

Schools and colleges teach many things—but they don't teach electronic equipment manufacturing. Most of us pick up that type of knowledge through on-the-job experience or through our hobbies. (In fact, hobby magazines like **Radio-Electronics** probably are the most common teachers of practical design and construction.) However, many engineers graduate from school and enter the work force with little or no practical experience.

Recently I had to set up a training center to instruct young engineers at my company in basic manufacturing processes. My task was to create a small manufacturing factory where students would build an electronic product. In the process, they would experience every stage of the manufacturing process: interpreting engineering drawings, buying parts, testing them, building the product, and shipping it to the customer. The project was dubbed the Manufacturing Technology Facility (MTF).

With a limited budget and limited time in which to teach more than 400 people, I searched for automation aids that would speed up the mundane work without attenuating the manufacturing experience we were trying to impart. One area I attacked was incoming inspection. In a normal manufacturing plant, parts are bought from many sources. When they arrive, they are tested to ensure they work, because it can cost more than \$10 to find and fix a bad \$1 part in a finished product.

The product we built had more than thirty different types of electronic parts, which fell into five groups: resistors, capacitors, diodes, transistors, and IC's. What I needed was a low-cost way for students to inspect the parts quickly, but with minimum chance for error. The result was what we now call the Component Inspection System (CIS). It incorporates a capacitance meter, an IC tester, and a computer-controlled voltmeter used for testing resistors and diode and transistor junctions. The CIS software includes a database containing each component's specifications, complete with pass/fail criteria. In addition, the system

contains a data-logging function that allows us to maintain a record of each vendor's quality history, which is useful in selecting vendors.

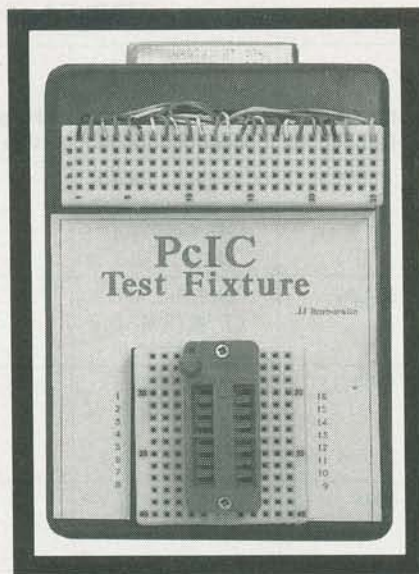
In future articles, we will describe different components of the CIS hardware and software. This time we'll present a \$15 two-IC circuit that lets you use your PC as a capacitance meter. Later installments will include complete details for building sophisticated component and IC testers. When space is available, we'll provide the software listings (all of which are in QuickBASIC); compiled programs and source code are also available.

PC-based capacitance meter

The first project is a capacitance meter. It will be incorporated into the next project, a combined voltmeter, ohmmeter, and capacitance meter on a PC board, but it can also be used as a stand-alone test instrument. First let's discuss the details of hardware operation.

The circuit, shown in Fig. 1, consists of two IC's: a 555 timer (IC1) and a quad bilateral switch (IC2). The 555 is the heart of the circuit. When configured in the astable (one-shot) mode, the length of the pulse generated at pin 3 is directly proportional to the value of the timing resistor (R1) and the timing capacitor (C_x). With a fixed timing resistor, the duration of the output pulse will be directly proportional to the value of the timing capacitor. Thus, by connecting a known resistor and an unknown capacitor to IC1, triggering it, and then measuring the length of the resultant output pulse, we can calculate the value of the capacitor.

We wanted to obtain an effective meter range of 20 pF to 20 μ F. To achieve such a wide range, we had to use two different timing resistors—and that's where the 4066 comes in. By driving pin 12 of IC2 high, the IC effectively parallels a second resistor (R2) with the main timing resistor (R1). Doing so makes it easier to measure large-value capacitors. With R2 switched in, the effective range of the meter is 0.1 μ F to 20 μ F. If the value of the unknown capacitor is less than 0.1 μ F, R2 can be switched out of the circuit (automatically by the software)



and the capacitor can then be re-measured.

