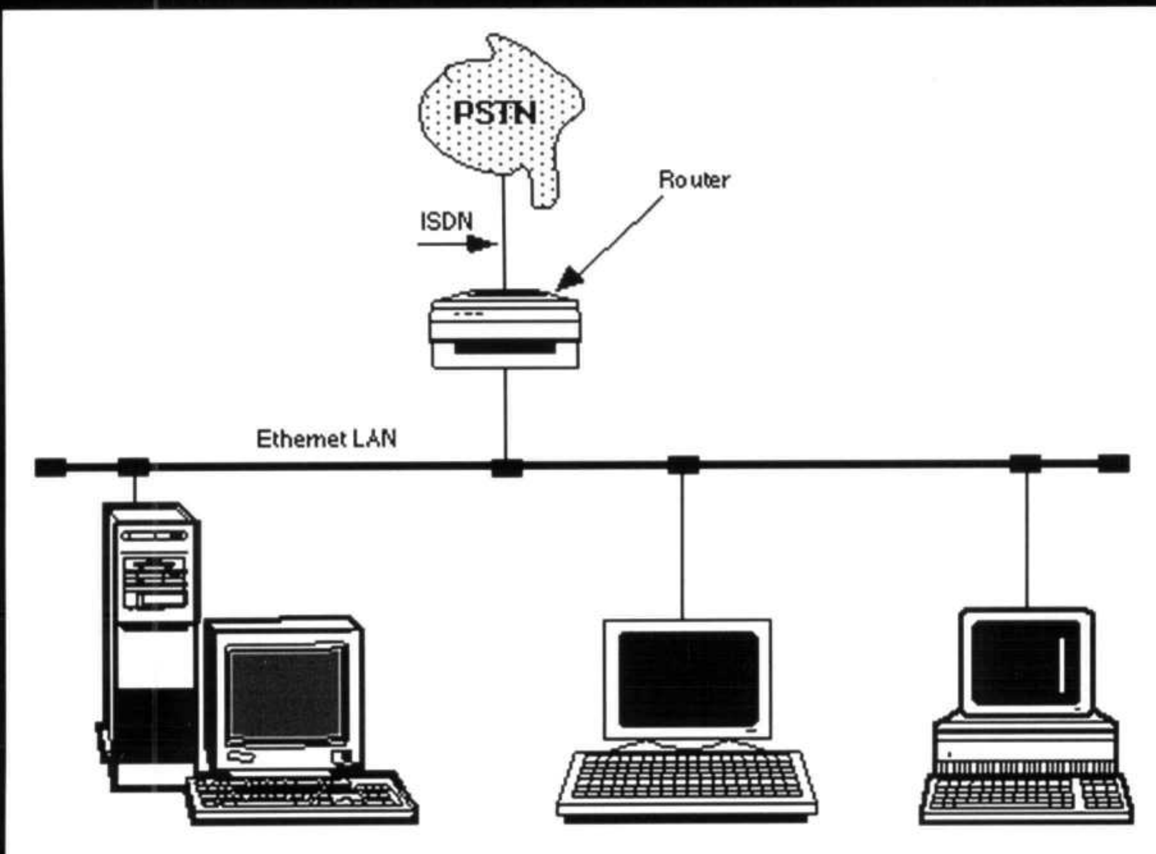
$1.75

QEX

ARRL Experimenter's Exchange

October 1996



The Speedy Solution: ISDN

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

QEX (ISSN: 0886-8093) is published monthly by the American Radio Relay League, 225 Main Street, Newington CT 06111-1494. Subscription rate for 12 issues to ARRL members is $12; nonmembers $24. Other rates are listed below. Periodicals postage paid at Hartford CT and at additional mailing offices.

POSTMASTER: Form 3579 requested. Send address changes to: QEX, 225 Main St, Newington CT, 06111-1494

Issue No. 177

Advertising Information Contact:

Brad Thomas, KC1EX, *Advertising Manager*
American Radio Relay League
860-667-2494 direct
860-594-0200 ARRL
860-594-0259 fax

Circulation Department

Debra Jahnke, *Manager*
Kathy Fay, N1GZO, *Deputy Manager*
Cathy Stepina, *QEX Circulation*

Offices

225 Main St, Newington, CT 06111-1494 USA
Telephone: 860-594-0200
Telex: 650215-5052 MCI
Fax: 860-594-0259 (24 hour direct line)
Electronic Mail: MCIMAILID: 215-5052
Internet:qex@arrl.org

Subscription rate for 12 issues:

In the US: ARRL Member $15, nonmember $27;

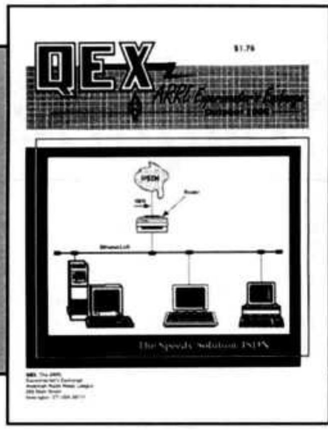
US, Canada and Mexico by First Class Mail: ARRL Member $28, nonmember $40;

Elsewhere by Surface Mail (4-8 week delivery): ARRL Member $20, nonmember $32;

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

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

In order to insure 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.



About the Cover

Richard Parry, W9IF, tells how ISDN is moving wireline connections to new heights.

Features

3 **A Practical Approach to Computerized Control of Stepper Motors and Relays**
By Ron Pierce, NIØL

11 **ISDN: The New Legal Limit**
By Richard Parry, W9IF

15 **A Simple T/R Sequencer**
By Zack Lau, KH6CP/1

Columns

19 **Upcoming Technical Conferences**

October 1996 QEX Advertising Index

American Radio Relay League: 21, 22, Cover IV
Communications Specialists Inc: 23
Down East Microwave, Inc: 22
HAL Communications Corp: 24
NOBLE Publishing: 23
PC Electronics: 23
Sescom, Inc: 24
Tucson Amateur Packet Radio Corp: 23
Z Domain Technologies, Inc: 22

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: 860-594-0200
Telex: 650215-5052 MCI
MCIMAIL (electronic mail system) ID: 215-5052
FAX: 860-594-0259 (24-hour direct line)

Officers

President: RODNEY STAFFORD, KB6ZV
5155 Shadow Estates, San Jose, CA 95135

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*.

Any opinions expressed in *QEX* are those of the authors, not necessarily those of the editor or the League. While we attempt to ensure that all articles are technically valid, authors are expected to defend their own material. Products mentioned in the text are included for your information; no endorsement is implied. The information is believed to be correct, but readers are cautioned to verify availability of the product before sending money to the vendor.

Empirically Speaking

The Electronic ARRL

It's an electronic age—that's hardly news—and ARRL is striving to keep up. You already know about *ARRLWeb* (**http://www.arrl.org/**), or you should—we've certainly mentioned it often enough. But that's not the only thing we're doing in the electronic venue.

These days, more and more information is being disseminated as computer data. For several years now, we've been getting suggestions from ARRL members that they'd much rather keep their old ARRL publications on CD-ROM instead of on musty bookshelves. In 1995, we responded with the *1995 ARRL Periodicals CD-ROM*, containing the complete contents of the 1995 issues of *QST*, *QEX* and *NCJ*. And there will be a 1996 version, too. (Probably with a better name, though!) But what of those older *QST*s? Shouldn't those be in electronic form, too? Indeed. And now they are.

The problem with the older issues of *QST* was that they didn't exist in usable electronic form. With over 80 years of published *QST*s, only the last couple being done wholly electronically, the solution was limited to scanning the pages of older issues, and that's what we've done. The result is *QST View*, a collection of past issues of *QST* on CD-ROM. Using *QST View*, you can display and print pages from past issues of *QST*. To make it more useful, *QST View* includes a searchable database of *QST* articles, allowing you to easily find the one you're looking for.

Initially, *QST View* sets are available for the years 1985-89 and 1990-94. Sets for earlier years will follow soon.

The other ARRL publication that members mentioned a lot was, of course, the *Handbook*. As one of the premier reference works for amateurs, electronics technicians and engineers, a CD-ROM version of the *Handbook* is potentially highly useful. That's available now as the *ARRL Handbook CD 1.0*. It contains all of the text and illustrations from the seventy-fourth edition of the *Handbook*. And we've added sound clips you can play to *hear* what the *Handbook* is talking about!

We've got other projects in the works, and we're actively considering future additions to our line of electronic publications and software. If you have ideas, we'd love to hear about them, most easily via email to **jbloom@arrl.org**.

Mea Culpa

You'll have noticed, no doubt, that this issue of *QEX* arrived *very* late. The reason has to do with the aforementioned electronic publication products. We were committed to getting those out in time for Christmas but were a bit short of human resources. Something had to give; unfortunately, *QEX* was that something. This doesn't reflect on the perceived importance of *QEX*, a concern some readers have expressed. It has more to do with the fact that *QEX* information is not particularly time-critical. Zack Lau's T/R sequencer is just as useful this month as last month.

Still, the delay in getting *QEX* out is not something we like—or will accept. With the high-priority projects out of the way, we plan to catch up *QEX* quickly. And we're looking at ways of reallocating resources to avoid any repetition of this situation. For now, though, our abject apologies.

This Month in *QEX*

It may be a solid-state world, but a lot of applications end up requiring moving mechanical components. The challenge is to marry solid-state electronics with electromechanical devices. To do that, Ron Pierce, NI0L, shows us "A Practical Approach to Computerized Control of Stepper Motors and Relays."

That 28.8 modem seem slow? (It seemed so *fast* when you first got it!) You've probably heard about ISDN as an alternative. What is it? What are its advantages and disadvantages? Richard Parry, W9IF, explains all about "ISDN: The New Legal Limit."

Lashing together that microwave station requires careful interfacing. One of the key components is the control circuitry that turns circuits on and off at the proper time. "A Simple T/R Sequencer," by Zack Lau, W1VT, keeps the electrons in line.

You can't have too much information about the state of the experimenter's art, and once again, "Upcoming Technical Conferences" tells you where it's happening. *—KE3Z, email:* ***jbloom@arrl.org***.

A Practical Approach to Computerized Control of Stepper Motors and Relays

Using your PC's parallel port to run stepper motors or control relays.

By Ron Pierce, NIØL

Many articles have been written using a compatible PC as a controller. And much has been done concerning implementation of stepper motors. This project draws these various techniques together to make a practical controller, as simply and inexpensively as possible. The devices used here are controlled by an "old" 386SX33. Even this supposedly slow machine must be delayed to actuate the much slower electromagnetic devices that are used. Most of the testing was done on a Tandy XT laptop. After testing, it was determined that you can control up to 22 relays and control two motors with one 8-bit port. And if you don't want to control any motors, you can control 30 relays. A state-of-the-art computer is not needed to do this type of controlling.

