QEX

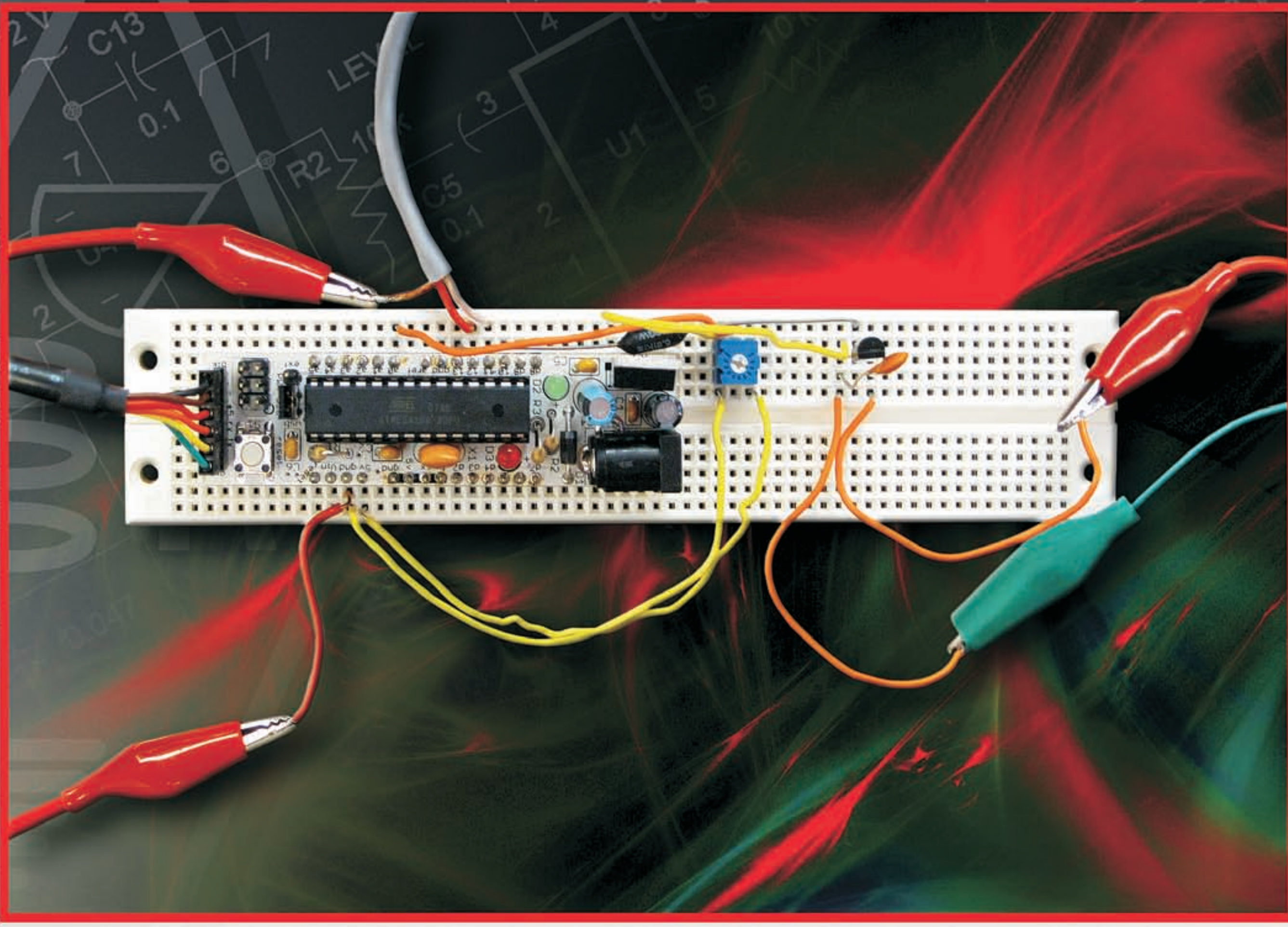
$5

September/October 2009

www.arrl.org

A Forum for Communications Experimenters

Issue No. 256



KC4IFB introduces the Arduino microcontroller board as a development platform. He shows us how to design and program an electronic keyer in this issue.

ARRL The national association for AMATEUR RADIO
225 Main Street
Newington, CT USA 06111-1494

KENWOOD

Listen to the Future

A portrait of extraordinary excellence

KENWOOD U.S.A. CORPORATION
Communications Sector Headquarters
3970 Johns Creek Court, Suite 100, Suwanee, GA 30024
Customer Support/Distribution
P.O. Box 22745, 2201 East Dominguez St., Long Beach, CA 90801-5745
Customer Support: (310) 639-4200 Fax: (310) 537-8235

www.kenwoodusa.com

ADS#25109

JQA-1205 091-A
ISO9001 Registered
Communications Equipment Division
Kenwood Corporation
ISO9001 certification

QEX (ISSN: 0886-8093) is published bimonthly in January, March, May, July, September, and November by the American Radio Relay League, 225 Main Street, Newington, CT 06111-1494. Periodicals postage paid at Hartford, CT and at additional mailing offices.

POSTMASTER: Send address changes to: QEX, 225 Main St, Newington, CT 06111-1494
Issue No 256

Harold Kramer, WJ1B
Publisher

Larry Wolfgang, WR1B
Editor

Lori Weinberg, KB1EIB
Assistant Editor

Zack Lau, W1VT
Ray Mack, W5IFS
Contributing Editors

Production Department

Steve Ford, WB8IMY
Publications Manager

Michelle Bloom, WB1ENT
Production Supervisor

Sue Fagan, KB1OKW
Graphic Design Supervisor

David Pingree, N1NAS
Senior Technical Illustrator

Advertising Information Contact:

Janet L. Rocco, W1JLR
Business Services
860-594-0203 – Direct
800-243-7768 – ARRL
860-594-4285 – Fax

Circulation Department

Cathy Stepina, *QEX Circulation*

Offices

225 Main St, Newington, CT 06111-1494 USA
Telephone: 860-594-0200
Fax: 860-594-0259 (24 hour direct line)
e-mail: **qex@arrl.org**

Subscription rate for 6 issues:

In the US: ARRL Member $24, nonmember $36;

US by First Class Mail: ARRL member $37, nonmember $49;

International and Canada by Airmail: ARRL member $31, nonmember $43;

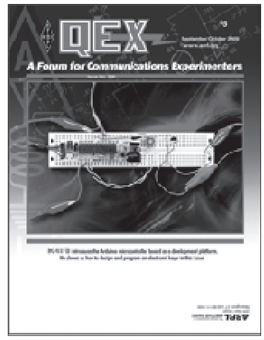
Members are asked to include their membership control number or a label from their QST when applying.

In order to ensure prompt delivery, we ask that you periodically check the address information on your mailing label. If you find any inaccuracies, please contact the Circulation Department immediately. Thank you for your assistance.

September/October 2009

About the Cover

Richard Chapman, KC4IFB, used the Arduino microcontroller board to design and program an electronic keyer. The Arduino is a development platform that can be used to design and implement many microcontroller-based Amateur Radio projects.



In This Issue

Features

Columns

Index of Advertisers

The American Radio Relay League

The American Radio Relay League, Inc, is a noncommercial association of radio amateurs, organized for the promotion of interest in Amateur Radio communication and experimenta-tion, for the establishment of networks to provide communications in the event of disasters or other emergencies, for the advancement of the radio art and of the public welfare, for the representation of the radio amateur in legislative matters, and for the maintenance of fraternalism and a high standard of conduct.

ARRL is an incorporated association without capital stock chartered under the laws of the state of Connecticut, and is an exempt organization under Section 501(c)(3) of the Internal Revenue Code of 1986. Its affairs are governed by a Board of Directors, whose voting members are elected every three years by the general membership. The officers are elected or appointed by the Directors. The League is noncommercial, and no one who could gain financially from the shaping of its affairs is eligible for membership on its Board.

"Of, by, and for the radio amateur," ARRL numbers within its ranks the vast majority of active amateurs in the nation and has a proud history of achievement as the standard-bearer in amateur affairs.

A *bona fide* interest in Amateur Radio is the only essential qualification of membership; an Amateur Radio license is not a prerequisite, although full voting membership is granted only to licensed amateurs in the US.

Membership inquiries and general correspondence should be addressed to the administrative headquarters:

ARRL
225 Main Street
Newington, CT 06111 USA
Telephone: 860-594-0200
FAX: 860-594-0259 (24-hour direct line)

Officers

President: JOEL HARRISON, W5ZN
528 Miller Rd, Judsonia, AR 72081

Chief Executive Officer: DAVID SUMNER, K1ZZ

The purpose of *QEX* is to:

1) provide a medium for the exchange of ideas and information among Amateur Radio experimenters,

2) document advanced technical work in the Amateur Radio field, and

3) support efforts to advance the state of the Amateur Radio art.

All correspondence concerning *QEX* should be addressed to the American Radio Relay League, 225 Main Street, Newington, CT 06111 USA. Envelopes containing manuscripts and letters for publication in *QEX* should be marked Editor, *QEX*.

Both theoretical and practical technical articles are welcomed. Manuscripts should be submitted in word-processor format, if possible. We can redraw any figures as long as their content is clear. Photos should be glossy, color or black-and-white prints of at least the size they are to appear in *QEX* or high-resolution digital images (300 dots per inch or higher at the printed size). Further information for authors can be found on the Web at **www.arrl.org/qex/** or by e-mail to **qex@arrl.org**.

Larry Wolfgang, WR1B

lwolfgang@arrl.org

Empirical Outlook

Can You Teach an Old Dog New Tricks?

After four decades as an Amateur Radio operator, this hobby continues to amaze me. New experiences and learning opportunities abound. Field Day has always been one of my favorite ham radio events. I've participated with a number of clubs over the years, and the camaraderie and pooling of resources for a successful weekend of emergency communications practice is always wonderful. In recent years my wife, WB3IOS, and I had been holding our own Field Day operation, camping in several states and enjoying an extended weekend "vacation."

This year that wasn't going to work out, however, so I opted to do Field Day with a local Amateur Radio club. I hadn't been to Field Day with this club before, although they have been going to the same local State Forest for years. Quite a few of the club members camp overnight, so that fits right in with my idea of Field Day.

There were plenty of operating opportunities for those who wanted to be on the air, although many who came out to the site seemed content to just enjoy some ham radio conversation. Everyone seemed to have a lot of fun.

I noticed that no one had brought a digital-mode station, and that started me thinking (again) about the NUE-PSK modem kit that was sitting in a box at home, waiting for me to get the nerve to start the construction. That box has been sitting in my shack since March 2008. (The NUE-PSK modem was described in the Mar/Apr 2008 issue of *QEX*, with an overview in the Mar 2008 issue of *QST*.) I've opened the box several times to read the directions and consider the process of building the modem.

I really enjoy building projects, and consider myself to be a rather experienced builder. Still, there was something daunting about this project. Not only does the NUE-PSK modem involve soldering surface mount components, but there are several surface mount ICs. Now, some of those ICs are similar to common 8 or 16 pin DIP through-hole-mount ICs, but one is a 64 pin microcontroller that is about ½ inch square, with 16 leads on each side!

I decided I had put this off long enough. Although I was very reluctant to try placing that microcontroller IC on the circuit board, I was also not ready to accept defeat and send it back to the article authors, George Heron, N2APB, or Milt Cram, W8NUE, who had offered to solder at least the dsPIC IC for me.

With space cleared on my operating desk and my magnified lamp clamped to the side of the desk, it was time to go for it. I tried one of the larger ICs first, and it really wasn't all that bad. With the lighted magnifying lens I could actually see the leads very well, and it was not too difficult to line up the pins with the circuit board pads. Tack solder one corner lead, and double check to make sure the IC pins are all aligned with the pads. Then tack the opposite corner lead and check again. Yes, it looks like everything is aligned perfectly. Next I heated the row of pins and circuit pads, and just ran the solder along the board. That resulted in one blob of solder along each side, but after laying some solder wick on top of the row of leads and heating with my soldering iron, it looked pretty good. Actually, it looked *very* good.

I followed the same procedure for the 64 pin dsPIC, but found it was much more difficult to really see when the pins all lined up with the circuit board pads. In fact, something always looked a bit out of place along one side. The other 3 looked perfect, though, so I soldered the IC in place. After adding the surface mount resistors and capacitors, along with a few through-hole components, I was ready for the ultimate test. After carefully examining the board under the magnifying lens, and finding no shorts or other obvious problems, I connected the two 9 V batteries and turned on the power. I could see the display backlight come on, but there was no sign-on message, and no line for the frequency display.

I consulted with Milt, W8NUE, and he suggested that there were probably some pins shorted on the dsPIC, but I really could not see it well enough to decide. My dilemma was how to gain more magnification. It finally occurred to me that I might try taking a digital photo through the magnifying lens, and then enlarging that photo on my computer screen. Well, it worked! As soon as I blew up the best photo, I could see several pins that appeared to be bent, and touching the neighboring circuit board pads.

Armed with this enlarged image, I was able to heat the appropriate pins with my soldering iron and then apply some lateral pressure with a heavy straight pin to move the leads back into place. Another photo confirmed that the pins looked to be back in place, so I again applied power. Success! Oh, the joy of seeing something I built come to life for the first time. It wasn't long before I had my first PSK-31 QSO with the new modem. The next time I am faced with soldering such a small IC with so many leads, I'll tack a corner in place and take a photo. Then I'll tack another corner and take another photo. With that increased magnification, I will be able to spot any problems such as bent leads. Perhaps my "discovery" will encourage you to try building a project with surface mount ICs, too. It's fun!

Richard Chapman, KC4IFB
1300 Hickory Ln, Auburn, AL 36830; chapman@eng.auburn.edu

Build a Low-Cost Iambic Keyer Using Open-Source Hardware

Use the Ardvino prototyping board to learn how to build with and program a microcontroller.

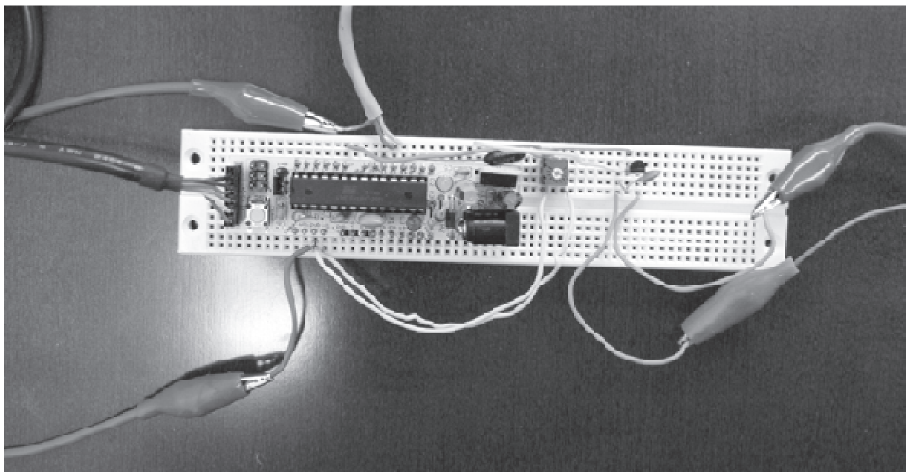


Figure 1 — The author's keyer is shown here. The "DC Boarduino" version of the Arduino from www.ladyada.net plugs into a prototyping board to add external components as well as the leads to the paddle, key line and power. You can see the "barrel" style dc connector on the right end of the Boarduino. There is also a version of the Boarduino with a USB connector instead of the barrel style dc connector.

Introduction

Have you ever wanted to prototype a ham radio project with a microcontroller, but thought you needed a degree in computer engineering to do it? Programming microcontrollers *can* be difficult or expensive, but it doesn't have to be. In this article I will try to convince you that anyone with basic electronics and programming skills can build microcontroller based projects for ham radio.

I've been a ham for nearly twenty years, but mostly I have operated on VHF and UHF, primarily packet, with a focus in the last few years on satellites. Recently I began to feel the desire to operate CW on HF once again. I am old enough to have passed the 5 WPM Novice test when I was first licensed, and I even made a CW QSO or two with my old Drake TR-4 before the blandishments of the faster digital modes lured me away. As I began to try to revive my rusty skills with CDs and computer training, I felt the desire for an iambic keyer. My current transceiver does have one built in, but I wanted to practice with a code practice oscillator before going on the air. There are plenty of transceivers still in use that do not have an internal keyer. Also, some hams would prefer a keyer over which they have more control than the built-in one allows.

There are plenty of commercial keyers on the market, ranging from designs based on the venerable Curtis integrated circuits, (first developed in the 1960s by John Curtis, K6KU) to microprocessor-controlled keyers with memory and many configurable options, but I am cheap. So, for some time I had been turning over in my mind the idea of designing and building my own keyer. A microcontroller-based design seemed like the way to proceed, to minimize the hardware cost and simplify the design.

Many ham homebrewers, however, shy away from including microcontrollers in their designs. Some microcontrollers require expensive programming devices for downloading binaries from the development computer to the chip. Others can be programmed over a serial or USB port from a computer, but require the purchase of an expensive software development environment if you want to write your own software. Sometimes there are licensing issues related to releasing software with any code written by third parties. Some vendors provide the software development kit free, yet the chips themselves are expensive. Yet other microcontrollers require the developer to program in assembly language. I wanted to avoid all those problems.

I have taught students to program microcontrollers for many years, using most of the designs produced by the major semiconductor manufacturers, as well as many products produced for the hobbyist market. Yet, until this project I had not been able to find a chip that met all the following requirements:

- Low chip cost and easy availability
- Low-cost and easy-to-use development environment
- Adequate variety and number of analog and digital inputs
- Low-cost or no special hardware required to download software to the chip

I recently became aware of the open-source Arduino microcontroller (**www.arduino.cc**).[1] The Arduino microcontroller attempts to do for hardware development what *Linux* has done for personal computer

[1]Notes appear on page 7.

operating systems. That is, all aspects of the design are freely made available to the developer, including the software development environment and compiler, as well as all schematics and specifications for the hardware. I thought it was worth a try to implement my keyer using the Arduino chip and software development environment. In this article I outline my design process, the decisions I made, and why I made them, as well as providing instructions for you to build your own version of my keyer. The keyer I ended up with is nice, and was produced at very low cost, but more importantly, I learned how to develop using this microcontroller, and my main goal in this article is to pass on to you, the reader, the capability of making your own designs using the Arduino, without spending much on the development tools.

The Arduino Platform

Arduino is based on the Atmel AVR family of 8-bit microcontrollers (**www.atmega.com**). There are a large number of configurations of the AVR chip, all with the same basic instruction set but differing in quantity of memory (both RAM and flash), number of digital and analog I/O pins, and available interfaces. Most of the available Arduino boards make use of the Atmel AVR Atmega 168 chip as the basic hardware platform. I first gained experience with this chip family when supervising student design teams developing control and data handling software for Auburn University's AS-1 Aubiesat student satellite project.[2] At the time, I used a gcc-based command-line toolset for prototyping and development, which did not meet my stated goals above.

What Arduino adds to the bare microcontroller is a run-time package that is downloaded onto the chip, and a development environment, available for free download, that compiles programs (called "sketches" by the Arduino creators) written in a *C*-like language into executables that can be downloaded to the chip over a USB or serial port. Massimo Banzi, the original developer of Arduino, has written a guide for novice programmers to get started with Arduino.[3]

Since the hardware designs are freely made available, various manufacturers produce Arduino-compatible development boards, and these are available from a number of vendors (for example, **www.sparkfun.com**, **www.makershed.com**). Virtually any Arduino board will work for the keyer I have designed. I used a board called Boarduino (**www.ladyada.net/make/boarduino/**), which I purchased from Makershed. All of these boards can be programmed with a cheap USB-to-5V RS232 cable (available from **http://ftdichip.com**) or with a simple serial cable if your computer has a serial port that can handle TTL voltage levels.

The Arduino makes available to the programmer a UART-style serial interface, 14 digital I/O pins with programmable internal pull-up/down resistors, and six analog input pins with A/D conversion. About 14.3 kbytes of program memory is available to the developer.

Arduino programs consist of two main parts: a *setup()* routine that runs once when the chip is powered up or reset, and a *loop()* routine that runs over and over again once the setup routine completes. These replace the *main()* routine typically found in a *C* program, and they can be augmented by any number of routines written by the developer. Typically the programmer must set up the event loop described above by her or himself, and it is a nice touch that it is handled for you by the Arduino software package. Anyone with any experience in programming *C*, *C++*, or *Java* should have little trouble figuring out how to program the Arduino, and even those without much experience should not find it hard — Banzi's book does not require any programming experience to read. (See Note 3.) For a more thorough introduction to the *C* programming language, I still recommend the book by the folks who invented that language.[4] Only a handful of Arduino specific functions are defined, all for controlling access to the hardware. These are well documented on both the Web site and in Banzi's book on the Arduino. Examples of such functions include:

pinMode(pinNumber,INPUT);
  // define the mode of a digital pin as input or output
x = digitalRead(pinNumber);
  // read from a digital input pin
digitalWrite(pinNumber, value);
  // write to a digital output pin
y = analogRead(pinNumber);
  // read from an analog input pin
time = millis();
  // return the number of milliseconds since reset
delay(numMilliseconds);
  // wait the specified number of milliseconds

With these tools in hand, I set out to design my iambic keyer.

Hardware Design

I first built a prototype on a breadboard, using the Boarduino mentioned above (an Arduino, TTL-level USB programmer header, two LEDs, and a 5 V regulated power supply on a small circuit board with 0.1 inch headers that easily plugs into a standard breadboard). The board can also take its power from the USB port. I used a Bencher dual-paddle key and a code practice oscillator I built many years ago to test it. My prototype is shown in Figure 1. At the left edge of the breadboard you can see the USB programming cable. The dual-paddle key is connected by the 3-conductor cable on the top edge, and the code practice oscillator by the orange wires on the right side of the photo. Power and ground are connected by the red and black wires along the bottom of the photo.

My goal was to use as few external components as possible, and to do as much as possible in software on the microcontroller. For the basic design, I added only a potentiometer to adjust the speed of the keyer, and a resistor, capacitor, and transistor to drive the keyer input on the transmitter (my design only supports open collector keying, and may not work on some older tube transmitters). Each of the digital input pins on the Arduino has internal, programmable, pull-up or pull-down resistors, so I did not need to add additional components to keep the inputs from floating. I wired the two paddle switches to two of the digital input pins, pulled up to +5 V when the switches were open. The schematic is shown in Figure 2.

Software Design, Version One

I wanted a software design that was also simple and easy to understand, since part of my purpose with this article is to introduce first-timers to microcontroller development. The listing in Figure 3 shows how simple a basic keyer is to program.

At the beginning of the program (lines 9-12) I define four constant values, which give symbolic names to the pins on the Arduino chip that my program will use (2 inputs from the paddles, one input from the speed potentiometer, and one output to the transistor that drives the key input on the transmitter). I also define some local variables on lines 14-17: two variables to hold the length of a dot at the current speed setting (*dotLength*, in milliseconds), a variable to hold the current state of each of the two paddle inputs (*dotVal* and *dashVal),* and a variable to hold the value read from the speed potentiometer (*speedDial*).

In the *setup()* routine, which runs once whenever the Arduino is powered up or reset, I put code to set the direction of input/output activity on the digital pins mentioned above, in lines 20-22. There is no need to set the direction on the analog input pin, since digital-to-analog output is not supported on the Arduino. I also activate the internal pull-up resistors on the paddle input pins on lines 27-28.

The *loop()* routine is very simple. Remember that the loop routine runs over and over forever, once the *setup()* routine completes. During each iteration of *loop()* the Arduino does the following steps:

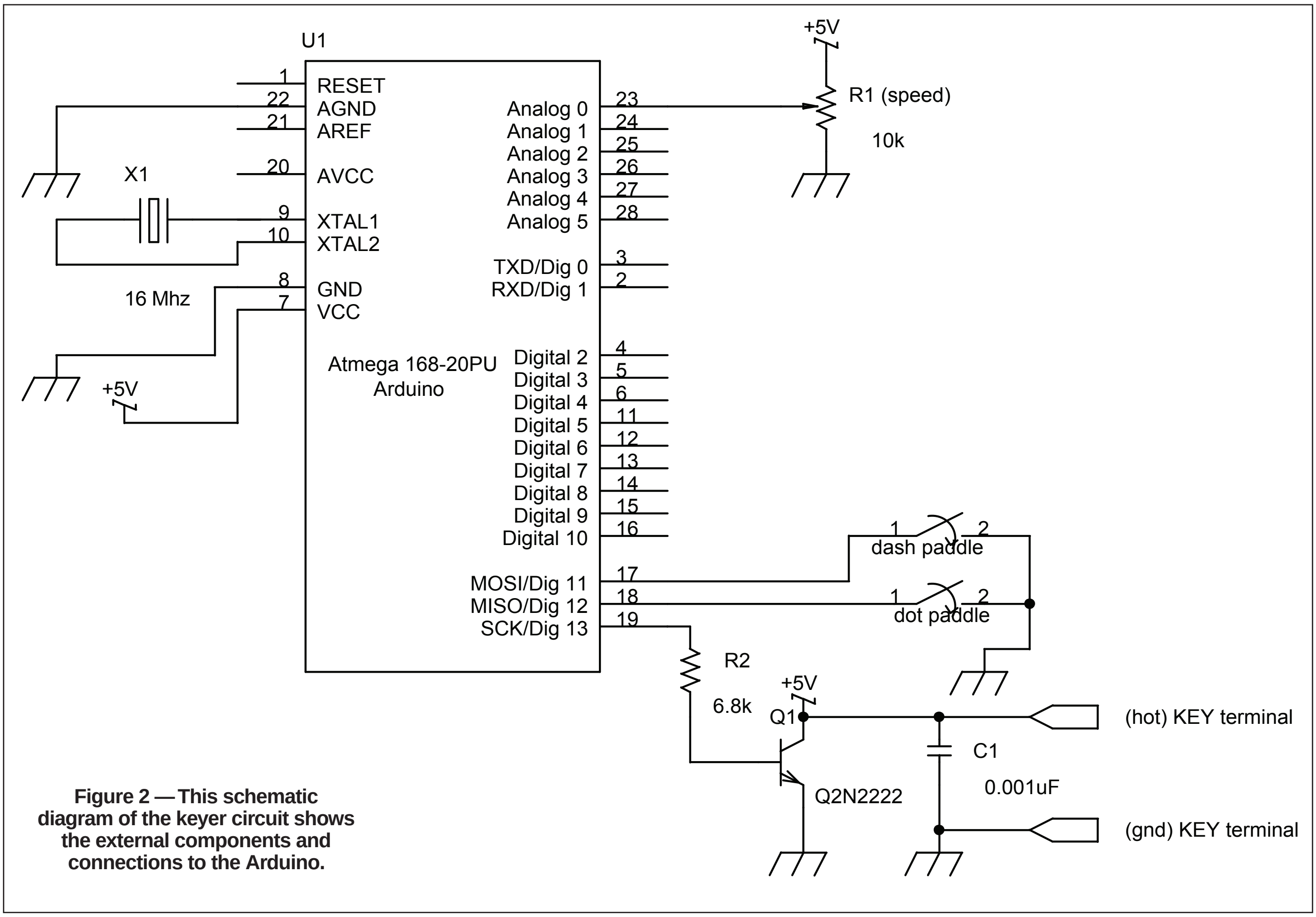


Figure 2 — This schematic diagram of the keyer circuit shows the external components and connections to the Arduino.

1. Read the state of the potentiometer and compute the current sending speed

2. Read the state of the two paddle switches into the variables.

3. Based on that state, either generate a dot, a dash, or dot followed by a dash with one dot-length delay in between.

4. Wait one dot-length delay.

I wrote three short functions of my own: a delay function that waited the length of one dot element, plus functions to generate a dot and to generate a dash. All three depend on the *delay()* function provided as part of the Arduino software development environment. This function causes the program to wait a specified time (in milliseconds) before proceeding. Both the dot and dash functions have the same structure: drive the keyer output pin high, wait a prescribed time, then drive the keyer output pin low.

Now, normally, when using a microcontroller to detect the position of mechanical switches such as the paddles on a key, one must be careful to *debounce* the values read. This is because the metal contacts typically do not go immediately from a non-conducting state to a conducting state when pushed together, but rather, because the metal surfaces are not perfectly smooth at a microscopic level, go back and forth between conducting and not conducting very rapidly, over the course of several milliseconds, before making solid contact. It is easy enough to code this into a program – one just waits a few milliseconds after detecting a change in the state of a switch before reporting it to the rest of the program. I have included code to do this in the second version of the program (see below), but in my testing of this keyer, I did not find it to be necessary, so I omitted it from the first version of the program for simplicity.

The total length of the source code for this version of the program is 64 lines of code, counting blank lines and headers. The compiled executable that is downloaded to the Arduino is 1444 bytes. I think it demonstrates how easy it is to get something running on an Arduino chip in a very short period of time, with minimal external hardware.

I was not happy to stop here, however, since this version does not support dot-insertion mode and dash-insertion mode, which most users of iambic keyers expect. These features allow a user who is sending a stream of dots, say, to briefly tap the dash paddle, and get one dash inserted immediately after the current dot completes, then go back to sending dots. This feature significantly cuts down on the number of hand motions required to send characters, so it is desirable. Adding that feature forced me to rethink the design of my software.

Software Design, Version Two

The problem with adding these features to the first version of the program is that the first version cannot check the state of the paddles (open or closed) while it is generating a Morse code element. It only checks the paddle switches when it is idle. In order to do dot- and dash-insertion, which most CW operators expect from an iambic keyer, I needed my program to be able to do other things while it was generating a dot or dash. In particular the program should be able to schedule things to happen in the future based on paddle presses and the current state of the keyer. In the second version of the program, listed in Figure 4, my thinking shifted from simply having the keyer read the paddles and generate a dot or dash at that instant, to having the keyer read the paddles very frequently, many times per millisecond, and schedule dots and dashes to happen in the future at

```
/*
 * Arduino iambic CW keyer
 * for Ham radio usage
 * Richard Chapman
 * KC4IFB
 * January, 2009
 */

#define SPEEDIN 0              // analog pin to read the speed value
#define DOTIN 11               // high -> dot paddle closed
#define DASHIN 12              // high -> dash paddle closed
#define KEYOUT 13              // drives the 2N2222 that closes the key connection

int dotLength;                 // length of a dot in milliseconds
int dotVal;
int dashVal;
int speedDial;
void setup()                   // run once, when the sketch starts
{
  pinMode(KEYOUT, OUTPUT);     // sets modes on the i/o pins
  pinMode(DOTIN, INPUT);       //
  pinMode(DASHIN, INPUT);
  // no need to set SPEED pin as input since it is analog
  digitalWrite(KEYOUT, LOW);   // initially the key is open

     // activate internal pullup resistors
   digitalWrite(DOTIN,HIGH);
   digitalWrite(DASHIN,HIGH);
}

void loop()                    // run over and over again
{


                               // get the speed
   speedDial = analogRead(SPEEDIN);
   dotLength = 1200 / ( 5 + (speedDial/64));
   dotVal = digitalRead(DOTIN);
   dashVal = digitalRead(DASHIN);

   if ((dotVal == LOW) && (dashVal == HIGH)){ // dot only
       sendDot();
   } else if ((dashVal == LOW) && (dotVal == HIGH)) {       // dash only
       sendDash();
   } else if ((dashVal == LOW) && (dotVal == LOW)) { // both pressed
       // iambic
       sendDot();
       dotDelay();
       sendDash();
   }  else {                   // nothing pressed
   }
   dotDelay();
}


void sendDot() {
  digitalWrite(KEYOUT,HIGH);
  delay(dotLength);
  digitalWrite(KEYOUT,LOW);
}

void sendDash() {
  digitalWrite(KEYOUT,HIGH);
  delay(3*dotLength);
  digitalWrite(KEYOUT,LOW);
}

void dotDelay() {
  delay(dotLength);
}
```

Figure 3 — The program code for the simplest version of the keyer is listed here. The program code file is available for download from the ARRL *QEX* Web site.[5]

certain times, based on the current, past, and future states of the keyer. I took advantage of a built-in Arduino function, *millis()*, which returns the time in milliseconds since the current program began executing to do this. The program always keeps track of the current time, and when it schedules an event to happen in the future, it also keeps track of the time at which that event should end.

Let's look at the code in Figure 4. Starting at the top again, I added some variables to keep track of both the previous and current condition (open or closed) of the dot and dash paddles, plus variables to keep track of the state of the keyer (by "state" I mean what Morse element it is generating at that time). I defined four states for the keyer: *IDLE*, *DOT*, *DASH*, or *DELAY*. I then added variables for the current state (*currElt)* and for the previous and next states (*lastElt* and *nextElt*, respectively). Each of these variables could take any one of the four state values defined above. I also added a variable to schedule the end of the Morse code element currently being generated (*currEltEndTime)* and a variable for keeping track of the current time (*time)*. I modified the *setup()* routine to initialize the state variables to *IDLE* and set the time.

The hard part of developing this program was making sure that the right action was taken in all situations, which led to some "if-then" logic that I cannot honestly say is entirely simple, but I think this is because the actual situation is a bit complicated. The *loop()* routine required the most significant modification. In this version of the program, the loop will iterate as fast as possible, rather than waiting for a Morse element to complete before continuing. Each time through the loop, the time is checked, and then one of four sets of instructions is carried out, based on the current state of the keyer.

