Crystal Bank for Vintage Gear

It used to be one could go to a ham fest, rummage around for an hour or so, buy 15 or 20 used crystals, and, depending on one's negotiating ability, return with them and enough change from a $20 bill to buy a hamburger and a coke. Folks, where I live, those days are gone. Today I'm lucky if I can get a ham band crystal for less than $5 or $10, assuming you can find them. It makes operating a boat anchor with any frequency agility both problematic and costly.

I began to examine direct digital synthesis (DDS) as a solution to the scarcity of piezoelectric crystals.

The inexpensive (about $28 from Mouser) AD9850 chip (http://www.analog.com/static/imported-files/data_sheets/AD9850.pdf) provides a very nice basis for a synthesized frequency source. However, I didn't want to go to the trouble and expense of designing an AD9850 controller and all of the necessary supporting electronics and software to interface to one. In addition, the AD9850 is a surface mount device (SMD) and I just don't have the equipment, inclination, or patience to tackle a SMD project.

I examined other AD98xx implementations hams have developed and are marketing and about a year ago and purchased and built the N3ZI DDS2 variable frequency oscillator (VFO). (http://www.pongrance.com/super-dds.html). In this VFO an AD9834 DDS SMD is already soldered to the board and the other components are easily soldered using a small tip soldering iron.

The N3ZI DDS2 VFO drove my Viking II after I designed and built a buffer amplifier for it; but tuning was sometimes jerky with the mechanical contact rotary encoder Doug supplies with his kit. I was able to upgrade this VFO with a high quality optical rotary encoder and now obtain very smooth tuning that emulates an analog VFO touch and feel.

Due to the unplanned additional price of the optical encoder, the overall cost for this VFO exceeded the $150 limit I had initially established. Nevertheless, it turned out to be a very nice addition to the shack; albeit a bit of an overkill as all I really needed were 20 or 30 discrete frequencies. The other problem was the amount of RF feed through that was detectable in my receiver. The VFO has an offset capability, so I used that feature to solve that problem, but it meant the displayed frequency, when receiving, was always lower that the actual transmit frequency.

Recently Doug (N3ZI) began selling a mother board that will allow two AD9850 daughter cards to be plugged in and obtain 64 discrete frequencies (http://www.pongrance.com/DDS-9850.html).

The daughter cards are easily obtained from eBay at a price, including shipping, of about $5.00 as of this writing (just search for DDS 9850). When you compare these cards to the basic price of the AD9850, they are quite a bargain. A picture of one of these daughter cards is shown below.

These cards include a 125 MHz clock as well as all of the necessary supporting components and interfaces to two standard 0.1" header pins (10 pins each side). Using the N3ZI motherboard, two such DDS cards are needed to obtain 64 channels, but you could just use one DDS card if you only require less than 32 channels.

I worked up a budget for this project and came up with the following estimate:

- N3ZI Xtal Bank Board - $30
- 2 DDS Daughter Cards - $10
- Control Circuit - $20
- Metal Enclosure - $30
- Shipping - $10

