SDR Cube
A Portable Software Defined Radio Utilizing An Embedded DSP Engine for Quadrature Sampling Transceivers

By George L. Heron N2APB and Juha Niinikoski, OH2NLT

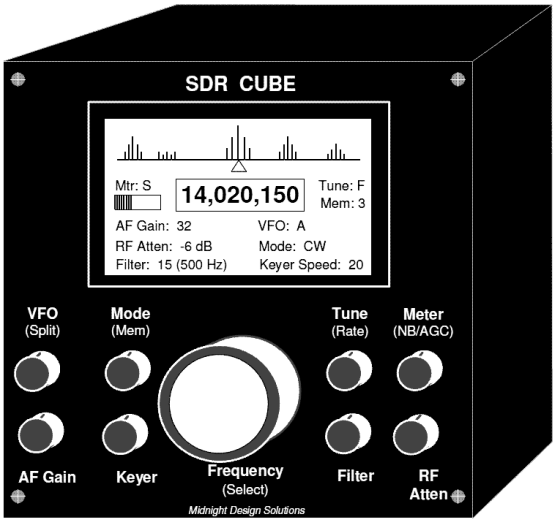


Figure 1: Graphic depiction of the completed SDR Cube Transceiver. Dimensioned at 4" x 4" x 4", design contains full user interface, DSP processing, I/O connectors on the rear panel, and 27 in^3 shielded space internally for Softrock.

Photo 1: Cube prototype prior to application of black powder coating and white silkscreen labeling. Designed as initial companion to NUE-PSK digital modem, future growth will encompass modem functions and more.

1 Introduction

Software Defined Radio (SDR) in the Amateur Radio community has been making great strides in recent years. From the innovative and ground-breaking products of Gerald Youngblood, KSDR of Flex Radio some four years ago, to the most recent state-of-the-art designs in the HPSDR group, SDR technology has really now come of age. This current state-of-the-art has also been greatly enabled by the tireless work of Tony Parks, KB9YIG, the father of the Softrock designs, has enabled many thousands of hams worldwide with his series of inexpensive radios that work with a PC sound cards.

Each of these radios in their most basic form consists of electronics that sample the incoming RF after it has been converted down to baseband and send the results to a PC for digital conversion. The PC then performs the complex demodulation computations so we operators can understand the SSB, AM or digital mode communications coming in. And of course the reverse happens for transmit, whereby the PC presents electronic signals to the front end electronics for mixing and ultimate transmission.

However throughout all the excitement of PC-based software defined radio, there has also been a quieter background quest for a form of SDR that is not tethered to a PC. While the PC offers seemingly unlimited processing power, gorgeous user interfaces and lots of memory, the PC is still an expensive and cumbersome accessory to take to the field when the need arises for portable operations. Classic problems are encountered with regard to the ability to see the PC display in bright sunlight, and

powering the PC and the power-hungry SDR front-end electronics is tough with limited-capacity batteries. In general, lugging around an expensive and delicate laptop is not something one wants to regularly do. Also, many hams just do not care to operate a ham radio with a mouse controlling knobs shown on a on a PC screen.

However most would still agree that the *performance* of such a PC-based radio is of great value, given the flexibility offered with SDR's innate ability to handle virtually any operating mode – SSB, AM, CW, and digital modes – with just a new software file or program loaded into the radio.

Numerous experimenters have been seeking to develop a portable SDR transceiver to address the shortcomings of using a PC in the field, while retaining enough of the benefits of SDR in general so as to have an inexpensive-yet-powerful transceiver. Embedded SDR projects are in progress with in the High Performance SDR Group (HPSDR) and yet another is being described in an article series in QRP Quarterly Magazine. Many of us experimenters are standing on the shoulders of previous pioneers in the field of DSP for Amateur Radio: Rob Frohne KL7NA, Lyle Johnson KK7P, Rick Campbell KK7B and others. The handbooks from the ARRL and RSGB, as well as the seminal work of "Experimental Methods for RF Design" by Hayward/Campbell/Larkin, are replete with the basic building blocks that we designers of today are employing in our SDR implementations.

This paper chronicles and describes the path that our team has been taking, resulting in what we call the "SDR Cube".

2 Brief History of the Design

Before getting into a full design break down, we wanted to provide a brief background how we came to this point.

The genesis of this embedded DSP approach to a Software Designed radio has its root in Finland, at the shack of OH2NLT (co-author of this paper). Over the course of four years, Juha Niinikoski had been experimenting with many of those basic building blocks mentioned earlier in order to show how a "cheap DSP" solution can be used to create a ham radio. In fact, using that very name for his progression of early designs with Microchip's dsPIC processors, Juha was able to demonstrate a minimalist working radio based on a dsPIC30F processor to control the phasing of two quadrature-related sampled data streams.

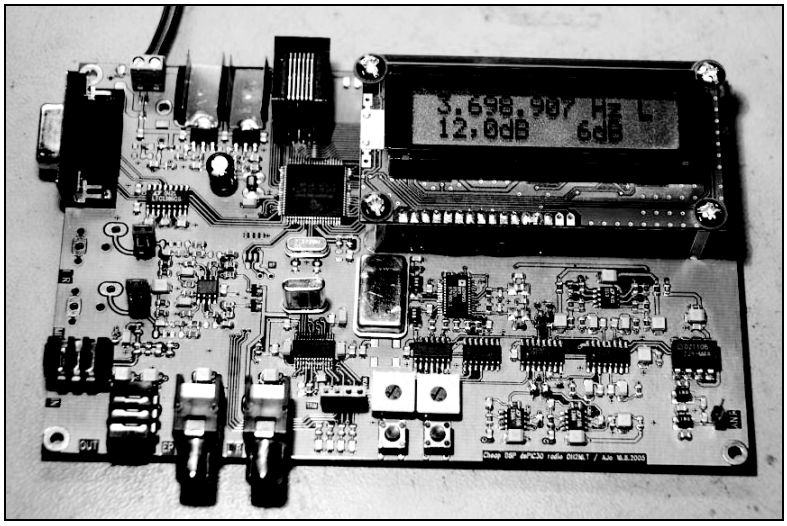


Photo 2: First prototype
A single board controller with integrated transceiver.

The CheapDSP results were very encouraging and early in 2009, N2APB (the other author of this paper) learned of Juha's work and it became something that had to be duplicated in his own workshop in Maryland. Heron had always been focused on providing portable radio solutions, as evidenced by his recent co-design of the NUE-PSK Digital Modem, which also used the dsPIC processor.

In the course of replicating the OH2NLT CheapDSP controller, George and Juha began collaborating on an evolved and more complete SDR implementation, this time using the more capable dsPIC33F processor at the heart of the design. Using this Microchip controller we would have greater processing power, more input/output pins for an integrated and capable user interface, and we would be able to inherit some of the features and drivers that made the NUE-PSK modem so popular – namely, the spectrum display and FFT processing, and the drivers for the rotary encoder, field programming and bootloading, and the USB card interface.

Together, N2APB and OH2NLT embarked on an enhancement campaign, working "transoceanic" and across many time zones to ultimately create the encompassing SDR Cube design.

Photo 3: Expanding "CheapDSP" to Cube Capabilities
Prototype hardware on bench being modified as design evolves to the "SDR Cube".

3 SDR Cube Overview

The SDR Cube is a single-band QRP transceiver consisting of an embedded DSP controller coupled with the Softrock RxTx v6.3. Sized at 4" x 4" x 4", the appropriately-named Cube also contains a full complement of built-in user interface: graphic LCD for spectrum display, typical controls for frequency, mode and signal management, and I/O connectors for connection to the outside world. The Cube design is optimized to internally accommodate the popular Softrock RXTX v6.3 electronics. Different from other experimenter "single board" solutions, the Cube was designed from the start to provide a full transceiver with minimal component hassle imposed on the builder.

3.1 Software Discussion

The design is the classic "phasing method" of SSB generation, whereby a 90-degree delay is imposed on the source audio signal coming from the microphone, or from a NUE-PSK Digital Modem. The resultant signals are applied to a pair of mixers – in our case these are provided on the Softrock board – with 0

and 90-degree LO signals. The output of the mixer is amplified on the Softrock and presented to the output BNC connector on the rear of the Cube.

The receive path works in a similar manner, but in reverse. The received signal is mixed with LO signals on the Softrock card that are at 0- and 90-degrees, producing the I and Q quadrature signals. These component signals are digitized by the Cube hardware and the dsPIC controller imposes a 90-degree phase shift between them by passing them through two balanced and matched FIR filters, one advancing the signal 45 degrees and the other delaying the signal by 45 degrees. Maintaining accurate phase delays and corresponding amplitudes have a direct impact on the quality of the demodulated signal. The demodulation occurs at the summing junction, shown in Figure 2 below; when the dsPIC performs the computations for the desired mode, the receive audio is able to be heard in the headphones.

Note that either USB or LSB may be demodulated merely by reversing the I and Q signals where indicated in Figure 2.

Recalling the computation for AM demodulation: AM = SQRT (I^2 + Q^2), even this mode is able to be accomplished, which is a feature of all SDRs.

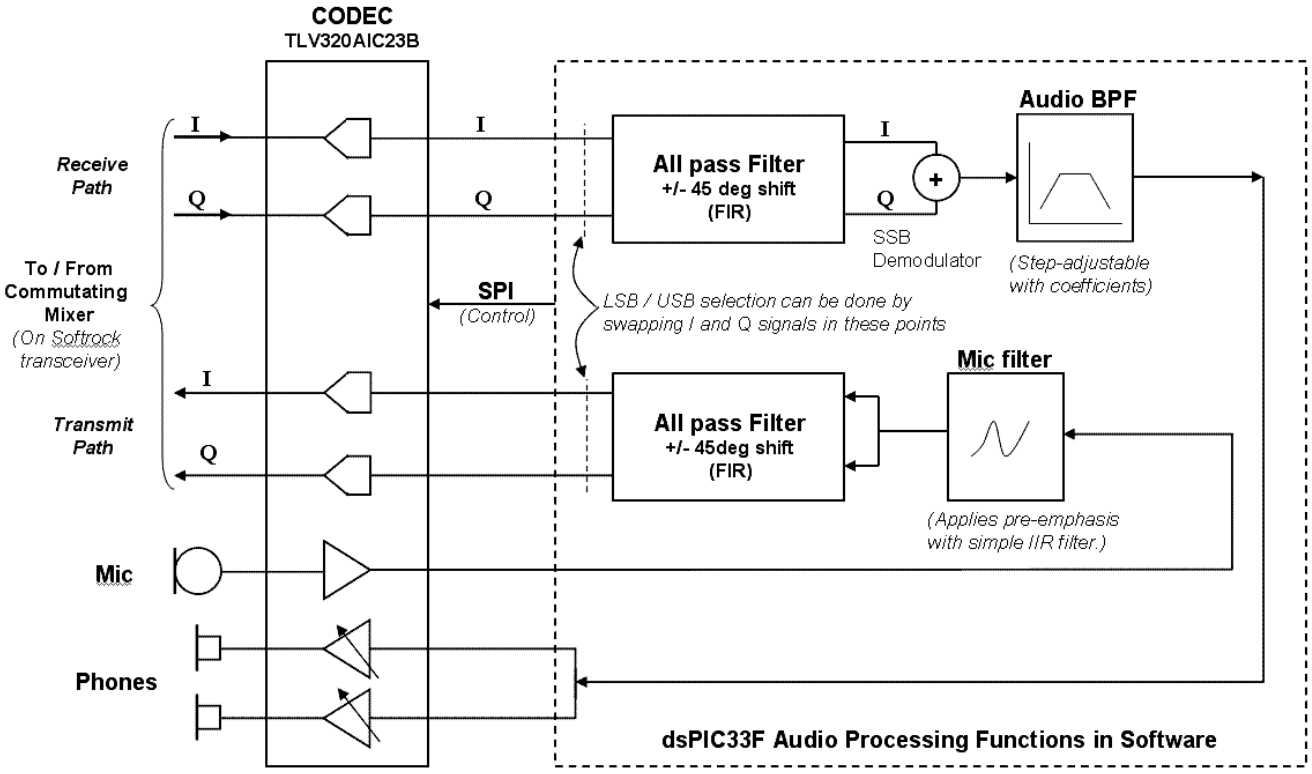


Figure 2: Software Processing Block Diagram

3.2 Hardware Discussion

Turning next to the hardware block diagram shown below in Figure 3 we see the dsPIC33F at the heart of the design, taking inputs from many user controls, displaying text and graphics to a 128 x 64 pixel graphic LCD, and implementing the audio data paths through the codec and over to the Softrock board on the right.

