

```

                dd
                dd
          aaaaaa      dd      ssssss
                aa      dddddddd  ss
          aaaaaaaaa  dd      dd      ssssss
aa      aa      dd      dd      ss
          aaaaaaaaa  dddddddd  ssssssss

```

```

PPPP RRRR OOOOO M M W W W RRRR IIIII TTTT EEEEE RRRR
P P R R O O MM MM W W W R R I T E R R
PPPP RRRR O O M M M W W W RRRR I T EEEE RRRR
P R R O O M M W W W R R I T E R R
P R R OOOOO M M W W R R IIIII T EEEEE R R

```

USER'S MANUAL
 =====
 version 4.0
 2/23/84

Ackerman Digital Systems, Inc.
 216 W. Stone Ct.
 Villa Park, Illinois 60181
 (312) 530 - 8992

ADS INC. reserves the right to make changes to any products herein to improve reliability, function or design. Although the information in this document has been carefully reviewed and is believed to be accurate, ADS does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent or copyrights nor the rights of others.

Copyright (c) 1984 by ads, inc. ALL RIGHTS RESERVED

TABLE OF CONTENTS

I.	Introduction.....	1
II.	Operation.....	1
III.	Commands.....	2
	a. P - Program eprom.....	2
	b. U - Unprogram eeprom.....	3
	c. V - Verify eprom.....	3
	d. R - Read eprom.....	4
	e. T - Type of eprom.....	4
	f. D - Display memory.....	5
	g. B - Base address change.....	5
	h. O - Offset for memory.....	5
	i. L - Load .HEX file.....	5
	j. W - Write .HEX file.....	7
	k. Q - Query file directory.....	7
	l. C - Check eprom for unprogrammed.....	7
	m. E - Exit to CP/M.....	8
	n. F - Fill memory.....	8
	o. S - Sum memory.....	8
	p. H - Help command list.....	8
	q. X - Exercise PROMBLASTER II.....	9
	r. " - Ditto (repeat) last command.....	9
	s. M - Mode of Program, Read & Verify...	9
IV.	Error Messages.....	10
V.	Notes on earlier versions.....	11

IMPORTANT: AT NO TIME SHOULD AN EPROM BE PLACED INTO THE PROMBLASTER WITHOUT FIRST HAVING SPECIFIED THE TYPE TO THE PROMWRITER. FAILURE TO DO SO MAY RESULT IN DAMAGE TO THE EPROM AND/OR PROMBLASTER. ALSO NEVER RESET THE COMPUTER WITH AN EPROM IN THE PROMBLASTER.

I. Introduction

The ads PROMWRITER software is a package written in 8080 assembly language for execution under CP/M. It provides the capability of programming 26 different EPROMS, 1K, 2K, 4K, 8K 16K and 32K parts, both 24 and 28 pin packages when used with the ads PROMBLASTER II. Commands are provided to read, program and verify EPROMS with optional offsets. Memory load, examine, display and sum functions as well as CP/M hex file load and save operations are provided. Fast programming algorithms are used for some parts above 8K in size. A Mode command allows 16-bit memory data to be programmed into even and odd proms and a repeat last command function has been added. Utilizing the built-in test circuitry of the PB-II an Exercise command allows the PROMWRITER to perform a limited functional-test of the PROMBLASTER II board.

II. Operation

The ads PROMWRITER is a transient program that operates under CP/M. It is invoked by typing its name, PROMxx, on the command line (xx=version). When started the PROMWRITER prints a signon message, a command list, and the current part type then issues a prompt (:). All commands to the PROMWRITER consist of a command character followed by zero to three arguments. Type an H then a carriage return. An abbreviated list of commands should be displayed. In this list, X, Y, and Z represent hexadecimal command arguments, the word TYPE represents a seven character string identifying an EPROM type, and the F represents a CP/M file name.

If your PROMBLASTER II is configured for a device address other than C0H-C3H, the first command you must use is the Base command. This configures the PROMWRITER for whatever PROMBLASTER device address you have selected. The next command you should enter is a Type command. This allows the PROMWRITER to setup the PROMBLASTER for the correct standby voltages for whatever type of EPROM you will be programming. Note that the default values for these two commands can also be changed by patching and re-saving the PROMWRITER. This is explained under the description of each command.

