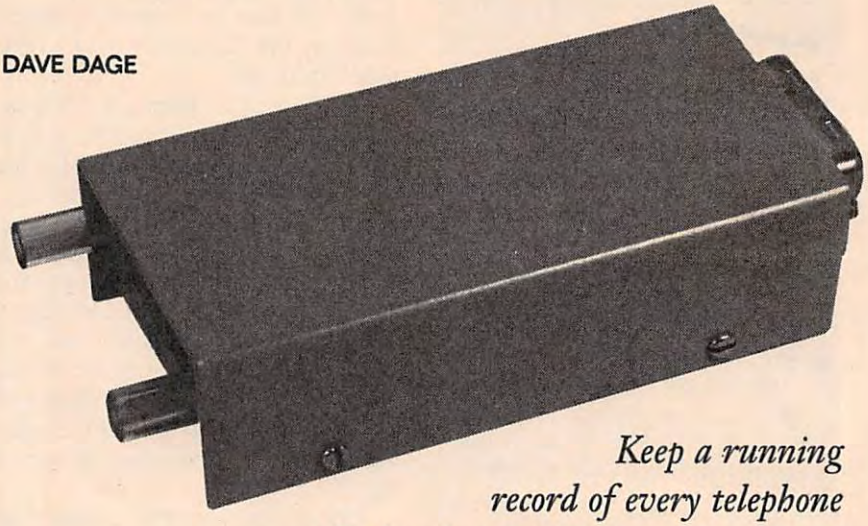
BUILD A TELEPHONE-USAGE MONITOR AND CONTROLLER

DAVE DAGE



Keep a running record of every telephone call placed or received, or even control other devices with a simple telephone call!

Perhaps you have a small business in your house. If so, you probably need to keep a log of your telephone expenses. You might also wish to block certain outgoing telephone calls at your office or house. It also might be convenient to be able to control some devices remotely using your telephone. Wouldn't it be great to have a device to handle all those tasks, expecially if it was small, cheap, and easy to use?

The Phone Troll presented here is just such a device. Roughly the size of a cigarette pack, it connects to your PC's game port on one end and to the telephone line on the other. It converts phone-line current and DTMF dialing tones into digital signals that are compatible with a game port. With a simple Qbasic program, the Phone Troll can be used for many different tasks.

The "Troll" part of the device's name as used here actually has two meanings. The obvious one relates to the mythical creature that lurks undetected, lying in wait to monitor and record every telephone transaction in your home or business. Information that can be logged each time the phone is used includes the start and end times that the phone was in use, what number was dialed, what dialing buttons were pressed during the call, when an incoming call was received, how many times the phone rang before being answered, and whether or not it was answered.

When the telephone is in use (off-hook), the Phone Troll records all valid DTMF signals—including the additional four tone combinations that are not normally a part of the standard 12-button keypad. This will provide a permanent record of all telephone numbers dialed as well as any other digit activity.

The second meaning of "troll" can be loosely translated into "control". Since the DTMF signals are decoded by a QBasic program, they can be used to signal the program to do some particular task—including control some electrical device connected to the computer's parallel port. The QBasic program is able to detect special control numbers that you can set. By simply lifting the telephone and pressing the proper control code, you can individually turn on and off any of up to eight outputs on the parallel port. To demonstrate that function, a parallel-port monitor plug with light-emitting diodes will also be built.

The control codes used in the Qbasic program are three digits long. When the Phone Troll is active, pressing a three-digit code will turn on or off a single output bit. You can activate the control by simply picking up any telephone that the Phone Troll is connected to, entering a three-digit code, and then hanging up. That feature will also work while talking to someone else. The three-digit code can be entered at any time without disconnecting the call. You can also place a call to your home or office from anywhere in the world, enter the desired control code, and turn an output bit on or off. Of course, the telephone must be answered in order to activate the Phone Troll, but that can be done either by a person or by an answering machine.

Calling home from your car phone just to flip the status of an LED on the back of your computer might not sound like an exciting way to use a pile of expensive high-tech gear. But when the LED is replaced with an opto-isolator acting as a solid-state relay, practically anything electrical can now be controlled by your PC. For instance, turning on a furnace or an air conditioner a couple of hours before you arrive home from a vacation would definitely have some merit. The program listed here needs no modification; it will accomplish all those tasks as written.

One non-obvious use for the Phone Troll is placing the contacts of a single-pole normally-closed

