



**LITTLE BOARD/186 UTILITIES**

**USER'S MANUAL**

**P/N: A74026-A**

## **DISCLAIMER**

AMPRO Computers, Incorporated makes no representations or warranties with respect to the contents of this manual or of the associated AMPRO software package, and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. AMPRO shall under no circumstances be liable for incidental or consequential damages or related expenses resulting from the use of this product, even if it has been notified of the possibility of such damages. AMPRO reserves the right to revise this publication from time to time without obligation to notify any person of such revisions. If errors are found, please contact AMPRO at the address listed on the title page of this document.

## **TRADEMARKS**

**DBase:** Ashton-Tate  
**IBM:** International Business Machines, Inc.  
**Flight Simulator, Multiplan:** Microsoft, Inc.  
**Little Board and SCSI/PLUS:** AMPRO Computers, Inc.  
**Lotus 1-2-3, Symphony:** Lotus Development Corp.  
**SuperCalc:** Sorcim/IUS Micro Software  
**Wordstar:** MicroPro International Corp.

**COPYRIGHT 1985,86,87 AMPRO COMPUTERS INCORPORATED  
ALL RIGHTS RESERVED**

No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of AMPRO Computers, Incorporated.

## TABLE OF CONTENTS

### CHAPTER 1 - GENERAL INFORMATION

1.1	Introduction . . . . .	1-1
1.2	Manual Organization . . . . .	1-1
1.3	Conventions . . . . .	1-1
1.4	References . . . . .	1-2

### CHAPTER 2 - OPERATION WITH PC-DOS

2.1	Introduction . . . . .	2-1
2.2	Hardware Considerations . . . . .	2-1
2.3	Operating System . . . . .	2-1
2.3.1	Device Assignment . . . . .	2-2
2.3.2	Internal Commands . . . . .	2-2
2.3.3	Drivers and Utilities . . . . .	2-2
2.4	Applications Software . . . . .	2-4
2.4.1	MS-DOS Generic Programs . . . . .	2-4
2.4.2	Programs Which Make ROM-BIOS Calls . . . . .	2-4
2.4.3	Programs Which Make Direct Hardware Access . . . . .	2-5
2.4.4	Designing Software for Compatibility . . . . .	2-5
2.5	Configuration Options . . . . .	2-5
2.5.1	The Little Board/186 Utilities . . . . .	2-6
2.5.2	The AMPRO SCSI Utilities . . . . .	2-7

### CHAPTER 3 - SOFTWARE INSTALLATION

3.1	Introduction . . . . .	3-1
3.2	Two Special Files . . . . .	3-1
3.2.1	CONFIG.SYS . . . . .	3-1
3.2.2	AUTOEXEC.BAT . . . . .	3-2
3.3	Terminal . . . . .	3-2
3.3.1	Baud Rate . . . . .	3-2
3.3.2	Display Control Codes . . . . .	3-2
3.3.3	PC Function Key Simulation . . . . .	3-3
3.3.4	Hardware Handshaking . . . . .	3-3
3.4	Printers . . . . .	3-4
3.4.1	Using a Parallel Printer . . . . .	3-4
3.4.2	Using a Serial Printer . . . . .	3-5
3.5	Modems . . . . .	3-6
3.6	Unique Floppy Configurations . . . . .	3-6
3.6.1	Single-Floppy Configurations . . . . .	3-7
3.6.2	Using DRIVER.SYS . . . . .	3-7
3.6.3	Using the "DRIVPARM" Option . . . . .	3-8
3.6.4	Custom Floppy Parameters . . . . .	3-9
3.7	Hard Disk Drives . . . . .	3-9
3.8	PC-DOS RAM Disk . . . . .	3-9
3.9	CP/M File Access . . . . .	3-10
3.10	Example CONFIG.SYS . . . . .	3-10

## CHAPTER 4 - AMPRO DEVICE DRIVERS AND UTILITIES

4.1	Introduction . . . . .	4-1
4.2	Program Descriptions . . . . .	4-1

AMPROCLK.EXE . . . .	Hardware RTC support
AMPROKEY.SYS . . . .	ASCII terminal keyboard driver
CHKROM.COM . . . . .	ROM-BIOS checksum and version
CPMDRVR.SYS . . . . .	CP/M file access driver
FUNKEY.SYS . . . . .	Keyboard driver enhancement
KT7KEY.SYS . . . . .	Keyboard driver for Kimtron KT7
KT7TERM.SYS . . . . .	Display driver for Kimtron KT7
LBCOMM.EXE . . . . .	Communications utility
MONITOR.EXE . . . . .	EPROM monitor demonstration
SETCOM1.COM . . . . .	Aux serial port initialization
SETCON.SYS . . . . .	Console port initialization
SETCPM.COM . . . . .	CP/M format selection
SETFDC.COM . . . . .	Floppy drive parameter initialization
TERM.SYS . . . . .	ASCII terminal display driver

## CHAPTER 1

### GENERAL INFORMATION

#### 1.1 INTRODUCTION

This manual is for users and integrators of systems based on the AMPRO Little Board/186 single board computer, utilizing the IBM PC-DOS operating system. It describes the PC-DOS support software included on the AMPRO Little Board/186 Utilities software diskette, and how to perform typical software installation for a variety of system configurations. Additional support software for the board's SCSI interface is contained in the AMPRO SCSI Utilities package, and is described in that software's User's Manual (AMPRO P/N A74027).

