

IND-486H SBC User's Manual

For MCSI PART NO. 73300101 IND-486H
All-In-One Single Board Computers
For Industrial/Embedded Systems Applications

MICRO COMPUTER SPECIALISTS, INC.
"The Embedded PC Specialists"
1070 Joshua Way
Vista, CA 92083
U.S.A.

Voice (760) 598-2177 - Fax (760) 598-2450
email: mcsi@mcsi1.com WebSite: <http://www.mcsi1.com>

First Edition (September 8, 2000)
Revision 2.00

Changes are made periodically to the information contained herein; these changes will be incorporated into new editions of this document.

Requests for copies of this publication or the product(s) which it describes should be made to MCSI.

While every effort has been made to insure that this document and its accompanying product(s) are free from defects, MCSI, its distributors, representatives, and employees shall not be responsible for any loss of profit or any other commercial damage including, but not limited to, special, incidental, consequential, or other damages occasioned by the use of this product(s).

In the event of defect the buyer's sole recourse is to receive a refund or replacement unit at MCSI's discretion if notified within the time period covered by the product warranty.

©2000 MCSI Micro Computer Specialists, Inc. All Rights Reserved.
IBM is a registered trademark of International Business Machines Corporation.
PC/XT & PC/AT are registered trademarks of International Business Machines Corporation
MS-DOS is a registered trademark of Microsoft Corporation.
All other trademarks are the properties of their respective holders.

PREFACE

This manual provides information about the MCSI IND-486H All-In-One Single Board Computer. This information is intended for users who must implement IBM PC/AT compatible computer solutions to a wide variety of applications which cannot be satisfied using conventional desktop computers. This manual assumes that the reader has a good understanding of MS-DOS and the standard IBM PC/AT compatible architecture. For more information on the IBM PC compatible hardware and software architecture, refer to any of the many books available on the subject. A few suggestions are listed below:

- *Advanced MS-DOS Programming*, Microsoft Press
- *Programmers Guide to the IBM PC*, Microsoft Press
- *Programming the 80386*, Sybex
- *Undocumented DOS*, Addison Wesley

INVENTORY CHECKLIST

The complete IND-486H All-In-One Single Board Computer package consists of the following:

IND-486H All-In-One Single Board Computer
This Manual
PROMDISK Software Utilities Diskette

If any of the above is missing or appears to be damaged, inform MCSI immediately.

Please look for a file called README on the distribution disk for this product. It contains "last minute" changes and updates to this hard copy manual. This file will also document new utility programs.

Table of Contents

Section 1 - Introduction	1
1.1 Features	1
Section 2 - System Description	2
2.1 Processor	2
2.2 Co-Processor	2
2.3 System Memory (DRAM)	2
2.4 DMA Controller	2
2.5 Interrupt Controller	3
2.6 Timers	4
2.7 Clock/Calendar and CMOS RAM	4
2.8 Battery	4
2.9 Keyboard	5
2.10 Speaker Port	5
2.11 Reset Switch	5
2.12 Printer Port	5
2.13 Standard Serial Ports	5
2.14 Floppy Disk Port	6
2.15 EIDE Hard Disk Port	6
2.16 VGA Port	6
2.17 WatchDog Timer	7
Section 3 - Setup	8
3.1 Using the Internal Setup Utility	8
3.2 Clearing the CMOS Memory	8
Section 4 - PROMDISK Disk Emulator	9
4.1 PROMDISK Drive Configuration	9
4.2 PROMDISK Memory Setup	9
4.3 PROMDISK Power Selection	10
4.4 Dynamic Drive Assignment	10
4.5 PROMDISK FlashCom Feature	10
Section 5 - Using the PROMDISK Software	11
5.1 Using SRAMs with PROMDISK	11
5.1.1 Formatting SRAMs in the Floppy Disk Mode	11
5.1.2 Formatting SRAMs in the Fixed Disk Mode	12
5.1.3 Software Write Protect for SRAM Drives	13
5.2 Using EPROMs with PROMDISK	13
5.2.1 Creating a PROMDISK List File or Work Disk	13
5.2.2 Sizing the PROMDISK	15
5.2.3 Programming FLASH EPROMs	16
5.2.4 Programming Standard EPROMs	17
5.2.5 Programming the EPROM Drive A	17
5.3 Emulating Floppy Drives C through Z	17
5.4 Dynamic Drive Assignment	18
5.5 PROMDISK Custom Features	19
Section 6 - FlashCom	20
6.1 How Does FlashCom Operate?	20
6.2 FlashCom Components	20

6.2.1 UPLOAD	21
6.2.2 Upload Programming and Verification Options.....	22
6.3 General Purpose RS-232 File Transfer Programs	22
6.3.1 Using PUTFILE	22
6.3.2 Using GETFILE	22
6.4 Other Uses For FlashCom.....	23
Section 7 - Installation	24
7.1 Installing the SIMMs	24
7.2 Installing the PROMDISK Memory	24
7.3 Completing the Installation.....	25
Appendix A - Specifications	26
Appendix B - Board Outline	28
Appendix C - Memory and I/O Maps	29
Appendix D - Connectors	30
Appendix E - Configuration Jumpers	33
Appendix F - BIOS Error Beep Codes	35
Appendix G - Suggested PROMDISK Memory Chips	36
Appendix H - FlashCom Serial Interface	37

SECTION 1 - INTRODUCTION

The IND-486H All-In-One Single Board Computer (SBC) is a high performance system board that provides the primary elements for building an IBM PC/AT compatible computer for a wide variety of embedded systems applications. The IND-486H SBC contains all the basic elements found in a standard IBM PC/AT compatible desktop computer system.

The IND-486H SBC is fully compatible with the IBM PC/AT (ISA Bus) which means virtually all the software written for the IBM PC/AT will run on the IND-486H SBC.

1.1 FEATURES

A complete list of features is listed below:

- IBM PC/AT Compatible Plug-in Computer
- ST Micro STPC Client Chipset with 486DX2-66 CPU Core and 387 Math Coprocessor
- AMI BIOS
- 66MHz Operation
- Integral PCI VGA Controller and Graphics Accelerator
- 64M-Byte FPM or EDO DRAM System Memory (2-72pin SIMMs)
- Optional On-board System Memory expandable to 8MB
- 2M-byte PROMDISK Disk Emulator for Diskless Applications
- Includes FLASHCOM software to program Flash EPROMs remotely over COM1 serial port.
- Dual Floppy Disk Port Supports Two 3½" or 5¼" Drives up to 2.88M-bytes
- PCI Extended IDE Hard Disk Port supports up to Two Drives
- PS2/AT Compatible Keyboard Port
- PS2 Compatible Mouse Port
- Two High Speed 16C550 Compatible RS-232 Serial Ports
- Multimode Bi-directional Parallel Printer Port
- Infrared Data Access Port for future applications
- Clock/Calendar with Battery Back-up
- Low Power CMOS Design
- Half Size AT Plug-in Multilayer Board for Low EMI and High Reliability
- WatchDog Timer and Power Monitor
- On-board Mini Speaker
- Optional *Datalight* ROM-DOS 6.22 Operating System

SECTION 2 - SYSTEM DESCRIPTION

The following sections describe the major system features of the IND-486H All-In-One Single Board Computer.

2.1 PROCESSOR

The IND-486H SBC uses an ST Micro STPC Client chipset. The STPC chipset has an internal 486DX2-66 microprocessor. It is fully object code compatible with the 8086/8088, 80286, 80386, and 80486 family of products. Some of the distinctive features include:

- 16-bit External Data Bus
- 32-bit Internal Architecture
- 64M-byte Directly Addressable Memory Space
- Internal 14 Word by 16-bit Register Set
- Operand Addressing Modes
- Bit, Byte, Word, and String Operations
- 8 & 16-bit Signed/Unsigned Arithmetic
- Integrated Memory Manager

2.2 CO-PROCESSOR

The STPC Client chipset includes a 387 Math co-processor core. The 387 math coprocessor provides up to 100 times the performance of the CPU alone. Over seventy numeric processing instructions are added to the instruction set, and eight 80-bit registers are added to the register set. The coprocessor is compatible with the IEEE Floating Point Standard 754-1985.