The dsPIC33F is a 40 MIPS controller (40 million instructions per second), which comparatively is only moderately capable with today's technology. But unlike some of the more powerful embedded DSP controllers from Texas Instruments or Freescale, the dsPIC package is relatively straightforward for homebrewers to attach on pc boards. Even with the 100-pin package of the dsPICmodel we use is able to be attached in the conventional "flood solder and wick off" technique. Microchip offers a free, very capable and intuitive development environment called MPLAB, enabling experimenters to focus on the design and not the tool. For these reasons, the dsPIC controller remains popular in the experimenter community.

The dsPIC offers enough capabilities for designers to provide some demanding solutions: a 16-bit fixed point architecture, yielding 96 dB dynamic range in its computations. It has 40-bit accumulators and hardware support for division operations, barrel shifters, multipliers and a large array of 16-bit working registers – all of which provide ample DSP power for the operations performed in such an embedded SDR. Admittedly, the dsPIC controllers are getting "long in the tooth", however microchip is committed to continuing the product line, perhaps due to continuing applications like those here in the ham radio community.

For the all-important A/D and D/A conversion block we used the Texas Instruments TLV320AIC23B high performance stereo codec. In our first implementation, 8 kHz sampling is used, as anything much more than this is difficult for the dsPIC to handle, as a higher sampling rate has a two-fold performance cost: (1) Filtering must be performed more often (at the pace of the sample rate); and (2) More filter taps, resulting in more computation time, is required at a higher sample rate, which is a subtle and often-overlooked consideration in design. In later versions of Cube software we will be implementing additional processing hardware to accommodate increased computational power; for example providing a wider spectrum band scope and digital modem functionality. So keeping the computing overhead lower is just as important as achieving adequate quality. The integrated headphone amplifier, programmable gain microphone amplifier, and SPI control bus were important selection criteria for this codec.

Optimal gain distribution in the system is not necessary known yet, and AGC is an important (and difficult) feature to implement. Thus we have started getting our hands around receive path gain by implementing a controllable RF Attenuator as a plug-in for the Softrock board in place of its existing BFP board. In this way we can have several levels of signal attenuation to assist in establishing optimal signals throughout the system.

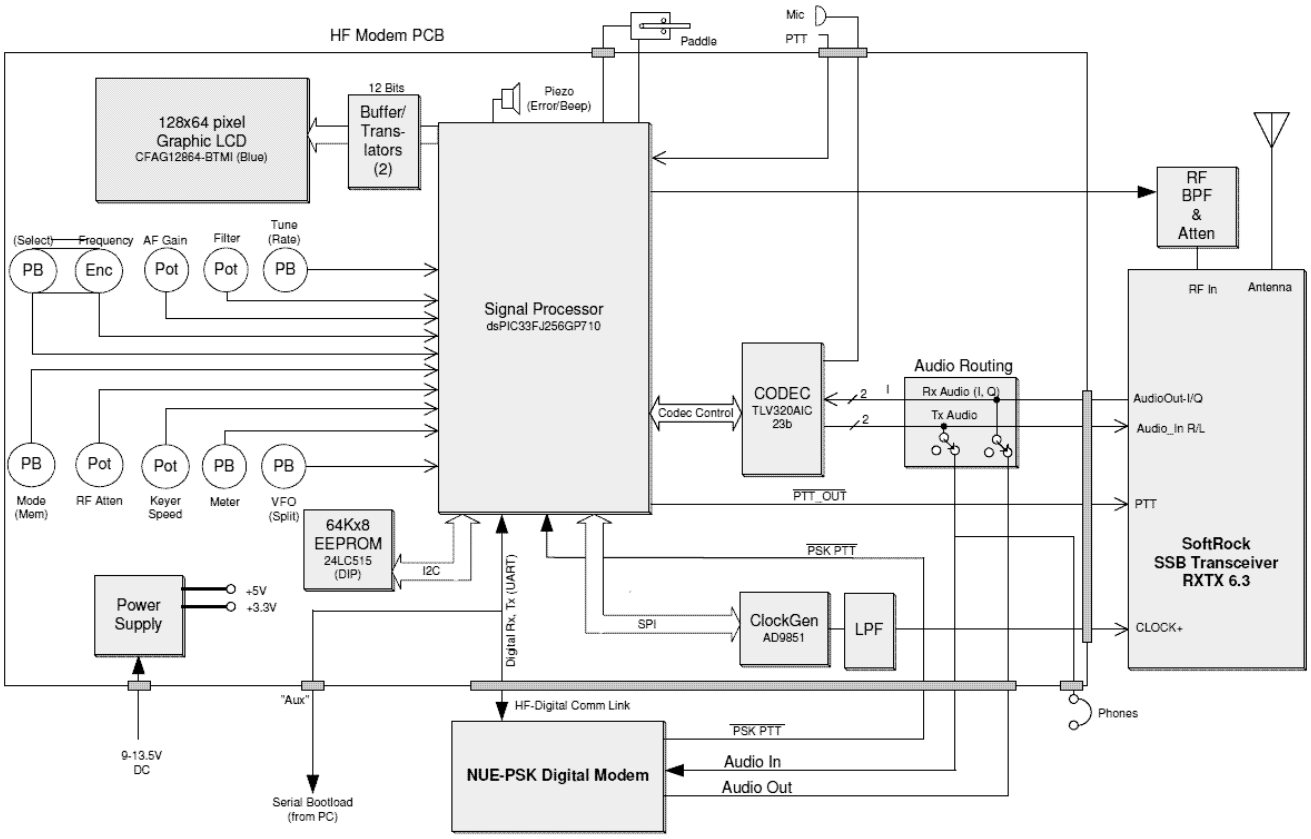


Figure 3: SDR Cube Hardware Architecture

Illustrates designed-in controls, graphic display, and tight integration with NUE-PSK Digital Modem

Clock generation is optionally provided for on the DSP board. In fact, because of the difficulty that homebrewers have in obtaining the ideal clock generator (Si570), we have also provided for use of the classical DDS signal generator.

Special connection to the NUE-PSK Digital Modem is provided in order to optimally interoperate with the modem to provide digital mode capability for the SDR Cube.

The display system is a step up as compared to conventional minimalist SDR implementation that uses a character-based LCD. As illustrated in the Cube screen graphics we are able to provide a multitude of graphical representations of system status. Additionally, and most exciting is the ability planned for the Cube to display a spectrum band scope function that is a graphical representation of the 2.5 kHz swath of spectrum being processed. Admittedly, this close-in view of the spectrum is not as sexy as the 25 kHz spread offered by commercial rigs; but using only 8 kHz sampling presents limitations, and various tradeoffs are necessary in embedded solutions. This is an area we intend on improving in future iterations of the design – either by improving the processing software or by substituting a more powerful DSP in place of the dsPIC.

4 Features, Goals & Specifications

This section lists the features of the SDR Cube in a rather list-oriented manner, primarily for the sake of brevity. It was important for us to understand the natural constraints offered in embedded systems with limited resources and computing power. The designer's goal is always to provide "just enough" without over-taxing the available resources; or conversely without creating a radio with more bells and whistles than are actually warranted.

4.1 Portable, standalone, QRP-level, all-mode SDR transceiver for Amateur Radio

a. Portable: 12-volt battery operated, easily transportable, hand-held form factor for convenient field use *(EmmCom, Field Day, Trail usage, etc.)*

b. Standalone: Design uses embedded microcontroller for all signal processing - no PC or Laptop required. *(Decouples the product from PC complexities, cost & usage concerns.)*

c. Band Coverage: 160-10 meters (1.8-30 MHz). *(Base design provides for 20m BPF/Output modules, with other plug-ins accommodated.)*

d. Low Power: Low current draw from power source, approx. < 500 ma. *(Maintains battery life in field use.)*

e. QRP: RF transmissions < 5 Watts, typical. *(QRP output levels are achievable in small form factor. Can later add options for power amp.)*

f. Modes: Voice, CW, and select Digital modes. Digital modes achieved by interoperation with NUE-PSK Digital Modem. *(These modes cover the wide range of anticipated user needs: bench/field use, casual use, EmComm use, etc.)*

4.2 Built-in Transceiver "RF front end"

a. QSE/QSD-based quadrature signals provided to HF modem for all-mode modulation/demodulation of signals. *(Easiest , least expensive and most convenient architecture for implementing SDR.)*

b. Softrock RxTx 6.3 transceiver assumed in base design of enclosure. *(Best performing and most compact Softrock transceiver. 50 kits already stocked for this SDR use.*

c. Other Softrock models or other QSD-based transceivers able to be plugged into core signal processing of the HF modem. *(Leverage the > 10,000 Softrocks already in the field, lowers the user's cost of ownership by optionally selling the built-in RxTx)*

4.3 HF Modem

a. Microchip dsPIC33FJ used as the primary embedded signal processor performing the HF modem functions *(Software architecture available, easy to port in critical functions from digital modem: display, USB, bootloader, keyboard)*

b. TLV320AIC23B codec used as the multi-channel, gain-controlled analog-digital-analog signal conversion *(Driver available, sufficient bits & sampling)*

4.4 Physical User Interface (Display & Controls)

a. AF gain: potentiometer *(Adjust of audio volume to headphones and digital modem)*

b. RF Gain: potentiometer *(Adjustment of incoming RF levels to help with difficult ALC operation)*

c. Frequency: rotary encoder with integrated Select pushbutton

d. Tune/Rate: pushbutton *(Press-Hold turns on constant tone for transceiver/antenna tuning. Tap cycles through fast/med/slow tuning rates.)*

e. Keyer Speed: potentiometer *(Controls speed of internal electronic keyer.)*

f. Mode: pushbutton *(Tap selects the operating mode: USB, LSB, Digital, CW, AM. Press-Hold to select Memory.)*

g. Filter Select: potentiometer *(Selects desired DSP filter for audio output.)*

h. Power On/Off: recessed slide switch *(Protects unit from accidental turn-on while being transported)*

i. Display: graphic LCD, 128x64 PELs *(Displays indicators for frequency, mode, keyer speed, memory meter and spectrum)*

j. Beeper: piezo sounder *(Internal annunciator for user interface feedback (clicks) and error indicator.)*

4.5 Connectors

a. Power: 2.1mm coaxial DC jack *(Most common in the QRPer's bench and backpack for field use)*

b. Antenna: BNC *(Most common for QRP equipment, better/easier RF connection than RCA jacks or SO-239)*

c. Mic: 3.5mm stereo jack for microphone and PTT input. *(Most common on bench and in backpack)*

d. Phones: 3.5mm stereo jack. *(Most common on bench and in backpack)*

e: Paddle: 3.5mm stereo jack *(Most common on bench and in backpack)*

f. NUE-PSK Interface: 8-pin miniDIN *(Custom connect to NUE-PSK digital modem for optimized performance.)*

g. Aux: 3.5mm stereo jack. *(Convenient serial connection to PC for bootloading, external control, etc.)*

4.6 Form factor

a. Size: Small, driven by graphic LCD and controls, also including Softrock. *(Facilitates convenient and integrated portable operation.)*

b. Material: Aluminum. *(Needed for RFI shielding)*

4.7 Software Field Upgradeability

a. Integrated bootloader (*Allows user to download improved software versions from the website and load into SDR.)*

5 Mechanical Construction

The SDR Cube design is embodied across three printed circuit boards, as depicted in Figure 3 on the next page …

- **Controls Board** – Oriented as the "front panel" and containing all the user controls, graphics display, and clock generation;
- **DSP Board** – Plugs into the end of the Controls Board at a right angle and sits front-to-back along the left side of the enclosure; and
- **IO Board** – Plugs into the DSP card and contains most of the electrical connections to the outside world.

