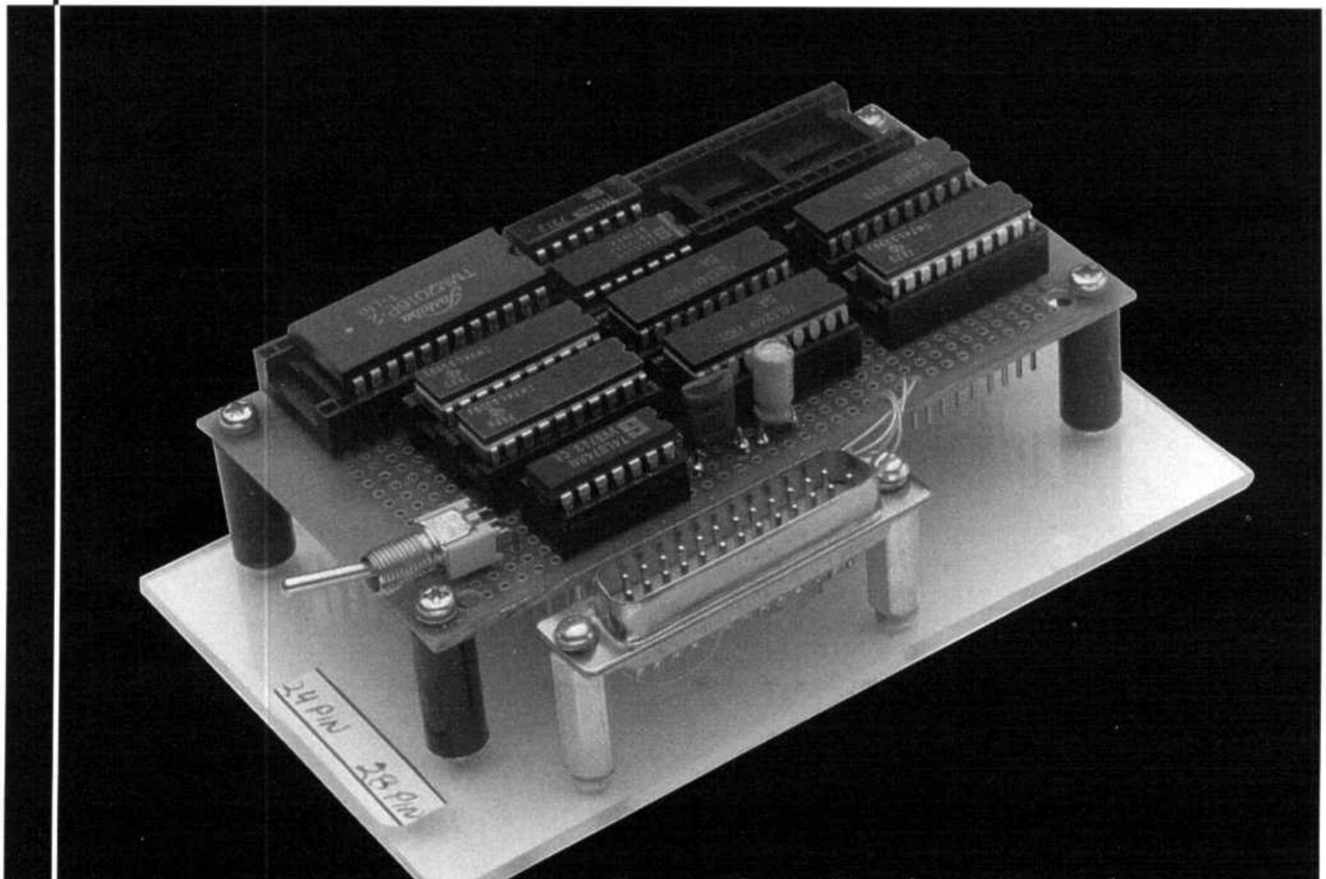
QEX

$1.75

ARRL Experimenter's Exchange

JANUARY 1993



Low-Cost Project Development Using A ROM Emulator

QEX: The ARRL
Experimenter's Exchange
American Radio Relay League
225 Main Street
Newington, CT USA 06111

Non-Profit Org.
US Postage
PAID
Hartford, CT
Permit No. 2929

QEX (ISSN: 0886-8093) is published monthly by the American Radio Relay League, Newington, CT USA.

David Sumner, K1ZZ
Publisher

Jon Bloom, KE3Z
Editor

Lori Weinberg
Assistant Editor

Mark Forbes, KC9C
Geoffrey H. Krauss, WA2GFP
Harold Price, NK6K
Zack Lau, KH6CP
Contributing Editors

Production Department

Mark J. Wilson, AA2Z
Publications Manager

Michelle Bloom, WB1ENT
Production Supervisor

Sue Fagan
Graphic Design Supervisor

Dianna Roy
Senior Production Assistant

Circulation Department

Debra Jahnke, *Manager*

Kathy Fay, N1GZO, *Deputy Manager*

Cathy Stepina, *QEX Circulation*

Offices

225 Main St, Newington, CT 06111 USA
Telephone: 203-666-1541
Telex: 650215-5052 MCI
FAX: 203-665-7531 (24 hour direct line)
Electronic Mail: MCI MAIL ID: 215-5052
Internet:arrl@mcimail.com

Subscription rate for 12 issues:
In the US by Third Class Mail: ARRL Member $12, nonmember $24;

US, Canada and Mexico by First Class Mail:
ARRL Member $25, nonmember $37;

Elsewhere by Airmail:
ARRL Member $48 nonmember $60.

QEX subscription orders, changes of address, and reports of missing or damaged copies may be marked: *QEX* Circulation.

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

About the Cover:

Developing microprocessor-based projects can be a chore unless you have high-priced professional tools—or low-cost amateur replacements, such as this ROM emulator.

131

TABLE OF CONTENTS

COLUMNS

JANUARY 1993 QEX ADVERTISING INDEX

THE AMERICAN RADIO RELAY LEAGUE

The American Radio Relay League, Inc. is a noncommercial association of radio amateurs, organized for the promotion of interests in Amateur Radio communication and experimentation, for the establishment of networks to provide communications in the event of disasters or other emergencies, for the advancement of 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 two 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 at 225 Main Street, Newington, CT 06111 USA.

Telephone: 203-666-1541 Telex: 650215-5052 MCI.
MCI MAIL (electronic mail system) ID: 215-5052
FAX: 203-665-7531 (24-hour direct line)

Canadian membership inquiries and correspondence should be directed to CRRL Headquarters, Box 7009, Station E, London, ON N5Y 4J9, tel 519-660-1200.

Officers

President: GEORGE S. WILSON III, W4OYI
1649 Griffith Ave, Owensboro, KY 42301

Executive Vice President: DAVID SUMNER, K1ZZ

Purpose of *QEX*:

1) provide a medium for the exchange of ideas and information between Amateur Radio experimenters
2) document advanced technical work in the Amateur Radio field
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 correspondence for publication in *QEX* should be marked: Editor, *QEX*.

Both theoretical and practical technical articles are welcomed. Manuscripts should be typed and doubled spaced. Please use the standard ARRL abbreviations found in recent editions of *The ARRL Handbook*. Photos should be glossy, black and white positive prints of good definition and contrast, and should be the same size or larger than the size that is to appear in *QEX*.

Empirically Speaking...

Low-Cost Development Tools

Most of the technology we experiment with is relatively low in cost. The revolution of the past three decades in semiconductor integration has provided us with incredibly complex chips that cost a mere pittance. We take advantage of these devices by developing ever-more-complex applications at ever-decreasing cost. This is a trend that's bound to continue.

Another, related, trend is toward device programmability. The complexity that can be packed into a small, low-power device these days is such that use of a programmable part often makes more economic sense than use of a more direct circuit using nonprogrammable parts. This is exhibited at various levels of functionality. For example, simple digital logic circuits are often implemented with one-time-programmable (OTP) logic devices; programmable analog circuits such as adjustable-gain amplifiers and programmable filters are common, too. At a higher functional level, microprocessors and audio subsystems are examples of programmable digital and analog systems.

All of which is wonderful except for one thing: translating the paper design to the physical reality often requires assistance from design and development tools. And often, these tools are hard to find or are prohibitively expensive. There are outstanding software and hardware tools available to take the labor out of programming parts and systems, but most of the best tools were developed for use in engineering environments—where paying big bucks for development tools is just a cost of doing business.

That's not to say there are no low-cost development tools. Amateurs can develop—and have developed—some of their own tools, as demonstrated by the ROM simulator in this issue of *QEX*. But you have to gather a number of tools to put together a toolbox you can use for serious experimentation and development.

So, this is a call for information sharing. Let me know, by writing to me at the address on the masthead of this magazine or by sending me electronic mail at the Internet address listed at the end of this column, what low-cost development tools you use and like. They can be programming aids or other design tools. We'll compile a summary of the information we receive and publish it in *QEX*. Maybe then we can all find the tools we need to push our development activities to new levels, with new technology.

This Month in QEX

Debugging software that runs on microcontrollers and microprocessors is made easier when you can easily reload the computer's program memory. With the "2k × 8 Bit ROM Emulator," by Dave Lichtenstein, AA6FJ, and John Nemec, NS6Z, this becomes a simple matter of telling your PC to reload the memory with new data.

In "Reflections on the Reflection Coefficient: An Intuitive Examination," Wes Hayward, W7ZOI, takes a non-mathematical look at the subject to clarify it and ends up showing why the Smith chart looks the way it does.

Does your helical antenna have that drooping look? Maybe its design just isn't up to long-term exposure to the elements. Ron Lile, KØRL, shows us an almost-all-metal design that will last years in, "Improved Mechanical Design for the Helical Antenna."

Mark Forbes' "Components" column this month describes a professional RF design package he's been trying out. In "VHF+ Technology," Geoff Krauss explores the results of the June 1992 ARRL VHF contest and other topics discussed at this year's Microwave Update conference, and expounds on the radial line stub.

