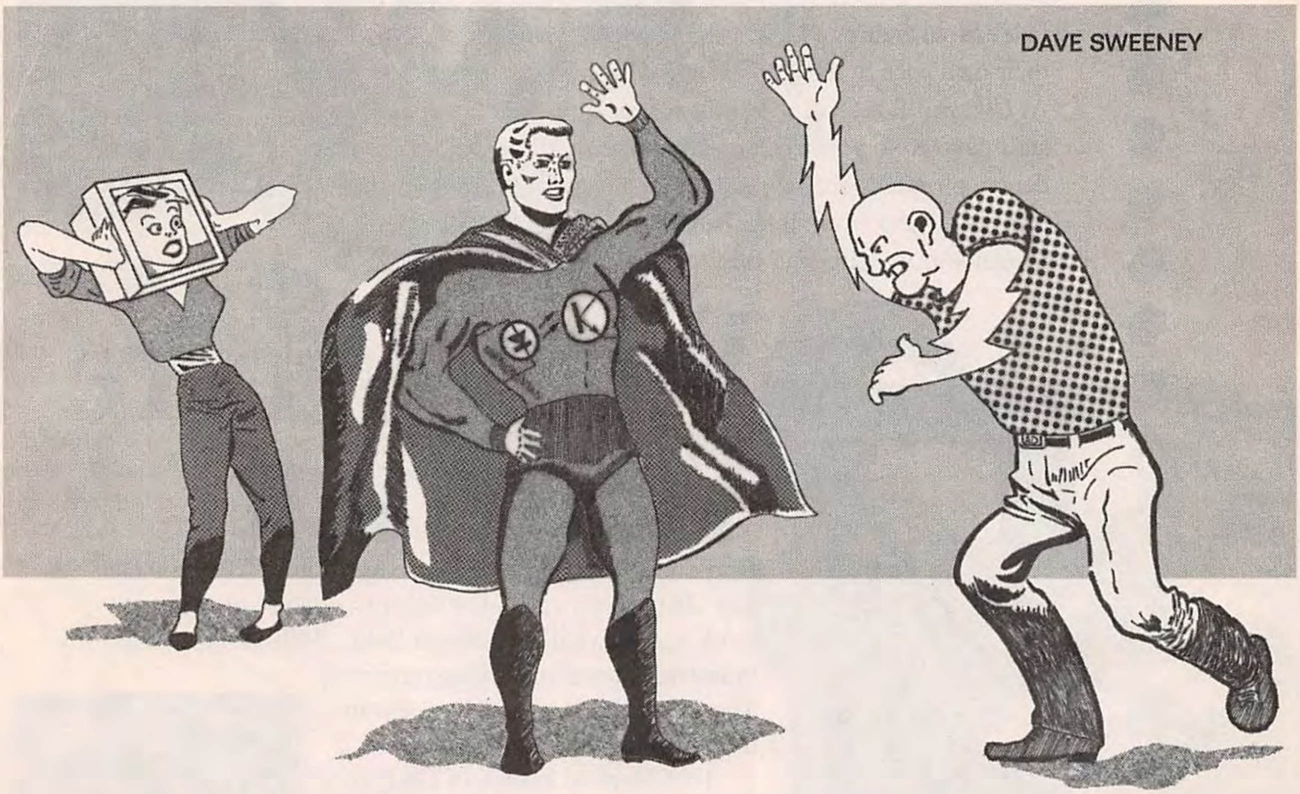
PROTECT YOUR COMPUTER'S PARALLEL PORT WITH THIS OPTICAL ISOLATOR

DAVE SWEENEY



Accidentally burning out a computer's internal circuitry is a thing of the past with this simple device.

There are an ever-increasing quantity of projects and devices that are controlled by the output signals available from a computer's printer port. Examples of those devices include relays, stepper motors, lights, and even electric trains. There have also been several projects that have appeared in both **Electronics Now** and our sister magazine, **Popular Electronics**. In fact, if a way can be found to use a 5-volt digital signal to control some device, it can be hooked up to a printer port and controlled by some simple software commands. With that ability at your fingertips, you could (dare we say it)...run the world!

Seriously, though, interfacing a computer to a piece of hardware and having the software react to feedback from external signals has opened up a new level of control and sophistication in electronics. The problem many times has become one of taking proper safety steps to safeguard your computer from any errors that you might have made in assembling your latest experimental project. The days where you could build a new project and "watch for smoke" are sadly (or thankfully) gone. An ungrounded soldering iron, voltage on a ground return, or a short at the wrong moment could zap a computer's internal circuits. Even an inductive pulse on the computer's output could cause a smokeless tragedy.