#### 1.2 MANUAL ORGANIZATION

This manual is organized as follows:

**Chapter 1 - GENERAL INFORMATION** - An introduction to the structure of this manual, including conventions used in program operation descriptions.

**Chapter 2 - OPERATION WITH PC-DOS** - Options available with the PC-DOS operating system, a discussion of compatibility with standard PC-DOS software, and a brief description of the software utilities supplied by AMPRO with the Little Board/186.

**Chapter 3 - SOFTWARE INSTALLATION** - How to configure system software for use with a wide variety of configurations and peripherals.

**Chapter 4 - PROGRAM DESCRIPTIONS** - Detailed descriptions and operating instructions for each of the Little Board/186 device drivers and utility programs.

Chapter 4 is intended to provide a convenient user reference. The program descriptions are arranged alphabetically.

#### 1.3 CONVENTIONS

In the descriptions of the use of software utilities, terminal keyboard inputs which you will make to the system are shown underlined. This has been done to make it easy for you to distinguish between the computer's prompts and the operator's keystrokes. For example,

A>DIR<RETURN>

means that you type "DIR" and then press the <RETURN> key on your keyboard.

Also, certain keys on your terminal's keyboard have special uses. The control key, generally labeled CTRL, is meant to be pressed at the same time as another key. In this manual, such two-key combinations are written: <CTRL-C>, which means "hold down the CTRL key while you press the C key."

Two other special keys are the "escape" key, indicated by <ESC> and the "return" key (also called the "carriage return" or "enter" key), indicated by <RETURN>. In general, all commands you enter from the PC-DOS command prompt require you to press the <RETURN> key to begin the operation, as in the example above.

#### 1.4 REFERENCES

Some familiarity with the IBM PC-DOS operating system is assumed. Please refer to the appropriate IBM PC-DOS reference manuals for further information on the use of the IBM PC-DOS operating system and its drivers and utilities.

In addition, reference to the following AMPRO documentation is recommended:

**AMPRO Little Board/186 Technical Manual (P/N A74011)** - Information on the operation and capabilities of the Little Board/186, and its configuration and setup.

**AMPRO Little Board/186 Utilities User's Manual (P/N A74026)** - You are reading it now!

**AMPRO SCSI Utilities User's Manual (P/N A74027)** - Information on the installation and operation of the AMPRO SCSI drivers and utilities software for use with the Little Board/186 and PC-DOS operating system.

## CHAPTER 2

### OPERATION WITH PC-DOS

#### 2.1 INTRODUCTION

This chapter provides an overview of the system configuration options and utilities that are available under the PC-DOS operating system, as well as a discussion of the degree of compatibility that the Little Board/186 offers relative to software written for operation on the IBM PC and compatibles (the "standard PC").

A combination of standard IBM PC-DOS utilities, along with utilities supplied by AMPRO, allows you to create a highly customized system based on the Little Board/186. Common system software configurations based on the functions provided by the Little Board/186 Utilities are covered in Chapter 3 of this manual, while Chapter 4 describes that software in detail. In addition, the AMPRO SCSI Utilities User's Manual (P/N A74027) covers the procedures and software required for installation and operation of hard disks and other SCSI devices.

#### 2.2 HARDWARE CONSIDERATIONS

The Little Board/186 single board computer is based on an 80186 integrated, high-performance 16-bit microprocessor, which provides a functional superset of the 8-bit 8088 microprocessor used in the "standard PC." Programs written for an 8088 microprocessor can run on an 80186 without modification, but with a performance improvement of up to 400 percent.

Many of the hardware devices present on the Little Board/186, and their I/O port addresses, differ from those of the "standard PC." This includes: the 2681 serial communications controller; the 1772 floppy disk controller; the hard disk interface (SCSI); and the DMA, interrupt, and timing controllers contained within the 80186 microprocessor. Furthermore, the Little Board/186 utilizes an RS232C ASCII terminal as a user console (keyboard and monitor) rather than the keyboard and display controllers used in a "standard PC."

As a result, programs which make direct access to board hardware, including both I/O ports and display controller "video RAM," rather than using the operating system or ROM-BIOS functions provided for the same purpose, can not be used without I/O driver modifications.

Refer to the Little Board/186 Technical Manual (P/N A74011) for further technical information on the hardware.

#### 2.3 OPERATING SYSTEM

The Little Board/186 ROM-BIOS allows the use of IBM PC-DOS versions 2.x or 3.x as the board's operating system. PC-DOS version 3.x is required for hard disk usage, while version 3.2 is required for operation with 80-track (720K) mini or micro floppy drives.