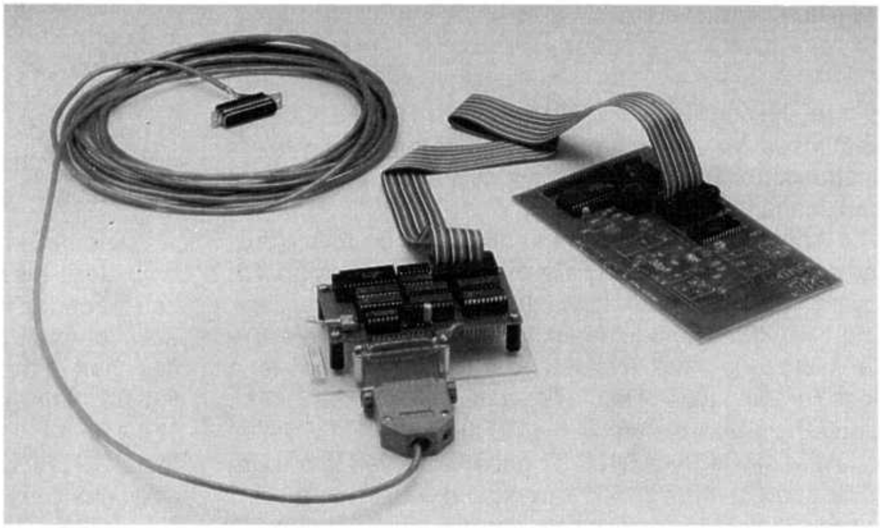
—KE3Z, email: jbloom@arrl.org (Internet)

A 2k × 8 bit ROM Emulator

By Dave Lichtenstein, AA6FJ
6543 Hampton Drive
San Jose, CA 95120

and

John Nemec, NS6Z
2707 Estella Drive
Santa Clara, CA 95051



Introduction

Many amateur projects can utilize microcontrollers. A major advantage of this approach is that the hardware construction can be much simpler than a discrete logic implementation. The possibilities range from simple circuits like an electronic keyer or a frequency counter and display, to more complex applications such as station control (local and remote), repeater control, DSP, packet TNCs and AMTOR. In all cases, however, a program must be developed, integrated with the hardware, and then debugged. With a few simple tricks and techniques, this process is quite straightforward. It is far easier to type a few additional lines of code than to get out the wire-wrap tool or soldering iron and move wires, add ICs and reprogram an EPROM. This article describes an easy-to-build tool to make the development process painless.

Microcontrollers generally have two ways of accessing program instructions. One way stores the instructions internally using an EPROM or a metal mask coded ROM. The second way stores the instructions externally, usually in an EPROM. This article focuses specifically on a microcontroller using external program storage. From a socket standpoint this circuit will emulate any 24 or 28 pin EPROM, i.e., any 2716 through 27256 device, but the size is limited to 2 kbytes.

From a practical standpoint, the selection of internal or external program storage is often dictated by the application. While attaching an external EPROM for program store is very direct, the use of the I/O ports may conflict with the use of external EPROM. If the application uses many of the I/O port bit manipulation capabilities, then giving these up for external EPROM may not be desirable. But if the number of I/O ports is not critical for the application, the external program can be used as is. The worst case occurs when all of the available I/O ports are required for the application. In this situation, the emulator can still be used to develop and debug a significant portion of the code, but the final code debugging will be done by actually loading the instructions into the onboard microcontroller ROM and completing the remainder of the work using trial and error.

Since this project is generic and can be used to develop and debug any microcontroller application, some basic assumptions have been made. First, an application (called the "target system") has been identified. Second, the application hardware already exists. Third, an assembler or C compiler that produces an object module (the actual binary code interpreted by the microcontroller) in Intel HEX format is available. The task at hand is to integrate the software and hardware for the target system and get everything working.

The debug process is iterative. Individual sections of code are run and their results checked. In this way, various modules of code are debugged and then put together in the main program. One "cheap and dirty" trick frequently used during this process is to add code specifically for trouble-shooting purposes which is later removed when the debug is completed. By its very nature, this process requires a method which allows the code to be readily changed, re-loaded into ROM and re-run.

In the commercial world, microcontroller manufacturers supply in-circuit emulator (ICE) systems with development software to address this specific issue. This is usually done with a special integrated circuit (IC) that uses pins not normally available in the production ver-

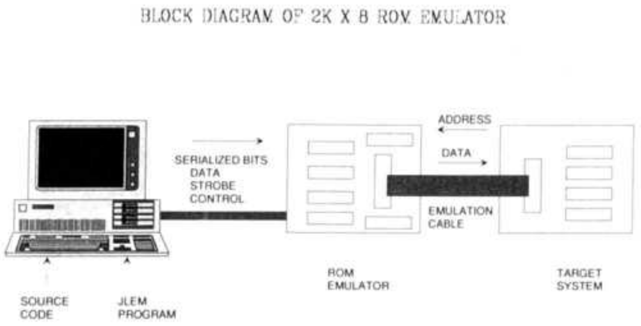


Fig 1—Block diagram of how everything is hooked together

sion of the chip. In addition to the custom IC, the ICE system uses an emulation RAM for storing the system instructions. The primary functions that all ICE systems provide are shown in Table 1.

At this point the problem appears to be trivial. Just go down to your favorite parts distributor and buy an ICE system. The bad news is that the price for a typical commercial ICE system is about $1500 for one of the popular microcontrollers. A midrange ICE system or one for a less common type of microcontroller costs about $3000, and a high-end system with all of the bells and whistles could easily carry a $10,000 price tag. Even if you are willing to part with $1500 or can find a deal on a used system, there is another catch: These systems are device specific and require additional attachments or perhaps a dedicated ICE system for each microcontroller device type or variant.

A good approach for the experimenter operating on a limited budget is to use a ROM emulator. The ROM emulator is a static RAM that is loadable through one port

TABLE 1

ICE Systems and ROM Emulator Features

Feature	ICE System	ROM Emulator
Can examine program memory contents by specific address.	Yes	Yes
Can dis-assemble binary code into assembly language.	Yes	No
Can edit single byte.	Yes	Yes
Can edit a group of bytes.	Yes	Yes
Allows program execution to start at specific location.	Yes	No
Allows program execution to stop at specific location (breakpoint).	Yes	No

and, at the end of a cable, can look to the microcontroller exactly like any popular EPROM. Fig 1 shows a block diagram of the ROM emulator. As Table 1 shows, this emulator does not have all of the features of a full-blown ICE system. However, it has more than enough capability for almost any application.

The schematic diagram for the ROM emulator is shown in Fig 2. There are four basic sections to the circuit:

1. *Instruction Storage:* Provided by the 2K X 8 SRAM.
2. *Emulation Mode Section:* Isolation circuitry (74LS244 tri-state buffers) used to separate the address and data lines from the microcontroller in the target system.
3. *Learn Mode Section:* A 25-bit shift register formed by the cascaded 74LS299s to receive the address, data bit and control bit via the data cable from the printer port of the personal computer (PC).
4. *Logic Circuitry:* Various gates to implement the logic functions.

The JLEM program running on a PC controls the emulator. This software takes the microcontroller instruction file coded in Intel HEX format, serializes the data and downloads it into the emulator RAM. A high-level flowchart of the program is shown in Fig 3. Bits from the PC are loaded serially into the shift register with the CONTROL line held high. When the register is fully loaded, the CONTROL line is asserted low. This prevents any additional bits from shifting through the register and enables the RAM write enable ($\overline{WE}$) signal line. On the next strobe, data is written into the RAM at the specified address. The initial load consists of setting the Learn/Emulate Flag (the last bit of the shift register) to "0". This disables the EMULATE section and enables the LEARN section. Next, each address and data combination is shifted into the register and the data is written into the RAM. This process is repeated until all instructions have been stored into the RAM. After the last instruction is written, data is shifted into the register which sets the Learn/Emulate Flag to "1". The D flip-flop, U8, then switches the circuit into the EMULATE mode. In the EMULATE mode, the LEARN mode circuitry is tri-stated and the address drivers that receive the instruction address from the emulation socket are enabled. At this point, the microcontroller of the target system can be manually reset and the program will begin to execute.

Note that the emulator is powered from the target system via the +5 volt (V_{CC}) and ground pins of the target system's socket. The target system must be powered in order for the emulator to successfully receive the data in the LEARN mode. Since the microcontroller address does not reach the emulation RAM during the LEARN sequence, the microcontroller is receiving random instructions at this time. Depending upon the application, you may want to hold the microcontroller in the reset condition until the EMULATE mode is entered. In any case, the microcontroller must be reset after the EMULATE mode is entered to insure that the target system begins executing instructions from the correct address.

Building the hardware

Circuit construction is not critical and can be done on a vector board using wire-wrap techniques. Although the original ROM emulator did not use them, bypass capacitors should be included. Standard practice calls for a 0.1 µF capacitor on each socket and a 10 µF capacitor across power supply input connections. It is probably best to keep the emulation buffers as close to the RAM and emulation socket as possible since the circuit will be running at the microcontroller's clock rate. The SRAM socket can be a 28-pin socket wired to permit an upgrade to 8 kbytes of instruction storage. To do this, wire pins 3 through 26 of the 28-pin socket as if they were pins 1 through 24 of a 24-pin socket. A few wiring changes will effect the upgrade.

The emulation socket should also be 28-pin and wired according to the schematic. Emulation cables can be purchased or you can make them yourself using ribbon cable and headers. Two cables should be built (24-pin and 28-pin) and used for their respective applications. The 24-pin cable is used to emulate older 24-pin EPROMS such as the 2716 and 2732, while the 28-pin cable is used for 27C64 and 27C256 EPROMs. Using the 28-pin cable for the 24-bit applications is awkward and will result in bent

Bill Of Materials For 2k ROM Emulator

Identifier	*Description*
U1, U2, U3, U4	Each is ⅙ of 74LS14 (hex inverting Schmitt trigger)
U5, U6, U7	74LS299 (8 bit shift register with tri-state output)
U8	½ of 74LS74 (dual D flip-flop)
U9, U10, U11	Each is ¼ of 74LS32 (quad or gate)
U12	2016 (2k × 8 SRAM)
U13, U14, U15	74LS244 (tri-state buffer)