What is needed then is a way to electrically isolate the printer port's signals yet let them pass through to the device that they are to control. The ideal solution is optical isolation. That way the parallel port's pins are disconnected electrically from the outside world, but signals can flow back and forth on a beam of light. Combining the Parallel-Port Optical Isolator presented here with software that controls the separate pins on the parallel port results in a system that's safe for computer control of your experimental devices.

Light Signals. By using light to conduct a signal, optical isolators interrupt the electrical connection between their input pin and output pin. A typical arrangement, shown in Fig. 1, demonstrates the concept. Inside the optical isolator are a light-emitting diode and a phototransistor. When a positive voltage is applied to the LED through current-limiting resistor R1, the LED shines. The light from the LED falls on the phototransistor, causing it to conduct as if a current were applied to its (non-connected) base lead. Note that on some real-world optical isolators, the phototransistor's base lead is brought out to an additional pin.

Normally, the collector lead of the phototransistor is at a positive voltage supplied through R2. When the phototransistor conducts, the current through R2 is shunted to the output ground through the phototransistor's emitter. Because of the voltage drop inherent in all semiconductors, the voltage on the phototransistor's collector will generally be below one volt—it will never reach a zero-volt potential unless the emitter is connected to a negative voltage that is equal to the phototransistor's voltage drop.

At this point, the output signal is an inverted version of the input signal; if that is fine for your application, no further conditioning to the signal is needed. However, it is usually best to keep the polarity of the output and input signals matched. In the Fig. 1 example, the inverted and isolated signal is buffered and re-inverted by an op-amp. An additional advantage of using an op-amp is having a low-impedance output.

It is very important to note that the input and output grounds are not connected together in any way; the purpose of optical isolation would be defeated if they were. To further enhance electrical isolation, a separate power supply should be used for the output portion of the circuit; that is, the phototransistor and op-amp.

Circuit Design. The full schematic diagram for the Parallel-Port Optical Isolator is shown in Fig. 2. Note that it is very similar to the single isolation circuit of Fig. 1. While any opto-isolator chip that has a similar pinout can be used in place of the devices specified, keep in mind that the suggested units have an isolation rating of 5000 volts—much higher than most other units that are commonly available.

In our discussion of the basic circuit, we did not mention any detailed description of the op-amp. Traditionally, an op-amp in its linear mode always tries to keep its two input lines at the same voltage level. By feeding back the op-amp's output voltage to one of the inputs, the output can be made to follow an input signal at whatever ratio and pattern set by external components. While that explanation is very simplistic, it is sufficient for our purposes here; a detailed discussion can fill a book.

When used in a non-linear mode (linear mode with "infinite" gain), an op-amp compares the voltage levels of the inputs. If the non-inverting input is higher than the inverting input, the output will go high. A reference voltage is needed to set the point at which the output will change state. A reference voltage of one-half the power supply voltage is created by R41 and R42; that reference voltage is applied to all of the non-inverting inputs. As long as the voltage from the phototransistors does not fluctuate beyond that level, the output of the op-amps will remain stable.

An additional opto-isolator, IC3, lets the Parallel-Port Optical Isolator send isolated feedback signals to the computer. Note that op-amp buffers are not used. Since the parallel-port input circuit is a known load that does not change, buffering is not necessary. However, note that the phototransistor's load resistor is connected between the emitter and ground, acting as a pull-down resistor when the phototransistor is off; the result is a non-inverting output to the computer. Although there are five input lines in a standard parallel port, only four are implemented; the opto-isolator chip that we are

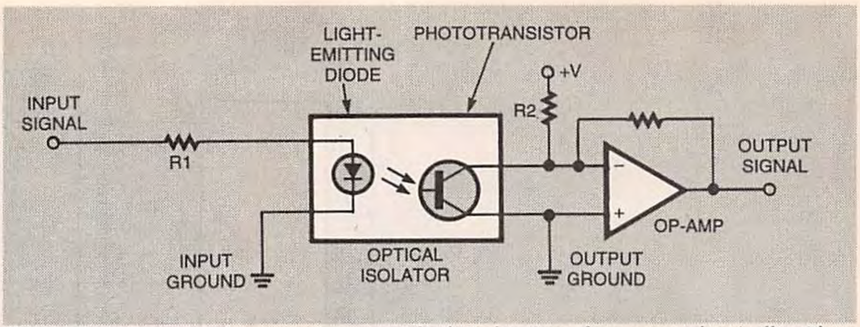


Fig. 1. A simple opto-isolator circuit can completely isolate input from output electrically without impeding the signal itself; the signal travels on a beam of light.

