Build an intelligent cable tester for only $25!

JIM BARBARELLO

A single fault in a printer or modem cable can disable the device just when you need it most. A cable fault is not a problem if you're a cable manufacturer, because you can use your sophisticated (and expensive) cable tester to troubleshoot the cable. The problem is that most of us can't afford that type of equipment. There is, however, a simple and inexpensive alternative. If you own a PC and have about $25 in spare change, you can build your own multi-line cable tester.

Our tester connects to your PC's printer port and allows you to test any cable with as many as 24 active lines. The tester works by performing a continuity check on the 576 possible interconnections; a complete test takes only about ten seconds. The tester uses inexpensive, readily available components, is built on a solderless breadboard (so no PC board is required), and is controlled by a simple BASIC program, which you're free to modify or enhance.

Multiplexing

To understand how the cable tester works, let's review the concept of multiplexing. Consider the circuit shown in Fig. 1. The two rotary switches are ganged, so both sections change simultaneously. In each position a different device glows: LED1, LED2, or LMP1. However, if we rotate the

switches quickly, all three devices will appear to be on at the same time.

That process is called *multiplexing*; it lets each device share the common transmission line part of the time. Multiplexing is the basis of modern phone transmission; that's how hundreds of calls are sent over a single pair of wires.

Our cable tester reverses the standard multiplexing scheme to allow testing many wires with just a single input and a single output. The principle is shown in Fig. 2; note that the rotary switches there are not ganged.

If we place both switches in position 1, LED1 will light, indicating continuity. But if we leave S1 in position 1 and move S2 to position 2, the LED should go out. If it doesn't, there must be a connection—intentional or otherwise—between lines 1 and 2.

We accomplish that type of switching with a 4051 multiplexer/demultiplexer, shown in Fig. 3. As shown in Table 1, depending on the binary code present at inputs A, B, and C of the 4051, we can connect the common line to any of the eight transmission lines. Note that if we apply a high level to the Inhibit line, all connections are broken between the common and transmission lines, regardless of the state of the A, B, and C inputs. Like the rotary switches in Figures 1 and 2, the 4051 can accommodate AC and DC signals in both directions.

The circuit

If the PC had a standard interface with 24 input and 24 output lines, multiplexing would be unnecessary. It doesn't, so the cable tester uses the 11 output lines of the PC's parallel port to control six 4051 IC's, and one of the port's five input lines to sense transmission status. The complete circuit is shown in Fig. 4.

To select IC1, we must bring its Inhibit line low while keeping the Inhibit lines of IC2 and IC3 high. Then, depending on what we apply to the A, B, and C inputs, we
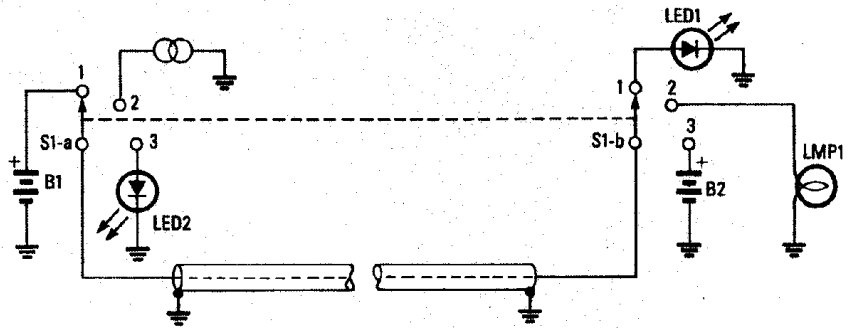

Fig. 1. A MULTIPLEXED CIRCUIT allows a single cable to carry several different circuits.
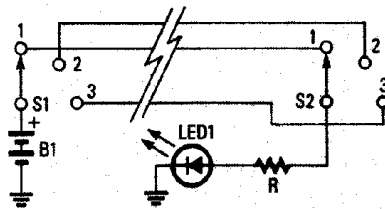