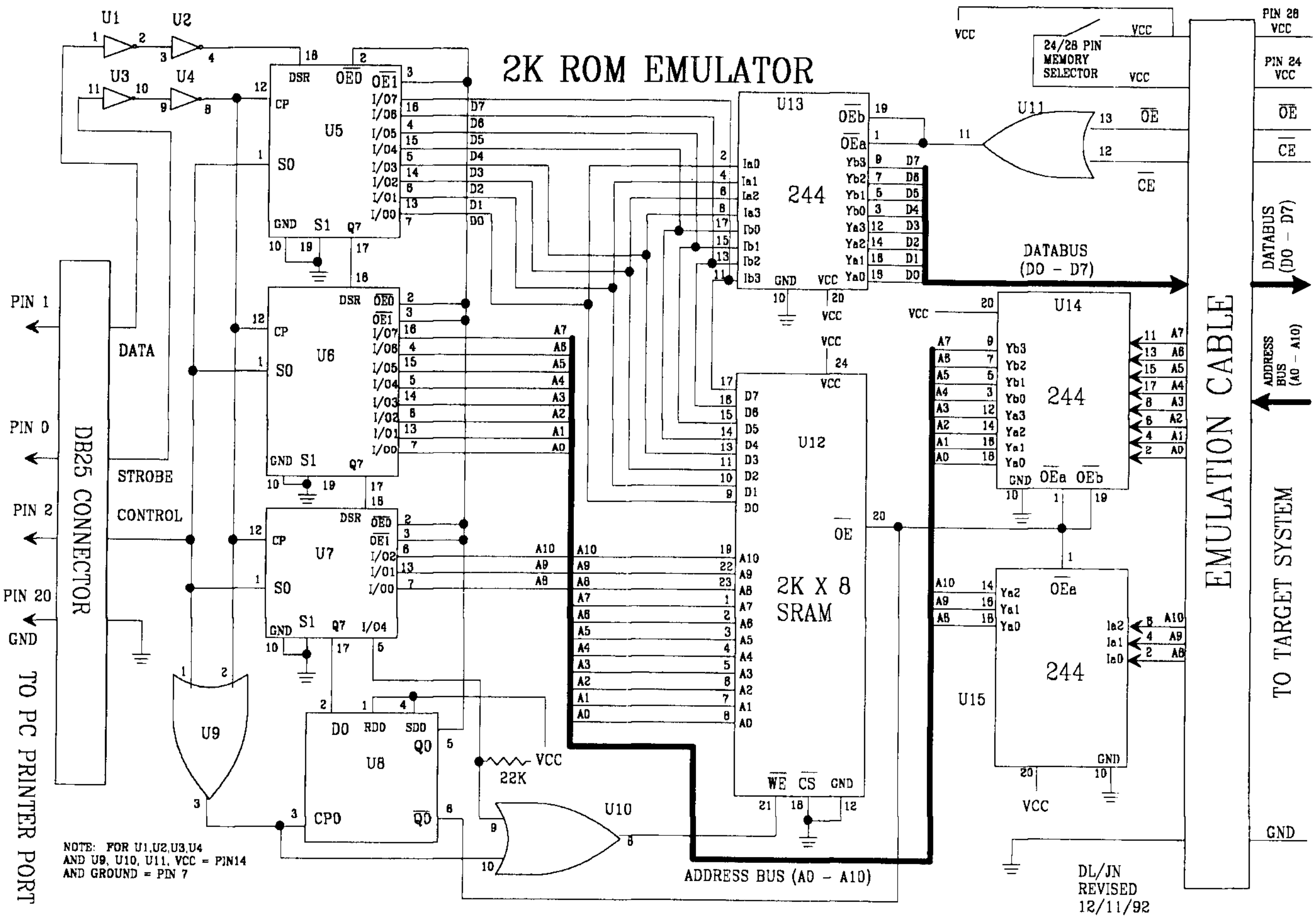


Fig 2—Schematic diagram of ROM emulator

JLEM PROGRAM FLOWCHART

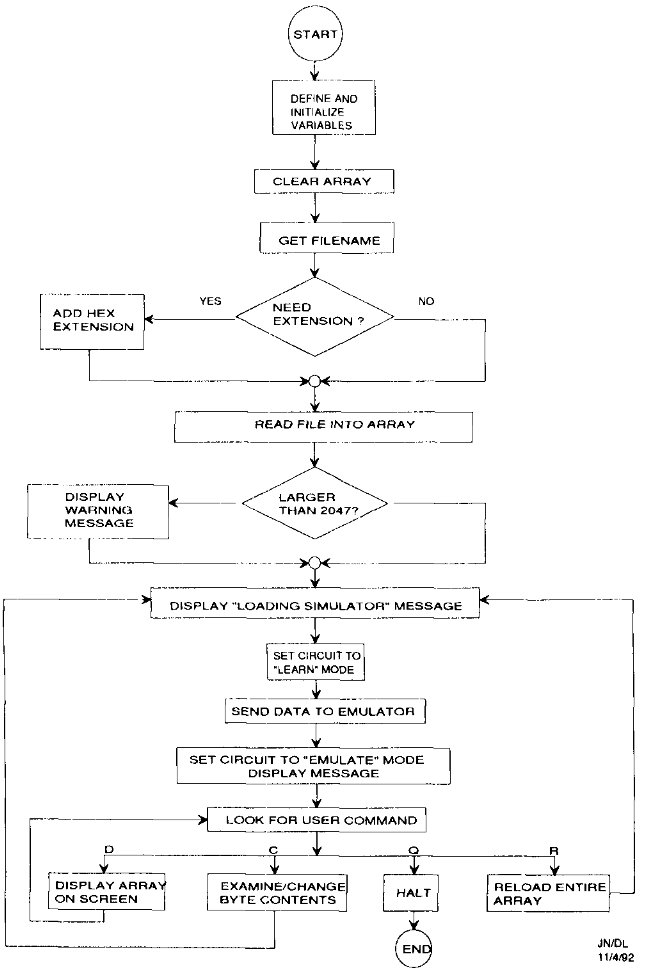


Fig 3—JLEM overview flowchart (see note 1)

pins or a poor fit in tight places. Provide adequate labeling for the emulation socket so that the cables will be plugged in correctly. The cable should be keyed so that the 24-pin emulation cable plugs into the pins opposite the pin 1/28 end of the socket. Mark the 24/28 pin selection switch clearly to avoid confusion. The switch should be guarded to prevent accidental change when the module is in use. Having the switch set for 24 pins when emulating 28-pin EPROMs will cause an address line of the target application to be connected to V_{CC}.

Mount the entire unit in a box with an insulated base so that the emulator can be set on top of the target system circuit board without danger of shorting any connections. This will let you keep the emulation cable short while still allowing access to the target system while debugging the code.

The only critical components are at the interface to the PC printer port. It is *mandatory* that 74LS14 or 74HCT14 Schmitt triggers are used. These devices provide hysteresis on the inputs and prevent multiple clocking of the shift registers by slow-rise-time signals. The remaining components are not critical since all LS or HCT logic devices are fast compared to microcontroller speeds. If a high-speed microcontroller is used in the application, an SRAM with an appropriate access time should be used. The use of HCT devices in the LEARN side of the module is recommended because they consume very little power. This minimizes the current drawn from the target system.

The PC connector can be easily constructed from a DB25 connector. Only four connections are required: D0, D1, strobe and ground.

Circuit check-out

Carefully review the wiring to make sure there are no obvious errors. To gain confidence that the module is correct, build a "target socket" that has power, pull-up resistors (4.7kΩ is a typical value), switches to ground on each of the address lines, and the $\overline{OE}$ and $\overline{CE}$ pins grounded. Load the module with code as described under "Using the emulator." Turn all of the address switches on to ground all of the address inputs (input address 00) and, using a VOM, examine each of the data output pins D0 through D7. Compare the output with the known content at the address. After verification, walk a 1 through the address bits (up to 1023 decimal) and verify the data output at each address. If all of the bits match, the emulator is probably wired correctly. The code contained in each of these locations should be different so that there are unique contents at each address. The contents should also be varied so that each output bit is exercised, verifying that there are no bits that are miswired. If everything checks out, a very simple program can be assembled and run in an application board. The program should produce some simple, easily observable output to verify functionality. The next step is to extend this to jumps throughout the 2k address space to again verify that all of the address connections are correct.

Should any of these tests fail, you will have to signal trace the circuit. Start in the LEARN mode and modify the section of the program containing the "tdly" instruction. In this part of the code add:

```
until keypressed;
```

statements and recompile the JLEM program. This statement causes the program to wait for any key to be pressed before continuing. This will let you single-step through the serial load operation. With a VOM, examine the DATA, STROBE and CONTROL signals at each stopping point and verify that they are at the correct logic levels. Next, follow the data through the shift registers and verify the contents as the data moves through the registers. Follow the DATA, CONTROL signal, address and $\overline{WE}$ lines all of the way to the SRAM to verify that the connections are correct.

If this part of the circuit is working, make sure that the control signals from the output of the Emulate/Learn flip-flop are correct and appear at all of the connections. Do this for both the LEARN mode and the EMULATE mode.

After all of the corrections have been made and you are confident that the LEARN circuitry is operating correctly, remove the "until" statements and load the emulator with unique data as previously described. Start at the target socket and track the address and data signals at the SRAM socket. With the final debug complete, return to running short diagnostic programs in an application board for the final testing.

Using the emulator

The subject of program debugging is broad and complex. Here are a few tips focusing on the emulator operation itself that will streamline the process.

Verify that the 24/28 pin selector switch has been correctly set and that the emulation cable is correctly plugged into the target system. Connect the PC printer port cable to the emulator and apply power to the application. On the PC keyboard, type JLEM followed by Enter and wait for the prompt.[1] At the prompt, enter the name of the Intel HEX file that contains the object code for the application. (This code is produced by the microcontroller assembler and is a common standard to transfer data to EPROM programmers.) After the file has been read, the console will display "LOADING SIMULATOR MODULE." This indicates that the module is in the LEARN mode and is not free to be accessed by the application. Loading may take several minutes so be patient. When loading is complete, the monitor will display "SIMULATION MODE." At this point the module has entered the EMULATION mode and the microcontroller should be reset to start execution at the controller's reset starting address. Since the module is powered from the application, power cannot be turned off without losing the LEARNED data. Reset the microcontroller by pulling the external RESET pin to the appropriate level and then releasing. The microcontroller will then begin executing the code contained in the emulator.

There are several program control commands which can now be utilized. Each command is invoked by entering one of the command letters summarized in Table 2. Look at the annotated example session to see what the display looks like for each command.

Use of the Display command is shown in the example session. The left column is the hex address followed by the data at that address. Successive address contents (also in hex) are displayed down the row. Note that this display is actually the image of the emulator contents as stored in the PC.

At the prompt for the Change command (C:), enter the hex address of the byte to be changed followed by Enter. The actual contents will then be displayed. Type in the new value followed by Enter. If you don't want to change the contents, simply press Enter to exit the change routine. If you want to leave the contents unal-

EXAMPLE SESSION FOR ROM EMULATOR

JLEM asks for hex file
User input gives file name
Message telling user that emulator is in LEARN mode
Display command (no carriage return)
Message telling user that emulator is in EMULATE mode
JLEM echoes display command
User enters starting location
Microcode (by location) showing contents in hex

```
Enter Intel hex code file name: a:cdtime [CR]
Loading Simulator Module.
Simulation mode.
D
D: 0000 [CR]
0000: 75  0001: 81  0002: 0F  0003: 75  0004: 21  0005: 00  0006: 75  0007: 24
0008: 00  0009: 75  000A: 89  000B: 20  000C: 75  000D: 8D  000E: 72  000F: 75
0010: 98  0011: D0  0012: D2  0013: 8E  0014: D2  0015: AC  0016: D2  0017: AF
0018: 02  0019: 04  001A: 00  001B: 00  001C: 00  001D: 00  001E: 00  001F: 00
0020: 00  0021: 00  0022: 00  0023: C0  0024: E0  0025: D2  0026: D3  0027: C2
0028: 98  0029: 20  002A: 9A  002B: 12  002C: 20  002D: 7F  002E: 0A  002F: A6
0030: 99  0031: 08  0032: E8  0033: 54  0034: 03  0035: 70  0036: 02  0037: D2
0038: 7F  0039: C2  003A: D3  003B: D0  003C: E0  003D: 32  003E: C2  003F: 7F
0040: E5  0041: 99  0042: C3  0043: 33  0044: C3  0045: 33  0046: 24  0047: 40
0048: 40  0049: 05  004A: B4  004B: 68  004C: 06  004D: 40  004E: 04  004F: D2
0050: 7F  0051: 80  0052: E6  0053: F8  0054: C2  0055: D3  0056: D0  0057: E0
0058: 32  0059: 00  005A: 00  005B: 00  005C: 00  005D: 00  005E: 00  005F: 00
0060: 00  0061: 00  0062: 00  0063: 00  0064: 00  0065: 00  0066: 00  0067: 00
0068: 00  0069: 00  006A: 00  006B: 00  006C: 00  006D: 00  006E: 00  006F: 00
0070: 00  0071: 00  0072: 00  0073: 00  0074: 00  0075: 00  0076: 00  0077: 00
0078: 00  0079: 00  007A: 00  007B: 00  007C: 00  007D: 00  007E: 00  007F: 00
```

