PROBURNER™

USERS MANUAL

THOMPSON ELECTRONICS LTD.

7 JACKES AVENUE • SUITE 502 • TORONTO, ONTARIO • M4T 1E3 • (416) 960-1089

1074 KENSINGTON AVENUE • SUITE 188 • BUFFALO, N.Y. • 14215

PLEASE READ THIS BEFORE ATTEMPTING TO USE THE PROBURNER

WARNING

Incorrectly inserting an EPROM/ROM/EEPROM into the PROBURNER socket can cause PERMANENT DAMAGE to both PROBURNER and the memory chip. See INSERTING A MEMORY DEVICE.

IMPORTANT

The PROBURNER has a hardware startup timing device that interferes with COPY, ERASE CHECK, VERIFY and BURN-IN commands after power is first applied to the computer. ALWAYS WAIT FOR 25 SECONDS after turning on the computer before using these commands.

WARNING

A mistake in the selection of the chip type for burn-in can result in permanent damage to the chip.

WARNING

DO NOT WRITE TO ADDRESSES $D500 THROUGH $D5FF while a chip is in the PROBURNER socket. These addresses can enable the high voltage used to program EPROMS.

CONTENTS

1. CAPABILITIES

The PROBURNER is a general purpose EPROM programmer for the ATARI 400,800,600XL and 800XL computers. It handles all common EPROMS, a variety of single voltage supply EEPROMS and all the ATARI ROMS. Types include:

27 series EPROMS
- 2716 (INTEL, single supply, 2K by 8)

NOTE: The PROBURNER does not program the less common TMS2716 from Texas Instruments.
- 2732,2732A (4K by 8)
- 2764 (8K by 8)
- 27128 (16K by 8)

25 series EPROMS
- 2516 (2K by 8)
- 2532 (2K by 8)

52 series EEPROM (from SEEQ TECHNOLOGY)
- 52B13 (2K by 8)
- 52B23 (4K by 8)
- 52B33 (8K by 8)

X28 series EEPROM (from XICOR)
- X2816 (2K by 8)

ATARI ROMS — 2K, 4K, 8K types

The PROBURNER can copy all or part of a chip and put the contents anywhere in RAM.

It can check if an EPROM is properly erased. Then either a portion of or an entire chip can be programmed. The programming data can be put in any block of RAM.

There is a Verify function that will check a section of data in a chip against any block of RAM.

Data can be loaded into RAM from disk or cassette files for programming. Conversly it can be saved to disk or cassette files from RAM.

A machine language monitor is available for moving blocks of memory, displaying and changing memory and dumping memory contents to a Printer.

The PROBURNER has a RUN mode that allows up to 8K chips to be run at the $A000 to $BFFF address space in the socket.

2. USERS OVERVIEW

All programming and copying of chips is done directly from or to RAM. This means that you must choose where in RAM the chip is programmed from or copied to. If, for example, you want to duplicate a EPROM there are two ways you can proceed. First you could read the source EPROM into RAM, then save it on disk or cassette, turn off your computer to install the new unprogrammed EPROM, load the contents of the previously saved disk or cassette file into RAM and program the EPROM. Alternately you can use the STAND-ALONE capability. This method uses the fact that EPROMS,

EEPROMS and ROMS can be taken in and out of the PROBURNER while the power to the computer is still on. You must not remove or insert chips into the PROBURNER socket during a BURN-IN cycle, however. For the ATARI 400/800 you need to override the power interlock on the cartridge door. The duplication process is simple. First copy the source EPROM into RAM, remove it with the power still on, then insert the blank EPROM and program it from the same RAM block.

The monitor can be used to display or manipulate the data that has been or is to be programmed.

3. INSERTING A MEMORY DEVICE

3.1 A notch will be found in the center of one end of the EPROM/ROM/EEPROM device.

FOR 28 PIN DEVICES such as 2764 and 27128 insert the EPROM with the notched end toward the center of the PROBURNER.

FOR 24 PIN DEVICES such as the 2716, 2732, 2532, 52B13 and ROMS insert the device so that the notched end is toward the center and the device is plugged in as far away from the center as possible. This will leave 4 empty socket terminals closest to the center of the PROBURNER.

3.2 Removing the Memory Device.

Put a small screwdriver under the chip at its center and using lever action lift out the socket.

4. ERASE CHECK

Type E to check if all locations in an EPROM are erased.

