AAØZZ IQPro

A High-Performance, Dual-Output VFO using the AD9854 DDS

By Craig Johnson, AAØZZ

aa0zz@cbjohns.com

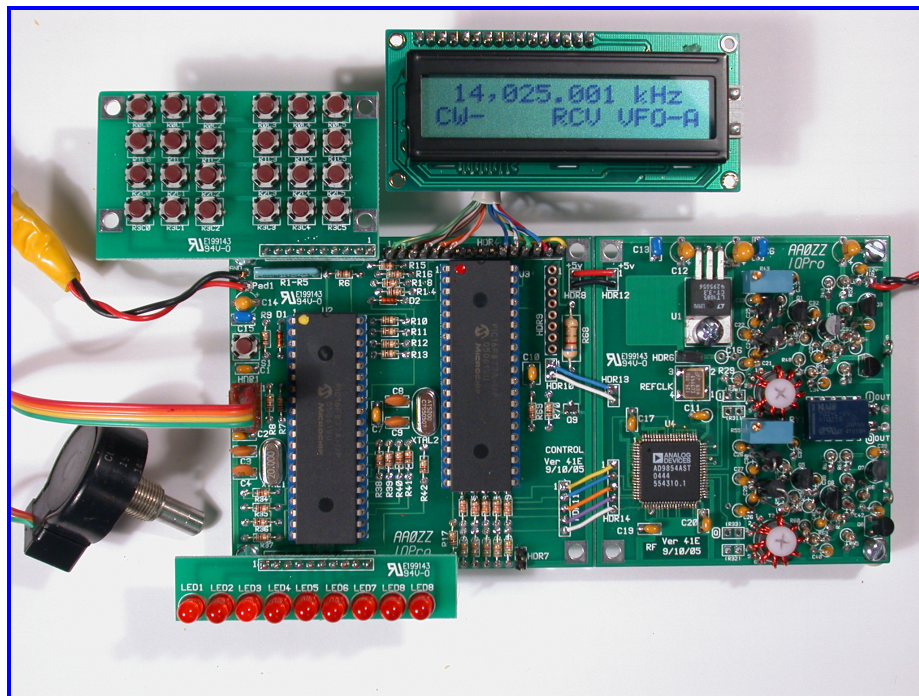www.cbjohns.com/aa0zz

TABLE OF CONTENTS

1   Introduction

In the July, 1997 issue of QEX magazine, Curtis Preuss, WB2V, published an article featuring a signal generator which used an Analog Devices AD9850 DDS device controlled by a PIC16F84 microcontroller. Since that time, a newer DDS device has been introduced by Analog Devices - the AD9854. This device has some significant advantages over the AD9850.

First, the AD9854 produces two outputs with a 90-degree phase difference. This phase relationship remains constant regardless of the frequency.

Second, the AD9854 has a 12 Bit DAC (D/A converter) while the AD9850 has a 10 bit DAC. This is important in reducing the undesirable byproducts (spurs).

Third, the version of the AD9854 used in this project has a 200 MHz maximum clock speed, compared to 125 MHz for the AD9850. The faster clock speed raises the Nyquist limit, allowing the VFO to operate cleanly up through 30 MHz.

This project is an extension of the concepts introduced in that original AD9850 project, but there have been many major changes. It still has an optical encoder for frequency selection and it still uses an LCD for displaying the frequency. However, the IQPro VFO has been changed in these ways:

- The IQPro VFO now uses two PIC microcontrollers - one controlling the user interface (optical encoder, pushbuttons, and LEDs) and the other controlling the AD9854, the LCD, and the band-switching relay interface.

- The IQPro VFO now uses a two-line LCD so more information is displayed.

- The IQPro VFO now operates as two VFOs with two frequencies. The active VFO is selectable via a pushbutton.

- The two-part VFO supports split-frequency operation. This means you can transmit on one frequency and receive on another.

- The two AD9854 outputs are amplified to produce signal levels which are compatible with common receiver requirements (such as the R2, MiniR2, Binaural or R2Pro by Rick Campbell, KK7B.)

Early versions of this VFO project (WB2V follow-on) used a single PIC microcontroller for all VFO functions. However, the PIC's main code execution loop was very large and it varied a great deal in length, depending on if the LCD and DDS needed to be updated and if pushbuttons were being pressed. Since the single PIC needed to monitor the input signals from the optical encoder, detecting the number of changes in the "gray code" to determine how fast the encoder was being turned, the variable size main execution loop made accurate timing and smooth operation very difficult to accomplish.

In the next version of project, which I developed with Bruce Stough, AAØED, we decided to use a dedicated PIC16F628 microcontroller to handle the optical encoder. Why? They are inexpensive and provide a rather elegant solution to the problem of accurate timing for smooth tuning operation.

It is changed even more in the IQPro. Now there are two 16F877 PICs. One is defined as the Interface PIC and the other is the Driver PIC. The encoder logic has been moved into the Interface PIC. The Interface PIC monitors the "gray code" output of the encoder and periodically sends information to the Driver PIC (by way of an encoder message) to indicate how far the encoder shaft has been turned since the last update as well as the direction of turn. The Interface PIC sends this data to the Driver PIC by setting up the tick-count (up to 6 bits - indicating the number of encoder "transitions" detected since the last update) and the direction bit in a message byte and then sending this encoder message to the Driver PIC. When the Driver PIC gets the encoder message, it extracts the tick-count and direction from the message, updates the working registers with the new frequency, and sets a flag that indicates the DDS and LCD need updating. A very short time later, code in the main execution loop of the Driver PIC will detect the "changed" flag and call the appropriate routines to update the frequency in the DDS and the LCD.

The Interface PIC also handles the pushbuttons. When a pushbutton is pressed, the Interface PIC detects it and sends a message to the Driver PIC, causing the Driver PIC to take the appropriate action. Similarly, whenever the Driver PIC needs to turn an LED on or off, it sends a message to the Interface PIC and the Interface PIC performs that action. In addition, the Driver PIC sets up output signals on a header when the VFO frequency changes to a different band. These signals are intended to set/reset band relays to activate band-specific filters which remove harmonics from a transmitter output before sending the signal to the antenna.

2   Theory of Operation

Early versions of this project used a single 16F84 PIC microcontroller for all VFO functions. However, the PIC's main code execution loop was very large and it varied a great deal in length, depending on if the LCD and DDS needed to be updated and if pushbuttons were being used. In the next version we used a dedicated PIC16F628 microcontroller for the optical encoder. Now, in the IQPro main (control) board, the IQ-VFO's PIC16F628 is replaced by a PIC16F877. This larger PIC also provides additional interface pins. The IQPro divides the functions between two PIC16F877s - the Interface PIC and the Driver PIC.

The IQPro takes another step forward in improving the "user interface". The Interface PIC handles all operator inputs and outputs. In addition to handling the optical encoder, the pushbuttons and the LEDs are controlled by the Interface PIC. The Driver PIC handles the AD9854, the LCDs, and the filter relays on the Filter / Relay Board.

The pushbutton board is a small board on which 24 pushbuttons can be mounted. The pushbutton board interface header plugs into the main (control) board's header HDR3. For testing purposes, the builder can install the pushbuttons directly on the pushbutton board; however, the builder will eventually want to move the pushbuttons to the front panel of the project's chassis. It is still convenient to use the pushbutton board since it provides an easy way of connecting pushbuttons to the four rows and six columns. Even so, there are several options. The pushbutton board can remain plugged into the main (control) board header with individual wire-pairs to bring the pushbuttons to the front panel or the pushbutton board can be mounted directly behind the front panel, connected to header HDR3 by way of a 10-wire harness.

The LED Board has positions for the nine LEDs to be mounted on it. For testing purposes it can be plugged directly into HDR2 of the main (control) board. Then it can be moved to the chassis front panel or individual wires (one wire per LED, plus a common ground) can be moved to the front panel.

The band-switch test LED Board has six LEDs and is for testing purposes only. It plugs into main (control) board header HDR5. This header provides a way attach an 8-wire harness to bring the six relay signals (plus the ground and the optional +5v power line) to an external filter / relay board (not supplied).

The I and Q signals generated by the AD9854 DDS are passed through low pass filters and amplified before going off-board to the receiver or transmitter. The I / Q reverse relay, activated by pressing a pushbutton, is used to switch the I and Q signals before going off-board. This allows the transmitter or receiver to toggle between upper and lower sideband. For additional details see the I-Q Reversal section.

Figures 1 shows a snapshot of the I and Q signals that are generated by the AD9854. Note that this 90-degree (quadrature) phase relationship is consistent for all generated frequencies, from 0 to 30 MHz.
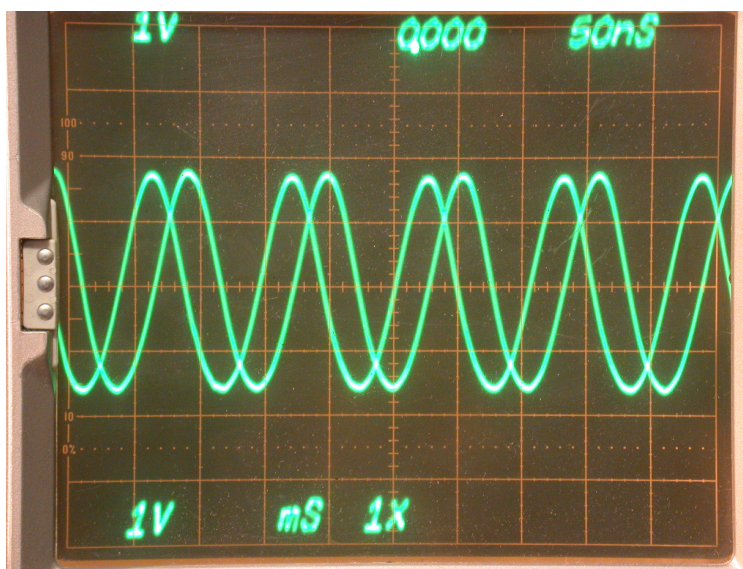

Figure 1 – I and Q Signals
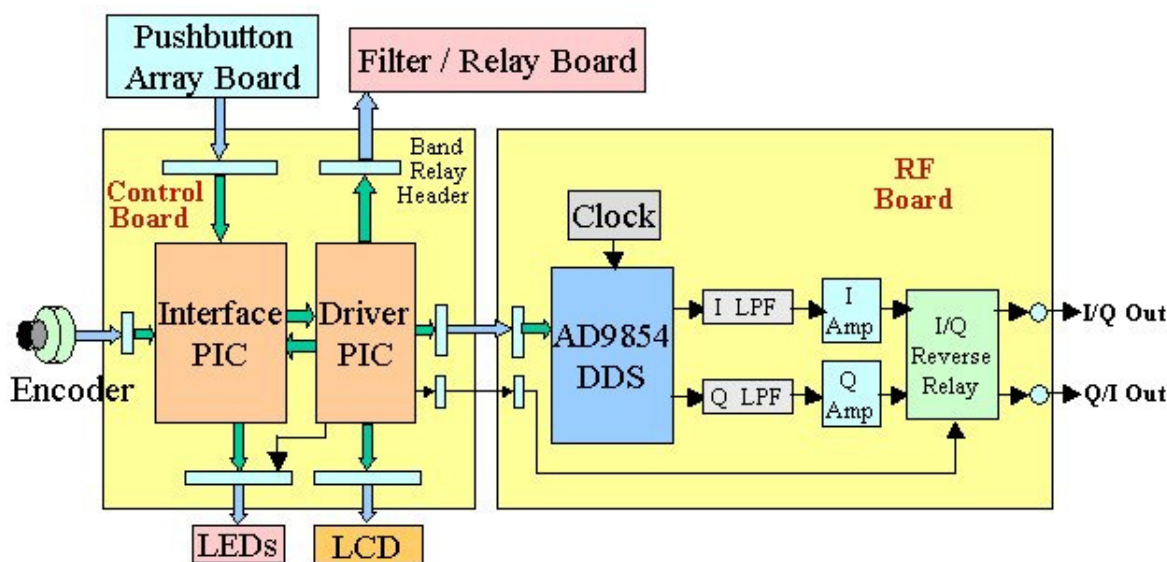
Figure 2 shows a block diagram of the IQPro.

Figure 2 – IQPro Block Diagram

3 Main Board Description

3.1 Control and RF Board Description

Figures 23 and 24 shows the two-part schematic for the IQPro. The IQPro is divided into two boards - a control board and an RF board. The reason for the division into two boards is to allow the builder the option of completely shielding the RF board if desired.