Annotated sample session (continues on following page)

EXAMPLE SESSION FOR ROM EMULATOR (cont'd)

```
                    Change command
JLEM echoes         (no carriage return)
Change command
      C                          User enters
                                 starting location
      C: 0008 [CR]
      C: 0008: 00  11 [CR]  <--- Change contents of location 0008 from 00 to 11
      0009: 75  [spacebar]  <--- Use spacebar to step through next two locations
      000A: 89  [spacebar]  <---
      000B: 20  00 [CR]     <--- Change contents of location 000B to from 20 to 00
      000C: 75  [CR]        <--- Carriage return to exit Change mode

                    Display command
JLEM echoes         (no carriage return)
display command
      D                          User enters
                                 starting location
      D: 0000 [CR]
      0000: 75  0001: 81  0002: 0F  0003: 75  0004: 21  0005: 00  0006: 75  0007: 24
      0008: 11  0009: 75  000A: 89  000B: 00  000C: 75  000D: 8D  000E: 72  000F: 75
      0010: 98  0011: D0  0012: D2  0013: 8E  0014: D2  0015: AC  0016: D2  0017: AF
      0018: 02  0019: 04  001A: 00  001B: 00  001C: 00  001D: 00  001E: 00  001F: 00
      0020: 00  0021: 00  0022: 00  0023: C0  0024: E0  0025: D2  0026: D3  0027: C2  <--- Microcode (by location) showing
      0028: 98  0029: 20  002A: 9A  002B: 12  002C: 20  002D: 7F  002E: 0A  002F: A6       contents in hex
      0030: 99  0031: 08  0032: E8  0033: 54  0034: 03  0035: 70  0036: 02  0037: D2
      0038: 7F  0039: C2  003A: D3  003B: D0  003C: E0  003D: 32  003E: C2  003F: 7F
      0040: E5  0041: 99  0042: C3  0043: 33  0044: C3  0045: 33  0046: 24  0047: 40
      0048: 40  0049: 05  004A: B4  004B: 68  004C: 06  004D: 40  004E: 04  004F: D2
      0050: 7F  0051: 80  0052: E6  0053: F8  0054: C2  0055: D3  0056: D0  0057: E0
      0058: 32  0059: 00  005A: 00  005B: 00  005C: 00  005D: 00  005E: 00  005F: 00
      0060: 00  0061: 00  0062: 00  0063: 00  0064: 00  0065: 00  0066: 00  0067: 00
      0068: 00  0069: 00  006A: 00  006B: 00  006C: 00  006D: 00  006E: 00  006F: 00
      0070: 00  0071: 00  0072: 00  0073: 00  0074: 00  0075: 00  0076: 00  0077: 00
      0078: 00  0079: 00  007A: 00  007B: 00  007C: 00  007D: 00  007E: 00  007F: 00

      R  <--- Reload command (no carriage return)

      Loading Simulator Module.
      Simulation mode.          <--- Messages from JLEM
                                                                           DL/JN
      Q  <--- Quit command (no carriage return)                            11/9/92
```

Annotated sample session (continued from previous page)

tered but want to step to the next address location, press the space bar.

When a content change is performed, the emulator will enter the LEARN mode, receive the new instruction and return to the EMULATE mode. Since conditions are the same as during the initial download, the microcontroller has once again been disconnected from the emulator address bus and is receiving random instructions. Consequently, it will most likely be executing in areas of the memory space that do not exist or do not contain executable code. Because of this, it will be necessary to once again reset the microcontroller after editing the code.

If re-loading the entire emulation module is required, use the "R" command. This command would be appropriate if the power to the application has been interrupted. As usual, reset the microcontroller after entering the EMULATE mode.

Use the Quit command to exit JLEM. The PC printer port connector may be removed at this time. The circuit will remain in the EMULATE mode as long as power is applied.

As you can see, the operation is relatively simple. Since it is not a true development system, however, there are a few things you can do to make things simpler.

When debugging the application, add a visual indicator that can be removed later. A typical example would be a single LED connected to an unused I/O port. The

TABLE 2

JLEM Program Commands

Letter	Command	Description
D	Display	Display the contents of the emulation memory to the screen.
C	Change	Change a byte in the emulator memory.
R	Reload	Reload all of the emulation code.
Q	Quit	Exit JLEM and return to DOS.

LED can be controlled by instructions included in the source code for debug purposes. The state of the LED can indicate that a certain portion of the program has been reached, or a calculated result compares successfully to a predetermined value and so on. Another option is to add a register connected to a 7-segment LED display. This would provide more definitive information by displaying results directly. Another possibility is to connect a switch to one of the I/O pins of the microcontroller so that its closure can be sensed. This can be used by special code added for debug purposes to stop at a predetermined point and advance when the external switch is closed.

The ROM emulator will not let you view the internal registers or, port contents or trace the path of the instruction execution. The best way to tackle this is to break the source program into discrete modules which perform a specific function and execute each one apart from most of the other code in the application. Debug each module separately and thoroughly so that there is a high confidence that each is running correctly. Then proceed to the next level by combining these modules into larger segments. The emulator allows real time edits to the code either by patching (using the Change command) or by altering the source code, reassembling and downloading the simulator with the new code. Overall, this approach is more tedious than use of professional level development systems, but it certainly is orders of magnitude better than trial and error and repeated EPROM program and erase cycles.

Notes:

[1]The JLEM program, including source code, may be obtained electronically from 1) CompuServe HAMNET, 2) the ARRL telephone BBS (203 666-0578) or 3) anonymous FTP from ftp.cs.buffalo.edu (/pub/ham-radio directory) on the Internet.

Reflections on the Reflection Coefficient: An Intuitive Examination

By Wes Hayward, W7ZOI
7700 SW Danielle Avenue
Beaverton, OR 97005

Much of modern electronics is steeped in mathematics, some of it very complicated and sometimes difficult. The math is powerful, a fact beyond dispute. The strength is in the generality afforded by a mathematical description; one equation can describe an entire family of complicated phenomena.

It is easy to become entrenched in the mathematics of a subject. The utility and even the very structure can be exciting. However, an overly analytic approach can cause some things to appear to be more complicated than they really are. This may be the case with the subject of this note where we ponder the nature of the voltage reflection coefficient.

Power Transfer and Impedance Matching

We begin our discussion with a look at a source of energy, shown in Fig 1. This could be an ac generator, such as an RF signal source, or something as simple as a battery. The source is modeled as a voltage source with a magnitude of 2. The generator has a series resistance of value R_0. A typical value for R_0 might be 50 ohms. Our generator is terminated with (attached to) a resistance of value R_X. The "x" indicates that the value might not be known. We assume initially that the impedances are simple resistances; reactance is re-served for a bit later.

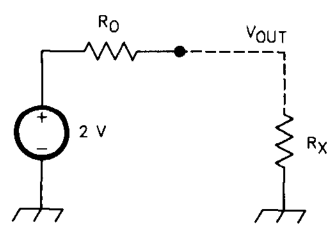


Fig 1

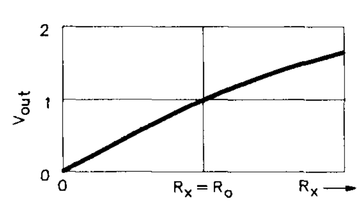


Fig 2

The output voltage is well established. Assume that R_0 is fixed and known, but R_X is allowed to vary. The output voltage is found with Ohm's Law. Some results are plotted in Fig 2. When the termination resistor, R_X, is a short circuit, the output is zero. A voltage appears when R_X is nonzero, and the output continues to climb as R_X gets larger. It does not climb forever; when R_X has become extremely large, V_{out} reaches 2 volts.

A corollary result relates to power. The power delivered to the load, R_X, is a function of the load resistance. The calculation yields the familiar result that the power is maximum when $R_X=R_0$. Either larger or smaller values of R_X will cause less power to be dissipated in R_X. The load is said to be matched to the source. The power appearing in the matched load is termed the available power from the generator.

Further Reflections

Fig 3 shows a modified version of the earlier signal source. A second voltage generator has been added. The second generator, with a 1-volt strength, floats below the earlier V_{out} node, with an output labeled with the Greek capital letter gamma (Γ). The polarity of the added generator is opposite that of the first generator. The voltage at gamma is 1 volt less then V_{out}.

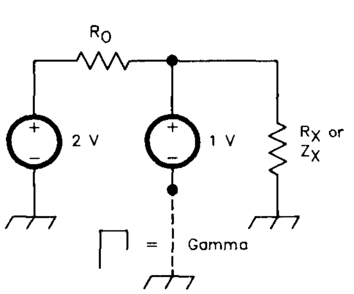


Fig 3

The value for gamma (a voltage) is interesting and useful. Assume that a very high impedance is attached to the R_X port of Fig 3. Little current flows, so the output voltage (V_{out} of Fig 1) is 2, the open circuit value. The corresponding value for gamma is 1 volt.

If the terminating impedance is set to R_0, the reference, or source resistance, the signal across the termination is 1 volt, causing gamma to become zero.

The third case of interest is a short-circuit termination. This causes gamma to be −1. If the experiment

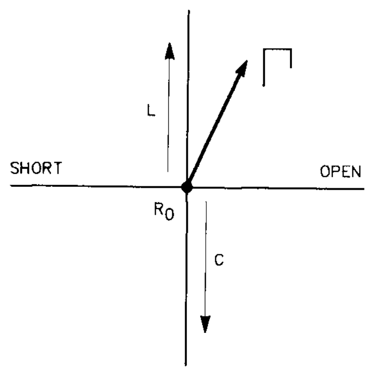


Fig 4

described was done with batteries, a negative voltage would happen for gamma. If the generators were ac sources, the short-circuit termination would yield a gamma with magnitude of 1. It would, however, have a phase that is 180 degrees away from the driving 2-volt generator.

Gamma has significance. Gamma was 0 volts when the perfect impedance match was present, $R_X = R_0$, but was highest when there was a severe impedance mismatch. Gamma is, hence, a measure of the degree of mismatch. Energy that is not transferred from a source to a termination is said to be reflected. Accordingly, gamma is called the voltage reflection coefficient.

Our examples have used pure resistance for the termination. The general case would allow complex terminations: impedances consisting of both resistance and reactance. The reactance would arise from either inductors or capacitors. Fig 4 is a plot showing a gamma value resulting from a complex termination.

Consider first the resistances we have already discussed. Gamma takes on values from –1 to 1 for terminating resistors, R_X, from a short to an open circuit. This variation is shown on the horizontal axis of Fig 4. This line, of length 2, is a "map." All possible resistances from 0 to very large values correspond to points on the horizontal line. The most exciting detail of this result is that a strictly infinite range of resistance has been mapped onto a line of finite extent.