The custom-designed 4” x 4” x 4” aluminum enclosure is designed in a clamshell arrangement to facilitate easy access to the inner workings of the Cube during assembly, troubleshooting or modification. The final enclosure will be black powder coat finished with white silkscreen labels.

An inner L-shaped aluminum bracket, is provided in order to shield the digital environment of the DSP and front panel from the RF compartment in which the Softrock resides.

The RF compartment is sized to ideally contain the Softrock RXTX V6.3, however other electronics (or an entirely different Softrock transceiver) may be fit into this space.

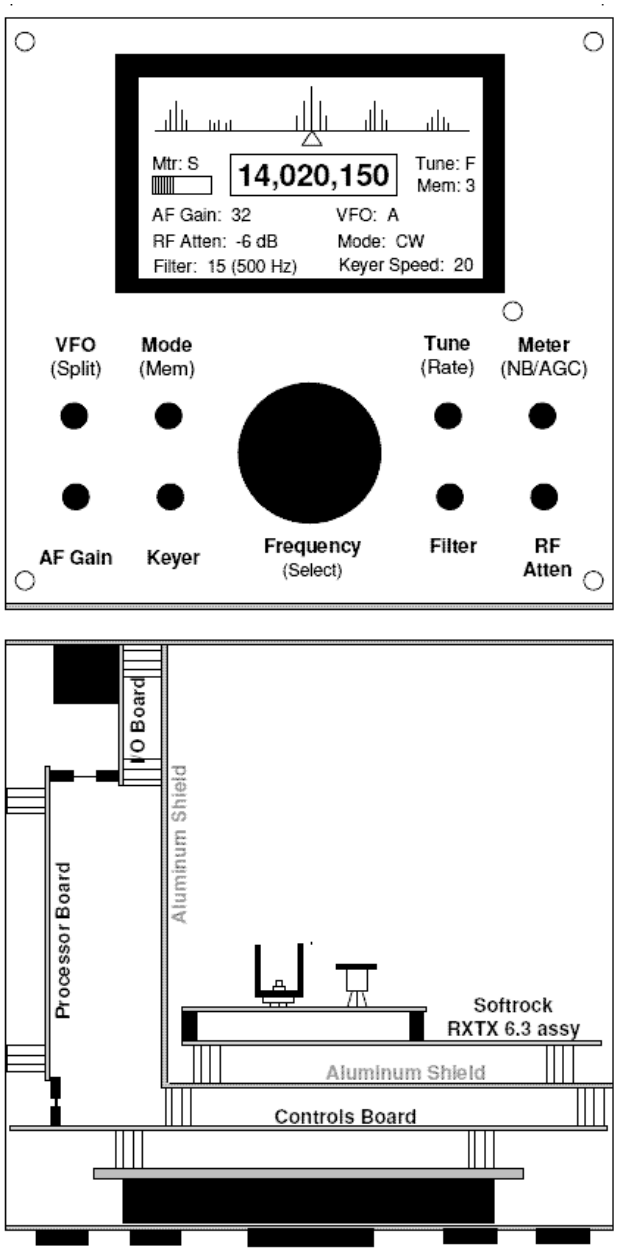


Figure 4: Front and Top view of the SDR Cube

6 Current State

At the time of this writing the prototype system is operational, yet we have not performed many comparative system performance measurements. We are currently working with a set of improved printed circuit cards (see Photo 4 below), and we are starting to see the final system configuration come together. However we still envision a significant number of software updates being needed as we continue to fine tune the user interface and its many control aspects.

By the time of the actual Conference, we bring along the prototype system for demonstration, and we fully expect to also have the "final" implementation of the hardware, running an optimized version of the Cube software.

Also in the timeframe of the Conference, we expect to update this paper with actual system performance results and further implementation detail. Printed versions of the updated paper will be provided to those in attendance of our presentation.

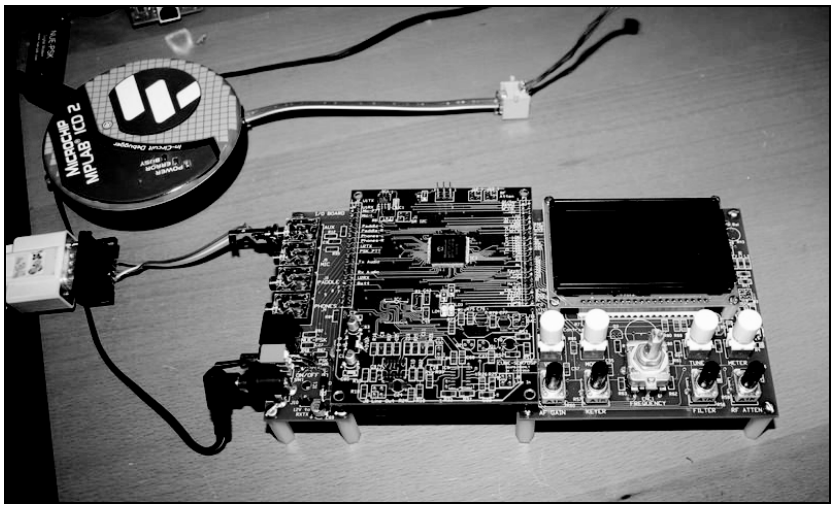


Photo 4: 3-Board Set Cube

Three-board SDR Cube boards connected horizontally as a development platform

Photo 5: 3-Board Set Cube

Custom Aluminum Enclosure (unpainted, before milling) and the three board set of SDR Cube

7 About The Designers

George Heron, N2APB, *has been a technology manager located in the northeastern US for more than three decades, working as a cyber security professional to help protect computer users from viruses and malware. First licensed in 1968 as WN2WVZ, George currently holds an Amateur Extra class license and is an avid homebrewer in RF and digital circuits, with a special interest in DSP and microcontroller applications to QR.P. He co-developed the NUE-PSK Digital Modem and the Micro908 Antenna Analyzer, leads the New Jersey QRP and the American QRP clubs, and has previously edited/published Homebrewer Magazine. George can be reached at 2419 Feather Mae Ct, Forest Hill, MD 21050, or at n2apb@verizon.net.*

Juha Niinikoski, OH2NLT *lives in Espoo, Finland (near Helsinki) with his wife and two sons – one son is also a ham! Juha is a co-owner of a small hardware and software design house and is a co-developer of the Finnish JUMA kit line of ham radio products. Juha enjoys rag chewing and especially experimenting with new technologies and designing equipment. He received his first license in 1988 - a Technical class license which was later upgraded to General class. OH2NLT can be reached at Etuniementie 11 C, 02230 Espoo, Finland or at juha.niinikoski@sitecno.fi*

8 References

1. CheapDSP, Juha Niinikoski, OH2NLT,
http://www.kolumbus.fi/juha.niinikoski/Cheap_dsp/Cheap_dsp.htm

2. FlexRadio, Gerald Youngblood K5SDR, http://www.flex-radio.com/Default.aspx

3. NUE-PSK Digital Modem, George Heron, N2APB and Milt Cram, W8NUE, http://www.nue-psk.com/

4. "Experimental Methods in RF Design", Chapter 11, Wes Hayward, W7ZOI, Rick Campbell, KK7B, Bob Larkin, W7PUA, http://www.qrp.pops.net/emrfd.asp

5. The ARRL Handbook, American Radio Relay League, http://www.arrl.org/shop/The-ARRL-Handbook-for-Radio-Communications/

6. Various articles in DSP Processing, ARRL, http://www.arrl.org/dsp-digital-signal-processing

7. High Performance SDR (HPSDR), http://openhpsdr.org/

8. HPSDR Boards, TAPR, http://www.tapr.org/kits_vna.html

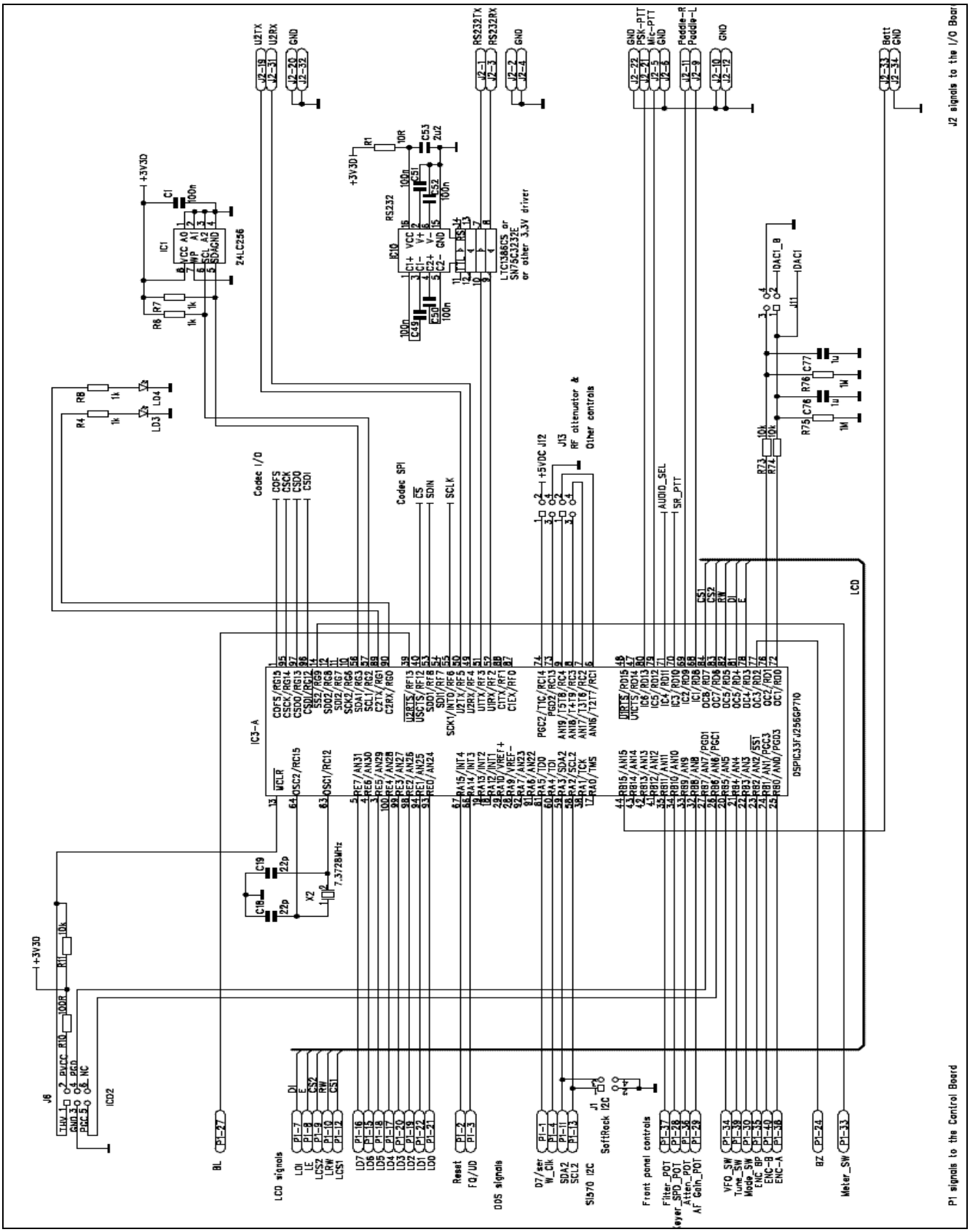
9. RSGB Radio Communication Handbook, Radio Society of Great Britain, http://www.vhfcomm.co.uk/rsgb-handbook.htm

