

Allpro88

Users Guide

This manual describes the operation of Logical Devices' ALLPRO software driven device programmer.

The information contained in this manual has been reviewed for accuracy, clarity, and completeness.

Please report, in writing, any errors to:

**LOGICAL DEVICES, INC
P.O. BOX 8400
DENVER, CO 80201**

Logical Devices, Inc. reserves the right to use and distribute any information supplied, without obligation.

Limitations of Liability

- . This manual is subject to change without notice
- . Logical Devices does not assume responsibility for any errors which may appear in this document nor does it make a commitment to update the information herein.
- . Logical does not assume any liability arising out of the application or use of its product described herein.
- . Logical Devices will not be responsible for any damage to either the user, any hardware, or any devices that may result from accidents during operation of this unit according to this manual or associated material.
- . Logical Devices reserves the right to make changes without further notice to any products in order to improve reliability, function, or design.
- . Logical Devices products are not authorized for use with components intended for life support applications.
- . Logical does not convey any license under its patent rights nor the rights of others.

Logical Devices, LDI, STAMP-IT, SERIALPRO, BATCHPRO, and ALLPRO are trademarks of Logical Devices, Inc. UNIX is a trademark of AT&T Bell Laboratories. PC-DOS and IBM are trademarks of International Business Machines Corp. MS-DOS is a trademark of Microsoft Corp. Tekhex is a trademark of Tektronix, Inc.

ALLPRO 88 Development Group:

Project Engineer: Steve Harr
Project Leader: William Snow
Systems Engineer: Michael Appleton

Software Engineer: Jun Flores
Device Support: Tony Tran
Software and Support Coordinator: Joleen Rasmussen
Software Testing: Vincent Figundio

Drafting: Adrienne Licudine

This manual was produced on the Apple Macintosh 
Text was created in Microsoft Word™

Graphics were produced using MacDraw™ and Canvas™
Paste-Up and Final Assembly was accomplished using various versions of Pagemaker™

ALLPRO 88 Manual Group:

Concept: William Snow
Technical Writers: Jack Mack
Jim Barnes

Copyright © 1990 by Logical Devices, Inc
All rights reserved. Printed in the United States of America

No part of this book may be reproduced or copied in any form or by any means without prior written permission of Logical Devices, Inc.

Table of Contents

INTRODUCTION

About this manual	1
Safety Note Conventions	1
Typeface Conventions	2
The Allpro System	3
Allpro C	3
Allpro 88	3
Allpro Interfaces	4
Modules	4

GETTING STARTED/INSTALLATION

ALLPRO Systems	1
Comprix Total Support Programming System	1
Memory Programming System	1
Logic Programming System	2
Installation Requirements	3
Memory Map Conflicts	4
Hardware Installation	5
Precautions for User and Machine	6
Eraseable Devices and UV	7
Interface Card.....	8
Standard Interface Card	8
Turbo Interface Card.....	8
Module Installation	10
LED Indicators	11
LED Indication Table	11
DIP Socket	12
Device Insertion/Removal	12
PLCC Socket	13
PLCC Device Insertion/Removal	13
PLCC Module.....	14
Software Installation	15
Software Description	15
Program Organization.....	16
Hard Disk Installation	16
Installation Environment	17
Selective Installation of Libraries	17

Expanded Memory System	18
Introduction	18
Requirements	18
RAM Disk and LIM EMS	18
Extended Memory and LIM EMS	19
Installation of EMS	19
RAM Disk Installation	20
Opening Display Screen	20

QUICKSTART Programming Examples

Programming Example 1	1
Programming Example 2	4

OPERATIONS

User's Interface	1
Main Display	1
Message Center	1
Operator Input	1
Quick Select of Items	2
Performance Monitor	3
Max Retries Scale	3
ALLPRO Main Menu and Sub-Menus	4
How Descriptions are Arranged	4
Device Select	6
Device Libraries	6
LDI's Unique Codes	7
LDI Unique Five-Digit Device Code	7
LDI Unique Six-Character Library Filename Code	7
Choose Library and Device	8
Select Device	8
Package Select	9
Reserved	9
Return to Main Menu	9
Blank Check Device	10
Load Device	11
Program Device	12
Verify Device	13
Secure Device	14
Edit Data Map	15
Device Arrays	15
Editing a Data Map	16
Byte Swap Function	18
Edit PLD Data Map	19

Read Formatted File	20
Onscreen File Select	20
Intel Hex Format (Read)	21
ASCII Hex Format (Read)	22
Binary Format (Read)	22
Straight Hex Format (Read)	22
Motorola Hex Format (Read)	23
JEDEC Format (Read)	23
Tektronix Hex Format (Read)	24
Hex Download Parameters	25
Number of Devices in Set	25
Current Device in Set	25
Start File Address	25
End File Address	25
Start Device Address	26
Nibble to Download	26
Start Set Number	26
Write Formatted File	27
Intel Hex Format (Write)	27
ASCII Hex Format (Write)	27
Binary Format (Write)	28
Straight Hex Format (Write)	28
Motorola Hex Format (Write)	28
JEDEC Format (Write)	29
Tektronix Hex Format (Write)	30
Utilities	
View/Change Working Directory	31
Specify Library Path	32
Data Initialization Toggle	32
Edit File	33
Program in Sets	33
Auto Secure Toggle	34
Automatic Test Vector Toggle	34
Range Settings	35
Auto-Serialization (Stamp-It™)	35
Checksum	35
Handler Operations	36
Functional Test Device	37
Quit	39

ERROR CODES

Error Message Listing	1
------------------------------------	---

TROUBLESHOOTING

Problem Analysis	1
DIAG4: Universal Programmer Diagnostic	3
Main Menu	3
System Status	3
Statistics	4
Cycle Command	4
Pindriver Selection	4

TECHNICAL REFERENCE

System Overview	1
“Pindriver” defined	2
System Block Diagram	3
Hybrid Pin Driver	4
Pindriver Boards	6
Modular Construction	6
Intelligent Power Control	6
Main Power Switch	6
Power-On Reset	6
System ID	7
Module ID	7
Pindriver Physical/Logical Translation Table	8
ALLPRO Memory Mapping	9
VGA Card Memory Conflicts	9
ALLPRO Specifications	10
PLCC Module	11

APPENDICES

A: Configuration File	
B: Set Programming	
C: Data File Formats	
D: BATCHPRO: Production Programming Utility	
E: STAMP-IT: Auto Serialization Facility	
F: IC Handler	

TECHNICAL SUPPORT

Procedure	1
Rapid Response Fax Service Sheet	2
Bulletin Board Service	3
Returning the Allpro for repair	6

WELCOME !

Thank you for choosing Logical Devices, Inc. for your programming needs. By making the investment in an ALLPRO universal programmer, you have chosen a system to meet your current needs as well as future device programming challenges. Since its inception in 1976, Logical Devices, Inc. has been committed to providing a superior product at reasonable prices. As part of our customer base, you have access to the largest device support base in existence. Only a system that offers a DAC-per-pin can provide the wide-ranging support we are able to deliver.

Our excellent support staff is ready to assist you if you have any questions or encounter any problems. We promise to be there when you need us.

Anyone can provide hardware and empty promises. But the more you use your ALLPRO, the more you understand what Logical Devices, Inc. is all about ... PEOPLE!

David Mot
President
Logical Devices, Inc.

ABOUT THIS MANUAL

Most of the information in this manual is applicable to the ALLPRO 88 and the ALLPRO C models. Where there are differences between the ALLPRO 88 and the standard ALLPRO, either with software or hardware, these differences are explicitly noted in the manual. If there are any questions on the capabilities of a particular ALLPRO programming system, please consult the factory.

This manual is designed to serve both as a reference manual and as a learning aid. Make certain that all safety notices and warnings listed throughout the manual are strictly adhered to. If there are addendums provided, be certain to read them also.

The Getting Started / Installation section is a guide that will greatly expedite the installing of all hardware and software necessary to get the ALLPRO running. A Quickstart example section is a helpful guide through the basic operating procedures of the ALLPRO.

Safety Note Conventions

Throughout this manual, three types of user alert notations are used. They are **NOTE**, **CAUTION** and **WARNING**. They are intended to be interpreted as follows:

NOTE assists the user in accomplishing a task. It makes the job more easily understood.

CAUTION alerts the user that, if an instruction is not followed, unexpected results or damage to the device may occur.

WARNING alerts that harm to the user or severe damage to a device, drive, directory or computer peripheral will occur if the instructions are misinterpreted or are incorrectly followed.

Typeface Conventions

This manual gives step-by-step procedures and examples. To make the manual easier to read, the following conventions are used. In general, the screen input and output, diagrams, notes, cautions, warnings and special items appear in the right columns on the manual pages, while text appears in the left columns.

User Input

Input that is entered by the user is shown in bold type. An example of this typeface is: Type in the name **ALLPRO** and then press **F10**.

>ALLPRO

Screen Output

Output that appears on the computer screen is printed in the manual in a type style similar to the style on the screen. For instance:

Press any key to continue.

Variable names that appear on the screen are indicated by angle brackets, such as:

<filename>.SCH.

Filenames

Within text, program names and filenames are shown in bold type the first time they appear in a particular section. The first time **ALLPRO.CFG** appears, it is bold type but, if **ALLPRO.CFG** appears again, it is not bold. This avoids too much bold type in the text.

File names are bold the first time they appear in text.

ALLPRO.CFG
ALLPRO.CFG
ALLPRO.CFG

Within text, menu items such as **return to main menu** are bold type each time they appear. Note that there are no capital letters in **return to main menu**, because, in some cases, different letters are capitalized within the item on the actual screen menu. One way to select a menu item is by pressing the first capitalized letter that appears in the item. See the manual for more details on this feature.

Menu items are always bold.

choose library and device

Italics represent variable filenames, such as *filename.SCH*. See the Screen Output section for more information.

Italics represent names.

type in this name

THE ALLPRO SYSTEM

This manual contains information for two similar ALLPRO units: the ALLPRO C and the ALLPRO 88.

The ALLPRO is a software driven device programmer that supports a wide variety of programmable devices, in most applications without the use of external adaptors. The ALLPRO, together with an IBM Personal Computer or compatible machine, becomes an effective engineering workstation. Connection to the computer is by way of a proprietary PC bus interface.

Each pin at the ALLPRO's ZIF socket is driven by software-controlled power sources that produce the signals necessary for the selected device. Control of each pin is handled by software resident in the computer. Details regarding voltage levels, waveforms and timing required by each device are contained in a set of device library and overlay files.

In order to complete the system, the components provided by Logical Devices, Inc. require integration with an IBM PC or compatible computer. Installation involves connecting the ALLPRO to the computer through the interface card and installing the software and library files on disk.

ALLPRO C

The ALLPRO C was the precursor to the ALLPRO 88; it has essentially the same basic architectural characteristics. The only major differences are:

- Discrete Quad Pindrivers: 4 channels / board
10 boards maximum capacity
- No programmable clock or current pulse capability.
- One DIN pindriver output connector on programmer top.

ALLPRO 88

The ALLPRO 88 is an enhanced version of the ALLPRO C with these additional features:

- Flexible System Configuration Options:
Various hardware and software configurations available.
- Octal Hybrid Pindrivers: 8 channels / board
11 boards maximum capacity
- Hybrid Technology: Employed to provide maximum channel/circuit density
- Programmable Clock: Any pin can be clocked, at a rate of up to 4MHz.
- Current Pulsing: Capability available on every socket pin.
(Requires use of Turbo Interface.)
- Two DIN pindriver output connectors on programmer top.

The additional DIN connector on the ALLPRO 88 is located under the module. During normal use, the connectors are not visible, and the ALLPRO 88 exterior appears identical to the ALLPRO C.

ALLPRO INTERFACES

There are, at present, two different interfaces for use with the ALLPRO. These are the standard interface and the turbo interface. Both interfaces can work with either programmer, although each has its own characteristics and restrictions.

Standard Interface

With the standard interface, all timing and control signals are generated from the computer bus. The standard interface is memory-mapped in PC memory at a location that is user definable.

With this interface, the software references the programming waveform timing to the host CPU. This is acceptable for standard applications and devices. However, for timing-critical devices, very fast or slow computers can have a negative impact upon the programming timing specifications.

Turbo Interface

The turbo interface card contains its own processor which can take control of the ALLPRO bus away from the computer. The processor can provide programming control signals at an established set clock rate for devices that are supported on the turbo interface's internal processor.

For device algorithms that are not supported on the turbo interface card, the CPU section of the card is transparent, and the card behaves like a standard interface.

The turbo interface card has the ability to be accessed as standard I/O and not as memory-mapped, thus eliminating memory-map conflicts. This function is not currently employed, and the card is memory-mapped in the same way as the standard interface.

Note: Only algorithms which have been supported on the internal processor of the turbo interface card are immune to timing problems caused by fast (or slow) computer clock speeds.

MODULES

Various modules are available for use with both ALLPRO versions. Modules are units which plug directly onto the ALLPRO DIN connector(s), and should not be confused with adaptors which plug into module ZIF sockets. Modules, in general, provide programmable decoupling to some pins on their respective sockets.

Note: Modules designed for use with the ALLPRO C are upwardly compatible with the ALLPRO 88. However, due to the absence of the second DIN connector, modules designed for use with the ALLPRO 88 are not downwardly compatible with the ALLPRO C.

Some module versions intended for use with the ALLPRO C provided a clock for use with embedded micros (40, 32, 28 and 24 pin); this compensates for the C versions lack of a clock driver.

GETTING STARTED/ INSTALLATION

ALLPRO Systems

Logical Devices offers several different software packages to meet specific programming and budget requirements. The following section details the software package differences; refer to the ALLPRO catalog for descriptions of package hardware differences. In general, the software differentiates between the programming systems; the hardware is interchangeable between systems.

Comprix Total Support Programming System

The Comprix Total Support programming package offers the **complete** set of device libraries available for the ALLPRO. This package provides the ultimate flexibility in meeting programming needs with support for programmable Micros, Memory and Logic devices.

All functions that are supported by the ALLPRO are available with the Comprix software.

Memory Programming System

The Memory Programming System is a cost-effective alternative for the user who presently only needs to work with programmable memory or embedded micro devices. Featuring full upgradability to the COMPRIX programming system and all of the advanced hardware features to work with today's high-density memories.

The Memory software contains support for all of the device types listed in the Comprix package with the exception of PLDs. In addition, certain menu options or functions, as they relate to PLDs, are not available with the Memory package.

PROMs
EPROMs
EEPROMs
PLDs
MICROS

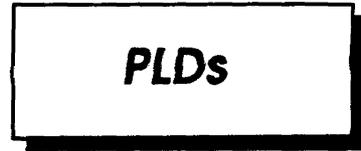
PROMs,
EPROMs,
EEPROMs,
MICROS.

Not available on the Memory system menus:
Functional test device
Read formatted file: JEDEC format
Write formatted file: JEDEC format
Utilities: Automatic test vector toggle

Logic Programming System

The Logic Programming System is an exceptional opportunity for a user on a budget to take full advantage of the flexibility of the ALLPRO 88. With access and support for today's popular PLDs, the Logic System offers the additional advantage of future system expansion.

The Logic ALLPRO software contains support only for PLD devices. In addition, certain menu options or functions that relate to other device types are not available with the Logic package.



The only item available under the **read formatted file** and **write formatted file** menu selections is:

JEDEC format

Items that are not available on the Logic system menus are:

Utilities: Program in sets
Utilities: Range settings
Utilities: Auto serialization

Installation Requirements

DOS Version

The ALLPRO works with DOS 2.0 or greater. However, if available a more current version of DOS should be used.

Drives

The software installation procedure outlined in this manual requires use of a hard drive. However, the ALLPRO may be operated on a floppy-based system consult the factory for details.

The minimum system requires a double-sided disk drive, but two disk drives or a hard drive are recommended when multiple libraries are used. Users with one floppy drive should refer to the Selective Installation of Libraries section.

