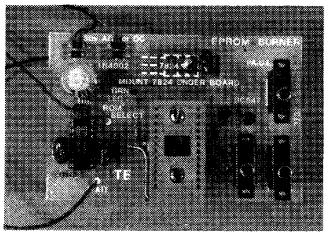
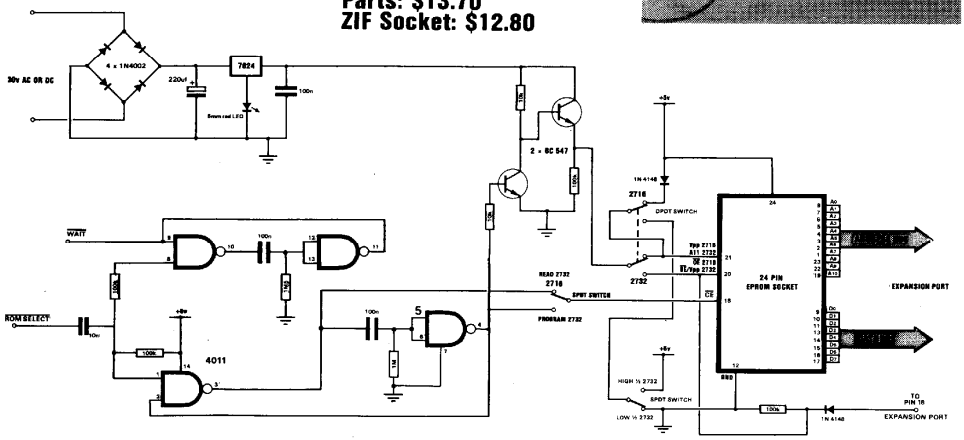
EPROM BURNER

PC Board: $3.50
Parts: $13.70
ZIF Socket: $12.80





An EPROM BURNER is the greatest thing to hit the TEC since the regulator was put under the board!

It adds the versatility you have wanted for ages.

To be able to save a program in a permanent form is the final goal of programming.

The TEC RAM CARD and EPROM BURNER combine to make a system capable of generating, testing and producing programs in hard form which can be saved, stored or sold - programs capable of emulating almost any task imaginable.

You can take any project from any magazine or book and convert it to a micro design with a consequent saving in parts, space and cost. Its capability can be increased and its reliability improved by using a tried-and-proven micro design.

By using the NON-VOLATILE RAM as the intermediate stage and the MON 2 monitor (with insert and delete functions) for the production of the program, you can generate, and have running, any machine code program, before burning it permanently into an EPROM.

Burning an EPROM is the final stage and you should be thoroughly satisfied with the performance of a program BEFORE-HAND as it cannot be changed once it is burnt.

This is not entirely true as you can change some values and 'burn-down' any value to zero.

This is an important fact to remember when programming and we will explain what we mean:

EPROMs are purchased in blank form and this means the cells (of which there are 16,384 in a 2716 and 32,768 in a 2732) do not hold any charges of electricity.

Due to buffering circuits in the EPROM, the output from a blank device will be a set of HIGHs. Advantage is made of this as you will see. Eight cells are accessed at a time and if the value is read, it will be 1, 1, 1, 1, 1, 1, 1, 1. But we don't want to read a blank ROM - we want to program it with useful commands and data.

In Hexadecimal notation, the blank EPROM produces FF's from each set of 8 locations. This is called a byte and as we burn each byte in the EPROM burner, we convert the FF's into a lower value. If we don't burn a particular location, its value remains FF.

If we burn all 8 cells, the resulting value will be 00 and the designers of micro-processors (such as the Z-80) have given a very clever command to this value. It is "NO-OPERATION" in which the processor glides over the location with-out affecting any of the remaining program.

PARTS LIST

2 - 10k
3 - 100k
1 - 1M
1 - 1M5

1 - 10n greencap
3 - 100n
1 - 220uf 35v electro

2 - 1N 4148 signal diodes
4 - 1N 4002 power diodes
1 - red LED

2 - BC 547 transistors
1 - 4011 IC
1 - 7824 regulator

1 - 14 pin IC socket
1 - 24 pin wire-wrap socket
1 - 24 pin DIP header
1 - 24 pin ZIF socket **12.80 EXTRA**

2 - 10cm hook-up flex
2 - matrix pins (for TEC)
2 - matrix pin connectors
4cm heat-shrink tubing
1 - 6BA nut and bolt
3 - DPDT slide switches

1 - EPROM BURNER PC BOARD

The advantage of having a No-OPeration command as **00** means any location can be 'burnt-down' to 00 if it is required to be removed.

This is where the term 'burn-down' comes from. Whenever an EPROM is burnt or programmed, the value produced is less than the starting value for the location.

We said values cannot be changed once burnt, but in some cases you can reduce the value if the following conditions are met:

The main criteria is: **the cells you wish to change must be 1's.**

The table below shows the values which can be burnt down and those which cannot.

BURN-DOWN TABLE:

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
ANY VALUE	CA86420	C9851 0	840	A9832 10	8 2 0	8 1 0	0	6 5 4 3 2 1 0	4 2 0	4 1 0	0	2 1 0	0	0	0

The bold value can be burnt down to the values shown in the column.