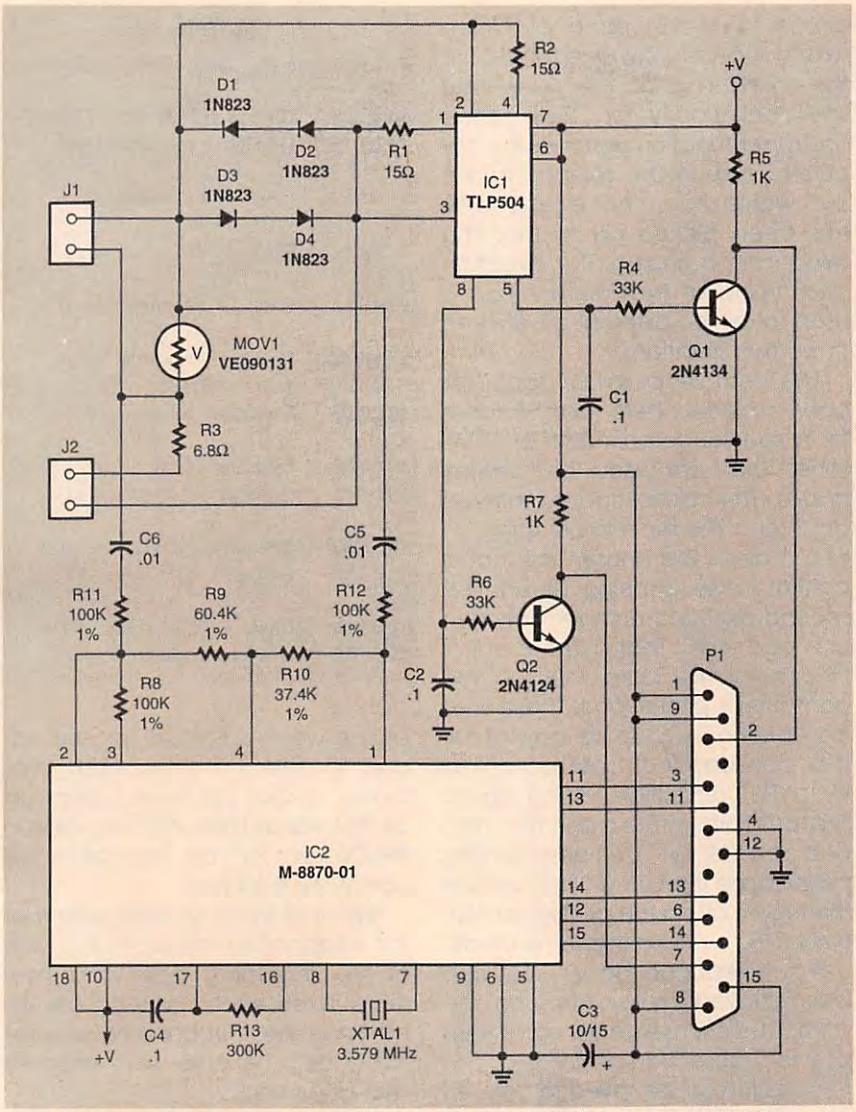


Fig. 1. The Phone Troll is a simple circuit that can sense both telephone usage and DTMF tones. A QBasic program logs activity and can even control other devices.

relay in series with the telephone line. Modify the program to sense the first digit or several digits being dialed on a new call. If the number being dialed starts with a "1" or "1900", activate the relay for a half second. Guess what? That will stop unauthorized persons from using your telephone to place long distance calls or calling any of the "1-900" numbers. The program should also contain a code to enable and disable that function. No one without the proper code would be able to access any long-distance numbers using your telephone.

Note that this arrangement would not be able to prevent operator-assisted calls, unless you also include "0" in the list of numbers on the restricted list. However, doing that is not a good idea—especially as it could be necessary to contact the operator in the event of an emergency.

Yet another use for that "hang-up-the-telephone" technique relates to ordering merchandise from one of those "shop-at-home" TV channels. It's easy to write a QBasic program to test each DTMF digit being entered on the telephone. Have the Phone Troll watch for a string of digits that match any of the family or business credit-card numbers. When one of those numbers is detected, pulse the relay as detailed above. The person will be wondering why the telephone would mysteriously disconnect.

Project Overview. The Phone Troll circuit attaches to the game port on a PC with a standard 15-pin connector. Power for the Phone Troll is supplied by the PC though the game port. The telephone line connects to the opposite end of the Phone Troll through a standard telephone jack. The device is secured to the game port with two mounting screws. Load and run the program, and you're in business.

The Phone Troll has two separate

PARTS LIST FOR THE PHONE TROLL TELEPHONE-USAGE MONITOR

SEMICONDUCTORS

IC1—TLP504 opto-isolator, integrated circuit
IC2—M-8870-01 DTMF decoder, integrated circuit
D1–D4—1N832 silicon diode
MOV1—VE090131 metal-oxide varistor, 38-joule, 130-volt rms
Q1, Q2—2N4124, NPN transistor

RESISTORS

(All resistors are ¼-watt, 5% units, unless otherwise noted.)
R1, R2—15-ohm
R3—6.8-ohm
R4, R6—33,000-ohm
R5, R7—1000-ohm
R8, R11, R12—100,000-ohm, 1%
R9—60,400-ohm, 1%
R10—37,400-ohm, 1%
R13—300,000-ohm

CAPACITORS

C1, C2, C4—0.1-µF, ceramic-disk
C3—10µF, 15-WVDC, electrolytic
C5, C6—0.01-µF, 5%, Mylar

ADDITIONAL PARTS AND MATERIALS

XTAL1—3.579545 MHz crystal, HC49/U case
J1, J2—RJ11 telephone jack, PC-mount
P1—15-pin "D"-style male connector, PC-mount
Case, hardware, etc.
Note: The following items are available from: Dage Scientific, PO Box 144, Valley Springs, CA 95252, Tel: 209-772-2076: Kit of all parts, PC board, and manual (without case or program diskette), No. FT-1, $26.95; Case, No. FT-2, $9.95; Program on 3½-inch diskette, No. FT-3, $2.00. Please add $4.00 for shipping and handling. CA residents must add appropriate sales tax. Visa and Mastercard are accepted on telephone orders.