If they are separated, it is suggested that the RF board be mounted in an RF-shielded box (e.g. Hammond 1590BB) and feed-through capacitors used to bring the power and control lines from the control board into the RF board in the shielded box.

If the control and RF boards are not separated, jumpers need to be installed between the matching pins of headers HDR7 and HDR10, HDR8 and HDR11, and HDR9 and HDR12.

3.1.1 Control Board

The two socketed PIC microcontrollers control the IQPro functions. Both are type 16F877 and are 40-pin parts. Each PIC pin that is used as an input has a 10k ohm pull-up resistor attached to +5v.

The optical encoder feeds its two "gray code" signals to the Interface PIC.

3.1.2 RF Board

The RF Board uses a 125 MHz DIP clock oscillator for the AD9854. This is a tradeoff. We could have used a lower frequency clock oscillator and then used the AD9854's internal clock multiplier but that introduces additional spurious products as well.

The output current (Iout) from the AD9854 is set by the size of the Rset resistor. We are using an Rset of 3.9k which translates to 10ma of output current. (20 ma is maximum, but 10 ma is selected because it provides the best Spurious Free Dynamic Range- SDFR.)

3.2 Control and RF Board Details

The two boards together measure 3" x 6". There is a "partial-cut" line between the two boards so the builder can chose to break them apart or to keep them together. If they are kept together as a single board, simple jumpers can be installed between the appropriate pins in HDR7 and HDR10, HDR8 and HDR11, and HDR9 and HDR12.

If you chose to mount the RF board in an RF shielded box, it fits nicely into a Hammond aluminum diecast box (Hammond 1590BB, DigiKey part number HM152-ND) which is 3.7" x 4.7" x 1.2".

External power to the control board (passed to the RF board) is 5 volts. If the transistor amplifier option is selected, +12v is also needed. There is no 12v to 5v converter on the IQPro boards. However, a small 12v to 5v converter board is supplied with the IQPro kit. It can be mounted at any convenient location in the case.

The AD9854 must run at 3.3 volts so its reference clock (125 MHz) is also a 3.3v part. However, since the PIC16F877 runs at 5 volts, a method of converting the voltages of the signal lines that connect these PIC16F877 to the AD9854 was needed. It is done with simple voltage dividing resistors on each signal line.

Five-volt power consumption for the entire board is about 325 ma with the LCD backlight off and about 400 ma with the backlight on. About 265 ma of this current goes to the AD9854 and the 125 MHz clock. The 12-volt transistor amplifiers draw a total of 40 ma if this options is implemented.

3.3   I-Q Reversal

The AD9854 puts out two output signals, I and Q, with a 90 degree phase difference. The I and Q outputs are reversed by the activation of the IQ Reverse Relay.   The signals to set and reset this latching relay are generated by the Driver PIC when the Mode Cycle pushbutton is pressed.

3.4   Band Relay Driver

The IQPro is set up to cover the frequency range of 0 to 30 MHz.   As seen in the block diagrams, the IQPro is intended to drive a receiver directly with the I and Q signals. However, transmitters need low pass filters at the outputs, just before going to the antenna, to remove harmonics.   Obviously a single low pass filter (such as 30 MHz) cannot be used to eliminate the harmonics for the entire range. The IQPro has a scheme in which signals are generated (HDR5) which can directly trigger latching relays on another board (the filter/relay board) which can engage the proper low pass filter.

Within the IQPro frequency range there are currently nine U.S. amateur bands (ignoring the 60 meter band). While nine individual filters could be used, that is not really necessary. The IQPro has a scheme in which five low pass filters are sufficient to remove the second (and higher) harmonics for all nine amateur bands. Builders can easily change these ranges.

    Relay 1   enables LPF with 3.5 MHz cut-off   (For 160 Meters - 2nd harmonic removed)
    Relay 2   enables LPF with 7.0 MHz cut-off   (For 80 Meters - 2nd harmonic removed)
    Relay 3   enables LPF with 14 MHz cut-off   (For 40 and 30 Meters - 2nd harmonic removed)
    Relay 4   enables LPF with 18.1 MHz cut-off (For 20 Meters - 2nd harmonic removed)
    Relay 5   enables LPF with 30 MHz cut-off.   (For 17, 15, 12 and 10 Meters - 2nd harmonic removed)
    Relay 6   (Unused.   Available for future use.)

Since the IQPro is continuous coverage from 0 to 30 MHz, the following eleven band numbers and ranges are implemented.

    Band Number 1   Relay 1   0 - 1.799999 MHz
    Band Number 2   Relay 1   1.8 - 3.499999 MHz
    Band Number 3   Relay 2   3.5 - 6.999999 MHz
    Band Number 4   Relay 3   7.0 - 10.099999 MHz
    Band Number 5   Relay 3   10.1 - 13.999999 MHz
    Band Number 6   Relay 4   14.0 - 18.067999 MHz
    Band Number 7   Relay 5   18.068 - 20.999999 MHz
    Band Number 8   Relay 5   21.0 - 24.889999 MHz
    Band Number 9   Relay 5   24.89 - 27.999999 MHz
    Band Number 10   Relay 5   28.0 - 29.999999 MHz
    Band Number 11   Relay 5   30.0 - 30 MHz

The five band relays are activated by signals on header HDR5.   Each relay on the filter board is assumed to have one side connected to a unique header pin (coming from the Driver PIC on the main control board) and the other side of each relay connected to a common connection point.   Thus, to set and latch a relay, the

Driver PIC simply puts a low level on all pins on the header and then raises the appropriate header pin to set that relay. The pin is kept high for 8 ms and then brought back to a low level. The selected relay is now latched in a set state. Similarly, to reset a relay, the Driver PIC puts a high level on all header pins (causing no current to flow) and then makes the appropriate relay's header pin a low. Current will then flow though through the selected relay's coil in the reverse direction and the relay will go to the reset state. After 8 ms, all pins are brought back to a low level again. The relay is now latched in a reset state.

This method of setting/resetting the relays is used because it minimizes the number of PIC pins (and header pins) required. Note that during a reset operation, forward current is flowing through the coils of all of the non-selected relays. However, the forward current through the non-selected relay coils is not enough to inadvertently set them.

3.5 Band-switch Test Board

A band-switch test board is included with the IQPro kit. The schematic is shown in Figure 3. This board is designed to plug into HDR5 of the main (control) board to allow the builder see the relay signals pulse as the frequency ranges change. Surface mount (size 0805) 1K resistors are used on this test board.
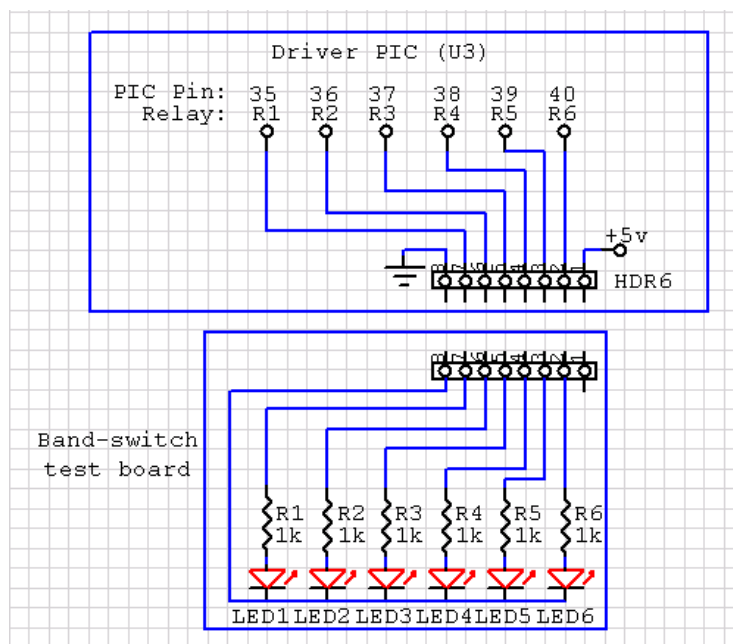
Figure 3 - Band-switch Test Board Schematic

The Relay/Filter Board section for more information about implementing a relay/filter board.

3.6 I and Q Amplifiers

The I and Q signals generated by the AD9854 are not sufficient in amplitude to drive most mixers. As mentioned in the RF board section, the output current (Iout) from the AD9854 is set by the size of the Rset resistor. The default value for Rset of 3.9k, which translates to 10ma of output current. 20 ma is maximum, but 10 ma provides the best Spurious Free Dynamic Range (SDFR).

The output current of 10 ma into a 50 ohm load translates into 500 mv p-p (177 mv RMS) which is about -2 dBm. However, many common mixers (such as the TUF-3 used by the R2Pro) require +7 dBm signal levels so some amplification is required.

There are three amplifier options available on the IQPro board:

3.6.1 Option 1 – Toroids only (no transistor amplifiers)

In this option, two FT37-43 toroids with trifilar windings are used. The toroid connections go directly to the AD98954.

The transistor amplifiers parts are not installed. The AD9854's two complementary outputs are "added" to the primary outputs by way of the toroid transformers.

If +7 dBm signal outputs are required, the nominal value of 10 ma output on the 4 signals is not sufficient. To increase the I and Q signals to 14 ma, change the Rset resistor (R29) from 3.9k to 2.7k.

Figure 4 shows the basic schematic for the trifilar transformer. (This is duplicated for the Q signal.)
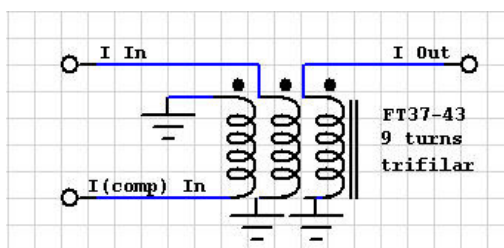

Figure 4 - Toroid Transformer

Pros:
1. The two complementary outputs from the AD9854 as well as the two "normal" outputs are used.
2. No 12v power is needed on the IQPro board.

Cons:
1. Two toroids with trifilar windings must be constructed and connected properly.
2. If +7 dBm outputs are wanted, the Rset resistor (R29) may have to be changed from 3.9k to a smaller value such that more current is put out by the two AD9854 outputs. This might cause a slightly higher level of spurious output products.

To use this option:
1. Termination resistors R30 – R33 are NOT installed on the board.
2. Install a wire jumper from the output of each Low Pass Filter to the relay (RY1) inputs.
3. Transistor amplifier components are NOT installed.

3.6.2 Option 2 – Transistor amplifiers only
If this option is selected, the transistor amplifier components are used and the two toroids are not installed. This gives +7 dBm signal outputs.

Pros:
1. The AD9854 should be putting out the cleanest signal (highest Spurious Free Dynamic Range) with the nominal Rset value of 3.9k (corresponding to 10 ma output on I and Q).

Cons:
1. +12v power is required.

To use this option:
1. Termination resistors R30 - R33 are installed.
2. Rset resistor (R29) is 3.9k.
3. Toroids T1 and T2 are NOT installed.
4. Transistor amplifier components are installed.
5. Install a wire jumper from Pin 2 to Pin 3 of T1 and from Pin 2 to Pin 3 of T2.

3.6.3 Option 3 – Toroids and Transistor amplifiers

Use this option if signal outputs larger than +7 dBm are needed. With Rset (R29) at its "nominal" value of 3.9k, producing 10 ma signals, the typical outputs of +17 dBm are produced.

Pros:
1. This option produces the highest output signal levels of the three options.

Cons:
1. +12v power is required.

To use this option:
1. Termination resistors R30 - R33 are NOT installed.
2. Rset resistor (R29) is 3.9k.
3. Toroids T1 and T2 are installed.
6. Transistor amplifier components are installed.

3.7 PIC Software

3.7.1 AD9854 Initialization
The IQPro runs the AD9854 in serial mode. This means it uses one data line, a clock line, and an external update line to send the frequency updates from the PIC to the AD9854. Two reset lines are also controlled by the PIC.

Determining how to get the AD9854 initialized was a real challenge, especially with a 125 MHz reference clock. The problem stems from the fact that the AD9854 powers up in internal update mode while it really needs to be in external update mode to operate in this type of application.   See the AD9854 data sheet for details.

3.7.1.1 Debug Mode for AD9854 Initialization

The AD9854 initialization sequence is sent from the driver PIC to the AD9854 immediately after the IQPro is powered up and after the reset pushbutton is pressed. However, the AD9854 initialization signals are fairly short in duration and may be difficult to observe with an oscilloscope. To aid in debugging, a special debug mechanism was implemented in the driver PIC code. This code, when triggered, forces the driver PIC to go into an endless loop in which the initialization sequence is continually repeated. This debug mode is activated by installing a jumper between the two pins of header HDR6 during power up. The initialization sequence will be continuously sent until the short between the two header pins is removed. Now the signals can easily be seen with an oscilloscope. When the jumper is removed the PIC will continue initializing the IQPro.