Let's start with the simplest case to understand, in which the current state of the keyer is *IDLE* — that is, no element is being generated, no paddles are pressed, and nothing is scheduled for the future. This is in lines 94 to 111 of Figure 4. If a paddle is pressed, there are three sub-choices based on whether the dot paddle, dash paddle, or both are pressed:

1. If the dot or dash paddle (only one) is pressed, then set the current state to be DOT or DASH, respectively, and set the appropriate time for that dot or dash to end.

2. If both paddles are pressed, and the next element is not yet defined, set the current element to be a DOT, and the next element to be a DASH (iambic mode).

Note that the last case here is extremely rare — the keyer is checking hundreds of thousands of times a second, and it is very unlikely that the user will grab both paddles at exactly the same microsecond. Users instinctively grab, say, the dot paddle first by a discernible fraction of a second if they want the first element to be a dot.

If the current state of the system is that the keyer is generating a dash (or dot, if you reverse the words dot and dash in the pseudocode below), then the only event you care about is if the *other* paddle, that is, the dash paddle is pressed, meaning that the user wants a dash inserted after the current dot ends. Most keyers only look one element into the future. In other words, if somehow the user is fast enough to press and let go of the dot paddle twice before the current dash ends, still only one dot is inserted after the current dash ends (dial down the speed and try it if you have a commercial keyer). Another issue that needs to be handled when the system is generating a dot or dash is to end the current dot or dash when the appointed time (in variable *currEltEndTime)* is reached. Finally, if a dot or dash is being generated, the keyer output must of course be closed, to actually send the dot or dash to the radio. Formalizing this, we get the following summary of what happens in lines 72 to 93 of Figure 4:

1. If the dot paddle is also pressed, and the next element is not yet defined, schedule a dot to happen next (dot-insertion).

2. If the current time has reached the time to end the current dot, schedule it to end on the next iteration of loop and for a dot-length delay to happen next.

3. Set the keyer output high to keep sending the current dot.

Finally, the system may be waiting for the mandatory dot-length delay between Morse elements after generating a dot or dash (the state is *DELAY)*. We keep track of how long this delay should last in the variable *currEltEndTime*, as with a dot or dash element. Of course we want the keyer output to be open during this time. We also have to notice if the user presses a dot or dash paddle during the delay, and schedule it to happen after the delay ends. Then, since an element is ending, we need to shift the contents of variable *nextElt* into *currElt*, and so on. The next element will be started on the next element of the loop automatically. This code can be found in lines 112 to 131 of Figure 4:

1. If the current time has reached the time when the delay should end, and a next element has already been scheduled to follow the current delay, shift the state variables accordingly so that the next element will begin the next time through the loop.

2. If either paddle is pressed during the delay, and a next element has not been scheduled, set *nextElt* accordingly (dot or dash insertion).

3. Drive the keyer output low for the current delay.

Summary

The length of the code increased to 136 lines, still a very short program, as a result of the changes required to add dot and dash insertion, but I think most operators will welcome the new feature. The size of the downloaded executable increased to 2144 bytes. Once I had done this, I was as happy with the code that this keyer generates as I am with the code generated by a commercial keyer based on the Curtis chip, to which I compared my keyer.

Further Possibilities

There are additional features that can be added to the basic iambic keyer at very low additional expense. We can begin to really see the advantage of using a microcontroller over discrete logic. Some possible features include:

- Switchable iambic mode Curtis type A and type B operation
- Generation of a variable frequency sidetone
- Adding message memory
- Semiauto ("bug") operation

I hope you enjoy building and programming your keyer. Please share your modifications and new features with *QEX* readers.

Notes

[1]Daniel Jollife, "Arduino Fever," *Make: Technology on Your Time,* O'Reilly Media, Sebastapol, CA. Vol. 7, pp 52-53.

[2]Richard Chapman, KC4IB, Jean-Marie Wersinger, KI4YIU, Thor Wilson and John Klingelhoeffer, WB4LNM, "Aubiesat 1: A Student-Designed Cubesat Developed at Auburn University," *Proceedings of 2008 AMSAT Symposium*, October 24-26, Atlanta, GA.

[3]Massimo Banzi, *Getting Started With Arduino,* O'Reilly Media, Sebastapol, CA, 2008.

[4]Brian Kernighan and Dennis Ritchie, *The C Programming Language,* 2nd edition, Prentice Hall, 1988.

[5]The keyer program code files are available for download from the ARRL *QEX* Web site. Go to **www.arrl.org/qexfiles** and look for the file **9x09_Chapman.zip**.

Richard Chapman has been a licensed Amateur Radio operator since 1989. He is a an associate professor in the Department of Computer Science and Software Engineering at Auburn University, and an affiliate faculty member in the Department of Electrical and Computer Engineering. He is director of the Center for Innovation in Mobile, Pervasive, Agile Computing Technology (IMPACT) in Auburn's Samuel Ginn College of Engineering. His interests in Amateur Radio have mostly focused on digital packet and network communications, including particularly satellite operations. He is currently serving as control and data handling advisor for the AUBIESAT-1 project, a student designed and built satellite being developed at Auburn University.

```
/*
 * Arduino iambic CW keyer v2.0
 * for Ham radio usage
 * Richard Chapman
 * KC4IFB
 * February, 2009
 */


// I/O pin numbers
#define SPEEDIN 0              // analog pin to read the speed value
#define DOTIN 11               // high -> dot paddle closed
#define DASHIN 12              // high -> dash paddle closed
#define KEYOUT 13              // drives the 2N2222 that closes the key connection

// state of the machine
#define IDLE 0                 // doing nothing
#define DASH 1                 // playing a dash
#define DOT  2                 // playing a dot
#define DELAY 3                // in the dot-length delay between two dot/dashes


int dotLength;                 // length of a dot in milliseconds
int dotVal;                    // value of the dot paddle this sycle of main loop
int dashVal;                   // value of the dash paddle this cycle of main loop
int oldDotVal;                 // value of dot paddle last cycle
int oldDashVal;                // value of dash paddle last dial
int speedDial;                 // raw value read from the potentiometer for speed
int currEltEndTime;            // what time did the current element start sounding (in milliseconds
                               since powerup)
int currElt;                   // sate of what the keyer output is sending right now
int nextElt;                   // state the keyer will go into when current element ends
int lastElt;                   // previous state of the keyer
int time;

void setup()                   // run once, when the sketch starts
{
  pinMode(KEYOUT, OUTPUT);     // sets modes on the i/o pins
  pinMode(DOTIN, INPUT);
  pinMode(DASHIN, INPUT);
  // no need to set SPEED pin as input since it is analog
  digitalWrite(KEYOUT, LOW);  // initially the key is open

  // activate internal pullup resistors
  digitalWrite(DOTIN,HIGH);
  digitalWrite(DASHIN,HIGH);

  // initialize the state variables
  lastElt = IDLE;
  currElt = IDLE;
  nextElt = IDLE;
  time = millis();
  currEltEndTime = time;

}

void loop()                    // run over and over again
{

   // get the speed (need to do this every iteration)
   speedDial = analogRead(SPEEDIN);
   dotLength = 1200 / ( 5 + (speedDial/64));

   // read the paddles
   oldDotVal = dotVal;         // save the old values to detect a transition, so you can debounce
   oldDashVal = dashVal;
   dotVal = digitalRead(DOTIN); // read the current values of the paddles
   dashVal = digitalRead(DASHIN);

   // short delay for a debounce  -- turns out not needed
   //if ((oldDotVal != dotVal) || oldDashVal != dashVal) {
   //    delay(5);
```

Figure 4 — This program code shows the author's revised keyer version, which checks the paddle input status while generating code elements. This program code file is available for download from the ARRL *QEX* Web site. See Note 5.

```
   //}

   //get the current time
   time = millis();

   switch(currElt) {          // cases based on what current state is
   case DASH:
      if ((dotVal == LOW) && (nextElt == IDLE)) {      // going from dash to iambic mode
           nextElt = DOT;
      }
     if (time >= currEltEndTime) {      // at end of current dash
       lastElt = DASH;        // a delay will follow the dash
       currElt = DELAY;
       currEltEndTime = time+dotLength;
     }
     digitalWrite(KEYOUT,HIGH);  // close the keyer output while the dash is being sent
     break;
   case DOT:
     if ((dashVal == LOW) && (nextElt = IDLE)) {       // going from dot to iambic mode
           nextElt = DASH;
     }
     if (time >= currEltEndTime) {      // at end of current dot
       lastElt = DOT;         // a delay will follow the dot
       currElt = DELAY;
       currEltEndTime = time+dotLength;
     }
     digitalWrite(KEYOUT,HIGH);  // close the keyer outout while the dot is being sent
     break;
   case IDLE:                 // not sending, nor finishing the delay after a dot or dash
     if ((dotVal == LOW) && (dashVal == HIGH)) {       // only dot paddle pressed, go to DOT mode
          lastElt = IDLE;
          currElt = DOT;
          currEltEndTime = time + dotLength;
     } else if ((dotVal == HIGH) && (dashVal == LOW)) {// only dash paddle pressed, go to DASH mode
          lastElt = IDLE;
          currElt = DASH;
          currEltEndTime = time + 3*dotLength;
     } else if ((dotVal == LOW) && (dashVal == LOW) && (nextElt == IDLE)) {
          // if both paddles hit at same time (rare, but happens)
          lastElt = IDLE;
          currElt = DOT;
          nextElt = DASH;
          currEltEndTime = time + 3*dotLength; // it is an iambic keyer, not a trochaic keyer
     }
     digitalWrite(KEYOUT,LOW);// keyer output is open in IDLE mode
     break;
   case DELAY:                // waiting for a dot-length delay after sending a dot or dash
     if (time >= currEltEndTime) {      // check to see if there is a next element to play
       currElt = nextElt;
       if (currElt == DOT) {
            currEltEndTime = time+dotLength;
       } else if ( currElt == DASH) {
            currEltEndTime = time+3*dotLength;
       }
       lastElt = DELAY;
       nextElt = IDLE;
     }
     // during the delay, if either paddle is pressed, save it to play after the delay
     if ((lastElt == DOT) && (dashVal == LOW) && (nextElt == NULL)) {
        nextElt = DASH;
     } else if ((lastElt == DASH) && (dotVal == LOW) &&(nextElt == NULL)) {
        nextElt = DOT;
     }
     // key output is open during the delay
     digitalWrite(KEYOUT,LOW);
     break;
   default:
     break;
   }


}
```

W. G. Moneysmith, W4NFR

w4nfr@earthlink.net

A Homecrafted Duplexer for the 70 Centimeter Band

Do you need a duplexer for your 440 MHz repeater? This project may be just what you need

As a newly minted repeater trustee, I was faced with either buying or building a duplexer for the 70 cm band. So I chose the hard way! Not really, it was fun to build, once I got past the "black magic," it was not as difficult as I first thought.

Over the years, I have tuned and retuned many duplexers. Mostly, they were commercial types, so to build one from scratch was a new challenge. In researching this subject, I found information on 2 meter duplexers, but not much on UHF (440 MHz) models.

This unique design uses materials I had in my junkbox and items commonly available at a local hardware store. The BNC connectors are available at hamfests and via the Internet.

Last spring, when I changed out the traps on my Mosley TA-33 triband antenna, little did I know those old traps would get recycled into another Amateur Radio project! I tend to save most usable metal, and in this case the 2 inch diameter aluminum outer sleeves would be the basis for my homecrafted UHF bandpass / band reject duplexer project.

In addition to the aluminum sleeves left over from my Mosley TA-33 traps, I had two pair of 8 × 3 × 2 inch aluminum angle brackets in my junk box. Those angle brackets formed the ends for the duplexer project.

While searching around the shop, I came up with the five BNC "T" connectors and eight BNC Male crimp on connectors for the feed lines. I had to make a trip to the hardware store to purchase a 5 foot long, ½ inch copper pipe and 17 pieces of 12-inch-long "all thread rod," nuts, and lock washers. Everything else was in my junk box.

[Editor's Note: When we reviewed this article for publication acceptance, several Technical Advisors commented against using dissimilar metals in the tuned cavities, and that any *ferromagnetic* metals in the tuned cavities would greatly reduce the unloaded *Q* and increase losses in the duplexer. "Even a

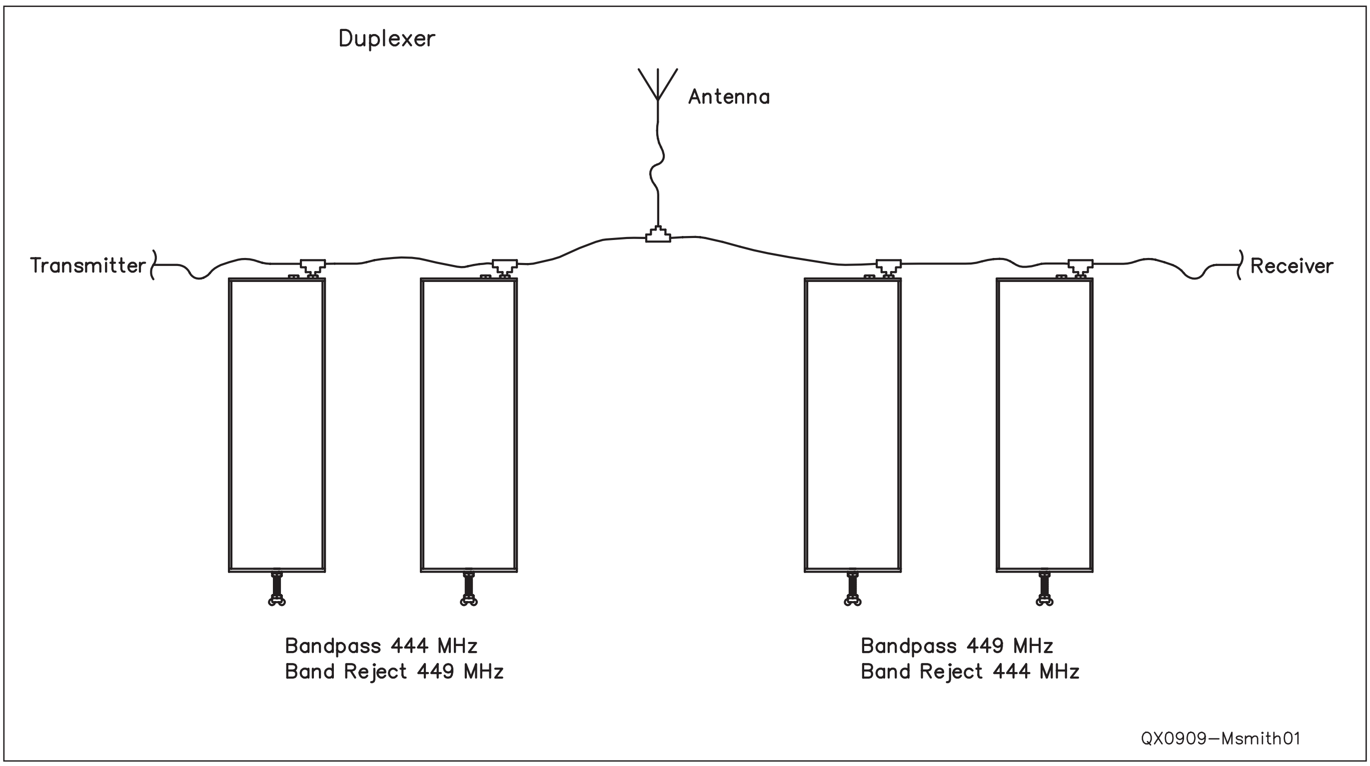


Figure 1 — By using a pair of duplexer "cans" on the transmitter side and another pair on the receiver side, a single antenna can be used with a repeater system.

single steel nut can have a major effect," said one Technical Advisor. Copper pipe could be used for the outer sleeves and brass all thread rod, nuts and lock washers are available from a number of suppliers, including Small Parts (**www.smallparts.com**). The author acknowledged that improved performance may be possible by using all copper or brass components, and suggested that 3 inch copper pipe could be used for the outer sleeves. The length of the sleeves is probably not extremely critical. The spirit of the project was to use available junkbox parts and readily available hardware, however. The project is presented here as it was constructed. Interested builders may want to consider the increased cost of copper pipe and brass hardware for possible performance improvements.]

A duplexer provides the means for simultaneous operation of a repeater station having separate transmit and receive frequencies, when using a common antenna. Figure 1 shows a simple block diagram of the station.

I decided to construct a single-loop, series-resonant, notch/bandpass duplexer. This design, shown in Figure 2, uses only one loop and a series capacitor to adjust the notch frequency above or below the passband. A single BNC connector with an external T is used for the feed line connections. Each of the four cavities is tuned with a 1.5 inch diameter circuit board mounted on one end of a threaded rod. Each rod then passes through a threaded hole at the bottom, grounded, end of the cavity and is secured using a lock nut.

The bandpass circuit consists of a 6 inch copper pipe soldered to one end of each cavity and the pass frequency is tuned by adjustment of the circuit board capacitor at the grounded end. The notch frequency is adjusted by a small piston capacitor at the coupling loop, in the series resonant input circuit.

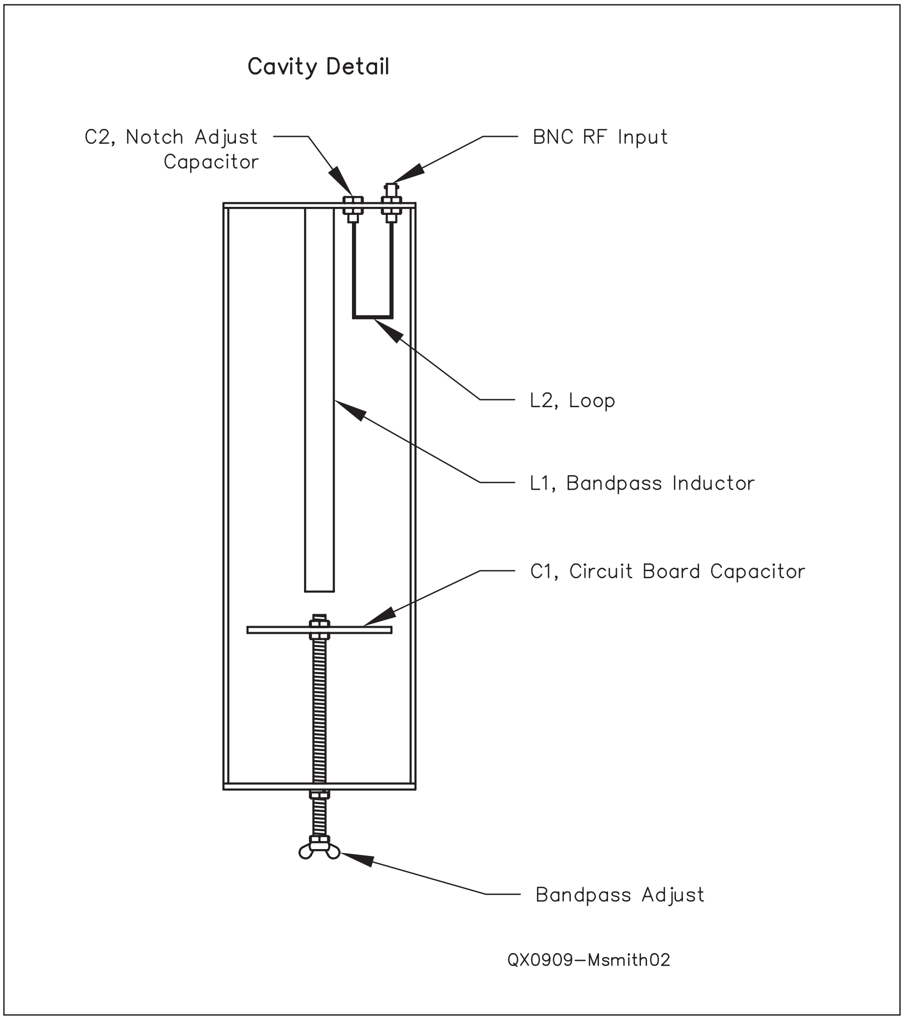


Figure 2 — This drawing shows the internal workings of a duplexer cavity.

Construction

After all the parts are acquired, begin by laying out the four copper-clad circuit board pieces (3 × 3.75 inches) shown in Figure 3. Drill the holes for the female BNC input connector, piston capacitor and "all thread" rods. It is a good idea to use a center punch to mark the hole centers before drilling. Cut out the brass pieces for the four loops, as shown in Figure 4.

A drill press is handy, but a hand drill will work. I used the first circuit board as a pattern for the other three. Once the boards are drilled, lay them on your angle brackets and mark those holes to be drilled, so all the threaded rods will line up for final assembly.

Table 1
Parts List

Quantity	*Description:*
16	"All thread" rods, 12 inches long, no. 8-32.
1	"All thread" rod, 12 inches long, no. 10-24.
4	Aluminum tube, 2 inch diameter, 8 inches long.
32	Nuts, no. 8-32.
16	Nuts, no. 10-24.
20	Split washers, no. 10 size.
4	Screws, ½ inch long, no. 10-24.
4	Lock nuts, no. 10-24.
4	Aluminum angle brackets, 8 x 3 x 2 inches, 1/8 inch thickness.
5	BNC, T connectors.
8	BNC, male, Amphenol RFX, crimp on connectors.
4	C1, circuit board discs, 1.5 inch diameter each, 1/8 center.
4	Circuit boards, single sided copper clad, 3" x 3.75".
4	C2, small piston capacitor, 4-15pf, VOL AP10.
4	L1, copper pipe, 6" long, ½" diameter, cut with tubing cutter.
4	L2, brass or copper flashing for loop construction.
1	RG-142, Mil-C-17, double shielded coaxial cable, 24" long.

Construct four each cables, 6" long. Terminate with BNC male RFX connectors.

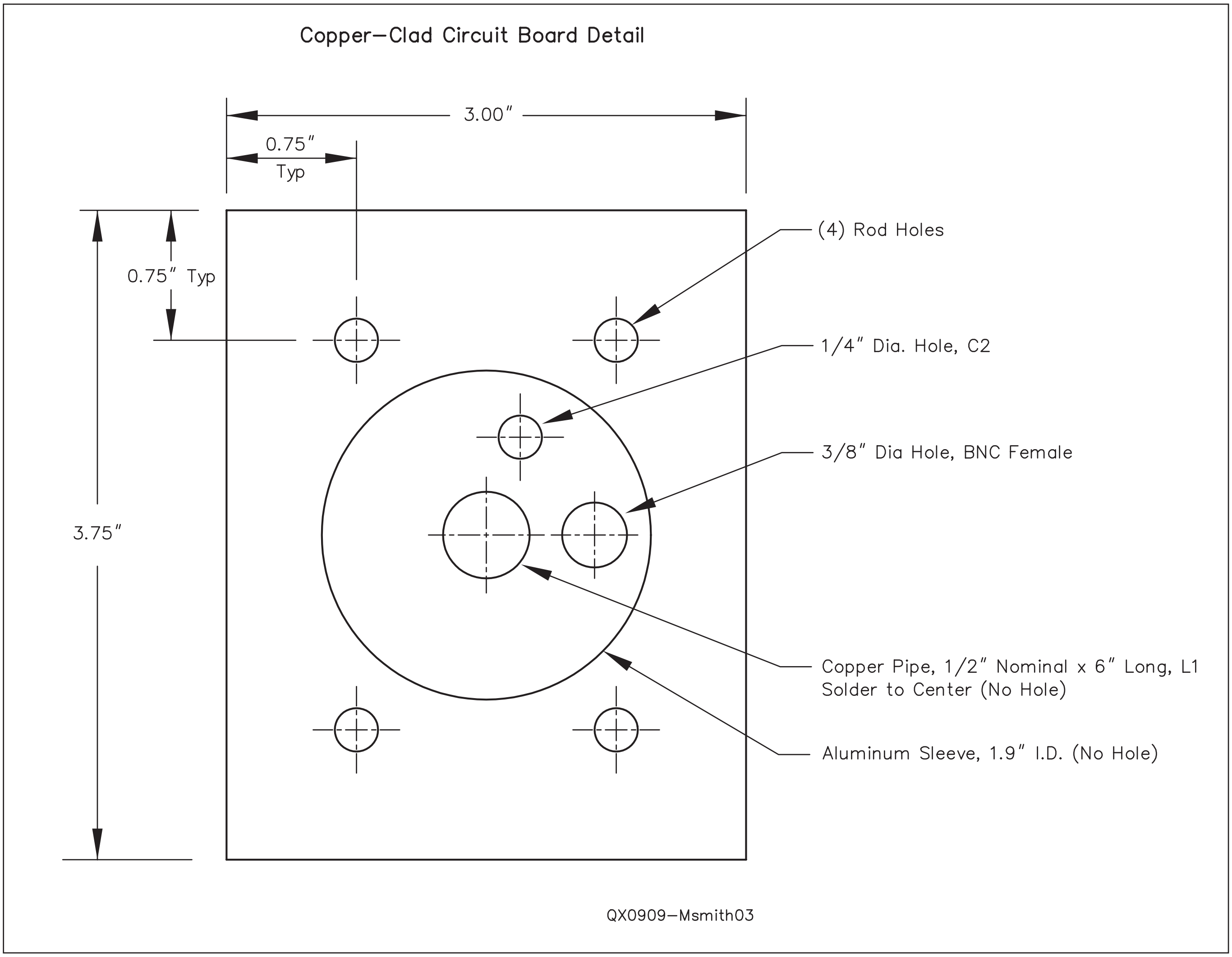


Figure 3 — This illustration shows the construction details for the copper-clad circuit board material used to form the end caps for the duplexer cavities.

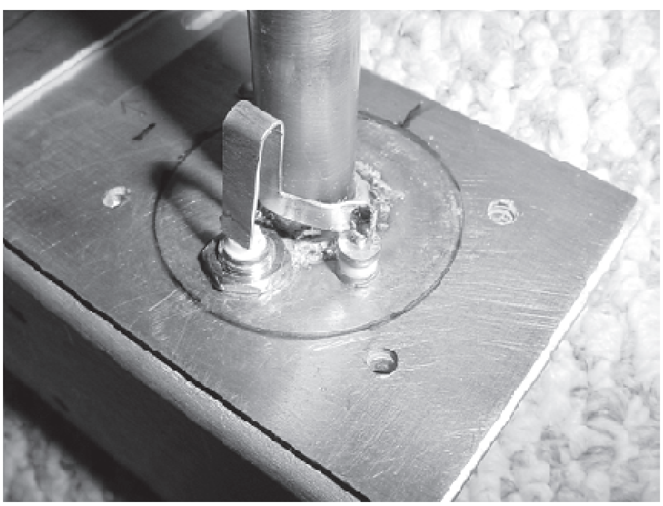


Photo A — This close-up view shows the assembly of one copper-clad circuit board material end plate with the copper pipe for L1 soldered to the circuit board, the BNC connector, copper flashing loop for L2 and the notch-adjust capacitor, C2, installed.

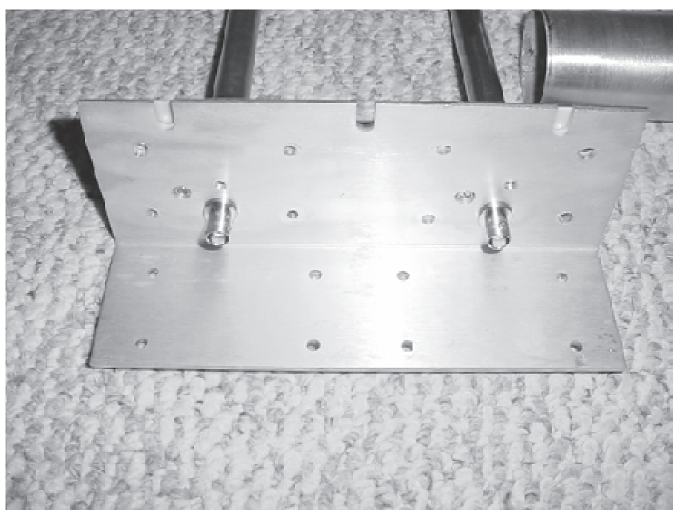


Photo B — A pair of end plate and copper pipe assemblies are attached to one piece of the aluminum angle bracket.

The brackets and rods give the duplexer its mechanical strength.

Clean the copper boards with a rotary brass wire brush on a drill to shine up the surface. Next, cut four 6 inch long pieces of ½ inch copper pipe and clean with a wire brush until shiny bright. Use a tubing cutter to make sure the tubes are cut with perfectly square ends. Be careful not to squeeze the tubing out of round.

Using a propane torch, solder each tube to the center of a piece of the circuit board material. Do not over heat or burn the circuit board material. Make sure each piece of pipe is vertical and perpendicular to its circuit board. Use very little solder, but enough to keep the tube solid to the circuit board.

After the copper pipes have cooled off, mount the small piston capacitors on the circuit boards. Make sure the notch-tuning access holes in the end brackets are drilled 5⁄16 inch diameter, so the circuit board lays flat against the bracket. Next, mount the BNC female input RF connectors, the end brackets and circuit boards.

Make the input brass loops, as shown in Figure 4. Solder the loops to the BNC input and to the notch piston capacitor as shown in Photo A. Make sure the loop is spaced ⅛ inch from the ½ inch copper pipe. Photos B and C show one pair of copper end plates and copper tubing assemblies attached to one of the aluminum angle brackets. Photo D is a side view of this assembly.

The circuit board tuning capacitor shown in Figure 5 is made using a 1.5 inch hole saw to cut out four discs of copper clad circuit board material. Drill a small ⅛ inch diameter hole in the center of each, and mount it on the end of a 3 inch long piece of no. 10-24 "all thread" rod. Tap and thread a small hole in the angle bracket. Thread the capacitor

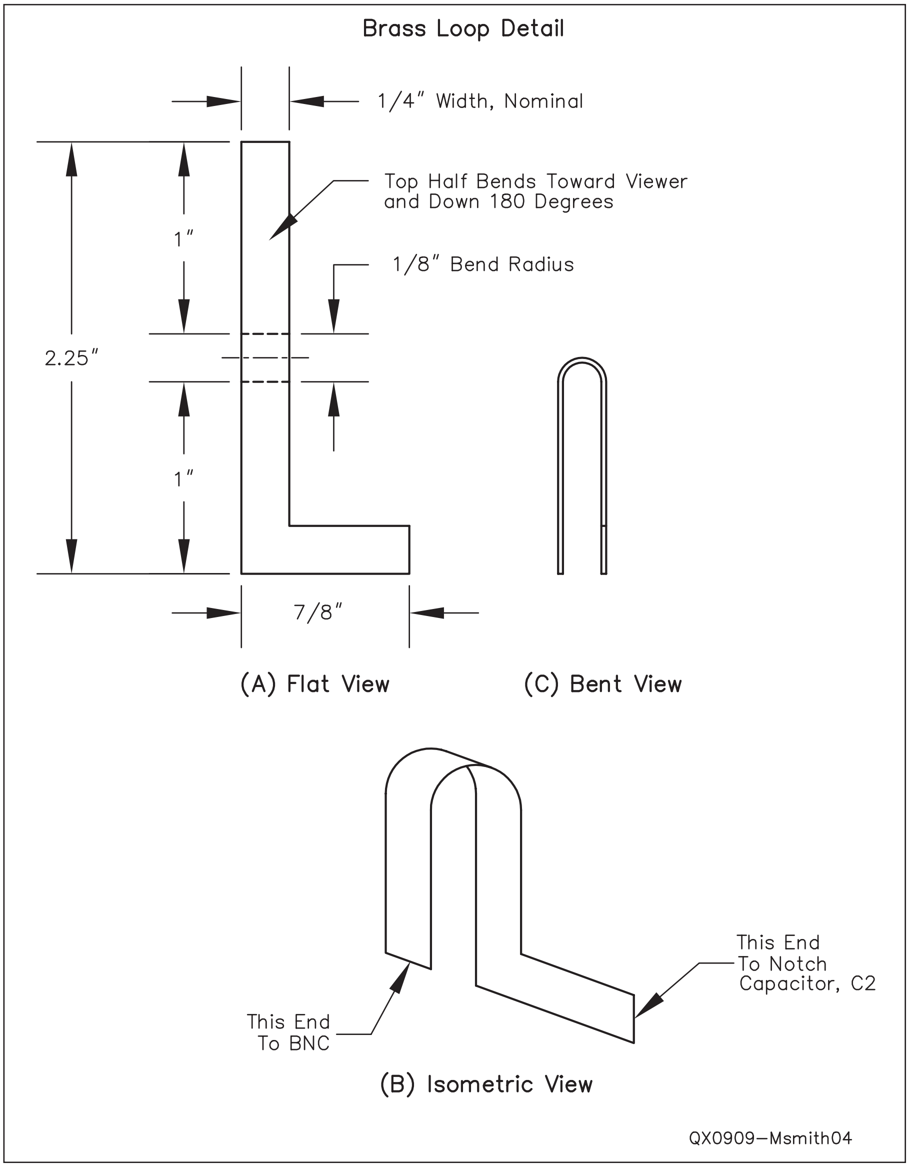


Figure 4 — Use brass or copper flashing to form loop L2 for each cavity.