IC2 can be ordered in single quantities from: ICS, Inc., 2222 E. Camelback Rd. # 222, Phoenix, AZ 85016-3427; Tel: 602-224-5322.

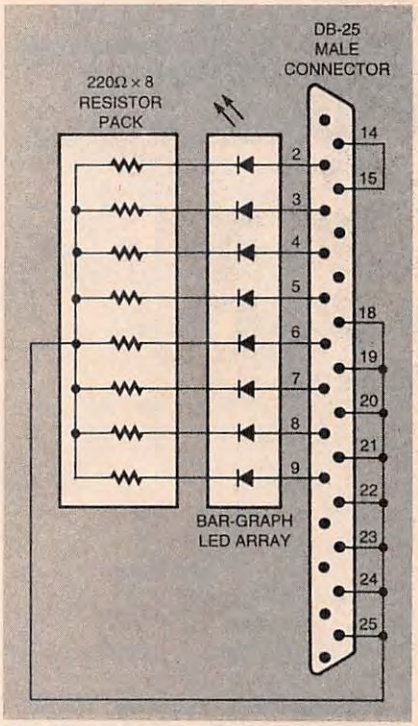


Fig. 2. The test fixture simply connects 8 LEDs to the data lines of a printer port. Connecting pins 14 and 15 together helps the Phone Troll program find the port that the test fixture is connected to.

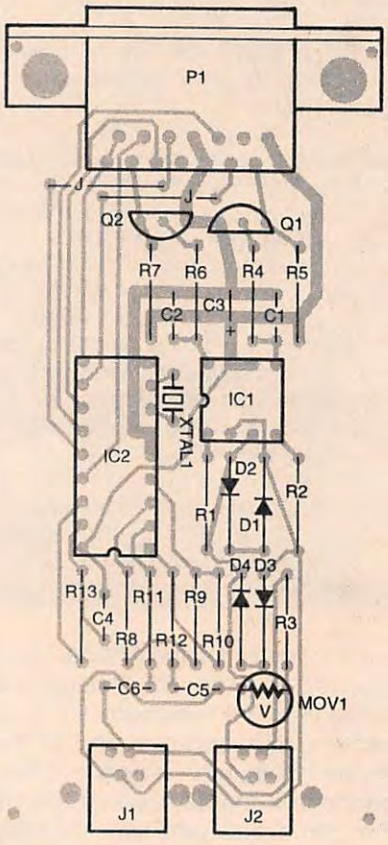


Fig. 3. The Phone Troll fits easily on a single-sided PC board. Only two jumpers are needed.

circuit functions, each of which runs independently. One function is the monitoring of the telephone line continuously for DTMF tones. The other function watches for any current flow in the telephone line that would mean that a telephone has been picked up or that the telephone is ringing. The direction and timing of that current flow is used to tell the difference between those two situations.

The sample program that has been included here also watches for a special combination of DTMF tones that are used as "control codes." The control codes are three digits long. The first number is the "#" key; it alerts the Phone Troll that a control code is being issued. The second digit is a number between one and eight. That number points to one of the data lines on the computer's printer port. Those lines can be hooked up to any circuit that can be electrically controlled by a TTL-compatible digital signal. The last digit is either a one (for "on") or a zero (for "off"). As an example, pressing the three keys "#81" will turn the eighth bit on the printer port on, while "#80" will turn the same bit off.

Any telephone activity will also be displayed immediately on the computer's screen. After each telephone transaction is finished, the information will be saved to a file on disk under the filename "phonelog.txt". That file will be created in the same directory as the QBasic program. Each time the Phone Troll program is started, any additional entries will be added to the end of that text file. Since it is a simple text file, any convenient text editor in either DOS or Windows can be used to view and print the log. The file can be renamed to save it, or it can be deleted. In either case, a new file will be created and the logging started fresh.

An example log file is shown in Listing 1. The appearance is the same on screen and when saved to disk. The first time the program is run, the starting date and time are recorded. After that, just the time is recorded until midnight. At that point, the next record will be preceded by the new date.

In the example, we can see that according to the records, a telephone went "off-hook" (picked up) and a telephone number was dialed at 2:43 PM (which appears as 14:43 as all times are recorded in the 24-hour format). The call lasted about ten minutes.

LISTING 1

Phone log starting 08-16-1997 14:40:57

14:43:08	Off-Hook, DTMF active 7722076
14:53:13	On-Hook - call terminated
16:52:57	Ring - 3 not answered.
17:05:12	Ring - 2 not answered.
17:15:31	Ring - 2 not answered.
17:22:44	Ring - 1 answered. DTMF active - #81#80
17:27:09	On-Hook - call terminated
19:54:39	Off-Hook, DTMF active - #21
19:54:41	On-Hook - call terminated
19:55:53	Off-Hook, DTMF active - #31
19:55:55	On-Hook - call terminated
19:59:18	Off-Hook, DTMF active - #30#20
19:59:22	On-Hook - call terminated

The next three records show that the telephone rang. Both the time of the incoming calls and how many times each call rang are listed. Since the telephone never went "off-hook", the calls were listed as "not answered".

