume. This headphone hum can be a pretty common problem with tube receivers.

In the SSB mode, the IF signal at the grid of the V8 product detector is mixed with a BFO signal applied to the tube's cathode. RF appearing at the plate of the tube is filtered by the two 470 pf capacitors and the 47k resistor in the circuit when S2G is in the LSB or USB position. From here the signal follows the same path as an AM signal.

The power supply is straightforward and uses conventional design. The AC power switch is ganged with the R3 IF gain control. Transformer T2 is a Stancor PC8420 unit, and choke L19 is a Stancor C-2303. The choke has an inductance of 2.5 henry, which is close to the critical inductance for the 90 to 100 ma of DC current being drawn. But it's what I had on hand and it appears satisfactory, as the +225 DC voltage at its output rises only a few volts when the receiver is muted or the IF gain is reduced. Both these conditions reduce the B+ current being drawn by the AGC controlled tubes. This is offset somewhat by increased current into the V10 regulator circuit as the B+ voltage attempts to increase. There is nothing special about the 0B2 regulator circuit; its parameters were described earlier. There is additional filtering for the B+ applied to audio output tube V9 to eliminate an annoying 120 Hz hum I was getting in the speaker at minimum audio levels. The 680 ohm resistor and 220 µf capacitor in the circuit form a low pass filter that is totally effective.

Transformer T4 is a reverse connected filament transformer. Its 120 VAC output is half wave rectified and filtered to provide negative bias for the tube circuits. The output of T3 is full wave rectified and filtered to provide unregulated +18 DC to IC regulators used for the solid state circuits. This supply was sized for the original configuration of the radio, and can source up to 500 ma of current. In the radio's final configuration it is totally underutilized, but I decided to leave it the way it is.

Overview of Arduino Hardware

For this project I used the very popular Arduino Uno type modules for all experimentation and final implementation into the completed radio. The Arduino Uno is a microcontroller board based on the ATmega328 chip. Features include 14 digital input/output pins (of which 6 can be

used as pulse width modulated outputs), 6 analog inputs, a USB connection for program loading and +5 DC power input, a separate power jack with an on board regulator to accept higher input voltages, and a reset button. Two examples of the Arduino Uno are shown in Figure 9. For reasons that will be explained later, the module on the right with two quartz crystals is the one to use for a frequency counter.

In many cases connections are not made directly to an Arduino module. Instead, a prototype board called a "shield" is plugged into the top of the Arduino module. The Arduino pins are carried through the shield on headers and bear the same pinout IDs. A typical shield used during my experiments is shown in Figure 10. The shield has lots of plated through holes, as well as power and ground buses. These provisions allow you to build up ancillary circuitry right on the shield, and install additional headers to make external connections. In the Figure 10 example, an amplifier module like that used for the BFO has been soldered into place on the shield, as well as some headers for external connections and a potentiometer for controlling LCD contrast. An RF synthesizer module plugs into the headers that have been installed along the sides of the amplifier module.

As mentioned earlier, there are presently two different chips that are popular for generating RF signals using Arduino control. These are the AD9850 and the Si5351. Both are available already mounted on small modules that include extra circuitry needed for the Arduino interface. Figure 11 shows an example of each type. The AD9850 board on the left is very prevalent, and comes preassembled. It has 10 pins on each side that plug into headers on an Arduino shield. It puts out a sinewave of about 1.0 volts peak to peak. The Si5351 module on the right was assembled from a QRP Labs kit. It has the same footprint and almost the same pinout as the module on the left. Only a few wires have to be changed on the shield to switch from one to the other. The QRP Labs module has the surface mount Si5351 chip already installed, with the rest being very easy to assemble. It puts out a square wave up to perhaps 2 volts peak to peak.

Output from either RF module is generally too low for direct use in vacuum tube circuits. There are some simple transistor amplifier circuits described on the internet that will produce a few volts of RF at lower frequencies. But their output drops off rapidly above 10 MHz or so. For my initial configuration of the radio I needed at least two volts rms of RF at a fairly constant level up to 33 MHz. That prompted development of a homebrew IC based amplifier using the LMH6703, a current mode op amp. This chip has a useful bandwidth of 1 GHz, and without careful layout and bypassing it will oscillate just by blowing on it. I initially tried to build it right on the Arduino shield, but could never get it tamed down. So I procured some prototype boards designed to accept the LMH6703 surface mount chip and tried again. I finally arrived at the configuration shown in Figure 12. It required a lot of intricate micro-soldering. The back of the board is covered with copper foil tape to provide a ground plane. It would be nice if somebody would make a kit of this amplifier, with a circuit board layout designed for unconditional stability and the surface mount chip already soldered in place.

Figure 13 shows packages for the proto board used for the LMH6703 amplifier and the QRP Labs Si5351 kit. Internet addresses are clearly labeled for both items. Figure 14 shows the complete BFO module. The Arduino module is on the bottom, and has nylon standoffs epoxied to the mounting holes for fastening it to the back of the vertical chassis. On top of it is the shield. The Si5351 module is plugged into the shield, with part of the amplifier module visible just below it. This complete assembly is what you see in the left compartment in Figure 15, the rear view of the solid state chassis. In the center compartment is the counter and LCD driver

assembly. Here you are looking directly at the shield, where a transistor and other components are visible, as well as some typical cable connections.

Figure 16 shows a rotary encoder and LCD with homebrew ribbon cables attached. These were used during the experimentation phase.

Solid State Circuit Details

Figure 17 shows the schematic of the solid state circuits. It is also available at http://www.kg7tr.com/75s-2b-receiver.html as a .pdf. At the lower left of the schematic are two IC regulators that provide +10 DC for the BFO amplifier and +8 DC for the two Arduino modules. These TO-220 packaged regulators are mounted to the bottom rear of the main chassis, and are visible in Figure 4.

When discussing the Figure 17 schematic, it should be pointed out that many schematics, including mine, will not depict the actual shield interconnections to the Arduino module. It is generally understood that a shield is present.

An Arduino controlled Si5351 module is used to generate the BFO signal. At this point in the radio I was able to use a module like this without spurious signal issues because it is after the high gain IF circuitry. This was fortunate because 455 kHz BFO crystals are expensive, if you can even find them. INRAD lists them in their catalog for a current price of $41 each. And there are no cheap microprocessor crystals available at these frequencies. Plus you can't easily tweak the frequency of a 455 kHz crystal to accommodate the optimum carrier insertion point for the SSB filter in use. With the Arduino controlled module this is done simply by changing the frequency value in the sketch. The frequency can be changed by as little as 1 Hz.

The BFO Arduino module selects the LSB or USB frequency by reading an associated grounded contact on mode switch section S2E. These contacts are connected to pins D11 and D12, which are configured in the Arduino sketch as digital inputs using internal pullup resistors. Pin D2 is also configured as a digital input, and reads a grounded contact on section S1D of the bandswitch in the 80 meter position. Because the receiver is single conversion on 80 meters, a sideband inversion occurs on that band. The grounded contact causes the sketch to switch to the opposite BFO frequency. In other words, on 40 through 10 meters no inversion takes place, meaning LSB is about 456.350 kHz and USB is about 453.650 kHz. On 80 meters the two frequencies are reversed.

The Arduino module is powered from +8 DC. An internal regulator makes +5 DC available to its own circuits, as well as externally to power the Si5351 module. The QRP Labs kit includes another on board 3.3 volt regulator to power the Si5351 chip. The set frequency commands are sent to the Si5351 module over the I2C SCL and I2C SDA lines. Square wave output from the CLK0 pin is fed into the LMH6703 amplifier module. A simple transistor amplifier would have been adequate here, but since I already had the LMH6703 module built and soldered to the shield I used it. In AM mode, output from the Si5351 module is inhibited in the sketch. In addition, +10 DC is removed from the BFO amplifier by S2F.

The output of the amplifier module is a faithfully reproduced square wave at over 6 volts peak to peak amplitude. Given the problems I had already experienced with spurious signals, I decided there might be trouble ahead if I used this signal as is, what with square waves bouncing all over the place. So I built a simple filter using back to back connected transistor IF transformers from a Radio Shack assortment. These were laid sideways inside a large "D" connector backshell and mounted to the main chassis at the right rear, as can be seen in Figure 2.

The input to the filter is run through RG-174/U coax from the BFO module. The output is a short wire run to pin 7 of V8. This filter makes nice sine waves for injection into the cathode of V8.

The other Arduino module is used as a frequency counter, and is based on the design contained in chapter 15 of Purdum and Kidder's book. There is a frequency counter routine built into the microcontroller that is used to implement the function. The counter input on pin D5 of the Arduino requires something resembling a TTL level square wave signal. It is also necessary to avoid loading the VFO as much as possible. So a preamplifier circuit is used to provide these functions. The MPF-112 FET connected to the VFO output operates as a source follower. It presents a very high impedance load, such that the VFO doesn't even know it's there. The FET circuit is housed in a small "D" connector backshell that can be seen directly behind the mechanical filter in Figure 2. This location allows a very short lead to be run to the VFO output at pin 1 of V4. The output from the FET source terminal is run through RG-174/U coax to the Arduino shield, where the 2N708 amplifier circuit resides. The 2N708 is a dinosaur when it comes to transistors, but it was handy and provided enough gain to give me the waveform needed. The counter's Arduino module is provided +8 DC, with +5 DC from its internal regulator picked off and used to power the LCD and preamplifier circuits.