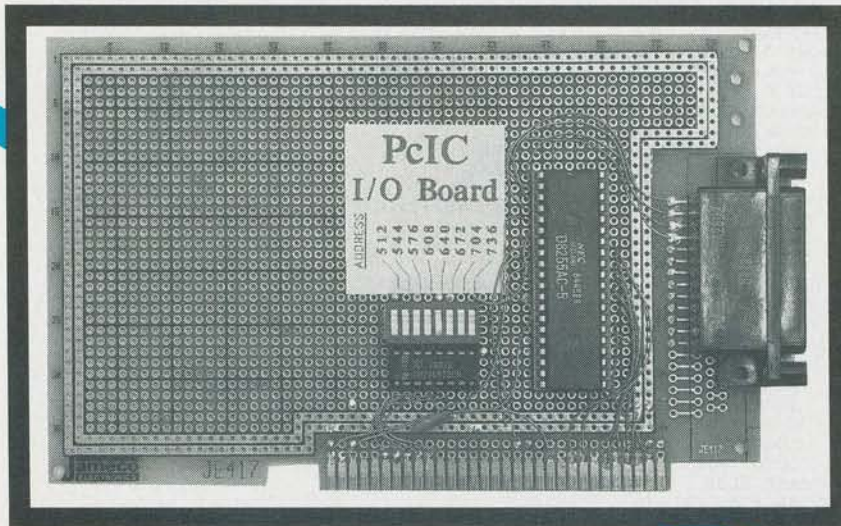
The PC connection

The capacitance meter attaches to your PC via three lines (plus ground) of a standard parallel port. The $\overline{\text{AUTOFD}}$ signal connects to IC1's TRIGGER input (pin 2) to initiate the timing cycle. The $\overline{\text{ACK}}$ line from the port connects to IC1's output (pin 3). In operation, our software drives $\overline{\text{AUTOFD}}$ momentarily low and then high again to trigger IC1. While the PC counts the time interval, the capacitor charges until IC1's internal comparator drives its output from low to high. At that time, $\overline{\text{ACK}}$ senses the end of the timing cycle and alerts the PC to stop counting. The software then converts the elapsed time interval to a capacitance value.

The D6 line from the parallel port drives the bilateral switch. Initially, the switch is on, which places R2 in parallel with R1, and allows you to measure the high range (0.1–20 μ F). If the resultant count is less than a specified value, the software switches R2 out of the circuit, obtains a new count, and then scales it for the low range (20 pF–0.1 μ F).

Because the software depends on a timing loop, it must be calibrated before use. Calibration consists of making two readings: one accounts for stray circuit capacitance; the other involves measuring a known capacitor to provide a standard. The circuit

EXPERIMENTING WITH PC-BASED TEST EQUIPMENT



must be recalibrated if the PC's operating speed changes (e.g., via a turbo mode), or if you connect the meter to another PC.

Software

The simplicity of the hardware is made possible by the software. Although we won't discuss all the details here, there are a few points you should be aware of in the event you wish to modify the software. First, there are two programs, CAP and CAPFAX; CAP is the main program and CAPFAX helps calibrate the software. Both programs are written in Microsoft's QuickBASIC. The software must be compiled; interpreters (including BASICA and GWBASIC) do not run quickly enough to measure capacitors less than 1 μ F. If you want to use the software as is, you don't need to own a copy of QuickBASIC; a compiled version is available from the author, as mentioned in the parts list. However, to modify the software, you will need a copy of QuickBASIC.

Now let's talk about CAP. Before doing anything else, the software locates the I/O address of your