Next, the telephone rang once, was answered, and a control code of #81 was entered, activating the device that is connected to data line 8 on the printer port—let's say that it is a tape recorder whose microphone is connected to the telephone line. Before the call was terminated, control code #80 turned the tape recorder off. That could have been done from any telephone that is monitored by the Phone Troll and not just the one that was picked up for the call. An arrangement like that can be a handy substitute for pencil and paper when a message or important information needs to be saved.

The final three entries are additional demonstrations of controlling devices connected to the computer's printer port from a telephone. First, a lamp connected to bit 2 was turned on. Minutes later, another call turned on the home stereo con-

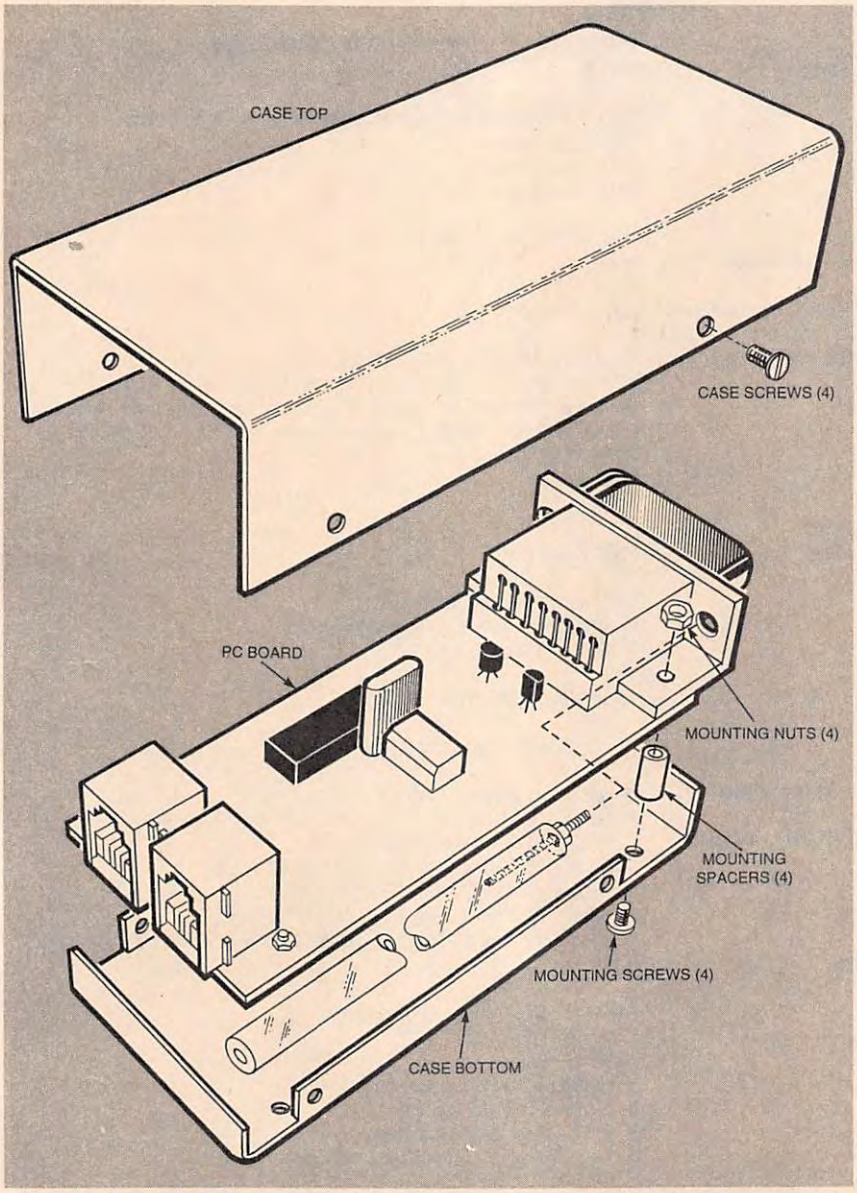


Fig. 4. Placing the Phone Troll in a compact case lets the unit attach directly to the game port on a computer.

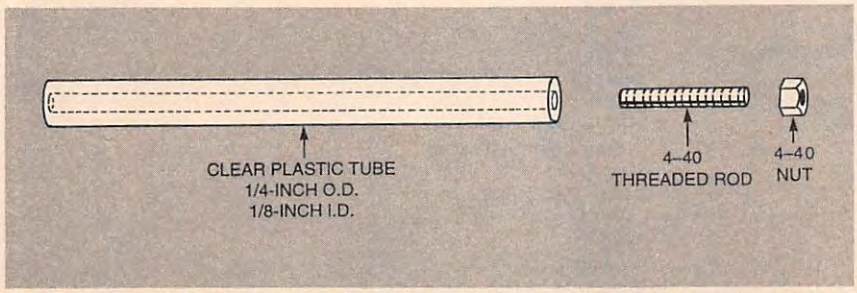


Fig. 5. The special hold-down screws for the Phone Troll are easily made from a piece of plastic tube and a length of threaded rod. You can easily make the threaded rod by cutting off the head from a 4-40 screw.

nected to bit 3. Several minutes later, the final call turned off both the stereo and the light in that order. Understand that any device that can be interfaced to a data line on a printer port can be controlled—lights, heat, air conditioning, or any other device that you can devise a control method for.