The first iteration of the counter used the Arduino Uno module shown at the left in Figure 9, which is perhaps the most common type to be found. I discovered that it would not stay calibrated. The counter is calibrated in the sketch by setting a constant that corrects for errors in the microcontroller clock frequency. Any counter can only be as accurate or stable as its clock. Using the Arduino Uno on the left, I would no sooner measure the required error constant and put it in the sketch, and it would be wrong again. I could not get better than about 1 kHz of accuracy for any length of time. After some internet research, I found out that the microcontroller clock in the module on the left in Figure 9 is actually using a ceramic resonator. While there is a crystal on this module, it is used for the USB interface chip, not the Atmega328 chip. If you look closely, you can see the very tiny ceramic resonator inside the double white oval right next to the 28 pin DIP, which is the ATmega328 microcontroller.

Needless to say, a ceramic resonator is not going to be as accurate or stable as a quartz crystal. It will be good enough for almost any other Arduino function where timing is not particularly critical. So with a little searching around the internet I was able to locate the Arduino Uno on the right in Figure 9. It has two quartz crystals, one for the USB controller and one for the microcontroller clock. Using this module solved the problem.

The basic counter function in the Arduino measures the raw VFO frequency in MHz. Like the BFO module, data from the mode switch and bandswitch is sensed on pins configured as digital inputs. Note that some of the pins on the two Arduino modules are connected together. With the pins configured as digital inputs with internal pullup resistors, it simply means that both modules are pulling the pins high, and both will go low when grounded. This did not cause any problems. The data from the switch grounds is used by the counter sketch to apply offset calculations for the crystal oscillator (40 through 10 meters only) and BFO frequencies, so that the frequency displayed on the LCD is correct on all bands and modes. The offsets are based on measured frequencies such that the readout is accurate to better than 100 Hz. A ribbon cable is connected to bandswitch section S1D and runs up to the counter module. Because this switch wafer is also shared with the preselector, some digital hash from the Arduino was being picked up. So the ribbon cable leads are run through a low pass filter to remove the noise.

Bandswitch and mode switch data is also used to display the band and mode on the bottom line of the LCD. Of course this can also be determined by simply looking at the positions of the front panel switch knobs. But there was a second line of display capability available on the LCD, it was an easy thing to do, and it looks cool, so why not? The LCD interface to the Arduino module is standard stuff available on a multitude of websites and won't be discussed here.

The final Arduino sketches used for the BFO and counter/display are available at http://www.kg7tr.com/75s-2b-receiver.html. These are in .txt format because I could not post the Arduino files with the .ino extension on my website. The .txt files can be opened in Notepad, then copied and pasted into the Arduino Integrated Development Environment (IDE) on your computer. In order to use them you must also download the various libraries called out in the *include* statements at the beginning of the sketches. This is standard Arduino stuff.

Project Evolution and Problems Along the Way

This receiver project started out with the acquisition of a new old stock (NOS) Collins mechanical filter, a 455 kHz "bathtub" unit with a 2.1 kHz bandwidth for SSB. Having built several SSB transmitters and a transceiver at this point, I decided to try another receiver. My first thought was to attempt cloning a Collins 75S type receiver since I already had the filter, perhaps one of the most essential components. Of course it would be impossible to duplicate the permeability tuned preselector used in the Collins radios, but a preselector built around a dual section broadcast (BC) type variable capacitor appeared feasible. These less expensive alternative preselector designs were popular in some Drake and Hallicrafters receivers of the early SSB era. The slug tuned coils I had from the junked piece of test equipment covered the range of 500 kHz to 30 MHz, and so were useable for the circuits in an HF receiver. The preselector and IF circuits seemed doable.

Next, I would need to address the three oscillators to be used in the dual conversion clone. The Collins 75S series of receivers use a band switched, crystal controlled first oscillator to convert the incoming signal down to a first IF frequency of 2.955 to 3.155 MHz. This is then mixed with a 2.5 to 2.7 MHz permeability tuned VFO (PTO) to produce the second IF of 455 kHz. A crystal controlled BFO provides a nominal 455 kHz LSB or USB injection signal to the product detector to recover audio from the IF signal. While I was pretty sure I could duplicate all three oscillators using tubes, getting the required crystals was not going to happen. A stock Collins 75-Sx receiver uses a total of 12 crystals for the first oscillator and two for the BFO. When International Crystal was still in business, new crystals were running more than $50 each. Used Collins crystals, if they can be found, are in the $30 range. In addition, the classic Collins architecture requires a 14 position bandswitch due to the limited 200 kHz span of the PTO. With a digital display to take care of accuracy, a VFO range of 500 kHz seemed more reasonable. Although that would require only five crystals for the first oscillator, I would need a 500 kHz bandpass for the first IF around 3 MHz or so, which is pretty wide. With these considerations in mind an alternate architecture for the oscillators began to emerge.

Arduino controlled RF modules have already been discussed in previous sections. They are definitely capable generating accurate and stable RF signals. The AD9850 chip from Analog Devices uses Direct Digital Synthesis (DDS) technology. Key to its operation is a digital to analog converter that creates a sine wave as a series of small amplitude steps over time. The Si5351 chip from Silicon Labs is described by the manufacturer as a programmable clock

generator, and uses frequency multipliers/dividers and phase lock loops (PLL) to do the job. You can read details about how they work on the manufacturer's websites. I do not pretend to fully understand their inner workings.

The output frequencies from these modules range from a few kHz into the VHF range, so they appear ideal for an HF receiver. The flexibility of the Arduino system allows simple interfaces to a rotary encoder for changing frequency and frequency steps, as well as a LCD display that can show commanded or offset frequencies and other parameters.

There is a wealth of information on the internet about how hams are applying these capabilities to various QRP applications with apparent great success. It seemed like I should be able to use this technology at the three locations in my receiver where I needed stable and accurate RF sources to replace the vacuum tube oscillators. Some amplification might be needed to get the RF levels up to a few volts for injection into the tube circuits, but that seemed feasible.

So I procured three Arduino Uno type modules, three AD9850 DDS modules, five rotary encoders, and three LCDs, and began experimenting. It occurred to me that if the first oscillator (replacing the crystal heterodyne oscillator) could generate any frequency in the range of 5.9 to 32.9 MHz, I could make a general coverage receiver for 3.0 to 30 MHz by using a 2.9 MHz first IF. To be useable, the first oscillator would need the capability to change frequency in selectable steps ranging from 10 Hz to 1MHz. Plus, the frequency display would need to be accurate to 10 Hz. But all of these requirements were within the capabilities of available Arduino sketches and the hardware I had. The second oscillator in this scheme (replacing the PTO/VFO) would only need to generate frequencies in the range of 2.445 MHz to arrive at a final IF of 455 kHz. This frequency would be changed slightly to accommodate LSB and USB signals. The BFO would need to generate the usual carrier insertion frequencies for the product detector. The preselector would need to cover the entire HF range as well.

With this grandiose architecture in mind, I assembled the first Arduino/DDS/encoder/LCD combo together and started work. The whole assembly and preliminary Arduino sketch were based on the design by AD7C that is available on the internet. I was able to get this working without too much difficulty, and the DDS module produced what looked like beautiful sine waves from 455 kHz to about 40MHz. The output was about 1.0 volts peak to peak at lower frequencies, dropping off to about .5 volts approaching 35 MHz. This was caused by a low pass filter on the module.

Clearly the output of the AD9850 modules would need to be amplified to provide adequate injection signals for the vacuum tube circuits. Since I would need to go as high as 33 MHz, I wanted an amplifier that would do the job with flat response. Based on internet research, none of the circuits using discrete transistor designs would work. VK5TM reported good results with the LMH6703 current mode op amp, so that's the approach I pursued. The module I ended up with was described earlier. There is a lot of information available on this chip at the Texas Instruments website. The circuit I used is their non-inverting gain circuit, as modified slightly for a single ended power supply.

The three Arduino sketches and shields were configured identically, with the capability to connect a rotary encoder and LCD to set and readout frequency. Each shield had an LMH6703 amplifier module installed. The outputs were injected into the cathodes of the first mixer, the second mixer and the product detector, just like the 75S-3 does. An LCD was installed in the front panel, and a rotary encoder was mounted to the chassis in a small minibox, with its shaft brought out to the front panel. These were intended to be the main frequency controls, and were connected to the first Arduino in the signal path, the one that would tune 5.9 to 32.9 MHz. The

other two Arduinos had LCDs and rotary encoders, as shown in Figure 16, temporarily connected for test purposes. The first IF at 2.9 MHz was implemented using two coupled coils, similar to the 75S-3 bandpass IF. Sketches were written and loaded into the modules for the three frequencies that would be required.

Figure 18 shows a top view of the radio in this initial configuration using the three Arduino/AD9850 modules. Figure 19 shows the solid state chassis. As can be seen, everything was well thought out and the installations were nice and neat. There were no extra holes anywhere. Because there were fewer vacuum tubes in the initial configuration, the B+ current demand was lower and I was able to use a 6X4 rectifier tube and smaller filter choke. These are visible next to the power transformer in Figure 18. There was no need for the 0B2 voltage regulator tube in this configuration, since all three signal sources used quartz crystal frequency references. Comparing this layout to the final configuration shown in Figure 2, differences are: a) the 6X4 had to be replaced with silicon diodes, b) regulated B+ was needed for the VFO and crystal oscillators, so the 0B2 used the socket formerly occupied by the 6X4, and c) a larger filter choke was needed, which required swapping locations with the low voltage transformer.