Adding reactance to a resistance generates impedances that map to points that are no longer on the horizontal axis. An inductance with the resistance generates a reflection coefficient point somewhere above the horizontal line. Resistor-capacitor circuits generate points below the line. Recall that a complex impedance is represented by a complex number, one with a "real" and an "imaginary" part. It is reasonable that a complex impedance would generate a complex reflection coefficient. Fig 4 is just a polar plot of various complex reflection coefficients.

Just as resistance was allowed to vary over a very large range, reactance can vary from large capacitive (negative) values to zero to large inductive (positive) values. Rather, the behavior is more complicated. What is found is that any possible impedance, Z=R+jX, will have a corresponding gamma value that has a magnitude of 1 or less. Every possible complex impedance with a positive value of R will generate a unique point within a circle of unity radius.

The reader with a mathematical bent can analyze the circuit of Fig 3 with a complex termination, Z, and show that

$$\text{Gamma} = \frac{Z - R_0}{Z + R_0}$$

A common variation of this equation divides both the numerator and the denominator by R_0 to "normalize" the result, producing

$$\text{Gamma} = \frac{z - 1}{z + 1}$$

where the lower case value is the normalized impedance, $z = Z/R_0$.

Circles

Fig 5A summarizes the polar plot of reflection coefficients. Associated with a given gamma is a point. A vector from the origin to that point has a length equal to the magnitude of gamma. The angle between the vector and the horizontal line is the angle of the reflection coefficient. The various circles shown in Fig 5A represent gamma values of 0.2, 0.4, 0.6, 0.8 and 1.

A variation of the polar plot of reflection coefficients is presented in Fig 5B. The circles of constant gamma have been eliminated. Instead, they have been replaced

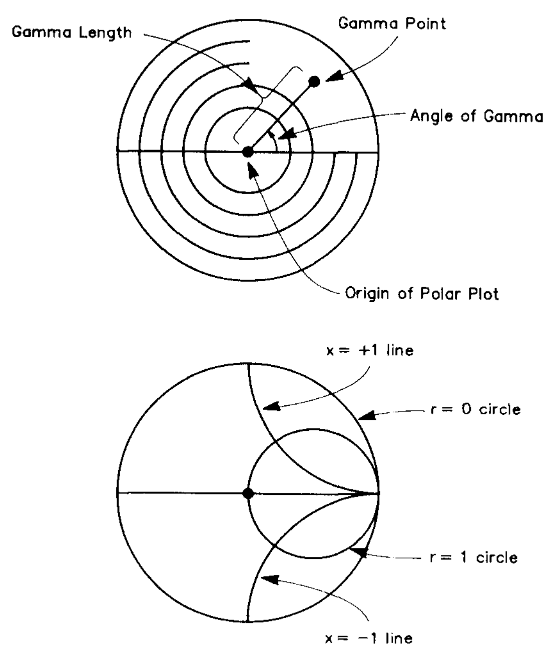


Fig 5A (top), 5B (bottom)

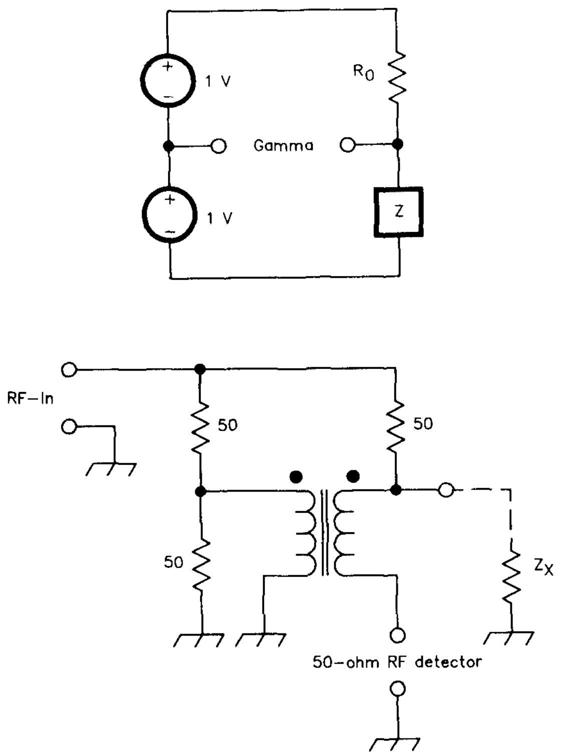


Fig 6A (top), 6B (bottom)

with smaller circles or circular segments. These represent lines of constant resistance or reactance. The horizontal line is our earlier map of all possible resistances; this is a line of impedances with zero reactance. A circle is shown that passes through the origin. This is a collection of impedances that all have constant resistance of R_0, or normalized r=1, at all possible reactance values. Other circles (missing from Fig 5B) are positioned inside the "unit circle" and represent other contours of constant R; all of these pass through the right-hand end of the center line. Two circular segments are shown. These represent lines of constant reactance with normalized values x=1 and x=-1. Even the outside edge is significant. The unit circle is the map of all possible impedances with zero resistance. The upper half of the unit circle represents all possible values of inductive reactance while the lower half maps all possible capacitive reactance.

Fig 5B is, of course, the familiar Smith Chart. This design and analysis tool is usually associated with transmission lines. Clearly, that is not required. The Smith Chart is useful as a generalized view of the impedance world. It is most useful for the study of transmission lines that have the same characteristic impedance as the R_0 value of the chart. A mismatched (or matched) transmission line has a constant reflection coefficient along

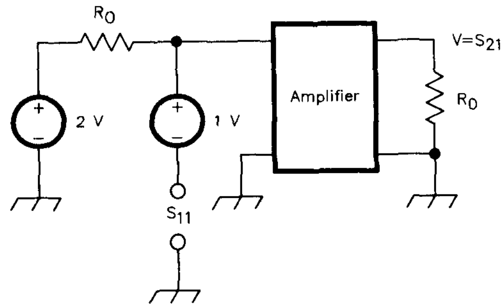


Fig 7

the line. The angle of gamma is related directly to the length of the line, or the position along the line. This is the basis for the usual circular plots that are done on the Smith Chart.

Further Extensions

The figures presented so far have illustrated a way to view the famous Smith Chart. The circuit forms are most convenient as a match to standard mathematics. Shown in Fig 6 is a variation that comes closer to a familiar measurement. Fig 6A has the previous 2-volt source replaced with a pair of 1-volt sources. The 1-volt subtraction that we did with a floating generator in Fig 3 is now done by measuring the voltage difference between the V-out node and the tap between the two generators.

Fig 6B shows a practical circuit, the familiar return-loss bridge. The two left-hand resistors serve to generate a reference at their junction. The current balun allows a grounded RF detector to "appear" between the floating nodes where gamma is measured. Other bridges can be similarly viewed.

Fig 7 is another extrapolation of the same ideas. The gamma measurement tool of Fig 3 is used to excite an amplifier or other network of interest. The amplifier output is terminated in a pure impedance, usually the same R_0 that was used for the input "bridge." The resulting circuit forms a definition for scattering parameters, the bread-and-butter tool of the RF engineer. The so-called forward gain results when the measured output is compared with the 1 volt that would appear across a matched load at the input generator. The amplifier output could be driven with a gamma measuring set, or network analyzer, while simultaneously terminating the input in R_0. The resulting reverse signals would then be s_{22} and s_{12}.

Scattering parameters are usually defined in terms of waves. While the more detailed definitions are extremely powerful, they often obscure the simplicity (yes, simplicity!) afforded by the scattering parameter view of a circuit.

Improved Mechanical Design for the Helical Antenna

By Ron Lile, KØRL
2822 Woodside Drive
Quincy, IL 62301

The helical antenna is an effective way to obtain circular polarization, but the wooden mechanical design of the antenna as typically used by amateurs leaves something to be desired. In this article, an almost all-metal mechanical structure is presented. It can be made from readily available material and can be built with common hand tools.

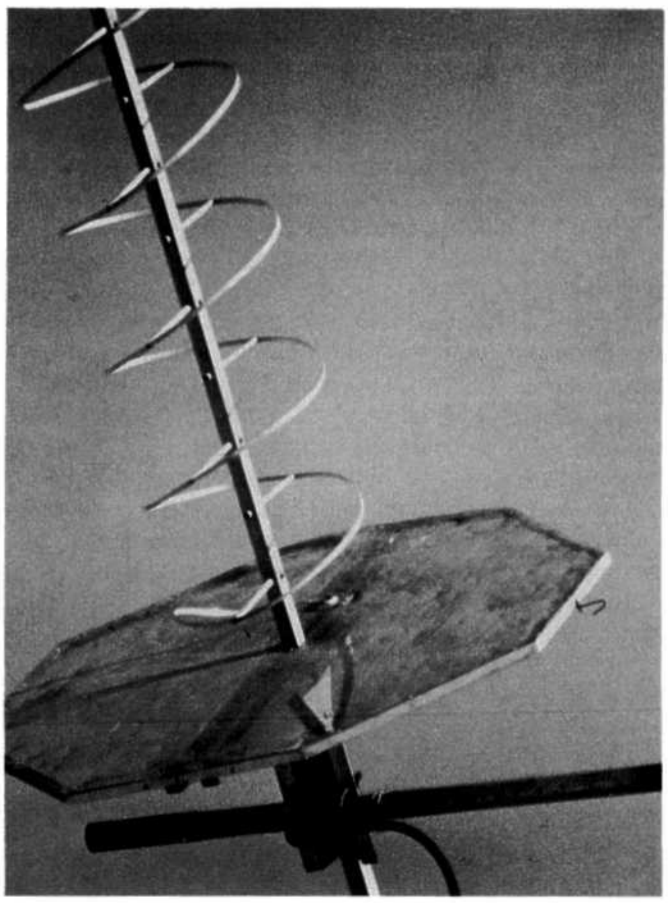
The helical antenna is a widely used satellite antenna in the amateur community as well as in the professional world. It is an antenna that can be built easily and provides good performance since the broadband nature is very forgiving of dimensional errors.

Past Techniques

For amateur use, the only shortcoming of the helical antenna is the mechanical construction. I've spent hours in local lumber yards trying to find the exact piece of wood for the antenna element supports.[1] That is, a piece of wood without knots and perfectly straight. Once the perfect piece of wood is found and brought home, antenna construction is well underway. Protection of the wood from the kind of weather it will experience on the top of the tower is an art in itself. Probably the best way to protect the wood is to apply several coats of a good marine varnish or paint (in a favorite color). The proper selection of paint or varnish will allow the antenna to last through several good years of operation. However, even with the best paint, the antenna will require repainting every few years.