307 11th Ave NE
Independence, IA 50644
emai **NIØL@juno.com**

We are going to cover five main areas: the parallel port(s); an external integrated circuit whose primary function is to turn relays on and off; a simple interface IC and switching transistors to make a stepper motor rotate; the software which controls it all, and finally; we will look at a method of controlling stepper motors using small relays and small switching transistors. This will allow control of three motors with four data bits.

As far as the stepper motors are concerned, they are being controlled in the unipolar mode. That is, the direction of the current in the motor windings does not change. The stepping and direction of rotation of the motor is controlled by the sequence of actuation. Other modes of operation can be found in the literature listed in References.

The unipolar is the simplest and most economical. If you are a new or beginning ham and like to play with computers, here is your chance.

Your parallel port is the DB-25 female connector on the back of your computer. Most computers will allow use of up to three parallel ports, LPT1, LPT2 and LPT3. These ports are addressed at decimal 888 for LPT1, decimal 632 for LPT2, and LPT3 is located at decimal 956. To confuse things, some of the older MGA/CGA video controllers have an LPT1 parallel port with its address at decimal 956. Keep this in mind as you proceed.

You will need a small supply of DB-25M connectors to mate with the computer port and you will have to make yourself some cables. Pins 2 through 9 are DB0 through DB7. Pins 18 through 25 are ground. That is all we are going to consider.

Pins 2 through 9 will output a high or a low. We will have to program these pins to output the correct bit pattern. Obviously, the port does much more since its original function was to drive a computer printer. There are other bits which could be used as a 1-byte read back into the computer for status of tasks accomplished. But that is beyond the scope of what we are doing here.

If you are even slightly interested in parallel-port controllers, get access to a copy of *Controlling the World with Your PC*, by Paul Bergsman, N3PSO. This book is the "Bible" for parallel-port implementation and is inexpensive as books go these days. Also recommended is an archived file on the Circuit Cellar BBS, PARALLEL.ARC. See References for phone numbers). This file contains all the addresses you need to implement a port, plus the input addresses, in case you ever need that information. If you are not familiar with archived or zipped files, download the uncrunching files. They are listed at the beginning of each section.

With all this in mind, it is time to start thinking about hanging some external electronics across the aforementioned port.

A 4-Phase Unipolar Stepper Motor-Driver Circuit.

This method of motor implementation has been criticized as being inefficient—which it is, as we only utilize half of the motor coils at any given time. However, it is perfectly valid for our purposes. If a large enough motor is chosen in the first place, you will have more than enough torque. The physical size of the stepper motor can be quite small, ie, one that drives an air variable. The most efficient method is a full-bipolar driver. Wherein, all the coils are energized and the magnetic field is reversed in one coil side or the other, depending upon the direction of rotation desired. For bipolar implementation, a transistor bridge or bipolar-driver IC is required. The complexity of the circuit and number of components is increased.

The 4-phase method requires four transistors, four diodes and four resistors, plus an interface IC. It is the simplest method of driving a 5- or 6-wire motor and adequate torque to rotate electronic devices, such as air-variable capacitors, and roller inductors can be generated. This method will not work on 4-wire motors.

Now that the cheap-and-simple-method warning has been issued, a couple of comments on motor hookup. Two motors salvaged from the same printer, same manufacturer, were found to be wired internally in a different manner. In this case one motor has to be wired up with the windings pulsed sequentially and the other alternatively. And when being tested on the test board in a single-coil mode, they work properly. But when attempts are made to use dual-phase, ie, two coils, they get fussy about how they are hooked up. The windings must be phased correctly. So don't be surprised if some experimentation is necessary. Once the transistors are wired, manual stepping can still be done, just move the switches over to the base resistors of the driver transistors.

This is why you need to build a motor test board. Before attempting to rotate your motor with the computer port, it is instructive to make the motor rotate manually. Building a motor test board is recommended. See Fig 1. This does not have to be elaborate. A

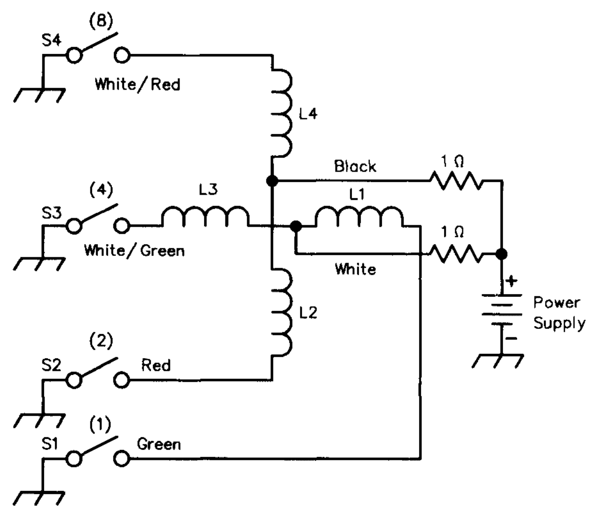


Fig 1

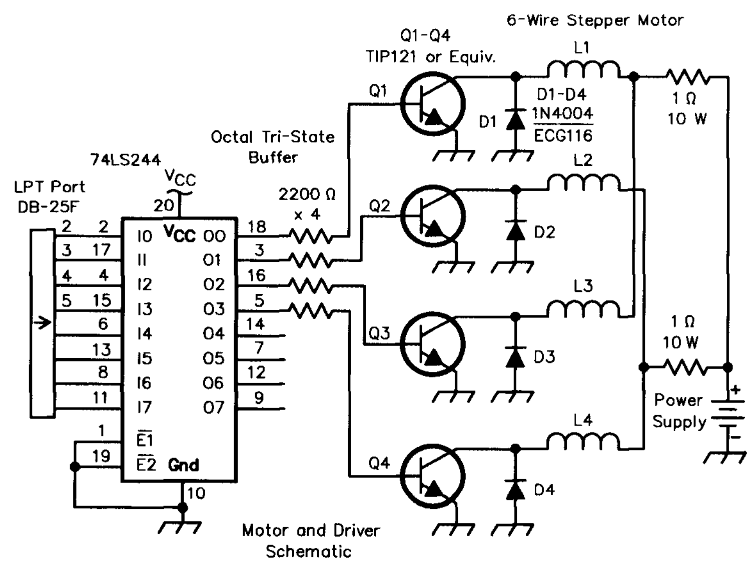


Fig 2

piece of board and four switches will do. Mine is a sawed-off scrap of an old shelf. The motor mounts are 3/4-inch aluminum material. Mount the four switches in a row for easy manipulation. Hook up the motor according to the schematic in Fig 1. Make sure all the switches are open and apply power. Close S1; the motor should step once. Open S1. Close S2 and the motor should step again in the same direction. If it doesn't, the motor is not hooked up correctly. A piece of masking tape around the shaft makes it easier to tell which direction the motor is turning. If you now have two steps in the same direction, open S2 and close S3. Once again another step in the same direction should take place. Now open S3 and close S4 and you should get your final step. Now start the whole sequence again, only this time start with S4 to S3, etc. The motor should rotate in the opposite direction. If it does, you are on the road to implementing your motor. The four switches are the equivalent of four switching transistors. S1 is the equivalent of a binary 1, S2 a binary 2, S3 a binary 4 and S4 a binary 8.

Now, for the fun of it, let's dual phase this critter. If you energize two coils simultaneously you get maximum torque for a given motor. Once again all switches open and power on. Close S1 and S2 at the same time. You should get one step. Open S1 and S2. Closing S2 and S3 should give another step in the same direction. Open S2 and S3. Close S3 and S4, another step same direction. Open S3 and S4. Now close S4 and S1 and you should get one more step in the original direction, completing the cycle. If you want to rotate in the opposite direction using dual phase, S4 and S1 would be closed first, then S4 and S3, and so on, to complete the cycle. The binary codes in the first instance are 3, 6, 12 and 9; and 9, 12, 6 and 3 in the second. In each case, two bits of the half byte or nibble are high, and each bit energizes a motor coil.

Now that we have made the motor rotate manually we can move on to making it do so electronically. In place of each switch we will put an NPN transistor. The driver and motor schematic, Fig 2, shows how these devices are connected. If we output a hex 3 from the port, pins 2 and 3 go high. Pins 18 and 3 of the 74LS244 go high. The transistor bases go high and the transistor turns on, energizing coils L1 and L2, and the motor steps one step. The 74LS244 is an octal tri-state buffer and is there to protect your parallel port. From here on we will just consider it as transparent since a high from the port gives a high on the transistor base. After a delay to give the motor time to step, the port output is reset to zero, or all low. We need this slight delay as computers are just a whole bunch faster than stepper motors. Next, the computer will output a hex 6 and the bases of Q2 and Q3 go high, energizing the next pair of coils, L2 and L3. Next a hex 12 is output and L3 and L4 are actuated, then a hex 9 to energize the last pair of coils L4 and L1. A zero is output after each step to de-energize the coils to prepare for the next step. To rotate in the opposite direction, the order in which the coils are energized is reversed. L4 and L1, L4 and L3, L3 and L2, L2 and L1, just as was done in the case of manual rotation.

Two motors can be run from one port using this configuration. The hexidecimal numbers for bits D4 through D7 will be different, but the effect will be the same. The output codes using the upper four bits are 48, 96, 192 and 144. And 144, 192, 96 and 48 to reverse the motor direction. Do not omit the flywheel diodes D1 to D4. They are there to protect your transistors!

Some circuits have been published using the parallel port to directly drive external circuits. It's not a good idea, which is why the 74LS244 is in this one. This chip sells for less than a dollar, parallel ports cost more. Not to mention the hassle of replacing the whole port.

Now let's see how relays can be actuated, and other interesting things, using this port.

If two CD4514s, 4-bit latch/4 to 16 line decoders are connected to a parallel port, 30 relays can be controlled. The relay control schematic, Fig 3, shows the connections. Only two relays are shown as they are all the same. The chip does all the decoding for you. If for example a 15 is caused to be output from the port, pins 2, 3, 4 and 5 go high. On the CD4514, pin 15 (S15) goes high. Through the base resistor, the transistor connected to S15 turns on and the relay in its collector circuit calls up. It's that simple. It will stay called up until we send a hex 0 to clear the latch and turn off the relay. Once again we build in a slight delay to give the device time to actuate.

The matching device shown in Fig 4 is for illustration. Figs 3 and 4 are identical, except for the type of relays being used and the manner in which they are called up. RY1 through RY12 are magnetic latching relays. But since each relay has two separate coils, we are only dealing with six relay units. If, for instance, we wish to switch in the entire inductor, we would latch up RY11. When we want to change bands, RY12 needs to be called up to clear the latch. Then the desired new tap position can be connected. New latching relays are expensive. Fortunately, Fair Radio Sales sells an open frame model at a reasonable price. It comes in a plastic case soldered to an octal plug. If you don't want to mount them in old tube sockets, remove six screws, snip off eight wires and you have a relay which can be mounted anywhere. The contacts are gold plated and should be good for at least 10 amps.