2.3 SYSTEM MEMORY (DRAM)

The IND-486H SBC features a shared memory which supports up to 64M-bytes of dynamic random access memory (DRAM) organized as two banks of 32M-byte. The board will support either standard FPM (fast page mode) or high performance EDO (extended data output) DRAM. The memory is configured using two single in-line memory module sockets, which will accept 72-pin single in-line memory modules (SIMMs) organized as 1MB, 2MB, 4MB, 8MB, 16MB, or 32MB with a maximum access time of 60ns. In addition, the PCB is designed to accommodate up to 8M-bytes of on-board DRAM by the installation of four 1Mx16 Fast Page Mode (FPM) DRAM chips, for applications where the use of SIMMs is prohibited. The video controller shares 512K to 4M-bytes of system memory as configured in the CMOS set up. **Both SIMM sockets, or all four on-board DRAMs must be populated when using the VGA port.**

2.4 DMA CONTROLLER

The IND-486H SBC memory refresh and DMA functions are included in the STPC Client chipset, which includes the equivalence of two 82C37 DMA controllers. The two DMA controllers are cascaded to provide four DMA channels for transfers to 8-bit peripherals (DMA1) and three channels for transfers to 16-bit peripherals (DMA2). DMA2 Channel 0 provides the cascade interconnection for the two DMA devices thereby maintaining IBM PC/AT compatibility. The DMA channel assignments are listed below:

DMA Channel 0: Not Used (8-bit)
 DMA Channel 1: Not Used (8-bit)
 DMA Channel 2: Floppy Disk (8-bit)
 DMA Channel 3: Multi-mode Parallel Port (8-bit)
 DMA Channel 5: Not Used (16-bit)
 DMA Channel 6: Not Used (16-bit)
 DMA Channel 7: Not Used (16-bit)

The DMA request (DRQx) and acknowledge (DACKx/) lines are available on the P1/P2 98-pin edge connector.

2.5 INTERRUPT CONTROLLER

The IND-486H SBC has the equivalence of two 82C59A interrupt controllers included in the STPC Client chipset. The controllers accept requests from peripherals, resolve priorities on pending interrupts and interrupts in service, interrupt the CPU, and provide the vector address of the interrupt service routine. The two interrupt controllers are cascaded in a fashion compatible with the IBM PC/AT. The interrupt priority and assignments are shown below in descending order of priority:

<i>Highest</i>	IOCHCK/	Parity Check (Non-maskable)
	IRQ0	System Timer (Not Available)
	IRQ1	Keyboard (Not Available)
	IRQ8	Real Time Clock (Not Available)
	IRQ9	S/W Redirect to INT 0AH (IRQ2)
	IRQ10	Not Used
	IRQ11	Not Used
	IRQ12	Not Used
	IRQ13	Co-processor (Not Available)
	IRQ14	Not Used
	IRQ15	Not Used
	IRQ3	Serial Port 2
	IRQ4	Serial Port 1
	IRQ5	Parallel Port 2 (Not Used)
	IRQ6	Floppy Disk Controller
<i>Lowest</i>	IRQ7	Parallel Port 1

The interrupt request lines IRQx and IOCHCK/ are available on the 98-pin edge connector except as noted above.

2.6 TIMERS

The IND-486H SBC has the equivalence of an 82C54 Programmable Timer included in the STPC Client chipset. The 82C54 is a three channel Programmable Counter/Timer chip. The three timers are driven by a 1.19MHz clock source derived from the on-board 14.3818MHz crystal oscillator. The three timers are used as follows:

TIMER Channel 0: System Timer
TIMER Channel 1: Timer for DRAM refresh
TIMER Channel 2: Tone Generation for Audio

2.7 CLOCK/CALENDAR AND CMOS RAM

The IND-486H SBC has the equivalence of an MC146818 real time clock/calendar with 128 bytes of CMOS RAM included in the STPC Client chipset. An on-board lithium battery is used to provide power to the system chip when the system power is off.

The 128 byte CMOS RAM consists of 14 bytes used by the clock/calendar, and 114 bytes used by the system BIOS.

Should your CMOS become corrupted, i.e. loss of battery power or accidentally clobbered, strange errors may occur while attempting to run your programs. Refer to Section 3.0 for instructions on how to clear the CMOS and reset the initial SETUP values.

2.8 BATTERY

The IND-486H SBC contains an on-board Lithium battery used for powering the STPC Client chipset during standby operation in the absence of the normal +5 volt supply.

During normal operation, the battery should last greater than five years before needing replacement. A replacement battery may be ordered directly from MCSI.

Care must be taken when handling, shipping, and storing the IND-486H board. The following precautions should be adhered to in order to prevent accidentally damaging the battery:

DO store the board in a non-conductive, static free container, or **INSIDE** the original Anti-Static Bag. This bag has been specially designed to prevent battery discharge.

DO NOT allow the board to come in contact with the outside of the bag; the outer surface of the bag is conductive.

DO NOT place the board on a metal bench or other conductive surfaces at any time.

DO NOT connect any equipment, circuits, etc., to the on-board battery as serious damage may result. Any additional current drain will dramatically reduce its life.

2.9 KEYBOARD

The IND-486H SBC contains an IBM PC/AT compatible keyboard controller for interfacing to a generic IBM PC/AT compatible keyboard. The keyboard controller assembles the serial data from the keyboard into bytes and interrupts the CPU via IRQ1 after each byte is ready to be read. The IRQ1 service routine reads port 60H to get the keyboard scan code and acknowledges by sending a positive pulse to port 61H to clear the interrupt for the next byte. Refer to Appendix D for the keyboard connector location and pin assignments.

2.10 SPEAKER PORT

The IND-486H SBC contains an on-board sub-miniature audio speaker to provide audio interface to the user. Because of the small size of the speaker, the sound output is much reduced over that of the larger speaker found in most desktop computers. A connector is provided to connect an external speaker if the sound output is not sufficient.

Refer to Appendix D for the speaker port connector location and pin assignments.

2.11 RESET SWITCH

The IND-486H SBC includes an on-board power detector and power on reset circuit to reset the computer after power is applied, and to hold the computer reset during low power, brown-out conditions. In addition, there are provisions for connecting an external, normally open, push button reset switch.

Refer to Appendix D for the reset switch connector location and pin assignments.

2.12 PRINTER PORT

The IND-486H SBC contains the equivalence of an IBM PC/AT Parallel Printer Port. The Parallel Printer Port can be configured as a standard IBM PC/AT compatible printer port, an 8-bit bi-directional port, or disabled completely. The interface consists of twelve buffered outputs which are latched and can be written and read under program control. An additional five inputs are provided for reading printer status. Refer to the IBM PC/AT Hardware Reference Manual for detailed programming and applications information.

The address, configuration, and enabling/disabling of the port are configured using the BIOS's INTERNAL SETUP program. Refer to Section 3.0 for more information. Refer to Appendix D for the connector location and pin assignments.

Note: The default interrupt for the printer port is IRQ7.

2.13 STANDARD SERIAL PORTS

The IND-486H SBC has the equivalence of two NC16C550 UARTs. The two UARTs can be configured as standard IBM PC/AT RS-232C compatible serial ports or individually disabled.

The port addresses, and enabling/disabling of the serial ports are configured using the BIOS's INTERNAL SETUP program. Refer to Section 3.0 for more information. Refer to Appendix D for the connector location and pin assignments.

Note: Interrupts for Serial Ports 1 & 2 default to IRQ4 & IRQ3, respectively.

2.14 FLOPPY DISK PORT

The IND-486H SBC contains an IBM PC/AT compatible dual floppy disk port with the equivalence of an NEC PD72056B Floppy Disk Controller, an on-chip digital data separator, and an IBM PC/AT compatible floppy disk adapter bus interface circuit. An on-board digital data separator provides optimum performance with the following disk drive types:

5.25"	360K Double-Sided
3.5"	720K High Capacity
5.25"	1.2M High Capacity
3.5"	1.44M High Density
3.5"	2.88M High Density

The port address, and enabling/disabling of the floppy port is configured using the BIOS's INTERNAL SETUP program. Refer to Section 3.0 for more information. Refer to Appendix D for the connector location and pin assignments.

Note: The default interrupt for the floppy port is IRQ6.

2.15 EIDE HARD DISK PORT

The IND-486H SBC contains a PCI Extended Integrated Drive Electronics (EIDE) Port which directly interfaces to two hard disk drives with embedded controllers. The EIDE Disk Port can be disabled using the CMOS Setup utility. Refer to Appendix D for the connector location and pin assignments.

The enabling/disabling of the EIDE port is configured using the BIOS's INTERNAL SETUP program. Refer to Section 3.0 for more information. Refer to Appendix D for the connector location and pin assignments.

2.16 VGA PORT

The IND-486H SBC includes a VGA display controller as part of the STPC Client chip set. The VGA display port is 100% backward compatible with the VGA standard specification. Resolutions of up to 1024x768 and color depths of 8, 16, 24 and 32 bits per pixel are supported. The integrated RAMDAC supports digital to analog conversion rates up to 135 MHz. This along with peak video bandwidth of 320MBytes/sec (using EDO DRAM) enables the VGA controller to support 1024 x 768 x24 and 800 x 600 x 32 resolutions at 75 Hz refresh rate.

The VGA controller shares up to 4MB of system DRAM for video RAM. **Both SIMM sockets or all on board DRAM chips must be populated when using the VGA port.**

If an external add-in VGA card is placed in the system, the IND-486H VGA controller is automatically disabled. Refer to Appendix D for the connector location and pin assignments.