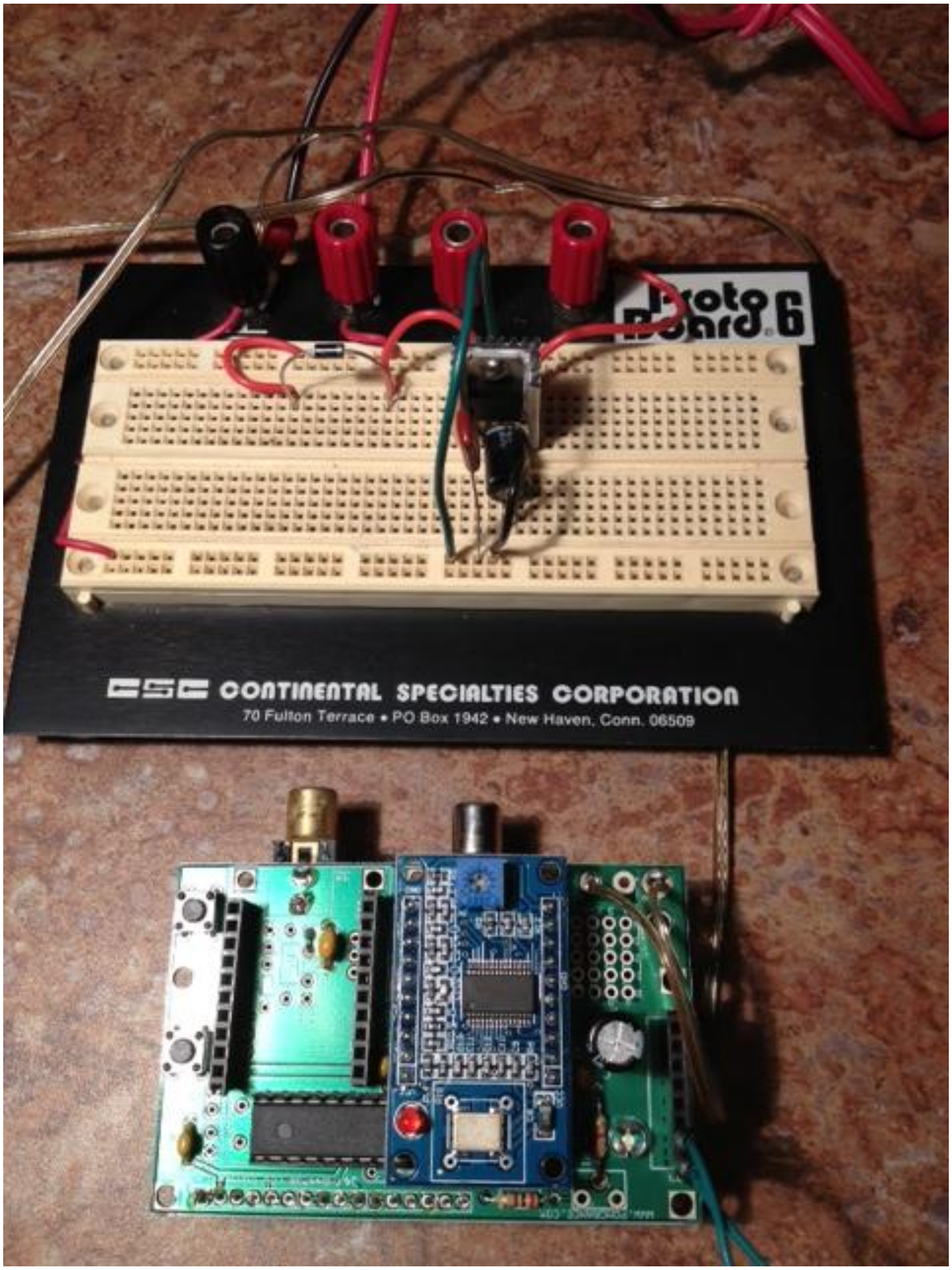
Total - $100

This estimate does not include the LCD 1602 display and a few other parts, such as connectors, header pins, wire, and switches that I had on hand in my parts supply. If you need to buy these additional parts, then the price could increase considerably.

The biggest cost item is the enclosure. While you can get a less expensive plastic enclosure, a metal enclosure is beneficial in reducing stray RF from either leaving the crystal bank or becoming susceptible from strong radio frequency in the shack.

The remainder of this paper discusses some of the engineering issues I encountered along the way in building this crystal bank. I'll also briefly discuss the use of an Arduino UNO or a Texas Instrument LaunchPad to replace the N3ZI mother board in case you would like to build your own controller or even use one of these DDS cards as the basis for a highly stable VFO.

Initial Testing and Breadboard: The N3ZI card uses 5 vdc. Initially a 1N4007, 7805 regulator, heat sink and two capacitors were lashed up to provide a source of 5 vdc from my 12 vdc bench supply. It turns out that the current drain (~325 mA) of the N3ZI card and the two DDS cards caused the heat sink to get quite hot. While it was ok for initial testing, I decided to approach the power source slightly different for the final implementation. More on that later. Here is a picture of the initial testing. Only one DDS card is installed here. The 5 vdc power supply is on the breadboard.



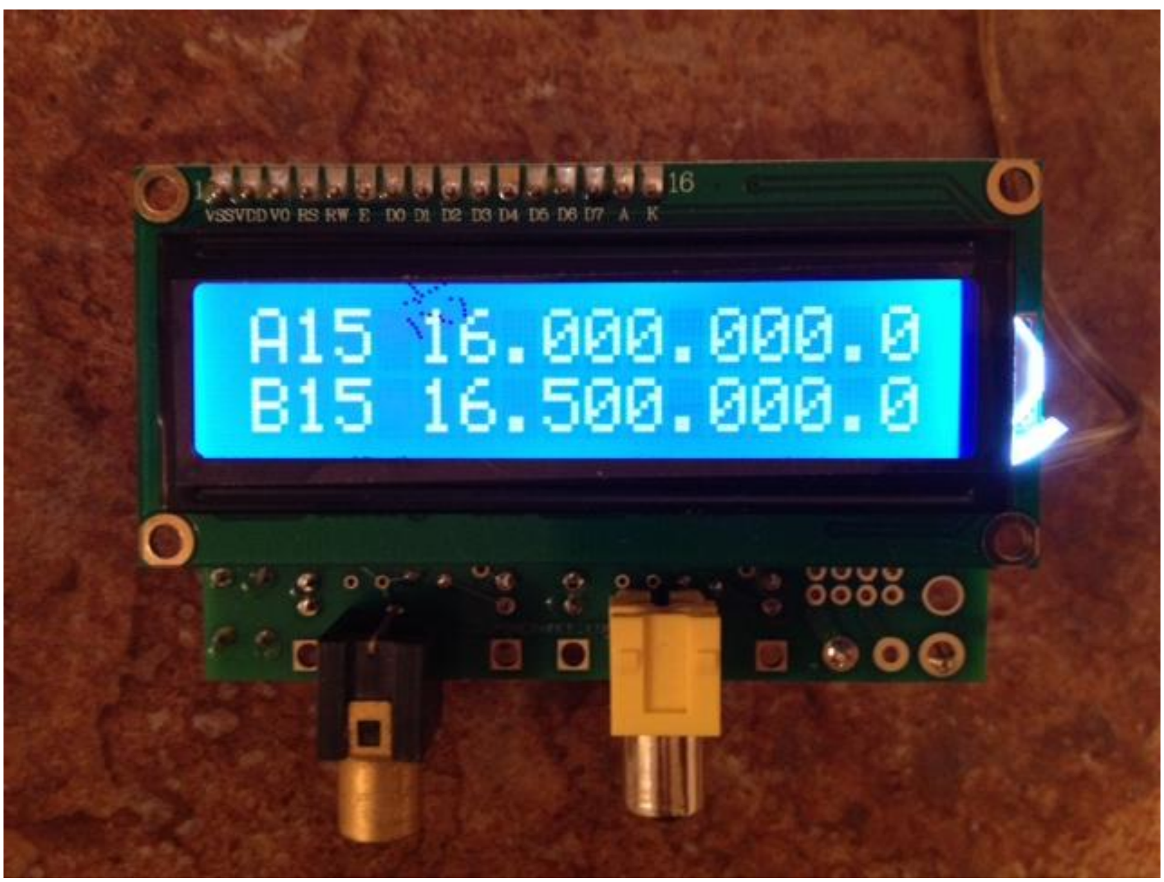
There is a tuning command on the N3ZI that takes a 5 bit word. Each of these are bits are held high with pull up resistors, so you need to short them to ground to enable a zero. Hence the default channel is 31. See the digital to 5 bit table below to review.

Channel		Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
0		0	0	0	0	0
1		0	0	0	0	1
2		0	0	0	1	0
3		0	0	0	1	1
***						
31		1	1	1	1	1

The top row of an optional LCD display shows DDS card Axx ff.fff.fff.f, and the bottom row card Bxx ff.fff.fff.f, where xx is the channel number and f.fff.fff.f is the frequency displayed in Mhz.kHz.Hz. and either .0 or .5 Hz.