RAM Memory

A minimum of 256K bytes of PC memory is required to program smaller devices; larger memory devices require more RAM. As a reference, a device 1Mb (bits) large requires at least 640K (bytes) of computer RAM. The ALLPRO base software requires 4Kbytes of memory which must not be used by any other memory mapped device or RAM. If it is intended that 4 Meg or larger devices are to be programmed EMS memory must be used. Refer to the EMS section for additional details.

CPU Requirements

Either an IBM PC or compatible computer is required to operate the ALLPRO. Because the ALLPRO is controlled directly by software executing in the computer, overall performance is determined by the computer being used.

WAIT Signal

The ALLPRO interface uses the WAIT signal line, which is usually available on the bus at the interface card connector. The WAIT signal is primarily used for accessing memory devices. Since the memory access time is typically slower than the processor's cycle time, the WAIT signal causes the processor to wait until the data is ready. If the WAIT signal is active, the processor waits until the signal becomes inactive.

Consult the PC reference manual or technical support for the particular brand of computer being used if this is a problem. Typically, this signal is present on all standard PC compatibles.

A complete ALLPRO operating system requires the following components:

- ALLPRO Programmer
- IBM PC or compatible
- ALLPRO Standard or Turbo Interface Card
- Interconnecting cable, 37 conductor

NOTE:

Hardware and software support is available for the IBM PS/2 computer. Call the LDI factory for details.

CAUTION:

The WAIT signal line must be present or proper communication between the computer and the ALLPRO cannot be guaranteed.

Memory Map Conflicts

If problems are experienced while installing ALLPRO, it is possible that there is a memory map conflict. A typical error message encountered states:

The use of a VGA, network or other expansion cards with the ALLPRO may cause a conflict in computer memory. The memory requirements of these cards may overlap the memory location used by the ALLPRO.

Additional information about memory mapping and resetting the used default memory area are contained in the Technical Reference section and the Configuration File appendix.

ALLPRO is not properly connected to PC

HARDWARE INSTALLATION

Prior to operation, the ALLPRO must be integrated with an IBM or compatible personal computer. The following section details the procedure for accomplishing the hardware installation. After the hardware installation is completed the software installation should be accomplished.

Procedure

The interface card must be installed and connected to the ALLPRO before the power is applied. The following steps detail the procedure for interface card installation.

1. Read the Precautions for User and Machine section before using the ALLPRO.
2. The computer and the ALLPRO must be unplugged from the AC power source before installing the interface card.
3. After power has been disconnected, remove the computer system cover. (Refer to the computer manual for instructions on removing the cover.)
4. Select an expansion slot in the computer. Remove and retain the bracket-retaining screw. Remove the bracket.
5. Install the ALLPRO interface card in the selected expansion slot. Ensure that the card edge connector is properly seated in the expansion slot connector. Secure the interface card with the bracket retaining screw removed in the previous step.
6. Set I/O port jumpers on interface card as necessary. Refer to the interface card section for details on setting these jumpers.
6. Replace the system cover. (Refer to the computer manual for instructions on replacing the cover.)
7. Select a suitable location for the programmer. Place the ALLPRO on a flat surface. Do not cover the unit; the ALLPRO is fan cooled.
8. Install the interface cable. Connect the cable provided from the 37-pin connector on the ALLPRO interface card to the 37-pin connector located on the rear of the ALLPRO programmer.

Refer to the Software Installation section of the manual to complete system installation.

NOTE:

This procedure assumes installation in an IBM or compatible personal computer. If installation is in a compatible computer, the procedure must be adapted according to the machine layout.

WARNING

Make certain that the computer is disconnected from the AC power source before removing the cover.

WARNING

Connect the interface card to the ALLPRO before power is applied.

Precautions for User and Machine

The list of precautions on the right should be followed when operating the ALLPRO, handling devices and using floppy disks. When using devices that must be erased with ultraviolet (UV) light, a UV eraser is required. The section on the next page details the care that should be taken when working with ultraviolet light.

WARNING

Unplug the ALLPRO and the computer from AC power sources before installing the interface card or servicing the equipment. Servicing should be performed only by qualified personnel.

1. The computer and the ALLPRO must be unplugged from the AC power sources before installing the interface card or cables or servicing the equipment.
2. The power to the ALLPRO must be off when installing or removing the header.
3. Make certain that the ALLPRO is properly connected to a grounded AC receptacle that meets power requirements.
4. The interface card must be installed in the ALLPRO before the power is applied.
5. Do not leave a device in the socket when power is switched on or off to the ALLPRO.
6. Insert a device in the socket only when an operation is to be performed and only when the software or LEDs indicate that it is safe.
7. Do not attempt to program or examine a device which is not listed in the library. Select the proper code for the device. Do not change the code while a device is in the socket.
8. The ALLPRO is air cooled. Do not inhibit air flow to the machine.
9. Do not mount homemade adaptors in the sockets.
10. Do not use original (master) disks for everyday use. Make back-up copies, and safely store the originals.
11. Do not use the ALLPRO during severe electrical storms or during any situation in which a power interruption is likely to occur.
12. Devices should be stored in antistatic tubes or on antistatic foam.
13. Be advised that, if formatting of the hard drive is required, all information previously stored on the drive will be lost. All files on the hard drive should be backed-up. See the manual for the computer.

WARNING

Ultraviolet light is dangerous.

Erasable Devices and UV

Certain types of programmable devices can be erased with ultraviolet light and then programmed again. The acronym EPROM, for instance, stands for Erasable Programmable Read Only Memory. EPROMs can be erased using ultraviolet (UV) light. If the device programmed can be erased with UV light, place an opaque label over the device window to prevent accidental erasure.

The wavelength and intensity of light in ultraviolet erasers is dangerous to the eyes, the skin and all living things. What is accomplished in 20 minutes in a UV eraser would take about one week in sunlight or about three years under fluorescent lighting. Never look directly at a lit UV tube.

The length of time required to erase a device varies with distance of the device from the UV tube and the age of the UV tube. Inadequate erasure of a device's data can yield unpredictable results. Normal erasure time is 15 to 20 minutes. Replace old UV tubes if complete erasure is taking an inordinate amount of time.

Several different models of ultraviolet erasers are available from Logical Devices, Inc. Models range from hobby-style units that can erase one or two devices at a time to industrial units that can rapidly erase up to 100 devices at once. Refer to the appendix and to the Sales Department at Logical Devices for more details about ultraviolet erasers.

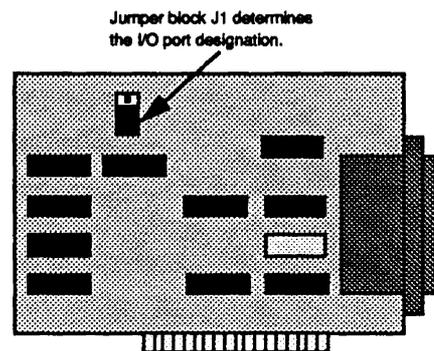
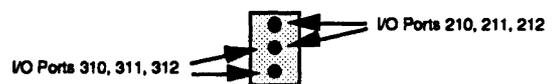
Interface Card

A proprietary interface card that is included with the ALLPRO is installed in the personal computer to provide the interface between the programmer and the computer. Connection from the card to the ALLPRO is via a 37-conductor cable. The section Interface Card Installation provides details on how to install the card. There are two types of interface cards available.

Standard Interface Card

The standard interface card requires a half-length expansion slot. For conditions where the standard interface card must be replaced with a turbo interface card, see the Interface section in the Technical Reference section..

A jumper block is located on the standard interface card. This jumper block is used to connect two pins at J1. The ALLPRO will not function if this jumper is not correctly located. The default setting from the factory is to I/O ports 310h, 311h and 312h. The user can change the settings to I/O ports with hex addresses 210h, 211h and 212h by moving the jumper and changing the UDIPORT command in the ALLPRO configuration file.



Standard Interface Card

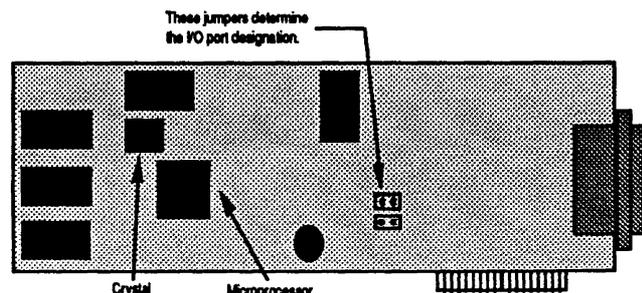
CAUTION

UDIPORT in the ALLPRO.CFG file must agree with the jumper setting on the interface card.

Turbo Interface Card

This card requires a full-length expansion slot. See the Technical Reference section for additional information.

The ALLPRO automatically checks to see if the standard or turbo interface card is installed. For details on particular devices that require the use of the turbo card, see the Device Usage Notes booklet.



Turbo Interface Card

On the turbo card, jumpers J2 and J3 are used to select the I/O ports; they may be changed according to the table to select the I/O designations. The turbo interface card is set at the factory to use I/O ports with hex addresses 310h, 311h, 312h and 313h. The ports can be set by changing the jumpers as shown in the I/O Port Settings table and changing the UDIPORT command in the ALLPRO.CFG file.

I/O Port Settings

I/O Ports Hex Address	Jumper J2	Jumper J3
310, 311, 312, 313	IN	IN
314, 315, 316, 317	IN	OUT
318, 319, 31A, 31B	OUT	IN
31C, 31D, 31E, 31F	OUT	OUT

Module Installation

Several modules are available for use with the ALLPRO, including models with both DIP and PLCC sockets, gold pin DIP sockets or 32-pin DIP sockets. This allows the user to create an ALLPRO that is practically custom-designed for specific devices and use. In addition, there are device-specific modules available, consult the Appendix for these and other accessories for use with the ALLPRO. The words "module" and "header" will be used interchangeably in this text.

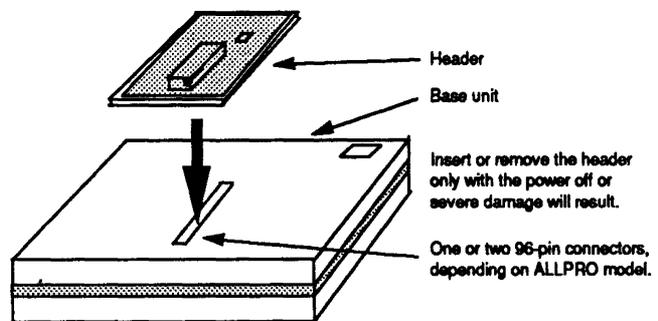
The ALLPRO programmer is shipped with the module installed into the socket on top of the programmer base unit. Under normal conditions, it is not necessary to either install or remove the module. If it is necessary to remove or exchange the module please follow the instructions below carefully. **Never insert or remove the header from the ALLPRO with the power on.** If the power is on and the header is installed or removed, damage to the ALLPRO or the module may result.

The following steps describe the insertion or removal of the module.

1. Turn the ALLPRO power switch to off.
2. Gently lift the module from the base unit. The module is held to the base unit with one or two 96-pin connectors, depending on the model. There are no hardware screws to remove.
3. Install the new module. The 96-pin connectors are keyed and can only be installed in one direction. Carefully align the connectors and press the header onto the base unit.
4. Connect the interface cables to the ALLPRO.
5. The ALLPRO may now be powered-up.

WARNING

Never insert or remove the module from the ALLPRO programmer when the power is on. Damage can result.



Inserting Module into ALLPRO

Note:

Modules designed for use with the ALLPRO 88 cannot be used with ALLPRO C due to the absence of the second DIN connector.

LED Indicators

Common to all modules are red and green LED indicators. These lamps indicate the status of the programming socket(s).

The green indicator is lit when the socket is inactive. During this time, devices may be safely inserted and removed. When the red indicator is lit, the socket is active and high programming voltages may be present. Any devices inserted or removed from the socket during this time may be damaged.

If an error is encountered during programming or verification, the red LED remains lit, and an error message appears on the screen. To cancel the red LED, press any key on the keyboard. Refer to the LED Status Table.

LED Indication Table

The table lists the conditions that are indicated by the red and green LEDs.

The LEDs do not indicate a pass or fail condition. While the red LED may remain lit if a device fails, the red LED only indicates that the socket is active. It is necessary to refer to the message center to determine the actual results of an operation. If an operation is complete and the red LED remains lit, press any key on the computer keyboard to deactivate the socket. The green LED lights, and the device may safely be removed.

WARNING

Never insert or remove a device from the ZIF socket when the red LED is lit.

LED Status Table

Socket	LED Lamps
At Power-Up	green
Active socket	red
Error Condition	red
Inactive socket	green

DIP Socket

The DIP socket that is mounted on most modules can accept devices up to 40 pins large and from 300 mils to 600 mils wide. To insert a device into the socket, the socket handle must be up. Never insert or remove a device unless the software conditions or the LEDs on the panel indicate that it is safe to do so. Voltages may be present which could damage a device. See the Device Insertion/Removal section for details.

Pin routers are available to perform functions on other package types and configurations. See the appendix for information about pin routers, such as those that are available from Emulation Technology.

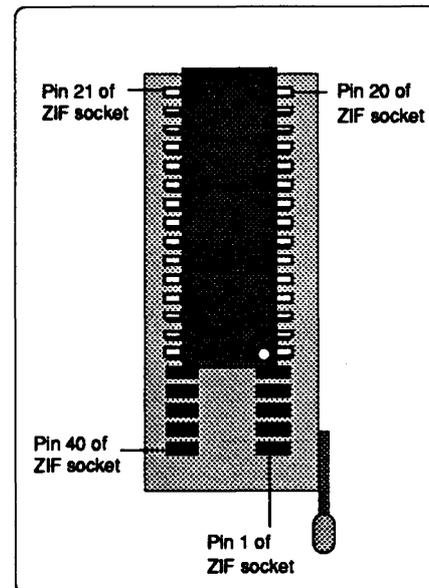
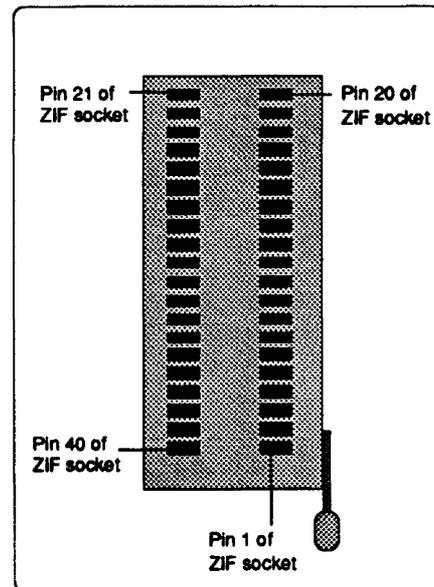
Device Insertion/Removal

The device type and parameters should be selected before inserting the device into the DIP socket.

To insert a device into a socket, the socket handle must be up, so that the pin contacts are physically open. Insert the device and lower the handle. This closes the socket contacts onto the pins of the device and assures good electrical contact. If needed, gently slide the device back and forth in the socket before the handle is completely lowered to remove oxidation on the device leads and ensure good contact.

Never insert or remove a device unless the green LED is lit, indicating that it is safe to do so. If the red LED is lit, voltages may be present that could damage the device.

Devices must be inserted into the socket in the proper orientation. The device position should be justified to the back of the socket (the end away from the lever). The position where pin 1 is inserted is indicated by the pattern surrounding the socket. To locate the proper position for pin 1, identify the line corresponding to the size of the device to be inserted, then follow that line to the socket. Pin 1 on a device is typically indicated by a dot or cut-out mark on the device.



CAUTION

A device is always justified to the back of the socket. When devices have fewer pins than the socket, the DIP socket pins nearest the handle must be empty.

WARNING

Never insert or remove a device from any socket when the red LED is lit.

PLCC Socket

PLCC programming sockets are provided on several modules. The PLCC sockets provided for use with the ALLPRO are compatible with industry standard footprint, height, and shadow. They feature one-step insertion and removal for ease of use. These dedicated sockets prevent cumbersome adaptors from having to be used to program PLCC devices. This section will apply in general to all modules containing PLCC sockets except as otherwise noted in this and other documents. Never insert or remove a device unless the software conditions or the LEDs on the module indicate that it is safe to do so. Voltages may be present which could damage a device.