Fig. 2. BY DECOUPLING the switches and driving (B1) and sensing (LED1) the common lines, it's possible to test various combinations of connections.
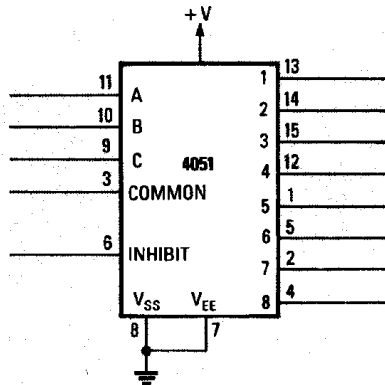

Fig. 3. THE CMOS 4051 allows a single common line to connect to one of eight input/output lines when the inhibit line is low.

can select any of the eight transmission lines. The Common line (pin 3) is connected to +V, so +5 volts will be transmitted through the selected transmission line to the cable under test via J1.

The other end of the cable is connected to J2. There, a line is selected by similar use of the Inhibit and A, B, and C lines of IC4, IC5 and IC6. The Common output line is sensed by pin 11 of the printer port. Resistor R1 serves as a pull-down to prevent false readings.

Think of the combination of IC1, IC2, and IC3 as a 24-position electronically controlled rotary switch. Similarly, the combina-

tion of IC4, IC5, and IC6 acts like a second rotary switch. Each "switch" is independently adjustable. For example, line 1 can be selected as the input to the cable, while any of lines 1 through 24 can be sensed at the output of the cable. That independent selection allows checking of all possible combinations of input and output lines. Power is provided by three 1½-volt AA batteries.

Addressing the 4051's

On the input side, lines A, B, and C are connected to printer-port pins 4, 3, and 2, respectively. Those pins correspond to the three lowest bits (D2, D1, and D0) of the 8-bit parallel data. To select transmission line 1, we connect the A, B, and C lines to ground; to select transmission line 8, we connect A, B, and C to +V.

Note that all three IC's (IC1–IC3) are addressed; we avoid contention by releasing the Inhibit line of only one IC at a time. The Inhibit line for IC1 is pin 1 (STROBE). For IC2, it is pin 14 (AUTOFD), and for IC3 it is pin 16 (INIT). To select IC1, send a decimal 5 to port 890 (OUT 890,5). A decimal 6 selects IC2, and decimal 0 selects IC3.