Values cannot be 'burnt-up' as we cannot produce HIGHs in an EPROM BURNER.

The only way we can restore HIGHs or 1's to the cells of an EPROM is to put it under an ultra violet light source, whereby ALL the locations will be erased and converted to 1's.

THE CONCEPT

Locations in an EPROM can be burnt if a voltage of 25v is applied to the Vpp pin and pin CE pulsed for 50 milliseconds.

The cells which will be given a charge of electricity will depend on the address which is being accessed and the value of data present on the data lines.

These are the only requirements and programming can be done with a simple set of switches. Unfortunately this would take an enormous length of time as 11 switches would be required for the address lines 8 switches for the data lines and each would have to be set for each byte of information.

A 2716 contains 2048 bytes and if a byte is burnt incorrectly, the whole procedure would have to be repeated.

The other inconvenience is all the bytes in the program would have to be converted to binary so that they can be loaded via the switches.

All this would take so long that the operation of burning would become a head-ache.

By using the TEC, the address values increment automatically as the program advances and the values of data are automatically converted to binary when each location is being burnt. This means you can program in Hex.

In this way an hours' work is converted to only a few minutes.

This is the function of an EPROM BURNER. It connects an EPROM to the address and data buses, provides the necessary 50 millisecond programming pulse and the 25v supply.

To hold the data steady on the data bus for the duration of the burn, it is necessary to HALT the computer. This is achieved by using another monostable connected to the WAIT line, with a pulse length which is slightly longer than 50 milliseconds.

When you think of it, 50 milliseconds is 20Hz and when you include a short additional delay for the wait function and a number of machine cycles for the execution of a program to carry out the burn operation, you arrive at a burn rate of about 15 locations per second.

Divide this value into the number of locations you wish to burn and you arrive at the length of time for burning an EPROM. That's why it may take a minute or so.

HOW THE CIRCUIT WORKS

The circuit is very simple and consists of a number of building blocks which come together via the ROM SELECT line from the computer.

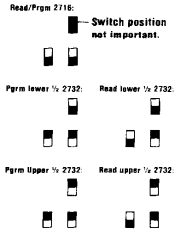
Starting at the top of the diagram, the 25v is derived from a 7824 voltage regulator which has been 'jacked up' by 1.7v by the inclusion of a red LED in the COMMON line. This gives and output of 25.7v and by the time it reaches the EPROM, a voltage drop of .5v has occured across the switching transistor. This transistor is switched via the output line of the 50 millisecond monostable.

The 25v line need not be switched ON and OFF when programming but must not be present when the EPROM is to be removed from the socket. By switching the voltage as we have done, the EPROM can be removed without damage.

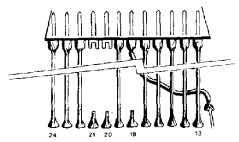
In this article we have included only a very simple burning routine which you can load into **0800** and get the project working.

The final two circuit blocks are monostables or one-shots, created from NAND gates. The lower monostable produces a 50 millisecond delay and the upper a 65 millisecond delay.

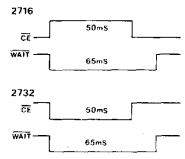
It places data on the data lines, turns on the required address lines and turns on a Chip Select line (located near the edge of the TEC PC).



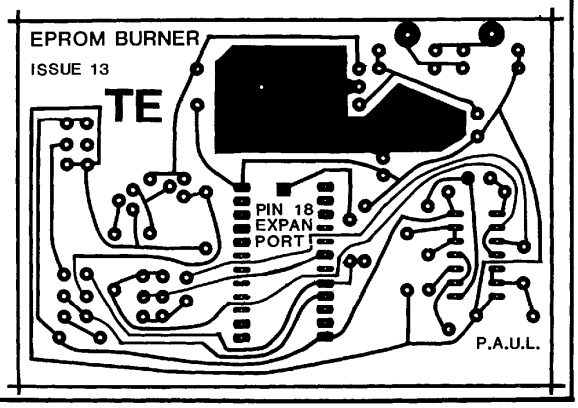
Switch positions for programming and reading 2716's and 2732's



Don't forget to add the jumper lead from pin 18 of the wire-wrap to the PC board and cut pins 18, 20 and 21.



The programming pulses differ between the 2716 and 2732.



The timing diagrams on the previous page start at the commencement of the program (see page 48 for the program). The program runs for the first 4 lines and in the fifth line the instruction is to load the contents of the accumulator into the location pointed to by the DE register pair. This instruction does three things; not necessarily in this order:

It places data on the data lines, turns on the required address lines and turns on a Chip Select line (located near the edge of the PC).

This line accesses either a RAM or ROM chip connected to it and the address starts at 1800.

This action triggers both monostables and the WAIT monostable immediately goes LOW to HALT the computer.

The program is stopped and the EPROM BURNER circuit takes advantage of the data appearing on the address and data lines. This address is loaded into the EPROM and the data placed on the cells at this particular location.

The high voltage is turned on for 50 milliseconds and at the same time CE is pulsed. This permanently puts the value of data in the EPROM.

The 50 millisecond monostable ends it timing cycle and 15 milliseconds later the WAIT monostable goes HIGH. This enables the computer to continue through the program and come to a JUMP RELATIVE instruction to bring it back to line 5.