2.17 WATCHDOG TIMER

The IND-486H SBC includes a WatchDog Timer circuit. The WatchDog Timer ensures that if an application program gets "lost or bombs", the system will reset or a non-maskable interrupt will be issued to the CPU. The WatchDog Timer is enabled by writing a byte xxH to memory address F000:FF55H. Once enabled, the WatchDog Timer must be triggered by writing a byte (xxH) to memory address F000:FF55H within the time out period, otherwise the WatchDog Timer will force a hardware reset or activate the IOCHCK/ line, generating a non-maskable interrupt (NMI). The WatchDog Timer can be disabled by writing a byte (xxH) to memory address F000:FB55H. A jumper is provided to select the time out period and to enable the WatchDog Timer circuit. Refer to Appendix E for the WatchDog Timer configuration jumpers

SECTION 3 - SETUP

The IND-486H uses an American Megatrends (AMI) BIOS. The BIOS includes an internal Setup Utility for configuring the system. The system configuration settings are stored in the on-board CMOS memory which is backed up by a Lithium battery.

3.1 USING THE INTERNAL SETUP UTILITY

The IND-486H SBC uses an AMI BIOS which contains an internal Setup Utility for configuring the system. The system configuration settings are stored in the on-board CMOS memory, which is backed up by a Lithium battery. Should your CMOS become corrupted, i.e. loss of battery power or accidentally clobbered, strange errors may occur while attempting to run your programs. A jumper at JP16 has been provided to force the BIOS to use its internal default SETUP values. Refer to section 3.2 for details.

The Setup Utility can be invoked by first causing a cold boot (reset) or a warm boot (**Ctrl Alt Del**) and pressing the **Del** key when instructed. This will cause the memory diagnostics to be aborted and the Setup Utility to display the MAIN SETUP MENU. Using the **↑↓** cursor keys, move the highlighted bar to the option you wish to modify and then press **Enter** to select it. When in the MAIN SETUP MENU, the **F2** and **F3** keys are used to select the colors used in the setup screens, and the **F10** key is used to save the changes before exiting the Setup Utility. The **Esc** key may be used to exit the Setup Utility without saving the changes. The **PgUp** and **PgDn** keys are used to scroll through the selections for a given setting. **PgUp** is also used to decrease the setting and **PgDn** to increase the setting.

After making the desired selections from the various setup menus press the **Esc** key to exit the current menu. You save your selections by pressing the **F10** key or by selecting the appropriate selection from the MAIN SETUP MENU.

Notes:

1. The user should be aware that improper selection of certain values in the ADVANCED SETUP UTILITY may cause unpredictable results. If this occurs select the AUTO CONFIGURATION WITH OPTIMAL SETTINGS from the MAIN SETUP MENU.

3.2 CLEARING THE CMOS MEMORY

Should your CMOS become corrupted, i.e. loss of battery power or accidentally clobbered, strange errors may occur while attempting to run your programs. A jumper at JP16 has been provided to force the BIOS to use its internal default SETUP values. This is accomplished by first removing power from the IND-486H and momentarily interrupting the battery power to the STPC Client chipset. To interrupt the battery power, remove the shunt from JP16 pins 1 & 2 and momentarily connect it to pins 2 & 3 (the "Clear CMOS" position). After waiting a few seconds, return the shunt back to its original position.

WARNING: DO NOT apply power to the IND-486H with the shunt in the clear CMOS position ---- serious damage will result.

Refer to Appendix B for the location of the CMOS shorting jumper.

SECTION 4 - PROMDISK DISK EMULATOR

The IND-486H SBC includes an on-board 2M-byte PROMDISK Disk Emulator for applications requiring diskless operation. The PROMDISK memory is made up of four 32-pin JEDEC standard byte-wide memory sockets which may be configured to accept EPROMs, FLASH EPROMs, NVRAMs, and SRAMs. The four sockets are arranged as two groups consisting of sockets U16&U12, and U8&U5. Each group of sockets can be configured separately allowing up to two drives to be implemented with different type chips, or a single drive with the same type chips. The grouping of the sockets was chosen to provide the most flexibility with the minimum number of configuration jumpers and switches.

4.1 PROMDISK DRIVE CONFIGURATION

During the power on diagnostics, the PROMDISK control program determines the drive configuration by looking for the PROMDISK signature at sockets U16 and U8. Drives configured from different type chips must start on the group boundaries, i.e. U16, and U8. Read/Write drives using SRAMs or NVRAMs must use sockets U8 and U5. The allowable PROMDISK drive configurations are shown below:

PROMDISK Disk Drive Configuration

U16	U12	U8	U5
A	A	A	A
B	B	B	B
C	C	C	C
D	D	D	D
A	A	B	B
B	B	A	A
A	A	C	C
C	C	A	A
A	A	D	D
D	D	A	A
B	B	C	C
C	C	B	B
B	B	D	D
D	D	B	B
C	C	D	D
D	D	C	C

Note: A drive can consist of 1 to 4 chips.

4.2 PROMDISK MEMORY SETUP

The PROMDISK memory can be configured to accept EPROMs, FLASH EPROMs, and SRAMs. For proper operation the devices should have a maximum access time of 150 nanoseconds or less.

The memory sockets can be configured to emulate up to two fixed or floppy disk drives simultaneously. The type of chips, i.e. EPROM or SRAM used for each drive is selected by the PROMDISK Configuration jumpers JP18, JP14, JP9 and JP15. Sockets U16&U12 are controlled by JP18

and JP14, and sockets U8&U5 by JP15 and JP9. Refer to Appendix E for the PROMDISK configuration jumpers.

4.3 PROMDISK POWER SELECTION

When using low-power CMOS SRAMs in sockets U8&U5, an on-board power control and battery circuit is selected by JP4 to provide nonvolatile operation for the PROMDISK Disk Emulator drive 2. Extreme care must be taken when configuring these jumpers. Refer to Appendix E for the PROMDISK power control jumpers.

4.4 DYNAMIC DRIVE ASSIGNMENT

The IND-486H PROMDISK supports dynamic drive assignments for its two drives. This feature is extremely useful during the development stages when the IND-486H board is installed in a development system with a different drive configuration than the final target system. The dynamic drive assignment feature uses the BIOS CMOS configuration to determine whether to reassign the drives. The PROMDISK drive is reassigned to the next logical drive letter or disabled whenever the PROMDISK drive matches the BIOS CMOS configuration. In other words, if you have a PROMDISK drive A and the CMOS floppy drive is selected, the PROMDISK drive A will be reassigned to drive B, and if a PROMDISK drive B were present, the PROMDISK drive A will be disabled. The table below shows how typical PROMDISK drive configurations are reassigned.

PROMDISK Dynamic Drive Assignment

Original PROMDISK		CMOS				Reassigned PROMDISK	
Drive 1	Drive 2	FDD #1	FDD #2	HDD #1	HDD #2	Drive 1	Drive 2
A	None	Installed	None	X	X	B	None
A	B	Installed	None	X	X	Disabled	B
A	B	None	Installed	X	X	A	B
A	B	Installed	Installed	X	X	Disabled	B
None	A	Installed	None	X	X	None	B
None	A	Installed	Installed	X	X	None	B
B	A	Installed	None	X	X	B	Disabled
B	A	None	Installed	X	X	B	A
B	A	Installed	Installed	X	X	B	Disabled
C	None	X	X	Installed	X	D	None
None	C	X	X	Installed	X	None	D
C	D	X	X	Installed	X	Disabled	D
D	C	X	X	Installed	X	D	Disabled

4.5 PROMDISK FLASHCOM FEATURE

The PROMDISK control firmware includes the MCSI FlashCom remote download and programming firmware for COM1/COM2. Refer to Section 6.0 for a detailed description of the features. The FlashCom firmware is enabled by installing a shunt jumper between JP15 pins 7-8 for COM1 or 5-6 for COM2. Refer to Appendix E for the PROMDISK configuration jumpers.

SECTION 5 - USING THE PROMDISK SOFTWARE

The IND-486H PROMDISK can be configured to emulate up to two drives simultaneously, each of which can be operated as a Read-Only or Read/Write floppy or fixed disk drive. The paragraphs that follow describe how to use PROMDISK in the floppy and fixed disk emulator modes. The operating modes of the two drives can be mixed for added flexibility. The fixed disk drive emulator assumes that fixed disk drives start with drive "C", and that drives "A & B" are always floppy drives. Remember that the PROMDISK hardware portion of the board must be properly configured prior to changing the types of memory used for each drive.

5.1 USING SRAMs WITH PROMDISK

This section describes how to use the PROMDISK Disk Emulator with SRAMs or NVRAMs. The paragraphs that follow assume that the PROMDISK drives have been configured, and the SRAMs are installed on the board in accordance with Sections 4 and 7 of this manual. The board must then be installed in the target system with a floppy or fixed disk, and your current version of DOS loaded before continuing on.

