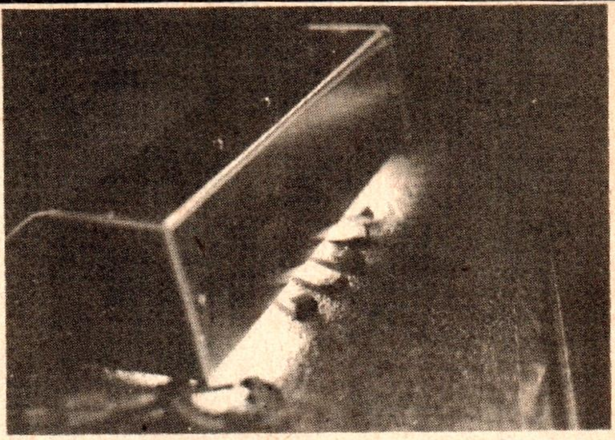
EPROM ERASER

A simple and inexpensive way to erase EPROMs from Marty Ing.



VIRTUALLY all EPROMs require a light source of 2537 Angstroms to erase the contents back to its original state of all outputs high or all outputs low. For a long time the only method of getting that type of light source that I knew of was to buy a commercial EPROM eraser. By the time I landed the cheapest one into Canada I was looking at over one hundred Canadian dollars. Since then I've found you can have an EPROM eraser for as low as $27.00.

At 2537 Angstroms light does not pass through common glass or the atmosphere very well. This means ordinary mercury vapour ultra-violet sources such as the ones you use for exposing sensitized PC boards or making blueprints do not work when it comes to erasing EPROMs. Exposing to direct sunlight doesn't seem to work either because of the atmospheric absorption of light rays in this region. I can safely say this because I've tried these methods including "black lights" and plant lights.

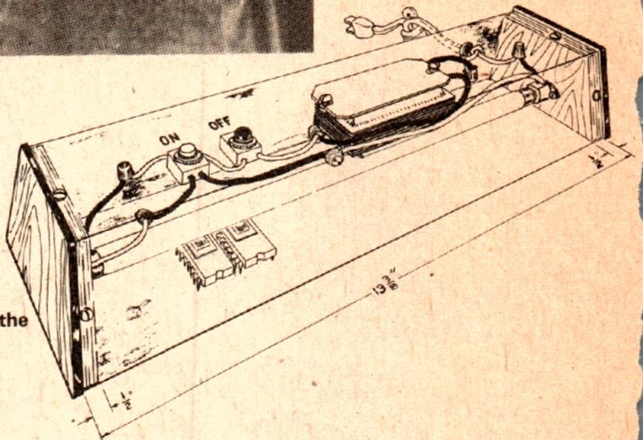


Fig. 1. Basic construction of the Eprom Eraser.

The Right Bulb

After searching around I finally found the G8T5 bulb (better known as germicidal lamp). The beautiful part of this lamp is that it will fit any standard desk top flourescent lamp that uses a 12" bulb. All you have to do is replace the existing bulb with the G8T5 or equivalent. This bulb isn't cheap. At $27.00 a bulb you better not drop it too many times. The secret to this bulb is quartz tubing instead of glass.

Placed at 3/4" away from the EPROM window it will erase the EPROM contents in about 20 minutes, even less time for 1702/4702 types. Because of varying situations, i.e. age deterioration, dust particles on the EPROM window, centering of the light source, etc., you should run a complete "read" of all the addresses on the newly newly erased EPROM to ensure there is not partial erase on any of the bits. It can be very frustrating to find a partially erased bit at the 1023 location after you've keyed in everything correctly up to that point.

Warning! Although light sources in this range of the spectrum may not look very dangerous or even very bright, it can cause skin cancer under continued exposure. At 2537A, it is well beyond the light spectrum you can see but it will still destroy the cells in your eye, eventually causing blindness. The moral of this story is to clearly label a warning on your lamp fixture and don't try to read with this bulb.

For safety reasons or if you don't have a suitable lamp fixture already you can build an inexpensive unit such as shown in the drawing. It's simply a piece of .032 aluminum folded into an inverted trough with 3/4" hardwood plywood ends mitered as shown to keep the light from escaping at the ends. The wiring is shown in figure A. Simply centre this light source over your EPROMs, turn it on and come back after the required exposure time. If you're building from scratch, be sure to mount the lamp sockets so that there is at least 3/4" clearance between the bottom of the lamp and the base line. You can even save on the sockets by soldering the leads directly to the bulb.●

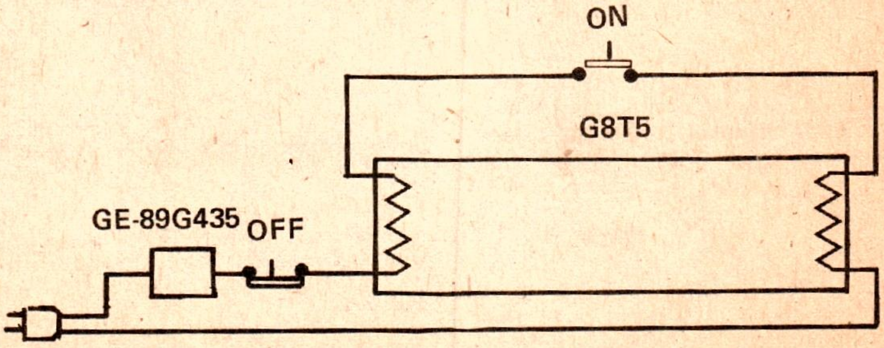
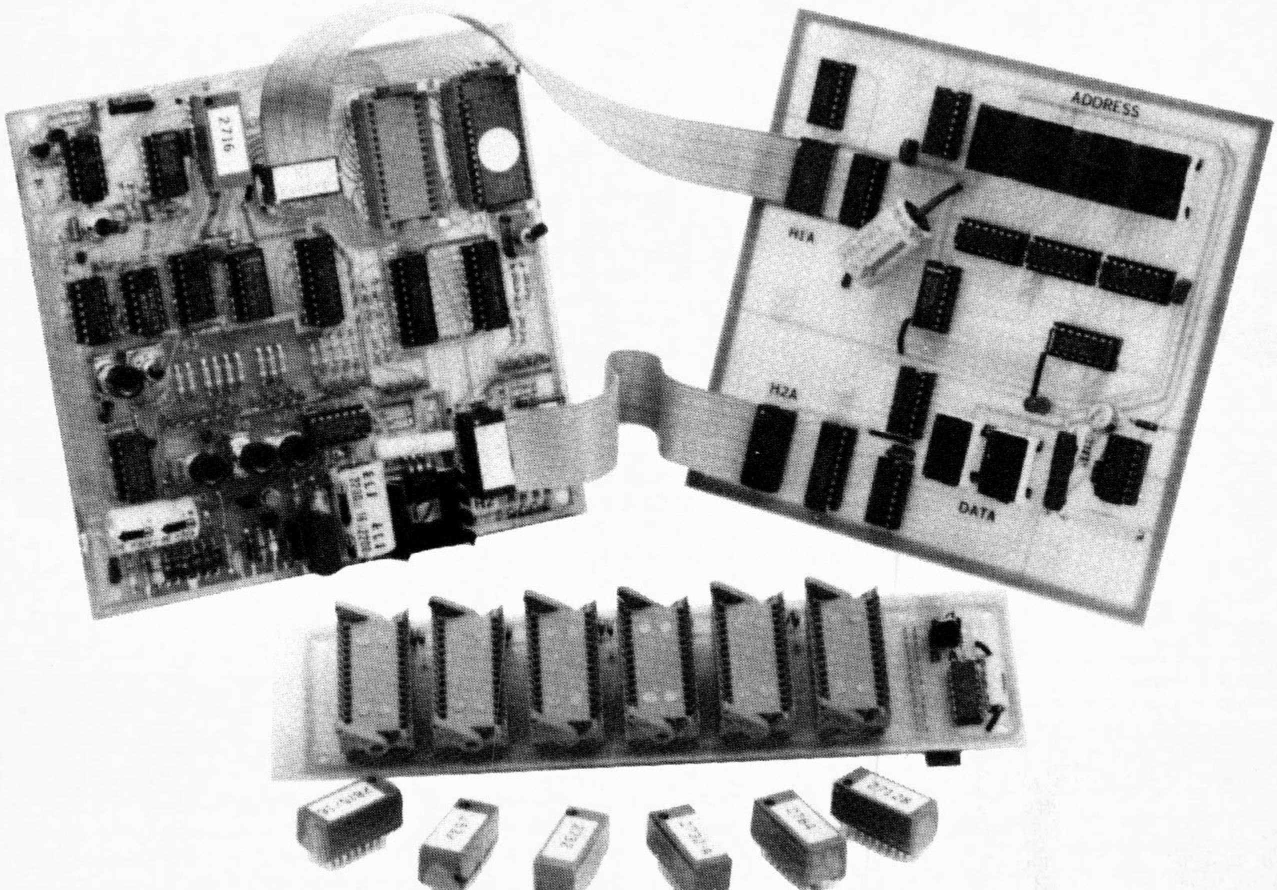


Fig. 2. The schematic.



EPROM PROGRAMMER

Our low-cost EPROM programmer can burn all the popular types, including EEPROM's!

LUBOMIR B. SAWKIW

THE MICROPROCESSOR IS TURNING UP IN some very unlikely places these days. You might find one in a TV remote control, an automotive ignition and timing circuit, a video game, or even a microwave oven. But a microprocessor can't do the work all by itself: It requires permanent memory that stores the data and instructions that allows it to do its job. And that's where the EPROM comes in. It can provide a low-cost way of developing, testing, and even distributing data and programs for many types of computer systems.

EPROM's aren't used solely with computers, of course. Often a complex logic problem that would require numerous individual gates can be solved with a small EPROM.

So, sooner or later, whatever your involvement with digital electronics, you'll find it necessary to *burn (program)* an EPROM. And we've got an inexpensive way of doing so. Our EPROM programmer can be built for about $60 (less PC board) in its basic form, and it can burn all of the popular five-volt EPROM's in both 24- and 28-pin packages, as well as several popular types of EEPROM's. The unit allows you to burn single locations one by one, burn one value into a number of consecutive locations, or copy one entire EPROM to another. An optional multi-EPROM board allows you to program up to six EPROM's simultaneously.

Features

The programmer is a stand-alone unit; no computer or ASCII display terminal is required to operate it. But it has input/output lines (labeled A–G in Fig. 1) that you can use to automate control of all functions.

In the basic programmer, data input is provided by an eight-position DIP switch (S8), and addresses are selected by means of FAST (S5) and STEP (S6) switches. Address and data lines are displayed in binary by 22 discrete LED's (LED1–LED22). An optional display board allows you to view the address and data lines in hexadecimal.

The programmer has a verification feature that allows you to view the contents of each EPROM location after that location has been programmed.

Personality modules are used to accom modate a variety of EPROM's. The per sonality module matches the operatin requirements and pin assignments of eac EPROM to the address, data, progran ming and timing signals developed on th programmer board. By using the appr priate personality module, you can pr gram the 2716, 2732, 2732A, 276

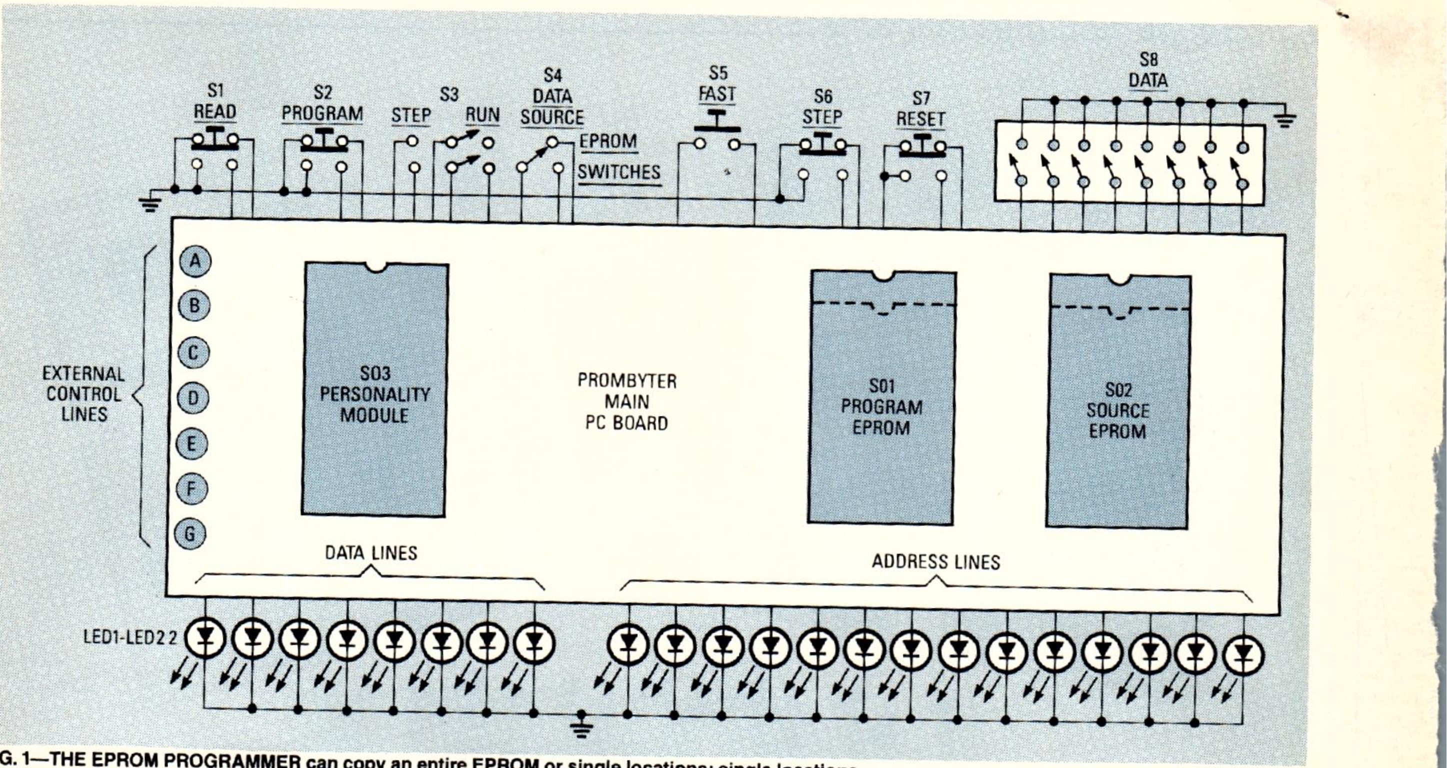


FIG. 1—THE EPROM PROGRAMMER can copy an entire EPROM or single locations; single locations can also be programmed manually. Separate sockets are provided for source and program EPROM's, and for plug-in personality modules that feed each type of EPROM the proper signals.

27128, 2758, 2516, and 2532 EPROM's; the CMOS versions of those devices; and the 2815/2816 EEPROM's.

Manual operation is simple. You set the DIP switches to the desired data, then set the FAST and STEP switches to the desired address, and last press the PROGRAM switch (S2). When that location has been programmed, the programmer automatically moves to the next location.

Circuit description

Except for the AC transformer, the entire power supply is contained on the main PC board. As you can see in Fig. 2, the regulated five-volt supply is derived from an eight-volt AC input. That AC is rectified by bridge BR1, filtered by capacitor C7, and then regulated by IC14, a 7805 regulator. Output capacitor C6 and bypass capacitors C10 through C17 further filter the +5-volt line. Optional resistor R57 provides extra power for the optional display board.

The 25-volt AC input is rectified and filtered for two purposes: to provide the +35-volt programming voltage (V_{PP}), and to provide the 120-Hz clock signal that is the core of the circuit's timing chain. Diodes D4 and D9 isolate the clock circuit from the V_{PP} output. Zener diode D10 clips the positive half-cycles of the unfiltered output at about 9 volts. Then IC7-b and IC7-c square up those pulses.

The programming voltage, V_{PP}, is that "burns" each bit into an erased EPROM. Digital control of V_{PP} is provided by IC7-f and IC7-a. When a high level signal is fed through the personality module (SO3) to pin 13 of IC7-f, the output of IC7-a goes high, Q1 turns on, Q2 turns off, and that allows D2 to bring the base of Q3 up to +25 volts.

At that point Q4 turns on and allows approximately +25 volts to appear at its emitter. That voltage is fed back to pin 8 of SO3. Capacitors C2 and C3 prevent a transient overshoot of the +25-volt line during switching. An overshoot of more than one volt could ruin the EPROM you are programming. Components in some personality modules reduce the +25- to +21-volts, for EPROM's that need +21 volts.

Transistor Q5 is used to prevent V_{PP} pulses from damaging an EPROM when power is removed from or applied to the circuit. As stated previously, Q2 normally conducts and shorts out Zener diode D2. However, Q2 is biased from the +5-volt supply line. If for any reason during power up the +35-volt line receives power an instant before the +5-volt line, Q3's base would shoot up to about +26-volts, and a +25-volt program pulse would appear on the V_{PP} line.

To prevent that surge, Q5's emitter is connected to the +5-volt supply, and its base is biased at +4.7 volts from the voltage at Q3's base by means of Zener diode D3. So, if the base of Q3 were trying to go up to +26-volts, and the +5-volt supply were not quite up yet, Q5 would be forward biased, so it would conduct through R52, and thereby reduce the voltage at the base of Q3.

Address circuit

Overall operation of the circuit is governed by the FUNCTION switch, S3. When that switch is in the RUN position, the 120-Hz clock signal causes operations to occur sequentially every 1/120th of a second. When S3 is in the STEP position, addresses must be set manually, and PROGRAM switch S2 must be pressed for each location that is to be programmed by the EPROM programmer.

Addresses are selected by pressing switches S5 and S6. When switch S6 is pressed, it is debounced by IC5-a, which delivers one pulse through IC6-b and IC6-d to IC8, a 4040 12-stage binary counter. That pulse increments the address held in IC8 by one. On the other hand, when switch S5 is pressed, it connects the 120-Hz clock signal to IC6-b and IC6-d, and then to IC8, which then increments at a rate of 120 Hz.

Since larger EPROM's have as many as 14 address lines, and the 4040 has only 12 outputs, IC3-b is also used as an address counter. When pin 1 of IC8 goes low, IC3-b increments. Its A, B, and C outputs provide the A12, A13, and A14 address lines. Two IC's buffer the current address for display on LED1–LED14: IC9 and IC10.

Reset

Switch S7 forces the system to reset. When that switch is pressed, a reset pulse is fed to all counters and flip-flops. Also, a reset pulse occurs at power up by means of C1's charging up through R8.