3.7.1 Inter-PIC communications

A completely home-grown synchronous communications scheme is implemented in the IQPro to pass messages between the PICs. Yes, the PIC16F877 supports two industry-standard synchronous communications schemes (SPI and I^2C) as well as an asynchronous scheme (USART/SCI), but I chose not to use them. Why? Because these communications schemes are really designed for efficiency in long data transfers but are rather inefficient for the single-byte messages of the IQPro.

The IQPro synchronous message-passing scheme uses the following general principles:

1. Use SYNC sequence: 111111111 (9 one-bits).
   Note that this sequence of bits is impossible for a "normal" data byte plus parity.

2. Send a data byte consisting of 8 data bits followed by a parity bit
   - Use EVEN parity
   - The first valid data code would be 00000000+p (which is 00000000 0)
   - The last valid data code allowed would be 11111111+p (which is 11111111 0)
   - Gives 256 usable codes. All 8 data bits are usable.

This communications scheme uses 3 data lines to send a message from one PIC to another. Two of these lines are controlled by the "originating" PIC (Data and Clock) and one line is controlled by the "receiving" PIC (Acknowledge). Two sets of lines are used to implement the two-way message passing.

Assume a message is to be passed from PIC-A to PIC-B. The timing diagram of Figure 5 shows the signals as one bit is transferred from PIC-A to PIC-B.
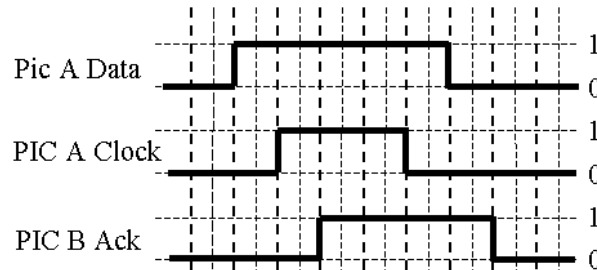
Figure 5 – PIC Communication Signals

To transfer a single BIT from PIC-A to PIC-B the following actions are taken:

1) PIC-A detects B-ack is low. (If not low, wait until it is.)
2) PIC-A sets A-data to 0 or 1
3) PIC-A sets A-clk high
4) PIC-B detects A-clk high and extracts data from data line
5) PIC-B raises B-ack.
6) PIC-A detects B-ack high (within X cycles) and drops A-clk (and A-data)
7) PIC-B detects A-clk is low and drops B-ack.
8) (Go back to step 1 for next bit)

Note that this scheme is completely synchronous so it is very fast. There are no "baud rates" to set up.

How well does it work? As implemented in the IQPro hardware and software, with the PICs running at 20 MHz, a message can be passed in 165 microseconds (measured).

Yes, there are some dangers in this scheme. The primary danger is if a collision occurs - with both sides originating messages simultaneously. Since the PIC code is not multi-threaded (it doesn't release control and switch between activities), if both PICs start executing the "SEND" code simultaneously, neither side will be able to receive a message so the message-passing will fail. This will result in a timeout. The Driver PIC is designed to time out first. (The loop constants in the Driver receive_bit and send_bit code) are smaller for the Driver PIC so the Interface PIC will detect the condition and will reset the Driver PIC. Obviously this is annoying so the occurrences must be extremely rare to make this acceptable.

How are collisions avoided? By being careful about when messages are originated in the Driver PIC.

3.7.2 Message Definitions

Figure 6 shows the messages that are sent from the Interface PIC to the Driver PIC.

Interface to Driver Messages (Hex)	Description
0	Keypad 1
1	Keypad 2
2	Keypad 3
3	MHz UP
4	Split Toggle
5	VFOA/VFOB Toggle
06 – 0F	(unused)
10	Keypad 4
11	Keypad 5
12	Keypad 6
13	MHz DN
14	Mode Cycle
15	VFOA = VFOB
16 – 1F	(unused)
20	Keypad 7
21	Keypad 8
22	Keypad 9
23	Band Up
24	Tone Set Toggle
25	Fast Tune Toggle
26 – 2F	(unused)
30	Keypad *
31	Keypad 0
32	Keypad #
33	Band Down
34	Backlight Toggle
35	Calibrate
36 – 3F	(unused)

Figure 6 - Interface PIC To Driver PIC Messages

Figure 7 shows the messages that are sent from the Driver PIC to the Interface PIC.

Driver to Interface Messages (Hex)	Description
40	LED 1 ON (SSB) and LED 2 OFF
41	LED 2 ON (CW) and LED 1 OFF
42	LED 3 ON (VFOA) and LED 4 OFF
43	LED 4 ON (VFOB) and LED 3 OFF
44	LED 5 ON (SPLIT)
45	LED 6 ON (FTUNE)
46	LED 7 ON (CAL)
47-53	(unused)
54	LED 5 OFF (SPLIT)
55	LED 6 FF (FTUNE)
56	LED 7 OFF (CAL)
57 – 5F	(unused)
80 - 8F	ENCODER messages

Figure 7 - Driver PIC To Interface PIC Messages

4   IQPro Operation

4.1   Optical Encoder

A good quality optical encoder is recommended for the IQPro, although a mechanical encoder will also work. (Note that capacitors C1 and C2 are installed to eliminate the "noise" generated by the bounce of the switches in a mechanical encoder.)

The recommended optical encoder has 128 positions per revolution. At design time we decided we want the PIC to be able to capture all transition pulses when we turn the tuning knob (optical encoder) at a rate of 2 revolutions per second. (When we turn the knob faster than 2 revolutions per second we allow some transitions to be missed.)   The IQPro was designed to send encoder messages from the Interface PIC to the Driver PIC every 25 ms. At this rate, the calculations for the maximum number of transitions per update is given in Figure 8.

```
512 transitions/rev * 2 rev/second = 1024 transitions/sec

1024 transitions/sec = 0.98 ms / transition

Then:        25 ms/update
       _____    = 25.5 transitions/update
         0.98 ms/transition
```

Figure 8 – Encoder Transitions Per Second Calculation

We can send this number over to the Driver PIC with 5 "binary" bits. Since we have 6 bits (plus a direction bit) available we can exceed our design requirement and use the transitions when the tuning knob is turned even faster than 2 revolutions per second.

The IQPro has a communications scheme in which 8-bit messages are passed. (See the Inter-PIC Communications section for details regarding the inter-PIC communications scheme.) Since encoder messages are the most common message passed and efficiency is essential, a special scheme was implemented in which bit 7 of the message byte indicates that it is an encoder message. Then, bit 6 of any encoder message indicates the direction (1 = clockwise). This leaves 6 message bits (maximum value of 63) to pass the number of transitions that were detected in the past 25 ms period. A slight variation was implemented in which 0 indicates one transition and no message gets sent if no transitions are detected within the 25 ms period. Thus, 64 possible values can be sent in a message.

One additional factor that needs to be discussed is the PIC processing overhead that occurs on every update. The PIC processor must stop looking for encoder transitions when it is busy sending a message to the Driver PIC. Any transition that occurs while the PIC is busy sending a message will be lost. Fortunately message passing is very fast (about 165 us) so the length of time that the PIC is taken away from looking for encoder transitions is very small (165 us out of 25 ms).

The two encoder data pins are connected to the Interface PIC. Outputs A and B are the two encoder output signals and give "gray code" outputs (90 degrees out of phase - quadrature).

Figure 9 shows how "gray code" works while Figure 10 shows the algorithm that the PIC uses to determine the direction the encoder is rotating.
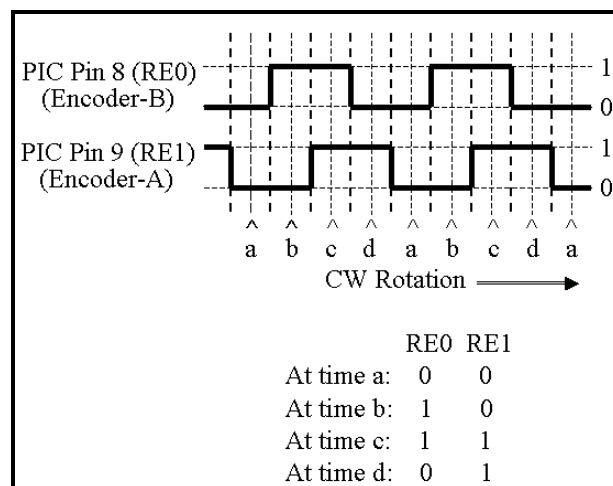

Figure 9 – Encoder Quadrature

Going UP, the sequence is a,b,c,d,a,b,c,d,a etc. so the sequence is:
    00, 10, 11, 01, 00, 10, 11, 01, 00 etc.

Going DOWN, the sequence is a,d,c,b,a,d,c,b,a, etc. so the sequence is:
    00, 01, 11, 10, 00, 01, 11, 10, 00, etc.

To determine if the sequence is UP or DOWN:
    1) Take the "Right-Bit" of any pair.
    2) XOR it with the "Left-Bit" of the next pair in the sequence.
    3) If the result is 1 it is UP; If the result is 0 it is DOWN

Figure 10 – Method for Determining Encoder Direction

The encoder message is sent from the Interface PIC to the Driver PIC. If no acceleration algorithm was used and the transition count was used to increment or decrement the frequency, the maximum frequency change is about 64 Hz per 25 ms update cycle, or 2.56 kHz per second. (This is because an encoder message can pass 6 bits of encoder change magnitude.) This is too slow for comfortable operation so some acceleration is necessary. In fact, two tuning rates were implemented (normal-tuning and fast-tuning). At the same time it is sometimes important to be able to increment/decrement in 1 Hz steps. To spread out the tuning rate, the mechanism shown in Table 1 is used:

Tick Count Range	Maximum Rev Per Sec	Normal Tune Mult. Factor	Normal Tune Max Hz Per Update	Normal Tune Max Hz Per Sec.	Fast Tune Mult. Factor	Fast Tune Max Hz Per Update	Fast Tune Max Hz Per Sec.
0	0.1	1	1	200	1	1	200
1	0.2	2	2	400	8	8	1600
2 - 3	0.3	4	12	2,400	16	48	9,600
4 - 7	0.6	8	56	11,200	32	224	44,800
8-15	1.3	8	120	24,000	32	480	96,000
16 - 31	2.5	16	496	99,200	64	1,984	396,800
32 - 63	5.0	16	1,008	201,600	64	4,032	806,400

Table 1 – Tuning Rate Acceleration

In Table 1 the acceleration constants of 1,2,3,3,4,4 are used for normal tune. For fast tune each of the acceleration constants is increased by 2. These acceleration constants are easily modified in the code and can be changed to suit the individual's preferences.

The Maximum Revolutions Per Second column shows the how fast the encoder can be turned while keeping within that tick count range. Again, this assumes a 128 pulse/revolution (512 transition) optical encoder and an update interval of 25 ms.

The update frequency was chosen carefully such that updates are not done too often. In fact, the 25 ms update interval could have been reduced to 5 ms or less. However, shorter intervals between updates produce encoder magnitude values that are always small so acceleration is rarely invoked. Again, it means that the maximum tuning rate is less than what is usually preferred.