This time DE will be pointing to the next higher location and A will contain a new value of data. The count-register-pair BC will be one less than previously. The program continues to loop until BC pair is ZERO.

At the conclusion of the burn routine the program jumps to address 0000 and the monitor program is executed to bring 0800 on the screen with the reset beep to indicate the end of burn.

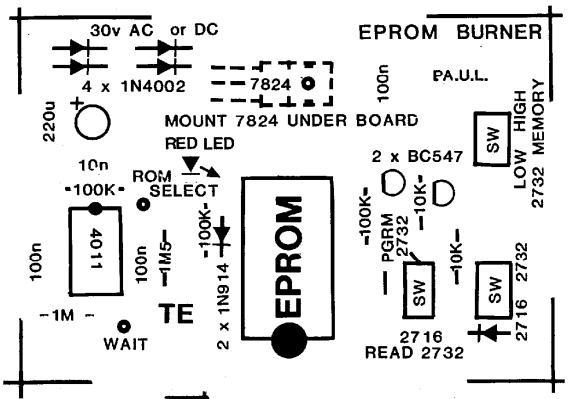
RECAP:

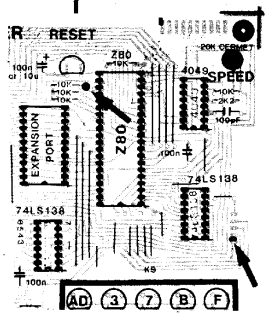
The program burns the EPROM at address **1800 - 1FFF**. To look at the data in EPROM: address **1000 - 17FF**. When the EPROM is removed from the programmer, its address will depend on the project you are using it in, but more than likely it will be the only programmed chip and thus it will be **0000 - 07FF** for a 2716 and **0000 - 0FFF** for a 2732.

CONSTRUCTION

Begin with the resistors and signals diodes. Keep them close to the board before and after soldering but be careful not to damage them with heat. Four power diodes are needed for the 30v bridge. Fit them next, along with the 10n and 100n greencaps.

There is one jumper link on the PC board and this can be made from a lead cut from one of the components.





The Chip Enable pin and Wait pin on the TEC.

The 7824 voltage regulator is mounted under the board and fixed to it with a nut and bolt. The pins fit through the holes provided and are trimmed on the topside of the board.

The two transistors and 14 pin IC socket are the next to the be added and then the red LED.

The 220uf filter electrolytic must be mounted around the correct way and the board is ready for the hardware parts.

Push the 24 pin wire-wrap socket through the holes in the PC and carefully solder it in position. **Cut off pins 18, 20 and 21.**

A short jumper goes from pin 18 of the DIP header to a solder-land on the PC board (located between pins 12 and 13). The easiest way to add this jumper is to solder one end to pin 18 of the DIP plug and the other end to the PC board BEFORE soldering the DIP plug to the wire-wrap socket.

Keep the pins of the wire-wrap full length and solder the 24 pin DIP header into position.

The three slide switches are mounted through the board if the holes are large enough. But if only small holes have been drilled, the legs will need to be extended with short lengths of tinned copper wire and the switches mounted above the board.

Finally connect two jumper leads for the WAIT and ROM SELECT lines and a length of twin flex for the 30v line.

The jumpers require matrix connectors and a short piece of heat-shrink tubing over them to make them sturdy. Heat the tubing with a flame to make it shrink over the connector.

The twin flex requires a 2 pin DIN plug so that it will fit the 30v socket on the TEC POWER SUPPLY.

Fit the 4011 IC and the board is ready.

Two matrix pins will be required on the TEC PC board to take the jumpers. These are soldered as shown in the diagram opposite and are included in the kit.

If you intend to produce a number of EPROMs you will need a ZIF socket. These are expensive (too expensive), but are essential if you want to avoid the damage caused by constantly inserting and removing EPROMs from an IC socket. The wire-wrap socket will accept about 50 insertions and removals before it gets a little weak. If the pins do not make good contact, the wrong values will be burnt.

When fitting a ZIF socket, push it firmly into the wire-wrap by starting at one end and gradually introducing the pins, two at a time. You cannot force all the pins together at once.

Pin 1 of the EPROM is towards you and this means the lever of the ZIF socket is also towards you.

Cont. P. 48. . .

EPROM BURNER

. . .cont. from P. 22.

The lower switches must be set for 2716 or 2732.

When burning 2716's the upper switch position does not matter as it is not in circuit.

When burning 2732's the upper slide switch selects the UPPER 2k or LOWER 2k of the 2732.

The switch closest to the EPROM is placed in the upper position when programming 2732's and in the lower position to read them.
This switch is placed in the lower position when programming and reading 2716's. Refer to the set of diagrams before carrying out any operation.

The high voltage is derived from a 30v supply. This can be the TEC POWER SUPPLY or from a 30v AC transformer. Very little current is required, however the voltage must not go below 30v or the regulator will dop out. This is because we are generating 25.7v and the regulator requires 3-4v across it for regulation.

Connection of the EPROM BURNER board to the TEC is via a 24 pin wire-wrap and DIP header plug. The board fits in the expansion socket and requires a WAIT line and ROM SELECT line.

The ROM SELECT line is pin 12 of the 74LS138 and WAIT is pin 24 of the Z-80.

