



**E/EPROM  
PROGRAMMER CARD  
LEAP-EP1**

Written on 1994.11



**COPYRIGHT (C) 1991  
LEAP ELECTRONIC CO., LTD.**

LEAP assumes no responsibility for any errors which may appear in this manual nor does it make a commitment in updating information contained herein.

LEAP reserves the right to make changes to this manual at any time and without giving prior notice.

Information stated in this manual is considered property of LEAP and may not in any way be duplicated or copied.

LEAP is a trademark of LEAP Electronic Co., Ltd.  
LEAP-EP1 is a trademark of LEAP Electronic Co., Ltd.  
IBM PC/XT/AT/386 is a registered trademark of International Business Machine.  
MS-DOS is a registered trademark of Micro Soft Corporation.

## CONTENTS

	PAGE
<b>I. INTRODUCTION</b> .....	1
A) LEAP-EP1 Overview .....	1
B) LEAP-EP1 Components .....	2
<b>II. HOW TO START LEAP-EP1</b> .....	4
A) Important Reminders Before Starting LEAP-EP1 ...	4
B) Installation Procedures .....	5
C) Product Outline .....	7
D) Device Installation .....	7
E) I/O Addressing Conflicts .....	8
1) I/O Port Address .....	8
2) I/O Wait Select .....	9
<b>III. STARTING WITH EPROM/EEPROM</b> .....	10
A) Applying the Functions .....	10
1) Type .....	10
2) Read Device to Memory .....	13
3) Check Device Blank .....	17
4) Programming .....	18
5) Verify Device .....	21
6) Disk .....	23
7) Help .....	29
8) Process .....	31
9) Parameter .....	36

# I. INTRODUCTION

---

## A) LEAP-EP1 OVERVIEW

Thank you for utilizing LEAP E/EPROM Programmer card. With LEAP-EP1 programming tool, we have accomplished best quality combined with user upgradability and affordable price.

This LEAP-EP1 is an excellent programmer with a 32 pin socket, which presents users with an easy-to-understand and simple-to-use working environment. No ambiguous commands nor complicated procedures are required to be memorized, just follow the simple functions written in the window driven software and we assure you will enjoy the best programming time.

LEAP-EP1 has the capability of handling EPROM (Electrically Erasable Programmable Read Only Memories) and EEPROM (Electrically Erasable Programmable Read Only Memories) ranging from 2716 to 2 Mg IC or more. New devices are easily updated in software and without modifying the hardware.

In order to help us and help you achieved the fastest after sales service, be sure to write a detailed description of the error encountered and mail back the unit at the earliest possible date.

## B) LEAP-EP1 COMPONENTS

LEAP-EP1 package consists of the following:

### 1) An IBM Compatible Plug-in Board

LEAP-EP1 plug-in board contains all the hardware logic that is required to interface between the IBM compatible PC/XT/AT computer and the device socket. The LEAP board produces all the necessary voltages that are required to program the supported logic devices.

It is all right for the board to be permanently installed in the user's IBM compatible PC/AT/XT computer, it does not interfere with normal operation.

### 2) Floppy Disk Based Software

The standard LEAP package contain one floppy disk with all the program and utilities to enable you to use LEAP-EP1. The program are all menu driven and easy to use. The Programs on the disk directly control the hardware logic on the LEAP-EP1 plug-in board and socket pod to produce the correct programming voltages for selected device.

### 3) An External Device Socket Module

Using the socket to provide connection to the programmable device. This box is called 'Module or POD', it supports devices in DIP packages with up to 32 pins.

- 4) A 1 Meter Long Cable
- 5) This manual.

## **II. HOW TO START LEAP-EP1**

---

### **A) IMPORTANT REMINDERS BEFORE STARTING LEAP-EP1**

- 1) The "repeat key" located at the right hand side of the textool socket was purposely designed to allow users to repeat reading, programming, verifying a device.

Take for example, after accomplishing programming a certain device, user can just make use of the "repeat key" and repeat the same procedure.

- 2) To prevent damaging the IC, always make sure to remove whatever device in textool socket when starting or ending the system.
- 3) Always make sure that the textool socket is free from dirt, otherwise when users execute read or program commands, it will provide unstable results.
- 4) If IC pin has become dirty because of prolong use, make sure to clean the device before application.

200-207	Game I/O
278-27F	Parallel printer port 2
2F8-2FF	Serial port 2
300-31F	Prototype card
360-36F	Reserved
378-37F	Parallel printer port 1
380-38F	SDLC, bisynchronous 2
3A0-3AF	Bisynchronous 1
3B0-3BF	Monochrome display and printer adapter
3C0-3CF	Reserved
3D0-3DF	Color/graphics monitor adapter
3F0-3F7	Diskette controller
3F8-3FF	Serial port 1

This is current I/O port address of any personal computer, so again we suggest users to select the 200-2FF section for the 300-3FF sections are usually occupied by a lot of other add-on cards.

## B) INSTALLATION PROCEDURES

All the instruction listed in the installation section should be followed carefully.

- 1) Make sure to switch off your computer system.
- 2) Check the JUMP of the computer system remote control card.
- 3) Insert the control board carefully into the slot and fasten it to the computer system with the slot cover screw.

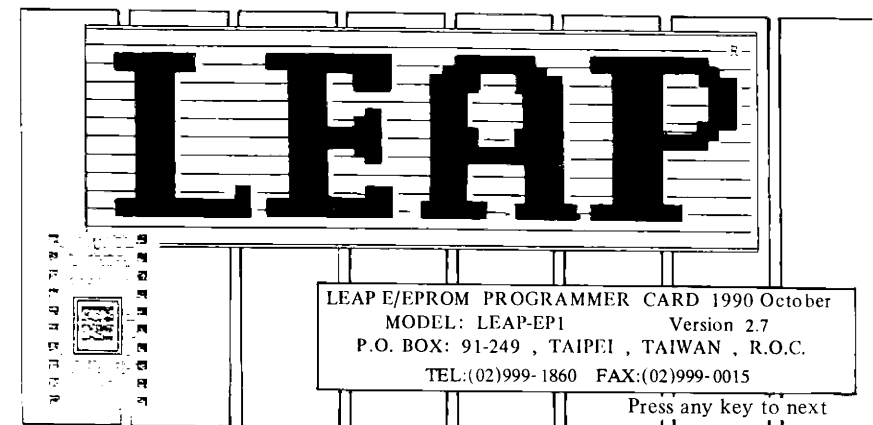
- 4) Finally, connect the socket 'MODULE OR POD' to the board by using the attached cable. The male cable end should be connected to the plug-in board and the female end to the main module.
- 5) After booting the system with DOS in drive A. Insert LEAP-EP1 software in drive A.

Enter the command: A>EP1 [Enter] — (Under 2M)

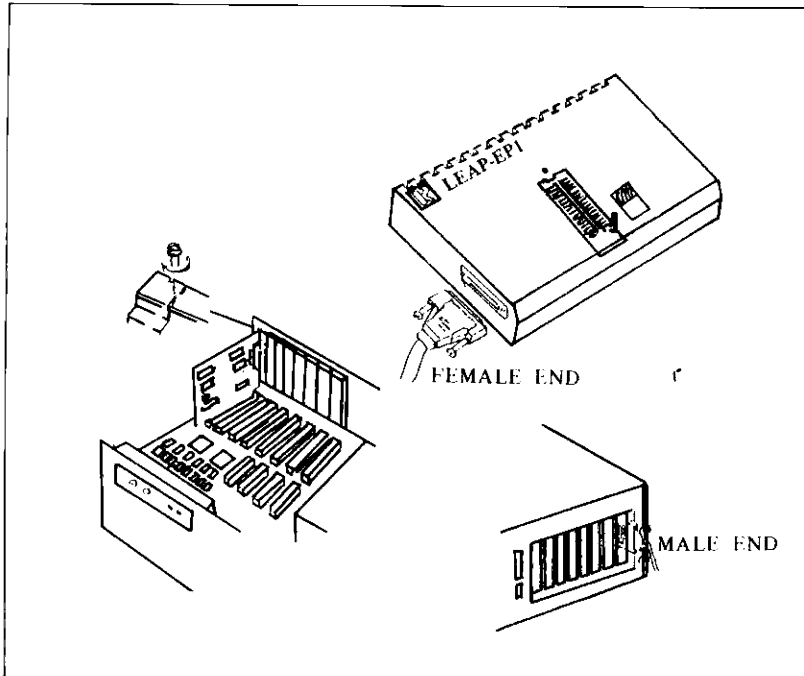
or

A>EP1-4M[Enter] — (4M)

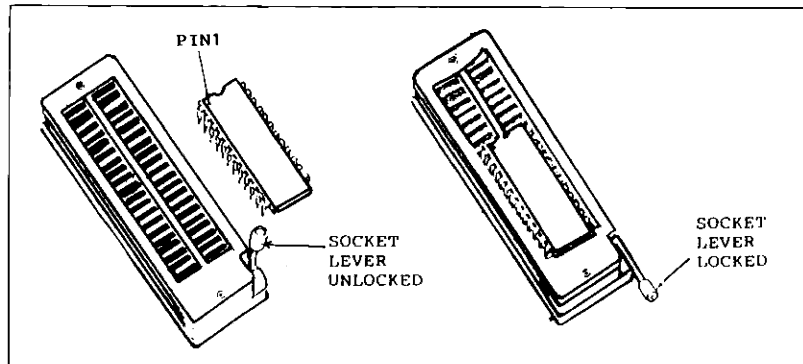
At this time, the big LEAP mark will appear on the screen and the system will automatically self-test. Press any key to start and the LEAP master menu as shown below will appear.



### C) PRODUCT OUTLINE



### D) DEVICE INSTALLATION

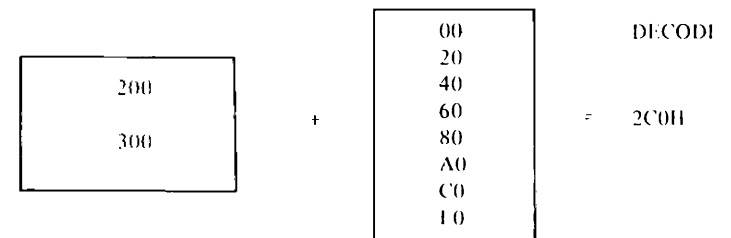
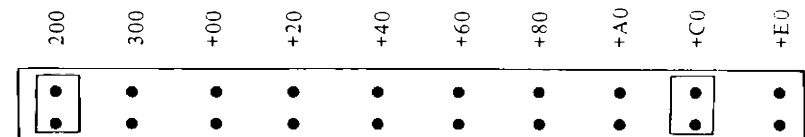


### E) I/O ADDRESSING CONFLICTS

#### 1) I/O PORT ADDRESS

I/O Address is divided into two big portions. The first section covers 200-2FF and the second section covers 300-3FF. The DIP Switch (SW1) handles 200-2FF and 300-3FF respectively, while Switch (SW2) handles the space range 01F(HEX).

In LEAP-EP1 the port address is default at 2C0h. If this I/O port address does not operate correctly in the system or comes in conflict with other interface cards, be sure therefore to temporarily remove other add-on cards or change the address of LEAP-EP1 (or others) by turning dip switches to on or off.

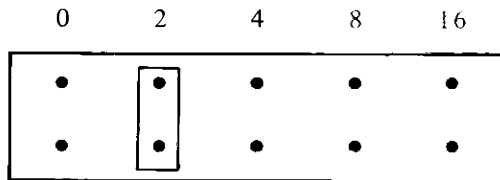


The illustration above identifies the I/O address position 2C0F. Users can select correct Jumper as per own usage.

## 2) I/O WAIT SELECT

Jumpers are provided on the LEAP-EP1 card to enable moving specified location on the hardware. The I/O wait of LEAP-EP1 is preset at 0,2,4,8,16. In altering the PC speed, user must expand the I/O wait. For example, PC XT will jump to 0 I/O wait and AT will jump to 2-8 I/O wait.

Take illustration below as example:



JUMPER 2

## VI. STARTING WITH E/EPROM

### A) APPLYING THE FUNCTIONS

\*(All the functions show below is only for reference only.

The detail functions should be refer to the software).

#### 1) TYPE

LEAP E/EPROM PROGRAMMER CARD MODEL: LEAP-EP1								
E/EPROM Programmer V1.0		Select device type	MEMSUM:01FE0000		CHECK SUM.0000			
TYPE	READ	CHECK	PROGRAM	VERIFY	DISK	HELP	PROCESS	PARAMETER
①	②	③	④	⑤	⑥	⑦	⑧	⑨
DEVICE DEFINITION			MODE: Intelligent2		BUFFER: 00000 - 07FFF			
DEVICE : 8bit EPROM			Tpw : 500uS		SIZE : 32K x 8 bits (7FFFh)			
VENDOR: Don't Care			Vcc : 5.0v		PROC : Check+Program+Verify			
TYPE : 27C256			Vpp : 13.0v		FILE : Haven't file.			
DATA : 8bit BUS			AREA: ALL address		FORMAT: MS-DOS Normal format			
[↑↓] Move cursor.		[Enter] Execute		[Esc] Escape		[H] Help		MAIN MENU

- |                                 |                                      |
|---------------------------------|--------------------------------------|
| 1) To select IC type and vendor | 6) To load/save disk data            |
| 2) To read device data          | 7) To display operating instructions |
| 3) To blank checks device       | 8) To display main buffer            |
| 4) To program device            | 9) To select Vpp, Vcc, Toe, Tpw      |
| 5) To verify device             |                                      |

LEAP E/EPROM PROGRAMMER CARD MODEL:LEAP-EP1		
E/EPROM Programmer V1.0 Select device type MEMSUM:01FE0000 CHECK SUM:0000		
TYPE READ CHECK PROGRAM VERIFY DISK HELP PROCESS PARAMETE		
<p>T Set vender/type. Space Set type only. sF1 8 bit EPROM. sF2 8 bit EEPROM.</p>		
<b>DEVICE DEFINITION</b> DEVICE : 8 bit EPROM VENDOR: HITACHI TYPE : 27C256G DATA : 8 bit BUS	MODE: Quickly Twp : 5ms Vcc : 6.00v Vpp : 12.50v AREA: ALL address	BUFFER : 00000-07FFF SIZE : 32K x 8 bits(7FFFh) PROC : Check + Program + Verify FILE : Haven't file. FORMAT: MS-DOS Normal format
[↑↓]Move cursor. [Enter]Execute. [H]Help. [tab]Automated [^Fz]Detail DEBUG MAIN MENU		

This section described the procedures in selection of device, vendor, and type of IC to be programmed. After each selection, this will be entered in the lower portion of the screen under "DEVICE DEFINITION".

8 bit EPROM Devices Manufacturer / Vender			
[Spc]	Geneyal type	[9]	MITSUBISHI
[0]	AMD	[A]	MOSTEK
[1]	ATMEL	[B]	MOTOROLA
[2]	EA	[C]	MXIC
[3]	EUROTECHNIQUE	[D]	NEC
[4]	FUJITSU	[E]	NS
[5]	HITACHI	[F]	OKI
[6]	HYUNDAI	[G]	RICOH
[7]	Intel	[H]	SeeQ
[8]	MATSHUSHITA	[I]	SHARP
[J]	SEIKO	[K]	SGS(ST)
[L]	Signetics	[M]	TI
[N]	TOSHIBA	[O]	UMC
[P]	VLSI		

[T] Select Vendor and Type.

Device type can be changed by using "T" for type select. The selection sequence are as follows. User is prompted by first menu with the list of currently supported vendors and followed by a 3rd window with list of devices type. Both selection can be entered by using the single key code for each vendor or device type.

After selection, LEAP-EP1 will return to main menu while both vendor's name and type will be listed in the lower portion under "DEVICE DEFINITION" of the screen.

General type DEVICE TYPE					
[0]	2716	25.0v	[D]	27256A	12.5v
[1]	2716P	21.0v	[E]	27C256	13.0v
[2]	27C16B	12.5v	[F]	27512	21.0v
[3]	2732	25.0v	[G]	27512A	12.5v
[4]	2732A	21.0v	[H]	27C512	13.0v
[5]	2732B	12.5v	[I]	27C101G	13.0v
[6]	2764	21.0v	[J]	27C301G	13.0v
[7]	2764A	12.5v	[K]	27C020	13.0v
[8]	27C64	13.0v	[L]	27C2001D	13.0v
[9]	27128	21.0v			
[A]	27128A	12.5v			
[B]	27C128	13.0v			
[C]	27256	21.0v			



**[Space]** Set IC Type only.  
 If user don't care about vendor selection, he can go direct into this command and screen will display list of currently supported IC type or number.

Press **[Ctrl][ESC]** escape key, at any time during the change device phase will return user to the previous master menu without affecting the currently selected device.

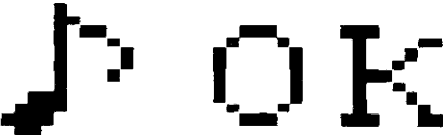
**[Shift][F1]** 8 Bit EPROM  
 This command allows user to go direct into selection of device type and vendor of 8 Bit EPROM.

**[Shift][F2]** 8 Bit EEPROM  
 This command allows user to go direct into selection of device type and vendor of 8 Bit EEPROM.

2) READ DEVICE TO MEMORY

LEAP E/EPROM PROGRAMMER CARD MODEL: LEAP-EPI								
E/EPROM Programmer V1.0	Read master data	MEMSUM:01FE0000 CHECK SUM:0000						
TYPE	READ CHECK PROGRAM VERIFY DISK HELP PROCESS PARAMETER							
<table border="1"> <tr> <td>R</td> <td>Read into address 0000.</td> </tr> <tr> <td>↑R</td> <td>Read into any address.</td> </tr> <tr> <td>sR</td> <td>Read device successive.</td> </tr> </table>			R	Read into address 0000.	↑R	Read into any address.	sR	Read device successive.
R	Read into address 0000.							
↑R	Read into any address.							
sR	Read device successive.							
<b>DEVICE DEFINITION</b>	<b>MODE:</b> Quickly	<b>BUFFER :</b> 00000 - 0FFFF						
DEVICE : 8bit EPROM	Twp : 1mS	SIZE : 32K x 8 bits (7FFFh)						
VENDOR: HITACHI	Vcc : 5.0v	PROC : Check+Program+Verify						
TYPE : 27C256	Vpp : 12.7v	FILE : Haven't file.						
DATA : 16bit BUS	AREA: ALL address	FORMAT: MS-DOS Normal format						
[↑↓] Move cursor.	[Enter] Execute	[Esc] Escape [H] Help. MAIN MENU						

When R' of Read is entered, the contents of the device in the module will be read into the memory buffer. After reading the data, a new check sum value will be calculated. Users are obliged to wait a few seconds when reading a large device. All correct check sum is displayed on the upper right hand side of the main menu screen. If the check sum of the current read in data is not the same with previous one, then the current stored checksum displayed in the main menu screen will blink.

LEAP E/EPROM PROGRAMMER CARD MODEL: LEAP-EP1		
E/EPROM Programmer V1.0 Read master data MEMSUM:00000000 CHECK SUM:0000		
TYPE READ CHECK PROGRAM VERIFY DISK HELP PROCESS PARAMETER		
<div style="border: 1px solid black; padding: 5px;"> <div style="border: 1px solid black; padding: 2px;"> <span style="float: left; margin-right: 10px;">R ^R sR</span> <span style="float: right;">i</span> <div style="clear: both;"></div> <p style="text-align: center;">READ DEVICE DATA</p> <p>Device start address [ 00000 ]  end address : [ 07FFF ]  Buffer start address : [ 00000 ]  Data check sum : [ 8000 ]</p> <div style="text-align: center; font-size: 2em; font-family: monospace;">  </div> </div> </div>		
DEVICE	07FFF	
DEVICE : 8 bit EPROM	Tpw : 500uS	SIZE : 32K x 8 bits (7FFFh)
VENDOR : Don't Care	Vcc : 5.0v	PROC : Check+Program+Verify
TYPE : 27C256	Vpp : 13.0v	FILE : Haven't file.
DATA : 8bit BUS	AREA: ALL address	FORMAT: MS-DOS Normal format

[Space] Escape display mode return main menu

LIST

[Ctrl] [R] Read in any address.

The 'CONTROL R' command is to read in any address. This allows user to select data device ranges that is to be read into buffer address from 'Start Address' to 'End Address'.

[Shift] [R] Read device successive.

With [SHIFT] [R] command, user can read in several small EPROMs successively into higher memory address of the memory buffer. It can combine data in small EPROMs and to program this data into a large EPROM.

Example when combining the contents of two 2732 data into a single 2764.

- 1) Select the vendor and device 2732 which you intend to read.
- 2) Read one 2732 into buffer memory 0-FFFh.
- 3) Load second 2732 into socket pod.
- 4) Depress [Shift] [R] to read into next address starting from 1000-1FFFh.
- 5) Select the vendor and device 2764.
- 6) Insert 2764 device into socket and program with [P] command.

For example in reading the contents of 4M bits data to buffer, follow the instructions listed below:

- 1) Select the vendor and device 27C4001 which you intend to read.
- 2) Enter "DISK" function and select { Shift } { I } command to open a cach file (U1.BUF) into disk. The capacity for the cache file are 1048576 bytes.
- 3) A window will appear and user is prompt to enter { pat9 h } { filename } to initial mega cache file.

### 3) CHECK DEVICE BLANK

LEAP E/EPROM PROGRAMMER CARD MODEL: LEAP-EP1		
E/EPROM Programmer V1.0 Check device blank MEMSUM:01FE0000 CHECK SUM:0000		
TYPE READ <b>CHECK</b> PROGRAM VERIFY DISK HELP PROCESS PARAMETER		
<div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;">           C Device blank check.            N Check device data sum.         </div>		
<b>DEVICE DEFINITION</b>	MODE: Quickly	BUFFER : 0000 - 0FFFF
DEVICE : 8bit EPROM	Twp : 1mS	SIZE : 32K x 8 bits (7FFFh)
VENDOR: HITACHI	Vcc : 5.0v	PROC : Check+Program+Verify
TYPE : 27C256	Vpp : 12.7v	FILE : Haven't file.
DATA : 16bit BUS	AREA: ALL address	FORMAT: MS-DOS Normal format
[↑↓] Move cursor.    [Enter] Execute.    [Esc] Escape    [H] Help.    MAIN MENU		

Blank checks the device before programming.

- [C] Device blank check.  
The [C] for device blank check allows user to verify or check a device before programming and to make sure that it is empty. Screen will display "PASS", if device is verified blank and "ERROR", if device is verified with data.
- [N] Check device data sum.  
Check the master data sum of IC oh socket without covering the original data in main buffer. User can get

the check sum even without reading master data into memory buffer.

### 4) PROGRAMMING

LEAP E/EPROM PROGRAMMER CARD MODEL: LEAP-EP1		
E/EPROM Programmer V1.0 Programming device MEMSUM:01FE0000 CHECK SUM:0000		
TYPE READ CHECK <b>PROGRAM</b> VERIFY DISK HELP PROCESS PARAMETER		
<div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;">           P Program from address 0000.            ↑P Program from any address.            sP Program device successive.            B EEPROM device chip erase.            ↑A Auto blank check . . . ON         </div>		
<b>DEVICE DEFINITION</b>	MODE: Quickly	BUFFER : 0000 - 0FFFF
DEVICE : 8bit EPROM	Twp : 1mS	SIZE : 32K x 8 bits (7FFFh)
VENDOR: HITACHI	Vcc : 5.0v	PROC : Check+Program+Verify
TYPE : 27C256	Vpp : 12.7v	FILE : Haven't file.
DATA : 16bit BUS	AREA: ALL address	FORMAT: MS-DOS Normal format
[↑↓] Move cursor.    [Enter] Execute.    [Esc] Escape.    [H] Help.    MAIN MENU		

The [P] programming command allows you to copy the data stored in the memory buffer into the device on the socket.

Before performing programming command, system will always blank checks device first, and after programming, the contents of the device will be verified against the contents of the memory buffers to confirm the programming outcome of the device. The message "PASS" will appear on the screen when everything is successfully completed. If there are any error, then the "FAIL" message will appear.

**[P]** Program from address 0000.  
Programming device data starting from buffer address 0000. User will always be required to program from address 0000, then enter targeted end address, buffer start address and checking address.

**NOTE:** Do not remove or insert any device into socket when program starts working. If device is verified with data, screen will display "ERROR", then enter [Space] to return to main menu.

**[Ctrl][P]** Program from any address.  
Program device data from any address.  
After typing [Ctrl][P], user will be prompted to enter start address to end address and buffer start address to programming address.

**NOTE:** Do not remove or insert any device into socket when program starts working.

**[Shift][P]** Program device successively.  
With [Shift][P] command, user can program one large EPROM successively into smaller memory address of the memory buffer.

Take for instance in programming one 27512 into two 27256:

- 1) Select type to 27512.
- 2) Read 27512 data to buffer.
- 3) Select type to 27256.

- 4) Enter [P] command to program the first IC from start address 000000-07FFF.
- 5) Enter [Shift][P] command to program the second IC from 08000-0FFFF.

**[B]** EEPROM device chip erase.  
Bulk erases the electronically erasable parts before LEAP-EP1 attempts to program them.

To erase a device, simply insert the device you want to erase and press "RETURN". LEAP-EP1 will erase the device and a message will appear to indicate that the operation has been completed.

If incorrect device type is entered, screen will display  
Operation error: Function suitable in erasing EEPROM only!

Press [ESC] to return to main menu and select the right type.

**[Ctrl][A]** Auto blank check . . . ON (OFF).  
To modify program enter this command, and screen will display the "Program Check" window. Choose [0] Turn-off, if you wish to eliminate blank checking device before program and [1] Turn-on, if you wish to blank check device automatically before program.

## 5) VERIFY DEVICE

LEAP E/EPROM PROGRAMMER CARD. MODEL: LEAP-EPI			
E/EPROM Programmer V1.0		Verify device data	MEMSUM:01FE0000 CHECK SUM:0000
TYPE	READ	CHECK	PROGRAM VERIFY DISK HELP PROCESS PARAMETER
		V Verify with address 0000. ^V Verify with any address. sV Verify with next address. E Display verify error . . OFF	
<b>DEVICE DEFINITION</b>	MODE: Quickly	BUFFER : 00000 - 0FFFF	
DEVICE : 16bit EPROM	Twp : 1mS	SIZE : 32K x 8 bits (7FFFh)	
VENDOR: HITACHI	Vcc : 5.0v	PROC : Chick+Program+Verify	
TYPE : 27C256	Vpp : 12.7v	FILE : Haven't file.	
DATA : 16bit BUS	AREA: ALL address	FORMAT: MS-DOS Normal format	
[↑] Move cursor. [Enter] Execute. [Esc] Escape. [H] Help. MAIN MENU			

The "V" verify command allows you to compare the contents of the device in the socket to the data in memory buffer.

**[V]** Verify with address 00000.  
 Verify device data starting from buffer address 00000. Data will always be verified from buffer starting from address 00000, then enter targeted end addresses, buffer start address and verifying address.

**WARNING:** Do not remove or insert any device on the socket when program starts working.

**[R]** Read into address 00000.  
 Reading device data will always starts at address 0000.  
 User will not be given the choice of selecting own addresses, but can also go direct into the current file.

**[Ctrl][V]** Verify with any address.  
 Verify device data from any address.  
 After typing [Ctrl][V], user will be prompted to enter start address to end address and buffer start address to verifying address.

**[Shift][V]** Verify device successive.  
 With [Shift][V], user can verify several small EPROMs successively into higher memory address of the memory buffer.

LEAP E/EPROM PROGRAMMER CARD MODEL: LEAP-EPI																																			
E/EPROM Programmer V1.0		Verify device data	MEMSUM:00000000 CHECK SUM:0000																																
TYPE	READ	CHECK	PROGRAM VERIFY DISK																																
		<table border="1"> <thead> <tr> <th colspan="2">VERIFY DEVICE DATA</th> </tr> </thead> <tbody> <tr> <td>Device start address :</td> <td>  00000  </td> </tr> <tr> <td>end address :</td> <td>  07FFF  </td> </tr> <tr> <td>Buffer start address :</td> <td>  00000  </td> </tr> <tr> <td>Verifying address :</td> <td>  00000  </td> </tr> </tbody> </table>		VERIFY DEVICE DATA		Device start address :	00000	end address :	07FFF	Buffer start address :	00000	Verifying address :	00000																						
VERIFY DEVICE DATA																																			
Device start address :	00000																																		
end address :	07FFF																																		
Buffer start address :	00000																																		
Verifying address :	00000																																		
		<table border="1"> <thead> <tr> <th colspan="2">ERROR LIST</th> </tr> </thead> <tbody> <tr> <td>NO BUFFER</td> <td>DEVICE</td> </tr> <tr> <td>00001.00000:00</td> <td>11 00000</td> </tr> <tr> <td>00002.00001:00</td> <td>11 00001</td> </tr> <tr> <td>00003.00002:02</td> <td>11 00002</td> </tr> <tr> <td>00004.00003:00</td> <td>11 00003</td> </tr> <tr> <td>00005.00004:04</td> <td>11 00004</td> </tr> <tr> <td>00006.00005:00</td> <td>11 00005</td> </tr> <tr> <td>00007.00006:06</td> <td>11 00006</td> </tr> <tr> <td>00008.00007:00</td> <td>11 00007</td> </tr> <tr> <td>00009.00008:08</td> <td>11 00008</td> </tr> <tr> <td>0000A.00009:00</td> <td>11 00009</td> </tr> <tr> <td>0000B.0000A:0A</td> <td>11 0000A</td> </tr> <tr> <td>0000C.0000B:00</td> <td>11 0000B</td> </tr> <tr> <td>0000D.0000C:0C</td> <td>11 0000C</td> </tr> <tr> <td>0000E.0000D:00</td> <td>11 0000D</td> </tr> </tbody> </table>		ERROR LIST		NO BUFFER	DEVICE	00001.00000:00	11 00000	00002.00001:00	11 00001	00003.00002:02	11 00002	00004.00003:00	11 00003	00005.00004:04	11 00004	00006.00005:00	11 00005	00007.00006:06	11 00006	00008.00007:00	11 00007	00009.00008:08	11 00008	0000A.00009:00	11 00009	0000B.0000A:0A	11 0000A	0000C.0000B:00	11 0000B	0000D.0000C:0C	11 0000C	0000E.0000D:00	11 0000D
ERROR LIST																																			
NO BUFFER	DEVICE																																		
00001.00000:00	11 00000																																		
00002.00001:00	11 00001																																		
00003.00002:02	11 00002																																		
00004.00003:00	11 00003																																		
00005.00004:04	11 00004																																		
00006.00005:00	11 00005																																		
00007.00006:06	11 00006																																		
00008.00007:00	11 00007																																		
00009.00008:08	11 00008																																		
0000A.00009:00	11 00009																																		
0000B.0000A:0A	11 0000A																																		
0000C.0000B:00	11 0000B																																		
0000D.0000C:0C	11 0000C																																		
0000E.0000D:00	11 0000D																																		
<b>DEVICE DEF</b>	DEVICE : 8bit EPROM	Tpw : 500uS	SI																																
VENDOR: Don't Care	Vcc : 5.0v		PR																																
TYPE : 27C256	Vpp : 13.0v		LI																																
DATA : 8bit BUS	AREA: ALL address		LOR																																

**[S]** Stop list, Any key to continue.

**[ESC]/[C]** End list error

**ERROR LIST**

[E] Display verify error. . OFF (or ON)  
 Enter this command, and screen will display the "Display Error" window. Choose [O] Turn-off, if you wish to eliminate verifying device after program and only display the verified error data in device. And [1] Turn-on, if you wish to verify device automatically after program and only display portion of verified error data in device.

## 6) DISK

LEAP E/EPROM PROGRAMMER CARD MODEL: LEAP-EP1		
E/EPROM Programmer V1.0 Disk file process MEMSUM:01FE0000 CHECK SUM:0000		
TYPE READ CHECK PROGRAM VERIFY DISK HELP PROCESS PARAMETER		
↑D List disk directory. L Load disk data file in buffer. S Save buffer data on disk. sL Load encryption data in buffer. sS Save encryption data on disk. ↑M Input macro key to buffer. ↑E Erase macro key data. ↑T List macro key to screen. ↑L Load macro key file in buffer. ↑S Save macro key file on disk. sT Type file character to screen. ↑I Rest time, System lock. ↑Q End work, Return main menu.		
<b>DEVICE DEFINITION</b> DEVICE : 8bit EPROM VENDOR: HITACHI TYPE : 27C256 DATA : 16bit BUS	<b>MODE: Quickly</b> Twp : 1mS Vcc : 5.0v Vpp : 12.7v AREA: ALL address	<b>BUFFER : 0000 - 0FFFF</b> SIZE : 32K x 8 bits (7FFFh) PROC : Check+Program+Verify FILE : Haven't file. FORMAT: MS-DOS Normal format
[↑↓] Move cursor. [Enter] Execute. [Esc] Escape. [H] Help. MAIN MENU		

LEAP E/EPROM PROGRAMMER CARD MODEL: LEAP-EP1		
E/EPROM Programmer V1.0 Disk file process MEMSUM:01FE0000 CHECK SUM:0000		
TYPE READ CHECK PROGRAM VERIFY DISK HELP PROCESS PARAMETER		
		↑D List disk directory
LOAD DATA FILE FORMAT SELECT		
[0] MS-DOS fn.COM format.	[C] ASCII BHLF format.	file in buffer.
[1] MS-DOS fn.EXE format.	[D] ASCII BIOF format.	ta on disk.
[2] MS-DOS Normal format.	[E] ASCII OCT ( ) format.	ryption data.
[3] Intel HEX format.	[F] ASCII OCT ( % ) format.	ryption on disk.
[4] Tektronix HEX format.	[G] ASCII OCT ( , ) format.	files in buffer.
[5] Signetics HEX format.	[H] ASCII OCT SMS format.	file on disk.
[6] Motorola (%S) format.	[I] ASCII HEX ( ) format.	system.
[7] Intel 80/86 HEX format.	[J] ASCII HEX ( ? ) format.	
[8] Motorola EXOR ciser (S1)	[K] ASCII HEX ( , ) format.	
[9] Motorola EXOR macs (S2)	[L] ASCII HEX ( , ) format.	
[A] Motorola 32 bit format.	[M] ASCII HEX SMS format.	
[B] ASCII BPNF format.		
[0-Z] Direct set. [↑] [↓] Move cursor. [Enter] Set. [Esc] Escape. SELECT U		

[CTRL] [D] List disk directory

Enumerate all filenames of disk on screen.

[L]

Load disk data file in buffer

Loads the data from the file selected into memory buffer. After typing [L], the system will display the file formats ranging from [0] to [M]. Select the right format and user will be prompted to enter the filename.

**[S]** Save buffer data  
Save the data memory buffer into disk. After typing [S], user will be prompted to load the filename, start and end addresses. If data are all saved, then display will show "OK". If filename is not loaded, display will read DISK ERROR. If another file exist, display will read DISK ERROR, file exist, enter [Y] Yes or [N] No.

**NOTE:** LEAP-EP1 utilizes the normal format, so we suggest that when activating [L] & [S] commands, kindly use the same [2] normal format.

**[Shift] [L]** Load extra/ encryption data  
Load added password number into extra memory buffer.

**[Shift] [S]** Save extra/encryption data  
Save added password number into disk.

**[Ctrl] [M]** Input and set macro key.  
Macro command enables user to capture, delete, initialize (erase), list, read data to a file and load macros from a file.

In [Ctrl] [M] command user can eliminate the primary set-up procedures and use the keystroke commands to be recorded as macros and recall by pressing any of the F1-F10 user's define keys.

Take for example in setting EPROM Intel 27010, please follow the instructions below:

- 1) Enter "DISK" function and select [Ctrl] [M] command.
- 2) A window will appear and user is prompt to enter [F1] key to macro key capture and 27010 to macro key remark.
- 3) Press [Enter] to lock selected file.
- 4) Press [Alt][T] to enter select IC type window.
- 5) Press [sF1] key to directly select 8bit EPROM.
- 6) Press number [7] key to go direct into manufacturer "INTEL".
- 7) Press [K] key select type 27010 directly.
- 8) Press [Ctrl][M] to enter macro command.
- 9) Press [F1] key and system will run macro.

**NOTE:** Do not use the cursor key to command instructions, otherwise the whole macro memory procedures will be distorted. Just make use of the proper function keys.

**[Ctrl] [E]** Erase Macro key data  
Enter this command and the screen will display the macro key list with 10 [F1]-[F10] function keys, type the selection, which you intend to erase in the Macro key capture caption. Once macro key is entered, the system will automatically erase the macro key. Enter [Space] to escape display mode and return to main menu..

- [Ctrl][T]** List macro key data  
Enter this command to open the window with list of [F1] to [F10] Macro key. This will allow user to review which macro key was canceled.
- [Shift][S]** Save extra/encryption data  
Save added password number into disk.
- [Ctrl][L]** Load macro key file in buffer  
This command is to load 2nd set of macro key files into memory buffer and without cancelling contents of the main macro keys. Just enter [Ctrl][L] and the second set of users define keys will be loaded into the memory buffer.
- [Ctrl][S]** Save macro key file on disk  
Save macro key file into disk. In case another file exists screen will display disk error, user will be prompted to renew or erase the previous file.
- [Shift][T]** Type file character to screen  
To scroll the data of any file on screen.  
After typing [Shift][T], screen will display the "List file to screen" window. User will be prompted to enter the filename of disk to view. Enter [Y] for yes, if you want to scan file page by page, and enter [N] for No, if you intend to scroll file on screen.

View file control keys are as following:

- [A]** In page list mode.  
Means to scroll thru file.
- [S]** Stop display, followed by any key to continue.  
Means to halt file on display.
- [E]** Escape display, return to master menu.  
Means to exit display.
- [Ctrl][I]** Rest time, system lock.  
The [Ctrl][I] command was especially designed to allow user to have a "time out" period. Enter your own password and system will locked data automatically. Return to work by entering the same password. This can also prevent intruders from pirating or using your file.
- [Ctrl][Q]** End work, Exit System  
This command allow you to escape LEAP-EP1 and return to DOS. After typing [Ctrl][Q], screen will verify if you are sure to exit system, enter [Y] yes, if you want to escape or [N] No, if you intend to remain.

When LEAP-EP1 exit and return to DOS, all parameters set by user will be saved in DISK.



## 7) HELP

LEAP E/EPROM PROGRAMMER CARD MODEL: LEAP-EP1		
E/EPROM Programmer V1.0 Select device type MEMSUM:01FE0000 CHECK SUM.0000		
TYPE READ CHECK PROGRAM VERIFY DISK HELP PROCESS PARAMETER		
H Operation explain help. I Device information.		
<b>DEVICE DEFINITION</b> DEVICE : 8bit EPROM VENDOR: HITACHI TYPE : 27C256 DATA : 16bit BUS	<b>MODE: Quickly</b> Twp : 1mS Vcc : 5.0v Vpp : 12.7v AREA: ALL address	BUFFER : 00000 - 0FFFF SIZE : 32K x 8 bits (7FFFh) PROC : Check+Program+Verify FILE : Haven't file. FORMAT: MS-DOS Normal format
[↑↓] Move cursor. [Enter] Execute. [Esc] Escape. [H] Help. MASTER MENU		

**[H]** Operation explain help  
 Open this command and user will find 6 pages compilation of the basic operations of LEAP-EP1. Use [Pg][Up] or [Pg][Dn] to turn the pages. [ESC] to escape display.

Main DEV	DEVICE PIN CONFIGURATION	
[0] EPROM 2716 2K x 8bit	Vpp■ 1	28 ■Vcc
[1] EPROM 2732 4K x 8bit	A12■ 2	27 ■A14
[2] EPROM 2764 8K x 8bit	A7■ 3	26 ■A13
[3] EPROM 27128 16K x 8bit	A6■ 4	25 ■A8
[4] EPROM 27256 32K x 8bit	A5■ 5	24 ■A9
[5] EPROM 27512 64K x 8bit	A4■ 6	23 ■A11
[6] EPROM 27101 128K x 8bit	A3■ 7	22 ■/OE
[7] EPROM 27301 128K x 8bit	A2■ 8	21 ■A10
[8] EPROM 2804 512 x 8bit	A1■ 9	20 ■/CE
[9] E2PROM 2816 2K x 8bit	A0■ 10	19 ■D7
[A] E2PROM 2817 2K x 8bit	D0■ 11	18 ■D6
[B] E2PROM 2864 8K x 8bit	D1■ 12	17 ■D5
[C] E2PROM 28256 32K x 8bit	D2■ 13	16 ■D4
	GND■ 14	15 ■D3
[0~Z]	[0~K]/[↓][↑] Move. [Esc] Exit.	

**[I]** Device information  
 Display IC pin configuration. Use cursor keys to select. [ESC] to exit display.

## 8) PROCESS

LEAP E/EPROM PROGRAMMER CARD MODEL: LEAP-EP1		
E/EPROM Programmer V1.0 Buffer data process MEMSUM:01FE0000 CHECK SUM:0000		
TYPE READ CHECK PROGRAM VERIFY DISK HELP PROCESS PARAMETER		
<p> <b>D</b> Dump buffer HEX data.  <b>U</b> Display buffer used map.  <b>sD</b> Edit encryption table.  <b>↑X</b> Divide 16/32/64 to 8bits.  <b>sX</b> Combine 8 to 16/32 bit  <b>sC</b> Buffer data lock/unlock.  <b>↑N</b> Get memory check sum.  <b>↑F</b> Buffer fill (FF)h data.  <b>sO</b> Buffer fill (00)h data.  <b>s9</b> Fill sequential word.  <b>s8</b> Fill sequential byte.                 </p>		
<b>DEVICE DEFINITION</b>	MODE: Quickly	BUFFER : 00000 - 0FFFF
DEVICE : 8bit EPROM	Twp : 1mS	SIZE : 32K x 8 bits (7FFFh)
VENDOR: HITACHI	Vcc : 5.0v	PROC : Check+Program+Verify
TYPE : 27C256	Vpp : 12.7v	FILE : Haven't file.
DATA : 16bit BUS	AREA: ALL address	FORMAT: MS-DOS Normal format
[↑↓] Move cursor. [Enter] Execute. [Esc] Escape. [H] Help. MAIN MENU		

**[D]** Dump and simple EDIT buffer HEX data.  
 Select this command and screen will display the address and the current contents of the address in EPROM. User will be prompted to edit contents of the memory buffer by applying HEX and binary format.

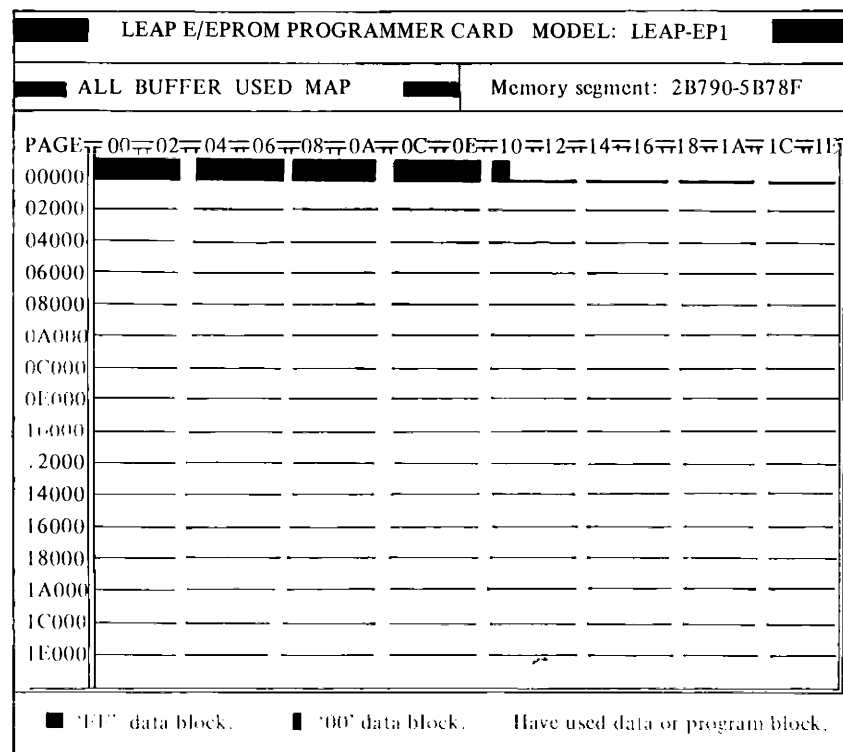
For application follow the list of keys below:

- [Ctrl][Home] Dump to first (start) page.
- [Ctrl][End] Dump to last page.
- [Shift][PgUp] To Change 40000h address.
- [PgDn]

- [Ctrl][PgUp] To change 1000h address.
- [PgDn]
- [Ctrl][E] To Edit data in HEX format.  
Initiate command by using [0] ~ [9], [A] ~ [F] key.
- [Ctrl][A] To Edit data in ASCII format.
- [Ctrl][D] To enter Dump page new address.
- [Ctrl][I] Set buffer index point in cursor address.
- [Ctrl][J] Jump to index point.
- [ESC] To exit command and return to master menu.

LEAP E/EPROM PROGRAMMER CARD MODEL: LEAP-EP1																								
BUFFER PROCESS Ver1.0				HEX data edit/dump				Memory segment: 10B70-60B6F																
EDIT		PROCESS		FILL		SEARCH		HELP		DISK														
ADDRESS	0	.1	.2	.3	.4	.5	.6	.7	.8	.9	A	B	C	D	E	F	ASCII Code							
3FF60	60	FF	62	FF	64	FF	66	FF	68	FF	6A	FF	6C	FF	6E	FF	b	d	f	h	j	l	n	
3FF70	70	FF	72	FF	74	FF	76	FF	78	FF	7A	FF	7C	FF	7E	FF	p	r	t	v	x	z	~	
3FF80	80	FF	82																					
3FF90	90	FF	92																					
3FFA0	A0	FF	A2																					
3FFB0	B0	FF	B2																					
3FFC0	C0	FF	C2																					
3FFD0	D0	FF	D2																					
3FFE0	E0	FF	E2																					
3FFF0	F0	FF	F2																					
00000	00	00	00	02																				
00010	10	00	12																					
00020	20	00	22																					
00030	30	00	32	00	34	00	36	00	38	00	3A	00	3C	00	3E	00	0	2	4	6	8	:	<	>
00040	40	00	42	00	44	00	46	00	48	00	4A	00	4C	00	4E	00	@	B	D	F	H	J	L	N
00050	50	00	52	00	54	00	56	00	58	00	5A	00	5C	00	5E	00	P	R	T	V	X	Z	\	^
Cursor address: 3FF60													HEX [60]		BIN [0110 0000]			ASCII						

- [>] Move cursor.
- [Ins] Insert
- [Del,] delete.
- [Enter] Input.
- ENTER



[Space] Change Mode. [PgUp][PgDn] Change page. [Esc] Exit. List

[U] Display buffer used map.  
To display all/portion used map. Open this command to view the scope of area blocked with data.

[Shift][D] EDIT extra buffer data.

Select this command to examine or modify the extra (use 87C51) encryption data in memory buffer. For instance, if the 32 byte password of 87C51 is registered into the extra buffer, it will no longer be readable by those who should not have access to it.

After entering [Shift][D], user will be prompted to load the address to be edited, followed by the HEX address, and the carriage return. The screen will display a 256 byte page of data from memory buffer.

[Ctrl][X] Divide 16/32/64 bit to 8 bit.

This command is to allow division of 16/32/64 bits into 2,4,8 bits files. Upon entering command, and the screen will display the compile Bit number. User will be prompted to select between [0] 16 bit source to [2] 64 bit source. After entering the right choice, the 2nd window will display the Target byte position, type your choice and the 3rd window will prompt you to enter start and end addresses.

**Example.** The 16 bit source files can be divided into 8 bit files. The [Ctrl][X] command can divide a 16 bit files into two 8 bit files that will be programmed to an EVEN EPROM and an ODD EPROM.

[Shift][X] Combine 8 bits into 16/32 bit.

This command allow combining two or four 8 bits blocks into 16/32 bit data blocks.

**Example:** When combining two 8 bits blocks into a 16 bits file. The 1st 8 bit file will be collected to LOW byte of 16 bits file and the 2nd 8 bits file will be collected to HI byte of 16 bits file.

**[Shift] [C]** Buffer data lock/unlock.  
This command allows user to utilize data of extra buffer in creating password to secure data of main buffer. The length of password can readily be changed.

**[Ctrl] [N]** Get memory check sum.  
To retrieve and calculate the check sum of memory buffer.  
**Example:** Select device type to 27256, enter [Ctrl] [N] command and it will automatically add the sum of 0000-7FFF, this will again be translated into 4 byte hexadecimal.

**[Ctrl] [F]** Buffer fill (FFh) data  
Fill buffer memory with 'FF' data. After typing [Ctrl] [F], user will be prompted for inputting selection [Y]es or [N]o to initialize the command.

**[Shift] [φ]** Buffer fill (φφh) data  
Fill buffer memory with 'φφ' data. After typing [Shift] [φ], user will be prompted for inputting [Y]es or [N]o to initialize the command.

**[Shift] [9]** Fill sequential word  
Fill sequential word data into all buffer.

**Example:** (00000:00 00 02 00 04 00 ~  
FC FD FE FF)

**[Shift] [8]** Fill sequential byte  
Fill sequential byte data into all buffer.

**Example:** (00000.00 01 02 03 04 05 ~  
FC FD FE FF)

## 9) PARAMETER

LEAP E/EPROM PROGRAMMER CARD MODEL: LEAP-EPI																																		
E/EPROM Programmer V1.0 System parameter MEMSUM:01FE0000 CHECK SUM:0000																																		
TYPE READ CHECK PROGRAM VERIFY DISK HELP PROCESS PARAMETER																																		
<table border="1"> <thead> <tr> <th colspan="2">PROGRAM FLOW CHART</th> </tr> </thead> <tbody> <tr><td>{0}</td><td>Standard/1 ... 50mS</td></tr> <tr><td>{1}</td><td>Standard/2 ... 10mS</td></tr> <tr><td>{2}</td><td>Quickly ... 5mS</td></tr> <tr><td>{3}</td><td>Intelligent/1 ... 1mS</td></tr> <tr><td>{4}</td><td>Intelligent/2 ... 500 uS</td></tr> <tr><td>{5}</td><td>Quick pulse ... 100 uS</td></tr> <tr><td>{6}</td><td>Interactive/1 ... 1mS</td></tr> <tr><td>{7}</td><td>Interactive/2 ... 500 uS</td></tr> </tbody> </table>		PROGRAM FLOW CHART		{0}	Standard/1 ... 50mS	{1}	Standard/2 ... 10mS	{2}	Quickly ... 5mS	{3}	Intelligent/1 ... 1mS	{4}	Intelligent/2 ... 500 uS	{5}	Quick pulse ... 100 uS	{6}	Interactive/1 ... 1mS	{7}	Interactive/2 ... 500 uS	<table border="1"> <thead> <tr> <th colspan="2">PROGRAM Tpw</th> </tr> </thead> <tbody> <tr><td>{0}</td><td>50mS</td></tr> <tr><td>{1}</td><td>10mS</td></tr> <tr><td>{2}</td><td>5mS</td></tr> <tr><td>{3}</td><td>1mS</td></tr> <tr><td>{4}</td><td>500 uS</td></tr> <tr><td>{5}</td><td>100 uS</td></tr> </tbody> </table>	PROGRAM Tpw		{0}	50mS	{1}	10mS	{2}	5mS	{3}	1mS	{4}	500 uS	{5}	100 uS
PROGRAM FLOW CHART																																		
{0}	Standard/1 ... 50mS																																	
{1}	Standard/2 ... 10mS																																	
{2}	Quickly ... 5mS																																	
{3}	Intelligent/1 ... 1mS																																	
{4}	Intelligent/2 ... 500 uS																																	
{5}	Quick pulse ... 100 uS																																	
{6}	Interactive/1 ... 1mS																																	
{7}	Interactive/2 ... 500 uS																																	
PROGRAM Tpw																																		
{0}	50mS																																	
{1}	10mS																																	
{2}	5mS																																	
{3}	1mS																																	
{4}	500 uS																																	
{5}	100 uS																																	
		<table border="1"> <tbody> <tr><td>M</td><td>Programming flow chart.</td></tr> <tr><td>C</td><td>Programming Vcc voltage</td></tr> <tr><td>B</td><td>Programming Vpp voltage</td></tr> <tr><td>W</td><td>Programming pulse width (Tpw)</td></tr> <tr><td>sW</td><td>Read pulse width adjust</td></tr> <tr><td>sA</td><td>8 bit BUS ALL address</td></tr> <tr><td>sE</td><td>16 bit BUS Even address</td></tr> <tr><td>O</td><td>Odd address</td></tr> <tr><td>↑O</td><td>Select driver I/O decode</td></tr> </tbody> </table>	M	Programming flow chart.	C	Programming Vcc voltage	B	Programming Vpp voltage	W	Programming pulse width (Tpw)	sW	Read pulse width adjust	sA	8 bit BUS ALL address	sE	16 bit BUS Even address	O	Odd address	↑O	Select driver I/O decode														
M	Programming flow chart.																																	
C	Programming Vcc voltage																																	
B	Programming Vpp voltage																																	
W	Programming pulse width (Tpw)																																	
sW	Read pulse width adjust																																	
sA	8 bit BUS ALL address																																	
sE	16 bit BUS Even address																																	
O	Odd address																																	
↑O	Select driver I/O decode																																	
<table border="1"> <thead> <tr> <th>DEVICE DEFINITION</th> <th>MODE: Quickly</th> <th>BUFFER: 0000 - 0FFFF</th> </tr> </thead> <tbody> <tr> <td>DEVICE: 8bit EPROM</td> <td>Twp: 1mS</td> <td>SIZE: 32K x 8 bits (7FFFh)</td> </tr> <tr> <td>VENDOR: HITACHI</td> <td>Vcc: 5.0v</td> <td>PROC: Check+Program+Verify</td> </tr> <tr> <td>TYPE: 27C256</td> <td>Vpp: 12.7v</td> <td>FILE: Haven't file.</td> </tr> <tr> <td>DATA: 16bit BUS</td> <td>AREA: ALL address</td> <td>FORMAT: MS-DOS Normal format</td> </tr> </tbody> </table>		DEVICE DEFINITION	MODE: Quickly	BUFFER: 0000 - 0FFFF	DEVICE: 8bit EPROM	Twp: 1mS	SIZE: 32K x 8 bits (7FFFh)	VENDOR: HITACHI	Vcc: 5.0v	PROC: Check+Program+Verify	TYPE: 27C256	Vpp: 12.7v	FILE: Haven't file.	DATA: 16bit BUS	AREA: ALL address	FORMAT: MS-DOS Normal format																		
DEVICE DEFINITION	MODE: Quickly	BUFFER: 0000 - 0FFFF																																
DEVICE: 8bit EPROM	Twp: 1mS	SIZE: 32K x 8 bits (7FFFh)																																
VENDOR: HITACHI	Vcc: 5.0v	PROC: Check+Program+Verify																																
TYPE: 27C256	Vpp: 12.7v	FILE: Haven't file.																																
DATA: 16bit BUS	AREA: ALL address	FORMAT: MS-DOS Normal format																																
<table border="1"> <tbody> <tr> <td>[↑↓]</td><td>Move cursor.</td> <td>[Enter]</td><td>Execute.</td> <td>[Esc]</td><td>Escape.</td> <td>[H]</td><td>Help.</td> <td>MAIN MENU</td> </tr> </tbody> </table>			[↑↓]	Move cursor.	[Enter]	Execute.	[Esc]	Escape.	[H]	Help.	MAIN MENU																							
[↑↓]	Move cursor.	[Enter]	Execute.	[Esc]	Escape.	[H]	Help.	MAIN MENU																										

The programming algorithm is automatically selected when you change the manufacturer and type, but you can change it by yourself and only if you made it in accordance with the programming rules.

**[M]** Programming flow chart  
Depress [M] to select required programming mode. Each selection will be entered in the lower portion of the screen for programming reference.

**[Ctrl][C]** Programming Vcc voltage  
Select required Vcc voltage prior to programming. Enter this command and user will be given three (0) 5.00v (1) 6.00v (2) 6.25v different choices to select and each selection will be listed in the lower portion of the screen for programming reference.

**[Ctrl][B]** Programming Vpp voltage  
Alter required Vpp voltage prior to programming. User is given 0.1 up to 25.0 to key in choice of voltages and each selection will be listed in the lower portion of the screen for programming reference.

**[Ctrl][W]** Programming pulse width (TPW)  
Alter required pulse width prior to programming. User is given 50ms (lowest) to 100us (fastest) choice of pulse width to select. Each selection will be recorded in the lower portion of the screen for programming reference.

**NOTE:** Each manufacturer have designed their devices with it's respective pulse width, so make sure to match the device to be programmed with its corresponding programming algorithm otherwise the device data will not be properly programmed.

After selecting programming flow chart, the Tpw will

automatically be changed.

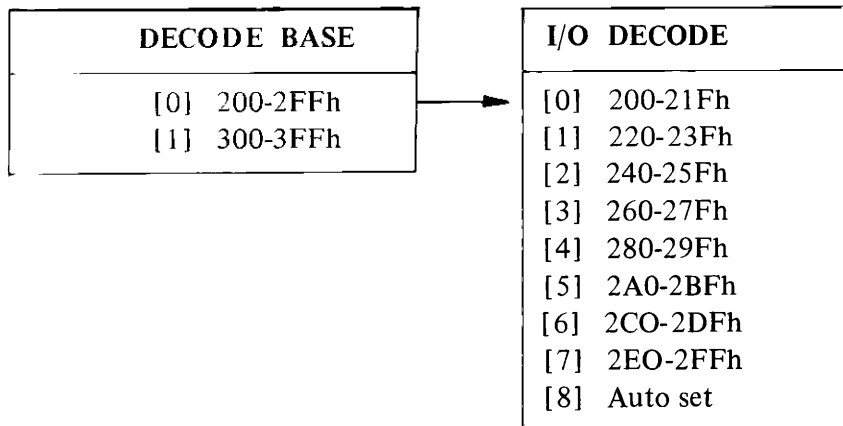
**[Shift][W]** Read pulse width Toe  
This command signifies the time interval in reading data to a device. The length of reading pulse width varies from [0] 1us to [5] 200us. 1us is considered the fastest, while 200us considered the slowest. User will be prompted to enter the correct choice.

**NOTE:** If IC encounters unreliable conditions in READ/CHECK/VERIFY commands, user can adjust pulse width to slowest us in attaining stable results.

**[Shift][A]** 8 Bit BUS ALL address  
Even in programming or verifying operations, the programmer is capable of selecting an address to be programmed. If [Shift][A] is selected, an All memory address will be programmed.

**[Shift][E]** 16 Bit EVEN address  
If [Shift][E] is to be selected, an even memory address will be programmed.

**[Shift][O]** 16 Bit ODD address  
If [Shift][O] is selected, an odd memory address will be programmed.



**[Ctrl][O]** System driver I/O decode.02C0h (HEX)  
 [0] 200-2FFh and [2] 300-3FFh are 2 sets of decode base option in LEAP-EP1 remote control card.

The drive I/O decode of LEAP-EP1 is permanently stationed in 02C0h. If I/O decode is not correctly selected, screen will display SYSTEM ERROR: I/O drive not found!

*MEMO*