When operated under PC-DOS, the Little Board/186 provides full PC file and data compatibility. Diskettes may be copied, formatted, verified, etc. on either for the other. All of the PC-DOS 40-track mini floppy formats, and the PC-DOS Version 3.2 80-track micro floppy formats, are supported, including single- and double-sided, and eight and nine sectors per track.

### **2.3.1 Device Assignment**

In addition to supporting the standard PC-DOS functions, the Little Board/186 ROM-BIOS provides a software interrupt interface which is a subset of that of the "standard PC" ROM-BIOS software interrupt structure.

Serial Port A is supported as the PC-DOS console device ("CON"), so that an RS232 ASCII terminal can be used for the required keyboard and monitor functions. Serial Port B is supported as the PC-DOS "COM1" port; the parallel printer port is "LPT1". The time of day clock and floppy disk I/O functions are also supported in the standard manner. Hard disk support (INT 13H) is also provided in the ROM-BIOS, using the board's SCSI interface.

The video display interrupt (INT 10H) within the board's ROM-BIOS supports the "Write TTY" function only, but can be enhanced to provide additional functions such as clear screen, cursor positioning, and scrolling by installing the Little Board/186 video display driver (TERM.SYS). Keyboard drivers (AMPROKEY.SYS and FUNKEY.SYS) can also be installed in the system to enhance the compatibility of a terminal keyboard with that of the "standard PC."

### **2.3.2 Internal Commands**

Most of the "internal" PC-DOS commands may be used without restrictions. These include: ECHO, FOR, GOTO, IF, PAUSE, REM, CHDIR, COPY, CTTY, DATE, DIR, DELETE, ERASE, MKDIR, PATH, RENAME, RMDIR, SET, TIME, TYPE, VER, VERIFY, VOL, and PROMPT (Version 3). In addition, the powerful PC-DOS batch processing facility is fully usable.

### **2.3.3 Drivers and Utilities**

Many of the drivers and utilities included on the IBM PC-DOS operating system diskettes may be used with the Little Board/186. Of these, most work with no restrictions, while a few have some limitations. This section indicates which programs provided on the standard distribution PC-DOS diskettes may be used with Little Board/186, and what restrictions, if any, apply.

**PC-DOS Utilities/Drivers With No Restrictions** - The following standard IBM PC-DOS (Version 2 and 3) programs have no known restrictions:

ATTRIB.EXE	Set or clear file attributes
BACKUP.COM	Copies hard disk files to floppies
CHKDSK.COM	Disk directory
COMP.COM	Compares files
DEBUG.COM	Program development tool
DISKCOMP.COM	Compares entire disks
DISKCOPY.COM	Copies entire disk
DRIVER.SYS	Modifies floppy defaults (DOS 3.2)
EDLIN.COM	Text editor
EXE2BIN.EXE	Program development tool
FIND.EXE	Searches files for specific text
FORMAT.COM	Formats floppy disks
LINK.COM	Program development tool
MORE.COM	Displays a screen full of data
PROMPT.COM	Set new prompt
RECOVER.COM	Recovers files from disks
RESTORE.COM	Restores floppy files to hard disk
SORT.EXE	Sorts text data
TREE.COM	Displays all directory paths
VDISK.SYS	RAM disk driver
XCOPY.EXE	Enhanced copy utility (DOS 3.2)

**PC-DOS Utilities/Drivers With Restrictions** - The following standard IBM PC-DOS (Version 2 and 3) programs have known restrictions, as indicated:

Program: MODE.COM  
Function: Configures printer and display port characteristics.  
Restriction: MODE "Option 2" not supported.

Program: ANSI.SYS  
Function: Converts PC video display into an ANSI terminal.  
Restriction: Requires use of AMPRO terminal driver (TERM.SYS)

Program: SYS.COM  
Function: Writes operating system to a drive  
Restriction: Use for floppy disks only.

**PC-DOS Utilities/Drivers which are Unusable** - The following standard IBM PC-DOS programs or commands do not work at all: BREAK, GRAPHTABL, GRAPHICS, KEYBxx, BASIC, BASICA, and the programs associated with the IBM ROM-resident Basic (the Little Board/186 ROM-BIOS does not contain Basic). FDISK is replaced by the AMPRO hard disk partition editor, which is supplied in the AMPRO SCSI Utilities software.

## 2.4 APPLICATION SOFTWARE

Programs written for use on PC's can be categorized in three groups:

- (1) Programs which confine themselves to operating system calls
- (2) Programs which make ROM-BIOS calls
- (3) Programs which make direct hardware access

Of these, nearly all of group (1) are usable, some of group (2) are usable, and none of group (3) are usable (without modification). The following sections explain this in more detail.

### 2.4.1 MS-DOS Generic Programs

Programs and utilities written for operation on any MS-DOS system are called "MS-DOS Generic." These programs restrict their system access exclusively to functions provided by the operating system. MS-DOS Generic programs are "hardware independent"; they can be used on a variety of hardware implementations, including systems such as the Little Board/186 that use RS232C ASCII terminals as the system console. Such "well behaved programs" nearly always run without modification on the Little Board/186. MS-DOS Generic programs generally have a terminal installation utility which allows you to specify the terminal to be used as the system console device, usually from a menu.