PLCC Device Insertion/Removal

Device type and parameters should be selected before inserting the device into the PLCC socket. Never insert or remove a device unless the green LED is lit, indicating that it is safe to do so. If the red LED is lit, voltages may be present that could damage the device.

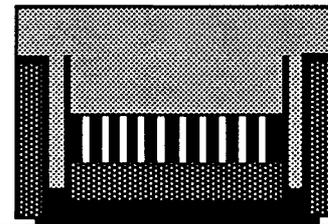
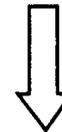
Insertion: Ensure green "READY" LED is lit indicating that it is safe to insert device. The cutout edge of the device should be in the same corner as the cutout edge of the socket. Place package dead bug into the socket and push the package down fully. This will lock the socket into the up, closed position.

Removal: Ensure green "READY" LED is lit indicating it is safe to remove device. Push down on the top of the socket. The package then comes easily and simply out of the socket.

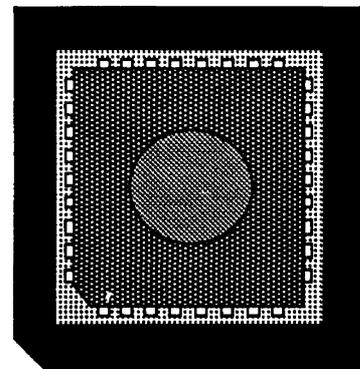
WARNING:

Never insert or remove a device from any socket when the red LED is lit.

Insertion Direction
(Dead Bug)



Side View
PLCC Socket



Pin 1
Top View
PLCC Socket

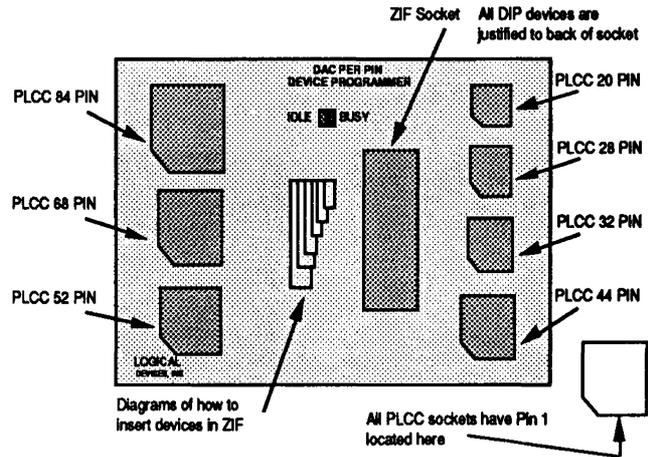
PLCC Module

The PLCC module on the ALLPRO88 is designed to support multiple sockets and a number of different devices in each socket. Because of this, the module does not support decoupling on every pin. Programming signals are applied to all sockets in parallel, so only one device may be operated on at a time. For more details, refer to the technical reference section of this manual.

The figure details a typical layout of the DIP and PLCC sockets on the module. The advantage of this module is that separate socket adaptors are not required for PLCC devices of different sizes.

CAUTION:

Only one device may be inserted into the module sockets at any one time.



ALLPRO PLCC Module

SOFTWARE INSTALLATION

Installation copies files to the working disks or to the hard drive. The original ALLPRO disks are then stored for safe keeping. An installation program is provided to assist.

The ALLPRO software is sent in archived format with the install program. The archived version requires the use of a hard drive. If a hard drive is not available, a non-archived version of the ALLPRO software may be obtained from Logical Devices.

The power switch for the ALLPRO is in the right corner of the top panel. At power-up, the green LED on the ALLPRO is lit. The personal computer and the ALLPRO programmer can be powered up in either order.

Software Description

The ALLPRO is controlled by a program that executes on the personal computer. All operations are initiated from this program. The software system consists of a main menu and sub-menus; selection of certain items causes a sub-menu to appear. Various screens and windows indicate the status of set-up or operation of the ALLPRO. Descriptions of menu items appear under the Menu Selection Descriptions section of this manual.

Menu items are selected either by using the cursor arrows and [Return] key or by pressing the first capital letter in a menu item name. Additionally, in some cases, specific function keys listed on the screen may be used.

Integral parts of the software are the device library database files and algorithm overlay files. Database files describe the device configurations and overlay files contain programming algorithms.

NOTE:

If a hard drive is not available, contact Logical Devices for a non-archived floppy disk version of the ALLPRO software.

.OVR files contain programming algorithms

.DB files contain device configurations

Program Organization

The control program that executes in the computer consists of menus, and database and library management functions. Overlay program segments are included for data editing and hex, JEDEC and binary file read and write. Other overlay programs provide the required algorithms. The overlay is automatically loaded when a device operation is selected.

Computer memory not in use by the control program, resident programs or operating system is available for device data. Sufficient memory must be available to store the selected device pattern. The amount of memory installed in the computer must be considered when programming large devices, such as EPROMs.

Hard Disk Installation

The use of a hard drive is recommended when using the ALLPRO software. After the ALLPRO files are installed on the hard drive, store the disks for safe keeping. ALLPRO now has an intelligent program that automates the installation process. It creates the required directory if it does not already exist, checks for sufficient disk space and optionally installs the IC handler overlay file. The installation program displays instructions on the screen as it proceeds, and it may be used for first time installation or for updates. Installation can be done from any drive, but the following steps assume installation from drive A.

To install ALLPRO, put the original Disk 1 into drive A. Change the default to drive A by typing:

Start the installation program by typing at the prompt:

Follow the instructions on the screen. After the installation is complete, put the original ALLPRO disks in a safe place. Installation may be repeated as often as necessary. In the event that the hard drive fails, it may be necessary to re-install the ALLPRO software from the disks provided.

CAUTION

To prevent unintentional loss of data, always keep back-up copies of all files that are on the hard drive.

WARNING

Formatting the hard drive removes all information previously stored on the hard drive. Use this command with care. Refer to the manual for the computer being used for more information.

A: [Return]

A>INSTALL [Return]

Installation Environment

If an error message is received while running the install program, it is possible that the DOS environment has been modified. The error encountered appears as such:

The install program invokes a second copy of COMMAND.COM. The above error is displayed if the COMMAND.COM cannot be found. The full path name of this file must be specified in the DOS environment with the COMSPEC variable for the install program to execute properly. To set this variable to look for COMMAND.COM in the root directory of the C drive, type:

Selective Installation of Libraries

This section applies to users of floppy drive based systems that wish to operate off one diskette.

Not all the database and overlay files can fit on one disk. In systems that have only one floppy drive frequently used libraries may be selectively installed.

Database files have the format of TTTMMM.DB, where TTTMMM is the library name. The TTT represents type, and the MMM represents manufacturer. See the Library ID codes section for details.

Along with the .DB file, additionally required overlay files for a library are listed in the DEVICESC.TXT text file. To access this file, type:

To print out a copy of this document, type:
(The document is approximately 100 pages long.)

Unarchiving file error - Installation Terminated

```
set COMSPEC=C:\COMMAND.COM
```

To construct a complete library file, begin by including the following files. These files are needed at all times.

Required Program Files

ALLPRO.EXE	PROMEDIT.OVR
ALLPRO.MSG	HEX_LOAD.OVR
MODULE.OVR	DEV_SEL.OVR
JED_LOAD.OVR	ASC_LOAD.OVR
MOT_LOAD.OVR	PAEDIT.OVR
ALLPRO.CFG	INT_LOAD.OVR
CONFIG.OVR	BIN_LOAD.OVR
TEK_LOAD.OVR	

As an example, to install the Lattice PLD library, include the following files listed in DEVICESC.TXT in addition to the above listed base files:

PLDLAT.DB	LAT6001.OVR
LATTICE.OVR	LAT22V10.OVR

In the above files, the Lattice devices are stored in the LATTICE.OVR file, with the exception of the devices GAL6001 and GAL22V10, which are stored in their own files.

```
A>TYPE DEVICESC.TXT
```

```
A>COPY DEVICESC.TXT PRN
```

Expanded Memory System

By using the expanded memory system (EMS), the ALLPRO is capable of programming devices of 4 megabits or larger. The LIM EMS is supported on ALLPRO version 1.50C and later.

Introduction

LIM EMS was developed by Lotus Development Corp, Intel Corp. and Microsoft Corp. (thus the L-I-M). This expanded memory system should not be confused with the extended memory system that is used in personal computers. The expanded memory system consists of a circuit card and a software driver. This package is not manufactured by Logical Devices, but is available from third-party sources.

Requirements

At present, DOS is limited to accessing only 640K of memory. The ALLPRO uses 128K of RAM, and the operating system and drivers use another substantial portion. Because of these memory restrictions, devices of four megabits and larger require that the EMS be used to provide additional memory. If the EMS is not installed, the system indicates that there is insufficient memory available.

Unlike a standard memory system, the expanded memory system is divided in pages and a page frame. Each page is 16K bytes, and the page frame consists of four pages, for a total of 64K bytes. The page frame is located above the 640K DOS limit and below the 1 meg addressing limit.

More information on the EMS expanded memory system may be found in DOS manuals.

RAM Disk and LIM EMS

RAM disk space can be allocated in the computer memory. There are restrictions, however, for distributing data in the RAM disk. If a device requires a single block of memory and a check of RAM disk indicates that there is sufficient space available, it is still possible that the RAM disk may not have the storage capability. In these cases, expanded memory is required. Refer to the section RAM Disk Installation in this manual and

to the DOS manual for a more thorough explanation of the RAM disk.

Extended Memory and LIM EMS

For computers that are equipped with extended memory, drivers are available that emulate LIM EMS. The driver permits the extended memory on the computer to be used as expanded memory.

Installation of EMS

Follow the installation instructions provided with the EMS card. With power disconnected, install the card into one of the computer's expansion slots. Power up the system and install the software.

After a device is selected using the ALLPRO software, the ALLPRO attempts to allocate DOS memory. If DOS reports that insufficient memory is available, an attempt is made to locate the EMS driver. If the driver is not found, an insufficient memory message appears. If the driver is found, ALLPRO tries to allocate memory for the device. If not enough EMS memory exists, ALLPRO reports insufficient EMS memory for the selected device.

If adequate space is found in memory, normal ALLPRO operation continues.

RAM Disk Installation

To speed up program execution, a section of computer memory can be designated as a RAM disk. The advantage of using RAM disk is that, although it behaves like a hard drive, access time is ten to twenty times faster. RAM disk can either be placed into extended memory above 1 Meg or into low memory below 640K. In either case, sufficient memory must remain to allow the operating system and ALLPRO software to be loaded.

NOTE: RAM disk storage is volatile, and data is lost when the computer is reset or turned off. After the RAM disk is created, use a batch file to copy the ALLPRO software and libraries from permanent disk to RAM disk. For a complete explanation of the RAM disk utility, refer to the DOS manual for the computer being used.

Opening Display Screen

The opening display screen appears when ALLPRO is executed. The screen may be bypassed by pressing any key before or during the display. To permanently disable the display screen, delete or rename the LOGO.OVR file. It may be renamed LOGO.BAK.

QUICKSTART

Examples

This section includes two examples of programming with the ALLPRO designed to help the user become familiar with the basic operating concepts. Concepts used here apply to most programming applications.

Blank devices are included with the ALLPRO so the user may get started immediately. Both sample devices are in DIP packages and use the DIP ZIF socket.

PROGRAMMING EXAMPLE 1

In the first example, we will learn how to blank check, program and verify a 27256 EPROM.

Turn on the ALLPRO.

From the host computer keyboard, call the ALLPRO program. At the DOS prompt, type:

The main screen appears:

Use the cursor control keys to select the entry:

When the entry is highlighted, press:

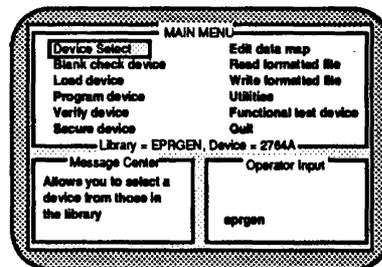
Use cursor and return keys to select:

The screen shows:

Use the cursor keys to select the library and device.

The green LED is illuminated.

C:\ALLPRO <return>



* Device Select

<return>

* Choose library and device

Search for all libraries

NOTE:

If the first three letters of the library name are known, entering them makes the search much shorter. In this example, EPR.

EPRINT

Intel EPROM library

The screen shows an extensive list of Intel devices. From this list, select the following:

At this time, the selected device entry, including the designation of DIP as the active socket, appears in the center of the screen.

The data file is now read into the computer data buffer (ALLPRO does not have internal memory). The ALLPRO software includes two example files called EX1.HEX and EX2.JED. The first is the data file loaded at this time. From the main menu, select:

The data file supplied with the ALLPRO is in Intel hex format, so select:

Now a list of all .HEX files is displayed and from this list select:

Press the F10 function key to indicate parameter modification is complete, and the file may be loaded. The message center displays:

The cursor control keys are used to select:

Under this menu, select:

The message center shows:

This is the valid checksum of the data contents now in the computer buffer.

10325 EPROM_IT_27256 Intel 27256-Intelligent

NOTE:

Select the Intel library, even though the sample chip included with your ALLPRO may have a different manufacturer's name.

* Read formatted file

* Intel hex format

EX1.HEX

NOTE:

Do not modify any parameters in the example at this time. You will modify them later when you use the advanced programming capabilities of the ALLPRO. The defaults are correct for this example. For information, see **Read Formatted File** section later in this manual.

Loading Intel hex file
Please wait

* Utilities

* Checksum

Checksum = 9496

NOTE:

The correct value for this example is 9496. If this value is not shown, the files have been corrupted.

The 27256 device is now inserted into the ZIF socket, notch down (towards you) and positioned towards the back of the socket (away from you). See the illustration:

Pull the handle down to secure the device.

To ensure that the device has not been previously programmed, select:

This checks that all cells in the device are blank. The completion scale is displayed, and when completed, the message center shows:

At this point, the data to be programmed is in the computer buffer and an unprogrammed device is in the ALLPRO ZIF socket. The next step is to program the device. From the main menu, select:

The message center reports:

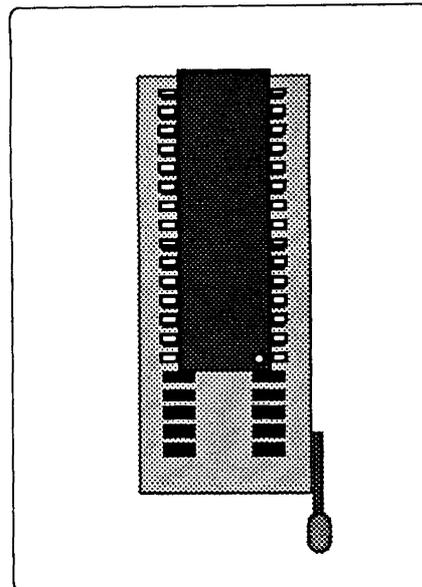
Elapsed time is displayed on the Performance Monitor screen. This device normally programs in approximately three minutes. In addition, the completion scale is displayed along with a Max tries display. During programming:

After programming, the device is automatically verified and the checksum calculated. The checksum should be 9496, the same value as observed previously.

✓ **The EPROM is now in a programmed state.**

It is safe to remove the device from the socket when the green LED is lit.

end of example



* **Blank check device**

Device is blank

NOTE:

If device is not blank, erase it and try again.

* **Program device**

Programming device

The red LED is illuminated

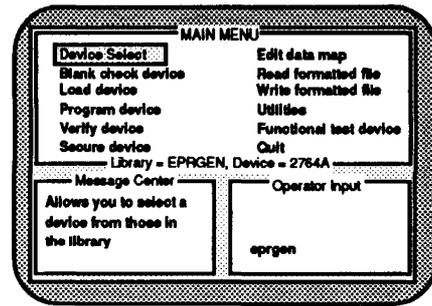
WARNING

Do not touch the device while the red LED is lit.

The green LED is illuminated

PROGRAMMING EXAMPLE 2

In this example, we learn how to blank check, program, verify and functionally test the TI TIBPAL16L8-25 PLD device.