The frequency stability of these units is on the order of ½ of a Hertz or less. More on frequency stability and accuracy later, but rest assured, they are certainly more than adequate to use as a stable frequency source for HF and possibly VHF applications.

In the picture below, bit 5 is shorted to ground hence channel 15 has been selected. Note that bank A's channel 15 frequency has been set 500 kHz below that of bank B's channel 15. Each bank channel can be set independent of each other. Instructions on how to set frequencies are provided later. The output of bank A is the RCA phone jack to the right, bank B is to the left.

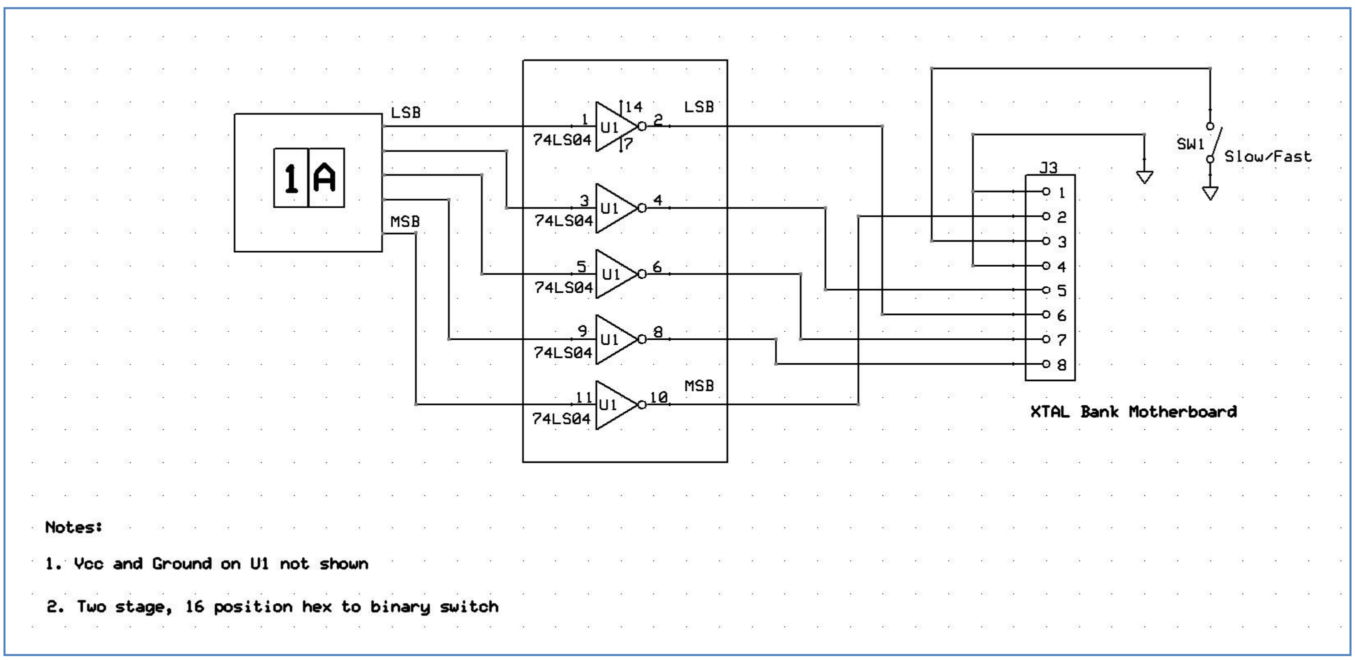


Channel Selection Circuit: Since the N3ZI board requires five bits to be selected in a word so as to access a specific channel, some sort of channel selection circuit needed to be designed and implemented.

Initially I was thinking about just using five SPST switches to ground or a small dip switch to select the binary code pertaining to each channel, but then decided I wanted an easier and more operator friendly way to increment or decrement to the next channel. My plan is to group frequencies by band and then within that grouping have sequential frequencies to select for operating convenience.