LISTING 1

```
                'Continuous pulse output to pin 3
10              'Start
OUT &H3BC,0     'Reset all pins – port 3BC is assumed. Port 278 or 378 possible
D=2             'Value selects pin – 1=pin 2, 2=pin 3, 4=pin 4, 8=pin 5, 16=pin 6,
32=pin 7, 64=pin 8, 128=pin 9
PW=5            'Controls the time the output is high (pulse width)
                'Begin pulse generation
OUT &H3BC, D    'Sets pin 3 high
DO WHILE I,PW
        I=I+1
LOOP
I=0
PI=10           'Controls the interval between pulses
OUT &H3BC,0
DO WHILE I<PI
        I=I+1
LOOP
GOTO 20
```

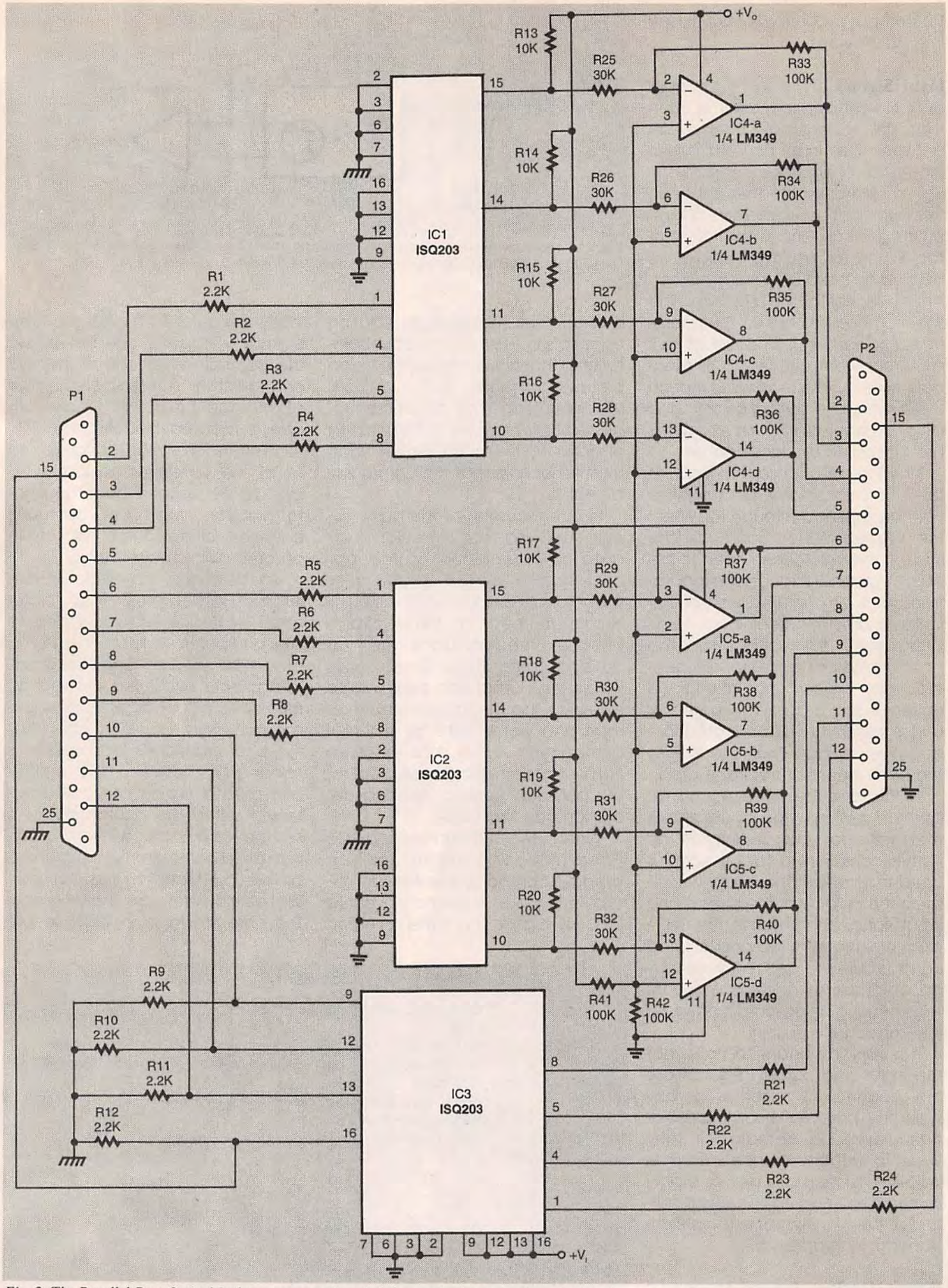


Fig. 2. The Parallel-Port Optical Isolator uses a series of eight opto-isolator circuits to isolate the port's output lines. An additional set of four circuits lets feedback signals enter the computer. Note that two completely separate grounds are used.