4.2 Pushbutton Array Board

The IQPro has 24 pushbuttons to control the VFO. Twelve of them form a "telephone-type" keypad (which can be used for direct frequency entry) while the remaining twelve pushbuttons control other VFO functions.. Figure 11 shows the format of all 24 pushbuttons as they are arranged on the pushbutton array board.
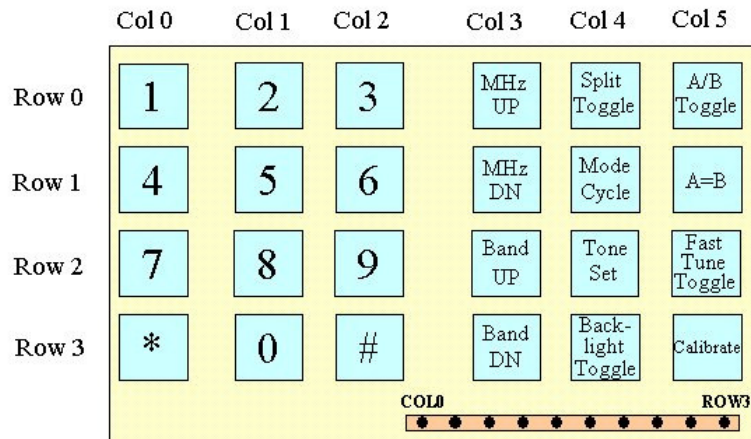

Figure 11 - Pushbutton Board Layout

The pushbuttons are connected to the PIC on the main (control) board by way of a 10-pin header (HDR3) on the main (control) board. The 10 pins represent four rows of six columns of pushbuttons. The pushbutton array board has connection positions for 24 SPST pushbuttons, arranged on a grid of four row busses and six column busses. One side of each pushbutton connects to a row bus and the other side of each pushbutton connects to a column bus. Figure 12 shows a schematic of the pushbutton array board and its connections to the main (control) board via header HDR3.
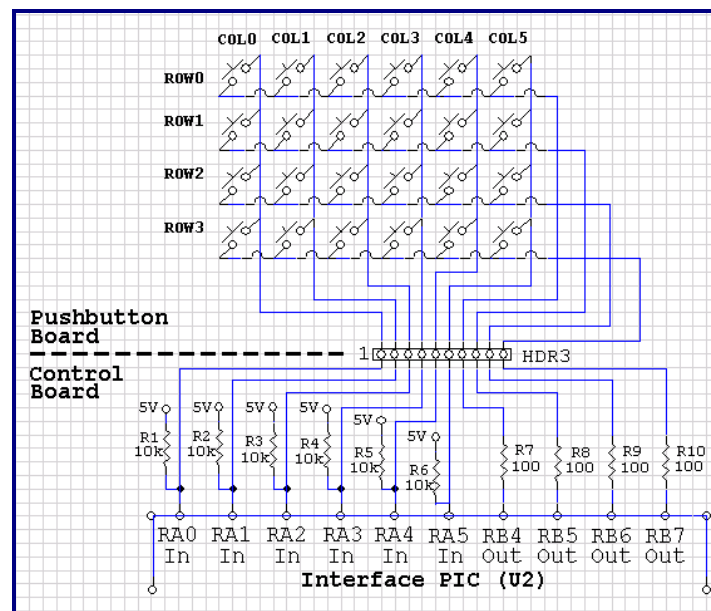

Figure 12 - Pushbutton Board Schematic

The pushbutton array board may be directly plugging it into the main (control) board's header HDR3 for testing purposes but it is intended to eventually be located on the front panel of the chassis and connected with a 10-conductor ribbon cable. The pushbutton array board may be mounted directly behind the enclosure's front panel with the pushbuttons soldered into the pushbutton array board and the pushbuttons protruding through holes in the front panel. However, this is not necessary and may not be optimum, given the close spacing of the pushbuttons on the pushbutton array board. Individual pushbuttons may be relocated to the front panel and connected to the pushbutton array board with a pair of wires.

The Interface PIC detects a pushbutton press by executing the following algorithm:
1. Set all row pins high
2. Select one row by setting it low
3. Look at each column input to see it it's being pulled low by a pushbutton.
4. If column input is low, that pushbutton (row and column) is being pressed. Exit.
5. If all column inputs are high, select the next row and repeat

Pushbuttons Row-0 Col-0 through Row-0 Col-2,
                   Row-1 Col-0 through Row-1 Col-2,
                   Row-2 Col-0 through Row-2 Col-2,
                   Row-3 Col-0 through Row-3 Col-2:   Direct-Frequency-Input Keypad

This group of pushbuttons (four rows by three columns) forms a keypad which has the same format as a telephone keypad.   It is used to allow the operator to jump to a desired frequency. The frequency is entered in kHz. To enter a frequency you start by pressing the "*" (star) button in the lower left corner of keypad. Then enter a number between 0 and 30,000. After entering 5 digits, the VFO will automatically jump to this frequency. To enter a frequency which has less than 5 digits, you can enter one or more leading zeros before the desired number, or you can enter the desired digits and then press the "star" key again to complete the direct frequency entry.

Pushbutton Row-0 Col-3:   1 MHz UP

Each time this button is pressed the frequency will increase by 1 MHz.   Frequency maximum is 30.000 MHz.

Pushbutton Row-0 Col-4:   Split Toggle

This pushbutton allows the operator to toggle between NORMAL and SPLIT mode.
SPLIT mode operates in conjunction with the keying interface header on the board.

When this keying circuit is activated (by grounding the key line or shorting the interface pins together) and it is in SPLIT mode, the PIC will switch from one VFO to the other. In NORMAL mode the frequency remains at the displayed frequency (VFO-A or VFO-B) as the keying circuit is activated.

The operator sets desired transmit frequency in one VFO and the desired receive frequency in the other VFO. The current VFO selection is done by means of the Rx VFO Toggle pushbutton. The LCD displays the frequency of the receive VFO when in receive mode and it displays the frequency of the transmit VFO when in transmit mode.   The LCD displays the word XMIT when the keying circuit indicates transmit mode and RCV when in receive mode.

The LCD displays the word SPLIT when in SPLIT mode. This portion of the LCD is blank when in NORMAL mode.

Pushbutton Row-0 Col-5:   VFOA / VFOB Toggle

Two VFO frequencies are used in the IQPro. This pushbutton allows the operator to toggle between VFO-A and VFO-B as the currently active VFO.   The LCD indicates whether VFO-A or VFO-B is active and an LED. Two VFO LEDs also indicate which VFO is active.

Pushbutton Row-1 Col-3:   1 MHz Down

Each time this button is pressed the frequency will decrease by 1 MHz.   Frequency minimum is 0.000 MHz.

Pushbutton Row-1 Col-4:   Mode Cycle

When this button is pressed the mode will change. The progression is from LSB to USB to CW- to CW+ and then back to LSB.   The LED1 will be lit if it's LSB or USB while LED2 will be lit if it's CW- or CW+.

Pushbutton Row-1 Col-5:   VFOA = VFOB

When this button is pressed the frequency of the VFO that is not currently being displayed will be changed to match the frequency of the VFO that is currently being displayed.

Pushbutton Row-2 Col-3:   Band Up

Pressing this button will advance the frequency to the low end of the next higher U.S. amateur band. Currently the band table has entries for 9 amateur bands. (The 60 meter band was not included because of its unique spot frequency requirements.) The table can easily be updated to cover the bands for other countries.

Pushbutton Row-2 Col-4:   Tone Set

This pushbutton is used to change the CW offset tone.   Pressing the pushbutton will bring up a display on the LCD which shows the current tone, as retrieved from the PIC16F877's EEPROM.   If the mode is LSB or USB, the tone value cannot be changed. However, if the mode is CW- or CW+, the tone value can be changed (within the range of 0 - 32,767 Hz) by turning the encoder.   When the tone button is pressed again, the tone value is saved in the PIC's EEPROM.

When the VFO is in RCV mode, the tone is ADDED to the base (transmit) frequency if the mode is CW+ while it is subtracted from the base frequency if the mode is CW-. The tone is not added or subtracted if the VFO is in XMT mode.

Pushbutton Row-2 Col-5:   Normal/Fast Tuning Toggle

Pressing this pushbutton toggles the optical encoder tuning rate between normal and fast tuning.
The default tuning rate allows fine tuning in 1 Hz steps when the encoder is turned very slowly. When turned slowly, one complete revolution of the encoder shaft will change the frequency by 512 Hz. However, the acceleration algorithm provides greater rate of frequency change when the encoder is turned faster. In normal tuning mode the maximum amount of frequency change in one revolution of the encoder shaft has a practical limit of about 25 kHz.

After toggling the pushbutton to fast tuning mode, the operator can still tune in 1 Hz steps by turning the shaft very slowly. However, the acceleration algorithm ramps up much faster, so the maximum tuning rate is about 150 kHz per revolution of the encoder shaft.

When in fast-tuning mode, the right-most character on the first line of the LCD displays a "+" (plus sign) if fast-tuning acceleration is in use.

Pushbutton Row-3 Col-3:   Band Down

Pressing this button will advance the frequency to the low end of the next lower amateur band. The bands are described in the section describing the Band Up pushbutton operation.

Pushbutton Row-3 Col-4:   Backlight Toggle

This pushbutton is used to toggle the LCD backlight on and off.

Pushbutton Row-3 Col-5:   Calibrate

The Calibrate button is used to put the VFO into a special mode to ensure the DDS is generating a sine wave for the exact frequency that is displayed on the LCD.

When the Calibrate button is first pressed, the VFO switches to 10 MHz and the optical encoder tuning rate is set to a very small number.   Now tune a nearby receiver to WWV at 10 MHz and listen, in CW mode, to WWV's tone at some offset (e.g. 700 Hz above). You can hear the IQPro putting out a faint signal at what "it thinks" is 10 MHz with the same tone offset. While listening to the WWV tone and the IQPro tone, turn the encoder to adjust the IQPro's tone. As the two frequencies get closer together, you can hear the "beats" get slower and slower. Listen for "zero beat" which indicates the tones are identical. When the two tones are identical it is calibrated. Now press the Calibrate button again to exit this mode. The updated constant is now saved in EEPROM.

4.3   Other Controls

A keying circuit interface header is provided on the board for a connection to external keying circuitry. It is used for switching between the two VFO frequencies when in SPLIT mode. The LCD changes from RCV to XMIT when the keying interface pin is grounded (or the two header pins are logically connected together). In addition, if SPLIT mode is selected, the PIC changes to use the opposite VFO frequency when the keying interface pin is activated (grounded).

Test this feature by entering different frequencies in VFO-A and VFO-B. Select SPLIT operation. Now short the two header pins together and watch the display change from one VFO to the other and the display change from RCV to XMT.   By design, the currently selected VFO is always the "receive" VFO, and the receive frequency is displayed. The other VFO contains the "transmit" frequency and it is displayed when transmitting.

4.4   LCD Display

The LCD display shows the current frequency that is being generated by the DDS. It also shows the state of the SPLIT toggle, the current transmit/receive mode, and the currently active VFO (A or B). When direct frequency entry is being entered via the pushbuttons, the digits are displayed on line 2.

A pair of resistors (R15 and R16), forming a voltage divider, are used to adjust the LCD brightness/contrast. These resistors are not be necessary for all LCDs but are often are required for proper contrast. For some types of LCDs, pin 3 may be tied directly to ground, while in other LCDs need a small positive voltage is required on pin 3 for maximum brightness. The resistor divider has R15 (820) and R16 (5.6k) connected in series, with R16 connected to VDD (+5v) and R16 connected to ground. Pin 3 is connected to the junction of R15 and R16. Thus, the voltage at LCD pin 3 is calculated in Figure 13. If you use an LCD other than the one supplied in the IQPro kit you may have to determine the correct bias voltage empirically.

$$V = 5 * \frac{820}{(5600+820)} = .64 \text{ v}$$

Figure 13 – LCD Contrast Bias Calculation

If 0.64v is not correct for the contrast control for your LCD, simply adjust the size of R15. R15 can be shorted out (replaced by a piece of wire) if 0 v is the appropriate voltage for Pin 3. In this case you may leave out R16. Obviously you don't want to short out R16.

Virtually any 16 x 2 LCD which uses a HD44780 controller can be used. However, the backlight current requirements need to be considered. The current-limiting resistor (R68) in the backlight's power line is 6.2 ohms for the LCD supplied in the kit. If another LCD is used, this value may have to be adjusted. The LCD backlight is toggled on and off by a PIC pin driving a MOSFET. A pushbutton (Row-3, Column-4) toggles a PIC pin which drives the gate of MOSFET Q9 through R69. When the PIC pin is set high, the Q9 is "turned on" so current flows through the MOSFET from Drain to Source. When the PIC pin is set low, Q9 is "turned off" so current flow is shut off. An IRLML2502 MOSFET was selected for this application because of its very low resistance (.045 ohms) when the device is turned on, allowing the high current flow with the 5v supply.

4.5   LEDs

There are no Light Emitting Diodes (LEDs) on the Main Board. Instead, LED connections are made via Header 2 on the Main Board. This header has nine active pins (one per LED) plus a ground. Each header pin goes to +5v when the LED is turned on. Figure 14 illustrates the LED layout.
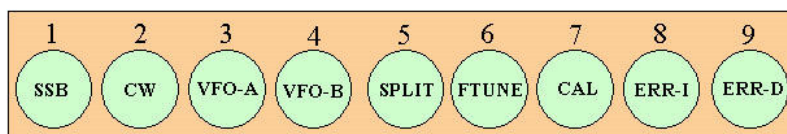

Figure 14 - LED Layout

LED 1 – This LED indicates that Single Sideband mode is active.
LED 2 – This LED indicates that CW mode is active.
LED 3-- This LED is turned on when VFO-A is active in receive mode.

