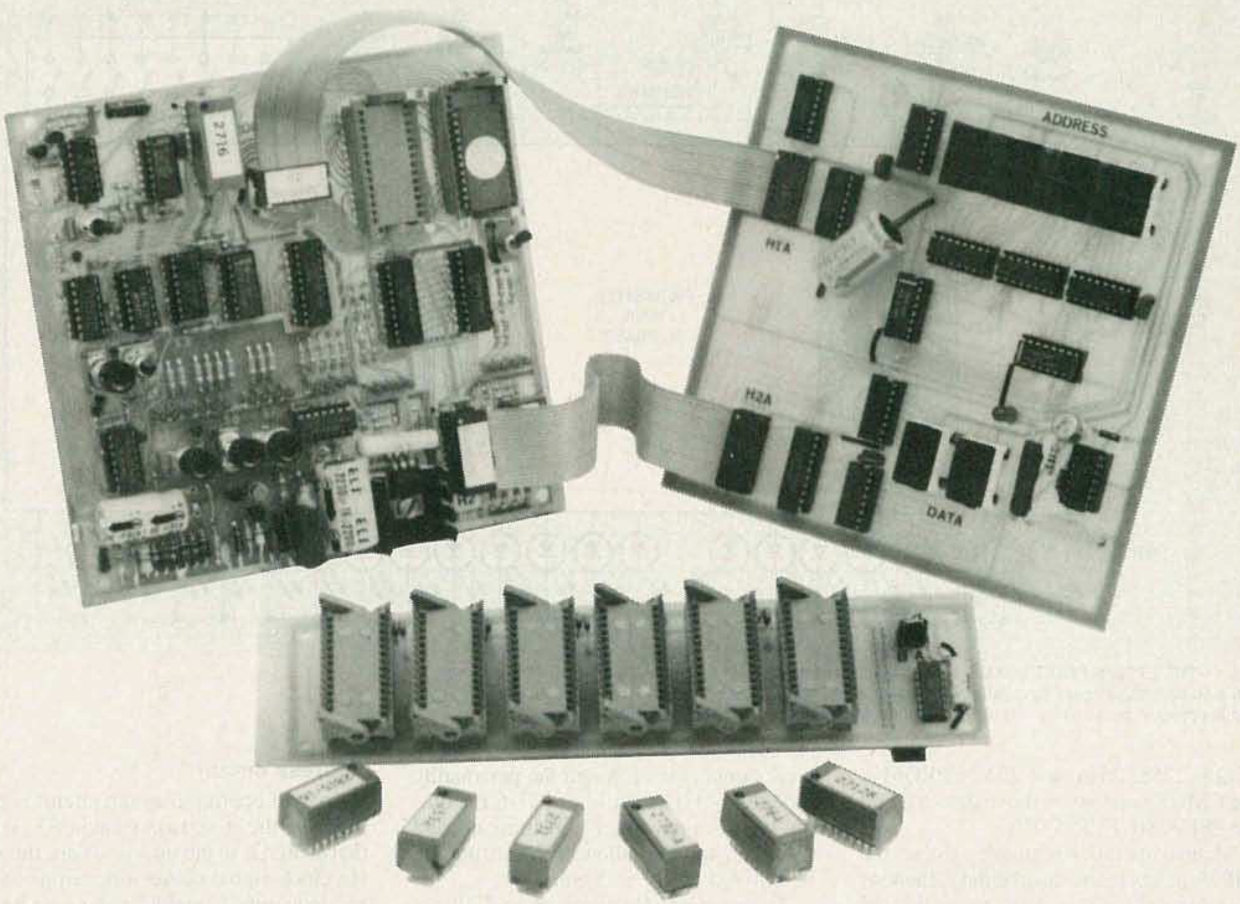
BUILD THIS



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 accommodate a variety of EPROM's. The personality module matches the operating requirements and pin assignments of each EPROM to the address, data, programming and timing signals developed on the programmer board. By using the appropriate personality module, you can program the 2716, 2732, 2732A, 2764,