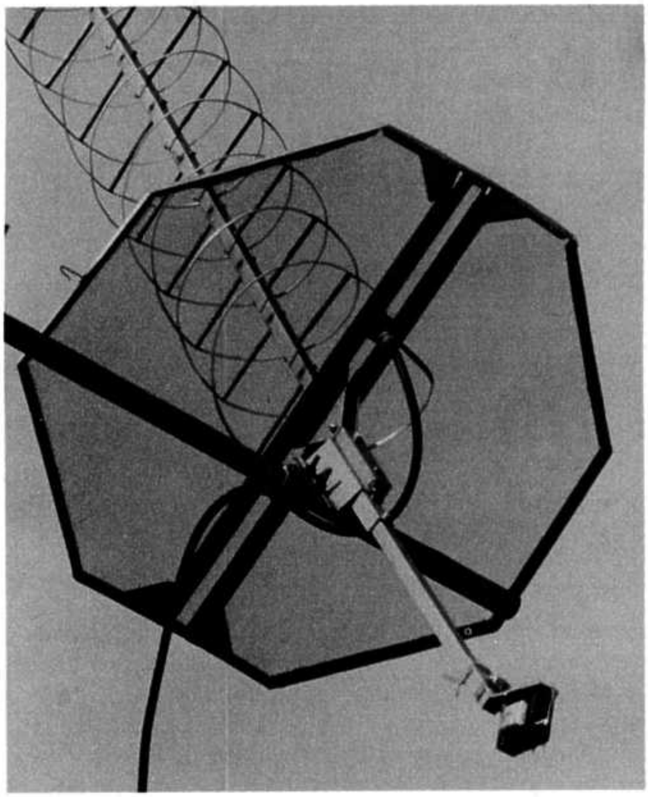
A hidden defect of the wooden-support approach does not appear for several months, perhaps a year or so. Most wood that can be purchased at local lumber yards has not been thoroughly dried before being cut to the shape or size the lumber yard sells. So in most cases, the wonderful antenna begins to twist, sag or bend after a time as the wood really dries out. Not only does the antenna begin to look bad, but the bend or twist changes the pattern thereby losing the gain and circularity which was the reason for building this type of antenna in the first place.

Okay, so wood is not the best material for the antenna even though it's readily available. Another recommended material for the support is fiberglass. Up until recently, (at least in this area) fiberglass tubing was not easily found. Paint roller extension handles are now available in approximately 4-foot lengths at local paint and discount houses, but information is not available as to the weatherability of this material. It is lighter than wood, but equal to or heavier than aluminum of the same size. Splicing for the desired length and strength can be a problem as well. It is definitely more expensive than wood or aluminum.

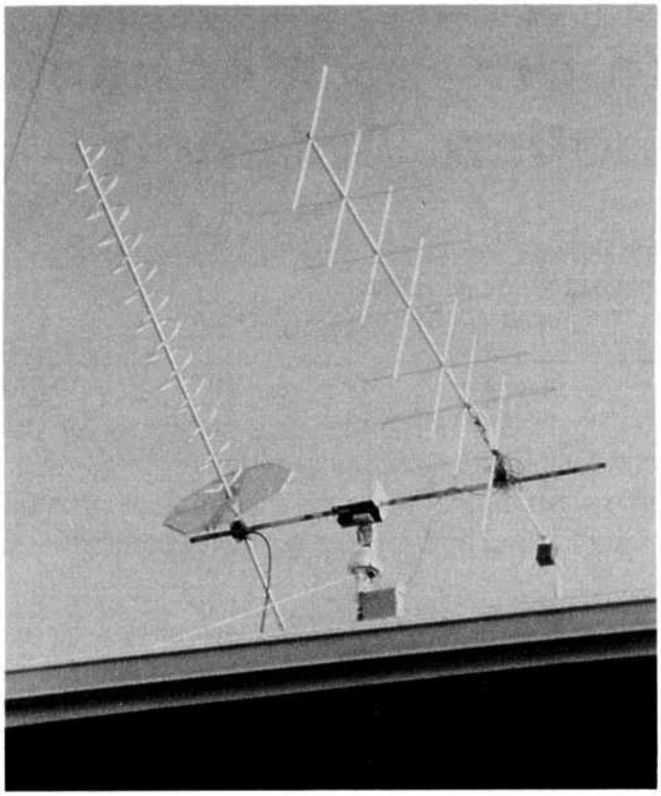


Helical Front (photos courtesy KØRL)

[1]Notes appear on page 16



Helical Back View



Satellite Antennas

Why Not All Metal?

What to do? A possible solution would be to build an all-metal antenna. The reflecting screen for the helical is already metal wire or screen, and the working radiator is metal wire or coaxial cable, but what about the radiator support? Amateur construction techniques reported to date in the popular magazines have used wood or plastic materials for the radiator support. For a long-life design, why can't a metal support be used?

I became aware of the possible use of the metal support while scanning the professional literature for the radiating element design equations. Many professionals indicted that they used a metal support for the radiating element on their antennas.[2] The reported effects varied with the size of the support, of course. However, at certain diameters of the support, as compared to the radiating element diameter, little effect was noted. A few questions came to mind: How big can the support be before major effects are noted? What is the maximum diameter that the support can be, relative to the diameter of the radiating element?

In this report, a metal support for the radiating element is proposed. In fact, the only nonmetal parts of the antenna are the radiating element insulators which hold the radiator away from the support.

Calculation of the Radiating Element

Generally, the radiating element for the helical antenna is given a circumference of one wavelength.[3,4] In an empirical investigation it was found that the gain would peak slightly higher if the circumference was approximately 1.13 wavelengths.[5] The antenna built for the measurements in Note 5 had an aluminum support tube internal to the radiating element. The diameter of the tube was 1.25 inches, the diameter of the radiating element was 4.35 inches and the operating frequency was in the 900- to 1000-MHz range. The maximum gain occurred at approximately 950 MHz. The pitch angle between turns was 12.5 degrees, a common pitch used to provide maximum gain.

For the antenna described here, the number of turns was a trade off between the ability to provide mechanical support and the desired gain, the gain being determined by the transmitter/receiver capabilities and the feed line losses. With the support material available and the transmitter power assumed, it was decided that a twelve-turn helix would do the job.[6] The antenna gain was calculated to be 14.5 dB.

The empirical equation used to calculate gain included additional gain due to the cup that is part of the reflector. Many of the helical antenna designs shown in amateur literature do not use this ¼-wavelength cup because of the added difficultly encountered mechanically. It is possible that amateurs using a helical are giving up 1 or 2 dB of gain by not using it. I felt that with the decrease in weight of this design, I could use the cup. (The drawings and pictures accompanying this article do not show the cup for clarity.)

Support Scaling

Scaling of the support, from the size used in the experiment to the frequency for amateur use, 435 MHz, was done to obtain the maximum size that could be used for the driven element support. Since the dimensions are linear factors according to the frequency, a linear scale factor can be made by dividing the higher frequency (the frequency of the model) by the desired operating frequency:

$$\frac{f_0}{f_1} = \text{scale factor}$$

where, f_0 = frequency of operation of the model antenna
f_1 = frequency of operation for the antenna to be built

In this case, 435 MHz is the frequency range for the antenna to be built and the model antenna was operating on a frequency of 950 MHz, therefore the linear scale factor is calculated to be:

$$\frac{f_0}{f_1} = \frac{950}{435} = 2.2$$

The radiator support dimension of the model antenna is multiplied by the scale factor to create the actual dimensions for the radiator support at the desired operating frequency. Based on this scale factor, the antenna radiating element support could be as large as 2.5 inches in diameter.

Design Trade-Offs

The next step was to find out what aluminum tubing was available for the support construction. Checking local supplies, I found several candidates. Round and square tubing was available in several diameters up to 1.5 inches. The larger sizes were available in 8- and 10-foot lengths. Larger sizes and longer lengths are available through mail order or metal-supply houses in larger cities. The metal grade (the familiar 6061-T6 given for aluminum is the strength rating and is reached by alloy composition and heat treatment) is not given for material in the usual handyman's store. Some of the material is soft for easy cutting and use at home; it is not suitable for antenna building. If the metal strength can not be determined from the information available, it would be wise to order an appropriate size tube from one of the antenna material vendors listed as advertisers in the back of any ham magazine or purchase it from a distributor of tubing stock. Reynolds-type 6061-T6 is good, readily available material suitable for use in this application.

In trading off strength against weight, something around 1 inch in diameter appeared to be reasonable. About this time I was given a piece of 13⁄16-inch furniture tubing, which was used in the final design. (This just happened to be very strong square tubing.) Square tubing has been used before and was found to work satisfactorily.[7] A previous author (see Note 7) used Reynolds type 4860 for an antenna support.

The square tubing also made attaching the rotor boom and reflecting screen somewhat easier, although the final brackets used in the design can accommodate a round support.

Mechanical Design

The final design is shown in Fig 1. The element insulators on this version are made from wooden dowel rods, but plastic insulating materials could be used. The radiating element is attached to the dowel rods via short pins and epoxy. The radiating element was made from flat, aluminum

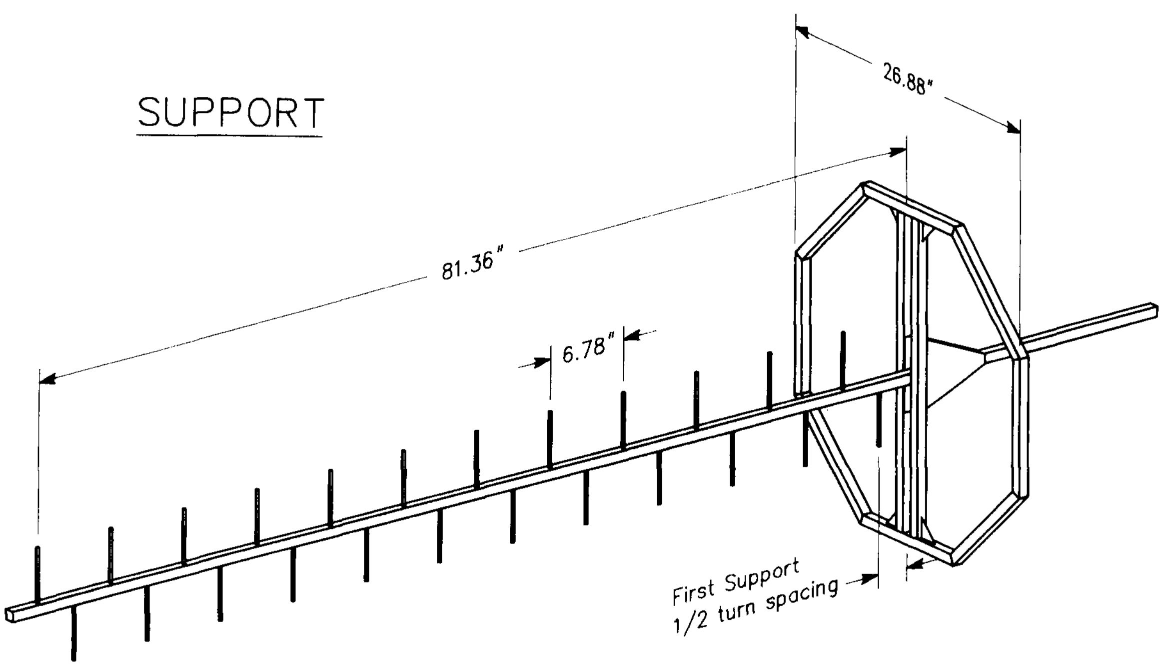
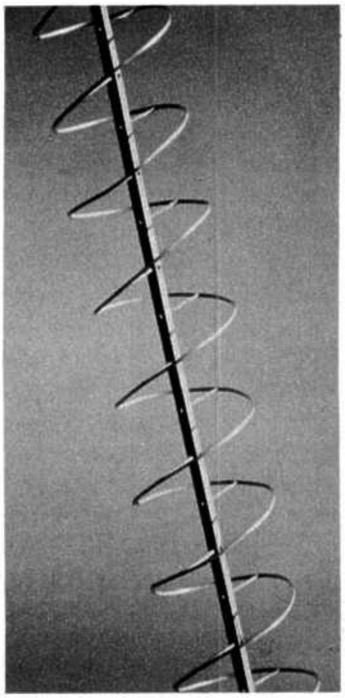


Fig 1—Metal supporting structure dimensions for the helical antenna.



