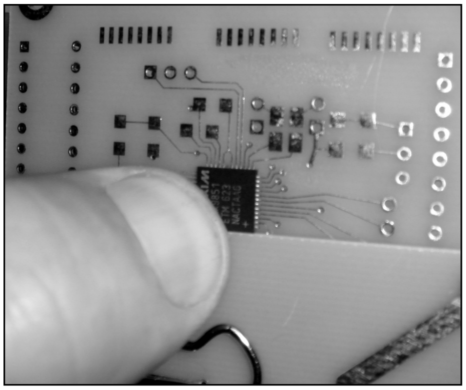

**Figure 5—CODEC ready for soldering.**

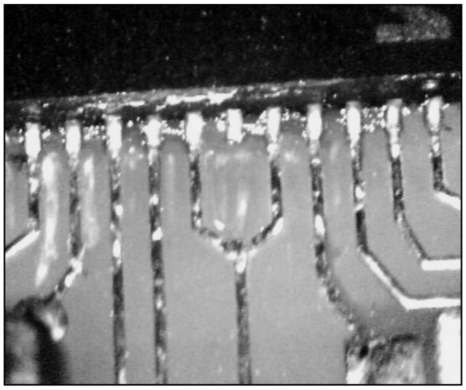

**Figure 6—CODEC after soldering.**

Here's one way it can be done: First, using a couple of report clips, a scrap piece of p.c. board was attached to the target board. Figure 4 shows how this is done. Because the boards are rectangular, the rotational alignment of the two PC boards is trivial. While the two report clips hold the board together snugly, it is still possible to move one up and down. Movements as small as a few thousands of an inch are easily controlled. The CODEC chip is lightly pushed against the "alignment" board. Thus, with minimal effort the CODEC chip can be aligned longitudinally and rotationally.

Using a pair of jeweler's glasses simplifies the alignment of the CODEC along the edge of the alignment board. Again, this turns out to be a relatively simple exercise. The CODEC is easily held in place with a single hand. Figure 5 shows the CODEC in place ready to solder.

The actual soldering also turned out to be rather simple. A small amount of solder paste was purchased at a local electronics store and a small syringe at a craft store. To make the solder paste flow more evenly it was cut with a few drops of rubbing alcohol. While holding the chip in place with a thumb, a small amount of solder paste was applied to one corner of the chip. With the chip aligned and the solder paste touching the actual pad and trace, a brief touch with a fine point soldering iron applied the necessary heat. The solder paste immediately melted and flowed directly to where it was needed… one corner was attached.

Once the chip was held firmly in place it was possible to remove the clips and alignment p.c. card. The alignment of the chip is examined on all sides. If alignment is not good enough, a little hot air or a glob of solder could be used to de-solder the part. After using some solder wick (dipped in flux) to remove any extra solder on the chip and p.c. board another attempt could be made.

Once the chip alignment was ensured, the next step was to solder the rest of the pads. This can be done by running a very small bead of solder paste down the edge of the chip. Then, using a very fine tip on a normal soldering iron, the paste was melted. Although it sounds cliché, solder really does want to go to the right places. This technique ALWAYS delivers solder between the chip pads and the PC etches; the solder wicks into the tiny spaces where it is needed. The only problem, unfortunately, is that even with solder paste which has been thinned extensively, too much solder is delivered and there will be shorts between the PC etches. However, solder being what it is, it a little flux and a very fine piece of solder wick quickly and easily removes the excess solder. Repeat this procedure on the other three sides and VOILA!!

Like many other skills, this process took a little practice. Ultimately the CODEC could be placed quickly and reliably. Figure 6 shows the result of this soldering approach before the rosin was cleaned away.

## The SDR Digital Subsystem

Having developed a technique to successfully mount the CODEC it was time to start the design of the digital part of the SDR. SDR software can be divided into two pieces: the control plane and the data path. The control plane is used for both human interface and for control of the CODEC. The data path is used for manipulation and forwarding of signal-related data. The original thinking was to use two independent processors, one for each path, connected using some kind of serial link. While this may ultimately prove necessary, the present implementation uses a single processor to implement both subsystems.

Even though the processor DDS and CODEC had already been chosen, there remained numerous, less critical decisions. For example, the majority of digital logic these days runs on 3.3 volts (or lower) and the CODEC and DDS run on 2.5 and 1.8 volts respectively. How should one provide these lower voltages? One could use simple linear voltage regulators but these would be very inefficient; less than 20%. Alternatively, DC/DC converters could be used which would be better than 90% efficient but be more expensive and require significantly more design work. The decision was made to use a DC/DC converter (a Linear part) for digital voltages and "Low DropOut" (LDO) linear regulators for the lower and more critical voltages. This approach has worked extremely well and is highly recommended. The LDO linear regulators are very small and can be placed immediately beside the component using the power.