10. Softrock Transceivers, Tony Parks, KB9YIG, http://www.kb9yig.com/

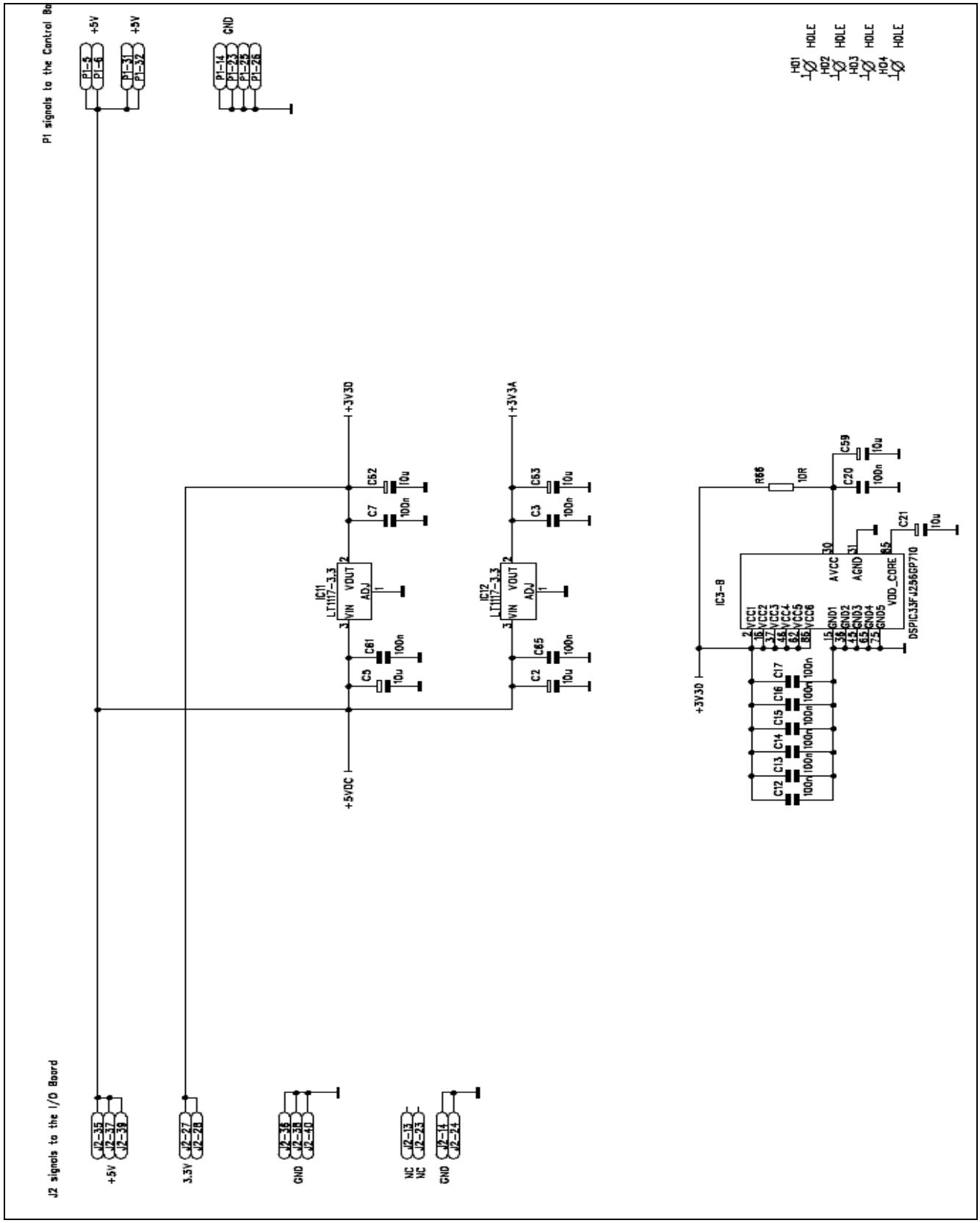
11. Softrock RXTX V6.3, http://www.wb5rvz.com/sdr/RXTX_V6_3/

9 Schematics

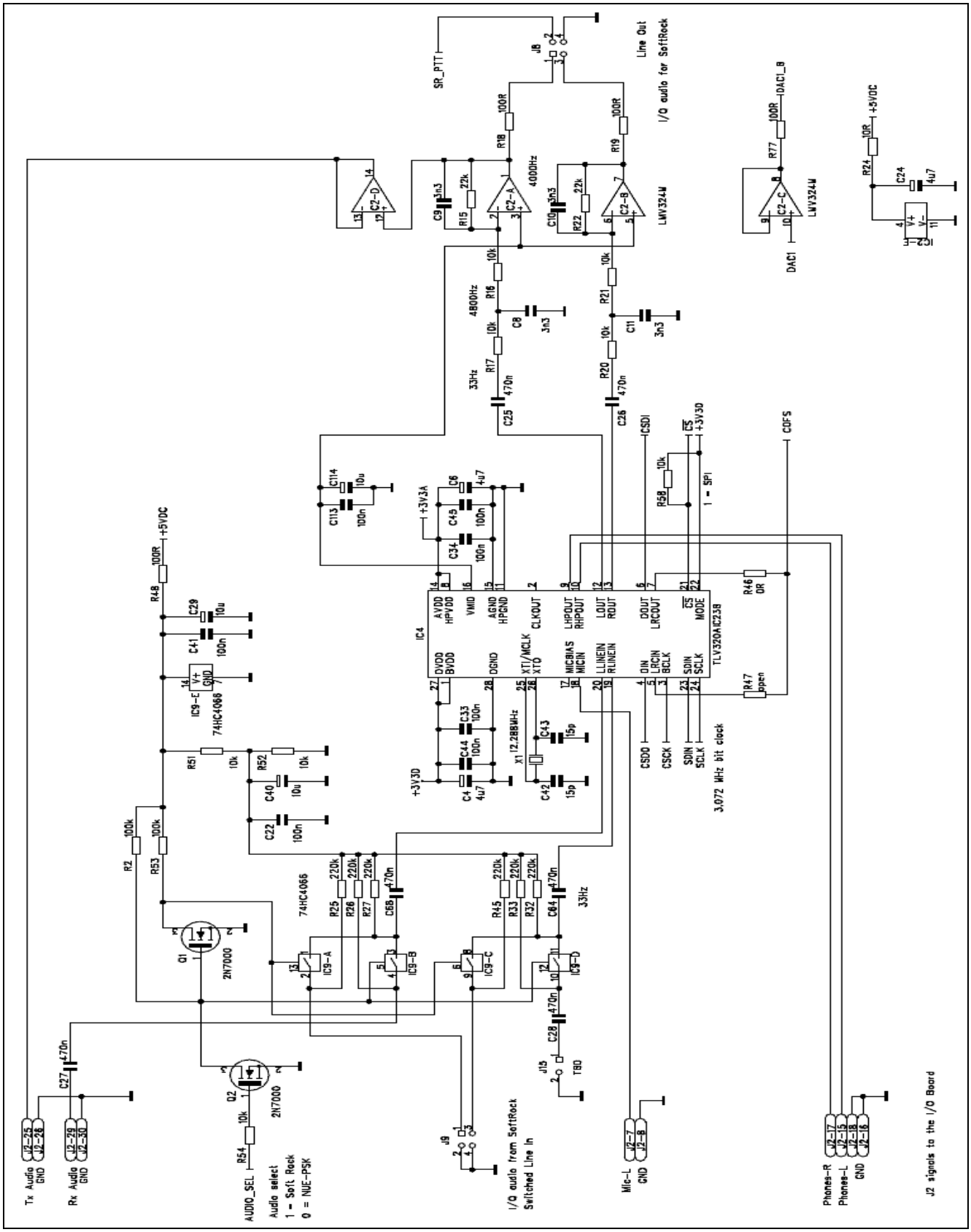
See following pages …



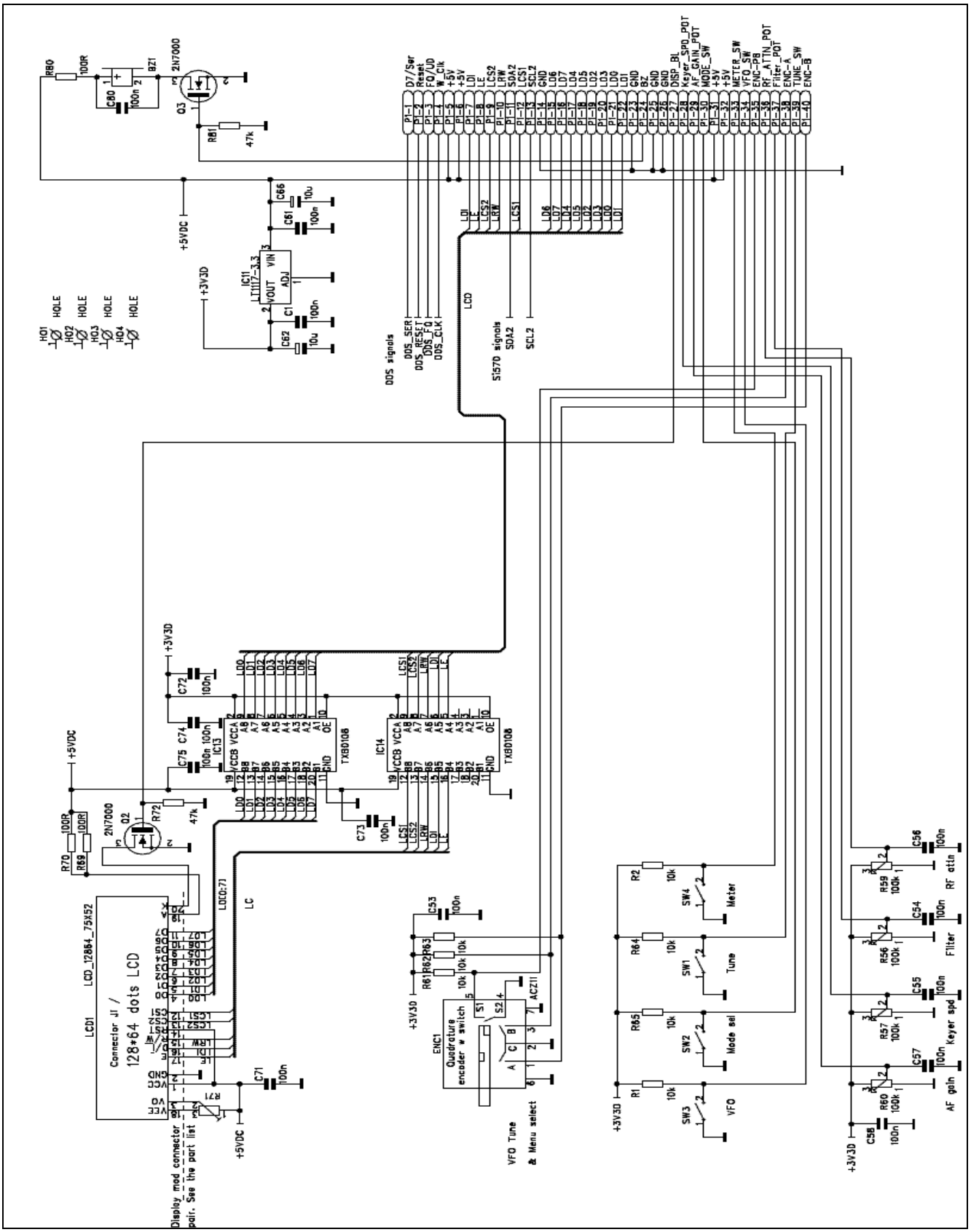
Schematic (Sheet 1/8): DSP Board-1 *(Copyright 2010 N2APB and OH2NLT)*



