CONSUMER ELECTRONICS SHOW ROUNDUP

MAY 1998

Electronics NOW®

Build a

Swipe-Card Reader

Keep your secrets safe from prying eyes

Get the

Big Picture

Large-screen TVs that you can hang on a wall are finally here

Fuel-Cell-Powered Cars and Buses

Will they someday replace today's polluting vehicles?

Build a

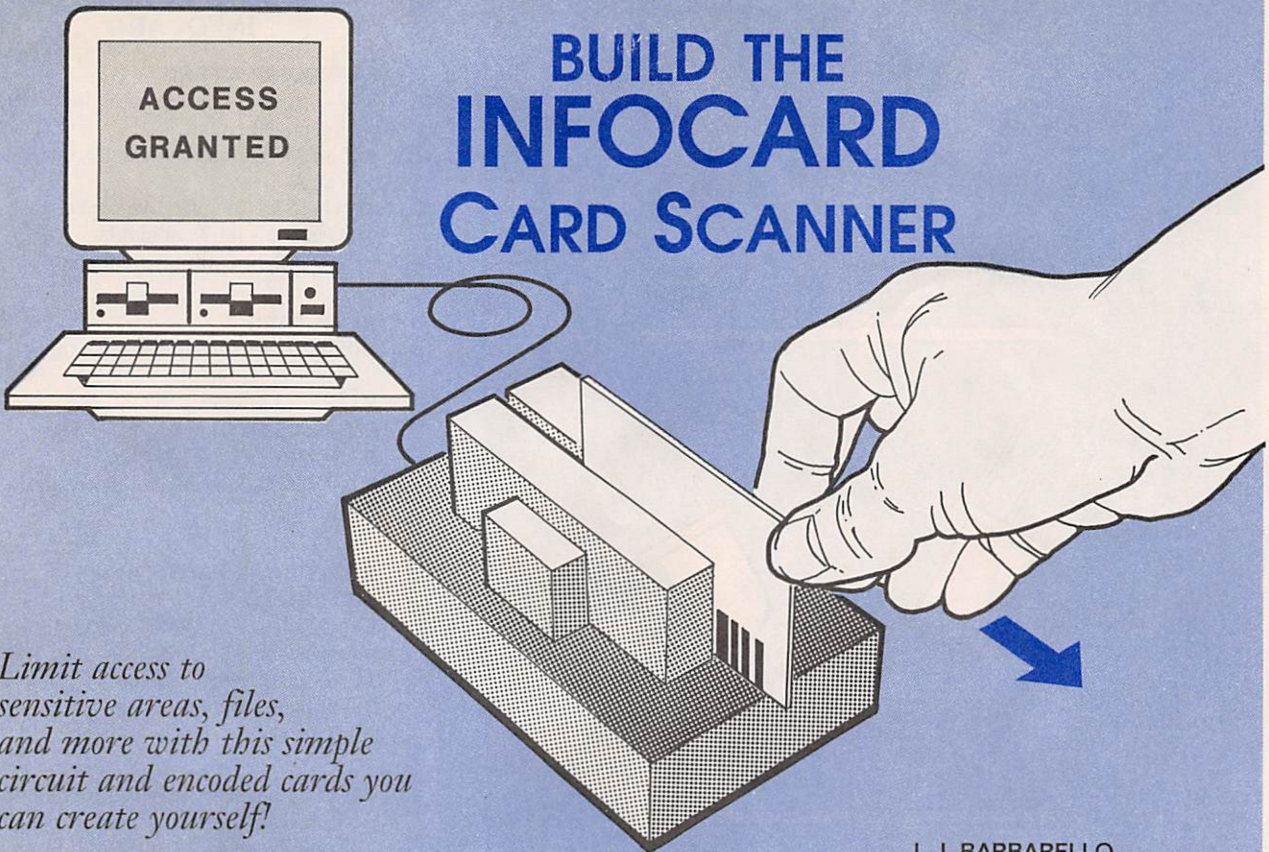
Digital Storage Oscilloscope

Turn any PC into a high-powered storage scope

A GERNSBACK PUBLICATION

$4.50 U.S.
$4.99 CAN.

BUILD THE INFOCARD CARD SCANNER



Limit access to sensitive areas, files, and more with this simple circuit and encoded cards you can create yourself!

J. J. BARBARELLO

Reflective-optical sensors are non-contact devices that use a light source and a photo-sensitive detector to detect the presence or absence of a reflective surface. To sense that type of surface, the source and detector are positioned so the light hits the surface at an angle; the reflected light is then seen by the detector. Infrared (IR) sensors produce better results than visible light sensors because IR energy in the 900-nanometer range is not normally present in large amounts in visible light. An IR sensor can be used easily where ambient light might cause false triggering of a visible-light detector.