Another secondary decision was how much non-volatile memory to provide. This memory would be used to store calibration parameters and perhaps short CW messages. Most of the Microchip PIC processors provide some small amount of non-volatile memory. Unfortunately, Microchip decided not to include any non-volatile memory with the particular processor chosen. Consequently, it was necessary to include a small flash memory; "small" meaning 8 pins and 256k bytes.

A third decision was what kind of display to provide. One my main complaints with most modest radio designs is the limited display capability. With this in mind a 16 × 4 alphanumeric character LCD was chosen. Should this prove overkill, future implementations could use a smaller display. As an interesting aside, this display consumes considerable power and some form of power management will ultimately be necessary.

Another substantial decision to be made was choosing the input method for controls. There are several choices, buttons, knobs, keypads, etc. The first attempt at providing a user interface was a failure; this is not the place to be frugal. In the present design there are four knobs, one for each line of text. There is also a 12 key keypad for direct entry of parameters such as frequency. Even this proves to be inadequate and more knobs will probably be introduced in any future design.

### The Digital Circuit Design

Having chosen the processor, the CODEC, the DDS, the display, the switches, knobs and oscillator; the time had come to do the detailed design, using the basic block diagram of Figure 2. The block diagram demonstrates the simplicity of the digital piece of this SDR: there are really only three significant digital-oriented blocks: the DDS, the CODEC and the dsPIC.

These three blocks are connected by relatively few signals: the I2S and the I2C busses connect the dsPIC and CODEC, and a simple two-wire interface connects the dsPIC to the DDS. The I2S (which stands for Intra-IC Sound) conveys the 16 bit A/D and D/A information. The CODEC is the master of this interface because it provides the clock and framing signal. Two samples in each direction are conveyed at up to 48 kHz when the oscillator runs at 26 MHz. Thus the SPI clock itself runs at roughly 1.5 Mbits/second. However, a 25 MHz oscillator was available from previous projects and substituted for the specified 26 MHz. Other than the attendant reduction in sample rate, no adverse affects have been discovered.

The I2C (which stands for Inter Integrated Circuit) is a low speed bus which runs at several hundred kilobits/second. It is used to initialize the CODEC and to control the gain of the various CODEC amplifiers. In addition, the I2C bus is used to communicate with the flash memory chip to store and retrieve configuration data.

The display module is a Hantronix HDM16416L, LCD display. The connection to this part is directly from the manual. Not shown is a simple transistor switch which controls the LED backlight of the display. This LED is extremely power hungry and an operator specified timer is used to disable the LED a short time after any display update.

The keypad and knobs are the usual culprits. A background software routine polls for switch closures. Turning a knob causes events to occur so fast that interrupts are necessary to reliably track knob turns. The dsPIC provides hardware for generating "interrupt on change" and the knob signals were run directly to dsPIC pins with no external logic required.

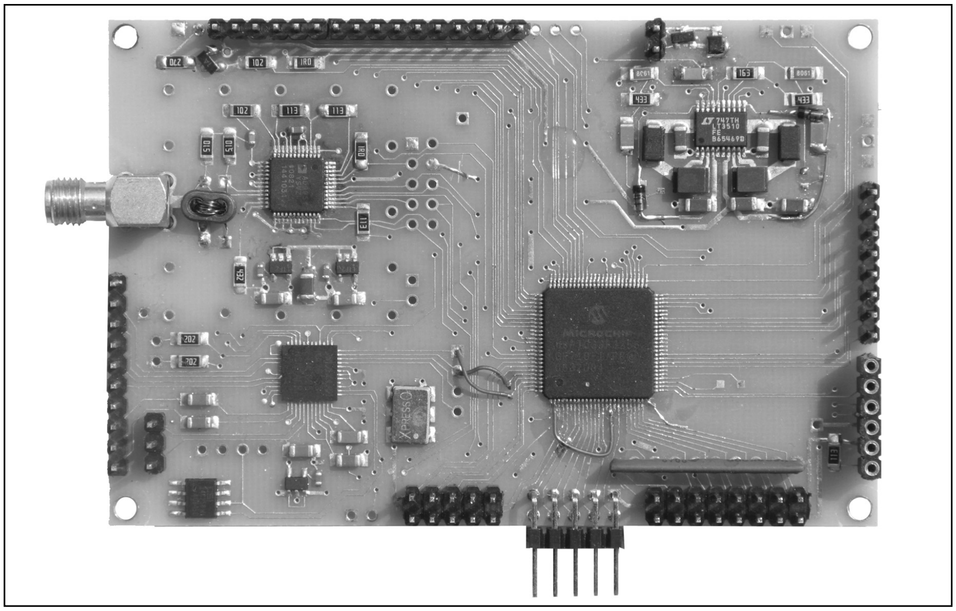

**Figure 7—SDR Processor Board at completion.**

In reality, the digital hardware portion of this SDR project is extremely straight forward and represents no significant challenges. There are no particularly high speed signals and the dsPIC and CODEC follow the standards well. The design really was simplicity itself. Hooking up the various parts as specified by the manufacturers delivered a solid and functional subsystem which operated exactly as expected.