LED 4-- This LED is turned on when VFO-B is active in receive mode.
LED 5-- This LED is turned on when SPLIT operation is active.
LED 6-- This LED is turned on when Fast Tuning is active.
LED 7-- This LED is turned on when Calibrate is active.
LED 8-- This LED is turned on when an error is detected in the Interface PIC.
LED 9-- This LED is turned on when an error is detected in the Driver PIC.

5   Relay/Filter Board

A relay/filter board is not included in the IQPro kit because requirements for individual applications will vary widely. The concept is shown in Figures 15, 20 and 22. Note that the two sides (a and b) are implemented with the two poles of a DPDT relays rather than with separate relays for the two sides. The band relay latching signals from header HDR5 are pulsed when the frequency changes to a new band range. See the Band Relay Driver section regarding generation of the band relay signals.

The signals generated by the Driver PIC and presented on HDR5 are pulsed for approximately 8 ms. The amount of current that is "sourced" by a PIC pin (to HDR5) is limited to about 20 ma which is exactly the amount of current that this particular type of relay (RY1) requires to set or reset. However, 20 ma is probably NOT sufficient to drive several relays in parallel. The design assumes 20 ma single coil latching relays are used, each with a common side tied together. To set a relay, the PIC drops all of the band-switch pins to a low level and then raises the single pin for the relay to be set to a high level. Approximately 20 ma of current flows in the "forward" direction through the coil and sets (and latches) the relay. Note that this implementation requires that the other relays in the bank will "share" the return current (5 ma each). To reset a relay, the PIC raises all of the band-switch pins to a high level concurrently and then drops the single pin of the relay to be reset. Approximately 20 ma of current flows in the "reverse" direction through the relay's coil and resets (and latches) the relay. Note that the 20 ma current in the 'forward" direction flows through the other four relays in the bank but is low enough such that it does not set any of them.

If you want to drive several relays concurrently from one HDR5 pin you should consider using switching components which in turn enable sufficient current flow to latch the relays. MOSFETs (e.g. 2N7000 or IRLML2502 or IRF510) work well in this type of application. For an example, see the LCD backlight enable mechanism using Q9 (an IRLML2502 MOSFET) in the main (control) board schematic (Figure 23). Q9 is driven by a PIC pin which enables the LCD backlight current (75 ma) to flow when the PIC pin is set to a high level.

Note that the PIC software remembers which relay was previously set and resets it before setting the new relay. The exception is on power-on in which case the previous value is unknown. For this reason, the PIC initialization routine resets all the relays, one at a time, before setting the default relay.
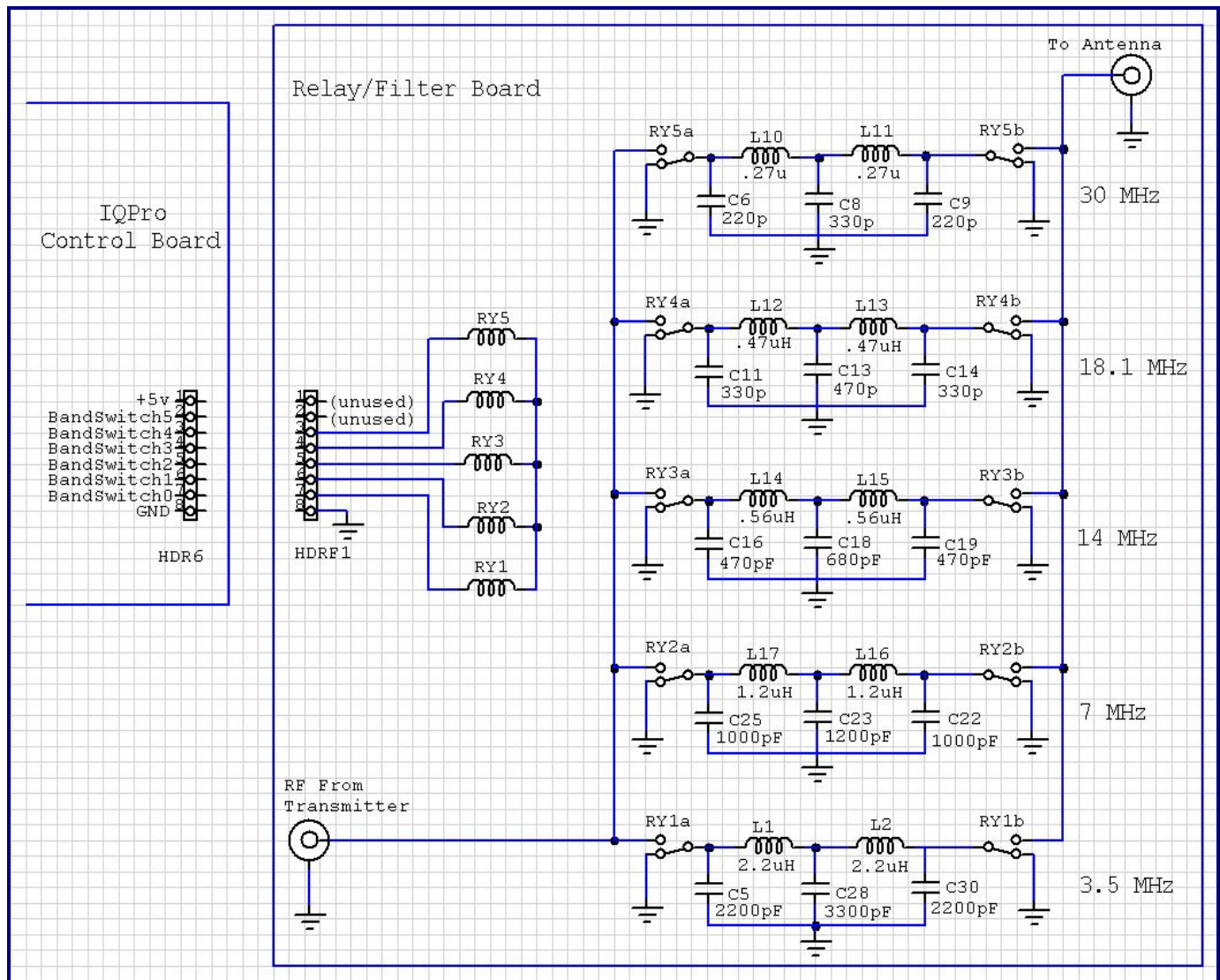
Figure 15 – 5-Band Filter Board Example

6 Switching Power Supply Board

The main board requires a 5v input power source and it is customary for most builders to use 12v power for their equipment. To make this conversion easy, a small switching power supply board is supplied with the IQPro kit. This power supply board converts 12v to 5v without dissipating a lot of heat. Figure 16 shows the schematic for this power supply. The conversion from 5v to 3.3v (required by the AD9854 DDS and its reference clock) is done on the board.
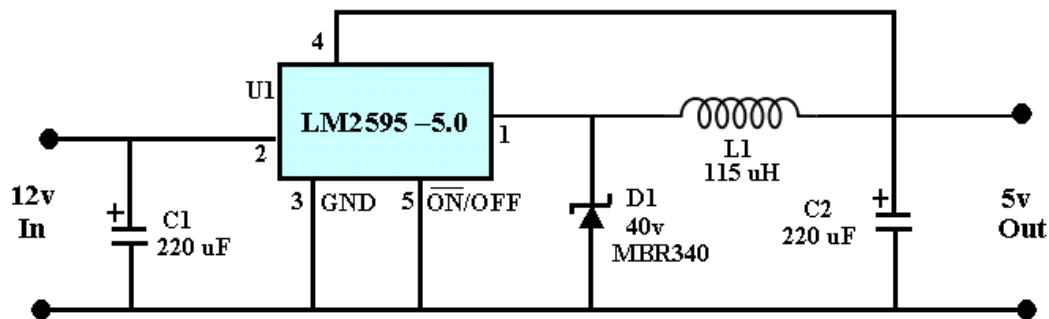
Figure 16 – Switching Power Supply Schematic

7 IQPro Applications

Here are some examples of how the IQPro can be used in a variety of applications.

7.1 IQPro in R2 (or R2Pro, etc) Receiver
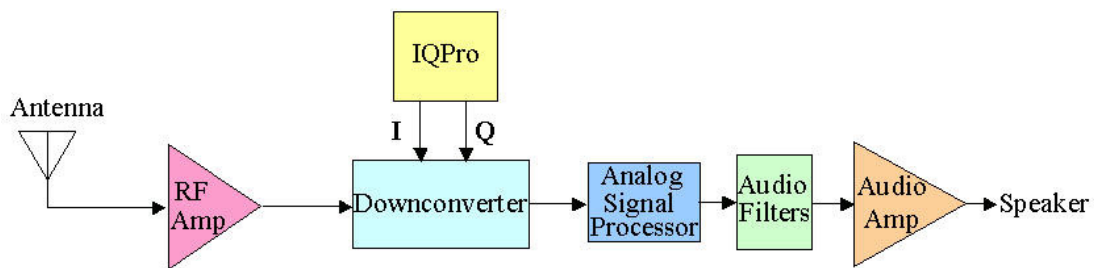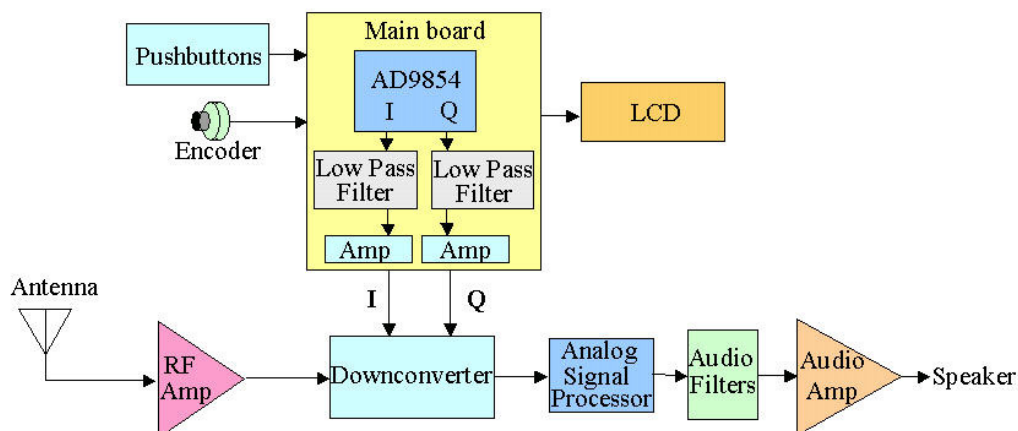

Figure 17 – IQPro in R2 Receiver – Basic

Figure 18 – IQPro in R2 Receiver – Details

7.2 IQPro in CW Transmitter
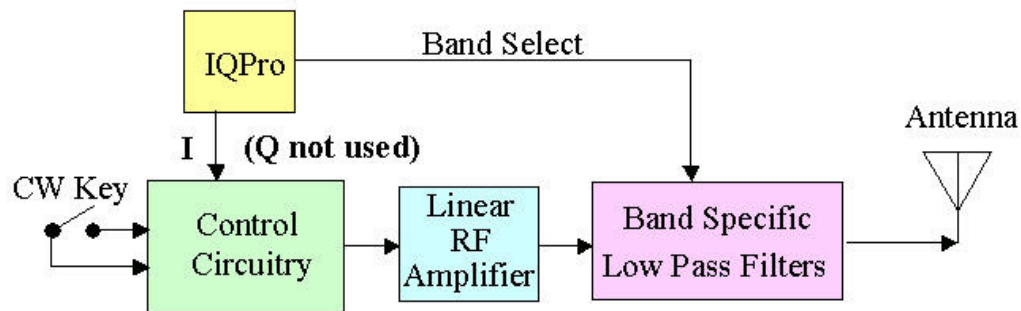
Figure 19 – IQPro in CW Transmitter - Basic
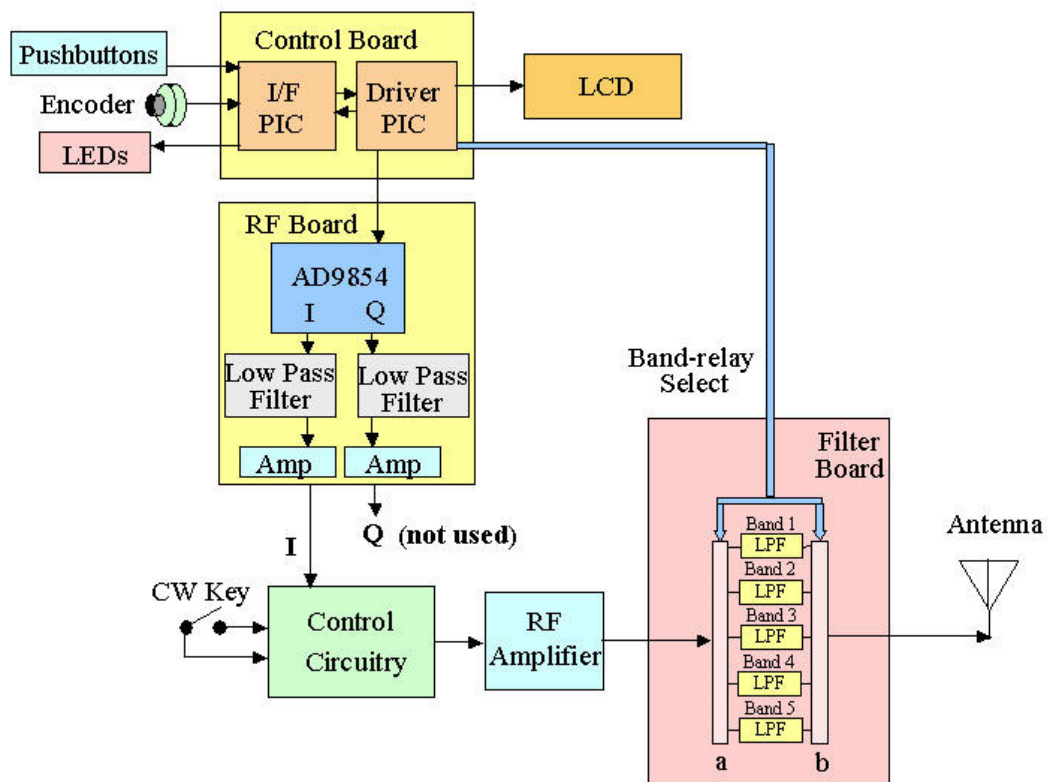
Figure 20 – IQPro in CW Transmitter - Details