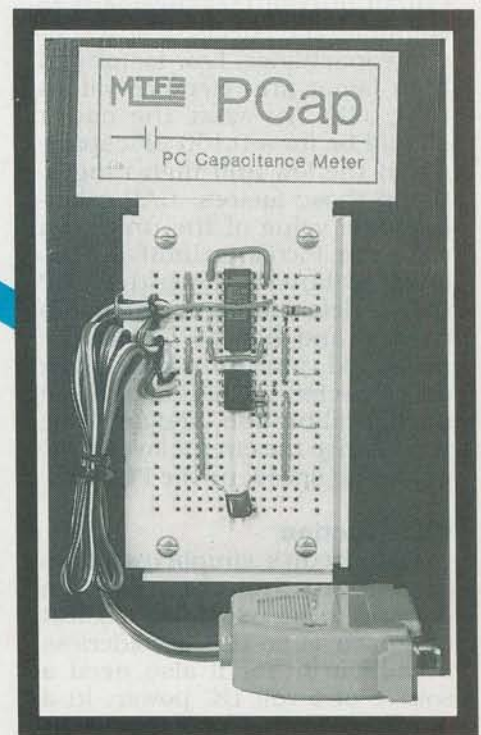
parallel printer port. That is necessary because some cards don't address their printer ports at the standard location (0378h or 888 decimal). In Listing 1, line 16 locates the address and stores it in variable C. (Note that line numbers are optional in QuickBASIC; they are shown here for reference only.) Variables B and A, which are derived from C, are used to read IC1's output and to operate IC2, respectively.

The measuring function begins in line 18, which switches in the 100K resistor. The next line initializes the counting variable, X. The next line contains two functions. The first, consisting of the OUT statements, generates a negative-going pulse to trigger IC1. The WHILE/WEND loop then continually increments X until pin 3 of IC1 goes low.

The remainder of the program determines if R2 should be switched in, converts the count to a capacitance value, and displays the value in an appropriate form (pF or μ F).

**Build your own
low-cost PC-based
test equipment**

JAMES J. BARBARELLO



You use the second program, CAPFAX, to create a data file (CAPFAX.DAT) that contains information required by CAP. (CAPFAX is shown in Listing 2.) Five values are required: zero offset, low factor, high factor, picofarad limit, and microfarad limit. Zero offset is the count obtained with no capacitor connected to the circuit. In operation, CAP subtracts this value from the count ob-

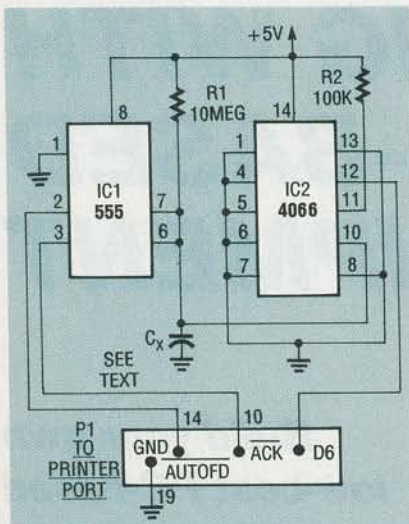


FIG. 1—SCHEMATIC DIAGRAM. The capacitance meter consists of two IC's: a 555 running in the astable mode, and an analog switch that allows the software to perform automatic range switching.

tained during measurement to eliminate the effects of stray circuit capacitance. Low factor and high factor are reference values that tell CAP what the count should be for a 0.100 μ F capacitor in the low and high ranges. Using those factors, CAP calculates the value of the unknown capacitor. Picofarad limit and microfarad limit are values that CAP uses to determine when to switch ranges, and how to format the measured value for presentation on the screen. CAPFAX is used during calibration, and any time you change the circuit layout, PC operating speed, or the PC itself.

Construction

The circuit's simplicity allows just about any construction method to be used. The easiest approach is to use a solderless breadboard. You'll also need a source of 5-volt DC power; in a pinch you could power the circuit with three batteries in series (the 4.5 volts produced should be adequate to generate the required TTL logic levels.) If you use a CMOS 555, the batteries will last a long time.

As shown in Fig. 1, several pins of IC2 must be grounded to ensure proper switching operation. If those pins are not grounded, the meter may operate erratically.