Type:

```
ALLPRO <return>
```

The main screen appears. Select:

```
* Device select
```

Select:

```
* Choose library and device
```

At the prompt, type:

```
PLD
```

When the library list screen for PLDs appears, select:

```
PLDTI_ TI PLD Library
```

From the resulting list, select:

```
10385 TIBPAL16L8-12/15/25 Octal 16 Input And-Or...
```

From the package menu, select:

```
20 pin DIP
```

Message center displays:

```
Select Package Type
```

Display at center of screen reads:

```
Library = PLDTI_, Device = TIBPAL16L8-12/15/25-DIP
```

At this time, the data file is read in from the computer. Select:

```
* Read formatted file
```

Choose file named:

```
EX2.JED
```

Message displays:

```
Loading JEDEC file
```

A moment later, the display changes to:

```
Checksum = 73ca
```

NOTE:

The correct value for this example is 73ca. If this value is not shown, the files have been corrupted.

At this time, the device is inserted in the socket, notch down and rear justified, as shown in the illustration.

From the main menu, select:

The message center reports:

Select:

After the device is programmed, the message center reports:

The ALLPRO provides the capability of testing devices, if test vectors are available. In this example, test vectors are included in the .JED file. From the main menu select:

Select:

The message center displays:

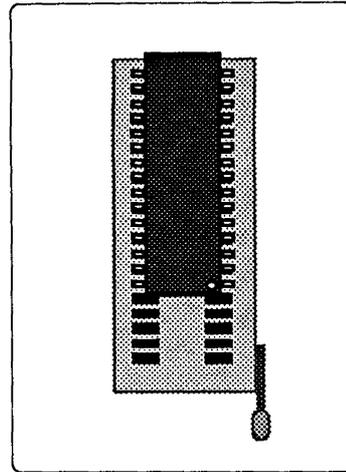
Select:

The message center reports:

Press any key for main menu:

The screen displays a graphic of the device, showing the state of each pin. See the Functional Test section of Operations for explanations.

The next few steps are included to illustrate the effectiveness of securing a device.



* Blank check device

Device is blank

* Program device

Checksum = 73ca

Note:

If a different checksum is obtained the device will not verify after programming.

* Functional test device

EX2.JED

Single step (Y/N)?

N

Passed functional test

<return>

Select:

The message center reports:

This confirms that an unsecured device can be selected and verified.

From the main menu, select:

The message center reports:

After Secure, any attempts to verify or load the device will not be successful. From the main menu, select:

The message center reports:

confirming that the device is indeed secure.

Even though the device cannot be verified or loaded, it still is a functioning device. This is illustrated as follows:

Select:

Select:

Single step (Y/N)?

The message center reports:

end of example

* Verify device

Device verified
Checksum 73ca

* Secure device

Reading device
Device is now secure

* Verify device

Device did not verify

* Functional test device

EX2.JED

N

Passed functional test

OPERATIONS

This section is logically arranged in an identical structure as the main menu to enable the user to quickly locate a specific menu item or section.

User's Interface

The screen is divided into three display areas: the main display, the message center and the operator input area.

Main Display

The main display area presents menus or work space as appropriate for the operation. This area alternates between selection menus, selection lists and work space.

The top boundary of each display area contains a title block, while the bottom border of the main display window contains information pertinent to the current operation. On the main menu, the currently selected library and device type is displayed. During other operations, the contents of this line may change.

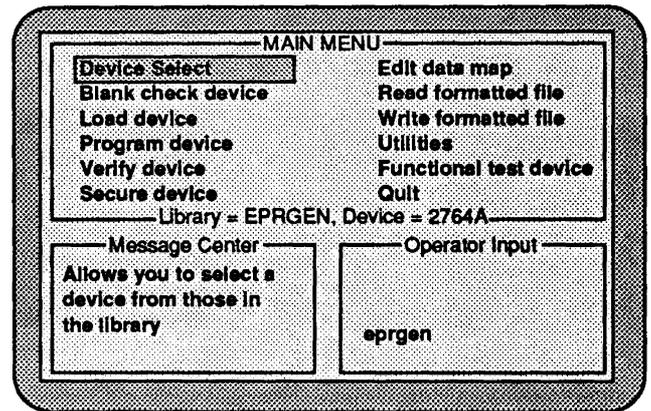
Message Center

The message center displays status messages, error messages or instructions relating to the current operation. The message center is located on the lower left portion of the screen. This area should be monitored for instructions and error messages during normal operation.

Operator Input

Operator input is either via keyboard entry or the cursor arrows. Use the keyboard to type information in; input is displayed on the operator input window. Additionally, use the cursor arrow keys to highlight a selection, and then press the [Return] key.

The operator input window is located on the lower right side of the display screen. This area contains the operator prompt when keyboard entry is expected. Previous entries are scrolled up as new entries are made.



User's Interface

NOTE:

When a menu item is highlighted, a description of that item appears in the message window.

The main menu illustration shows the three display areas that appear on the screen. In this case, the user has already selected a 2764A EPROM. The message center text changes when different items are highlighted on the main menu.

The highlighted text points to selectable items. Movement between items is accomplished by depressing the cursor movement keys (up arrow, down arrow, right arrow and left arrow) until pointing to the desired item. The operation is selected by depressing the [Return] key. When a menu item is highlighted, a description of that item appears in the message window.

Quick Select of Items

A hot key may be used to quickly select a menu item. The first capital letter of each item designates which is the hot key for that item. Press the hot key to immediately choose the operation.

The hot key is designated by the capitalized letter in a menu item.

aUto serialization

└─ the hot key is **U**

Cchecksum

└─ the hot key is **C**

Performance Monitor

During load, blank check, verify and program operations, the Performance Monitor screen is displayed. A Completion Scale bar graph indicates the percentage of operation completed. This graph is displayed only if the device selected is an EPROM, PROM or MICRO with a data area of 1K or more. Smaller size EPROMs, PROMs and all PLDs do not display this graph.

The Performance Monitor also displays the elapsed time while an operation is being performed. A Maximum Retries scale is also displayed for certain device operations.

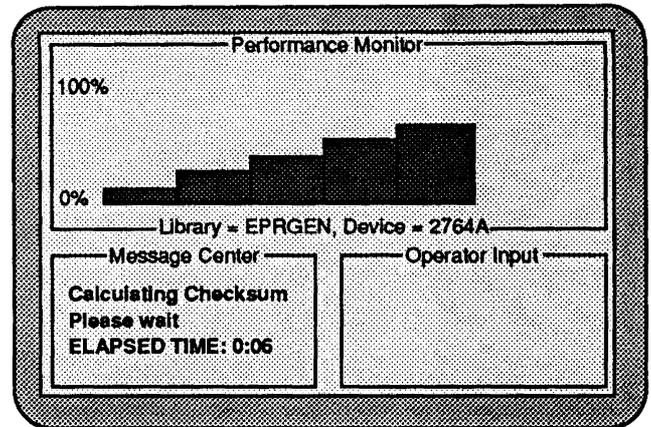
Max Retries Scale

The Max Retries (Maximum Retries) scale indicates the number of attempts required to successfully program a device. The Max Retries scale appears on the Performance Monitor screen.

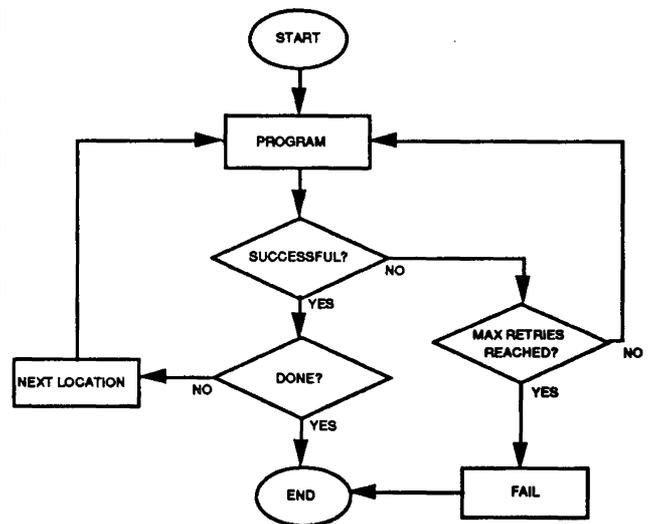
When programming devices for which the manufacturer allows more than one programming try per location, the Max Retries scale appears to the right of the Completion Scale on the Performance Monitor screen. This screen appears only for memory devices of 1K and larger. After programming is successfully completed, the Max Retries scale indicates the greatest number of attempts that was required to program a bit or byte in the device.

The maximum number of programming retries allowed varies by device. The Max Retries scale, however, displays only up to 17 retries, and this number is not user selectable. The Completion Scale only advances after a bit or byte (depending on the device) is successfully programmed. If programming is unsuccessful after the maximum number of retries is reached, programming is abandoned and the device is failed.

The flowchart shows the sequence followed by the max retries (maximum retries) algorithm. Notice that a programming attempt is made on the next location only if the previous location is successfully programmed.



Performance Monitor Screen



Maximum Retries Algorithm

ALLPRO Main Menu and Sub-Menus

The table may be used as a ready reference for locating items on the main menu and sub-menus. The menu items that are followed by an asterisk (*) do not appear on the screen as menu items. They are screens that automatically appear following the completion of a menu item. See the text for the particular item for a more detailed description.

Selecting **device select**, **read formatted file**, **write formatted file** and **utilities** from the main menu causes a sub-menu to be displayed.

The table of menu items shows the menu items that are available from the main menu. If a menu choice pulls down a sub-menu, the sub-menu choices are shown in the second column. The capitalized letter in each item is the letter that may be entered to select that item.

Certain sub-menu items may cause another sub-menu to be displayed. These menus are usually parameter menus, such as those for set programming, and are detailed in other sections of this manual.

Other sub-menus may display a list of filenames. If a list appears, and no filename is highlighted, press the Escape [Esc] key to cause the highlight to appear, then use the arrow keys to move the highlight.

The [Esc] key is also used to return to the Main Menu window from the Operator Input window on the Main Menu display.

How Descriptions Are Arranged

The text on the following pages that describes the items on the menus is arranged in the following manner.

The first item in the text is the first item that appears on the main menu. The items that follow in the text are all the items that appear on the sub-menus under the main menu item. After all sub-menu items are described, the next item on the main menu is described, followed by its associated sub-menu items, and so on.

Table of Menus and Sub-Menus

Main Menu	Sub Menu	Sub Menu
Device select	Choose library and device	Package Select*
	Select device	Package Select*
	Reserved	
	Return to main menu	
Blank check device		
Load device		
Program device		
Verify device		
Secure device		
Edit data map		
Read formatted file	Intel hex format	
	ASCII hex format	
	Binary format	
	Straight hex	
	Motorola hex format	
	JEDec format	
	Tektronix hex format	
	Return to main menu	
Write formatted file	Intel hex format	
	ASCII hex format	
	Binary format	
	Straight hex	
	Motorola hex format	
	JEDec format	
	Tektronix hex format	
	Return to main menu	
Utilities	View/change working directory	
	Specify library path	
	Data initialization toggle	
	Edit file	
	Program in sets	
	auto secure Toggle	
	Automatic test vector toggle	
	Range settings	
	auto serialization	
	Checksum	
	return to Main menu	
	Handler operations	
	Functional Test Device	
Quit		

NOTE:

The Main Menu and Sub-Menus list the operations supported by the ALLPRO. Depending on the ALLPRO software being used, certain items may not appear on menus. For a description of the various software systems that are available, see the Getting Started/Installation section in this manual.

NOTE:

Most items on the menu require that a device be selected from the libraries before the menu item is selected.

In the case of certain menu items, such as **return to main menu**, these items are described only once, although they appear several times throughout the menus.

Device Select

Select a manufacturer and a device from those in the libraries

When this item is selected, a sub-menu appears. Device selection is a two-step process.

The first step requires that the library expected to contain the desired device be selected. On the sub-menu, **choose library and device** displays each set of library files found. The path searched for library files may be specified in a **utilities** menu operation. Initially, this path is the path specified in the configuration file.

To select a library, point and select the desired item, either by using the arrow keys or by typing in the first six letters of the LDI device library filename. See the LDI Unique Six-Character Library Filename Code section for information about the device library filename conventions.

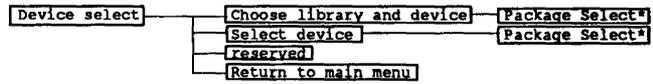
The second step requires that the specific device be designated. The device may be selected with the arrow keys, or the device code may be keyed in from the keyboard. See the LDI Unique Five-Digit Device Code section for information about the device code format.

If a library has already been chosen, then the **select device** item on the sub-menu may be selected. This displays all devices available under the specified library.

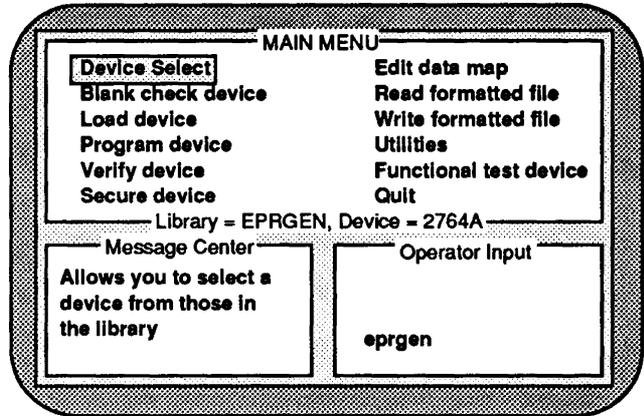
Device Libraries

Once a device is selected by the user, the ALLPRO accesses the characteristics of the selected device from a library file. The library files describe the electrical characteristics and operational sequences that are used to program or read a device.

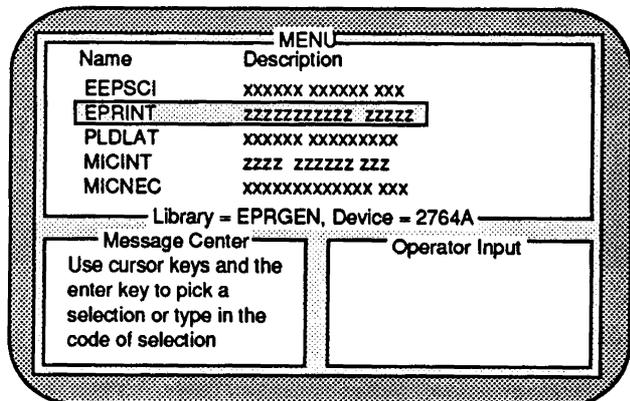
The ALLPRO provides a full range of electrical characteristics for each device pin. The software configures pins according to the library description and then sequences through the steps to program, read and verify devices. User selection of a device causes the ALLPRO to read a library description entry and set up the parameters for the device. Libraries providing support for most programmable devices are available.



Device Selection Menu Tree



Main Menu: Selecting a Device



Sub Menu: Highlighting a Desired Library

LDI's Unique Codes

To facilitate the process of selecting devices, unique codes created for internal use by Logical Devices may be used. These codes only are used with equipment from Logical Devices, Inc. and are detailed below.

LDI Unique Five-Digit Device Code

Logical Devices has assigned a unique five-digit device code to each device in the libraries. This code is used in a field which specifies a particular device in the selected library.

A list of all the devices currently supported by the ALLPRO, and their corresponding LDI device codes, is displayed when select device is chosen on the main menu. These codes are also listed in the Logical Devices Master Device List booklet.

LDI Unique Six-Character Library Filename Code

Each device library consists of a database file (file extension .DB) as well as program containing the device algorithms.

The DOS command DIR may be used to determine which libraries are available.

The library filenames are coded such that the first three characters represent the device library type, and the last three characters represent the manufacturer. If specified, this library is automatically selected at execution time.

There are many three-character device manufacturer types MMM designated by Logical Devices. If the manufacturer type is unknown, the listing may be accessed as follows.

Choose device select from the main menu, then choose library and device. The message center window lists the five device library types. Type in a library type, such as EPR, then press [Return]. All supported manufacturers of EPROMs have the manufacturer type codes listed.

