

USER'S MANUAL

**E(E)PROM
PROGRAMMER**

E(E)PROM PROGRAMMER

TABLE OF CONTENTS

	page
Introduction -----	1
Function Description -----	5
Setup Program -----	14
SYSTEM ADAPTER card I/O address selection -----	15
Troubleshooting -----	16
Appendix -----	17

SECTION 1

INTRODUCTION

The following procedures are described for proper application of the package on your personal computer:

A. PACKAGING

This package contains the following parts:

1. User's manual.
2. UTILITY disk.
3. SYSTEM ADAPTER.
4. Connecting cable.
5. PROGRAMMER unit.

B. SOFTWARE INSTALLATION

The following steps will make the UTILITY disk capable of booting your computer:

1. Start your computer using the DOS disk.
2. Place the DOS disk in Driver B.

Place the UTILITY disk in Driver A.

3. At the A> prompt, type the INSTALL <cr> command.

The screen will display the following messages.

```
A>INSTALL
A>REM      * * * * *
A>REM      * Place the DOS DISK in Driver B.      *
A>REM      * Place the UTILITY DISK in Driver A.  *
A>REM      * Make sure the disks are loaded correctly. *
A>PAUSE    * * * * *
Strike a key when ready . . .
A>B:
B>SYS A:
System transferred
```

```
B>COPY B:COMMAND.COM A:
      1 File(s) copied
```

```
B>COPY B:DEBUG.COM A:
      1 File(s) copied
```

```
B>REM      * SOFTWARE INSTALLATION COMPLETED ! *
```

```
B>A:
```

4. The UTILITY disk is now ready for use.

* For Hard-disk system, just type the ASSIGN B=C <cr> command before software installation.

* The UTILITY disk we supplied contains the the following files:

1. EPP-xx.EXE -- the main program to be executed.

(xx stands for 01, 04 or 08 in different model)

2. EPP.DAT ----- to save the previous running data while EPP-xx.EXE terminated.

3. SETUP.EXE --- to modify the system data (system clock and I/O address) to match your system.

4. SETUP.DAT ---- to save the system data.

5. INSTALL.BAT -- to do SOFTWARE INSTALLATION.

6. AUTOEXEC.BAT - to help you to run the package a little more conveniently.

7. HEXOBJ.EXE -- to convert files in HEX format into BIN format.

8. READ.ME ---- to tell you some new information about this package.

* After SOFTWARE INSTALLATION COMMAND.COM, DEBUG.COM and DOS are copied into the UTILITY disk.

C. Make a copy of the UTILITY disk and keep it at a safety place.

D. RUNNING THE PACKAGE

a. Computer system and minimum memory capacity required:

1. IBM PC XT/AT or compatible system with 256K RAM

(If your chips are 1024K type and you use DEBUG under EPP-xx.EXE then you need at least 384K RAM)

2. PC DOS V2.0 (or later) or compatible DOS.

b. HARDWARE INSTALLATION

1. Power off your computer.

2. Plug the SYSTEM ADAPTER card into any of the expansion slot.

3. Put the PROGRAMMER unit at a convenient position.

4. Connect the SYSTEM ADAPTER and the PROGRAMMER unit with the connecting cable.

c. Boot your computer system with the UTILITY disk.

d. Execute the EPP-xx.EXE program (or run by AUTOEXEC.BAT).

Then the menu will be displayed on the screen.

E(E)PROM PROGRAMMER V1.0

* MFG.: * ZIP.:
* TYP.: * PROG.:
* Vpp.: * VCC.:

MAIN MENU :

- 1. DIR
2. LOAD OBJ FILE TO MEMORY BUFFER
3. SAVE MEMORY BUFFER TO DISK
4. DEBUG MEMORY BUFFER
5. GANG SIZE
6. PROGRAMMING ALGORITHM
7. SET MEMORY BUFFER SIZE
M. MANUFACTURER
T. TYPE
B. BLANK CHECK
P. PROGRAM A. AUTO
R. READ V. VERIFY
C. COMPARE D. DISPLAY & EDIT
Q. QUIT

SELECT WHICH NUMBER ?