A "reflective surface" might lead you to think of a surface such as a mirror that's either present (reflective) or absent (non-reflective). However, while paper is a good reflector of infrared light, many different types of inks (and the graphite in a lead pencil) do *not* reflect IR energy very much. With the right type of ink, we can use a reflective IR optical sensor to make a device that reads a series of different width bars separated from each other by a white space on a card. One such bar pattern is the Universal Product Code, or UPC—otherwise known as "bar" codes. However, such a code is not limited to the UPC standard. For example, we can use a basic binary code instead. A system with simple wide and narrow bars to code the ones and zeros in a binary code results in a bar code that is easy to sense and can have up to 32,767 possibilities with only 15 bars.

The InfoCard project presented here uses a low-cost reflective-IR optical sensor and a simple interface circuit to create an electronic identification system using any PC. One such use of the InfoCard is for coded-entry access by reading a card and energizing an electric door latch for authorized codes. If you are good at programming PIC chips or Basic Stamp devices, you can readily adapt the InfoCard's Basic software routines to use the reader as a stand-alone device.

The swipe cards themselves can be made quickly and cheaply from almost any available paper stock. Best of all, the project cost is about $10–$15, depending on the parts you might already have on hand.

Design Considerations. To find out how far a reflective surface has to be from the sensor in order to see the most reflected light, we can use the law of optics that says that when light hits a surface at an angle, the angle at which it hits the surface (the *angle of incidence*) is equal to the angle at which it is reflected (the *angle of reflection*). That is shown in Fig. 1A.

As you can see, the reflecting surface is at the perfect distance from the light source and sensor to pick up the greatest amount of reflected light. If the surface is closer or farther away, the amount of energy that the sensor can detect

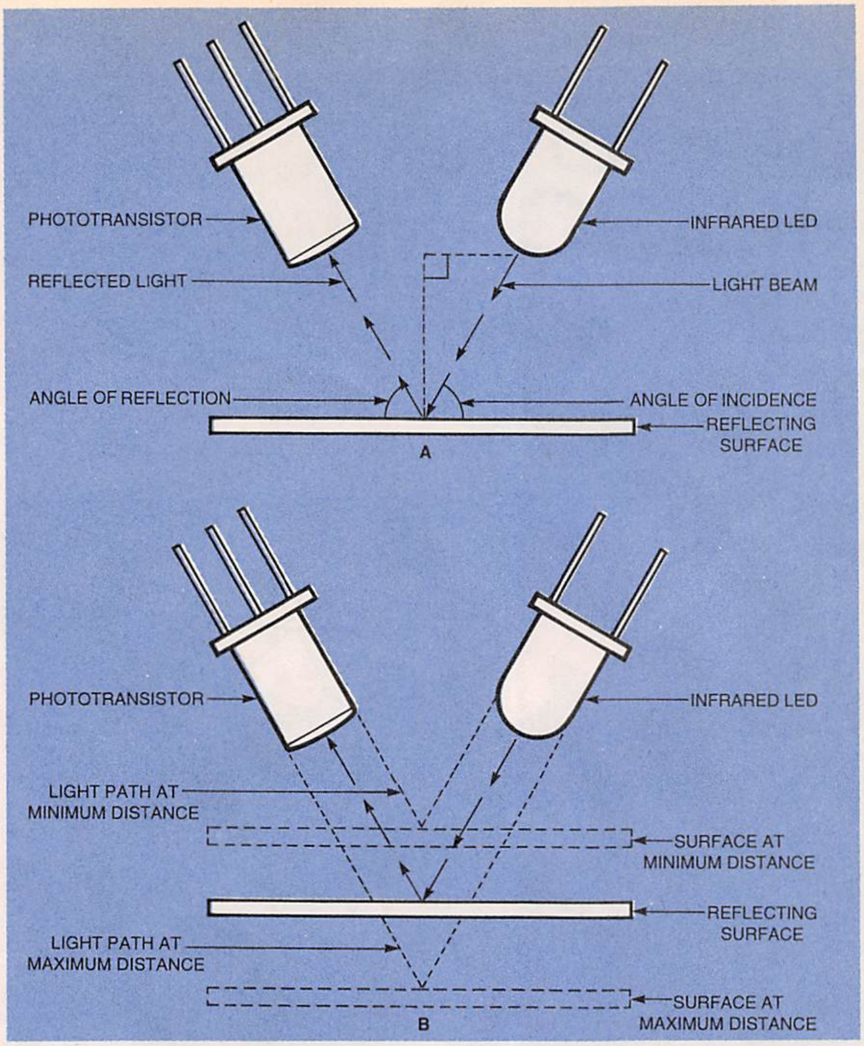


Fig. 1. If a light beam strikes a reflective surface at an angle, it will bounce off at a "mirror-image" angle (A). For best sensing, the surface must be within the area that the light source and detector are focused on (B). If the surface is too close or too far away, the light will not hit the detector.