Here are a few popular application programs known to be available in MS-DOS Generic versions: Multiplan (Version 1), Wordstar (Version 3), DBase II, SuperCalc2, the T/Maker Integrated Software package. In addition, most languages and software development tools have MS-DOS Generic versions, including: Microsoft C (Version 3), Basic, and MASM; Turbo Pascal, Palasm, Abel; the Intel 86-family development tools.

### 2.4.2 Programs Which Make ROM-BIOS Calls

Many popular programs written for use on the "standard PC" make direct ROM-BIOS calls. As long as these programs do not also access hardware directly -- including video RAM -- these programs can generally be used, providing that the display and keyboard drivers from the Little Board/186 Utilities are properly installed. One exception is that ASCII terminals do not generally provide pixel graphics functions, so this is not supported by the Little Board/186 terminal driver (TERM.SYS).

### 2.4.3 Programs Which Make Direct Hardware Access

Programs that talk directly to hardware (floppy controller, serial ports, keyboard port, video RAM, etc.) generally do not run on the Little Board/186 without modification. Some examples include:

- Copy protected programs: often use floppy controller, serial port, or printer port hardware.
- Communications programs: generally access serial port hardware
- Programs using graphics: generally access display controller hardware
- Programs that write to Video RAM

Included in this group are many "consumer programs," including: Lotus 1-2-3, Symphony, DBase III, and Flight Simulator.

### 2.4.4 Designing Software for Compatibility

To provide full compatibility between the Little Board/186 and the standard PC, the application software must simply confine itself to standard PC-DOS and the supported ROM-BIOS functions, and must not perform direct hardware accesses.

If you use the standard PC ROM-BIOS video functions (INT10H) for display control (cursor position, clear screen, scrolling, and intensity control), then the software you write will run on the Little Board/186 regardless of what type of terminal device is used -- providing that the board's display driver (TERM.SYS) is properly installed. In fact, your software will also run on a standard PC with any of its standard keyboard video controller devices! Refer to the Chapter 4 for details on the INT10 video support provided by the AMPRO TERM.SYS display driver.

## 2.5 CONFIGURATION OPTIONS

A wide degree of system configurability is available through options provided by: (1) the IBM PC-DOS operating system and its drivers and utilities; (2) the AMPRO Little Board/186 Utilities; and (3) the AMPRO SCSI Utilities.

The configuration options include:

- Terminal selection and setup (baud rate, etc.)
- Parallel and Serial printer port selection and setup
- Modem interface
- Additional -- or different types of -- floppy disk drives
- One or more SCSI hard disk drives
- RAM disk

Examples of how a typical system can be configured for each of these options are given in Chapter 3. A brief summary of the features of the AMPRO Little Board/186 Utilities and the AMPRO SCSI Utilities appears in the following two sections.

## 2.5.1 The Little Board/186 Utilities

Functions available through the Little Board/186 Utilities software include:

- Hardware initialization utilities
- PC display and keyboard simulation drivers
- Modem communications utility
- CP/M diskette file access driver and utility

The following drivers and utilities are contained on the Little Board/186 Utilities diskette(s) at this time. Actual utility names and descriptions may vary. Refer to Chapter 4 for detailed program descriptions and operating instructions.

**AMPROCLK.EXE** - Sets the PC-DOS system clock from one of two optional real time clock devices (Expansion/186 or SCSI/CLOCK).

**AMPROKEY.SYS** - Allows any ASCII terminal to simulate PC function keys.

**CHKROM.COM** - Reports ROM-BIOS checksum and version.

**CPMDRVR.SYS** - Allows file transfer to/from CP/M formatted diskettes, including the AMPRO 8-bit formats.

**FUNKEY.SYS** - Allows the function keys of specific ASCII terminals to directly simulate PC function keys. Enhances the operation of the AMPROKEY.SYS keyboard driver.

**KT7KEY.SYS** - Keyboard driver for the Kimtron KT7 PC terminal.

**KT7TERM.SYS** - Display driver for the Kimtron KT7 PC terminal.

**LBCOMM.EXE** - General purpose communication utility. Includes auto-dial, and both ASCII and XMODEM file transfer protocols.

**MONITOR.EXE** - Little Board/186 EPROM monitor demonstration. Although run under PC-DOS, it is equivalent in function to the optional EPROM monitor.

**SETCOM1.COM** - Port setup for Serial Port B, the DOS "COM1" port. Sets baud rate (110 to 57.6K baud), parity, data length, stop bits, and option of automatic hardware handshaking.

**SETCON.SYS** - Port setup for Serial Port A, the PC-DOS "CON" device. Similar in function to SETCOM1.COM. By including this driver as the first entry in the system CONFIG.SYS file, a console baud rate other than 9600 baud (110 to 38.4K baud) can be used.

**SETCPM.COM** - Selects desired CP/M format for use with CPMDRVR.SYS.

**SETFDC.COM** - Floppy drive parameter initialization utility. Used to set step rates for each floppy drive to other than the default value (6 mS).