Some sample codes are:

10315 2764 EPROM in Intel family
10540 16L8 PAL in AMD family

Libraries are referenced by six-character codes:

TTTMMM

MMM is the manufacturer
TTT is the device library type

Device Library Types

EER	Electrically Erasable PROMs (EEPROMs)
EPR	UV Erasable PROMs (EPROMs)
MIC	Microcontrollers with on-board EPROMs (MICROs)
PLD	Programmable Logic Devices (PLDs)
PRM	Programmable Read Only Memories (PROMs)

NOTE: The code must contain six characters. If fewer than six letters are in the code, the remaining characters are underscores (_).

Examples of library filenames:

EPRINT	EPROMs from Intel device library
PLDNS_	PLDs from National Semiconduc tor device library
none	No library chosen

Choose Library and Device

Select a manufacturer and device from the libraries.

Prior to selecting a device, a device library must be chosen. Each library contains details on a series of components and contains related devices from a single manufacturer. For example, AMD PAL10L8 and AMD PAL16R8 are in one library, whereas Intel EPROMs are in a different device library.

Refer to the LDI Unique Six-Character Library Filename Code section for details about using the codes for faster library searching and device type selection.

When **choose library and device** is selected, a search for device library files is initiated. To shorten the search time, one of the three-character device types listed can be entered before searching, and only libraries under that device type are displayed.

Once a device library is selected, the library code is displayed on the bottom border of the main display area. Each library is identified by the first six characters of the library filename.

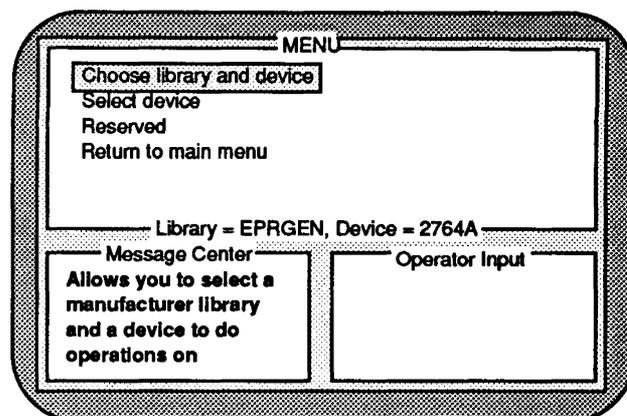
Select Device

Select a device if manufacturer is already selected.

When choosing a device, devices in the selected library are listed in the main display area. Each line displays the LDI device code and the device name and description. If more devices are included in the library, they may be displayed by pressing the page down [Pg Dn] or page up [PgUp] keys or the arrow keys.

Selection is made by pointing and selecting the desired device with the arrow keys.

As an alternative to locating the device in the displayed list, the device's identifying code may be entered from the keyboard when the list appears on the screen. Keyboard input is displayed in the user-input window. Refer to the LDI Unique Five-Digit Device Code section for information about using the device codes for faster device selection.



Sub Menu: Choose Library and Device

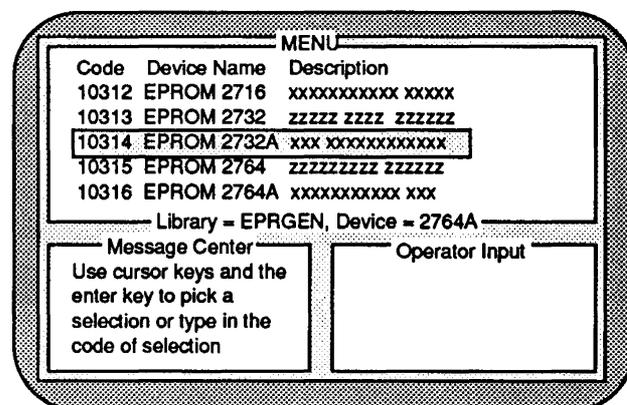
For example:

If **EPR** [Return] is typed in, ALLPRO searches all EPROM libraries.

After searching, libraries and their descriptions are displayed alphabetically by library names.

To select a library, use the arrow keys, or the LDI six-character library code may be typed in.

To rapidly select the library or library and device family, type in the three or six letter codes as described above.



Selecting a Particular Device Type

Only devices contained in the selected library may be chosen. If the desired device is not contained in the selected library, the correct library should be specified.

During device selection, the internal memory buffer used to store data is initialized to a default state. If desired, however, this initialization may be overridden by setting the **data initialization toggle** off in the **utilities** menu. See the data initialization toggle section for more details.

Package Select

Select package style for the device selected.

This item does not appear on the screen as a menu item; it is a screen that automatically appears following the selection of a device.

After the library and device type are selected, the Package Select screen appears. The package types that are available for the selected device are listed on this screen. If a device only is available in one package, the list of package types does not appear.

Highlight the desired item and press [Return] to select.

Reserved

No function currently available.

This is an item reserved for future use. Do not select this item, as it has no application at the present time.

Return to Main Menu

Leave current screen. Go to main menu.

Selecting this item causes the main menu to appear. If this option is not offered, then pressing the Escape [Esc] key may return to the previous menu. In some instances, keys other than the Return or Escape key (such as the **F10** function key) cause a return to the main or previous menu. If so, this is noted on the screen.

CAUTION

Part numbers for different devices are often very similar. Select the proper device code before examining or programming a device. Attempting to program or test a device using the wrong device code may result in damage to the device.

When a device is available in more than one package, the following is typical of the selections that appear:

Key	Size	Package	Description
0	20 Pin	DIP	
1	24 Pin	PLCC	

Description may appear with additional comments when a device is available in two or more similar packages.

WARNING

When using PLCCs with a PLCC adaptor that is plugged into the DIP socket, the package style selected must be DIP, not PLCC.

Blank Check Device

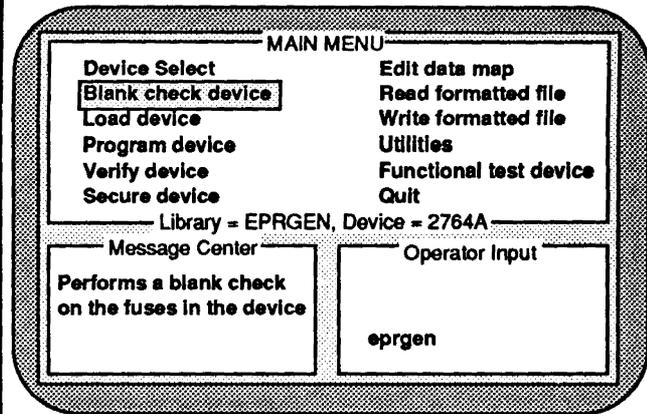
See if the device in socket is in an unprogrammed state.

Every device has a state called a blank state. In some devices, a blank state occurs when all bits are set to zero; in other devices, a blank state is defined as all bits being set to one. New devices that have never been programmed are usually in a blank state. Devices that have been completely erased after having been programmed are in a blank state. Depending on the device, a blank state may be shown as all ff (hex) or 00 (hex) on a fuse map.

Blank check verifies that a device is in an erased or unprogrammed state. Each valid fuse or location is compared against the expected unprogrammed condition for the selected device. If the device does not verify as being blank, an error condition is displayed in the message center.

Blank check completes with one of two possible results. A message is displayed indicating either that the device is blank, with no error conditions, or that the device is not blank.

The error message includes the device array in which the error was encountered. The address in error is displayed as a hex value for memory devices and in an input/product/output format for PLDs. The device data is displayed in parallel format as read from the device.



Main Menu: Blank Checking a Device

NOTE:

Blank Checking EEPROMs

The algorithm for EEPROMs performs an erase prior to programming the device. Because of this, blank checking an EEPROM is a useless procedure. EEPROMs also can be programmed to a blank state.

A manufacturer may leave a checksum in an EEPROM, anticipating that it will be automatically erased before programming.

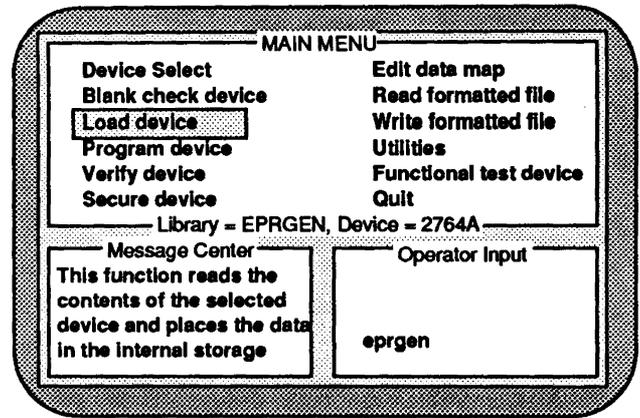
Load Device

Copy the data pattern from device in socket to computer buffer.

Load device copies the patterns programmed in a device into the internal computer memory. This data may then be programmed into new devices or displayed and modified with the data map editor or stored to a file for later use.

After loading the data pattern, the checksum of the data read is computed and displayed in the message center window. For PLDs, the checksum is calculated in accordance with JEDEC Standard No. 3B. For memory devices, the checksum is the 16-bit value calculated by summing all valid memory contents.

Refer to the Checksum in the Utilities section for more details.



Main Menu: Loading a Device

Program Device

Place a new data pattern from computer buffer into the device in socket.

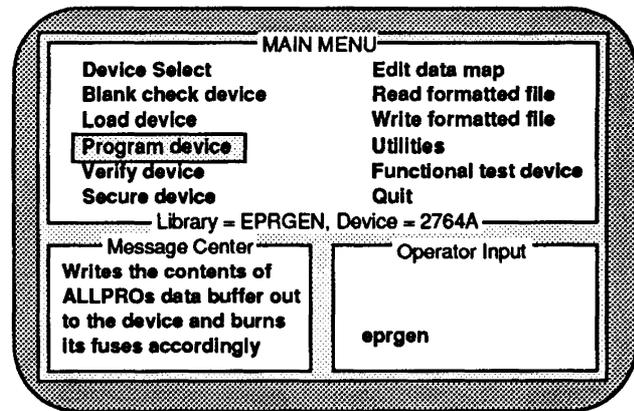
This selection programs the data pattern stored in memory into a device. Data stored in memory (diskette or hard drive) is placed into the computer buffer and then programmed into the device. After programming is complete, verification is performed per the manufacturer's specifications, including both a high and a low Vcc margin verify, if needed.

Prior to programming, a blank check is recommended, to ensure that the device does not contain previously programmed cells.

If an error condition is detected during the post-program verify, the message window indicates the array and address in error and both the expected data (RAM data) and the actual data (device data). The address is displayed as a hex value for memory devices and in input/product/output format for PLDs. Device data is displayed in parallel format as read from the device.

If no verify error is detected, a message so indicating is displayed.

Red and green LEDs on the ALLPRO also indicate the status of the socket. If an error is encountered during programming or verification, the red LED remains lit, and an error message appears on the screen. To cancel the red LED, press any key on the keyboard. Refer to the section in this manual that describes the LED status conditions.



Main Menu: Programming a Device

CAUTION

If a key is pressed during an operation, the operation is cancelled. Do not touch or remove a device during an operation.

Verify Device

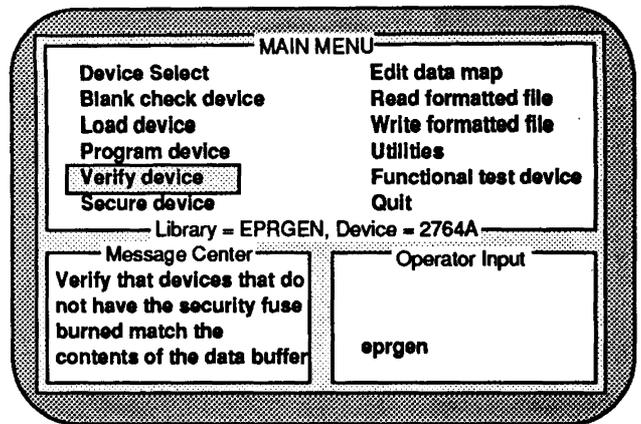
Assures that data in the device matches data in the computer buffer.

Only devices that do not have the security fuse blown can be verified. Not all devices have a security fuse. For more details, see the Secure Device section.

The verify operation requires that the exact data pattern or file that was used to program the device be resident in the computer buffer.

The verify procedure validates the pattern read from a device against the current pattern in memory. If recommended by the device manufacturer, a verify pass is done for both the high and low Vcc margin. This type of verify detects marginally programmed or erased cells that may not be found during a standard verify at normal Vcc levels. If a discrepancy is found, the message center indicates the array in which the error was encountered, the location or fuse in error, and both the expected data pattern (RAM data) and actual data pattern (device data) read.

If no verify error is detected, a message so indicating is displayed.



Main Menu: Verifying a Device

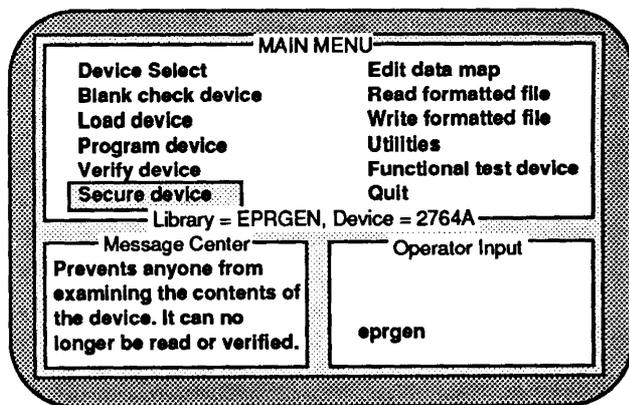
Secure Device

Disables the device so that its contents can no longer be examined or modified.

The secure device function prevents the pattern programmed into the device from being read or copied. This feature is not supported by all devices. Consult the manufacturers' data books to check the availability of the secure option.

To ensure that the security fuse has been blown, the **secure device** operation is preceded by a read of the device and followed by a verify. Security is confirmed when valid data can no longer be read or verified against a previously read pattern. Once a device has been secured, it is seen as a blank chip even though it is not actually blank.

Additional information about automating the secure process is available in the Auto Secure Toggle section.



Main Menu: Securing a Device

CAUTION

Once a device is secured, it cannot be unsecured to read the data in the device. With erasable devices, the data is erased when the security bit is erased.

NOTE:

The Secure Device function is not supported by the ALLPRO for all devices that have this feature.

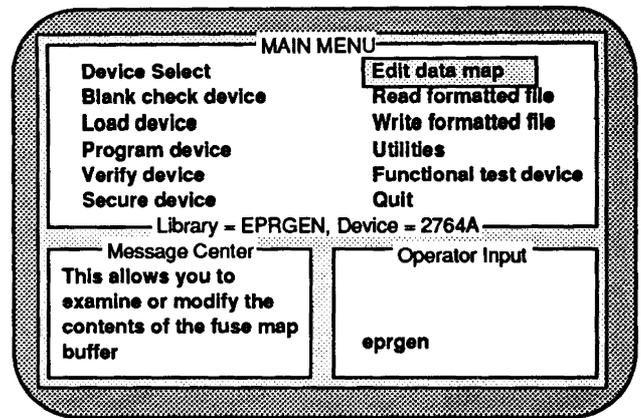
NOTE:

Some devices have an encryption array which is programmable. Refer to the Device Usage book for more details.

Edit Data Map

Examine or modify data in the computer buffer.

A data editor is available to modify data in memory. The editor has different features, depending on whether an EPROM or a PLD is selected. The Edit Data Map, Edit PLD Data Map and Byte Swap Function sections explain editor operations for each device type.



Main Menu: Edit a Data Map

Device Arrays

Many devices contain groups of fuses or addressable locations that are organized into different functional arrays. For example, an EPROM may be page addressable, where each page is considered to be a different array, whereas a PLD may have four arrays: an AND/OR array, an OUTPUT ENABLE array, a PRESET/RESET array and an OUTPUT ARCHITECTURE array.

Generally, an array represents fuses or locations that are grouped according to the addressing arrangement in the device. In some PAL or PLD devices, the array groups differ from the JEDEC fuse numbering order. When viewing or modifying fuses in PLDs, note that the JEDEC fuse number may be displayed non-sequentially.