PARTS LIST FOR THE INFOCARD

SEMICONDUCTORS

IC1—78L05 5-volt regulator, low-power, integrated circuit
IC2—LM339 quad comparator, integrated circuit
SEN1–1STS708 reflective optical switch (Jameco 138093 or similar)
LED1—Light-emitting diode, red

RESISTORS

(All units are ¼-watt, 5% units unless otherwise noted.)
R1, R5—220-ohm
R2, R8—22,000-ohm
R3, R4, R6, R9—2200-ohm
R7—10,000-ohm potentiometer, multi-turn

ADDITIONAL PARTS AND MATERIALS

C1, C2—1µF, 16-WVDC, electrolytic capacitor
J1—DB25 connector, male
9-volt battery, battery connector, hardware, etc.

NOTE: Enhanced software, part number ICARD-S (both source and executable code) is available for $12.00 from: James J. Barbarello, 817 Tennent Road, Manalapan, NJ 07726. In addition, the author will be happy to answer any questions sent to the adress above (please include a self-addressed, stamped envelope).

will drop off. That is shown in Fig. 1B. If the reflecting surface is too close or too far, the reflected light will miss the sensor completely. Based on the specifications of the reflective sensor we want to use, we can apply simple trigonometry formulas to find the perfect distance from the sensor to the reflective surface. For the device called for, that distance works out to about $^5/_{32}$ inch.

But we shouldn't stop at just the optimum distance; we should also find the minimum and maximum distances. Again, using basic trigonometry, we find that the surface should not vary more than $^1/_{32}$ inch in either direction from the $^5/_{32}$-inch distance. With those numbers, the card guide that we design should keep the card between $^1/_8$ inch and $^3/_{16}$ inch from the sensor.

One more piece of information to keep in mind when working with light has to do with the tendency of light to disperse from its source. An example of that can be seen in Fig 2A. The width of the beam will affect how narrow a barcode stripe can be seen. By adding an aperture as shown in Fig. 2B, the resolution improves. The width of the aperture must be large enough to allow enough light to be reflected back, but small enough to obtain the desired resolution. The InfoCard will be using a $^1/_{16}$-inch-wide aperture. That will let us space the bars on the card $^1/_{16}$ inch apart.

Circuit Description. As you can see in the schematic diagram shown in Fig. 3, the interface circuit for the InfoCard is quite simple and straightforward. Reflective IR sensor SEN1 contains a matched IR diode and an IR detector in a single case. The current for the IR diode is limited by R1; R2 is a pull-up resistor for the collector of the detector. That lets a proper voltage level always be present at the positive input of IC2-b. The negative input is connected to an adjustable voltage divider made up of R3, R7, and R8. By varying R7, the voltage level at which the comparator switches can be set.

When the positive input of IC2-b is less than the negative input, the output will be grounded. That occurs when a reflecting surface is close enough to SEN1 to reflect IR light onto the detector. Otherwise, the output will have no voltage on it. The LM339 was designed that way so that the output could be used in circuits that needed the output level to be different than the supply voltage. Because of that, R4 is used to pull up the output of IC2-b when the output should be high. That signal is connected to pin 11 of the PC's printer port.