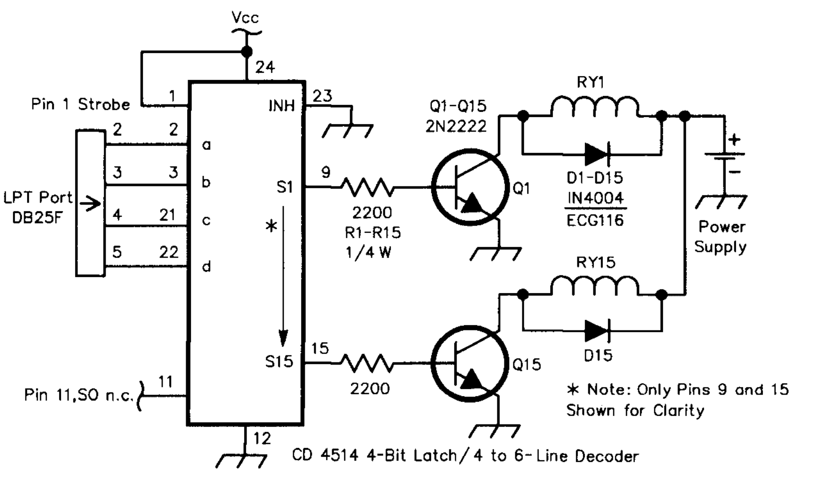


Fig 3

As this project evolved, the plans were to use two parallel ports. But at about the three quarter point, it was decided to try to do it all on one port. In this configuration, three stepper motors can be driven and still have 18 relays left to switch whatever you want switched. But adding another port is still another option if you need more headroom. All the data is here if you choose to do so.

We are now ready to mention a method stepper-motor control using relays instead of switching transistors. But first a few comments about steppers in general. Stepper motors come in a bewildering variety of configurations. Some step 15 degrees for 24 steps per revolution. Some step 7 degrees 30 minutes for 48 steps per revolution. One of the yet-to-be-used prizes in my collection steps 0.225 degrees for 1600 steps per rotation. We don't need anything that exotic. A 1.8-degree motor will give 200 steps, which is plenty of resolution for our purposes. These motors are common in old computer printers, not necessarily PC compatible. The best ones seen so far are the Honeywell/NEC printers. They have two beautiful STEP-SYN motors by the Sanyo-Denki Co. These are high-quality motors, made to last. A commercial-quality printer is preferred as they have better motors.

Until you have some experience, stay away from 4-wire motors and stick with either 5- or 6-wire models. Research also says 4-wires can be run, switched in quadrature, whereby two ends of the coils are taken to the motor supply voltage through ballast resistors. The remaining two coil ends are pulsed with one coil pulsed leading by 90 degrees which determines the direction of rotation. It should be an interesting programming challenge, but not for now as the others are much easier to use.

On the motor test schematic where we manually operated the motors, the wire color scheme on the bulk of my motors has been included to help when you go to ohm out your motor windings and see how to make the hookup. When you get your motor(s), consider yourself lucky if you get any kind of diagram. You will most likely have to get out the ohmmeter, find the center-taps and then experimentally find the proper phasing.

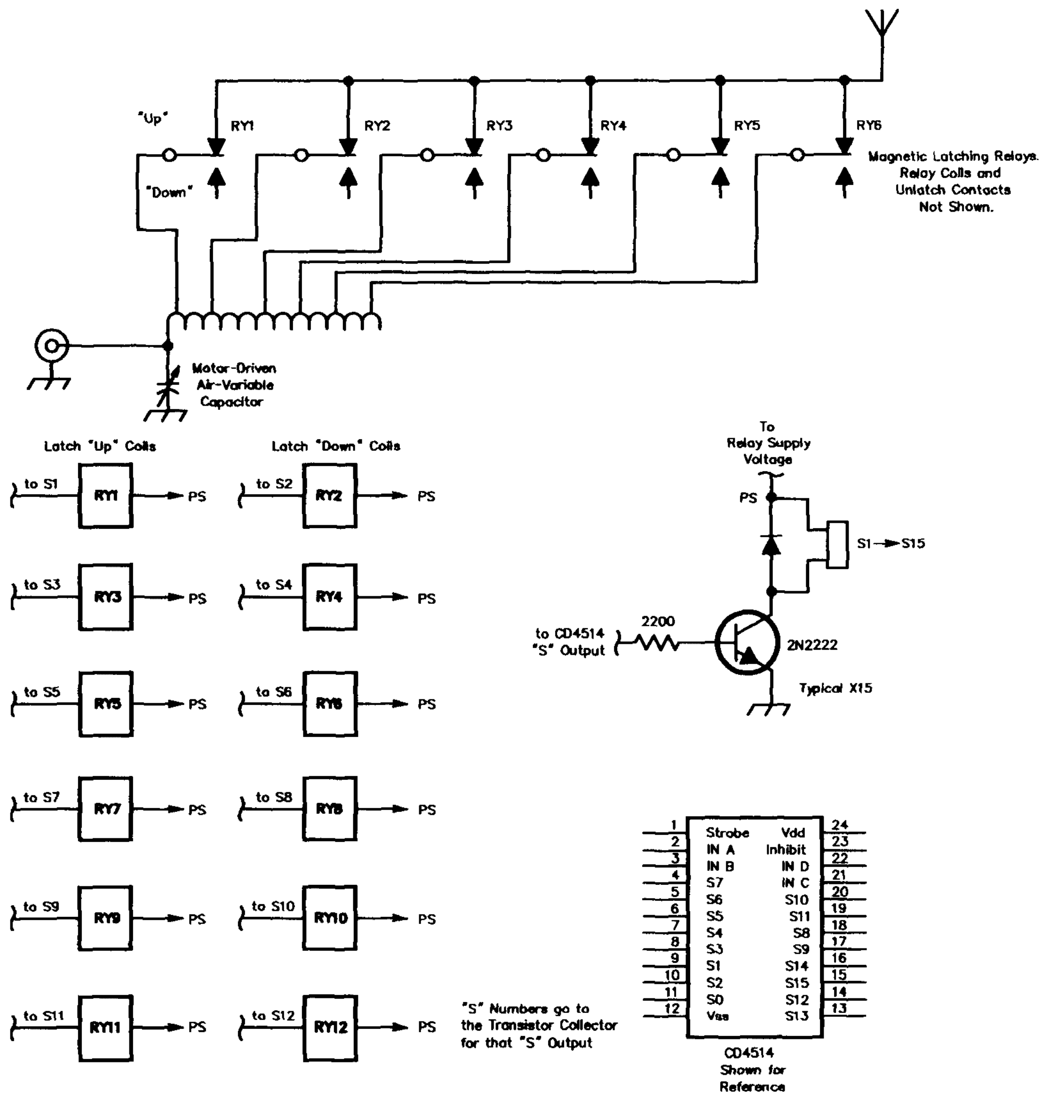


Fig 4

You may be wondering why a different method of driving the motors is being considered—after building up a motor test board, manually operating the motor, then making the motor rotate with transistor switches. For one thing, this method is simpler—and you can drive six motors with eight bits, but only two simultaneously. Plus all this can be done on one port.

We are essentially hardwiring the code into the relays. Both relay commons go to a particular pair of motor coils. The NO contacts reconnected to ground. The first relay is connected to L1 and L2. The second to L2 and L3 and so on. Now all we have to do is have the computer call up the right relay in the right sequence.

It is deceptively simple; you wire the motor switching into 4-DPDT relays. Fig 5 shows the hookup. The output codes to the CD4514 are 1, 2, 3, 4 or 4, 3, 2, 1 for a first motor. A second motor would be 5, 6, 7, 8 and 8, 7, 6, 5. It was first thought that this method might work, but the relays would soon hit a wall and not allow any faster rotation. This has *not* proved to be true. Furthermore, one of the main reasons for all this work on stepper motors is that you can throttle them down to dead slow and stop that air variable right at the point of minimum SWR.

If you cannot find subminiature DPDT relays, try the Potter & Brumfield KHP17D11 series. These are small relays. Make sure you get a 24-V dc coil as the 110-V ac coil model is identical in appearance. Fortunately, they are usually well marked. This relay is an industry standard and has been around for decades. It shows up in surplus industrial control devices, timers, sequencers, etc.

Now we look at the software. It is written *Micro-Soft Quick Basic* which allows creation of .EXE files after your software is debugged and running.

There are two versions, nearly identical. The first version assumes the motor is to be driven with transistors; the second assumes relays.

Lines 10 through 900 establish the main menu, and then waits for the keystroke for the next step. Assume 80 meters is selected and "b" is depressed. The software goes first to the 2200 subroutine and sets all the latching relays to off. This is necessary as we are not providing read-back status to the computer to know which device is latched up on which band. After this subroutine is done, the second relay is set. A delay function times out and the CD4514 is set to zero out, all outputs low, no relays called up.

If "t" is depressed the program goes to the 3000 subroutine and asks for motor direction. "f" for forward "r" for reverse. Upon receipt of the keystroke, the motor starts to rotate clockwise or counter clockwise, and will continue to do so until "s"is depressed. When "s" is depressed the motor stops and the software returns to the main menu. To exit to DOS press "q."

You will note there are no FOR NEXT loops in the motor driver routine. A severe aversion to these loops was developed during testing, which is probably why I think the motors run smoother without them.

Finally, line 3200. This establishes how fast the motor is going to turn. The bigger the number the slower it turns. If it is too short it may not work either. So experiment with your computer to see what speed develops for a given setting, and adjust accordingly. On the old XT Laptop, 1000 is a slow rotation.

Just in case latching relays are not available, 4PDT contactors can be used in their place. BG Micro has these relays at very good prices; there are no worries about contact current ratings here. See the circuit details in Fig 7. This makes these relays self-latching. Just one latching relay here will make it a lot easier to clear the latches. Otherwise the quick-and-dirty way is to shut off the relay supply. The latching relay can shut off the power, clear the latched contacts, and then turn the power back on for the next relay latch. Of course the computer software will have to be modified slightly. TUNER.BAS does just this function. Of course the necessary wiring changes will have to be made to the relay rack and coils taps.

The motor driver which uses small relays to drive the motors is TUNER2.BAS and is nearly identical to the original. Note here if the top four bits were to be utilized, the variables would be X1=16, X2=32, X3=48 and X4=64. This can be used to test motors. It is only slightly different than the driver used in TUNER2.BAS.

Continued on page 10.