4.1 Enter the EPROM type.

4.2 Make sure 25 seconds have elapsed since turning on the computer before proceeding.

4.3 If the EPROM is erased (this is the case when all data bits are ones) the response is "DONE". IF a number of locations are not erased then the total number of BITS (not bytes) that are not erased will be displayed.

5. READING AN EPROM/EEPROM

5.1 Insert the chip into the PROBURNER socket, as described above and put the PROBRUNER into the cartidge slot (use the left slot for the ATARI 800). Turn on the computer.

5.2 Type C to get copy mode and enter the device type.

5.3 Enter RAM start address. This is the start of the RAM addresses to which the device contents will be transfered. You must use a start address that is low enough that the contents of the chip can be stored above it. The free memory range given in brackets on the screen can be used as a guide.

5.4 Enter the chip start address. Each device has locations from 0 to its maximum capacity. Enter the address (in hex) of the starting location of the EPROM/ROM/EEPROM to be read from. This will be 0 if all the device contents are to be read. The acceptable range is given in brackets on the screen.

5.5 Make sure at least 25 seconds has elapsed since the computer was turned on before proceeding.

5.6 Enter the chip end address. This will be the last location in the device to be read. To read out all of the contents use $7FF, $FFF, $1FFF, $3FFF for 2K through 16K devices. The range limits are given in brackets on the screen.

5.7 Now you can display the chip data immediately or you can examine the data with the monitor. You might want to go to the disk operating system for storage, or store the data on tape.

NOTE: If you want to return to the menu in the middle of a copy cycle Press X then RETURN.

6. READING A ROM

Since all ATARI ROMS have the same basic configuration and differ only in their sizes the best way to read one is to assume that it is a 8K chip. Use the procedure above to read the chip into RAM with 0 and $1FFF as the chip start and end locations. When the chip has been read go to the monitor. Assuming that the chip data is in a RAM space starting at $4000 then you would display the following four sections of memory: $4000-4010, 4800-4810, 5000-5010 and 5800-5810. Notice if the data in these block is all $FFs or if two blocks are the same. From this analysis the size of the ROM can be determined. A 4K ROM, for example, might have no data ($FFs) from $4000 to $4FFF.

7. PROGRAMMING AN EPROM/EEPROM

7.1 Insert the chip into the PROBURNER as described above then turn on the computer.

7.2 Load your machine code into RAM from disk (See LOAD FILE FROM DISK), cassette (SEE GET A FILE FROM CASSETTE) etc.

7.3 Enter chip type. WARNING! A mistake here could permanently damage the chip.

7.4 Enter the RAM start address (in hex). This is the start address of your data in RAM.

7.5 Enter the chip start location. This is the lowest location in the chip that the data in RAM will be programmed to. Use 0 if you want to program the entire device. The range limits are given in brackets on the screen.

7.6 Make sure that 25 seconds has elapsed since you first turned on the computer before proceeding.

7.7 Enter the chip end location. This is the last location in the EPROM/EEPROM to be programmed. Use $7FF, $FFF, $1FFF, $3FFF for 2K through 16K devices if the entire devices is to be programmed. These are shown in brackets on the screen.

7.8 When the timer in the right hand corner of the screen counts down to zero programming is complete.

For the 52B13 EEPROM the screen will go blank for about forty seconds while programming is taking place.

NOTE: To get back to the menu from the Programming mode type X then RETURN or use the BREAK key.

8. VERIFY CHIP

Type V for the verify function.

8.1 Enter chip type.

8.2 Enter RAM start address. This first address of the block of data that will be compared with the data in the chip.

8.3 Enter the chip start location. This is the lowest location in the chip that the data in RAM will be programmed to. Use 0 if you want to verify the entire device. The range limits are given in brackets on the screen.

8.4 Make sure 25 seconds has elapsed since the computer was turned on before proceeding.

8.5 Enter the chip end location. This is the last location in the chip to be verified. Use $7FF, $FFF, $1FFF, $3FFF for 2K through 16K devices if the entire device is to be verified.

8.6 If all the locations to be checked have the same data as the RAM the response will be "DONE". All differences between the chip and RAM will be listed sequentially.

9. MACHINE LANGUAGE MONITOR

The MONITOR is entered by typing M. An asterisk will be displayed as a prompt after each monitor command.