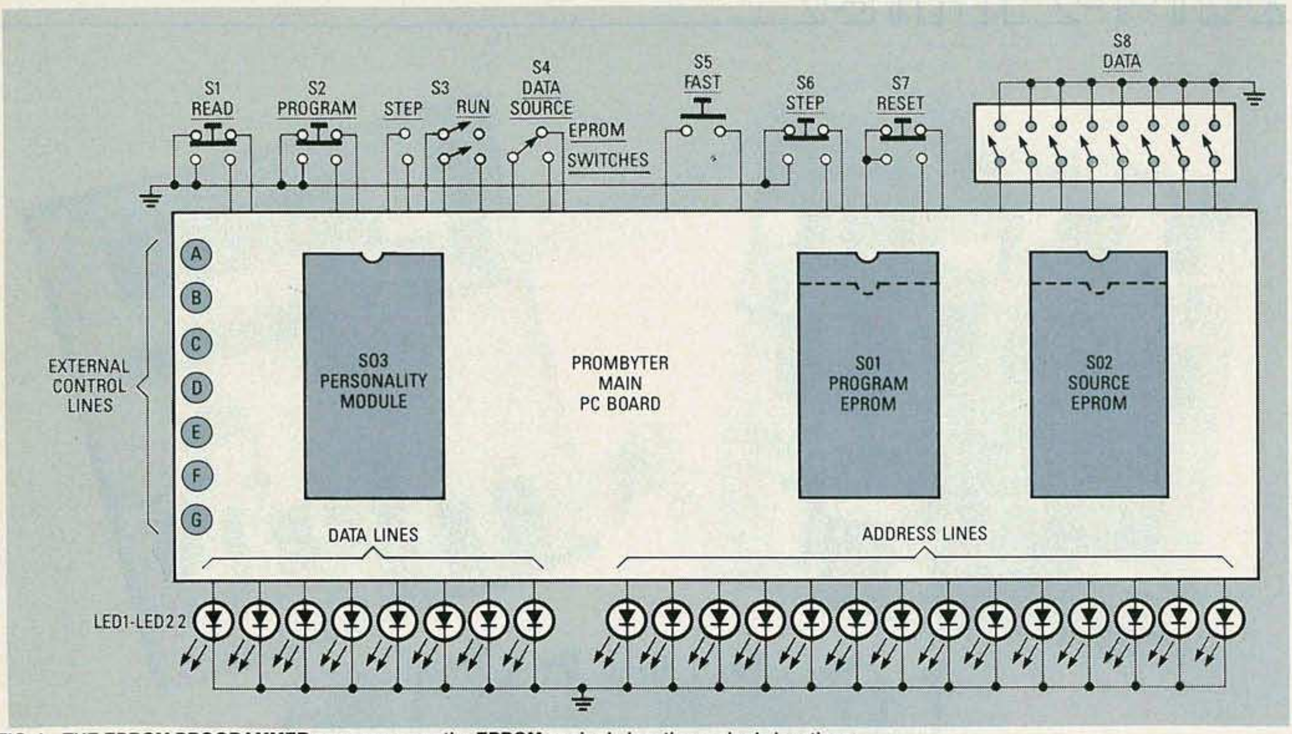


FIG. 1—THE EPROM PROGRAMMER can copy an entire EPROM or single locations; single locations can also be programmed manually. Separate sockets are provided for source and program EPROM's, and for plug-in personality modules that feed each type of EPROM the proper signals.

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 what "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.