7.3 IQPro in SSB Transmitter
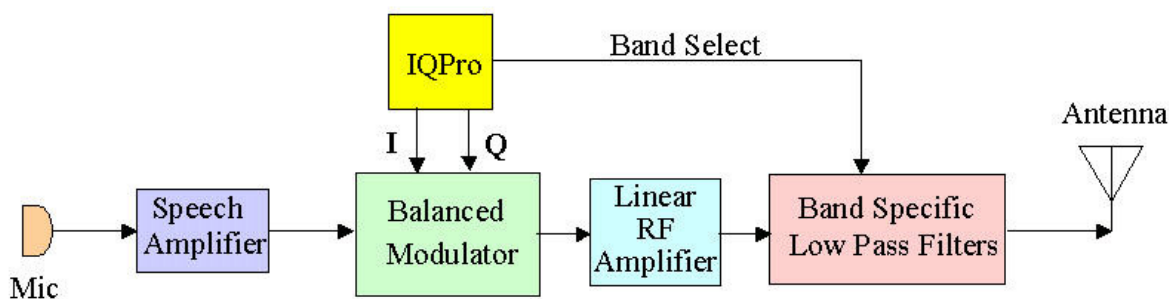
Figure 21 – IQPro in SSB Transmitter - Basic
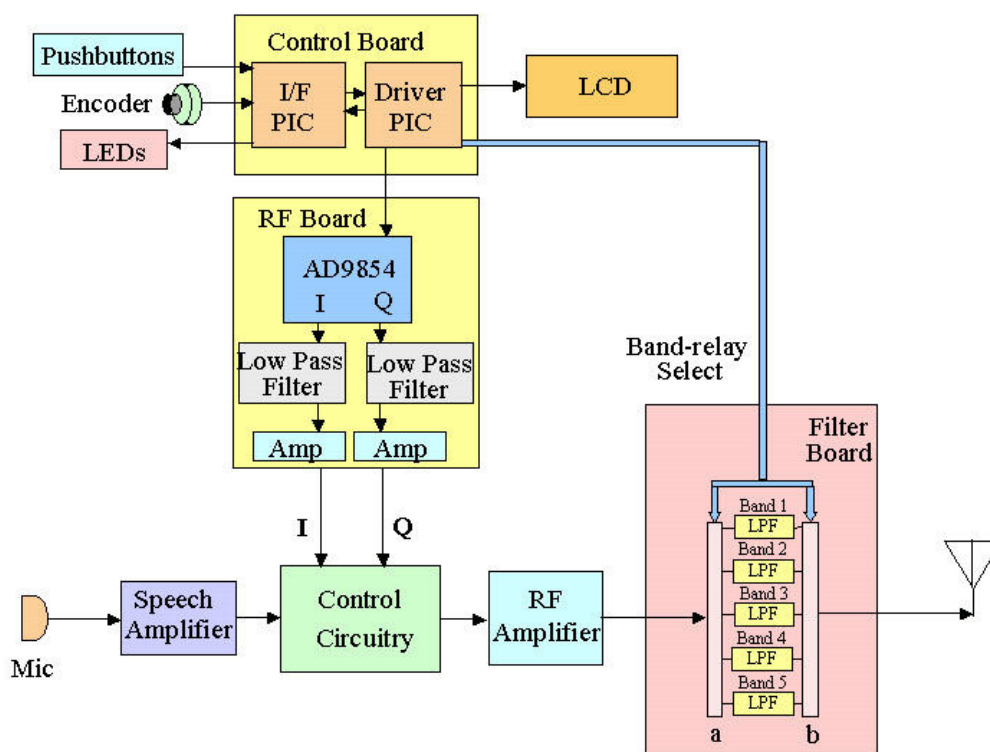
Figure 22 – IQPro in SSB Transmitter - Details

8 DDS Frequency Calculation Details

When we want the AD9854 to generate a sine wave at a particular frequency, we need to send it a 6-byte number referred to as the Frequency Tuning Word. The AD9854 uses the Frequency Tuning Word and the reference clock to generate the output waveform.

The AD9854 can generate an output frequency with 48-bit accuracy. However, in this application, since we are always going to be generating a frequency which is a whole number of Hertz, we keep a 4-byte numeric constant in the PIC's EEPROM which represents a 1 Hertz change in output frequency when used with the specified reference clock. We refer to these 4 bytes as:

ref_osc_3 (the most significant byte)
ref_osc_2 (next byte)
ref_osc_1 (next byte), and
ref_osc_0 (the least significant byte)

Note that the AD9854 has a 48-bit frequency tuning word and we are only saving a 32-bit constant in EEPROM. For performance reasons, and since 48-bit accuracy is not required for this application, we will construct the 48-bit tuning word from the 32-bit value. We will use this value to calculate the 48-bit frequency tuning word. We will then multiply the 32-bit value by the desired output frequency, in Hertz, to obtain another 32-bit product. This 32-bit product will be used as the most significant 32 bits of the 48-bit frequency tuning word and we'll fill the remaining 16 bits with zeros.

This numeric constant is interpreted as a fixed-point real number in the format:

<refosc3> . <refosc2> <refosc1> <refosc0>
> where:
>   **refosc3** = $(2^{32} / osc\_freq\_in\_Hz)$
>   **refosc2**, **refosc1**, and **refosc0** are the fractional part of: $(2^{32} / osc\_freq\_in\_Hz)$ times 2^{24}.

Example:
> For a 125 MHz clock:
>   **refosc3** is $(2^{32} / 125 \times 10^6)$ = 34.359738368 truncated to 34 (0x22)
>   **refosc2** is the high byte of $(.359738368 \times 2^{24})$ = 6035408.303
>   6035408.303 = 0x5C17D0, so high byte is 5C.
>   **refosc1** is the next byte of 0x5C17D0, or 17
>   **refosc0** is the last byte of 0x5C17D0, or D0

The values for common oscillator frequencies are as follows:

Frequency	refosc3	refosc2	refsc1	refosc1
125.00 MHz	0x22	0x5C	0x17	0xD0
120.00 MHz	0x23	0xCA	0x98	0xCE
100.00 MHz	0x2A	0xF3	0x1D	0xC4
90.70 MHz	0x2F	0x5A	0x82	0x7A
66.66 MHz	0x40	0x6E	0x52	0xE7
66.00 MHz	0x41	0x13	0x44	0x5F
60.00 MHz	0x47	0x95	0x31	0x9C
50.00 MHz	0x55	0xE6	0x3B	0x88

Once again, we can think of this constant number as the step size which represents 1 Hz of movement. When we want to set the VFO's frequency to a specific number of Hertz, we simply have to multiply this constant by the desired number of Hertz to obtain a Frequency Tuning Word. We send this Frequency Tuning Word to the AD9854 and, *voila*, it generates the sine wave at the specified frequency.   In fact, it generates two outputs exactly 90 degrees apart.

Now it is obvious why the Calibrate function is necessary. When the PIC microcontroller is programmed, preset values are loaded into the PIC's EEPROM, based on the nominal frequency of the AD9854's external reference clock. However, it is never absolutely accurate as specified. The Calibrate function is used to adjust the 4-byte constant (step size) such that when the AD9854 uses it with this reference clock, the output

frequency is accurate.   The updated 4-byte constant is then stored back in EEPROM for future use in calculating the Frequency Tuning Word. It is not lost when the IQPro is powered down.

9   IQPro Spectral Measurements

The question if spectral purity is always the subject of a lot of discussion in any forum in which DDS VFOs are discussed. This is a very legitimate question.

The IQPro has 30 MHz low-pass filters for the AD9854's I and Q signals. These filters are located between the toroids and the transistor amplifiers. (The toroids or transistor amplifiers may be bypassed, depending on the amplifier option selected. See Section 3.6.) The main purpose of these low-pass filters is to remove the DDS fold-back signals (signals that occur at the DDS reference clock frequency minus the fundamental frequency) that are produced by any DDS. The amplitude of these fold-back signals follow the SIN(X)/X curve. See the AD9854 specification sheet for more details.

Measurements of the IQPro's DDS outputs were made with a TEK 2712 spectrum analyzer. For these measurements, the toroids and transistor amplifiers were in use and the signals levels were set at the maximum.

Harmonics were seen in the output spectrum. When the DDS frequency is 10 MHz, the second harmonic of about -26 dBc was seen as well as a third harmonic of about –31 dBc. When the DDS frequency is set to 25 MHz, the second harmonic is about -27 dBc but the third harmonic is about –24 dBc.

Harmonics can be produced by nonlinearities anywhere in the RF chain so band-specific filtering on the DDS board would not be sufficient. For a receiver, harmonics at these levels should not be detrimental. For a transmitter, the output signal must be filtered by a band-specific low-pass filter at the transmitter's output so, once again, these harmonics should not be detrimental. Obviously, one low-pass filter cannot cover all of the HF ham bands. A low-pass filter that is used to remove the second harmonic from a 40 meter (7 MHz) fundamental could not be engaged when running at 20 meters (14 MHz fundamental), etc.

I found that the spectrum was very clean with no noticeable spurs above the noise level until the frequency reached 125 MHz.   Then the reference clock was seen (about –38 dBc).

The IQPro supplies signals at a header (HDR5) which facilitate the engagement of a band-specific low-pass filters at the transmitter's output, just before the signal is sent to the antenna. Low-pass filtering at this level will bring the transmitted spectrum to a level that is FCC compliant.

10 Questions/Support

For up-to-date details and documentation regarding this project, please see my web page, **www.cbjohns.com/aa0zz** .
The latest version of the code for the two PIC microcontrollers can also be found on the web site.

For additional support questions, see the YAHOO group DDS-VFO.

11 Conclusion

Is a DDS-based VFO "right" for your radio? It really depends. There are trade-offs in any design and clearly, a DDS-based VFO is not right for every application. However, the ability of the AD9854 DDS to produce I and Q signals throughout the 0 – 30 MHz tuning range is intriguing.

It is my opinion that most homebrew projects are never completely done. By the time you finish one version you are tempted to do it again - to make it even better. The IQPro is one building block of a radio that can be customized forever. That's why the source code is distributed. As we all experiment with various configurations and share them with others, we learn together. That's the real value of building the IQPro DDS VFO.

Appendix A – Assembly Notes

Note: This is not intended to be an all-inclusive set of building steps.

1) Decide whether you want to divide the two boards (Control and RF). If you want to decide to keep them together you can simply install jumpers between the appropriate pins on HDR7 and HDR10, HDR8 and HDR11, HDR9 and HDR12 without installing the actual headers. If you separate them you should install all the headers.

2) Make sure you use a socket for the two PIC microcontrollers i.e, don't solder the PICs in. You may want to update the firmware at some later time and the programming cannot be done in-circuit.