A second comparator in IC2 is used as an LED driver. When pin 2 of

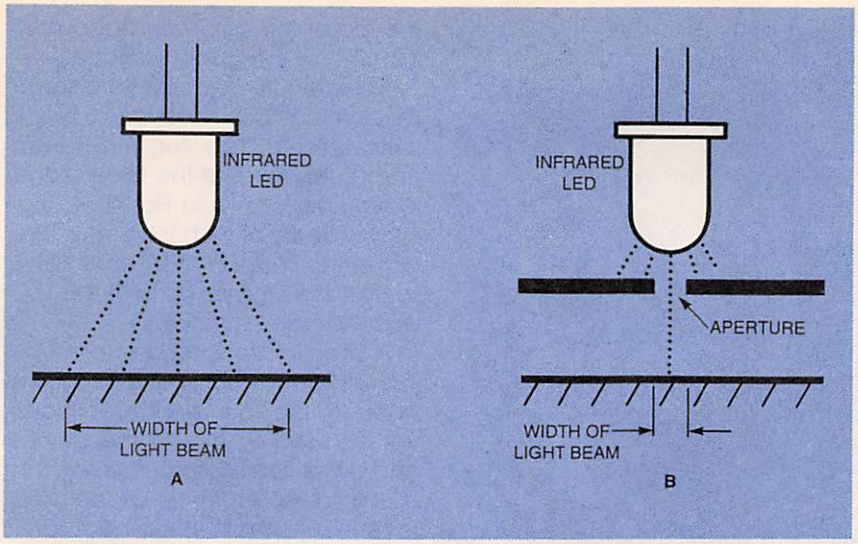


Fig. 2. Even laser light spreads out as it leaves the source (A). That spreading is called divergence. *If the spreading light is blocked by an aperture (B), the light beam will have a greater pinpoint accuracy.*

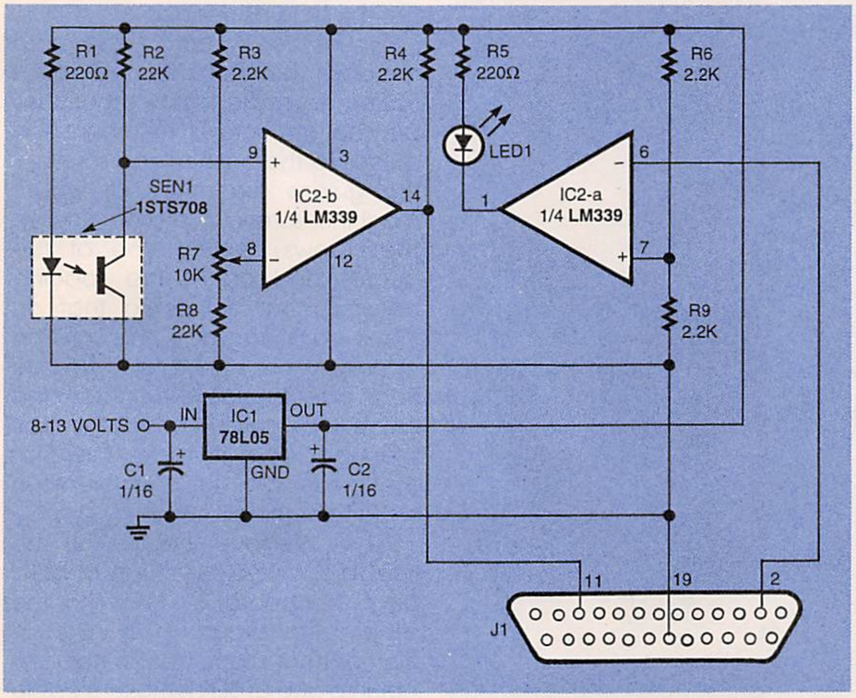


Fig. 3. The InfoCard's circuit is built around a reflective optical sensor (SEN1) and a comparator (IC1).

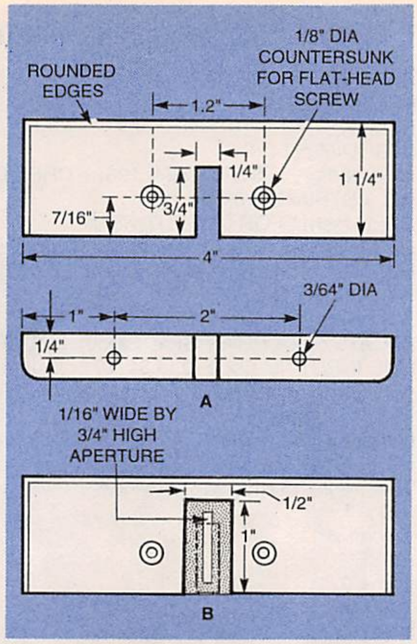


Fig. 4. The card guides need a slot in one of them to mount the sensor. Use flat-head screws to mount the sensor so that the guide slot will be smooth.

the parallel port is brought high, IC2-a's output will be open, turning off LED1. A low input to IC2-a will ground its output, letting current flow through R5 and LED1, turning it on. Voltage-divider string R6 and R9 set a switching level for IC2-a that is compatible with the voltage levels present on the parallel port.

Power from any suitable voltage source between 8 and 13 volts is regulated to 5 volts by IC1, with C1 and C2 providing filtering.

Construction. The construction details presented here are only for purposes of building an InfoCard demonstrator. Once you are familiar with the design and operation of the unit, no doubt you'll be designing your own custom housings and installations. To begin, we'll start with the card guides.