When the initial configuration was finally integrated and testing began, the first thing I noticed was an abundance of "birdies" and spurious signals all across the bands, some occurring every few kHz or so. Changing the frequency of the first oscillator by as little as 10 Hz would cause changes in the birdie frequencies and overall tones, not by a few Hz as expected, but by unpredictable amounts. Changing the second oscillator frequency caused similar results. Substituting a first oscillator signal from my HP 8640B signal generator eliminated some of the noises, but changing the second oscillator frequency would still cause digital sounding tones. Since I didn't have a second signal generator, I was unable to test the effects of two clean oscillator signals.

From the digital sounding nature of the noise I was getting, I suspected the DDS chips were generating some kind of low level spurious signals, or "spurs". So I decided to look at the AD9850 signals on my HP 141T spectrum analyzer. I was unable to see any significant spurious signals within the 60 db or so of useful dynamic range on the analyzer. I also tried all sorts of additional bypassing, decoupling and moving wires around; none of these made the slightest bit of difference.

It was then that I decided to see if other hobbyists have encountered similar issues with these DDS chips. There is a lot of information out there on how to make circuits and Arduino sketches with the AD9850, but what about problems? Doing an internet search on the keywords "spurious noise problems with ad9850" returned several discussions, including this blog on eham: http://www.eham.net/ehamforum/smf/index.php/topic,115343.0.html.

By far, the most sensible explanations I found anywhere on the internet were those by N3QE found in the above blog. It is recommended reading for anyone thinking about using this chip. Especially interesting was his comment, "…do small changes like a single step frequency change in the DDS VFO cause a complete change in the buzzes and whines? If so, you are getting the spurs." Well, that is exactly the way my radio was behaving. And as N3QE goes on to explain, there are no simple methods to make it better.

More internet research revealed that predicting the exact characteristics of these spurs at any given frequency is apparently a complex math exercise that is impractical. And some of them will be very close to the desired carrier. The Analog Devices data sheet for the chip specifies that the Spurious Free Dynamic Range (SFDR) is at least 50 db down from the desired carrier. While that may sound good, it must be realized that in my receiver's architecture there is perhaps 80 db

of IF gain after the second mixer that is provided by the two 6BA6 amplifiers. I suppose the mixers see the close-in spurs as some type of modulation that is indistinguishable from the input signal, so it appears in the mixer output products along with the desired signal. Since these products are close-in, they can't be filtered out by any reasonable means. Needless to say, these results were very disappointing. And unfortunately it required integration of the complete radio for the problems to become apparent.

Nevertheless, in a last ditch effort to stay with my initial approach I decided to try the Si5351 chip, since it uses an entirely different method to generate a clock signal. Plus the QRP Labs Si5351 modules had the same footprint as the AD9850 modules, so the conversion would not be too difficult. I procured and built several kits from QRP Labs, rewired the Arduino shields, revised the sketches and began testing. Results were just about the same as the AD9850, with digital sounding tones across the spectrum that would change with the slightest step in output frequency. Since this chip generates square waves, I thought maybe that was the problem. So several low pass filters from QRP Labs were acquired, assembled and installed in an attempt to filter out the harmonics. These made little to no difference. I could totally bypass the filter and the noise stayed the same, once again suggesting I was dealing with close in spurs that can't be filtered out.

At this point I was no longer inclined to pursue the solid state approach for the first two signal sources. I realized that to make the radio work I was going to need a crystal controlled oscillator for the first and a VFO for the second, and I might as well use vacuum tubes. When I started looking at available crystals, it soon became apparent that the 75S-3 architecture would have to be abandoned. But there were cheap microprocessor crystals readily available in the frequencies needed for a Drake 2-B mixing scheme. This would in turn require a 3.5 to 4.0 MHz bandpass for the first IF that would synchronously tune with a 3.955 to 4.455 MHz VFO. I just happened to have a three section variable capacitor from a junked ARC-5 receiver that would do the job. It had a beautiful vernier drive mechanism. Two of the sections could tune the variable IF bandpass coils, while the third could handle the VFO. So once I figured out how to mount this variable, the final configuration of the radio described earlier quickly emerged. A Si5351 module was still used for the BFO. When I powered up the new configuration and began testing, there were no birdies!

There were a couple of other modifications on the way to the final configuration worth mentioning. The first iteration of the crystal oscillator used the pentode section of the V2 6U8 connected as a triode (i.e., the screen grid was connected to the plate). I then used the triode section of that tube for the first mixer, mostly because of all the fanfare about triodes being less noisy mixers than pentodes, and the fact that the 75S-3 uses the triode section of a 6U8 for its first mixer. Then again, the Drake 2-B uses the pentode section of its 6U8 for the first mixer, and it seems to do just fine. Well, it kind of concerned me that with my new Drake 2-B architecture I would be using a triode as a straight through amplifier on 80 meters. This seemed like it might turn an amplifier into an oscillator. While I didn't notice any outright oscillation, I did eventually discover that on 80 meters the bandpass IF seemed to have a lot more gain at the 4.0 MHz end. Well, sure enough I was getting regeneration that increased at the higher frequency. Changing over to the pentode section of the 6U8 put a screen grid in the way of the signal grid and plate, and stopped the problem. So now I think I know why the Drake 2-B did the same thing.

Another feedback problem occurred between the plates of V4 and V7. The plate of V4 has the input of the first IF on it, while the plates of V7 carry the output of the second IF. The tubes

are about three inches apart, which was just close enough to almost cause an oscillation with the IF gain control at maximum. It took a lot of troubleshooting to run this one down. As seen in Figure 2, these tubes now have shields which completely eliminated the problem.

Final Thoughts

It took about a year and a half to get this radio to its final configuration and complete the documentation. A lot was learned along the way, especially in regard to Arduino technology. If my initial configuration had worked, no doubt the total project time could have been cut in half.

Although I had my share of problems with the Arduino controlled RF synthesizers, I am in no way suggesting that these clever devices should be avoided like the plague. They are inexpensive to buy and fun to work with, including writing the Arduino sketches. My goal in this TD is simply to point out some of their limitations, and advise any prospective users to research their applications beforehand. If you are going to use one of these with a vacuum tube radio, I would strongly suggest building up a prototype module and testing it out before drilling any holes or cutting metal. Who knows, maybe yours will work just fine. -MB

