ELAN DIGITAL SYSTEMS LTD.

EF-PER™ SERIES PROGRAMMERS
MODEL 3000 USER MANUAL

MODELS 3-132, 3-142, 3-232, 3-832,
3-840, 3-154, 3-187, 3-145

The 3000 PC Remote Disk is stored at the back of this Manual.

WESTERN EUROPE
Elan Digital Systems Ltd.
Elan House, Little Park Farm Road
Segensworth West, Fareham, Hants.
PO15 5SJ
U.K.
Tel: (01489) 579799
Fax: (01489) 577516

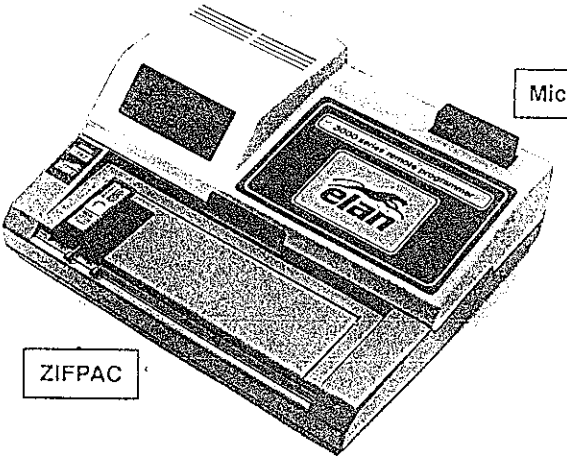
NORTH AMERICA
Elan Systems/Ascend
1328, Concannon Blvd.,
Livermore.
CA 94550. U.S.A.
Tel: 001 510 606 2000
(800) 541 ELAN
FAX: 001 510 606 2006

REL: 11/01/95 SW: EF7.02 REMOTE DRIVER REL:X5.00

CONTENTS

CHAPTER 1,

3000 Series Remote Programmer



3000 SERIES OVERVIEW

3000 Series Programmers are PC controlled with Elan REMOTE DRIVER software supplied on disk with this manual. (An RS232 serial I/O cable is optionally available from Elan, order code 004061).

The Elan REMOTE DRIVER software converts standard file formats to ELAN BINARY FORMAT (EBF). This allows very fast data transfer rates with the minimum overhead whilst maintaining data integrity with block transfer checksums.

Programming is concurrent with data transfer giving exceptionally fast transfer/programming times. eg. 64K bytes of data transferred and programmed to a 27512 device in less that 37.5 seconds @ 19,200 baud.

Elan's Fast Programming Enhancement Routine "EF-PER" considerably reduces the overall programming time by a unique combination of hardware and software design techniques. It will be noticed in particular when blank checking large devices, where a conventional programmer may take at least 14 seconds to blank check a 27512 device, the Model 3000 Programmers will check any size of device in less than 1 second.

The speed of the 3000 programmers compliments device manufacturers' technological achievements in reducing programming times although memory size are doubling in size.

The Model 3000 Programmers have a plug-in firmware Micro Cartridge to allow software enhancements and extended device coverage for the future.

The Model 3000 Programmers have the facility to extend the range and number of devices programmed, by the insertion of user selected ZIFPACs giving extensive coverage from 24 pin to 40 pin devices in DIL. With Adapters other surface mount packages are supported.

During power-up the programmer runs a comprehensive self-check routine and ensures all power rails are steady and accurately set. The unit continually monitors for over-current and voltage tolerance on internal supplies, Vcc and Vpp when applied to the ZIFPAC. Address and data lines are also continually monitored during program, verify, blank & bit test functions to maintain programmer and device integrity. (ZIFPAC's are described in Chapter 3.)

3000 REMOTE DRIVER Automatic Serial Configuration:- Baud Rates; 9,600 *or* 19,200.

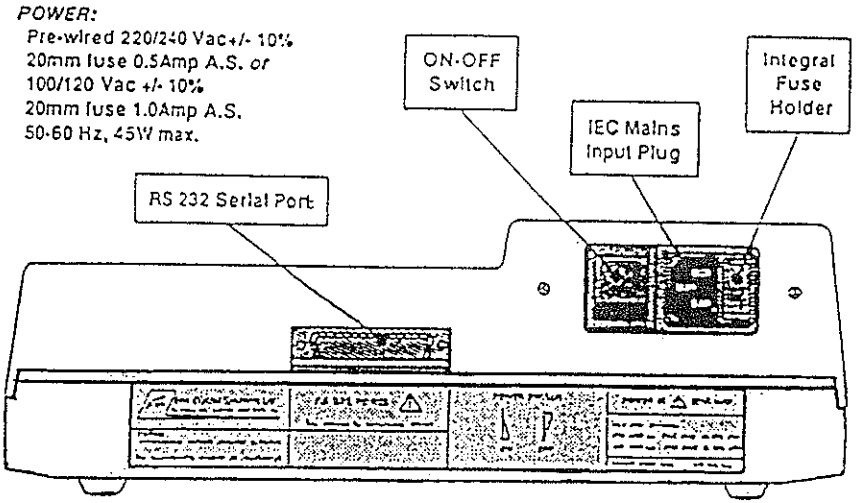
It should be noted that currently available versions of DOS do not permit selection of baud rates in excess of 9,600. As some computers are known to be capable of faster transfer rates REMOTE DRIVER runs a benchmark test to determine if a baud rate of 19,200 will support reliable data transfer and will set the serial port accordingly.

Manual selection: 38,400 baud

A manual selection of 38,400 baud will be allowed by the software subject to a handshake speed test between the serial ports. This selection is not guaranteed but reliable transfer can be determined by the user of IBM PC or compatible XT PCs.

INSTALLATION

Connect serial communciations cable between programmerRS232 25 way D socket and IBM or compatible Com 1 or Com 2 plug. (Available from Elan, Part No. 004061.)



Rear View of Programmer

Interface:Serial I/O RS232 connections;

(all 25 ways can be connected pin to pin e.g. ribbon cable with IDC connectors)

IBM 9 way D type plug on PC	To	ELAN 25 way D type socket
1 N/C	.....	1 GROUND/EARTH
3 DATA OUTPUT	.....	2 DATA INPUT RX0 PORT A
2 DATA INPUT	.....	3 DATA OUTPUT TX0 PORT A
7 REQUEST TO SEND	.....	4 CLEAR TO SEND INPUT CTS0
8 CLEAR TO SEND	.....	5 READY TO SEND OUTPUT RTS0
6 DATA SET READY	.....	6 INTERNAL LINK TO PIN 20
5 SIGNAL GROUND	.....	7 Ov
	.....	

(N/C: 10, 12, 13,
14, 15, 16, 17,
19, 21, 23, 24)

If DSR awaits DTR join 4 to 6 on this connector.

Convert 9 way to 25 way with converter cable available from IBM dealers, Part No. 0745 764.

Additional Information

10 -5v
12 Vpp
13 +5v
14 AUXILIARY DATA INPUT RX1 PORT B
16 AUXILIARY DATA OUTPUT TX1 PORT B
20 LINK TO PIN 6
25 GROUND/EARTH
(N/C: 8, 9, 11, 15, 17, 18, 19, 21, 22, 23, 24)

IBM 25 way D type plug on PC	To	ELAN 25 way D type socket
1 N/C	.....	1 GROUND/EARTH
2 DATA OUTPUT	.....	2 DATA INPUT RX0 PORT A
3 DATA INPUT	.....	3 DATA OUTPUT TX0 PORT A
4 REQUEST TO SEND	.....	4 CLEAR TO SEND INPUT CTS0
5 CLEAR TO SEND	.....	5 READY TO SEND OUTPUT RTS0
6 DATA SET READY	.....	6 LINK TO PIN 20
7 SIGNAL GROUND	.....	7 Ov
	.....	

(N/C: 10, 12, 13,
14, 15, 16, 17,
19, 21, 23, 24)

Additional Information

10 -5v
12 Vpp
13 +5v
14 AUXILIARY DATA INPUT RX1 PORT B
16 AUXILIARY DATA OUTPUT TX1 PORT B
20 INTERNAL LINK TO PIN 6
25 GROUND/EARTH
(N/C: 8, 9, 11, 15, 17, 18, 19, 21, 22, 23, 24)

Installing 3000 REMOTE DRIVER on your PC:

To install Elan REMOTE DRIVER on your PC, the following steps must be undertaken.

1) Create a subdirectory called "ELAN3000".

2) Copy the master diskette supplied into the new directory.

e.g. Type COPY A:*.* C:\ELAN3000

Because the batch facility requires the use of your own text editor the DOS environment will require setting. Follow the instructions below to perform this.