The dimensions of the card guides are shown in Fig. 4. Use two pieces of ½-inch-thick wood or metal with the dimensions shown in Fig 4A. Both pieces should have their edges rounded as shown. Keep in mind, however, that the rounding shown is greatly exaggerated. It is important that the edges be smooth so that a swipe card will not hang up or tear as it is slid through the guide. The edges that will be rounded will be facing each other to form the guide.

Both guides will have holes drilled in their bottom edge as shown. If you are using metal for the guides, you will have to tap threads into the holes. Once that is done, set one piece aside—it is finished. The second piece will have a notch cut in it as shown for SEN1. The countersunk holes will be used to mount SEN1; the countersink goes on the inside of the guide. The holes should be sized to fit a 4-40 flathead screw.

Install a pair of flathead screws in the countersunk holes with nuts. A second pair of nuts should then be threaded onto the protruding screw threads. Take a piece of perfboard that is large enough to mount SEN1 and drill two holes to match the spacing of the flathead screws. Mount SEN1 on the perfboard. That assembly is then mounted on the screws and held in place with an additional pair of nuts. The second and third nuts will form a "sandwich" that will hold SEN1 in position. Adjust the position of the nuts so that the front of SEN1 is perpendicular to the

LISTING 1

```
REM** INFOCARD.BAS
REM** V970719
REM**
CLS : DIM n(14)
x$ = "##### " + STRING$(4, 196) + CHR$(62) + CHR$(32) + CHR$(179)
sp$ = STRING$(10, 32)
OPEN "scrn:" FOR OUTPUT AS #1
LOCATE 1, 27: PRINT "PcInfoCard Card Generator"
LOCATE 2, 1: PRINT STRING$(80, 220);
'PROGRAM BEGINS
getnumber:
LOCATE 5, 27: PRINT SPACE$(50): LOCATE 5, 27
LINE INPUT "ID Code (0 to 32767): "; number$
 IF LEN(number$) = 0 THEN number$ = "0"
 IF VAL(number$) < 0 OR VAL(number$) > 32767 THEN BEEP: GOTO getnumber
number = VAL(number$): n = number: code$ = ""
FOR i = 14 TO 0 STEP -1
 IF n >= 2 ^ i THEN
 n(i) = 1
 n = n - 2 ^ i
 ELSE
 n(i) = 0
 END IF
NEXT i
FOR i = 14 TO 0 STEP -1
 IF n(i) = 1 THEN
 code$ = code$ + CHR$(219) + CHR$(221)
 ELSE
 code$ = code$ + CHR$(221)
 END IF
NEXT i
code$ = code$ + CHR$(222)
code = LEN(code$): delta = 34 - code
IF delta / 2 <> INT(delta / 2) THEN
 d1 = INT(delta / 2): d2 = INT(delta / 2) + 1
ELSE
 d1 = delta / 2: d2 = d1
END IF
LOCATE 7, 1
CLOSE : OPEN "scrn:" FOR OUTPUT AS #1: GOSUB card
LOCATE 5, 27: PRINT SPACE$(50)
LOCATE 8, 52: PRINT "Press"
LOCATE 9, 52: PRINT "——"
LOCATE 11, 52: PRINT "<H> for Hardcopy"
LOCATE 13, 52: PRINT "<Esc> to End"
LOCATE 15, 52: PRINT "Any Other Key to Try Again"
getoption:
a$ = INKEY$: IF a$ = "" THEN GOTO getoption
a$ = UCASE$(a$)
SELECT CASE a$
 CASE IS = CHR$(27)
 CLS : LOCATE 18, 1: END
 CASE IS = "H"
 CLOSE : OPEN "lpt1:" FOR OUTPUT AS #1: GOSUB card
END SELECT
FOR i = 8 TO 15
 LOCATE i, 52: PRINT SPACE$(26)
NEXT i
GOTO getnumber
'
'SUBROUTINE TO PRINT/DISPLAY CARD
card:
PRINT #1, sp$; CHR$(218); STRING$(code + delta, 196); CHR$(191)
FOR i = 1 TO 7
 PRINT #1, sp$; CHR$(179); STRING$(code + delta, 32); CHR$(179)
NEXT i
PRINT #1, sp$; CHR$(179); STRING$(22, 32);
PRINT #1, USING x$; number
FOR i = 1 TO 3:
 PRINT #1, sp$; CHR$(179); STRING$(d1, 32); code$; STRING$(d2, 32); CHR$(179)
NEXT i
PRINT #1, sp$; CHR$(192); STRING$(code + delta, 196); CHR$(217)
RETURN
```

surface of the guide and about $1/8$ inch below the guide's surface.