IMPORTANT: AT NO TIME SHOULD AN EPROM BE PLACED INTO THE PROMBLASTER WITHOUT FIRST HAVING SPECIFIED THE TYPE TO THE PROMWRITER. FAILURE TO DO SO MAY RESULT IN DAMAGE TO THE EPROM AND/OR PROMBLASTER. ALSO NEVER RESET THE COMPUTER WITH AN EPROM IN THE PROMBLASTER.

After specifying the EPROM type, any of the other commands may be used. To program an EPROM, place the unprogrammed part in the programming socket U8, or in the programming socket of the PROM EXTENDER accessory. MAKE SURE TO USE ONLY THE LOWER 24 PINS FOR 24 PIN EPROMS. Now using the Check command verify that the part is indeed unprogrammed. If the data you wish to program is

already in memory, then give the Program command specifying the address range and EPROM offset. If the data is not in memory a Load command may be used to read it off of the disk. To copy an EPROM, place the programmed part into the socket U8 then use the Read command to copy the data to an area of memory. Now place an unprogrammed part into U8, Check it, then Program it. Any errors detected after programming will be printed on the console.

The failure of an EPROM to program may be caused by many problems. An EPROM that is not fully erased will not program ones. The Check command should always be used before programming any part to verify that it is totally erased. Any pattern of addresses or bits that fail to program is a sign of a bad EPROM. Since the Verify command is automatically performed after a Program command, any attempts to program from an area of memory that is changing will cause apparent programming errors. Specifying the wrong EPROM type to the PROMWRITER can be catastrophic. Damage to the EPROM and/or PROMBLASTER II may result. BE CAREFUL !

III. Commands

All commands to the ads PROMWRITER are given in upper or lower case. Any arguments required follow on the same line as the command. A leading space before the first argument is optional. Input is via the CP/M read line function so any of the line editing functions may be utilized before the carriage return. All numeric arguments required are in hexadecimal. The Read, Load and Fill commands will not overwrite the PROMWRITER or CP/M. All numeric arguments are checked against the size of the EPROM being programmed. Note that if the PROMWRITER is operating in even or odd mode that the apparent size of the EPROM is doubled since only even or odd memory locations are being referenced.

IMPORTANT: AT NO TIME SHOULD AN EPROM BE PLACED INTO THE PROMBLASTER WITHOUT FIRST HAVING SPECIFIED THE TYPE TO THE PROMWRITER. FAILURE TO DO SO MAY RESULT IN DAMAGE TO THE EPROM AND/OR PROMBLASTER. ALSO NEVER RESET THE COMPUTER WITH AN EPROM IN THE PROMBLASTER.

a. **P** - Program eprom

This command is used to program a range of memory into a previously specified type of EPROM. Three arguments are required; a starting memory address, an ending memory address, and an EPROM address offset. Both memory addresses may also be offset with the Offset command. The EPROM is programmed one location at a time starting with the EPROM offset location, and continuing through the memory address range given. When the PROMWRITER is operating in even or odd mode memory addresses are incremented by two and forced to be even or odd. For most EPROMS a single 50 millisecond long programming pulse is applied to each location. Therefore the time to program is approximately $t = (N$

x .05) seconds where N is the number of locations being programmed. For an 8K EPROM, $t = (8192 \times .05) = 409.6$ seconds = 6.83 minutes. Some 8K EPROMS require only a 2 millisecond programming pulse with multiple passes. Many of the newer 8K and larger EPROMS can utilize a Fast-Programming algorithm which interacts with the part being programmed. Basically a series of 1 millisecond pulses are applied (up to a limit) and a verify is attempted after each one. When the location verifies a longer over-program pulse is applied whose width is determined by the number of program pulses that were issued. The resulting savings in programming time is impressive. An 8K EPROM that used to take 6.8 minutes now typically programs in 1.5 minutes. A 32K EPROM typically programs in 2.75 minutes. Only typical program times are given as the actual time is part-dependent. At the end of programming, a control-g (bell) is sent to the console and a Verify command is attempted. Any discrepancies are reported on the console. The command format is:

P x,y,z Program from x to y offset z

where: x = starting memory address
 y = ending memory address
 z = prom offset

b. **U** - Unprogram eeprom

This command is used to unprogram (erase) a previously programmed EEPROM. The erase time varies for the different EEPROMS but it is usually under one second. A verify of the EEPROM's erasure is automatically performed after an unprogram is attempted. Any locations that fail to show erasure are reported. The format of the command is:

U Unprogram EEPROM

c. **V** - Verify eeprom

This command verifies the contents of an EPROM against memory. It is called automatically at the end of a Program command. The EPROM locations are compared one a time starting with the offset address and ranging from the starting memory address to the ending memory address. The memory addresses may also be offset via the Offset command. When the PROMWRITER is operating in the even or odd mode memory addresses are incremented by two and are forced even or odd. Any discrepancies between the EPROM and the memory contents are displayed on the console. The format of this command is:

V x,y,z Verify from x to y offset z

where: x = starting memory address
 y = ending memory address
 z = prom offset

d. **R** - Read eprom

This command reads the contents of an EPROM into memory. The contents of the EPROM locations are transferred into memory one at a time starting with the offset address and ranging from the starting memory address to the ending memory address. It is important to remember that the memory addresses may also be offset by a previous use of the Offset command. When the PROMWRITER is in the even or odd mode memory addresses are incremented by two and forced even or odd. The Read command will not allow the PROMWRITER or CP/M to be overwritten. The command format is:

R x,y,z Read from x to y offset z

where: x = starting memory address
 y = ending memory address
 z = prom offset

e. **T** - Type of eprom

This command determines the type of EPROM being programmed. Any commands which are EPROM dependent; i.e. Program, Verify, Read, and Check all require that the EPROM type is correctly specified before their use. A default part TYPE may be specified by changing the seven bytes at 105H to the Ascii representation of the desired part type. An attempt to enter a part TYPE not shown in the EPROM TYPES list will give an 'Unknown TYPE of EPROM' error message before an eprom type has been specified. The type is specified as a seven character string. The command format is:

T type Type of eprom

where: typestr = eprom type string

The following table lists the EPROM types that the PROMWRITER version 4.0 will program:

	TYPE	DESCRIPTION	SIZE	# OF PINS	FAST	SUPPLIES
1K	TMS2508	- Texas Inst	2508	- 1K x 8	- 24 pins	-
	INT2716	- Intel	2716	- 2K x 8	- 24 pins	-
	TMS2516	- Texas Inst	2516	- 2K x 8	- 24 pins	-
2K	MCM2716	- Motorola	2716	- 2K x 8	- 24 pins	-
	HN42716	- Hitachi	2716	- 2K x 8	- 24 pins	-
	INT2732	- Intel	2732	- 4K x 8	- 24 pins	-
	IN2732A	- Intel	2732A	- 4K x 8	- 24 pins	- Vpp=21v
	HN42732	- Hitachi	2732	- 4K x 8	- 24 pins	-
4K	TMS2532	- Texas Inst	2532	- 4K x 8	- 24 pins	-
	MCM2532	- Motorola	2532	- 4K x 8	- 24 pins	-
	HN42532	- Hitachi	2532	- 4K x 8	- 24 pins	-
	INT2764	- Intel	2764	- 8K x 8	- 28 pins	- F - Vpp=21v
	IN2764A	- Intel	2764	- 8K x 8	- 28 pins	- F -Vpp=12.5v
8K	INT2763	- Intel	2764	- 8K x 8	- 28 pins	- F -
	HN42764	- Hitachi	2764	- 8K x 8	- 28 pins	-
	TMS2564	- Texas Inst	2564	- 8K x 8	- 28 pins	-
	MC68764	- Motorola	68764	- 8K x 8	- 24 pins	-
16K	IN27128	- Intel	27128	-16K x 8	- 28 pins	- F - Vpp=21v
32K	IN27256	- Intel	27256	-32K x 8	- 28 pins	- F -Vpp=12.5v
EE	HN48016	- Hitachi	48016	- 2K x 8	- 24 pins	-
PROMS	MCM2816	- Motorola	2816	- 2K x 8	- 24 pins	-
	INT8748	- Intel	8748	- 1K x 8	- 40 pins	- adapter
SPECIAL	INT8749	- Intel	8749	- 2K x 8	- 40 pins	- reqd
	INT8755	- Intel	8755	- 2K x 8	- 40 pins	-
	INT8048	- Intel	8048	- 1K x 8	- 40 pins	- read
	INT8049	- Intel	8049	- 1K x 8	- 40 pins	- only