5.1.1 Formatting SRAMs in the Floppy Disk Mode

When PROMDISK is configured to emulate a Read/Write floppy disk drive using SRAMs or NVRAMs, it is necessary to format the PROMDISK using the PROMDISK Format Utility FORM733.COM. Once formatted, the PROMDISK drive can be treated just like a physical floppy disk drive and can be randomly accessed at memory bus speed, thereby increasing system through-put orders of magnitude over that of a typical floppy disk system. The standard PROMDISK firmware does not allow SRAMs to be configured as a bootable floppy drive, i.e. drive A. If this limitation presents a problem, a special version of the PROMDISK firmware can be requested from the factory. If you wish to configure the SRAMs as a bootable drive, you must configure the SRAMs as a PROMDISK fixed disk drive C (refer to Section 5.1.2). To format the PROMDISK SRAMs in the floppy disk mode you must first boot the target system with your operating system. Once the system is booted, you must format the PROMDISK floppy disk drive by typing:

```
FORM733 {pddrive} [/p]
```

Where:

pddrive is the PROMDISK drive to format

/p is the optional step by step confirmation command

Example 1: To format PROMDISK drive B as a *floppy disk drive* starting at chip number three and containing one SRAM, type:

```
FORM733 B
```

Example 2: To format PROMDISK drive B as a *floppy disk drive* starting at chip number three and containing two SRAMs, with step by step confirmation, type:

```
FORM733 B /p
```

NOTES:

1. The standard DOS format program, **FORMAT.EXE**, cannot be used to format a **PROMDISK** floppy disk drive.
2. Be sure to include the device driver **DRIVER.SYS** in your **CONFIG.SYS** file when emulating **PROMDISK** floppy drives **C** through **Z**. Refer to Section 5.3 for more details.

If the selected **PROMDISK** drive was never formatted or the data has been corrupted, the **PROMDISK** Format Utility will prompt you to reboot the system so that the **PROMDISK** control program can initialize and configure the drive. After the system is rebooted, you must run **FORM733** again to format the drive.

5.1.2 Formatting SRAMs in the Fixed Disk Mode

When **PROMDISK** is configured to emulate a Read/Write fixed disk drive using **SRAMs** or **NVRAMs**, it is necessary to first create a valid **DOS** partition that can be recognized by both the system **BIOS** and your operating system's Format Program. Once a valid **DOS** partition has been created, you simply use the Format Program supplied with your operating system to format the **PROMDISK** **SRAMs**.

To initialize the **PROMDISK** **SRAMs** with a valid **DOS** partition you first boot the target system using a floppy disk containing your operating system. Once the system is booted, you initialize the **PROMDISK** fixed disk drive using the **PROMDISK** Fixed Disk Initialization Program **FDISK733.COM**.

Example 1: To initialize **PROMDISK** drive **C** starting at chip number three and containing two chips, type:

FDISK733 C

Example 2: To initialize **PROMDISK** drive **D** starting at chip number three and containing one chip, type:

FDISK733 D

After the **SRAMs** have been initialized, the program will instruct you to reboot the system. After rebooting the system, the **PROMDISK** **SRAMs** must be formatted using the Format Program supplied with your operating system. You must remember to copy the system files if you plan to make the **PROMDISK** a bootable drive.

Example 1: To format **PROMDISK** drive **C** as a bootable drive under **MS-DOS**, type:

FORMAT C: /u/s

Example 2: To format **PROMDISK** drive **D** as a non-bootable drive under **MS-DOS**, type:

FORMAT D: /u

After the **PROMDISK** drive is formatted, you may reboot your system and use the **PROMDISK** drive just like a physical fixed disk drive.

5.1.3 Software Write Protect for SRAM Drives

The IND-486H PROMDISK includes a software write protect feature. If you wish to incorporate the write protect capability on your R/W drives, use the PDPRO.COM Utility Program. To write protect PROMDISK drive B, type:

Example 1: To write protect (*disable*) PROMDISK drive B, type:

```
PDPRO B:/p
```

Example 2: To unprotect (*enable*) PROMDISK drive B, type:

```
PDPRO B:/u
```

Note: The protect bit is reset to the unprotected state when the system is rebooted.

5.2 USING EPROMS WITH PROMDISK

This section describes how to use the IND-486H PROMDISK with standard (ultraviolet light erasable) EPROMs and FLASH EPROMs. Standard EPROMs must be programmed using an external EPROM programmer from binary image files created with the PROMDISK PDCR733.EXE Software Utility, whereas FLASH EPROMs can be programmed on-board. The paragraphs that follow assume that the PROMDISK drives have been configured and, if using FLASH EPROMs, they are already installed on the board. The board must then be installed in the target system with a floppy or fixed disk, and your current version of DOS loaded before continuing on.

5.2.1 Creating a PROMDISK List File or Work Disk

The PROMDISK PDCR733.EXE Programming Utility provides two methods of programming FLASH EPROMs or creating binary image files. The first method uses a *list file* to specify the source and structure of the PROMDISK. This method has no limit on the size of the PROMDISK it can create and assumes the files are DOS compatible. The second method involves creating a *work disk* containing the files you wish to emulate on the PROMDISK. This method limits the size of the PROMDISK to the size of your *work disk*, and also requires the *work disk* to be reformatted each time changes are made to it. The only advantage of the *work disk* method is that it uses BIOS calls to read the disk sector by sector and therefore is able to create a PROMDISK containing a non-DOS type operating system. The *work disk* method **cannot** be used to create a PROMDISK fixed disk drive. Therefore, you must first decide whether to use a PROMDISK *list file* or *work disk* to specify your PROMDISK files.

If you choose the PROMDISK *list file* method, you must first create an ASCII file containing a list of files you wish to include in your PROMDISK disk emulator using a text editor of your choice. This file will be referred to as the PROMDISK *list file* and must end with the extension *.LST*. This file is used by the PROMDISK Programming Utility PDCR733.EXE to specify the files, their location, and for creating PROMDISK subdirectories. The first line of the *list file* must contain the PROMDISK keyword [BOOT SOURCE], followed by the location of the boot sector for the operating system you plan to use. If you wish to create a bootable PROMDISK drive, the next line of the list must contain the PROMDISK keyword [SYSTEM], then followed by the list of system files. The remainder of the list specifies the PROMDISK directories, subdirectories, and their associated files. The format of the *list file* is shown below:

<i>Format for List File</i>	<i>Remarks</i>
[BOOT SOURCE]	PROMDISK Boot Source Keyword
drive:	Location of Boot Sector and System Files
[SYSTEM]	PROMDISK System Keyword
[filename.SYS]	Hidden System File(s)
[filename.COM]	System Command File
[VOLUME]	PROMDISK Volume Keyword
volumeName	Volume Name (11 characters max)
[\]	PROMDISK Root Directory
[drive][path]MYFILE1.EXT	Location of File(s) in Root Directory
[\subdirectoryName]	PROMDISK Subdirectory
[drive][path]MYFILE2.EXT	Location of File(s) in Subdirectory

Example 1: A typical *list file* for a bootable fixed disk is shown below:

PROMDISK List File

```
[BOOT SOURCE]
b:
[SYSTEM]
io.sys
msdos.sys
command.com
[\]
autoexec.bat
config.sys
[\myproject]
c:\myproject\myfile1.ext
d:\myfile2.ext
[\promdisk]
c:\pdiii\PDCCR733.exe
c:\pdiii\pdformat.exe
[\promdisk\lists]
d:\data\als.lst
d:\data\johns.lst
```

PROMDISK Directory Structure

```
C:\
|
| io.sys
| msdos.sys
| command.com
| autoexec.bat
| config.sys
|
|----- \MYPROJECT
|         myfile1.ext
|         myfile2.ext
|
|----- \PROMDISK
|         pdcreate.exe
|         pdformat.exe
|
|         \LISTS
|         als.lst
|         johns.lst
```

Example 2: A typical *list file* for a non-bootable floppy disk is shown below:

<i>PROMDISK List File</i>	<i>PROMDISK Directory Structure</i>
[BOOT SOURCE]	
c:	B:\
[\]	
autoexec.bat	autoexec.bat
c:\project\myfile1.ext	myfile1.ext
d:\myfile2.ext	myfile2.ext
[\apps]	
c:\doug\program1.exe	\APPS
c:\daenah\program2.com	program1.exe
	program2.com

If you do not have access to a text editor, you can use the standard DOS copy command to create the PROMDISK *list file*. To create a PROMDISK *list file* named MYLIST.LST using the list of files from example two above, at the DOS prompt you would type:

```
copy con mylist.lst
[boot source]
c:
[\]
autoexec.bat
c:\myproject\myfile1.ext
d:\myfile2.ext
[\apps]
c:\doug\program1.exe
c:\daenah\program2.com
^z                               (control Z)
```

Once you have created your PROMDISK *list file*, you can proceed to Section 5.2.2 to determine the size and number of memory chips that will be required to emulate your specified drive.

If you choose to use the *work disk* method to specify your PROMDISK drive, you must first format a floppy diskette using the version of DOS that your files require for operation. Be sure that the format utility you use completely erases the diskette. If you are using DOS 5.0 FORMAT.EXE, be sure to use the /U option. Also, if you are making a bootable PROMDISK drive A, be sure to transfer the system files using the FORMAT /S option. You then copy the files you wish to emulate onto the newly formatted work diskette. You can then proceed to Section 5.2.2 to determine the size and number of memory chips that will be required to emulate your specified drive.