Editing a Data Map

This section applies to PROMs, EPROMs, EEPROMs and MICROs. If PLDs are being edited, see the Edit PLD Data Map section.

When memory and MICRO devices are being edited, the main window displays data in hex format on the left side and in ASCII format on the right. The hexadecimal address of the first word of each line is displayed left of the hex data display.

Each hex data pattern is displayed in device words. If an 8-bit device is being edited, one byte is displayed in each group. If the device is 16-bit, it can be displayed in groups of one or two bytes. This feature in the editor is referred to as word addressing and can be invoked using the F4 key.

The ASCII format equivalent of the hex pattern is displayed on the right side of the screen. The ASCII character for hex data 00 is displayed as a period (.). During an edit session, additional information is presented on the lower border of the main display area. The left portion displays the hexadecimal address of the cell currently being pointed to. The right portion displays the array number or the page number, if a multiple-paged device is selected.

Cursor arrow keys position the cursor to a desired location. The [PgUp] and [PgDn] keys cause the previous or next set of data bytes to be displayed. The [Home] and [End] keys position the cursor to the beginning or end of the device pattern. If the device has multiple pages or arrays, the [Home] and [End] keys apply to only the current page or array. Changes are accepted as the hex characters 0 to 9, A to F and a to f.

Data editing features can be invoked by using the following function key sequences:

- F1** displays one of three help screens. Subsequent pressing of the F1 key displays the next help screen.
- F3** fills memory with a specified byte. When F3 is pressed, the ALLPRO prompts for a two hex digit (one byte) data pattern. If the range is not set, the entire memory reserved for the device is filled with this pattern. If the range is set, only that part of memory bounded by the range is filled with the specified byte. If a multiple-paged

Data Map Editor Commands

F1	Help screen rotate
F2	Not used
F3	Fill <2 hex digit>
F4	Word addressing (if 16-bit device)
F5	Goto <address>
F6	Search <string>
F7	Previous array (if multiple array device)
F8	Next array (if multiple array device)
F9	Search next
F10	Done
(Alt)-F1	Set range <address>
(Alt)-F3	Begin range
(Alt)-F4	End range
(Alt)-F5	Reset range
(Alt)-F6	Swap nibble
(Alt)-F7	Fill low nibble <1 hex digit>
(Alt)-F8	Fill high nibble <1 hex digit>
(Alt)-F9	Swap bytes

device is selected, this function fills all of the pages (arrays).

- F4** toggles the display between byte addressing and word addressing. If the device is a 16-bit device, the data can be displayed in both addressing modes. In the byte addressing mode, data is displayed in groups of one byte (two hex digits). In the word addressing mode, data is displayed in groups of two bytes (four hex digits).
- F5** positions the cursor to a specified location. After pressing F5, the ALLPRO prompts for an address. Entry is accepted in three different formats: decimal, octal or hex. The entered number must be immediately followed by a format specifier: n for decimal, o for octal or h for hex. The specifying character may be either upper or lower case. If no specifier character is included, the address is interpreted as hexadecimal.
- F6** locates the occurrence of a string pattern. The search string may be entered in either hex or ASCII format. A hex string must be preceded by a slash (/). ASCII strings must be preceded by a single quote (').
- F7** changes the display to the beginning of the previous page if a multiple-paged device is selected.
- F8** changes the display to the beginning of the next page if a multiple-paged device is selected.
- F9** initiates a search for the next occurrence of the current search string. For this function to operate properly, a search string must be entered prior to execution.
- F10** returns the system to the main menu.

The following commands are initiated by holding down the [Alt] key and pressing the function key. These commands adjust the range, which sets the lower and upper address boundaries for the fill (F3) command.

[Alt]-F1 sets the range. The range is used with the fill command. The ALLPRO prompts for the addresses for the beginning and the end of the range.

Examples Using the F5 Key (Cursor Locate)

To go to a decimal address of 100, press **F5**, then enter 100n (Return).

To go to an octal address of 500, press **F5**, then enter 500o (Return).

To go to a hexadecimal address of 3ff, press **F5**, then enter 3ffh (Return) or 3ff (Return).

If a multiple-paged device is selected, this feature only allows positioning to an address in the current page.

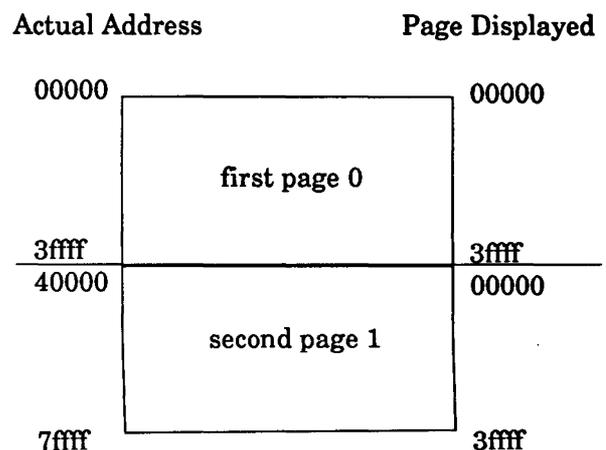
Examples Using the F6 Key (String Search)

To search for a hex string of 0d0a, press **F6** and enter /0d0a (Return).

To search for an ASCII string of any message, press **F6** and enter any message (Return).

Notes on the F8 Key (Page)

When editing devices larger than 2 Meg, the memory is paged. The chart shows the actual addresses and the addresses that appear on the screen. An offset of 40000 hex is added to the addresses on the second half.



[Alt]-F3 sets the cursor position address as the beginning of the range.

[Alt]-F4 sets the cursor position address as the end of the range.

[Alt]-F5 clears the range setting.

[Alt]-F6 swaps the high and low nibbles in all bytes. If the range is set, this operation is limited to the range setting.

[Alt]-F7 fills the low nibbles in each byte with user-entered data. If the range is set, this operation is limited to the range setting.

[Alt]-F8 fills the high nibbles in each byte with user-entered data. If the range is set, this operation is limited to the range setting.

[Alt]-F9 swaps the high bytes with the low bytes within a specified range.

Byte Swap Function

Byte swap interchanges the contents of an even address location with the contents of the next higher odd address location. For example, the contents of address 0000 hex are swapped with the contents of address 0001 hex. This function affects the entire device buffer and only works if the device buffer size is a multiple of two.

Byte swap is useful if, for example, an assembler creates a file in Intel format, in which the low byte is read before the high byte, but the file must be in Motorola format, in which the high byte is read before the low byte.

Byte swap may be used to swap the low bytes and high bytes of 16-bit words (16-bit devices). After selecting a PROM type device, select edit data map from the main menu. The fuse map editor is displayed, and its contents may be modified. The message center has three help screens. Press [F1] until Help Screen 3 appears. This screen displays the available swap functions. Press key [Alt-F9] to swap the bytes. The screen displays the swapped bytes when the operation is complete.

Edit PLD Data Map

When a PLD data map is being edited, each valid intact fuse is displayed as a zero (0), and each blown fuse is displayed as a one (1).

The lower left side of the main window provides the information listed on the right.

Some devices display the PLN in a cumulative fashion, from zero to the maximum number in the device. Other devices display the PLN relative to the output. The lower right side of the main window describes the type of array, such as AND/OR, being viewed.

The fuse number display, normally on the left side of the fuse display, is suppressed when the device-addressable architecture differs significantly from the JEDEC fuse numbering convention. In such cases, the fuse number is only displayed in the FUS: border area.

Cursor arrow keys position the cursor to a desired location. The page up [PgUp] and page down [PgDn] keys cause the preceding or following fuses to be displayed. The [Home] and [End] keys position to the beginning or end of the array. The message center shows which function keys are available.

Each time the space bar is pressed, the state of the fuse at the cursor position toggles from 1 to 0 or from 0 to 1. Optionally, a one (1) or zero (0) may be entered from the keyboard.

PLD Data editing features can be invoked by using the following function key sequences:

- F7** changes the display to the beginning of the previous page if a multiple-paged device is selected.
- F8** changes the display to the beginning of the next page if a multiple-paged device is selected.
- F10** returns the system to the main menu.

PLD Data Map Commands

- F7** Previous array (if multiple array device)
- F8** Next array (if multiple array device)
- F10** Done

PLD Fuse Map Editor Indicators

The position of the cursor on the fuse map changes the values of the following items:

- FUS:** The fuse number underscored on the data map. This is the JEDEC fuse number.
- PLN:** The product line number, displayed per manufacturer's convention.
- ILN:** The input line number.
- OUT:** The corresponding output of the PLN and ILN.

Read Formatted File

Copy a file from storage to the computer buffer for viewing or editing.

The pattern to be programmed into a device can be entered using the data map editor, loaded from a similar device or read from a disk file. The **read formatted file** selection reads a formatted disk file and converts it to fuse or data patterns. When this function is selected, a sub-menu allows selection of the file format to be read. The supported formats are Intel hex, ASCII hex, binary, straight hex, Motorola hex, JEDEC standard data transfer and Tektronix hex.

After a format is selected, a list of filenames is displayed. Press [Esc] to highlight a filename, and then select a filename using the arrow keys. The name of the file that was read or used last for functional test purposes is displayed in the Operator Input window. This is the default filename and may be selected by hitting the [Return] key.

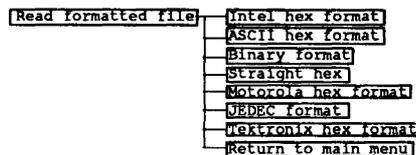
The formatted file being read must be in the directory specified by the working directory. This can be set to any directory automatically by specifying the path in the configuration file, or it can be changed from the utilities menu. For further details, see the file format section in the appendix.

Onscreen File Select

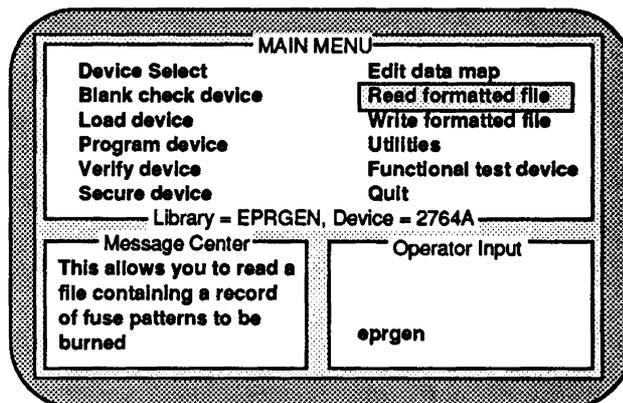
After choosing a file format, the screen displays a list of files contained in the working directory. If Intel, Motorola, Tektronix, ASCII or straight hex file format was chosen, only files with HEX extension are displayed; if JEDEC format was chosen, only files with JED extension are displayed; if Binary format was selected, all files having all extensions are displayed.

To display different files, change the directory mask using the F4 key. Changing the directory mask to *.* causes all files in the working directory to be displayed.

The F3 key displays the files in alphabetical order.



Read Formatted File Menu Tree



Main Menu: Reading a Formatted File

NOTE:

To change directories, type:

C:\

or C:\dirname\dirname

Formats supported for Memory devices are:

INTEL HEX
 ASCII HEX
 BINARY
 STRAIGHT HEX
 MOTOROLA HEX
 TEKTRONIX HEX

Formats supported for PLD devices are:

JEDEC

The files in sub-directories are highlighted. To display these files, move the cursor onto the desired sub-directory and press [Return].

To display the files under the parent directory, which is indicated by two dots (..), move the cursor to the directory and press [Return].

The working directory is set to the directory that was last displayed.

Intel Hex Format (Read)

Select this format to view or edit data pattern.

The Intel hex format contains hexadecimal data coded in an ASCII file. The format contains the load address, data, and a checksum for validation. This format transfers and manipulates data as text.

Intel hex downloads recognize three record types: extended address records, data records and end of file records. When extended address records specify the upper segment base address, the file may contain up to one megabyte of data (20 address bits). Without extended address records, a maximum of 64 Kbytes of data may be contained in the file (16 address bits).

When this format is selected, the filename can either be selected from the list displayed or entered from the keyboard. During a file read operation, only data that fits into the selected device is accepted. For more details on filename selection, see the Onscreen File Select section.

Files may be read for sets of devices used in systems having data paths wider than 8-bit using the **program in sets** menu selection. To read part of a file, change the download parameters according to the Hex Download Parameters section.

ASCII Hex Format (Read)

Select this format to view or edit data pattern.

The ASCII hex format contains 4-bit or 8-bit hexadecimal data in fixed-length records, each containing 16 data entries. A checksum word at the end of the file validates data receipt.

When this format is invoked, the filename can be selected from the list displayed or entered from the keyboard. To read part of a file, change the download parameters according to the Hex Download Parameters section.

Binary Format (Read)

Select this format to view or edit data pattern.

The binary format contains neither the addresses nor the checksum of the file. The file contains the actual binary data. An example of this format is a DOS executable file having a **.COM** extension.

When this format is invoked, the filename can be selected from the list displayed or entered from the keyboard. Since binary format file does not specify the address associated with the data, the file is loaded at the address starting at 0000.

Straight Hex Format (Read)

Select this format to view or edit data pattern.

Straight hex format is generated for PROM type devices. The data may be either 4-bit or 8-bit data, with spaces following each byte. Unlike ASCII hex format, straight hex format does not have the address preceding the data bits.

Motorola Hex Format (Read)

Select this format to view or edit data pattern.

The Motorola hex format contains hexadecimal data coded in an ASCII file. The format contains the load address, data, and a checksum for validation. This format transfers and manipulates data as text.

This selection recognizes S1-S9, S2-S8 and S3-S7 Motorola formats. S1 data records contain 16 address bits, S2 contain 24 address bits and S3 contain 32 address bits.

The filename can either be selected from the list displayed or entered from the keyboard. Sets of devices for target systems requiring greater than 8-bit data paths may be downloaded with the **program in sets** menu selection. To read part of a file, change the download parameters according to the Hex Download Parameters section.

JEDEC Format (Read)

Select this format to view or edit data pattern.

The JEDEC Standard Data Transfer Format defines a data format for transferring open or intact fuse states to a device programmer. This format can also be used for transferring files creating by CUPL™, Logical Devices' Universal Compiler for Programmable Logic.

Three JEDEC field types are recognized by the ALLPRO: the L field, F field and C field. These fields are interpreted in accordance with JEDEC Standard No. 3B. All remaining field types are ignored during a JEDEC file read. Structured test vectors are handled by the **functional test device** menu item.

When **functional test device** is selected, the filename can be selected from the list displayed or entered from the keyboard. Because fuse numbering in the JEDEC file is relative to the device selected, ensure the the correct device is selected prior to initiating the read operation.

Tektronix Hex Format (Read)

Select this format to view or edit data pattern.

The Tektronix (Tekhex) format contains ASCII records that may be in either standard or extended format.

Hex Download Parameters

After selecting **read formatted file**, choose the required format, then select a file from the list that appears. After selection is made, the Download Parameter Editor is displayed. This editor can modify the parameters that are used when reading a hex file.

The following describes the parameters that the Hex Download Parameter editor can modify. The parameters that can be changed depend on the format type that was selected; not all parameters appear with every format selection. Refer to the procedure that pertains to the device manufacturer being used.

Number of Devices in Set

The number of devices needed to form the width of one word in the target system. Refer to the Programming in Sets section.

Current Device in Set

This only applies if the number of devices in set is greater than one. The current device in set range is from zero (0) to one less than number of devices in set.

Start File Address

The address where the ALLPRO starts reading the file. All data preceding this address is ignored. The number must be in hexadecimal format.

End File Address

The address where the ALLPRO stops reading the file. All data following this address is ignored. The default address is dependent upon the device selected. The number must be in hexadecimal format.

Download Parameter Editor	
# of devices in set	1
current device	0
start file address	0
end file address	7ff
start device address	0
nibble to download	2
Library = EPRGEN, Device = 2764A	
Message Center	
Use cursor keys to move highlight.	
Press F1 for help.	
Press F10 when done.	
Operator Input	
eprint	
s	
c	

Download Parameter Menu

For example, if a system uses a 16-bit bus, and two 2764 devices are being used to achieve an 8K x 16-bit memory system, the number of devices in set is two (2).