Circuit Description. The schematic diagram shown in Fig. 1 reveals how simple the Phone Troll's interface circuit is. Power for the circuit is supplied from the host computer's 5-volt supply through pins 1, 8, 9, and 15 of the game port. Ground is supplied from pins 4 and 12. Telephone jack J1 connects to the telephone line and is surge protected with MOV1. The other telephone jack, J2, is the jack through which the Phone Troll can monitor telephone activity. Any telephones connected to the J1 side of the telephone-wiring system will not activate the Phone Troll. Because of that, the Phone Troll will work best when connected between the telephone company and the telephones in the house—much like connecting additional telephone-line devices to the pass-through connector on a modem, fax machine, or answering machine.

Current on the telephone line is sensed by opto-isolator IC1. Two identical circuits within IC1 sense the current flow depending on the direction that it is flowing. Current flow in one direction will pass through R2 and pins 3 and 4 of IC1 while current flow in the other direction will pass through R1 and pins 1 and 2 of IC1. A wide range of DC current can flow from the telephone's central office, so diodes D1—D4 regulate the current through IC1. The small-signal audio impedance through the diodes and IC1 is matched by R3.

By monitoring the current and its direction, normal telephone activity can be determined. When current flows steady in one direction, the telephone has gone "off hook". On the other hand, current flowing rapidly in both directions means that the telephone is ringing. Typically, the "off hook" pattern is one of current flowing for ½ second in one direction, then switching to the other direction for the remainder of the call.

The second part of the Phone Troll circuit is the DTMF sensing and decoding. DTMF signals fit within the telephone band of 300 to 3000 Hz. Further information on DTMF signals can be found on the Web at http://www.2xtreme.net/dage/dtmf_tut.html.

The DTMF receiver (IC2) is made by Teltone Corp. That chip makes

LISTING 2

```
REM Release version 1.00 Date 10/16/97
REM No global variables, auto detection of output plug

DECLARE SUB OneSecTD ()
DECLARE SUB FindPort ()
DECLARE SUB LineSample ()
DECLARE SUB Ring ()

REM ratio = time current flows in one direction, divided by
REM time current flows in other direction during sample window.

REM The DATA below are the normal addresses for LPT1 to LPT3 in
   decimal.
REM Add additional output port (in decimal) to DATA, ending with 0 only.
REM "SUB FindPort ()" will read DATA below
DATA 888,632,956,0

CLS
CALL OneSecTD ' initialize
REM BEEP
REM PRINT "OneSec = "; OneSec; " FOR/NEXT loops"
REM FOR t = 1 TO OneSec: NEXT t ' wait one second
CLS
CALL FindPort

CLS

RecordDate$ = DATE$

OPEN "PHONELOG.TXT" FOR APPEND AS #1

PRINT #1, "Phone log starting "; RecordDate$, TIME$: PRINT #1, :
PRINT #1,
PRINT "Phone log starting "; RecordDate$, TIME$: PRINT : PRINT
CLOSE #1

inactive: '////////// START LINE MONITORING //////////

ringct = 0: noanswer = 0: incoming = 0     'initialize

DO
CALL LineSample
LOOP WHILE ratio = 9 'no current, ratio =9


CALL LineSample 'get fresh reading
IF ratio = 9 THEN GOTO inactive 'glitch remover

REM Phone line now active - either off-hook or ring

OPEN "PHONELOG.TXT" FOR APPEND AS #1

IF RecordDate$ <> DATE$ THEN 'Prints date once each day.
 RecordDate$ = DATE$
 PRINT #1, "Phone log for", RecordDate$: PRINT #1, : PRINT #1,
 PRINT "Phone log for", RecordDate$: PRINT : PRINT
END IF

PRINT #1, TIME$,
PRINT TIME$,

IF ratio > .5 THEN incoming = 1: CALL Ring

CallHandeling:

IF incoming = 0 THEN
 PRINT #1, "Off-Hook, DTMF active - ";
 PRINT "Off-Hook, DTMF active - "; 'DTMF digits displayed here
 ELSEIF noanswer THEN
 PRINT #1, "Ring - "; ringct; " not answered."
 PRINT #1, : CLOSE #1
 PRINT "Ring - "; ringct; " not answered.": PRINT : GOTO inactive
 ELSE
 PRINT #1, "Ring - "; ringct; " answered. DTMF active - ";
 PRINT "Ring - "; ringct; " answered. DTMF active - ";
 END IF

 CALL LineSample 'used for delay when phone lines switch
 CALL LineSample
 CALL LineSample
 CALL LineSample

 REM ////////// start DTMF //////////
 dtmf:

 DO

 DO UNTIL INP(513) AND 128 'loops until next digit
 IF (NOT INP(513) AND 48) = 0 THEN 'exits when on-hook
 PRINT #1, : PRINT #1, TIME$, "On-Hook - call terminated": PRINT #1,
 PRINT : PRINT TIME$, "On-Hook - call terminated": CLOSE #1
 PRINT : GOTO inactive
 END IF
 LOOP

 REM Read DTMF

 OUT 513, 0 'start one-shots
 n = INT(OneSec * .01) 'wait 10 mSEC
 FOR t = 1 TO n: NEXT t

 value = NOT INP(513) AND 15

 REM //// print all 16 valid DTMF codes ////

 IF value = 0 THEN PRINT "D"; : PRINT #1, "D";
 IF (value > 0) AND (value < 10) THEN PRINT CHR$(48 + value); :
    PRINT #1, CHR$(48 + value);
 IF value = 10 THEN PRINT "0"; : PRINT #1, "0";
 IF value = 11 THEN PRINT "*"; : PRINT #1, "*";
 IF value = 12 THEN PRINT "#"; : PRINT #1, "#";
 IF value > 12 THEN PRINT CHR$(52 + value); : PRINT #1, CHR$(52 +
    value);

 REM //// Start Control Application ////
 IF bas% THEN
 contdig = value
 IF contdig = 10 THEN contdig = 0
 IF cont = 1 THEN s = s + 1
 IF s = 1 THEN addr = contdig
 IF s = 2 THEN dta = contdig: s = 0: cont = 0: doit = 1

 IF contdig = 12 THEN cont = 1

 IF doit = 1 THEN
 op = 2 ^ (addr - 1)
 IF dta = 1 THEN
 outreg = outreg OR op
 ELSE
 op = op XOR 255
 outreg = outreg AND op
 END IF
 OUT (bas%), outreg
 doit = 0
 END IF
 END IF
 REM //// End Control Application ////

 DO WHILE INP(513) AND 128 'loops until digit gone
 IF (NOT INP(513) AND 48) = 0 THEN 'exit when on-hook
 PRINT #1, : PRINT #1, TIME$, "On-Hook - call terminated": CLOSE #1
 PRINT : PRINT TIME$, "On-Hook - call terminated"
 PRINT : GOTO inactive
 END IF
 LOOP
LOOP

END
```