3) Using an editor or DOS command EDLIN modify your AUTOEXEC.BAT file in BATCH EDIT. Add at the end of the file the DOS environment variable "EBTEDIT", this should include the complete pathname and extensions to your own editor.

e.g. If your editor is Wordstar and resides in the directory WS4 type:-

SET EBTEDIT=C:\WS4\WS.EXE

When the above is completed you may then run Elan REMOTE DRIVER.

Note: If you get the message "out of environment space" when the machine is "booting up", increase the environment size by using the /E option in the shell specification. This is in your config.sys file.

e.g. SHELL=C:\COMMAND.COM /E:300 /P

Selecting ELAN3000 on your PC:-

To run Elan 3000 REMOTE DRIVER software on your PC simply change to the ELAN3000 directory

e.g. Type CD\ELAN3000 "ENTER"

On Screen *C:\ELAN3000*

Type \3000 "ENTER"

or from Start Up

C:>\ELAN3000\3000\ "ENTER

(Enter = Return)

ELAN3000 will automatically configure to the 3000 series programmer and present a choice of five pull down menus on the following pages.

Note: Type "3000/BW" to display in monochrome only. This may be useful for portable machines using Plasma or L.C.D. displays.

CHAPTER 2

PRESS ON PC FUNCTION KEY F1 FOR HELP AT ANY STAGE

THE 3000 REMOTE DRIVER SOFTWARE PULL-DOWN MENUS.

INTRODUCTION and PREPARATION.

Connect the serial communication cable between the 3000 SERIES programmer and IBM PC or compatible, COM1 or COM2 socket. Ensure correct connections.

Install ZIFPAC and corresponding FW Cartridge.

For Memory devices(PROM), one of -132 -142 -832 -840 -PLC ZIFPAC's are to be installed with cartridge type E5.xx.

For Micro, INTEL architecture, devices one of -154 -187 ZIFPAC's are to be installed with cartridge type U1.xx.

For Logic devices, install the -145 ZIFPAC provided the 3000 MCU, main unit, is equipped with 3000- PLD RAM enhancement and PC driver.

Switch on the programmer and run the ELAN 3000 REMOTE DRIVER by typing in *" \3000 "* and press RETURN. With the system installed and correctly configured, the main menu is displayed on the controlling PC screen as detailed in the next section.

3000 REMOTE DRIVER PULL-DOWN MENU.

There are five sections in the main menu. *PROGRAM, FILES, SETUP, COMMS, BATCH.*

On power-up the *PROGRAM* menu is shown.

The horizontal cursor keys cycle to each of the other main sections of the menu. The vertical cursor keys cycle to sections within the menu. The ENTER key enters the selection made.

PROCEDURES

Parameters need to be set in the *SETUP* and *COMMS* menus, to define target device and communication selections.

The selections in the *COMMS* menu are usually set by the system.

The programming data files need to be converted, and may be edited, in the *FILE* menu, in preparation for programming.

In the *PROGRAM* menu, target devices are programmed from selected program data files, nnnn.EBF.

For volume programming select the *BATCH* menu.

SELECTIONS FROM THE COMMS MENU

SELECT PORT: This displays the systems communication port in use and allows manual selections to be made.

BAUD RATE: Displays the selected baud rate and allows for other manual selections as displayed. 38400 - 150.

PARITY: Displays parity selection in use and other selections available for the communication words - i.e. ***NONE, ODD*** or ***EVEN PARITY.***

STOP BITS: Displays selection in use and other manual selection of ONE or TWO STOP BITS in the communication routine protocol.

RESTART: This enables a forced RESTART of communication of the ELAN REMOTE DRIVER software with the programmer.

FILE I/O FORMAT: See the section for the programming of logic devices.

The **ENTER** key on the PC keyboard enters the selection.

SELECTIONS FROM THE SETUP MENU

PRESS ON PC FUNCTION KEY F1 FOR HELP AT ANY STAGE

SELECT DEVICE: The whole range of manufacturers and device selections are uploaded from the programmer and displayed on the screen of the PC.

The horizontal cursor keys enable the selection of device manufacturers and the vertical keys enable the selection of devices to be made.

The range of devices available are subject to the type of ZIFPAC and FW cartridge installed.

WORD SIZE: The programming data file (EBF) may be set up for 8 bit, 16 bit or 32 bit number in the binary word size of the program.

Take care to ensure the correct size for the device and file in preparation for programming.

BYTE ORDER: For the programming of devices in 16 bit wordsize, the byte order may be selected to suit.

- LS- MS byte order is used with INTEL BASED SYSTEMS.
- MS- LS byte order is used with MOTOROLA BASED SYSTEMS.

When the object file is converted to EBF file format, the byte order is set. If on subsequent PROGRAMMING, VERIFY and BIT-TEST from an EBF file, the order is changed, then the settings in the EBF file are valid.

The default in LEAST SIGNIFICANT - MOST SIGNIFICANT order, INTEL SYSTEM format and the reverse, MOTOROLA SYSTEM order can be selected.

Note that with the -132 and -142 ZIFPAC, 16 and 32 bit wordsize programs, can be set out over two and four devices, each in defined order for programming. The system automatically takes care of the programming order.

NUMBER OF BLOCKS: All fitted programming sockets can be used for programming devices. Incorrect choices made are reset automatically.

See description of SET'S, WORD SIZE and BLOCKS in this manual.

ALTERNATIVE SOCKETS: This selection is used for programming in gang, when large adapters are located in every second site.

ELECTRICAL ID: When this section is made to ON, the electrical ID code of the device inserted in the sockets is compared with the ID code in the current programming data file. When this selection is in the OFF state no comparison is made.

AUTO TEST VECTORS: See section for the programming of logic devices.

SELECTIONS FROM THE FILES MENU

DIRECTORY: This option allows the search for object and other files by their file extensions in this directory.
i.e. *.obj or *.ebf and with *.* give a full listing of all files in this directory.

DOS SHELL: This option allows DOS commands to be executed and other programs to be run. To RETURN to the MAIN menu type *EXIT*. It is advised not to run software with access to ports from this shell.

NOTE: All existing and new object data files must be converted to ELAN Binary Format for transfer to the programmer and for the programming target devices.

CONVERT TO EBF (ELAN Binary Format): This function allows the conversion of object files into EBF file format for programming. Note that EBF format files are device linked and only applies to PROM and MICRO CONTROLLER devices.

A prompt allows a choice between *HEX* and *BIN*ary object files to be converted, followed by the name of the target EBF file used for the programming of the target devices.

Object file format may be
- INTELHEX or EXTENDED INTEL HEX.
- MOTOROLA S RECORD.
- TEKTRONIC STANDARD TEKHEX.
- TEKTRONIC EXTENDED TEKHEX.
- Others on request.
- BINARY FILE WITHOUT HEADERS.

A further prompt requests for an offset. This means that conversion of object data file to EBF format can be from current starting address 0, or any later address in the object file. If an address other than the start address is specified, the result of file conversion shows a note regarding data in the file before the specified address, and if appropriate, data extending beyond the device range selected. This is for information only.
This facility allows large files to be split up for programming into any required devices, (select the required device parameters prior to file conversion).

In other words, a SET is formed for programming.

If with an 8 bit data file, the data starts at address 80H and the base offset selected is 50 (reduction) the start address is then 30H. If the file is converted to 16 bit wordsize, then start address is at 40H with base offset reduced to 28, thus the start address is now at 18H. If the the file is converted to 32 bit word, then the start point is at 24 with base offset reduced to 14, and the file then starts at 0CH.

During file conversion the percentage completed of the file is displayed in the top right hand side of the screen.

CONVERT FROM EBF (ELAN Binary Format): This function allows data read in from PROM devices on ZIFPAC sockets to be converted to non device dependent object data files. EBF files may be converted to the following formats:

- INTEL HEX or EXTENDED INTEL HEX
- MOTOROLA S-RECORD
- TEKTRONIX STANDARD TEKHEX
- TELTRONIX EXTENDED TEKHEX
- BINARY (no header)

(Other Formats maybe available on request. Please contact your local supplier.)

EDIT EBF FILE: A prompt for the file name in this menu allows any Elan Binary Format (EBF) content to be edited on the screen. *See separate page on this topic.*

When this option and the particular file is selected, the file content is shown on the screen, in the HEX code and ASCII coded translation. File content can be changed at will, and the changes can be saved or abandoned. Upon selecting to save the changes, the file can be saved under its original name or a different specified name. Initiate with **Alt.** and **X** keys.

It is thus possible to take a basic EBF file and make some alterations and store it under a new name, repeatable at will.