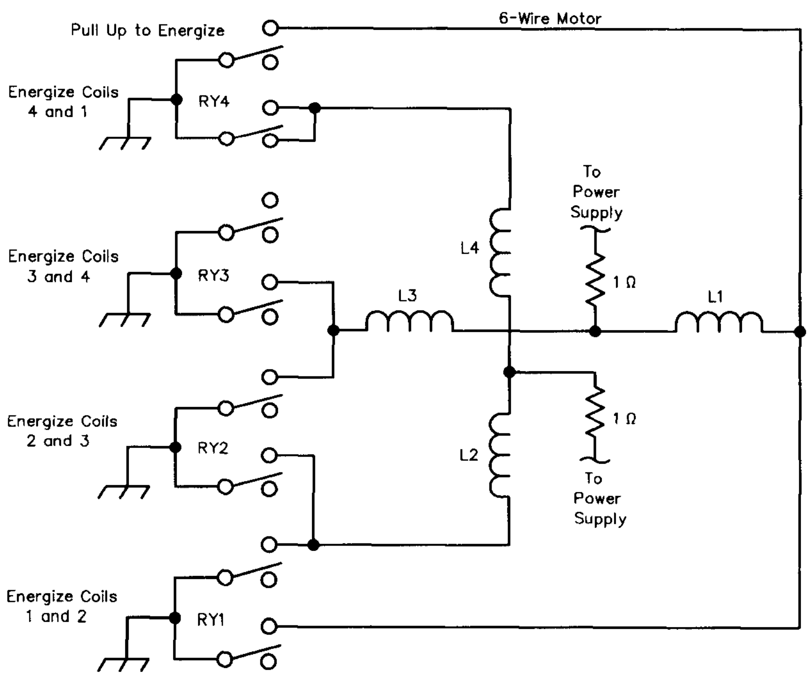


Fig 5

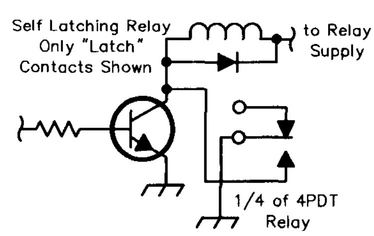


Fig 6

```
0 REM:TUNER.BAS
20 REM:to control remote devices via Parallel Port
25 REM NOTE**LPT1= 888***LPT2=632
30 REM*****WRITTEN BY RON PIERCE NI0L- JUNE 1996 ********
50 SCREEN 0
70 COLOR 15, 1, 1:
80 CLS
90 LOCATE 4, 16: PRINT "REMOTE CONTROL MENU "
95 LOCATE 5, 10: PRINT "**************************************************************"
100 LOCATE 6, 16: PRINT "SELECT BAND."
110 LOCATE 7, 16: PRINT "PRESS a FOR 160 METERS,b FOR 80 METERS"
120 LOCATE 8, 16: PRINT "PRESS c FOR 40 METERS,d FOR 20 METERS"
130 LOCATE 9, 16: PRINT "PRESS e FOR 15 METERS,f FOR 10 METERS"
140 LOCATE 10, 16: PRINT "PRESS t TO TUNE CAPICATOR"
150 LOCATE 11, 16: PRINT "PRESS q TO EXIT TO DOS"
190 LOCATE 12, 10: PRINT
"***********************************************************"
200 a$ = INKEY$: IF a$ = "" THEN 200
220 IF a$ = "a" THEN GOSUB 2200: OUT 888, 1: GOSUB 2500
240 IF a$ = "b" THEN GOSUB 2200: OUT 888, 3: GOSUB 2500
260 IF a$ = "c" THEN GOSUB 2200: OUT 888, 5: GOSUB 2500
280 IF a$ = "d" THEN GOSUB 2200: OUT 888, 7: GOSUB 2500
300 IF a$ = "e" THEN GOSUB 2200: OUT 888, 9: GOSUB 2500
360 IF a$ = "f" THEN GOSUB 2200: OUT 888, 11: GOSUB 2500
370 IF a$ = "t" THEN GOTO 3000
900 IF a$ = "q" THEN END
2110 GOTO 200
2200 FOR x = 2 TO 12 STEP 2: OUT 888, (x): FOR a = 0 TO 500: NEXT a: NEXT x: RETURN
2210 REM** x loop sets all latch relays to open, s loop is a delay to give
2215 REM****   relays time to actuate
2500 FOR s = 0 TO 250: NEXT s: OUT 888, 0: RETURN
3000 REM *** MOTOR DRIVE ROUTINE
3110 N0 = 0: N1 = 1: N2 = 2: N3 = 3: N4 = 4
3120 N$ = ""
3130 X1 = 3
3140 X2 = 6
3150 X3 = 12
3160 X4 = 9
3200 DELAY = 50
3210 LOCATE 13, 16: INPUT " PRESS f FOR CW , r FOR REVERSE ROTATION , s TO STOP"; a$
3220 IF a$ = "f" THEN 3300
3230 IF a$ = "r" THEN 5000
3300 OUT 888, X1: LOCATE 15, 16: PRINT " MOTOR IS RUNNING CLOCKWISE"
3310 GOSUB 6000
3340 OUT 888, X2: GOSUB 6000
3380 OUT 888, X3: GOSUB 6000
3420 OUT 888, X4: GOSUB 6000
3500 IF INKEY$ = "s" THEN CLS : GOTO 50
3510 IF INKEY$ = N$ THEN 3300
5000 OUT 888, X4: LOCATE 15, 16: PRINT " MOTOR IS RUNNING COUNTER CLOCKWISE"
5010 GOSUB 6000
5040 OUT 888, X3: GOSUB 6000
5080 OUT 888, X2: GOSUB 6000
5130 OUT 888, X1: GOSUB 6000
5170 IF INKEY$ = "s" THEN CLS : GOTO 50
5180 IF INKEY$ = "" THEN 5000
6000 D = N0
6010 D = D + N1
6020 IF D < DELAY THEN 6010
6050 RETURN
```

```
0 REM:TUNER2.BAS
5 REM**TO REMOTELY CONTROL DEVICES VIA PARALLEL PORT
10 REM **MOTOR DRIVER SECTION CALLS UP RELAYS WHICH STEP MOTOR(s)
20 REM**OTHERWISE IDENTICAL TO TUNER.BAS
25 REM NOTE**LPT1= 888***LPT2=632
30 REM*****WRITTEN BY RON PIERCE NI0L— JUNE 1996 ********
50 SCREEN 0
70 COLOR 15, 1, 1:
80 CLS
90 LOCATE 4, 16: PRINT "REMOTE CONTROL MENU "
95 LOCATE 5, 10: PRINT "*************************************************************"
100 LOCATE 6, 16: PRINT "SELECT BAND."
110 LOCATE 7, 16: PRINT "PRESS a FOR 160 METERS,b FOR 80 METERS"
120 LOCATE 8, 16: PRINT "PRESS c FOR 40 METERS,d FOR 20 METERS"
130 LOCATE 9, 16: PRINT "PRESS e FOR 15 METERS,f FOR 10 METERS"
140 LOCATE 10, 16: PRINT "PRESS t TO TUNE CAPICATOR"
150 LOCATE 11, 16: PRINT "PRESS q TO EXIT TO DOS"
190 LOCATE 12, 10: PRINT
"*************************************************************"
200 a$ = INKEY$: IF a$ = "" THEN 200
220 IF a$ = "a" THEN GOSUB 2200: OUT 888, 1: GOSUB 2500
240 IF a$ = "b" THEN GOSUB 2200: OUT 888, 3: GOSUB 2500
260 IF a$ = "c" THEN GOSUB 2200: OUT 888, 5: GOSUB 2500
280 IF a$ = "d" THEN GOSUB 2200: OUT 888, 7: GOSUB 2500
300 IF a$ = "e" THEN GOSUB 2200: OUT 888, 9: GOSUB 2500
360 IF a$ = "f" THEN GOSUB 2200: OUT 888, 11: GOSUB 2500
370 IF a$ = "t" THEN GOTO 3000
900 IF a$ = "q" THEN END
2110 GOTO 200
2200 FOR x = 2 TO 12 STEP 2: OUT 888, (x): FOR a = 0 TO 500: NEXT a: NEXT x: RETURN
2210 REM** x loop sets all latch relays to open, s loop is a delay to give
2215 REM****   relays time to actuate
2500 FOR s = 0 TO 250: NEXT s: OUT 888, 0: RETURN
3000 REM *** MOTOR DRIVE ROUTINE
3110 N0 = 0: N1 = 1: N2 = 2: N3 = 3: N4 = 4
3120 N$ = ""
3130 X1 = 1
3140 X2 = 2
3150 X3 = 3
3160 X4 = 4
3200 DELAY = 50
3210 LOCATE 13, 16: INPUT " PRESS f FOR CW , r FOR REVERSE ROTATION , s TO STOP"; a$
3220 IF a$ = "f" THEN 3300
3230 IF a$ = "r" THEN 5000
3300 OUT 888, X1: LOCATE 15, 16: PRINT " MOTOR IS RUNNING CLOCKWISE"
3310 GOSUB 6000
3340 OUT 888, X2: GOSUB 6000
3380 OUT 888, X3: GOSUB 6000
3420 OUT 888, X4: GOSUB 6000
3500 IF INKEY$ = "s" THEN CLS : GOTO 50
3510 IF INKEY$ = N$ THEN 3300
5000 OUT 888, X4: LOCATE 15, 16: PRINT " MOTOR IS RUNNING COUNTER CLOCKWISE"
5010 GOSUB 6000
5040 OUT 888, X3: GOSUB 6000
5080 OUT 888, X2: GOSUB 6000
5130 OUT 888, X1: GOSUB 6000
5170 IF INKEY$ = "s" THEN CLS : GOTO 50
5180 IF INKEY$ = "" THEN 5000
6000 D = N0
6010 D = D + N1
6020 IF D < DELAY THEN 6010
6050 RETURN
```

Summary

We have now seen how to call up relays, either latching or conventional. And we have seen how to make a stepper motor rotate using two different methods. If you are a beginner and into computers, give it a try. This is a chance to make your computer do what you want it to do, not what someone else programmed it to do. This projects covers a lot of territory and assumes some prerequisites. Not the least of which are power supplies for the motors/relays. If you don't want your project to look like a rats nest, which mine does, Radio Shack sells wire-management gizmos to make things look neater.

A large percentage of the parts used in this project came from salvaged equipment. The relays came from industrial timers and sequencers used for burn-in testing. The switching transistors came from old computer printers, some of which have 30 to 40 of these in the range of IC max 8 amps. The CD4514s and 74LS244 sell for less than a dollar from BG Micro.

If you are not a beginner and want to delve deeper into full bipolar drives, see *Electronic Motors and Control Techniques* by Gottlieb. This book has the best section on stepper motors that I've encountered so far. It covers the techniques used here and progresses on through full-bipolar drives. Shift registers and switching transistors are used in tandem with a digital input to make the motors go. All the motor coils are used, but the drive electronics are consequently more complex.