Helical Radiating Element

wire stock, and a small hole was measured and drilled in each of the mounting locations at the circumference distance used in the design. This aided in forming the spiral that the radiating element makes on the support. Other attachment methods could be used depending on the material used for the radiating element.

The wooden dowel rods (after being treated with paraffin) are attached to the radiator support using aluminum sheet metal screws. Here the square tubing is really a bit better than round tubing. When pulling the screws down tight, the square tubing is not deformed. The round tube would be flattened and this would form a weak spot in the support.

A clearance hole for each sheet metal screw was drilled at each of the spacer locations. Then a large clearance hole for the dowel rod was drilled at 90 degrees to the screw clearance hole. The hole for the dowel rod was drilled all the way through the aluminum tube, thus giving support to the dowel rod so that the screw would tightly clamp the dowel rod to the support.

The wire-mesh screen is attached to an aluminum angle frame using pop rivets. Aluminum screen wire was used for the reflecting screen. Many of the published helical antennas use hardware cloth or chicken wire. However, my goal was a long-life antenna without structural failure or noise generation. The use of galvanized hardware cloth or chicken wire with aluminum could cause RFI due to the dissimilarity of metals between the wire and aluminum used for the frame. Corrosion due to the differing materials would also shorten the structural life of the antenna.

Performance

The antenna electrical performance is not easy to verify without a good antenna range. Crude measurements made using ground based sources and satellite signals give pattern results approximately as calculated. Gain comparisons between a Cushcraft 416TB and this antenna show them to be roughly comparable, with the helical averaging slightly higher numbers (more consistent numbers) when satellite signals are used for the signal source.

The point of this antenna design was not to verify the calculations and measurements of the many amateurs and professional who had built antennas prior to this one. Rather, the antenna was designed to withstand the elements and provide many years of reliable performance. This design has lived up to the expectations desired.

Table 1 Antenna Dimensions

Center Frequency	435 MHz
Wavelength	68.73 cm (27.06 in.)
Diameter of Reflector	68.73 cm (27.06 in.)
Diameter of Radiating Element	0.006 to 0.05 λ
or	0.413 cm to 3.44 cm (0.162 to 1.35 in.)
Alpha (pitch angle)	12.5 degrees
K for maximum gain	
circumference of radiator	1.13
diameter of radiator	24.73 cm (9.73 in.)
Turn spacing for pitch	17.21 cm (6.778 in.)
Support boom length (12 turns)	206.59 cm (81.336 in.)

Conclusions

The mechanical structure shown has been in service for 10 years with no mechanical or electrical failures. The antenna has suffered through heavy Texas winds and thunderstorms and Illinois ice, wind and snow.

The structure is lighter in weight than previous wooden supports, reducing wear and tear on the elevation and azimuth rotors.

This same construction technique should be suitable for helical designs on the other frequencies used for amateur satellite communications. It can produce a light but strong mechanical structure sure to give years of maintenance free service.

Notes

[1] Allen, B., W7US, "A Mode J Helix," *AMSAT Newsletter*, Vol 12, No. 2, June 1979, p 30.

[2] Donn, C.H., "A New Helical Antenna Design for Better On-and-Off Boresight Azial Ratio Performance," *IEEE Transactions on Antennas and Propagation*, Vol AP 28, No. 2, March 1980.

[3] *The ARRL Antenna Book*, Chapter 19, pp 22-23.

[4] Kraus, J.D, *Antennas*, New York: McGraw-Hill, 1950, Ch 7.

[5] Wong, J.L. and King, H.E., "Characteristics of 1 to 8 Wavelength Uniform Helical Antennas," *IEEE Transactions on Antennas and Propagation*, Vol AP 28, No. 2, March 1980, pp 291-296.

[6] Harris, E.F., "Helical Antennas," *Antenna Engineering Handbook*, H. Jasik ed., New York: McGraw-Hill, 1961, Ch 7.

[7] Kunde, Keith, K8KK, "Try a 'Dopplequad' Beam Antenna for 2 Meters," *QST*, February 1985, pp 28-31.

Components

By Mark Forbes, KC9C
PO Box 445
Rocklin, CA 95677

DESIGN AUTOMATION HEPA-PLUS

Design Automation manufacturer's CAD/CAE design software for RF power amplifiers should be interesting to some of you who want to design amplifiers for amateur applications. The program is called HEPA-PLUS (High Efficiency Power Amplifier-PLUS).

With the program, you can design linear (single ended or push-pull), class AB, B, or C or switching-mode amplifiers (class E or F). The program automatically optimizes the proposed design and also identifies and helps prevent design errors and omissions.

The package consists of four modules. HEPA-SIM is the simulation program. It directly simulates circuit steady-state periodic response, input power, output power, and all power dissipations. According to the company, simulation computation takes only 280 ms on a 20-MHz 386 computer.

HEPA-OPT (or HEPA-OPT/WB) automatically optimizes the design when used with HEPA-SIM. HEPA-OPT optimizes at one frequency; HEPA-OPT/WB (wide band) optimizes at up to sixteen frequencies simultaneously.

HEPA-DES allows you to generate a rapid, preliminary design to meet your specified parameters. And the final module, HEPA-TEV, evaluates power transistors to determine parasitic losses and plot efficiency vs frequency and output power.

I ran the evaluation copy of the program and found that it could not only save a tremendous amount of time in RF amplifier design, but it can eliminate many errors without having to build and troubleshoot the circuit.

The only down side to the program is that it is definitely not inexpensive. If you are interested in the program, contact Design Automation, Inc, 809 Massachusetts Ave, Lexington, MA 02173-3993, or phone 617 862-8998.

3-V DUAL OP-AMP

There is a growing trend toward components that operate down to three volts. The thrust behind the trend is the growing number of battery-powered and hand-held products. Analog Devices have introduced the OP-295 dual op amp that operates on either 3 V or 5 V. Maximum supply current is only 300 micro amps, and the typical gain-bandwidth product is 75 kHz.

The parts cost about $2.00. To get more information on the OP-295, contact Analog Devices, One Technology Way, Box 9106, Norwood, MA 02062, or call 617 329-4700.

NEW RF CONNECTORS

Molex, Inc has introduced a new line of RF connectors, including the old favorites PL-259, BNC, and F, and also connector adapters. They also include some of the more exotic connectors such as TNC, Twinax, and Mini-UHF.

All assembly options are included...solder on, crimp, and screw on. They are also available in nickel, silver, gold, and black chromate plating. Contact Molex for more information at 2222 Wellington Ct, Lisle, IL 60532, or phone 708 969-4550.

IN-RUSH CURRENT LIMITERS

Siemens has a new line of thermistors which limit in-rush current in circuits. These are especially useful in digital, microprocessor, and CMOS circuits. The ICL (In-rush Current Limiters) can handle currents up to 20 amps.

There are six models available: S153, S234, S235, S364, and S464. Call Siemens to get an information kit on these valuable power conditioning components: 800 888-7729.

VHF+ TECHNOLOGY

By Geoff Krauss, WA2GFP
1927 Audubon Drive
Dresher, PA 19025

In my last column, I mentioned the Rohm RHF1204 GaAsFET, a very low-noise device for microwave LNAs, and gave two sources for small quantity purchase; I have since been in contact with a helpful young lady (Sharon) at one of the sources, Garrett Instruments and Components/IEU Inc (3130 Skyway Drive #701, Santa Maria, CA 93455, tel 805 922-0594). They have a $10/order minimum and will take phone orders paid with MC/VISA; this makes things very easy for ordering one or two devices, with a single-piece price of $11. Observant readers will note that the Rohm ads were for devices with pricing of about $4 or $5, but that was in 10k lots ($10-15 is about what most GaAsFETs are going for, in small quantity, today). Should you build any RHF1204 LNAs, we would certainly like to know what sort of performance *you* get.

I have tended recent columns toward more coverage of the UHF/microwave area, because that appears to be where most new amateur VHF+ work is being done. A good indicator of this is the number of 902-and-up stations present in 1992 contests. The June '92 VHF Contest results were finally published in December *QST,* and the single/multioperator lists clearly show the results up to 2304 MHz. Keeping in mind that fantastic E-skip and tropo conditions were present during both the June and September contests, which led to some extremely high station and grid totals on 50-1296, I want to make sure the growth at even higher frequencies is documented. Once I found out what some of the scores were, I requested *QST* to include them in the box scores, but there is apparently only so much available space... While I only have the W2SZ/1 September '92 numbers, they are still instructive, even if one figures, keeping in mind that a goodly percentage are the MGEF's own rover group, that maybe one-half to two-thirds of the total station count are nonaffiliated stations. The maximum number of rover-attributable grids is about 7 (as shown by the 24-GHz grid total); any number of grids over 6 or 7 is due to nonaffiliated stations and definitely shows that there is much more growth above 2 GHz than below:

	902	1296	2304	3456	5760	10 GHZ	24 GHz
QSOs	73	115	62	46	38	42	28
Grids	29	43	21	15	15	11	7

Obviously, the 2.3/3.4/5.7 GHz work is narrowband, as is most of the 10-GHz work beyond 7 grids.

It is unfortunate that the contest results do not list each QSO over a distance greater than some standard, because that list would certainly tell us how good propagation was and also give us some idea of the station distribution. From what I've heard, there are now a goodly number of stations on microwave bands with the capability to make "forward scatter" contacts well beyond the horizon, but we could use lots more of them. If such a listing were to be published for each contest, and if it caused more stations to get on, for any reason, it would be counted a success. I would certainly like to see any UHF/microwave (especially 3456 and up) listings for other stations. Being more indicative of hard work and good operating than merely being present for skip/tropo openings, such activities should be acknowledged.