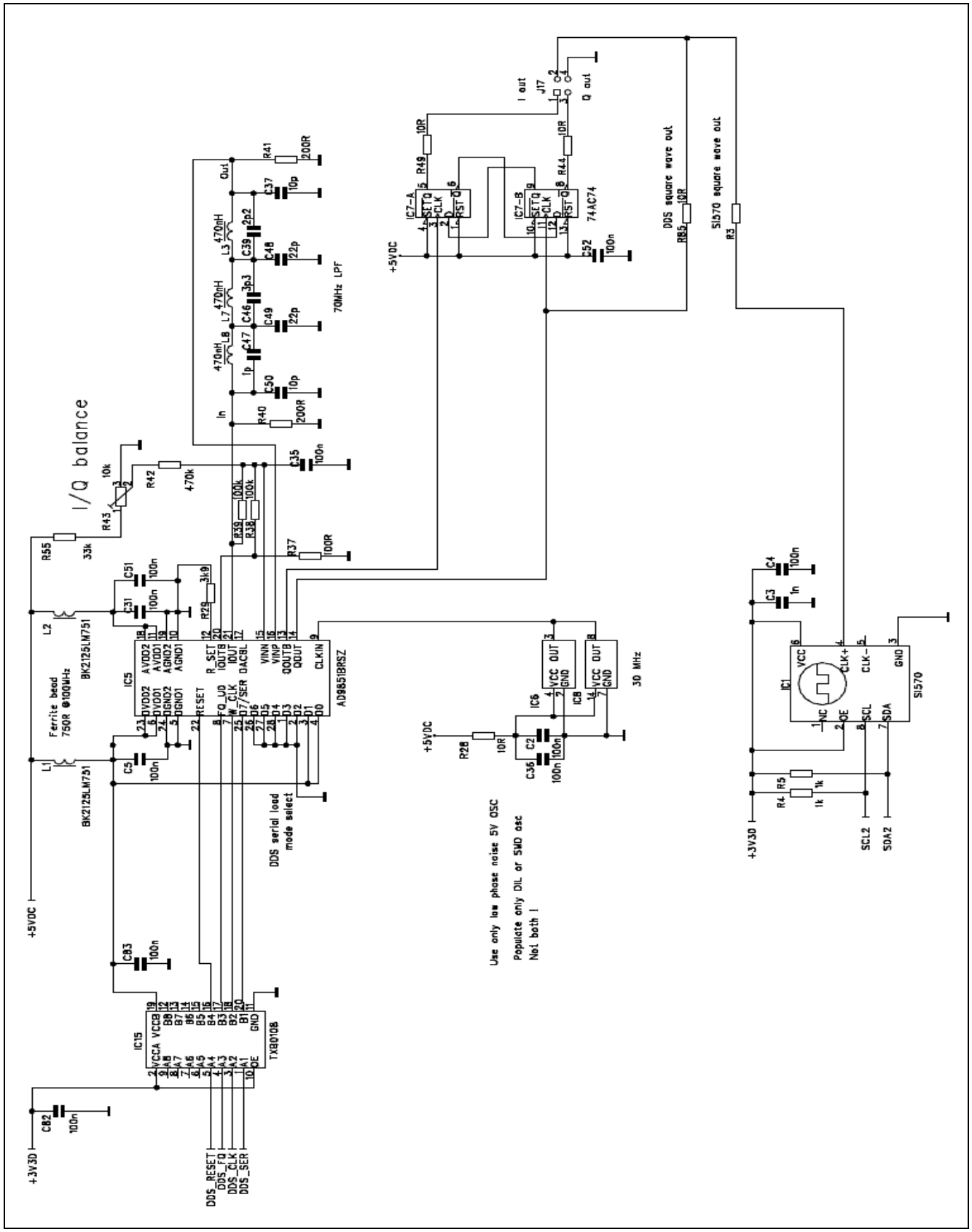
Schematic (Sheet 2/8): DSP Board-2 *(Copyright 2010 N2APB and OH2NLT)*



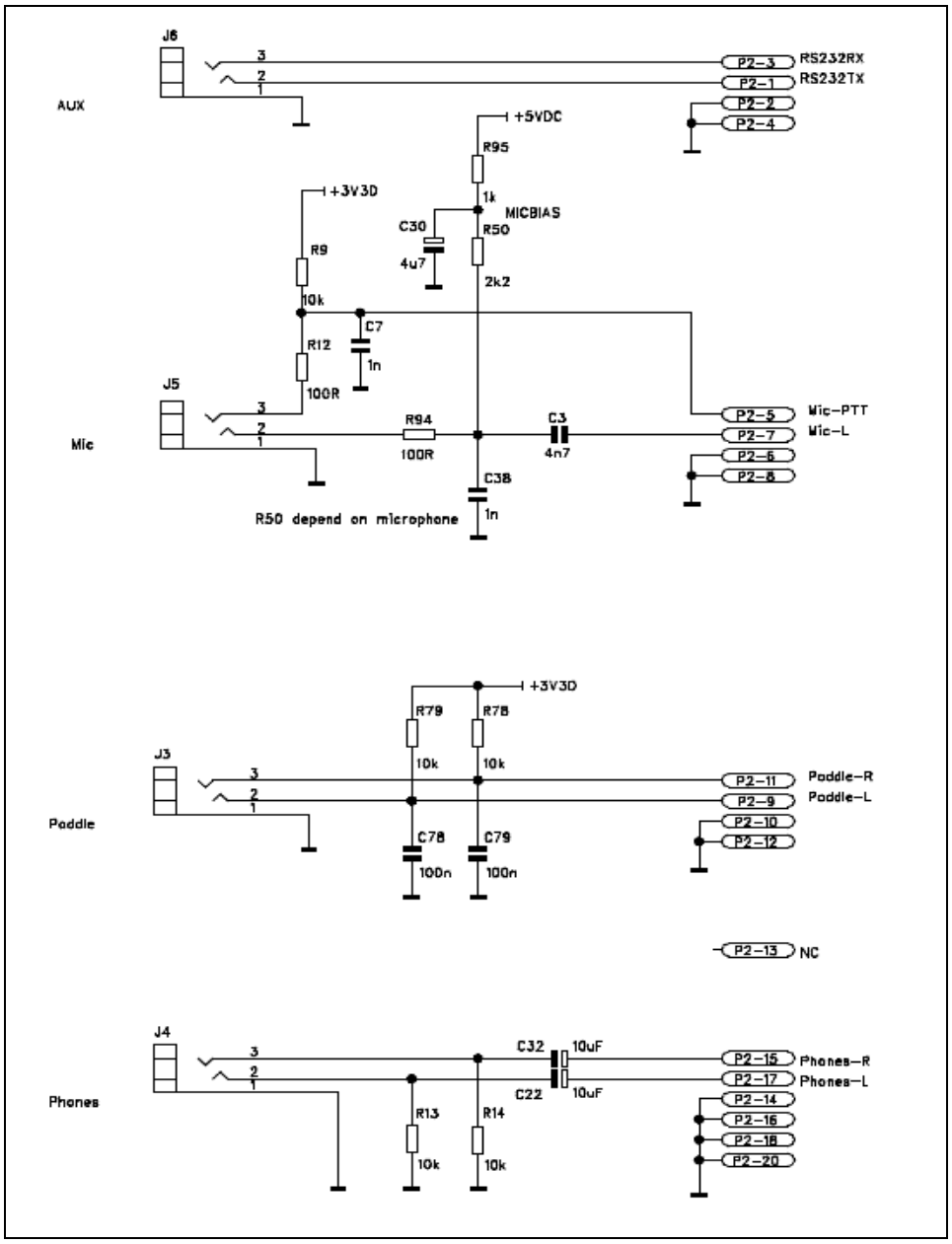
Schematic (Sheet 3/8): DSP Board-3 *(Copyright 2010 N2APB and OH2NLT)*



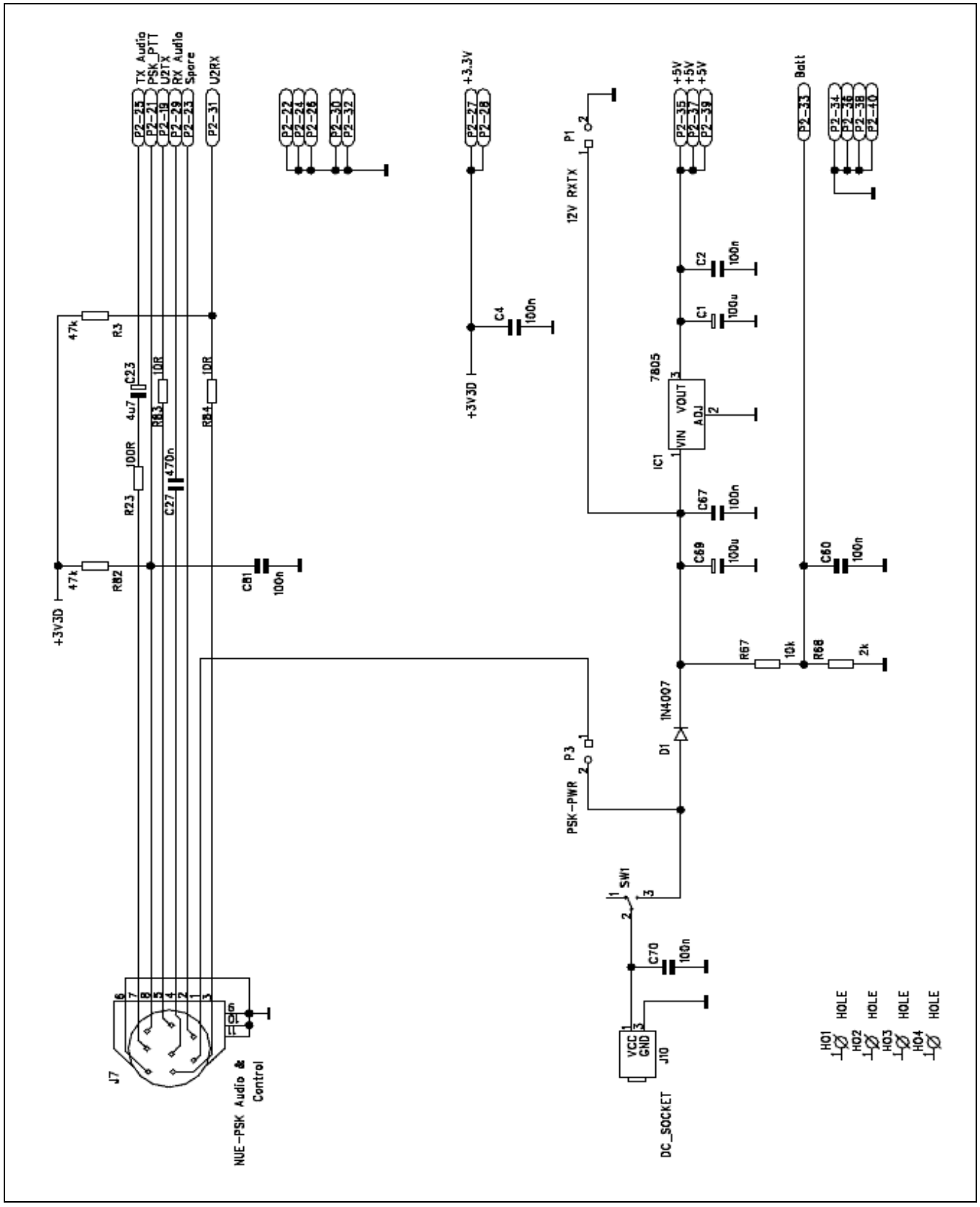
Schematic (Sheet 4/8): Controls Board-1 *(Copyright 2010 N2APB and OH2NLT)*