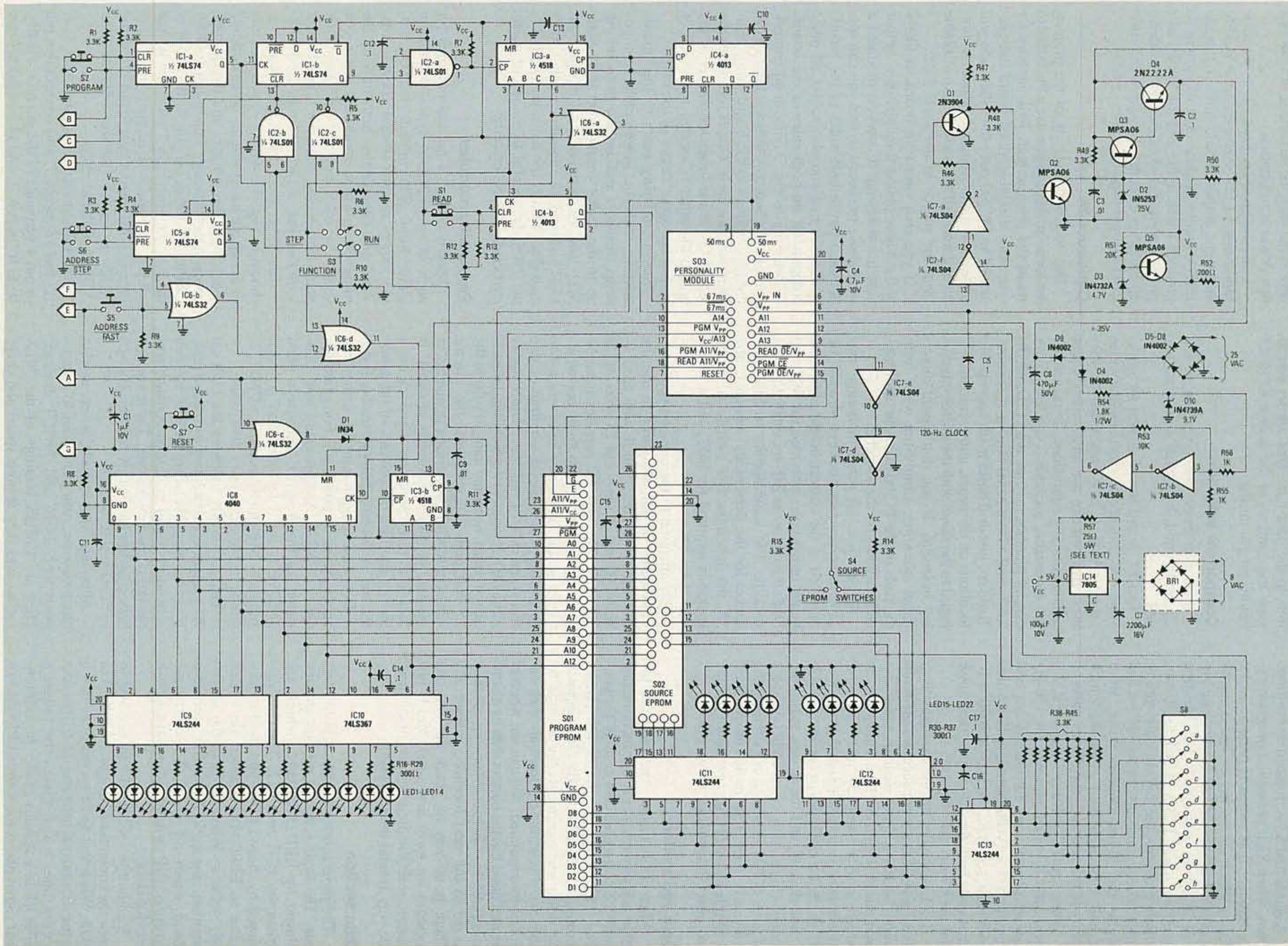


FIG. 2—THE EPROM PROGRAMMER IS DRIVEN BY A 120-Hz clock circuit that is generated by IC7-c and IC7-d; LED's indicate addresses and data in binary.