using has four isolation circuits. With those input signals, the computer software can react to signals from whatever external hardware is plugged into the Parallel-Port Optical Isolator. Similarly, only the eight main data lines are used for output. Since not all of the standard signals are connected, the Parallel-Port Optical Isolator can't be used with traditional computer equipment such as printers or other devices that plug into the parallel port. Those limitations should be kept in mind when you are connecting any type of hardware to the Parallel-Port Optical Isolator.

Power Supply. As mentioned above, the Parallel-Port Optical Isolator needs a separate power supply to prevent any possible electrical damage to the computer. The schematic for the power supply is shown in Fig. 3. Each supply voltage is regulated. While IC6, a simple 5-volt regulator supplies power to the input circuit (IC3), a higher voltage is needed for the output (computer-to-hardware) circuit. In order for the op-amps to achieve a 5-volt peak output, their supply voltage must be at about 6.5 volts. An LM317T adjustable regulator is used for IC7. With a 12-volt input, the values shown for R1 and R2 will set the output to 6.5 volts. Variations in the supply voltage might change IC7's output voltage; adjusting the value of R2 will compensate for any supply variations.

Construction. The Parallel-Port Optical Isolator is simple enough to be built on a piece of perfboard using standard construction techniques. However, a neater assembly results if a PC board is used. For those that would like to use the printed-circuit approach, foil patterns for the single-sided boards have been provided. Note that the power-supply regulators have been placed on a separate board. Having two smaller boards instead of one large one gives you more flexibility in choosing a suitable enclosure for the unit.

If you decide to etch your own PC boards using the foil patterns, follow the parts-placement diagrams when assembling the boards; Fig. 4 is for the optical-isolator circuit itself while Fig. 5 is for the power-supply regulators.

It is a good idea to use sockets for the integrated circuits. Not only will that make it easier to replace a "zapped" IC when repairing the unit, it will make testing easier. For now, just install the sockets; do not install the ICs at this time. When building the main board, don't forget to install the several jumpers that are indicated; use insulated wire to prevent any accidental shorts.

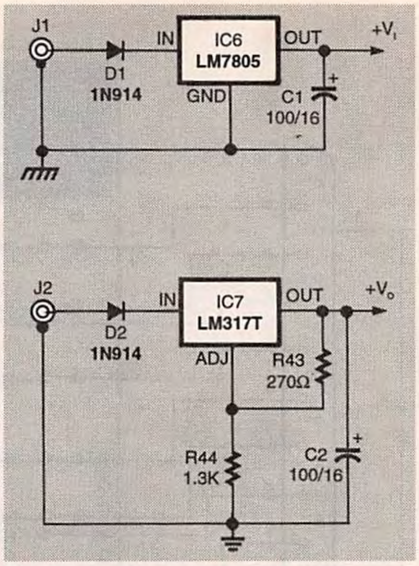
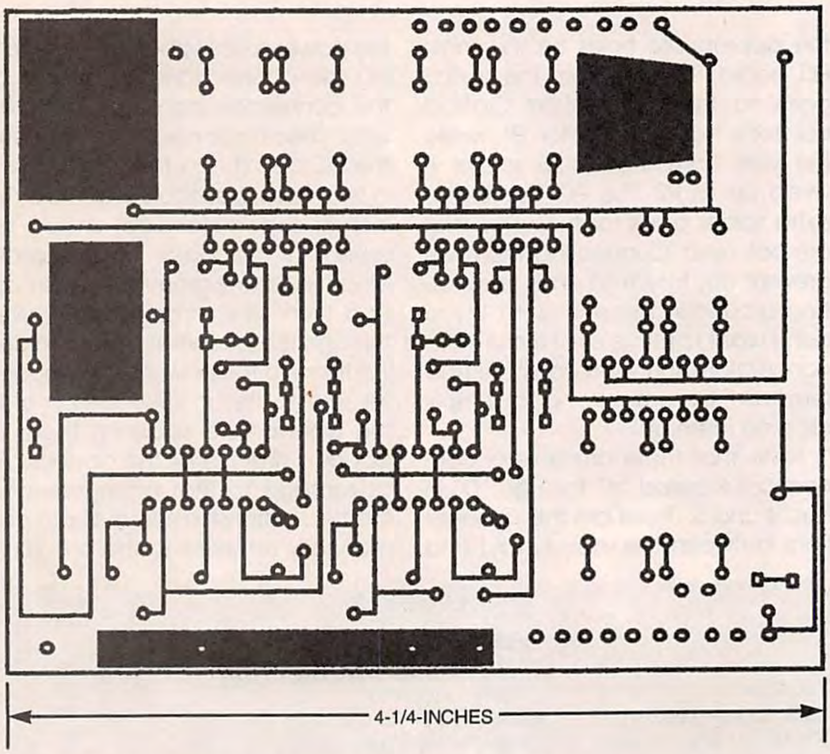