Addressing the output side is more complicated. Bits D7 and D6 (pins 9 and 8, respectively) are used to generate the A, B, and C signals through IC7 and IC8. To select IC4, set D7 = 1 and D6 = 1. Select IC5 with D7 = 1 and D6 = 0. Finally, select IC6 with D7 = 0 and D6 = 1. The A, B, and C lines are driven by D5, D4, and D3 (pins 7, 6, and 5, respectively). So, to select transmission line "5" in IC5, we would make D7 = 1, D6 = 0, D5 = 1, D4 = 0 and D3 = 0.
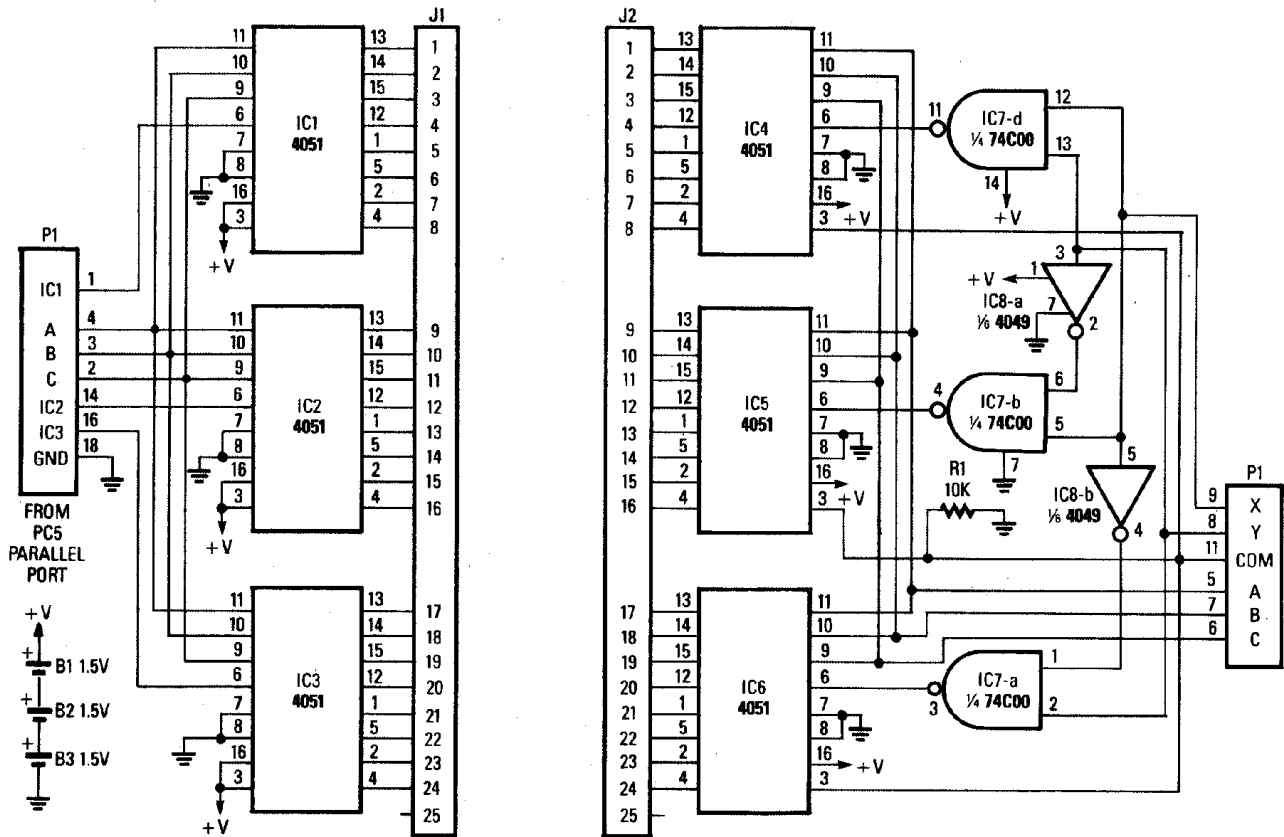
Fig. 4. *COMPLETE SCHEMATIC DIAGRAM of the cable tester. The circuitry to the left of J1 is the input side; to the right is the output. Plug P1 connects to the PC's parallel port.*

However, we must be careful not to disturb bits D2, D1, and D0, which were previously set. We do that by reading the status of port 888 and masking off the higher bits with the function INP(888) AND 7. That gives us a decimal value indicating the status of bits D2–D0. We can now add the decimal value from bits D7–D3 to that number and send it back out. To continue the previous example, we had D7 = 1, D6 = 0, D5 = 1, D4 = 0, and D3 = 0, or 10100XXX (with the X's meaning we don't care what those bits are). The decimal number for that pattern is $128 \times 1 + 64 \times 0 + 32 \times 1 + 16 \times 0 + 8 \times 0$, or 160.

If D2 = 1, D1 = 1, and D0 = 0, that pattern's decimal equivalent is 6. Add that to 160 and we get 166. So to select input line 15 and output line 13, we OUT 890,6 (to select IC2), and OUT 888,166 (to select transmission line 7 in IC2 and transmission line 5 in IC5).

Although the calculations seem somewhat involved, the computer program accomplishes them quickly; in Listing 1, see lines 110 and 120 for the input side and 250–290 for the output side of the PC Cable Tester.