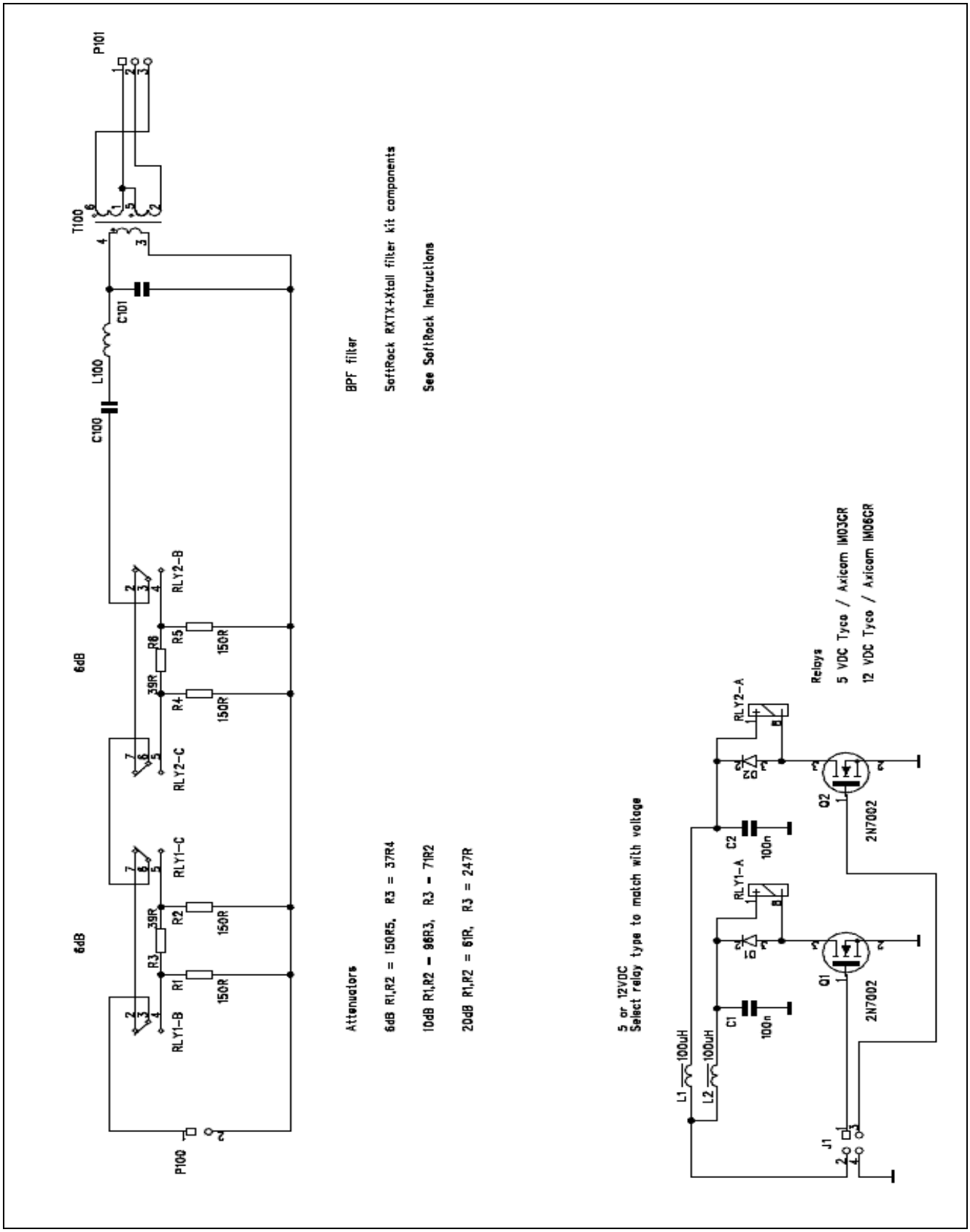
Schematic (Sheet 5/8): Controls Board-2 *(Copyright 2010 N2APB and OH2NLT)*



Schematic (Sheet 6/8): I/O Board-1 *(Copyright 2010 N2APB and OH2NLT)*



Schematic (Sheet 7/8): I/O Board-2 *(Copyright 2010 N2APB and OH2NLT)*



Schematic (Sheet 8/8): RF Attenuator Board *(Copyright 2010 N2APB and OH2NLT)*

A Simple SDR Receiver

Michael Hightower, KF6SJ
13620 White Rock Station Road
Poway, CA 92064
858-592-0555
mike@simplecircuits.com

Abstract:

This article discusses the design and operation of an HF radio receiver operating in the 3.5 to 18 MHz range. The receiver architecture is based on software defined radio techniques and incorporates a Cypress PSoC CY8C3866 component that contains both analog and digital circuits, thus decreasing the receiver's component count.

Key Words:

SDR, software defined radio, PSoC, programmable system-on-chip

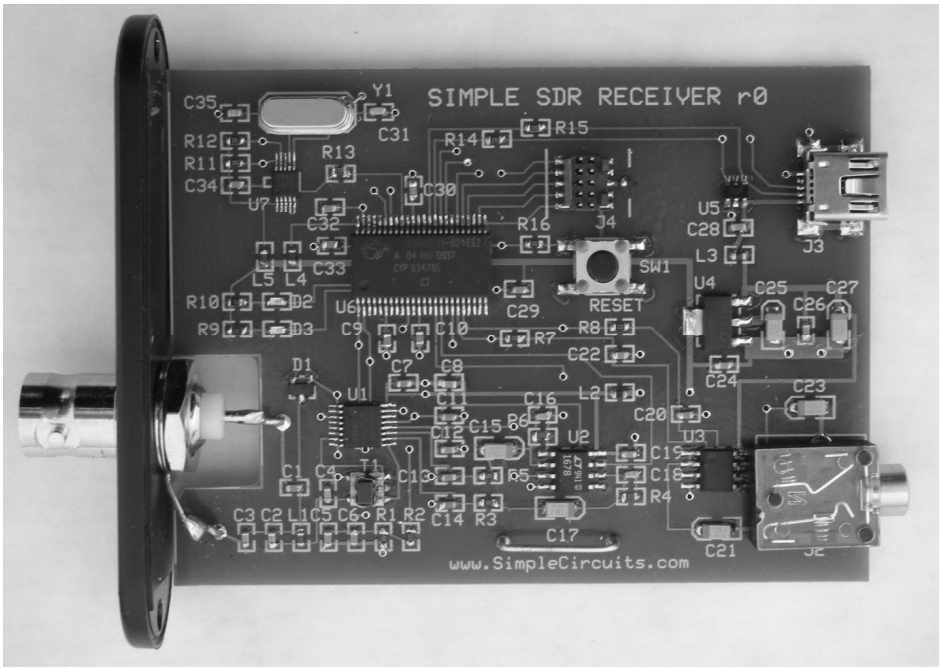


Figure 1: The complete receiver fits on a 3.2 by 2.3 inch printed circuit board.

Background:

Laziness can be an asset for an engineer. Minimizing the number of components in a design decreases the effort of board layout, parts procurement, board construction, and, most importantly, reduces the cost. Being a lazy engineer, I want my designs to have as few of parts as possible, striving for elegantly simple circuits.

During the last few years, I have built a few homebrew receivers using software defined radio (SDR) techniques. Each successive design is simpler, performs better, is less expensive, and uses few components than its predecessor. One of my favorite components to use is the Cypress Semiconductor PSoC series of components. These parts are far more than just a microcomputer; they also contain software configurable analog and digital peripherals on a single chip. Cypress calls the family a PSoC in reference to it being a programmable embedded system-on-chip. The newest series of parts, which Cypress calls the PSoC 3 family, contains a 67 MHz 8051 class microcomputer, an analog to digital converter fast enough and with enough resolution for an SDR receiver, and other valuable functions that are desirable in a receiver design. When I saw this part, I immediately saw its use in an HF receiver.

Receiver design goals:

My goal was to use the PSoC 3 component as the cornerstone of an SDR receiver design. The receiver should be used for casual, conversational listening, not a higher performance receiver for DX use. It should also be built with a minimum number of components, small, and easy to use. I considered adding an LCD display and controls to select the frequency and modes of operations, but decided that the design would have fewer parts and cost less if a PC is used for user input and output. Since the PSoC has a USB port, the receiver can connect to the PC with a USB cable and take power from the PC over the USB cable, saving a power jack, and external power source. Control of the receiver is accomplished by the receiver USB port appearing as a standard com port to the PC. The Ham Radio Deluxe (HRD) program works perfectly to control this receiver.

The basic SDR receiver:

A common SDR receiver is built using a quadrature sampling detector, as shown in the block diagram of figure 2. The quadrature sampling detector is nothing more than a set of analog switches that are enabled and disabled in the particular sequence that samples the input signal four times for each cycle of the desired receive frequency. The four samples represent the 0, 90, 180, and 270 degrees of a sine wave. The output of the detector is amplified by a pair of op amp low-pass filters. After the op amps, the remaining signal processing is performed inside the PSoC microcomputer using digital processing techniques. The processing will digitize the baseband signal, remove the undesired sideband from the received signal, limit the bandwidth of the audio, and then convert the digital samples back into an analog audio signal.