**TERM.SYS** - Terminal display driver. Enhances the INT10H video support

present in the board's ROM-BIOS. Converts IBM PC video display functions for cursor position, clear screen, intensity control, scrolling, etc., into the control codes required by a selection of ASCII terminals. Allows use of many programs which lack "MS-DOS Generic" terminal installation options, but do not use video RAM or bit-mapped graphics. Terminal customization file source code is included, allowing easy addition of user-defined terminals.

## 2.5.2 The AMPRO SCSI Utilities

Functions available through the AMPRO SCSI Utilities software include:

- Hard disk formatter
- Hard disk partition editor
- Hard disk park utility
- SCSI block copy utility
- SCSI compare utility
- SCSI debugger utility

The following drivers and utilities are contained on the AMPRO SCSI Utilities diskette(s) at this time. Actual utility names and descriptions may vary. Refer to the AMPRO SCSI Utilities User's Manual (P/N A74027) for detailed program descriptions and operating instructions.

**HARD.SYS** - Hard disk device driver. Allows addition of one or more hard disk drives or partitions, on one or more SCSI hard disk controllers.

**HEDIT.EXE** - Hard disk partition editor. Allows partitioning of a hard disk drive up to four PC-DOS drive letters, 32 megabytes or smaller.

**HFORMAT.EXE** - Formats hard disk drives.

**HPARK.EXE** - Hard disk park. Used to position hard disk heads over a defined safety zone, to reduce likelihood of head "crash" on powerdown or during transit.

**SCSICOMP.EXE** - SCSI device block compare utility. (For technical users.)

**SCSICOPY.EXE** - SCSI device block copy utility. (For technical users.)

**SCSITOOL.EXE** - SCSI device exerciser and debugger. (For technical users.)

**TBACKUP.EXE** - Hard disk streaming tape backup. (Available soon.)

## CHAPTER 3

### SOFTWARE INSTALLATION

#### 3.1 INTRODUCTION

Assuming you have successfully booted PC-DOS as described in your board or system manual, you will probably want to take advantage of the flexibility designed into the Little Board/186 ROM-BIOS and Utilities software to create a customized PC-DOS based system. You can tailor your software configuration to a wide variety of hardware configurations, as outlined in Chapter 2. This Chapter provides a discussion of some common configuration options, including examples of typical installations.

#### 3.2 TWO SPECIAL FILES

Before reading about the various options available for customizing your system, you should understand the use of two special PC-DOS files: CONFIG.SYS and AUTOEXEC.BAT. These are discussed in detail in the PC-DOS reference manual, but will now be discussed briefly as an introduction to this chapter.

##### 3.2.1 CONFIG.SYS

"CONFIG.SYS" is a text file which you must create. It is used to instruct the operating system to install one or more "drivers" or system extensions. Each driver is a program which will be loaded by PC-DOS from floppy or hard disk. For each driver program that you want included in your system's operating system configuration, you simply include its name in a text file called "CONFIG.SYS" which is on your boot drive (floppy or hard disk). Here is a simple example of a CONFIG.SYS file:

```
DEVICE = DRIVER.SYS
```

This is a single text line, which can be entered using the EDLIN editor, using a word processing editor (in non-document mode), or using the COPY command, like this:

```
A>COPY CON CONFIG.SYS<RETURN>  
DEVICE = DRIVER.SYS<CTRL-Z><RETURN>  
A>__
```

When you want to include more than one driver in your system environment, you simply enter more lines beginning with "DEVICE =" to this file. In a few cases, drivers work with each other and must be installed in a specific order. When this is the case, the description of the driver (Chapter 4) specifies what other drivers are required, and what order must be used in the CONFIG.SYS file.

It is not enough to simply create the CONFIG.SYS file. You must now reboot your system to have PC-DOS actually install DRIVER.SYS as part of your system's operating environment. Then, the function provided by that driver will be available as a system resource.

### 3.2.2 AUTOEXEC.BAT

The second special file is called "AUTOEXEC.BAT." This is a file of commands that you would like to have your system run automatically on powerup or reset. For example, you could have your system display the directory of the boot drive using the following line as the contents of AUTOEXEC.BAT:

```
DIR
```

In this chapter, you will see that there are a number of AMPRO utilities which you might wish to include in your AUTOEXEC.BAT file. You do not necessarily need an AUTOEXEC.BAT file at all.

You create the AUTOEXEC.BAT file in the same manner as the CONFIG.SYS file, described above.

### 3.3 TERMINAL

Nearly any RS232C ASCII terminal can be used with your system. The initial terminal characteristics for first time booting must be set as described in your board or system manual (9600 baud, 8 data bits, etc.).

Once your system is initially booted, you can configure a custom system boot diskette for alternate console parameters, and add keyboard and video display appropriate to the ASCII terminal you are using, by means of several AMPRO drivers. As the various terminal options are explained, a cumulative installation example will be generated.