Note: The work disk method cannot be used to create a PROMDISK fixed disk drive.

5.2.2 Sizing the PROMDISK

Use the *size* option of the PROMDISK utility PDCR733.EXE to determine the size and number of chips required for emulation.

Example 1: To determine the number of chips required to emulate the files specified by your PROMDISK *list file* named MYFILES.LST located on your drive B, type:

PDCR733 B:MYFILES SIZE

Note: DO NOT add an extension to the list file filename. PDCR733 assumes the extension .LST.

Example 2: To determine the number of chips required to emulate the files specified by your PROMDISK *work disk* located on your drive B, type:

PDCR733 B SIZE

Note: DO NOT add a colon after the drive letter when sizing a work disk.

The PROMDISK sizer will report the type, size, and number of chips required for drive emulation.

5.2.3 Programming FLASH EPROMs

Use the PROMDISK Programming Utility PDCR733.EXE to program the FLASH EPROMs. Using the size information from Section 5.2.2 above, program the FLASH EPROMs by typing:

```
PDCR733 source numChips kSize destfile pddrive [option]
```

Where:

source is the location and name of the *list file* (with no extension) or the location of the *work disk*.

numChips is the number of chips from Section 5.2.2.

kSize is size of the chip(s) in K-bytes, i.e. 128, 256, or 512.

destFile enter Flash for Intel/AMD 28F010/020 Flash EPROMs, AMD for AMD 29F040 Flash EPROMs, SST for SST Flash EPROMs, or output filename without extension to create the binary images.

pddrive PROMDISK drive to program, i.e. A, B, C, etc.

options:

/dN = Use DOS to read floppy (0 = no, <1>= yes)

/fN = Starting chip <1>, or 3

/hN = Fixed disk partition (0 = no, <1>= yes)

/pN = Pause before reading disk (0 = no, <1> = yes)

/sN = Pre-size before writing (0 = no, <1>= yes)

/vN = Verbose setting (0, <1>, or 2)

<> denotes the default value

Note: DO NOT add an extension to the destFile filename when making binary images; PDCR733 adds the extension .PD1, .PD2, etc.

The PDCR733 program will check to see if the FLASH EPROMs are erased, erase them if necessary, and program them accordingly. After the chips are programmed, PDCR733 will prompt you to reboot the computer to initialize the PROMDISK drive.

Example 1: To build a 1M-byte PROMDISK drive C using four INTEL FLASH 256Kx8 chips starting at chip one from your *list file* named MYFILES.LST located in drive A, type:

PDCR733 A:MYFILES 4 256 FLASH C /f1

Example 2: To build a 128K PROMDISK drive B using one INTEL FLASH 128Kx8 chip starting at chip three from your *work disk* located in drive A, type:

PDCR733 A 1 128 FLASH B /f3

5.2.4 Programming Standard EPROMs

Standard EPROMs must be programmed from an external EPROM programmer using the binary image file(s) created with the PROMDISK PDCR733.EXE Programming Utility before installing them on the PROMDISK board. The procedure for creating binary image file(s) is the same as the procedure for programming FLASH EPROMs (Section 5.2.3 above) except that an output filename is specified instead of the word "FLASH". After programming the EPROMs with an external EPROM programmer, they must be installed on the board in accordance with Sections 4 and 7 of this manual before installing the board in your target system.

Example 1: To create the binary image file(s) named MYIMAGE.PDx for a 512K PROMDISK drive B using 4 128Kx8 EPROMs from your PROMDISK *list file* named MYFILES.LST located in drive A, type:

PDCR733 A:MYFILES 4 128 MYIMAGE B

Example 2: To create the binary image file for a 256K PROMDISK drive B using one 256Kx8 EPROM from your PROMDISK *work disk* located in drive A, type:

PDCR733 A 1 256 MYIMAGE B

The build program will create the binary images for the files specified in your file list and write the images to the selected filename. The selected filename extension .PDx will be incremented for each chip in the set.

5.2.5 Programming the EPROM Drive A

When PROMDISK is configured as drive A, your system will automatically try to boot from PROMDISK. If your PROMDISK drive A is corrupted or the EPROMs have been erased, your system will hang-up since there is no valid boot sector. If this occurs, simply set the CMOS floppy drive A to Installed, which will reassign the PROMDISK drive to B. Refer to Section 4.4 for more information on the PROMDISK reassign feature.

5.3 EMULATING FLOPPY DRIVES C THROUGH Z

When emulating floppy disk drives C through Z in systems that do not allow floppy disks to be configured for these drives or in systems with hard disks, you must specify the DOS device driver DRIVER.SYS in your CONFIG.SYS file. The CONFIG.SYS and DRIVER.SYS files must be contained in the root directory of your boot drive.

Example: To configure your system with a physical disk for drive A and PROMDISK floppy drive C, you must first set the system CMOS for a physical floppy disk installed for drive A and *NONE* installed for drives B through D. Next, you make a CONFIG.SYS file containing the following:

```
DEVICE = DRIVER.SYS /d:2
```

5.4 DYNAMIC DRIVE ASSIGNMENT

The IND-486H PROMDISK supports dynamic drive assignments for its two drives. This feature is extremely useful during the development stages when the IND-486H board is installed in a development system with a different drive configuration than the final target system. The PDASSIGN.COM software utility allows the user to reassign or even hide the PROMDISK drives. The format for PDASSIGN is shown below:

```
PDASSIGN {oldDrive}={newDrive}
```

Where:

{oldDrive} is the PROMDISK drive you wish to reassign.

{newDrive} is the letter A through Z for the new PROMDISK drive. Entering a dash (-) will hide the drive until the system is rebooted.

Notes:

- 1.0 PROMDISK fixed disk drives cannot be reassigned.**
- 2.0 The PROMDISK drives are reset to their original assignments when the system is rebooted.**
- 3.0 If you experience problems accessing a real floppy disk drive after using PDASSIGN, make sure your source diskette is the same media type, i.e. if you need to access a 1.44M floppy diskette, you should build your PROMDISK with PDCR733 using the same size drive.**

Example 1: To reassign PROMDISK drive A to B, type:

```
PDASSIGN A=B
```

Example 2: To reassign PROMDISK drive B to A, type:

```
PDASSIGN B=A
```

Example 3: To hide PROMDISK drive A, type:

```
PDASSIGN A=-
```

The following is a typical application example:

The PROMDISK is configured as drives A and B. Drive A contains DOS, and drive B the application program. To protect (hide) the application program on drive B, you would execute PDASSIGN B=- from within the application program or from the DOS prompt. This would cause the PROMDISK drive B to be inaccessible, and the physical drive B, if installed, to be accessible.

5.5 PROMDISK CUSTOM FEATURES

The PROMDISK control firmware can be customized to include custom drive configurations and/or small setup or user defined control programs which can be executed as part of the start-up procedure just prior to the boot process. Contact the factory for more information concerning these custom features.

SECTION 6 - FLASHCOM

FlashCom is a firmware package designed to work with MCSI's IND-486H Single Board Computer (SBC) products. The package can be used for two basic functions:

- Remote downloading/programming of PROMDISK Flash Memory
- Remote download and execution of special software out of the remote embedded PC's DRAM without modifying PROMDISK (this feature is for future expansion by MCSI or the OEM)

For both of these tasks, FlashCom utilizes a special program on the embedded PC which resides in MCSI's PROMDISK firmware ROM. FlashCom first checks the Data Set Ready (DSR) Modem Status Line for the selected COM port, if it is active, then it monitors the port for a preset length of time at BAUD rates from 9600 to 115000. If a valid startup message is received, the software sets itself up for performing one of the above remote tasks, if a valid start-up message is not received it passes control back to the BIOS.

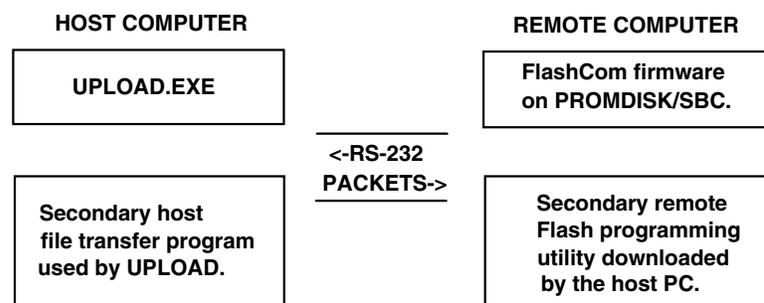
6.1 HOW DOES FLASHCOM OPERATE?

What FlashCom does is determined by the commands it receives over the serial port. Commands include the ability to copy RS-232 received data to the PC's conventional memory and to transfer control to the received data. This protocol is used by the MCSI supplied Flash update software run on an external DOS compatible PC, which would typically be a notebook or laptop computer. As of the writing of this manual, the Flash update software is the only utility that uses this protocol, but future utilities are planned. The technical specifications of the protocol are available from MCSI for OEMs wanting to write their own downloadable programs. Please contact MCSI for more information.