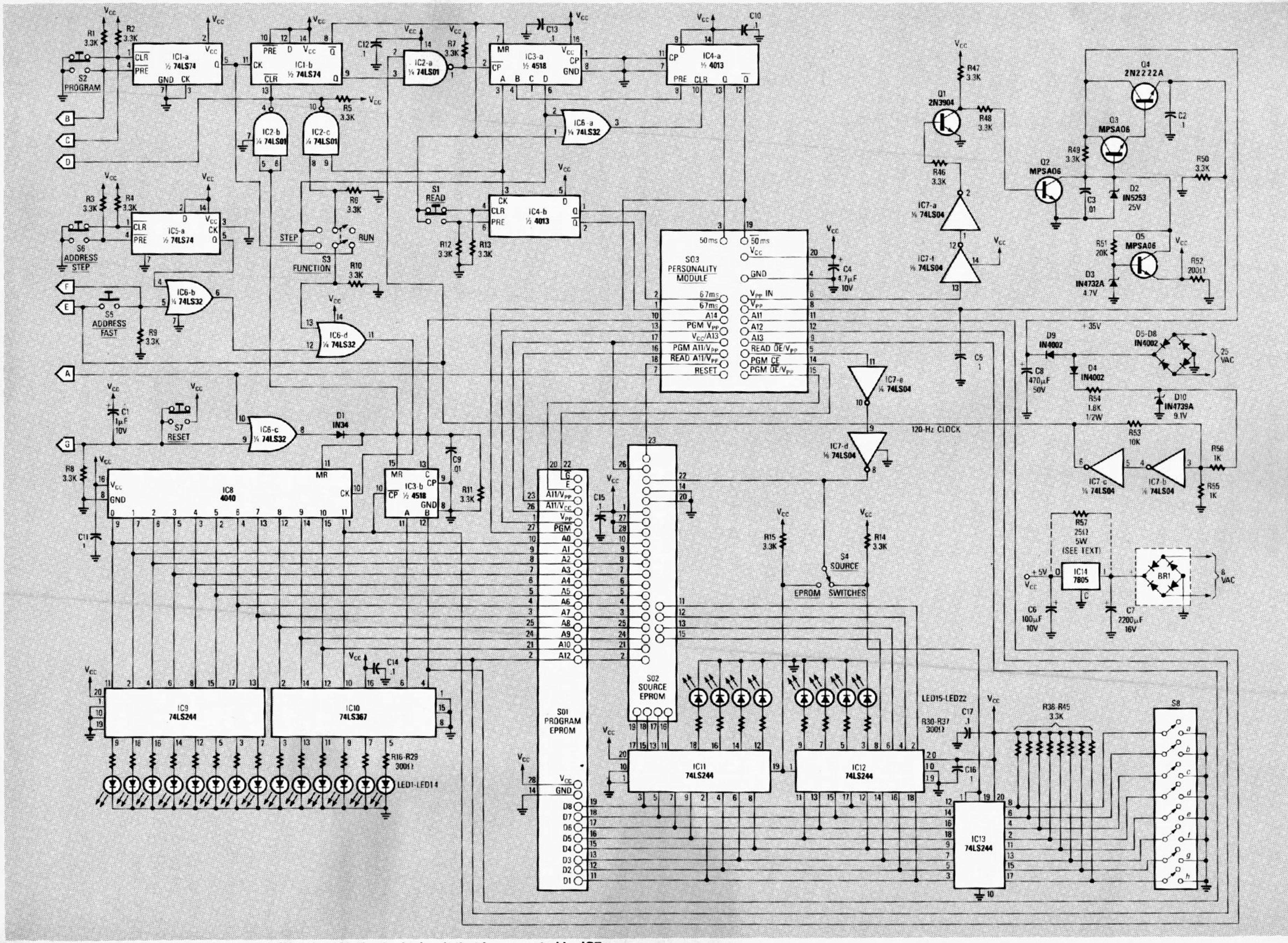


FIG. 2—THE EPROM PROGRAMMER IS DRIVEN BY A 120-Hz clock circuit that is generated by IC7-c [illegible] addresses and data in binary.

Programming circuitry

Assume that S3 is in the STEP position. Then, when S2 is pressed, a program cycle begins. That switch is debounced by IC1-a, which clocks IC1-b. The Q output of of IC1-b allows the 120-Hz clock (at pin 2 of IC2-a) to be fed to IC3-a. The first count to reach IC3-a is fed to IC4-b, which provides the 67-ms data gating pulse. The second pulse presets IC4-a and begins the 50-ms program pulse. Both signals stay high until count 8, which clears IC4-a and ends the 50-ms pulse.

When count 9 arrives NAND gate IC2-c clears IC1-b, which prevents the 120-Hz signal from passing through IC2-a and counter IC3-a. At that time $\overline{Q}$ of IC1-b also goes high, resets IC3-a, and clears IC4-b. That ends both the 50-ms program-pulse and the 67-ms gating pulse.

If S3 is toggled to the RUN position, NAND gate IC2-c never receives the ninth count, so IC1-b is not cleared, and IC3-a continues to count until the tenth pulse, at which time it internally resets to zero. Since there is no reset signal from $\overline{Q}$ of IC1-b, IC3-a continues to count, so IC4-b is not cleared, and the 67-ms pulse line remains high. Also, count eight (available at pin 6 of IC3-a) is used to increment the address counter (IC8) in this mode. If S3 is toggled to STEP, the next count of nine will reset the programming cycle.

Data verification

When S3 is in the STEP position, the 4040 address counter is not incremented until S2 is released and Q of IC1-a goes low. That low-going transition passes through S3 and IC6-d to the clock input of the 4040 and increments the address by one. So, if you hold S2 down after a location is programmed, you can see whether that location has the correct data by examining LED15–LED22. When you're ready to go to the next location, release S2. There is no way to "back up" by one location (or more than one). You'll just have to reset the prom burner, or cycle all the way around through ØØØØ.

Data section

The programmer has an eight-bit data bus. The logic levels of all bits are displayed by LED15–LED22, which are driven by IC11 and IC12. The data displayed by those LED's (and used to program an EPROM) is chosen via switch S4, SOURCE. When S4 is in the EPROM position, the EPROM in SO2 provides data; when S4 is in the SWITCHES position, DIP switches S8-a through S8-h provide data.

Pushbutton switch S1 can be used to view the contents of the source EPROM in SO2. When that switch is pressed, IC4-b is preset, and that activates the 67-ms data gating line, which places the program EPROM in a standby mode. In addition, that signal places the source EPROM in a read mode, and connects its data input pins to the data bus. That allows you to view the contents of the source EPROM before programming.

Component selection

With that basic understanding of the circuit in mind, let's build a programmer now.

Our PC board was designed for miniature, PC-mountable switches like those sold by Alco, C & K, and other companies. But you can use any switches that are functionally equivalent. If you use the PC-mountable switches, you can, with careful planning, secure the PC board directly to the front panel of your enclosure. To do so, just drill holes in the appropriate locations on the panel, and secure the PC board to the panel with the switch-mounting hardware. If you don't want direct-panel mounting, simply run wires from the pads on the PC board to the appropriate terminals on the switches.

In addition, for panel mounting, two 28-pin wirewrap sockets should be mounted at the correct height above the board in the holes provided for SO1 and SO2; then plug the ZIF sockets into those sockets. The same applies to SO3, the socket for the personality modules, and S8, the data input DIP switch. A 20-pin socket is required for SO3, and a 16-pin socket for S8.

Pay attention to the circuit's power requirements. If you are going to use the two option boards, you'll will need 8-volts AC at 1.2 amps and 25-volts AC at 280 mA. The two AC supplies must be separate; they cannot be taps on one winding or in any other way be connected to each other. Two separate windings on one transformer will suffice. Don't apply more than 10 volts to the 8-volt input pads on the board, nor more than 30 volts to the 25-volt pads. In addition, the circuit works only with a power-line frequency of 60 Hz, because the timing circuitry is locked to that frequency. Also, it's not a bad idea to fuse the primary of the transformer. A ¼-amp, 250-volt fuse will do.

Don't use bargain transistors; they can be destroyed at power up. For example, Q4 is a 2N2222A. Make sure your transistor has the A suffix, because a plain 2N2222 is rated for operation at lower voltages. Nor should you use a PN2222, which has limited power dissipation.

The personality modules are built on 20-pin headers which may be hard to find. If you have trouble locating them, you can substitute 20-pin machined-contact, solder-tail IC sockets instead. Those sockets have pins that are sturdy but thin enough to fit into an IC socket.

Although 74LS series IC's were used in our prototype, you may want to experiment with 74HC and 74HCT devices. They are CMOS versions of the 74LS series, and they feature speed and drive capacity comparable to those of the 74LS series, but with the advantage of CMOS's low power consumption.

Last, use good quality LED's. We have found that LED's vary greatly in quality and light output. Some hobbyist-grade LED's require a great deal of current to produce much light, and the 7805 can't provide a great deal of current and drive all the other circuitry. So stay with prime LED's or get high-brightness LED's.

Construction

Due to the complexity of the project, we recommend that you use a PC board to build the programmer. Foil patterns for one side of double-sided board are presented in PC Service; the other side will appear in next month's issue. A pre-etched and drilled board is also available from the source in the Parts List.

If you etch your own board, it likely will not have plated-through holes. In that case use wirewrap IC sockets mounted a little above the board so that you can solder each pin to both sides of the board. Make sure that you also solder all other components on both sides of the board.

Referring to Fig. 3, mount all components on the board as follows. First solder the two jumpers to the board using insulated wire for each. One is located to the left of Q3 in the lower left corner of the board; the other is located to the left of IC8 in the upper center of the board.

Begin soldering parts to the board, starting with the lower-profile components (resistors and diodes) and working up to the larger components (C8, SO1, SO2, and all switches). All the discrete LED's should be mounted so that their cathodes (usually the flat side) face IC14, the 7805. Be careful to mount all polarized components correctly.

Personality modules

Shown in Fig. 4 are the modules for each of the EPROM types mentioned above. Some of the modules are very simple; others require several components. We'll give construction hints only for the more complicated modules. After you verify that each module works correctly, pot it with epoxy, mark pin 1, and label it with the type of EPROM it is used with.

The 2532, 2732, and 2372A modules, shown in Fig. 4-*c*–Fig. 4-*e*, are the only ones that are hard to build. When building them, wire the jumpers and the discrete components first, and then install the 4001 CMOS IC. Break off pins 4, 10, and 11 of the IC. Then bend pin 3 under the IC and solder an insulated jumper to it; that jumper will connect later to pin 15 (for the 2532) or pin 14 (for the 2732 and the 2732A) of the module.

Short the unused input pins (5, 6, 8, 9, 12, and 13) to pin 7, and connect that pin to pin 4 (ground) of the module. Finally, bend IC pins 1, 2, and 14 so that they can

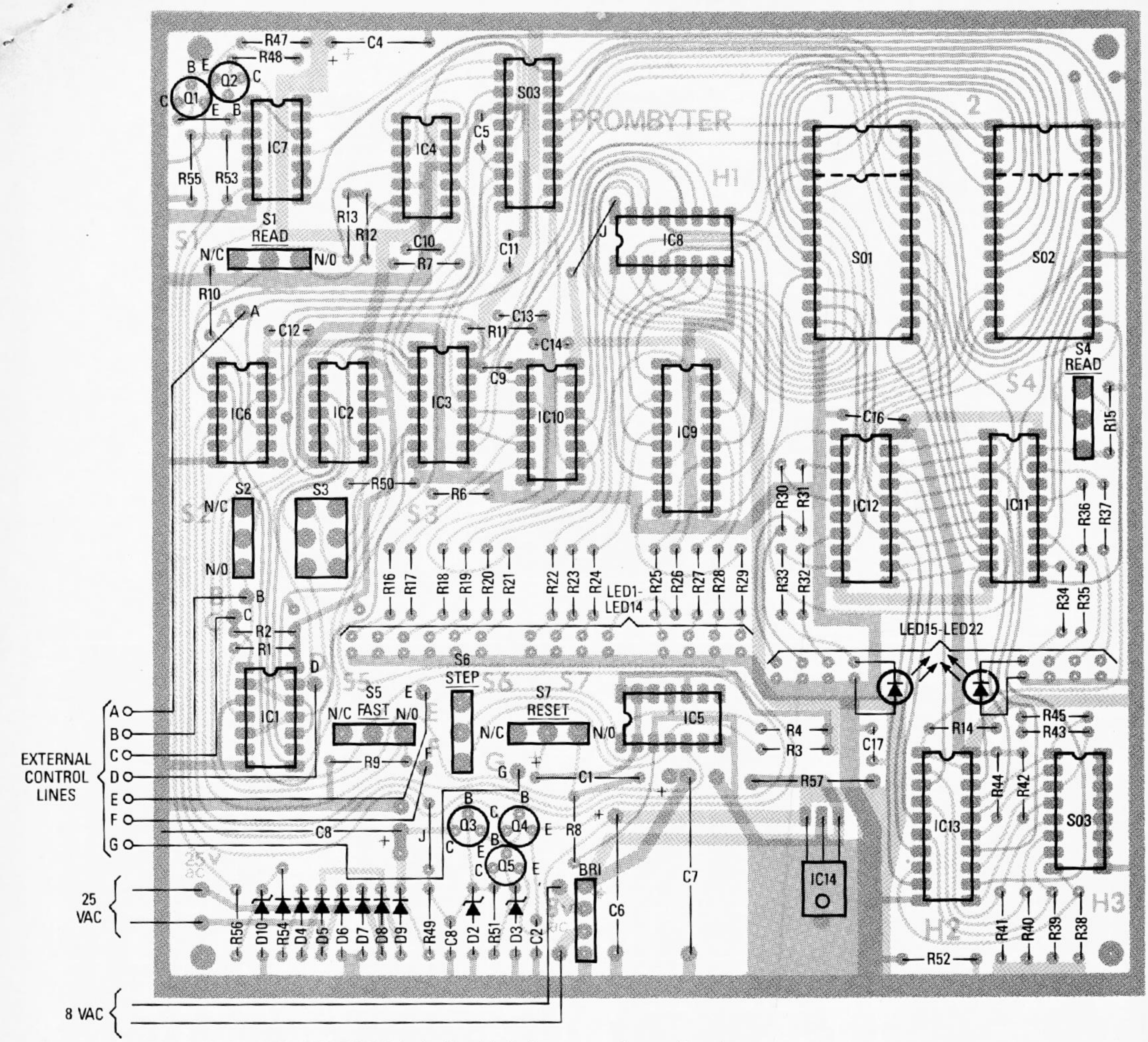


FIG. 3—THE EPROM PROGRAMMER'S PC BOARD is double-sided; all components mount as shown here. Since a home-made board's holes are not plated through, the IC sockets must be mounted slightly above the board so that the socket pins can be soldered to the top side of the board.

PARTS LIST

All resistors are ¼-watt, 5% unless otherwise noted.

R1–R15, R38–R50—3300 ohms
R16–R37—300 ohms
R51—20,000 ohms
R52—200 ohms
R53—10,000 ohms
R54—1800 ohms, ½ watt
R55, R56—1000 ohms
R57—25 ohms, 5 watts (see text)

Capacitors

C1—1 μF, 10 volts, electrolytic
C2, C5, C10–C17—0.1 μF, disk
C3, C9—0.01 μF, disk
C4—4.7 μF, 10 volts, electrolytic
C6—100 μF, 10 volts, electrolytic
C7—2200 μF, 16 volts, electrolytic
C8—470 μF, 50 volts, electrolytic

Semiconductors

IC1, IC5—74LS74 dual flip-flop
IC2—74LS01 quad 2-input NAND gate
IC3—4518 CMOS dual BCD counter
IC4—4013 CMOS dual flip-flop
IC6—74LS32 quad 2-input OR gate
IC7—74LS04 hex inverter
IC8—4040 12-stage binary counter
IC9, IC11–IC13—74LS244 octal 3-state buffer
IC10—74LS367 hex 3-state buffer
IC15—7805 5-volt regulator
BR1—1-amp 50-PIV bridge rectifier
D1—1N34A germanium signal diode
D2—1N5253 25 volts, 1 watt, Zener diode
D3—1N4732A 4.7 volts, 1 watt, Zener
D4–D9—1N4002 rectifier
D10—1N4739A 9.1 volts, 1 watt, Zener diode
LED1–LED22—miniature, high-brightness LED
Q1—2N3904 NPN transistor
Q2, Q3, Q5—MPSA06 NPN transistor
Q4—2N2222A NPN transistor

Other components

S1—SPDT, toggle, momentary
S2, S6—SPDT, pushbutton, momentary
S3—DPDT, toggle
S4—SPDT, toggle
S5, S7—SPST, pushbutton, momentary
S8—8-position DIP switch

Miscellaneous: Dual-secondary transformer: 8-volts at 1.2 amps, and 25-volts at 280 mA; heatsink for IC14; two 28-pin ZIF sockets; IC sockets, wire solder, case, etc.

Note: The following are available from Lubomir Sawkiw, P.O. Box 555, Rensselaer, NY 12144: A transformer with 25 and 10 volt AC secondaries, $8.00 plus $3.00 shipping and handling; 9368 Fairchild IC's, $4.25 each postpaid. New York residents must add 7% sales tax.

The following are available from E²VSI, P.O. Box 72100, Roselle, IL 60172: main circuit board, $25.00; hex display board: $15.00; ga[redacted]board, $10.00; set of three boards, [redacted].00.

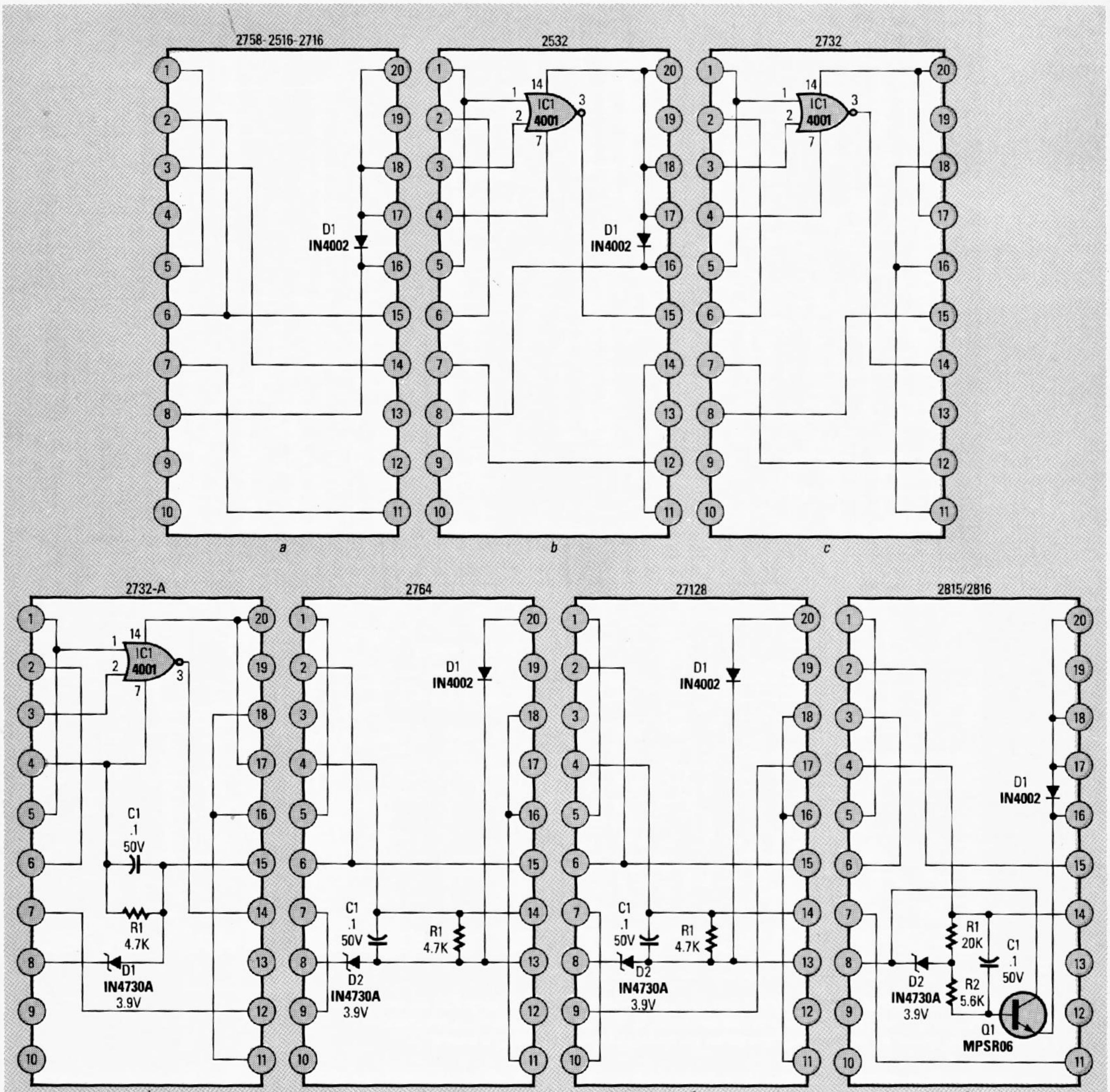


FIG. 4—EACH PERSONALITY MODULE is built on a 20-pin header; see the text for information on building the 2532, 2732, and 2732A modules.

attach directly to module pins 1, 3, and 20, respectively. When soldered in place correctly, the IC should sit firmly above the carrier wiring and components.

Initial check-out

Install all IC's and a personality module in the proper sockets, but don't install an EPROM yet. Measure the +5-volt DC output of the regulator IC as you power the board. To measure that (or any other) voltage on the board, do not clip the negative lead of your meter to the edge of the board; you would short out the power supply. Rather, connect your negative test lead to the negative side of C6, C7, or C8.