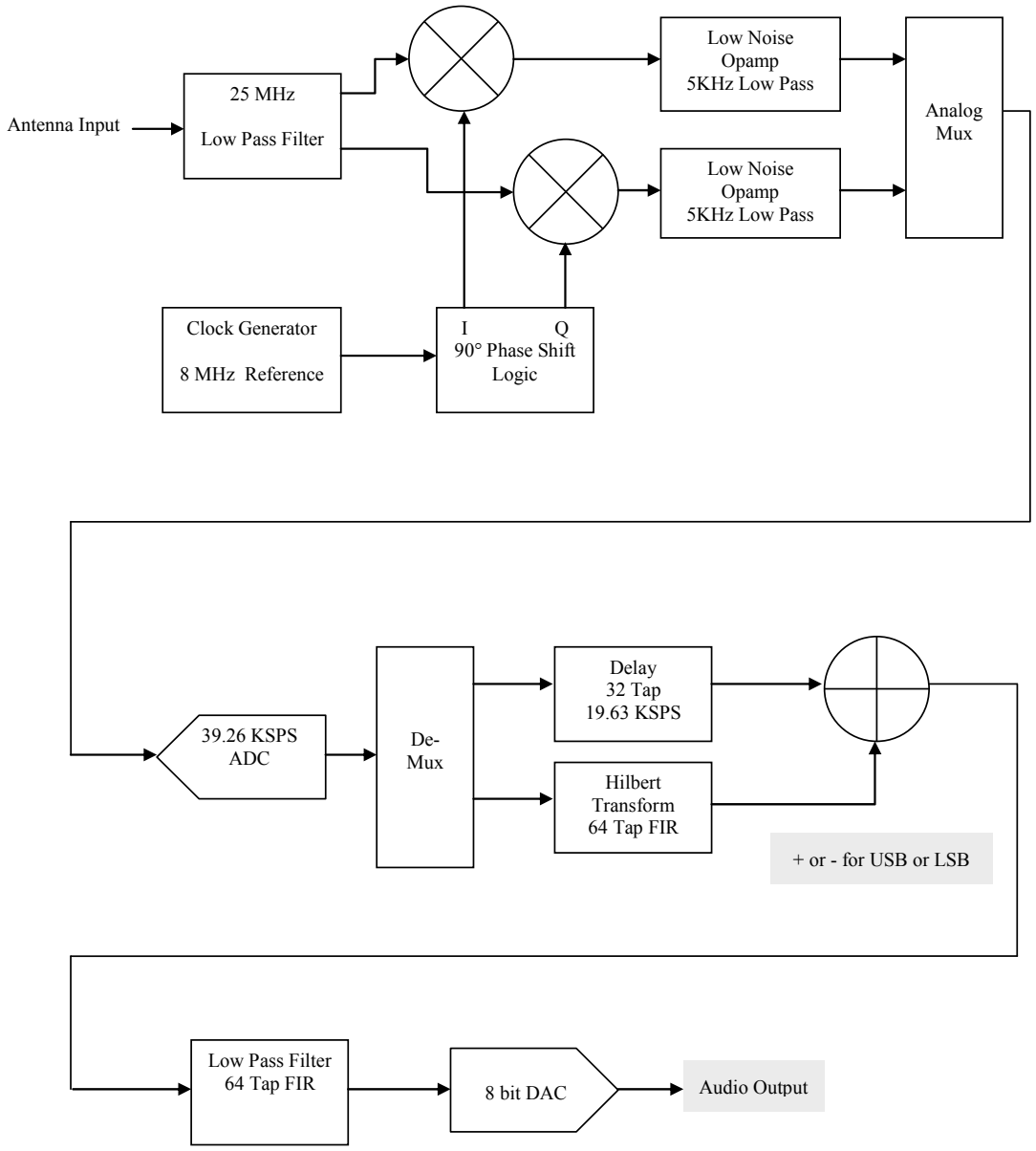


Figure 2: SDR Receiver block diagram.

Circuit Description:

At the antenna input terminals, an RF low pass filter having a 25 MHz corner frequency suppresses signals above the receiver tuning capability. Figure 3 shows the frequency response for the filter.

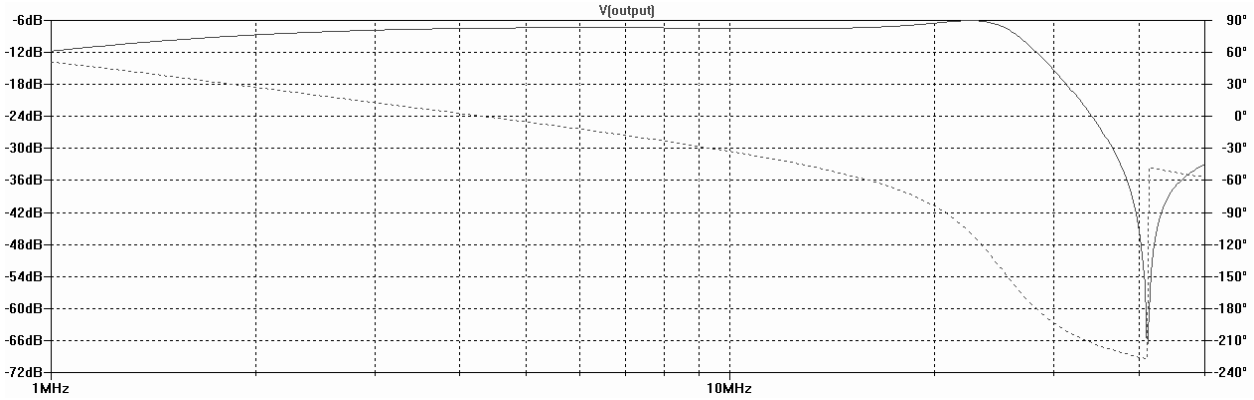


Figure 3: Input RF Filter Response.

Referring to the schematic in figures 6 and 7, U1, a dual 1-of-4 multiplexer/demultiplexer, is used as the sampling detector. This process is similar in functionality as a local mixer, only the control is performed with digital logic switching levels.

A Cirrus Logic clock generator chip, U7, is used to generate a microcomputer controller frequency. It is called a fractional-N clock multiplier and is basically a phase lock loop. The advantages of this component over a discrete digital synthesizer chip (DDS) is that is it cheaper, it has less phase jitter in its output, and it outputs a logic level signal that can be connected directly to the sampling detector multiplexer's control inputs without the need to add a comparator circuit to slice the DDS sine wave output. The clock generator operates at twice the desired receive frequency. The PSoC inputs this clock signal, one rising edge divide by two circuit generates the I clock signal, another falling edge divide-by-two circuit creates the Q clock signal. These two signals are output to the sampling detector multiplexer. They are square waves at the desired receive frequency that are shifted by 90° from one another.

Per the PSoC specification, its inputs are guaranteed to 33 MHz. That means that the clock generator input is the limiting factor in the upper frequency operating range of the receiver. Since this frequency is twice the receive frequency, the highest guaranteed operation is 16.5 MHz. However, after testing several receivers, all have worked to at least 18 MHz (36 MHz clock input) which is near the 17 meter amateur band.

After the mixing process, a pair low noise op amps, U2A and U2B, amplify the base band signals. The part was chosen because of its low noise performance and the ability to operate with inputs and outputs near the ground and power rails. The inputs to the op amps are typically in the microvolt range. Since the op amp circuit voltage gain is on the order of 40 dB at 1 kHz, the output signals are on the order of a few hundred microvolts to a few millivolts. This circuit includes a first order low-pass filter having a 3 kHz corner frequency. It is important to reduce the frequencies above half of the analog to digital converter sampling rate which is called the Nyquist frequency. Otherwise, images will appear at frequencies near the sampling rate. Figure 4 shows the frequency response of the op amp circuit.

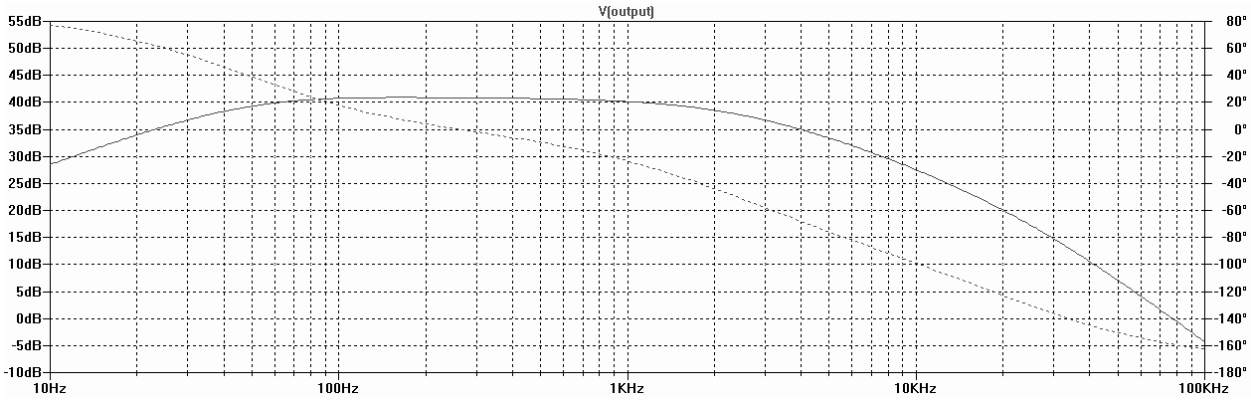


Figure 4: Op amp circuit frequency response.

From this point, all of the signal processing is performed in the digital domain, and, specifically, inside the PSoC, U6.

The PSoC has a single analog to digital converter (ADC). Since there are two base band signals to process, the I and the Q channels, an analog multiplexer is used to switch one of the two inputs to the ADC input. It is desirable to have a high sample rate and a high number of bits, but the best tradeoff I could find is to use the ADC in a 14 bit mode and sampling at 39,260 samples per second. Since a sample from each input channel is necessary, the equivalent sample rate per channel is 19,630 samples per second. Therefore, the Nyquist frequency is almost 10 kHz. The op amp frequency response at 10 kHz is about 15 dB below the desired passband. This is not great, but leaves room for improvement in a future version.

The rest is all digital processing. The PSoC 3 family of parts have an interesting internal hardware feature they call a digital filter block, or DFB, and it consists of a 24-bit fixed point, programmable limited scope DSP engine. This is a dedicated hardware accelerator block that operates independently of the main 8051 processor. It consists of a dedicated multiplier and accumulator that calculates a 24-bit by 24-bit multiply 48-bit accumulates in one system clock cycle. It is optimized to implement a direct form Finite Impulse Response (FIR) filter that approaches a computation rate of one FIR tap for each clock cycle. This block is used as two independent, 64 tap, digital filters.

Alternating outputs from the ADC are loaded into either a 32 sample long delay line or one of the two digital filters. This digital filter uses a set of coefficients that form an all-pass filter having a flat magnitude response, but phase shifts all frequency in its passband by 90°. This is called a Hilbert filter. Suppressing either the upper or lower sideband is accomplished by phase shifting the Q channel baseband data and either subtracting or adding the filter output to the delayed I channel baseband data. The delay is necessary to compensate only for the delays incurred by the processing of the Hilbert filter. The output of the addition is one of the two sidebands.

After the removal of the undesired sideband, the data stream is feed into the other half of the PSoC digital filter block configured as a low-pass filter. This filter has a steep rolloff as shown in the figure 5 for a 2 kHz filter. This is the advantage of processing in the digital domain as compared to a set of analog filters. Steep rolloffs and repeatability of the filter performance over wide temperature ranges and from part to part variations are the reasons to use digital processing.

The output of the low-pass filter is fed into one of the PSoC's 8 bit digital to analog converters (DAC) that converts the data stream back into an analog signal. This signal is buffered with a unity gain op amp, U3, and passed to the output connector.