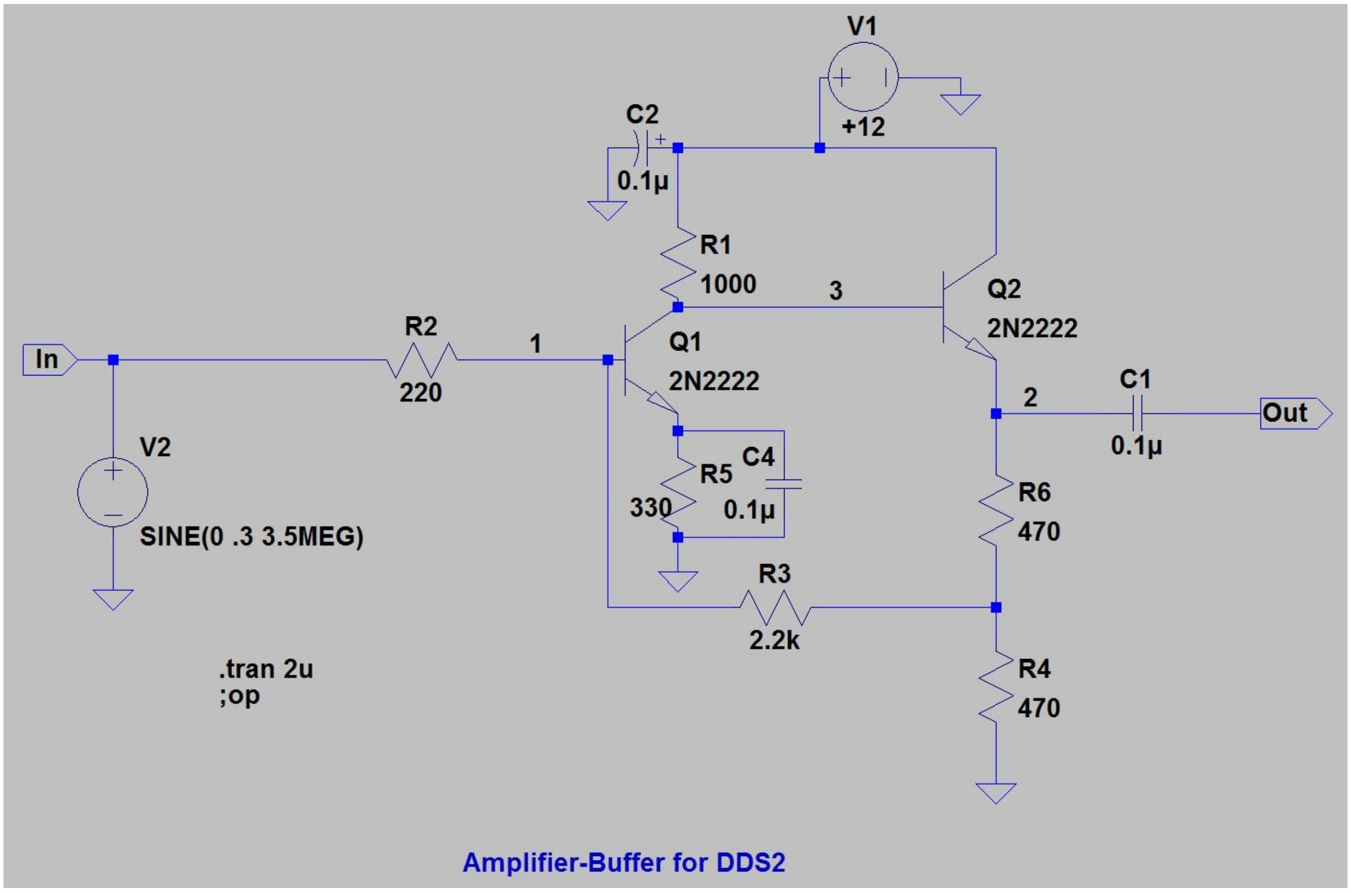
A quick look on eBay uncovered a 16 position, hexadecimal to binary, dual segment, rotary switch. Each segment can count up to 16 channels. In the diagram below, hex 1A is actually decimal channel 26. Since the LCD will provide you with the decimal readings, this seemed like an easy way to select the channels especially if you implement the optional LCD. Each segment has four contacts representing bits 1, 2, 4, and 8 (LSB to MSB). The left switch segment only requires a connection to bit 1, since that is actually the MSB that will be presented to J3, pin 2. The common pin on each segment is connected to ground.

U1, a 74LS04, is necessary to properly invert the bit sense (e.g., when a bit is selected in the switch it is grounded). But since the mother board has pull up voltages applied, we want to represent a bit high as a 5 vdc digital signal. U1 does that inversion for us.



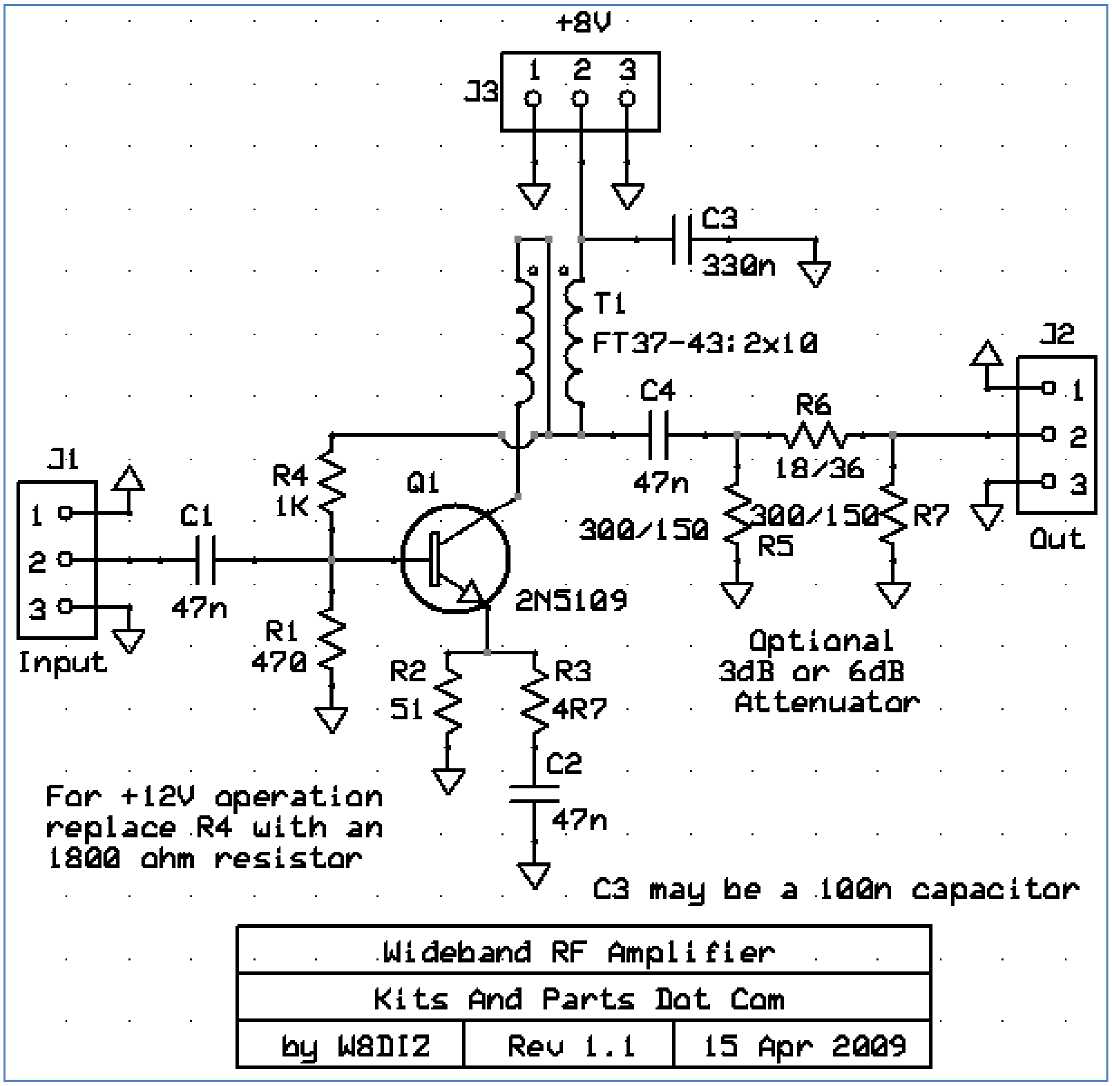
SW1 on pin 3 of J3 is not part of the channel selection process, but is used as an aid when loading a frequency into a channel. We will discuss frequency assignment shortly.