designing a DTMF-receiver circuit very simple because all of the processing is performed on-chip. All that is needed for operation is a power supply, the audio signal, a timing crystal, and adjusting the guard time.

An AC-balance connection is made to the telephone line through C5 and C6. Precision resistors R8–R12 provide the balance network to IC2—it is important to balance both sides of the telephone line with respect to the op-amp portion of IC2. A 3.579545-MHz color-burst crystal, XTAL1, is directly connected to pins 7 and 8 of IC2. All of the necessary drive circuitry is internal with no external components needed. The guard time is set by R13 and C4. The values listed set the valid length for the "tone-present" and "tone-absent" timing to 40 ms. Those values should work perfectly; they can be adjusted if needed.

When a valid tone pair has been detected, the four-bit binary code that represents that tone pair appears on output pins 11–14. Pin 15 also goes high for 40 ms as a signal that the data on the output lines is valid and stable. That feature is very important in order to be able to use the game port in such a non-traditional manner.

An IBM game port has 8 inputs—

LISTING 3

```
SUB FindPort
SHARED OneSec
SHARED bas%

READ bs

DO
 test1 = 0: test2 = 0
 OUT (bs + 2), 4
 IF INP(bs + 1) AND 8 = 8 THEN test1 = 1
 OUT (bs + 2), 6
 IF NOT INP(bs + 1) AND 8 THEN test2 = 1
 IF test1 AND test2 THEN bas% = bs
 READ bs
LOOP WHILE bs

IF bas% = 0 THEN
 BEEP: PRINT "Test plug not found. Press  any key to enter address,"
 PRINT "or, wait for program to start without the control function.": PRINT : PRINT
 REM loop is about three times as long as  OneSec
 DO WHILE INKEY$ = "" AND t < 3 * OneSec
 t = t + 1
 LOOP
 IF t < 3 * OneSec THEN
 INPUT "Enter output port address in decimal."; bas%
 ELSE EXIT SUB
 END IF

ELSE PRINT "Base address = ";
HEX$(bas%); " HEX, or decimal"; bas%

REM Cycle output LEDs
 FOR n = 1 TO 3
 OUT bas%, 255: BEEP: FOR t = 1 TO OneSec / 5: NEXT t
 OUT bas%, 0: FOR t = 1 TO OneSec / 10:
NEXT t
 NEXT n
END IF

END SUB
```

LISTING 4

```
SUB LineSample
SHARED ratio

Ia = 0: Ib = 0: LoopStartTime = TIMER
DO WHILE (TIMER - LoopStartTime < .2)
 IF NOT INP(513) AND 16 THEN Ia = Ia + 1 'check bit 4, active low
 IF NOT INP(513) AND 32 THEN Ib = Ib + 1 'check bit 5, active low
 REM correct LoopStartTime when passing thru midnight
 IF (TIMER - LoopStartTime < 0) THEN LoopStartTime = LoopStartTime - 86400
LOOP

IF Ia = 0 AND Ib = 0 THEN 'Ideal samples
 ratio = 9 ' ratio = 9 Assigned; Line is calm
 ELSEIF Ia < Ib THEN ' ratio = 1 Active Ring
 ratio = Ia / Ib ' ratio = 0 Off-Hook
 ELSE ratio = Ib / Ia
END IF

END SUB
```