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

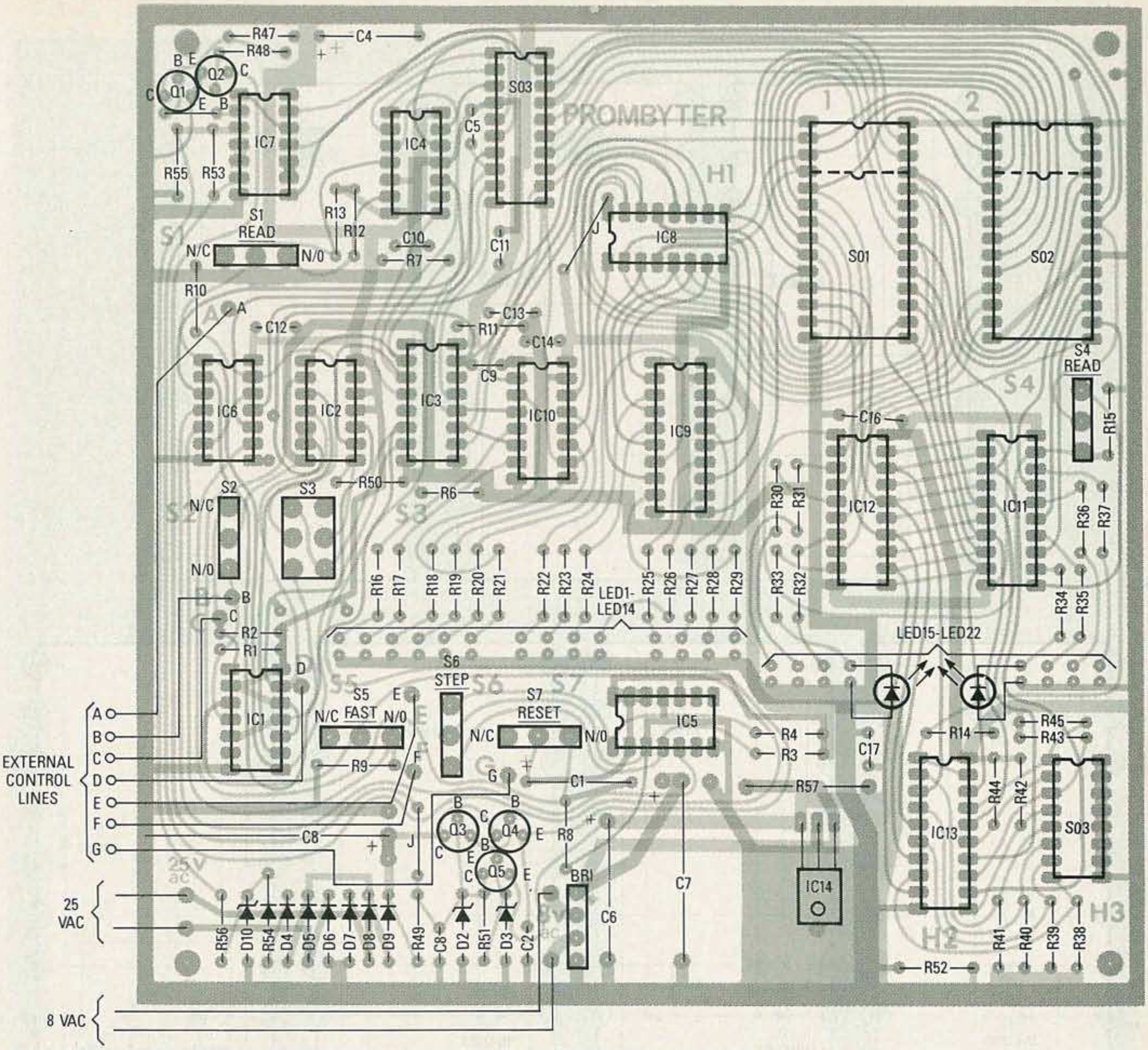


FIG. 3—**THE EPROM PROGRAMMER'S PC BOARD is double-sided; all components mount as shown here. Since a home-made board's holes are not plated through, the IC sockets must be mounted slightly above the board so that the socket pins can be soldered to the top side of the board.**

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; gang board, $10.00; set of three boards, $45.00.