To connect the unknown capacitor, you could insert it directly into the solderless breadboard. For more convenient

LISTING 1

```

1  REM*****
2  REM**      CAP.BAS      *
3  REM**      V900114      *
4  REM*****
5  REM
6  REM** NOTE: THIS PROGRAM MUST BE COMPLIED WITH QUICKBASIC 4.0 PRIOR TO
7  USE
8  REM
9  DEF SEG = 64: DEFINT A-C, I, K: DEFLEN X
10 OPEN "r", 1, "capfax.dat", 50
11 FIELD 1, 10 AS ol$, 10 AS fl$, 10 AS fh$, 10 AS pf$, 10 AS uf$
12 GET 1, 1
13 offsetlo = VAL(ol$): FACTORLO! = VAL(fl$): FACTORHI! = VAL(fh$)
14 apf = VAL(pf$): auf = VAL(uf$)
15 CLOSE
16 FACTORLOPF# = FACTORLO! / 1000000: factorhipf# = FACTORHI! / 1000000
17 GOSUB S1: C = PEEK(8) + 256 * (PEEK(9)) + 2: B = C - 1: A = B - 1
18 RESTART:
19 OUT A, 64
20 X = 0: LOCATE 12, 31: PRINT "  MEASURING...  ";
21 OUT C, 2: OUT C, 0: WHILE (INP(B) AND 64): X = X + 1: WEND
22 SELECT CASE X
23   CASE IS > auf
24     measuredvalue = X / FACTORHI!
25     LOCATE 12, 31: PRINT SPACE$(18);
26     LOCATE 24, 1: PRINT "Hi - uf";
27     LOCATE 12, 36: PRINT USING "####.# uf"; measuredvalue
28     GOTO SKIPIT
29   CASE ELSE
30     OUT A, 0
31   END SELECT
32 OUT C, 2: X = 0: OUT C, 0: WHILE (INP(B) AND 64): X = X + 1: WEND
33 SELECT CASE X
34   CASE IS > apf
35     measuredvalue = (X - offsetlo) / FACTORLO!
36     measuredvalue = INT(measuredvalue * 10000) / 10000
37     LOCATE 12, 31: PRINT SPACE$(18);
38     LOCATE 24, 1: PRINT "Lo - uf";
39     LOCATE 12, 36: PRINT USING "#.### uf"; measuredvalue
40     GOTO SKIPIT
41   CASE ELSE
42     AA = 0: XX = 0
43   REDO:
44     OUT C, 2: X = 0: OUT C, 0: WHILE (INP(B) AND 64): X = X + 1: WEND
45     IF AA < 5 THEN AA = AA + 1: XX = X + XX: FOR i = 1 TO 100: NEXT i:
46     GOTO REDO
47     mean = XX / 5: X = mean
48     LOCATE 12, 31: PRINT SPACE$(18);
49     measuredvalue = (X - offsetlo) / FACTORLOPF#
50     LOCATE 24, 1: PRINT "Lo - pf";
51     LOCATE 12, 36: PRINT USING "#### pf"; measuredvalue
52   END SELECT
53   SKIPIT:
54   LOCATE 24, 62: PRINT SPACE$(17); : LOCATE 24, 62: PRINT "Count: "; X;
55   COLOR 7, 0: LOCATE 20, 29
56   COLOR 0, 7: PRINT " M "; : COLOR 7, 0: PRINT "easure, or ";
57   COLOR 0, 7: PRINT " E "; : COLOR 7, 0: PRINT "nd ... ";
58   Tryagain:
59   Reply$ = UCASE$(INPUT$(1))
60   SELECT CASE Reply$
61     CASE "M"
62       LOCATE 12, 37: PRINT SPACE$(6)
63       GOTO RESTART
64     CASE "E"
65       CLOSE : END
66     CASE ELSE
67       BEEP: GOTO Tryagain
68   END SELECT
69   END
70   S1:
71   CLS : LOCATE 3, 23: COLOR 0, 7: PRINT SPACE$(35)
72   LOCATE 4, 23: PRINT "      PC CAPACITANCE METER      "
73   LOCATE 5, 23: PRINT "          (c) 1990, JJ BARBARELLO      "
74   LOCATE 6, 23: PRINT SPACE$(35): COLOR 7, 0
75   LOCATE 11, 30: PRINT CHR$(222); STRING$(19, 223); CHR$(221)
76   LOCATE 12, 30: PRINT CHR$(222); LOCATE 12, 50: PRINT CHR$(221)
77   LOCATE 13, 30: PRINT CHR$(222); STRING$(19, 220); CHR$(221)
78   RETURN

```

access, you could use a pair of binding posts connected via short lengths of wire. Although doing so adds stray capacitance to the circuit, it can be canceled during calibration.