List of functions:

9.1 MOVE MEMORY BLOCK

To move a block of memory aaaa through bbbb to cccc through dddd type:

Mcccc<aaaa,bbbb

For example, to move the operating system from where it resides at $E000 through $FFFF to $4000 through $5FFF type:

M4000<E000,FFFF

9.2 DISPLAY MEMORY LOCATION(S)

To display the data at a single location type Dxxxx where x is the address in hexadecimal. For multiple locations type Dxxxx,yyyy where xxxx is the first address and yyyy is the last. To read out locations $203 through $768, for example, type:

D203,768

The screen displays data in groups of eight. From the left there is the address, next comes a byte of data (hex) for that address, then the data of the next address and so on for 6 more bytes of data. Since this command displays a group of eight bytes the data for the first address you specify might be in the middle of the line. This first address data will be displayed in inverted video so that it can be easily picked out. Similarly when a single location is to be displayed it will be highlighted in this way. On the right of the screen is the ASCII equivalent of each byte of data. Non readable characters are replaced with a decimal point.

9.3 PRINT MEMORY LOCATIONS

This command is essentially the same as the Display Memory Locations command except that the printer prints what is displayed on the screen. See Display Memory Location for details but substitute P for D.

9.4 CHANGING MEMORY LOCATIONS

The data at any RAM location xxxx (represents any address) can be set to dd (represents any data value) by typing:

Cxxxx<dd

For changing consecutive locations use the form:

Cxxx<dd,dd,dd

Example: to put in data values of 24,F5,86 at addresses 2C5, 2C6 and 2C7 type:

C2C5<24,F5,86

NOTE: When displaying a series of memory locations use the CTRL E to stop the display and the BREAK key to return to the monitor. When you want to get back to the menu type X then RETURN.

10. DISK FUNCTIONS

To get to the disk operating system type T.

10.1 SAVE FILE TO DISK

To create a file from a block of RAM type S and follow the instructions on the screen or read USING DOS below.

10.2 LOAD FILE FROM DISK

To load a file from disk to RAM type L and follow the instructions on the screen or those given below.

10.3 USING DOS

When under control of the DOS you are able to save RAM data in a file with the K command or load a EPROM/ROM/EEPROM file to RAM with the L command.

To return to PROBURNER control use the M (Run from address command) at address $8000.

11. CASSETTE FUNCTIONS

11.1 PUT A FILE ON CASSETTE

To save a block of RAM using the PROGRAM RECORDER type P.

11.1.1 Enter the save address. This is the first address on the block of RAM to be saved.

11.1.2 Enter the length of the block of RAM in kilobytes. For the 2716,2532,2732,2764 and 27128 enter 2,4,4,8,16 respectively.

11.1.3 There will be two beeps from the computer. These are a reminder to Press both the RECORD and PLAY button on the recorder. After this is done type RETURN. The recorder will run until all the data is stored then the computer will return to the menu.

11.2 GET A FILE FROM CASSETTE

11.2.1 To get a file from the program recorder and put it in RAM type G.

11.2.2 Enter the load address. This is the address that you want the file to start at in RAM.

11.2.3 There will be a single beep from the computer at this point. This is a reminder to Press the PLAY button on the recorder. After this type RETURN. The recorder will run until all the data has been transfered to RAM then the computer will return to the menu.

12. RUN MODE

This mode allows up to 8K EPROM/EEPROM to be run as though it were part of the computer's memory. The run mode is latched in the PROBURNER hardware once it is set. The address space to which the socket is converted is $A000 to $BFFF. This is the same address space as the 8K cartridges used by ATARI computers. For machine language programs to run in this mode they must be assembled for this address space.

12.1 To enable the run mode type R then the start address of the program (it must be $A000 through $BFFF).

For the 52B13 EEPROM and the 2716 EPROM always RUN from an address in the range $A800 through $AFFF and $B800 through $BFFF. If these start addresses are not used for the 52B13 its memory could be altered.

For the 4K and 8K chips any start address in the $A000 through $BFFF range may be used.

Only 27 series EPROMs and 52 and X28 series EEPROM can be run from the PROBURNER socket.

You may also use the RUN mode to go to a Program written in RAM. Using it this way makes it similar to a run-from-address monitor command.

MANUAL REVISIONS

1. You can ignore the warning in the manual to wait 25 seconds after turning on the computer. The PROBURNER now displays "WAIT" during this time.

2. The PROBURNER can be used in the XE models.

3. When programming an E/EEPROM the PROBURNER will automatically perform an erase check before programming and a verify after programming. The erase check will display the number of unerased bits in the chip if any exsist. The verify will list sequentially all differences between the chip and RAM.

4. Do not attempt to program INTEL type 2764A and 27128A. These chips use a lower programming voltage than the standard 2764 and 27128.