Connect these lines to the TEC and plug the EPROM BURNER board into the expansion socket. Connect the 30v supply and the red LED will illuminate to indicate all is ready.

You can burn a new EPROM or blank locations in an old ROM. You can even burn old locations providing they fulfill the requirements mentioned previously.

THE PROGRAM

The BURN PROGRAM can be placed anywhere in the Monitor ROM or typed into the RAM. If placed in the RAM, it will need to be typed each time an EPROM is to be burnt.

The program is very simple and does not have any checking facility to prevent it burning over previous program.

The absence of this means you can burn or reburn any location(s) anywhere without having to break a safety lock.

The first three lines of the program contain variables which have to be set each time you want to burn an EPROM. For this reason, the three lines must be typed into RAM with a fourth line to provide a call or jump to the remainder of the program. The rest of the program can be located in ROM (at say 0700).

Here's how it is done:

You will need a blank 2716.

The first stage is to transfer the MONitor program into the new EPROM. Load the following into 0800:

```
LD DE 1800      800   11 00 18
LD HL 0000      803   21 00 00
LD BC 06FF      806   01 FF 06
LD A(HL)        809   7E
LD (DE),A       80A   12
PUSH BC         80B   C5
DJNZ FE         80C   10 FE
DJNZ FE         80E   10 FE
POP BC          810   C1
INC HL          811   23
INC DE          812   13
DEC BC          813   0B
LD A,B          814   78
OR C            815   B1
JR NZ           816   20 F1
Restart 0000    818   C7
```

Make sure the switch selects 2716. Push RESET, GO. The TEC screen will blank for about 2 minutes while the program is burning.

When the screen reappears you can check the operation by addressing 1000 and read the locations. Compare them with 0000 and confirm the program has been transferred.

The next stage is to add the burn routine to the MONitor ROM. This is done at 0700. Change the values at 0800 to:

```
11 00 1F
21 09 08
01 10 00
```

Push RESET GO and the program will be transferred in a few seconds.

You have now produced a MONitor ROM with a burn routine at 0700. Place the new ROM into the TEC and it will start up with 0800.

To use the BURN ROUTINE, type the following at 0800:

```
11 ____ ____ TO: ROM address + 1800H
21 ____ ____ FROM: RAM address
01 ____ ____ No of hex bytes
C3 00 07
```

Programs to be burnt into EPROM are placed at **0900** and can extend to **0FF0**. To transfer these programs to EPROM, place the following at **0800:**

```
11 00 18
21 00 09
01 F0 0E
C3 00 07        Push RESET GO.
```

EX: 80 byte program at 0900 to 0000 in EPROM:

at 0800:

```
11 00 18
21 00 09
01 80 00
C3 00 07        Push: RESET, GO.
```

A6 byte program at 0A00 to 0180 in EPROM:

at 0800:

```
11 80 19
21 00 0A
01 A6 00
C3 00 07        Push: RESET, GO.
```

40 byte program at 0C00 to 02C0 in EPROM:

at 0800:

```
11 C0 0A
21 00 0C
01 40 00
C3 00 07        Push: RESET, GO.
```

If you type a program at **0800**, the BURN ROUTINE can be located at **0900:**

```
11 xx xx
21 00 08
01 xx xx
C3 00 07
```

decrement to **0900** Push: GO, GO.

Before starting any programming you should fill the TEC RAM with FF's. This will allow any non-program locations to be transferred and retain the value FF.

To fill RAM with FF's:

```
11 FF FF
D5
C3 03 08
Reset, GO.
```

Programs can be transferred from EPROM to the TEC memory via the following routine:
at 0C00:

```
11 00 08
21 00 10
01 ____ ____ (No of bytes)
ED B0
C7
```

decrement to **0C00 Push: GO, GO.**

Program will transfer very quickly - This is not a burn routine but a DUMP ROUTINE which can also be used for the non-volatile RAM project.

Example: 80 bytes in EPROM at 0000 to 0900 in TEC RAM.

at 0800:

```
11 00 09
21 00 10
01 80 00
ED B0
C7          Push Reset, Go.
```

(0000 in EPROM means page-zero in EPROM).

Ex: **80 bytes in EPROM at 0630 to 0900 in TEC RAM.**

at 0800:

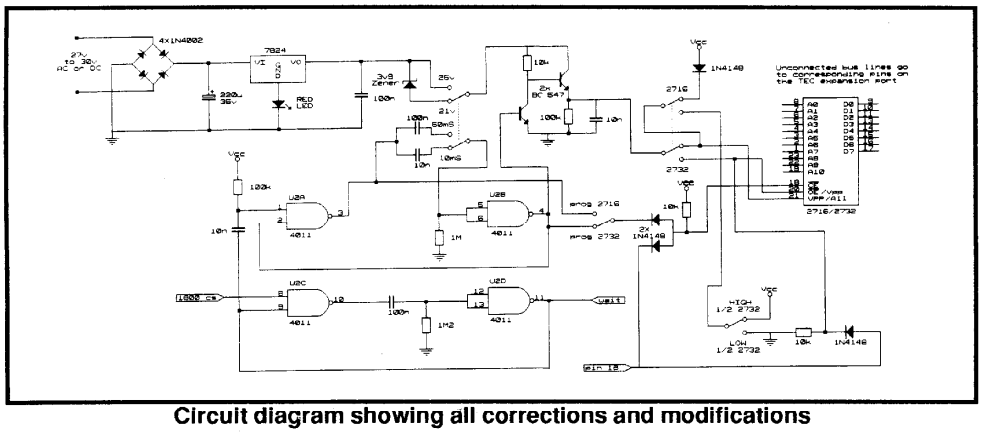
```
11 00 09
21 30 16
01 B0 00
ED B0
C7          Push: Reset, Go.
```

Before attempting any transfer, you must write the necessary program on a piece of paper using one of the examples in the text. Check it carefully then type it into the TEC at 0800 (or other location as explained).