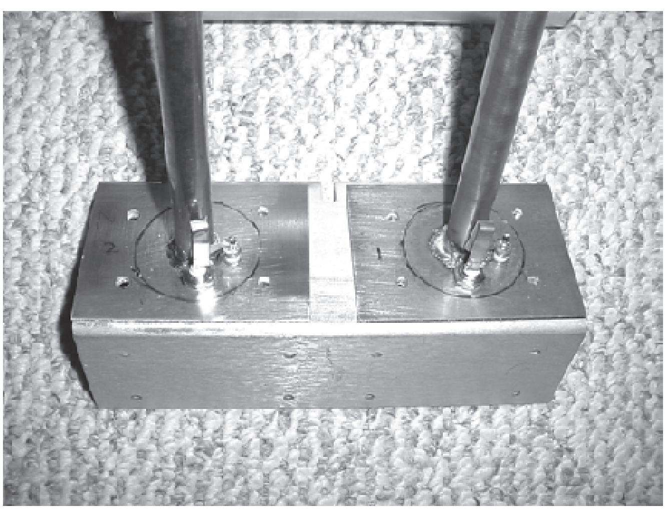


Photo C — This photo shows a completed end assembly for one pair of the duplexer cavities.

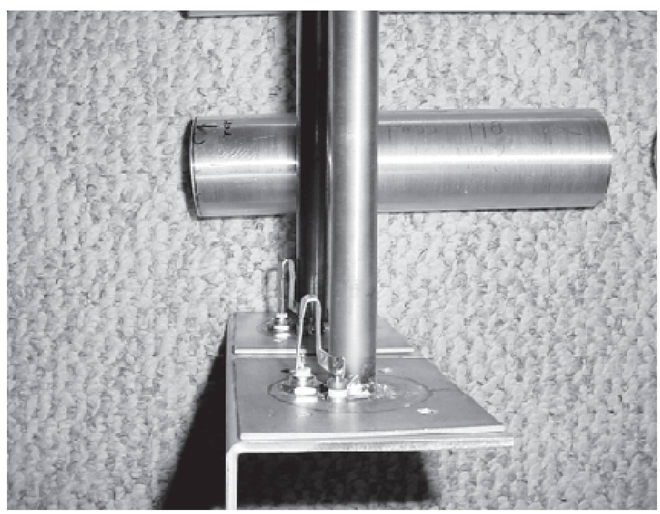


Photo D — Here is a side view of the completed end assembly.

rod into this hole and lock it in place with a no. 10-24 nut and no. 10 lock washer.

The 2 inch diameter aluminum tubes were originally 12 inches long, and they are cut to a length of 8 inches for this project. Make sure to cut squarely to ensure a nice tight fit against the brackets and circuit boards.

Mechanical Assembly

You are now ready to assemble the duplexer. Put two of the 8 inch aluminum sleeves over a pair of the ½ inch copper pipes, between end brackets. Secure the brackets with no. 8-32 "all thread" rods, using a no. 8-32 nut and lock washer on one end and a no. 8-32 nut on the other end. When completed, cut off the excess rod with a hacksaw. Center up the aluminum sleeves and tighten the nuts evenly until all are solid.

Tuning

I tune each cavity separately in the beginning, using a 5 W 440 MHz handheld transceiver and a dummy load with built-in wattmeter. A BNC T connector is secured to the BNC female on the cavity. Install a shielded RF cable from the transceiver to the T and place another RF cable from the T to the dummy load/wattmeter. Set

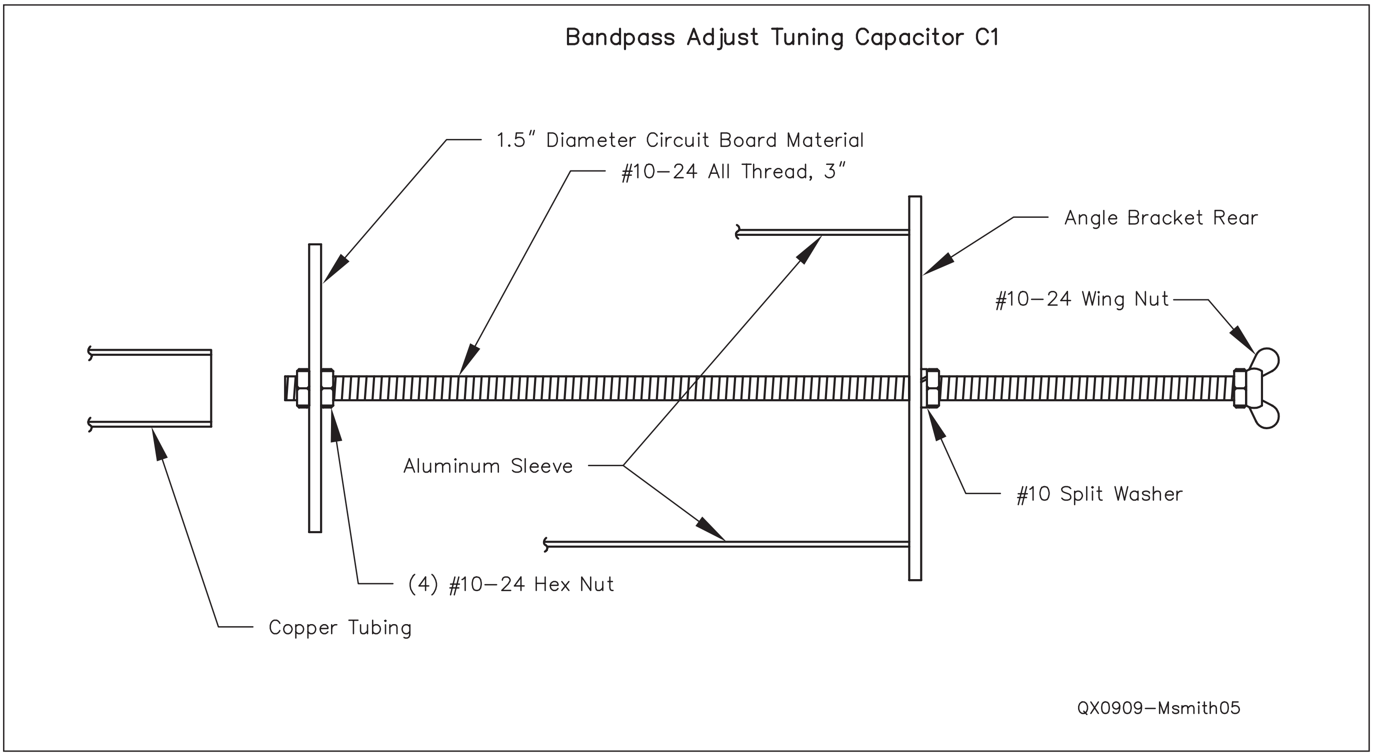


Figure 5 — Here are the construction details for the bandpass tuning capacitors used in each cavity.

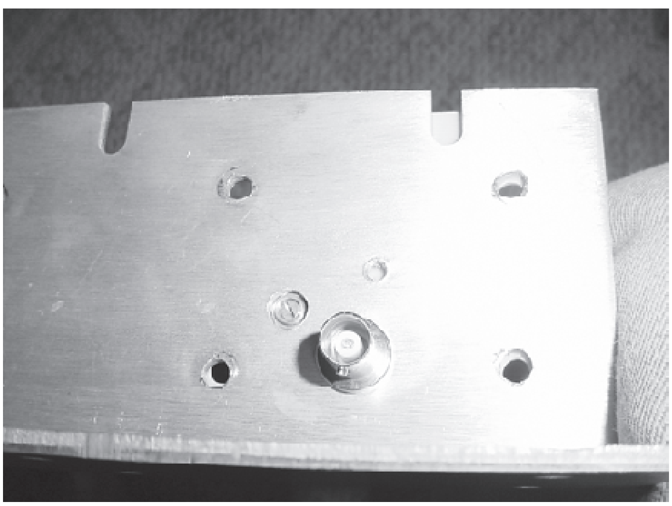


Photo E — This close-up view shows the RF input BNC connector and the notch adjust capacitor through holes in the angle bracket. The holes for the all thread rods are also visible.

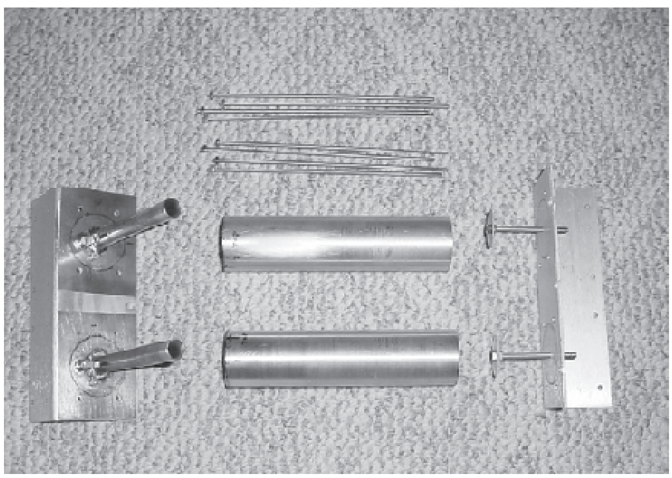


Photo F — The copper pipe assembly, aluminum tube "cans" and all thread rods are laid out ready for assembly. The bandpass adjustment capacitors have been installed in holes in the second aluminum angle bracket.

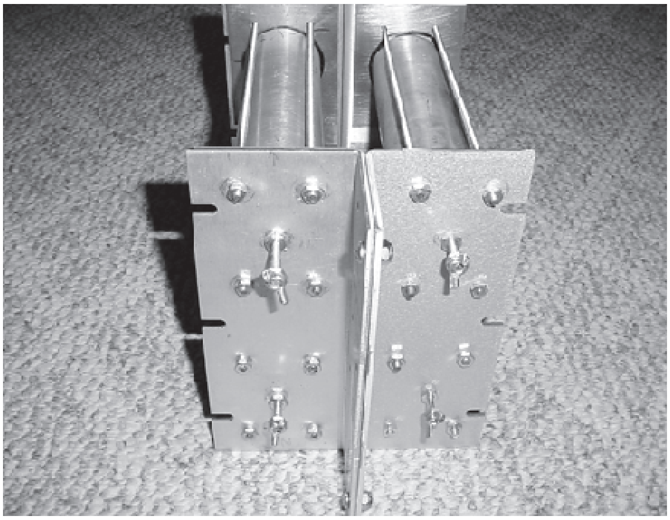


Photo G — This end view of the completed pair of cavities shows the bandpass tuning adjustment all thread rods through the angle brackets.

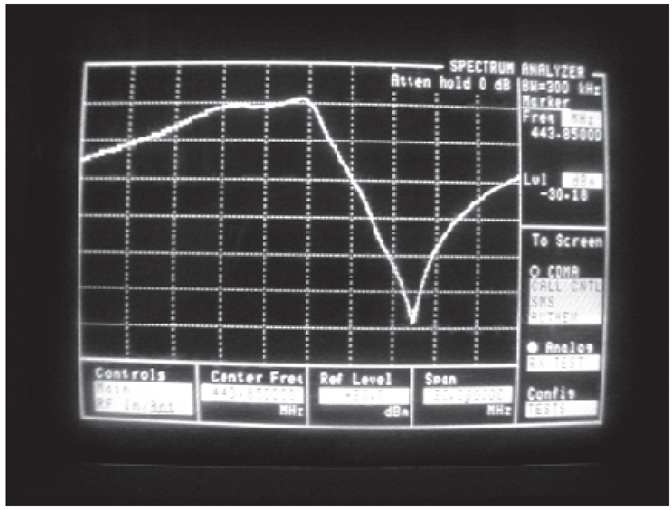


Photo J — This spectrum analyzer sweep shows the transmit frequency passband, centered at 443.85 MHz. Note that the received signal at 448.85 MHz is greatly reduced. The sweep was done using the two cavities on the transmitter side of the duplexer.

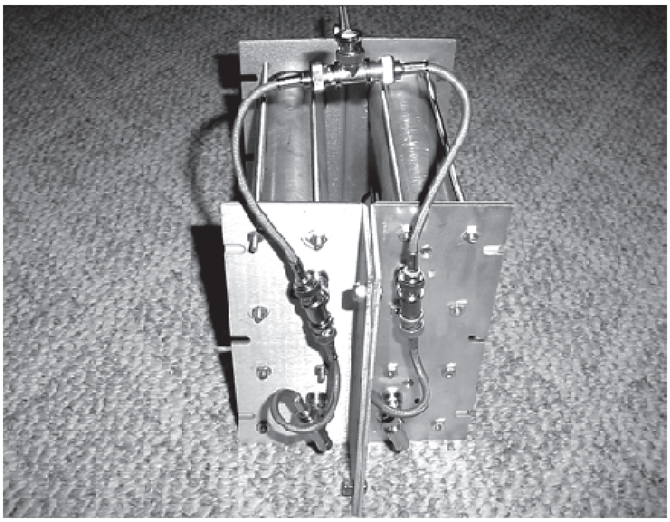


Photo H — Here is the completed duplexer assembly, including the interconnecting coaxial cables.

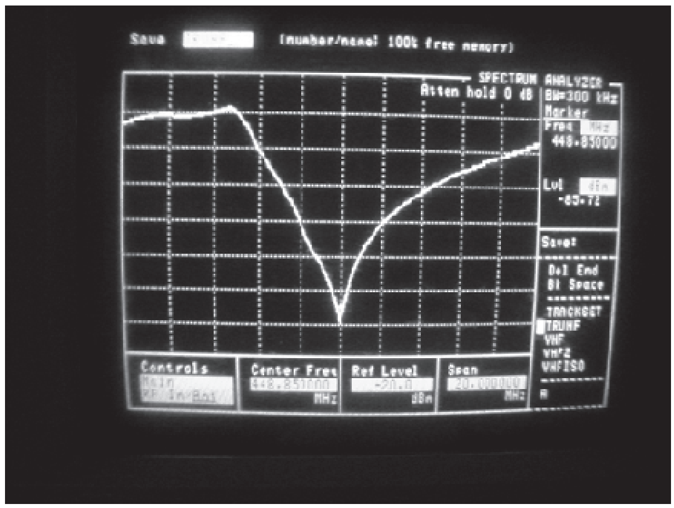


Photo K — This spectrum analyzer sweep shows the receive frequency notch, centered at 448.85 MHz, also using the two cavities on the transmitter side of the duplexer.

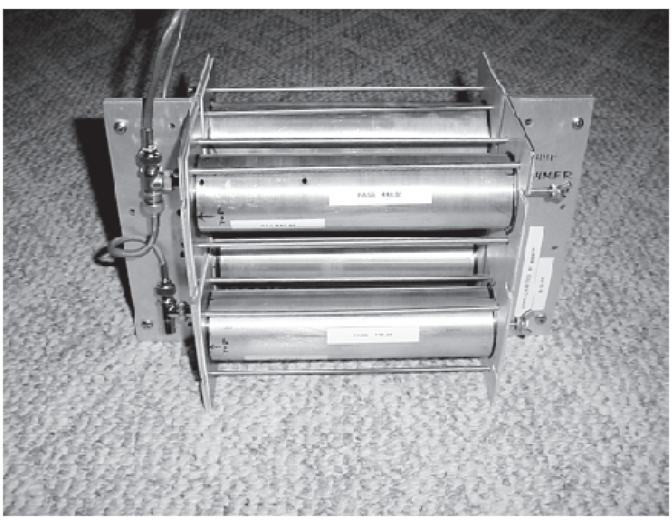


Photo I — This side view of the completed duplexer assembly shows all four duplexer cavities.

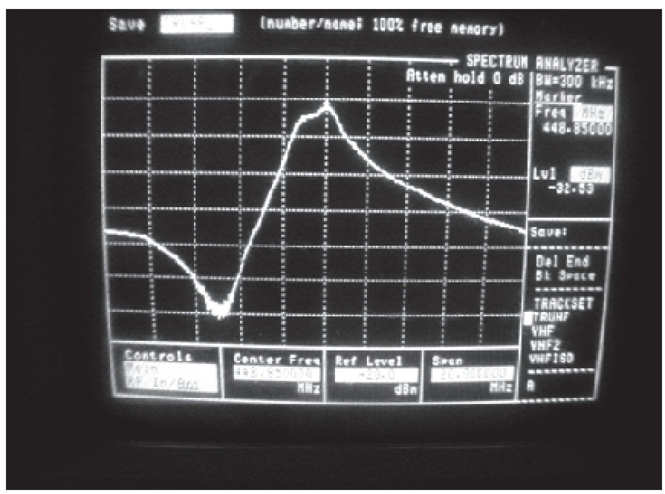


Photo L — The spectrum analyzer sweep of the receive frequency passband, centered at 448.85 MHz, using the two cavities on the receiver side of the duplexer.

the radio transmit frequency to the bandpass frequency (448.85 MHz in this case) and key the transmitter. Adjust the circuit board tuning capacitor rod on the end of the mounting bracket for maximum throughput as indicated on the wattmeter.

Next, change the handheld transceiver frequency to the notch frequency (443.85 MHz), key the handheld and tune the notch capacitor for a minimum reading on the wattmeter. Repeat this for the second cavity and set the assembly aside for now. Tune the other set of cavities in a similar manner, but reverse the passband and notch frequencies.

After all of the cavities have been roughed in, install T connectors on all cavities. Install quarter wavelength (6 inch) feed lines using RG-142 double shielded coaxial cable and BNC male connectors.

Both sides of the duplexer can now be mounted back to back using no. 10-24, ½ inch long screws and lock nuts. The cables from both sides can now be connected to the antenna BNC T connector. Recheck your passband and reject adjustments. Terminate the unused ports with a 50 Ω load. Touch up your bandpass and notch adjustments and this completes the project.

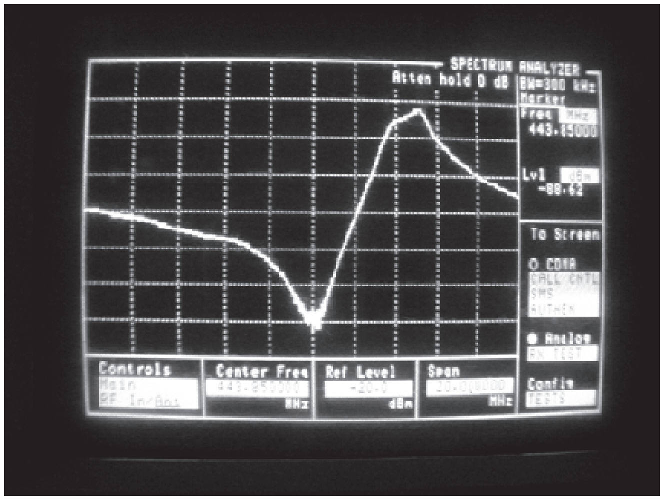


Photo M — This is the spectrum analyzer sweep of the transmit frequency notch, centered at 443.85 MHz, using the two cavities on the receiver side of the duplexer.

Test Results

I used a spectrum analyzer to record the results of frequency sweeps of the duplexer cavities. Photo J shows the transmit frequency passband sweep, while Photo K shows the receive frequency notch sweep. These photos are for the two cavities on the transmit frequency side of the duplexer.

Photo L shows the receive frequency passband sweep, while Photo M shows the transmit frequency notch sweep. These photos are for the two cavities on the receive frequency side of the duplexer.

W.G. Moneysmith, W4NFR, has been enjoying ham radio as an Extra class licensee since 1961. Bill enjoys all facets of radio, especially AM, CW, PSK-31, RTTY, Packet, DSSTV, QRP and chasing DX. He is an avid electronics and ham radio enthusiast, with a particular interest in antennas. He has written many technical articles and is a retired Electronics Engineer. Current activities include RV camping/traveling, gardening, wood carving, gold panning and metal detecting with my wife, Lydia.

Robert Zimmerman, VE3RKZ

54 Radford St, Hamilton, ON L8S 3E6, Canada; rkzimmerman@gmail.com

An Easily Erected 20 Meter Antenna for Emergency Use

This antenna hangs from a single support, so it is easy to set up for portable and emergency operations.

Antennas for emergency use have requirements beyond those for home use:

- Easily transported
- Quickly deployed
- Small footprint
- Omni-directional
- Preferably no ground radials
- Directly fed with 50 Ω coax

Additionally, one still wants good electrical performance. The 20 meter vertical dipole described here fills all these requisites while providing excellent electrical performance. As such, it will also be of interest to backpackers and QRP operators.

Theory

If a uniform current distribution is used on a dipole rather than a sinusoidal current distribution, a dipole length of only ¼ λ develops a full 50 Ω radiation resistance with wideband performance. This is much preferable to using a low radiation resistance antenna and a step-up transformer, which yields a high-Q (low bandwidth) antenna. A nearly uniform current may be easily realized with a ¼ λ wire by using top hat capacitors at both ends of the wire. Figure 1 shows an *EZNEC* model of the

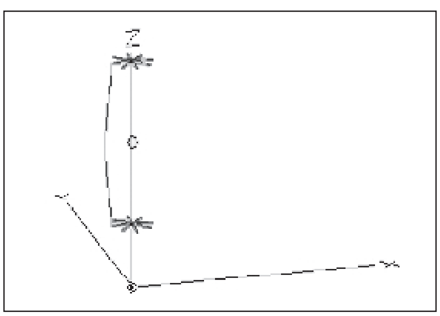


Figure 1 — This EZNEC model shows the nearly uniform current distribution along the length of the 20 m emergency antenna.

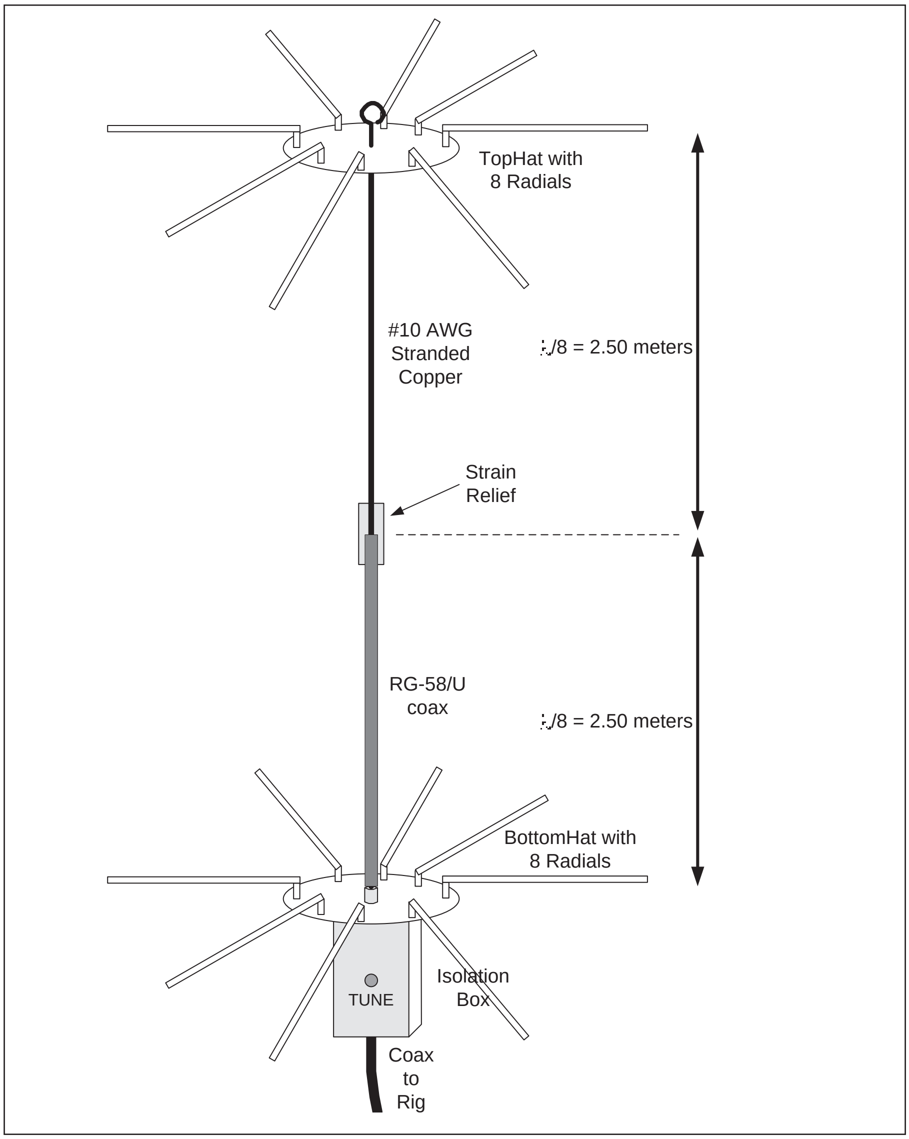


Figure 2 — This drawing illustrates the construction of the 20 m emergency antenna.

antenna construction. The computer model for this antenna is available for download from the ARRL *QEX* Web site.[1]

Construction

The overall antenna plan is shown pictorially in Figure 2. The dipole itself is ¼ λ, which is 5.0 meters for the 20 meter band. There is a top hat capacitor with lanyard at the top, and a "bottom hat" capacitor at the bottom, as well as an integral coaxial isolation box (described later).

The top hat and bottom hat capacitors are each made of eight extendable whip antennas (67 cm — 26⅜ inches — extended length), which collapse for easy carriage, as shown in Figure 3. *EZNEC* shows that eight radials perform about the same as a solid sheet-metal top hat. To make the system ultimately portable, the extendable whips may be removed using banana plugs. Figure 4 shows the banana plug soldered to base of the whip antenna. The whip antennas are available from Nutech Electronics for $5.00/each (16 are required).[2]

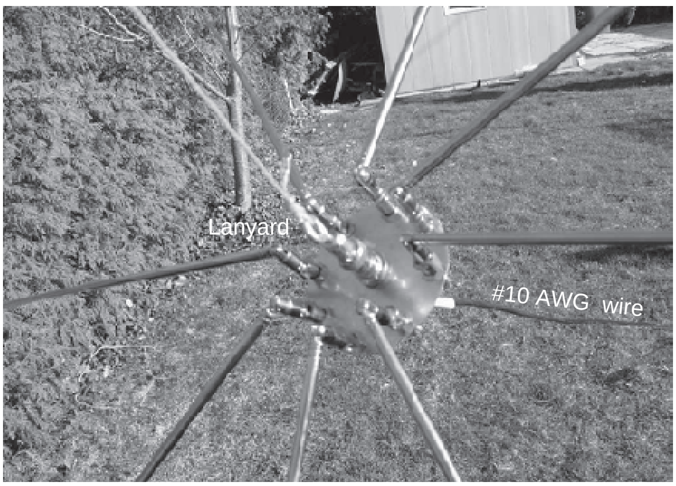
[1]Notes appear on page 20.

The dipole is 5.00 meters (16 feet 4.85 inches) long. The feed point is located at the very center of the antenna. The upper half of the antenna (above the feed point) is made of AWG no. 10 stranded wire, leading to the top hat. The lower half of the dipole (below the feed point) is realized using the RG-58/U coaxial cable shield, leading to the bottom hat. At the feed point, the coaxial center conductor is directly soldered to the no. 10 wire, as shown in Figure 5; the center conductor current continues right up the wire. Meanwhile, the coax internal shield current "folds back" onto the outer surface of the shield, as shown pictorially in Figure 6.

The bottom hat capacitor is "hot" with RF voltage. To ensure no RF is present on the rig itself, an isolator box is used just below the bottom hat. This box interrupts the exterior shield current with more than 3 kΩ isolation resistance, without affecting the internal coax current needed to feed the antenna center. Inside the box the coaxial cable is coiled into a 6 turn inductor, across which is placed an air variable capacitor in shunt, making a parallel LC circuit. Figures 7A and B show the construction of the isolator in a plastic Hammond box. (Do *NOT* use a metal box!). The capacitor shown resonates the coil anywhere between 9.2 and 17.5 MHz. If power levels greater than 100 W are anticipated, use a wider spaced capacitor. When constructing the isolator box, ensure that the capacitor rotor and shaft are connected to the rig side of the coax – this will eliminate the chance of an RF burn when adjustments are made with RF applied.

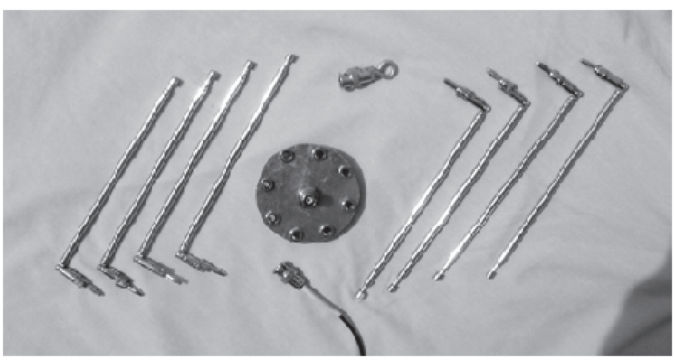
Performance

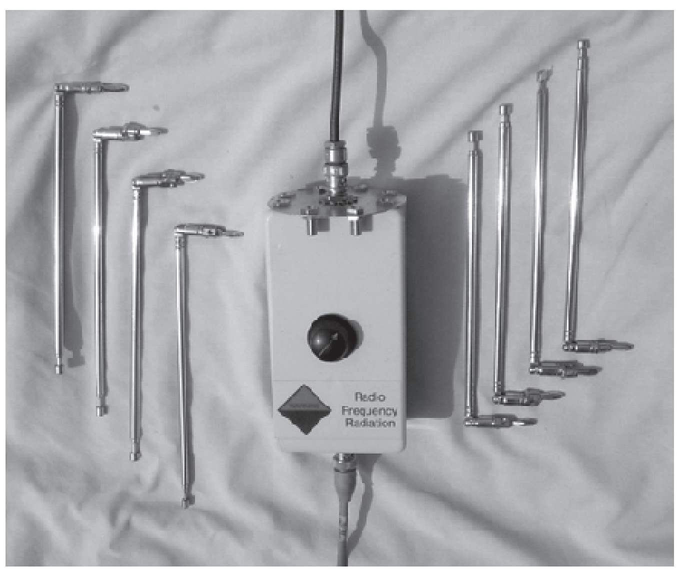
The isolator box provides coaxial isola-



(A)


(C)


(B)



(D)

Figure 3 — Part A shows the top hat capacitor with extended radial elements. A BNC(F/F) feedthrough connector is used at the center so that the lanyard and wire may be easily removed. Part B shows the top hat capacitor with the telescoping whip elements unplugged. Part C shows the bottom hat capacitor with extended radial elements. Part D shows the bottom hat capacitor and the integral isolator box, with the radial elements unplugged.

Figure 4 — This photo shows the base of a whip radial, showing a banana plug soldered to the end.

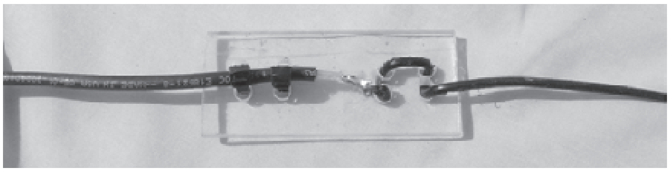


Figure 5 — Here is a photo of the center feed point, showing the coax at left and the AWG no. 10 stranded wire on the right. The coax center conductor is soldered to the stranded wire. The shield is left unterminated, so the shield current will fold around the open end to the exterior surface.

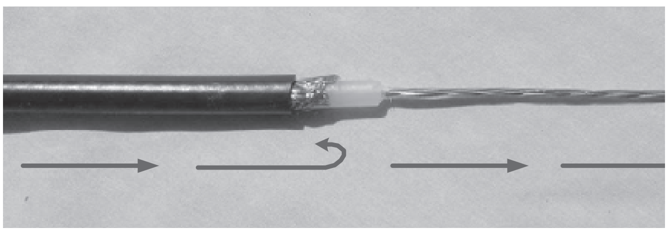
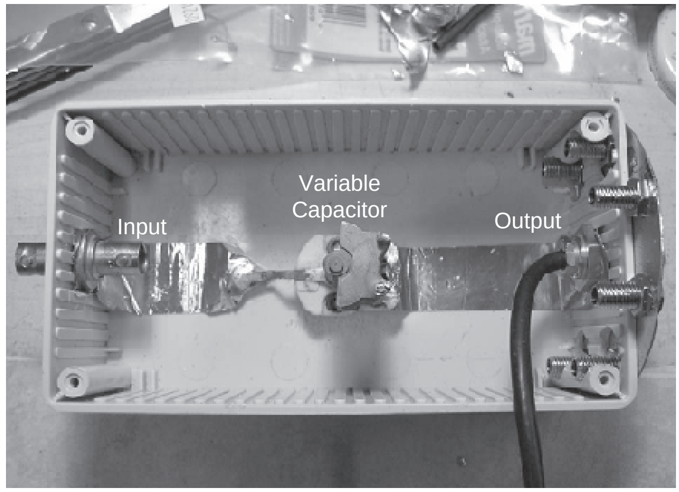
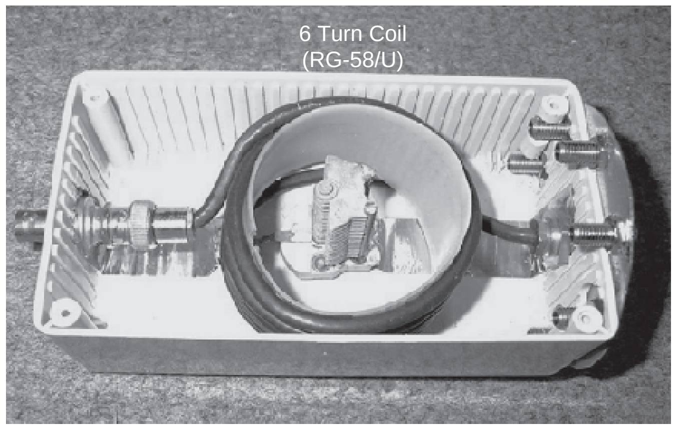


Figure 6 — Here is an illustration of the current foldback at end of the coaxial shield.