#### 3.3.1 Baud Rate

SETCON.SYS allows you to use a terminal which operates at a choice of baud rates, from 110 baud to 38.4K baud. In addition, SETCON.SYS permits: 7 or 8 bit data format; 1 or 2 stop bits; odd, even, or no parity; and the option of automatic hardware handshaking (RTS/CTS). Refer to the instructions for using SETCON.SYS in Chapter 4, and to the section on Hardware Handshaking, below. For example,

```
DEVICE = SETCON.SYS B38400 D8 S1 PN
```

Would initialize the terminal port of your system for operation with a terminal set for 38.4K baud, 8 data bits, 1 stop bit, no parity, and no hardware handshaking.

#### 3.3.2 Display Control Codes

PC software often uses ROM-BIOS functions to position the cursor, clear the screen, etc. The AMPRO TERM.SYS driver provides mapping of the PC ROM-BIOS functions into the control codes required by your specific terminal. TERM.SYS is supplied with a set of terminal characteristics files, of which one is used to customize your system for your terminal. Refer to Chapter 4 for details on

the use of TERM.SYS. Assuming your terminal is a WYSE WY-50, which is represented by the WY50.TRM terminal characteristics file, your CONFIG.SYS file might now contain a second line, as follows:

```
DEVICE = SETCON.SYS B38400 D8 S1 PN
DEVICE = TERM.SYS T\WY50.TRM P1 X0
```

### 3.3.3 PC Function Key Simulation

The AMPROKEY.SYS driver allows you to simulate the standard PC's function keys from any terminal's keyboard. This is generally desired when using TERM.SYS, since many programs request PC function key inputs. As indicated in Chapter 4, adding the AMPROKEY driver requires an additional CONFIG.SYS line:

```
DEVICE = SETCON.SYS B38400 D8 S1 PN
DEVICE = TERM.SYS T\WY50.TRM P1 X0
DEVICE = AMPROKEY.SYS
```

The WYSE WY-50 terminal in this example has function keys on it; it would be nice to use them directly. In the case of the WY-50, its function are programmable, so follow the instructions in the terminal's installation manual to program the keys to generate the keystroke sequences listed in the description of AMPROKEY.SYS in Chapter 4.

For terminals that do not have programmable function keys, you will need to use the keystrokes required by AMPROKEY. An additional keyboard driver, FUNKEY.SYS, converts several terminals' function keys directly into the keystrokes required by AMPROKEY, reducing the PC function keys back to single keystrokes. Refer to the instructions for using the FUNKEY driver in Chapter 4. As mentioned above, the WY-50 terminal used in this example has programmable function keys, so it does not require FUNKEY. If it did, you would add an additional CONFIG.SYS line to the growing list:

```
DEVICE = SETCON.SYS B38400 D8 S1 PN
DEVICE = FUNKEY.SYS
DEVICE = TERM.SYS T\WY50.TRM P1 X0
DEVICE = AMPROKEY.SYS
```

Notice that in this case, the driver was added before TERM.SYS and AMPROKEY.SYS. This is an example of an order-dependent driver.

### 3.3.4 Hardware Handshaking

Automatic hardware handshaking is often used to prevent a computer from sending data to a terminal or printer faster than the device can process it. It is usually used with serial printers at 1200 baud data rates or faster, and often required by terminals at data rates in excess of 9600 baud.

Each of the two serial ports on the Little Board/186 provides an option of automatic hardware handshaking. If required, this feature must be enabled by an application program or special utility, since it is disabled on by the ROM-BIOS on powerup or reset. The AMPRO SETCON.SYS driver and SETCOM1.COM utility support this option for the computer's two serial ports (see Chapter 4).

Here is a brief technical discussion of how the automatic handshaking option works:

When the automatic hardware handshaking feature is enabled, the serial controller stops sending data to the external serial device whenever the port's HANDSHAKE IN signal is inactive. To make use of this feature, the port's HANDSHAKE IN signal must be connected to an appropriate handshake output signal from the external serial device.

In addition, if the external device attempts to transmit data to the respective port faster than the computer's serial port can read it, that port's HANDSHAKE OUT signal will be brought to an inactive level. For this to be useful, the port's HANDSHAKE OUT signal must be connected to an appropriate handshake input signal on the external serial device.

When the automatic hardware handshaking feature is disabled for a port, the port's HANDSHAKE IN signal is ignored, and it's HANDSHAKE OUT signal is left permanently active.

Regardless of whether the automatic hardware handshaking feature is enabled, software can utilize the ROM-BIOS COM1 port status call to sense the state of the HANDSHAKE IN status signal on Serial Port B. (Refer to the Little Board/186 Technical Manual for details.) Whether automatic hardware handshaking is -- or is not -- required, depends on the specific software application. On the other hand, in some cases you may need to tie Serial Port B's HANDSHAKE OUT and HANDSHAKE IN signals to each other due to the monitoring of handshake signal status by software.

In the system installation example based on the WYSE WY-50 terminal, no hardware handshaking for the terminal is required.

### 3.4 PRINTERS

The system can be used with both Centronics type parallel printers, and with RS232C ASCII serial printers.

#### 3.4.1 Using a Parallel Printer

No special configuration is required for using the system with a Centronics type parallel printer. Simply reference the parallel printer as the DOS "LPT1" device. Most application software uses LPT1 as the default printer port, so printing to the parallel printer interface is automatic.