Buffer Amplifier Circuit: You will need to build some type of buffer circuit in order to drive many of the boat anchor transmitters. The output from the AD9850 is on the order of 200-400 mv p-p, but when loaded with 50 Ohms is significantly less. When I built my N3ZI DDS2 VFO I used LtSpice to design a small two stage 2N2222 buffer amplifier. That amplifier works up to about 10 MHz but after that the output starts to fall off fairly rapidly. However, since most boat anchors have doubling or tripling circuits, I didn't find that an issue since the fundamental frequencies are less than 10 MHz. Here is the circuit that I used with my DDS2 VFO.



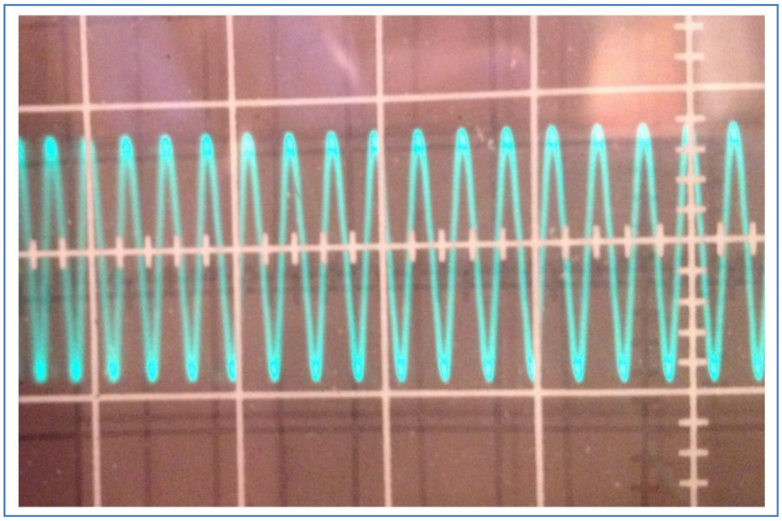
One additional problem with this amplifier was the fact that it didn't produce a nice sine wave output. It appeared there was some clipping and distortion of the input signal. Again, since most of the boat anchors have tuned circuits, this lack of fidelity wasn't a significant problem and it didn't seem to cause any issues with my EF Johnson Viking II on 160 through 10 meters. However, in some instances, it may serve to trigger parasitic oscillations and/or spurious emissions.

I decided to look at other designs and ran across a small 2-30 MHz, 20 dB gain amplifier for less than $10 at kitsandparts.com (http://www.kitsandparts.com/rfamp1.1.php). Below is a schematic of that amplifier.



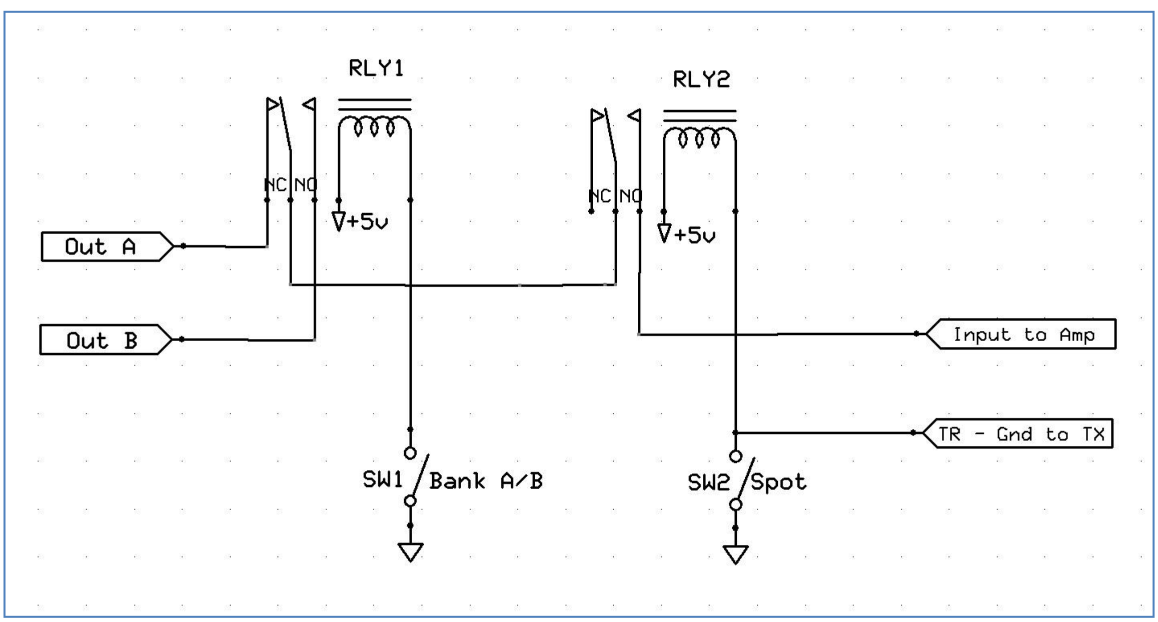
So the choice is pretty much up to what one wants to do – the amplifier/buffer design of mine seems to be adequate and drives a low impedance of 50 Ohms quite well. But the amplifier by W8DIZ has better fidelity of the input signal, provides RF output in excess of 1 volt p-p driving into a 50 Ohm load at 16 MHz, can be powered by 5 vdc, and has a fairly constant gain to 30 MHz.