(A)



(B)

Figure 7 — Part A shows the isolator box under construction. In the photo at B, you can see the completed isolator box. Six turns of RG-58/U coaxial cable form a choke.

tion (an operating requirement) along with some series reactance above and below resonance. Above the isolation center frequency, the reactance is capacitive, while below resonance the reactance is inductive. This is precisely opposite the reactance change with frequency of the dipole itself, and helps make the antenna system broadband. The measured SWR bandwidth around 14.250 MHz is shown in Figure 8, without any retuning of the isolator box. The 2:1 SWR bandwidth is 640 kHz. Of course, in use you may easily adjust the isolator box tuning. This would allow operation on adjacent MARS frequencies above and below the amateur band, for example.

On the Air

Figure 9 shows the antenna collapsed and in a small handbag the size of a book. The total weight is half a pound. Unpacking, assembling, and raising the antenna takes about 10 minutes, if done casually. With the isolator box easily within reach (at head level), adjust the isolator box variable capacitor for best SWR. Then it is desirable to raise the antenna as high as your support will permit. This lowers the radiation "take-off" angle, as shown in Figure 10, as calculated on *EZNEC* above real earth (dielectric constant = 13, conductivity = 0.005 S/m). Extra height also slightly increases the antenna main lobe gain, as shown in Figure 11. In Figure 12 I am operating the antenna near my home in Hamilton, Ontario.

I operate with 20 W from an SG-2020 miniature rig, so that making contacts is more

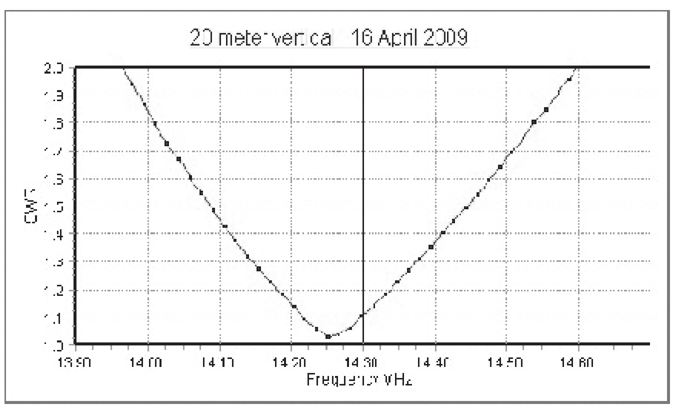


Figure 8 — This graph is the measured SWR bandwidth centered at 14.250 MHz. The SWR is less than 1.5:1 over a 350 kHz bandwidth. Measurements were taken with an AEA Bravo antenna meter.

of a challenge. On trying the antenna, in one hour I spoke with Dan, AL7RT, in Fairbanks, Orelvis, CO6LC, in Cuba, and Dan, AC5O, in Louisiana. I almost snagged Lu, DF1OM, in Germany but we had to give up when QRM took out my signal. Operating is easy, with no retuning upon moving around the band.

Robert K. Zimmerman, Jr. was born in 1951 in Dupo, Illinois. He graduated from Southern Illinois University, Edwardsville, with BS and MS degrees in physics (1973, 1975) and then attended the University of Illinois, Urbana-Champaign, where he was awarded the MSEE degree in 1980. He has spent his entire career in radio science, working for Cornell University (Arecibo Observatory), NASA Goddard Spaceflight Center, Voice of America (US Information Agency), Los Alamos National Laboratory (accelerator division), and most recently as a radar engineer on Kwajalein Atoll. He is presently involved in microwave antenna research at McMaster University, Hamilton, Ontario. He has been licensed as WN9PXG (1965), WA9ZSF, NP4B, V73BZ, and now as VE3RKZ. Zimmerman is active on 40 m and 23 cm.

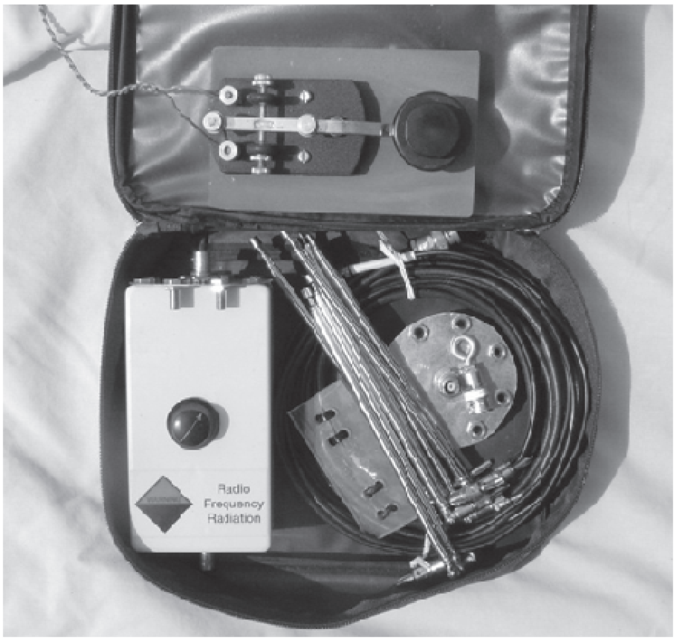


Figure 9 — Here is the antenna stowed in a handbag for easy transportation. The telegraph key illustrates the size of the package.

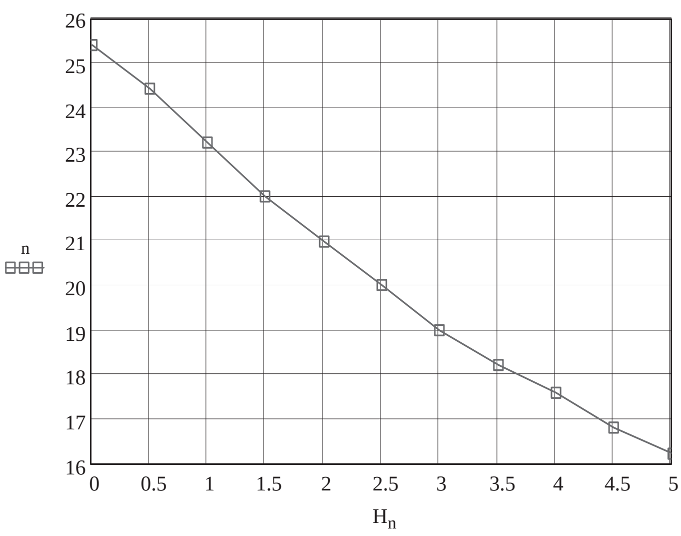


Figure 10 — This graph shows the *EZNEC* calculated take-off angle in degrees above the horizon, as a function of bottom hat height above real Earth, in meters.

Notes

[1]The *EZNEC* model files for this antenna are available for download from the ARRL *QEX* Web site. Go to **www.arrl.org/qexfiles/** and look for the file **9x09_Zimmerman.zip**.

[2]The author purchased the collapsible whip antennas used for the capacitance hats for this antenna from Nutech Electronics, 166 Parkdale Ave. N, Hamilton, ON L8H 5X2, Canada; Phone (905) 547-8420; FAX (905) 547-8422; Order Desk (800) 263-8620; **www.nutechelectronics.com**; e-mail **nutech2@bellnet.ca**

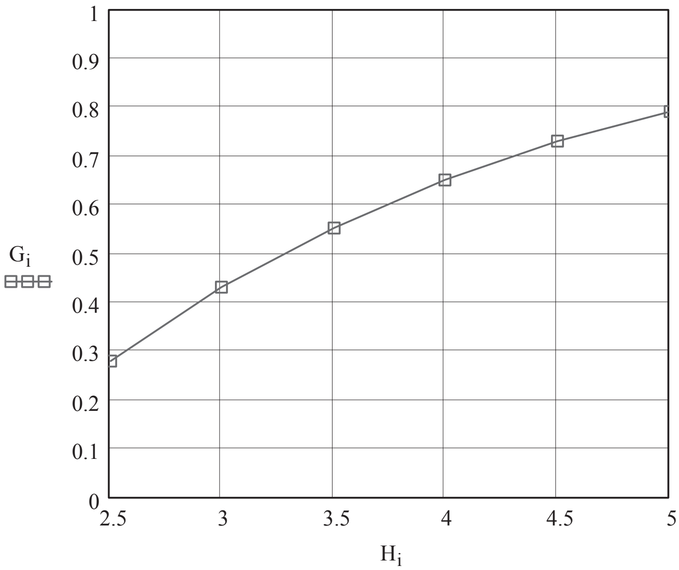


Figure 11 — This graph shows how the *EZNEC* calculated gain in dBi increases with bottom hat height above real Earth, in meters.

Figure 12 — The author operating his portable station beneath the deployed 20 m emergency antenna.

Gary Steinbaugh, PE, AF8L

9534 Appomattox Court, Loveland, OH 45140-7140, gsteinbaugh@yahoo.com

A Cybernetic Sinusoidal Synthesizer: Part 4

This part in the series describes an RF power level control module and concludes with a 53.3 MHz low phase noise synthesizer.

Before proceeding to the last two construction projects in this series, permit me to make a few salient points about control theory. Many people have unrealistic expectations of automation, in part due to its history of success. Nevertheless, the addition of automatic control will not allow you to violate the laws of physics. For example, if your kitchen oven can heat only to 400° without feedback, adding controls will not allow it to reach 500°, and if you overheat the oven, just lowering the setpoint will not obligate the temperature to drop. Similarly, it is impossible to control a process variable precisely at a setpoint, since the process variable *must* deviate from the setpoint before any corrective action can be produced. Corrective feedback is rather like steering an automobile by looking in the rear-view mirror!

From Part 3, you should have a feeling for the operation of a proportional-integral-derivative controller. Now, I would like to introduce you to lag/lead compensation, a close approximation to PID control that is implemented in many practical controllers. Those familiar with pole-zero filter design will recognize lead/lag compensators as filters, which are used to improve the dynamics of a loop by modifying its response characteristics. In Part 3, I alluded to the similarity between integral action and a low-pass filter when I presented their Laplace transforms ($1/s$ and $1/(s\tau + 1)$, respectively). Figures 40 and 41 are Bode plots that visually confirm this similarity. [Note: For readers who are unsure what a Bode plot is, my *IEEE Standard Dictionary of Electrical and Electronics Terms* says that Bode diagrams are graphs of the log-gain and phase-angle

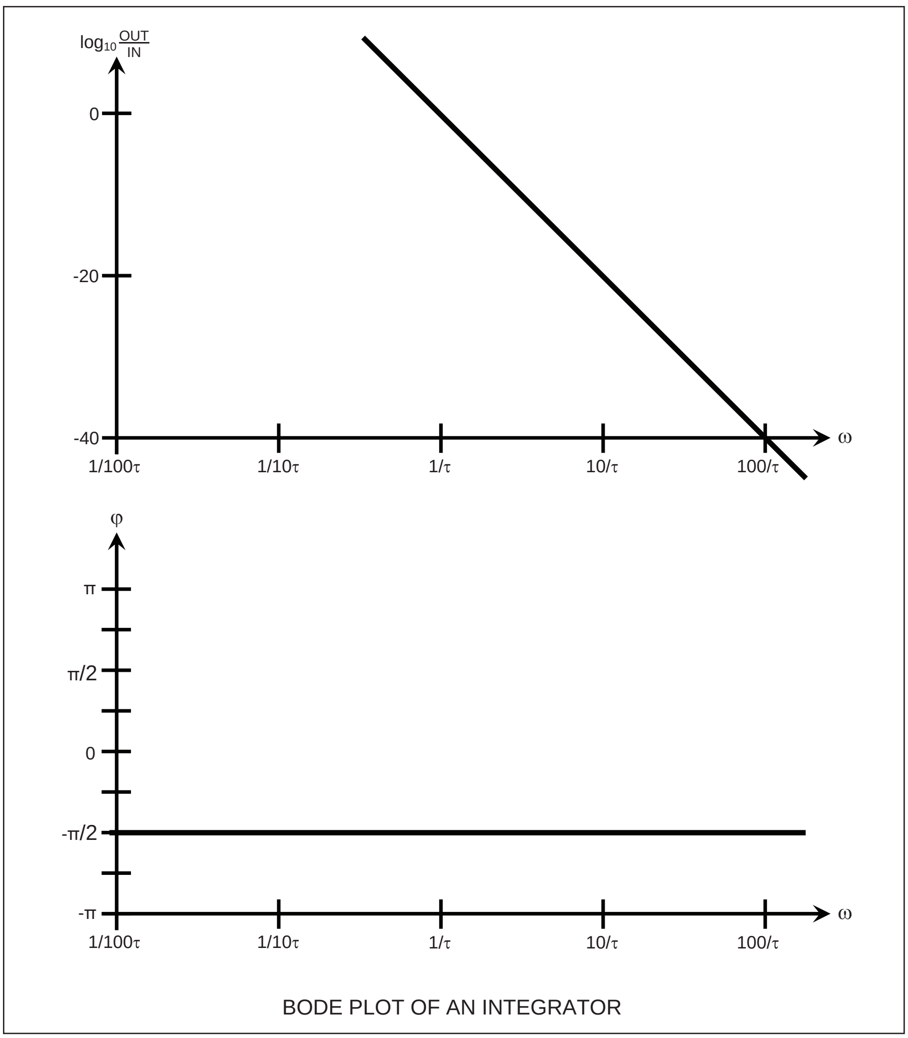


Figure 40 — This Bode plot represents the log gain and phase angle values versus log frequency of a pure integrator.

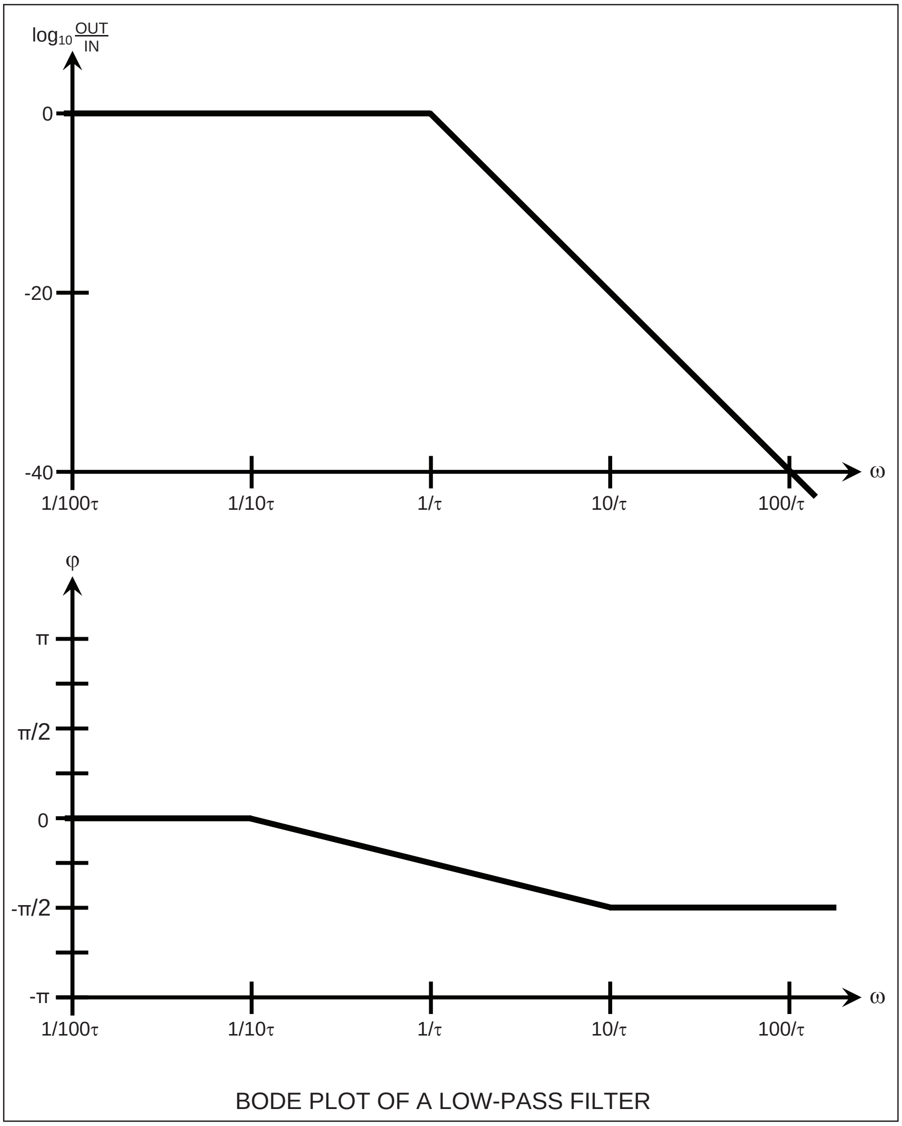


Figure 41 — This Bode plot represents the log gain and phase angle values versus log frequency of a low-pass filter. Note the similarity to Figure 40.

values on a log-frequency base for an element transfer function, G(*j*ω). — *Ed*.] Lag compensation, as used in both remaining modules, approaches a PI controller's near-zero steady-state error, but is significantly simpler to implement. In control engineering terminology, lag compensation improves steady-state conditions with some sacrifice of transient response.

An RF Power Level Controller

I found this to be the most challenging module to design. I made several attempts, both discrete and integrated, at designing an electrically-variable-gain RF amplifier, but was unhappy with some aspect of each. The best of the lot uses a National Semiconductor LMH6505, which is essentially an operational amplifier that can operate at radio frequencies. Figure 42 shows the RF power level controller schematic, and Figure 43 is a list of components. The LMH6505 is well mannered, with one exception: when the gain control input exceeds 2.3 V with respect to the virtual ground, its gain control circuitry saturates and the gain *reduces*. This is practically a prescription for an unstable loop, so a Zener clamping diode (D4) was added to preclude such an unsavory situation. The LMH6505 is a current feedback (on the inverting input) op amp whose bandwidth is not limited by a gain-bandwidth product: the maximum gain of the first amplifier is set by the gain resistor (R2) at 28 dB (reduced to a maximum of 22 dB by the 6 dB loss in R4, the output impedance matching resistor), and the bandwidth is set by the feedback resistor (R3) to about 150 MHz.

The second amplifier takes a sample of the first amplifier's output and produces an identical output, which is intended to be fed into the power meter. Its maximum gain is 10 dB, and the overall gain is adjusted to 0.00 dB with the sample adjust trimpot to compensate for the 6 dB loss in its output impedance matching resistor (R8). It is configured for a 150 MHz bandwidth as well.

Figure 44 shows the control loop diagram; lag compensation is used to provide PI-like action. The first operational amplifier of the LT1638 presents a high-impedance input to the AD8307 output, supplies lag compensation with R10 and C15, and shifts the level appropriately while dividing by two. The 6.3 V virtual ground is generated by using the same temperature compensated Zener / signal diode trick described in Part 2. The second amplifier subtracts the RF power level input from the set point, and amplifies the result, for an overall gain of 23.5; R18 and C4 provide more lag compensation. Note that D4, the 1.8 V clamping diode, has a relatively high Zener impedance, and thus is useful only with low-level signals. A six-position switch selects one of four preset power levels (that I arbitrarily set for –20, –10, 0, and +10 dBm), or a continuous level adjustment potentiometer, which varies the output in either an automatic (closed-loop) mode, or in a manual (open-loop) mode that is useful for adjusting the trimpots or for trouble-shooting. The closed-loop step response, shown in Figure 45, corresponds to a ζ of 0.3, and produces the quarter-amplitude damping recommended by Ziegler and Nichols (see the Notes in Part 3).

The LMH6505 is available only as a surface mount component, necessitating a printed circuit board, which was produced by the toner-transfer method mentioned in Part 2. To mitigate handling and presbyopia problems, I used 1206-size components. LMH6505s are a little fussy about parasitic reactances, so I suggest the layout shown in Figures 46 and 47. On the other hand, the controller layout shown in Figures 48, 49, and 50 is intended for reference only; my controller had to fit within a pre-existing volume, which turned out to be entirely too small. (Note that all of the circuit board patterns are shown at twice the actual size for clarity. Full-size patterns are available for download from the ARRL *QEX* Web site.[15]) One would have to be barking mad to duplicate this tiny circuit board without a very good reason. The traces are ½ mm wide or larger, with ¾ mm minimum separation; it is possible to produce traces on the order of a tenth of a millimeter wide, but these become rather fragile. Figure 51 depicts the circuit boards (and toner-transfer paper) before and

[15]Notes appear on page 30.

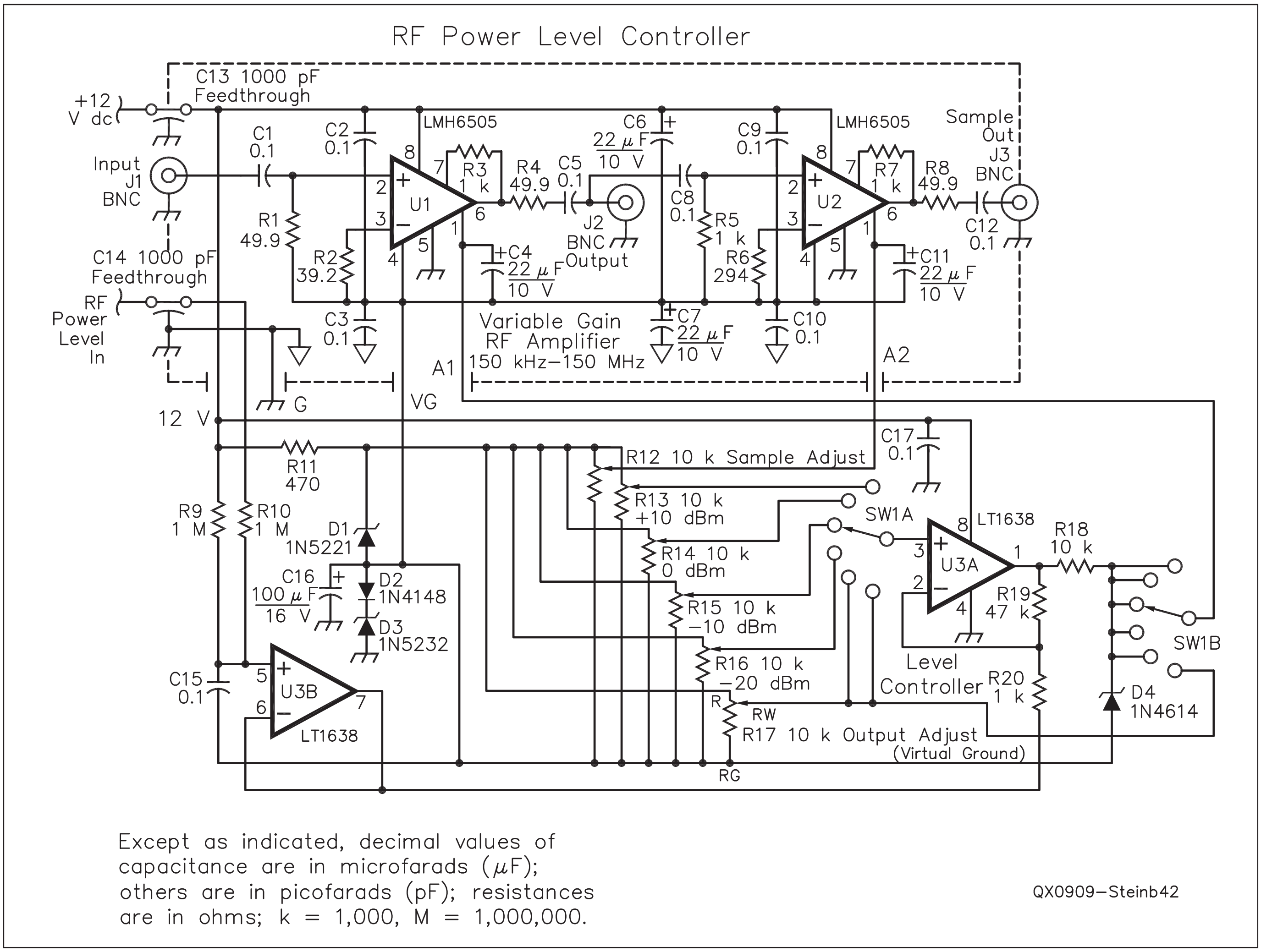


Figure 42 — Here is the schematic diagram of the RF power level controller.

RF Power Level Controller Component List

R1, R4, R8	49.9 Ω	¼ W, 1% Thick Film 1206 SMT
R2	39.2 Ω	¼ W, 1% Thick Film 1206 SMT
R3, R5, R7	1 kΩ	¼ W, 1% Thick Film 1206 SMT
R6	294 Ω	¼ W, 1% Thick Film 1206 SMT
R9, R10	1 MΩ	¼ W, 5% Carbon Film
R11	470 Ω	¼ W, 5% Carbon Film
R12, R13, R14, R15, R16	10 kΩ	Trimpot
R17	10 kΩ	Potentiometer, 9 mm (Alpha)
R18	10 kΩ	¼ W, 5% Carbon Film
R19	47 kΩ	¼ W, 5% Carbon Film
R20	1 kΩ	¼ W, 5% Carbon Film
C1, C2, C3, C5, C8, C9, C10, C12	0.1 µF	Ceramic 1206 SMT
C4, C6, C7, C11	22 µF 10 V	Tantalum 1206 SMT
C13, C14	1000 pF	Feedthrough
C15, C17	0.1 µF	Ceramic
C16	100 µF 16 V	Aluminum
D1	1N5221	2.4 V 500 mW Zener
D2	1N4148	100 V 500 mW
D3	1N5232	5.6 V 500 mW Zener
D4	1N4614	1.8 V 250 mW Zener (Central Semiconductor)
U1, U2	LMH6505MA	Variable Gain Amplifier (National Semiconductor)
U3	LT1638	Low Power Rail-to-Rail In/Out Dual Op Amp (Linear Technology)
SW1	MRK206	6-position Double-Pole Rotary Switch (NKK)

Pomona Model 3754 Miniature-size Cast Aluminum Enclosure
Pomona Model 2400 A-size Cast Aluminum Enclosure
Phoenix MPT 0,5/ 6-2,25 6-position Terminal Block
Misc.: BNC Jacks, Hardware

Figure 43 — The components list for the RF power level controller is given here.

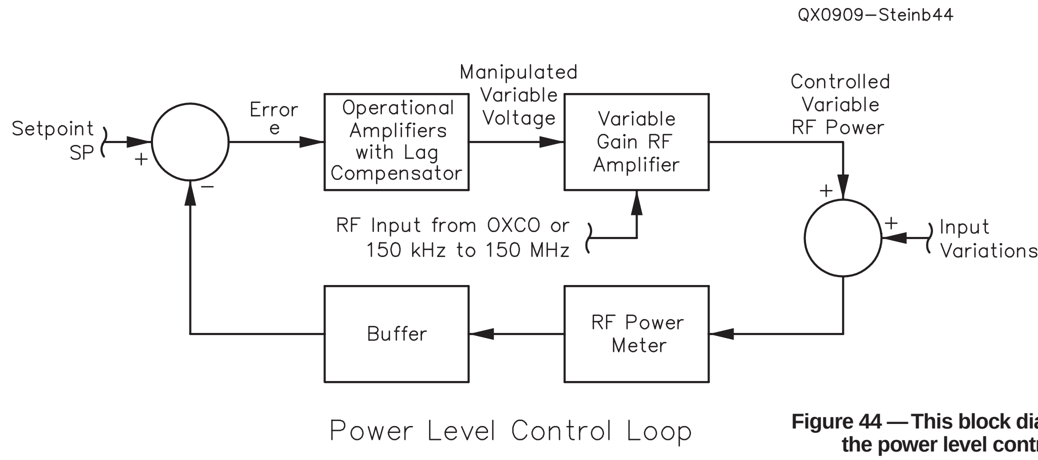


Figure 44 — This block diagram shows the power level control loop.

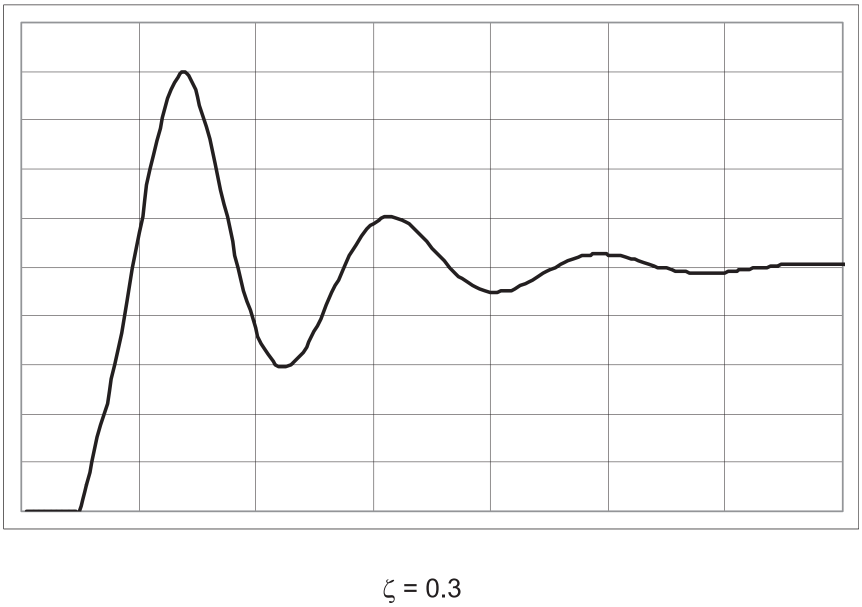
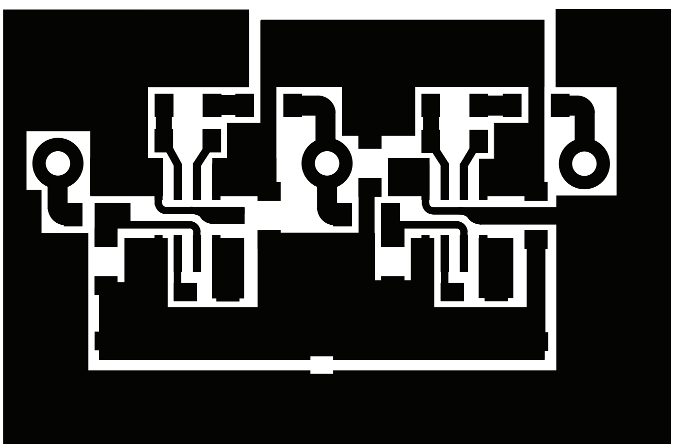
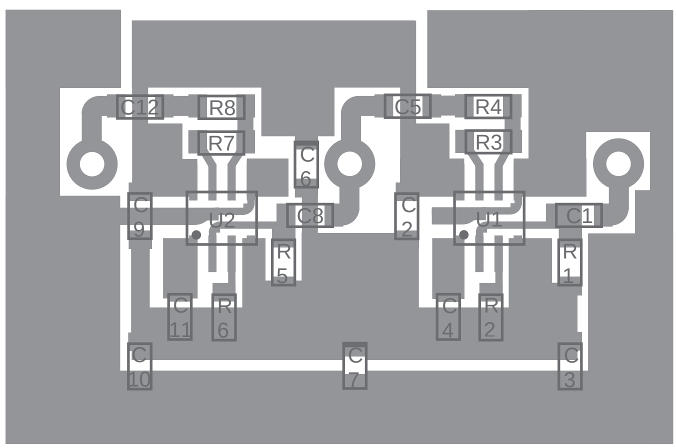


Figure 45 — This graph, with a damping factor ζ = 0.3, shows the RF power level controller response. This represents quarter-amplitude damping, a compromise that trades some overshoot for a quicker response.



AMPLIFIER PCB — 2X MIRROR IMAGE

Figure 46 — This is the circuit board pattern for the RF amplifier, shown as a mirror image, at twice its actual size. The mirror image pattern will be reversed onto the circuit board material by the toner-transfer process to make the circuit boards.



AMPLIFIER PCB — COMPONENT PLACEMENT

Figure 47 — This diagram shows the placement of the surface-mount components on the RF amplifier printed circuit board.

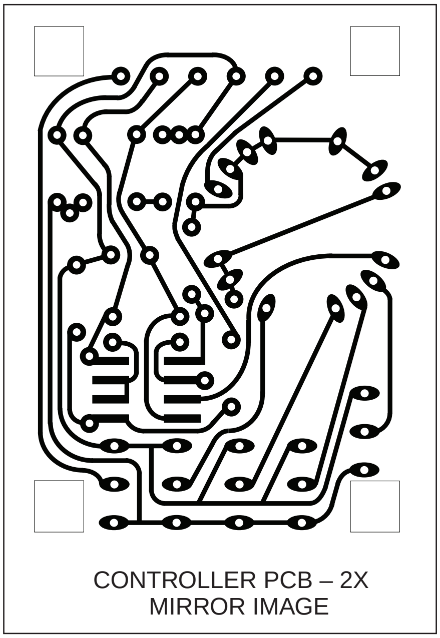


Figure 48 — This is the circuit board pattern for the power level controller, shown as a mirror image, at twice its actual size.