For example, the following PC-DOS commands will result in printing to the parallel printer:

```
A><u>COPY CONFIG.SYS LPT1</u><RETURN>    ...prints contents of CONFIG.SYS
A><u>DIR >LPT1</u><RETURN>              ...prints the directory
```

In addition, the "Printer Echo" function can be used to cause all characters written to the console to also be written to the LPT1 device. The Printer Echo function is enabled with <CTRL-P>. Once enabled, Printer Echo is disabled when another <CTRL-P> is used.

### 3.4.2 Using a Serial Printer

Serial Port B is supported as the PC-DOS "COM1" device. You can copy files to the serial printer port as follows:

```
A><u>COPY FILENAME.EXT COM1</u><RETURN>
```

You can use the PC-DOS MODE utility to reassign COM1 as the default printer port, as follows:

```
A><u>MODE LPT1=COM1</u><RETURN>
```

If you wish to have this done automatically on reset or powerup, place the above command line in a text file called AUTOEXEC.BAT on your system boot diskette. The command lines contained in the AUTOEXEC.BAT file are always executed automatically on system startup (reset or powerup). Later, you can restore the default printer port to the parallel interface again, with:

```
A><u>MODE LPT1</u><RETURN>
```

Before using Serial Port B, the port must be initialized. This can either be done with the PC-DOS MODE utility, or with the AMPRO SETCOM1 utility. SETCOM1 (see Chapter 4) allows the option of automatic hardware handshaking (RTS/CTS), while MODE does not. Here is an example of using SETCOM1.COM to initialize Serial Port B for use with a 1200 serial printer that requires hardware handshaking:

```
A><u>SETCOM1 B1200 D8 S1 PN H</u><RETURN>
```

The following two lines might be included in your system's AUTOEXEC.BAT file, to automatically initialize your system to use a 1200 baud serial printer as the printer device, on powerup or reset:

```
MODE LPT1=COM1  
SETCOM1 B1200 D8 S1 PN H
```

If you include the "H" parameter in the SETCOM1 command line, be sure the handshake line is connected properly to the printer, as described in your board or system documentation, or your system may hang up when you attempt to print. Refer to the technical notes on above hardware handshaking (Section 3.3.4), and to your board or system documentation for additional details on the use of this option.

### 3.5 MODEMS

In addition to its use as a serial printer port, Serial Port B can also be used as a modem interface, accessed as the PC-DOS "COM1" device. Serial Port B data characteristics are initialized in the manner described above for serial printer use (i.e. using MODE or the AMPRO SETCOM1 utility). In most cases, automatic hardware handshaking is not desirable when using modems, as the software generally "polls" for modem status.

You may wish to keep two batch files on your system disk, one called MODEM.BAT and one called PRINTER.BAT, to change between the serial data parameters (baud rate, handshaking, etc.) required for printer and modem functions. Each of these files would contain the appropriate line of text to invoke the MODE or SETCOM1 utilities, setting the parameters as required for your peripherals.

Most communications programs perform direct serial port I/O, and must be customized before use. The AMPRO LBCOMM.EXE program (Chapter 4), included with the Little Board/186 Utilities, is a full-featured communications program developed specifically for the Little Board/186. LBCOMM offers remote system access and both XMODEM and ASCII file transfer protocols.

In addition, the following two high quality communications programs are available preconfigured for the Little Board/186:

#### **MEX-PC**

NightOwl Software, Inc.  
Rt. 1, Box 7  
Fort Atkinson, WI 53538  
Phone: (800) 648-3695  
(414) 563-4013

#### **MICROLINK II**

by Wordcraft  
Digital Marketing Corp.  
Phone: (415) 947-1000

MEX-PC is a sophisticated "Modem Executive" program which allows you to create systems based on the Little Board/186 which can perform communications tasks completely automatically. It supports six upload/download protocols, including XMODEM, KERMIT, and Compuserve, and offers a powerful script programming facility.

### 3.6 UNIQUE FLOPPY CONFIGURATIONS

When your system initially boots from a standard PC-DOS system diskette, the floppy configuration is set for one to four 40-track (48 tpi) drives, depending on the setting of jumpers at position J7 (refer to your board or system manual).

### 3.6.1 Single-Floppy Configurations

One handy feature of PC-DOS is its built-in support for single-drive systems. If you jumper the Little Board/186 drive quantity jumpers (J7) to "one," the operating system will automatically assign drive letters "A" and "B" to the single floppy drive. With this configuration, you can copy files between two diskettes as though your system has two drives; PC-DOS will prompt you to change diskettes when needed.

### 3.6.2 Using DRIVER.SYS

The PC-DOS 3.2 DRIVER.SYS device driver allows a number of unique floppy customizations, including:

- 720K byte (80-track) mini- and micro-floppy support
- File copying between 2 diskettes on any drive
- Allowing additional 40-track drives beyond the setting of your system's drive quantity jumpers (J7)

When DRIVER.SYS is used to alter the system parameters of a floppy drive, an additional PC-DOS drive letter is assigned by PC-DOS to that drive. The letter assigned is the next available one, and depends on the setting of the computer board's drive quantity jumpers (at J7), and on whether other mass storage device drivers (e.g. HARD.SYS, CPMDRVR.SYS, and VDISK.SYS) are also in use.