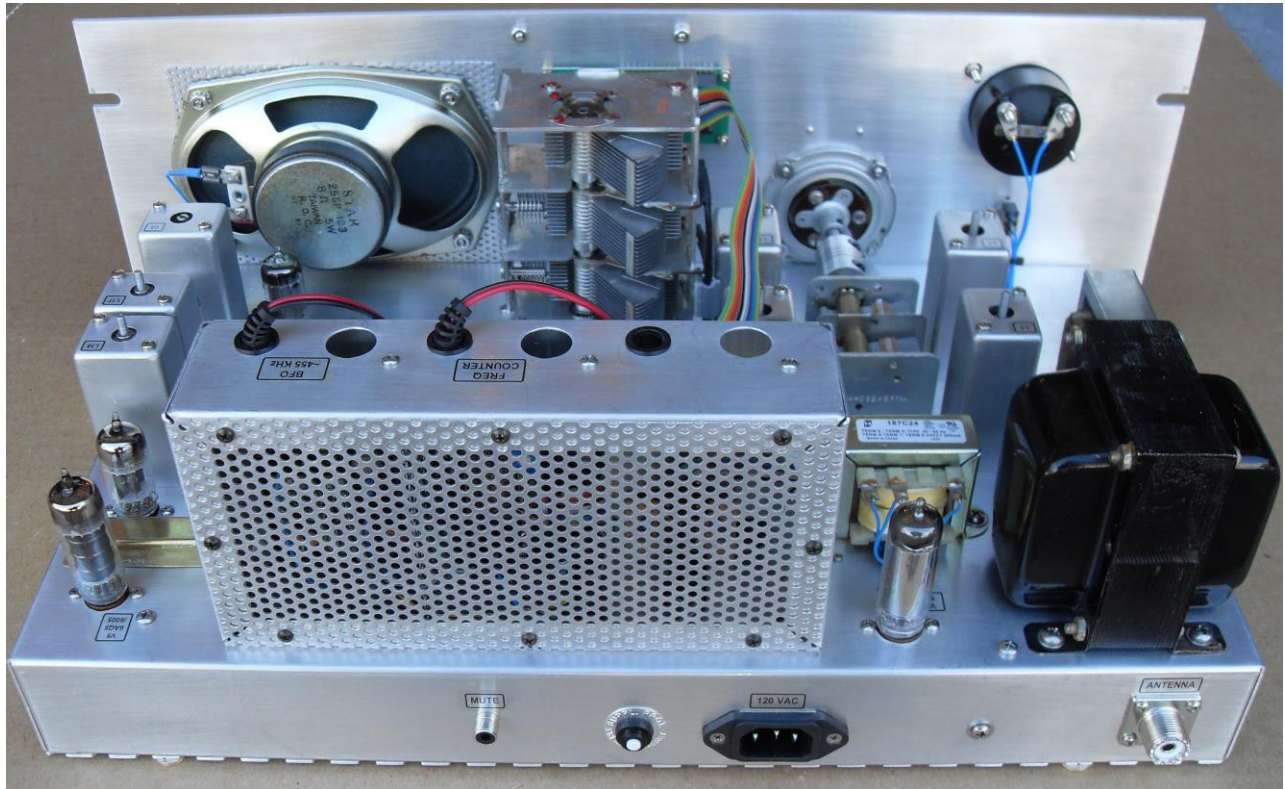**Figure 1: The 75S-2B Receiver**
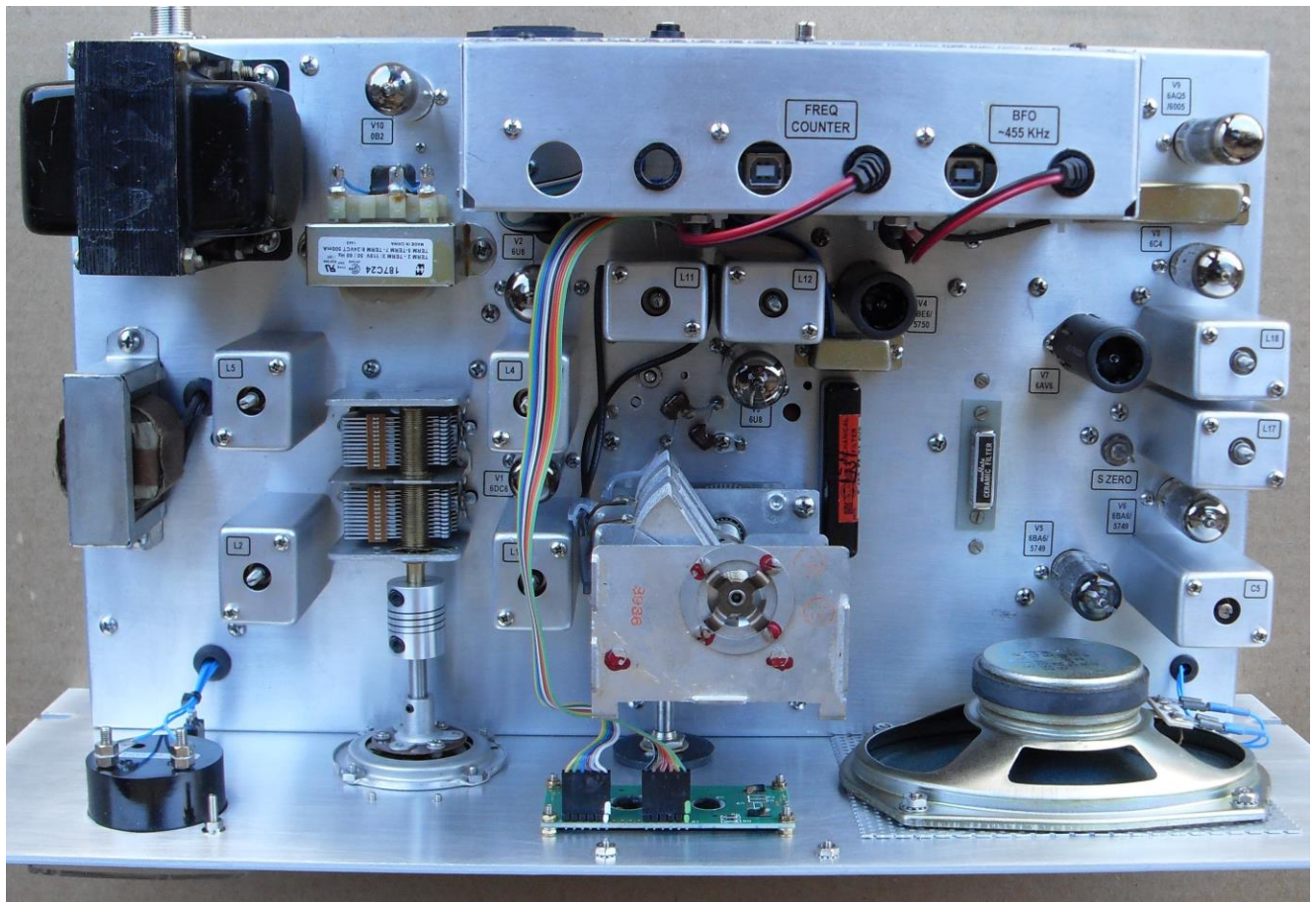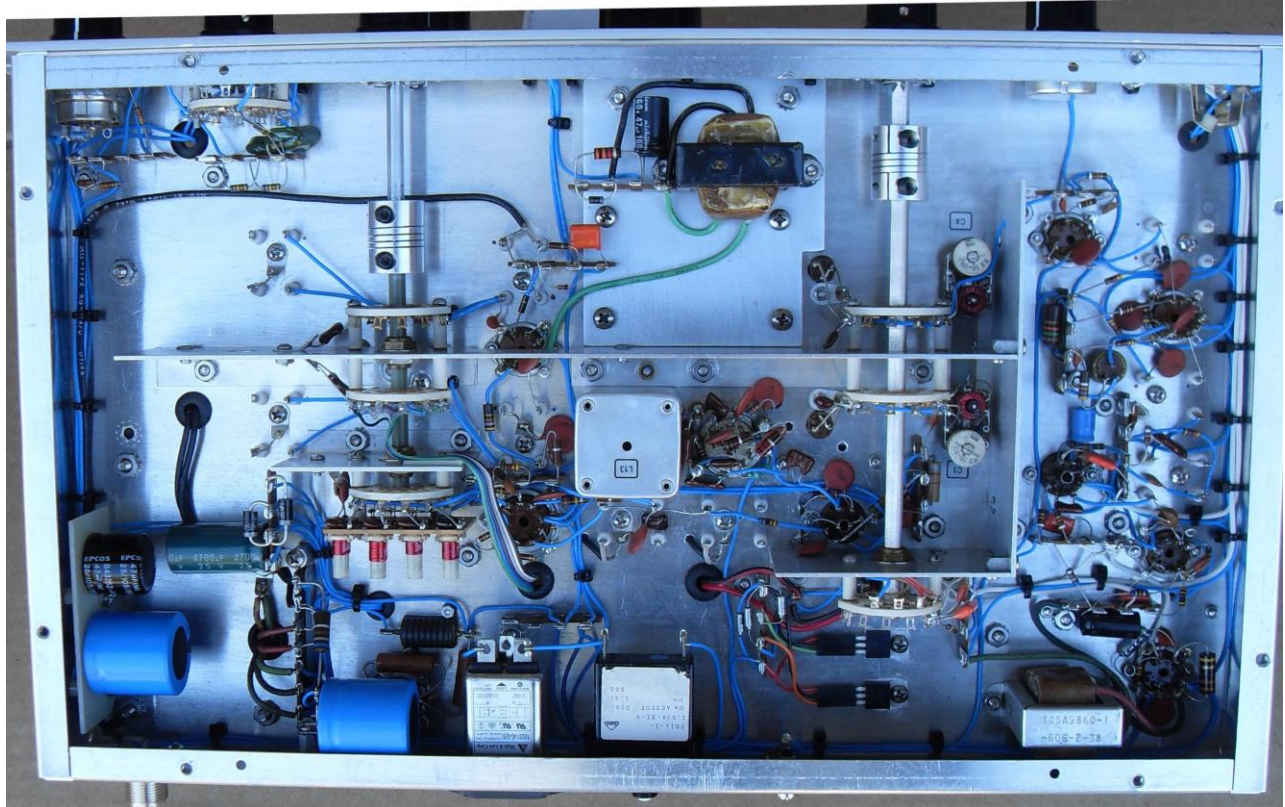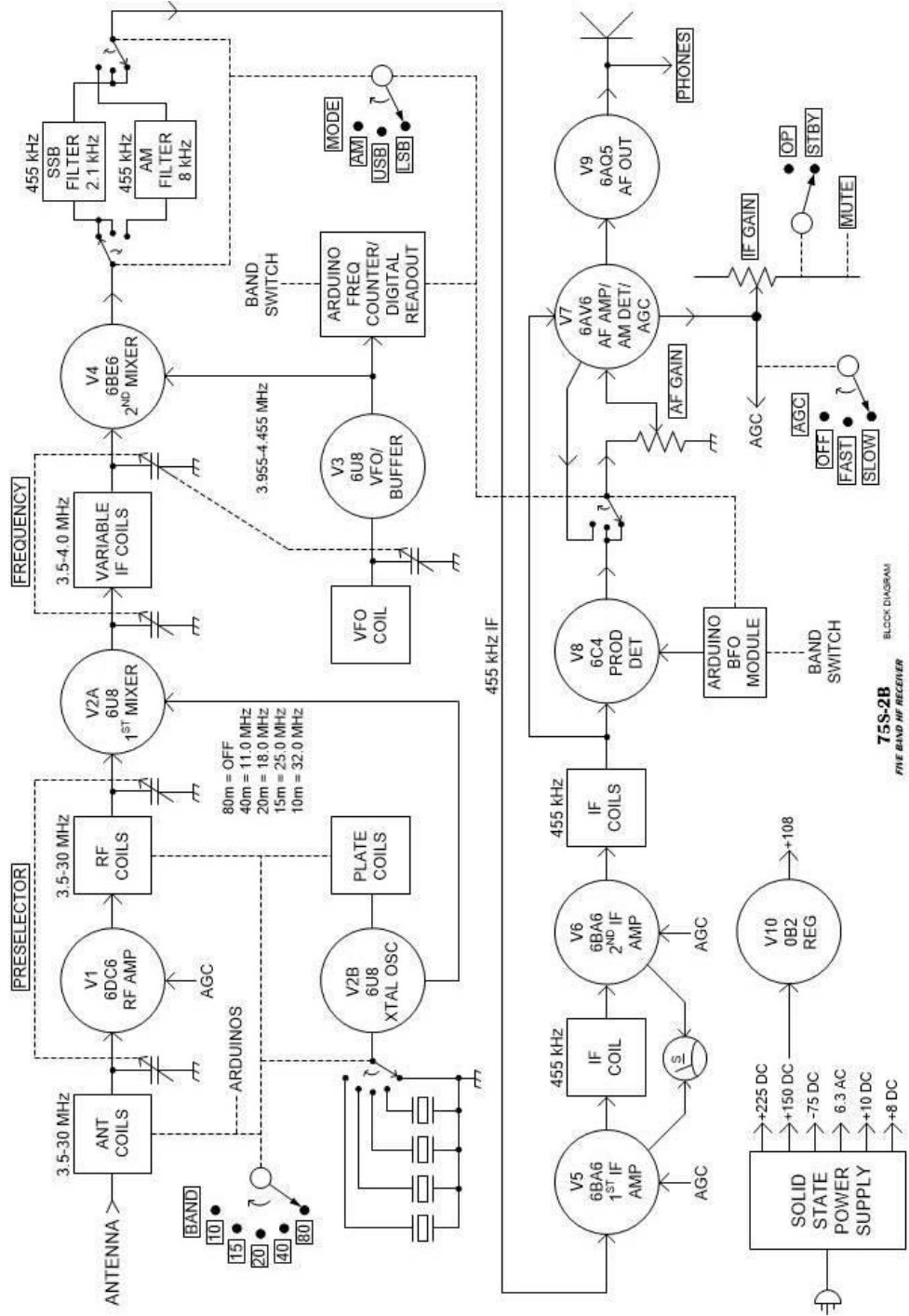