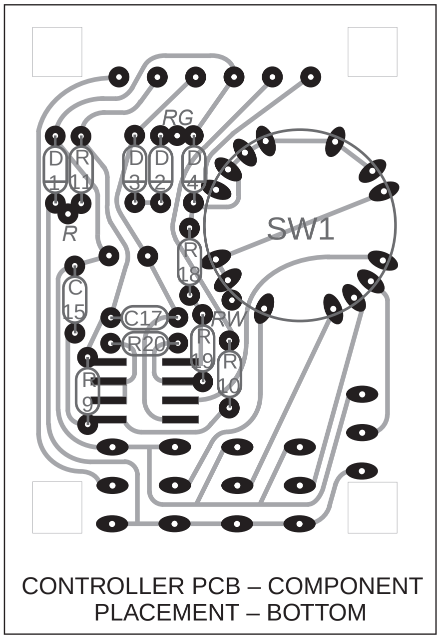


Figure 49 — This diagram shows the placement of components on the bottom side of the RF amplifier printed circuit board.

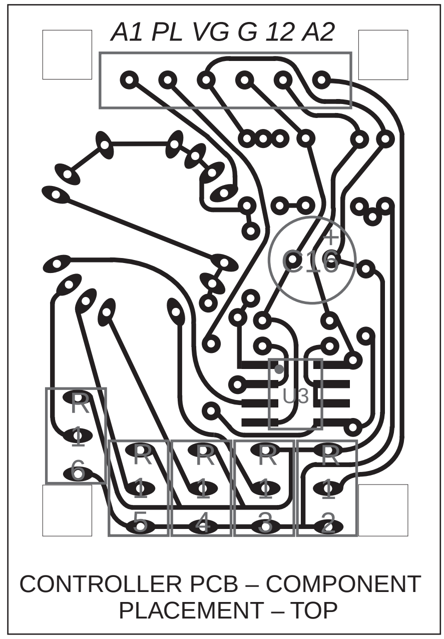


Figure 50 — This diagram shows the placement of components on the top side of the RF amplifier printed circuit board.

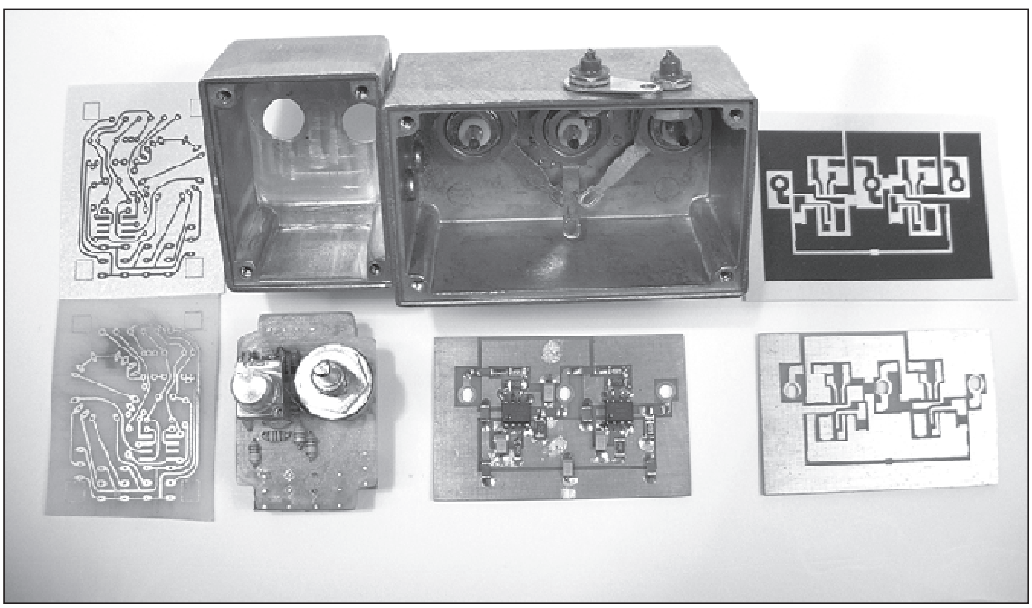


Figure 51 — Here is a photo of the RF power level controller subassemblies. Note that the mirror images on the toner-transfer paper are reversed on the etched printed circuit boards.

after assembly, and Figure 52 shows the completed module.

To verify operation, connect the sample output from the RF power level controller to the RF power meter input, and the power level output from the meter to the controller power level input, as indicated in Figure 44. Connect the OXCO through a step attenuator to the RF power level controller input, and the controller main output to an oscilloscope with a 50 Ω termination. Select manual mode, set the level adjustment potentiometer for any convenient value (0 dBm, for example), and observe the amplitude changes in the oscilloscope trace and the RF power meter display while switching in various attenuation levels. Now switch to automatic mode, readjust the level adjustment potentiometer for 0 dBm, and switch in the same attenuation levels; the output should now remain at practically the same amplitude, until enough attenuation is added that the amplifier overall gain capability of 22 dB is exceeded and control is lost. (Remember the 400° kitchen oven analogy?) Note the "springy" amplitude response on your oscilloscope as power levels are varied; this is a consequence of the closed-loop quarter-amplitude damping, and naturally will not be observed in the manual mode.

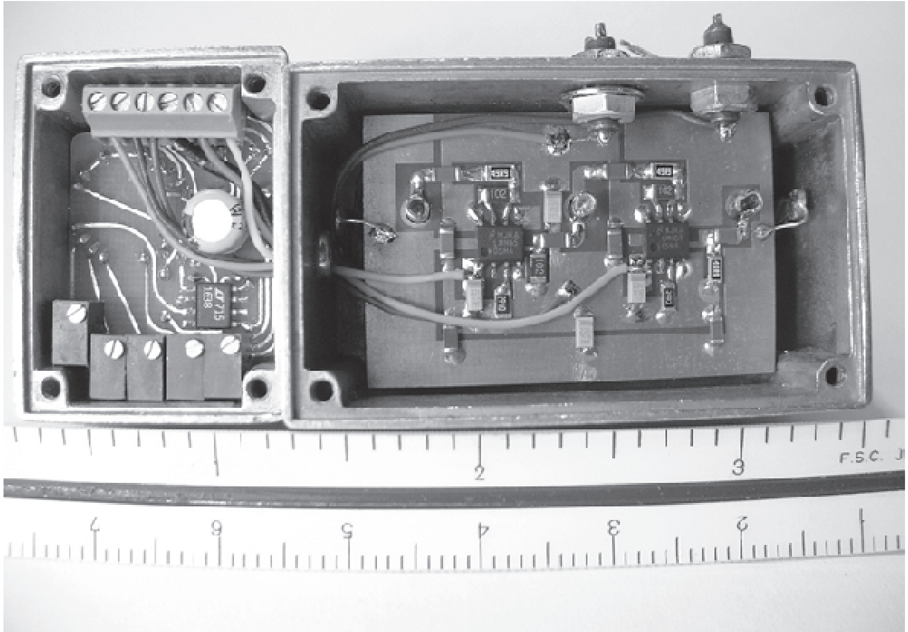


Figure 52 — Here is the RF power level controller after assembly.

A Low Phase Noise Synthesizer

We now have a high-purity 10 MHz sinusoid that is stabilized in both frequency and amplitude. My spectrum analyzer needed a 55.3 MHz local oscillator, however, so this frequency was synthesized by using a feedback system commonly referred to as a Phase Locked Loop (PLL).

Figure 53 shows the frequency synthesizer schematic, Figure 54 is a list of components, and Figure 55 shows the synthesizer control loop diagram. The step response for $\zeta = 1$ as shown in Figure 56 is termed *critically damped*, the lowest damping that can be used without overshoot, and is equivalent to a phase margin φ_M of about 1 radian.

With increasing system integration, it may seem rather antique to implement a PLL synthesizer with individual counters and similar components. Well, it irritates me to come upon an interesting circuit in a magazine that is a couple of years old, only to find that a primary component is no longer available; PLL IC systems seem predisposed to becoming obsolete quickly. Moreover, they require a microprocessor interface, and although this would not be difficult to design, there is something vaguely revolting about having to implement a microcontroller that does nothing except to load a few bytes of data into a PLL IC at power-up.

I chose a Mini-Circuits POS100 for the voltage controlled oscillator (VCO), as it has good phase noise characteristics (see Figure 57), is relatively inexpensive, and is easy to order;[16] at 55.3 MHz, its gain (K_V) is about 2.7×10^7 rad / (s V). Note the power supply LC filter, which was motivated by an article by Colin.[17] I would recommend Leeson (W6NL)[18] or Goldberg[19] to anyone who is interested in phase noise mathematics. As with the OXCO, the output impedance of the VCO is matched to 50 Ω with a 6 dB resistive attenuator.

Before they are sent to the counters, the sinusoidal outputs of the VCO and OXCO are converted to logic-level square waves. A 74HC390 dual decade counter divides the 10 MHz OXCO input frequency down to a 1 MHz square wave, and then to the 100 kHz reference frequency; this establishes the 0.1 MHz resolution of the synthesizer. Either frequency is jumper-selectable as a buffered auxiliary output for use as a marker generator, or other stable-frequency source (note that this auxiliary signal is a harmonic-rich, 5 Vp-p square wave, and not a controlled-impedance sinusoid).

Three 74AC161 synchronous counters implement a divide-by-N circuit; in order to synthesize a 55.3 MHz local oscillator using a 100 kHz reference frequency, N must be 553. (Another control engineering observation: an operation β in the feedback path will produce an inverse effect on the overall loop; thus division results in frequency multiplication.) This divisor is obtained by presetting the counters to DD7 (decimal 3543) and letting them step up to their terminal count of FFF (decimal 4095), which sets up a synchronous preset on the next (553rd) count. If a different LO frequency is desired (within the capabilities of the VCO, of course), it is not difficult to recalculate the counter preset number.

The individual counters will operate at 170 MHz, but the maximum frequency of the divide-by-N circuit is limited by the time required to detect the terminal count and prepare for the preset before the next clock pulse arrives. A lookahead carry arrangement sidesteps several gate delays; this trick produces a divider that can be clocked at about 70 MHz with a 5 V V_{CC}. Note that this division ratio will add 20 log (553) = 54.9 dB to the phase detector noise (see Figure 57), so choosing a low-noise phase detector becomes important.

I was attracted to the Philips 74HCT9046A by the data sheet's promise that its phase comparator #2 "allows a virtually ideal performance." [20] The venerable CD4046 may be substituted, but its noise performance degrades when there are very small phase differences between its two inputs, whereas the 9046A phase / frequency detector (PFD) has additional circuitry to eliminate this "dead zone" problem, and also has a charge pump output that makes a loop filter amplifier (another potential contributor of phase noise) unnecessary. With a bias resistor (R_B) of 24.9 K, the PFD gain (K_P) is 2.7×10^{-4} A / rad. The positive and negative current pulses of its output are absorbed by a third-order loop filter that functions as a lag compensator, modifying the loop response by suppressing undesirable frequency components. The overall loop exhibits proportional-integral action.

Since the synthesizer operates on a single frequency, switching speed is irrelevant, sug-

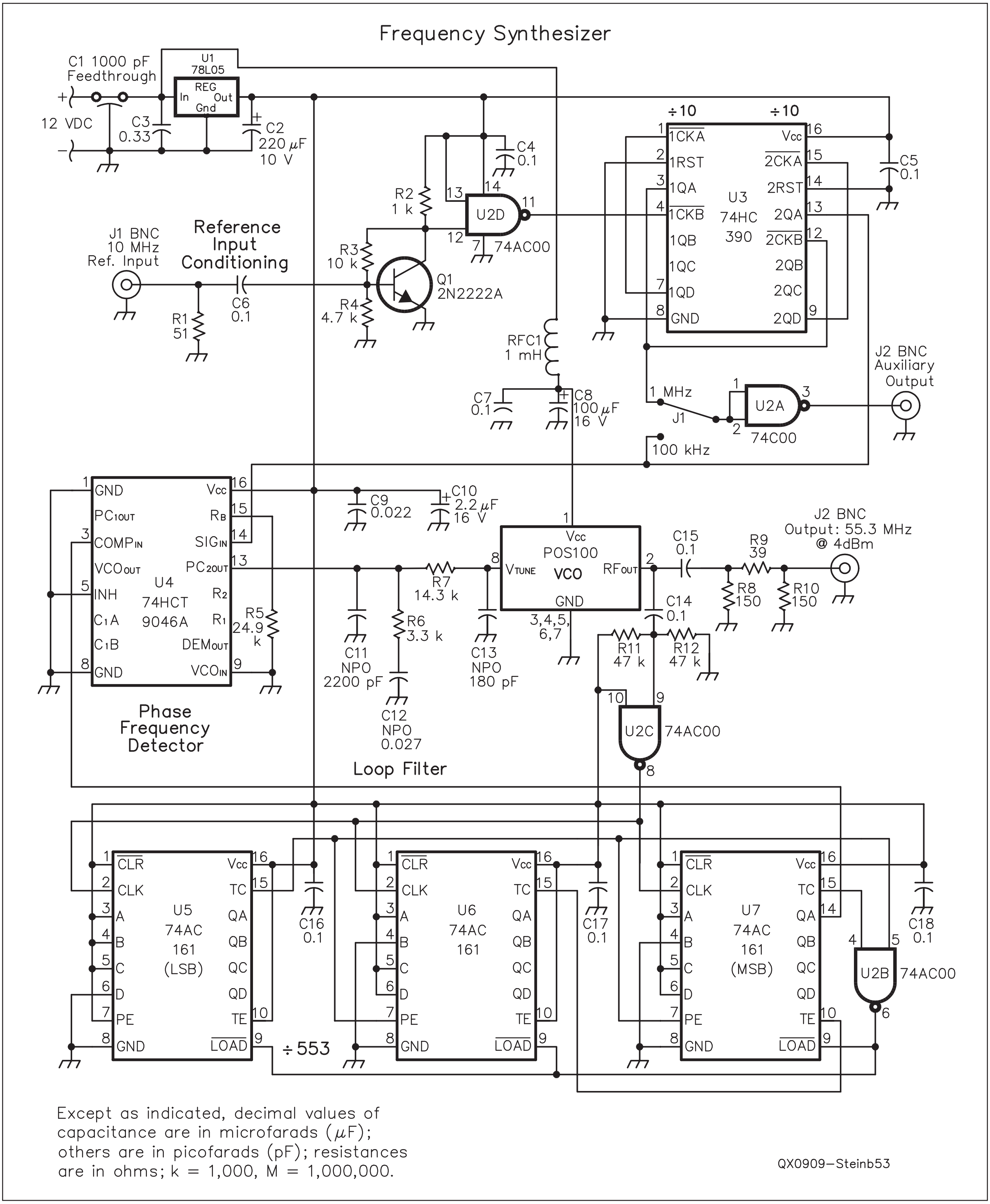


Figure 53 — Here is the schematic diagram of the frequency synthesizer.

gesting that the loop filter bandwidth (ω_C) should be made as narrow as possible … but doing so would forfeit an important loop advantage: noise within the loop bandwidth is predominately from the PFD + divider combination, while the VCO has large low-frequency noise components that decline in amplitude as their frequencies increase, at some point becoming less than the PFD + divider noise (see Figure 57). By setting the bandwidth at this crossover point, the large oscillator noise components are effectively rejected, thus lowering the maximum overall phase noise.

Unfortunately, Philips could not characterize the 74HCT9046A phase noise, so I had to make an educated guess: an active PFD has an SSB noise power density ($\mathcal{L}(f_m)$) in the neighborhood of –140 to –160 dBc/Hz, which, with the divider noise, would yield an overall $\mathcal{L}(f_m)$ of about –100 dBc/Hz.[21] The VCO $\mathcal{L}(f_m)$ drops below –100 dBc/Hz at roughly 6 kHz (4×10^4 rad / s), so I chose this as the loop bandwidth for filter component calculations (for the mathematically inquisitive, I have included the pertinent formulas in the Appendix).

Measuring phase noise with a spectrum analyzer has its problems: the display is modified by logarithmic compression of the IF signal, the resolution bandwidth is not 1 Hz, and, except for the region between about 1 kHz and 10 kHz (see Figure 57), the $30,000 Agilent 4395A is noisier than this $30 synthesizer. Considering this, the synthesizer's spectrum as shown in Figure 58 suggests that its performance is close to what was desired. The spurious signal at 55.4 MHz is an artifact of the 100 kHz (0.1 MHz) reference frequency.

Figure 59 is a photograph of the completed synthesizer; note the jumper for selection of the auxiliary output frequency above the top left IC. The auxiliary output jack is shielded with a BNC cap when it is not being used.

Conclusion

The four modules presented in this article may be combined in several ways, or any one might be a useful component in a system of your design. I hope that some of the electronic tricks sprinkled here and there will infiltrate the heart of one of your circuits.

If you are intrigued about automatic control engineering, I would suggest getting some books like Melsa & Schultz[22] or Raven[23] on your radar. Should your interest in phase locked loop design exhibit a $\zeta < 0$, Gardner[24] is the standard text in the field. My address is listed at the beginning of each Part; I would be pleased to receive any … feedback. Use your Cybernetic Sinusoidal Synthesizer in good health!

Frequency Synthesizer Component List

R1	51 Ω	¼ W, 5% Carbon Film
R2	1 kΩ	¼ W, 5% Carbon Film
R3	10 kΩ	¼ W, 5% Carbon Film
R4	4.7 kΩ	¼ W, 5% Carbon Film
R5	24.9 kΩ	¼ W, 1% Metal Film
R6	3.3 kΩ	¼ W, 1% Metal Film
R7	14.3 kΩ	¼ W, 1% Metal Film
R8, R10	150 Ω	¼ W, 5% Carbon Film
R9	39 Ω	¼ W, 5% Carbon Film
R11, R12	47 kΩ	¼ W, 5% Carbon Film
C1	1000 pF	Feedthrough
C2	220 µF 10 V	Aluminum
C3	0.33 µF	Ceramic
C4, C5, C6, C7	0.1 µF	Ceramic
C8	100 µF 16 V	Aluminum
C9	0.022 µF	Ceramic
C10	2.2 µF 16 V	Tantalum
C11	2200 pF	NP0 (C0G) Disc
C12	0.027 µF	NP0 (C0G) Disc
C13	180 pF	NP0 (C0G) Disc
C14, C15, C16, C17, C18	0.1 µF	Ceramic
RFC1	1 mH	
Q1	2N2222A	NPN Bipolar
U1	78L06	6 V @ 100 mA Regulator
U2	74AC00	Quad NAND
U3	74HC390	Dual Decade Counter
U4	74HCT9046A	Phase Locked Loop (Philips / NXP)
U5, U6, U7	74AC161	Synchronous Presettable 4-bit Counter
VCO	POS100	Voltage Controlled Oscillator (Mini-Circuits)
Hammond Model 1590G Cast Aluminum Enclosure		
Misc.: BNC Jacks, 14- and 16-pin DIP Sockets, Hardware		

Figure 54 — The components list for the frequency synthesizer is given here.

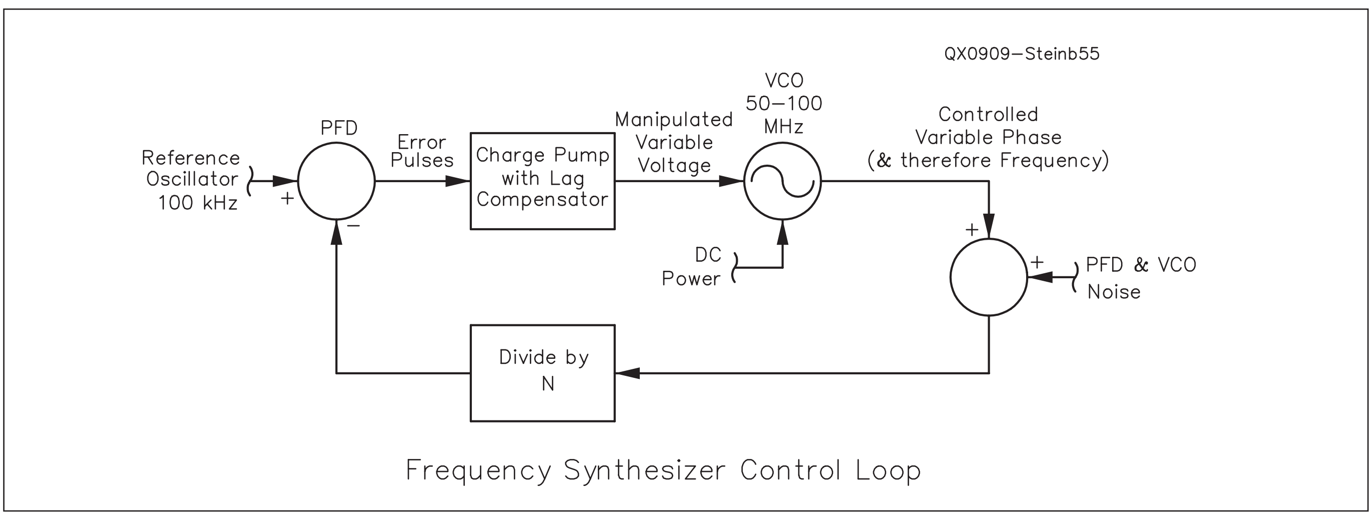


Figure 55 — This block diagram shows the frequency synthesizer control loop. Compare this with the control loop diagrams for the previously described modules.

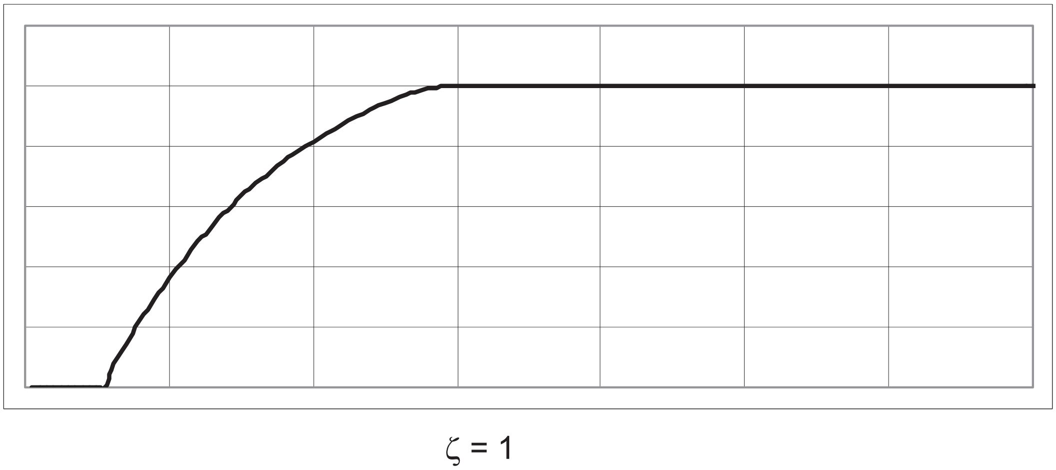


Figure 56 — This graph, with a damping factor ζ = 1, shows the frequency synthesizer response. This represents critical damping, the lowest damping that can be used without overshoot.

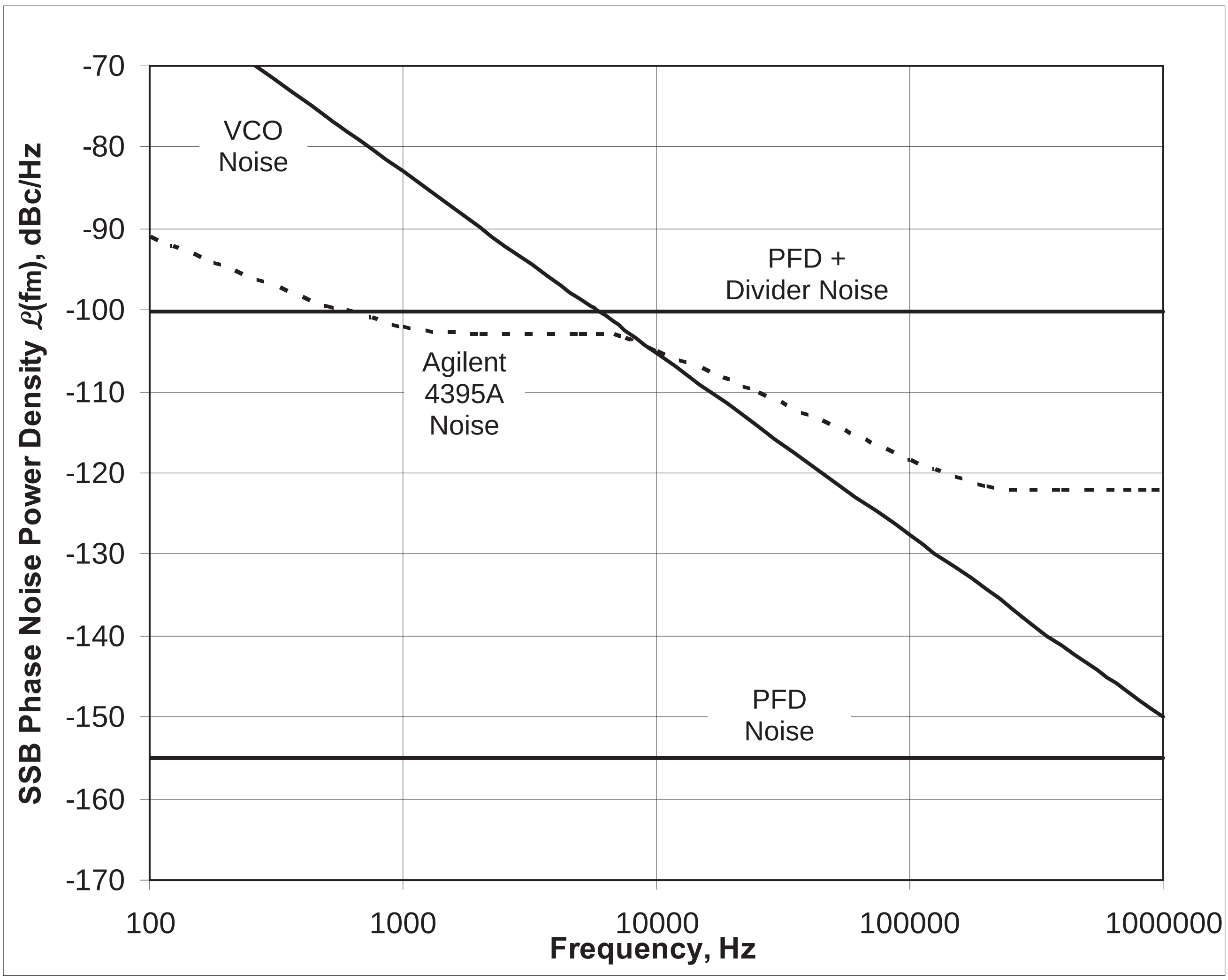


Figure 57 — This graph shows the SSB phase noise power density for the various elements of the phase locked loop system, including the $\mathcal{L}(f_m)$ of the measuring instrument, an Agilent 4395A spectrum analyzer. To optimize loop performance, the frequency where the PFD + divider noise intersects the VCO noise (6 kHz) was chosen for the loop bandwidth, ω_C.

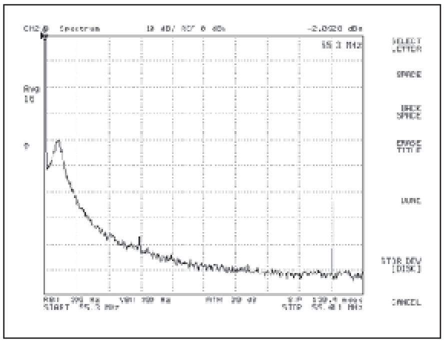


Figure 58 — This graph shows the spectrum produced by the phase locked loop, with 55.3 MHz at the left and 55.4 MHz at the right (the spike at 55.4 MHz is an artifact of the 100 kHz reference frequency derived from the ovenized oscillator).

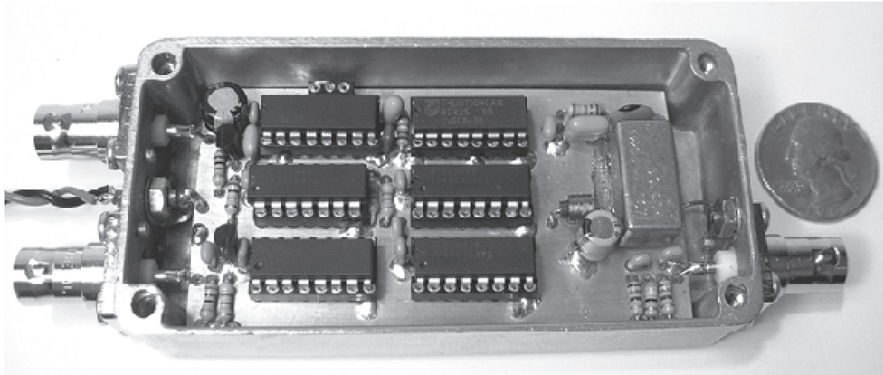


Figure 59 — This is a photo of the completed frequency synthesizer.

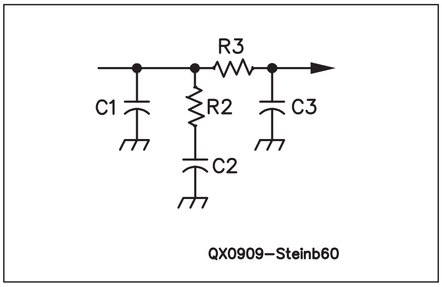


Figure 60 — Here is the schematic diagram of a third-order phase locked loop filter.

Notes

[15]Full size etching pattern files for the circuit boards described in Part 4 are available for download from the ARRL *QEX* Web page. Go to **www.arrl.org/qexfiles** and look for the file **Steinbaugh_9x09.zip**.

[16]Mini-Circuits, PO Box 350166, Brooklyn, NY 11235-0166, (718) 934-4500, **www.minicircuits.com**.

[17]Dennis Colin, "Externally Induced VCO Phase Noise," Norwood: *Microwave Journal*, Vol 45, No. 2, February 2002.

[18]David B. Leeson, "A Simple Model of Feedback Oscillator Noise Spectrum," *Proceedings of the IEEE*, Vol 54, February 1966.

[19]Bar-Giora Goldberg, "Phase Noise Theory and Measurements: A Short Review," Norwood: *Microwave Journal*, Vol 43, No. 1, January 2000.

[20]Philips Semiconductors, Data Sheet, 74HCT9046A, PLL with band gap controlled VCO, October 2003.

[21]Bar-Giora Goldberg, *Digital Frequency Synthesis Demystified*, Eagle Rock: LHH Technology, 1999.

[22]James L. Melsa and Donald G. Schultz, *Linear Control Systems*, New York: McGraw-Hill 1969.

[23]Francis H. Raven, *Automatic Control Engineering*, New York: McGraw-Hill 1968.

[24]Floyd M. Gardner, *Phaselock Techniques*, Hoboken: John Wiley & Sons, 2005.

Continuously licensed since 1964, Gary Steinbaugh, AF8L, is an ARRL Life Member. Holding a BSEE from Case Institute of Technology and several patents, he is a licensed Professional Engineer, and the author of many technical articles. He is Senior Electronic Engineer for AtriCure, Inc., a manufacturer of RF electrosurgical instruments used for soft tissue ablation. Gary is also a Certified Flight Instructor and a semi-pro musician.

Appendix
Design Formulas for a Third-Order Phase Locked Loop Filter

ω_C	bandwidth	rad / s	6.4 kHz = 4.0 × 10^4 rad / s
φ_M	phase margin	rad	1 rad
K_V	VCO gain	rad / (s V)	4.3 MHz / V = 2.7 × 10^7 rad / (s V)
R_B	74HCT9046A R_{BIAS}	Ω	2.49 × 10^4 Ω
F_{OUT}	output frequency	Hz	5.53 × 10^7 Hz
F_{REF}	reference frequency	Hz	1.00 × 10^5 Hz
T3 / T1 ratio		(no units)	0.4
C1 / C3 ratio		(no units)	12
$N = F_{OUT} / F_{REF}$		(no units)	= 553
$K_P = \frac{6.764}{R_B}$		A / rad	= 2.7 × 10^{-4} A / rad

(Be sure your calculator is in radian mode.)

$T1 = \frac{\frac{1}{\cos\varphi_M} - \tan\varphi_M}{\omega_C}$		s	= 7.3 × 10^{-6} s
$T2 = \frac{1}{\omega_C^2 \; T1}$		s	= 8.5 × 10^{-5} s
T3	= 0.4 T1	s	= 2.9 × 10^{-6} s
$C1 = \frac{K_V \; K_P \; T1}{N \; \omega_C}$		F	= 2.4 × 10^{-9} F (→ 2200 pF)
$C2 = \frac{K_V \; K_P \; T2}{N \; \omega_C} - C1$		F	= 2.6 × 10^{-8} F (→ 0.027 μF)
C3	= C1 / 12	F	= 2.0 × 10^{-10} F (→ 180 pF)
R2	= T2 / C2	Ω	= 3.3 × 10^3 Ω (→ 3.3 kΩ)
R3	= T3 / C3	Ω	= 1.44 × 10^4 Ω (→ 14.3 kΩ)

Mike Hamel, WO1U

5 Ridge Rd, Essex Junction, VT 05452; wo1u@arrl.net

Phase Controlled Differential Drive for EER Amplifiers

Eliminate the high level modulator in high efficiency envelope elimination and restoration amplifiers.