### Digital Circuit Layout

The layout of the SDR processor board pushed the limits of both designer and tool. The layout went through several iterations, during which the best pinouts of the dsPIC were chosen and the best connector layout was reached.

Once the final placement of the ICs and connectors was chosen, the layout of the signal etches was a long but straightforward effort. The layout of the SDR board took several nights working into the wee hours of the morning. After verifying the pinouts and IC footprints, the SDR digital board was ordered.

As always, PCB fabrication was flawless. Assembly of the board was time consuming but uneventful. Power was applied and a few design errors were discovered; notice the wires on the board in Figure 7. All in all, careful layout and almost obsessive checking of every detail yielded a workable board the first time out.

### Remaining Development

At this point, the entire hardware portion of my SDR has been discussed with the exception of the band-specific boards. These boards each contain a low noise preamplifier and a 5 watt transmitting amplifier tuned to the band in question. This amplifier operates in Class E and will be considered in a separate part of this article series to allow more discussion of design principles and methods.

The other remaining development is, of course, the software used in this project. Since it is the heart of an SDR, it too deserves a separate article of its own. This article will discuss the methods of software development used as well as the actual software itself and the functions it provides.

Watch for these two articles in future editions of QRP Quarterly.

### Personal Retrospection

At the end of so large an effort it is probably a good idea to sit back and think about what might have been done differently. The word "MIGHT" is important here. The whole project was conceived as a learning vehicle and, as such, it succeeded beyond my expectations. In fact, this transceiver has become my main rig and is used on a daily basis. Still, here are a few ideas:

First, while the dsPIC is a perfectly fine processor, a true DSP from TI or Analog

Devices might be a better choice. These manufacturers continue to advance their offerings while Microchip has largely abandoned the dsPIC line. Further, a floating point DSP might be a good idea as well. Alternatively, the design of the SDR board could be entirely avoided: there are now small DSP evaluation boards which are very inexpensive and quite functional. Connecting one of these boards to a complete user interface may be problematic.

Second, the decision to design the down converter could have been avoided and a commercial version used. For example, a SoftRock40 or SoftRock20 could have been substituted.

Third, a DDS may not be the easiest way to provide the VFO functionality. A new technology called a "silicon oscillator" is becoming mainstream. A good example is the SI570 from Silicon Labs. This part is attractive because it is easier to mount than most DDS chips and provides a square wave output which can be connected directly to logic. Further, unlike most DDS ICs, the silicon oscillators do not need an anti-aliasing output filter.

Enough second guessing. Here are a few important lessons learned:

*First and foremost*—the user interface is NOT incidental to the design of a radio. Yes, there are countless rigs which have only a few knobs and no readouts. While these rigs can be effective and fun, this author finds them to be frustrating when used for long periods. Indeed, more knobs are better and more display is better. Any future effort will certainly increase the number of knobs and provide a larger keypad.

*Second*—one should use components which have solid documentation and IF AT ALL POSSIBLE have an evaluation board with a published schematic. There's no reason to purchase the evaluation board but there is good reason to believe the schematic will work if the manufacturer offers an evaluation board.

*Third*—local power regulation is a small expense with huge payoff. Small DC/DC converters work well and are all but required by the low voltages of modern, large scale ICs. Analog power supplies should use small linear regulators placed immediately beside the ICs using them. Don't share analog supplies between devices. ALWAYS POLARITY-PROTECT EVERY BOARD'S POWER IN….. ALWAYS! A 10 cent diode can prevent hours of rework. A couple seconds of thoughtlessness can be very costly.

*Fourth*—most manufacturers provide samples; one or two are usually free. Even relatively large and expensive ICs are often sampled. All the voltage regulators used, the CODEC and the op amps were all samples in this project.

Also, however tempted, don't crowd things. Packing things close together complicates every aspect of the project. Leave size optimizations for rev 2 or even 3!

Oh, get comfortable with surface mount parts. Nearly every modern IC is surface mount and relying on old through-hole parts is too restrictive. Never fear, if you can see the pins, even with jeweler's glasses, you can solder it down.

**Final Note**

Here is the most important lesson learned: If you have a project in mind, don't shy away from it. It is a timeless cliché but it is true, nothing teaches like doing! ●●