Figure 2: Top View

Figure 3: Rear View

Figure 4: Chassis Underside View

Figure 5: Close Up of Preselector and Main Tuning Controls

Figure 6: Receiver Block Diagram

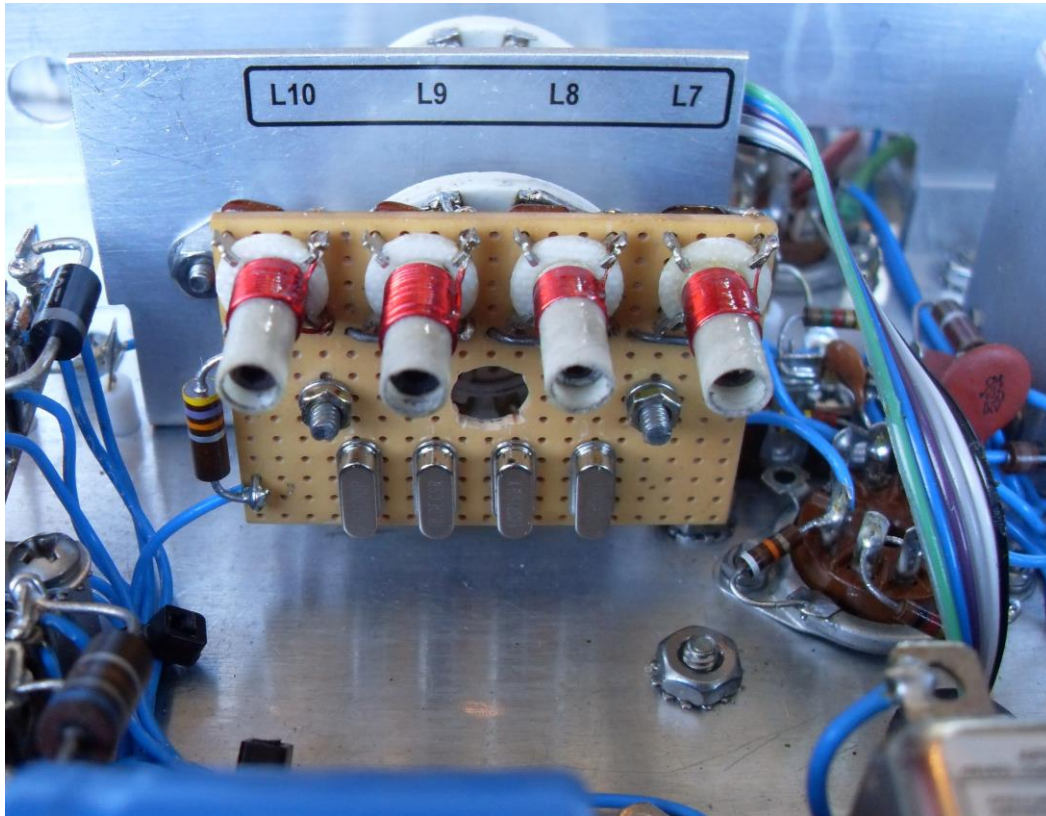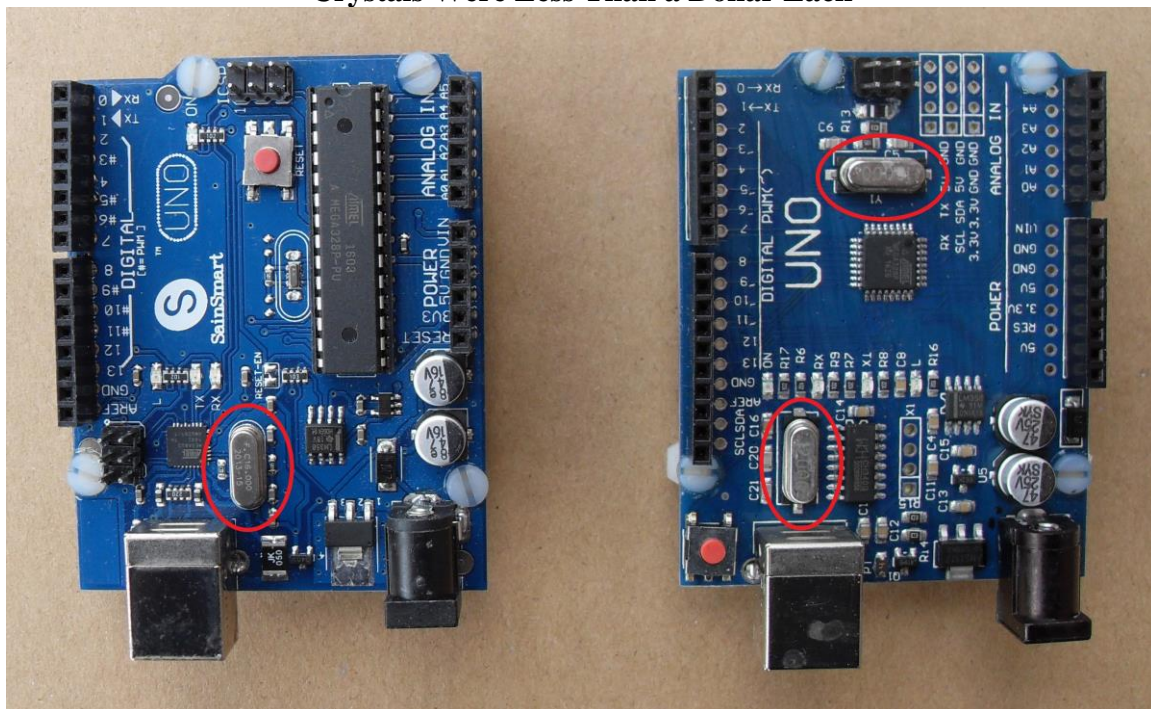**Figure 7: Schematic of Hollow State Circuits**