SAVE CONFIGURATIONS: The current screen parameters and device details may be saved in a file called "ELAN.ECF" or any other named file. These files are automatically saved with the ".ECF" extension.

LOAD CONFIGURATIONS: Any previous saved configuration file may be recalled and automatically restores the saved screen configuration.

QUIT: This controls the exit to the DOS domain.

SELECTIONS FROM THE PROGRAMMING MENU

PROGRAM:
In this menu programming is controlled from a selected EBF, Elan Binary File. This selected file may be the current or any other in this directory of EBF programming files.

The EBF programming file may be derived from an existing object file converted for programming, or read in from a master device, or devices, in to a named EBF file.

Blank check before programming of devices form part of the routine.

BLANK CHECK: Devices(s) are blank checked in less than one second.

VERIFY: The programmed device(s) in the ZIFPAC are verified against the programming file for this device.

BIT-TEST: Devices will be bit checked against an the EBF file to show any PROM bits is set to 0 and the file defines a 1.

READ: Devices are read to a named EBF file, current or new. *For logic devices, see separate section.*

INSERTION CHECK: Connection to devices in the sockets are checked. *For logic devices, see separate section.*

SPECIAL: In this section additional options are available namely:

EXIT TO SPECIAL FUNCTION MENU: This exits to main menu.

ERASE: This option permits the erasure of certain EEPROMS and FLASH EPROM devices. The programmer checks that after erasure the device is correctly erased.

SECURE: *See logic and micro section for this option.*

DISABLE PROTECTION: This de-activates the Software Write Protection feature available on some EEPROM devices.

All the remaining selections in this menu apply to the logic device section of this manual.

PRODUCTION PROGRAMMING WITH BATCH FILE COMMAND

PRESS ON PC FUNCTION KEY F1 FOR HELP AT ANY STAGE

SECTION A FOR ELAN 3-132, -142, -232, -832, - 840, 3-plc/1, 2

This User Manual has described what the required commands are for programming a PROM device.

To program devices for production, with variations in type, size and data files, the number of required commands is reduced with BATCHFILE.

In a BATCHFILE, all the required instructions for programming selected target devices are written.

BATCHFILE is operated by simply selecting the named BATCHFILE.

HOW TO RUN A BATCHFILE

In ELAN REMOTE DRIVER software select the BATCHFILE window.

In BATCHFILE window, select RUN.

In RUN, select the BATCHFILE required.

Enter this selection provided the programmer is in ELAN REMOTE DRIVER software and in communication with the programmer.

Follow instructions on screen of the PC indicating when to insert devices in the ZIF-sockets of the programmer.

HOW TO CREATE A BATCHFILE

In ELAN REMOTE DRIVER software, select the BATCHFILE window.

In BATCHFILE window, select EDIT.

The window asks to select a master BATCHFILE *"TO EDIT"* or create a new BATCHFILE with *"OTHER FILE"*.

New BATCHFILEs are written in the Text Editor, when selected. (e.g. WS)

Note: Refer to installation section Chapter 1, for setting up the link to the text editor in AUTOEXEC.BAT file in the host pc.

The content of this text file uses specific, all typed in words, in a defined order, complemented by the specific date required for the programmer to follow.

BATCHFILE can instruct the Programmer to *READ* a master device and store data into a target file, for later use, or *PROGRAM* devices from named data files.

TO READ A DEVICE AND STORE IN NAMED FILE

Type in as follows (within <> insert selections):

DEVICE <FAMILY><DEVICE NUMBER>
NUMBLOCKS <1-8>
WORDSIZE <8-16-32>
READ <data into named file .EBF>
FROM_EBF <convert this file.EBF> <into named object file.OBJ> <format>

The data on the master device is now read into named EBF file, converted to named format and stored into named object file for later use.

TO PROGRAM A DEVICE FROM A NAMED OBJECT FILE

Type in as follows (within <> insert selections).

DEVICE <FAMILY><DEVICE NUMBER>
NUMBLOCKS <1-8
WORDSIZE <8-16-32>
TO_EBF <convert named object file to EBF file> HEX YES FF O
PROGRAM YOURFILE.EBF

NOTES:

- 1. Number of blocks used is subject to word size selected. e.g. word size 8 max. number of blocks possible is 8. e.g. word size 16 max. number of blocks possible is 4. The number of blocks selected is subject to the number of sockets installed on the selected ZIFPAC.
- 2. When converting to EBF format just before programming, the statement reads, "to convert to EBF format this HEX format file and fill empty spaces in the target device with FF and start reading from address 0 or other".
- 3. Object files intended for large size devices can be converted to multiples of smaller devices by specifying smaller size devices and increasing the number of blocks used. The same is true in the other direction by increasing the size of selected target device and reducing the number of blocks required when converting to EBF format for programming. It is useful to note that object files are independent of device selection.
- 4. HEX object files may be in:
  IXH - INTEL HEX
  EHX - EXTENDED INTEL HEX
  MS - MOTOROLA S RECORD
  TKS - TEKTRONICS STANDARD TEKHEX
  TKE - TEKTRONICS EXTENDED TEKHEX
  BIN - BINARY
- 5. When converting object file to EBF format and the object file is not HEX but BINARY format, then instead of HEX state BIN for both reading and programming.
- 6. As part of the programming routine in BATCHFILE all target devices are blank checked before programming.
- 7. ERASE for EEPROM devices can be included before programming a device.
- 8. VERIFY file.EBF and BITTEST file.EBF can be added to the BATCHFILE if required, but verification is carried out with programming routine.
- 9. IDON and IOFF enables and disables the electrical identification of the device.
- 10. The BATCHFILE shown below can also be called on the PC, in ELAN REMOTE DRIVER directory with \3000/F ELAN-A.EBT
- 11. When the BATCHFILE has been completed, save the file in text editor and exit to MAIN MENU.

PROGRAM SAMPLE BATCHFILE

ELAN-A.EBT

```
DEVICE AMD AM27C010
NUMBLOCKS 1
WORDSIZE 8
READ ELAN-B.EBF
FROM_EBF ELAN-B.EBF ELAN_C.OBJ MS
DEVICE AMD AM27C010
NUMBLOCKS 1
WORDSIZE 8
TO_EBF ELAN-C.OBJ ELAN-D.EBF HEX YES FF 0
PROGRAM ELAN-D.EBF
DEVICE INTEL IN27128
NUMBLOCKS 8
WORDSIZE 8
TO_EBF ELAN-C.OBJ ELAN-F.EBF HEX YES FFO
PROGRAM ELAN-F.EBF
DEVICE AMD AM27C010
NUMBLOCKS 2
WORDSIZE 8
READ ELAN-L.EBF
FROM_EBF ELAN-L.EBF ELAN-M.OBJ BIN
DEVICE INTEL INT27512
NUMBLOCKS 4
WORDSIZE 8
TO_EBF ELAN-M.OBJ ELAN-N.EBF BIN YES FF O
PROGRAM ELAN-N.EBF
:RUNA
PROGRAM ELAN-N.EBF
LOOP RUNA 8
```

SECTION B FOR ELAN 3-154, -187

This section is specifically for the programming of MICRO CONTROL DEVICES. *See separate section for basic data in this manual.*

For programming MICRO CONTROL DEVICES in BATCHFILE, the pattern is very similar to what has been shown before in Section A.

It is important that the required ZIFPAC is installed to the programmer.

The following sample program shows how to read from an unsecured master device and then how to program into a target device, ending proceedings with secure device.

EXAMPLE BATCHFILE PROGRAM FOR MICRODEVICES

```
DEVICE AMD AMD87C51
WORDSIZE 8
IDON
READ ELANMICR.EBF
FROM_EBF ELANMICR.EBF ELANMICR.OBJ MS
DEVICE AMD AMD87C51
WORDSIZE 8
IDON
TO_EBF ELANMICR.OBJ ELANMICR.EBF HEX YES FF O
PROGRAM ELANMICR.EBF
VERIFY ELANMICR.EBF
BITTEST ELANMICR.EBF
SECURE
```

NOTES:

- 1. Optional VERIFy and BITTEST has been added.
- 2. When the file has been created then **SECURE** and **EXIT** from the text editor back to MAIN MENU - BATCHFILE. Run for programming.

Chapter 3,

MEMORY ZIFPACS

ZIFPACs provide a choice of socket configuration to suit all programming requirements. They are available to order as separate items and can be easily interchanged by the user, see diagram overleaf.