An aperture is made from some black electrical tape or a black write-protect tab for a 5¼-inch computer disk. Cut the tape to the dimensions shown in Fig. 4B. A $1/16$-inch wide by ¾-inch long slit is also cut in the tape. Place it over SEN1 so that the slit lines up with the LED and the detector in SEN1.

A base plate is made from $1/8$-inch hardboard or metal as shown in Fig. 5. The four elongated holes will be used to mount the guides and adjust the width of the slot. The $1/8$-inch diameter hole will be used for the wires that will connect SEN1 to the circuit board. If you wish to use a rubber grommet, you should size the hole as needed.

Mount the two card guides onto the base with their rounded edges facing towards each other. Use appropriate screws to mount the guides to the base (wood screws if the guides are made of wood; machine screws if metal). Loosen the screws holding one of the guides and adjust the spacing between the guides so that an index card can pass through the opening with ease, but not move from side to side. Re-tighten the guide in place and check the fit of the index card to make sure that the guide did not shift or move while tightening it down.

The interface circuit for the InfoCard is simple enough to build on a perfboard by following the schematic diagram in Fig. 3. Component placement is not critical. In place of SEN1, make a cable from three equal lengths of insulated wire that will be able to reach from the board to SEN1. Make note of which wires will be connected to the various pins on SEN1.

A second three-wire cable will be used to connect the InfoCard to the PC. Again, three equal lengths of 22- or 24-gauge stranded wire are used. Connect them to pins 2, 11, and 19 of a 25-pin connector. The other ends of the wires connected to pins 2, 11, and 19 are connected to the circuit according to Fig. 3.

Thread the three-wire cable for SEN1 through the hole in the base plate and connect the wires to SEN1. If you are not sure which

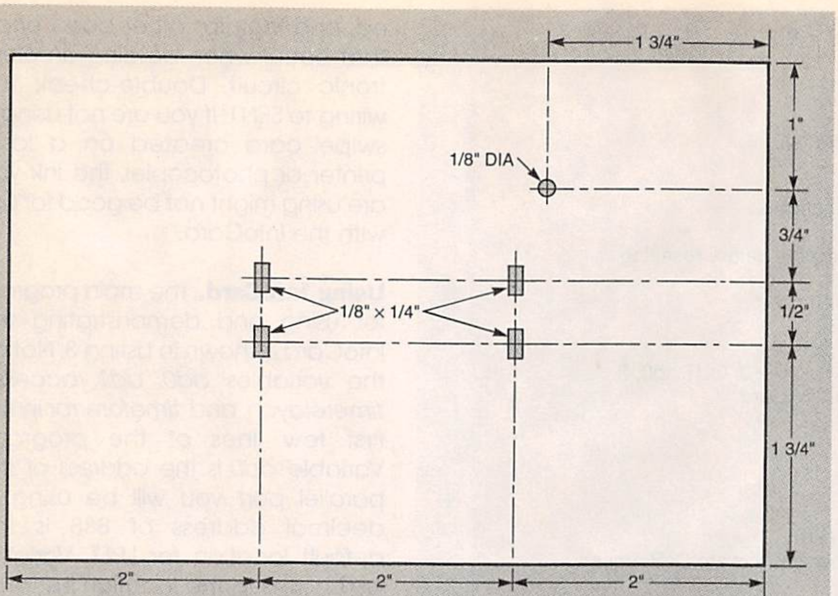


Fig. 5. The base plate has elongated slots so that the width of the card guide can be adjusted. The wires from the sensor pass through the hole in the upper corner. If you are going to use metal for the base plate, make that hole a bit larger and use a rubber grommet to protect the wires.

LISTING 2

```
''' ICARDTST.BAS
''' Version 970715.1
CLS : DEF SEG = 64: ad0 = 888: ad1 = 889
OUT ad0, 2
PLAY "l64t240"
LOCATE 1, 1
COLOR 0, 7: FOR I = 1 TO 3: PRINT SPACE$(80); : NEXT i
LOCATE 2, 22
PRINT "PcInfoCard Test (Press <Esc> to End)"; : COLOR 7, 0
LOCATE 10, 25: PRINT "Emitter "; STRING$(3, 219); ") \"
LOCATE 11, 40: PRINT "\"
LOCATE 12, 42: PRINT "\"
LOCATE 15, 24: PRINT "Detector "; STRING$(3, 219); ")"
loop00:
a = INP(ad1) AND 128
 IF a = 0 THEN
 OUT ad0, (INP(ad0) AND 254)
 FOR i = 10 TO 15: LOCATE i, 43: PRINT CHR$(32): NEXT i
 LOCATE 13, 41: PRINT " "
 LOCATE 14, 39: PRINT " "
 PLAY "n48l64p4"
 ELSE OUT ad0, (INP(ad0) OR 1)
 FOR i = 10 TO 15: LOCATE i, 43: PRINT CHR$(222): NEXT i
 LOCATE 13, 41: PRINT "/"
 LOCATE 14, 39: PRINT "/"
END IF
a$ = INKEY$: IF a$ = "" THEN GOTO loop00
IF ASC(a$) <> 27 THEN GOTO loop00
END
```