**Figure 8: Close Up of Crystal Oscillator Board Mounted to Bandswitch
Crystals Were Less Than a Dollar Each**

**Figure 9: Typical Arduino Uno Modules
Module on Right Has Two Quartz Crystals and is the One to Use for a Frequency Counter**
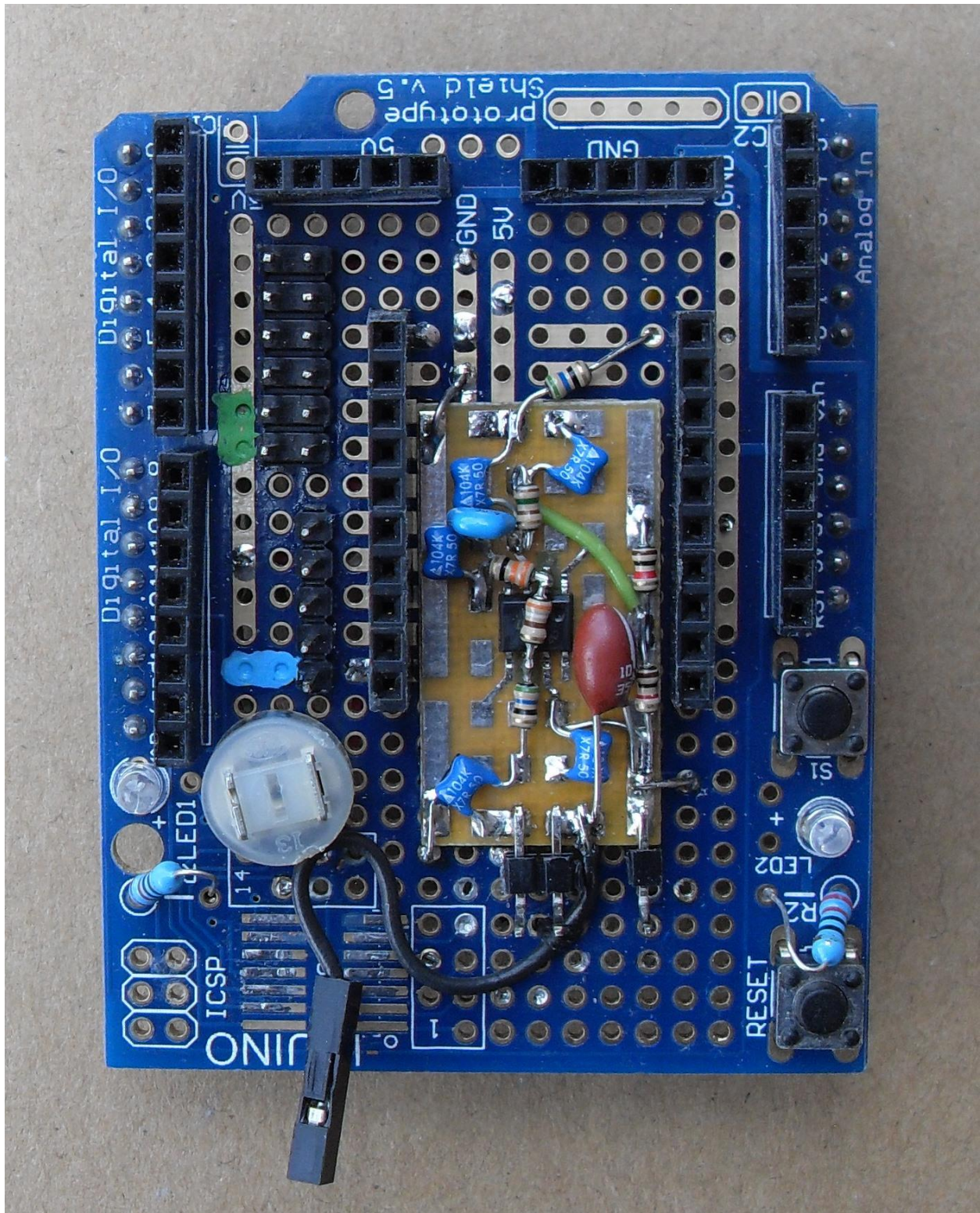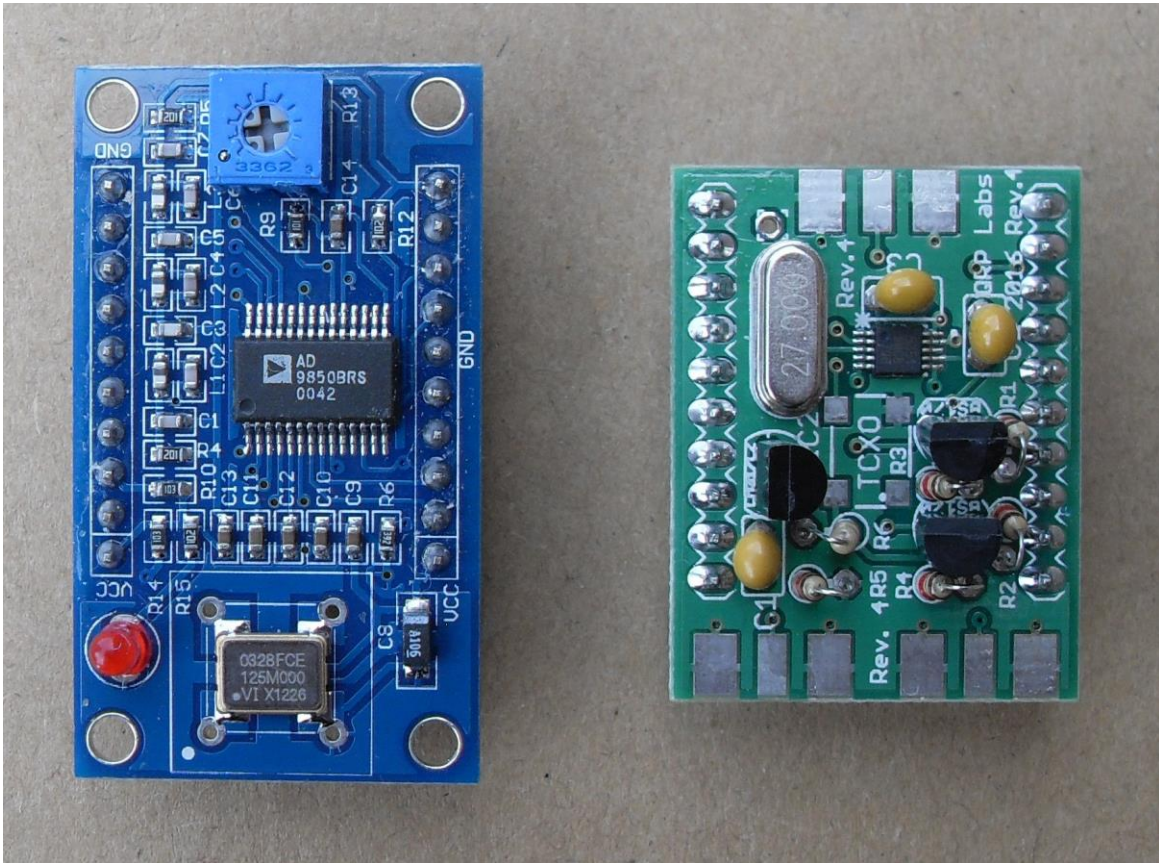
Figure 10: Typical Arduino Shield, Shown with Amplifier Module in Place

Figure 11: Two Popular Synthesizer Modules
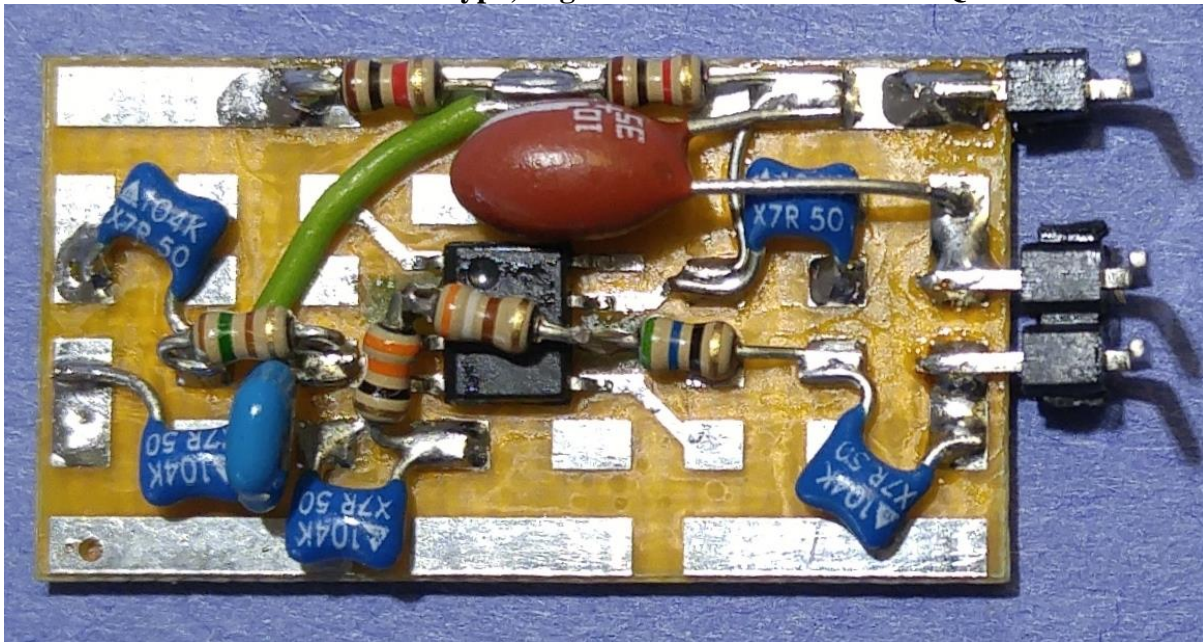**On Left is the AD9850 Type; Right Shows Si5351 Kit From QRP Labs**