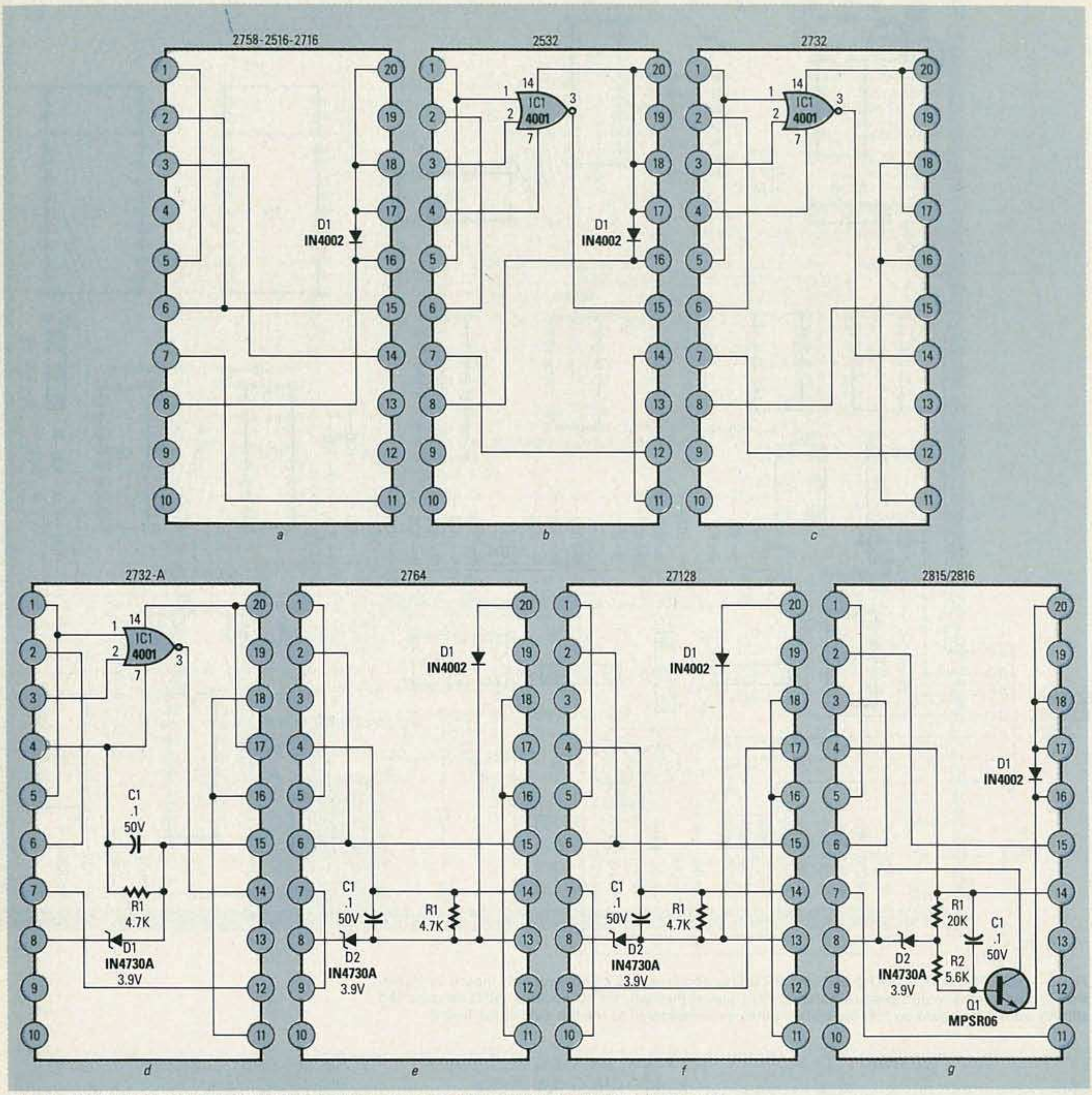


FIG. 4—EACH PERSONALITY MODULE is built on a 20-pin header; see the text for information on building the 2532, 2732, and 2732A modules.

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.

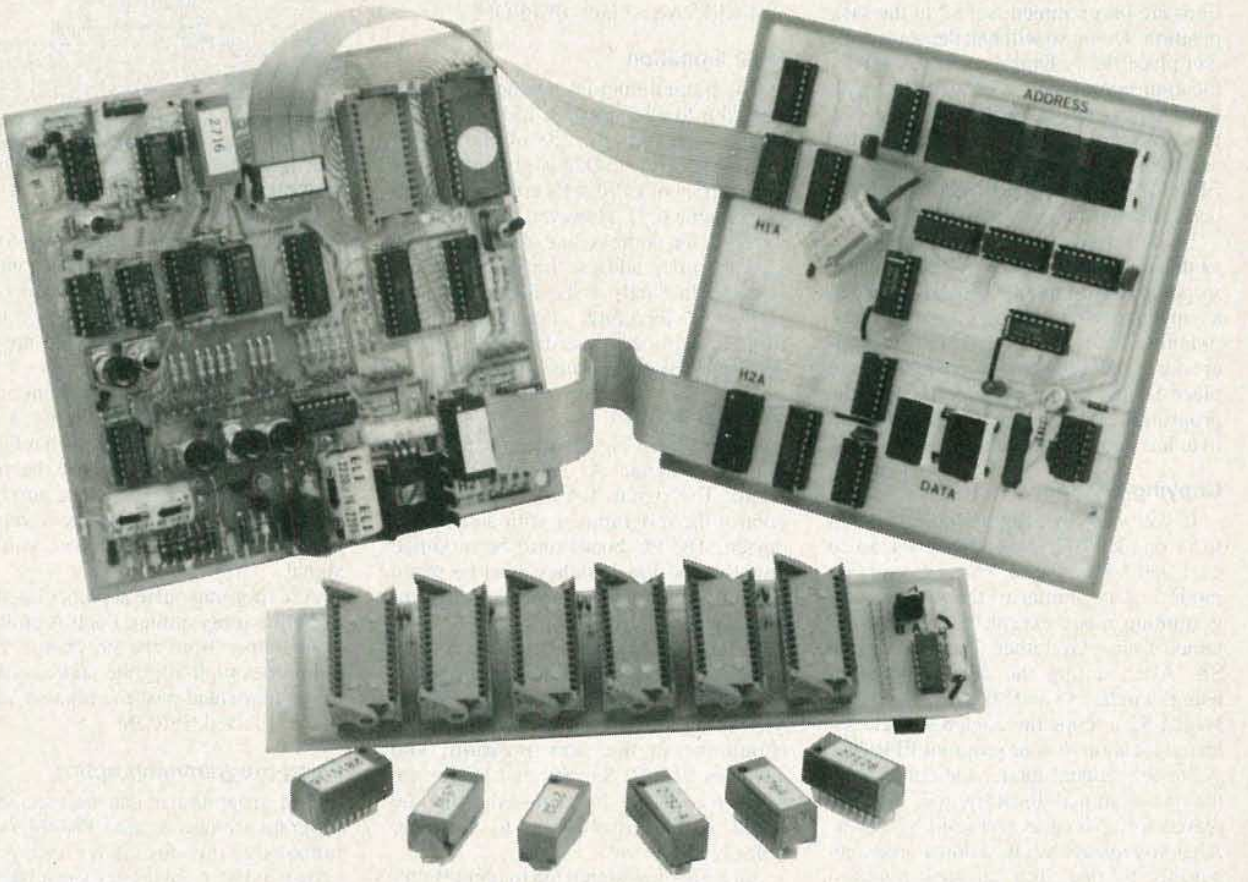
After checking the +5-volt supply, measure V_{PP} across C8. The voltage there shouldn't exceed 40 volts. You need only about 27 volts to do the job; a higher voltage could destroy one of the transistors.