The 3000 Series ZIFPACS for PROM devices are as follows:-

-132 A single 32 pin socket. Programming SETS. 16 and 32 bit word size. Programs 24, 28 and 32 pin EPROMs, Flash EPROMs and EEPROMs. (3000 Model 3-132).

-142 A 32 pin socket and a 40 pin socket (used separately). Programming SETS. 16 and 32 bit word size. Programs 24, 28, 32 and 40 pin EPROMs, Flash EPROMs and EE PROM. (3000 Model 3-142)

-232 Two 32 pin sockets programming SETS: 16 and 32 bit word size, Blocks or Gang. Programs 28 and 32 pin EPROMs, Flash EPROMs and EEPROMs. (3000 Model 3-232)

-832 Eight 32 pin sockets programming SETS: 32 bit and 16 bit word size Blocks or Gangs Programs 28 and 32 pin EPROMs, Flash EPROMs and EE PROMs. (3000 Model 3-832)

-840 Eight 40 pin sockets programming GANG of 40 pin EPROM.

-154 Twin 40 pin sockets for Intel/NEC 8748 88751 Series 8 bit devices.

-187 Two 40 pin and a single 48 pin socket for all Intel/NEC 8748-8751 to 87196 8 bit and 16 bit devices.

-145 A single 48 pin socket for EPLD/GAL/CMOS PAL and serial PROM devices

See separate section for microprocessors.

General device coverage; NMOS, CMOS, FLASH EPROM and EE PROM
See full listing for all manufacturers.

ZIFPACS

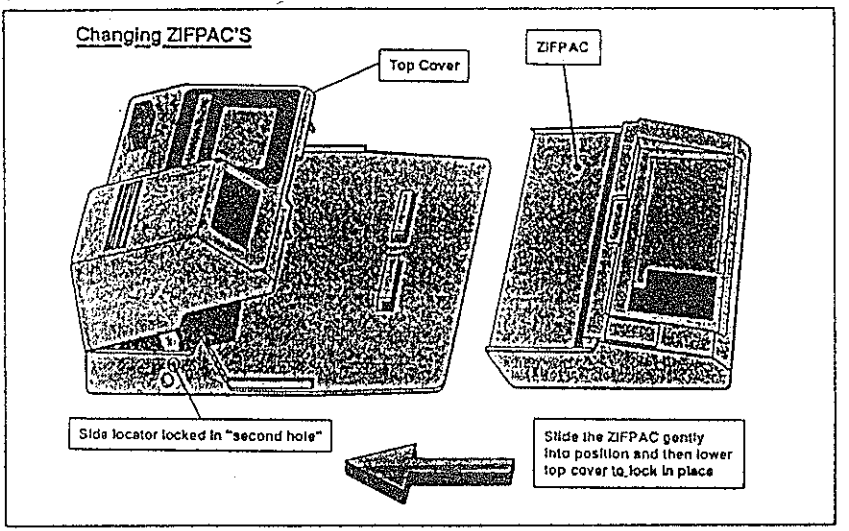
ZIFPACs are easily interchangeable as follows:

Switch off and withdraw the mains socket to isolate the programmer.

Depress the side locator while lifting the top cover and allow the locator to lock into the second hole. Do not open the unit beyond this point or the hinges may be strained. This action unplugs the 96 way ZIFPAC connector and allows the ZIFPAC to be withdrawn.

Slide in the alternative ZIFPAC and depress the slide locator to lower the front cover and allow it to be locked into place.

Re-apply power and the unit will self test and re-configure to the new ZIFPAC.



Zero Insertion Force (ZIF) Sockets

32 pin sockets; 32 pin devices are inserted with pin 1 at the top left hand corner. 28 pin devices must be in the lower part of the socket (bottom justified), with pin 1 uppermost similarly for 24 pin devices. The 40 pin socket is only suitable for 40 pin devices (model 3-142, & 3-840). The ZIFPAC sockets are only powered during a PROGRAM, READ, VERIFY, BIT or BLANK CHECK function. Devices must not be inserted or removed during these functions. Do not turn the programmer on or off when devices are fitted in the sockets.

Note: *The nature of ZIF sockets precludes any wiping action between the socket contacts and the EPROM pins, therefore it is important to ensure that the socket pins (and EPROM pins) are always clean to avoid poor contact.*

ZIF Socket LED's

RED LED(s):

Illuminated; These indicate an error. The display will show the nature of the error.

GREEN LED's;

Illuminated; Power applied to sockets.

Flashing; Indicate the correct socket(s) for the selected device.

SETS combinations with PROM Programming

SETS are combinations of WORDSIZE and BLOCKS. (See examples overleaf.)

If Wordsize and Blocks are not changed from their default settings of 8 and 1, then each copy device inserted will be programmed in Gang. For normal 8 bit word size, blocks are simply separate blocks of data in RAM programmed to each socket concurrently.

Thus, if two devices containing different data are read when using a SET selection comprising of 8 bit wordsize and BLOCKS set to two, a single binary file would be created containing the device data as two separate blocks, e.g. if the devices were 2764's, one block would be 0-1FFF and the second 2000-3FFF. Programming with this file would re-create these blocks of data into two further 2764's.

Selecting 16 bit wordsize (Blocks set to 1) automatically programs even bytes from RAM to socket 1 and odd bytes to socket 2. If 4, 6 or 8 devices were present in the ZIFPAC then the 16 bit pattern would be repeated in each pair.

For 40 pin 16 bit EPROMs the even bytes are programmed to D0-D7 (pins 12-18 inc.). odd bytes to D8-D15 (pins 3-10 inc.)

Selecting 16 bit wordsize with a number of blocks will allow separate 16 bit data patterns to be programmed to each pair of sockets.

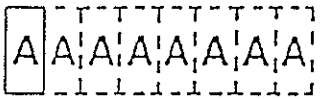
Selecting 32 bit wordsize automaticaly programs the appropriate RAM data to each of the first four ZIFPAC sockets. If 8 devices were present then the 32 bit pattern would be repeated in the second four sockets.

Selecting 32 bit word size with 2 blocks will allow a separate 32 pin data pattern to be programmed to the second four sockets.

Example of Sets comprising of Word Size and Blocks.

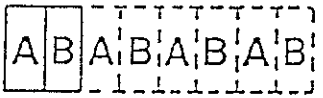
8 bit word size, 1 Block

(Dotted lines indicate how the sockets automatically configure as extra devices are inserted)



EPROM 2764	RAM ADDRESS Start	Finish
A	0000	1FFF

16 bit word size, 1 block



EPROM 2764		RAM ADDRESS
A	Even address bytes	0000 3FFF
B	Odd address bytes	

32 bit word size, 1 block

EPROM 2764	RAM ADDRESS 0000-7FFF
A	address bytes 0,4,8, etc.
B	address bytes 1,5,9,
C	address bytes 2,6,A,
D	address bytes 3,7,B,

Example of Sets comprising of Word Size and Blocks (con'l).

8 bit word size, 8 Blocks

EPROM 2764	RAM ADDRESS Start	Finish
A	0000	1FFF
B	2000	3FFF
C	4000	5FFF
D	6000	7FFF
E	8000	9FFF
F	A000	BFFF
G	C000	DFFF
H	E000	F000

16 bit word size, Block of 4

EPROM 2764	RAM ADDRESS
A even address bytes B odd address bytes	0000 3FFF
C even address bytes D odd address bytes	4000 7FFF
E even address bytes F odd address bytes	8000 BFFF
G even address bytes H odd address bytes	C000 FFFF

Examples given for a 2764 Device.

Total capacity 64K bits = 8Kbytes
Device addressed from 0000-1FFF hexadecimal

Chapter 4

MICROCONTROLLER ZIFPAC

MODELS -154 -187

The Microcontroller ZIFPAC family with the ELAN Model 3000 Programmer supports the INTEL ARCHITECTURE Microcontroller range of devices.

There are two variants, namely;

- The -154 ZIFPAC supports the 8748, 8751 families. 8 BIT devices.
- The-187 ZIFPAC supports the INTEL8748, 8751, 879x and 87196xx families. 8 bit and 16 bit devices.

As some microcontroller devices need pin conversion adapters, the device list with the manual indicate the adapter for programming when required.

GENERAL NOTES.

Each MICROCONTROLLER ZIFPAC is supplied with its supporting ELAN firmware cartridge. Upon installation of one of the ZIFPACs and its fw cartridge, the programmer will be configured for its supporting MICROCONTROLLER devices.

Programming procedures are generally as described for EPROM devices. With the Model 3000 programmer, remotely driven from a PC with ELAN PC Remote Driver, master devices can be read in and stored with other data in the PC workingdirectory. Stored programming data, may be in EBF format or as OBJECT FILES in various other formats.