These contest results were much discussed at the Microwave Update '92 Conference, held October 16-18, 1992, in Rochester, New York, under the sponsorship of the Rochester VHF Group. The *Proceedings* of the conference have been published and are available from ARRL for $12. The book covers most of the talks, which included two on HEMT LNAs, three on propagation/beacons and several on various systems aspects; missing from the published text is my lead-off talk on "Microwave Systems Principles" (see any amateur MW book), which was designed as a lead-in for the less-than-fanatic

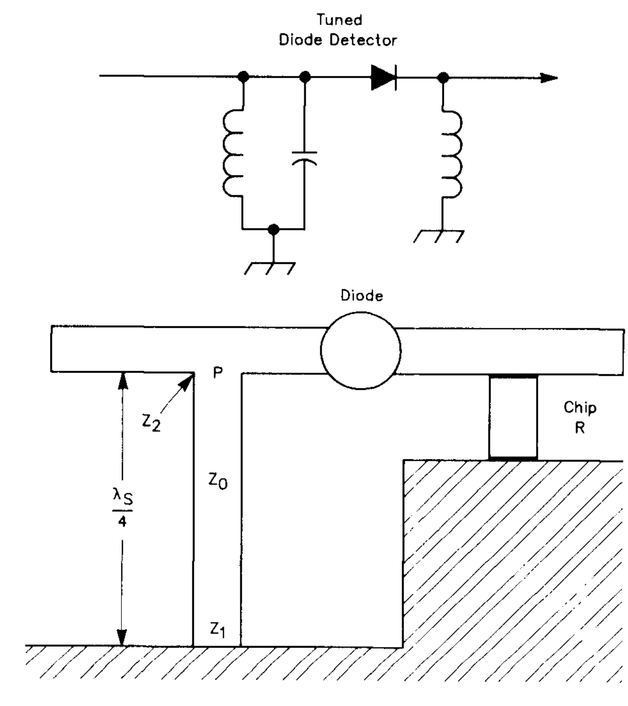


Fig 1

attendee (of which I personally met several, but very few who were totally novice), and the 24-GHz operation talk of Dick Frey, WA2AAU, about the W2SZ/1 work on that band. Dick's slides were copied as a hand-out, and contain a lot of interesting atmospheric data that serious 24-GHz workers should review. You can get your copy by either finding someone who brought back a copy from Rochester or by sending 'AAU (or me, if you get desperate enough) a letter request with a large SASE.

In addition to two days of fine talks (out of which I have to single out three especially noteworthy presentations by two very skilled individuals: Kent Britain, WA5VJB, and Paul Shuch, N6TX, both of whom teach at the college level—Kent's Stealth bomber presentation was, to my mind, all by itself worth the price of admission), door prize drawings and discussions with about 100 of the Faithful, including many of the premier and well-known MWers of North America and "selected other locales," the Thursday Surplus Tour and Friday Evening One-room Flea Market more than slaked the attendees' thirst for acquisition and exchange of MW goodies. I found this conference to be one of the best that I have attended over the course of something like a third of a century; I am told that the next conference, Microwave Update '93, is to be held in Atlanta, Georgia.

Another 1992 conference for which *Proceedings* are also available for $12 from ARRL is the 26th Conference of the Central States VHF Society. This conference incorporated the first Worldwide VHF Ionospheric Propagation Symposium, with 3 sessions of 9 papers, covering one of the few VHF areas which most of us will never know enough about, and in which area we can all work simply by being on-the-air to recognize and report interesting propagation. There are a number of other interesting articles in this proceedings in areas as diverse as EME, beacons, microwave and space communications.

The Radial Line Stub

If you have done any work above 50 MHz, you have probably encountered at least one of the quarter-wave or half-wave transmission line stubs. The quarter-wave stub (Fig 1) transforms a low impedance Z1, typically a short circuit, at one end of a transmission line with a characteristic impedance Z_0, to a high impedance Z2 at the end which is 90 electrical degrees away; this is the same as saying that $Z1/Z_0 = Z_0/Z2$, or $Z2 = SQR\ (Z_0{*}Z_0/Z1)$. This configuration can be very useful for providing a dc short which looks to RF signals like an open circuit (or at least a very high impedance) at the line end. Use as a filter element, particularly for bandpass filtering, is also well known. Note that the wavelength of the transmission line *does* change with impedance level, so that the quarter wavelength will also change. Any book on transmission lines should include the appropriate formulae for calculating wavelength in the transmission line. Similarly, the half-wave line (Fig 2) is also used for filters, and for biasing circuitry. The best known bias configuration (Fig 3) is a half-wave line of two different impedances: The first half of the half-wave line (or a quarter-wave) has a very narrow width, and therefore a high impedance, and is the line segment having one end connected to the circuit to be biased. The other quarter-wave line is a segment of much greater width, and lower impedance, and has an open circuit at its far end. The open circuit is reflected back to the attached point P as anoth-

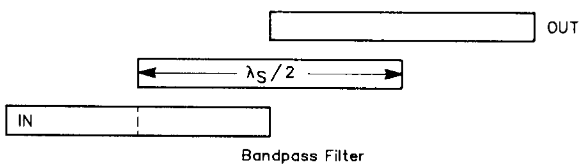


Fig 2

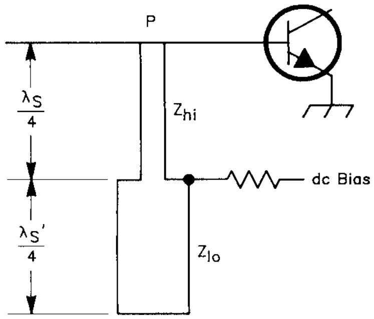


Fig 3

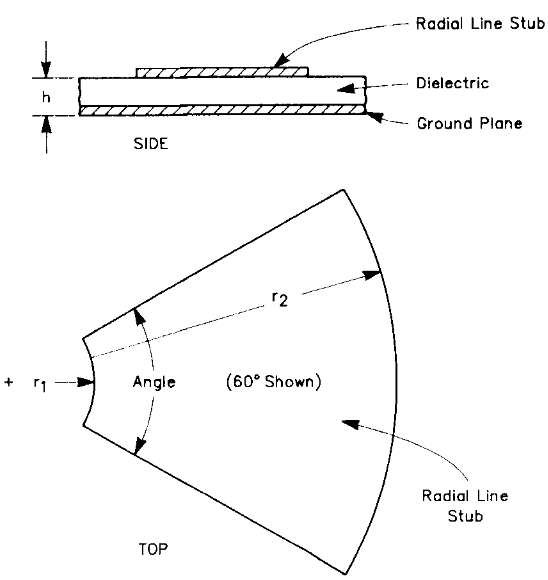


Fig 4

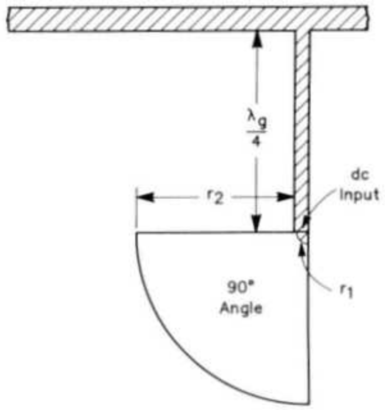


Fig 5

er open circuit for both RF and dc so there is no disturbance, but allows a dc voltage to be impressed at point P anyway. Again, note that the wavelengths are not the same in the high and low impedance sections, and these will be of somewhat different physical lengths for the same 90-degree electrical length. Another problem is that the resistive transformation effect of a quarter/half wave length is true only for the one design frequency; a few percent change in frequency will change the stub into a very reactive load at the attachment point P. As a mater of fact, with low impedance lines (having broad width) it is not easy to determine where the attachment P actually occurs, and small changes in attachment point can often have large and undesirable effects.

The way around these problems is to use a radial line stub, which not only has a broad resonance region, but also has a very well located attachment point. Fig 4 shows the stub: an angular sector, usually with the included angle being 60 or 90 degrees. The attachment point is at the smaller end, where the sector has a radius r_1. The size of the stub is determined by the maximum radius r_2, which is determined by the formula:

$$r_2=10^{A\log(fg\sqrt{\varepsilon r})+B\log(h)+C\log(r1)+D}$$

where f_g is the center frequency in GHz, ε_r is the dielectric constant of the microstrip material, h is the thickness of that dielectric and A, B, C and D are constants which depend on the angle of the stub. For the common 60 and 90 degree stubs, the constants are:

	A	B	C	D
60 degrees	–0.8232	0.0572	0.1169	–0.8082
90 degrees	–0.8510	0.0614	0.0877	–0.8695

To save you a lot of calculation, for a microwave substrate with ε_r of 2.2 and h=1/32 inch (0.03125 inch), for a minor radius r_1 of 2 mm (0.079 inch), the major radius r_2 would be:

	60 Degrees		:	90 Degrees	
Frequency (GHz)	mm	inches	:	mm	inches
0.902	81.412	3.205	:	67.995	2.677
1.296	60.412	2.378	:	49.950	1.967
2.304	37.620	1.480	:	30.612	1.205
3.456	26.944	1.061	:	21.679	0.853
5.760	17.694	0.697	:	14.036	0.553
10.368	10.906	0.429	:	8.511	0.335
24.192	5.430	0.214	:	4.139	0.163

Attachment is as shown in Fig 5.

Bits

Central States VHF Society: Call for Papers

The Central States VHF Society has announced a call for papers for their annual conference. The 1993 conference is scheduled for July 29 through August 1 at the Lincoln Plaza Hotel in Oklahoma City, Oklahoma. Emphasis this year will be on the beginner to weak-signal VHF-and-above operating, with a special session devoted to this subject.

If you're interested in presenting a paper or having one published in the *Proceedings,* contact Tommy Henderson, WD5AGO, 12476 E 13th, Tulsa, OK 74128, tel 918 438-0099. Papers must be received by June 1 for inclusion in the *Proceedings.*

Microwave Update '93

I know, you haven't quite recovered from this year's conference, but...it's not too soon to start thinking about Microwave Update '93! Atlanta will be hosting the '93 conference and your's truly had his arm twisted enough to cry "uncle." Actually, I'm really excited about having the conference here in Atlanta. With help from the conference committee (KK7B-technical program, WB4MBK-test equipment, and others), I hope to bring you one of the best conferences yet! My goal is to have everyone leave the conference with a renewed enthusiasm for operating, building and experimenting with amateur microwave equipment.

The Northwest Atlanta Hilton has been selected for the conference which will be held on September 24-25, 1993. I hope this date is a good choice, as it is difficult to find a date that has hotel availability and does not conflict with fall contests and other conferences. Details on registration and hotel reservations will be sent out after the first of the year to all past attendees. We will handle this as in the past with you making your own hotel reservation. I will be taking the conference registrations. If you want to be sure that you receive a registration package, drop me a note or call so that I can be sure your name is on the mailing list (404 333-2136 [W], 404 998-6971 [H]). A block of rooms is being held at the Hilton for $62 a night. Right now, I'm planning that the conference registration and banquet cost will remain the same as last year.

Rick, KK7B, has volunteered to put together the technical program. Since I've never known Rick to do anything short of "first class," I am especially excited about the program! We would like to hear from you soon if you have been working on something of interest to the microwave community. Give Rick a call at 906 487-2848 if you have an idea; he can help you develop it into a talk.

Watch for more details. Hoping to see you in Atlanta next September!—*Jim Davey, WA8NLC*