leads are for the LED, measure the resistance across the leads with an ohmmeter. If the ohmmeter shows conduction in one direction, those pins are the LED; if neither direction conducts, they are the detector. Checking the LED with an ohmmeter will also show its polarity.

If you want, you can mount the circuit board to the bottom of the base plate. A set of long screws or standoffs can be used as legs, if desired.

Making a Swipe Card. The coded card consists of narrow (to represent a "0") and wide (for a "1") bars separated by blank white spaces. The wide bar must be at least twice as wide as the narrow bar, and the blank must be as wide, if not wider, than the aperture. The bars should be positioned on the card so their bottoms are within 1/8 inch of the card's bottom. Finally, the bars should be at least ½-inch high.

The bars are read right-to-left when looking at the card with the bars at the bottom. The first bar is always narrow—it is a reference bar and is not part of the binary code. The next fifteen bars are a binary representation of a number between 0 and 32767. The first bar after the reference bar is bit 0, with each bit proceeding in order from there up to bit 14.

Making a swipe card is easy with the program in Listing 1. When you enter a number between 0 and 32767, the resulting barcode is displayed on the screen. To print that pattern on a card, press "H". Once printed, cut away the excess paper so that the card will be the correct size. The program marks where the edges of the card should be as a guide.

Some printer inks are not IR absorbing (such as the ink used in Hewlett Packard ink jet printers). The best results are obtained with a laser printer or by photocopying a card that has been printed. If you don't have access to a laser printer or a photocopier, you can fill in the bars with a felt-tipped pen that has IR-absorbing ink. You can use the InfoCard's testing and calibration program in the next section to find out if the particular felt-tipped pen you plan to use will do the job.

Testing and Calibration. Apply power to the InfoCard, connect it to a PC's printer port, and load the QBASIC program in Listing 2. Using a DC voltmeter, adjust R7 until the voltage on pin 8 of IC2 is about 4 volts.

Place a card in the card guide so that a wide bar is directly in front of the aperture. Check to see if LED1 is lit. If it is not, turn R7 to increase the voltage level on pin 8 of IC2 until LED1 comes on. Start turning R7 the other way slowly until LED 1 goes out, then add an additional ¼ turn. If the swipe card is moved so that SEN1 sees the white of the card, LED1 sould turn back on.

If that procedure does not work as it should, check the accuracy of the circuit wiring, make sure that all components are properly orientat-

LISTING 3

```
REM ** ICARD.BAS
REM ** Version 970720
REM
CLS : DEF SEG = 64: ON ERROR GOTO errorloop
DEFINT A-B, X: DIM a(25), b(25)
ad0 = 888: ad1 = 889
access1 = 226: 'If this code found, Relay will activate.
timerelayon = 10: 'Time Relay Stays On.
timeforerror = 5: 'Time to wait when an error occurs before resetting.
'
'**PROGRAM BEGINS
'

loop01:
CLS : LOCATE 10, 36: PRINT "Ready..."; : x = 0: xo = 0: OUT ad0, 3
WHILE (INP(ad1) AND 128) = 0: WEND: 'Wait for Card
'** Card In. Turn Off LED and Wait for Bars.
OUT ad0, 2
WHILE (INP(ad1) AND 128) = 128: WEND
'** First Bar Found.
redo1:
WHILE (INP(ad1) AND 128) = 0: x = x + 1: WEND
IF (x - xo) < 10 THEN x0 = x: x = 0: GOTO redo1: 'Software DeBounce
a(0) = x: b(0) = 0
'** Width of first (reference) line measured as a(0)
FOR i = 1 TO 15
redo01:
x = 0
'Wait for next line.
WHILE (INP(ad1) AND 128) = 128: WEND
'Determine Width of Line.
WHILE (INP(ad1) AND 128) = 0: x = x + 1: WEND
IF x < .1 * a(0) THEN GOTO redo01: ' Software Debounce
a(i) = x
NEXT i
'** Card Read Done. Ref is a(0). Bits are a(1)-a(15).
loop02:
CLS : code = 0
FOR i = 1 TO 15: b(i) = 0
SELECT CASE b(i - 1)
 CASE IS = 0
 IF a(i) > a(i - 1) * 1.5 THEN b(i) = 1: code = code + 2 ^ (i - 1)
 CASE IS = 1
 IF a(i) > a(i - 1) * .75 THEN b(i) = 1: code = code + 2 ^ (i - 1)
END SELECT
NEXT i
LOCATE 10, 30: PRINT "CODE: "; code
LOCATE 11, 27: PRINT "(";
 FOR i = 1 TO 15: PRINT RIGHT$(STR$(b(i)), 1); : NEXT i: PRINT ")"
LOCATE 18, 20: PRINT "<Esc> Ends, Any Other Key Continues...";
blink01:
IF code = access1 THEN blink1 = 0: blink2 = 1 ELSE blink1 = 2: blink2 = 3
 OUT ad0, blink1
start1! = TIMER
 WHILE (TIMER - start1!) < timerelayon
 OUT ad0, blink1
 start! = TIMER
 WHILE (TIMER - start!) < .15: WEND
 OUT ad0, blink2
 start! = TIMER
 WHILE (TIMER - start!) < .15: WEND
 a$ = INKEY$
 IF a$ <> "" THEN
 IF ASC(a$) = 27 THEN OUT ad0, 2: END ELSE GOTO loop01
 END IF
 WEND
GOTO loop01
errorloop:
LOCATE 10, 15
BEEP: PRINT "Error Reading Card. <Esc> to End, or Wait For Reset.";
start! = TIMER
WHILE (TIMER - start!) < timeforerror
 a$ = INKEY$: IF a$ <> "" THEN a = ASC(a$) ELSE a = 0
 IF a = 27 THEN OUT ad0, 2: END
WEND
RESUME loop01
```