After checking the +5-volt supply, measure V_{PP} across C8. The voltage there shouldn't exceed 40 volts. You need only about 27 volts to do the job; a higher voltage could destroy one of the transistors.

Now remove power and install an erased EPROM into SO1. Whether you use a 24-pin device (such as a 2716) or a 28-pin device (such as a 2764), the EPROM should be "bottom justified." In other words, pin 12 of a 24-pin device, or pin 14 of a 28-pin device should be plugged into pin 14 of the socket. After the EPROM is oriented properly, move the socket's locking lever to the closed position.

With power still off, insert the eight-position DIP switch in its socket. Mount it so that switch 1 is closest to the EPROM sockets. That places data bit Ø, on switch one and data bit 7 on switch 8. A closed DIP switch puts a low (logic Ø) on that line. An open switch puts a high (logic 1) on that line. At this point you should have an erased EPROM, the corresponding personality module, and the DIP switch inserted in the correct sockets. Place S3 in the STEP position, apply power, and press RESET (S7).

Next time we'll present more testing details, as well as plans for the optional display and gang-programming boards. **R-E**

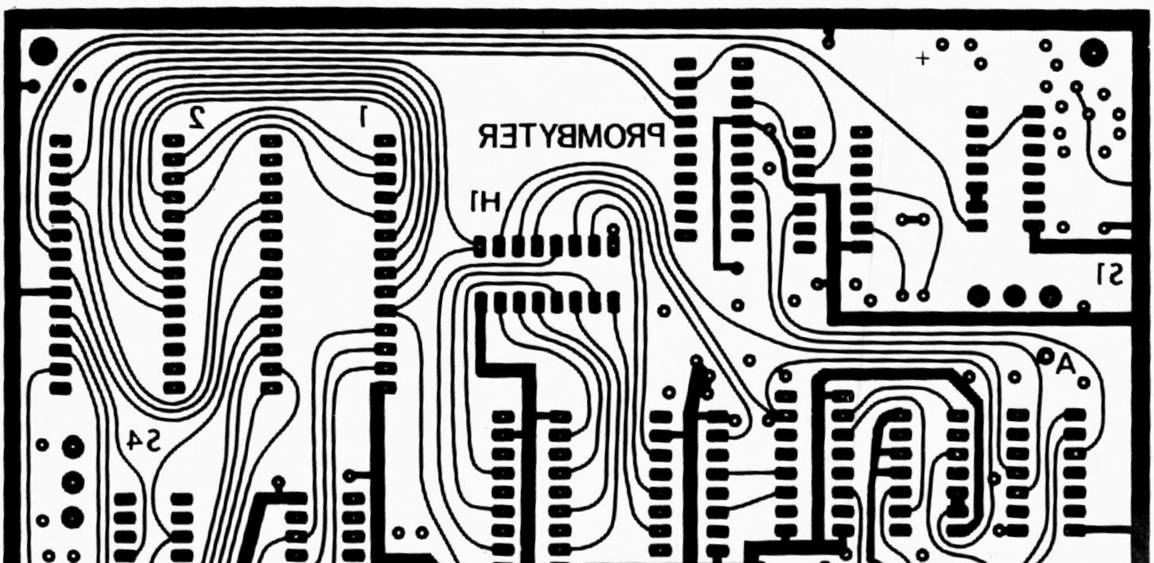
PC SERVICE

across the back of the artwork. That helps make the paper transluscent. Don't get any on the front side of the paper (the side with the pattern) because you'll contaminate the sensitized surface of the copper blank. After the oil has "dried" a bit—patting with a paper towel will help speed up the process—place the pattern front side down on the sensitized copper blank, and make the exposure. You'll probably have to use a longer exposure time than you are used to.

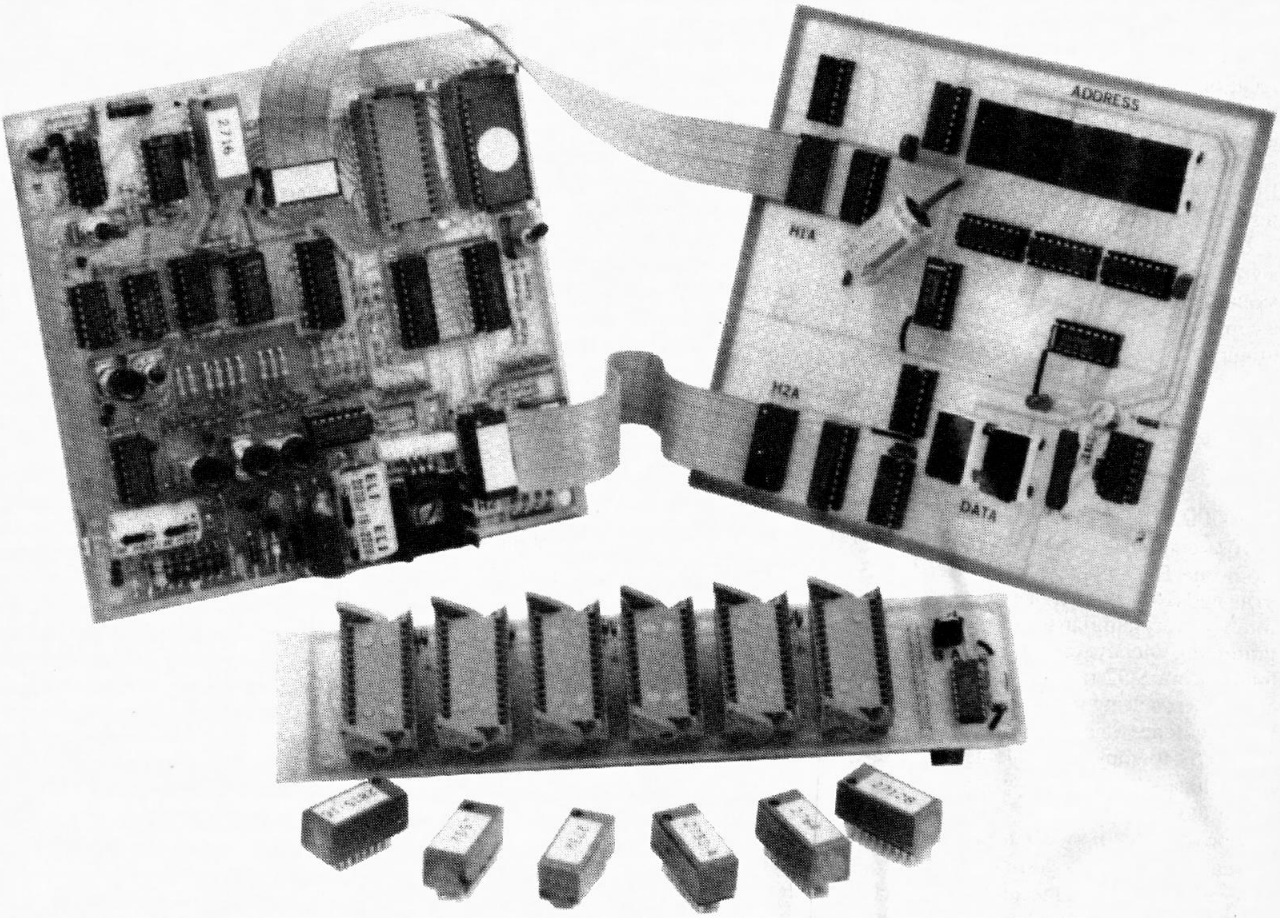
We can't tell you exactly how long an exposure time you will need as it depends on many factors but, as a starting point, figure that there's a 50 percent increase in exposure time over lithographic film. But you'll have to experiment to find the best method for you. And once you find it, stick with it.

Finally, we would like to hear how you make out using our method. Write and tell us of your successes, and failures, and what techniques work best for you. Address your letters to:

Radio-Electronics
Department PCB
500-B Bi-County Blvd.
Farmingdale, NY 11735



THE COMPONENT SIDE of the EPROM programmer PC board is shown here. Due to space limitations, the solder side will be shown next time.



EPROM PROGRAMMER

This time we discuss the programmer's many modes of operation and hardware options that increase its ease of use.

LUBOMIR SAWKIW

Part 2 WE HAD JUST FINISHED building and testing the programmer when we left off last time. Let's continue now and explain the programmer's four modes of operation. Then we'll discuss adding an optional gang-programming board that allows you to program as many as six EPROM's at once, and a display board that allows you to view addresses and data in hexadecimal on seven-segment LED's.

Manual programming

The programmer's modes of operation, and the switch settings that enable them, are summarized in Table 1. To program a byte manually, place S3 in the STEP position and S4 in the SWITCHES position. Set the data switches in DIP switch S8 to the desired values and then set the address using S5 and S6, as discussed last time. If the byte at that address is properly erased, all eight DATA LED's (LED15–LED22) will light up (indicating a value of FF hex).

TABLE 1—MODES SELECTION

Mode	Switch Settings S3	S4
Manual	Step	Switches
Copy EPROM	Run	EPROM
Copy byte	Step	EPROM
Repeat program	Run	Switches

If that location is erased, you can press PROGRAM switch S2 to program the new value. The programming occurs in 50 ms, and the new data will be visible for as long as you continue to press S2. If the LED's indicate an incorrect value, the EPROM may not be fully erased, or it may be damaged. Try erasing the EPROM; and if problems persist, try a new one.

When you release S2 after programming a location, the board advances to the next address. You can program that address or use S5 and S6 to go to another.

Copying

To copy an entire EPROM, place S4 i

the EPROM position and S3 in the RUN position. Then reset the circuit by pressing S7. Now place your EPROM's in their sockets. The source EPROM goes in SO2, and the blank EPROM in SO1. *Don't reverse them!*

To start the copying process, press PROGRAM switch S2. To stop before all locations are programmed, set S3 to the STEP position. Doing so will halt the copy cycle and place the programmer in the single-location copy mode. You can now continue to program the EPROM one location at a time. Later you can resume copying automatically, if desired. When the entire EPROM has been copied, the board will reset and await a new command.

It's not a good idea to use the RESET switch to halt the copying process because stopping the programmer in the middle of a copying cycle may leave a partially-programmed byte in your EPROM. The proper way to halt the copying procedure is to place S3 in the STEP position. Then the programmer will stop after the current byte has been programmed.

Copying bytes

If you want to copy a single location from one EPROM to another, set S3 to STEP and S4 to EPROM. The byte-copying mode is very similar to the manual-programming mode except that data is obtained from SO2 rather than DIP switch S8. After setting the desired location using switches S5 and S6, press PROGRAM switch S2 to copy the contents of the selected location to your program EPROM. As in the manual mode, the contents of the programmed location will be displayed for as long as you hold S2 down. After you release S2, the address counters advance by one. You can then program that location, or move to any other, using S5 and S6.

Switch S1 is handy when you are copying an EPROM one location at a time. Normally, the data LED's (LED15–LED22) display the contents of the program EPROM. But when you press S1, that switch allows you to view the contents of the source EPROM.

Repeat programming

The repeat mode allows you to program successive EPROM locations with the data byte in DIP switch S8. Why would you want to program many locations with the same byte? A delay or timing loop might require one byte to be programmed into a number of locations. You could also use repeat programming to erase an EEPROM.

To enter the repeat program mode, set switch S3 to RUN and set switch S4 to SWITCHES. Then set the switches in S8 to the desired value. As usual, to begin programming, press PROGRAM switch S2. To halt programming, move S3 to the STEP position. Now you can proceed in the manual programming mode. As mentioned earlier, do not use RESET switch S7 to halt the programmer.

To erase a 2815 or 2816 EEPROM, plug the appropriate personality module into SO3. Then set S8 to FF (hex), S3 to RUN, and S4 to SWITCHES. Reset the board and then press S2 to start. In about 2 minutes you will have a blank EEPROM.

2532 limitation

The programmer is capable of *copying* only the lower half of a 2532 EPROM. That is because pin 20 of the source EPROM socket (SO2) is grounded. The 27xx series of EPROM's uses that pin for chip enable ($\overline{CE}$). However, the 2532 uses that pin for address line A11. Since the highest-order address line is held low, only the first half of the EPROM can be read, and, therefore, copied. Pin 20 of SO1 is not grounded, so the upper half of a 2532 may be programmed manually.

External control

As we saw in Fig. 2 last time, there are 7 points labeled A through G on the board. The signals there may be used to control the programmer with an external device. The PC board must be modified slightly, and the switches must be set in certain positions to control the programmer externally.

Remove DIP switch S8 from its socket, and connect your data source at that point. Switch S4 must, of course, be in the SWITCHES position. In addition, S3 should be in the STEP position, and switches S1, S2, S5, S6, and S7 are not used. S2 should be removed from the board, or the ground trace to its center pole should be cut.

An active-low signal fed to point B presets IC1-a and starts the programming cycle. There must be no noise, glitches, or bouncing at that input. If the input signal isn't clean, you could program ten locations with what you thought was one pulse. After each location is programmed, a low must be fed to point C to clear IC1-a. The circuit shown in Fig. 5 may be used; that circuit can save you the trouble of having to supply a separate clear signal to point C. Just make sure that your input signal stays low for at least 67 ms, or else you will increment the address before the programmer finishes programming the current location.

Point D can be used as both an input and an output. When S3 is in the STEP position, point D carries a narrow active-low pulse at the end of a programming cycle when IC3-a reaches the count of nine. That point can be used to indicate when the board has completed programming one location and is ready for another cycle.

When S3 is in the RUN position, point D carries a narrow low-going pulse when the board has programmed an EPROM's last

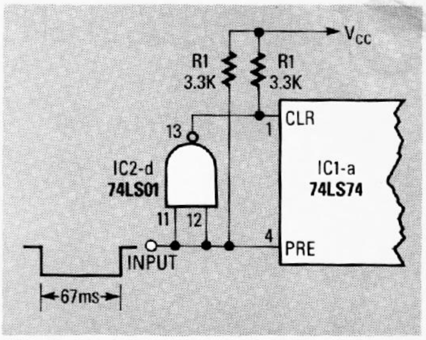


FIG. 5—TO AUTOMATE PROGRAMMING, connect the unused gate of IC2-d as shown here. Your control signal should be low-going and have a width of at least 67 ms.

address. That signal can be used to indicate when a copy operation is complete.

In addition, bringing point D low at any time during the programming cycle will clear IC1-b and will abort the programming operation.

You can increment the current address location by one by delivering a high-going pulse to point F. Point E has the 120-Hz clock signal that drives the timing chain. Your controller can connect that signal to point F for fast address stepping, or you can pulse point F with your own signal.

A high-going pulse at point G will reset the entire programmer. Point A provides a reset output from the programmer; that point goes high after the address counter has incremented past the highest address of the selected EPROM.

Gang-programming option

The programmer can be expanded to program as many as six EPROM's simultaneously; the circuit for doing so is shown in Fig. 6. In order to avoid possible data-bus contention, the board has logic that (optionally) disables the verification capability for sockets two through six. In other words, only the EPROM in SO1 would be read during verification. That feature was included because, if one or more of the EPROM's in SO2–SO6 were defective or did not program properly, both highs and lows could be present on the data bus simultaneously. An EPROM (especially a CMOS type) could be damaged thereby, but, even worse, a bad or misprogrammed EPROM might pass verification and wreak havoc later. To avoid that possibility, if you gang-program EPROM's, we recommend that you verify each one separately at a later time.

Notice the terminals labeled A, B, and C in Fig. 6. With A connected to B, SO2–SO6 will not be read during verification. But if you should wish to read all six sockets simultaneously, connect A to C.

The parts-placement diagram for the gang-programming board is shown in Fig. 7; the foil patterns for the PC board are shown in PC Service. Note that all components except PL1 mount on the component side of the board. You probably won't be able to find a 28-pin male IDC (*I*nsulation *D*isplacement *C*onnector), so use a 34-pin model and cut off pins 29–34. Then solder it to the board.

A cable that connects plus PL1 to socket SO2 (on the main PC board) is built using a 28-pin DIP socket header on the end that goes to SO2 on one end and a 34-pin female IDC header on the other end.

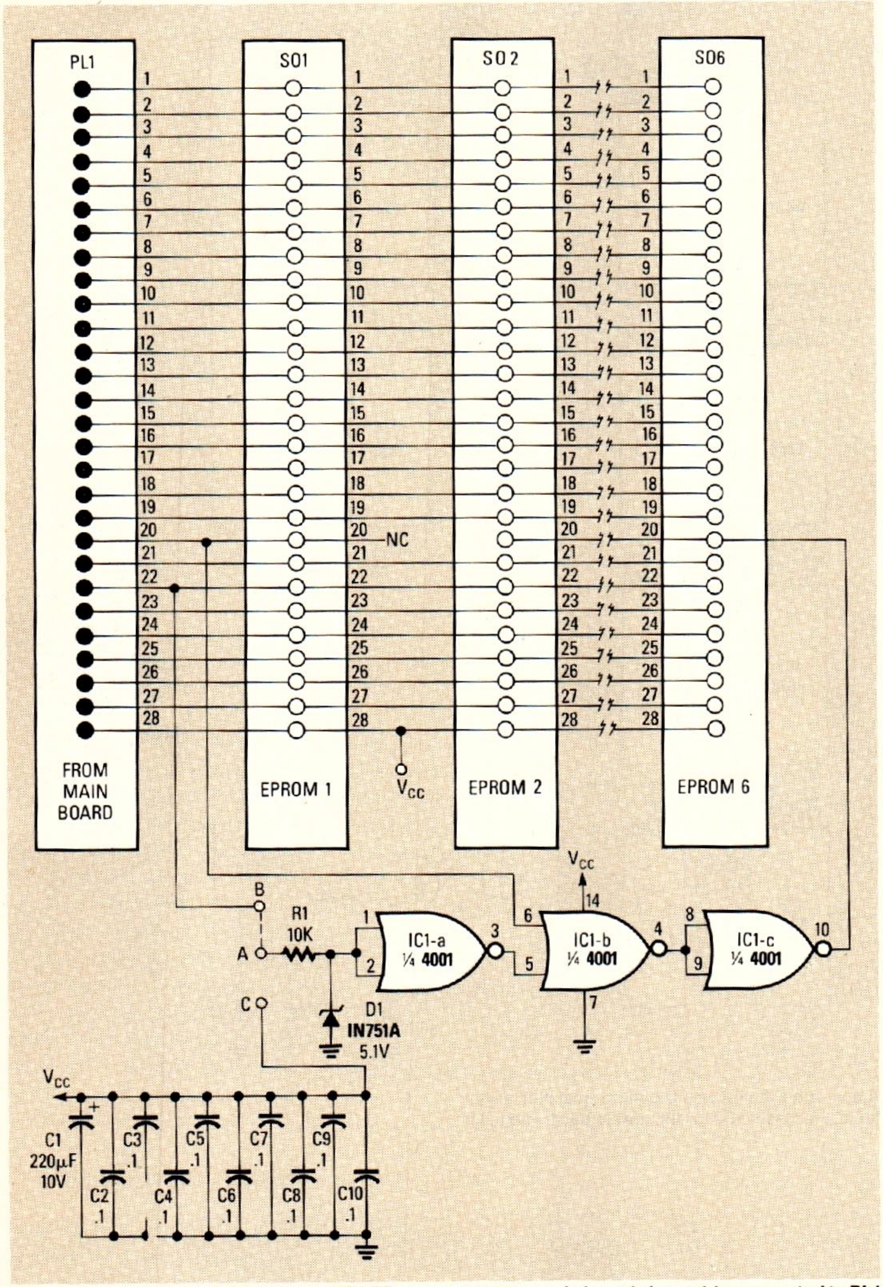


FIG. 6—THE GANG-PROGRAMMING BOARD connects to the main board via a cable connected to PL1. To avoid the possibility of data-bus conflict, the gates of IC1 allow you to disable reading EPROM's in sockets SO2–SO6 (when the jumper is connected between A and B).