REMARKS: *:items that may be changed by user.
#:items that are changed automatically.

NOTE : If "Error Identification !" displayed, it means that

- 1.the UTILITY software may be not authorized.
- 2.the hardware may not be our original.
- 3.the hardware may not be plugged in or not be connected properly.

You may press <Q> to Quit and correct it or press <cr> to continue, but some functions may not work properly if you continue.

e.Select function M and T to change the manufacturer and type number to match your chips.

f.Put your chips onto the ZIP sockets while the BUSY LED is off.

g.Select working functions to do your job.

h.After finishing your job, press <Q> under main menu to quit from EPP-xx.EXE and return to DOS.

NOTE : The running data such as : MFG.,TYPE and OBJ file name will be saved to EPP.DAT on the UTILITY disk,after you quit from EPP-xx.EXE, and will be loaded into the system again next time you run the package

CAUTION: Before you put chips on or take chips out of the ZIP sockets,be sure that the BUSY LED is off.Otherwise the chips may be damaged.

SECTION II

FUNCTION DESCRIPTION

The function menu will be automatically displayed on the screen after executing EPP-xx.EXE. Details are described below.

Function 1 : DIR

To list the files directory of the disk.

The format is

```
[d:][path][filename[.ext]][/P][W]
```

or press <ESC> to return to main menu.

Function 2 : LOAD OBJ FILE INTO MEMORY BUFFER

```
LOAD :
Enter file name to be loaded :
TEST.OBJ
Enter buffer starting address :
0000
Loading now...
Ok !

Press any key to continue.
```

To load data from disk file into memory buffer. You may specify the file name and the buffer starting address to be loaded or press <ESC> to return to main menu.

Function 3 : SAVE MEMORY BUFFER TO DISK

To save data from memory buffer into disk file. You may specify the file name, the buffer starting and end address to be

saved or press <ESC> to return to main menu.

```
SAVE :
Enter file name to be saved :
TEST.OBJ
Enter buffer starting address :
0000
Enter buffer end address :
01FF
Saving now...

FILE EXISTED! OVERRIDE(Y/N)?Y
Ok !
Press any key to continue.
```

Function 4 : DEBUG MEMORY BUFFER

To debug and modify your memory buffer. This function calls the DOS DEBUG.COM file, so you just follow the DOS DEBUG command.

Example 1 :

Memory buffer starting address at 203F:0000

```
-----
-RDS
DS 2B35
:203F
-D 0000,001F
203F:0000 AA 3D 4A 62 74 66 22 13-45 29 00 87 88 80 33 DA
203F:0010 DC DF FF FF FF FF FF FF-00 00 00 00 00 00 00
-E 0000
203F:0000 AA.23 3D.45 4A.67 62.

For memory buffer 0-64K,
key in "starting address".
```

Example 2 :

Memory buffer starting address at 3105:0000

```
-----
-RDS
DS 4417
:3105
-D 0000,001F
3105:0000 66 55 44 88 DE DF CD C5-C3 3A 8D 5E EC EE FF AA
3105:0010 44 33 00 00 00 00 00 00-00 00 00 00 00 00 00
-E 0000
3105:0000 66.33 55.5D 44.6C

For memory buffer 64K-128K,
key in "starting address + 1000H".
```

Function 5 : GANG SIZE

```
----- GANG SIZE : -----  
  
1 : 1 socket  
2 : 2 sockets  
3 : 3 sockets  
4 : 4 sockets  
5 : 5 sockets  
6 : 6 sockets  
7 : 7 sockets  
8 : 8 sockets  
  
<ESC> back to main menu.  
  
SELECT NUMBER ?
```

To select the number of chips in a working batch.

Function 6 : PROGRAMMING ALGORITHM

```
----- PROG. ALGORITHM : -----  
  
1 : normal -- 50 ms  
2 : normal -- 10 ms  
3 : normal -- 5 ms  
4 : intelligent -- 1 ms  
5 : interactive -- 0.5 ms  
6 : quick-pulse -- 0.1 ms  
  
<ESC> back to main menu.  
  
SELECT NUMBER ?
```

The programming algorithm is automatically selected when you change the MFG. or TYPE but you can also change it by yourself.