With the increasing popularity of high efficiency amplifier techniques such as Class E, there is a strong need for a simple approach to linearizing these amplifiers so that the majority of amateur RF amplifier builders can reap the benefits of high efficiency.[1, 2] High efficiency amplifiers are smaller and have lighter power supplies, lower output device heating per watt of output power, and higher reliability. For example, a common broadband linear solid state RF amp with 100 W output consumes on the order of 180 to 220 W of dc power at full output, depending on biasing. A Class E amplifier with comparable power output has the disadvantage of being tuned and therefore is only usable over typically one or two bands. Efficiency, however, can reach 90% in practice. This corresponds to a dc input power of 111 W for 100 W RF output.

The downside of the saturated switch mode classes of operation is non-linearity, meaning the output power is not directly proportional to the drive power. In a perfectly linear amplifier with a gain of 10 dB, the output power is exactly 10 times the input power regardless of the drive level, from very low levels to full rated output. With a single ended Class E amplifier, however, the output device must be driven to saturation, so if the drive amplitude were reduced, the efficiency would drop dramatically. In order to maintain Class E operation, a constant minimum drive level to the final stage at the operating frequency must be maintained. This means a Class E amplifier cannot be amplitude modulated by varying the drive signal amplitude.

A number of amateur builders are using Class E RF stages successfully in AM transmitters by modulating the power supply. A carrier at the operating frequency must be present to drive the final stage, even when the RF output must dip to zero during 100% modulation, otherwise distortion and splatter will occur. With high level AM transmitters, the audio information is already separated from the RF carrier. As such, a power supply may be high level modulated with an audio amplifier to achieve efficient AM operation with a Class E transmitter stage.

Unfortunately, with single sideband suppressed carrier voice operation, using an external high efficiency power amplifier isn't so easy. There isn't a continuous carrier between words, and the transition from no signal between words to a carrier with enough drive to operate the output stage correctly, can be a very abrupt transition which causes distortion and spurious signals.

Kahn was the inventor of the linearization technique called envelope elimination and restoration (EER) in the 1950s.[3] This technique involves taking a signal requiring linear amplification, such as from an SSB transmitter, and separating the amplitude envelope information and the carrier frequency information into two separate channels. The frequency information is extracted by clipping the signal into limiting to derive the carrier phase, and the AM envelope is detected for driving a high level modulated power supply. Done properly, this configuration can correctly and linearly amplify an SSB signal even though the RF final stage may be operating in a non-linear mode such as Class C, D, E, F or S.

What is presented here is a different approach to providing amplitude modulation of saturated switching amplifiers, using only low level modulation of the drive signal. This could be used to simplify the construction of an EER amplifier, or an AM transmitter. When the dc supply can remain fixed and unmodulated, the construction complexity is greatly reduced and the efficiency is higher because a high level modulator stage is not required. Reliability is also increased due to reduced complexity and component count.

One amplification topology rarely discussed is the Class D current mode amplifier.[4] This is a good topology to illustrate phase controlled differential drive. This type has also been described as a "push-pull Class E" type because the operating principle is similar. A Class D current mode amplifier uses a tuned resonant output transformer with saturated switch push-pull finals. In operation, the drain voltage and current waveforms are identical to a Class E single ended stage.

The traditional timing of differential drive to any push-pull stage is such that the output devices are never on at the same time, and the phase relationship is always 180° to ensure maximum output.

For a minute, let's examine what would happen if the phase of the drive was not 180° at the finals. An output transformer is required to transform the differential signal from the drains of the push-pull finals into a single-ended signal for the load. The output transformer can be thought of as a differencing device. Any in-phase signals applied to its primary are canceled and only differential signals pass through it. So it stands to reason if you can modulate the phase of the drive to the individual push-pull output devices, you can modulate the current delivered to the output transformer. When in-phase at zero degrees, there is no current in the output transformer primary and no RF power is induced in the secondary circuit; when driven at 180°, maximum possible current is delivered to the output transformer and

[1]Notes appear on page 34.

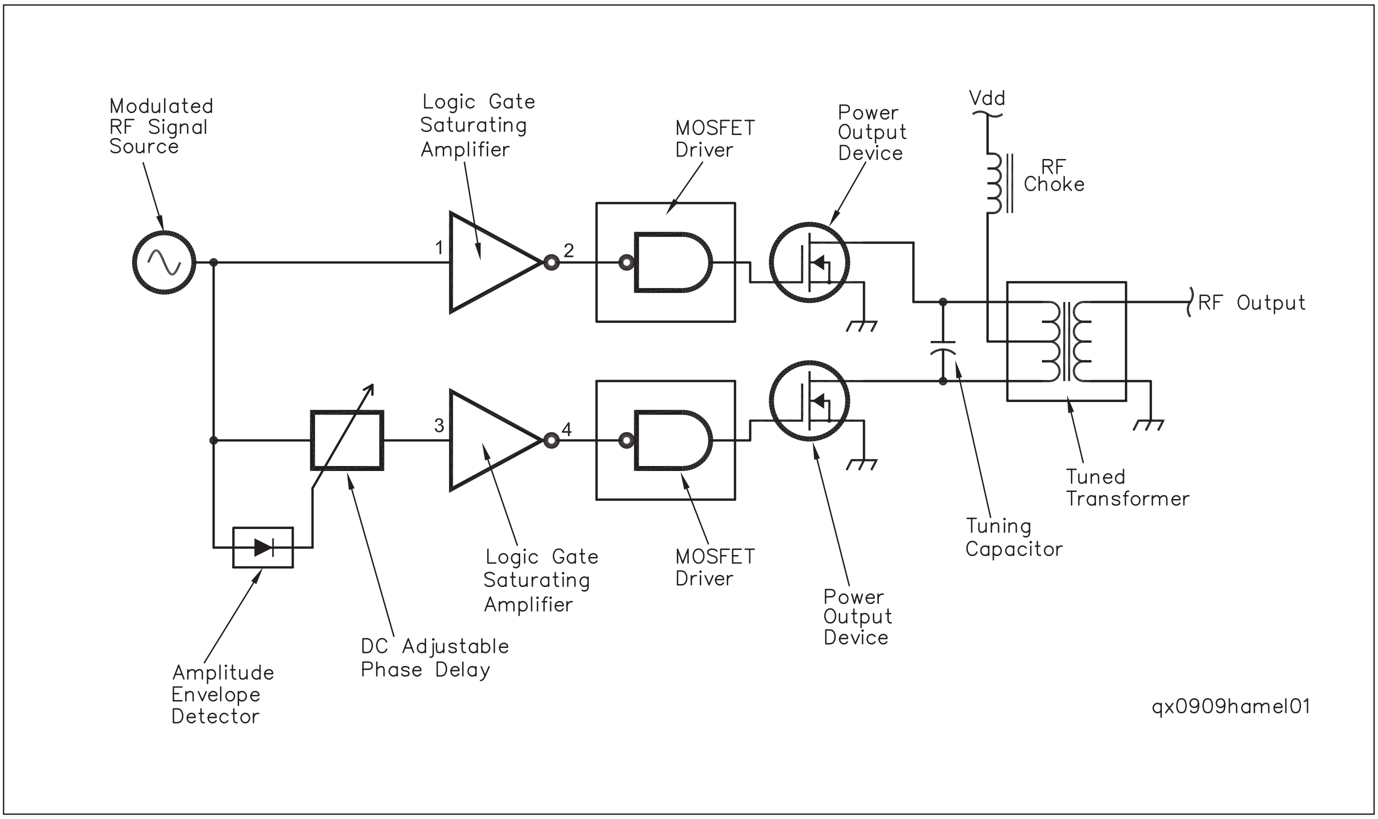


Figure 1 — This block diagram shows the operation of a Class D current mode amplifier, with phase controlled differential (PCD) modulation.

induced in the secondary circuit. The fact that the primary is tuned to resonance allows the current to ring with a sinusoidal waveform, whose amplitude is proportional to the phase difference in the drive signals, despite the non-sinusoidal drive signal.

With carefully controlled phasing, 100% modulation can be achieved over a zero to 180° phase range between the two drive signals. The block diagram in Figure 1 shows a Class D current mode amplifier topology with phase controlled differential (PCD) modulation. The efficiency for Class D current mode is comparable to that of a Class E amplifier, and can reach 90% when properly optimized.

In order to maintain high efficiency, dc must be supplied to the center tap of the output transformer through an RF choke as shown in Figure 1. When the driver phase is 0°, all of the RF drain voltage appears across the choke, which has a high impedance at the operating frequency, and very little current is drawn from the power supply. As the driver phase moves from 0° toward 180°, more current is delivered to the output transformer primary winding(s). Alternatively, individual drain chokes may be used. The center tap lead must not be bypassed to RF ground.

The key to PCD drive is an adjustable phase delay circuit, in which the phase delay

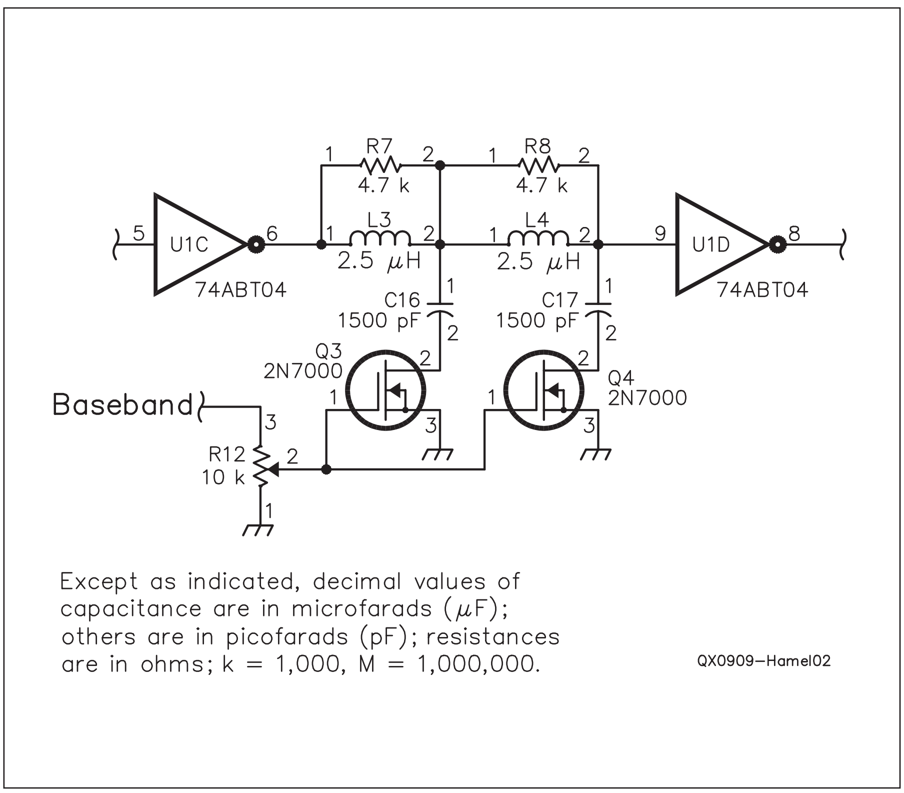


Figure 2 — This schematic diagram shows a simple dc adjustable LC network phase delay that can be used with a phase controlled differential (PCD) modulator.

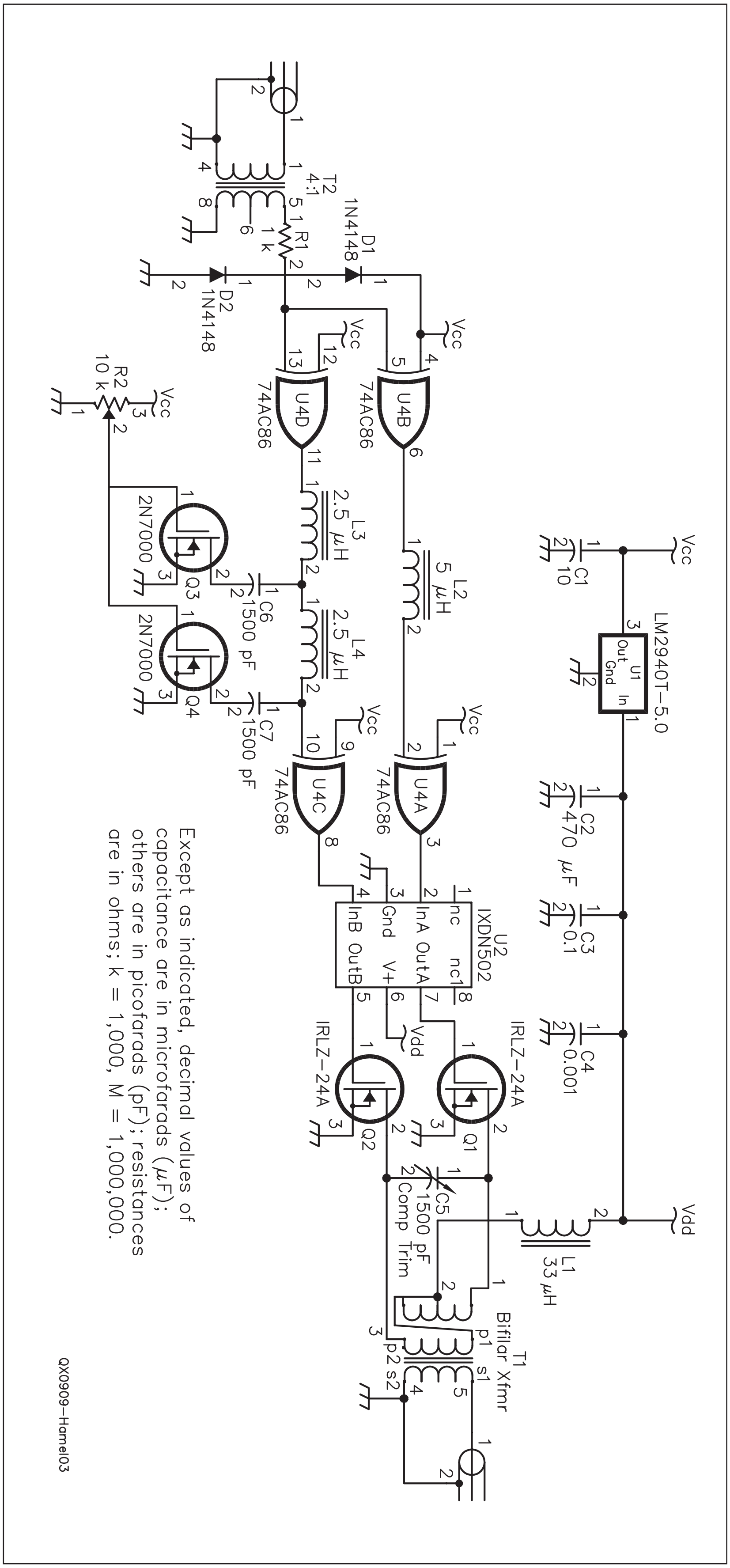


Figure 3 — Here is the schematic diagram of my first prototype amplifier.

is controllable with a dc voltage or baseband modulating signal. Figure 2 shows a simple dc adjustable LC network phase delay. Baseband modulation with dc offset is applied to potentiometer R12. A dc offset is required to bias Q3 and Q4 to the threshold of conduction. Phase performance of this network was modeled using the Quick Universal Circuit Simulator (QUCS) an open source modeling tool available for many Linux distributions. The 2N7000 MOSFETs in this example essentially determine the shunt capacitance of the network, which provides a variable phase delay.

In practice, the inductors and resistors of this network are applied to both final gate drivers, but only one has phase shift capacitors C16, C17 and delay capability. This is done to maintain drive symmetry when a low amount of phase shift is required to get near zero output. The network shown in Figure 2 is designed for 160 meter operation. It functions through 80 meters as well, but best linearity performance is achieved with single band networks.

In the first “brassboard” prototype amplifier, shown in Figure 3, a pair of IRLZ-24A logic level FETs was used with an Ixys IXDN502 two ampere high-speed dual driver. Running at 15 V, and with 1 W drive, the CW amplifier demonstrated adjustable power output from a milliwatt to more than 50 W using only a potentiometer to control the output power. It maintained 89% efficiency over most of the range.

In another way of applying PCD, a Class E single ended amplifier is driven with pulse width modulation using a PCD stage with a broadband transformer. The resonance of the Class E output network “cleans up” what would otherwise be a very ugly signal with rich harmonic content. Because of this resonant output network, it is not necessary that the drive signal be sinusoidal. It follows that the amplitude of a class E stage can be controlled by pulse-width modulation (PWM) of the gate drive signal. A properly adjusted Class E amplifier will put out a power level proportional to the duty cycle of the gate drive, within certain constraints and limitations. (The details of this are beyond the scope of this article.) With a broadband transformer, termination resistors are required to minimize ringing and resonance.

Why do all this to generate a PWM signal? Generating a linearly modulated PWM signal at high frequencies is very difficult. One typical approach is to use a fast comparator with a triangular waveform on one input and the baseband modulating signal on the other input. The difficulty with this approach, and with many logic or one-shot methods is, the upper frequency of operation is limited by the propagation delay or speed of the logic or comparator. When the frequency is high enough that the logic propagation delay becomes significant, phase noise and distortion due to jitter become problematic, especially at the lowest duty cycles.

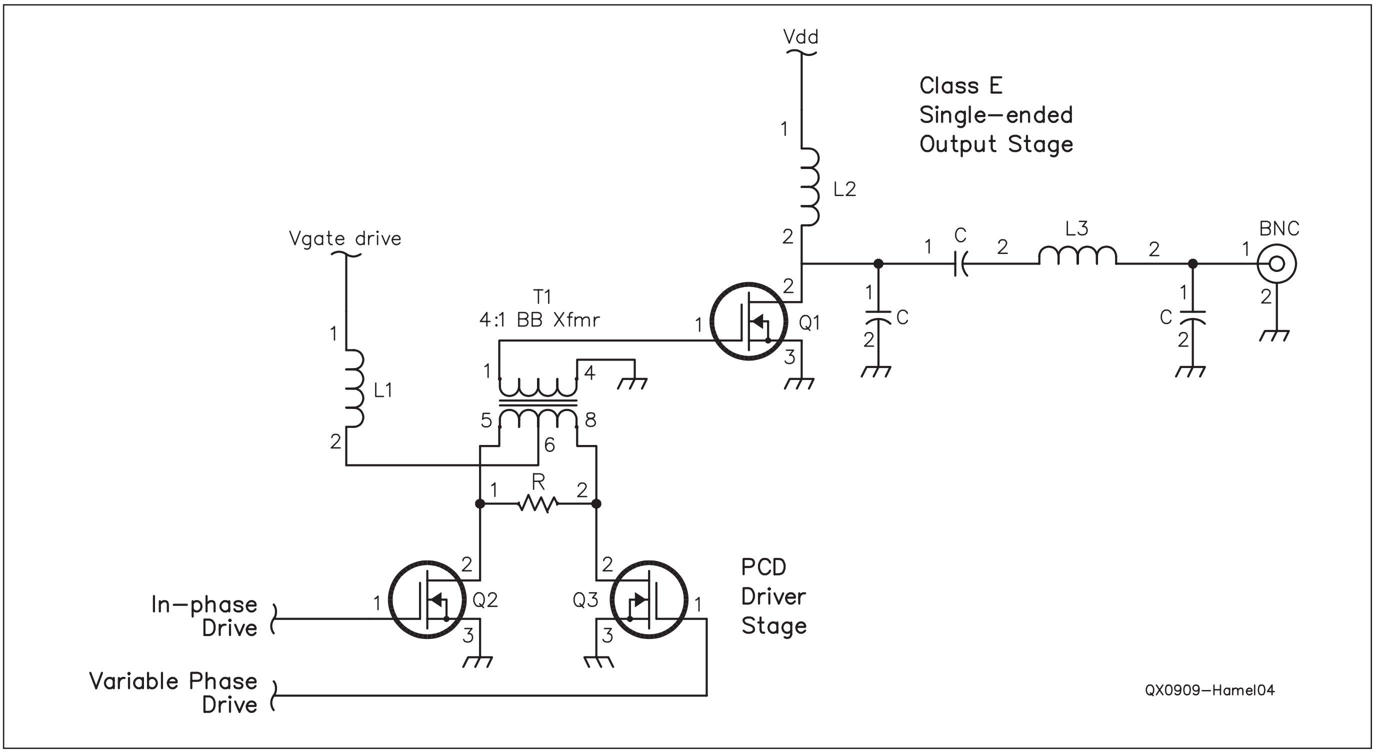


Figure 4 — This schematic diagram shows a PCD driver circuit for use with a typical Class E topology final stage.

These problems can be mitigated using a PCD stage to develop a broadband drive signal, rather than use it in a tuned final amplifier. Figure 4 shows such a PCD driver configuration with a typical single-ended Class E topology final stage. An important difference here is that the transformer in the PCD circuit is untuned and broadband. This is because the gate drive of the Class E stage does not require sinusoidal drive, and more importantly, it is very desirable to minimize the number of band specific tuned components when considering building an amplifier for multiple bands. This PCD pulse width modulator exploits the fact that it is much easier to control the phase of two signals than it is to control the propagation delay of saturated logic or comparators.

This PCD technique provides a different approach to low level modulation of saturated power amplifiers. PCD does not solve the problem of the abrupt transition from no signal between syllables in an SSB signal, with enough amplitude to fully saturate the output devices. This would be a problem in an external stand-alone EER linearized amplifier configuration. That issue could, however, be solved in the digital domain with software defined radio (SDR) technology. Another challenge with this implementation of PCD is the linearity of the phase modulator. It might be possible to compensate for nonlinearities in the phase modulator by using a closed loop, in which the output power is sampled and compared to the baseband drive signal of Figure 2, similar to a feedback bootstrap in audio amplifiers. PCD could also work well for Class E AM transmitters, eliminating the expensive modulator components.

Notes

[1]N. O. Sokal and A. D. Sokal, US Patent 3,919,656, 1975.

[2]Frederick H. Raab, "Idealized Operation of the Class E Tuned Power Amplifier," *IEEE Transactions on Circuits and Systems*, Vol CAS-24, no. 12, December 1977, pp 725-735.

[3]Leonard Kahn, "Single Sideband Transmission by Envelope Elimination and Restoration," *Proceedings of the IRE*, July 1952, pp 803-806.

[4]Herbert L. Krauss, Charles W. Bostian and Frederick H. Raab, *Solid State Radio Engineering*, J. Wiley and Sons, New York, 1980.

Mike Hamel, WO1U, has been a ham since 1977 and holds an Extra class license. Mike's ham radio interests include PSK31, resurrecting derelict HF parts radios, and longing for more sunspots. He has been an RF Engineer for the last 25 years, and holds 11 patents for energy harvesting converters and wireless powered sensors. His interests include flying, motorcycles and music.

QEX

Bob Miller, KE6F

9655 Appalachian Dr, Sacramento, CA 95827; MillerKE6F@aol.com

Atomic Frequency Reference for Your Shack

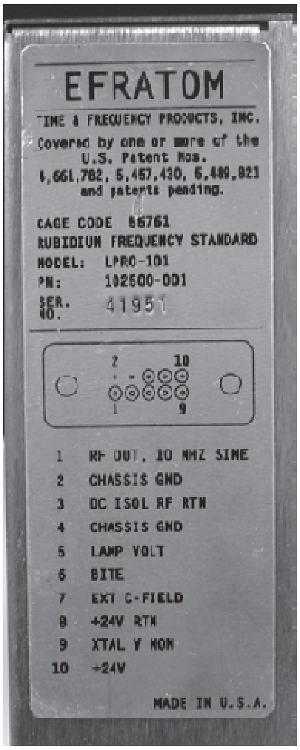
A quest for frequency accuracy and stability led to this atomic clock, along with some practical applications.

It's been the dream of many hams and electronics experimenters to have the precision of an atomic clock available in their shack or shop for many decades. Many schemes have been implemented to capture the low frequency transmissions from WWV to establish a stable reference greater than can be found with oven stabilized crystal oscillators. Propagation problems and noise can make this type of reception problematic in many areas of the US. Several schemes have been published where the atomic referenced signals from GPS satellites are stripped from the orbiting package signal, but that too presents a number of technical challenges for hams and hobbyists. While frequency accuracies around 10 parts in 1 million usually suffice in traditional Amateur Radio activities, the continued improvement in narrowband communications schemes such as PSK31 and its derivatives are pushing on the accuracy and stability envelope.

Most SSB and CW equipment in use today exhibit stabilities better than 100 parts per million, and calibration accuracies better than 10 Hz in the HF bands. With digital communications embracing narrower and narrower bandwidths, stability and accuracies to less than 1 part in 100 million are not that far fetched. Finally, there is just the quest and the fun of fine frequency measurement. At the heart of any frequency measurement or generation system is the master clock. This article will speak to the construction of and practical applications for a rubidium based reference signal generator.

Many rubidium oscillator units have been showing up on the surplus market of late, with many coming from the upgrading of cell phone sites. Functional units made by Efratom Ball are being sold on the Internet by Chinese entrepreneurs and many others

Figure 1 — PhotoA shows the Efratom Ball FRS Rubidium Oscillator name label. This is the unit used in this project. Photo B shows the special connector used with these units. A mating connector is available from Mouser.



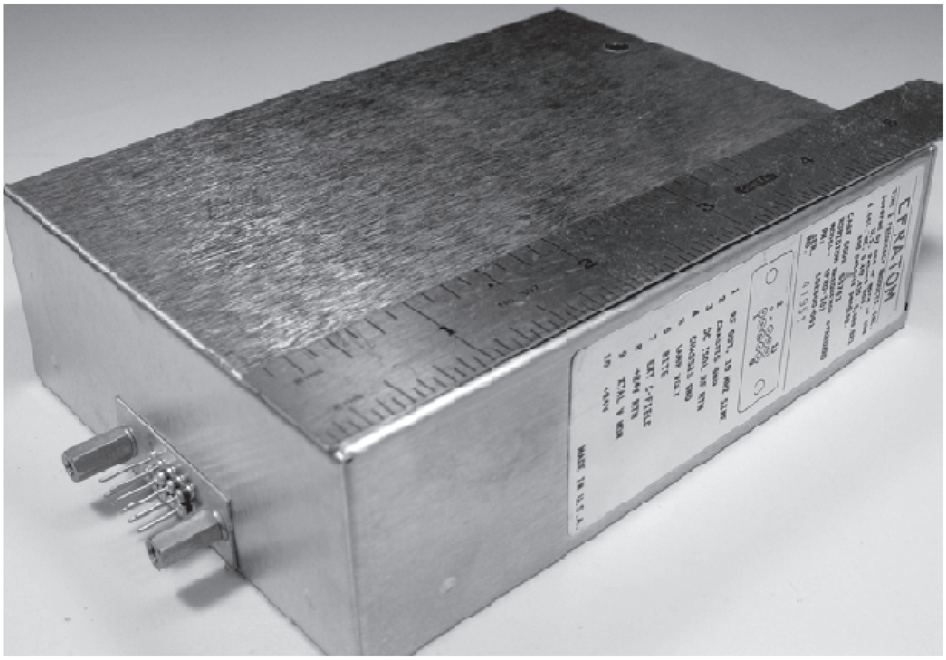


Figure 2 — Photo A is the LPRO-101 name label. Photo B shows the physical dimensions and the connector on the side of the LPRO-101 unit.

stateside. Prices for the popular FRS model range from $60 to $125. Figure 1A shows the surplus FRS module used in this project.[1]

This article will address the procedures used to adapt the unit as a generalized piece of test equipment to include power supplies, digital divider logic and buffered outputs for shop distribution. Also included is a simple netting system whereby the Rubidium "Physics Package" can be calibrated with a primary source if extreme accuracy is desired or it can be used as a long period frequency comparator that allows for calibration periods many magnitudes finer than most frequency counters will display.

Fundamental testing of the FRS unit can be accomplished with a bench supply capable of providing 24 to 28 V dc at 2 A. Figure 1A shows the connection decal that is nearly always attached to the surplus units. The connector is shown in Figure 1B. This connector will not be found at RadioShack. Mouser electronics, however, can supply the hybrid D connector with the miniature coaxial connector insert (Cannon P/N DAM11WIS). Testing can be performed without the special connector by drilling the center out of a DB-15 female connector. Positive power connections are made to pins 9 and 6. Power return is connected to pin 10. Signal output can be monitored with an oscilloscope and frequency counter by probing the center pin on the hybrid mini-coax connector. A voltmeter should be attached between pin 1 and pin 10 on the connector to monitor for a lock indication. When power is first applied a voltage level of around 5 V dc will appear and within a few minutes that voltage will drop to near zero indicating that the unit has locked to the rubidium source. Typical lock times are between 3 and 7 minutes. If the unit does not lock, contact the vendor for a replacement. Do not run the unit in a test configuration for extended periods without a proper heat sink attached. There is no need to connect the calibration potentiometer to the frequency adjust terms, as the unit centers that control voltage internally when the external adjustment circuit is not connected. The signal output should resemble a sine wave with a level of around 0.5 V RMS. Typical frequency measurements on used units will generally exceed the accuracy and resolution of most counters available to hams.

Another rubidium source that is finding its way to the surplus market is the Efratom Ball LPRO-101. It is similar in operation to the earlier FRS units and will work in this project without any significant changes. The LPRO-101 replaces the hybrid connector with a small connector consisting of two rows of wire wrap style posts. The only electrical difference is in the "Physics Package" calibration scheme. The FRS supplies a regulated source for the calibration potentiometer where the LPRO-101 relies on an external 5 V dc regulated source for that potentiometer. Like the FRS unit, all connections to the unit are silk screened on the case. The BITE terminal (Built In Test Equipment) is the logical term for the lock signal and goes low when the package locks up to the rubidium source. The Timenuts internet site has an excellent tutorial on the LPRO-101 rubidium source. See Figure 2 for photos of the LPRO-101 data label and mechanical properties.

To keep the cost of this unit within the reach of a large sector of the Amateur Radio population, extensive use of surplus materials were made. The only new parts used in the scheme are the few integrated circuits, the front panel, and the PCB. Everything else came from surplus resources. Figure 3 shows the completed package.

Unit's Capabilities and a Bit of Perfunctory Theory

The rubidium source used in my project was manufactured by Efratom Ball and relies on the natural resonance of optically excited rubidium 87 to establish a reference frequency of 6.8346875 GHz. Through a process of dividers, multipliers and mixing schemes a 20 MHz VCXO is phase locked to the rubidium signal with a long term stability greater than a few parts in 10^{-9} and with adjustments to the internal "Physics Package C Field," accuracies greater than a few parts in 10^{-11} are possible for short term measurements. The 20 MHz VCXO is divided by 2 for a final output signal of 10 MHz.

While a stable 10 MHz source is useful as a calibration source, additional hardware was designed to support other functions, such as a

[1]Notes appear on page 43.

digital divider system that provides a number of useful square wave test signals. These signals are brought out to BNC connectors on the unit's front panel at CMOS compatible levels. All front panel signals are isolated via low value resistors to protect the drivers from shorted test leads.

The following signals are available:

- 10 MHz
- 5 MHz
- 2 MHz
- 1 MHz
- 100 kHz
- 10 kHz
- 1 kHz
- 100 Hz
- 10 Hz
- 1 Hz

Front panel indicators provide an "I'm Alive" signal at a 1 Hz rate and an indicator confirming that the rubidium module is locked. A ten-turn surplus Helipot® is also brought out to the front panel to allow for fine frequency adjustment of the rubidium

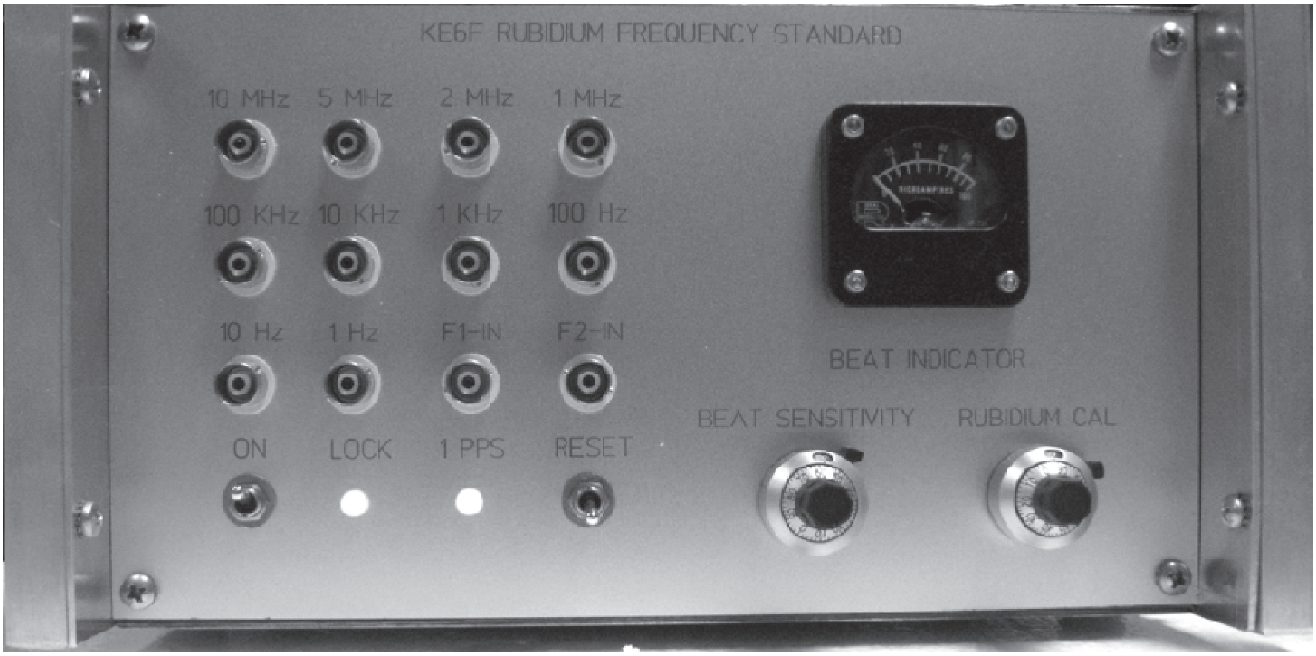


Figure 3 — Completed package with FRS module locked and 1 PPS indicator caught in the On Cycle of the 1 Hz signal.