Using the programmer and the non-volatile RAM in conjunction with the TEC will open up lots of possibilities. Programs can be used on the TEC or MICROCOMP and you will begin to see how everything is going together.

EPROM PROGRAMMER REVISITED

Parts $2.30



Circuit diagram showing all corrections and modifications

CIRCUIT DIAGRAM CORRECTION

A mistake has been made with the circuit diagram on page 20 in issue 13.

The 100k resistor between pins 8 and 1 of the 4011 does not exist on the board and pin 8 is actually directly connected to the ROM SELECT LINE. It is not coupled through the 10n capacitor (via the 100k mentioned above) as shown.

CIRCUIT UP-GRADES

If your EPROM programmer is working ok and you're completely satisfied with its performance, perhaps it is best left alone. There are two modifications though, that are HIGHLY RECOMMENDED:

The first is the 100k resistor on the left-hand side of the EPROM socket (next to a diode) SHOULD BE REDUCED to 10k. This will allow for far more reliable readings (if yours doesn't read at all or very poorly, then this will almost certainly fix it).

The second is a 10n greencap is connected across the 100k resistor next to the EPROM socket on the right-hand side of the board (when looking at it from the top).

This 10n greencap is to prevent spikes from damaging the EPROM.

There are some other very handy mods to make. This next one will make it possible to read from 2732 (4k EPROMs) without having to slide the switch across. The BIG advantage of this is that it is possible for the software to read from the 2732 just after you have programmed each location. The software can then diagnose a failure and re-try or abort quickly. The software routine is provided below which will do this for either a 2716 or 2732.

Three additional parts are required for this mod. They are two IN 4148 diodes and a 10k resistor.

The first diode is soldered between the DIP-HEADER and the EPROM socket. The cathode (the end with the band on it) is soldered to pin 18 of the DIP-HEADER and the anode is soldered to pin 18 of the EPROM socket. Next, the track running between pin 18 of the EPROM socket and the middle of (program 2716 read 2732)/program 2732 switch is cut. The anode of the second diode is soldered to the pin 18 side of the cut and the cathode is soldered onto the middle terminal of the switch. One end of the 10k resistor is soldered to the anode side of the second diode (the end connected to pin 18). The other end of the resistor is soldered to ground.

Once you have fitted this modification, it may be tested by fitting a 4k ROM into the socket and addressing 1000. You should be able to read the contents regardless of the position of the read/program 2732 switch. The high/low switch is still used to select which half of the EPROM you wish to read and the read/program switch is used to select the type of EPROM you wish to program.

The next mod is a little more involved but is an important one if you wish to re-program some of the EPROMs supplied by TE.

The programming requirements of some types of more modern (but now obsolete) EPROMs are not compatible with the current set-up of the EPROM programmer. This mod allows the EPROM programmer to be used with a wider variety of EPROMs. The mod does this by switching the programming voltage from 25v to 21v and reduces the programming pulse from 50mS to 10mS.

The parts required for this mod are: one DPDT switch, one 10n greencap,. one 100n greencap, a 3v9 zener diode

cont. P 45 . . . . .

PARTS LIST

(For all mods)

2 - 10k

1 - 10n greencap
1 - 100n greencap

2 - 1N4148 diodes
1 - 3v9 Zener diode

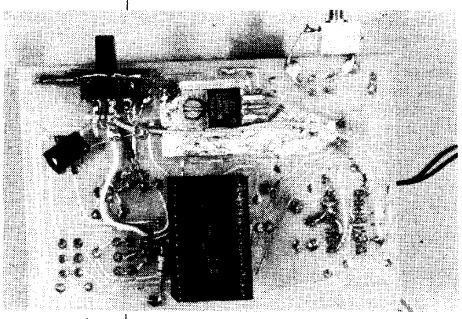
1 - DPDT switch
1 - 10cm tinned copper wire
1 - 10cm hook-up wire

. . . . .from P. 32.

and some hook-up wire. To start, mount the switch on the bottom of the PCB by drilling two holes and wrapping tinned copper wire around the switch (see photo). Next cut the track between the output of the 24v regulator and the transistor switching block. The bottom middle terminal of the switch is connected to the transistor side of the cut. Connect the bottom right-hand side terminal to the regulator output and also solder the cathode end of the zener to this junction. The anode end of the zener is soldered to the bottom left-hand side of the switch. The zener, which is connected between the regulator and the high voltage switching section, drops the programming voltage by about 4v.

This completes the voltage switching section. Below is the programming pulse length mod.

The photograph on the right shows how the parts on our prototype are mounted. The description of the parts placement in the text, corresponds to this photo.