GANG BOARD PARTS LIST

R1—10,000 ohms, ¼ watt
C1—220 µF, 10 volts, electrolytic
C2–C10—0.1 µF, 50 volts, disc
IC1—4001 quad NOR gate
D1—1N751A 5.1-volt, 1-watt Zener diode
SO1–SO6—28-pin zero insertion force sockets
PL1—34-pin PC-mount right-angle IDC connector
34-pin female IDC cable connector
28-pin IDC DIP plug
ribbon cable

Since there will be only 28 wires coming from the DIP end, be sure to leave a gap at the correct end of the female connector. Of course, you could simply solder the wires at both ends of the cable directly to the board but a cable is neater an affords you greater flexibility and we highly recommend that you use one.

There is no difference in operation when using the gang-programming board.

Hexadecimal display option

The circuit for the display board is shown in Fig. 8. The display board's operation is straightforward. IC2 on the display board replaces IC8 on the main board by means of a 16-conductor cable connected from socket SO1 on the display board to IC8's socket on the main board. The outputs of the two counters (IC1 and IC2) on the display board drive the display decoder/drivers (IC5–IC8), which in turn drive the address displays (DISP1–DISP4) directly.

You'll notice that there are no current-limiting resistors connected to any of the displays. Instead, Q1, driven by op-amp IC12, functions as a dynamic resistor that keeps the voltage across the segments of the display constant, thereby maintaining constant brightness. Any variation in display brightness causes a corresponding change in the brightness of LED2, which changes Q2's base bias. That varies the voltage at the non-inverting input of IC12-a, which varies Q1's bias, and thus its effective resistance to ground. Trimmer potentiometer R4 allows you to adjust the brightness of the display.

The transistor used for Q2 can be al-

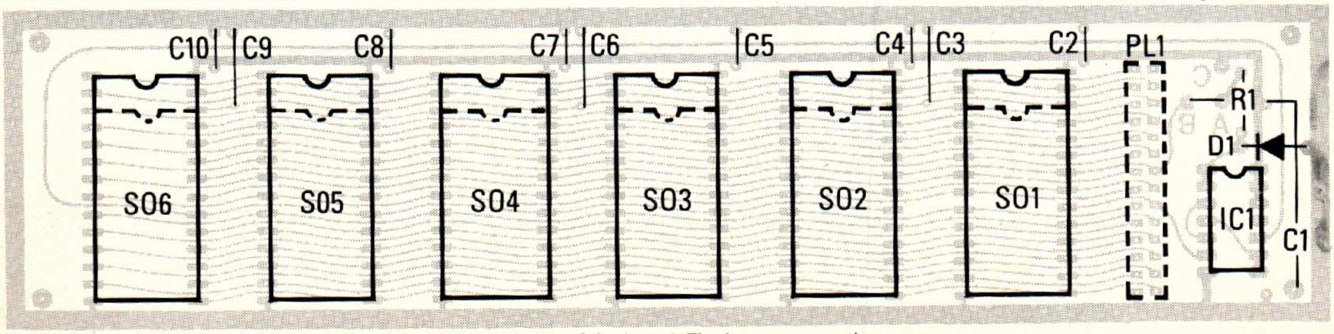


FIG. 7—MOUNT ALL PARTS except PL1 on the component side of the board. The jumper conncting points A and B disables reading from sockets SO2–SO6.

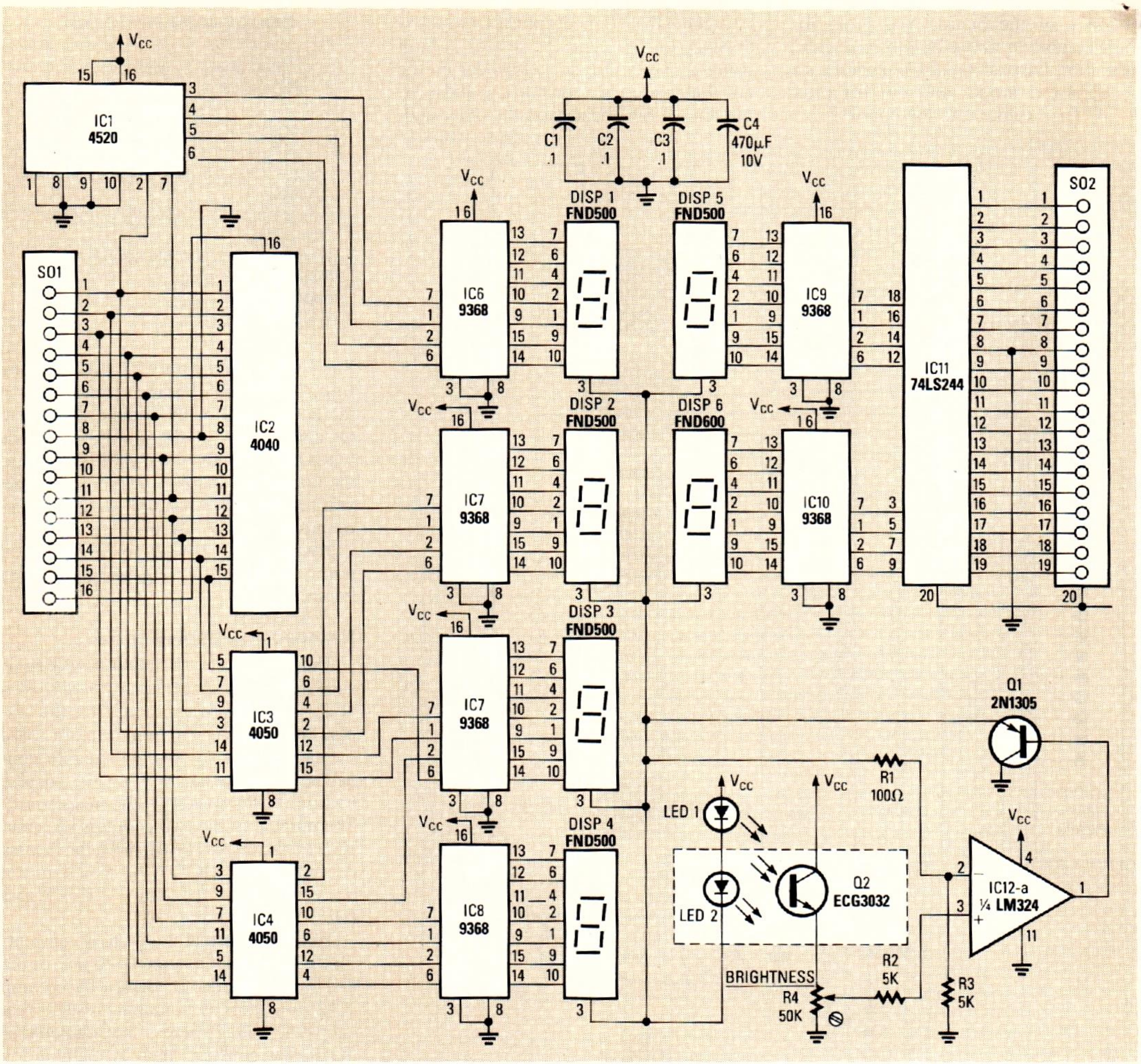


FIG. 8—THE DISPLAY BOARD attaches to the address bus and data bus on the main board through cables connected to SO1 and SO2, respectively. Counter IC2 replaces IC8 on the main board, and IC11 replaces IC13.

most any NPN phototransistor that will respond to the LED driving it. Use a good quality, efficient LED for LED2. You can mount it face to face with Q2 inside a short piece of heat-shrink tubing, or you can simply tape them together. Whichever method you use, keep the two parts very close to each other, and keep out all external light.

The data-bus display circuit is even simpler than the address-bus circuit. IC11 on the display board replaces IC13 on the main board via a 16-conductor cable that connects SO2 on the display board to IC13's socket on the main board. The outputs of IC11 drive a pair of 9368's, which drive two more seven-segment displays (DISP5 and DISP6).

Mount all components on the display board as shown in Fig. 9. The foil pattern for that board is shown in PC Service. Note that there are three jumpers on the board. Mount them first, then the re-

DISPLAY BOARD PARTS LIST

All resistors are ¼-watt, 5% unless otherwise noted.
R1—100 ohms
R2, R3—5000 ohms
R4—50,000 ohms, trimmer potentiometer, 20 turns, PC mount
Capacitors
C1–C3—0.1 µF disc
C4—470 µF, 10 volts, electrolytic
Semiconductors
IC1—4520 dual binary counter
IC2—4040 12-stage binary counter (from main board)
IC3, IC4—4050 hex buffer
IC5–IC10—9368 decoder/display driver
IC11—74LS244 octal tristate driver (from main board)
IC12—LM324 op amp
Q1—2N1305
Q2—ECG3032 or Radio Shack 276-130
LED1, LED2—Standard red LED
DISP1–DISP6—FND500 common-cathode, 7-segment LED display
Miscellaneous: 20-pin DIP-socket jumper cable, 16-pin DIP socket jumper cable.

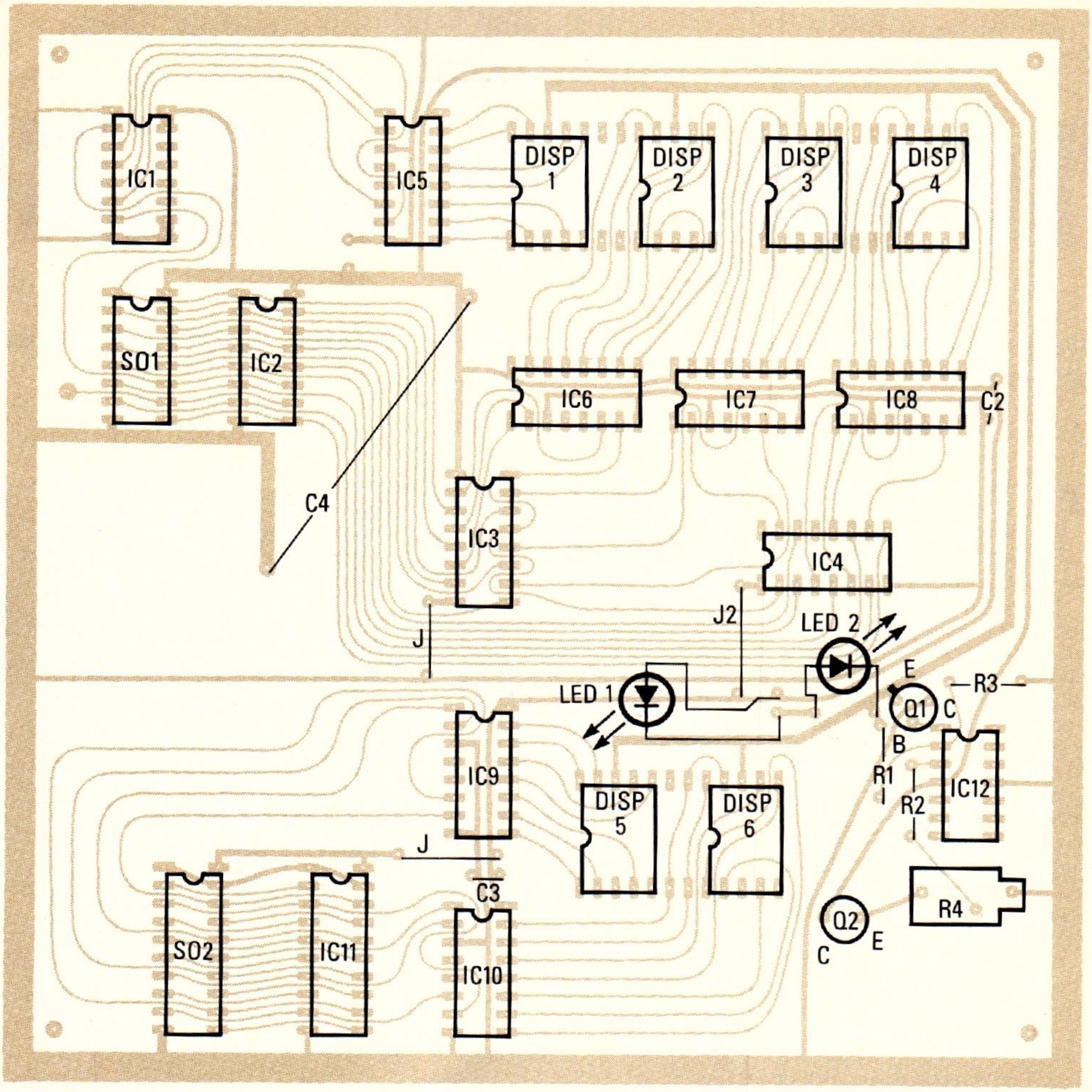


FIG. 9—PHOTOTRANSISTOR Q2 AND LED2 must be mounted so that their receptive and transmissive surfaces receive no ambient light. Heatshrink or tape them together. The foil pattern for this board is shown in PC service.

sistors, and so on. Its best to use sockets for all IC's; the six LED displays can be mounted in 24-pin 0.6" DIP sockets. A red filter will improve their readability.

ORDERING INFORMATION

The following are available from Lubomir Sawkiw, P.O. Box 555, Rensselaer, NY 12144: A transformer with 25 and 10 volt AC secondaries, $8.00 plus $3.00 shipping and handling; 9368 Fairchild IC's, $4.25 each postpaid. New York residents must add 7% sales tax.

The following are available from E²VSI, P.O. Box 72100, Roselle, IL 60172: main circuit board, $25.00; hex display board: $15.00; gang board, $10.00; set of three boards, $45.00.

To install the display board, remove IC13 from the main board and install it in the space allocated for IC11 on the display board. Then connect a 20-pin DIP jumper cable between socket SO2 on the display board and IC13's socket on the main board.

You could also solder all of connections directly to the appropriate points on the two boards, eliminating the need for a DIP cable or socket. In either case, make absolutely certain that every pin is connected correctly.

Remove IC8 from its socket on the main board and install the IC in the socket provided for IC2 on the display board. Connect a 16-conductor DIP jumper between SO1 on the display board and IC8's socket on the main board. Make sure that the correct pins line up on both ends. Do not plug the jumpers in or remove them (or IC's) while power is applied.

Remove IC9 and IC10 from the main board. Because they drive the 14 address LED's (LED1–LED14), they're no longer necessary. Removing the IC's will conserve some much needed power for the display board.

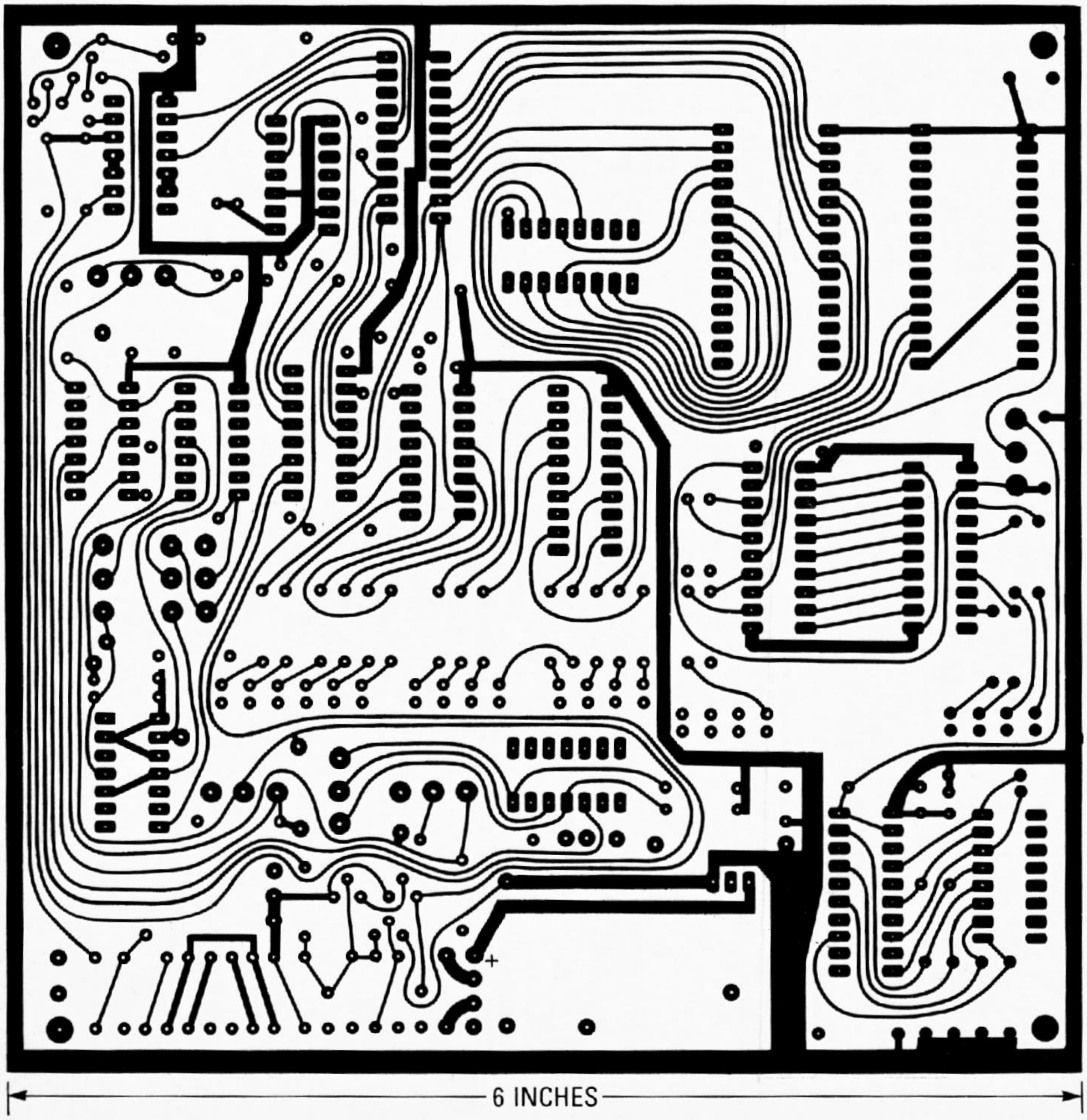
The voltage regulator (IC14) on the main board can normally run quite warm—as high as 70°C maximum. However, with the display board plugged in we would be pushing the regulator to the limit. So, to let it run a little cooler, solder a 25-ohm, 5-watt power resistor (R57) to the main board in the pads provided near the regulator. If you add that resistor, be sure that you do not run the programmer without the display board connected. **R-E** general, all the kinds of things you look for in the final etched board. You can clean up the published artwork the same way you clean up you own artwork. Drafting tape and graphic aids can fix incomplete traces and doughnuts, and you can use a hobby knife to get rid of bridges and dirt.

An optional step, once you're satisfied that the artwork is clean, is to take a little bit of mineral oil and carefully wipe it across the back of the artwork. That helps make the paper transluscent. Don't get any on the front side of the paper (the side with the pattern) because you'll contaminate the sensitized surface of the copper blank. After the oil has "dried" a bit—patting with a paper towel will help speed up the process—place the pattern front side down on the sensitized copper blank, and make the exposure. You'll probably have to use a longer exposure time than you are used to.

We can't tell you exactly how long an exposure time you will need as it depends on many factors but, as a starting point, figure that there's a 50 percent increase in exposure time over lithographic film. But you'll have to experiment to find the best method for you. And once you find it, stick with it.

Finally, we would like to hear how you make out using our method. Write and tell us of your successes, and failures, and what techniques work best for you. Address your letters to:

Radio-Electronics
Department PCB
500-B Bi-County Blvd.
Farmingdale, NY 11735



THE SOLDER SIDE of the EPROM programmer's main board. The component side was shown last month.

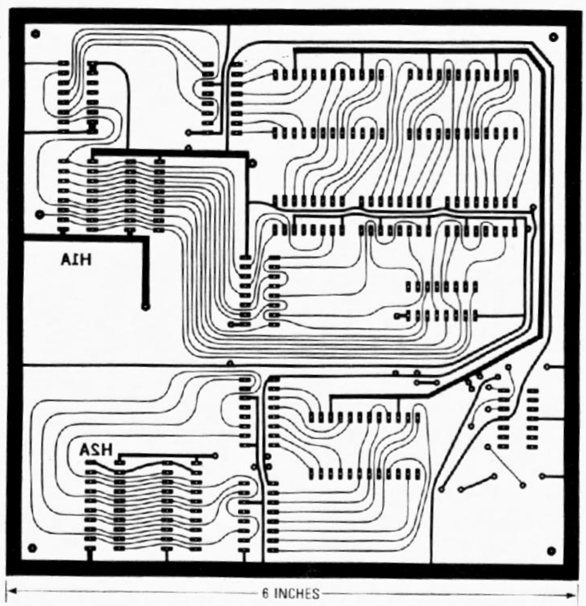
PC SERVICE

One of the most difficult tasks in building any construction project featured in **Radio-Electronics** is making the PC board using just the foil pattern provided with the article. Well, we're doing something about it.