FlashCom also includes two utility programs, PUTFILE and GETFILE, that allow transferring files between two computers. These are generic programs which you may also find useful in other embedded situations.

6.2 FLASHCOM COMPONENTS

The core FlashCom package consists of four components -- RS-232 uploading firmware, a Flash programming module, a remote communications module, and a control program for the host portable computer. For maximum flexibility, only the FlashCom firmware is kept on the embedded PC. The other three programs are kept on the host PC. The diagram below shows the relationship between these four files.



The actual file names are listed below.

File	Purpose
SENDEXE.EXE	Primary control program to send .exe file to execute in remote PC.
UPLOAD.EXE	Primary control program
GPUPLOAD.EXE	Secondary program used by UPLOAD that reads .PD* files and sends them to the remote PC.
AMD733.BIN	Flash programming program for AMD 29F040 Flash EPROMs
INT733.BIN	Flash programming program for Intel/AMD 28F010/020 Flash EPROMs
SST733.BIN	Flash programming program for SST 28F040 Flash EPROMs
PUTFILE.EXE	File transfer software (sending side)
GETFILE.EXE	File transfer software (receiving side)
PDCR733.EXE	PROMDISK Programming Utility

The two primary programs, UPLOAD.EXE and the PROMDISK/FlashCom firmware, establish communication via the serial port and then transfer control to secondary programs using a pre-defined protocol.

PUTFILE and GETFILE are simple general purpose file transfer programs for the COM ports. These files are not required for FlashCom, and are supplied as convenient utilities.

6.2.1 UPLOAD

The host computer control program is a DOS based command-line driven program. It is designed much like the small FlashCom driver in that once it downloads the remote program, it leaves the rest of the work to a separate .EXE program which it runs using a DOS shell. A sample command line to program Intel/AMD 28F010/020 Flash EPROMS is shown below:

```
UPLOAD 1 115000 INT733.BIN GPUPLOAD APNAME.PD1 2 256 1
```

The first two parameters list the COM port number followed by the BAUD rate which can be any standard DOS BAUD rate from 9600 to 115000. Unless you are experiencing problems, we recommend 115000 BAUD. The second two parameters list the names of the "real" programs that take care of transferring the PROMDISK data and programming the 28F010/020 Flash parts -- in this case INT733.BIN and GPUPLOAD.EXE (.EXE extension assumed).

For programming AMD 29F040 Flash parts, the following command would be used:

```
UPLOAD 1 115000 AMD733.BIN GPUPLOAD APNAME.PD1 2 512 1
```

The *.PDx files will be programmed into the remote Flash parts and each sector will be sent with a checksum to detect anomalies in the transmission process. The embedded PC will also report back any problem programming the Flash parts which will cause the sending program to stop and display an error message. You can tell by looking at the line above the error message which chip did not program.

6.2.2 UPLOAD PROGRAMMING AND VERIFICATION OPTIONS

To insure that all of the binary images were successfully programmed, you may want to perform a final read back step. To do this, add "/PV" at the end of the command line when you run UPLOAD.EXE. A complete byte-for-byte verification will be performed after all of the chips have been programmed.

To verify only, add "/V" instead of "/PV." This will not erase or program the Flash parts, but will rather compare their contents with a set of *.PDx files stopping when an anomaly is found. This is a quick method of determining if an embedded PC has the latest set of PROMDISK images.

6.3 GENERAL PURPOSE RS-232 FILE TRANSFER PROGRAMS

Beyond transferring complete PROMDISK images with FlashCom, it may be necessary or "convenient" to copy individual files -- either to an SRAM PROMDISK drive, an MS-DOS RAM drive, or to the Flash File System. Two MCSI utility programs GETFILE and PUTFILE are supplied to support this requirement. One is run on the sending side and one is sent to the receiving side.

6.3.1 USING PUTFILE

PUTFILE.EXE reads files on the host computer and sends them using a "FlashCom-like" protocol out either COM1 or COM2. It is invoked by specifying the COM port, the BAUD rate, and a filename which can contain wild cards (* or ?). Enter PUTFILE with no arguments at the DOS prompt for more detailed explanations. An example command is shown below.

```
PUTFILE 1 115000 APREX1.EXE
```

Since it is impossible to know the speed of the remote disk the data is eventually being copied to, the PUTFILE program does not time-out. It will exit after copying the last file. You can also exit early by pressing the <Esc> key.

6.3.2 USING GETFILE

GETFILE.EXE responds to messages from PUTFILE.EXE and creates new files as specified. The date and time of the file will not be as originally sent, but will match the remote computer's date and time at the instant the file was transferred.

To invoke GETFILE, specify the COM port number and BAUD rate -- optionally followed by a time-out from 5 to 30 seconds. If no time-out is specified, the time-out will default to 15 seconds. An example command is shown below:

```
GETFILE 1 115000 30
```

GETFILE will exit on its own if (A) a command from the host is received telling it there are no more files, (B) if no message is received from the host in the specified time-out, or (C) if a message is received out of synchronization. As with the PUTFILE program, you can also abort early by pressing the <Esc> key.

6.4 OTHER USES FOR FLASHCOM

The basic portion of FlashCom embedded in the PROMDISK firmware essentially does only four things -- synchronize on a BAUD rate, copy blocks of data from the serial port to memory, copy blocks of data from memory to the serial port, and jump to an address in memory. This allows the architecture to be used for much more than updating Flash EPROMs. The SENDEXE.EXE program is used to download a ".exe" program and run it out of memory without modifying the code in PROMDISK.

The following files are provided as examples:

TEST.ASM	Sample of user assembly source code
TEXT.BIN	Sample executable test program
T.BAT	File to convert .exe to .bin

The above sample test program can be sent and executed in the target FlashCom system by entering the following on the host system and rebooting the target system:

```
SENDEXE 1 115000 TEST.BIN
```

SECTION 7 - INSTALLATION

This section describes the procedures for installing the IND-486H All-In-One Single Board Computer into your system. The following is a list of typical peripherals required to build a minimum system:

- Passive Backplane and Power Supply
- IBM PC/AT Type Keyboard
- Display Adapter Card
- Display Monitor
- Floppy or Hard Disk with MS-DOS

7.1 INSTALLING THE SIMMS

When installing or removing the DRAM SIMMs, be sure to first touch a grounded surface to discharge any static electricity from your body. Use the following procedure to install the SIMMs:

1. Insert the first SIMM edge connector at a slight angle into the SIMM2 socket closest to the edge of the board. Note that the SIMMs are keyed and will only go in one way.
2. Push the SIMM back into the connector carefully until it snaps into place.
3. Check to make sure the SIMM is inserted securely.
4. If required, insert the second SIMM edge connector at a slight angle into the SIMM1 socket.

To remove a SIMM, use a small screw driver to pull back the holding clip on each side of the SIMM and lift the SIMM from the connector.

7.2 INSTALLING THE PROMDISK MEMORY

The IND-486H's PROMDISK supports the popular JEDEC standard byte-wide memory devices. For proper operation the devices should have a *maximum access time of less than 150 nanoseconds*. The memory chips are installed bottom to top starting at the bottommost socket. Be sure that the PROMDISK configuration jumpers are properly set in accordance with Section 4 of this manual, and that you first touch a grounded surface to discharge any static electricity from your body. Use the following procedure to install the PROMDISK memory chips:

1. Line pin one of the memory chip indicated usually with a notch or white dot with pin one (upper leftmost pin) of the socket.
2. Carefully push the chip into the socket until the bottom of the chip comes in contact with the bottom of the socket.
3. Check to make sure that all the pins of the chip are installed and that there are no pins bent under the chip.
4. Repeat Steps 1-3 for the remaining chips.

To remove a memory chip, wedge a very small screwdriver between the chip and the socket, and gently pry the chip out of the socket. Care must be taken not to insert the screwdriver under the socket otherwise serious damage to the socket may result.