Remove the 100n greencap on the extreme left-hand side of the board (top view). Solder one end of the new 100n to the top right-hand side of the switch. Take the 10n cap and solder one side of this to the top left-hand side of the switch. The other ends of the caps are soldered together and a jumper is also soldered onto this junction. The jumper is then soldered to pin 3 of the 4011. Another jumper is soldered between pin 6 of the 4011 and the top middle terminal of the switch.

When the switch is in the right-hand position (top view), the EPROM programmer is set up for the modern 21v/10mS EPROMs.

One of these types of EPROM is being supplied by TE. It can be identified by the following markings:

TMS
2732A-25JL
LHE XXXX (DATE CODE)

To increase the reliability of the programmer, another mod is suggested. Follow the track from the ROM select line to where it joins the 10n cap. Cut both the tracks that join to the cap at this junction. Then run a link from the ROM select input pad to pin 8 of the 4011. Now run a jumper from the wait pin to the now isolated end of the greencap.

This mod slightly delays the programming pulse to the EPROM by triggering it from the wait line, not the input ROM select line.

The software for burning EPROMs provided in issue 13 is only very basic. There is one VERY IMPORTANT ADDITION to make to the issue 13 software. After you have loaded BC, DE and HL, as described in issue 13, add the following:

XOR A (AF)
LD I,A (ED 47)
JUMP 0700 (C3 00 07)

These instructions stop the noise on the expansion port which is a result of several TEC design oversights.

The following software is designed to be burnt into either a MON-1 or MON-2 EPROM at 0700.

The software is JUMP TO with the "from address" in HL, the "to address" in DE and the number of bytes in BC.

Before it attempts to burn into the EPROM, it checks that the area to be programmed contains only FF's. If not, the routine displays an "F", for FULL in the data display and halts. You may continue on and burn the EPROM by hitting "GO". Each location is checked after it is burnt and if not correct, it is reprogrammed several more times before being aborted. The routine then displays "E" for ERROR.

You must do the "read 2732" mod to program 2732 EPROMS.

An added feature to this software is that it flashes the address being programmed on the TEC display.

EPROM BURNING SOFTWARE

```
0700  AF          XOR A
0701  ED 47       LD I,A
0703  CD 90 07    CALL 0790
0706  7E          LD A,(HL)
0707  12          LD (DE),A
0708  D5          PUSH DE
0709  D9          EXX
070A  D1          POP DE
070B  CB 9A       RES 3,D
070D  D5          PUSH DE
070E  01 F0 0F    LD BC,0FF0
0711  C5          PUSH BC
0712  CD 5A 07    CALL 075A
0715  7B          LD A,E
0716  CD 5A 07    CALL 075A
0719  7A          LD A,D
071A  CD 5A 07    CALL 075A
071D  C1          POP BC
071E  01 10 00    LD BC,0010
0721  C5          PUSH BC
0722  CD 6E 07    CALL 076E
0725  C1          POP BC
0726  0B          DEC BC
0727  78          LD A,B
0728  B1          OR C
0729  20 F6       JR NZ,0721
072B  D1          POP DE
072C  1A          LD A,(DE)
072D  D9          EXX
072E  BE          CP (HL)
072F  20 08       JR NZ,0739
0731  23          INC HL
0732  13          INC DE
0733  0B          DEC BC
0734  78          LD A,B
0735  B1          OR C
0736  20 CE       JR NZ,0706
0738  C7          RST 00
0739  C5          PUSH BC
073A  01 05 00    LD BC,0005
073D  CB DA       SET 3,D
073F  7E          LD A,(HL)
0740  12          LD (DE),A
0741  10 FE       DJNZ,0741
0743  CB 9A       RES 3,D
0745  1A          LD A,(DE)
0746  BE          CP (HL)
0707  20 03       JR NZ,074C
0749  C1          POP BC
074A  18 E5       JR 0731
074C  0D          DEC C
074D  20 EE       JR NZ,073D
074F  C1          POP BC
0750  3E C7       LD A,C7
0752  D3 02       OUT (02),A
0754  3E 01       LD A,01
0756  D3 01       OUT (01),A
0758  76          HALT
0759  C7          RST 00
075A  F5          PUSH AF
075B  CD 63 07    CALL 0763
075E  F1          POP AF
075F  0F          RRCA
```

```
0760  0F          RRCA
0761  0F          RRCA
0762  0F          RRCA
0763  E6 0F       AND 0F
0765  21 B0 07    LD HL,07B0
0768  85          ADD A,L
0769  6F          LD L,A
076A  7E          LD A,(HL)
076B  02          LD (BC),A
076C  03          INC BC
076D  C9          RET
076E  21 F0 0F    LD HL,0FF0
0771  06 06       LD B,06
0773  0E 01       LD C,01
0775  7E          LD A,(HL)
0776  D3 02       OUT (02),A
0778  79          LD A,C
0779  D3 01       OUT (01),A
077B  0E 40       LD C,40
077D  0D          DEC C
077E  20 FD       JR NZ,077D
0780  07          RLCA
0781  4F          LD C,A
0782  AF          XOR A
0783  D3 01       OUT (01),A
0785  23          INC HL
0786  10 ED       DJNZ,0775
0788  C9          RET
0789  FF          RST 38
078A  FF          RST 38
078B  FF          RST 38
078C  FF          RST 38
078D  FF          RST 38
078E  FF          RST 38
078F  FF          RST 38
0790  D5          PUSH DE
0791  C5          PUSH BC
0792  CB 9A       RES 3,D
0794  1A          LD A,(DE)
0795  FE FF       CP FF
0797  20 09       JR NZ,07A2
0799  13          INC DE
079A  0B          DEC BC
079B  78          LD A,B
079C  B1          OR C
079D  20 F5       JR NZ,0794
079F  C1          POP BC
07A0  D1          POP DE
07A1  C9          RET
07A2  3E 47       LD A,47
07A4  D3 02       OUT (02),A
07A6  3E 01       LD A,01
07A8  D3 01       OUT (01),A
07AA  76          HALT
07AB  18 F2       JR 079F
```

```
07B0 EB 28 CD AD 2E A7 E7 29
     EF 2F 6F E6 C3 EC C7 47
```

