PROmGRAMER

INSTRUCTIONS

IMPORTANT NOTICE

The software shipped with this PROmGRAMER should be considered preliminary. While it programs as advertised, there are some more features that we plan to add in the very near future. For this reason, it is <u>very</u> important that you fill out and return the enclosed post card, so we can send out the improved version of the software when it becomes available. This will be done at no charge.

INTRODUCTION

If you understand EPROMs and the mechanics of programming, set the switches, boot the disk, and ignore the rest of these instructions. If you are in a rush to use the PROmGRAMER, or are not interested in the technology behind this product, you may skip the next section, and GOTO "SETTING THE SWITCHES".

WHAT IS AN EPROM

Taking our APPLE as an example, we know that some of its memory is volatile, that is, when we turn off the power, the computer "forgets" what it had in it. This type of memory is called RAM (Random Access Memory), or, more correctly, R/W (Read-Write) memory, since you can write information into it, and read it out again. However, there is some memory that is not volatile. We know that as soon as we turn on the computer, some programs are available to us, even without a disk. Examples of these programs are APPLESOFT and the system monitor. These programs are stored in a non-volatile memory called ROM (Read Only Memory). [We leave it as an exercise to the reader to figure out what a WOM (Write Only Memory) is good for].

A major problem with ROMs is that they are manufactured with the program in place, and it is impossible for the user to change the memory. The cost to make just <u>one</u> is in the vicinity of $3,000 (U.S.). Obviously, this is too high to be practical, unless you make many of them, since each additional one is only a few dollars.

Along came the PROM (Programmable Read Only Memory). This device is manufactured with many tiny fuses in it. Where a fuse exists, a logical ONE would be read. To change it to a logical ZERO, a high pulse of current is sent through the fuse, melting it. The obvious problem with this is that you can program the device once. If you make a mistake, buy another one, and try again.

Which brings us to the EPROM (Erasable Read Only Memory).
This device is also manufactured with each "cell" reading a
logical ONE. If you want a logical ZERO, you have to program
it in. A rigorous explanation is beyond the scope of these
instructions, but, in short, this is done by "pumping" a charge
into an insulator. This is the job of the PROmGRAMER. The
wanted address and data are set up and relativly high voltage
is applied to the chip. This causes an insulator to temporarily
change into a conductor. A charge of electrons is injected
into the middle of the conductor, and, when the high voltage is
removed, these electrons now find that they are trapped,
surrounded by an almost perfect insulator. This "floating"
charge affects circuitry which "sees" a logical ZERO where such
charge exists.

     These EPROMs are programmed one byte at a time, i.e. eight
bits per programming pulse. The charge is as permanent as we
wish to make it (Well, not absolutly permanent. Most EPROM
manufactures will guarantee the charge retention for only(!)
ten years). We can change this insulator back into a
conductor, however, by shining a short-wavelength ultra-violet
lamp through the little window on the EPROM. This causes the
insulator to allow the charges to "leak" rapidly, thus erasing
the EPROM. This usually take about 20 minutes. Now, unless
you know what you are doing, please read the rest of these
instructions before you try programming your first EPROM.

SETTING THE SWITCHES

     On the PROmGRAMER is a set of ten switches. All switches
should be in the down position (OFF) except for those shown by
the white square above the switch. On the right edge of the
white squares are some numbers representing the different types
of EPROMs. This unit will program the single voltage supply
type of EPROM from the 2708 to the 27512. The number 27 does
not appear on the guide. For instance, to read or program a
27128, find the number 128 to the right of the guide. You will
notice that the squares above switch numbers 1, 4, 7, and 9 are
highlighted. These switches should be turned on (UP). NOTE:
It is especially important that switches 7 and 8 be in the
correct position for the type of EPROM used. These switches
control the level of programming voltage, and failure to have
them in the correct position may result in catastrophic failure
of the EPROM.

DIFFERENT EPROMS

     There are several EPROM types other than those listed on
the guide. It is important to remember that the PROmGRAMER can
only program the single voltage-type EPROMS. Some of the
earlier EPROM types need three voltages to operate. These
types should be considered obsolete, and the PROmGRAMER will
not program them. In particular, TI (Texas Instruments) had an
EPROM called a 2716 which uses three voltages. Their 2516 is
the equivilant of everyone elses 2716. The Intel type 2716,
which everyone else makes, can be programmed by the PROmGRAMER.

     Due to space limitations, we did not list all EPROM types
that you can use. If you have a single voltage 2708, use the
settings for the 2716. If you have a 2764A, use the settings
for the 2764, except switch 8 should be up, and switch 9 down.

A NOTE TO QUIKLOADER OWNERS

We realize how antiquated it would be to have to "boot" a disk, and type "BRUN whatever". To save you from this chore, we put a file on the disk called "quikloader". Just BLOAD it, and you will be able to "burn" a 2764 with the four PROmGRAMER programs. Naturally, you can take just the one that you need, and put it into an EPROM with other programs, but, with this ready-to-burn file, you can have your first program ready within 15 minutes.

JUST FOLLOW THESE EASY STEPS

1) Plug in a 2764 into the PROMGRAMER.

2) Install the card.

3) Type "BLOAD QUIKLOADER".

4) Type "BRUN EPB4" (for example).

5) Type "3" (EPROM type).

6) Type "B 1000 2FFF 0000"

7) Wait about 7 minutes.

8) After sucessful programming, in-
   stall the chip into your
   quikLoader.

9) You can catalog the chip in the
   usual way, and run the program by
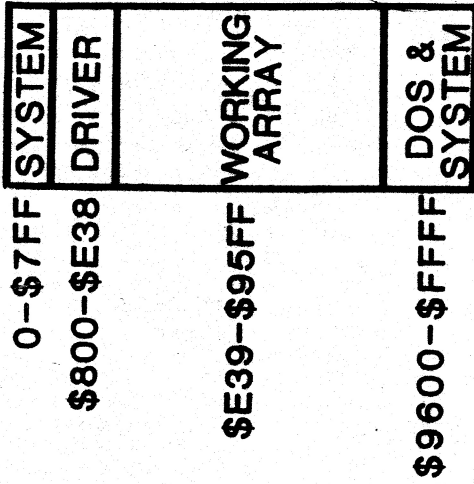   typing the index letter.
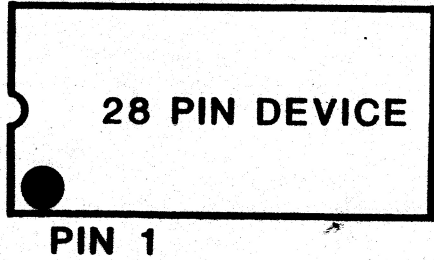
10) Any questions? We're here to
    help.

Address	Section
0-$7FF	SYSTEM
$800-$E38	DRIVER
$E39-$95FF	WORKING ARRAY
$9600-$FFFF	DOS & SYSTEM

FIGURE 2

28 PIN DEVICE

PIN 1

24 PIN DEVICE

PIN 1

FIGURE 1