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



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	S3	S4
	Switch Settings	
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 in

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

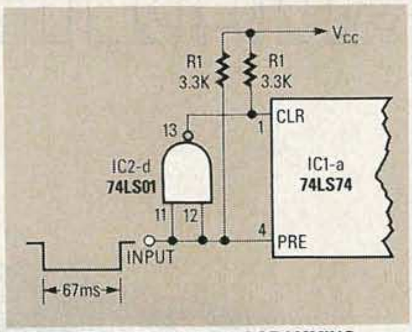


FIG. 5—TO AUTOMATE PROGRAMMING, connect the unused gate of IC2-d as shown here. Your control signal should be low-going and have a width of at least 67 ms.

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.

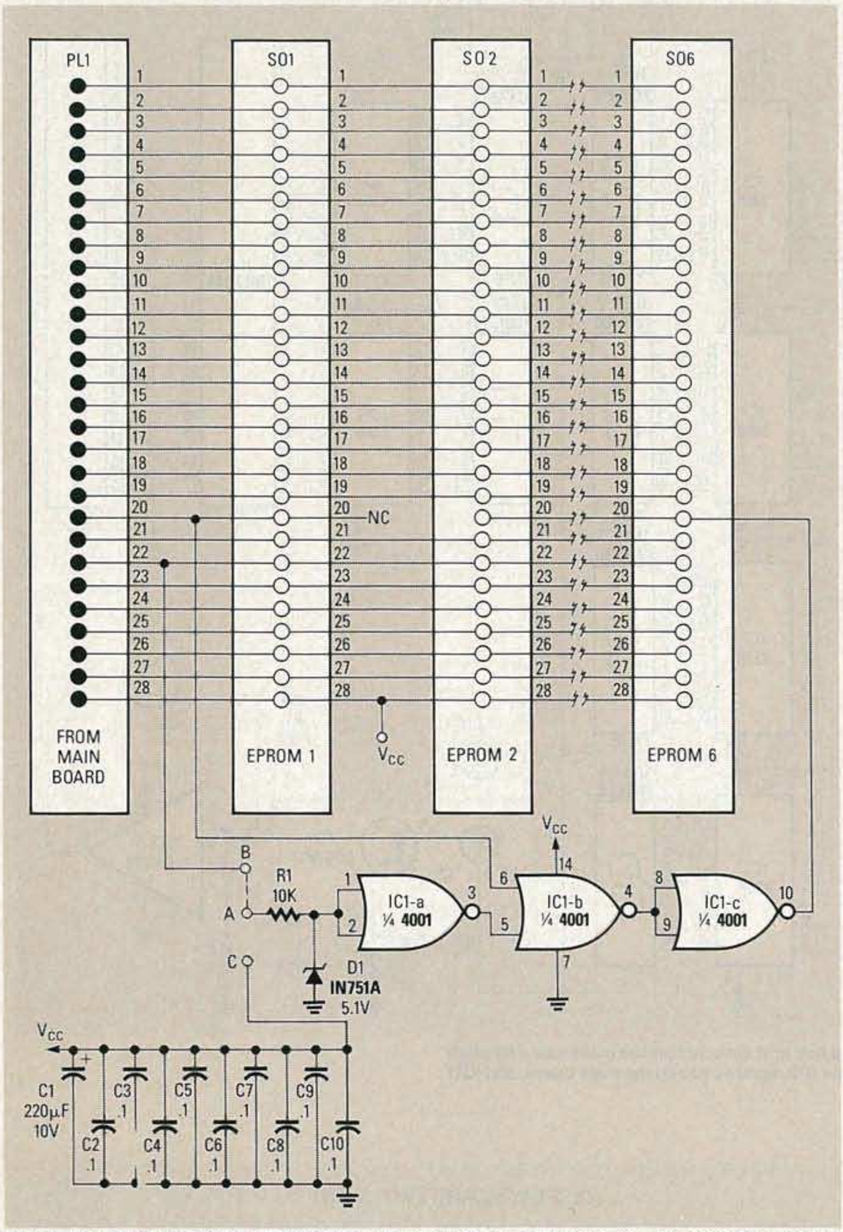


FIG. 6—THE GANG-PROGRAMMING BOARD connects to the main board via a cable connected to PL1. To avoid the possibility of data-bus conflict, the gates of IC1 allow you to disable reading EPROM's in sockets SO2–SO6 (when the jumper is connected between A and B).

