

AC 2007-1032: A SOFTWARE-DEFINED RADIO PROJECT FOR FIRST-YEAR ECET STUDENTS

Peter Goodman, Indiana University-Purdue University-Fort Wayne

PETER E. GOODMAN, P.E. is an Assistant Professor of Electrical and Computer Engineering Technology at IPFW. He earned his BS degree in Electrical Engineering from Rose-Hulman Institute of Technology and his MS degree in Electrical Engineering from Purdue University. He has worked for 28 years in industry and education, and is a member of the IEEE and the ASEE.

A Software-Defined Radio Project for First-Year ECET Students

Abstract: This paper discusses a software-defined radio project which was built by first-year students in an introductory circuit-analysis course. The project was intended to engage and motivate students by providing a real-world application to which they would connect the abstract concepts of circuit analysis. The effort was a success.

Acknowledgements

This work was supported by an Indiana University-Purdue University Fort Wayne Summer Instructional Development Grant. I would like to thank PCB Express, who donated the printed circuit boards, and the following companies who supplied parts as donations or samples: Molex, Inc., Mouser Electronics, Kemet Corp., Linear Technology Corp., and Analog Devices, Inc.

Introduction

Most of us are aware that engineers are reputed to be boring people who toil in a dry, boring profession. That stereotype is, of course, far from true, but our many of our first-year students and potential students have not learned yet that technology is fun and rewarding. Students who don't learn that lesson in the first or second semester may leave the profession for something more exciting, such as Accounting or Folklore.

Unfortunately, the introductory Electrical Circuits courses taken by first-year ECET students do little to inspire most of them. Homework assignments ask students to analyze networks of resistors (and, before long, capacitors, inductors and other elements) which illustrate concepts like Kirchoff's Laws and Thevenin's Theorem, but which do not actually do anything of use in the "real world". Typical laboratory exercises are similar, and similarly dry. If our retention percentage is not what we would like it to be, we should not be surprised.

Of course, there are a significant number of students who do not leave for the greener pastures of microeconomics. Some of them already know that our discipline is rewarding in many ways other than a paycheck, possibly because they began tinkering with technology while in high school. If we could give the rest of our students a first-year experience something like the experience these tinkerers and radio amateurs give themselves, it seems reasonable to think that our retention rates would benefit. That kind of experience can be provided by assigning all first-semester students to build a small project which is challenging but not beyond their skills. This project should do something useful and interesting, but it should also be useful for demonstrating and reinforcing the principles of circuit analysis. Some of us may be surprised to find that a software-defined radio (SDR) project meets all of these requirements.

In the Fall semester of 2006, each student enrolled in the introductory circuit analysis course was required to build the "front-end" hardware of a simple SDR. On completion, each student took his or her radio to the IPFW Amateur Radio Club station where it was connected to an antenna and to a PC running the "back-end" DSP software which defined the radio, and heard Morse code signals which originated thousands of miles away. The satisfaction and pride experienced by each of these students was very obvious. It is too soon to judge the effect on retention, but the results of an end-of-semester survey of the students seem encouraging in that regard.

The idea of requiring first-semester students, many of whom know little about electronics at the beginning of the semester, to build, test, debug, and experiment on a software-defined radio may

seem dubious to some, but the circuits involved are less complex than the phrase “software-defined radio” suggests. While it is a fairly simple project, it is complex enough that portions of it serve as examples of series and parallel networks, resistive voltage dividers, Thevenin equivalents, and so on, all focused on a very modern implementation of the oldest application of electronics: a radio receiver.

Background

An FM radio receiver project built by students in a junior-level course in electronic communications course was recently described.¹ That project was designed to link curricular material learned in two sophomore-level courses (one in RF and power electronics and the other in microcontrollers) and to “address problems such as lack of student motivation, poor retention, segment learning, and lack of integration”. These problems, especially motivation and retention, need to be addressed in the first year, because many students would not make it to the third year otherwise. The “reverse” approach is reported here: incoming students start building a radio project immediately, and fundamental concepts are linked to the project as the course progresses.

Students taking the sophomore-level Electronic Systems Fabrication² course are currently required to build a simple software-defined radio based on the “Softrock 40³”, a low-cost (\$20 or less in parts) software-defined receiver which has been the subject of much discussion and experimentation in the Amateur Radio community. Each of those students must design a PCB for the radio, have it fabricated, and assemble it using surface-mount parts. The first-year project was adapted from that project by providing the first-year students with PCBs and simply requiring them to assemble, test, and (if necessary) debug.