If it seems like stepper motor-driver integrated circuits have been avoided, your right. You have to go buy these devices, which is anathema to an old parts collector like this old ham.

Finally, I am not holding myself out as an expert on stepper motors. Neither am I a computer programmer, and it probably shows, being self taught. But it sure is a lot of fun and continues to be so. I will be glad to try to answer questions via email.

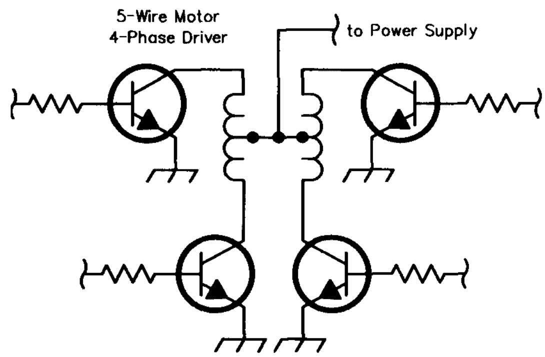


Fig 7

References

DELTROL 105ML Magnetic Latching Relay, Fair Radio Sales, Lima, OH.

Miscellaneous parts: B.G. Micro, Inc, Dallas, TX 1-800-276-2206.

DeMaw, Doug, W1FB, "The Half-Delta Loop: A Critical Analysis and Practical Deployment," *QST*, September 1982.

DeMaw, Doug, W1FB, "Antenna Matching, Remotely—Some Thoughts," *QST*, July 1982.

Controlling the World with Your PC, Paul Bergsman, 1994, HighText Publications, Inc, PO Box 1489, Solana Beach, CA 92075.

Antenna and Techniques for Low-Band Dxing, John Devoldere, ON4UN. ARRL.

Nuts 'n Volts Magazine, July and August 1995. Two-article series on Stepper Motors by John Iovine.

Solid State Motor Controls, John Kuechen. 1978, TAB Books.

Mechanical Devices for the Electronic Experimenter, Britt Rorabaugh. 1995, TAB Books/McGraw-Hill.

Electric Motors and Control Techniques, Irving M. Gottlieb. 1994, TAB Books/McGraw-Hill.

Interfacing to the IBM Personal Computer, Lewis C. Eggebrecht. SAMS.

Personal Computer Interfaces, Michael Hordeski. 1995, McGraw-Hill

The 8088 Project Book, Robert Grossblatt. 1989, TAB/McGraw-Hill.

Herbach & Rademan: surplus dealer: 1-800-848-8001.

Mobile Robots Inspiration to Implementation, Jones & Flynn. 1993, A K Peters Ltd, Wellesley, MA 02181. Excellent sections on motors and interfacing motors and the appendices are worth the price of the book.

PARALLEL.ARC downloadable from Circuit Cellar BBS: 860-871-1988 or 860-871-0549. Excellent parallel-port reference minituae.

The Secret Guide to Computers, Russ Walter, 617-666-2666.

ISDN: The New Legal Limit

ISDN connections to the Internet—improved connection speeds over analog modems.

By Richard Parry, W9IF

A Solution Waiting for a Problem

If you ask a QRP amateur-radio operator why he or she uses low power, the answer might be "the thrill and the challenge." Ask a ham running 1500 W "Why run high power?" and he or she may tell you that life is too short to be wasted with hit-and-miss attempts at communication. If you ask them about connect speeds to the Internet, however, I doubt you will get disagreement—the faster the better!

For most of the 1970s, 300 baud was the defacto standard for those telecommuting, although the word *telecommuting* did not exist at the time. Advances in technology and a larger market fueled the jump to 1200 baud, and then to 2400, which at the time was considered the limit to telephone data rates. Before jumping to 9600 we had to shed the practice of using the terms *baud* and *bit* interchangeably, because 9600 bps, still uses a *baud rate* of 2400. The heyday of 9600-bps modems was short-lived; almost overnight came the 14.4-kbps modems. Additional standards were developed, and again the speed limit increased to the present 28.8-kbps standard. Some modems push the envelope a little further, to 33 kbps and some claim faster rates, but that is a *throughput* rate from compression schemes rather than real physical-layer data rates.

If you're wondering what the next step is, it's already here. It is called Integrated Services Data Network (ISDN). ISDN has been around over a decade, but at first ISDN was a solution waiting for a problem. Without a doubt, the Internet, and more importantly, the World Wide Web with copious text and color graphics, has created the problem that ISDN was waiting to solve, albeit, not the problem that Ma Bell had envisioned solving so many years ago.

This article describes my experiences while implementing an ISDN connection at my home. It contains numerous references for future exploration. Lastly, I hope to dispel much of the stigma that has plagued widespread use of this technology. ISDN is here now, and it works!

28.8-kbps—It's the Law!

One might be tempted to trivialize the ISDN solution. After all, in a relatively short time, we have gone from 300 baud to 28.8 kbps. Ingenuity and know how will provide 56 kbps and

13842 Deergrass Ct
Poway, CA 92064-2276
e-mail **rparry@qualcomm.com**

even faster speeds very soon. Not so! Granted, we have developed new technologies that continue to amaze us, but they all obey the speed limit, the law. That law, or limit, is the Shannon-Hartley Law:

$$C = B \log_2 (1 + S/N) \text{ bps} \qquad \text{Eq 1}$$

Where C is the information rate, in bps. B is the line bandwidth, in hertz. S is the signal power, in watts, and N is the random-noise power, in watts. If you assume the bandwidth of the PSTN (Public Switched Telephone Network) is 3 kHz, and a typical signal-to-noise ratio is 20 dB, then C is 19.9 kbps—lower than the present 28.8-kbps standard. To get higher speeds you need either greater bandwidth, which is not a real option, or higher signal-to-noise ratios, also a difficult barrier to break. To go faster, you must leave the analog world and go 100% digital, which is exactly what ISDN does.

I Still Don't Know

It's a standard joke that ISDN stands for I Still Don't Know; unfortunately some people still believe ISDN is unfathomable. Nothing could be further from the truth. For many years I've been reading about ISDN applications. The writings all too often include nightmare stories about bad experiences that the authors had while implementing ISDN. Most of the stories start with, "When I called the phone company and asked for ISDN service, they said, What's that?" These stories prevented me from exploring further until recently.

Once I decided to try ISDN for myself, I was prepared to spend a whole day, if necessary, finding the right phone number and person. In my case, and I think in most cases now, the widespread use of ISDN makes getting service pretty easy. I was pleasantly surprised to find that Pacific Bell (my local telephone company) had a phone number set up specifically for ISDN orders, with a staff of knowledgeable people.

Ordering the service went quickly. The operator spent a good deal of time explaining rates and installation charges, and possible changes in rates that were impending. About the only technical question was, "What type of router or terminal adapter was I going to use?"

Shortly after I ordered the service, it was installed. No access to my home or additional wiring was necessary. It is pretty incredible to think that ISDN can use plain, old twisted-pair wires to send 128 kbps, the maximum ISDN rate. Since I was home at the time, the installer was good enough to make sure the connection was good right up to the wall socket, rather than just up to the side of the house. He did a quick BER (Bit Error Rate) test, gave me the new phone numbers for the service, the type of electronic switch at the local central office (this information is referred to as the *SPIDs*), and I was ready to hit the information superhighway at the new legal limit.

Terminal Adapters

A terminal adapter (TA) is a device that replaces your modem in ISDN implementations. As the name ISDN implies, the connection is digital—all digital. There is no modulation or demodulation. The data is 100% digital from your computer, through the TA, to the local central office, throughout the vastness of the PSTN to its final destination (presumably another computer). Because much of the world is still analog, however, manufacturers also make ISDN modems. These devices offer the best of both worlds. The ISDN modem can act as a 100% digital terminal adapter and a traditional analog modem to provide the user the ability to connect to services that do not offer ISDN connections.

A TA can be either an external standalone unit or a card that fits inside your computer, just as your present modem can be external or internal. The external style allows you to see the LEDs, which provide useful troubleshooting information and a warm fuzzy feeling that all is working well. Fig 1 shows the present standard analog modem connection. Fig 2 shows the same system using a TA. Note that they are virtually identical.

Consider the maximum allowable speed of your serial port when deciding between an internal and external TA. If your serial communications port cannot support 128 kbps, your computer will be a bottleneck and you won't get

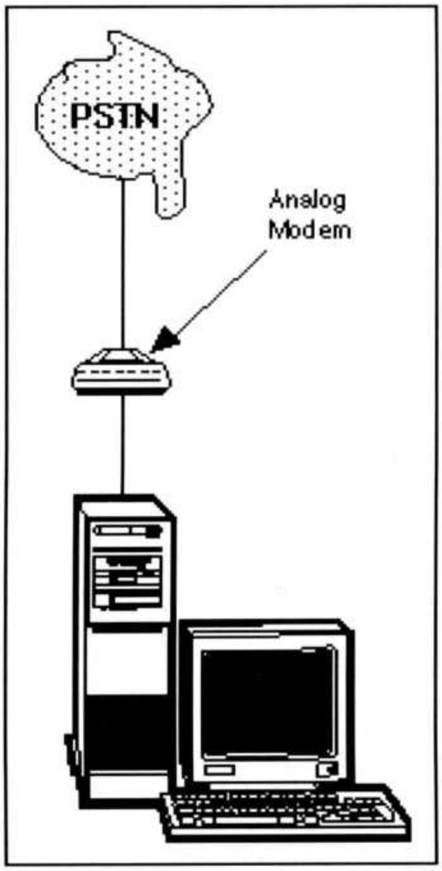


Fig 1—Here we see a traditional connection to the telephone company central office. A standard analog modem changes digital signals to audio signals.

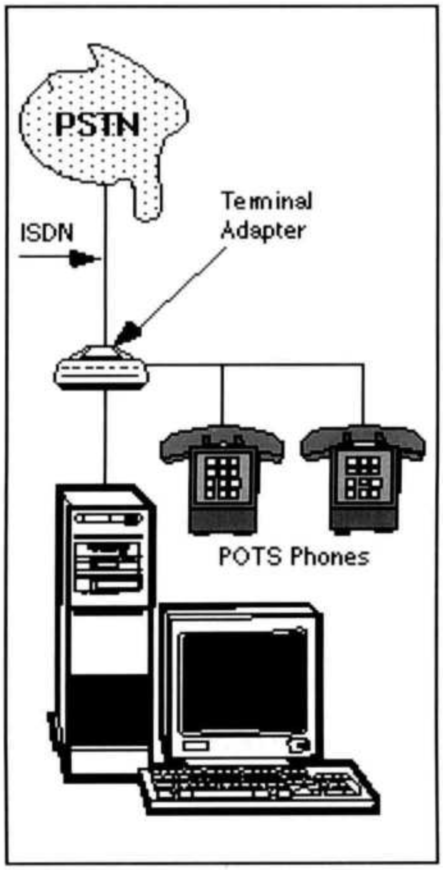


Fig 2—In this configuration, the terminal adapter (TA) takes the place of the modem and very little else changes—except the connect speed jumps to 128 kbps. Terminal adapters, and some routers provide the ability to connect additional analog phones.

the full advantage of ISDN. If you have a 486 or higher speed computer with a 16650 serial communications chip, you should have no trouble. On the other hand, an internal TA card bypasses the serial port, which should obviate any limitations of the serial port.