Construction

It's certainly possible to design a PC board for the tester, but we chose to build it on a standard 6 ½-inch solderless breadboard, which provides just enough room for the circuit's eight IC's. Use short lengths of no. 22 solid wire to make the breadboard interconnections.

Next, obtain a four-foot length

TABLE 1—4051 TRUTH TABLE

Inhibit	A	B	C	Line Selected
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	7
0	1	1	1	8
1	X	X	X	NONE

of 15-conductor cable; that cable connects to the PC's printer port. On the end that will attach to the

```
1 ─── 1
2 ─── 2
3 ─── 3
4 ─── 4
5 ─── 5
6 ─── 6
7 ─── 7
8 ─── 8
9 ─── 9
10 ─── 10
11 ─── 11
12 ─── 12
13 ─── 13
14 ─── 14
15 ─── 32
16 ─── 31
17 ─── 36
19 ─── 19
25 ─── 30
```

25-PIN
D
CONNECTOR
(MALE)

36-PIN
CENTRONICS
PARALLEL
PRINTER
CONNECTOR
(FEMALE)

Fig. 5. ADAPTER for testing a standard printer cable with a male Centronics plug on one end.

LISTING 1

```
1 REM**  PC Cable Tester Program
2 REM**  Name: PCCABLE
3 REM**  (c) 1987 J.J. Barbarello
4 REM**  V871017
10 DEF SEG=64:KEY OFF:DEFINT I,O:DIM I(576),O(576)
20 CLS:CNT=1:PRINT TAB(30);'P.C. CABLE TESTER':PRINT STRING$(79,223)
30 LOCATE 4,15:PRINT STRING$(50,32):LOCATE 4,15:LINE INPUT'How many wires in
   the cable (1-24)?...  ';NL$:NL=VAL(NL$)
40 IF NL<1 OR NL>24 THEN 20
50 N1=INT(NL/8):N2=NL MOD 8:IF N2=0 THEN N2=7 ELSE N2=N2-1
60 LOCATE 6,18,0:PRINT'Checking for Connections on Line:'
90 T=TIMER
100 FOR I=1 TO NL
110 IF I=1 THEN OUT 890,5 ELSE IF I=9 THEN OUT 890,6 ELSE IF I=17 THEN OUT 890,0
120 OUT 888,(I-1) AND 7:LOCATE 6,52:PRINT USING'##';I
130 GOSUB 250:NEXT I
140 CLS:PRINT TAB(30);'P.C. CABLE TESTER':PRINT STRING$(79,223)
150 LOCATE 3,22,1:PRINT USING'Test Completed in ###.## seconds.';TIMER-T
160 IF CNT=1 THEN LOCATE 5,28:PRINT'NO CONNECTIONS FOUND.':GOTO 190
170 LOCATE 5,1:PRINT'CONNECTION FOUND BETWEEN THE FOLLOWING INPUT AND OUTPUT
LINES (IN/OUT):'
180 FOR I=0 TO CNT-2:LOCATE (I MOD 10)+7,INT(I/10)*10+1:PRINT USING'##/##';
   I(I+1);O(I+1);:NEXT I
190 LOCATE 18,27:PRINT'Another Test (Y/N)?... ;
200 W$=INPUT$(1):W$=CHR$(ASC(W$) AND 223):IF W$<>'Y' AND W$<>'N' THEN 200
210 PRINT W$:IF W$='Y' THEN 20
220 LOCATE 18,27:PRINT'PC CABLE TESTER OFF LINE.
230 OUT 888,0:OUT 890,238:END:REM* TURN EVERYTHING OFF BEFORE ENDING
240 REM** SUBROUTINE TO CHECK EACH INPUT LINE
250 L1=7:IF NL<8 THEN L1=N2
260 MASK=192:C=1:FOR L=0 TO L1:OUT 888,(MASK+L*8+(INP(888) AND 7)):GOSUB 310:
   NEXT L:IF NL<9 THEN RETURN
270 L1=7:IF NL<16 THEN L1=N2
280 MASK=128:C=9:FOR L=0 TO L1:OUT 888,(MASK+L*8+(INP(888) AND 7)):GOSUB 310:
   NEXT L:IF NL<17 THEN RETURN
290 MASK=64:C=17:FOR L=0 TO N2:OUT 888,(MASK+L*8+(INP(888) AND 7)):GOSUB 310:
   NEXT L:RETURN
300 REM** SUBROUTINE TO CHECK FOR CONTINUITY
310 IF (INP(889) AND 128)=128 THEN RETURN
320 IF (INP(889) AND 128)<>0 THEN LOCATE 10,1:PRINT'** ERROR **':STOP
330 I(CNT)=I:O(CNT)=L+C:CNT=CNT+1:RETURN
```
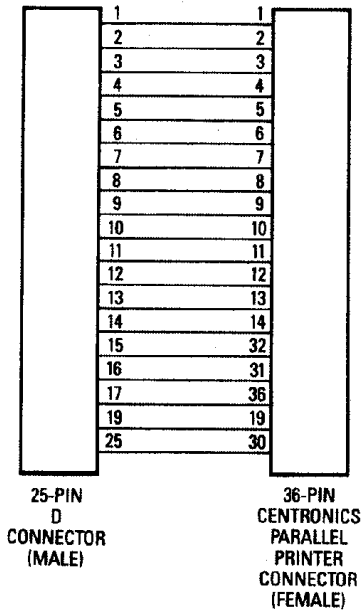
onnector, strip back the outer cable jacket about 1½ inches, exposing the 15 wires. Then strip ⅛ inch of insulation from 13 of the wires, and cut off the remaining two wires. Tin each of the exposed ends. Repeat the process for the other end of the cable, but strip 4 inches of outer jacket and ⅜ of an inch of insulation from each wire. The wires on the connector end are soldered to the 13 pins of P1 as shown in Fig. 4. The wires on the other end are inserted into the appropriate points on the breadboard.

Cut forty-eight 4-inch lengths of no. 22 solid wire, stripping ⅛ of an inch of insulation from one end and ⅜ of an inch of insulation from the other. Solder the ⅛-inch stripped end of a wire to pin 1 of J1. Repeat that process for pins 2–24 of J1 and pins 1–24 of J2 (J1 and J2 are standard 25-pin "D" connectors). Make a bracket out of a piece of aluminum and then mount J1 and J2 in the bracket. Finally, insert the free end of each of the forty eight wires into the appropriate points on the breadboard.

The three AA batteries are housed in a standard battery holder. You can use a power switch, but it's just as easy to insert or remove the +V battery wire to the breadboard. The breadboard, batteries and bracket can be mounted on a plywood base.

To test a cable without 25-pin D connectors, make a suitable adapter. One end of the adapter will be a 25 pin "D" plug. The other end will have a mating connector for the cable to be tested. These two connectors are then attached. An adapter for a 36-pin Centronics connector is shown in Fig. 5.

Operation

Type in and save the BASIC program shown in Listing 1 (or download PCCABLE. BAS from the RE-BBS—516-293-3000). Attach the cable tester to your PC's printer port and then run the program. Connect the cable to be tested to the input and output jacks of the tester. When the program asks *How many wires in the cable (1–24)?*, enter the appropriate number; if you're not sure, enter 24. The program will then check all possible combinations of lines and give you a list of connections found.

Each item in the list is of the form 1/1, meaning that pin 1 of the connector that is attached to the tester's input is connected to pin 1 of the other connector at the tester's output. On a standard 4.77-MHz PC, it takes about 10 seconds to test all 24 lines in all
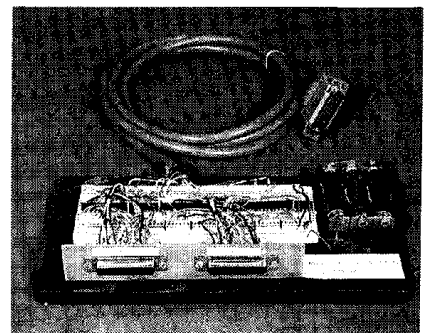

Fig. 6. The completed cable tester. Construction is not critical, and a prototyping board should give good results.