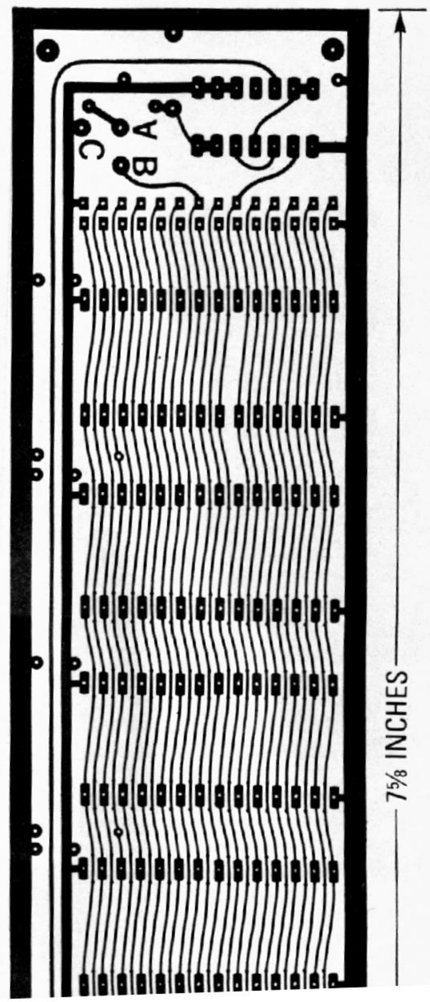
We've moved all the foil patterns to this new section where they're printed by themselves, full sized, with nothing on the back side of the page. What that means for you is that the printed page can be used directly to produce PC boards!

Note: The patterns provided can be used directly only for *direct positive photoresist methods.*

In order to produce a board directly from the magazine page, remove the page and carefully inspect it under a strong light and/or on a light table. Look for breaks in the traces, bridges between traces, and in



ADD A HEX DISPLAY BOARD to your EPROM programmer using this PC board. Note that the board is shown half sized. It will have to be enlarged photographically before it can be used to produce the board.



THE GANG PROGRAMMING BOARD for the EPROM programmer is shown here.

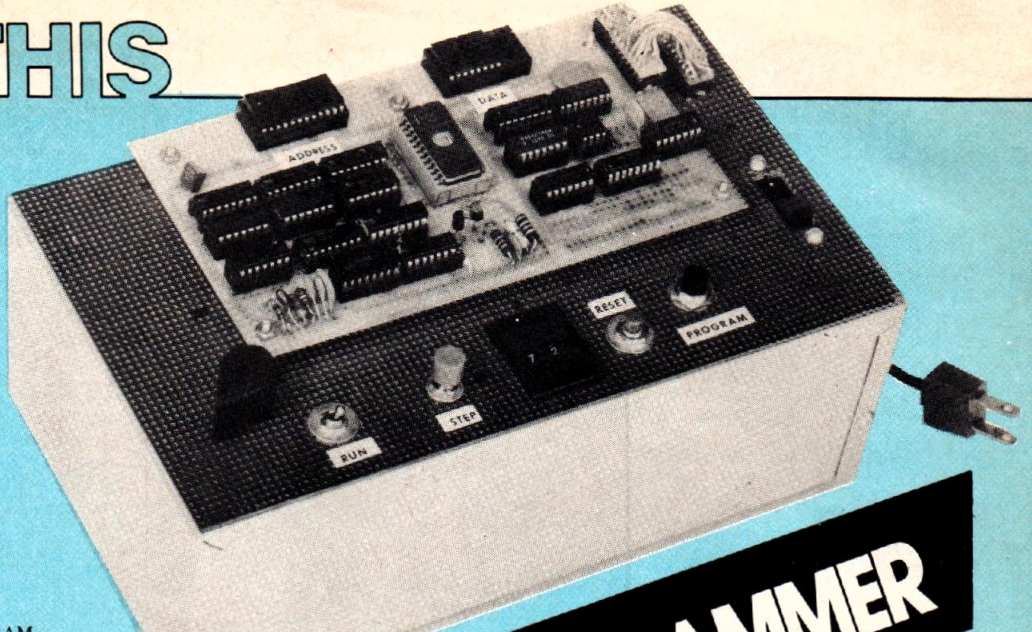
ON PROGRAMMING PROCEDURE

8223 PROM specifications sheet re-
nends a 390-ohm burn resistor, as op-
l to the 39-ohm value specified in
ow To Program Read-Only Memories" (July 1975). The exact magnitude difference between the two values makes me suspicious of the article's procedure. —*Mark Coffman, Cincinatti, Ohio*

I have a question on one of the steps for programming a PROM (page 30). The note in step 1 states: "NEVER operate *S1* when *S2* is set to BURN." If this is correct, there is no way to blow the fuses.—*David Peterson, Edmonton, Alberta, Canada*

It has been our experience that most PROM's need a great deal more burn current than is possible to obtain using a 390-ohm resistor. We experimented with a number of PROM's and found that 39 ohms is the optimum burn resistor value. The 390-ohm value failed to give the desired results. (Perhaps the PROM's we had, from several sources, were manufacturer fallouts; if so, most of the PROM's our readers will be able to obtain are in this category.)

As for the note in step one of the programming procedure, it should read: "NEVER operate S1 *when* S2 *is set to* TEST."



PROM PROGRAMMER

It is frequently convenient for the computer or microprocessor user to store his programs in reusable EPROM's (Erasable Programmable Read Only Memories). This programmer for 2K 2716 EPROMS allows the job to be done quickly and efficiently.

ROBERT N. BEABER

THE 2716 EPROM (*E*RASABLE *P*ROGRAMmable *R*ead *O*nly *M*emory)—currently available for ten dollars or less—is ideal for microprocessor-based projects. It permits storage of 2K (2048 bytes) of user program and operates from a single 5-volt supply. Because it can be erased by strong ultraviolet light, its contents can be changed if a modification must be made to the program it contains, or it can simply be erased for reuse in another project.

The programmer described here allows the project designer to program the 2716, using hexadecimal code, at each memory location by entering the required data with two thumbwheel switches and pressing a single PROGRAM switch. After programming a given memory location, the device automatically selects the next highest one. It also has a SINGLE STEP or CONTINUOUS RUN option to provide access to address locations at a rate determined by the user. Those options provide the control needed to verify the data contained by the EPROM. Both the memory location addressed and its contents are indicated by 7-segment LED's.

Circuit operation

Figure 1 is a schematic of the EPROM programmer. Refer to it as we discuss how the circuit works.

The programming sequence begins when the PROGRAM switch, S3, is pressed. That switch is debounced by IC11-c, and the resulting pulse is used to clock one-shot IC7-b. The output of the one-shot enables NAND gate IC8-c and allows the 40-Hz oscillator—IC9-b —to clock counter IC14.

Initially, the counter's output is set at 0. As it is clocked by IC9-b its outputs go to a logic-high (+5 volts) state one at a time, in sequence. *Note that only five of the ten possible outputs are used—pins 1, 4, 5, 7 and 11; they correspond to counts of 5, 2, 6, 3 and 9, respectively.* As the outputs of IC14 change state, they control the actions of latches IC13-a and IC13-b.

When pin 4 of IC14 goes high, pin 1 of IC13-a is latched high. That, through buffer IC2-c, brings pin 20, CS (*C*hip *S*elect—sometimes referred to as "data enable") of SO1, which contains the EPROM, high, and readies the EPROM for programming.

The output of IC14's pin 4 also turns on Q1 and Q2, which provide the programming voltage (25.5 volts) to pin 21 of SO1, and enables the three-state drivers, IC1 and IC2. The inputs of those two IC's are connected to S1 and S2, the BCD-output thumbwheel switches used to provide the programming data. The outputs of the two IC's are connected to pins 9-11 and 13-17 of SO1, which correspond to the eight data lines of the 2716 EPROM.

After 25 milliseconds—1/40-second, equal to one period of the 40-Hz clock —pin 7 of IC14 goes high and IC13-b is also latched high. That, in turn, takes pin 18 ($\overline{\text{CE}}$/PGM—$\overline{\text{CHIP ENABLE}}$/PROGRAM) of SO1 high and the data from the thumbwheel switches is programmed into the EPROM. The address programmed usually starts with 0, but can be selected by S4, (SINGLE STEP), and S5 (RUN).

When 50 milliseconds have passed, pin 1 of IC14 goes high. That resets IC13-b, removing the logic-high from pin 18 of the EPROM. After another 25 milliseconds, pin 5 of IC14 goes high, resetting IC13-a. That removes the CS voltage, removes the 25.5 volts from pin 21 by turning Q1 and Q2 off, and disables IC1 and IC2, thus removing the thumbwheel-switch data from the EPROM's data lines.

When pin 11 of IC14 goes high, it resets one-shot IC7-b, disabling NAND gate IC8-c and cutting IC14 off from the 40-Hz clock. Pin 11's going high also resets IC14 to 0 and advances counter IC3 to the next address to be programmed.

The remainder of the logic circuitry provides RESET, SINGLE STEP, and RUN functions.

Switch S6 (RESET) is used to reset the cascaded address counters IC3, IC4, and IC5, to zero. Switch S4 provides SINGLE STEP capability—every time the switch is pressed, IC7-a, a one-shot, supplies a pulse to counter IC3, causing the memory location being addressed

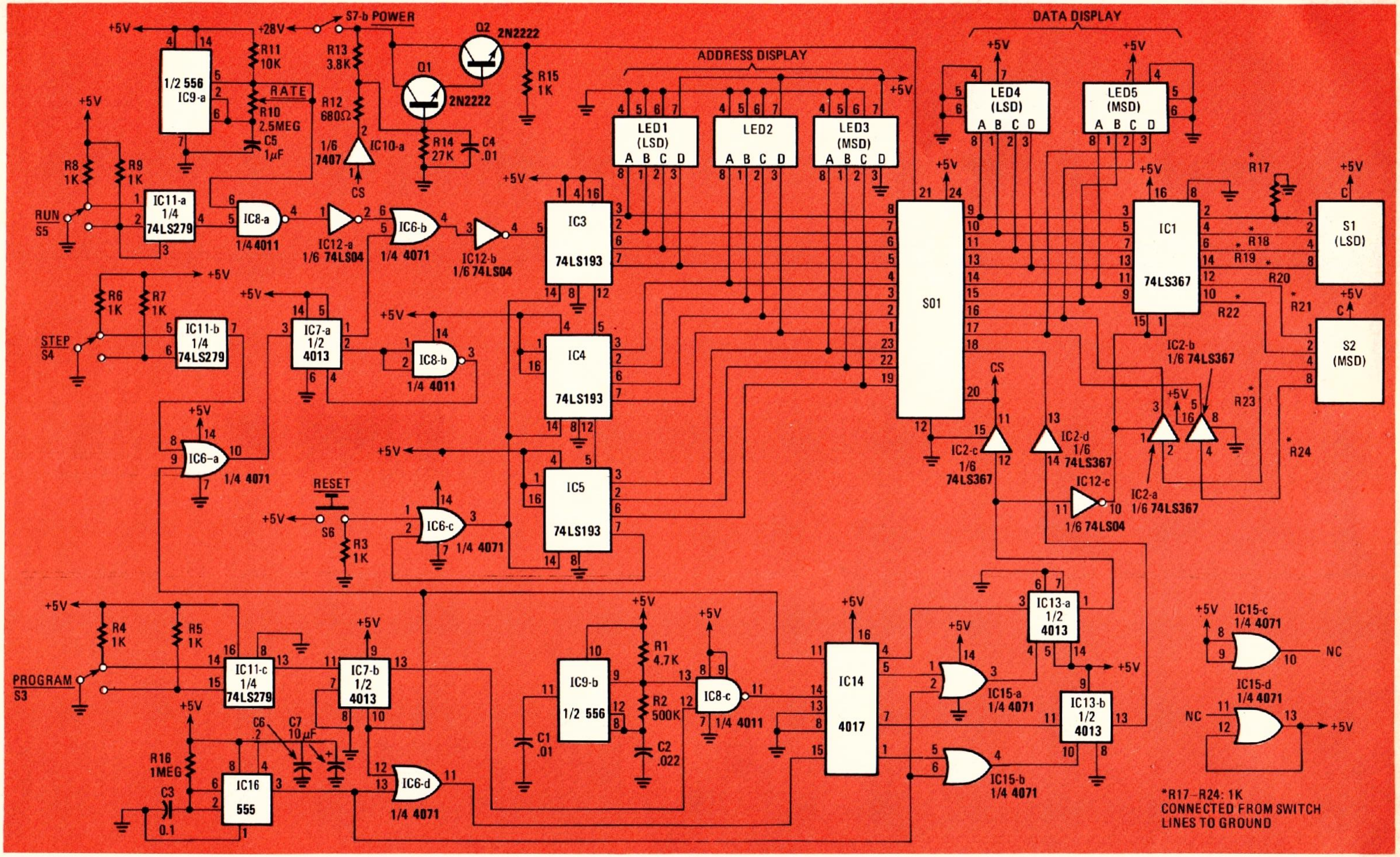


FIG. 1—WHILE ONLY R17 IS SHOWN, resistors R17-R24 are connected from the eight output-lines of S1 and S2 to ground.

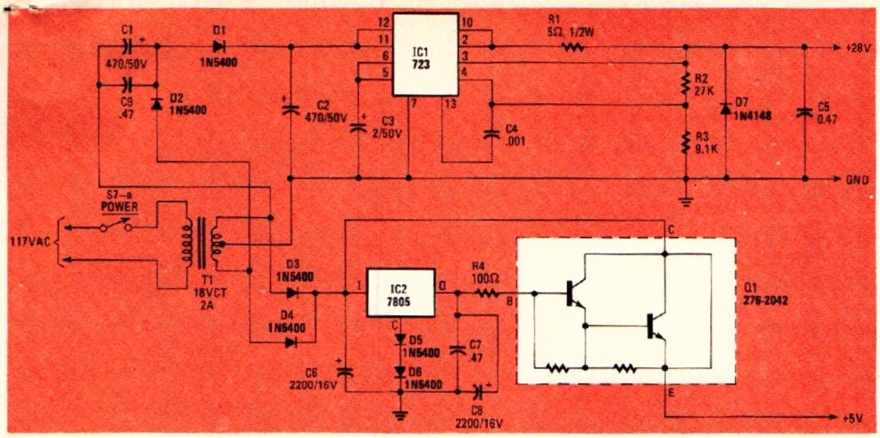


FIG. 2—POWER SUPPLY provides both the 5-volts needed for the logic circuitry and the 25.5-volts used to program the EPROM.

to be incremented by one. Finally, S5 (RUN) allows the variable-frequency oscillator formed by IC9-a and controlled by R10 to clock counter IC3 at a rate determined by the user.

The three switches provide the user with several means for accessing EPROM memory addresses to enter or verify data. The LED displays on the address and data lines provide a continuous indication of the memory address being accessed and of its contents.

Power supply

The power supply shown in Fig. 2 provides all the voltages required by the programmer. Power switch S7 has two sections—one for 117-volts AC (Fig. 2) and one for 28-volts DC (Fig. 1). That is necessary because when the AC voltage is cut off by S7-a, an induced current in the secondary of the transformer will cause a pulse to appear on the 28-volt DC line. Since that pulse can be transmitted to pin 21 (V_{PP}) of the EPROM—a pin used in the programming process—erroneous data may be entered into the IC. Therefore, S7-b is used to cut off the 28-volt supply at the same time the main 117-volt supply is turned off.

Construction

The 2716 EPROM programmer can be constructed on perforated construction

PARTS LIST—POWER SUPPLY

All resistors 5%, ¼-watt unless otherwise specified
R1—5 ohms, ½-watt
R2—27,000 ohms
R3—9100 ohms
R4—100 ohms
Capacitors
C1, C2—470 µF, 50 volts, electrolytic
C3—2 µF, 50 volts, electrolytic
C4—.001 µF, ceramic disc
C5—0.47 µF, ceramic disc
C6, C8—2200 µF, 16 volts, electrolytic
C7—0.1 µF, ceramic disc
Semiconductors
IC1—LM723N adjustable voltage regulator
IC2—7805 5-volt regulator
Q1—high-power Darlington, NPN-type (Radio Shack 276-2042 or similar)
D1-D6—1N5400, 3 amps, 50 PIV
T1—18 volts, center-tapped, 2 amps
Miscellaneous: TO-3 heat sink, PC board or perforated construction board, 14-pin IC socket, hardware, etc.

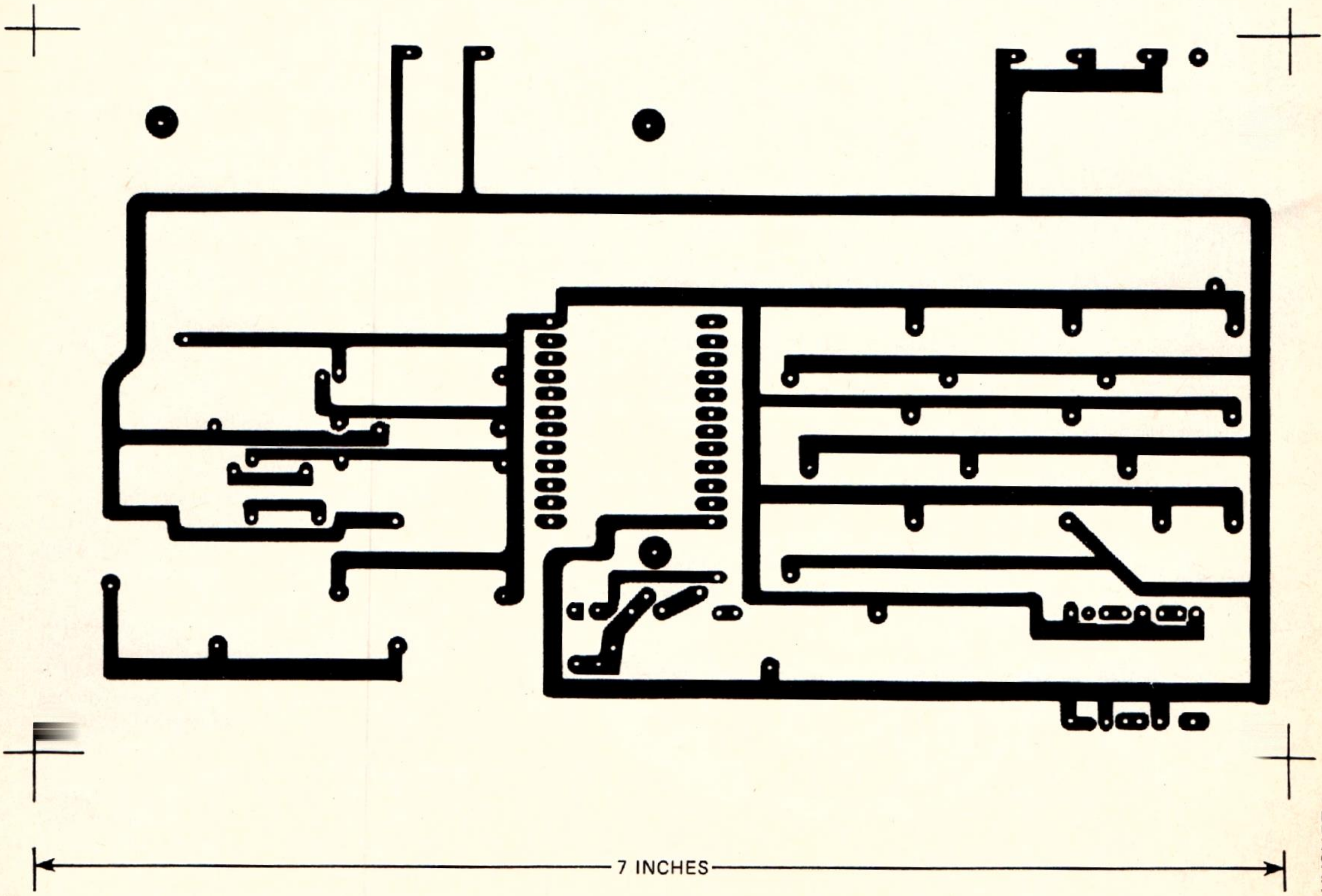


FIG. 3—PERFORATED CONSTRUCTION BOARD is used to mount components. Stick-on copper traces can be applied to serve as power and ground buses.