Routers

Many families own two cars, two televisions and now two computers. Those with more than one computer, must purchase an additional TA for each computer. In addition, you must acquire additional phone lines for each TA, assuming you want to use the computers simultaneously. Clearly, this is not cost effective. The solution is a router.

A router is a more complex device than a TA, so a router requires a little more work and knowledge to set up. When installing a router, it's best to first connect your computers on a local network. This requires additional hardware, besides the router.

Fig 3 shows a block diagram of a network connected to the Internet via a router. The computers are connected on a network using Ethernet Network Interface Cards (NIC) rather than serial ports. The router then serves as the interface between the local network and the Internet. Note that this arrangement lets the computers communicate with each other even when there is no connection to the Internet. This is not automatic. Additional software, such as Telnet and ftp daemons, must be installed to support this feature.

You must also get IP (Internet Protocol) addresses for each of the computers, most likely from your local ISP (Internet Service Provider) for an additional charge. If you have several machines, you may even receive a small block of numbers to make a subnet, which requires you to become a system administrator. It is a lot of work and requires a fair amount of expertise, but is a great way to learn about networks.

The router solution is a significant expense. A router costs $1000 to $1500; the cost of NICs to implement the network vary, and lastly, your ISP will charge for the additional IP addresses required. Having said that, it is an excellent solution if you have the need.

So What is ISDN?

I recently went to an amateur-radio club meeting that discussed the GPS (Global Positioning System). After the meeting a fellow ham said, "GPS is simple." From the user's standpoint, GPS is simple, yet GPS uses very advanced technology. There are numerous erudite books written exclusively on the subject that discuss the mathematics and physics used, the RF spectrum allocation, the spread-spectrum technique, the atomic clocks in the satellites, the ground control system, reliability and availability factors, selective availability, encryption, security issues, the satellite constellation and more. There is nothing simple about the GPS.

ISDN is also complex, but so is a television, and we don't think twice about using it. ISDN users don't need to know very much about the network's complexities. ISDN consists of two B (data) channels and one D (control) channel. This 2B+D is called the *Basic Rate Interface* (BRI). Each B channel can carry information (voice packets as well as computer data) at 64 kbps, giving a total throughput of 128 kbps when the channels are strapped together. The D channel, when used, transmits data at up to 16 kbps. It carries control and signaling information to set up and tear down the voice and data channels. When the D channel is used, the B channels revert to 56 kbps (each) to keep the total bandwidth at the 128-kbps limit.

If a call occurs while the B channels are both in use for data communication, one of the B channels is released to carry the call. The first user's throughput drops in half during this period, but returns to the higher rate when the second B channel is available again.

The separate D channel is very significant. It controls calls (eg, call setup, call forwarding, call waiting, etc) and greatly improves available bandwidth for voice calls because control signals need not ride along with the voice packets. ISDN is an *out-of-band* signaling system because the control function is separate from the voice channel. Control signals ride along with the voices in analog systems, so they are called *in-band* signaling systems.

There are three basic ISDN configurations. A *U* interface is the most common interface; it uses a single pair of wires. An *S* (sometimes called a *T*) configuration uses two pairs. The third ISDN option is a *Primary Rate Interface* (PRI), which is referred to as a 23B + D or 30 + D interface. The 23B + D is a US T1 line, which can carry 23 simultaneous calls. The 30B + D is a European E1 standard that carries up to 30 calls. Each of these lines has distance and data-rate limitations that differentiate the configurations.

From the user's standpoint, ISDN is simple, but don't take the TA apart to see what makes it tick. (It might be instructive, but it could be a humbling, experience.)

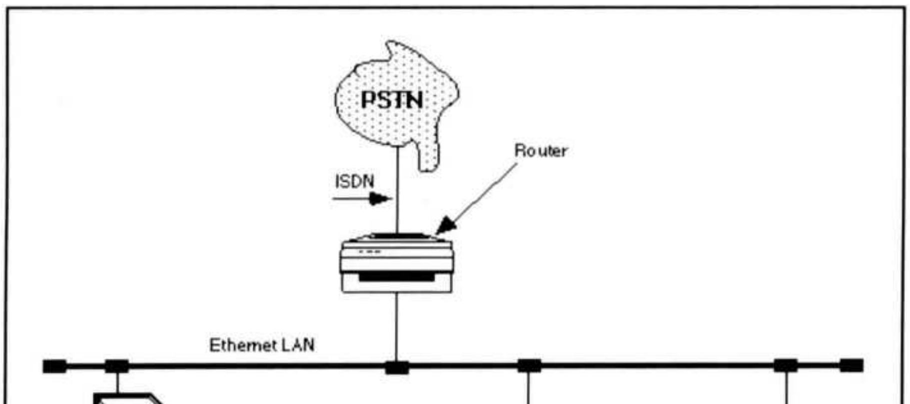


Fig 3—A router provides the interface between the ISDN and a small network of computers. This last solution, albeit a good solution for connecting multiple computers, requires significant additional costs and system administration expertise.

Side Benefits

The benefits of the faster connect speeds are clear: fewer long waits for megabyte files to download and less waiting for Web graphics to appear. However, there are a few less obvious fringe benefits that come with ISDN.

First, some TAs have two ports for POTS (Plain Old Telephone Service) analog phones, answering and/or fax machines. This means when the TA is idle, your single ISDN line can serve as two separate phone lines. (Try to forget that it is a single twisted-pair cable.) Even though you may pay a slight premium for an ISDN line, you are essentially getting two phone lines with two separate phone numbers. This is a big plus because it may even save you money if you drop one line that you presently use. Even though an incoming or outgoing voice call takes a channel and cuts your net-surfing throughput to about 56 kbps, that is still twice the present 28.8-kbps standard.

For those who want to use a single ISDN line for all telephone communication, remember that the TA or router requires conventional power. Therefore, you will lose *all* use of your phones in the event of a power failure. Electric power is highly reliable, and most hams are prepared for emergency communication, but this risk is something to consider nevertheless.

Second, the speed of connects and disconnects with the central office (CO) is very fast. When a normal analog modem makes a call, it picks up the phone, waits a few seconds for the dial tone, listens, and then dials the phone number. The modem at the other end must answer, synchronize to the tones and many other magical things occur during the call setup. When all of the handshaking is complete, 20 to 30 seconds have passed. ISDN connections take approximately 12 seconds. (Some users report connections as quick as 3 seconds.) The disconnect is even faster, typically 3 to 4 seconds. The difference is only a few seconds, but I do find that the connect time has been reduced sufficiently that I find checking e-mail or doing quick research more palatable.

Okay, How Much $$?

It would be a mistake to provide exact costs for an ISDN implementation without a warning that costs vary widely across the country, and they continue to change. My personal example (with Pacific Bell Telephone) may give you an idea of what to expect, but use it *only* as a starting point.

Telephone costs

Monthly Service Fee $24.50
Installation Fee $125.00 (waved with a two-year contract)
New Line Fee $34.50 to install an additional line
Usage $0.01 per minute during business hours, free otherwise

ISP and hardware costs

ISP $30 to $50 per month (varies greatly)
TA $300-600

For me, the monthly telephone fee is virtually the same as the normal analog phone line, so there was no increase in cost there. I accepted a two-year contract so the installation fee was waved. Since I had been using a telephone line dedicated to an Internet connection already, there was no new line fee. I merely switched the line from analog to ISDN. Therefore, *there was virtually no increase in cost for ISDN service.* There could be a cost *decrease* if I use the single ISDN line to obtain two phone lines and drop the original analog line. The only significant cost is the one-time cost of a TA and the service provider's monthly premium for an ISDN line.

The web is the best place to get up-to-date ISDN information. There is a plethora of pages to choose from, but if you have time for only one, go to **http://www.icus.com/kegel/**. There you will find most of what you need to know, and numerous helpful links. When you finally make the decision to break the speed limit, try to get the phone number of the ISDN department for your carrier and tell them you want a ISDN Basic Rate Interface. If you want to impress them, tell them you want the 2B + D U configuration to enable you to bypass Shannon-Hartley limitations.

Conclusion

Now you know a little about ISDN technology, and the References give you other sources of information. ISDN implementation today is not nearly so difficult and confusing as it was a few years ago.

You can never have enough computer memory, disk space or speed. As these resources increase, so does the variety and demand of the applications. Multimedia, high-fidelity audio and real-time video conferencing are only now possible due to leaps in computer power. But these applications are probably not in your home yet, and ISDN is only a partial solution. For even greater connect speeds than BRI ISDN can offer, there are few options. The phone company offers a PRI ISDN; a T1 line (1.544 Mbps) with a price tag to match. More reasonable and hopeful, the near future promises cable modems: local cable companies who will provide Internet access through the existing high-bandwidth CATV infrastructure. Cable modems are now enjoying some use in Canada. Only time will tell if this is the real answer to increased speed and the ultimate goal of having the information superhighway run through your living room. I dare not venture any further guesses. I think we all will agree these are exciting times, and the future will no doubt continue to amaze us, and more often than not, prove our prognostications wrong.

About the Author

Richard Parry, W9IF, was originally licensed in New York City in 1962 as WN2BIJ. Since that time he has progressed from one obsession to another: CW, RTTY, packet, and most recently satellite communication. He comes from an amateur-radio family consisting of WB2ILP (father), KC5PVL (wife), KK5SU (son), and daughter Megan (age 10) is hard at work for her first license.

References

Angell, David, "The Ins and Outs Of ISDN," *Internet World*, March 1996, pp 78-82.

Savetz, Kevin M., "My ISDN Experience," *Internet World*, March 1996, pp 84-85.

Key ISDN Web-page URLs for information on ISDN products, ISPs and telephone companies. If your time is limited, visit the first one.

http://www.icus.com/kegel/
http://www.pacbell.com/products/business/fastrak/networking/isdn/home-isdn/index.html
http://www.alpha-tele.com/
http://www.halcyon.com/tcs/
http://isdntek.com/
http://www.psi.net/
http://www.cisco.com/
http://www.mot.com/MIMS/ISG

A Simple T/R Sequencer

Protect sensitive RF circuits from burn out by sequentially switching from transmit to receive.