In this implementation I used the W8DIZ amplifier, although it did bump my budget by about ten percent. Note that the optional attenuator circuit was not used. A picture of the nice sine wave output of 16 MHz is shown below.



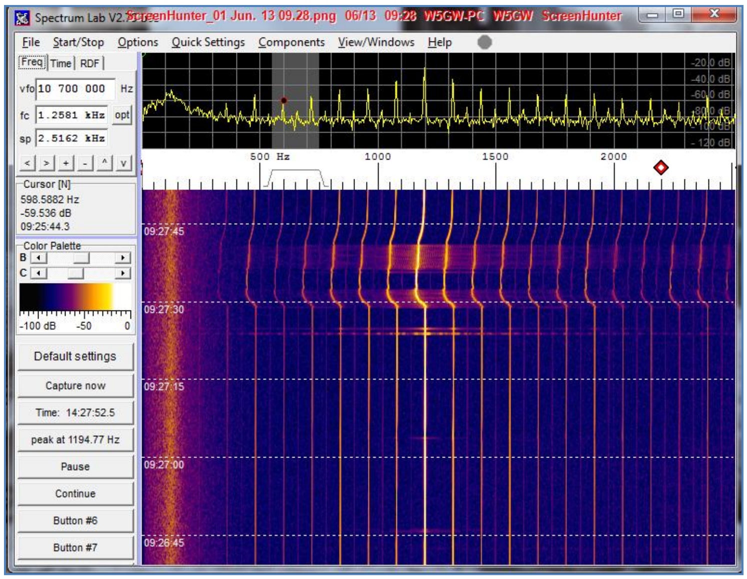
Another circuit was needed to be built to select either channel bank A or B. Additionally, I wanted to mute the output to prevent from hearing the signal while receiving. That, with a metal enclosure should be sufficient to prevent any RF from reaching the receiver while in receive mode. Be sure to use coax for the RF from the crystal bank to the transmitter, in my case I used a 2 foot section of RG 8X. A set of contacts from my T/R switch are used to un-mute the crystal bank. A front panel Spot switch is included so I can find my signal in the receiver for zero beating.

RG 174 was used for all of the wiring from the crystal bank's output ports (the RCA jacks were removed for the final construction) through the relays to the buffer amplifier and from the output of the buffer amplifier to a BNC connector on the back of the metal enclosure.

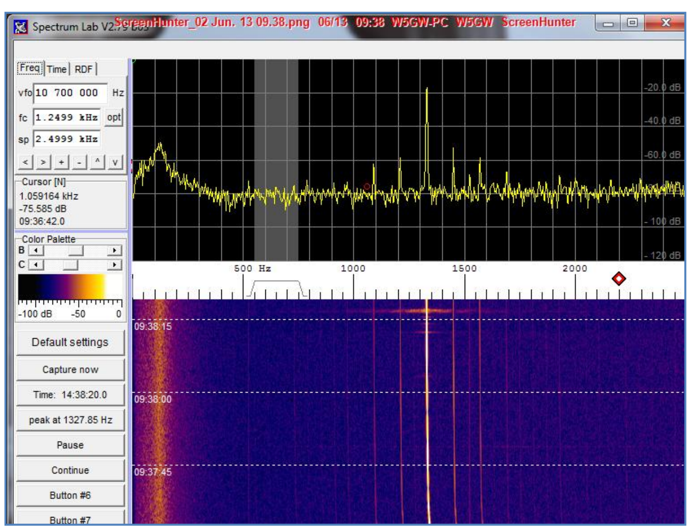


Power Supply and Frequency Stability and Accuracy: I decided rather than include a voltage regulator and operate from 12 vdc, to just supply the crystal bank with 5 vdc from an external source. My primary reason for this was to keep the heat dissipation minimal so as to improve the frequency stability of the 125 MHz crystal oscillator on each AD9850 DDS board.

Doug, N3ZI, recommends the use of an external wall wart power supply. I tried this and the switching hash caused frequency modulation (see below picture) and resulted in a terrible tone.

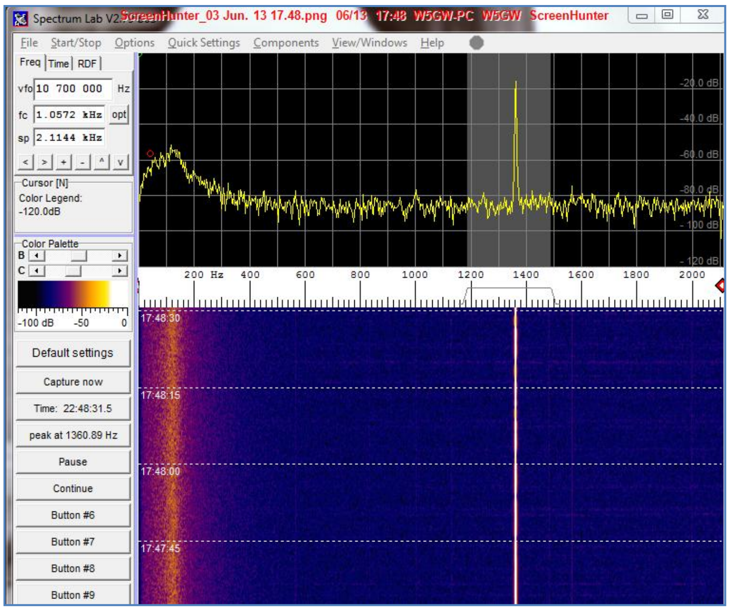


As a temporary test, I applied my lab power supply (also a switcher but better filtered) and obtained the results in this picture. Switcher spurs reduced but not eliminated.