Converted programming data (nnn.EBF) may be viewed and edited in the edit window of ELAN PC Remote software.

The menu gives the facility to the following options - to:

PROGRAM devices.
ID CHECK devices.
BLANK CHECK devices.
VERIFY program content in devices.
BITTEST to check state of all bits.
READ data from master devices.
EDIT programming file content.
BATCHFILE use of batch programming facility.

Device selection is not possible with a device in a socket. This to protect the device and the equipment.

Normal static handling procedures are to be adopted with the programming of these devices.

DO NOT POWER UP OR DOWN WITH DEVICE IN SOCKET.

Although the system is protected it is a recommended good practice to insert devices only when prompted on the screen.

LED FUNCTIONS:

GREEN LED: Illuminated. This indicates correct socket for device.

FLASHING GREEN LED: This indicates programmer active during READ, PROGRAM, VERIFY and BLANK functions.

FLASHING RED LED: This indicates an error condition.

FLASHING - ALL LEDS: If on power up, initialisation failure occurs, this is then indicated by all LEDs flashing in red and the buzzer sounds intermittently for 15 seconds.

VERIFY:

Programmed devices are verified to normal supply conditions.

DEVICE SECURITY PROTECT.

All current ZIFPACs for Microcontroller Devices have program security protection facilities. This is to prevent the program code inside the programmed Microcontroller Device being altered as well as unauthorised reading, software piracy of the program code.

Once the device has been programmed and verified, the device can be secured by the SECURE control within the ELAN PC Remote driver software. When the device has been secured, it is not possible to verify the device data, as the devices will read back as blank.

Attempting to program these devices at this point will result in a ***PROGRAM FAIL*** message.

A secured a device can only be released by full UV erasure.

Members of the 879x and 87c196xx families will fail an electronic ***ID CHECK*** if the device has been secured.

In general, the devices in ceramic packages are usually Ultra Violet light erasable and devices in plastic packages are "once only programmable" (OTP).

INTEL 879X AND 87196XX FAMILIES.

There are three important features to note when dealing with these device families as an example:

1. The EPROM part of these devices is mapped internally from 2000h upwards. The microcontroller only deals with the EPROM part, so the EPROM is treated as though it starts from zero. The 8k offset is compensated for.

2. There is a Chip Configuration Byte at address 18h (actual device 2018h). The two most significant bits of this, correspond to the security bits of the device. The firmware issue U1.05 will prevent these from being programmed. If these were to be programmed, then it would not be possible to VERIFY the programmed data.

3. Within the memory part of the device, there are certain locations reserved by INTEL and no attempt should be made to program these. The programmer firmware will allow these locations to be read, but will not allow them to be programmed.

The EPROM memory map for the INTEL 8795BH is shown in below as an example. Please consult the device manufacturer's datasheet for exact details of other devices.

INTEL 8795BH.

Address	Description	Device Address
0 - 11H	Interrupt Vectors	2000 - 2011H
12 - 17H	Reserved	2012 - 2017H
18H	Chip Configuration Byte	2018H
19H	Program to 20H	2019H
1A - 1BH	Self Jump Opcode 27H,FEH	201A - 201BH
1C - 1FH	Reserved	201C - 201FH
20 - 2FH	Security Key	2020 - 202FH
30 - 6FH	Reserved	2030 - 206FH
70 - 71H	Signature Word	2070 - 2071H
72 - 7FH	Reserved	2072 - 207FH
80-1FFFH	EPROM	2080 - 3FFFH

PROGRAMMING GUIDE WITH ELAN REMOTE DRIVER.

Switch on programmer and select on PC, 3000 and ENTER.

The system communicates to the programmer and the installed ZIFPAC is displayed on screen of host PC.

If settings are not compatible, then use **<CTRL and BRK>** to allow the driver software to automatically set in COMMS window Port, Baudrate, Parity and select ***RESTART.***

In select window, select the wanted device for programming etc. Secure selection with the ENTER key on PC.

In file window convert named Object File, in known format to named nnn.EBF format file. This is the Programming File.

When converting an Object File to nnn.EBF , the Object File maybe from address 2000 of the device memory map, then use base offset of 2000. This is to ensure that the PROM data is placed at thecorrect place in the memory map of the device.

If the Object File starts at 0 and this is the start of the PROM part of the device then use base offset at 0.

In ***PROGRAM*** window, select the required nnn.EBF file for programming the selected target device and insert device when prompted.

VERIFY and BITTEST of all locations of the device with the program can also be done in this window.

The device content may be secured using the ***SECURE OPTION*** in the ***PROGRAMMING*** window.

In the ***FILE*** window, the current nnn.EBF file maybe edited and stored under a new name.

Data in unsecured devices or master devices, can also be read in and stored.

Chapter 5

LOGIC ZIFPAC

MODEL -145 OVERVIEW

This extension of the PC controlled ELAN REMOTE DRIVERsoftware with the ELAN Model 3000 Programmer equipped with PLD RAM enhancement and the -145 ZIFPAC, will support the programming of logic devices.

In general, the controls of this issue of software are the same as for programming PROM and MICRO devices, as it shares the same pull down menu format.

Some options are not valid for this application of programming logic devices and are outlined below.

Press Function Key F1 on the PC for ***HELP*** at any time.

THE EXFILE PULL-DOWN MENUS FOR -145 ZIFPAC.

With the system powered up, it will configure for the -145 ZIFPAC, as displayed on the screen of the controlling PC .

GENERAL DESCRIPTION

On power up the ***PROGRAM*** section of the menu will be shown.

Moving the cursor key to the right, the windows read as follows:

BATCH: This option is not available with this ZIFPAC.

SELECTION FROM THE COMMS MENU:

This window permits the selection of:

SELECT PORT: Displays the current selected port detected by the software.

BAUD RATE: Displays the selected baud rate used by the system. Manual rate can be selected between 38400 -150.

PARITY: Displays the selected parity setting.

STOP BITS: Displays the selected choice of stop bits.

RESTART: A forced Restart may be selected to re-start communication if required.

FILE I/O FORMAT: This menu permits the selection of JEDEC, P.O.F, INTEL HEX and MOTOROLA S RECORD formats. The default selection is JEDEC format.

Note: Most PLD's only use JEDEC format whereas ALTERA MAX devices require P.O.F format. When the XILINX devices are selected extra communication formats for programming I.e. MOTOROLA-S and INTEL HEX can also be selected. For these XILINX devices the A116 adapter is required for programming.

Note: *If an invalid file format is selected, a valid format must be RE-SELECTED before continue operations.*

SELECTION FROM THE SET-UP MENU.

SELECT DEVICE: This option displays all the devices supported by the programmer's Firmware and the target device must be selected for programming at this stage.

If the programmer was not connected correctly to the PC at the time of power-up and booted-up, then it may be necessary for **RESTART** to be selected when no devices can be found. (**RESTART** command is in the *COMMS* window.)

WORD SIZE.} Not valid with -145

BYTE ORDER.} Not valid with -145

NUMBER OF BLOCKS.} Not valid with -145

ALTERNATIVE SKT.} Not valid with -145

ELECTRICAL ID.} Not valid with -145

AUTO TEST VECTORS: Select *ENABLE* or *DISABLE.*

If *ENABLE Is* selected and the source JEDEC file includes test vectors to value the programmed device functionality, then the device is tested as part of the programming cycle.

If *DISABLE* is selected the device is not tested with the test vectors in JEDEC file, whether these are included or not.

SELECTION FROM THE FILES MENU.

DIRECTORY: When selected, a prompt for File Spec. allows file names or file name extensions to be entered and listed on the screen of the controlling PC. Wild cards, i.e. *.*, may be used to givea full listing in this directory.

To edit the file *NAME,* some keys are available as follows:

Cursor key - to move around input text.

Backspace - to erase text backwards.

Delete - to erase text forwards.

<Ctrl> k - to erase the entire input line.

Insert - to toggle between insert and overstrike. To abandon the input, simply press ENTER on an empty input line in the window.

DOS SHELL: This command allows entry in to the DOS environment and allows re-entry into 3000 REMOTE DRIVER by typing *EXIT,* without losing the previous screen.

CONVERT TO EBF

CONVERT FROM EBF

EDIT EBF FILE: The system does not permit the editing of JEDEC data files, but for the XILINX devices EBF data files can be edited, converted and created.

SAVE CONFIG

LOAD CONFIG: These commands allow saving and re-loading of current screen parameters.

QUIT: This command exits from 3000 REMOTE DRIVER.