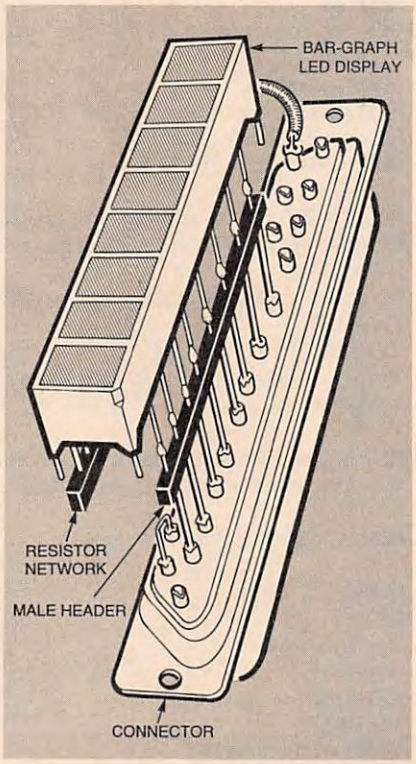


Fig. 6. The finished test fixture will look something like this. A straight-pin header helps connect the bar-graph display to the connector. The pins on the header will have to be spread to match the spacing on the header.

4 analog and 4 digital. Three of the digital inputs are used—two for the telephone-current sensing, and one for the "DTMF valid" signal. The data lines from IC2 are connected to the analog inputs. Here is where the "non-traditional use" comes into play. The DTMF data signals are connected to an individual resistor/capacitor circuit in the game card hardware that is used to control the pulse width of a one-shot multivibrator. If the digital signal is high, the one-shot times out fast; if the digital signal is low, the one-shot takes a long time to time out (if at all).

Since the DTMF signals are locked for at least 40 ms, that leaves enough time for the QBasic program to check the "joystick button" that indicates that DTMF data is available. The program reads the analog "joystick position" by triggering all four multivibrators. After a certain amount of time, the state of the one-shots are read out. The time period between triggering the one-shots and reading their state is long enough to give them time to time out if they are sensing a high signal, but not long enough to let them time out if they are sensing a low signal. Although it sounds complex and exacting in terms of critically-timed events, there is a wide margin available, making the readings very reliable. The QBasic program can accomplish those steps with time to spare.

Output Test Fixture. As mentioned before, the Phone Troll can control devices attached to the printer port. While the actual interface circuits are beyond the scope of this article, some way is needed to verify that the Phone Troll can actually send signals to the printer port. A simple test fixture is shown in Fig. 2. It is simply a set of 8 light-emitting diodes connected to the data lines of the port. A set of resistors limit the current through the LEDs, and the circuit is completed through the port's ground lines. With that arrangement, the LEDs can be lit individually with the control codes.

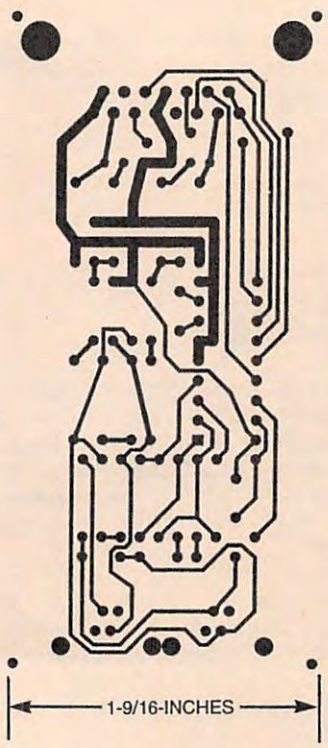
Lighting LEDs might not sound very exciting, but the LEDs can be replaced with any type of opto-isolator chip. Those chips can have different types of circuits on their output sides. Readily-available outputs include bi-polar transistors, silicon-controlled rectifiers, and Triacs.

LISTING 5

```
SUB OneSecTD
SHARED OneSec

REM Purpose:
REM To find the number of FOR/NEXT loops on a specific computer to cause
REM a one second delay, variable [OneSec] is number of loops/second.

REM Use:
REM For program delays > 3 mSEC to < 500 mSEC. Interrupts affect accuracy.

REM Example:
REM To delay 250 mSEC, insert this code:
REM n = INT(OneSec * .25)
REM FOR t = 1 to n : NEXT t

REM **** Calibrate host cmptr / Turbo must remain constant ****
PRINT "Calibrating FOR/NEXT delay loops. Please wait."
x = TIMER
DO WHILE x = TIMER: LOOP 'Wait until new clk tick
x = TIMER
 DO UNTIL TIMER - x >= .5
 n = n + 1 'see how many n's for half second
 LOOP

REM n is the number of do/loops in half sec. estimate for/next is 6X faster
n = n * 24 'should be close to 2 second delay

REM **** test and trim n for a 2 sec delay ****

lastloop = 2 'initialize
DO
 n = n * 2 / lastloop
 x = TIMER
 DO WHILE x = TIMER: LOOP
 x = TIMER
 FOR t = 1 TO n: NEXT t
 lastloop = TIMER - x
LOOP WHILE lastloop < 1.9 OR lastloop > 2.1
OneSec = INT(n / 2)

END SUB
```

Construction. The Phone Troll is a simple project to build. All of the components are mounted on a single-sided PC board. A foil pattern has been included for those who wish to etch their own board. As an alternative, a kit with a pre-etched board is available from the source given in the Parts List.