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-

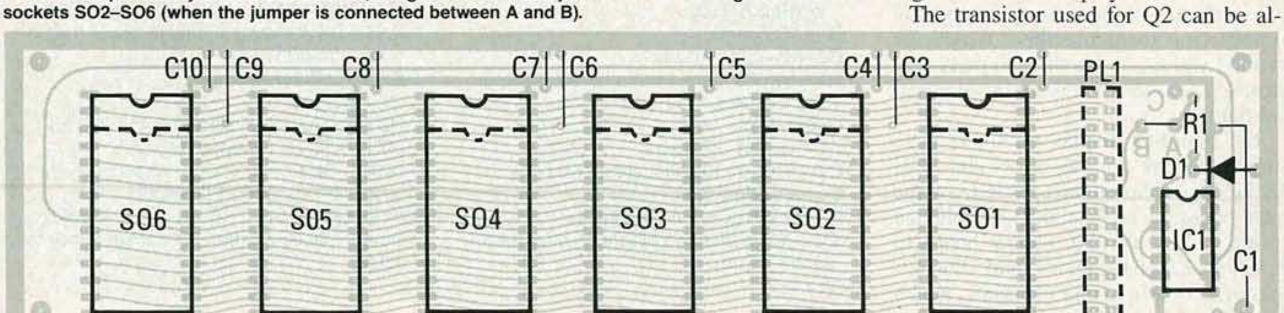


FIG. 7—MOUNT ALL PARTS except PL1 on the component side of the board. The jumper conncting points A and B disables reading from sockets SO2–SO6.

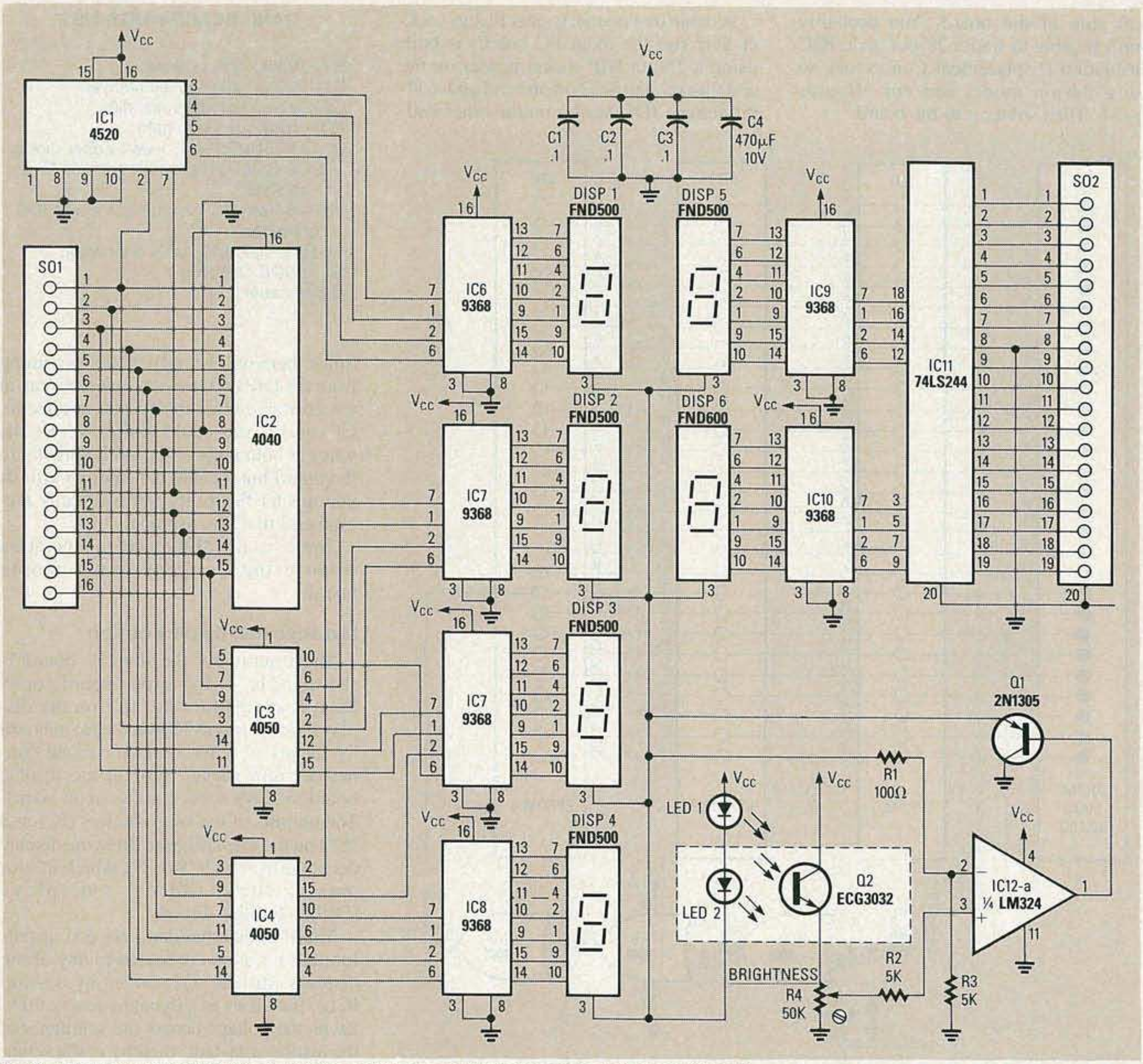


FIG. 8—THE DISPLAY BOARD attaches to the address bus and data bus on the main board through cables connected to SO1 and SO2, respectively. Counter IC2 replaces IC8 on the main board, and IC11 replaces IC13.