7.3 COMPLETING THE INSTALLATION

To complete the installation, the following steps should be followed:

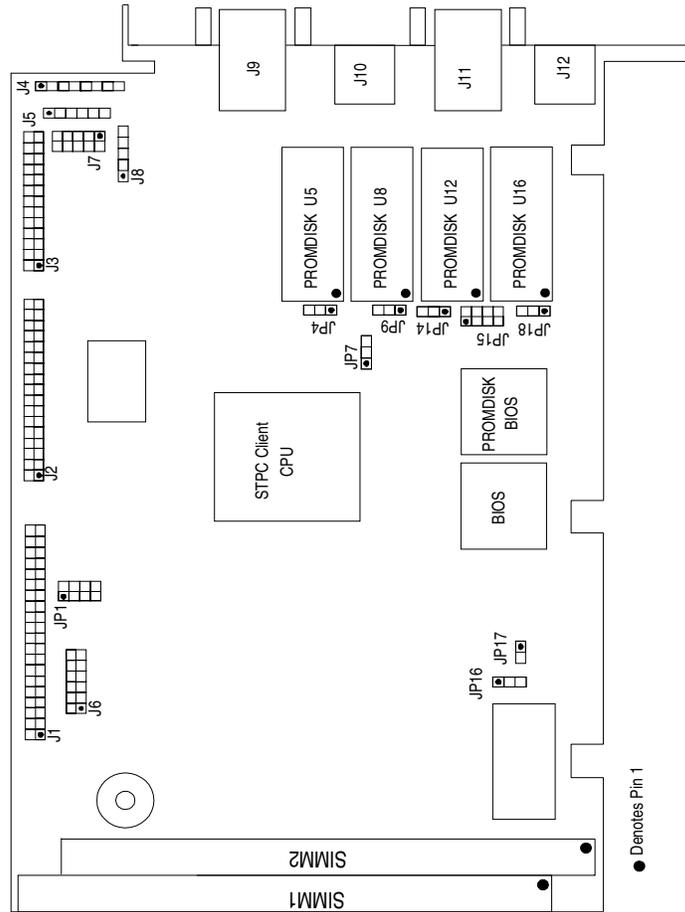
1. Set the configuration jumpers in accordance with Appendix E.
2. Install the IND-486H SBC into one of the 98-pin I/O slots in a passive backplane.
3. If the on-board VGA is *not* used, install a VGA Card into one of the slots in the backplane.
4. Connect the applicable I/O cables and peripherals, i.e. floppy disk, IDE hard disk, monitor, power supply, etc.
5. Connect an IBM PC compatible keyboard.
6. Turn power on to the display monitor.
7. Turn power on to the backplane power supply.
8. Reconfigure the IND-486H SBC using internal CMOS SETUP utility.
9. Boot the system.

APPENDIX A - SPECIFICATIONS

This appendix lists the specifications for the IND-486H All-In-One Single Board Computer.

CPU:	IND-486H includes ST Micro 486DX2-66 CPU Core
Co-processor:	Internal to STPC Client chipset
Memory:	System Memory Expandable to 64M-bytes. Supports: 8MB, 16MB, or 32MB SIMMs using two 72-pin SIMM sockets. Both standard fast page mode and high performance EDO DRAMs are supported. Optional on-board (soldered in) System memory up to 8MB.
Disk Emulator:	2M PROMDISK Disk Emulator, 4-32pin sockets, two sockets support EPROMs and FLASH EPROMs, and two sockets support EPROMs, FLASH EPROMs, NVRAMs, & SRAMs with on-board battery back-up.
BIOS:	AMI with Internal SETUP
Clock/Cal:	PC/AT Compatible with battery back-up
I/O Bus:	IBM PC/AT Compatible 98-pin Edge Connector
DMA:	7 Channels (DMA Channel 3 inoperative)
Timers:	3 Programmable
Interrupts:	16
Reset:	Controlled by on-board power detector with provisions for external reset switch
I/O Ports:	<p>I/O Controller Winbond W83977F-A</p> <p>2 - RS-232 Serial Ports (COM1 at rear connector J11 and COM2 at J7)</p> <p>1 - Parallel Printer Port (J3)</p> <p>1 - PS2/AT Compatible Keyboard Port (at rear connector J12 and J8)</p> <p>1 - PS2 Mouse Port (at rear connector J10)</p> <p>1 - IrDA Data Port (J5)</p> <p>1 - On-board Speaker with Speaker Port (J6)</p> <p>1 - Dual 3.5"/5.25" Floppy Disk Port (J2)</p> <p>1 - IDE Hard Disk Port (J1)</p> <p>1 - WatchDog Timer</p> <p>1 - External Power Connector (J4)</p>
	<p>1 - PCI VGA Video Port (at rear connector J9)</p> <p>Chipset: STPC Client</p> <p>VRAM: Shares system DRAM (2 SIMMs or 4 on-board DRAMs)</p> <p>Resolution: 1024 x 768, 64K color, 75Hz 800 x 600, full color, 90Hz</p>
Speed:	66MHz
Battery:	Lithium for Clock/Calendar (five years operation under typical conditions)
Benchmark:	LANDMARK v2.0 = 221 MHz
Size:	Half Size AT board 7.0"L X 4.8"H
Power:	+5VDC @ 1.4A, ±12VDC @ 0.020A

APPENDIX B - BOARD OUTLINE



APPENDIX C - MEMORY AND I/O MAPS

The following is the memory map for the IND-486H SBC. The addresses are fully PC/AT compatible, unless otherwise specified.

IND-486H SBC Memory Map

Address	Used For	Size
00000H - 003FFH	Interrupt Vectors	1.0K
00400H - 005FFH	BIOS Values	0.5K
00600H - 7FFFFH	User RAM (DOS)	510.5K
80000H - 9FFFFH	User RAM (DOS)*	128.0K
A0000H - AFFFFH	Reserved for EGA*	64.0K
B0000H - B7FFFH	Video RAM (MDA)*	32.0K
B8000H - BFFFFH	Video RAM (CGA)*	32.0K
C0000H - C3FFFH	Video BIOS*	16.0K
C4000H - C7FFFH	EMS Window	16.0K
C8000H - DFFFFH	ROM Scan Devices*	96.0K
E0000H - E3FFFH	PROMDISK Window	16.0K
E4000H - E4FFFH	PROMDISK Driver	4.0K
E5000H - EFFFFH	Custom Firmware	44.0K
F0000H - FFFFFH	BIOS**	64.0K
100000H - 3FFFFFFH	On-board System Memory	3.0M
400000H - 7FFFFFFH	On-board Memory Expansion	4.0M
800000H - 3FFFFFFH	SIMM Memory	63.0M

**External to the IND-486H*

*** Includes WatchDog Timer Address*

The following is the I/O map for the IND-486H SBC. I/O addresses are fully PC/AT compatible, unless otherwise specified.

IND-486H SBC I/O Map

Address	Function
000H - 01FH	DMA Controller #1
020H - 021H	Interrupt Controller #1
022H - 023H	Configuration Address Register
040H - 05FH	System Timers
060H - 063H	Keyboard, Status, & System Control
070H - 07FH	Clock/Calendar & CMOS Ram Access
080H - 083H	DMA Page Register
0A0H - 0BFH	Interrupt Controller #2
0C0H - 0DFH	DMA Controller #2
1F0H - 1F8H	IDE Hard Disk
238H - 23FH	Serial Port COM2 (Optional)
278H - 27FH	Parallel Printer Port LPT1 (Optional)
2F8H - 2FFH	Serial Port COM2 (Default)
338H - 33FH	Serial Port COM1 (Optional)
378H - 37FH	Parallel Printer Port LPT1 (Default)
3BCH - 3BFH	Parallel Printer Port LPT1 (Optional)
3F0H - 3F7H	Floppy Disk Controller
3F8H - 3FFH	Serial Port COM1 (Default)

APPENDIX D - CONNECTORS

J12 Keyboard Connector (PS2 type)

Pin	Signal
1	KBDATA
2	N/C
3	GND
4	+5VDC
5	KBCLOCK
6	N/C

J8 Keyboard Header

Pin	Signal
1	KBCLK
2	KBDATA
3	N/C
4	GND
5	+5VDC

J10 PS2 Mouse Connector

Pin	Signal
1	MSDATA
2	N/C
3	GND
4	+5VDC
5	MSCLK
6	GND

J5 IrDA Infrared Port Connector

Pin	Signal
1	+5VDC
2	FIR-RX
3	IR-RX
4	GND
5	IR-TX
6	CIR-RX

J4 External Power Connector

Pin	Signal
1	+5VDC
2	+12VDC
3	-12VDC
4	GND
5	GND
6	-5VDC
7	+12VDC
8	+5VDC

J2 Floppy Disk Port Connector

Pin	Signal Name
2	RPMLC
4	Not Used
6	Not Used
8	INDEX/
10	MOTOR0/
12	DRIVE SELECT1/
14	DRIVE SELECT0/
16	MOTOR1/
18	DIRECTION
20	STEP/
22	WRITE DATA/
24	WRITE GATE/
26	TRACK0/
28	WRITE PROTECT/
30	READ DATA/
32	HEAD SELECT/
34	DISK CHANGE/

All odd numbered pins are GND

J1 IDE Hard Disk Port Connector

Pin	Signal	Pin	Signal
1	IDERESET/	2	GND
3	IDED7	4	IDED8
5	IDED6	6	IDED9
7	IDED5	8	IDED10
9	IDED4	10	IDED11
11	IDED3	12	IDED12
13	IDED2	14	IDED13
15	IDED1	16	IDED14
17	IDED0	18	IDED15
19	GND	20	Not Used
21	NOT USED	22	GND
23	IDEIOW/	24	GND
25	IDEIOR/	26	GND
27	IOCHRDY	28	NOT USED
29	IDEDACK/	30	GND
31	IRQ14	32	NOT USED
33	IDESA1	34	NOT USED
35	IDESA0	36	LA19
37	HDCS0/	38	HDCS1/
39	IDEACT/	40	GND