By Zack Lau, W1VT

What, another T/R sequencer? Yes, I decided to make it as simple as possible, while adding +12-V outputs and reverse polarity protection. I think the reverse polarity protection is pretty nifty—who hasn't worried about hooking up batteries backwards and frying something? The 12-V outputs make it real easy to wire up my transverter designs, since they are made up of modules that run off 12 V. Hopefully, this design will spur more designers into adding reverse polarity protection.

The primary reason for using a sequencer is to protect the RF relay and the amplifiers hooked up to it. RF relays can be damaged by hot switching at high power levels. Unfortunately, what constitutes high power is rather fuzzy—I have not found any good references that adequately address this topic. Based on my experience with microwave transverters, I always put in a sequencer when switching more than half a watt. I don't bother with them in inexpensive systems running less than 100 mW.

Like the T/R sequencer designed by Chip Angle, N6CA, this sequencer uses a quad comparator to monitor the voltage on a charging or discharging capacitor.[1] I looked at using an integrator to get more uniform delay intervals, but decided to stick with the simpler circuit. An integrator can generate a nice triangular waveform, as opposed to the exponential curve generated by an R-C network. The 10-Ω resistors are used to provide hysteresis, so the outputs don't switch back and forth unnecessarily near the transition point. They are supposed to provide a little positive feedback, instead of the more common negative feedback used in other applications.

This sequencer first turns off the receiver, then activates the relay, amplifiers and finally the transmit IF outputs when switching from receive to transmit. When going back to receive, it turns off the transmit IF, amplifiers, and relay before reactivating the receiver. The idea is to introduce enough delay between these states to allow everything to settle down, and to reverse the order when switching from transmit to receive. This is especially

225 Main Street
Newington, CT 06111
email **zlau@arrl.org**

[1]Angle, Chip, N6CA, "TR Time-Delay Generator," *The ARRL Handbook for Radio Amateurs*, 1997, pp 22.53-22.56.

important with a mechanical relay which may make intermittent contacts for a few ms when switched.

A big advantage to using solid-state amplifiers is that you can switch them off during receive. Not only does this reduce the possibility of hot switching, it reduces the chance of amplified broadband noise getting into the receiver. Using a PIN diode switch to cut off transmit drive also helps to prevent hot switching—I use the transmit IF signal to control this switch.

Since I've never needed to change the polarity of the output of one of my sequencers, I decided some simplification was in order. Instead of the XOR gate Chip used, I decided to use a hard-wired switch based upon the principles of a transfer switch. To switch polarity with this design, you change a pair of resistors from horizontal to vertical, or vice versa. By labeling the blank area of the board between the resistors, identifying the resistors ought to be straightforward. The reduction in parts count ought to enhance reliability.

I've also taken advantage of the improvements in switching transistor technology. The International Rectifier P-channel IRF 9Z34 will easily switch 2 A, enough to power a 5-W GaAs FET power amplifier. Similarly, the Zetex ZTX 789 in a little TO-92 style case will actually switch a small SMA relay that draws a few hundred mA. The bonus to using more expensive PMOS/PNP parts is that the switched supplies are reverse polarity protected. I just needed to protect the comparators with a diode and use bipolar electrolytic capacitors which don't care about voltage polarity. Cheap 2N3906s are used for the RX and TX IF supplies, since they typically draw under 100 mA. You could use ZTX 789s instead of the 2N3906s, with the appropriate bias resistors, for higher current.

You may want to replace Q2 with a VN10LP N-channel FET. This will allow you to hook up the PTT line without pulling the voltage down significantly. Another advantage is that you can now hook up the PTT input to a stiff voltage source without frying Q2. But these FETs aren't as easy to get as common NPN switching transistors.

Of course, it makes sense to watch out for even newer FETs, which will undoubtedly offer better performance at lower cost. There is even a trend toward lower gate thresholds, which allow better performance at lower voltages. For instance, the IRF 7104

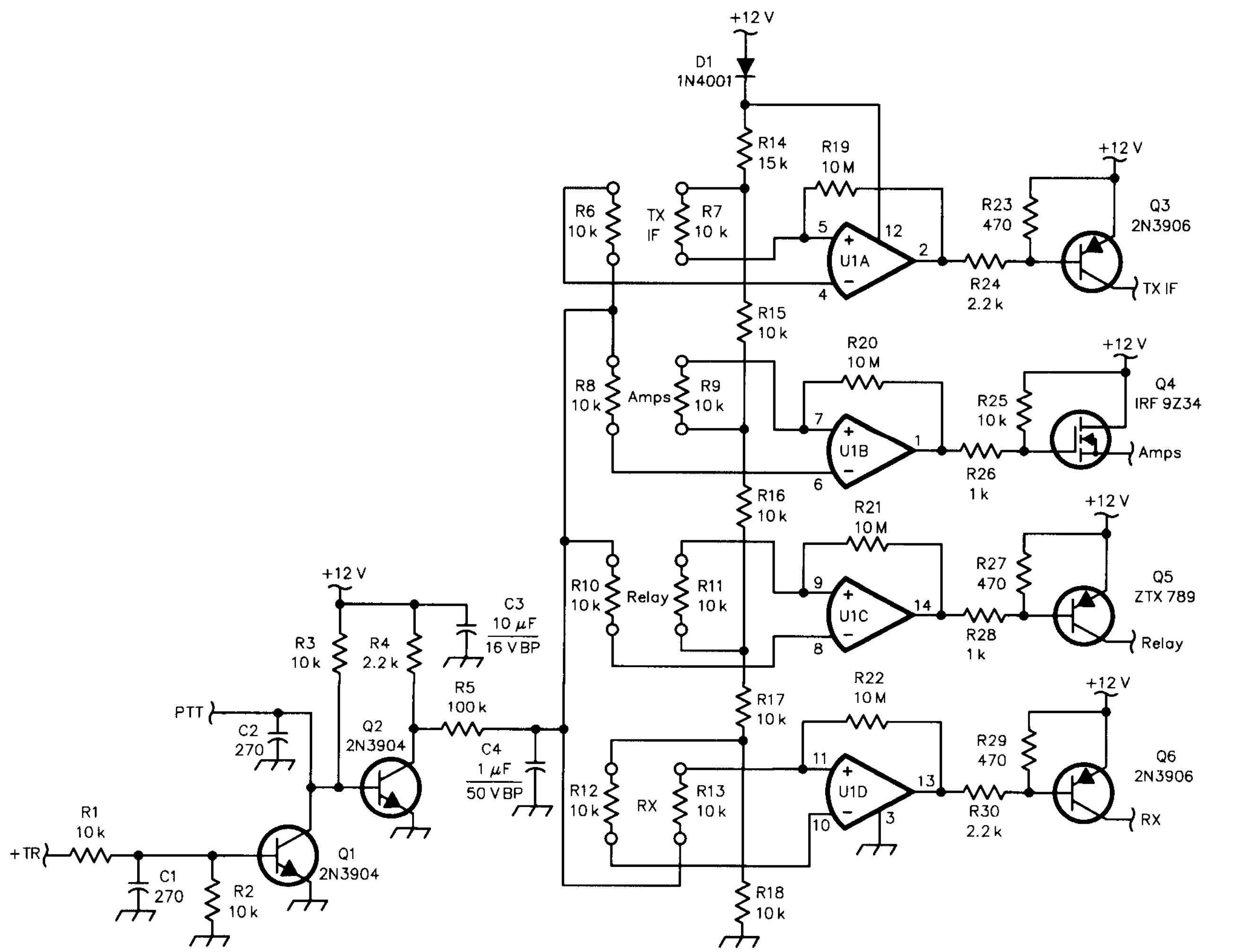


Fig 1—Schematic diagram of the transmit-receive sequencer.

drops only 154 mV when sourcing 0.57 A (V_{gs} = –5V). But, despite the marketing hype, you can do even better than "full enhancement." It only drops 117 mV with a V_{gs} of –10V.

Construction

I made the pads big enough to accommodate swaged terminals, which are a really nice way of making dc connections if you have the tooling to rivet them to a fiberglass circuit board. The board is a bit crowded—I wanted the board to fit nicely on the wall of a chassis box only two inches high. A mirror image of the etching pattern is provided—it simplifies the toner transfer process that some people use to make circuit boards. Similarly, a parts placement diagram using part values and component designations is also provided in Fig 4.

To test the board, I made a little fixture out of LEDs and dropping resistors. A separate fixture makes it easy to line up the LEDs in the proper sequence. It may be useful to slow down the sequencing by bridging the 1-µF timing capacitor with a 10-µF capacitor. This makes it easier to see the LEDs turn on and off.

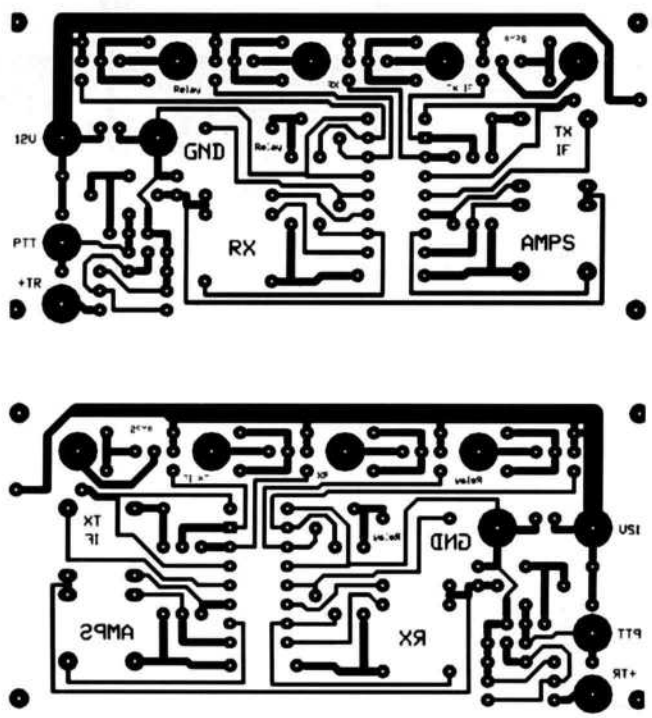


Fig 2—Etching pattern for the transmit-receive sequencer circuit board.