Fig. 3. A simple pair of power regulators supply the Parallel-Port Optical Isolator. Note that the output supply uses an LM317T to supply 6.5 volts to the op-amps. That way, the output of the op-amps can reach a full five volts for CMOS compatibility.

A Ready-Made Enclosure. The Parallel-Port Optical Isolator can be housed in any suitable enclosure. However, cutting the trapezoid-shaped holes for the DB-25 connectors can be difficult. It would also become annoying having to keep plugging and unplugging the Parallel-Port Optical Isolator every time you wanted to use it. The solution to those problems is to install the unit into an existing parallel-port A-B switch. Those devices, available at most computer stores, let you connect two pieces of equipment to a parallel port. By turning the switch on its front panel, you can connect either device to the computer without having to change or move any cables, saving wear and tear on the connectors, pins, cables, wires, and—most importantly—your sanity!

In selecting an A-B switch, it is important to choose a mechanical one. There are "automatic" devices available, but they are completely electrical in nature and sometimes require special software to use.



Here's the foil pattern for the Parallel-Port Optical Isolator's main board.

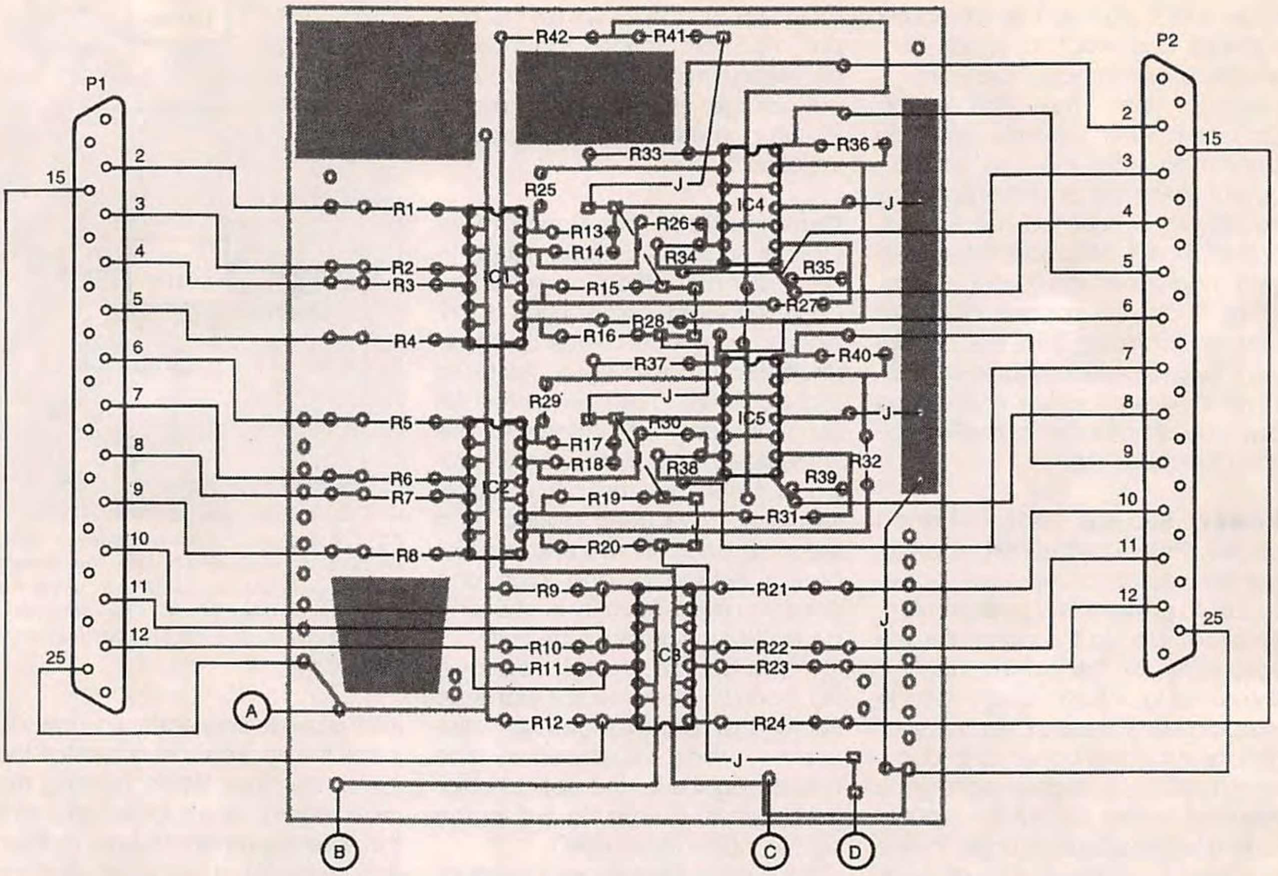


Fig. 4. The Parallel-Port Optical Isolator uses a single-sided PC board. Note that several jumpers are needed to complete some connections. Some of the resistors might be a tight fit for flat mounting and might have to be mounted vertically.

Another important feature to look for is that the A-B switch uses individual wires to make the connections between the connectors instead of a PC board. While the PC-board version is considered a "deluxe" model that minimizes crosstalk between the signals, the modification of the wire-type switch is much easier.

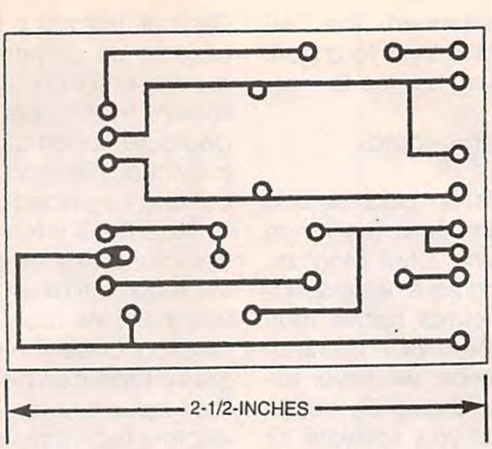
The author's prototype was installed in such a box; it can be seen in Fig. 6. Note how the main PC board sits at the bottom of the case while the power-supply board is tucked against the front panel next to the switch.