After building the circuit, wire a short cable from a standard DB-25 male connector and four

lengths of wire. Connect the appropriate pins on the connector to the circuit, and the DB-25 end to your PC's parallel port.

Calibration

Calibration must be performed prior to using the meter. To perform the calibration, you need a

LISTING 2

```

1 REM** CAPFAX.BAS
2 REM** REVISE CAPFAX.DAT ENTRIES
3 REM** V900114
4 CLS : OPEN "r", 1, "capfax.dat", 50
5 FIELD 1, 10 AS zero$, 10 AS low$, 10 AS high$, 10 AS pf$, 10 AS uf$
6 GET 1, 1
7 LOCATE 1, 30: PRINT "REVISE CAPFAX.DAT FILE"
8 LOCATE 3, 1: PRINT "Zero Offset: "; zero$
9 LOCATE 4, 1: PRINT "Low Factor: "; low$
10 LOCATE 5, 1: PRINT "High Factor: "; high$
11 LOCATE 6, 1: PRINT "pF Limit : "; pf$
12 LOCATE 7, 1: PRINT "uF Limit : "; uf$
13 LOCATE 9, 10: PRINT "Change (Y/N)? ";
14 GOSUB yesno: IF a$ = "N" THEN GOTO endit
15 z$ = zero$: l$ = low$: h$ = high$: p$ = pf$: u$ = uf$
16 getnewones:
17 VIEW PRINT 3 TO 23: CLS : VIEW PRINT
18 LOCATE 3, 1: PRINT "Zero Offset: "; z$
19 LOCATE 4, 1: PRINT "Low Factor: "; l$
20 LOCATE 5, 1: PRINT "High Factor: "; h$
21 LOCATE 6, 1: PRINT "pF Limit : "; p$
22 LOCATE 7, 1: PRINT "uF Limit : "; u$
23 LOCATE 9, 10: LINE INPUT "New Zero Offset..."; z$
24 IF z$ = "" THEN z$ = zero$ ELSE LSET zero$ = z$ + " "
25 LOCATE 9, 28: PRINT z$
26 LOCATE 10, 10: LINE INPUT "New Low Factor..."; l$
27 IF l$ = "" THEN l$ = low$ ELSE LSET low$ = l$ + " "
28 LOCATE 10, 27: PRINT l$
29 LOCATE 11, 10: LINE INPUT "New High Factor..."; h$
30 IF h$ = "" THEN h$ = high$ ELSE LSET high$ = h$ + " "
31 LOCATE 11, 28: PRINT h$
32 LOCATE 12, 10: LINE INPUT "New pF Limit....."; p$
33 IF p$ = "" THEN p$ = pf$ ELSE LSET pf$ = p$ + " "
34 LOCATE 12, 28: PRINT p$
35 LOCATE 13, 10: LINE INPUT "New uF Limit....."; u$
36 IF u$ = "" THEN u$ = uf$ ELSE LSET uf$ = u$ + " "
37 LOCATE 13, 28: PRINT u$
38 LOCATE 15, 10: PRINT "Change (Y/N)?";
39 GOSUB yesno: IF a$ = "N" THEN GOTO endit ELSE GOTO getnewones
40 endit:
41 VIEW PRINT 3 TO 23: CLS : VIEW PRINT
42 LOCATE 3, 1: PRINT "Zero Offset: "; zero$
43 LOCATE 4, 1: PRINT "Low Factor: "; low$
44 LOCATE 5, 1: PRINT "High Factor: "; high$
45 LOCATE 6, 1: PRINT "pF Limit : "; pf$
46 LOCATE 7, 1: PRINT "uF Limit : "; uf$
47 LOCATE 15, 10: PRINT "Save (Y/N)? ";
48 GOSUB yesno: IF a$ = "N" THEN CLOSE : PRINT "NO SAVE": LOCATE 18, 1:
49 END
50 PUT 1, 1: CLOSE : PRINT "New Data Saved.": LOCATE 18, 1: END
51 yesno:
52 a$ = UCASE$(INPUT$(1))
53 SELECT CASE a$
54 CASE "Y"
55 RETURN
56 CASE "N"
57 RETURN
58 CASE ELSE
59 GOTO yesno:
60 END SELECT

```

REVISE CAPFAX.DAT FILE

```

Zero Offset: 75
Low Factor : 2195000
High Factor: 25000
pF Limit   : 1000
uF Limit   : 10000

```

Change (Y/N)?