If you are building the kit or using



Here's the foil pattern for the Phone Troll.

LISTING 6

```
SUB Ring
SHARED ratio
SHARED ringct
SHARED noanswer

RINGADING:

DO
 CALL LineSample
LOOP UNTIL ratio < .5 OR ratio = 9 'loops while ringing, but may catch first silence

ringct = ringct + 1
CALL LineSample 'get full sample

REM ring has either stopped (ratio = 9) or phone goes off-hook (ratio < .5)
REM assume between rings, then check for off-hook

RingStartTime = TIMER

DO WHILE TIMER - RingStartTime < 5  AND ratio = 9
 CALL LineSample
 REM correct RingStartTime when passing thru midnight
 IF TIMER - RingStartTime < 0 THEN RingStartTime = RingStartTime - 86400
LOOP

CALL LineSample

IF ratio > .5 AND ratio <> 9 THEN GOTO RINGADING 'ring again
IF TIMER - RingStartTime >= 5 THEN noanswer = 1

END SUB
```

the included foil pattern, use the parts-placement diagram in Fig. 3 for locating the various components. Note that two jumper wires are needed; they are located near P1. When mounting the components, pay careful attention to the orientation of the semiconductors. The 1% resistors have three digit bands instead of two. The values, however, can still be read using the standard color codes.

The assembled board can be enclosed in a suitable case and mounted to the game port as shown in Fig. 4. With that arrangement, you will need to make a pair of extended screws for securing the Phone Troll to the game-port connector. Take a suitable length of stiff plastic pipe that is hollow. Glue a length of 4-40 threaded rod in one end so that it sticks out enough to screw down P1 to the game port. A nut should be threaded onto the rod to act as a stop. The length of the plastic pipe should be long enough to reach out the back of the case. That way, the screw can easily be turned by hand. The details for the extended screw are shown in Fig. 5. If you do not want to mount the Phone Troll like that, you can install it in a receptacle box and connect it to the game port with a cable and 15-pin "D" connector.

The test fixture can be made from an eight-LED bar-graph display, a single-inline resistor network, a 25-pin connector, and a strip of single-row straight header. The header will need to be bent on one side to match the spacing of the connector. That method makes a rugged assembly that matches up the different pin spacings between the connector and the bar-graph display.

Don't forget to install a jumper between pins 14 and 15 on the connector. The QBasic program will check all of the available parallel ports. The jumper will help identify the connector to the program. The completed fixture will look similar to the one shown in Fig. 6. As a side note, the test fixture can also be used as a stand-alone test tool. You can use it to automatically find, identify, and test an active parallel port including enhanced parallel ports.

Testing the Phone Troll. Connect the telephone lines to the Phone Troll and attach it to the PC's game port. Plug the test fixture into the printer port. The QBasic program will have to be typed in—follow Listing 2 through Listing 6 in order. Save the program as "FONETROL.BAS". Run the program, and the Phone Troll is up and running. If you aren't using the test fixture, the program will pause to let you set the parallel-port address. If no address is entered, the program will continue but without the control function.

After the program checks your computer's speed (for the timing function) and determines which parallel port is active, it will display the date, time, and start a log file. Pick up a telephone and the screen should show "Off-Hook, DTMF active". Press #11 through #81 and all eight LEDs should turn on one at a time, depending on which LED you're activating. Press #10 through #80 and all eight LEDs should turn off.

When building a new device, there's always the possibility that it won't work the first time. Recheck all of the parts for correct placement, orientation, and solder connections before continuing. Troubleshooting is more of an art form than a science and can't easily be explained in a few paragraphs. However, here are a few test points to check that will help isolate any trouble spots in the circuit. When the telephone goes off-hook, the voltage at R7 (next to Q2), or voltage at R5 (next to Q1) will drop from 5 volts to less than 1 volt. With a high-impedance device such as an oscilloscope or logic probe, verify that the oscillator at pin 8 on IC2 is running. Pick up a telephone that is being monitored by the Phone Troll and press a button. You should see that pin 15 on IC2 pulses low. Those tests will help pinpoint any spots in either the circuit or the software for additional tests.

Programming in QBasic. A nice feature with QBasic is that you are not restricted to the original program as would be the case with a compiled executable program. Running the Phone Troll directly from the source code lets you customize the Phone Troll's features to your own wants, needs, and desires. You will still need

to know how to program in QBasic, but it's very easy to modify the code and see what happens. Many budding programmers start gaining hands-on experience by modifying a program to achieve a particular goal. Just make sure that you have a backup copy of the program before you start experimenting. The program is well documented and will stand alone to act as a guide for those who are familiar with QBasic.

One example of a need to modify the program has to do with distinctive-ring service. The program will detect and count a normal ring cycle (two seconds of ringing followed by four seconds silence). Distinctive ringing might cause the Phone Troll to count each burst of rings, indicating two or three rings for each actual ring. Simply divide the variable *ringct* by two or three, or rewrite the program to indicate which sequence of rings was received.

Like any good servant, your troll awaits. Ω