576 combinations; testing fewer lines decreases the test time. Compiling the program reduces test time considerably.▶CD◀

LISTING 1

```
1 REM** PC Cable Tester Program
2 REM** Name: PCCABLE
3 REM** (cl 1987 J.J. Barbello
4 REM** V8971017
10 DEF SEG=64:KEY 0FF:DEFINT 1,0:DIM 1(576),0(576)
20 CLS:CNT=1:PRINT TAB(3O):'P.C. CABLE TESTER':PRINT STRING$(79,223)
30 LOCATE 4,15:PRINT STRING$(50,32):LOCATE 4,15:LINE INPUT'How many wires in
   the cable (1-24)?...';NL$:NL=VAL(NL$)
40 IF NL<1 OR NL>24 THEN 20
50 N1=INT(NL/8):N2=NL MOD 8:IF N2=0 THEN N2=7 ELSE N2=N2-1
60 LOCATE 6,18,0:PRINT'Checking for Connections on Line:'
QO T=TIMER
100 FOR I=1 TO NL
110 IF I=1 THEN OUT 890.5 ELSE IF I=9 THEN OUT 890.6 ELSE IF I=17 THEN OUT 890.0
120 OUT 888,(I-1) AND 7:LOCATE 6,52:PRINT USING'##';I
130 GOSUB 250:NEXT I
140 CLS:PRINT TAB(30);'P.C. CABLE TESTER':PRINT STRING$(79,223)
150 LOCATE 3,22,1:PRINT USING'Test Completed in ###.## seconds.';TIMER-T
160 IF CNT-1 THEN LOCATE 5,28:PRINT'NO CONNECTIONS FOUND.':GOTO 190
170 LOCATE 5,1:PRINT'CONNECTION FOUND BETWEEN THE FOLLOWING INPUT AND OUTPUT
LINES (IN/OUT):'
180 FOR I=0 TO CNT-2:LOCATE (1 MOD 10)+7,INT(I/10)*10+1:PRINT USING'##/##';
   I(I+1);O(I+1)::NEXT I
190 LOCATE 18,27:PRINT'Another Test (Y/N)?...;
200 W$=INPUT$(1):W$=CHR$(ASC(W$) AND 223):IF W$<>'Y' AND W$<>'N' THEN 200
210 PRINT W$:IF W$='Y' THEN 20
220 LOCATE 18,27:PRINT'PC CABLE TESTER OFF LINE.
230 OUT 888,0:OUT 890,238:END:REM* TURN EVERYTHING OFF BEFORE ENDINQ
240 REM** SUBROUTINE TO CHECK EACX INPUT LINE
250 L1=7:IF NL<8 THEN L1=N2
260 MASK=192:C=1:FOR L=O TO L1:OUT 888,(MASK+L*8+(INP(888) AND 7)):GOSUB 310:
   NEXT L:IF NL<9 THEN RETURN
270 L1=7:IF NL<16 THEN L1=N2
280 MASK=128:C=9:FOR L=0 TO L1:OUT 888,(MASK+L*8+(INP(888) AND 7)):GOSUB 310:
   NEXT L:IF NL<17 THEN RETURN
290 MASK=64:C=17:FOR L-O TO N2:OUT 888,(MASK+L*8+(INP(888) AND 7)):GOSUB 310:
   NEXT L:RETURN
300 REM** SUBROUTINE TO CHECK FOR CONTINUITY
310 IF (INP(889) AND 128)=128 THEN RETURN
320 IF (INP(889) AYD 128)<>0 THEN LOCATE 10,1:PRINT'** ERROR ● *':STOP
330 I(CNT)=I:O(CNT)=L+C:CNT=CNT+1:RETURN
```