3) Put a small paper washer under the 20 MHz crystal (XTAL) or else mount it with a small space between the board and the crystal. This will ensure that the plated-through holes are not shorted by the crystal case.

4) Use a 6-32 x 1/2" bolt to secure the 3.3-volt regulator to the board. It lays flat on the board and no additional heat sink is needed.

5) The "plus" end of the tantalum capacitors has a small "+" near the lead but the heavy printed bar near the "plus" lead is much more obvious.

6) The dot on the end of resistor network R1-R5 goes to the square pad on the PC board.

7) Assemble the 5-volt circuitry first. This includes

- The PICs (sockets),
- The headers HDR1 through HDR6,
- The 20 MHz crystals (XTAL1 and XTAL2)
- Capacitors C1 – C15
- Resistors R1 – R18, R34 – R42
- Diodes D1, D2
- Switch S1

6) Install R68, R69, R70 and Q9 before you mount the headers on the PIC board.
     Those parts control the backlighting of the LCD

7) Test 5-volt circuitry.   Apply 5 volts to the input pad (PAD1) and make sure you have 5 volts on these PIC socket pins:
   U2 pins 1, 11, 32
   U3 pins 1, 11, 32

8) Insert the PICs in the sockets

9) Connect the encoder to header HDR1

10) Connect the LCD to header HDR4. Pin 1 of HDR4 goes to Pin 1 of LCD, etc.

11) Plug the LED Test board into HDR2

12) Plug the pushbutton board into HDR3

13) Verify operation of all 5-volt circuitry. Encoders, pushbuttons, LEDs, and the LCD.

14) Mount the 3.3v regulator (U1).

15) Mount header HDR13 but DO NOT put the jumper across yet.

16) Mount the clock (REFCLK). Pin 1 of the REFCLK has the small dot at the corner (near the "BN" printed on the package).

17) Add AD9854 components:

- Capacitors C16, C17, C19, C20 (There is no C18)
- Resistors R19 - R33

18) Add a jumper to short the two pins of header HDR13

19) Add connections from HDR7 to HDR10 (+5v and ground)

20) Add connections from HDR8 to HDR11 (I/Q relay)

21) Add connections from HDR9 to HDR12 (AD9854 control signals)

22) Power up and observe the two outputs at the un-grounded ends of R30 and R33

23) Select your output amplifier method. Add parts for low pass filters and toroids/amplifiers

24) If toroids are used (amplifier options 1 or 3), R30 through R33 are REMOVED. If option 1 is used (transistor amplifier only) then R30 through R33 are left in.

25) Note that the resistors in the amplifier are 1% tolerance parts.

26) Resistors, capacitors and diodes in the amplifier are mounted vertically (on end).

27) Pin 1 of the I/Q relay is at the banded end of the relay

28) NOTE: Diodes D3, D4, D5, D6 are mounted with banded ended UP.

29) There is NO R67.

30) There is NO C18.

31) The "banded" end of relay RY1 goes to the left – toward the AD9854.

32) Power up and observe the two outputs, I-OUT and Q-OUT

33) I like to use four 6-32 x 1/2" bolts/nuts with 5/16" spacers for stand-offs in the mounting holes in the corners of the board. They prevent accidental short-circuits that can occur when you accidentally lay the powered-up board on a bit of wire or a piece of solder. Don't think it can't happen!

Appendix B - Schematic
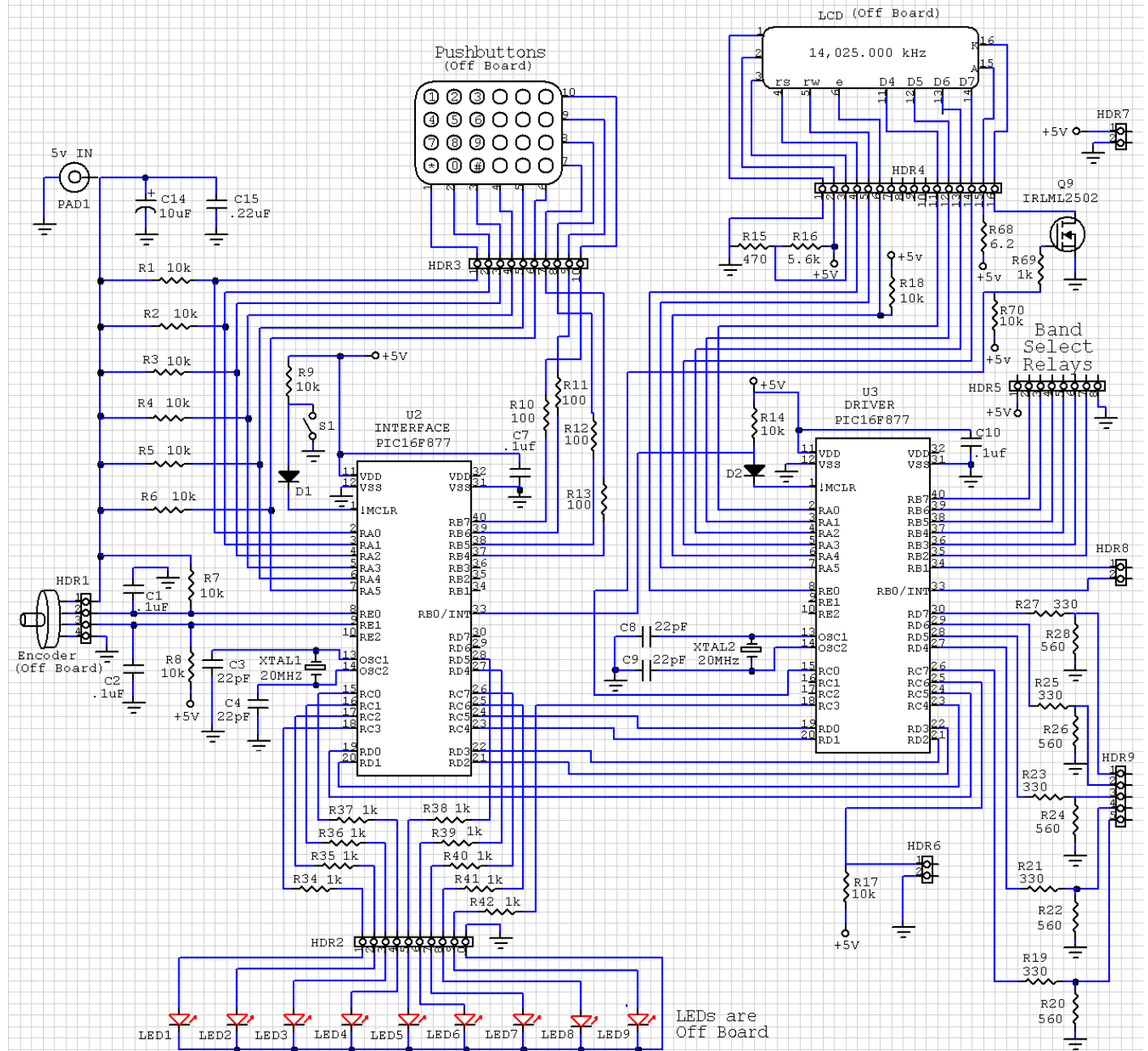
Figure 23
IQPro Schematic – Page 1 of 2
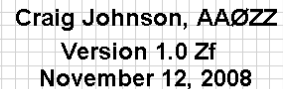
Figure 24
IQPro Schematic – Page 2 of 2

Appendix C - IQPro Amplifier Simulation

Figure 25 shows the output waveform and the DC operating voltages of a simulated complementary amplifier at 14 MHz.   (Simulation done with CircuitMaker2000.)

Figure 25 – Simulation - Output Voltage Waveform

Figure 26 shows the currents in the two complementary output transistors of the simulated amplifier. (Simulation done with CircuitMaker2000.)

Figure 26 – Simulation – Amplifier Output Transistor Currents

Appendix D – Layout

IQPro Parts Layout – Page 1 of 2
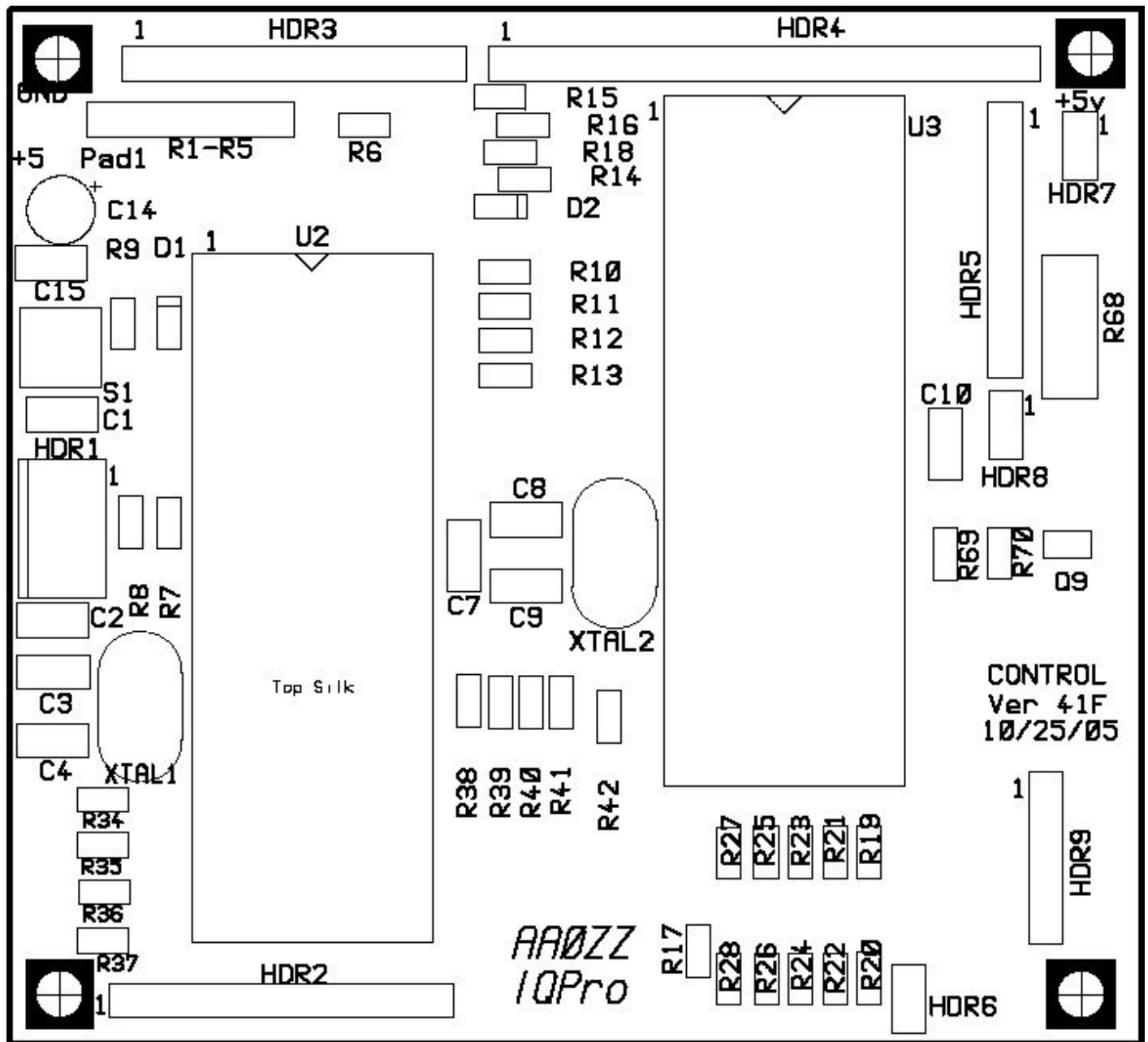
IQPro Parts Layout – Page 2 of 2