For example, if number of devices in set is two (2), then two downloads, or file readings, are needed to program the two devices. The first device, for even address bytes, has a current device in set value of zero (0). The second device, for odd address bytes, has a value of one (1).

For example, if 1000 hex is specified for this parameter, addresses from 0 to fff are ignored.

For example, if 3000 hex is specified for this parameter, addresses from 3001 to the end of the file are ignored.

Start Device Address

The beginning address in the device being loaded. The start device address, along with the start file address and end file address, define a window of data that is read from the data file and loaded into the device, starting at the specified start device address.

Nibble to Download

Specifies which nibble of the bytes is read. This applies to 4-bit PROM and EPROM devices, where the data may reside in either the upper or lower nibble. Zero (0) specifies read low nibble; one (1) indicates read high nibble; two (2) means read both nibbles. For 8-bit PROMs and EPROMs, this parameter must always be two (2).

Start Set Number

This is used for set programming only. If the number of sets defined is not the same as the number of sets in the computer RAM, start set number is used to choose which sets to download.

For example, if 100 hex is specified as the start device address, the addresses from 0 to ff in the computer RAM are not accessed.

Write Formatted File

Save the contents of computer buffer to storage.

Formatted files can be created from data contained in the computer's internal buffer. After a data pattern has been viewed or modified using the data map editor or the **read formatted file** function, it can be written in a specified format to storage on either the hard drive or disk. A file-name is requested when the file is saved.

When **write formatted file** is selected, a sub-menu allows selection of the file format to be written. The supported formats include Intel hex, ASCII hex, binary, straight hex, Motorola hex, JEDEC standard data transfer and Tektronix hex. Refer to the appendix for more details on formatted files.

Intel Hex Format (Write)

Select to save data pattern in this format.

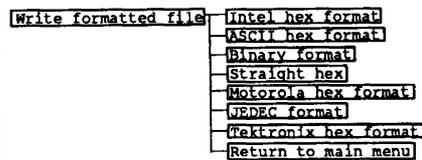
An Intel hex format file can be created with data read from a device. A device base address is requested. This address represents the first valid address in the device and must be entered in hexadecimal format. If necessary, extended address records are created to provide a maximum address space of 20 bits. Intel hex format is generated for programmable memory devices only.

ASCII Hex Format (Write)

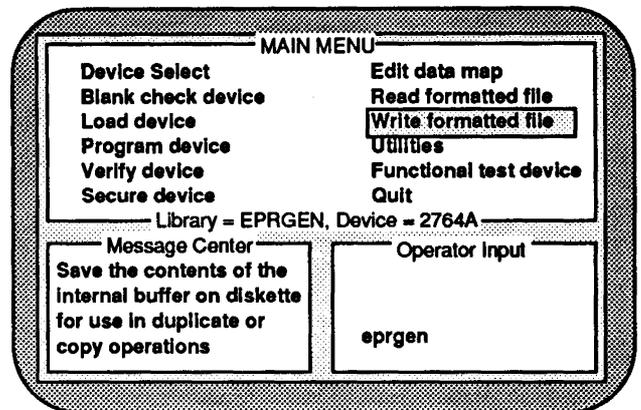
Select to save data pattern in this format.

This selection generates an ASCII coded hex format for either 4-bit or 8-bit PROMs. Each record contains a four-digit hex address (16-bit) followed by 16 data elements. A 16-bit checksum is at the end of the file.

When this format is selected, the device base address must be specified. This address represents the lowest address in the device. The file created



Write Formatted File Menu Tree



Main Menu: Writing a Formatted File

Formats supported for Memory devices are:

- INTEL HEX
- ASCII HEX
- BINARY
- STRAIGHT HEX
- MOTOROLA HEX
- TEKTRONIX HEX

Formats supported for PLD devices are:

- JEDEC

contains an entry for each location in this device. ASCII hex format can be created for programmable memory devices only.

Binary Format (Write)

Select to save data pattern in this format.

This selection generates a binary file. The internal memory contents are written to the specified file. The device base address cannot be specified, since the binary format does not have any addressing capability.

Straight Hex Format (Write)

Select to save data pattern in this format.

Straight hex format is generated for PROM type devices. The data may be either 4-bit or 8-bit data, with spaces following each byte. Unlike ASCII hex format, straight hex format does not have the address preceding the data bits.

Motorola Hex Format (Write)

Select to save data pattern in this format.

This selection creates Motorola hex format files for PROM type devices. The Motorola hex format contains hexadecimal data coded in an ASCII file. The format contains the load address, data, and a checksum for validation. This format transfers and manipulates data as text.

First, create a filename that ends with a **.HEX** extension. Then, enter the device base address, to establish the beginning address in the created hex file. The address must be in hexadecimal. Next, enter the Motorola file format. This selection recognizes S1-S9, S2-S8 and S3-S7 Motorola

.....
•
• **HINT:** •
• File formats can be converted. Read in a file •
• under the format in which it was written, then •
• save it under the desired format. •
•
•.....

formats. S1 data records contain 16 address bits, S2 contain 24 address bits and S3 contain 32 address bits.

The created file may contain addresses up to 32-bits long, which require the S3 format. The S2 or S1 format may be selected, however, if the file addresses require no more than the maximum number of bits available in those formats. The end of record entry contains a starting address of zero.

JEDEC Format (Write)

Select to save data pattern in this format.

A JEDEC fuse map may be created for PAL and PLD devices. This file conforms to JEDEC Standard No. 3B. This format can also be used for transferring files creating by CUPL™, Logical Devices' Universal Compiler for Programmable Logic. The following fields are created when this operation is selected.

STX The fuse map begins with an ASCII STX character (02 hex).

Design Specification This item is user specified. While no format rules apply, certain information, such as user's name, company, design date, part designation, revision and device part number, should be entered. This field is terminated by an asterisk (*).

QP This field specifies the number of pins in the device.

QF This field specifies the number of JEDEC fuses in the device.

L The fuse list fields contain the state of all fuse links in the device. The starting fuse number follows the L specifying the field type. The fuse list that follows contains a zero (0) for each intact link and a one (1) for each blown link. An L field is generated for each product term in the device.

C The checksum field contains the 16-bit sum of the link states of the 8-bit words.

ETX The fuse map ends with an ASCII ETX character (03 hex).

Sum Check A 16-bit sum of the ASCII values of the characters from STX to ETX inclusive. The sum check follows the ETX.

Tektronix Hex Format (Write)

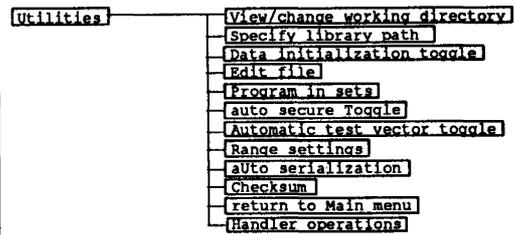
Select to save data pattern in this format.

The Tektronix (Tekhex) format contains ASCII records that may be in either standard or extended format.

Utilities

Contains many functions, including data path and flag setting.

Several functions are grouped in the **utilities** menu. Select this item to view the functions that are available, or refer to the Main and Sub-Menus section in this manual.



Utilities Menu Tree

View/Change Working Directory

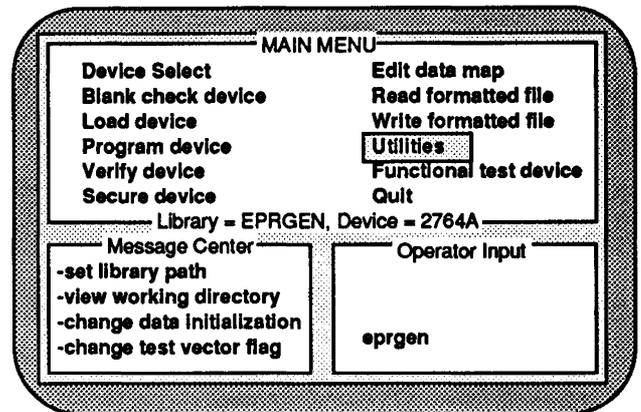
Display the files in the current directory.

The working directory is the drive and directory where files to be read or written are located. Initially, this directory defaults to the one from which the software was installed. It may be changed to default to one specified in the **ALLPRO.CFG** configuration file. **View/change working directory** displays all the files in the directory, including parent and sub-directories. Parent and sub-directories are highlighted, while normal files are not. The highlighted double dot (..) entry is the parent directory. The current path setting is displayed on the bottom main window border. To change the working directory, highlight and select the desired directory.

To set the working directory to a different disk, enter the drive letter and a colon.

Setting of the working directory may also be done by keying in the entire path name of the directory, as would normally be done in DOS using the CHDIR command.

When path setting is complete, press [Esc] to [Return] to the main menu.



Main Menu: Utilities Selection

NOTE:

Keep all of the data files in the working directory.

NOTE:

To change directories, type:
 C:\
 or C:\dirname\dirname

For example:
 C: (Return)

For example:
 \ALLPRO\HEX (Return)

Specify Library Path

Input the directory path for the library files.

The library path is the drive and directory where library database files (.DB files) and overlay files (.OVR files) are located. This path is initially set to the one from which the software was installed. It may be changed to default to one specified in the ALLPRO.CFG configuration file. If the library files need to be located in another directory, use **specify library path**. Enter the path name as if the DOS command CHDIR were being used. The directory path may be preceded by a drive designation. If the path entered does not exist, the old library path is not changed. The library path is displayed in the message window when the main menu is displayed.

Data Initialization Toggle

If set on, computer buffer is automatically initialized to intact fuse state upon device selection.

During device selection, the internal memory buffer used to store data is initialized to a default state. If desired, however, this initialization may be overridden by setting the data initialization flag to **off** under the **utilities** menu.

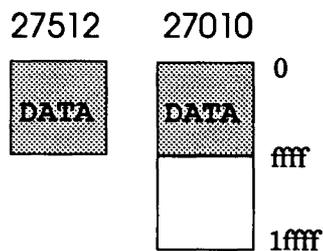
The data initialization flag is checked whenever a device is selected, to determine if the memory used as data space should be initialized to a default pattern. Normally, this switch remains in the **on** position. However, when copying different manufacturers' devices that are functionally compatible but require different programming characteristics, this switch is turned off. When in the **off** position, the data pattern in memory is not altered when a new device is selected. The toggle flag setting allows the source device to be selected, with initialize toggle on, and read. Then select the destination device, with initialize toggle off, and program the pattern.

This switch should be left in the **on** position unless it is specifically required to be off for a copy operation. If data initialization is off when selecting incompatible devices, unpredictable results may occur.

This toggle flag can be set on or off from the

For example, to specify a library path, type:

C:\OLDFILE



- With Data Initialization set to ON, select the 27512 (512K) device.
- Load the data from the 27512 device.
- Set the Data Initialization to OFF.
- Select the 27010 (1 Meg) device.
- Program the device.
- The data from the 27512 is now in the 27010.

Typical Application of Data Initialization

configuration file. For more information, see the Configuration File section.

Edit File

Change a text file using a text editor.

This function is used to edit a text file using a text editor, while remaining in the ALLPRO environment. The editor must be specified in the configuration file. Using **edit file** allows the entire contents of a file to be edited.

Selecting **edit file** displays the files that are in the current directory. Highlight and select the desired filename. If the desired file is not in the directory listed, type in the path and the filename and press [Return].

The ALLPRO software does not contain an editor; one must be supplied.

The editor must be set in the DOS path so that DOS can locate the editor file. The default editor specified in the configuration file is EDLIN, the DOS editor.

Refer to the configuration file installation section for more details.

Program in Sets

Split a file between several devices.

This selection allows programming two, three or four eight-bit EPROMs in parallel, for use in systems that have, respectively, 16-bit, 24-bit, or 32-bit data buses. Programming in sets reads the file in sets; multiple devices are programmed in parallel to provide a wider data path.

The procedure to program in sets using the Motorola and Intel hex format files is different from the procedure for ASCII, binary, and Tektronix hex formats.

Programming in sets is explained in detail in the appendix section of this manual.

To integrate an editor into the ALLPRO software, insert the following statement into the configuration file:

```
SET EDITOR <editor name>
```

where <editor name> is the name of the desired editor, such as PCWRITE.

Auto Secure Toggle

Automatically secures device after programming.

This function sets or resets the automatic secure flag. When the flag is set, the secure function is executed after a device is successfully programmed.

Securing a device prevents the data pattern that is programmed into the device from being read or copied. This feature is not supported by all devices. Check the device specifications to establish availability.

The automatic secure flag can also be set in the configuration file by using the command:

For more information, refer to the secure device section and the configuration file installation section.

Automatic Test Vector Toggle

If set to on, test vectors are executed after programming.

Initially, this flag is set to **off**. When set to **on**, test vectors in the JEDEC file are executed automatically after programming. Test vectors must be in the JEDEC file that was read for device programming. The ALLPRO does not execute test vectors automatically with the flag on when:

- > The selected device type is not a programmable logic device (PLD). In this case, test vectors do not apply.
- > There are no test vectors in the JEDEC file that was read.
- > The fuse map was created using the editor, instead of reading a JEDEC file.

A more detailed explanation of functional testing may be found in the Functional Test Device section.

CAUTION

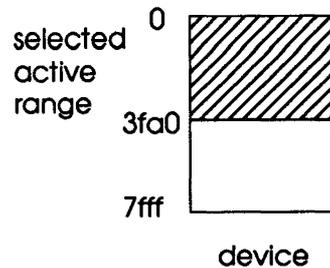
Once a device is secured, it cannot be unsecured to read the data in the device. With erasable devices, the data is erased when the security bit is erased.

SET AUTOSECURE ON

Range Settings

Sets the device address range accessed during an operation.

When using programmable memory devices, this selection sets the effective range for load, program, verify, blank check and editing operations. When **range settings** is selected, the page numbers and the starting and ending addresses of the range are entered, either in hex, decimal or octal. For example, if the **begin range** is set to 100 hex, and the **end range** is set to 4ff hex, only addresses from 100 to 4ff are affected by an operation.



Auto Serialization (Stamp-It™)

Devices are numbered with sequential serial numbers.

This menu item invokes the STAMP-IT Auto Serialization utility. This utility programs sequential serial numbers into programmable memory devices and Micros. When this item is selected, a sub-menu appears.

If the ALLPRO software is quit with the auto serial ON/OFF switch set to one (1), all parameters are saved to the **SERVALUE.DAT** file. To load the saved values, **set autoserial** must be set to ON in the **ALLPRO.CFG** file. The working directory is the drive and directory where files to be read or written are located. Initially, this directory defaults to the one from which the software was installed. It may be changed to default to one specified in the **ALLPRO.CFG** configuration file. For details about installing and using Stamp-It auto serialization, see the appendix section in this manual.

Checksum

Displays the checksum of a data pattern.

This selection calculates the checksum of data that is in the RAM buffer. For PLDs, the JEDEC checksum is displayed. For memory devices, the 16-bit checksum is the sum of all valid memory contents.

The checksum is a number that is the sum of data

bits in a file. It is used to compare data in devices or files for sameness. It is extremely improbable that a file could have errors yet produce a correct checksum; however, the only way data integrity can be guaranteed is with a bit-by-bit comparison.

Handler Operations

Initialize and activate the IC handler

This selection allows the software to be set-up for operation with an automated IC handler. This selection also initiates the operation of the handler. An optional IC labeler may also be use with the handler. Refer to the IC Handler appendix in this manual.

Functional Test Device

Functional test vectors are executed on a device.

When this item is selected, the JEDEC filename that contains the structured test vectors must be selected from the list displayed or entered from the keyboard. For more information on file selection, refer to the Onscreen File Select section. Functional testing may be done automatically by using the Auto Test Vector Toggle menu item.