The phrase “software-defined radio” may suggest that analog-to-digital (A/D) conversion is performed at the antenna connector, with all subsequent signal processing (upconversion and/or downconversion, intermediate-frequency (IF) filtering, demodulation, etc.) performed in reconfigurable software using digital signal processing (DSP) techniques. The term SDR may be defined more generally to allow some analog signal processing, such as downconversion, ahead of the A/D converter as long as most of the signal processing performed in software⁴. SDR architectures which fit this definition include the IF Sampling architecture, the Zero-IF architecture, and the Near-Zero-IF architecture⁵.

For obvious reasons, it was necessary to minimize the cost and complexity of this project. Experimentation with SDR hardware and software has recently become widespread in the Amateur Radio community, where minimization of cost and complexity are similarly important. Much of this activity has been focused on the Near-Zero-IF approach, in which the RF band of interest is downconverted to the audio frequency range. A/D conversion is then performed by a PC sound card. The sound card’s two input channels, normally used as “left” and “right” channels for stereo sound recording, are used as “in-phase” (I) and “quadrature” (Q) signal processing channels as shown in Figure 1. The Flex Radio Systems SDR-1000^{6,7,8,9}, a production high-frequency (HF, or “shortwave” in the popular literature) transceiver which costs approximately \$1000¹⁰, is one example of this architecture. Another example is the “Softrock-40” receiver, which can be built for \$15 to \$30. The software which defines both of these radios may be downloaded at no cost, and runs on a standard PC. “Rocky” is a simple SDR software package developed specifically for Softrock-40 type radios¹¹. The much more capable “PowerSDR” software is available from Flex Radio Systems. Other SDR software relevant

activity in this area includes the GNU Radio Project¹², a Free Software Foundation (FSF) effort to coordinate the development of open-source SDR hardware and software.

Hardware

A block diagram of the SDR project, which is a variant of the “Softrock-40 Version 5” is shown in Figure 1:

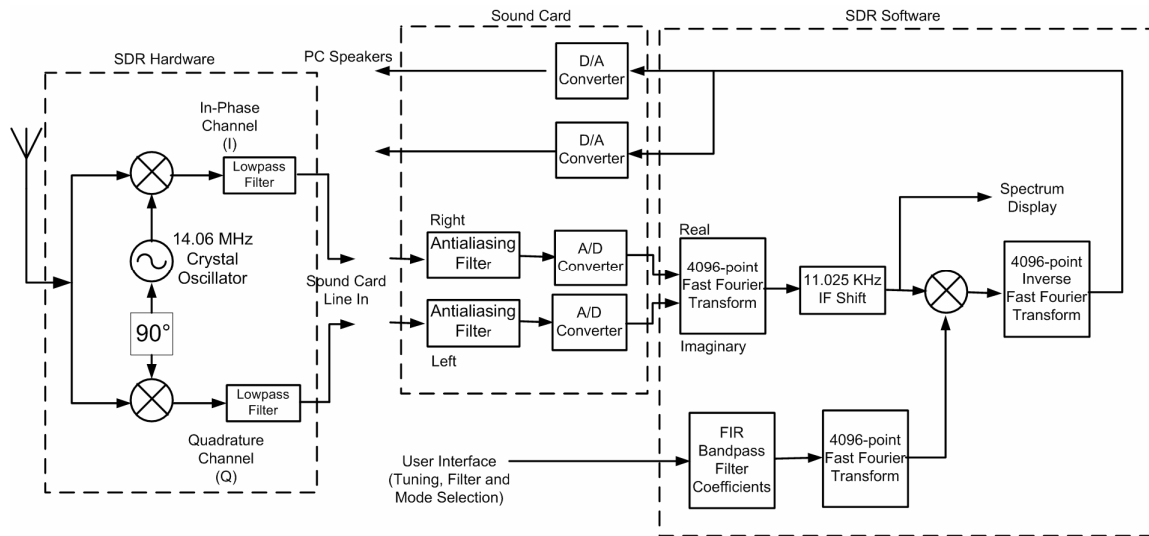


Figure 1
SDR System Block Diagram

This radio is designed to operate in the “20 meter” Amateur Radio band. The 20 meter version was chosen, because that band (14.000 – 14.350 MHz, approximately 20 meter wavelength) supports worldwide communication using low power during some portion of most days. For example, the author has contacted other Amateur Radio operators as far away as New Zealand using a 5-watt transceiver. Furthermore, an antenna for the 20 meter band is only half the size of an equivalent antenna for the 40 meter band.