Start by removing the switch and the connectors from the case without breaking any of the wires. Select one of the "output" connectors for optical isolation; don't use the "common" connector that will go to the computer. Cut each wire between that connector and the switch, strip the two ends, and install them into the appropriate holes on the main PC board. The wire from the switch goes to the Parallel-Port Optical Isolator's connections for P1, while the wire from the DB-25 socket is wired up as P2. The PC board has extra solder pads for the wires that are not used. Connecting them will prevent any lose wire ends from rattling around in the enclosure. If you don't want to make all of those extra connections, the wires can be simply removed by unsoldering or simply clipping them off.

Note that there are several connections labeled "A" through "D" in Figs. 4 and 5. Those are the connections between the main board and the power-supply board. When making the connections for J1 and J2, the connectors can be substituted with direct connections between the PC board and the transformers. In that case, a suitably sized hole (3/8-inch is a good choice) should be drilled in the back of the case. Mount a rubber grommet in the hole and run the transformer wires through the grommet before soldering them to the power-supply board. Be sure to tie an overhand knot in the wires before soldering them to act as a strain relief. The obvious disadvantage to that arrangement is the heavy transformers that are permanently tethered to the unit, mak-

LISTING 2

```
                        'Input sensing
10
DATA%=0                 'Initialize a variable
DATA%=INP(&H3BD)        'Read state of input pins
PRINT DATA%             'Display value of pin reading
GOTO 10
```



Here's the foil pattern for the Parallel-Port Optical Isolator's power-supply board.

ing it difficult to move the Parallel-Port Optical Isolator from one location to another.

Install the main PC board in the bottom of the case. Be sure to insulate the bottom of the box to prevent shorts; a piece of heavy gray "fish" paper, electrical tape, or gasket paper will work well. If you don't want to drill holes for screws, nuts, and spacers, you can bond a few pieces of hook-and-loop fastener material to the bottom of the PC boards and to the insulating paper. If you decide to go that route, be sure to glue the insulator to the case!