Note that pin compatible parts may be programmed by specifying their equivalent type; i.e. Fujitsu MBM2716 may be programmed by using INT2716 type command.

IMPORTANT: AT NO TIME SHOULD AN EPROM BE PLACED INTO THE PROMBLASTER WITHOUT FIRST HAVING SPECIFIED THE TYPE TO THE PROMWRITER. FAILURE TO DO SO MAY RESULT IN DAMAGE TO THE EPROM AND/OR PROMBLASTER. ALSO NEVER RESET THE COMPUTER WITH AN EPROM IN THE PROMBLASTER.

f. **D** - Display memory

 This command allows the user to examine a block of memory in hexadecimal. Two arguments are required; a starting memory

address, and an ending memory address. Both memory addresses may include offsets provided by the Offset command. Data is displayed 16 bytes per line in hexadecimal with the ASCII equivalents trailing the bytes on the same line. The display may be suspended with the CP/M suspend output character (control-S) or stopped with any other character. The command format is:

D x,y Display memory from x to y

where: x = starting memory address
 y = ending memory address

g. **B** - Base address

This command allows the user to change the I/O addresses that the PROMWRITER uses when it controls the PROMBLASTER. The default addresses used are C0H-C3H. The command format is:

B x Base addresses are now x to x+3

where: x = starting I/O address in hex

The user may wish to patch the PROMWRITER to have it always default to a different I/O address. Location 103H contains the default PROMBLASTER base address that the PROMWRITER uses. By changing this location with DDT and then saving the patched version with the CP/M SAVE command the user can permanently change the default device addresses that the PROMWRITER uses.

h. **O** - Offset command

All memory addresses used by the PROMWRITER may use an optional offset specified by this command. This allows the user to setup a base address of a data area and then use relative addresses for all the commands. This offset should not be confused with the EPROM address offset. The default offset is 0000H. Any offset remains in effect until it is changed. Note that the Load hex file command may also specify an offset which has the same effect as using the Offset command. The command format is:

O Offset examine
0000 0000 was old offset

O 1234 Offset change, 1234 is user's offset
1234 1234 is new offset

i. **L** - Load hex file

The PROMWRITER provides a means of loading INTEL hex format files from disk by name. This allows the output of an assembler to be programmed directly into an EPROM. The file is loaded at

the address specified in the hex format with whatever offset is in effect at the time. The PROMWRITER will not allow itself or CP/M to be overwritten. A new offset may be specified on the command line that will take effect for all subsequent commands. The command format is:

L filename,x Load filename .HEX with offset x

where: filename is CP/M .HEX file
 x is optional hexadecimal offset

j. **W** - Write hex file

The PROMWRITER provides a means of writing INTEL hex format files from memory to disk by name. This allows the user to save EPROM contents on disk. The data is written to the file from the address specified with whatever offset is in effect at the time to the ending address specified in hex format. If the file specified exists the user is given the option of overwriting the old file. The command format is:

W filename,x,y Write hex from x to y to filename.HEX

where: filename is CP/M .HEX file
 x is starting address (memory offset)
 y is ending address (memory offset)

k. **Q** - Query .HEX files

This command allows the user to examine the CP/M file directory on any disk for all .HEX files without having to exit the PROMWRITER. The names are printed 5 per line on the console. The command format is:

Q d Query .HEX files on drive d

where: d = CP/M drive specifier

l. **C** - Check for unprogrammed

Since EPROMS are only programmed by the addition of zeroes, not ones, it is necessary to verify that the EPROM is unprogrammed in a given range before an attempt is made to program it. This command checks a range of locations starting at a given offset in the EPROM for the unprogrammed state, and reports any locations that appear to already have been programmed. The command format is:

C x,y Check eprom for unprogrammed

where: x = number of locations to check
 y = offset from start of eprom

m. **E** - Exit to CP/M

This command performs an orderly return to CP/M via the warm-boot entry point. The PROMBLASTER is not affected by this command however it is advisable to remove any EPROMS from the PROMBLASTER first. The command format is:

E Exit to CP/M

n. **F** - Fill memory

This command allows the user to fill a range of memory with a constant. Three arguments are required; the starting memory address, the ending memory address, and the desired fill constant. Both memory addresses may be offset with the Offset command. All locations from the starting address through the ending address will be initialized to the specified constant. The PROMWRITER will not allow itself or CP/M to be overwritten. No attempt is made to verify that the constant stored correctly. Note that if both starting and ending addresses are the same only one location will be initialized. The command format is:

F x,y,z Fill memory from x to y with z

where: x = starting memory address
 y = ending memory address
 z = hexadecimal value

o. **S** - Sum command

This command computes a checksum value for a given range of memory. The checksum is given in two forms; a zero sum which is a value that when added to the sum of the range specified will give a zero result, and a ones sum which when added to the sum of the range specified will give an all ones result. The command format is:

S x,y Sum from x to y memory offset

where: x = starting memory address
 y = ending memory address

p. **H** - Help command

This command provides a brief summary of all PROMWRITER commands on the console as well as a list of all valid EPROM types. The command format is:

H Help with commands

This results in the following display of the PROMWRITER commands and EPROM types:

***** COMMANDS *****

V X,Y,Z - VERIFY X to Y Prom Offset Z	R X,Y,Z - READ X to Y Prom Offset Z
D X,Y - DISPLAY memory X to Y Offset	B Z - BASE PROMBLASTER Port Z
O X - Set and show memory OFFSET X	L F,X - LOAD File (.HEX) Offset X
W F,X,Y - WRITE File (.HEX) from X to Y	Q D - Directory of files (.HEX)
C X,Z - CHECK X bytes in Prom Offset Z	E - EXIT to CP/M
F X,Y,Z - FILL memory from X to Y with Z	S X,Y - SUM memory X to Y offset
H - HELP produces this list	X - Exercise PROMBLASTER
T TYPE - Set EPROM type to TYPE	" - Ditto (repeat) Command
M - MODE of Program, Read & Verify	P X,Y,Z - PROGRAM X to Y Prom Z

***** EPROM TYPES *****

TMS2508 INT2716 MCM2716 HN42716 TMS2516 TMS2532 MCM2532 HN42532 HN42732
 INT2732 IN2732A HN42764 INT2764 IN2764A INT2763 TMS2564 MC68764 HN48016
 MCM2816 IN27128 IN27256 INT8755 INT8748 INT8749 INT8048 INT8049

q. **X** - Exercise PROMBLASTER command

This command utilizes the built-in test circuitry on the PROMBLASTER II to attempt to verify correct operation. The user is prompted to remove any parts from the programming socket and then the Vcc and Vpp circuitry on the board is exercised and monitored in all four PROMBLASTER modes. The L.E.D. on the board (or on the PROM EXTENDER accessory) is on for each test for a total of four flashes. Any errors in detected versus programmed voltages are reported. A PROMBLASTER II that fails the Exercise command should not be used to program any EPROMs without an examination of why it failed. The format of this command is:

X Exercise PROMBLASTER II

r. **"** - Ditto command

This command repeats the last entered command. The previously entered command line is redisplayed then executed. The Ditto command will not repeat itself. The format of this command is:

" Repeat last command

s. **M** - Mode command

This command allows the PROMWRITER to Program, Read & Verify only the even, odd or all bytes into EPROMs. When in even or odd mode addresses given to the aforementioned commands are forced to be even or odd (by anding with 0FFFEH (even) & oring with 0001H (odd)) and are incremented by two during the range of the command. Also the effective EPROM size is doubled in even or odd

modes since only half of the bytes in the address range specified are being programmed, read or verified. The output of cross-assemblers for 16-bit machines can be loaded then the even and odd bytes burned into separate EPROMs for 16-bit wide system usage. Or 16-bit EPROMs can be read into even and odd bytes successively and then written to disk as a .HEX file. The format of this command is:

M Change PROMWRITER Mode

Select MODE of Program, Read and Verify commands.

All, Even or Odd bytes (A/E/O) :

IV. Error Messages

Selected Device is not an EEPROM. Unprogram IMPOSSIBLE !

This is sent in response to a unprogram command. The Unprogram command is invalid for non-EEPROM device types.

Unknown TYPE of EPROM

This is sent in response to a type command. The PROMWRITER doesn't recognize the type string as a valid EPROM type.

Syntax ERROR

This is sent in response to any command requiring arguments. The PROMWRITER requires more arguments than were supplied on the command line.

The SIZE of this command is IMPOSSIBLE with this EPROM TYPE

This is sent in response to any command that implies an EPROM size. The PROMWRITER computes a value from the starting and ending addresses and the prom offset which must be less than or equal to the size of the part being programmed. When in even or odd mode the apparent size of the EPROM type specified is doubled since only every other memory location is read or written to.

The ADDRESS of this command would OVERWRITE the PROMWRITER or CP/M

This is sent in response to any command that will load memory. The PROMWRITER computes a value from the starting and ending addresses and the start of CP/M and the end of the PROMWRITER. Overwrites of the PROMWRITER or CP/M are not allowed.

File ERROR

This is sent in response to a hex file load command. The

PROMWRITER detected a checksum error when loading an INTEL hex format file.

File NOT FOUND

This is sent in response to a hex file load command. The PROMWRITER couldn't find a file named on the command line.

The DIRECTORY is FULL

This is sent in response to a hex file write command. The PROMWRITER was unable to create the file named on the command line because the directory was full.

The DISK is FULL

This is sent in response to a hex file write command. The PROMWRITER was unable to finish writing the hex file named on the command line because the disk is now full.

?

This is sent in response to an invalid command.

OFST PR MM OFST PR MM OFST PR MM OFST PR MM OFST PR MM

This is sent in response to a Verify or Check command. The PROMWRITER found an error between the eprom(PR) and memory(MM) at the eprom offset(OFST). For a Check command the value of MM is FF or 00 hex.

x<-is NOT a VALID HEX digit

This is sent in response to any command requiring hex input. The PROMWRITER found the invalid hex character 'x' in the hex arguments of the command line.

V. Notes on earlier versions

Version 1.0

First release version.

Version 1.1

Fix erroneous SIZE ERROR on CHECK command with a prior memory offset specified.

Version 1.2

Fix LOAD command operation.

ads **PROMWRITER** User's Manual

Version 1.3

Fix LOAD command handling of drive specifiers.

Version 2.0

Add EEPROM capability. Add UNPROGRAM command. Add capability to interrupt long output messages on PROGRAM, VERIFY, CHECK and DISPLAY commands. Compressed by about 1/2 Kbytes over V1.3.

Version 3.0

Add lower VPP control for REV 2 PROMBLASTER and newer EPROMS. Improve three supply programming algorithm. Changed MC68764 programming algorithm. Add WRITE hex file command. Improved DISPLAY command output format.

Version 3.1

Fixed error in WRITE command.

Version 3.1X

Allows modified PROMBLASTER to program 27128's.

Version 3.2

Fixed error in handling of extent number in .HEX file FCBS for the Query, Load and Write commands for files over 6K bytes (16K characters) in size.

Version 4.0

Major changes for PROMBLASTER II. Added Ditto command, Exercise command and Mode command. Added many new part types and fast programming algorithm. Removed three-supply parts.