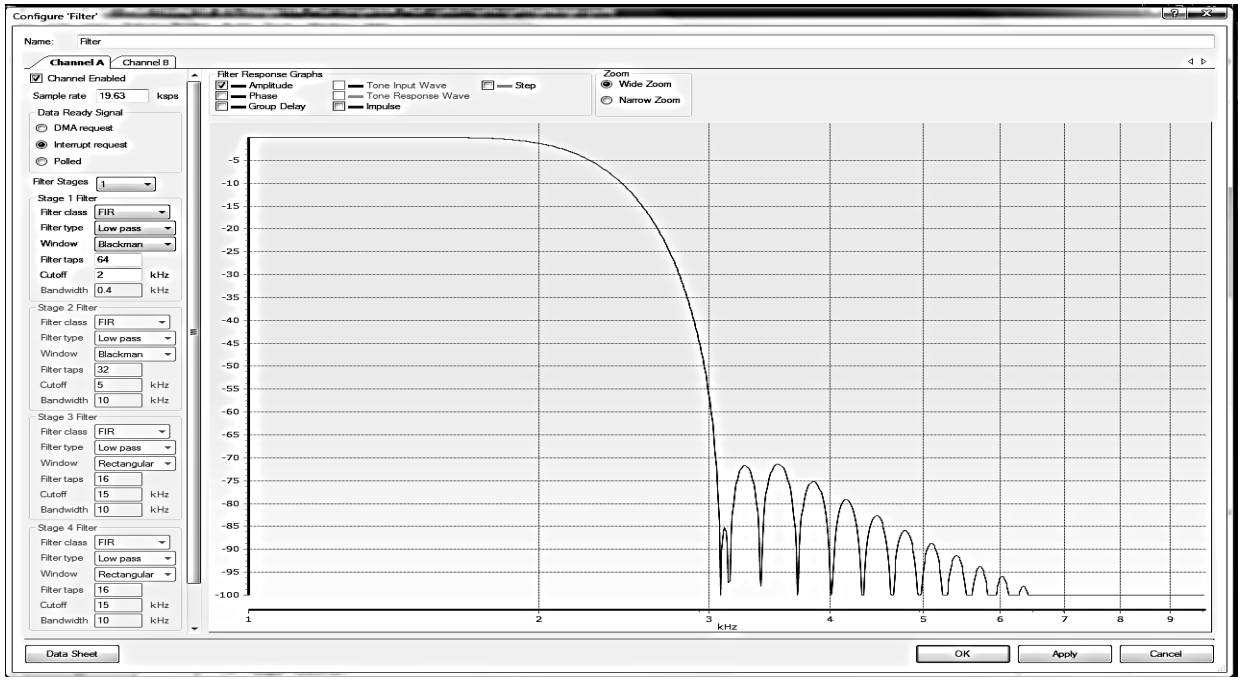


Figure 5: PSoC Digital low-pass filter response.

Operation:

The PSoC has an internal USB full speed port. I wrote firmware for this device to implement a serial communication port. When connected to a personal computer, the receiver will look like a serial device. Using the standard CDC (communication) drivers that are built into Windows, the receiver can communicate with Ham Radio Deluxe. The receiver firmware uses the Elecraft K2 communication protocol. I chose this protocol because it was one of the few that I could find that had a written specification available.

The receiver's audio output will directly drive low impedance headsets. The use of good computer speakers that have a built-in amplifier and volume adjustment is ideal. There is no volume control capability in the receiver. Headsets need their own volume adjustments.

The radio works on all frequencies from 3.5 MHz to at least 18 MHz (80 to 17 meters).

Any one of 4 different receive filter bandwidths can be selected using HRD. FL1 is a 2.5 KHz low pass filter, FL2 is a 2.0 KHz low pass filter, FL3 is a 1.5 KHz low pass filter, and FL4 is used for CW and RTTY and is a 1.0 KHz band pass filter that is 400 Hz wide (+/- 200 Hz).

The attenuator button (ATT) will turn decrease the volume by several dB.

Performance:

The minimum discernible signal (MDS) is approximately -117 dBm. A typical receiver would normally be more towards the -120 dBm number. However, on the lower HF frequencies, where atmospheric and man-made noise dominates, the performance is sufficient. In fact, in a head to head comparison with my Flex Radio Flex-3000, I find it difficult to hear any difference. Not bad for a $50 receiver.

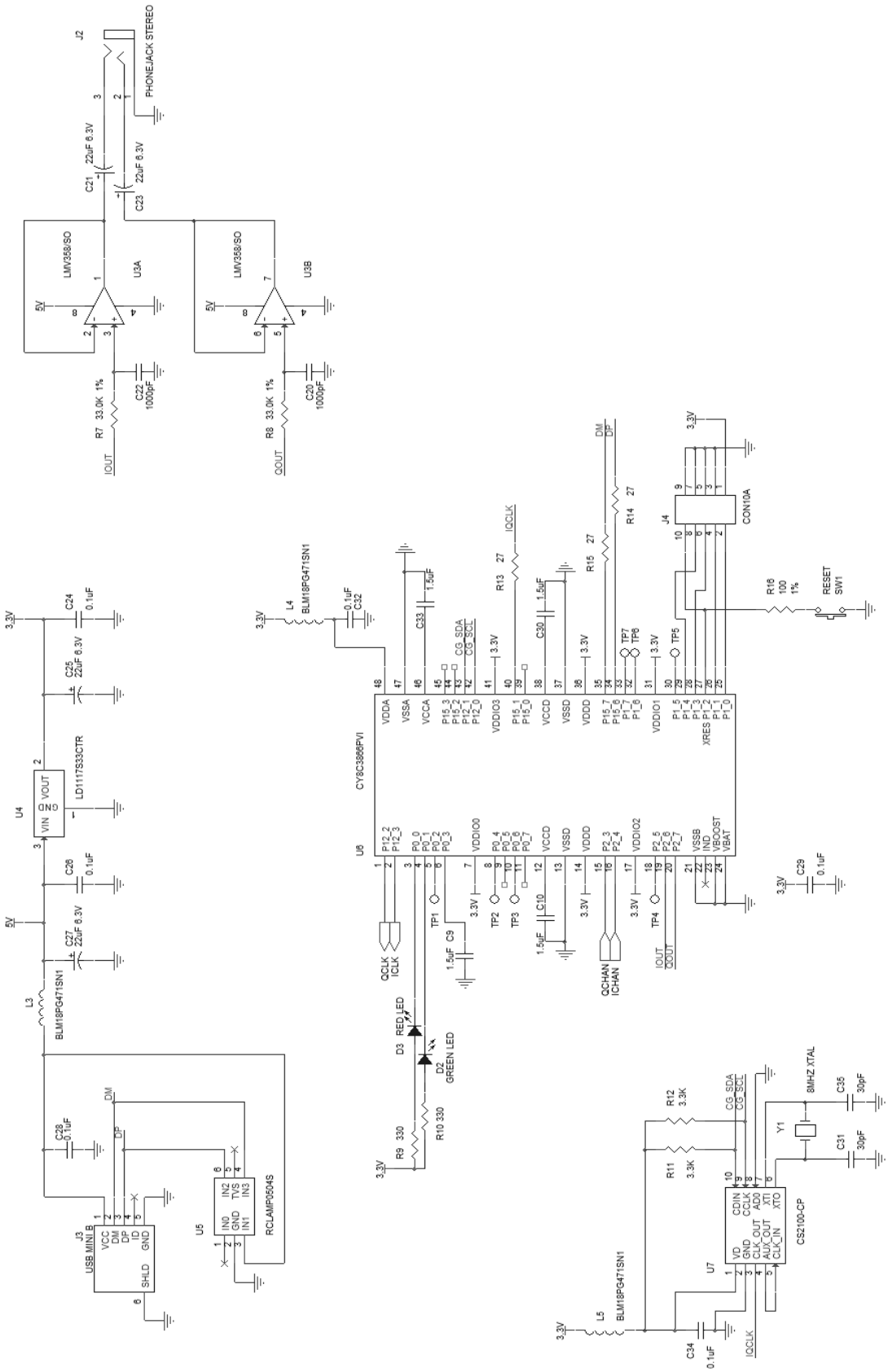


Figure 6: SDR Receiver schematic, sheet 1, digital.

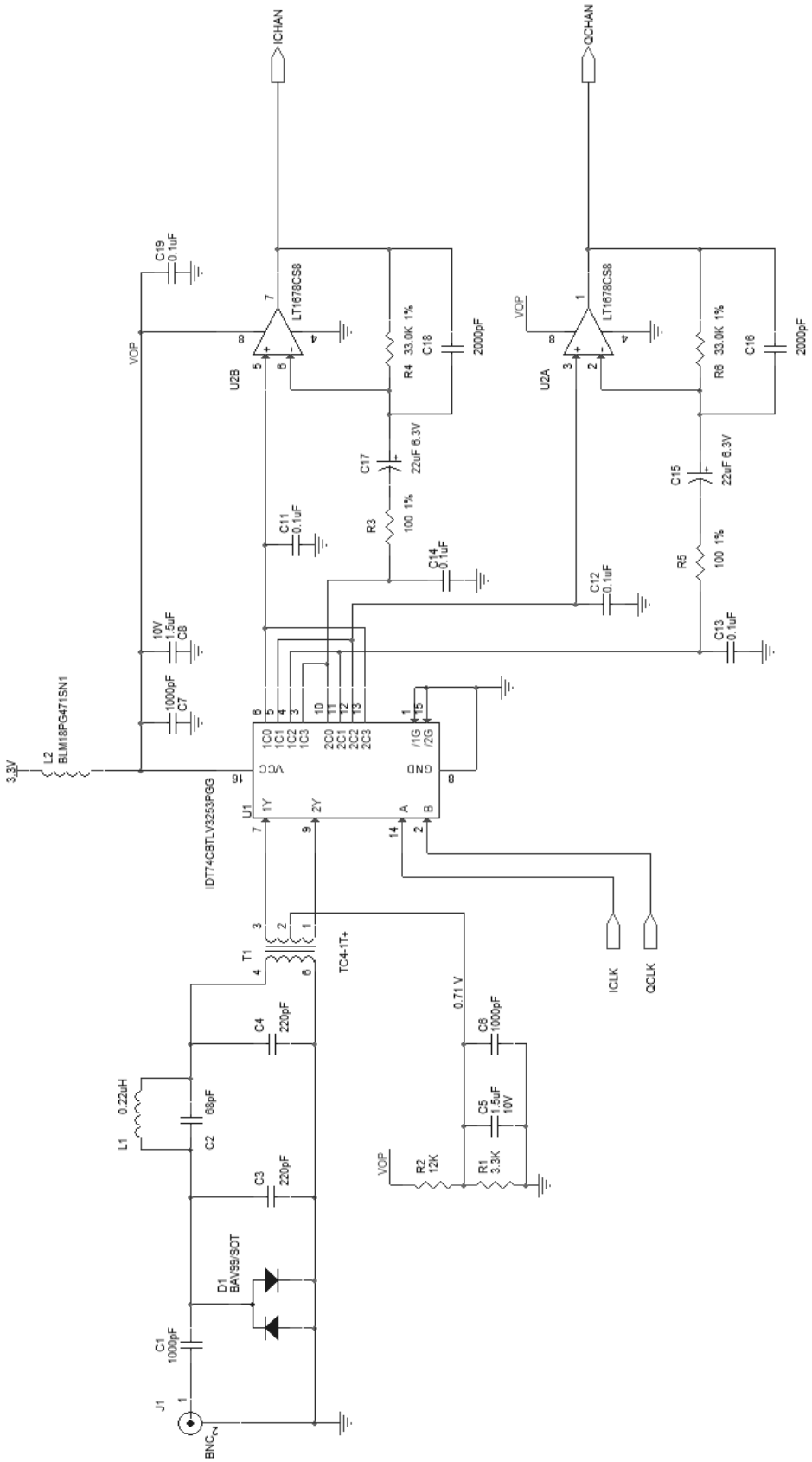


Figure 7: SDR Receiver schematic, sheet 2, RF.