Appendix E - Parts List

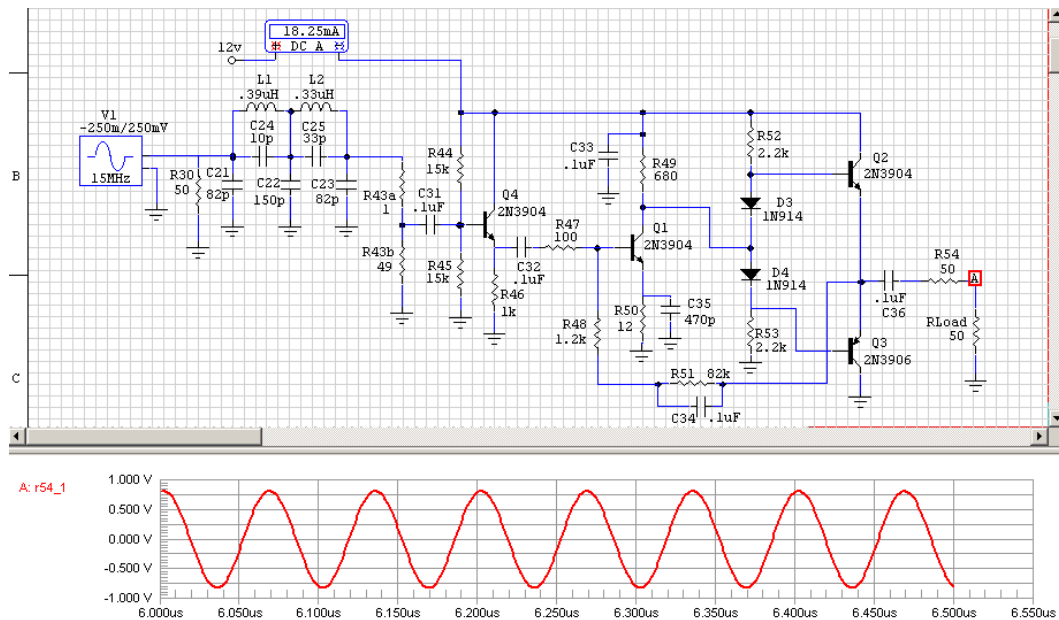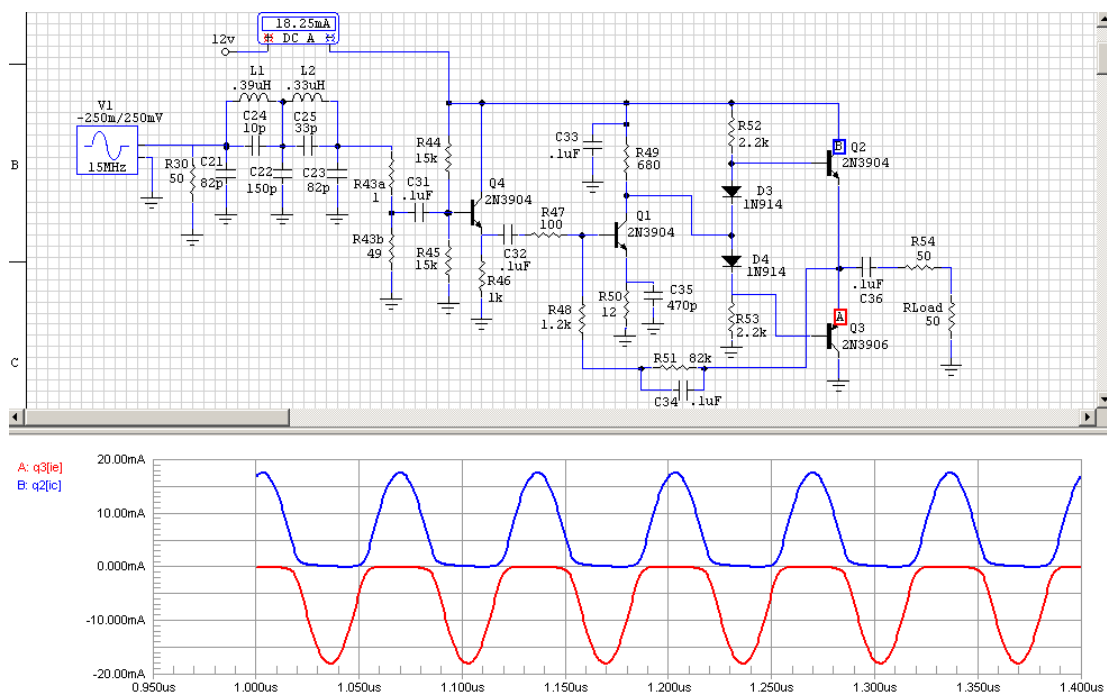| Part	Schematic Part Name	Part Number (DigiKey unless otherwise specified)	Quantity Needed	Small Quantity Price	Total at Small Quantity Price
IQPro Parts					
		Ver. 2.0g		1/3/2010	
Main Board					
3.3 volt regulator TO220	U1	LT1085CT-3.3#PBF-ND	1	6.50	6.50
Crystal Oscillator 3.3v 125 MHz (SMT)	REFCLK1	XC346CT-ND **(or 520-3953M-1250-BN)**	1	5.55	5.55
Crystal 20 MHz	XTAL1, XTAL2	CTX416-ND	2	0.80	1.60
Resistors (1/8 watt - 5%)					
51 ohm	R30, R33, R54, R66	51EBK-ND	4	0.05	0.21
24 ohm	R31, R32	24EBK-ND	2	0.05	0.10
100 ohm	R10-R13	100EBK-ND	4	0.05	0.21
330 ohm	R19, R21, R23, R25, R27	330EBK-ND	5	0.05	0.26
470 ohm	R15	470EBK-ND	1	0.05	0.05
560 ohm	R20, R22, R24, R26, R28	560EBK-ND	5	0.05	0.26
1k ohm	R34-R42, R69	1.0KEBK-ND	10	0.05	0.52
3.9k ohm	R29	3.9KEBK-ND	1	0.05	0.05
5.6k ohm	R16	5.6.KEBK-ND	1	0.05	0.05
10k ohm	R6-R9, R14, R17, R18, R70	10KEBK-ND	8	0.05	0.42
Resistors (1/8 watt - 1%)					
12 ohm	R50, R62	**(M) 270-12-RC**	2	0.11	0.22
100 ohm	R47, R59	**(M) 270-100-RC**	2	0.11	0.22
470 ohm	R46, R58	**(M) 270-470-RC**	2	0.11	0.22
680 ohm	R49, R61	**(M) 270-680-RC**	2	0.11	0.22
1.2k ohm	R48, R60	**(M) 270-1.2K-RC**	2	0.11	0.22
2.2k ohm	R52, R53, R64, R65	**(M) 270-2.2K-RC**	4	0.11	0.44
15k ohm	R44, R45, R56, R57	**(M) 270-15K-RC**	4	0.11	0.44
82k ohm	R51, R63	**(M) 270-82K-RC**	2	0.11	0.22

Resistor Network					
10k ohm x 5	R1-R5	770-61-R10KP-ND	1	0.28	0.28
Resistors (1/2 watt)					
6.2 ohm	R68	6.2H-ND	1	0.05	0.05
Variable Resistors					
50 ohm - 25 turn	R43, R55	490-2924-ND	2	0.85	1.70
Capacitors (ceramic)					
0.1 uF - radial	C1, C2, C7, C10, C11, C16, C17, C19, C20	BC1148CT-ND	9	0.07	0.59
0.22 uF - radial	C6, C13, C15	490-3862-ND	3	0.22	0.66
10 pF - axial C0G	C24, C29	1011PHCT-ND	2	0.12	0.24
33 pF - axial C0G	C25, C30	1016PHCT-ND	2	0.12	0.24
82 pF - radial C0G	C21, C23, C26, C28	490-3768-ND	4	0.28	1.12
150 pF- radial C0G	C22, C27	490-3689-ND	2	0.29	0.58
470 pF - axial X7R	C35, C41	1113PHCT-ND	2	0.08	0.16
.1 uF - axial	C31-C34, C36-C40, C42	399-4491-1	10	0.20	2.00
Capacitors (Ceramic disc)					
22 pF NP0	C3, C4, C8, C9	495-1004-1-ND<<obsolete **490-3639-ND <<new**	4	0.31	1.24
Capacitors (Tantalum)					
10 uF tantalum	C5, C12, C14, C43	399-3638-ND	4	0.71	2.84
Inductor					
.33 uH	L2, L4	**(M) 807-1641-331K**	2	1.15	2.30
.39 uH	L1, L3	**(M) 807-1641-391K**	2	1.15	2.30
Diodes					
1N4148 DO-35	D1- D6	1N4148DICT-ND	6	0.23	1.38

MOSFETs					
IRLML2502 (SMT)	Q9	IRLML2502PBFCT-ND	1	0.77	0.77
Transistors					
2N3904 TO-92	Q1-Q3, Q5-Q7	2N3904D26ZCT-ND	6	0.19	1.14
2N3906 TO-92	Q4, Q8	2N3906D26ZCT-ND	2	0.19	0.38
Toroids etc					
FT37-43	T1, T2		2	0.20	0.40
magnet wire (#28, 3 colors)	4 ft.of each color	Amidon, etc.	1	1.00	1.00
nylon screw (6-32 x 3/8")	(for T1 and T2)	H556-ND	2	0.09	0.19
nylon nut (6-32)	(for T1 and T2)	H620-ND	2	0.11	0.22
machine screw (6-32 x 3/8")	(for U1 regulator)	H356-ND	1	0.02	0.02
machine nut (6-32)	(for U1 regulator)	H220-ND	1	0.01	0.01
OTHER					
40-pin DIP socket (.600")	For U2, U3	A411-ND	2	1.94	3.88
Straight Header (.100" spacing) (2-pins)	HDR6, HDR13	WM4000-ND	2	0.18	0.36
Straight Header (.100" spacing) (16-pins)	LCD wiring harness	(D) **A26520-16-ND** (M) 571-16404526	2	1.10	2.20
Locking Header (.100" spacing) (4-pins)	HDR1	WM4202-ND	1	0.42	0.42
Locking Socket (.100" spacing) (4-pins)	For HDR1	WM2002-ND	1	0.29	0.29
SIP Socket (.100" spacing) (8-pins)	HDR5	(D) ED7150-ND (cut)	1	2.21	2.21
SIP Socket (.100" spacing) (10-pins)	HDR2, HDR3	(D) ED7150-ND (cut)	2	1.38	2.76
SIP Socket (.100" spacing) (16-pins)	On LCD	(D) ED7150-ND (cut)	1	2.21	2.21
SIP Socket (.100" spacing) (16-pins)	HDR4	(D) ED7150-ND (cut)	1	2.21	2.21
SIP Socket (.100" spacing) (2-pins)	HDR7, HDR8, HDR10, HDR11	(D) ED7150-ND (cut)	4	0.28	1.11
SIP Socket (.100" spacing) (5-pins)	HDR9, HDR12	(D) ED7150-ND (cut)	1	0.69	0.69
Pins for locking sockets	For HDR1	WM1114-ND	4	0.12	0.48
Shunt	For HDR6	929950-00-ND	1	0.23	0.23
Pushbuttons 5.0mm SPST	S1	P8079SCT-ND	1	0.38	0.38
Relay (DPDT Latching)	RY1	255-1004-5-ND	1	3.16	3.16
Optical Encoder (128 positions)	Encoder	600CS-ND (old but OK) **601VCS-ND is OK too**	1	29.93	29.93
LCD with backlight(16 x 2 characters with HD44780 controller)	LCD	08LCD9 fromwww.ElectronixExpress.com	1	8.95	8.95

AD9854AST DDS	U4	(Analog Devices)	1	29.52	29.52
(Installation of AD9854)			1	10.00	10.00
PIC16F877A 20 MHz (programmed)	U2, U3	AA0ZZ supplied	2	12.50	25.00
PCB (pair)	Control, RF		1	19.00	19.00
Printed Manual			1	7.50	7.50
Pushbutton Test Board (included)					
Pushbutton 5.0mm SPST	PB1-PB24	P8079SCT-ND	24	0.38	9.12
Straight Header (.100" spacing) (10-pins)	HDR1	WM4008-ND	1	0.87	0.87
PCB			1	2.00	2.00
BANDSWITCH Test Board (included)					
LED T1.75		**67-1110-ND**	6	0.17	1.01
Straight Header (.100" spacing) (8-pins)		WM4006-ND	1	0.77	0.77
1k ohm 0805		311-1.00KCRCT-ND	6	0.08	0.48
PCB			1	1.00	1.00
LED Test Board (included)					
LED T1.75	LED0-LED9	**67-1110-ND**	9	0.17	1.51
Straight Header (.100" spacing) (10-pins)	HDR1	WM4008-ND	1	0.87	0.87
PCB			1	1.00	1.00
POWER SUPPLY (included)					
Voltage Regulator	U1	LM2595T-5.0-ND	1	3.85	3.85
Inductor 115 uH, .67A	L1	**TE2175-ND**	1	2.35	2.35
Capacitor 220 uF, 25v	C1, C2	P12383-ND	2	0.32	0.64
Diode - Schottky	D1	MBR340IR-ND	1	0.83	0.83
PCB			1	2.00	2.00

			1		
					========
				Total Cost	**220.59**