J6 Miscellaneous I/O Header

Pin	Signal Name	Description
1	RESETSW/	Connect to ground to reset
2	GND	Ground
3	PDSEL1/	PROMDISK feature select 1
4	GND	Ground
5	PDSEL0/	PROMDISK feature select 0
6	GND	Ground
7	KEYLOCK/	Connect this pin to ground to inhibit keyboard.
8	GND	Ground
9	SPEAKER (-)	Connect to Speaker (-)
10	+5VDC (20ma)	Connect to anode of IDE LED
11	IDLED (-)	Connect to cathode of IDE LED
12	+5VDC (fused)	Connect to Speaker (+)

J3 Printer Interface Connector

Pin	Signal	Pin	Signal
1	STROBE/	2	AUTOFD/
3	PDAT0	4	ERROR/
5	PDAT1	6	INIT/
7	PDAT2	8	SLCTIN/
9	PDAT3	10	GND
11	PDAT4	12	GND
13	PDAT5	14	GND
15	PDAT6	16	GND
17	PDAT7	18	GND
19	ACK/	20	GND
21	BUSY	22	GND
23	PE	24	GND
25	SLCT	26	GND

J11 Serial Port #1 9-pin Sub D Connector

Pin	Signal Name
1	CARRIER DETECT #1
2	RECEIVE DATA #1
3	TRANSMIT DATA #1
4	DATA TERMINAL READY #1
5	GND
6	DATA SET READY #1
7	REQUEST TO SEND #1
8	CLEAR TO SEND #1
9	RING INDICATOR #1

J7 Serial Port #2 10-pin Header/Connector

Pin	Signal Name
1	CARRIER DETECT #2
2	DATA SET READY #2
3	RECEIVE DATA #2
4	REQUEST TO SEND #2
5	TRANSMIT DATA #2
6	CLEAR TO SEND #2
7	DATA TERMINAL READY #2
8	RING INDICATOR #2
9	GND
10	N/C

J9 VGA Display Connector (15-pin Sub-D)

Pin	Signal	Pin	Signal
1	RED	2	GREEN
3	BLUE	4	N/C
5	GND	6	GND
7	GND	8	GND
9	N/C	10	GND
11	N/C	12	DDCDAT
13	HSYNC	14	VSYNC
15	DDCCLK		

APPENDIX E - CONFIGURATION JUMPERS

This section provides a summary of the configuration jumpers. The tables below and the Board Outline in Appendix B serve as quick references for determining the jumper settings, and a means of finding detailed information elsewhere in the manual.

PROMDISK Memory Configuration Jumpers for Drive 1 (U16 & U12) JP18 & JP14 (Section 4.2)

Chip P/N	Type	Size	JP18	JP14
27C010	EPROM	128Kx8	OPEN	OPEN
27C020	EPROM	256Kx8	OPEN	OPEN
27C040	EPROM	512Kx8	OPEN	OPEN
28F010	FLASH	128Kx8	1-2	1-2
28F020	FLASH	256Kx8	1-2	1-2
29F040	FLASH	512Kx8	2-3	1-2
PH28SF040	FLASH	512Kx8	2-3	1-2

Note: Chips within a drive group cannot be mixed.

PROMDISK Memory Configuration Jumpers for Drive 2 (U8 & U5) JP15, & JP9 (Section 4.2)

Chip P/N (Generic)	Type	Size	JP15 1-2	JP15 3-4	JP9
27C010	EPROM	128Kx8	On	Off	OFF
27C020	EPROM	256Kx8	On	Off	OFF
27C040	EPROM	512Kx8	On	Off	OFF
DS1245	NVRAM	128Kx8	Off	Off	2-3
HM628128L	SRAM	128Kx8	Off	Off	2-3
Various	SRAM	512Kx8	Off	Off	2-3
28F010	FLASH	128Kx8	On	On	1-2
28F020	FLASH	256Kx8	On	On	1-2
28/29F040	FLASH	512Kx8	On	On	2-3
PH28SF040	FLASH	512Kx8	On	On	2-3

Note: Chips within a drive group cannot be mixed.

JP4 PROMDISK U8 & U5 Power Control Jumper

Mode	JP4	Section
Normal	1-2	4.3
Battery	2-3	4.3

Note: Setting the power control jumpers to the "battery power mode" is for SRAM use only.

JP15 PROMDISK FlashCom Feature Jumper

Function	5-6	7-8	Section
Disabled	OFF	OFF	4.5
Enabled - COM1	OFF	ON	4.5
Enabled - COM2	ON	OFF	4.5

JP16 Clear CMOS Memory Jumper

1-2	Normal Operation (Default)	Section 3.2
2-3	Clear CMOS Data	Section 3.2

JP7 WatchDog Timer Control Jumper (Section 2.17)

2-3	Generates hardware RESET when time out occurs.
1-2	Generates NMI (IOCHRDY) when time out occurs.
OFF	Disable (Default)

JP1 WatchDog Timer Time-out Period Jumper

Time	1-2	3-4	5-6	7-8
1second	OFF	OFF	ON	OFF
2 seconds	OFF	OFF	ON	ON
10 (Default)	OFF	ON	OFF	OFF
20 seconds	OFF	ON	OFF	ON
110 seconds	ON	OFF	OFF	OFF
220 seconds	ON	OFF	OFF	ON

JP17 Mouse Enable Jumper

ON	Enabled (Default)
OFF	Disabled

APPENDIX F - BIOS ERROR BEEP CODES

During the POST (Power On Self Test) routines, which are performed each time the system is powered on, errors may occur.

Nonfatal errors are those which, in most cases, allow the system to continue the boot up process. The error messages normally appear on the screen.

Fatal errors are those which will not allow the system to continue the boot-up procedure. If a fatal error occurs, you should consult with MCSI Customer Service for possible repairs.

These fatal errors are communicated through a series of audible beeps. The numbers on the fatal error list below correspond to the number of beeps for the corresponding error. All errors listed, with the exception of number eight, are fatal errors.

No. of Beeps	Error Message
1	Refresh Failure - The memory refresh circuitry is faulty.
2	Parity Error - A parity error was detected in the first 64K block of system memory.
3	Base 64KB Memory Failure - A memory failure occurred within the first 64KB of memory.
4	Timer Not Operational - Timer #1 has failed to function properly.
5	Processor Error - The CPU chip has generated an error.
6	8042-Gate A20 Failure - The keyboard controller (8042) contains the Gate A20 switch which allows the CPU to operate in virtual mode. This error message means that the BIOS is not able to switch the CPU into protected mode.
7	Processor Exception Interrupt Error - The CPU chip has generated an exception interrupt.
8	Display Memory Read /Write Error - The video adapter is either missing or the video memory is faulty. PLEASE NOTE: This is not a fatal error.
9	ROM Checksum Error - The ROM checksum value does not match the value encoded in the BIOS.
10	CMOS Shutdown Register Read/Write Error - The shutdown register for the CMOS memory has failed.
11	Cache Memory Read/Write Error - A Cache Memory failure occurred, do not enable the Cache Memory to resume operation.

APPENDIX G - SUGGESTED PROMDISK MEMORY CHIPS

The PROMDISK Disk Emulator supports the popular JEDEC standard byte-wide memory devices. For proper operation the devices should have a *maximum access time of less than 150 nanoseconds*. The following is a list of some of the more common generic chip types:

EPROM:	128Kx8	D27C010-150 <i>Intel</i>
	256Kx8	D27C020-150 <i>Intel</i>
	512Kx8	D27C040-150 <i>Intel</i>
	128Kx8	AT27C010R-15DC <i>Atmel</i>
	256Kx8	AM27C020-150DC <i>AMD</i>
	512Kx8	AM27C040-150DC <i>AMD</i>
	512Kx8	TMS27C040-15JL <i>TI</i>
SRAM:	128Kx8	KM681000ALP-15 <i>Samsung</i>
	128Kx8	M5M51008P-150L <i>Mitsubishi</i>
	128Kx8	TC551001PL-150L <i>Toshiba</i>
	512Kx8	58001 <i>MCSI</i>
NVRAM:	128Kx8	DS1245Y-150 <i>Dallas</i>
FLASH:	128Kx8	P28F010-150 <i>Intel</i>
	256Kx8	P28F020-150 <i>Intel</i>
	128Kx8	AM28F010-150C3PC, <i>AMD</i>
	256Kx8	AM28F020-150C3PC, <i>AMD</i>
	512Kx8	AM29F040-150, <i>AMD</i>
	128Kx8	SST29EE010-150, <i>SST</i>
	512Kx8	SST28SF040-150, <i>SST</i>

Note: For chip types not listed, please consult the factory.

APPENDIX H - FLASHCOM SERIAL INTERFACE

A standard Modem interface cable can be used to interface the Host to the Target computer. FlashCom uses a simple 4-wire interface as shown below:

