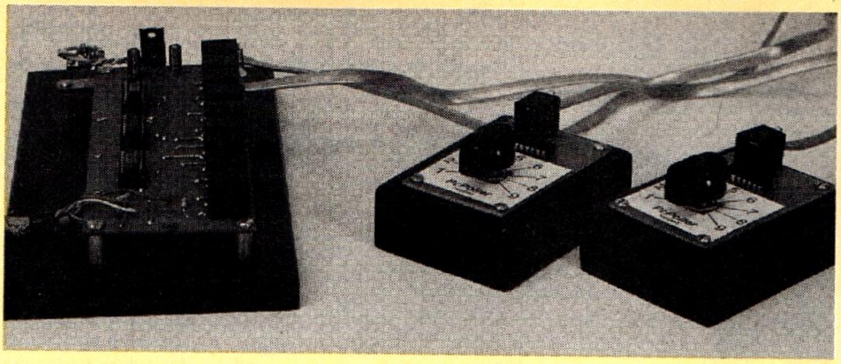
BY JAMES J. BARBARELLO



BUILD THE PC POLLER

Use it and your PC to poll the responses from several switches at once.

Anyone who watches audience-participation shows like *America's Funniest Home Videos* knows that there are instances where group responses need to be captured and analyzed. For example, there has to be some way for the audience to vote on which video is really the funniest.

Also, off the air, ad agencies routinely bring in groups of consumers to view potential commercials. During such screenings, each consumer continuously indicates his or her interest by adjusting a control that ranges from "bored" to "wow!" Such a system can also be useful in a classroom setting, where students can listen to a series of multiple-choice questions and provide their responses.

But no matter what application such a polling system is used for, it works in the following way: It immediately monitors the status (open or closed) of several switches, captures the responses, and logs them into a computer, where various analyses can be performed. When looked at that way, it becomes clear that a poller could even be useful in security or manufacturing operations.

The *PC Poller* described in this article is just such a versatile device. It allows you to poll the status of up to 36 keypad stations. Each keypad station can have up to nine switches, totaling 324 possible switches, which are all polled and sensed through any PC's parallel port. From there, the received data is interpreted using a simple QBasic program, which could be easily adaptable to almost any application.

Of course, you can't get up to 324 on-off indications through a 25-pin parallel port simultaneously. You can, however, sequentially scan the status of each individual switch very rapidly with a PC. So rapidly, in fact, that the scan can take less than one second on a fast computer. Let's see how.

Circuit Description. The schematic for the Poller control-board circuit is shown in Fig. 1. As shown in that diagram, the circuit will act as the first controller board. If you would like to connect more boards (remember, you can connect up to three more for a total of four), the subsequent circuits will be the same as the first except for the following changes:

For starters, the power-supply components, J10, C1, C2, and IC4 are only needed in the first board. Subsequent boards take their power from the first board.

Second, resistor R1 is only needed in the first board. In the other control circuits, the connection established by the resistor is not needed at all.

Finally, the DB-25 plug (PL1) that connects the Poller to a PC is only needed on the first board. The subsequent boards share the connections indicated in Fig. 1.

Now, back to how the circuit works. Jack J10 is connected to an 8- to 13-volt DC source. Two filter capacitors, C1 and C2, and a 7805 regulator, IC4, provide the circuit with a regulated 5-volt supply.

At the heart of the circuit is a 4017 decade counter, IC1. That chip has ten outputs, 0 through 9 (only 1–9 are actually electrically connected to the circuit). On each positive transition of the clock input (CLK) at pin 14, the current output (we'll call it "n") goes low, and the next output (n + 1) goes high. That occurs only if the clock-enable (EN) input, pin 13, is low. If the EN input is high, the CLK input has no effect. Finally, a high at reset-pin 15 resets the count.

Each of IC1's outputs, 1–9, is inverted by a section of a 4049 hex inverter. Because nine inverters are needed, two 4049s, IC2 and IC3, are used.

To understand how the inputs are detected by the controller board, let's take a look at the keypad circuit, which is shown in Fig. 2. That circuit contains a single 4017, IC1, which also has its 0 output unconnected. The remaining nine outputs are each connected to one terminal of rotary-switch S1. The pole of the switch is connected through diode D1 to a common point in jack J1, a six-contact modular jack. The same type of jack is used for J1–J9 on the control board.

For the circuits to work together, they need to be controlled by a PC, which is connected by PL1 of Fig. 1. We'll look at the software that controls the Poller later, but for now, we'll just stick to circuit functions. Initially, a master reset signal from pin 1 of PL1 resets all the 4017s in both the control and keypad boards to the 0 output. Because IC1 of the control board has its 0 output high, all other outputs are low. Those low outputs are inverted to highs by IC2 and IC3, which disables the 4017s on all the keypad boards (a high EN disables the 4017).

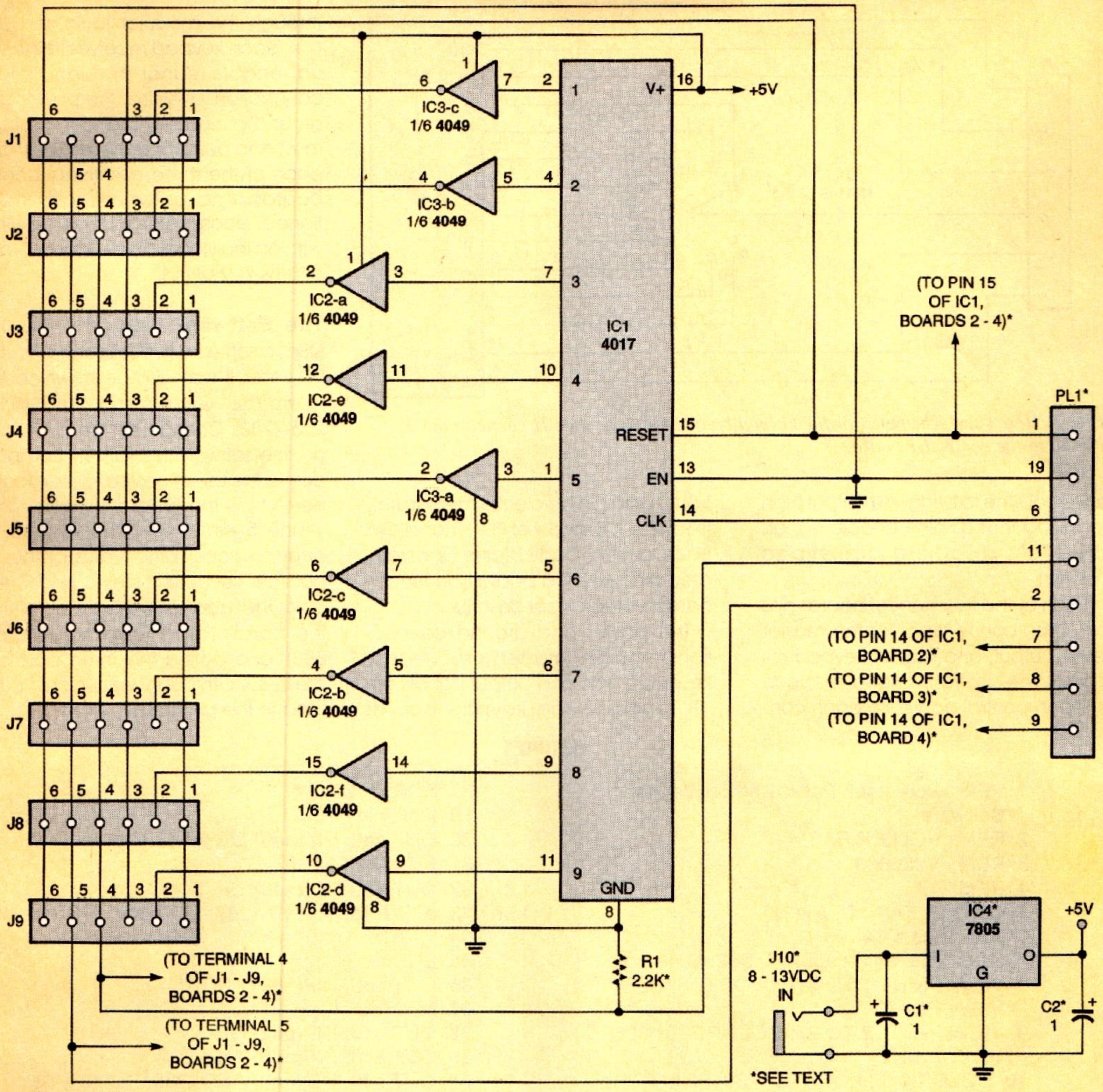


Fig. 1. This is the schematic for the PC Poller controller board. Although you can have up to four of these circuit boards working together, power-supply-components J10, C1, C2, and IC4; resistor R1; and plug PL1 are only needed in the first board.

Next, a control CLK signal clocks IC1 of the control board, bringing the 0 output low and the 1 output high. That is inverted in IC3-c, thereby enabling IC1 on the first keypad board. Then a keypad CLK signal begins clocking IC1 on the first keypad board (any other keypad-board 4017s are disabled, and therefore do not respond to the keypad clock).

Notice that in Fig. 2, the numbers on each setting of the rotary switch match the output numbers of IC1. Therefore, as IC1 of the keypad is clocked through outputs 1 through 9 by the CLK signal, the switch will pass the appropriate high level output through D1 to J1. In other words, if "3" is selected, the high level from output 3 will be passed on to the jack, and to the control board.

Back on the first control board (Fig. 1), resistor R1 acts as part of an OR gate (the diode on the keypad is the other part) to pass along the high level, while isolating the low levels from all other keypads. The software keeps track of which switch on which keypad board has been closed each time the software sees a high output.

Once the first keypad board has been clocked through all nine outputs, it is clocked one more time to bring the 0 output high again, completing the cycle and ensuring that none of the outputs connected to the keypads are high. Then IC1 on the control board is clocked, bringing its 1 output low and 2 output high. That disables the first keypad board and enables the second. The complete sequence is thus repeated for all nine keypad boards controlled by the first control board. As with the keypad boards, the control board is clocked

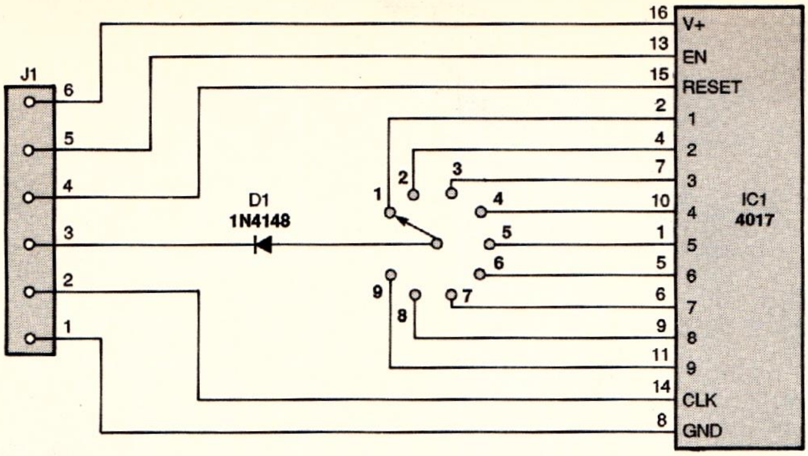


Fig. 2. Here's the schematic for the keypad board. Modular-jack J1 connects the circuit to the controller board.

one final time to bring its 0 output high (ensuring that all nine enable signals are high, disabling all keypad boards).

Each of the four control boards has common connections for the master reset, output, and master keypad CLK signals. The control CLK signal is unique to each control board, so each control board can be clocked separately (see pins 7, 8, and 9 of PL1). That allows each control board (and its associated nine keypad boards), in turn, to be activated as just described.

The power and ground connections, and the master reset, master keypad clock, and output signals are all bused to the keypad boards through the modular jacks. In addition, each keypad receives its individual enable signal through its jack connection. Likewise, the power and ground connections, and the master reset and output signals, are bused to each of the three additional control boards. And as we already mentioned, each control board is wired with an individual clock signal on one of pins 7–9 of PL1.

The Software. Listing 1 shows the QBasic software that is used to run the PC Poller. It can also be downloaded from the Gernsback BBS (516-293-2283). Considering what it has to do, the software is relatively compact due to the use of nested FOR-NEXT loops. Here's how the program works:

Line 5 dimensions the keypad (k) array to accept the status of the 36 possible keypads. Lines 6 and 7 enable the address space containing the parallel port (decimal address 888), and define the three port addresses AD0 through AD2, which will be used in the program.

LISTING 1

```
1 REM: Pc Person Poller Demonstration
  Software
2 REM: POLLER.BAS
3 REM: V950605
4 REM:
5 CLS : CLEAR : DIM k(36)
6 DEF SEG = 64
7 ad0 = 888: ad1 = ad0 + 1: ad2 = ad1 + 1
8 VIEW PRINT 1 TO 1: COLOR 0, 6: CLS :
  VIEW PRINT
9 VIEW PRINT 2 TO 25: COLOR 7, 0: CLS :
  VIEW PRINT
10 COLOR 0, 6: LOCATE 1, 30: PRINT " Pc
   PERSON POLLER "; : COLOR 7, 0
11 LOCATE 3, 20: PRINT "Press ANY key to
   repeat scan, ESC to end."
12 LOCATE 5, 32: PRINT "K E Y P A D S"
13 LOCATE 7, 4
14 FOR i = 1 TO 36
15 IF LEN(STR$(i)) > 2 THEN
16 PRINT MID$(STR$(i), 2, 1); " ";
17 ELSE
18 PRINT "0 ";
19 END IF
20 NEXT i
21 LOCATE 8, 4
22 FOR i = 1 TO 36: PRINT RIGHT$(STR$(i),
   1);    " "; : NEXT i
23 LOCATE 9, 4
24 FOR i = 1 TO 36: PRINT "- "; : NEXT i
25 grandloop:
26 REM: Pulse Reset Pin Hi/Lo
27 OUT ad2, 0: OUT ad2, 1
28 FOR i = 1 TO 4
29 FOR j = 1 TO 9
30 OUT ad0, INP(ad0) OR 2 ^ (3 + i)
31 FOR k = 1 TO 9
32 OUT ad0, INP(ad0) OR 1
33 IF (INP(ad1) AND 128) = 0 THEN
34 k(j + (i - 1) * 9) = k
35 END IF
36 OUT ad0, INP(ad0) AND 254
37 NEXT k
38 OUT ad0, INP(ad0) OR 1
39 OUT ad0, INP(ad0) AND 254
40 OUT ad0, INP(ad0) AND (255 - 2 ^ (3 + i))
41 NEXT j
42 OUT ad0, INP(ad0) OR 2 ^ (3 + i)
43 OUT ad0, INP(ad0) AND (255 - 2 ^ (3 + i))
44 NEXT i
45 LOCATE 10, 4
46 FOR i = 1 TO 36
47 IF k(i) > 0 THEN
48 PRINT USING "# "; k(i);
49 ELSE
50 PRINT "  ";
51 END IF
52 NEXT i
53 again:
54 a$ = INKEY$: IF a$ = "" THEN GOTO again
55 IF ASC(a$) <> 27 THEN
56 FOR i = 1 TO 36: k(i) = 0: NEXT i
57 GOTO grandloop
58 END IF
59 END
```

Lines 8 through 24 format the screen. The actual operation begins at line 25 (GRANDLOOP). Line 27 performs a master reset, pulsing pin 1 of the parallel port high and then low again. The actual polling and data capture is done in lines 28 through 44. Three nested FOR-NEXT loops are used to address each control board (i), each keypad for the given control board (j), and each switch on a given keypad (k).

The master reset signal is output from pin 1 of the parallel port (AD2, or 890 decimal). That output is provided to parallel port pin 11 (AD1, or 889 decimal). The keypad clock and control clocks are output respectively from parallel-port pins 2, 6, 7, 8, and 9 (all located at AD0, or 888 decimal). Pin 6 of the parallel port is the clock for the first control board, while parallel-port pins 7, 8, and 9 are the clocks for control boards 2, 3, and 4. In order to understand how the OUT commands are being used in this application, we need to review the bit patterns for address AD0.

Address AD0 contains eight binary bits, with the least significant bit (2^0, or 1) at pin 2, the next bit (2^1, or 2) at pin 3, and so on with the most significant bit (2^7, or 128) at pin 9. That is graphically represented in Table 1.

To clock the keyboards, we need to make bit 0 high and then low, without disturbing bits 4 through 7 (the control board clocks). Similarly, to clock a particular control board, we need to make its bit (4, 5, 6, or 7) high and then low, without disturbing bit 0. Referring to line 30, we first get the current status of AD0 (INP(AD0)) and then OR it with $2^{(3+i)}$. Note that i corresponds to the control board number, so an i = 1 is the first control board, and so on. So, we will OR the current status of AD0 with bit 4 (2^4) for control board one, bit 5 (2^5) for control board two, and so on. That has the effect of leaving all other bits undisturbed, but making the selected bit a 1.

Line 40 does the opposite. It ANDs the current status of AD0 with 255-$2^{(3+i)}$. That has the effect of keeping all other bits undisturbed, but making the selected bit a zero. So, the FOR-NEXT i loop allows us to calculate the proper bit to be toggled.

The actual clocking is done within the FOR-NEXT j loop (note that lines 30 and 40 are actually part of the j loop, although their calculations rely on the value of the i loop). That clocking determines which keypad will be selected for a given control board. Once the keypad has been selected, the FOR-NEXT k loop takes over. Line 32 ORs the current status of AD0 with a 1, bringing the master keypad CLK line (pin 2) high. Line 33 then checks to see if the output line (pin 11) is high. If it is, line 34 is executed, storing the number of the activated keypad switch. If pin 11 is low, no storage takes place. Line 36 then brings pin 2 low again, completing the master keypad CLK cycle.

When the FOR-NEXT k loop has been completed, all 9 positions have been checked. Lines 38 and 39 clock the keypad once more to bring its 4017 to the "0" output. When all nine keypads of a given control board have been scanned (the FOR-NEXT j loop), lines 42 and 43 clock the control board's 4017 once more to bring it to the "0" output. That sequence is repeated four times in the FOR-NEXT i loop, once for each control board.

Lines 45 through 52 display the results stored in the 36 k-array elements. If no keyclosures were detected, a blank is displayed (line 50). Otherwise, the switch number is displayed (line 48). Finally, lines 53 through 57 wait for any key press before repeating the scanning process. While complicated-looking, the process takes only a second or so on a 486DX-33MHz PC.

TABLE 1

Bit	Pin Number	Decimal Value
7	9	128
6	8	64
5	7	32
4	6	16
3	5	8
2	4	4
1	3	2
0	2	1

Construction. While any project-building method, even point-to-point wiring, can be used to build the Poller, the sheer volume of interconnections makes using printed-circuit boards almost a necessity. Figures 3 and 4 show the solder and component templates for a double-sided control board. The template for the keypad PC board is

PARTS LIST FOR THE PC POLLER CONTROLLER BOARD (Fig. 1)

SEMICONDUCTORS

IC1—4017 CMOS decade counter, integrated circuit
IC2, IC3—4049 hex inverter, integrated circuit
IC4—7805 5-volt regulator, integrated circuit (first board only)

ADDITIONAL PARTS AND MATERIALS

R1—2200-ohm, ¼-watt, 5% resistor (first board only)
C1, C2—1-µF, 16-WVDC, electrolytic capacitor (first board only)
J1–J9—RJ11 6-contact modular jack
J10—Power jack (should match AC adapter that is used; first board only)
PL1—DB-25 plug (first board only)
Printed-circuit materials, suitable enclosure, nine-conductor cable, 8- to 13-volt DC source, wire, solder, hardware, etc.

PARTS LIST FOR THE PC-POLLER KEYPAD BOARD (Fig. 2)

IC1—4017 CMOS decade counter, integrated circuit
D1—1N4148 silicon diode
J1—RJ11 6-contact modular jack
S1—12-position rotary switch (only 9 positions used)
Printed-circuit materials, suitable enclosure, 6-conductor cable with RJ-11 6-contact modular plugs on each end, knob for rotary switch, wire, solder, hardware, etc.

Note: The following are available from James J. Barbarello (817 Tennent Road, Manalapan, NJ 07726; Tel. 908-536-5499): RJ11 modular jack (RJ11J)—$1.00; RJ11 cable with RJ11 plug terminations on both ends (RJ11C)—$0.25 per foot (specify length); enhanced software (both source and executable code) on 3.5-inch disk (PCP-S)—$12.00. U.S. Funds only please. All prices include shipping and handling. Write or call for price and availability on Control and Keypad PC Boards, and complete kits. The author will cheerfully respond to any questions accompanied by a self-addressed, stamped envelope. Phone calls will be accepted between 6:00 P.M. and 8:00 P.M. EST Monday through Friday.

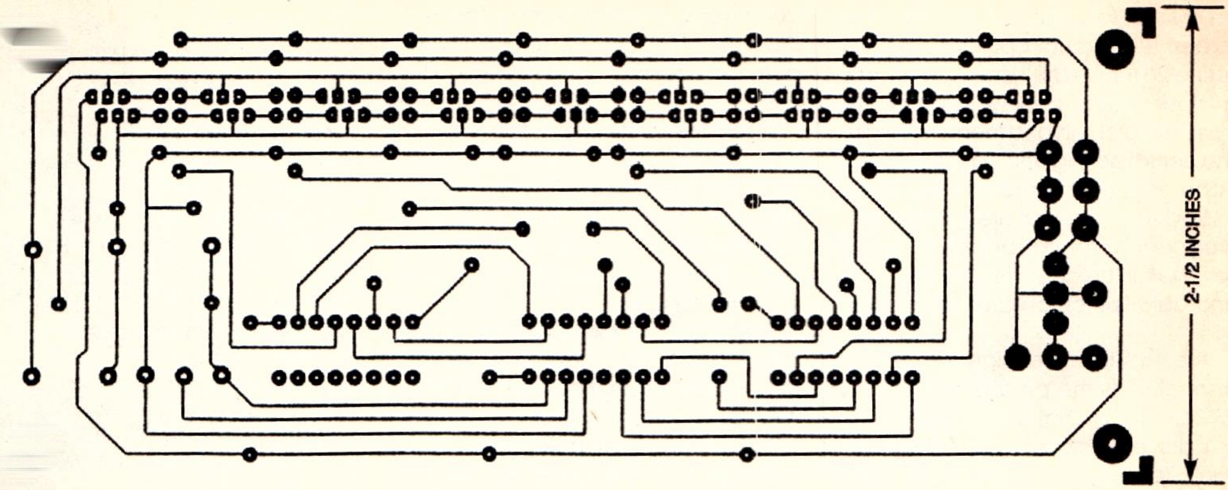


Fig. 3. The solder side of the controller board is shown here full size.

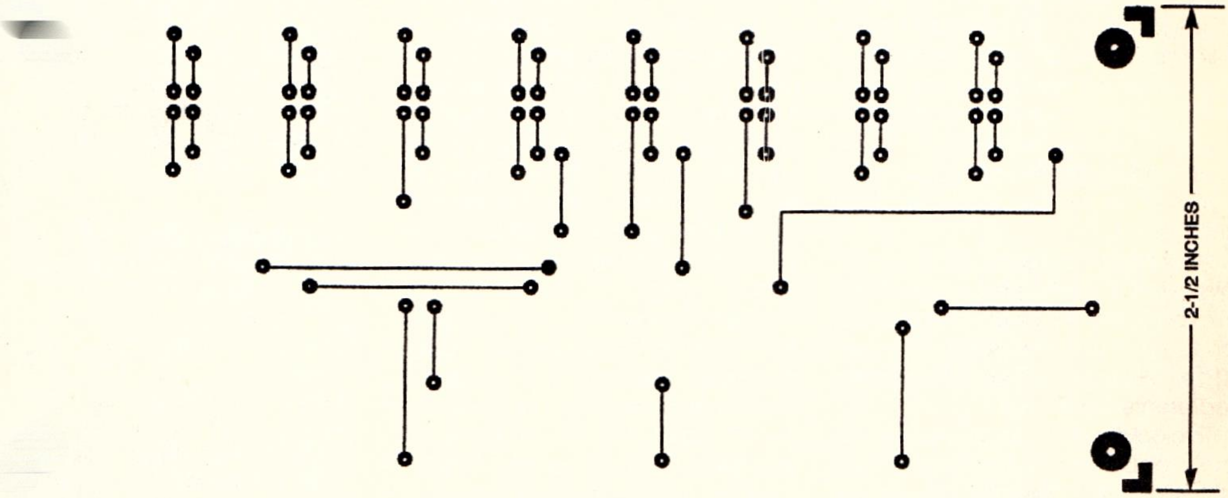


Fig. 4. This is the component side of the controller board.

shown in Fig. 5. You can either make your own boards or contact the author about the availability of etched and drilled boards (see the Parts List).

Etch or obtain the appropriate number of control boards. If you are using the double-sided PC board, the parts-placement diagram shown in Fig. 6 should make building the project easier.

Let's take a look at how to build the first control board. Begin by installing all the feedthroughs that are marked with an "X." Use little bits of wire, pass them through the holes, and solder them on both sides.

Next, mount 16-pin sockets for the three ICs. Go on to install resistor R1, and power-supply components J10, C1, C2, and IC4 (note that J10 is an off-board component).

For the first control board you will also need to make a 9-conductor cable with a DB-25 plug (PL1) on one end. As shown in Fig. 6, attach the nine wires to pins 1, 2, 6, 7, 8, 9, 11, and 19 of PL1. Then, attach the wires from pins 1, 2, 6, 11, and 19 to the appropriate pads, as shown in Fig. 6.

Continuing on the first board, mount RJ11 jacks J1–J9. Remember, those are six-contact jacks, not the more common four-contact versions used for standard telephone interconnection. The RJ11 modular jacks each have two snap-in posts protruding past their bottom surfaces. Snip off those posts flush with the jack bottoms before mounting the jacks. Position the jacks as close to the PC board surface as possible, and solder them into place.

If you're only using one control board, you only have to insert the ICs into their sockets to complete assembly. To add other control boards, you

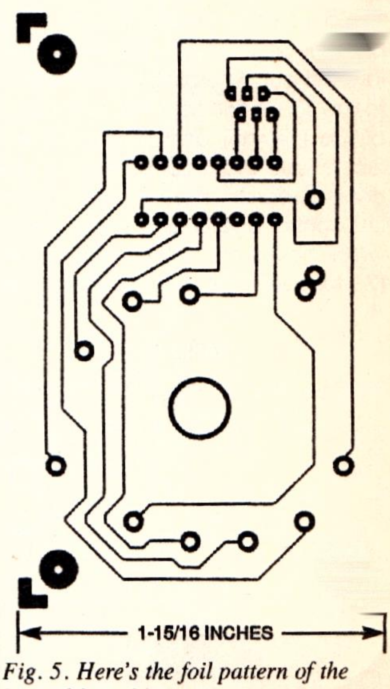


Fig. 5. Here's the foil pattern of the keypad board.