PRINT-2 AND PRINT-3 SOFTWARE

With the changes to the keyboard handler routines in both MON-2 and JMON, an up-dated printer ROM has been produced.

The new software is burnt into the same ROM at higher locations. When MON-2 was released, an up-dated ROM called print-2 was included in the printer interface kits. This gave you the same routines with an altered keyboard section. It was also a little more fancy as it showed the start address on the LED display as you typed it in. Unfortunately, Print-2 did not include a "dump string at 0900" routine to replace the dump from 0800 which is now unusable as MON-2 uses 0800 for its variable storage.

With the advent of JMON, the same arrangement has been used. The JMON printer routines are located higher again, so in the one ROM you have the printer software for all three MONitors. The list routine for JMON is an improvement on both earlier software packages, as JMON's routine uses the perimeter handler to allow you to enter both a START and END address. Print-3 includes a "dump from 0900" routine which can be used with MON-2.

The ROM with the JMON routines in it is called PRINT-3 and is supplied with the printer interface as standard.

JMON's hex dump routine is at 1A20, the typing routine at 1AA0 and the "dump string at 0900" routine is at 1AC0.

Below is a dump of PRINT-3. Burn the additional section(s) in PRINT-1/2 ROM.

The graphic demonstration routines in PRINT-1 will work with all MONitors.

```
1800 3E 0D D3 06 3E 0A D3 06
     76 ED 57 17 17 17 17 57
1810 CD 5D 18 76 ED 57 82 57
     CD 61 18 76 ED 57 17 17
1820 17 17 5F CD 5D 18 76 ED
     57 83 5F CD 61 18 C3 49
1830 18 3E 0D D3 06 3E 0A D3
     06 7A CD 5D 18 7A CD 61
1840 18 7B CD 5D 18 7B CD 61
     18 06 08 3E 20 D3 06 1A
1850 CD 5D 18 1A CD 61 18 13
     10 F1 C3 31 18 1F 1F 1F
1860 1F 21 6C 18 E6 0F 85 6F
     7E D3 06 C9 30 31 32 33
1870 34 35 36 37 38 39 41 42
     43 44 45 46 FF FF FF FF
1880 21 00 08 7E FE FF 20 05
     3E 11 D3 06 C7 D3 06 23
1890 18 F1 FF FF FF FF FF FF
     FF FF FF FF FF FF FF FF
18A0 21 C3 18 7E FE FF 28 05
     D3 06 23 18 F6 06 0A 21
18B0 CF 18 7E FE FF 28 05 D3
     06 23 18 F6 10 F1 3E 11
18C0 D3 06 C7 0D 0A 0A 0A 0A
     0A 0A 12 43 30 0D FF 49
18D0 2C 44 33 32 30 2C 30 0D
     4D 31 32 30 2C 30 0D 44
18E0 38 30 2C 2D 31 36 30 0D
     4D 32 32 30 2C 2D 38 30
18F0 0D 44 31 36 30 2C 2D 38
     30 2C 31 34 30 2C 2D 31
1900 36 30 2C 32 30 30 2C 2D
     31 36 30 0D 4D 31 35 30
1910 2C 2D 31 32 30 0D 44 32
     30 30 2C 2D 31 32 30 0D
1920 4D 33 32 30 2C 2D 38 30
     0D 44 32 36 30 2C 2D 38
1930 30 2C 32 34 30 2C 2D 31
     36 30 2C 33 30 30 2C 2D
1940 31 36 30 0D 4D 33 36 30
     2C 2D 31 32 30 0D 44 34
1950 30 30 2C 2D 31 32 30 0D
     4D 34 36 30 2C 2D 38 30
1960 0D 44 34 34 30 2C 2D 31
     36 30 0D 4D 32 2C 2D 32
1970 0D 43 33 0D FF FF FF FF
     FF FF FF FF FF FF FF FF
1980 76 ED 57 E6 0F 17 17 17
     17 57 76 ED 57 E6 0F 82
1990 D3 06 18 EC FF FF FF FF
     FF FF FF FF FF FF FF FF
```