I also discovered changing air currents around the crystal on the AD9850 board can cause several tens of Hz of frequency shift. The wiggly lines on the first photo are the result of waving my fingers close to or touching the crystal. By the way, it is normal for this crystal to run somewhat warm. Some folks have recommended using super glue to add a heat sink to the crystal (a copper penny?). I have found when the unit is buttoned up inside of the enclosure, ambient air currents are minimized and the unit will obtain thermal equilibrium fairly rapidly.

I decided to build a small separate enclosure so as to convert 12 vdc down to 5 vdc using a simple 7805 voltage regulator circuit. The result is shown in the following picture. All of the spurs are eliminated and no ac hum was discernible. (*Note: The noisy signal below 200 Hz is an artifact of the soundcard chip in my US Navigator Interface and is not caused by the AD9850.)*

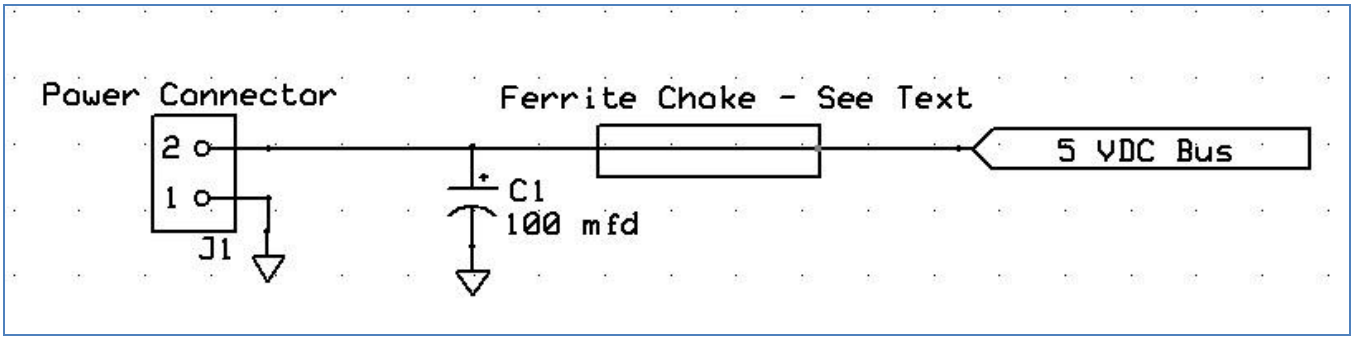


Three other precautions are:

- Addition of a filter capacitor inside the enclosure,
- Ferrite choke inside the enclosure (optional), and
- Shielded wire for the power line.

The filter capacitor is to prevent voltage spikes from damaging the DDS mother board or daughter cards on power application and provide RF filtering.

I had some ferrite chokes using type 43 material – this is a bit of a dual purpose component: 1. Prevent spurious (harmonic) RF from inside the case to get on the power lead back to the switching power supply and 2. Prevent RF from the shack from entering the innards of the crystal bank. Finally, be sure to use shielded wire for the power line for the same reasons. The circuit is very simple, but be sure to mount the filter capacitor and ferrite choke at the power jack on the inside of the metal enclosure.



The frequency stability of the DDS cards is quite good considering the modest 125 MHz crystal for the reference clock. After a 30 minute warm up period the DDS cards only drifted a few Hz at 10 MHz when running on my work bench in an ambient air temperature of about 76 degrees. Once warmed up, short term drift in ambient air is less than 1.0 Hz per minute. Over a 10 minute period, sampling every 10 seconds, I measured a maximum frequency excursion range of +/- 0.9 Hz when operating at 10 MHz.

Measured short term drift after a 30 minute warm up period is better defined using the statistical measure of standard deviation. Measuring for 10 minutes (again sampled every 10 seconds) of a radiated AD9850 signal at 10 MHz with an ICOM 756 PRO 3 using Spectrum Lab audio analysis software, the received standard deviation is 0.55 Hz. Inspection of these random errors indicate they are fairly symmetrical about the mean; hence, they appear to be a normal distribution. I'd expect to see a short term drift of less than +/- 1.65 Hz within 10 minutes once the unit has established thermal equilibrium. This type of frequency stability is adequate for high accuracy modulations such as QRSS.

Absolute frequency accuracy is a different matter. The AD9850 frequency is controlled by a common piezoelectric crystal assumed to provide an exact frequency of 125 MHz. In reality, of the five different AD9850 boards I had, all displayed some differences so that, at 10 MHz I had absolute frequency accuracy errors of several hundred Hz. Doug, N3ZI, includes a calibration method so that the AD9850 DDS board for each channel can be calibrated in software. I won't repeat his procedure here, but it was sufficient for me to obtain an absolute frequency accuracy of about +/- 3 Hz at 10 MHz. Remember this error will increase with frequency, so use the highest frequency you can when performing the calibration routine.

Therefore, I'd state the specifications of this unit as:

- Frequency stability less than +/- 50 Hz during a 30 minute warm up
- Short term drift 99.7% less than +/- 1.65 Hz in a 10 minute period after warm up
- Frequency accuracy within +/- 10 Hz at 30 MHz

All of the above assumes operating in a stable environment where the temperature will not vary by more than a few degrees and where the unit is not subject to air currents (in an enclosure) that will cause convective heating and/or cooling of the 125 MHz crystal oscillator. No provision was made to provide any thermal oven; however, this would likely be the next step if any better frequency stability were needed.

Other Applications and uProcessors: The crystal bank was a very successful project and is now being used with my vintage Viking II transmitter.