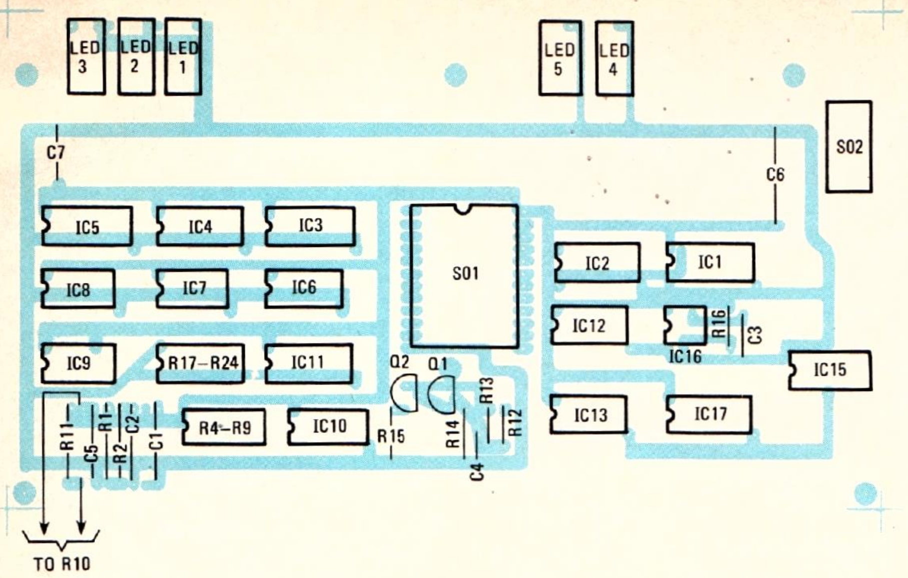


FIG. 4—SOCKET SO1 is a 24-pin zero-insertion-force socket used to hold the 2716 EPROM. Zero-insertion-force sockets allow IC's to be inserted and removed many times without causing wear or damage to either the socket or the IC.

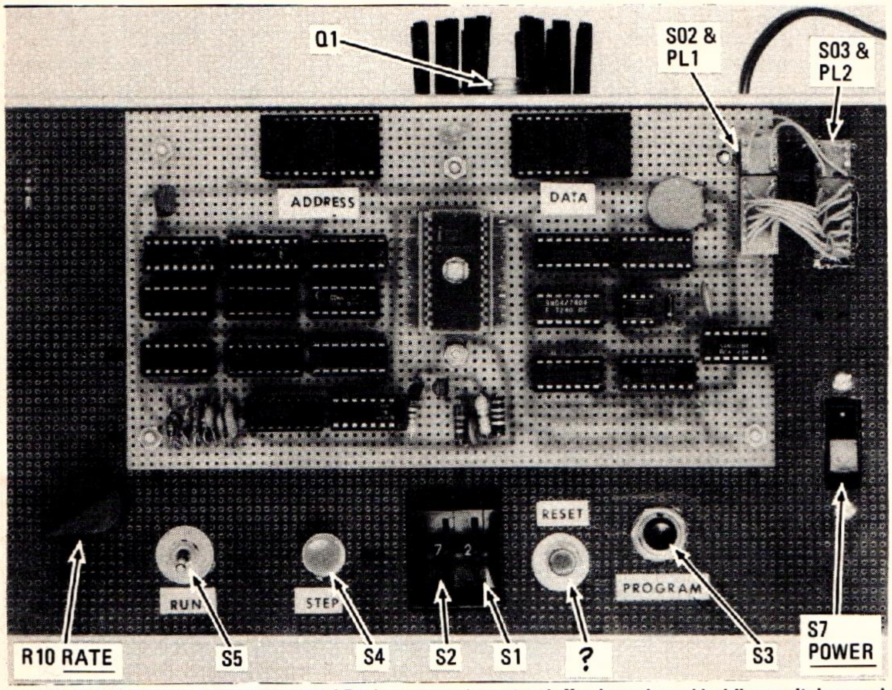


FIG. 5—LOGIC/PROGRAMMING BOARD is mounted on standoffs above board holding switches and controls. Heat sink and Q1 are part of power supply.

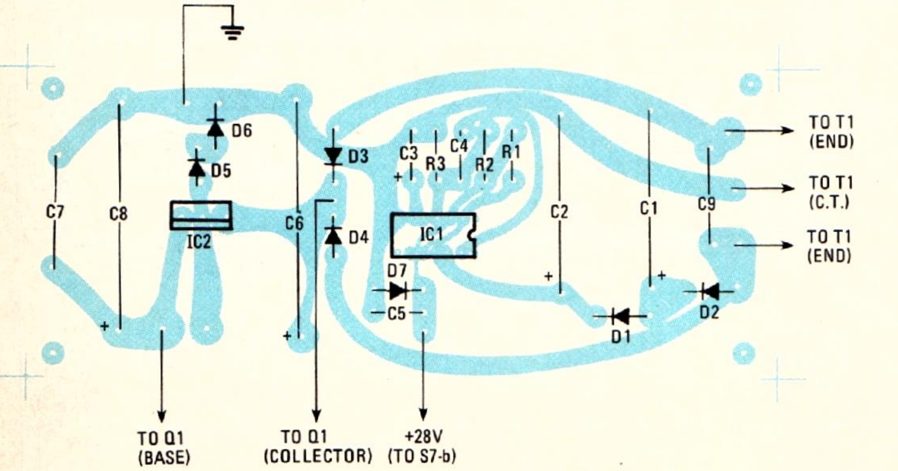


FIG. 6—PADS MARKED "TO T1 (END)" may be connected to either end of transformer's primary winding. Pad marked "C.T." must go to center tap.

PARTS LIST—PROGRAMMER

All resistors 5%, ¼-watt unless otherwise specified
R1—4700 ohms
R2—500,000 ohms
R3-R9, R15, R17-R24—1000 ohms
R10—2.5 megohms, ½-watt, potentiometer
R11—10,000 ohms
R12—680 ohms
R13—3800 ohms
R14—27,000 ohms
R16—1 megohm
Capacitors
C1, C4—.01 µF, ceramic disc
C2—.022 µF, ceramic disc
C3—0.1 µF, ceramic disc
C5—1.0 µF, tantalum
Semiconductors
IC1, IC2—74LS367 or 80C97 three-state hex buffer
IC3-IC5—74LS193 or 74C193 up/down binary counter
IC6, IC15—4071 CMOS quad 2-input OR gate
IC7, IC13—4013 CMOS dual-D flip-flop
IC8—4011 CMOS quad 2-input NAND gate
IC9—556 dual timer
IC10—7407 hex buffer, high-voltage, open-collector
IC11—74LS279 quad S-R latch
IC12—74LS04 hex inverter
IC14—4017 CMOS decade divider/counter
IC16—555 timer
Q1, Q2—2N2222 or similar NPN-type
LED1-LED5—7 segment hexadecimal decoder/display (H-P 5082-7340 or similar)
S1, S2—hexadecimal thumbwheel switch, BCD output (Unimax SF-54 or similar)
S3, S4—SPDT momentary pushbutton switch
S5—SPDT toggle switch
S6—N.O. pushbutton switch
S7—DPDT toggle or rocker switch, 125 VAC minimum rating
SO1—24-pin zero-insertion-force IC socket
SO2, SO3—18-pin wire-wrap IC socket
PL1, PL2—18-pin DIP header
Miscellaneous: wire-wrap IC sockets, perforated construction board, chassis-box, wire, etc.

board using wire-wrap techniques. A foil pattern for the power buses and several of the IC-to-IC connections is shown in Fig. 3. While parts-placement is not critical, a suggested layout is shown in Figs. 4 and 5.

Note that the logic/programming circuitry is located on one board, and that board is mounted on spacers over a second board that holds the switches and RATE control. Connections between the two boards are made using SO2, SO3 and PL1 and PL2.

A foil pattern for the power supply is shown in Fig. 7, and a parts-placement diagram for it in Fig. 6. The power supply can be mounted on spacers inside an aluminum chassis-box (Fig. 8). A heat sink should be used for the power supply's Q1.

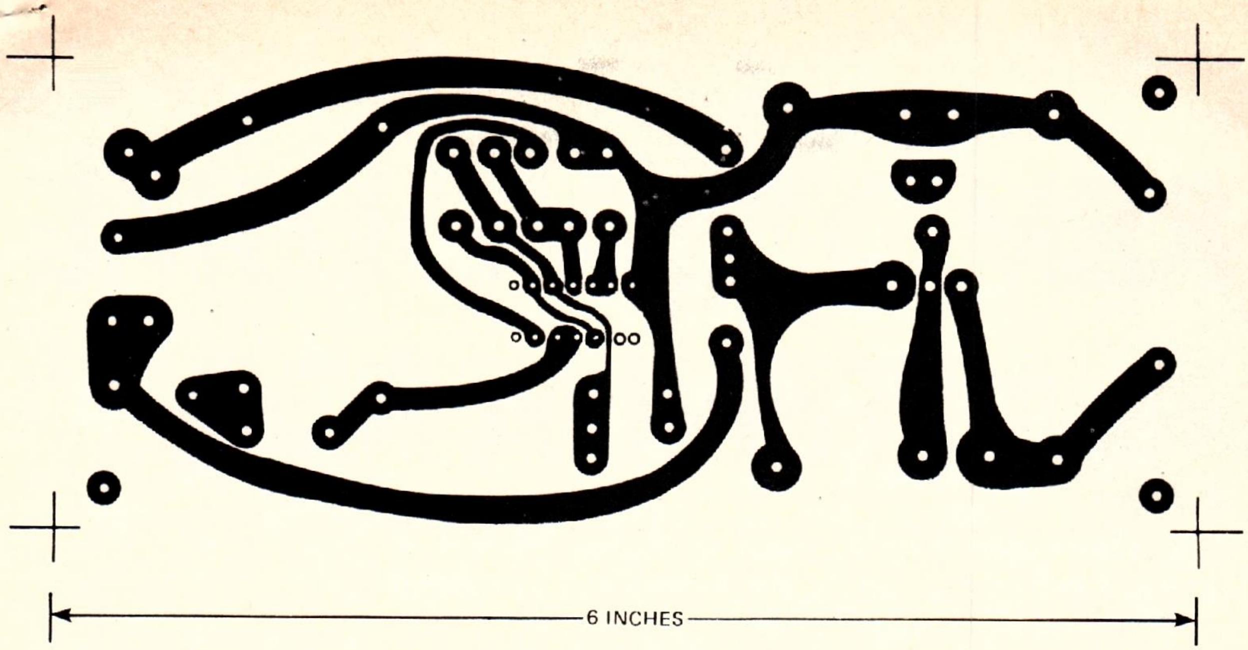


FIG. 7—FOIL PATTERN FOR POWER SUPPLY allows quick and easy construction. Ordinary copper-clad board can be used here.

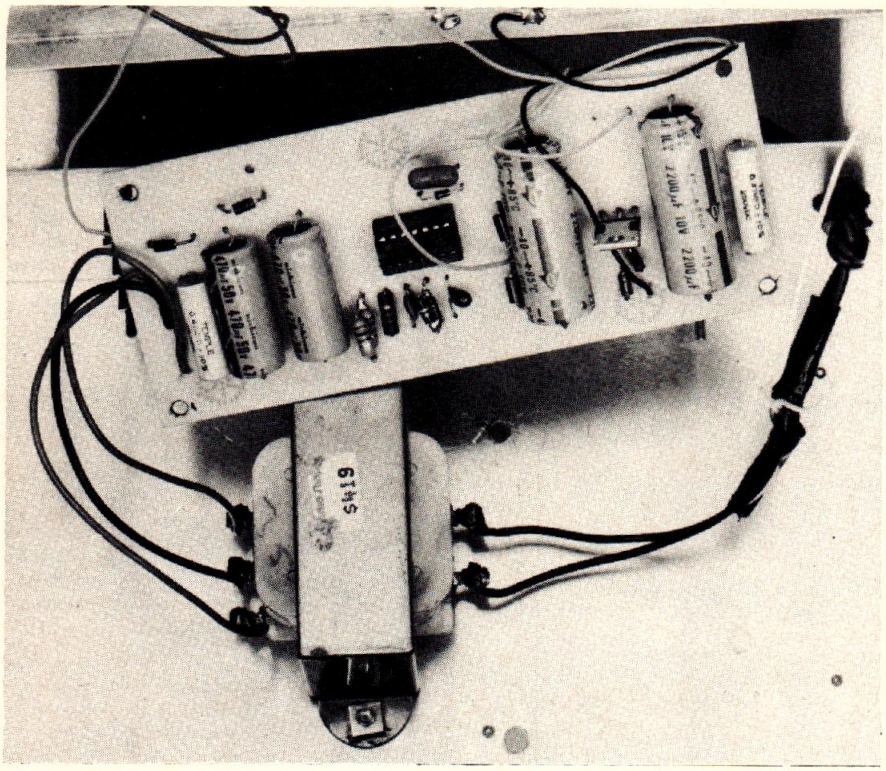


FIG. 8—TRANSFORMER IS MOUNTED on chassis-box bottom; power-supply board on standoffs at enclosure's rear.

Checkout and use

Before any of the logic IC's are installed in their sockets, the usual visual checks for correct component-orientation, solder bridges, etc. should be made. The unit can then be powered up and voltage checks made at the appropriate points. If everything checks out at that point, the IC's can be installed.

Now, with no EPROM in SO1, apply power again. The DATA LED's should show "FF," and the ADDRESS LED's should indicate "ØØØ" after the RESET has been pressed. Each time the SINGLE STEP switch is pressed, the ADDRESS display should advance one count. With the RUN switch in the ON position, the ADDRESS display should advance continuously, at a speed that can be varied by R10, the RATE control.

Still with no EPROM in the socket, set up a data byte on the thumbwheel switches. When you depress the PROGRAM switch, the DATA LED's should show that value briefly, and then return to "FF." If all those conditions are met, the programmer is ready to go to work.

With the power turned off, insert a 2716 EPROM into SO1. Apply power and depress the RESET switch to start programming at address ØØØ. Use the thumbwheel switches to set the data byte that is to be programmed into that location and press the PROGRAM switch. The DATA LED's should momentarily show this value, and then return to "FF"—the contents of the next memory location (ØØ1, shown on the ADDRESS LED's). The data for that address can then be set on the thumbwheel switches and the process repeated until the EPROM is completely programmed.

The EPROM's contents can be verified by resetting the counter to ØØØ and using the RUN switch and RATE control to automatically sequence through the memory locations. They, and their contents, will be shown on the LED's. Alternately, you can single-step through the addresses at leisure by using the SINGLE STEP switch. **R-E**

"I thought that voice synthesizer would tell me my weight—not scream in pain."

LETTERS

Address your comments to: Letters, **Radio-Electronics,** 200 Park Avenue South, New York, NY 10003

THE PROM PROGRAMMER

I read Robert N. Beaber's article on how to build a 2716 Prom Programmer in the February 1982 **Radio-Electronics,** and found his design very simple and straightforward. I'm planning to build an EPROM programmer, using his designs with three small changes.

By using an SPDT switch, a 74LS244 buffer, and a 24-pin ZIF socket, I will have the capability of copying another 2716 EPROM in whole or in part.

The advantage of that is clear: Many times, an EPROM has one or two errors that could be fixed just by changing the byte at the location of the error(s). Instead of recopying the whole EPROM by hand, the majority is recopied by single-stepping then hitting the program button. The necessity of having another 2716 available is not a disadvantage, as they cost less than five dollars.

The disadvantage is that you can't copy a program to a different address. That would require the addition of three more 74LS193's, three more address-display LED's, one more 4071, and one more reset switch. The same clock from pin 4 of IC12 is used on each address counter. That circuit is more complex than I need, but someone else might like the added option.

The power supply should not be a concern with either one of the design additions. The design-change I plan to use is shown in Fig. 1.

JEFF McDONALD
Tucson, AZ

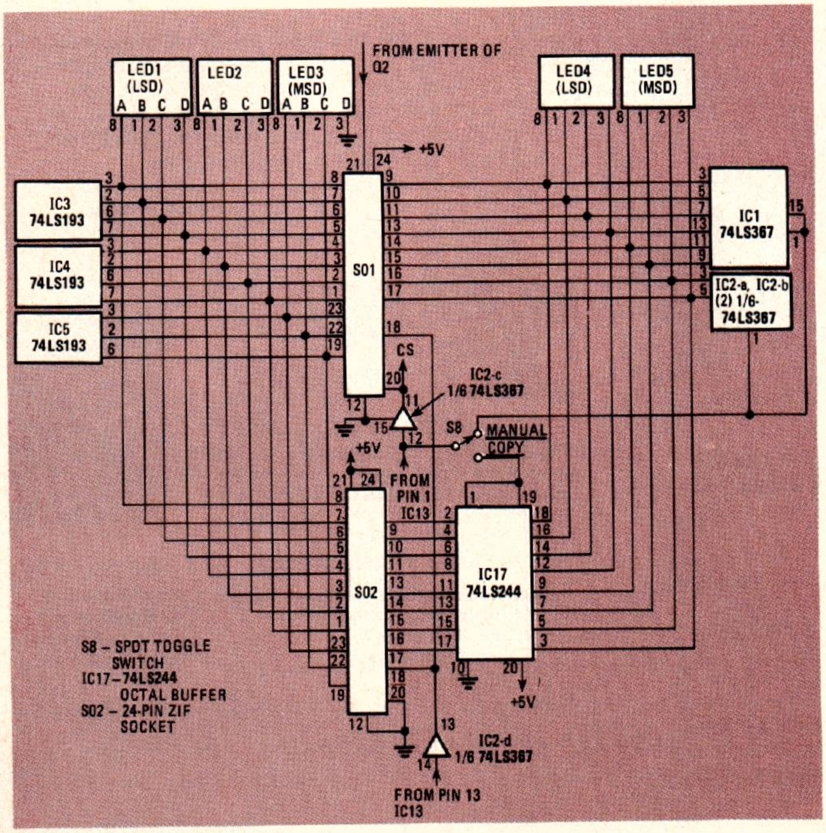


FIG. 1

RADAR DETECTORS

Mr. M. J. Rybicki's letter in the February 1982 issue of **Radio-Electronics** strikes me as being particularly illogical. He says: "I don't believe for one moment that the policeman operating the radar unit is primarily interested in saving my life ..." If it were only the lives of the drivers involved that were at stake, there would be some logic in letting them drive just as fast as they wanted to. Then, if they killed themselves, one could say, "Good riddance!" Unfortunately, all too often they kill someone else and not always themselves as well.

He also says: "To say that every radar-detector owner has it solely for the purpose of speeding without getting caught is like saying that anyone who owns a gun is planning a bank job or an assasination." Guns can be used for hunting, target-shooting, or self-defense—but I have yet to hear of a radar-detector in a car being used for *any* other purpose except to avoid getting caught when speeding. Perhaps Mr. Rybicki can tell us what he uses his for.

RICHARD KOLASINSKI
Richmond, MI

OOOOOPS

On page 59 of the March 1982 **Radio-Electronics,** there is a formula:

$Nf\ (g) = 10 \log_{10} N$

Where:

g = gain in dB
N = noise factor
Nf = noise figure.

The next formula:

$N = \log^{-1} \frac{Nf\ (g)}{10}$ is incorrect!

Solving equation (1) for (N) gives:

$N = \log^{-1} \left(\frac{Nf\ (g)}{10} \right)$ which is the correct formula for the noise factor.

In the same way, power gain (G) should be:

$G = \log^{-1} \left(\frac{g}{10} \right)$ *not* $G = \frac{\log^{-1} (g)}{10}$

These are the correct relationships.