Testing. With no ICs installed, turn the switch to connect to the optically-isolated connector. Measure for infinite resistance on all combinations of pins between the input and output connectors: pin 1 to pin 1, pin 1 to pin 2, pin 1 to pin 3, and so on. Once all 25 combinations have been checked, continue with pin 2 to pin 1, pin 2 to pin 2, pin 2 to pin 3, and so on.

On the same connector, pins 2 through 9 should read infinity to ground (pin 25). On the input connector (P1), pins 10–12 and 15 should read 2200 ohms to ground; on the optically-isolated connector, they should all read infinity to ground. Be sure to use pin 25 of the same connector as your ground reference when making those checks. You should also read infinite resistance between the two power supplies and the two grounds.

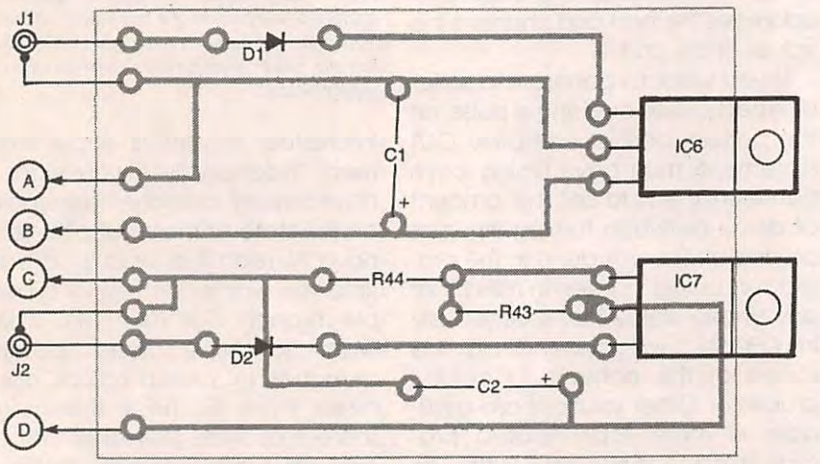


Fig. 5. The power-supply board is a simple single-sided construction. The four connections labeled "A" through "D" show the connections between the power-supply board and the main board.

Plug in the transformers. On IC1–IC3, pins 10, 11, 14, and 15 as well as pin 4 on IC4 and IC5 should read between 5 and 7 volts. On IC4 and IC5, pins 3, 5, 10, and 12 should read about 2.5 volts.

Unplug the transformers and insert the ICs. With the power turned on, apply pulses from a pulse generator to the input pins and check to see that the pulse appears on the corresponding pin on the other connector. Pins 2–9 pass signals from P1 to P2, while pins 10–12 and 15 go in the opposite direction. If you don't have a pulse generator to test the unit, the QBASIC code shown in Listing 1 can be used for testing the output pins. The comments in the program are self-explanatory. Monitor the pins on P2 with an oscilloscope and verify that the pulse rises to 5 volts and that there is no crosstalk between pins. The program will have to be modified to check each output pin in turn. Don't forget to choose the port address for the parallel port that the Parallel-Port Optical Isolator is plugged into. The example program is set for hexadecimal value 3BC; the other valid combinations on IBM compatible computers are 278 and 378 (again, in hexadecimal values). If you are unsure of the address of your port, you should check you computer's manual. If you have Windows95 or Windows98 installed on your machine, port addresses can be found in Device Manager, which can be found in the System Properties tool in Control Panel. If you are at an MS-DOS prompt, the utility MSD (Microsoft Diagnostics) can tell you the address of the parallel ports that are installed in your machine. Again, consult your computer's documentation if you are unsure how to go about doing that.

The simple program in Listing 2 can be used to check the input pins. As you apply a positive voltage to one input pin at a time, you'll see the value on the screen change. Again, check that there is no crosstalk between pins. Also note that the input port is one value higher than the address used in Listing 1. In the published examples, the output port is 3BC; the companion input port is 3BD. If you are using one of the other values, the input address would be x79 depending on which address your computer's parallel port is set for.

Once the Parallel-Port Optical

Isolator is verified as working without errors, it can be closed up and put to use.

Using the Parallel-Port Optical Isolator. If you know how to write your own software in your favorite language, you might already be familiar how to send and receive data from the parallel port. If not, QBASIC provides an extensive array of commands for building your own port-control software to switch port connector pins high or low. Keep in mind that since we are not using the Parallel-Port Optical Isolator to control a printer, the published standards that assign specific functions to each of the pins will not apply.