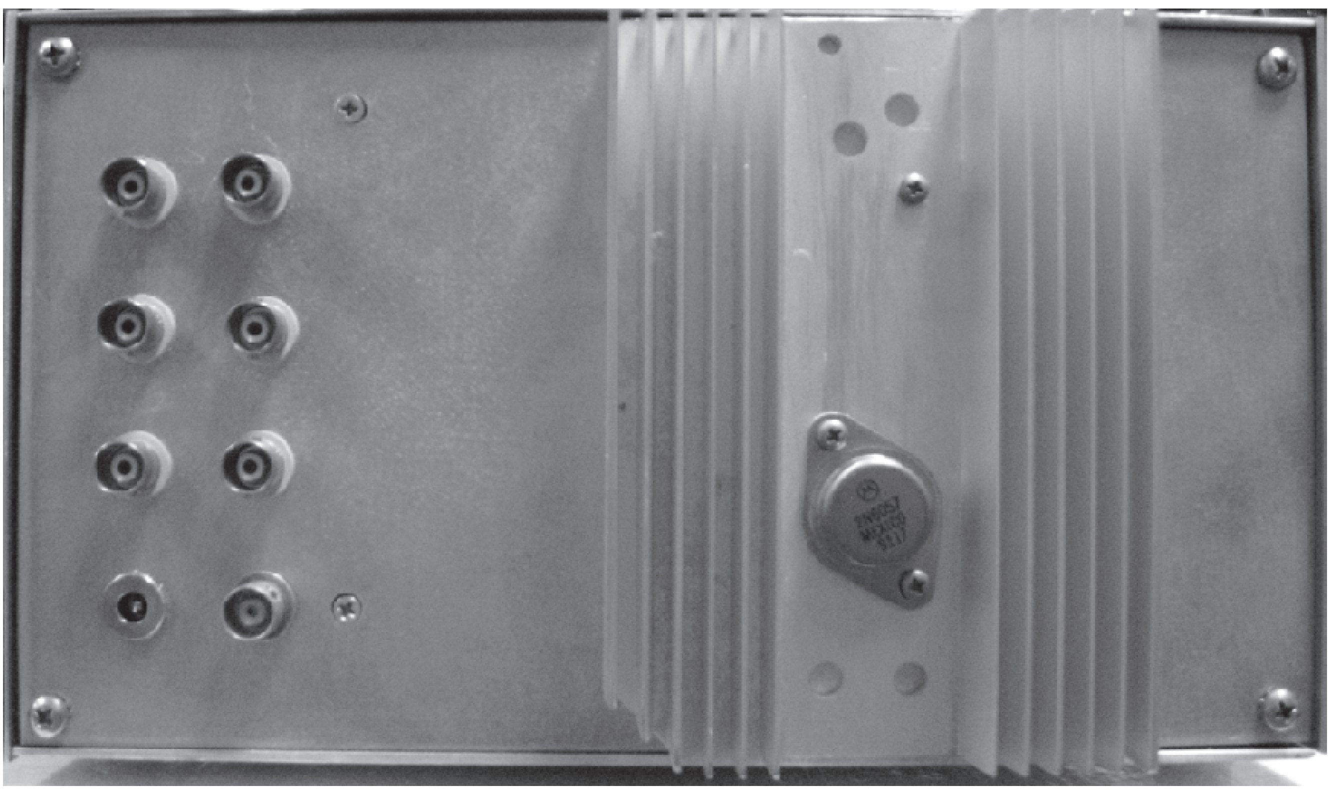


Figure 4 — Back panel connections for 10 MHz, 5 MHz and 1 PPS Signals. The transistor in the heat-sink is not active, but plugs up a hole.

"Physics Package C Field" when a primary standard is available as a reference. A complete explanation of the rubidium standard operation and calibration is available on-line from a number of sources.[2] To verify my unit's accuracy to 1 part in 10^{-9} I took it to a colleague who had access to a GPS locked rubidium clock, where I used the 100 second beat count at 10 MHz to ensure my unit was more accurate and more reliable than my aging quartz based standards. I made note of the calibration Helipot reading and locked the Helipot. I use the 100 second beat technique later in this article to measure and calibrate crystal oscillators and TCXOs.

Additional signals are brought out to the rear of the unit to provide reference signals for other frequency sensitive equipment in the shack such as frequency counters, synthesizers, signal generators, and other equipment that would benefit from an accurate clock source. The following back panel signals are available:

- 10 MHz (three BNC ports)
- 5 MHz (three BNC ports)
- 1 Hz (one BNC port)

Figure 4 illustrates the back panel signal connections, without labels.

Power Requirements

The primary consumer of power in the unit is the FRS module. Initial load current approaches 1.8 A at 24 V when the unit is first powered up. As the internal heater stabilizes, the current drain drops to approximately 500 mA. An external surplus power module (24 V dc at 2.5 A) was used in this application. The divider and interface logic is powered from the 24 V dc source via an LM7805 three terminal regulator in a TO-220 package. A small-tab heat sink keeps the regulator IC temperature in the safe zone.

Interface Requirements

The output of the FRS rubidium oscillator used in this project is a 0.5 V RMS sine wave. A level shifter in the form of a simple bipolar transistor inverter was used to convert this signal to a CMOS 5 V square wave. The circuit design for the interface was selected from a design in the ARRL publication *Experimental Methods in RF Design*.[3] It was necessary to stiffen up the collector load line on the interface to facilitate driving the logic and front panel loads. The interface drives both the multi-decade divider chain and the buffer array for direct distribution of the 10 MHz signal to other test equipment in the shop.

Environmental Considerations

The FRS module requires a heat sink mechanism to keep the unit's base-plate temperature below 70°C. With a turnover point of 25°C, the unit's accuracy will deviate $\Delta f/f$ of -1×10^{-10} at –25°C to $\Delta f/f$ of $+0.8 \times 10^{-10}$ at 70°C. The heat-sink and mounting system used in the unit keeps the FRS module temperature to around 25 to 30°C.

Construction Considerations

Most of the cabinet was built from scrap aluminum and some of the internal items were glued to the metal surfaces using E6000 clear adhesive to include barrier terminal strips, digital divider and driver boards, beat comparator board and many of the cables. The FRS module was bolted to the rear panel and a surplus heat sink was attached to the outside of the rear panel to dissipate heat generated by the FRS module. All BNC connectors and associated mini-coaxial cables were retrieved from surplus Ethernet coaxial cable distribution panels.[4]

The printed circuit board(s) that make up the divider and buffer system were designed using free software provided by Expresspcb.com and fabricated by that same company using their on-line fabrication facility.[5] SMT parts were used in the design of the multi-decade divider and signal buffer assemblies; the single exception being the 2N2222A translation interface that converts the 0.5 V RMS signal for the Rubidium oscillator to 0 to 5 V dc CMOS levels. The divider chain is made up of 74HCT390 dual decade ripple counters. Each decade counter is configured as a cascaded divide by 5-divide by 2. The second section in the third 74HCT390 provides 5 MHz and 2 MHz signals, where the two sections are driven by the 10 MHz source independent of the decade divider chains. The 5 MHz signal is a symmetrical square wave, but the 2 MHz signal, while delivered at a repetition rate of 2 MHz, produces square wave pulses with periods equal to 10 MHz. All other output divider module signals are symmetrical square waves. See Figure 5 for divider board detail and Figure 6 for the schematic.

Surface mount components were used in

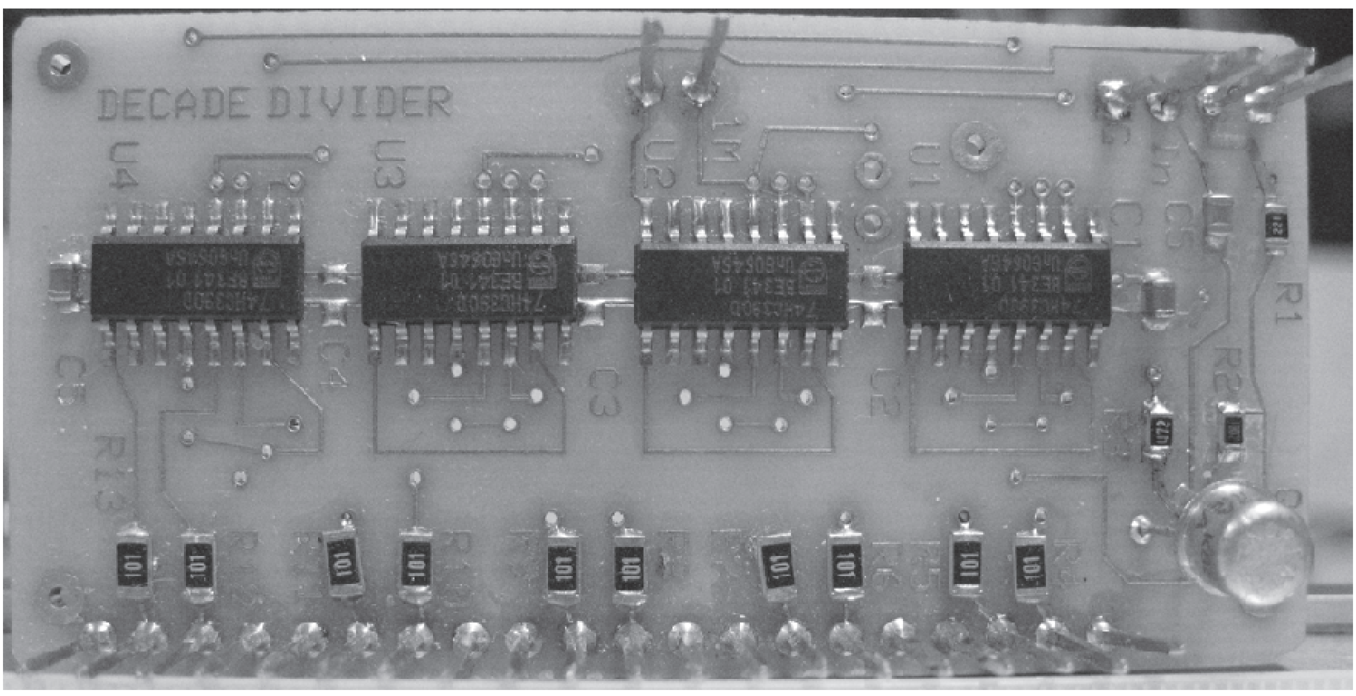


Figure 5 — Multi-Decade divider board.

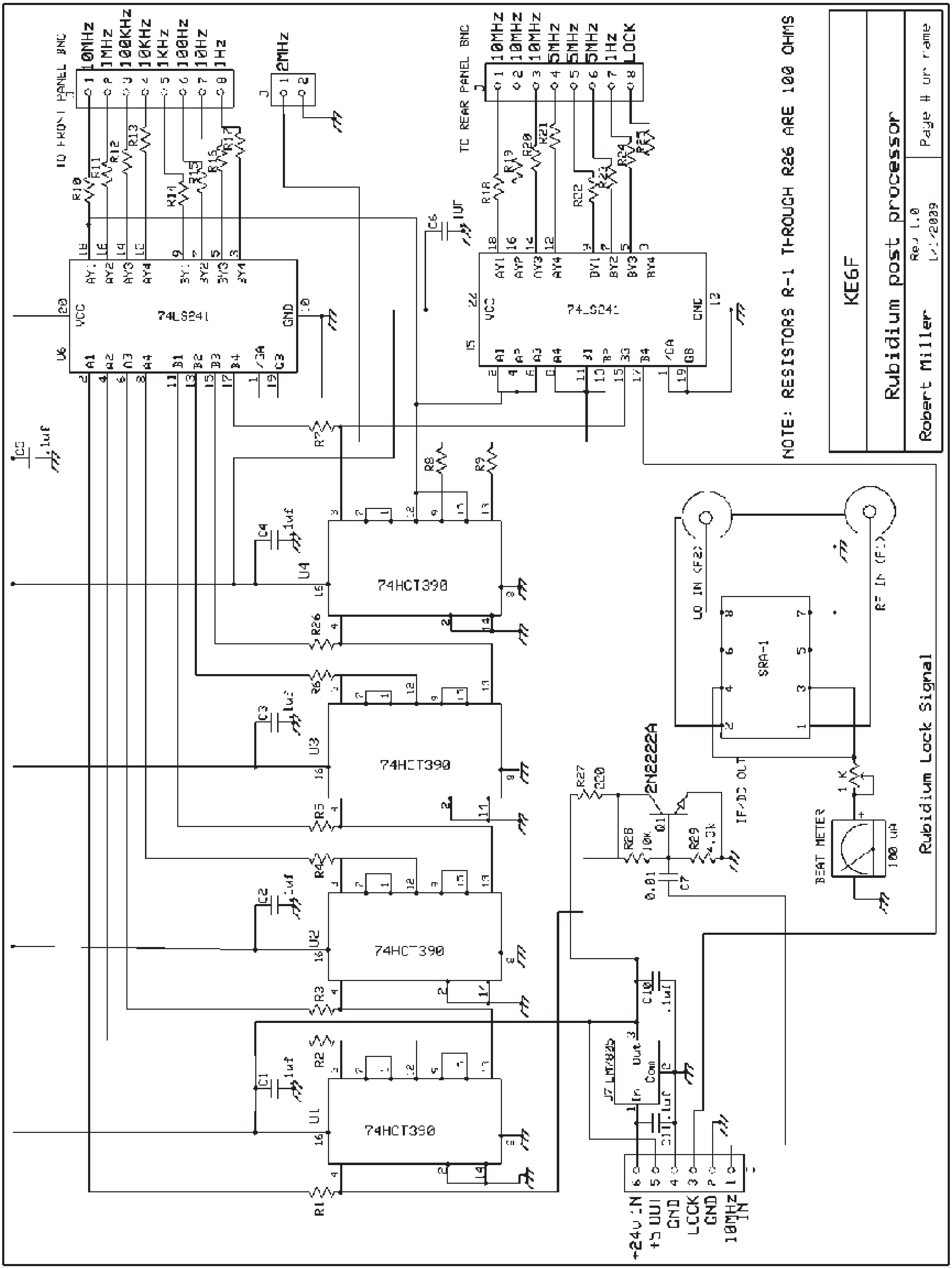


Figure 6 — Schematic detail of Divider and Buffer circuits.

the design of the multi-decade divider PCB, but standard leaded components would work quite well using wire wrap vector board construction. The lead pitch on the SMD parts is 0.05 inches and can be soldered with conventional soldering techniques with a little care. The layout is not that critical although rigorous bypassing of power connections to the active components should be embraced.

The inside of the unit is spacious enough to hold a pair of 12 V gel cell batteries if the unit is to be used in the field or as a transfer standard where the unit is calibrated at a primary source lab and then transported HOT to a final destination, where other items are to be calibrated. See Figure 7 for a detailed view on the unit's internal construction.

The front panel was designed using free software provided by frontpanelexpress.com who also fabricated the engraved front panel at a cost of around $50.[6] The 74HCT390 dual decade divider integrated circuits and the 74HCT241 octal inverting buffer were purchased from Mouser Electronics. The specialized D connector with integral mini-coaxial connector to mate up with the FRS module was also purchased from Mouser Electronics and is a bit delicate to fabricate so do not rush that process.[7]

How Accurate are Rubidium Based Frequency Standards?

Stabilized crystal oscillators have been perfected to the point that accuracies in many cases come close to those demonstrated by the small rubidium unit used in this project. One problem facing the crystal oscillator solution is the long term aging of the quartz element, and that in itself is why the rubidium units are seeing applications in many systems that heretofore relied on crystal oscillator technology. The rubidium oscillator solution is not without its own accuracy considerations. Frequency variation with the unit's physical temperature has already been noted. Short-term variations referred to as Allan Deviation can lower accuracy to around 1×10^{-10}. Data in graphs cited derived from the later LPRO rubidium oscillator design, is typical of these families of oscillators. See Figure 8 for the Allan Deviation graph.

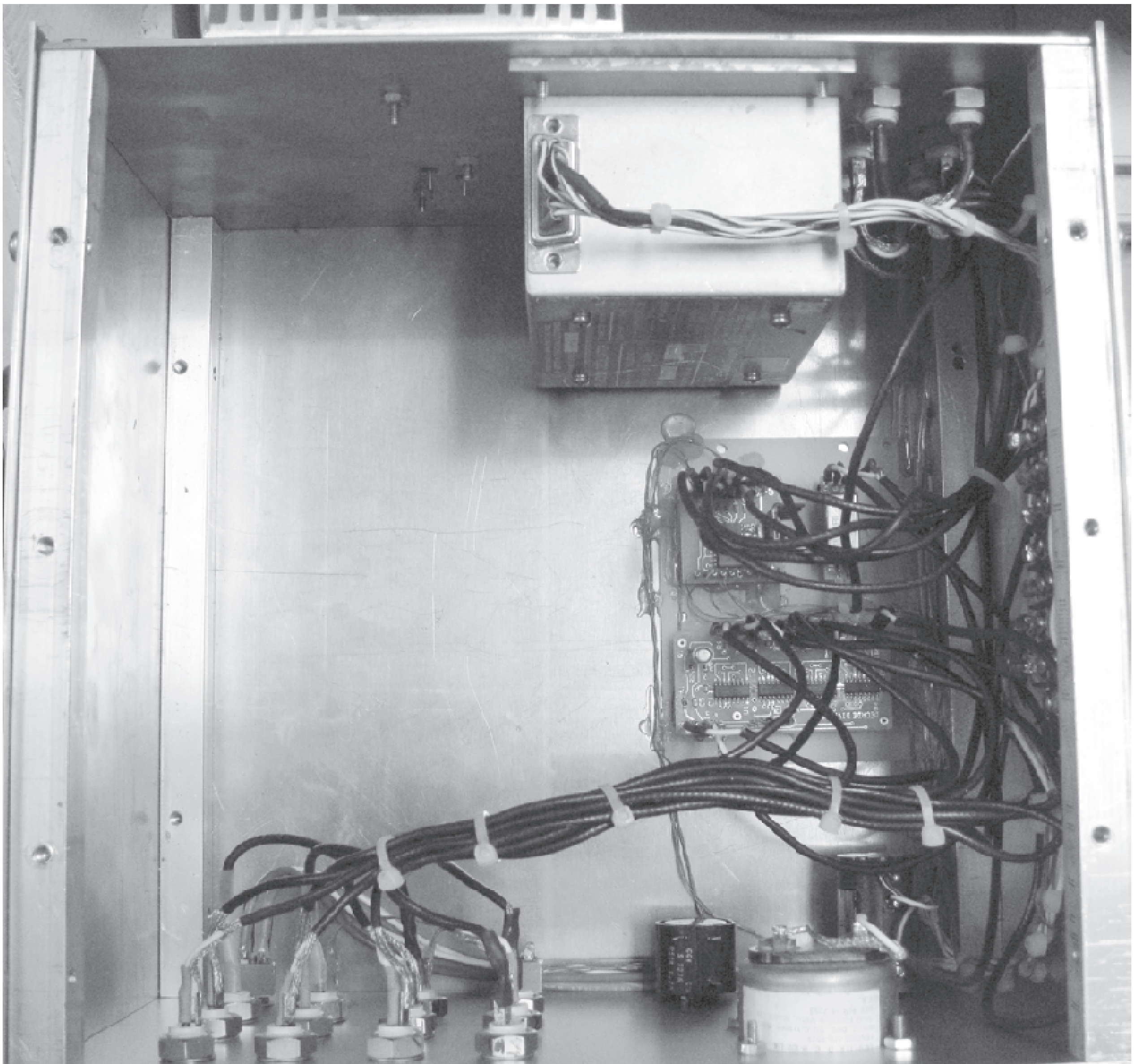


Figure 7 — Internal view of the unit showing the FRS module mounting and Internal cabling.

The graphs in Figures 8 and 9 illustrate the need to make long-term measurements to take advantage of the excellent accuracy these handy little frequency reference units can provide. See Figure 9 for data on short-term frequency excursions.

There is also quite a bit of difference in short-term performance between different brands and models of rubidium oscillators. See Figure 10, which compares a number of rubidium oscillators. Most of the difference occurs in the first 10 second averaging and by 100 seconds all units provide accuracies in the 1×10^{-12} to 1×10^{-13} region.

Applications (The Practical Side of Life)

Unit Provides Oscilloscope Time Base Marker Calibration Test

Many modern oscilloscopes provide convenient on-screen markers that enhance readout accuracy and reduce ambiguity both in period and amplitude measurements. Validation of scope period calibration is easily done by a simple measurement of the precision square wave signals produced by the unit. Figure 11 illustrates a 1 MHz signal being used to check the calibration markers on a Tektronix TAS 250 oscilloscope.

Unit Provides Precise Comb Markers for Spectrum Analyzer

Spectrum markers are often handy while doing spectrum analysis. Figure 12 shows how a 1 MHz signal from the unit provides a series of marker pips across the screen's display.

High-Resolution Calibration Capability

Most frequency counters aimed at the hobbyist or experimenter rarely offer resolutions greater than 1 Hz and for that matter many industrial grade instruments made by Hewlett Packard and others offer more than 1 Hz or 0.1 Hz resolutions. Resolution in these cases is predicated on the fundamental accuracy of the subject counter's time base and measuring a 100 MHz signal to 1 Hz represents a potential accuracy of one part in 10^{-8} ±1 time base accuracy, ± 1 count ambiguity in many designs, and count ambiguities attributed to noise on the signal being measured. Ten digits of resolution are meaningless if the fundamental accuracy of a unit's time base will only support 8 digits accurately.

The unit's internal beat meter is based on a balanced modulator using a Mini-Circuits SBL-1 and an analog meter. The scheme provides a simple but effective mechanism for comparing any of the unit's signal outputs with an external signal. While somewhat archaic in its methodology, it permits calibration of precision oscillators with resolutions in the millahertz (mHz) region. See Figure

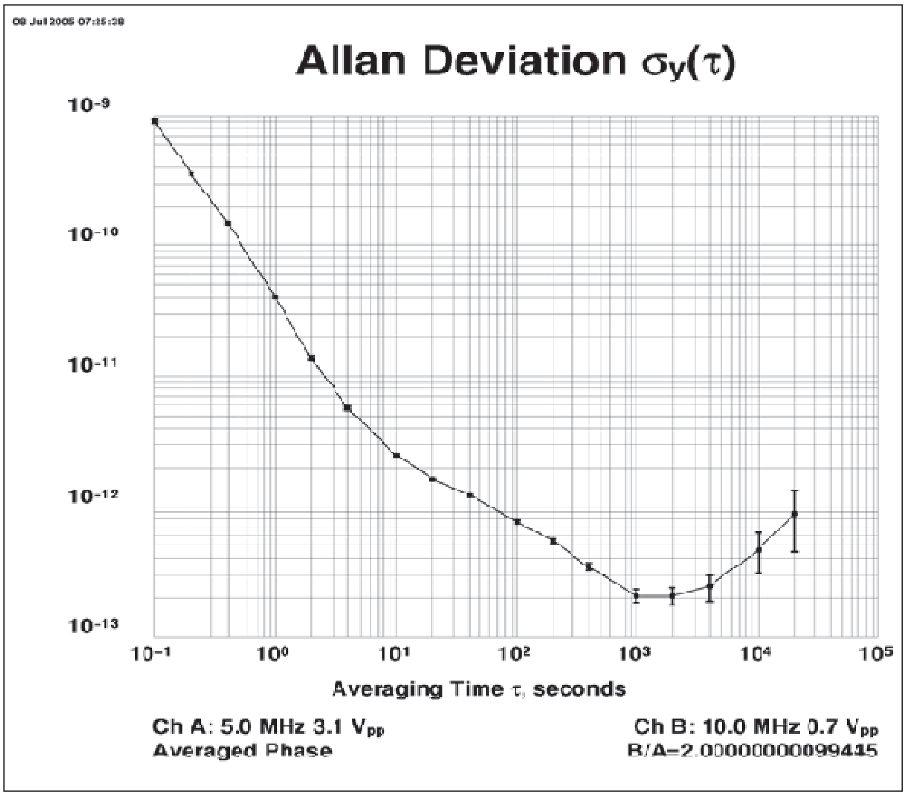


Figure 8 — Impact of averaging on Allan Deviation. The graph shows the oscillator stability from τ = 0.1 s to τ = 20 ks. Data taken from www.leapsecond.com/museum/lpro/.

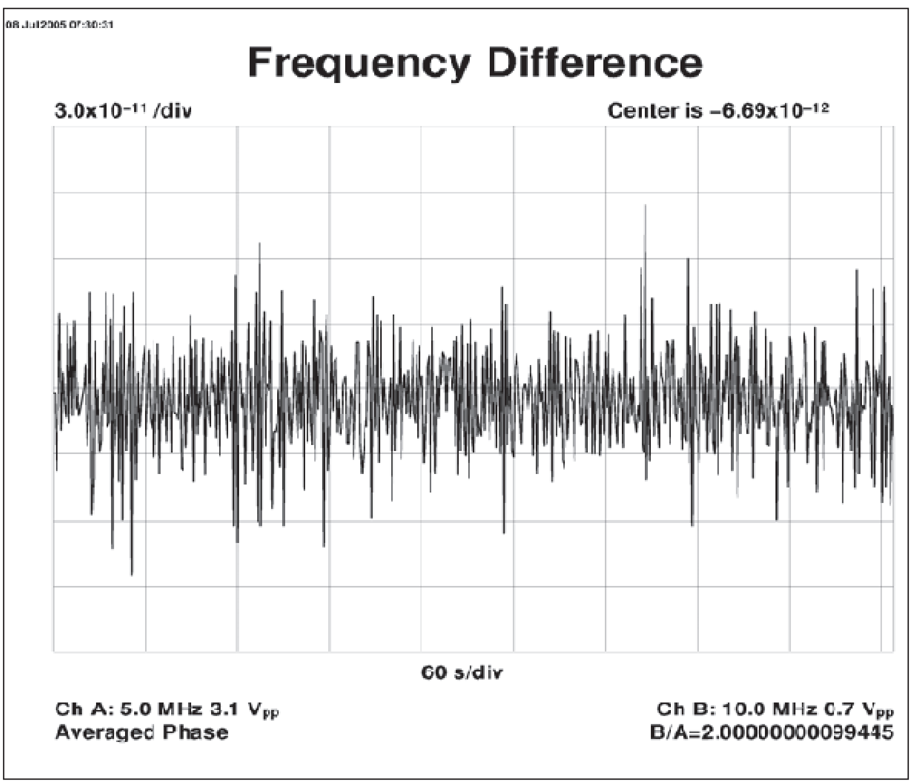


Figure 9 — Data taken from www.leapsecond.com/museum/lpro/ shows that over a ten-minute period excursions can approach $\pm 9 \times 10^{-11}$.

13 where a 1 MHz external signal is being compared to the 1 MHz signal from the unit.

The beat method relies on counting the full cycle movements on the beat meter. When the two signals are connected to the unit as noted in Figure 13, the meter will wiggle at a rate that is proportionate to the error of the measured device. In this case a full cycle of the meter every 10 seconds would indicate an accuracy of 1×10^{-7} and of one cycle in 100 seconds the accuracy

Distribution of 10 and 5 MHz signals in a Shop or Lab Environment

The frequency counter, synthesizers, function generators, and signal generators in my shop all have fairly accurate time bases. A few have oven stabilized crystal oscillators as their primary reference while others have temperature compensated crystal oscillators (TCXO) where the crystal is open to the environment and kept close to frequency with temperature sensing electronics. The problem with this arrangement is that while these units provide yeomen service for many applications, rarely do they agree with each other even at the 1 Hz accuracy level. The rubidium standard resolves this problem quite nicely by supplying a common 10 or 5 MHz reference signal to the equipments cited. The distribution buffer amplifiers in the unit are not up to the line driving standards of commercially available units, but satisfy most of my testing needs and certainly remove any ambiguity as to what item of test equipment is correct while making tests. See Figure 14 for an example a shop-lab distribution system.

Conclusions

This project has provided me with many benefits, to include: 1) a much better understanding of how rubidium standards operate and how to adapt them to applications associated with the Amateur Radio hobby; 2) honing skills applicable to SMD PCB design and layout and hot air soldering; 3) shop practices associated with mechanical layout, design, and fabrication of enclosures; 4) better maintenance of lab test equipment using the rubidium standard; 5) and finally a lot of fun trouble shooting the various low level to high level interface mechanisms that did not work as planned. Other uses around the shack include the following:

- Primary source for Direct Digital Synthesis such as the AD9850/51 chips and others in the Analog Devices DDS family.
- Stable source for microwave phase lock loops for SSB and where other frequency sensitive modes are used
- Synchronizing mechanism for signal detection schemes in high noise environments or signal detection below the ambient noise.

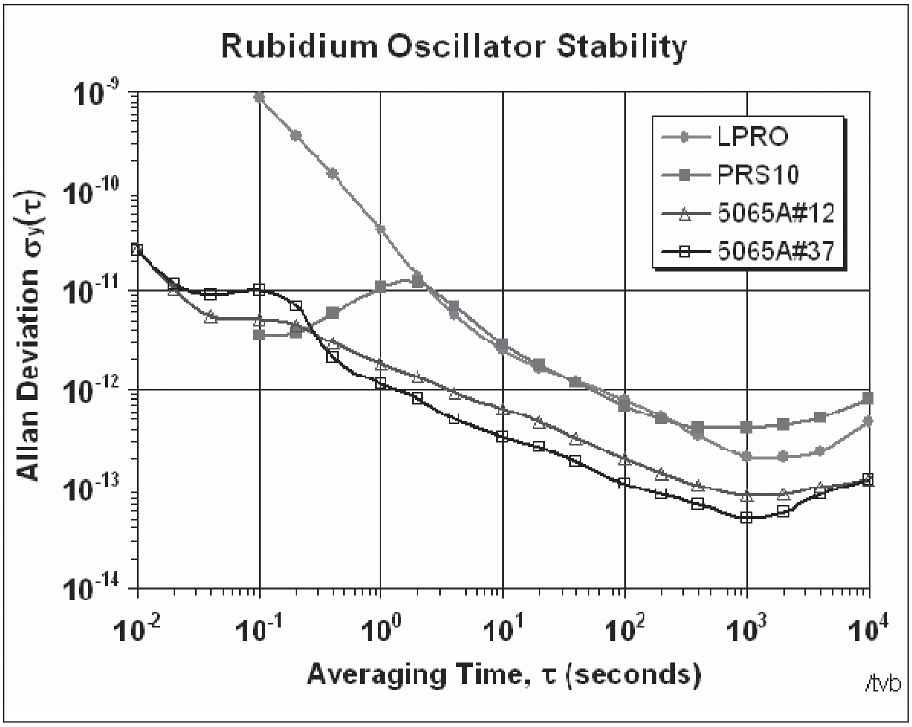


Fig 10 — Comparison of different rubidium oscillator units short term performance. This graph is from **www.leapsecond.com/museum/lpro/**.