ed, and look for other basic errors that occur when building an electronic circuit. Double-check the wiring to SEN1. If you are not using a swipe card created on a laser printer or photocopier, the ink you are using might not be good for use with the InfoCard.

Using InfoCard. The main program for using and demonstrating the InfoCard is shown in Listing 3. Notice the variables *ad0, ad1, access1, timerelayon,* and *timeforerror* in the first few lines of the program. Variable *ad0* is the address of the parallel port you will be using. A decimal address of 888 is the default location for LPT1. Variable *ad1* points to the location just after *ad0.* As an alternative, you can set *ad1* to equal *ad0* + 1. That way, if you change the port that you're using, *ad1* will automatically track which port you are using. Variable *access1* is the card code that will activate LED1. Variables *timerelayon* and *timeforerror* set the time duration, respectively, of how long LED1 will remain on and how long the program will wait before resetting itself if an error occurs. You can change each of those variables to suit your needs.

With the program running and the InfoCard hardware connected and powered up, the screen will simply say "Ready". Run a card through the reader. The card's value will be displayed. If the value matches the value in *access1,* LED1 will light. Pressing the Escape key will end the program, while pressing any other key or waiting for the time specified by *timerelayon* will reset the program to the "Ready" mode. If the card doesn't read properly, the screen will display "Error in Reading Card". You can either press the escape key to end the program or wait for the program to reset itself to the "Ready" mode. In that case, the delay time before reset is controlled by the *timeforerror* variable.

Relays and Other Options. Replacing LED1 with an optoisolator will let you interface the InfoCard with other hardware that can be controlled by an on-off-style switch. Devices such as the MOC3010

(Continued on page 49)

INFOCARD CARD SCANNER

(continued from page 42)

(which has a Triac-controlled output) can be used to control AC-based devices. With a simple transistor switch, a small relay could also be activated. If you do not want to remove LED1, the demonstration program also activates pin 3 of the printer port along with LED1. Multiple output circuits can be hooked up to pins 4 through 9 with modification to the program. If you use that pin, be careful of any circuit that you connect to the computer's hardware—you might end up damaging your printer port.

TABLE 1

J1 Pin No.	BASIC Function
10	INP (*address*) AND 64
11 I	NP (*address*) AND 128
12	INP (*address*) AND 32
13	INP (*address*) AND 16

Multiple InfoCards can be hooked up to the same computer if, for example, you want to read cards from different locations. The basic circuit built around IC2-b is all that is needed. If you do not use LED1, a single LM339 can be used for up to four InfoCard stations. Of course, the program will have to be modified to accept input from the different stations. Table 1 shows how to change the input statement in the program to read card information from different input pins.

The InfoCard software could also be integrated into another program or batch file to control access to lists or other data in the computer depending on a card's code. You can modify the program to have multiple access codes. You can also add routines to look up the code in a database to provide related information about the card holder. How you use the InfoCard is limited only by your imagination! Ω