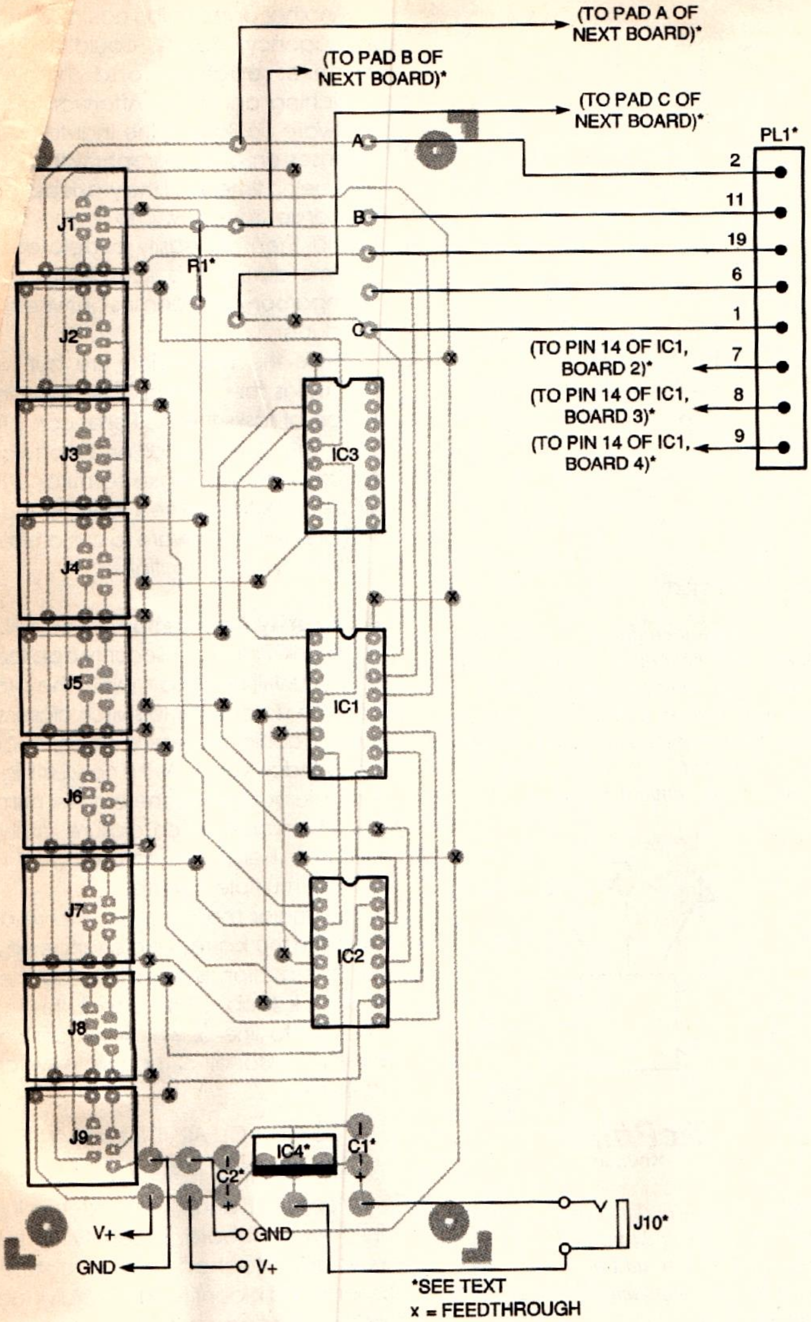


Fig. 6. Use this parts-placement diagram as a guide when building the controller board. If you're using more than one board, you will need to attach them using the various connection points indicated. Also note the points marked with an X; those are feedthroughs that need to be soldered on both sides (use little bits of wire).

will need to build each of them a little differently than the first one, and add some interconnections. Here's how to do all of that:

First, begin assembling each of the additional boards by installing the feedthroughs. Then, solder the IC sockets to the board. Mount the jacks next, being sure to cut the posts on them as you did earlier. Then insert the ICs and you're ready to interconnect the boards.

First of all, as you'll note in Fig. 6, the connections to pins 7–9 of the DB-25 plug each go to one of the additional boards. Connect those wires to the pads on the other boards where pin 6 would connect on the first board (because only one DB-25 plug is used, the pads on the other boards are free). That will establish a connection to pin 14 of each IC1 on each board.

As you'll also find in Fig. 6, three of the other pads that go to the DB-25 on the first board are labeled "A," "B," and "C." The pads located to the left of them are marked "To pad A of next board" and so on. On the first board, run wires from the latter pads to pads A, B, and C on the second board. Then, do the same from the second board to the third board, effectively daisy-chaining the boards together.

You will also have to make similar connections between boards for power. Note there are two V+ and GND connection pairs on each board. That enables you to connect the boards together to share the regulated 5 volts produced by IC4 of the first board. Use wires to attach one pair of V+ and GND connections to a pair on the second board. Then use wires to connect the free pair on the second board to the third board, etc. Following that technique means that the first and last board will each have an unused pair of power connections.

To complete all the boards, mount them in suitable enclosures. The author's prototype was mounted on a block of wood. If you're using more than one controller board, you might want to mount them in a row on a long piece of plywood to keep their interconnections from being strained by too much physical movement.

With the control board(s) completed, proceed to make the keypad boards. If you're using the PC-board pattern shown in Fig. 5, then assembly of each keypad will be simple. Just use the parts-placement diagram shown in Fig. 7.

Each of the author's prototype keypads uses 9 positions of a 12-position rotary switch. To make it easy to see which number the switch is set to, you might want to copy the full-size decal shown in Fig. 8 and apply it to the component side of the keypad PC board. Then begin assembly by installing a 16-pin socket for IC1.

Cut the posts on the RJ11 and mount it. Solder diode D1 to the board next. Then, from the component side of the board, pass the shaft of S1 through the large hole in the PC board. Secure the shaft with its hex nut. Next, use short wires to make the electrical connections to the switch on the solder side of the board. To complete each keypad circuit, add a knob to the switch and insert the IC.

When you're finished with the board, mount it in a suitable case. You will then have to fabricate an appropriate length (up to 25-feet long) six-contact cable for each keypad board. Attach an RJ11 plug to both ends of each cable.

Once all the control and keypad boards are complete, connect them together using the RJ11 cables. Then, connect a power source (supplying 8- to 13-volts DC) to power-jack J10 on the first control board.

Checkout. You can use the Poller with the software described earlier, or with advanced software available from the source in the Parts List. Once the software is in entered or loaded into your computer, connect PL1 to the PC's parallel port.

When you run the program, you will see a screen display that indicates which keypads are connected. The positions of the rotary switches will also be displayed. If you press any key, you can initiate another scan (or you can press the Esc key to end the program).