I had purchased a few extra DDS boards and began experimenting controlling them with an Arduino and Texas Instrument Launch Pad. There are several projects on the web using these boards along with some code examples. One of those code examples came from this web site http://webshed.org/wiki/AD9850_Arduino. I have reproduced this code below:

```
//AD9850 DDS test
#define DDS_CLOCK 125000000
#define  CLOCK  8  //pin connections for DDS – to clock on DDS board
#define  LOAD 9 // to FU_UD or FQUP (on alternative module type)
#define  DATA  10 // to Data or D7 (on alternative module type)
#define  RESET 11 // to Reset on DDS board
void setup()
{
  pinMode (DATA,  OUTPUT);
  pinMode (CLOCK, OUTPUT);
  pinMode (LOAD,  OUTPUT);
  pinMode (RESET, OUTPUT);
  AD9850_init();
  AD9850_reset();
  SetFrequency(10000000);
}
void loop()
{
}
void SetFrequency(unsigned long frequency)
{
  unsigned long tuning_word = (frequency * pow(2, 32)) / DDS_CLOCK;
  digitalWrite (LOAD, LOW);
  shiftOut(DATA, CLOCK, LSBFIRST, tuning_word);
  shiftOut(DATA, CLOCK, LSBFIRST, tuning_word >> 8);
  shiftOut(DATA, CLOCK, LSBFIRST, tuning_word >> 16);
  shiftOut(DATA, CLOCK, LSBFIRST, tuning_word >> 24);
```

```
  shiftOut(DATA, CLOCK, LSBFIRST, 0x0);
  digitalWrite (LOAD, HIGH);
}
void AD9850_init()
{
  digitalWrite(RESET, LOW);
  digitalWrite(CLOCK, LOW);
  digitalWrite(LOAD, LOW);
  digitalWrite(DATA, LOW);
}
void AD9850_reset()
{
  //reset sequence is:
  // CLOCK & LOAD = LOW
  //  Pulse RESET high for a few uS (use 5 uS here)
  //  Pulse CLOCK high for a few uS (use 5 uS here)
  //  Set DATA to ZERO and pulse LOAD for a few uS (use 5 uS here)
  // data sheet diagrams show only RESET and CLOCK being used to reset the device, but I see no output unless
I also
  // toggle the LOAD line here.
  digitalWrite(CLOCK, LOW);
  digitalWrite(LOAD, LOW);
  digitalWrite(RESET, LOW);
  delay(5);
  digitalWrite(RESET, HIGH);  //pulse RESET
  delay(5);
  digitalWrite(RESET, LOW);
  delay(5);
  digitalWrite(CLOCK, LOW);
  delay(5);
  digitalWrite(CLOCK, HIGH);  //pulse CLOCK
  delay(5);
  digitalWrite(CLOCK, LOW);
  delay(5);
```

```
  digitalWrite(DATA, LOW);    //make sure DATA pin is LOW
    digitalWrite(LOAD, LOW);
  delay(5);
  digitalWrite(LOAD, HIGH);  //pulse LOAD
  delay(5);
  digitalWrite(LOAD, LOW);
  // Chip is RESET now
}
```

Note there are only four output lines (DATA, CLOCK, LOAD, and RESET) necessary to drive the DDS-9850 board. The LOAD is marked as FU_UD on the board pictured at the beginning of this paper. As of this writing, there is another similar AD9850 board being sold on eBay. It has the LOAD on the pin marked FQUP. Also note, this board uses D7 for a data serial line. The CLOCK and RESET connections should be obvious.

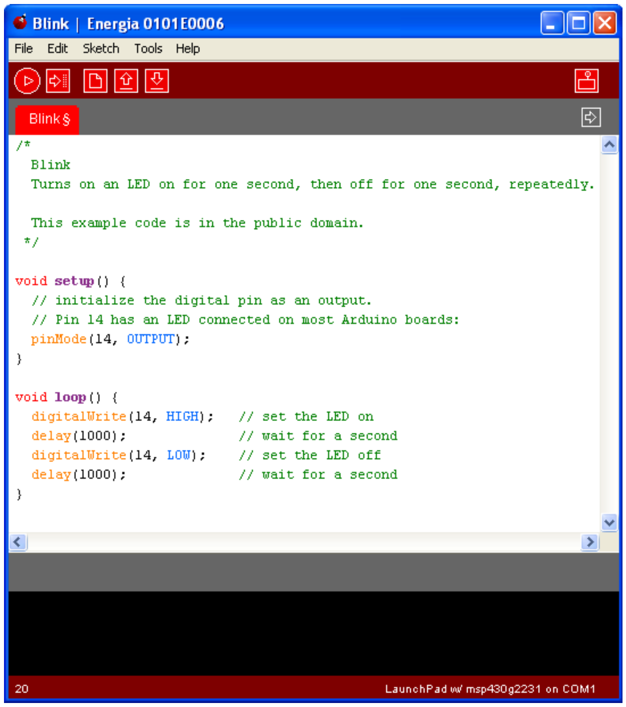
You should be able to easily upload this code as a sketch to an Arduino UNO using either of these two DDS modules and hear a tone very close to 10 MHz.

A Texas Instruments LaunchPad using a MSP430G2553 was also available and used with this example by making the following changes to the code – see snippet below:

```
//Replace these to use with a TI LaunchPad
#define  CLOCK  P1_2  //pin connections for DDS – to clock on DDS board
#define  LOAD P2_1 // to FU_UD or FQUP (on alternative module type)
#define  DATA  P2_2 // to Data or D7 (on alternative module type)
#define  RESET P2_4 // to Reset on DDS board
```

An Energia IDE (http://energia.nu/) was used to compile and make the LaunchPad behave as an Arduino. Note the Energia IDE has the same look and feel as an Arduino IDE (see below) except for the red coloring instead of green.

Pin mapping on a LaunchPad can have several conventions. For example, pin 12 is the same as P2_4. This is explained in the Energia documentation and there are several handy diagrams depending on which uProcessor your LaunchPad is equipped with.



Where to go from here? I'm looking at QRSS using the AD9850 and a LaunchPad as my frequency source. Other possibilities are:

- RF source for an automatic antenna tuner
- VFO
- Receiver calibrator

Here is a picture of the final crystal bank taken when I was determining current draw. The three push buttons on the top are for calibration and frequency adjustment. The large red knob selects either OFF (center), Channel A ON (left) or Channel B ON (right). The small pushbutton below the channel/ON/OFF switch is the SPOT switch.

The two variants of DDS daughterboard are shown in the foreground. The one on the left is the one that will plug into the N3ZI motherboard. The one on the right appears to work just as good, costs about the same on eBay, but will require a different method of mounting if the N3ZI control board is used.