The Softrock 40 Version 5 design was revised to be built on one PCB instead of two to reduce the cost, but provisions were made to allow the connection of “daughterboards” to change bands or add capabilities. A subsequent project will use that header to connect a Direct Digital Synthesizer (DDS) and filter module to extend the receiver’s frequency coverage to the entire HF spectrum (3 – 30 MHz).

A student-built SDR is shown in Figure 2:

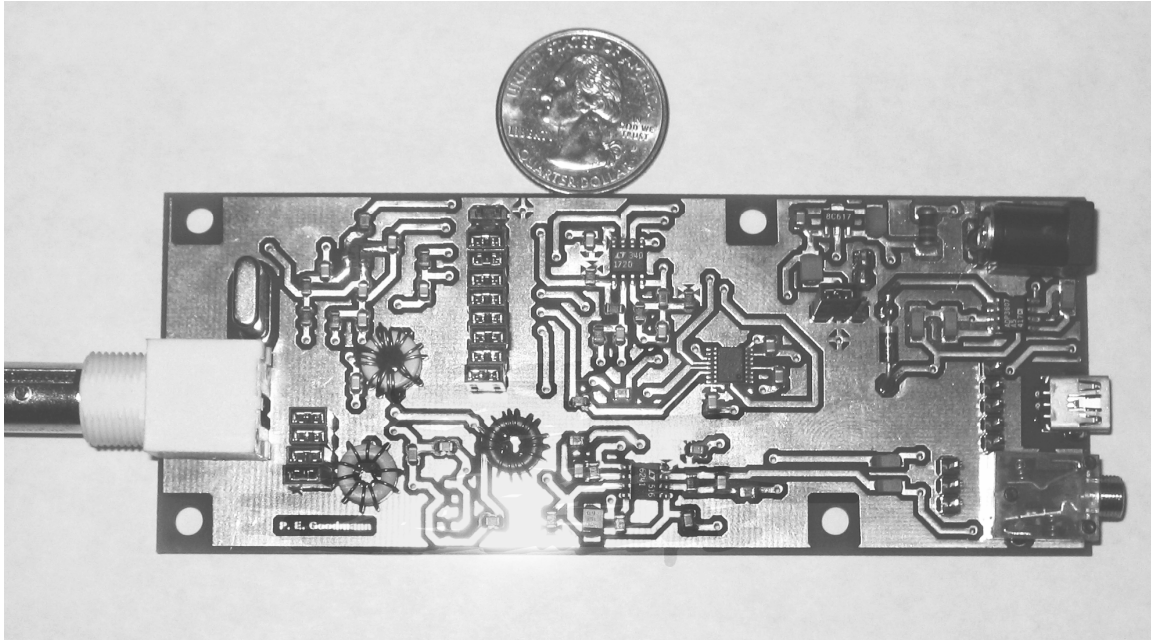


Figure 2
SDR Hardware

The original Softrock-40 design used surface-mount parts only where it was unavoidable. Most Radio Amateurs think surface-mount assembly is beyond their capabilities, but previous experience has shown that anyone can assemble a surface-mount project if good instruction and proper low-cost equipment is available. Therefore, this SDR project uses surface-mount parts wherever possible. Most students were initially apprehensive, but after assembling the radio think surface-mount assembly is actually easier than through-hole assembly. Figure 3 shows a group of students assembling their SDRs.



Figure 3
Students Assembling SDRs

Instructional Use of Project

The SDR project would probably have been worth doing if its only benefits were motivating the students, but one of the goals of this effort was to use it as an instructional tool by using it as a source of circuit examples and laboratory experiments in the first-semester DC Circuit Analysis course. This course, which is taught with the assumption that most students begin the semester knowing little or nothing of the nature of electrical circuits, covers the basics: charge, voltage, current, power, resistance, series and parallel circuits, Kirchoff's laws, mesh and node analysis, Thevenin's and Norton's theorems, and the superposition theorem. Very simple introductions to junction diodes, bipolar transistors, and operational amplifiers are also presented. Each of these principles and devices are present in the SDR, a slightly simplified schematic diagram of which is shown in Figure 4:

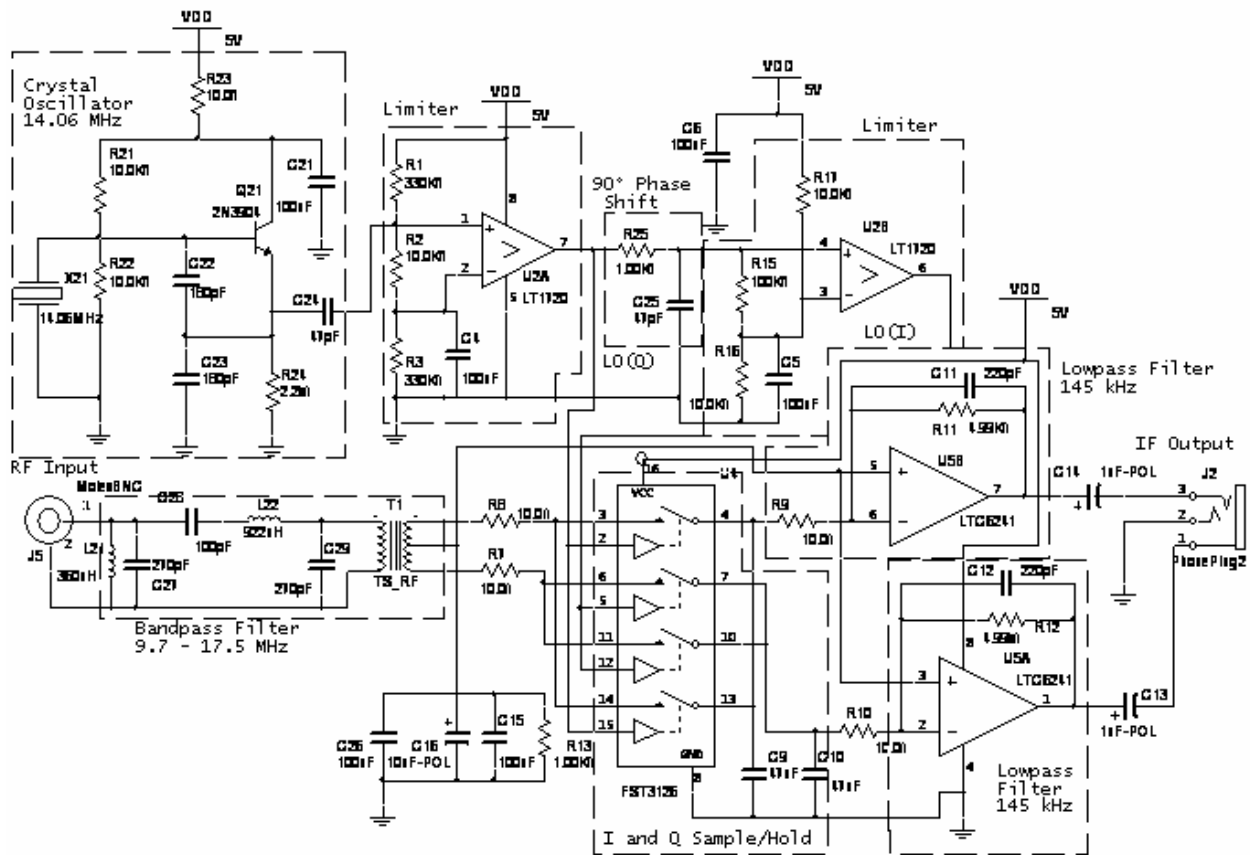


Figure 4
Simplified Schematic Diagram of SDR

Much of this course is concerned with Kirchoff's laws, series and parallel resistive circuits, and voltage dividers. Most of the students finish attaching the surface-mount resistors and capacitors to the PCB just before they are introduced to series circuits, so they are given the assignment of making a series of resistance measurements on their partially-complete PCBs. This introduces them to the use of an ohmmeter, and allows them to discover any poor solder joints or other assembly mistakes they might have made. They may find some of the resistances they measure to be less than they expect, but soon discover that the discrepancy is due to parallel circuits they

have not recognized. When the concept of the voltage divider network is introduced, the students are asked to analyze the networks which provide the base bias voltage of Q21 and which bias comparators U2A and U2B. After calculating the bias voltages, they are to find them using a circuit simulator (Electronics Workbench Multisim), and then they measure the same voltages in the actual hardware.