If you are going to be using QBASIC, a modified version of the test programs shown in Listings 1 and 2 can be used as simple subroutines. Those subroutines can be used in any program that you write whenever they are needed. Let's take a closer look at the two statements that the programming examples are built around: the "OUT" and "INP" statements.

PARTS LIST FOR THE PARALLEL PORT OPTICAL ISOLATOR

SEMICONDUCTORS

IC1–IC3—ISQ203 Quad opto-isolator, integrated circuit (Jameco 114083 or similar)
IC4, IC5—LM349 Quad op-amp, integrated circuit
IC6—LM7805 voltage regulator, integrated circuit
IC7—LM317T adjustable voltage regulator, integrated circuit
D1, D2—IN914 silicon diode

RESISTORS

(All resistors are ¼-watt, 5% units.)
R1–R12, R21–R24—2200-ohm
R13–R20—10,000-ohm
R25–R32—30,000-ohm
R33–R42—100,000-ohm
R43—270-ohm
R44—1300-ohm

ADDITIONAL PARTS AND MATERIALS

C1, C2—100-µF, 16-WVDC, electrolytic capacitor
J1, J2—Co-axial power connector
P1, P2—DB25 subminiature connector
12-volt DC power adapters, case or existing A-B port switch, hardware, wire, etc.

The "OUT" Statement. The OUT command sends a byte to a port; The OUT statement has the format

OUT <port>, <data>

where <port> is an address and <data> is a number in the range 0–255. In the Listing 1 test program, note that the address is being specified in hexadecimal rather than plain decimal. The choice is one of personal preference; whatever format you choose should be consistent across all of your software for readability's sake. We've already discussed what address to use to access the parallel port; you should already know that number at this point. The value for <data> can be a variable that is set by some other part of the program, the result of an INPUT statement (that will be discussed later), or even a constant number. Converting a decimal to binary will indicate which pins on the parallel port will be set high. The decimal values of the output pins are listed in the comments of Listing 1. Any output pin that is not addressed will automatically be set low by the OUT statement. The combination of high and low settings on the port pins will remain until another OUT statement addresses the port and changes the values of the port.

If you want to generate a series of timed pulses or a single pulse on the parallel port, consecutive OUT statements must have timing loops between them to set the amount of delay between turning the pins on and off (as was done in the program in Listing 1). Keep in mind that any simple delay loops, especially in QBASIC, will depend on the speed of the computer's central processor. Other solutions are available in more sophisticated programming languages such as Visual Basic, C++, and Turbo Pascal to name just a few. The use of those program-development packages are beyond the scope of this article and is left up to the reader's personal preferences.

The "INP" Function. The INP function returns a decimal value between 0 and 255 that is read from a port. The values that are returned from the Parallel-Port Optical Isolator's input pins will have to be determined by simple experimentation. Once they are known, the signals can be read and acted upon at one time or the individual bits can be separated out and examined one at a time.

Since INP is a function and not a command by itself, a destination for the read data must be specified. A variable is the usual way to use the function. Once the data is read in, the variable can be examined and the appropriate action taken. Another technique is to use the INP function as a part of an IF...THEN statement. However, it would become quite cumbersome to mask out the bits that you aren't interested in with a single statement. Additionally, if you want to have several branches depending on the state of the input pins, you'd have to read the input port each time you wanted to test for a possible branch. For example, if you want to have three possible branches in your program, you'd need three IF...THEN statements. There is also the possibility that the port value might change between readings for the various IF...THEN statements. Again, more sophisticated programming languages have many different approaches.

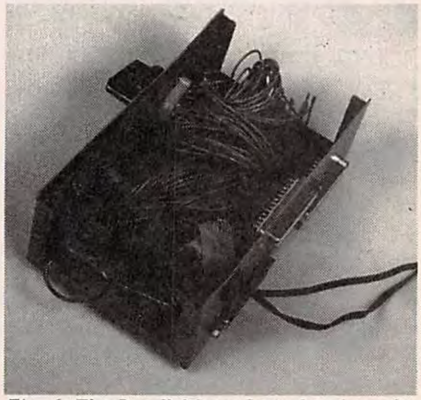


Fig. 6. The Parallel-Port Optical Isolator fits neatly into the bottom of a computer A-B switch. Note that the author's prototype, shown here, has the power transformers hard-wired to the supply board.

With the Parallel-Port Optical Isolator attached to your computer, you can experiment with controlling hardware without the fear of what a simple error could do to your computer. The worry of an expensive computer going "up in smoke" is a thing of the past! Ω