Function M : MANUFACTURER

```
----- MANUFACTURER : -----  
  
1 : DON'T CARE   A : Dki  
2 : AMD          B : Richo  
3 : Exel        C : Rockwell  
4 : Fujitsu     D : Seeq  
5 : Hitachi     E : SGS  
6 : Intel       F : Signetic  
7 : Mitsubishi G : TI  
8 : NS          H : Toshiba  
9 : NEC         I : UMC  
                J : VLSI  
                K : Xicor  
  
<SPACE BAR> select type.  
<ESC> back to main menu.  
SELECT NUMBER ?
```

To select the manufacturer of the chips.

After changing the MFG., the TYP., PROG. (programming algorithm), Vpp and Vcc will become blank. You can press <SPACE BAR> to select type or press <ESC> to return to main menu. But if you do not select types after selecting MFG, some functions (such as READ, PROGRAM...) will prompt you to select types.

Function T : TYPE

To select the type number of the chips of the selected manufacturer.

When you change the type number the programming algorithm, Vpp and Vcc will also be changed according to MFG. and TYPE.

Function R :READ

```
READ :  
-----  
CHIP STARTING ADR: 0000  
CHIP END ADR: 1FFF  
BUFFER STARTING ADR: 0000  
BUFFER CHECK SUM: E000  
Ready (Yes/Even/Odd/C/<ESC>)?  
Reading .....  
Ok !
```

To read the data on the chip into the memory buffer specified by "CHIP STARTING ADDRESS", "CHIP END ADDRESS" & "BUFFER STARTING ADDRESS"

Press "Y" to read , "C" to change the addresses and <ESC> to return to main menu.

Function V :VERIFY

To verify the data in the chips ranged from "CHIP STARTING ADDRESS" to "CHIP END ADDRESS" with memory buffer, starting at "BUFFER STARTING ADDRESS"

The first error address of the chips will be displayed.

Press "Y" to verify, "C" to change the

addresses and <ESC> to return to main menu.

```
VERIFY :  
-----  
CHIP STARTING ADR: 0000  
CHIP END ADR: 01FF  
BUFFER STARTING ADR: 0000  
BUFFER CHECK SUM: FE00  
Ready (Yes/Even/Odd/C/<ESC>)?  
Verifying .....  
NO.1 -- OK  
NO.2 -- OK  
NO.3 -- OK  
NO.4 -- OK  
NO.5 -- OK  
NO.6 -- OK  
NO.7 -- OK  
NO.8 -- OK
```

Function C :COMPARE

```
COMPARE :  
-----  
CHIP STARTING ADR: 0000  
CHIP END ADR: 01FF  
BUFFER STARTING ADR: 0000  
BUFFER CHECK SUM: FE00  
Ready (Yes/Even/Odd/C/<ESC>)?  
Comparing .....  
Ok !
```

To compare the data in the chip ranged from "CHIP STARTING ADDRESS" to "CHIP END

TYPE :

```
1 : 2716B -12.5V A : 27128A -12.5V
2 : 27(C)16 -25V B : 27C128 -12.5V
3 : 27(C)32 -25V C : 27(C)256 -21V
4 : 2732A -21V D : 27256A -12.5V
5 : 2732B -12.5V E : 27C256 -12.5V
6 : 27(C)64 -21V F : 27512 -12.5V
7 : 2764A -12.5V G : 27C512 -12.5V
8 : 27C64 -12.5V H : 27010 -12.5V
9 : 27(C)128 -21V I : 27100 -12.5V
J : 2816A -5V
K : 2817A -5V
L : 2864A -5V
```

<ESC> back to main menu.
SELECT NUMBER ?

Function B : BLANK CHECK

BLANK CHECK :

```
CHIP STARTING ADR: 0000
CHIP END ADR: 1FFF
BUFFER STARTING ADR: 0000
BUFFER CHECK SUM: 0000
```

Ready to check (Y/C/<ESC>)?
Blank checking

```
NO.1 -- OK
NO.2 -- OK
NO.3 -- OK
NO.4 -- OK
NO.5 -- OK
NO.6 -- OK
NO.7 -- OK
NO.8 -- OK
```

To check whether the chips are blank (i.e. no data) or not. The first address that is not blank will be displayed.