BILL HOSTMANN

INSUFFICIENCY ALERT

Before building the ambient illumination insufficiency alert ("Six Unique Projects for Your Car," April 1982 **Radio-Electronics**), interested readers should consider the spectral response of PC1 in Figure 10, page 48. That is so that the alert is not activated falsely by a failure to sense the illumination from the various types of lamps that may be present. The "colors" of the light from conventional tungsten-filament lamps, those of the halogen type (quartz-iodine), and those of the filtered amber fog lamp should be compared with the spectral response curve of the photocell.

A fringe benefit of your circuit became apparent to me, and I would like to mention it here so as to give the readers an additional

PROGRAMMING EPROM's WITH A SMALL COMPUTER

BY JOHN DOOLITTLE
AND SLOBODAN TKALCEVIC

Part 2

IN Part 1 of this article, we discussed how EPROMs can be put to good use in expanding your digital design work. We then presented a circuit that can be used as an interface with a small computer such as the TRS-80 to program EPROMs. We are ready to proceed now with constructing the circuit.

Construction. The EPROM Programmer is designed so that all components except for the ac power supply are mounted on a double-sided printed circuit board such as that shown in Fig. 7. The four lines from the transformer secondary (color-coded blue, yellow, red, and green) are soldered to the corresponding points on the printed circuit board marked B, Y, R, and G. Connection to the computer is done through a 40-pin card-edge connector that is compatible with the expansion bus of the TRS-80 CPU or Expansion Interface unit. All components are mounted on the bottom side of the board—except the zero insertion force EPROM socket, the personality module socket, and the indicator LEDs. These parts mount on top of the board. The printed circuit can be neatly mounted in a 7″ by 6″ by 3″ aluminum enclosure using ¼-inch spacers and allowing the topside components to protrude through the top panel. Access to the board's edge connector is made through a notch in the rear panel.

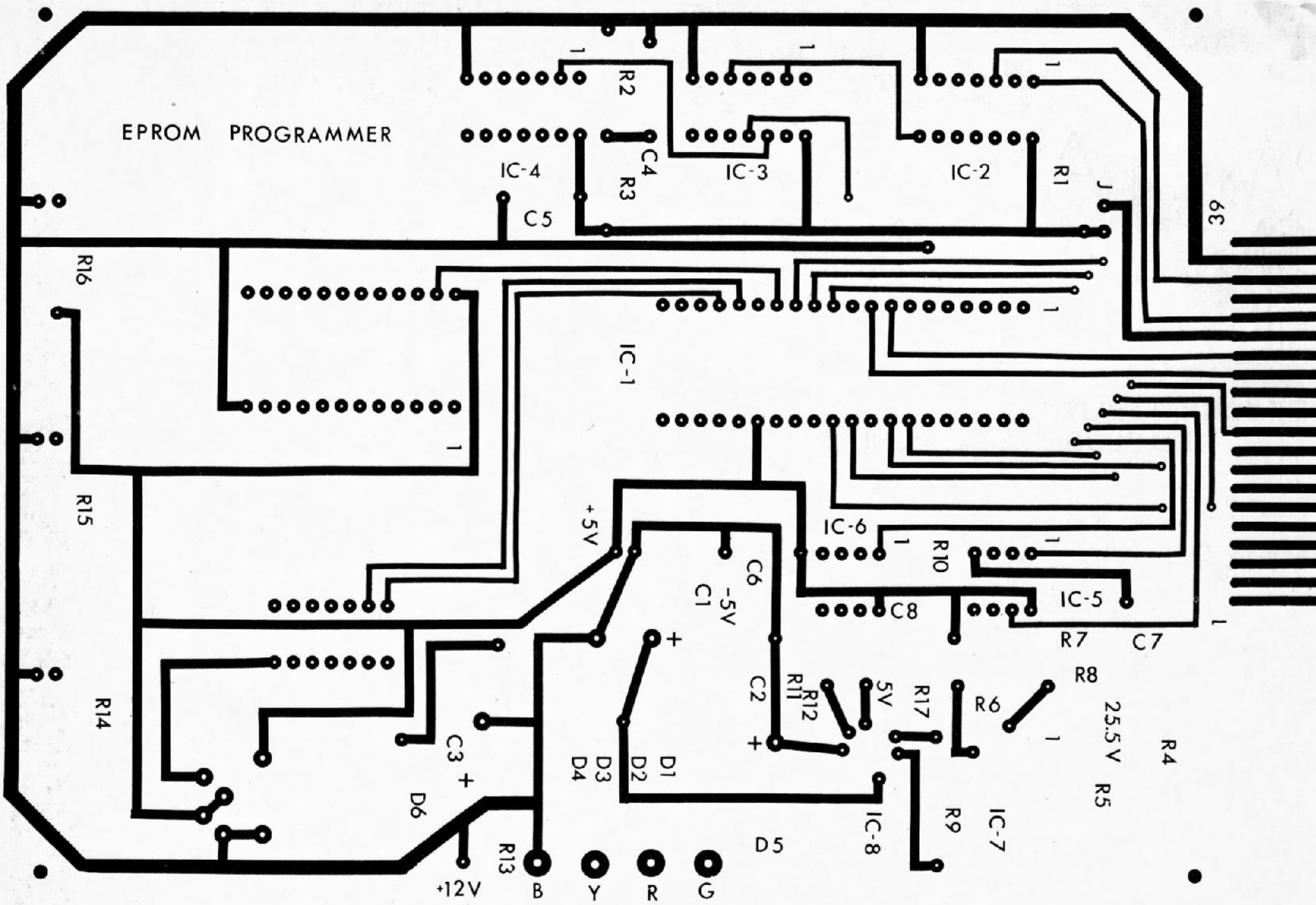
With all components soldered in place, switchable voltage regulators *IC7* and *IC8* should be adjusted (*R8* and *R12*) to develop 25.5 V and 5.0 V, respectively, before connecting the EPROM Programmer to the computer. The personality module socket is a convenient test point for the constant −5-V (pin 1) and 12-V (pin 4) supplies as well as for the switchable 5-V (pin 2) and 25.5-V (pin 3) supplies. The constant 5-V supply can be measured at pin 24 of the EPROM socket.

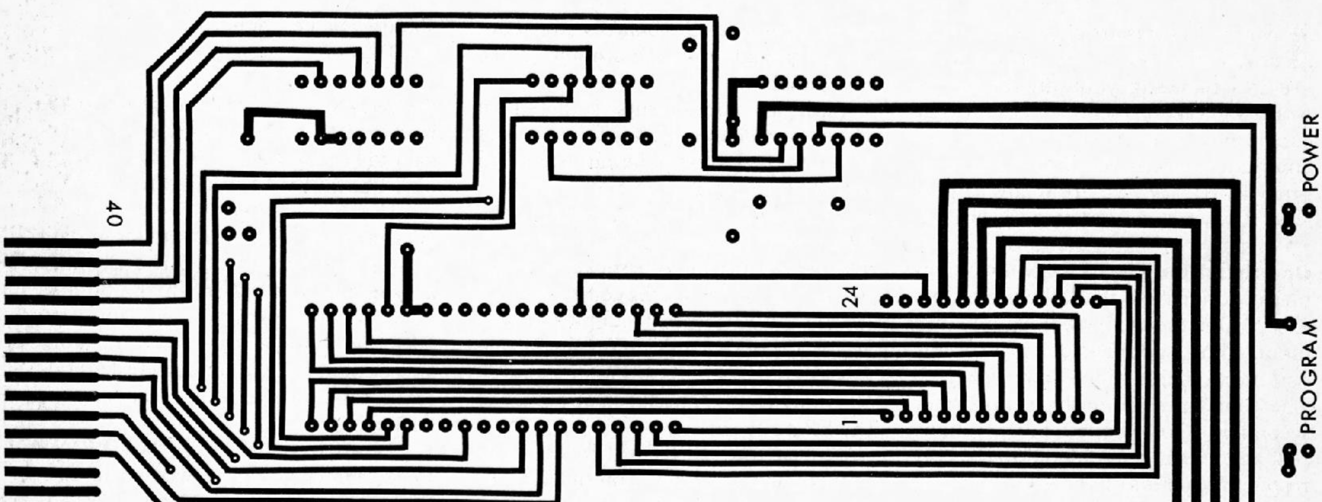
To use the EPROM Programmer with computers other than the TRS-80 Model I, a simple adapter card can be designed to interconnect the appropriate data, address, and control lines. Since ribbon cables terminated by identical connectors have the property of interchanging the lines top to bottom (but not right to left), care must be used to see that the routing is correct. In some applications, additional logic may be necessary to meet the I/O requirements of the host computer. If so, 5 V can be made available at the card edge to power external logic by simply inserting a jumper at point "J" on the printed circuit. Of course, this jumper should *not* be connected for use with the TRS-80.

Software. To accommodate up to 4K by 8-bit EPROMs, two 4K-byte buffers should be reserved in the TRS-80 memory. The first is a listing buffer which is used by the List, Verify, Copy, and Erasure verification routines to store data read in from an EPROM. The second is a programming buffer which contains data to be programmed into an EPROM. While the same memory space could be used for each task at different times, it is easiest to use separate buffers.

The software driver for the EPROM Programmer could be written in BASIC where the port-addressed I/O is handled by INP and OUT statements. Using a BASIC program for such a large task would require a large amount of memory and would operate slowly because of interpreter overhead. A better approach is to write the driver in assembly language. (The software driver written by the authors easily fits into 4K bytes of memory space.) See the Parts List in Part 1 of this article for information on ordering the software.

Sequential EPROM addresses are generated in software simply by incrementing a 12-bit counter and output-


EPROM PROGRAMMER
IC-1
IC-2
IC-3
IC-4
IC-5
IC-6
IC-7
IC-8
+5V
-5V
+12V
25.5V
B
Y
R
G
D5
R7
C7
R8


POWER
PROGRAM
READ
40
2
24

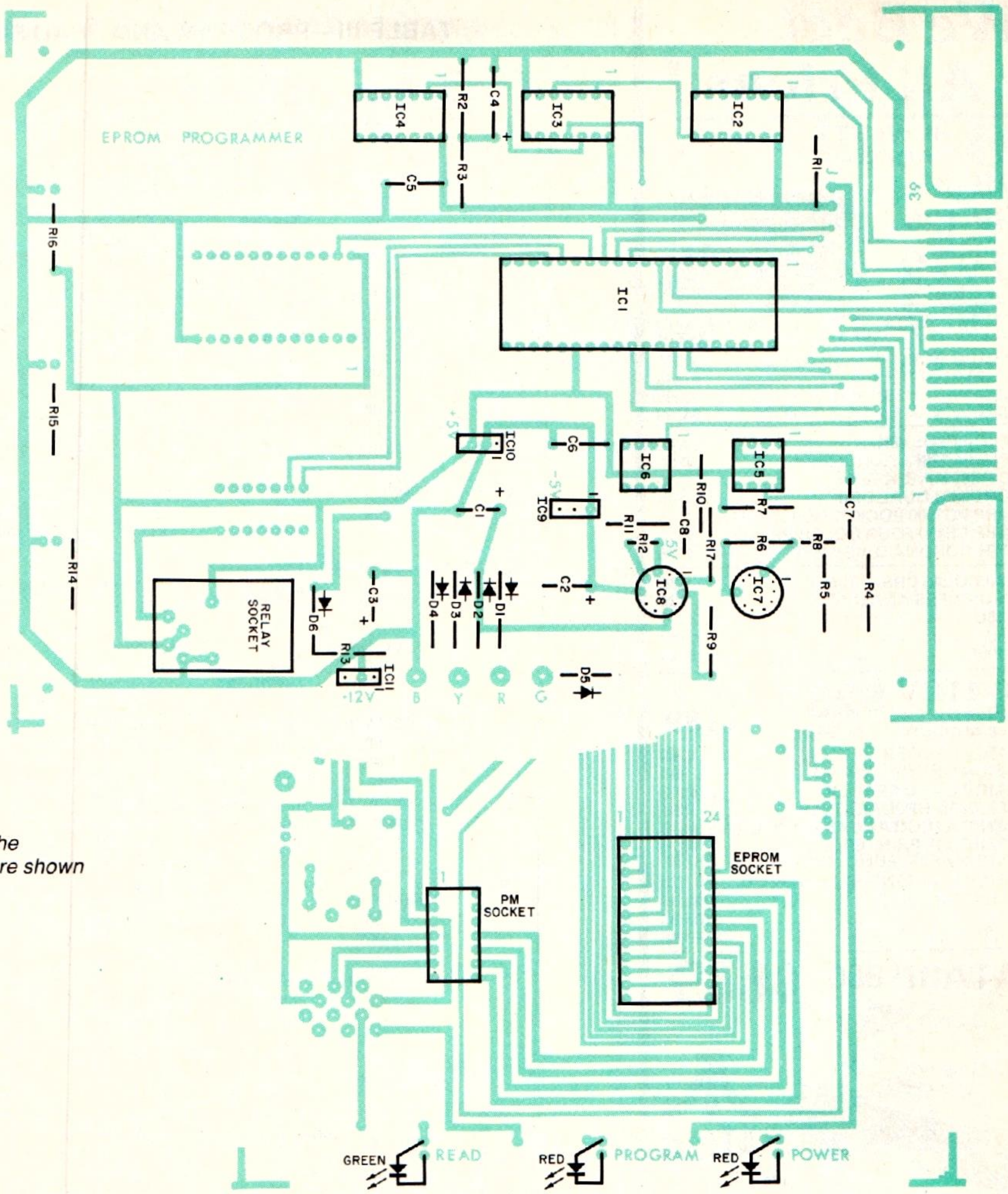


Fig. 7. Foil patterns for the double-sided pc board are shown on opposite page, while component layout is above and at right.

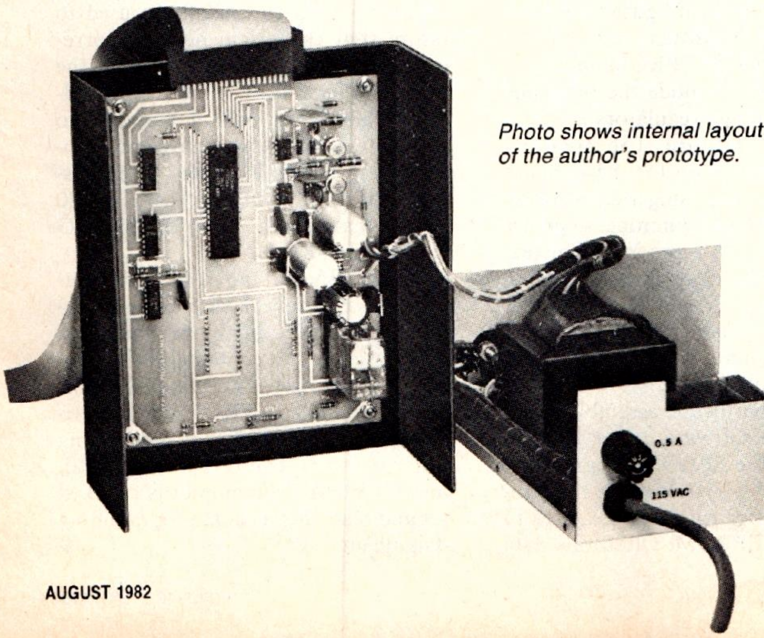


Photo shows internal layout of the author's prototype.

ting the low-order eight-bits to the 8255A's port A while the remaining high-order four-bits determine whether the port C bits (PC0-PC3) are set or reset. Address pointers to the corresponding buffer locations in the computer memory can be determined by adding the buffer starting address to the counter value. Table II shows the appropriate control bytes which must be sent to the 8255A's control port to set or reset the various lines of port C. Also shown in Table II are the control bytes which configure the 8255A for Read or Program modes.

Table III shows how to program and read data for a single location of a 2716 EPROM using Z80 assembly language. (The CPU I/O port addresses and control bytes used are those listed in Tables

TABLE III—PROGRAM AND READ SEQUENCE

```
; PROGRAMMING SEQUENCE
; Assuming that EPROM address 537H contains FF program it
; to 0E9H
        LD      A,82H           ;control byte for PROGRAM mode
        OUT     (0FBH),A        ;output it to 8255 CONTROL port
;  Set EPROM address 537H:
        LD      A,37H           ;output 8 LSBs of EPROM address
        OUT     (0F8H),A        ;to 8255 Port A
; Set 4 MSBs of address by setting proper 8255 I/O lines
;(PC0-PC3)
; In this example 4 MSBs = 0101 (5 from EPROM address)
        LD      A,01H           ;set EPROM address A8
        OUT     (0FBH),A        ;by setting 8255 I/O line PC0
        LD      A,02H           ;reset EPROM address A9
        OUT     (0FBH),A        ;by reseting 8255 I/O line PC1
        LD      A,05H           ;set EPROM address A10
        OUT     (0FBH),A        ;by setting 8255 I/O line PC2
        LD      A,06H           ;reset EPROM address A11
        OUT     (0FBH),A        ;by reseting 8255 I/O line PC3
;
; Switch relay into PROGRAM mode
        LD      A,0CH           ;reset 8255 I/O line PC6 to
        OUT     (0FBH),A        ;switch relay to PROGRAM mode
;
; Enable EPROM programming
        LD      A,09H           ;set 8255 I/O line PC4 to
        OUT     (0FBH),A        ;turn on switchable 25.5V
;
; Output byte (e.g. E9H) of programming data to EPROM
        LD      A,0E9H          ;data output to 8255 Port B
        OUT     (0F9H),A
;
; Apply 5V programming pulse of 50 msec duration
        LD      A,0BH           ;set 8255 I/O line PC5 to
        OUT     (0FBH),A        ;turn on switchable 5V
;
; Enter 50 msec delay loop (not shown here) and turn off
; programming pulse after loop falls thru
        LD      A,0AH           ;reset 8255 I/O line PC5 to
        OUT     (0FBH),A        ;turn off switchable 5V
;
; Disable EPROM programming mode
        LD      A,08H           ;reset 8255 I/O line PC4 to
        OUT     (0FBH),A        ;turn off switchable 25.5V
;
; Switch relay to READ mode
        LD      A,0DH           ;set 8255 I/O line PC6 to
        OUT     (0FBH),A        ;return relay to READ mode
;
; READ SEQUENCE
;  Set up 8255 for READ operation
        LD      A,80H           ;control byte for READ mode
        OUT     (0FBH),A        ;output it to 8255 CONTROL port
; Set EPROM address as shown above
;
; Read data from that EPROM address into register A:
        IN      A,(0F9H)        ;read data from 8255 Port B
```

I and II.) This example shows exactly how the 8255A is configured for Read and Program operations and how to specify a particular EPROM address.

In the Program mode the relay and switchable voltage regulators are activated by the software. The programming pulse is held on for the required programming time using a software delay loop. The programming sequence for other EPROM types is similar (except for the delay loop period). And in some cases, the 5-V and 25.5-V switchable supplies exchange roles as sources for the programming enable signal and the programming pulse.

With the hardware assembly complete and the software driver written, check-out of the EPROM Programmer can proceed by first loading the programming buffer with data destined to reside in the EPROM. Often this data will be a binary file generated by an assembler, where care has been used to assure that the origin address corresponds to the starting address of the programming buffer. A more direct, although laborious, loading method is to type data from the keyboard into programming buffer locations using system software such as the TRS-80 TBUG (cassette systems) or DEBUG (disk systems).

Next, verify that the EPROM is completely erased (check that every bit location is initially in the 1 state), and then program the contents of the programming buffer into the EPROM. The programmed EPROM can be verified against the original programming buffer contents. If no errors are found, your EPROM Programmer is checked out and ready for its next programming assignment. ◇

LOW-COST EPROM PROGRAMMER

BY DAN VINCENT

ART 2 *Power supply, construction and checkout.*

Power Supply. The supply (Fig. 5) delivers approximately +75 volts to a transistor switch/current limiter consisting of *Q1*, *Q2*, *Q3*, *R1*, *R2* and *R3*. Transistors *Q4* and *Q5*, in conjunction with *D5*, *R6*, *R7*, and *R8* regulate the +75-volt output down to +47 volts. Diode *D6* and resistor *R5* provide the V_{BB} bias supply. Resistor *R9* insures a minimum load on the regulator and provides a path for the *D6* zener current. Capacitor *C2* and resistor *R20* prevent the high-gain circuit of *Q5* from oscillating.