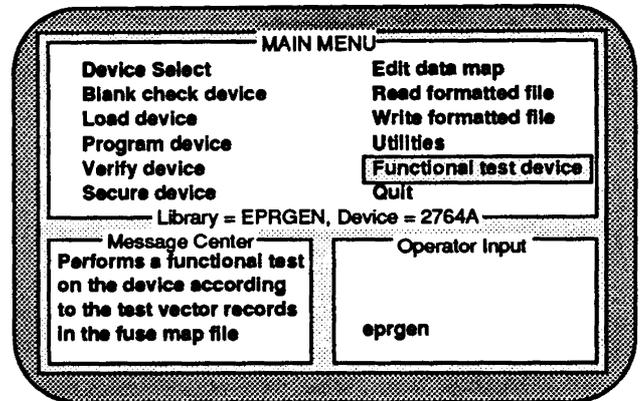
The test vectors are generated by CUPL TESTPLA™ or another logic compiler, and they must be present in the JEDEC file for functional test device to work. CUPL TESTPLA is a package that is used to automatically generate test vectors and is available from Logical Devices, Inc.

After the filename is selected, execution is started and the prompt displays:

If **y** is selected, the ALLPRO single steps through the vector list. After executing each test vector, the program pauses and displays **passed** or **failed**. If **n** is selected, the ALLPRO runs the entire list straight through, until all vectors are executed or until an error occurs.

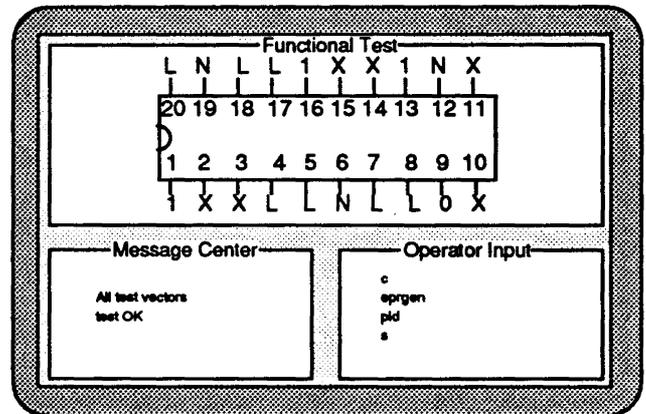
During execution, a pictorial representation of the device appears. A character indicating the test condition is displayed adjacent to each pin.

If an error is encountered, a message indicates the vector number or the line number of the file where the error occurred. The state of a pin in error is displayed next to the test condition character.



Main Menu: Functional Test Device

Single Step? y/n



The above diagram appears when functional testing is being done. The inputs and outputs are indicated with levels.

CAUTION

The ALLPRO does not support functional testing on devices that require negative voltages.

Certain high-speed devices that have speeds faster than 10ns may not pass the functional test.

Functional testing cannot be guaranteed for registered PLDs in PLCC packages.

Functional test conditions are indicated in the table.

The PLCC header on the ALLPRO88 is designed to support multiple sockets and a number of different devices in each socket. Because of this, the header does not support decoupling on every pin. Although this does not affect programming reliability, it may have an effect on the functional testing of PLDs in a state-machine design. A state-machine design is one in which the next state of flip-flops is dependent on the current state of other flip-flops along with the edge of a clock pulse. A PLD in which a state-machine design is implemented requires strong decoupling for proper operation.

Functional Test Conditions

1	input high	C	input low/high/low
0	input low	K	input high/low/high
H	output high	P	preload output register
L	output low	N	power pin
X	don't care		

Quit

Leave the ALLPRO program and return to DOS.

This selection exits from the ALLPRO control program and returns to the DOS command level. When asked to verify exit, press **Y** or **y** to exit to DOS, else press any other key to cancel the quit selection. To restart the ALLPRO software from DOS, type:

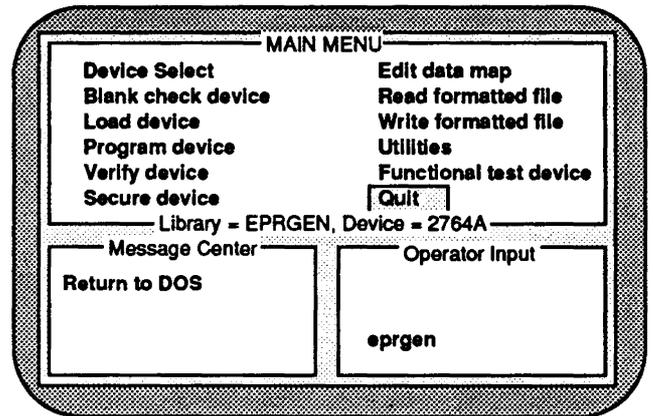
To invoke the ALLPRO program at power up, it may be necessary to go into the ALLPRO sub-directory, if there is no batch file. Type:

to change to the ALLPRO sub-directory and then type:

ALLPRO

cd ALLPRO

ALLPRO



Main Menu: Quitting from ALLPRO

ERROR CODES

Error codes and messages are displayed in the message window. Error messages are designed to be self-explanatory. Data is displayed as parallel data read from the device. For some devices, such as PALs, only half of the outputs are accessible at one time. For clarification, check the address information displayed with the error message; for PLDs, this includes the output number in error.

Error Message Listing

The message window should be monitored during operation. If an error message appears, refer to the following listing for a complete explanation. The first column is the error code number for LDI internal use, and the second column contains the error code message that appears on the screen. The third column explains the error message in detail.

1000	Unknown Machine ID. Enter PC for PC compatible or AT for AT compatible. (You can also enter Machine ID from the configuration file.	The configuration file does not contain a code that can be used to identify the computer. Machine ID must be either PC or AT. Enter one of the following lines into the configuration file: SET MID PC for a PC 8086/8088 machine or SET MID AT for an AT 80286/80386 machine.
1001	Cannot invoke command processor.	The Alt-D command to return to DOS cannot find the command processor. A copy of command.com must be in the ALLPRO directory.
1002	Programming in sets is available only for PROM type devices.	An attempt has been made to set program a device in which set programming is not supported.
1003	Device did not verify.	The data contained in the computer buffer does not match the data in the device. Either the device did not program correctly or the file in the computer buffer is not the exact file from which the device was programmed.
1004	Secure is not available for this device.	The device selected does not support the security option.
1005	Security fuse did not program.	An attempt to secure a device failed. Security is checked by reading the device pattern before blowing the security fuse and then verifying the pattern after the fuse is blown. If the device is secured, a verify error occurs. If the device failed to secure, the device pattern verifies.
1006	Overlay file not found: <i>filename</i>	The expected overlay file to execute is neither in the current directory nor in the library path.

1007	Invalid overlay file: <i>filename.OVR</i>	The overlay file was found in either the current directory or the library path, but it is not compatible with the base executable unit.
1009	ALLPRO unit is not properly connected to the PC.	Communication between the computer and the ALLPRO cannot be established. A memory conflict in the computer may be the cause. The address of the ALLPRO is defaulted to location D000, but can be relocated to any other open 4K byte area. Check that there is no conflict between these devices. This error may also occur if the ribbon cable is not connected or the unit power is turned off.
1010	A device must be selected before this function is activated.	An operation is selected that is specific to a device, but a device is not selected. Select the desired device library and device, then re-select the operation.
1011	Set number out of range.	For set programming, the number of sets must be within a specified range.
1012	No files with dir mask <i>filename</i> in this directory. Dir mask set to *.*	While browsing through the directories in the Onscreen File Select option, no files were found with the directory mask specified. The ALLPRO automatically changes the directory mask to *.* to display all the files that exist in that directory.
1013	Directory too large. Not enough memory.	While trying to read the filenames under a directory, the memory space runs out and not all the files under that directory can be displayed.

1014	Directory does not exist: <i>filename</i>	The specified directory, entered from View Working Directory command, does not exist and cannot display the files.
1015	File not found.	The filename entered was not found. The filename may have been mistyped or is in a different directory.
1016	Invalid test condition. Vector <i>XXX</i> Pin <i>XX</i>	An invalid test condition was found in the JEDEC file. This may be the result of the wrong device being selected.
1017	Vector <i>XXX</i> too large.	A test vector in the JEDEC file contains more entries than device pins.
1018	Vector <i>XXX</i> too small.	A test vector in the JEDEC file contains fewer entries than device pins.
1019	Preload is not supported for the selected device.	The registers in the selected device cannot be set to a known state during functional testing.
1020	File read error.	An error was encountered while reading the test vectors from disk.
1021	Unsupported data field. Line: <i>X</i> , Field Char: ' <i>Y</i> '	While executing functional tests, an illegal character was read from the JEDEC file. The error occurred at line number <i>x</i> , and the character is specified by <i>y</i> .

1022	Failed functional test.	Device may be defective or device speed is too fast. Do a verify to check that device was properly programmed. See functional test section of manual.
1023	No vectors found in file.	No test vectors were found in the selected JEDEC file.
1024	Value out of range. Must be within: x - y	The number entered is not in the expected range from x to y.
1025	Invalid choice.	A character was entered that is not allowed. Refer to display on screen for valid character choices.
1026	Not enough memory for multiple buffer mode. Number of buffers is reset to 1.	For set programming, there is insufficient computer RAM to allocate the buffers required by the selected devices.
1027	Invalid macro control character.	An invalid character exists in the ALLPRO.CFG file when using BATCHPRO. See manual for allowable control codes.
1028	Device is not blank.	After blank checking, some fuses in the device are not intact.

1032	Operation not supported.	The device does not support the selected operation.
1033	Fast sequence error.	x1033
1034	Device in socket contains incorrect manufacturer's code.	The signature read from the device is not the same as specified by the manufacturer. Some manufacturers require that the electronic signature be verified before performing an operation on the device.
1035	Device in socket contains unknown algorithm revision.	The algorithm revision read from the device cannot be recognized by the ALLPRO.
1036	Device in socket contains illegal device architecture.	The architecture of the device cannot be recognized by the ALLPRO.
1037	Device in socket is a master. It was not re-programmed.	Some GAL devices have a master bit that can be set. If the bit is set, the device cannot be programmed.
1038	Bulk erase failure. Not all arrays are erased.	After blank checking, some fuses in the device are not intact.

1039	Error re-programming manufacturer's ID code.	An error has occurred during the programming of the manufacturer ID array.
1040	Security fuse not intact. Operation aborted.	The selected operation cannot be completed because the device is secured.
1041	Invalid device ID.	The ID code read from the device cannot be recognized by the ALLPRO.
1042	Processor did not start.	The microprocessor in the socket did not power-up properly. Microprocessor could be defective or making poor contact in socket.
1043	Unable to read signature. Processor EPROM may be locked.	The EPROM of the microprocessor contains a signature code that cannot be read. The security bit has been set, and the contents can no longer be read.
1044	Unrecognizable device or not properly inserted.	The device in the socket is not recognized by the ALLPRO library or is inserted wrong in the socket.
1045	Bulk erase failure. PES array was erased.	Attempt operation again. If device fails again, replace device.

1050	Device is already secured.	The selected operation cannot be completed because the device is secured.
1053	Verify failure. Error in program memory.	Device specific error.
1054	Verify failure. Error in test EPROM.	Device specific error.
1055	Verify failure. Configuration fuses error was XXXX, expected XXXX.	Device specific error.
1056	Device did not erase.	After erasing the device, a blank check has determined that some fuse states are not intact.

1057	Device did not program.	The device did not program correctly. The data contained in the computer buffer does not match the data in the device.
1058	Warning! This device has been protected from verification of user EPROM and the encryption key contents.	A security or lock bit has been set. Contents can no longer be accessed.
1059	The device is not installed correctly or it has been fully secured. If the device is installed properly, then it must be erased before being programmed.	The device is positioned incorrectly in the socket or a security or lock bit is set and the device cannot be accessed for the selected operation. The security bit must be erased before the device can be programmed. Erasing the security bit also erases the data.
1060	Device is defective.	The device is defective and the operation selected cannot be performed. Check that the device is making good electrical contact with the socket pins.
1061	EPROM not erased.	After erasing the EPROM, a blank check has determined that some fuse states are not intact.
1062	Microcontroller did not verify.	The data in the computer buffer does not match the data in the micro. Either the micro did not program correctly or the file in the computer buffer is not the exact file from which the device was programmed.

1064	Operation aborted.	Fatal error. Operation cannot be continued.
1065	Library not found: <i>filename</i>	The library entered from the keyboard was not found. The filename can be constructed by taking the six character library name and concatenating it with .DB.
1066	Device not found: <i>filename</i>	An invalid device code was typed in. Either re-type the code or verify the code from the list of devices.
1067	<i>filename</i> No library found in specified library path.	There were no library files found in the library path.
1068	Cannot open library file: <i>filename.DB</i>	An error occurred while opening a library file.
1128	EMM driver failure.	Extended memory system error.
1129	EMM hardware failure.	Extended memory system error.

1131	Cannot find EMM handle.	Extended memory system error.
1133	No EMM handles available.	Extended memory system error.
1135	Insufficient EMS memory.	Extended memory system error.
1140	Device is not properly inserted.	The device is either plugged into the socket backwards, incorrectly offset, making poor connection or defective.
1141	Invalid module ID. Module ID = XX	The pin router module does not have a correct identification number. Contact the factory if this error occurs.
1142	Checksum error. Checksum = XXXX	A checksum error occurred while reading a JEDEC file.
1143	The specified file does not exist.	A filename keyed in from the keyboard was not found. Either verify that the working directory contains the desired file or key in the path as well as the filename.

1144	Error writing JEDEC file. Disk may be full.	When writing a JEDEC file to disk, either not enough space is available on disk or the file being written is a read-only file.
1145	File creation error.	Disk may be full or the existing file may be a read-only file.
1146	Error in checksum field. Line: X, Char: Y = 'Z'	While reading a JEDEC file, an unexpected character was encountered in the checksum field C. Line number and character position are indicated by x and y respectively. The character on which the error occurred is indicated by z.
1147	Error fuse default field. Line: X, Char Y = 'Z'	While reading a JEDEC file, an unexpected character was encountered in the default fuse field F. Line number and character position are indicated by x and y respectively. The character on which the error occurred is indicated by z.
1148	Unsupported data field. Line: X, Field Char: 'Y'	While reading a JEDEC file, an unexpected character was encountered as a data field heading. Line number and character are indicated by x and y respectively.
1149	Error in fuse list field. Line: X, Char Y = 'Z'	While reading a JEDEC file, an unexpected character was encountered in the fuse data field L. Line number and character position are indicated by x and y respectively. The character on which the error occurred is indicated by z.

1151	Invalid fuse in file. Line: X	The fuse is not defined for the selected device.
1153	Defined number of sets and number of sets in PC RAM must be equal for the shuffle operation.	Shuffle operation was attempted without the correct buffer number defined. Go to set programming under utilities and correct.
1154	Error writing HEX file. Disk may be full.	When writing a hex file to disk, either not enough space is available on disk or the file being written is a read-only file.
1156	Cannot create output file. Disk may be full.	When writing a file to disk, either not enough space is available on disk or the file being written is a read-only file.
1157	File address detected exceeding reserved RAM size.	During a hex file download, the address in the file exceeds the size of the device selected.
1158	Error reading binary file.	The file cannot be found or the file is corrupted or the incorrect format is selected.
1159	No files found.	The filename entered cannot be found.

1160	Device in socket contains incorrect device code.	The device code cannot be recognized by the ALLPRO. Verify that the correct device code was entered.
1162	Error writing binary file. Disk may be full.	When writing a binary file to disk, either not enough space is available on disk or the file being written is a read-only file.
1163	Device in socket contains unknown device's type.	The device in the socket cannot be recognized by the ALLPRO.
1164	Device in socket contains unknown device's wait state.	The wait state value is invalid. The wait state must be within the specified range.
1165	Load the data from a blank device then edit the desired data bytes, then repeat the programming operation.	x1165
1166	You must enter Y or N.	Only the character y or n can be accepted as input for this operation.
1167	Device is not blank or device has been secured or device improperly installed.	The data in the computer buffer does not match the data in the device or the device is secured and cannot be read or programmed or the device is installed improperly or is making poor contact in the socket.

1168	Incorrect pin size. Device pins = X File pin size = X	During the functional test operation, the QP field, which is the pin size field, does not equal the number of pins that are available on the device selected.
------	--	---