SELECTIONS FROM THE PROGRAM MENU:

PROGRAM: Upon selection of the PROGRAM menu the next screen requests to make the selection between:

- CURRENT I/O FILE for programming.
- NEW I/O FILE for programming.
- EFPER RAM.

The ***CURRENT I/O FILE*** is the current selected JEDEC file in directory as shown on the screen. When selected the system downloads this file to RAM for programming.

The selection ***NEW I/O FILE*** requests the name of the new JEDEC file to be downloaded to RAM for programming.

The selection ***EFPER RAM*** means, that if data has been read in from a master device into RAM, identical devices can be programmed from that data currently in held in RAM.

If data is residing in RAM and a programming JEDEC file is downloaded to RAM, this then overwrites current data.

The programming cycle consist of the following stages :

- Down loading JEDEC data file.
- Insertion and pin connection test.
- Blank check of the device.

*Note: If on **BLANK CHECK** the device fails, a choice to continue with **BITTEST** is given, to determine if the device can still be correctly programmed.*

- Bit-test
- Programming
- Verify programmed data with data file in RAM.
- Test the device for functionality with test vectors.
- Generate fuse map check sum, to give an indication of the program validity of the content on the device.

VERIFY: The menu requests the JEDEC file or RAM to verify against programmed dat. This selection enables separate verification of programmed data in the device.

BLANK: This selection enables the device to be blank checked.

BIT-TEST: This selection enables for separate BIT-TEST.

READ: This selection requests the name of the JEDEC data file, for reading in the data from the master device in the ZIF socket. JEDEC data can be read into the current selected file, or other named JEDEC data file. If the data is read into RAM, then this may be up-loaded by a separate path in the SPECIAL section of this menu.

Other sections of the menu can be viewed by moving the PC cursor key to the right.

INSERTION CHECK: This selection tests separately all the connecting pins, for connection quality, and reports which pin is suspect.

*Note: Specific devices require particular adapters and the system detects the correct insertion of a specific adapter for programming. If another specified adapter is inserted, and not correctly placed, the message **ADAPTER ERROR** is given.*

*Note: If the correct adapter is installed but the target device is wrongly inserted or wrong device is placed, then the message **DEVICE INSERTION ERROR** is given.*

SELECTION FROM THE SPECIAL MENU:

This selection generates a special additional menu, as follows:

EXIT SPECIAL FUNCTION MENU: This selection enables the return to the main menu.

ERASE: This selection is not valid with the -145 ZIFPAC.

SECURE: This selection, after a further ENTER request will secure devices, so that the program content cannot be read back. Secured devices cannot be verified with the original JEDEC fuse map. (Functionally the devices can be tested with the VECTOR TEST option in the PROGRAM menu.)

DISABLE PROTECTION: Not valid with the -145 ZIFPAC.

DOWNLOAD FILE TO 145 RAM: This selection enables a named JEDEC programming data file to be downloaded to the RAM of the programmer. The named file, current, new or other JEDEC Data file in the directory, when downloaded, overwrites the data currently held in EFPER RAM of the programmer.

UPLOAD FILE FROM -145 RAM: JEDEC programming data, read from a master device and held in RAM of the programmer, can be uploaded into a named, current or other data file in the directory of the controlling PC.

TEST VECTORS; The current selected device, can be separately functionally tested with the test vectors, held in the current JEDEC data file for this device.

RAM FUSE CHECK SUM: The checksum of fuse map is read from the data held in EFPER RAM and is displayed.

PROGRAM RESET POLARITY FUSE: This option PROGRAMs the polarity fuse of a particular device, currently only XILINX XC1765.

READ RESET POLARITY FUSE: This option reads the state of polarity fuse of a particular device, currently only XILINX XC1765.

SELECT FOR THE PROGRAMMING OF A MASTER DEVICE : When ***ENABLE*** is selected and this option is available, with the selected device, then the device will be programmed as a master.

If the device is then reprogrammed a WARNING message is given that this device is a ***MASTER DEVICE.***

This facility only applies to LATTICE GAL devices or equivalent.

MORE FEATURES WITH EXFILE/REMOTE DRIVER ISSUE X5.00.

With this issue new improvements and additional features are included for the programming of EPROM, EPROM, FLASH MEMORY and MICRO devices. It is recommended to understand the previous features and facilities as per standard manual first before using these new options.

INSTALLATION

This issue of software is to be installed in a new way by removing the current 3000 directory (previous software) and then from the provided software disk install the software with the INSTALL command. This will create a new 3000 Directory for use.

Note: It is recommended to save into a safe place all current data files that then can be re-imported into the new 3000 directory.

The main new feature with this issue is the new method of merging up to eight data files in-to one .EBF programming file with the full control of data file and target file start addresses.

It is recommended that if this operation is required to tabulate the various data for assembly first, detailing the file name, file start addresses, target start addresses and last address used. This will prevent the loss of data for the intended target device.

The example given shows the assembly of data to a 2 MEG device.

FILE	DATA	DATA USED	TARGET ADDRESS	LAST ADDRESS USED
TESTMS1.MS	0-1FFF	0-1FFF	0	1FFF
TESTMS2.MS	0-1FFF	0-500	2000	2500
TESTMS3.MS	0-1FFF	0-1FFF	2501	4500
TESTMS4.MS	0-1FFF	0-FFF	4501	5500
TESTMS5.MS	0-FFFF	0-FFFF	5501	15500
TESTMS6.MS	0-FFFF	0-FFFF	15501	25500
TESTMS7.MS	0-FFFF	0-AAFE	25501	2FFFF
TESTMS8.MS	0-FFFF	0-FFFF	30000	3FFFF

When this **CONVERSION TO_EBF** function is started for a previously selected device, the request is made for each identical format file name to be declared with its starting address to read from, followed by the question of MORE FILE Y/N, until up-to 8 files are declared. For less the 8 files More files, N is to be used to terminate this part.

The next stage in the procedure is the request to enter the target deposit starting address for each data block under the term **OFFSET ADDRESS**, and the target file name.

If the data length is to be reduced, this can be carried out by editing later in the EBF EDIT mode or by over-writing with other data. This could be a file with an all FF or other file. It is also possible to place the next data pack later on the memory map thus leaving space vacant in between.

With sufficient care taken in terms of files, starting, end and target addresses this function is a powerful facility, with the indicated listing method used prior to merging data files.

When the last data file is in place then the target programming file is to be declared, xxxxxxxx.EBF and nversion will start and % progress is shown.
LE<INPUT><BASE ADDRESS><OFFSET ADDRESS>
Note: When BINARY data files are used it is important to include the /b after the file name, ie: datafile.bin/b.

With this issue ASCII HEX SPACE format data files, xxxxxxxx.ahs, and INTEL hex 02 and 04 formats can be used.
Note: For 02 min xxxx start address and for 04 min xxxxxxxx start address is required.

TO EDIT DECLARED .EBF FILE AND SEARCH FOR CODE IN .EBF WINDOW

When the EDIT EBF file window associated with the target device is opened, with the F5 Function Key on the PC, data can be located in the map from the last point where the cursor is. If the cursor is moved with the TAB key to the ASCII window, then up to four CHAR must be declared. If this is not known then the wild char can be used. IE *z** ABI* ***E abcd. SHIFT AND F5 will start the arch for the next CHAR from that point.

When the EBF file has been selected for EDIT, the following facilities are available to edit the data on the map.

- ALT and A with the address number will move the cursor to the required address in the field.
- ALT and S will mark selected address as an start address for marking and to fill with data from or to copy from.
- ALT and E will mark selected end address for filling to with data or to end copying from.
- ALT and F will fill the marked area with the to be declared data, xx for 8 bit, xxxx for 16 bit and xxxxxxxx for 32 bit wordsize.
- ALT and U will UNMARK the marked area.
- ALT and C will copy the start to end marked area, to the new selected point in the memory map.

It is further possible to create and convert data files that have large unused space between the data blocks used , and is active in block or in set mode.

DIRECTORY/CHANGE DIRECTORY.
This is an expansion in the file search facility in the current directory using *.*, this will show all the files in that directory. If in CHANGE DIRECTORY, a path is added, then all the files in that directory will be shown.

SET PROGRAMMING

With this issue 16 bit devices can be programmed in set mode including batch mode.

ALT-SKT

The ALTERNATE SOCKET mode is now operational thus permitting the programming of memory devices that require wider device adapters, i.e. PLCC packaged devices. Up to four positions are available.