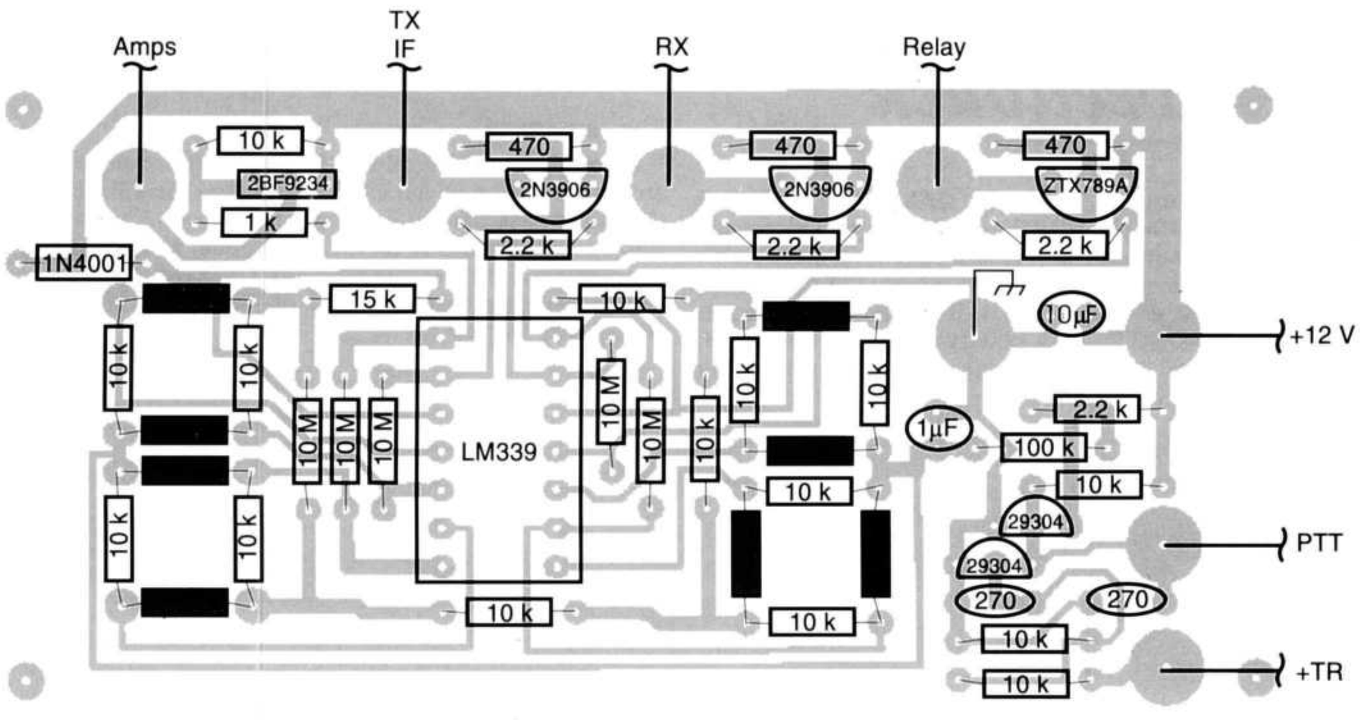


Fig 3—Parts placement diagram for the transmit-receive sequencer circuit. The shaded resistors indicate alternate positions for R6 to R13 to invert the signal sense. See text.

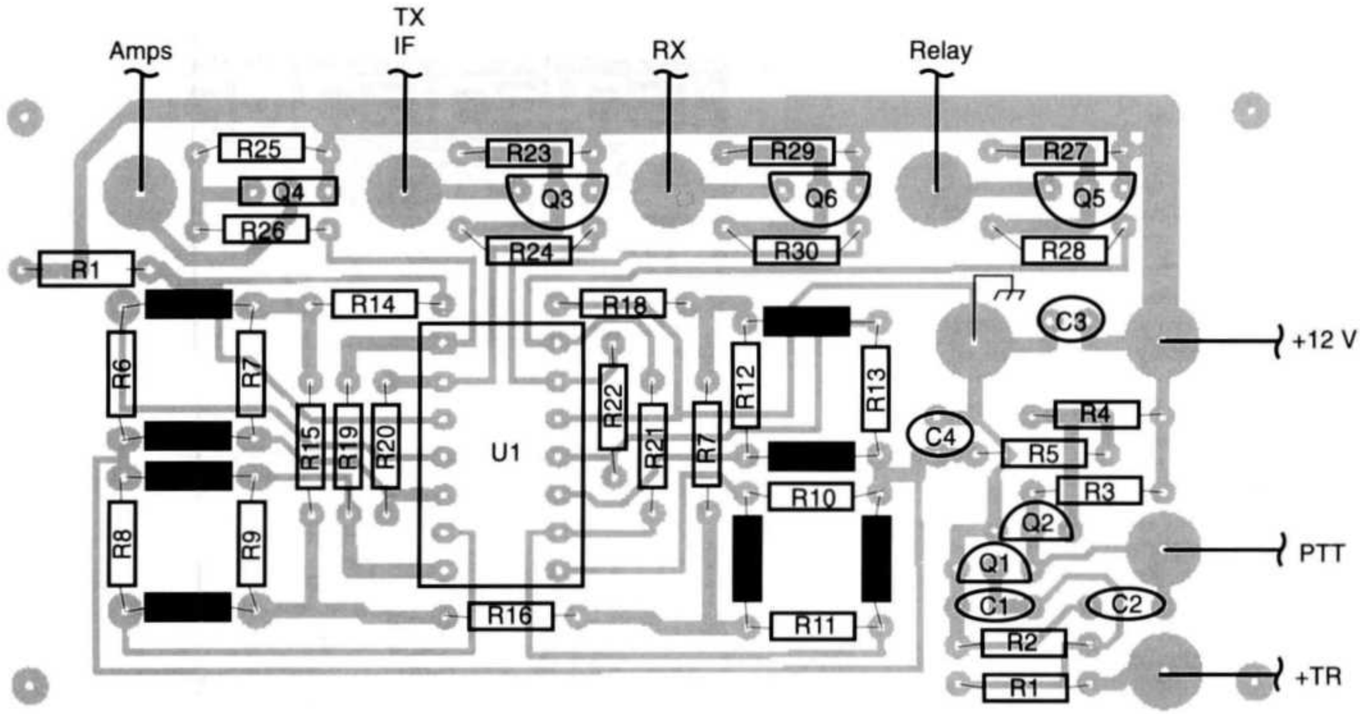


Fig 4—Parts placement diagram using component values and part numbers.

Feedback

In my recent (August 1996, p 21) *QEX* article, "A Better and Simpler A/D for the DDC-Based Receiver," I discussed the center-of-the-passband spurious output from the DDC. This is aurally very annoying, and confuses CW demodulators. I have just discovered that, if the HDF decimation ratio, *R*, in the DDC is set to exactly a power of two (2048, 4096, 8192, 16384, 32768), the tone is gone. The remaining "noise" really sounds like noise.—*Peter Traneus Anderson, KC1HR*

The Impedance of Square Coax Cable

In my May 1995 RF column I suggeted that 1.08 was just an empirically derived constant for the formula Z_0=138 (1.08D/a)—not true! It's a theoretically valid approximation as well. Thanks to Kevin E. Schmidt, W9DF, for sending me a derivation of this constant. One misleading source of this constant is page 13.11 of *The RSGB Microwave Handbook, Volume 2.*

Conference Proceedings Available

1996 AMSAT-NA Space Symposium and Annual Meeting was held November 8-10, 1996, in Tucson, Arizona. Here is a summary of the papers presented. Conference Proceedings are available from ARRL. ISBN: 0-87259-584-6; cost is $12, plus shipping; order number: 5846. Proceedings are also available from AMSAT.

Welcome to the 1996 AMSAT-NA Space Symposium & Annual Meeting, Bill Tynan, W3XO

Phase 3D Update, The Phase 3D Design Team

Phase-3D GPS Receiver Progress Report, Bdale Garbee, N3EUA

A Possible Phase 3D Follow-on Project, Bill Tynan, W3XO

The Amateur Satellite Service in 1996, Ray Soifer, W2RS

Amateur Satellites—Is There a Future?, Richard Limebear, G3RWLK, and John Branegan, GM4IHJ

A Summary of AO-16 Activity for 1994-96, Robert J. Diersing, N5AHD

ASUSAT 1: A Low-Cost AMSAT Nanosatellite, Shea Ferring, Joel D. Rademacher, Helen L. Reed, Jordi Puig-Suari, and members of the ASUSat 1 Team

The Picosat System, Peter Vekinis, KC1QF/EI4GV/SVØGV

Mir: Five Years of a Permanent Packet Radio Space Station, Gustavo Carpignano, LW2DTZ

Amateur Radio on the International Space Station, Frank H. Bauer, KA3HDO, and Matt Bordelon, KC5BTL

APRS SAREX Experiment on Mission STS-78, Bob Bruninga, WB4APR

AMSAT Satellites and ESA Launches, Gould Smith, WA4SXM

Amateur Radio Satellites on the Internet, Eric Cottrell, WB1HBU

SETI Sensitivity: Calibrating on a Wow! Signal, Dr. H. Paul Shuch, N6TX

The Mars Global Surveyor Project, Cliff Buttschardt, K7RR

A Time Code Reader/Display for Russian Tsikada Satellites, John M. Franke, WA4WDL

AMSAT: A Tutorial for Beginners, Barry A. Baines, WD4ASW

Keplerian Element Fundamentals, Ken Ernandes, N2WWD

The View from Below: Thoughts on Phase 3D Groundstation Requirements, Ed Krome, KA9LNV

(My) Hamshack of the Future, Andrew A. Skattebo, KAØSNL

Quick, Inexpensive and Effective: A Simple Satellite Mobile QRP Station for the Beginner, Douglas Quagliana, KA2UPW

My "Landfill Special" RS-10 Satellite Station, William D. Rausch. AA6PA

Upcoming Technical Conferences

Aurora '97

The Northern Lights Radio Society presents Aurora '97, the 14th annual upper midwest VHF/UHF/Microwave Symposium, to be held Saturday, April 19, 1997, at the Village North Professional Building in Brooklyn Park, Minnesota.

Activities will start at 9:00 AM with antenna-gain measurements and a VHF flea market, and continue through the afternoon and evening with a full schedule of talks and workshops, including noise-figure measurements.

This year's conference is being moved from its former date in February and expanded to a full day. It is expected that the move to Spring will allow many more people to attend. For more information, check the NLRS web page: **http://www.tc.umn.edu/nlhome/m374/husby002/** or contact: Donn Baker, WA2VOI, 3128 Silver Lake Road, Minneapolis, MN 55418, 612-781-1359 or email Rich Westerberg, NØHJZ, at **n0hjz@aol.com**. [Information is also available on the CSVHF Web Page: **http://www.umn.edu/nlhome/m042/liebe009/**.—*Paul Husby, WØUC*

Microwave Update 1997

Microwave Update 1997 will be held October 23-26, 1997, in Sandusky, Ohio.

Plans for this year's conference are going well. A surplus tour is sheduled for Thrusday—CTR Surplus, Fair Radio and ARE Surplus. The conference will be on Friday and Saturday. A flea market and noise-figure measurements are scheduled for Friday night. Saturday evening will be a Lake Erie Perch and Walleye fish fry, EME demonstration and flea market. A local group will provide entertainment. A spouse's program is also being planned.

We're still looking for speakers. If you're interested contact me, Tom Whitted, WA8WZG, at 419-732-2944 (voice and fax) or email **WA8WZ G@WA8WZG.COM**.