Construction. Although the Program-

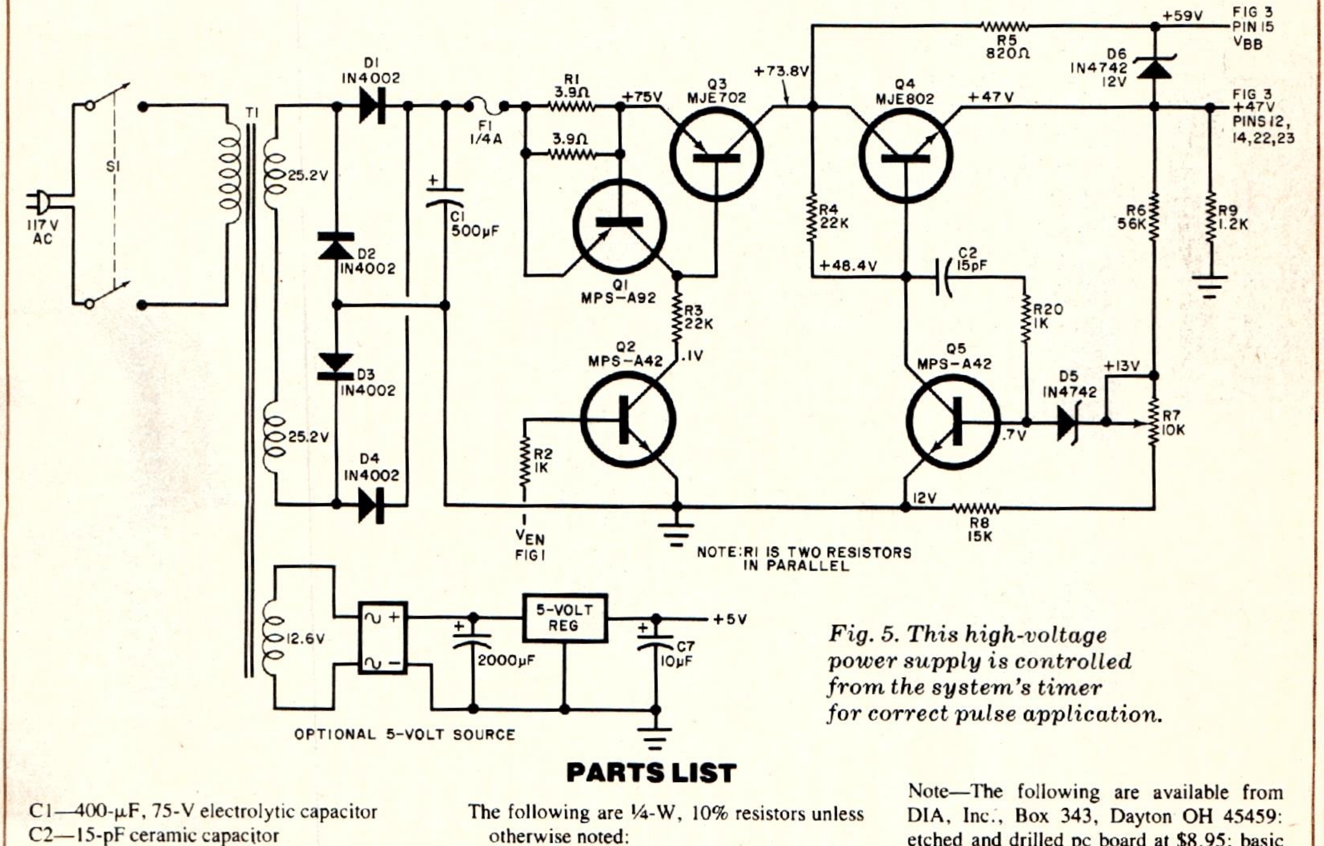


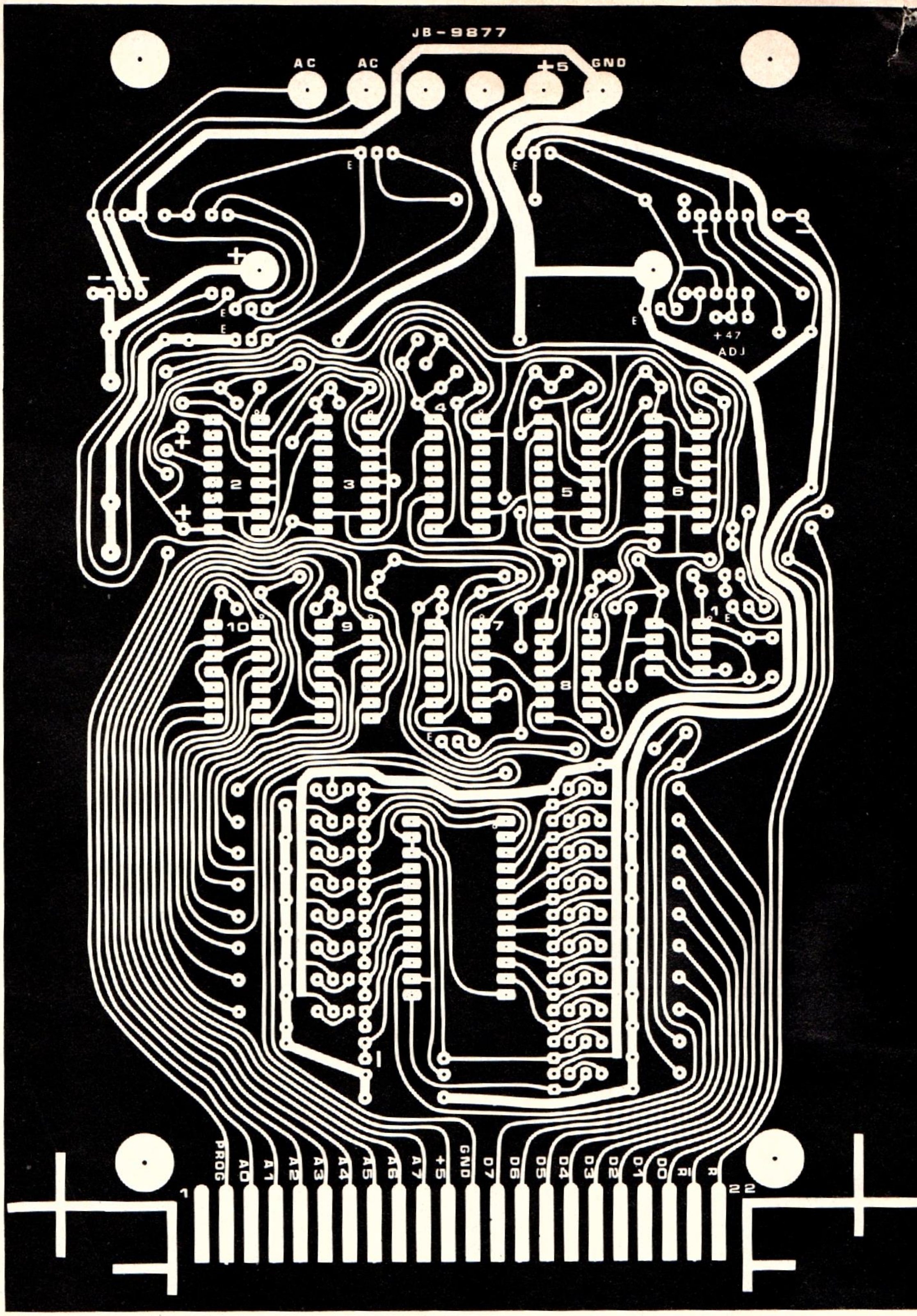
Fig. 5. This high-voltage power supply is controlled from the system's timer for correct pulse application.

PARTS LIST

C1—400-μF, 75-V electrolytic capacitor
C2—15-pF ceramic capacitor
C7—10-μF, 15-V electrolytic capacitor
D1 through D4—1-A, 200-V silicon diode (1N4002 or similar)
D5, D6—12-V, 1-W, 10% zener diode (IN4742 or similar)
F1—¼-A fast-blow fuse and holder
Q1—MPS-A92, 300-V silicon pnp transistor
Q2,Q5—MPSA42, 300-V silicon npn transistor
Q3—MJE702, 80-V silicon Darlington pnp transistor
Q4—MJE802, 80-V silicon Darlington npn transistor

The following are ¼-W, 10% resistors unless otherwise noted:
R1—1.9-ohm, 5%, 1-W resistor (two 3.9-ohm, ½-W, 5% in parallel)
R2, R20—1000 ohms
R3,R4—22,000 ohms
R5—820 ohms, ½-W
R6—56,000 ohms
R7—10,000-ohm, 10%, ½-W trimmer potentiometer (Bourns 3389 series)
R8—15,000 ohms
R9—1200 ohms, 2 W
S1—Dpdt, 1-A toggle switch
T1—2 each 25.2-V, 300-mA transformer with 5-V power-supply winding

Note—The following are available from DIA, Inc., Box 343, Dayton OH 45459: etched and drilled pc board at $8.95; basic 1702A EPROM Programmer kit (TTL option), including pc board and all components except power supply and connector, at $39.95; special transformer that also includes 5-volt power supply winding at $9.98; complete kit, including pc board and all components for stand-alone Programmer with switches, power supply, zero-insertion-force socket and case, at $79.95; complete stand-alone Programmer, assembled and tested, at $99.95. All prices postpaid. Check, money order, Visa, or Master Charge accepted. Ohio residents, please add 4½% sales tax.



mer can be built using any desired construction technique, a printed circuit board such as that shown in Fig. 6 is suggested. Observe the correct polarities when installing capacitors, diodes, transistors and IC's (using sockets, if desired). Do not install transistors *Q8* and *Q14* through *Q29* until after reading the checkout section of this article. Mount 1-inch by ½-inch thin metal heat sinks on transistors *Q3* and *Q4* . Using the fuse as a guide, install a fuseholder or fuse clips at the *F1* position. Do not install a socket at position *S02* or the LED for *LED1* if you are going to mount the board in an enclosure.

The component installation shown in Fig. 6 uses the TTL option so that the

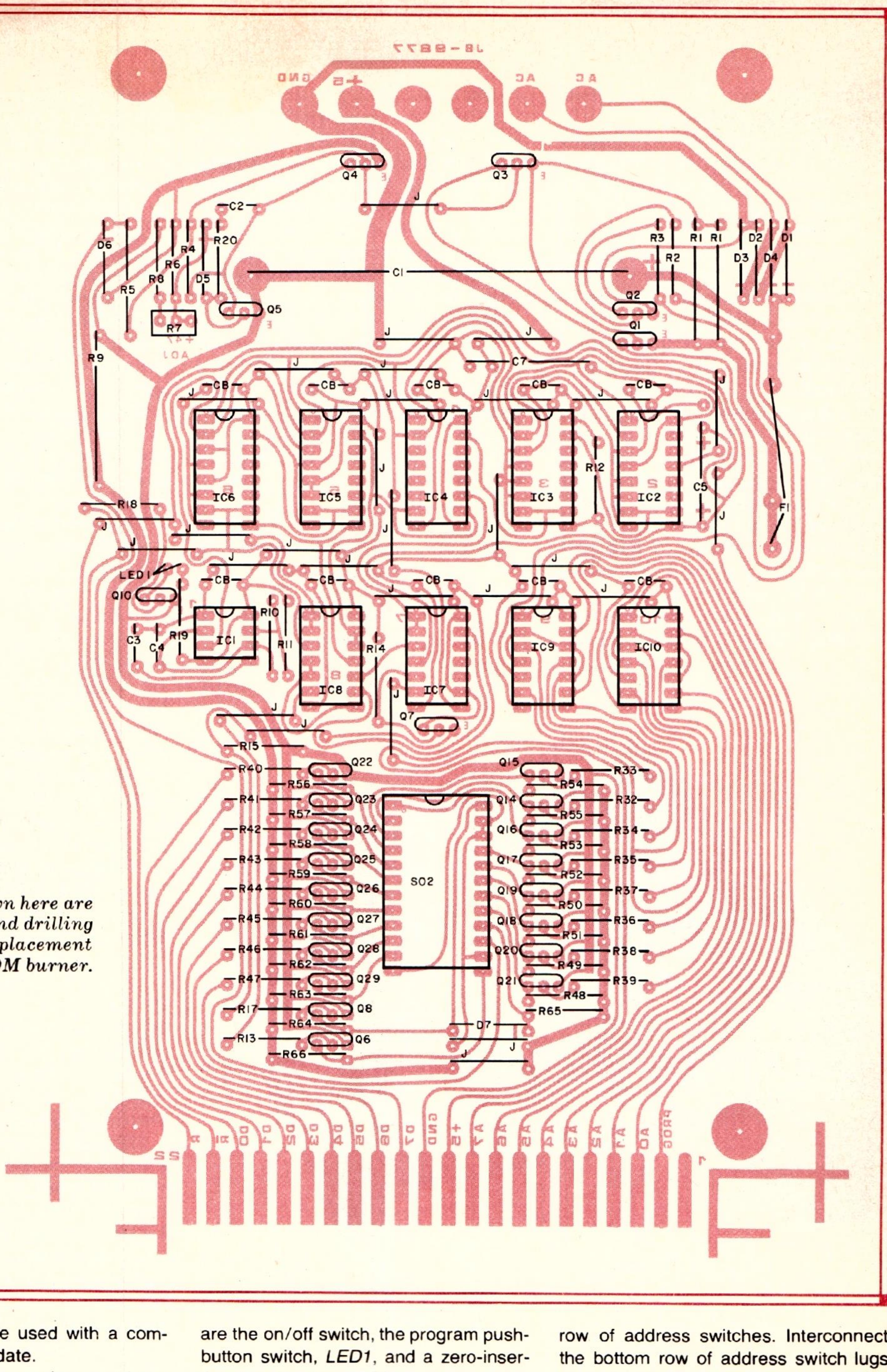


Fig. 6. Shown here are etching and drilling and component placement guides for PROM burner.

Programmer can be used with a computer at some later date.

Select a suitable enclosure whose front panel can support the eight address and write data switches in two rows (see photo). Also on the front panel are the on/off switch, the program push-button switch, *LED1*, and a zero-insertion-force 24-pin PROM socket. Identify the switches and controls properly.

Use a length of heavy bare wire to interconnect all of the upper lugs of the top row of address switches. Interconnect the bottom row of address switch lugs similarly. Use the same technique on the data switches. Using insulated wire, connect the upper lugs of the address switches to the upper lugs of the data switches. Do the same with the lower lugs—lower lugs to lower lugs.

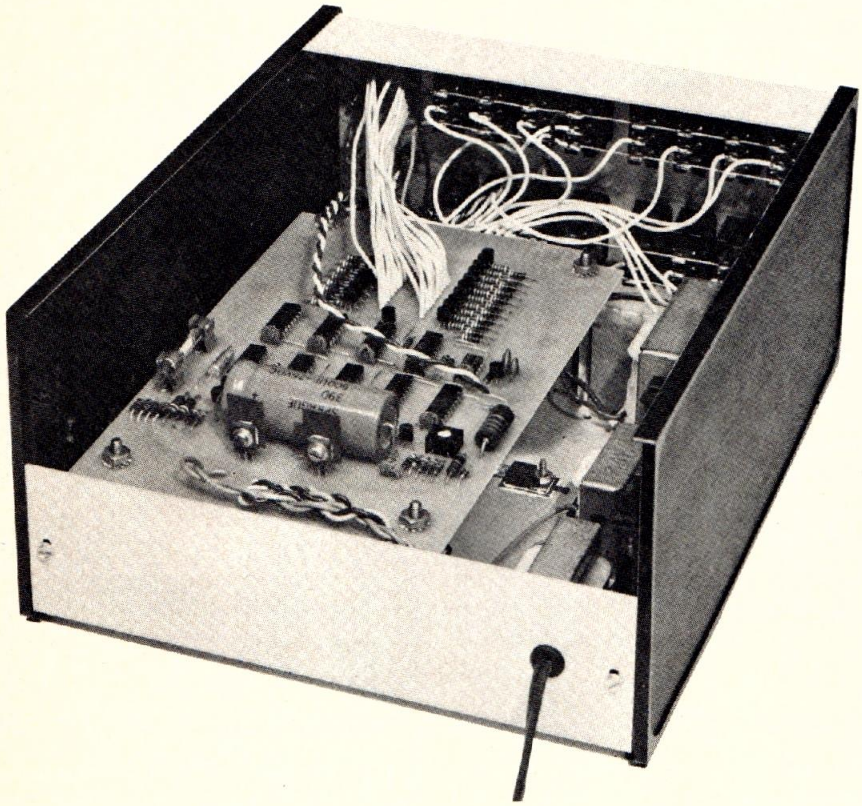
Using the small insert schematic of the *S18* circuit shown in Fig. 2, connect the normally closed contact of this switch to the top bare wire (gnd) of the address or data switches. Connect the two resistors and capacitor to the switch as shown, using the bottom lugs of either the address or data switches for the 5-volt connection.

Mount transformer T1 on one side of the chassis bottom plate. The rectifier, filter capacitor, and 5-volt regulator for this supply can also be mounted on the bottom plate of the chassis. The pc board will be mounted on spacers so that it will not contact the components mounted within the chassis. Using the four large corner holes in the pc board as a guide, and with the edge connector toward the front panel, mark and drill the four spacer mounting holes.

With the pc board held in its final mounting position (edge connector facing the front panel), cut lengths of insulated wire long enough to fit easily between the *S02* board position and the 24-pin front-panel socket. Do the same for the program switch and *LED1*. Make similar connections from the edge connector to the center lug of each of the address and data switches. A pair of wires will also be needed from the edge-connector 5-volt pad to the bottom lugs of the switches. You will also need insulated leads from the two ac-pads and the 5-volt ground pads (on the pc board edge opposite the connector) to interconnect to the power supply circuits.

Drill a hole in the rear apron of the chassis and put a grommet in it for the ac line cord. Make sure all ac connections are well insulated.

After all the wiring is installed, the board can be mounted on spacers. Do not tighten the mounting hardware, however, because the missing transistors will have to be installed after performing the following Checkout procedure.



The 5-volt supply is mounted under the pc board. With a little care, as shown here, a very professional look can be attained.

Checkout. Be sure transistors *Q8* and *Q14* through *Q29* and the +47-volt line connection are not installed until after the regulator checkout is complete.

After double checking the wiring (and pc board), adjust potentiometer *R7* to its maximum series resistance, then temporarily jumper the collector of *Q2* (Fig. 5) to ground to enable the regulator. Apply ac power to the high-voltage and 5-volt power supplies and check for the presence of +75-volt dc across filter capacitor *C1*. If necessary, reverse the secondary connections.

Using a dc voltmeter of known accuracy, monitor the voltage across *R9* (Fig. 5) and adjust *R7* to obtain +47 ±1 volts. Leave the voltmeter connected across the 47-volt line.

The current limiter is checked by *momentarily* shunting *R9* with a 68-ohm, 2-watt resistor. The voltage should drop to approximately 25 volts. If not, check *Q1*, *Q3* and *R1*.

Remove the temporary jumper from the collector of *Q2* and note that the output voltage drops to zero. If not, *Q2* is faulty or is being prematurely enabled by *IC7*. Between programming cycles, *IC7* should be completely cleared.

Using pushbutton switch *S1B* (Fig. 2), apply a pulse to the program command line and verify that the +47 volts occurs for about half a second. If it does, it is a good indication that the counters and clock are functioning normally.

The 47-volt line and the transistors can now be installed.

If you do not have a zero insertion-force socket, before installing the first PROM, loosen up the holes in the PROM socket using the leads of a ¼-watt resistor. This should be done since the pins of many 1702A PROM's are fragile and may be bent trying to force them into a tight socket.

With power applied, insert an erased EPROM in the socket, set the address and data switches in accordance with the first location of your truth table, and apply the programming command (*S18*). That location will be programmed within half a second. The optional LED programming indicator may be used to watch this timing.

You now have 255 more locations to go. If you use the microprocessor option (Fig. 3) and a suitable program, the EPROM can be programmed in just a few minutes. ◇