AUTO START-UP

When either the or 3000 DIR has been selected the pre-programmed BATCH file can be found and include in the startup routine i.e. \3000 /F MYFILE.EBT

BATCH FILE ROUTINE

The BATCH FILE ROUTINE has been expanded and improved as can be seen in the attached examples. Note: The devices used do not indicate any bias towards any semi house.

```
TEST1.EBT
DEVICE NEC UPD27C1001A
WORDSIZE 16
NUMBLOCKS 1
READ TEST1.EBF
```

This is a standard batch file entry for the reading in of two 8 bit devices in 16 bit wordsize.

If the devices happen to be in two in block then WORDSIZE = 8 and NUMBLOCKS = 2 is used.

To convert this .EBF file to Motorola S format, thus the data file is not device dependent, then add the following line:

```
FROM_EBF TEST.EBF TEST1.MOT MS
```

If two devices are thus to be programmed from a MOTOROLA S record as per this example the BATCH file will look like the following:

```
DEVICE NEC UPD27C1001A
WORDSIZE 16
NUMBLOCKS 1
FILE1 C:\EXFILE\TEST1.MOT 0 0
TO_EBF 1 TEST1A.EBF HEX YES FF
:RUNA
PROGRAM TEST1A.EBF
LOOP RUNA 5
```

This simple batch file shows the declared device, declared wordsize required, set mode in this case, one number of blocks used. This followed by declared first file test1.mot, format auto detected, and read from 0 starting address in that file and is to deposit data at 0 start address in the memory map.
Note: Up to 8 files (--- file8) can be used.

The next line is the conversion command TO_EBF and it expects in this case only 1 file for conversion and stating that the data files are hex coding, of the same type only, and that unused space on the memory map is to be filled with ff, in this case.

This stage also shows the notation of a programming repeat, followed by the programming .EBF file to be used and this concluded by the amount of programming cycles that have to be repeated.

The next example is the conversion and programming with a INTEL HEX 04 format in batch mode.

```
TEST6.EBT
DEVICE SGS-THOMSON N27C4001
WORDSIZE 32
NUMBLOCKS 1
FILE1 B:\BIGFILE.HEX FFE0000 0
TO_EBF 1 TEST6.EBF HEX YES FF
PROGRAM TEST6.EBF
```

If thus the data file format is INTEL HEX 02 the from xxxx start address is required, or for INTEL HEX 04 the start address should be xxxxxxxx as shown.

A further example shows the loading of 8 data files on a small device where in each case only two or three lines are used.

```
TEST7.EBT
DEVICE TEXAS TI2732A
WORDSIZE 8
NUMBLOCKS 1
FILE1 C:\3000\TESTAA.MS 20 0
FILE2 C:\3000\TESTBB.MS 20 20
FILE3 C:\3000\TESTCC.MS 20 40
FILE4 C:\3000\TESTDD.MS 20 70
FILE5 C:\3000\TESTEE.MS 20 90
FILE6 C:\3000\TEST55.MS 20 E0
FILE7 C:\3000\TEST44.MS 20 100
FILE8 C:\3000\TEST33.MS 20 120
TO_EBF 8 TEST7.EBF HEX YES FF
PROGRAM TEST7.EBF
```

A further useful screen notation is that at the top righthand of the screen the last checksum of the last cycle is detailed for each programming socket, and above this completed programming repeated cycles are shown.

The standard manual shows an example for the programming of MICRO devices.

The 5-J08 batch program will look like the following:
xxxxxxxx.ebt

```
DEVICE INTEL_FLASH IMC00FLKA
NUMBLOCKS 1
WORDSIZE 8
READ CARDT17.EBF
.RUNB
PROGRAM CARDT17.EBF
LOOP RUNB 99
```

There is no change in programming logic devices with 3-145.

PROGRAMMING MICROCHIP PIC DEVICES

Micro PIC devices can be programmed with the -940 ZIFPAC and active adapters A127-A130 for PIC 16c54-58 in DIL and surface mount packages.

With the correct device and adapter selected and installed, the EPROM can be programmed, but the Config fuse etc. can be programmed in the SPECIAL section of the PROGRAM menu.

The selections can be made as per table and note the Config fuse can also be read and programmed in this menu.

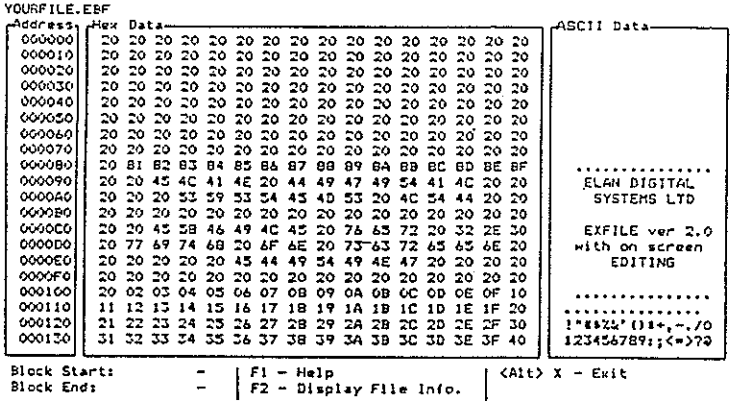
OSCILLATOR	FUSE VALUE																
Selector LF	0	X				X				X				X			
XT	1		X				X				X				X		
HS	2			X				X				X				X	
RC	3				X				X				X				X
SECURE (TAB) ON	0	X	X	X	X	X	X	X	X								
OFF	8									X	X	X	X	X	X	X	X
WATCH DOG ENABLE	0	X	X	X	X					X	X	X	X				
DISABLE	8					X	X	X	X					X	X	X	X
FUSE VALUE TO SET	FXX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Thus the fuse value in the first case will be FFO and in the last case the value will be FFF as shown in the table.

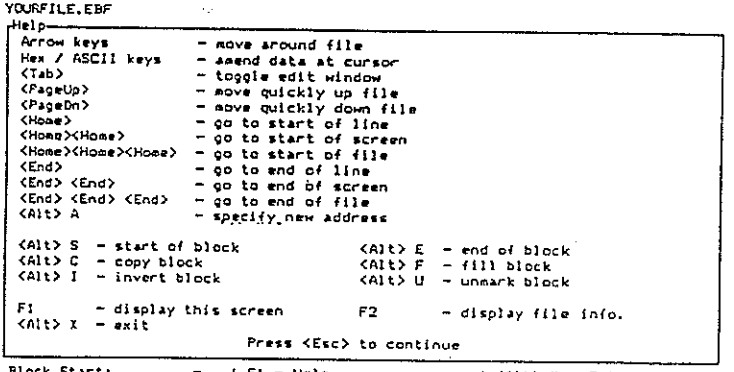
CHAPTER 6

3000 REMOTE DRIVER EDITING

Example of the Edit Screen



Description of the Edit Commands



APPENDIX A,

SERIAL COMMUNICATION FORMATS

The Model 3000 Programmer communicates using a proprietary protocol.

INTEL LOADER FORMAT

(Standard and Extended 16 bit Address records)

Characters	Description
1.....	**Record mark:** A colon is used to mark the beginning of a record.
2-3.....	**Record length:** A two-character hex representation of the number of bytes of data in the record. Character 2 is high order record length of characters.
4-7.....	**Load address:** A four-character hex address at which the first data bytes are loaded into successive (higher) memory locations. Character 4 is the high order address digit. In an end-of-file record, the load address is taken as the starting address.
8-9.....	**Record type:** A two-character hex code specifying the record type. All data records are type 00. The most significant digit is character 8. Extended address records are type 02, and end of file record is 01.
10-N.....	**Data:** Each byte of data is represented by a two-character hex number. The high order character precedes the low order.
N+1-N+2	**Checksum:** A two-character hex checksum, which is the negative sum of all bytes in the record except the colon and check-sum, evaluated modulo 256. The sum of all bytes in the record plus the checksum must be zero.
N+3.....	Carriage return
N+4.....	Line feed

MOTOROLA "S" RECORD

The programmer will action S0, S1, S2, S3, S7, S8 and S9 records.

where:-

- S0 is the header record.
- S1 is data from 16 bit address record.
- S2 is data from 24 bit address record.
- S3 is data from 32 bit address record.
- S7 is file terminator.
- S8 is file terminator.
- S9 is file terminator.