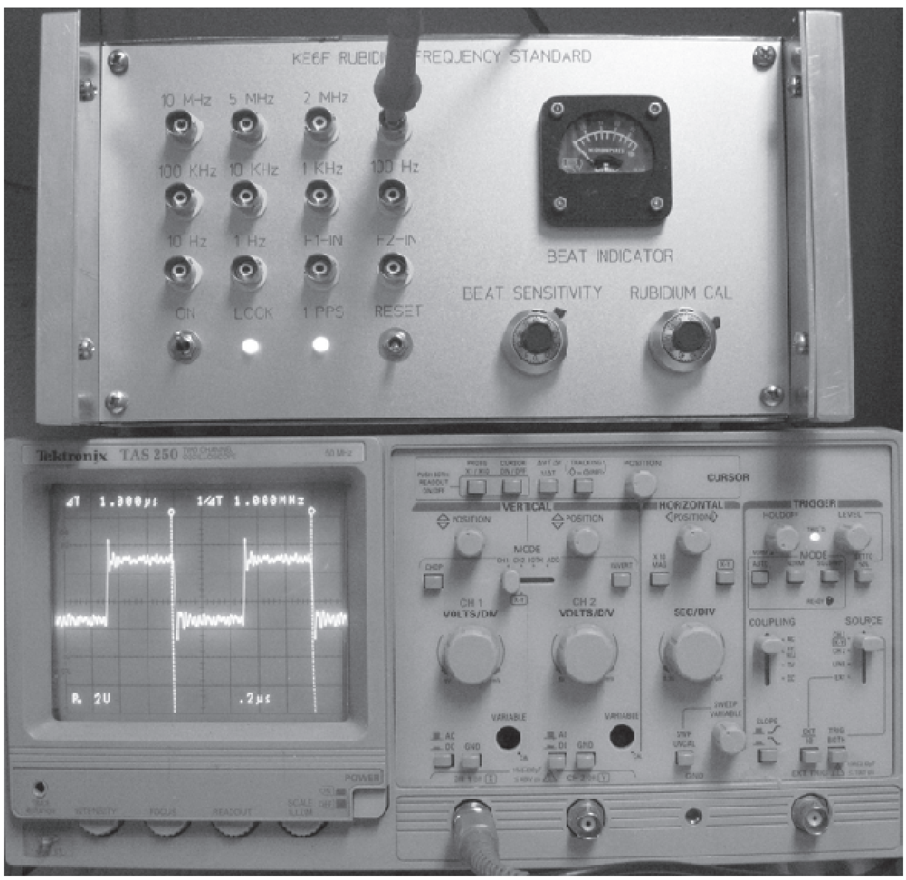


Figure 11 — Checking calibration of scope markers with a 1 MHz square wave signal.

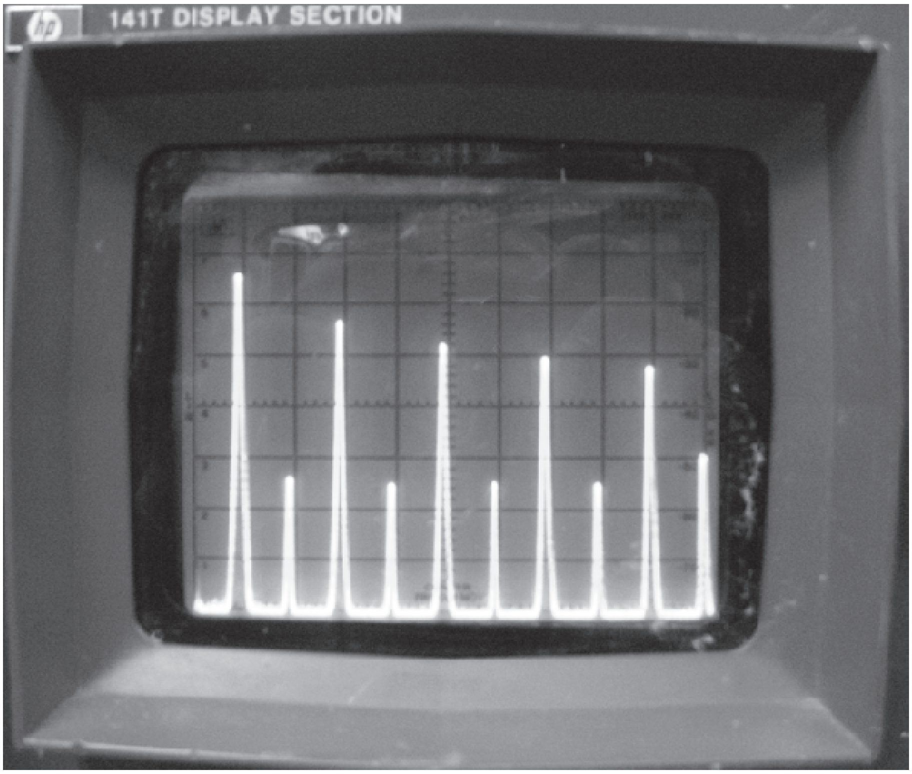


Figure 12 — Spectrum analyzer shot of 1 MHz pulse illustrates the odd-even distribution of a square wave energy.

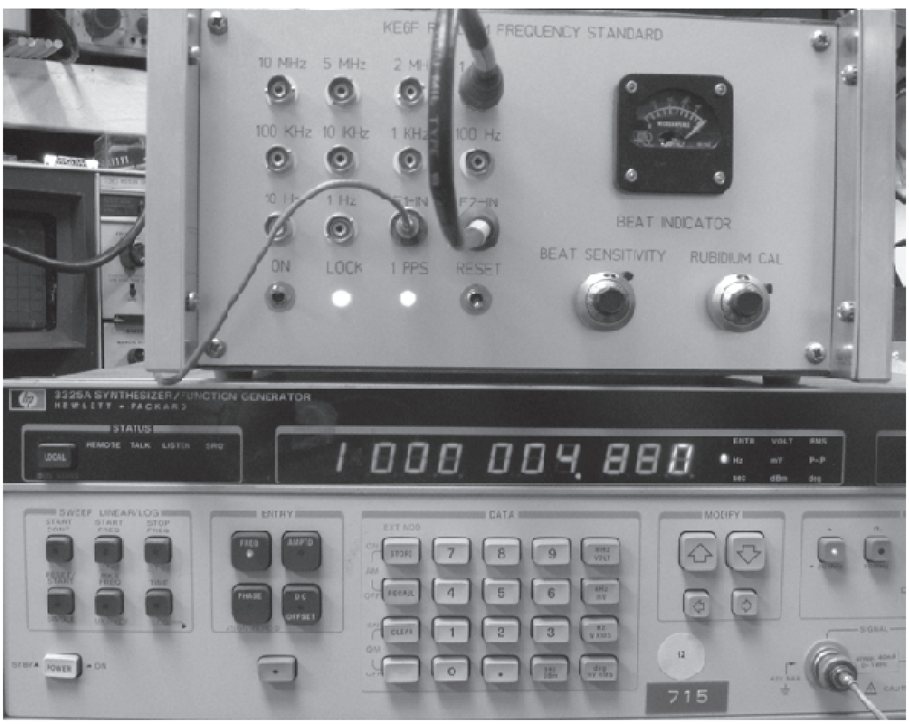


Figure 13 — Beat indicator being used to calibrate a 1 MHz oven stabilized synthesizer. The 1 MHz standard is applied to F2-IN and the signal from the synthesizer is applied to F1-IN. Adjusting the synthesizer frequency for a zero beat indicated that the internal oven stabilized reference was actually defective (it would not adjust on) with an error close to 5 Hz at 1 MHz.

• And of course the one-second pulse could be routed to a cheap digital wall clock for locally generated atomic time. Imagine having a wall clock that is never more than one second off in your lifetime!

Notes

[1]Internet source for rubidium FRS or LPRO oscillators: go to **www.ebay.com**, and search on rubidium. Also reference LPRO-101 manual at **www.2917.com/EBAY-images/LPRO-101/LPRO-101.PDF**

[2]Rubidium bibliographic: **www.leapsecond.com/museum/lpro/**

[3]Wes Hayward, W7ZOI, Rick Campbell, KK7B, and Bob Larkin, W7PUA, *Experimental Methods in RF Design,* American Radio Relay League, Newington, CT, 2003. See the section on Digital Interfaces.

[4]The first unit built used isolated BNC connectors and that caused a number of RF radiation problems in my shack. They were replaced with grounded shell connectors that negated the problem.

[5]PC boards were designed using free software from **expresspcb.com** and then fabricated by that firm at **www.expresspcb.com**. You can download the circuit board files in expresspcb.com format from the ARRL *QEX* Web site. Go to **www.arrl.org/qexfiles/** and look for the file **9x09_Miller_PCB.zip**.

[6]The front panel was designed using free software from **frontpanelexpress.com** and then fabricated by that firm at **www.frontpanelexpress.com** The front panel lay out files in frontpanelexpress.com format can be downloaded from the ARRL QEX Web site. Go to **www.arrl.org/qexfiles/** and look for the file **9x09_Miller_Front.zip**.

[7]Mouser Electronics online mail order **www.mouser.com**

Bibliography

Dennison, Mike, G3XDV, Lorek, Chris, G4HCL, *Radio Communications Handbook*. Herts England: Radio Society of Great Britain, 2005.

Hayward, Wes, W7ZOI, Campbell, Rick, KK7B, Larkin, Bob, W7PUA, *Experimental Methods in RF Design.* Newington, CT, American Radio Relay League, 2003.

Wilson, Mark J., K1RO, Editor, *The ARRL Handbook*, Newington CT, American Radio Relay League, 2008.

Wolfgang, Larry, WR1B, Editor, *QEX: A Forum For Communications Experimenters*. Newington CT, American Radio Relay League, various articles, 2007-2009.

www.leapsecond.com/museum/lpro/

User's Guide and Integration Guidelines, Rubidium Oscillator. 2000. Applications note S/O/10250D

The following references were taken from the LPRO-101 Manual

Sullivan, Allan, Howe, Walls, Editors, *NIST Technical Note 1337*, "Characterization of Clocks and Oscillators," March 1990.

Droupa, V., Editor "Frequency Stability: Fundamentals and Measurement,", IEEE Press, 1983

"General Considerations in the Metrology of the Environmental Sensitivities of Standard Frequency Generators," IEEE Frequency

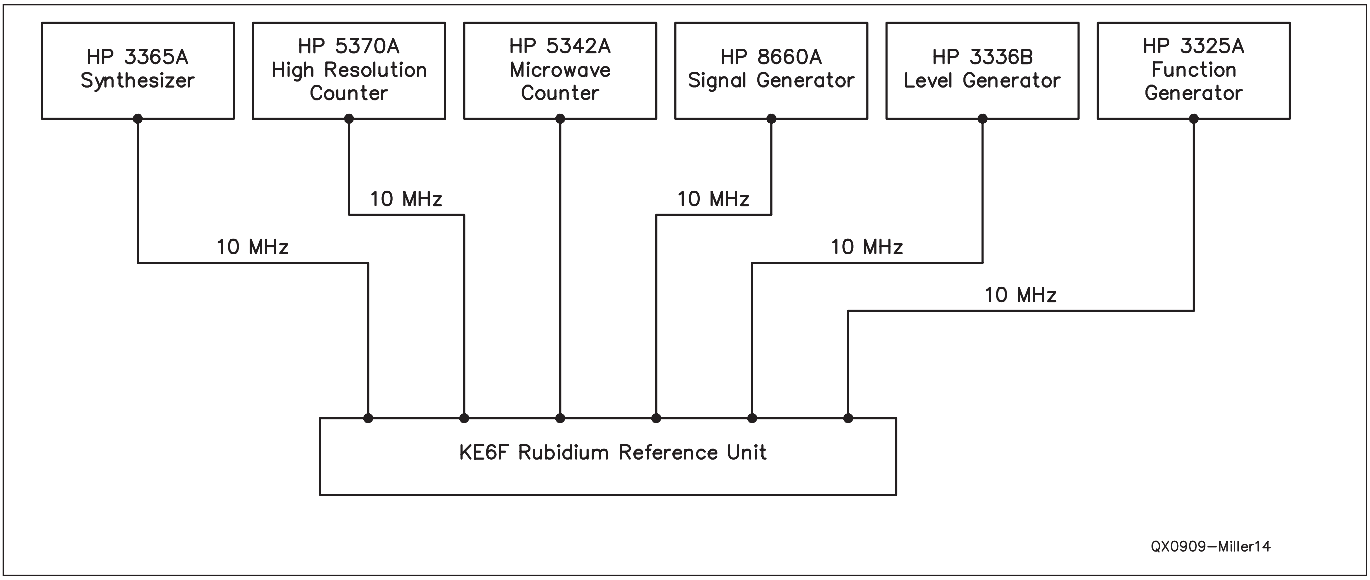


Figure 14 — The time base distribution scheme at KE6F.

Control Symposium, 1992, pp 816-830.

B. Taylor and C. Kuyalt, NIST Technical Note 1297, "Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results," 1994 Edition,.

DeWatts et al, "The Use of Statistics for Specifying Commercial Atomic Frequency Standards,", 1996, Frequency Control Symposium. iv

First licensed as WA6MGG in 1959, Robert M. Miller, KE6F, acquired his Advanced class license in 1967 and his Amateur Extra in the early '70s, along with a call sign change to KE6F. He has been active in nearly every phase of the hobby during the past 47 years.

Bob holds a BA in social science and an MA degree in history from California State University in Sacramento. Most of his working career has been within the telecommunications industry. He spent 12 years with the Western Union Telegraph Company in their Microwave and Space Communications group, spent the next 24 years with a local power utility in their telecommunications engineering group and a final four year stint supervising their communications maintenance group.

Bob has held an FCC First Class Radiotelephone license (while they were being issued) and currently holds a Second Class Radiotelegraph license with Ship Radar endorsement. Bob is also a retired member of the California Army National Guard after spending many years as a battalion communications sergeant. He enjoys power boating and water sports, reading and writing. He has published articles in CQ magazine and Communications Quarterly as well as numerous newspaper articles. QEX

Letters to the Editor

Fully Automated DDS Sweep Generator Measurement System (Nov/Dec 2008)

Dear Larry,

The circuitry for my article, "Fully Automated DDS Sweep Generator Measurement System" in the December 2008 issue of *QEX* requires modifications to operate properly with the 60 MHz DDS Daughtercard, which uses an AD9851. Figure 1 is a revised version of Figure 3 from that article. A pair of inverters in each address line replaces the pull-up resistors R1, R2, and R3. This arrangement prevents a problem that occurs when the parallel port powers up before the DDS, causing the DDS-60 to fail to start. The 30 MHz DDS uses an AD9850 and does not exhibit this problem. This same modification applies to Figures 6, 7 and 8 of the original article.

Input pin voltages on CMOS devices ought not to exceed the supply voltage of the package, as when the DDS card connects to the active computer LPT port with power off.

George Heron, N2APB, is designing a kit based on my article to aid hobbyists. The kit will be available on George's Web site: **www.midnightdesignsolutions.com**.

— 73, Dr. Sam Green, WØPCE, 10951 Pem Rd, Saint Louis, MO 63146; **wØpce@arrl.net**

Hi Dr. Green,

Thank you for sending along that correction.

— 73, Larry Wolfgang, WR1B, QEX Editor; **lwolfgang@arrl.org**

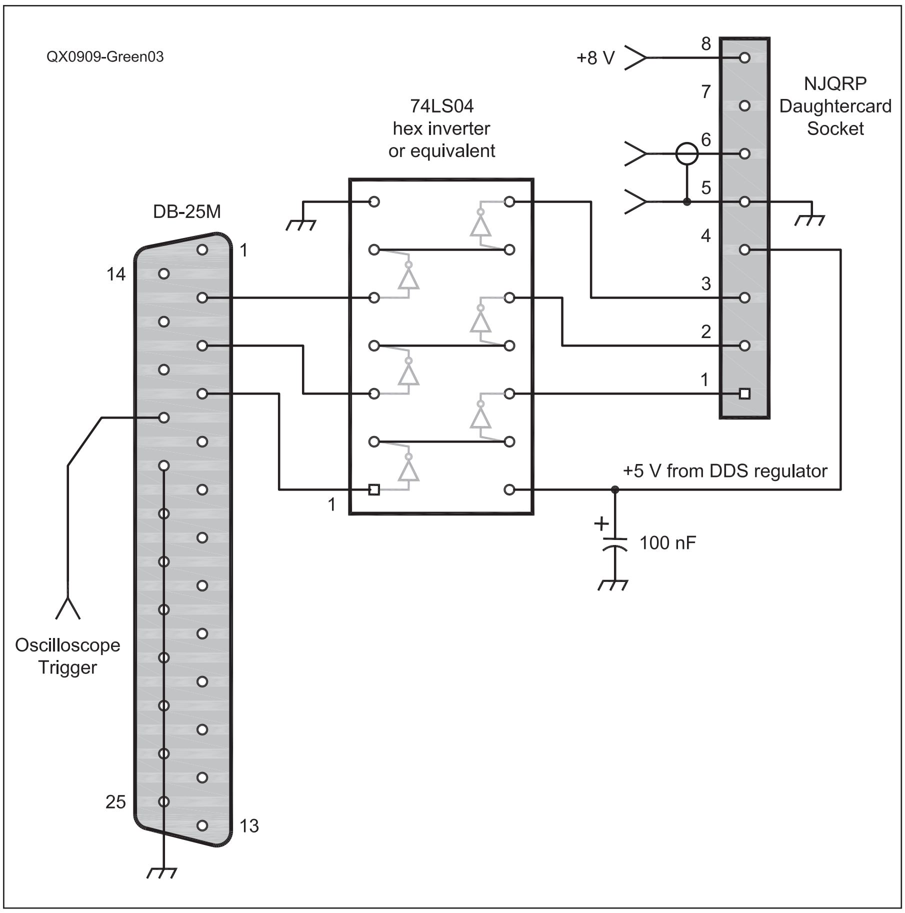


Figure 1 — This schematic diagram shows a revision to the circuit of Figure 3 in Dr. Sam Green's "Fully Automated DDS Sweep Generator Measurement System" in the Nov/Dec 2008 issue of *QEX*. The revision is only necessary if you are using the NJ QRP Club 60 MHz Direct Digital Synthesis DDS-60 Daughtercard. The modification involves replacing pull-up resistors R1, R2 and R3 with a pair of inverters in each data line. The same modification applies to Figures 6, 7 and 8 of the original article.

New Version of HAMCALC Program Collection

Hi Larry,

HAMCALC version 110, a single software collection of hundreds of menu-driven "painless math" design programs popular with radio amateurs, professionals and educators worldwide since 1993 is now available as a free download from **www.cq-amateur-radio.com**. There is a direct link to download the program collection on the CQ home page. There are detailed instructions given there for downloading and copying the files to your computer. I hope your readers will find these programs useful.

— 73, George Murphy, VE3ERP, 77 McKenzie St, Orillia, ON L3V 6A6, Canada; **ve3erp@rac.ca**

Hi George,

Thanks for passing along the information about the new version of your HAMCALC program collection.

— 73, Larry, WR1B

A Cybernetic Sinusoidal Synthesizer: Part 3 (Jul/Aug 2009)

Dear Larry,

I have just read Part 3 of Gary Steinbaugh's article in the July/Aug 2009 issue of *QEX* and am sympathetic to his plight in calculating the best resistor pair for his voltage divider.

I too have spent countless hours solving similar problems until I came across *Resistor CAD* a free *Windows* program by Terry Harris of Vader systems. It is small, lightning fast, and does not need to be installed. The program calculates series, parallel and divider pairs, and it knows all the standard resistance values for tolerance ranges from 1% to 20%.

It can be downloaded from Laurier Gendron's download page, **members.shaw.ca/roma/download.html**.

The downloaded file, **rescas.rar**, is compressed using an old pre-Microsoft era compression scheme, which was supplanted by the .zip standard. To uncompress the file, I suggest *7-zip*, a free program that can be downloaded from **www.7-zip.org**. That program will expand .rar files as well as many other compressed file formats including .zip files.

— 73, Juan A. Mónico, VA7IE, 15020 Ripple Rock Rd, Campbell River, BC, V9H 1N9, Canada; **juan@monico.org**

Thanks Juan,

Resistor CAD looks like a very useful program. Thank you for calling it to our attention, and also for the information about *7-zip* to uncompress the downloaded file. To assist readers who may not be able to, or may not want to install *7-zip*, I have also copied the *Resistor CAD* program file into a "standard" .zip file and placed it on the ARRL *QEX* files Web site. Interested readers can go to **www.arrl.org/qexfiles** and look for the file **9x09_rescad.zip**.

— 73, Larry, WR1B

Next Issue in *QEX*

Horst Steder, DJ6EV, and Jack Hardcastle, G3JIR, present "Crystal Ladder Filters for All." This article describes a computer program for the design of crystal ladder filters with predictable performance characteristics. The authors take us through the process of measuring the necessary physical parameters of individual crystal samples, so we can use that data in the design program. Then, by selecting crystals that most closely match the design parameters, we can construct filters that meet our design goals. The program will be available for free download from the ARRL *QEX* Web site.

Upcoming Conferences

The 28th Annual ARRL and TAPR Digital Communications Conference

September 25-27, 2009
Chicago, Illinois

There is still time to make plans to attend the premier technical conference of the year — the 28th Annual ARRL and TAPR Digital Communications Conference — to be held September 25-27, 2009, in Chicago, Illinois. The conference location is the Holiday Inn Elk Grove Village Hotel, Elk Grove Village, IL. This is the same location as last year's DCC.

The ARRL and TAPR Digital Communications Conference is an international forum for radio amateurs to meet, publish their work, and present new ideas and techniques. Presenters and attendees will have the opportunity to exchange ideas and learn about recent hardware and software advances, theories, experimental results, and practical applications.

Topics include, but are not limited to: Software defined radio (SDR), digital voice (D-Star, P25, WinDRM, FDMDV, G4GUO), digital satellite communications, Global Position System (GPS), precision timing, Automatic Position Reporting System (APRS), short messaging (a mode of APRS), Digital Signal Processing (DSP), HF digital modes, Internet interoperability with Amateur Radio networks, spread spectrum, IEEE 802.11 and other Part 15 license-exempt systems adaptable for Amateur Radio, using TCP/IP networking over amateur radio, mesh and peer to peer wireless networking, emergency and Homeland Defense backup digital communications, using Linux in Amateur Radio, updates on AX.25 and other wireless networking protocols.

This is a three-Day Conference (Friday, Saturday, Sunday). Technical and introductory sessions will be presented all day Friday and Saturday.

Join others at the conference for a Friday evening social get together. A Saturday evening banquet features an invited speaker and concludes with award presentations and prize drawings.

The ever-popular Sunday Seminar focuses on a topic and provides an in-depth four-hour presentation by an expert in the field. Check the TAPR Web site for more information and to register: **www.tapr.org**.

Registration

On-line registration is available at the TAPR Web site. The cost is $45 for one day, Friday or Saturday and $80 for both days. The Sunday seminar is $25. Register in advance for Friday and Saturday lunches, $20 each, and the Saturday evening banquet, $40.

AMSAT-NA Celebrates 40th Anniversary

October 9-11, 2009
Four Points Sheraton Hotel
Baltimore-Washington Airport

AMSAT-NA celebrates its 40th anniversary with the 2009 AMSAT Space Symposium October 9 through 11 at the Four Points Sheraton hotel near Baltimore-Washington Airport. More information and registration is available online at **www.amsat.org/amsat-new/symposium/2009/**, or by calling AMSAT-NA at 888-322-6728.

There is a free shuttle bus from the airport to the hotel. You can use public transportation to visit Washington and the inner harbor area in Baltimore. Annapolis will require a car. Within 5 minutes of the hotel is the National Electronics Museum.

Registration

Download the registration form at the AMSAT Web site or register on-line at the AMSAT store. Registration is $40 prior to Sep 15, $45 after Sep 15 and $50 at the door. The Saturday evening banquet is $45.

MICROWAVE UPDATE

October 23-25, 2009
Dallas, TX

Microwave Update 2009 will be held on Friday, October 23 through Saturday, October 24 in Dallas, Texas.

This is a microwave conference, and presentations will be on topics for frequencies above 900 MHz. Examples of such topics include microwave theory, construction, communication, deployment, propagation, antennas, activity, transmitters, receivers, components, amplifiers, communication modes, LASER, software design tools, and practical experiences.

Please contact Kent Britain, WA5VJB at **wa5vjb@flash.net** for additional information.

If you are interested in presenting at Microwave Update please contact Al Ward, W5LUA, at **w5lua@sbcglobal.net**.

Ray Mack, W5IFS

17060 Conway Springs Ct, Austin, TX 78717; w5ifs@arrl.net

Out of the Box

An Overview of Oscilloscopes

It wasn't long ago that a simple analog oscilloscope with modest capabilities would cost several thousand dollars. A storage scope was completely out of the picture for hobby use. The rapid drop in price of high speed analog and digital electronics over the past decade has brought modest storage scopes within a hobby budget (about the price of an ICOM IC-7000 or Yaesu FT-857D).

Stand Alone Oscilloscopes

All of the stand alone scopes mentioned come with probes, which are usually about $100 each.

Tektronix has a line of what we refer to at work as "lunch box" scopes because they are about the size of a large Superman or Batman lunch box. The TDS1001 is bottom of the line at $900. This is a monochrome 40 MHz, 2 channel scope with 500 million samples per second (MS/s) capture. For $1320 (about the most that I would still consider hobby priced) you can get a 2 channel monochrome TDS1012 100 MHz scope or a color TDS2002 60 MHz scope with 1 GS/s capture. All of these scopes come with digital storage up to 2500 samples per capture per channel. They also have two USB ports for I/O. One port is a host port for connecting Flash drives for storage of waveforms and setup information. The second USB port allows you to control the scope from a computer, and capture waveforms. These scopes also give you fast Fourier transform (FFT) calculations as part of the basic package, which means they can also display signals in the frequency domain, like a spectrum analyzer. There is a limited lifetime warranty with these scopes.

Agilent offers the 60 MHz 2 channel DSO3062 for $1198. This scope has a color display with 4000 samples per waveform and 1 GS/s sample rate. This scope includes software to do an FFT. This scope also includes both host and device USB ports. It comes with software to set the input filtering, so you can effectively look at the desired waveforms in the presence of noise. This model has a 3 year warranty.

Tektronix and Agilent are two of the most respected names in laboratory grade test equipment. B&K Precision has been known in the radio and TV service industry for many decades. I have 50 year old models of both B&K and Precision boat anchor test equipment in my collection. They have a much broader range of scopes that are well within the hobby budget of most amateurs. The least expensive monochrome digital storage scope is a 25 MHz, 250 MS/s model for $499. This includes many of the features of the more expensive models from Tektronix and Agilent. The 2542 is a 100 MHz, 2 channel, 1 GS/s, color scope for $1155. Like the others in the same price range, it has both host and device USB ports, digital filters, and FFT math. This model has a two year warranty.

LeCroy is another laboratory grade manufacturer. Their WaveAce series includes three models from $950 to $1200. The nicest is the WaveAce 202 for $1200. This includes 9000 samples per channel, 60 MHz bandwidth, 1 GS/s, the typical USB ports as well as an RS-232 port. All of the models have a color LCD. For the same price you can get the WaveAce 112 with 100 MHz bandwidth and 500 MS/s.

Jameco carries a Taiwan design Instek GDS-820 oscilloscope with 150 MHz bandwidth and 100 MS/s for $1400.

Computer Based Scopes

Unless otherwise noted, these products do not come with scope probes.

National Instruments is also known for laboratory grade test equipment. They have two USB based data acquisition modules that implement a digital oscilloscope in combination with a computer. The USB-5133 is a 2 channel, 100 MS/s, 50 MHz bandwidth module for $1199. The USB-5132 is a 2 channel, 50 MS/s, 50 MHz product. The main advantage of these over the standalone scopes is the depth of the memory. These have 4 MB of storage per channel.

BitScope makes a line of USB connected sampling systems. The unique feature of these is that they include both digitizers and logic analyzer inputs. The BS100U is a 40 MS/s, 100 MHz bandwidth, 2 channel oscilloscope with 8 logic inputs. This unit also has a deep acquisition depth of 64000 points. You can even look at the schematics for these products online to see how they do their job. All five of their products are 100 MHz bandwidth, 40 MS/s systems.

The Velleman PCS500 is available from Jameco for $350. This model requires a 25 pin parallel printer port on the computer for connection. It has a 25 MHz bandwidth.

Allied Electronics carries the Pico Technology line of USB based scopes. The PicoScope 3204 is a 50 MS/s, 50 MHz scope for $835. The PicoScope 3205 is a 100 MHz, 100 MS/s model for $1245.

Tektronix Inc.; **www.tek.com**

Agilent Technologies; **www.agilent.com**

LeCroy Corporation; **www.lecroy.com**

National Instruments; **www.ni.com**

BitScope; **www.bitscope.com**

Jameco; **www.jameco.com**

Allied Electronics; **www.alliedelec.com**

New RF MOSFET Source

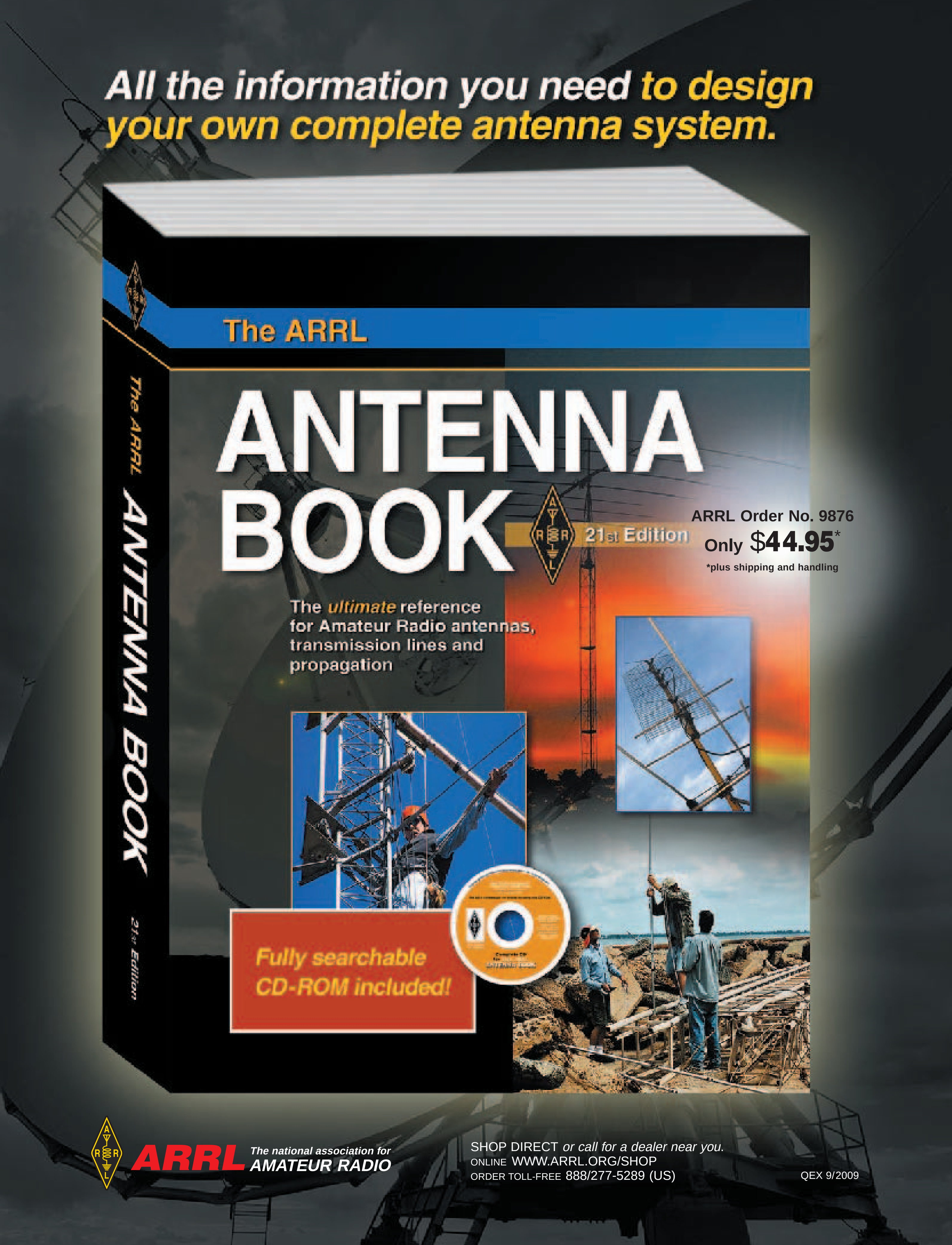
RF Parts carries a wide variety of RF devices. They recently added two new lines of RF power MOSFETs from Mitsubishi Electric and Toshiba. They carry 14 parts from Mitsubishi that cover dc to 900 MHz and 300 mW to 100 W power levels. Three of the parts are interesting because they are available in the TO-220 RF package, with the heat sink tab attached to the FET source lead. One of these parts is rated all the way up to 520 MHz. The parameter that I found most interesting is price. I am working on a 220 MHz amplifier, and the RD15HVF1, TO-220 part gives 18 W typical output for a cost of $5.25 each. The TO-220 RF package greatly simplifies the mechanical aspects of building an amplifier, and you don't have to worry about beryllia (beryllium oxide). Another interesting part is the RD100HHF1, which is capable of 100 W for only $28.20 each. The price of the parts range from $0.90 to $32.90.

The Toshiba parts are primarily aimed at low power applications for cell phone and consumer applications. One part is $120, however, and is rated at 190 W for use in VHF television transmitters. This is a push-pull device rated to 230 MHz in typical operation. The high peak to average power characteristics of TV are similar to those of SSB use, so the applications are very similar.

RF Parts is the official US distributor for all Toshiba RF semiconductors, including discrete power transistors and power modules.

RF Parts Company; **www.rfparts.com**.

QEX



All the information you need to design your own complete antenna system.
The ARRL
ANTENNA BOOK
21st Edition
The ultimate reference for Amateur Radio antennas, transmission lines and propagation
Fully searchable CD-ROM included!
The ARRL ANTENNA BOOK 21st Edition
ARRL Order No. 9876
Only $44.95*
*plus shipping and handling
ARRL The national association for AMATEUR RADIO
SHOP DIRECT or call for a dealer near you.
ONLINE WWW.ARRL.ORG/SHOP
ORDER TOLL-FREE 888/277-5289 (US)
QEX 9/2009