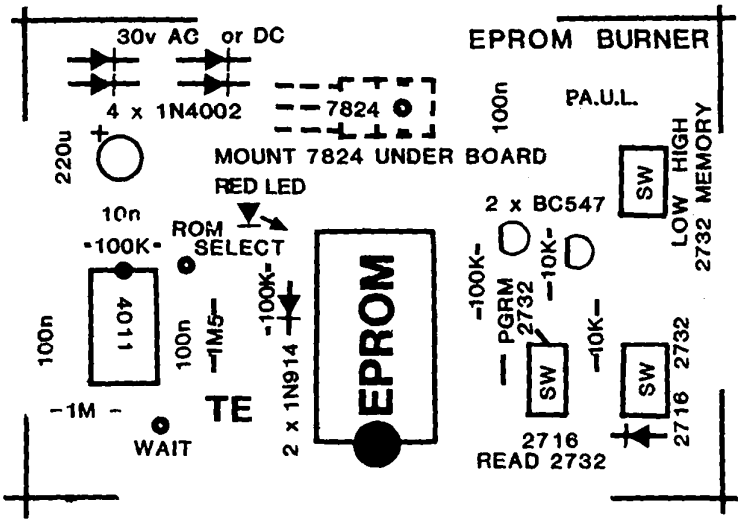
The next block is the PRINT-2 additions:

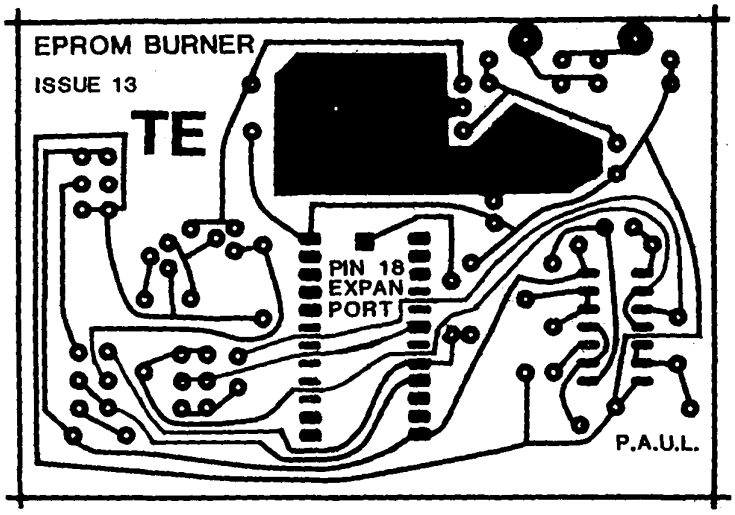
```
19A0 76 3A E0 08 E6 0F 17 17
     17 17 57 76 3A E0 08 E6
19B0 0F 82 D3 06 18 EA FF FF
     FF FF FF FF FF FF FF FF
19C0 3E 0D D3 06 3E 0A D3 06
     3E 29 21 D8 08 06 06 77
19D0 23 10 FC CD 00 1A 32 D8
     08 CD 00 1A 32 D9 08 CD
19E0 00 1A 32 DA 08 CD 00 1A
     32 DB 08 CD D8 01 CD 89
19F0 02 50 59 C3 31 18 FF FF
     FF FF FF FF FF FF FF FF
1A00 3E FF 32 E0 08 CD A0 02
     3A E0 08 FE FF 28 F6 E6
1A10 0F C6 FF CD 70 01 D6 01
     C9 FF FF FF FF FF FF FF
```

Below is PRINT-3 additions:

```
1A20 21 34 1A 11 80 08 01 0A
     00 ED B0 21 00 00 22 9C
1A30 08 C3 44 00 00 00 3E 1A
     99 08 00 01 50 1A 04 A7
1A40 04 C7 04 EB FF FF FF FF
     FF FF FF FF FF FF FF FF
1A50 3E 0D D3 06 2A 98 08 7C
     CD 82 1A 7D CD 82 1A 06
1A60 08 C5 3E 20 D3 06 7E CD
     82 1A 23 C1 10 F3 3E 0D
1A70 D3 06 3E 0A D3 06 ED 5B
     9A 08 E5 B7 ED 52 E1 38
1A80 D6 C9 F5 0F FF 0F 0F CD
     8B 1A F1 E6 0F C6 90 27
1A90 CE 40 27 D3 06 C9 FF FF
     FF FF FF FF FF FF FF FF
1AA0 CF E6 0F 07 07 07 07 57
     CF E6 0F 82 D3 06 18 F0
1AB0 D3 06 18 EC FF FF FF FF
     FF FF FF FF FF FF FF FF
1AC0 21 00 09 7E FE FF 20 05
     3E 11 D3 06 C7 D3 06 23
1AD0 18 F1 FF FF FF FF FF FF
     FF FF FF FF FF FF FF FF
```

TE


30v AC or DC
EPROM BURNER
4 x 1N4002
7824
100n
PA.U.L.
220u
MOUNT 7824 UNDER BOARD
RED LED
2 x BC547
SW
LOW HIGH
2732 MEMORY
10n
ROM
SELECT
100K
4011
100n
100n
1M5
100K
2 x 1N914
EPROM
100K
PGRM
2732
10K
10K
SW
SW
2716 2732
1M
TE
WAIT
2716
READ 2732


EPROM BURNER
ISSUE 13
TE
PIN 18
EXPAN
PORT
P.A.U.L.