TEKTRONIX HEXADECIMAL FORMAT

Characters	Description
1.	**Record mark:** A slash is to mark the beginning of a record.
2 - 5	**Load address:** A four-character hex address at which the first data bytes are loaded into successive (higher) memory locations. Character 2 is the high order address digit. In an end-of-file record, the load address is taken as the starting address.
6 - 7	**Record length:** A two-character hex representation of the number of bytes of data in the record. A record length of zero (00) indicates an end-of-file. Character 6 is the high order record length of character.
8 - 9.....	**Header checksum:** A two-character number representing the eight-bit sum, modulo 256 of the hex values of the six characters 2 - 7.
10 - N	**Data:** Each bytes of data is represented by a two-character hex number. The high order character precedes the low order.
N + 1 - N + 2..	**Data checksum:** A two-character number representing the eight-bit sum, modulo 256, of the hex values of the digits that make up the N data bytes.
N + 3 - N + 4..	Carriage return
	Line feed

EXTENDED TEKTRONIK HEXADECIMAL FORMAT

HEADER FIELDS

1 ..... **Record Mark:** A percent (%) specifies that the block is in extended TEKHEX format.

2 - 3..... **Block Count:** The number of ASCII characters in the block.
This count does not include the leading % or the end of line.

4 ..... **Block Type:**
6 = Data Block
8 = Termination Block
3 = Symbol Block (not supported thus ignored)

5 - 6 ..... **Checksum:** A two digit HEX no. representing the sum, MOD 256, of all the characters in the block, excluding the leading %, the checksum digits and the end of line.

EXTENDED TEKHEX DATA BLOCK FORMAT

1 - 6..... **Header:** As defined above with a block type of 6.

7 - N ..... **Load Address:** This is a variable length number, range is from 2 to 17 ASCII characters.
The first ASCII character indicates the length of the rest of the field.
The value of 0 (zero) indicates a length of 16 ASCII characters.
N.B **Only the least 8 significant characters of the address are actioned.**

N + 1..... **Data**

Carriage Return

Line Feed

EXTENDED TEKHEX TERMINATION BLOCK FORMAT

1 - 6 ..... **Header:** As defined above with a block type of 8.

7 - N ..... **Transfer Address:** The address where a program execution is to begin, a variable length number.
N.B **On input transfer address is ignored**
On input transfer address is 0 (zero).

APPENDIX B,

3000 Specifications

SYSTEM REQUIREMENTS (Min.):

IBM PC and PS/2 or compatible: (DOS 2.1 or later) plus one Floppy disk drive. A minimum of 256K RAM is required. Greater system efficiency will be achieved with 512K RAM or more. 3000 REMOTE DRIVER PC software auto-configures to any PC and display, (color or mono).

FUNCTIONAL SPECIFICATIONS:

Display selections of major manufacturers include:- AMD, Atmel, Fujitsu, GI, Hitachi, Hyundai, Intel, Matsushita, Mitsubishi, Motorola, National Semiconductor, NEC, Richco/Panatech, Rockwell, Seeq, SGS-Thomson, Signetics, Texas Instruments, Toshiba, Vitelec, VLSI, WSI, Xicor etc.

Devices:

Generic device type range accessable under manufacturer selection:

NMOS & CMOS 5V EPROMS: 2716B, 2532A, 27(C)32A, 2732B, 27(C)64(A), 87C64, 27(C)128(A)(B), 27(C)256, 87C256, 27(C)512, 27(C)513, 27(C)011, 27(C)010, 27(C)210.

EEPROMS: 2864, 28256, 48128

FLASH EPROMS: Intel 27F64, 27F256, 28F256

MICROs and PLD Devices

*****Ask for latest full device listing*****

Data Ram: The host PC DISK and RAM is made available as required.

Formats: All the standard formats available including: Intel, Motorola, Tek Hex, Ext. Tek Hex etc.

Interface: Serial I/O RS232 Auto configuring, remote operation, baud rates up to 38,400. This system is designed to accommodate the control of two concurrent bi-directional RS232 Ports if required via an optional split cable.

OPTIONS:

A range of Socket Converters is available for PLCC and SO packages.

ELECTRICAL SPECIFICATIONS:

Operating Voltages:

A: 220/240 V ac +/- 10%
50-60 HZ 45W max.

B: 110/120 V ac +/- 10%
50-60 HZ 45W max.

Switch-mode programmable power supply incorporates over-voltage and over- current protection.

Interference Suppression:

The 3000 is designed to meet all European and USA interference standards.

PHYSICAL SPECIFICATIONS:

Cast aluminum base, plastic injection moulded cover with a fully enclosed detachable front ZIFPAC.

Dimensions:
L.303mm x D.255mm x H.95mm.
(L. 12in. x D. 10in x H. 3.75in)

Weight:
5Kg (11 lbs)

Operating Temp:
+12deg.C to +35deg.C.
(54deg.F to 95deg.F)

Relative Humidity:
0-90% non-condensing.

All Programmers come with one year warranty on hardware and software.

Elan Digital Systems reserve the right to alter specifications without notice.

APPENDIX C,

3000 EXAMPLE

INSTALLING

Insert 3000 REMOTE DRIVER disk in computer drive A (or copy files COPY*.* to an ELAN3000 directory) and type *3000* and press ENTER. ELAN3000 will display a title screen for a few seconds, before starting an auto set-up routine to configure the Programmer/Computer serial link. (Should this fail, check the programmer is switched on and check serial cable connections. See INSTALLATION Chapter 1).

On successful completion, ELAN3000 will up-load device data from the programmer and present the "Set-up" Menu for device selection.

TO COPY A MASTER DEVICE

SELECT

Indentify your chosen master device and use ***SELECT DEVICE*** from the Set-up menu. (Unless already selected, see Device description on lower portion of screen). Use the cursor control keys to select the device then press ENTER.

READ

Go to "Read" in the Program Menu and press "Enter". Your choices will be to Read to the "Current EBF file" (Elan Binary Format file) or "Other File".

If you have not previously read your master device data to a file, move the cursor on the ***OTHER FILE*** position and press ENTER.

You may now type in up to 8 characters to name your file, (you may add a "." and file extension of to 3 characters and additional characters to define directories and paths if required). This file will now become the "Current EBF File". Press ENTER and follow the on-screen instructions to insert the master device and complete the Read function. The checksum will be displayed on-screen.

PROGRAM

Go to ***PROGRAM*** in the Program Menu and press ENTER. Press ENTER to program from the "Current EBF File" and follow the on-screen prompts to complete the programming function.

Any number from 2 to 8 master sets of devices may be Read, then copied concurrently using the 3-232 or 3-832 programmers (or the -232 or -832 ZIFPACs fitted to any Model 3000 Programmer). See SETS page 3-3.

APPENDIX D,

ERROR MESSAGES

EXFILE ERROR MESSAGE:	NOTES:
Cannot communicate at old baud rate;.....	Communications failure; turn off programmer & re-start.
Cannot communicate at new baud rate ;..... Cannot access selected port; Cannot access port COM1; Comms. setup aborted - User Break; No response from programmer ;.....	Reverts to previous choice. Check that the programmer is switched on & connected to P.C.
Unable to save configuration;.....	DOS error condition eg. "disk full".
Unable to load configuration;.....	Specified file not found.
Not enough memory for this operation;	
Full device details not known;.....	Check current device selections.
No current Binary Files selected;	
Cannot communicate using current setup; Device download aborted - User Break; Faulty Device(s) - (Vcc Error); Faulty Device(s) - (Vpp Error); Non-Blank Device(s); Illegible Bit Error(s); Device(s) failed to program; Verify Error(s); Read Error(s); Device Insertion Error(s); Premature "end-of-file" in binary file; Invalidchecksum in binary file;	Communications failure; Turn off programmer & re-start.
Selected device has incorrect size;.....	Currently selected device is not as size specification in binary file.

EXFILE ERROR MESSAGES:	NOTES:
Operation aborted - User Break;	
Programmer will not Blank Check devices;	Communications failure; turn off programmer & restart.
Cannot open binary file;.....	File not present.
Programmer will not Bit Test Device(s);	Communications failure; turn off programmer & re-start.
Programmer will not program Device(s);....	Communicationsfailure; turnoff programmer & re-start.
Programmer will not verify Device(s);.....	Communications failure; turn off programmer & re-start.
Programmer will not read Device(s);.....	Communications failure; turn off programmer & re-start.
Programmer will not accept selection;.....	Communicationsfailure; turnoff programmer & re-start.
Can't obtain device size from programmer;	Communications failure; turn off programmer & re-start.
Cannot obtain current device pinout ;.....	Communications failure; turn off programmer & re-start.
Comms. setup aborted - User Break;	
No response from programmer;	
Not enough memory for this operation;	
Programmer not Elan 3000 Series;	
Word Size must be 16 for this device;.....	40 pin 16 bit devices.
Word Size must be 8 for this device;.....	3-132, 3-142 programmers only.
Invalid number of blocks;	
Invalid Word Size / Num. Blocks setup;	