Figure 12: Homebrew LMH6703 Amplifier Module

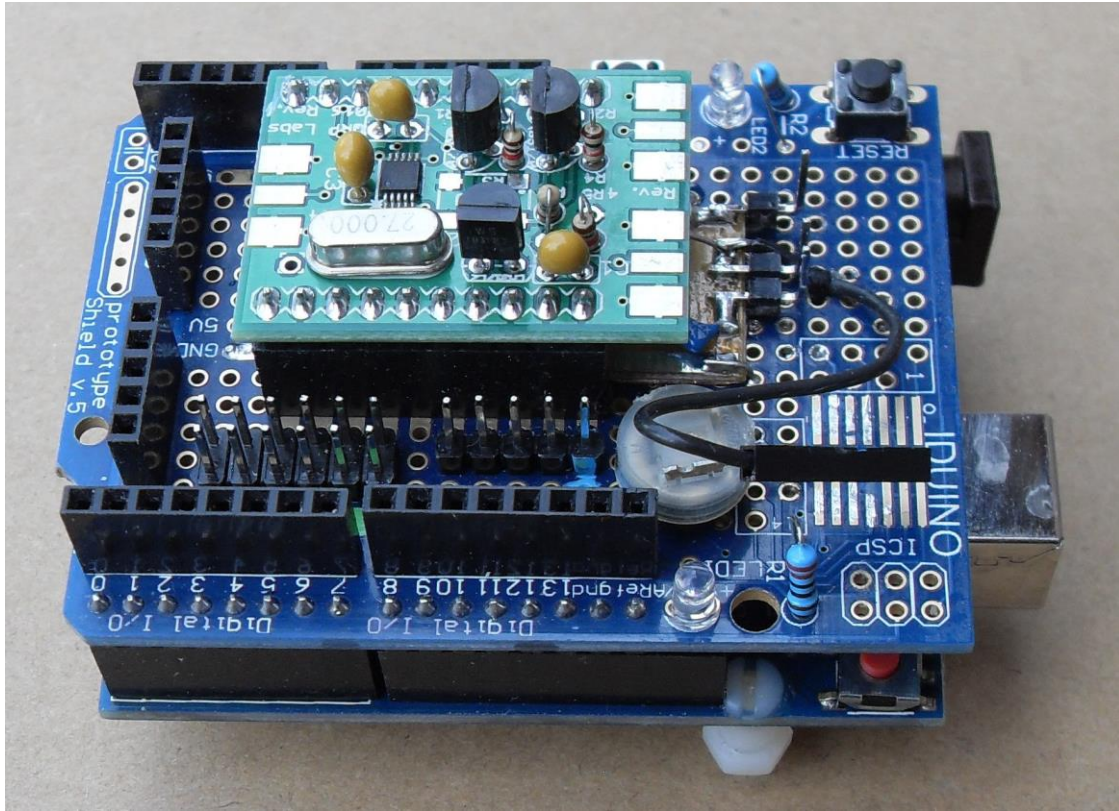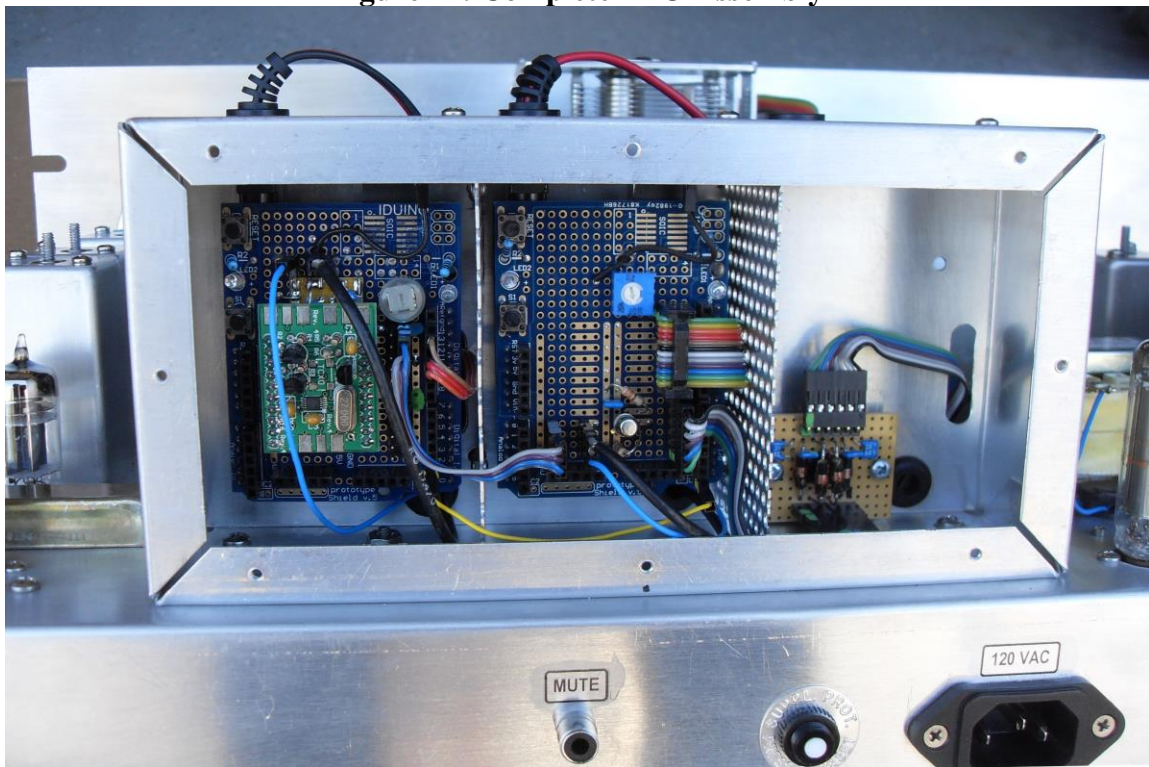**Figure 13: LMH6703 Proto Board and QRP Labs Si5351 Kit**