Press "Y" to check, "C" to change the

addresses and <ESC> to return to main menu.

Function P : PROGRAM

PROGRAM :

```
CHIP STARTING ADR: 0000
CHIP END ADR: 03FF
BUFFER STARTING ADR: 0000
BUFFER CHECK SUM: FC00
```

Ready (Yes/Even/Odd/C/<ESC>)?

Programming

Verifying

```
NO.1 -- OK
NO.2 -- OK
NO.3 -- OK
NO.4 -- OK
NO.5 -- OK
NO.6 -- OK
NO.7 -- OK
NO.8 -- OK
```

To program the chip ranged from "CHIP STARTING ADDRESS" to "CHIP END ADDRESS" from the data in the memory buffer, starting at "BUFFER STARTING ADDRESS".

It will VERIFY automatically after finishing program.

Press "Y" to program, "C" to change the addresses and <ESC> to return to main menu.

Press "O" to program only Odd byte, "E" to program only Even byte.

Function A : AUTO

To do Function B + Function P by pressing just one key.

SECTION III

ADDRESS" with the data in the memory buffer ,starting at "BUFFER STARTING ADDRESS" .

Press "Y" to compare,"C" to change the addresses and <ESC> to return to main menu.

The differences are displayed in the format:
CHIP ADDR:CHIP DATA-(BUFFER ADDR:BUFFER DATA)

Error at :
Press <ESC> to terminate display.

0000:FF-(0000:23) 0001:FF-(0001:45) 0002:FF-(0002:67)
0004:FF-(0004:4D) 0005:FF-(0005:4E) 0006:FF-(0006:43)
0008:FF-(0008:CD) 0009:FF-(0009:CE) 000A:FF-(000A:AA)

Press any key to back to main menu

While displaying the differences, you can press <CTRL-S> to hold the display or press <ESC> to terminate the display, then press any key to continue.

Function 7 :SET MEMORY BUFFER SIZE

To open memory buffer size, you may open 64k or 128k (for 128k*8 EPROM).

Function D :DISPLAY & EDIT

To display and edit memory buffer directly.

SELECT WHICH NUMBER ?m

Under main menu selection, Press <Q> to return to DOS.

SETUP PROGRAM

When you use this package at first time or when you change your computer system or when you want to change the I/O address of the SYSTEM ADAPTER,, you may use this program to modify the system data in an easy way.

1.Key in SETUP <cr> under DOS.

```

      SETUP PROGRAM
-----
CURRENT I/O ADDRESS IS : 280H PIN 5
CURRENT OSC.FREQUENCY IS : 4.77M

(A) CHANGE I/O ADDRESS.
(F) CHANGE OSC. FREQUENCY.
<ESC> EXIT.
      WHICH ONE ?
  
```

2.Select "A" to change the I/O address or select "F" to change the OSC. frequency.

```

      CHANGE I/O ADDRESS
-----
JUMPER IN WHICH POSITION (1-8) :
  
```

```

      CHANGE OSC.FREQ
-----
1.XT/4.77M  2.XT/6M  3.XT/8M  4.XT/10M
5.AT/6M    6.AT/8M  7.AT/10M  8.AT/12M
      SELECT :
  
```

3.Make sure that the system data displayed matches your system, then press <ESC> to return to DOS. The system data will be saved to the SETUP.DAT file automatically.

SECTION V

SYSTEM ADAPTER Card I/O address selection

This card may be plugged into any slot of the computer. The I/O address of the card has been pre-selected 280H in our factory. Some problems may happen while other card plugged on the computer has the same address as this card. The way to solve this problem is shown below.

step 1 : Power off your computer.

step 2 : Change the 8-position JUMPER to new I/O address. The I/O address map is as below.

JUMPER position	I/O address
1	200H
2	220H
3	240H
4	260H
5	280H
6	2A0H
7	2C0H
8	2E0H

step 3 : Plug the card into the computer again, execute the SETUP.EXE program, and enter new JUMPER position.

step 4 : Return to DOS by pressing <ESC>.