Later, the students given the assignment of finding the Thevenin resistance of the crystal oscillator's output port by using Multisim, then comparing the predicted value with a measured value. Those students whose SDRs do not work initially gain valuable debugging experience as they are coached through the debugging process, which is a very good opportunity to introduce them to the oscilloscope.

Most students seem to require about six hours to complete assembly of the SDR, about twice as long as required by the instructor. Those whose SDR does not work immediately may require an additional 2-4 hours for debugging.

Software

Development of digital signal processing software for this project is far beyond the capability of most first-year ECET students. Fortunately, several SDR software packages have been developed by Amateur Radio operators who have made them available to download at no cost. These include PowerSDR¹³, Rocky¹¹, and SDRadio¹³. The students who built this project were encouraged to download and use Rocky. A general description of the software, based on Youngblood's article⁸, follows:

The SDR hardware's I and Q channel outputs (see Figure 1) are connected to the "Left" and "Right" channels of the PC soundcard's "Line In" jack. Two input channels are absolutely necessary, so a monaural input such as the "Mic" jack (the only input available on most portable computers) will not work. The soundcard lowpass filters both input channels, then performs analog to digital (A/D) conversion at 16-bit resolution and a sample rate of 48,000 samples per second.

The composite sample rate (taking both channels into account) is 96,000 samples per second, making the system's Nyquist bandwidth 48 kHz. Thus, the SDR is capable of receiving signals within a 48 kHz-wide band centered at the frequency of the crystal oscillator, 14.06 MHz. Professional-sound cards which sample with 24-bit resolution and 96,000 (or even 192,000) samples per second (per channel) are available, and offer improved dynamic range and bandwidth at a much greater dollar cost.

Each pair of samples from the two channels is treated as a single complex-valued sample, with the real part supplied by the right channel and the imaginary part supplied by the left channel. The resulting sequence of complex-valued samples is divided into segments of 4096 consecutive samples. The first 2048 samples of each segment are identical to the last 2048 samples of the previous segment, which means successive segments overlap each other by 2048 samples. Each segment is transformed from the time domain to the frequency domain by a fast Fourier transform (FFT) algorithm. The coefficients of a user-selectable 2048th-order finite impulse response (FIR) filter are also transformed by a second instance of the FFT, but this instance only needs to be executed if the user changes the filter response. An IF Shift is performed on the transformed signal segment to bring the desired frequency into the filter's passband, then the segment is multiplied by the transformed filter coefficients. The result is transformed back to the

time domain by the inverse fast Fourier transform (IFFT) algorithm, and the process is repeated on the next segment. This results in a series of overlapping, but now filtered, segments of 4096 samples which are de-overlapped and assembled into a sequence of filtered, time-domain samples, completing a process called overlap-add¹⁴. This sequence is digital-to-analog (D/A) converted by the soundcard, amplified, and sent to the speakers.

The spectrum display, which shows the users the signal he has tuned the receiver to along with other signals above and below the tuning point, is a side benefit of using the FFT and IFFT algorithms. It provides operational advantages to an Amateur Radio operator, but it also introduces the students to the concept of the frequency domain much earlier in the curriculum than has traditionally been the case. This is a difficult concept for those to whom it is unfamiliar, so introducing it early may help more students to understand it by the time they graduate.

Survey Results

At the end of the semester, the students were given a voluntary survey of five questions relating to the SDR project's effect on the learning experience. Each of the questions presented a statement, and asked the student to agree or disagree. The statements were as follows:

Statement 1: "In my opinion, the approach taken in this course of linking the course material to a theme such as radio, and focusing the lab on a project related to that theme, leads to a better learning experience than the traditional approach in which lab exercises do not directly involve a real-world application."

Statement 2: "At the beginning of the semester I was undecided as to my major, but I have decided EET or CPET is the right choice at least partially because of the SDR project."

Statement 3: "The SDR Project helped me connect abstract principles such as Kirchoff's Laws and Thevenin's Theorem with real-world applications"

Statement 4: "The SDR Project was enjoyable and satisfying."

Statement 5: "When I saw the radio I built work for the first time I felt a significant sense of accomplishment."

The students were asked to indicate their degree of agreement or disagreement on a 0 to 4 scale, with 4 indicating strong agreement, 2 indicating neutral, and 0 indicating strong disagreement. The results are summarized in Table 1:

	Average	Agree or Strongly Agree	Neutral	Disagree or Strongly Disagree
Statement 1	3.35	16	1	0
Statement 2	2.41	7	7	3
Statement 3	2.59	11	2	4
Statement 4	3.29	14	2	1
Statement 5	3.41	13	4	0

Table 1
Results of Student Survey

The survey responses, especially those to statements 1, 4 and 5, clearly show that the students that students viewed the project as having a beneficial effect on their learning experience.