Figure 14: Complete BFO Assembly

Figure 15: Inside View of Solid State Chassis
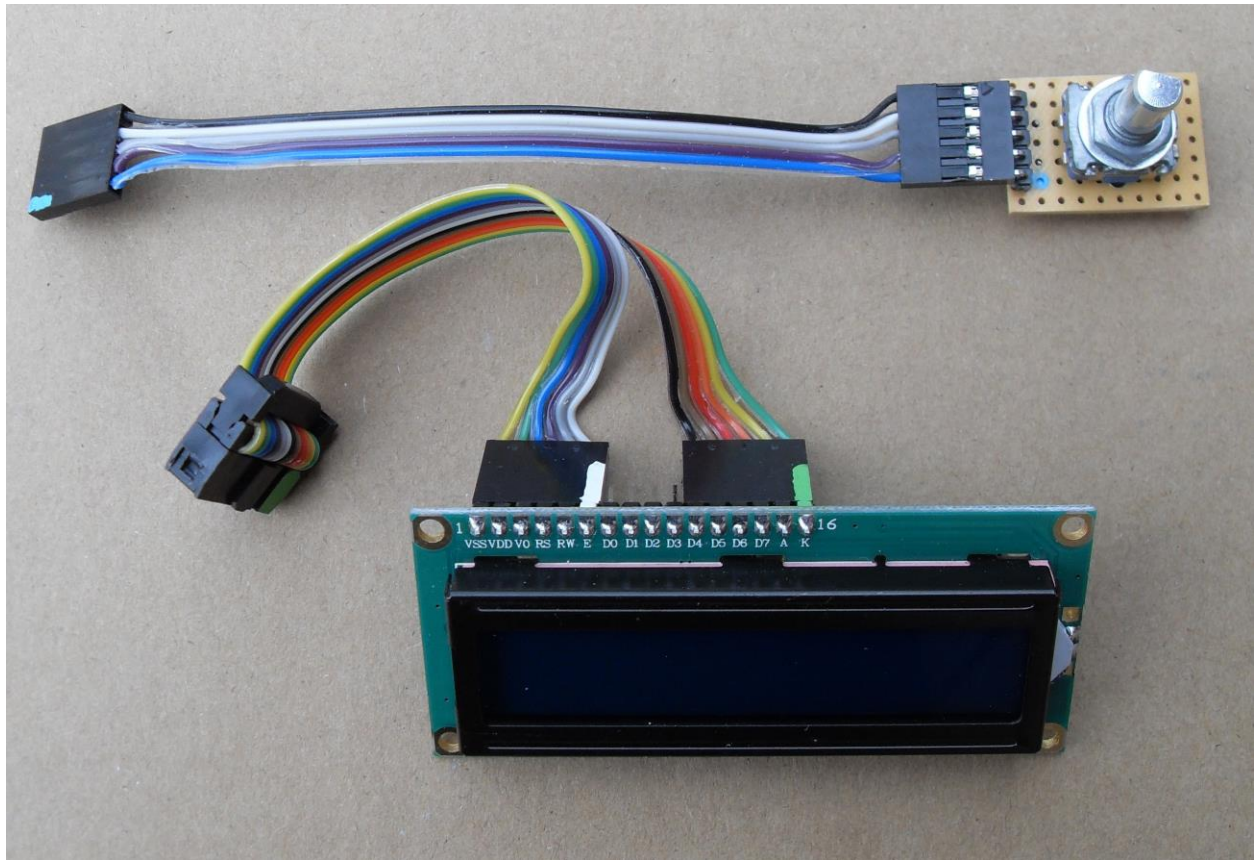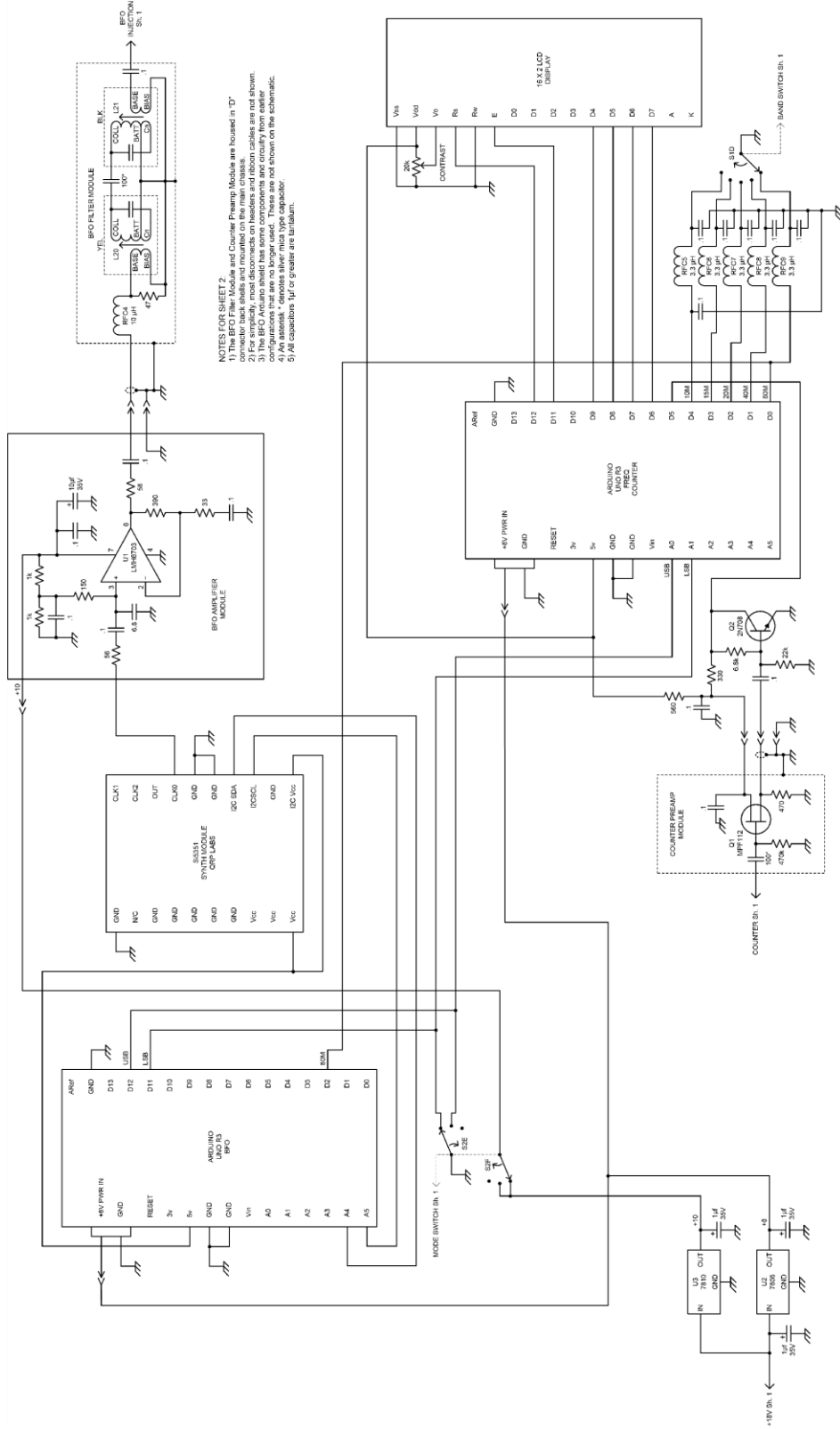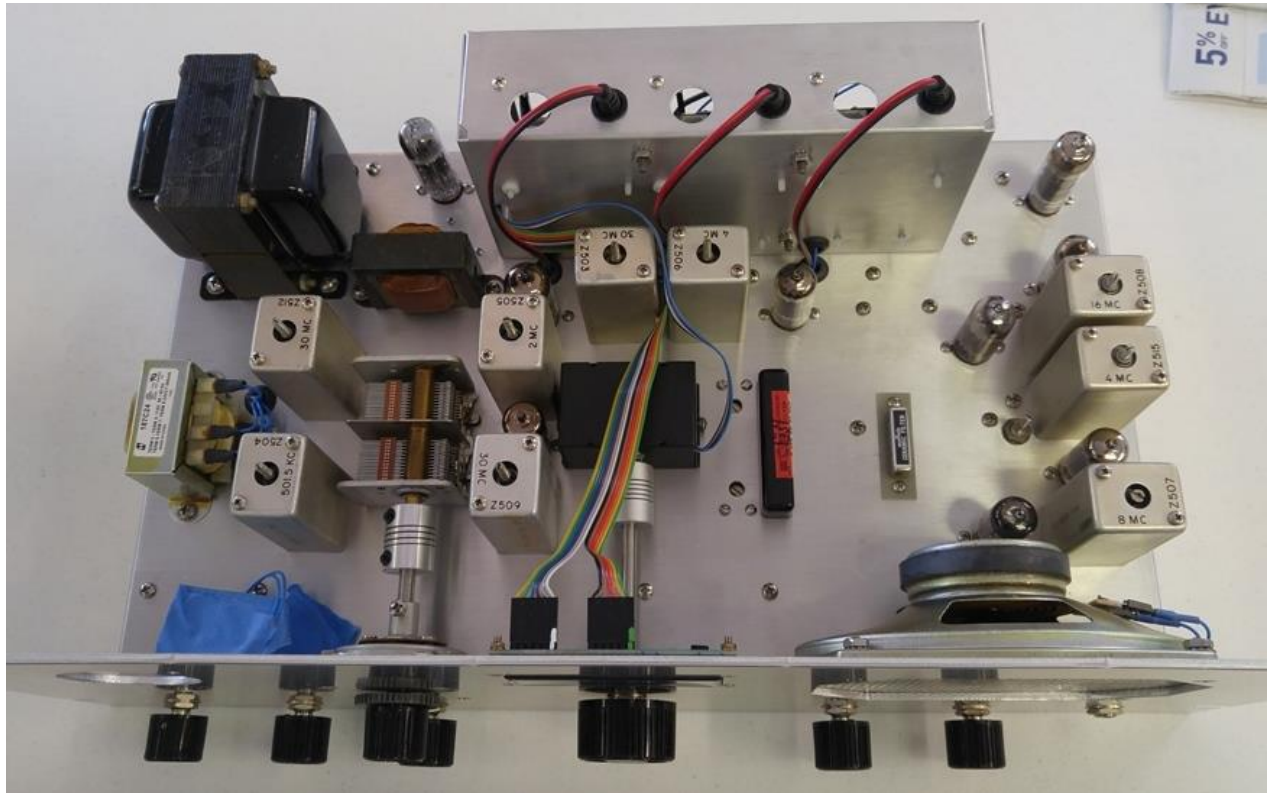BFO (left), Counter (center), Counter Filter (right)

Figure 16: Rotary Encoder and LCD with Ribbon Cables Attached

BFO FILTER MODULE

BLK
L21
COIL
BATT
100'
BASE
BIAS

YEL
L20
COIL
BATT
BASE
BIAS

RFC4
10 µH
47?

BFO
INJECTION
Sh. 1

NOTES FOR SHEET 2

1) The BFO Filter Module and Counter Preamp Module are housed in "D"
connector back shells and mounted on the main chassis.
2) For simplicity, most disconnects on headers and ribbon cables are not shown.
3) The BFO Arduino shield has some components and circuitry from earlier
configurations that are no longer used. These are not shown on the schematic.
4) An asterisk * denotes silver mica type capacitor.
5) All capacitors 1µf or greater are tantalum.

BFO AMPLIFIER MODULE

10uf
35V
.1
1k
1k
.1
100
6.8k
.1
56
U1
LM7070S
56
390
33
1

+10

Vss
Vdd
V0
Rs
Rw
E
D0
D1
D2
D3
D4
D5
D6
D7
A
K
16 X 2 LCD
DISPLAY

20k
CONTRAST

BAND SWITCH Sh. 1

S1D
RFC5 3.3 µH
RFC6 3.3 µH
RFC7 3.3 µH
RFC8 3.3 µH
RFC9 3.3 µH

ARef
GND
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
10M
15M
20M
40M
80M

ARDUINO
UNO R3
FREQ
COUNTER

+6V PWR IN
GND
RESET
3v
5v
GND
GND
Vin
A0
A1
A2
A3
A4
A5
USB
LSB

Q2
2N708
6.8k
330
22k
1
560

COUNTER PREAMP MODULE

Q1
MPF113
100'
470
470k
.1

COUNTER Sh. 1

CLK1
CLK2
OUT
CLK0
GND
I2C SDA
I2C SCL
GND
I2C Vcc

Si5351
SYNTH MODULE
QRP LABS

GND
N/C
GND
GND
GND
GND
GND
GND
Vcc
Vcc
Vcc

ARef
GND
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
USB
LSB
ROM

ARDUINO
UNO R3
BFO

+6V PWR IN
GND
RESET
3v
5v
GND
GND
Vin
A0
A1
A2
A3
A4
A5

MODE SWITCH Sh. 1
S2E
S2F

+10
1uf
35V
+9
1uf
35V

U3
7810
IN OUT
GND

U2
7808
IN OUT
GND

1uf
35V

+18V Sh. 1

75S-2B
FIVE BAND HF RECEIVER
HOMEBREWED BY KG7TR

SHEET 2
SOLID STATE CIRCUITS
REVISION 22 FEB 2021

Figure 18: Top View of Radio in Initial Configuration
Rotary Encoder is in Black Minibox in Center of Chassis

Figure 19: Solid State Chassis in Initial Configuration
Three Arduinos and Three AD9850 DDS Modules were Used