SECTION V

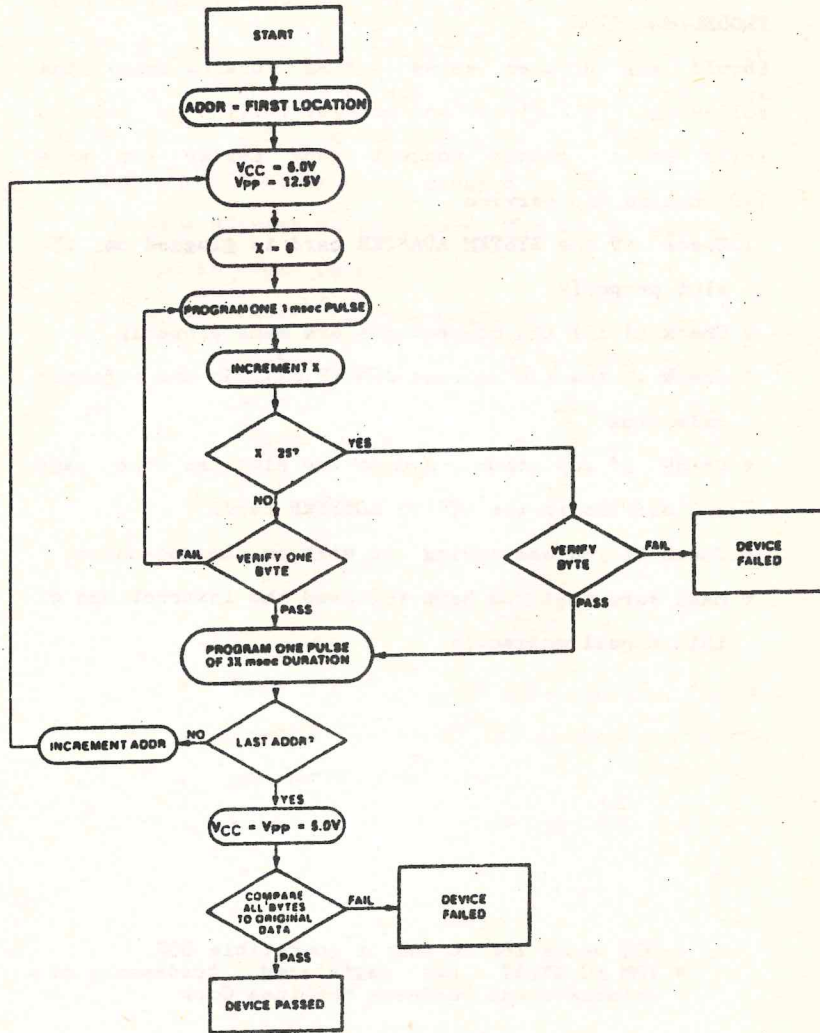
TROUBLESHOOTING

Should any problem arise during working, check the following list first and correct it. If the problem still exists, please contact your dealer for more information and service.

1. Check if the SYSTEM ADAPTER card is plugged on the slot properly.
2. Check if all the connections are made properly.
3. Check if the I/O address JUMPER matches the software selection.
4. Check if any other card on the slot has the same I/O address as the SYSTEM ADAPTER card.
5. Check if you are trying to work on damaged chips.
6. Make sure that you have followed the instructions of this manual correctly.

* DOS means IBM PC DOS or compatible DOS.
* IBM, PC, XT, AT are registered trademarks of International Business Machines Corp.

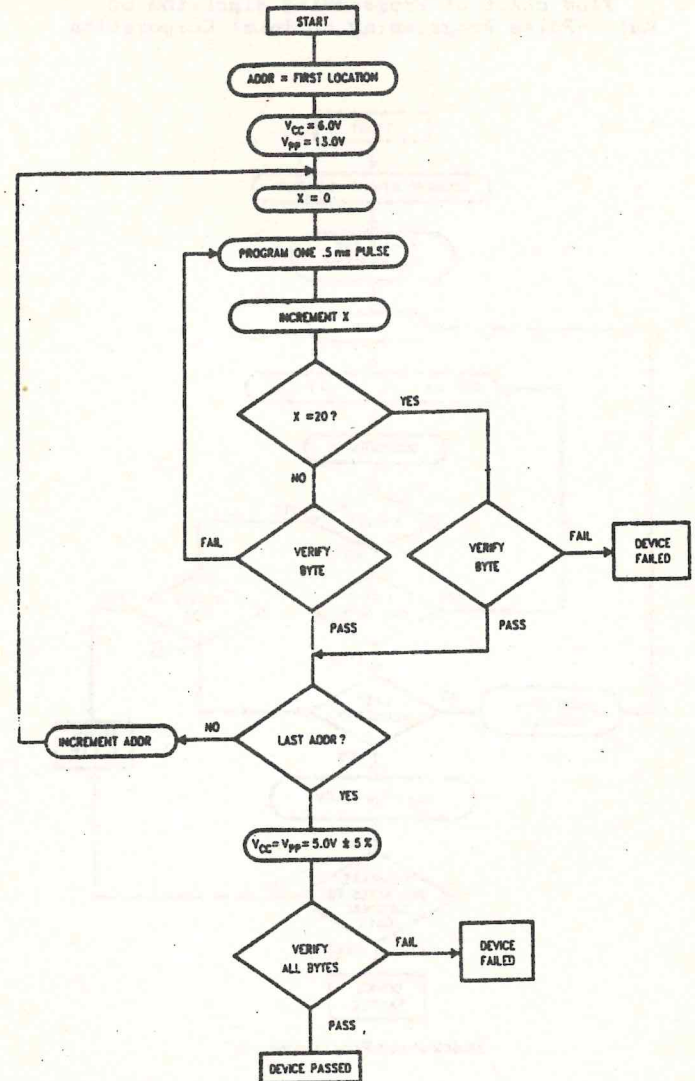
Flow chart of Programming algorithm of Intelligent Programming of Intel Corporation.



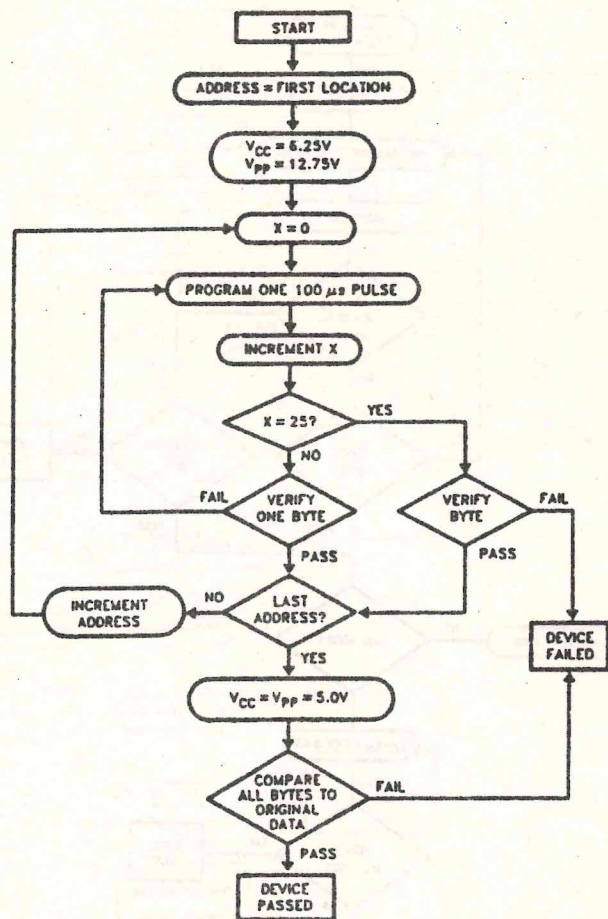
intelligent Programming™

* Intelligent Programming is a trademark of Intel Corporation.

Flow chart of Programming algorithm of Interactive Programming of National semiconductor Corporation.



Flow chart of Programming algorithm of Quick-Pulse Programming of Intel Corporation.



Quick-Pulse Programming™

* Quick-Pulse Programming is a trademark of Intel Corporation.