Statement 2 was aimed at those students who were undecided as to their major at the beginning of the semester. The response to that statement may appear inconclusive, but the wording of the statement makes it likely that some or all of the seven neutral responses came from students who were decided at the beginning of the semester. If the neutral responses are disregarded, 70% of the remaining responses indicate that the project was beneficial.

65% of responses to Statement 3 indicated that the project was helpful in connecting abstract principles to real-world applications. Individual students have different learning styles, and an experience which helps one student make the connection between an abstract concept and a concrete application may not help another. The fact that a majority of students indicated that the SDR project, which provided concrete experiences that had not previously been included in the course, helped them to make those connections seems very significant.

The overwhelmingly positive responses to Statements 4 and 5 speak for themselves.

Retention

This course is required for all ECET students, and is taken by a number of “undecided” students who are considering a major in ECET. Unfortunately, an examination of the academic records of the students who enrolled in the course revealed that a significant number of them were practically doomed to fail before they started the semester. The records of several students went back for several semesters (or even several years), showing a history of withdrawals and failures. One student had enrolled in 47 hours over a number of semesters, but only completed one 4-hour course (and that with a grade of “D”). Several others were in their first semester at this university and performed poorly (withdrawals and failures) in most or all of the courses they were enrolled in. Of the 42 students who were initially enrolled in this course, 9 clearly began the semester with serious problems. If the retention rate is calculated based on the remaining 33 students, 77% were retained in the ECET programs.

When the retention rate is based on the initial enrollment of 42 students, the result is 60%. However, this still compares well with 56% retention for the same course with traditional laboratory exercises instead of the SDR project.

Conclusion

The inclusion of the software-defined radio project in an introductory circuit analysis course was a success. The project was within the capabilities of the students, and improved their learning experience by providing a real-world application to which they could connect the abstract concepts of circuit analysis. It also provided excellent hands-on experience working with surface-mount technology, a powerful motivational experience of seeing the modules the students had built themselves working, and a demonstration of special-purpose hardware and software combined with a general-purpose computer to perform a specific task. The effect on retention was measurably positive.

Bibliography

1. Richardson, Jeffrey J., John Denton, and James Jacob, "Threading Topics and Creating Course Linkage Among Courses and Curricular Areas", *Proceedings of the 2006 American Society for Engineering Education Annual Conference and Exposition*, Chicago, Illinois.
2. Goodman, Peter E., "Geek Civilization: Amateur Radio and First-Year Projects to Improve Recruitment and Retention in an ECET Program", *Proceedings of the 2006 American Society for Engineering Education Annual Conference and Exposition*, Chicago, Illinois.
3. <http://www.amqrp.org/kits/softrock40/>
4. Tuttlebee, Walter H. W., "Software-Defined Radio: Facets of a Developing Technology", *IEEE Personal Communications*, April, 1999, pp. 38-44.
5. Nakajima, Nobuo, Ryuji Kohno, and Shuji Kubota, "Research and Developments of Software-Defined Radio Technologies in Japan", *IEEE Communications Magazine*, August, 2001, pp. 146-155.
6. Youngblood, Gerald, "A Software-Defined Radio for the Masses, Part 1", *QEX*, July/August, 2002, pp. 13-21.
7. Youngblood, Gerald, "A Software-Defined Radio for the Masses, Part 2", *QEX*, September/October, 2002, pp. 10-18.
8. Youngblood, Gerald, "A Software-Defined Radio for the Masses, Part 3", *QEX*, November/December, 2002, pp. 27-36.
9. Youngblood, Gerald, "A Software-Defined Radio for the Masses, Part 4", *QEX*, March/April, 2003, pp. 20-31.
10. FlexRadio Systems, <http://www.flex-radio.com/>
11. Shovkoplyas, Alex, "Rocky 1.5", <http://www.dxatlas.com/rocky/>, 13 December 2006.
12. "GNU Radio", <http://www.gnu.org/software/gnuradio/>, 13 December 2006
13. Di Bene, Alberto, "The Weaksignals Pages", <http://www.sdradio.org/>, 13 December 2006.
14. Oppenheim, Alan V. and Ronald W. Schaffer, *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1975, pp. 113-114.