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.

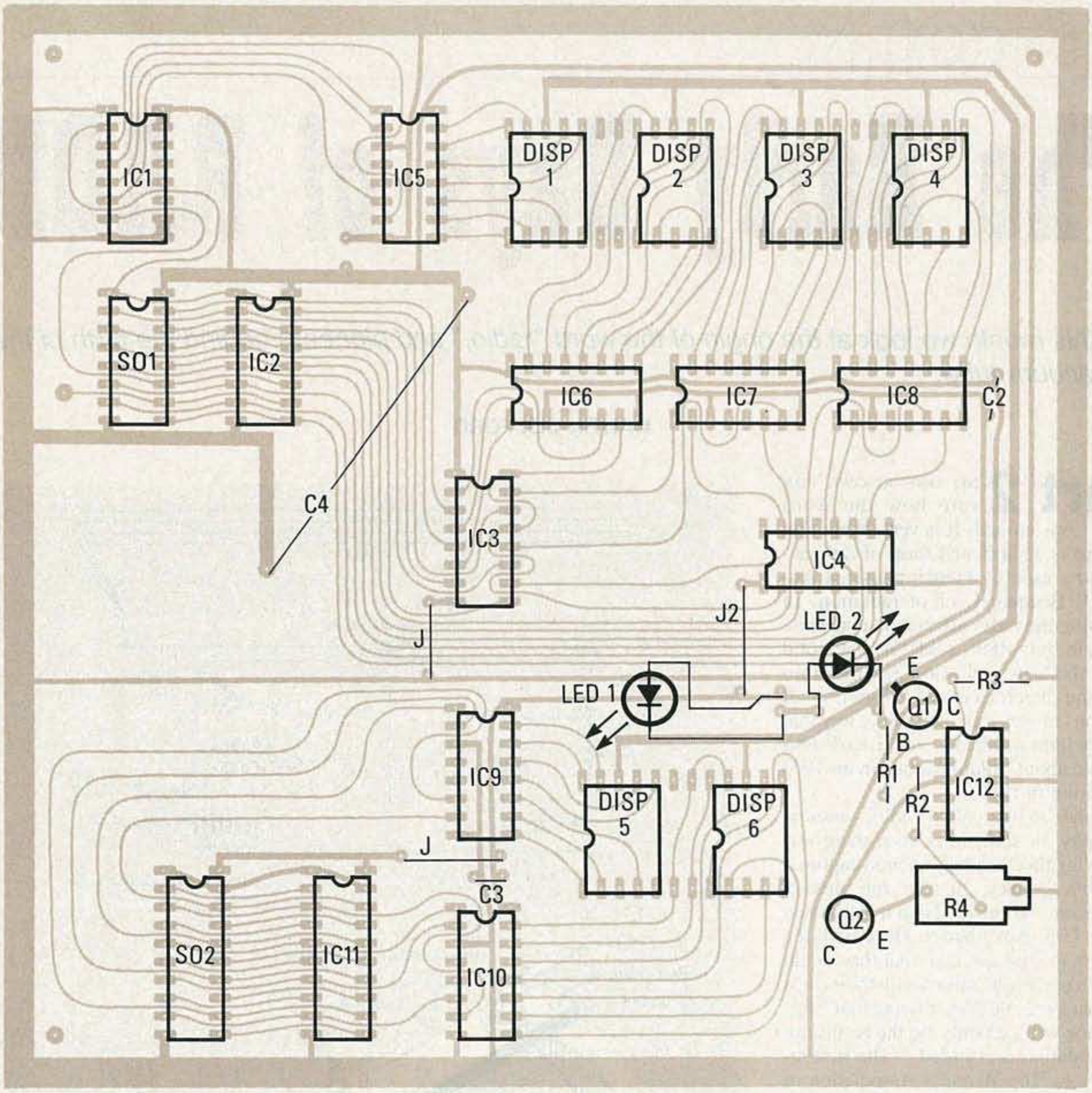


FIG. 9—PHOTOTRANSISTOR Q2 AND LED2 must be mounted so that their receptive and transmissive surfaces receive no ambient light. Heatshrink or tape them together. The foil pattern for this board is shown in PC service.

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.

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**