FIG. 2—CAPFAX.EXE calibrates and sets the range-switching values for the main program, CAPEXE.

0.1 μ F capacitor for which you know the exact value. If you don't have one, use a capacitor marked 0.1 μ F with the best tolerance you can find (at least 5% or 10%).

Begin by executing CAPFAX.EXE; you'll see a screen like that shown in Fig. 2, except that the five values will all be 0.0. To change values, press Y, and then

PARTS LIST

All resistors are 1/4-watt, 5%, unless otherwise noted.

R1—10 megohms

R2—100,000 ohms

Semiconductors

IC1—555 timer

IC2—4066 quad bilateral switch

Other components

P1—25-pin male D connector

Miscellaneous: 0.1 μ F high-tolerance capacitor for calibration, solderless breadboard, five-volt power source, wire, solder, etc.

Note: The following is available from JJ Barbarello, RD#3, Box 241H, Tennent Road, Manalapan, NJ 07726: Compiled version of the software (CAP and CAPFAX), with datalogging, on 5 1/4-inch double-density PC diskette, \$8.00. The author will be happy to answer any questions. Please include a self-addressed stamped envelope for reply.

enter the following values as a starting point.

- Zero Offset: 0
- Low Factor : 1
- High Factor: 1
- pF Limit : 1000
- μ F Limit : 10000

After entering those values, respond by pressing N (No) to the Change question. Respond Y (Yes) to the Save question to create the initial CAPFAX.DAT file.

Now connect the circuit to your PC's printer port, but make sure there is no capacitor connected to the test points. Apply power to the circuit and start the software by typing the command CAP at the DOS prompt.

The screen will display some value of capacitance, and a count in the lower right hand corner. Press M to measure again. Disregard the value displayed, but note the count, which should be between 5 and 100, depending on your particular PC and circuit construction). That value is the zero offset.

Next connect the known 0.1 μ F capacitor and press M to measure. Again, disregard the value displayed, but note the count. Multiply that number by 10 to obtain the low factor. For example, if the count is 123456, the low factor is 1234560. Leave the 0.1 μ F capacitor in the circuit.

Execute CAPFAX again and en-

ter the appropriate values into the Zero Offset and Low Factor fields. Also, change the μF limit to 1. Save the new data and re-execute CAP. Press M to measure. Disregard the value, but note the count, and multiply it by 10 to obtain the high factor. For example, if the count is 1234, the high factor is 12340. Execute CAPFAX one more time to enter the high factor and change the μF limit back to 10000. Save the revised data.

Use

Using the PC-based capacitance meter is straightforward. With your PC on and the meter connected to the printer port, apply power to the meter. At the DOS prompt, type CAP and press Enter. Insert the capacitor to be measured, and press M. The value appears in the middle of the screen, and the bottom indicates the range (pF, μF low, μF high), along with the timer count. To end the program, press E.

The μF limit and pF limit factors are used to determine how to format the measured value. If you measure a capacitor that displays 0.00 μF , but has a count greater than 0, you are in a "no-man's land" between the two limits. Execute CAPFAX and increase the pF limit factor. Doing so increases the pF formatting range and allows the measured value to be displayed properly.

Remember that if you vary the meter circuit or your PC, you should recalibrate the software to maintain accuracy.

Next Time

In the near future we'll incorporate the capacitance-measurement circuitry into a combination instrument that measures resistance, capacitance, and voltage. The device allows you to measure resistors, capacitors, diodes, and transistors. Because it is computer controlled, it can be the heart of an automated inspection system for your shop or business. You can save a lot of troubleshooting time by ensuring that the IC's you plan to use in a project function properly before you use them. The last project in this series is an IC tester that allows you to verify operation of most 14- and 16-pin TTL and CMOS IC's.

R-E