If, when you run the program, you do not get any indication for any connected keypad (even after resetting each rotary switch to various positions), check the parallel-port address of your PC. In some instances, the address might be different from the standard 888 decimal (378 Hex). Check your PC manuals for the actual parallel-port address, and substitute it for the 888 decimal in line 7 of the program. For instance, if your manual states your printer port is located at 3A0 Hex, change the first equality in line 7 to "ad0 = &H3A0" (without the quotes, of course).

Reconnect a keypad to each available jack on the control board, press Enter on the PC keyboard, and note the proper indication on the screen. If an individual keypad does not register a position, you might have installed the control knob so that it's pointing to the wrong position. Turn the knob and press Enter on the PC keyboard until you get an indication on the screen. Then adjust the position of the control knob to correspond with the screen indication. Likewise, if the keypad indicates a position, but it is inconsistent with the control knob, adjust the knob position to correspond with the screen indication.

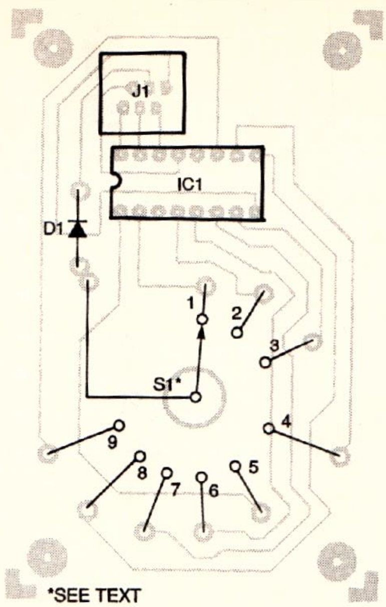


Fig. 7. As you could guess from this parts-placement diagram, the low parts count of the keypad board makes assembling it easy. Mount the rotary switch on the solder side using short wires, and put its shaft through the board to the component side.

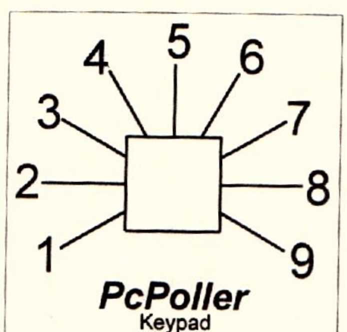


Fig. 8. If you use the keypad PC board and switch specified in the Parts List, you might want to add this actual-size decal to the component side of each of your keypads.

Program Modifications. You can expand the simple demonstration program to perform various analyses and storage tasks on the captured data. For instance, when using the system in a classroom environment, you could add a function that provides a bargraph of the responses after each scan (question) to view the response distribution. You could also compare the data for each student (keypad) against a set of desired responses. Then the software could calculate the percentage of desired responses obtained (*i.e.* a grade).

Another application could be in an ad agency. Viewers could give responses each second they are watching an event. Afterwards, the software could plot the individual responses on an x-y graph, with the x-axis being time, and the y-axis being a response from, say, 1 to 9. The data could then be easily correlated to events during the screening, and comparison of responses between individuals.

While the preceding are but two examples, they demonstrate the wide range of tasks the PC Poller can provide with some additional software programming. Those familiar with QBasic should feel free to experiment with the Poller software to customize it to their own applications.

A Security Application. If you'd like to use the Poller in a security application, you will need to modify the program to store multiple switch closures (which could represent open doors and windows, etc.). As it is now, Listing 1 is structured to store the highest numbered keypad switch closure during the scan. Here's how to change it to register multiple closures:

The major modification is to re-dimension the k array and change how the information is stored during the FOR-NEXT k loop. Make the following change to line 5, which dimensions the k array for all 324 possible switch activations:

```
5 CLS : CLEAR : DIM k(324)
```

Remember that there are three FOR-NEXT loops (i, j, and k) with i identifying the control board, j identifying the keypad attached to that control board, and k identifying the individual switch position on that keypad. So, any individual switch position can be represented by the formula:

$$((i - 1) \times 81) + ((j - 1) \times 9) + k$$

For instance, the fourth switch position on the first keypad connected to the first control board (i = 1, j = 1, k = 4) is

$$((1 - 1) \times 81) + ((1 - 1) \times 9) + 4 = 4$$

Similarly, the eighth switch position on the ninth keypad connected to the fourth control board (i = 4, j = 9, k = 8) is

$$((4 - 1) \times 81) + ((9 - 1) \times 9) + 8 = 323$$

(Continued on page 78)

PC-POLLER

(Continued from page 44)

Therefore, we can address the proper array element by changing line 34 to read

$$34\ k(((i-1)\times 81)+((j-1)\times 9)+k) = 1$$

From that formula we can readily see that as many as 81 individual switches can be scanned with just a single control board.

The remainder of the software modifications are up to you and depend on your individual application. You have to decide how you would like to store and/or display the results of the k array.

To use the security application, you will also have to make a simple hardware modification. Instead of installing a rotary switch on the keypad board, connect individual wires to the board's numbered pads and one terminal of the remote switches. The switches' other terminals should all be wired together and connected to the pad on the board that is connected to diode D1. ■