An undocumented PC-DOS function (DRIVPARM), discussed in the next section, is often preferable for 80-track floppy support. Here are two examples of using DRIVER.SYS with 80-track drives:

**Example #1:** Adding an 80-track drive. Assuming that your computer board's drive quantity jumpers (J7) are set for 1 floppy drive, that your system has a 40-track drive A, and that there is not a hard disk installed, the following CONFIG.SYS text line will define a second floppy drive as an 80-track drive:

```
DEVICE = DRIVER.SYS /D:1
```

In this case, PC-DOS will assign drive letter C to the 80-track drive, since the first drive will have two drive letters (A and B) due to the drive quantity jumper setting of 1, as discussed above.

**Example #2:** A system with only 80-track drives. Start with a system configuration consisting of a 40-track boot drive and an additional 80-track drive supported by DRIVER.SYS as described in Example #1. Then use the DOS FORMAT utility with the /S option to format a diskette in the 80-track drive and transfer the operating system to it. This 80-track diskette will be your new 80-track boot diskette. Transfer all desired files to the new diskette, including a CONFIG.SYS file containing the lines:

```
DEVICE = DRIVER.SYS /D:0  
DEVICE = DRIVER.SYS /D:1
```

Now change drive A to an 80-track drive, jumper the computer board's drive quantity jumpers to "2" (see board or system manual regarding J7 jumpering), and boot from the new 80-track boot diskette. Due to the use of DRIVER.SYS,

PC-DOS recognizes two drive letter designations for each drive: unit 0 is "A" and "C"; and unit 1 is "B" and "D". Be sure to call the drives "C" and "D" to have proper 80-track (720K) functionality, including the use of FORMAT, DISKCOPY, DISKCOMP, etc.

As you can see DRIVER.SYS can do funny things with drive letters! If you use DRIVER.SYS, be sure to read the appropriate material in the PC-DOS reference manual.

### 3.6.3 Using the "DRIVPARM" Option

PC-DOS version 3.2 has an undocumented function which provides 80-track floppy support without DRIVER.SYS. This is accomplished by adding a line to your system's CONFIG.SYS file as follows:

```
DRIVPARM = /D:n /F:2
```

where "n" equals 0, 1, 2, or 3 to represent the corresponding floppy disk drive selects. As you can see, this is similar in function to the use of the DRIVER.SYS device driver. It has two important differences, however:

- (1) Defining 80-track drives using the DRIVPARM command does not add additional drive letters, and results in the drive being accessed via its "normal" drive letter (i.e. you can have 80-track drives A and B).
- (2) The board's drive quantity jumpers (J7) must be set to allow the number of drives being used.

In short, the DRIVPARM function provides more straight forward 80-track support than does DRIVER.SYS. However, since it is not documented in the IBM PC-DOS 3.2 reference manuals, there is no guarantee that this function will be supported by future revisions of IBM's PC-DOS operating system, so use it at your own risk!

r

Here is an example of the use of the DRIVPARM function in a system with only 80-track drives:

**Example:** A system with 80-track drives "A" and "B": Build a system having a 40-track drive A and an additional 80-track drive (B), jumpering the drive quantity jumpers (J7) on the Little Board/186 to a quantity of "2" (as described in the board's technical manual). Place the following command in the system's CONFIG.SYS file:

```
DRIVPARM = /D:1 /F:2
```

Reboot the system based on this CONFIG.SYS setup. Then use the PC-DOS FORMAT utility with the /S option to format and transfer PC-DOS to a diskette in the 80-track drive:

```
A>>FORMAT B: /S<RETURN>
```

This 80-track diskette will be your new 80-track boot diskette. Transfer all desired files to the new diskette, including a CONFIG.SYS file containing the following two DRIVPARM lines:

```
DRIVPARM = /D:0 /F:2  
DRIVPARM = /D:1 /F:2
```

Now change drive A to an 80-track drive, and boot from the new 80-track boot diskette. Due to the use of the DRIVPARM function, PC-DOS recognizes the system's two 80-track drives as "A" and "B" -- just as with a normal 40-track drive system. You now have a fully 80-track system! You can use all system commands and utilities in their normal manner, referencing the drives as PC-DOS drive letters A and B. You might want to add a 40-track drive "C" so that you can read and write normal 360K PC-DOS floppies as well. (Don't forget to reprogram the board's drive quantity jumpers to 3, if you do.)

### 3.6.4 Custom Floppy Parameters

The default floppy drive step rate used by the Little Board/186 ROM-BIOS is 6 mS, which is the standard step rate required by 40-track (360K byte) floppy disk drives. Since most 80-track mini and micro floppy drives can step at 3 mS, an AMPRO utility (SETFDC.COM) has been provided to allow you to customize the floppy step rate on a drive-by-drive basis. Refer to the description and instructions for SETFDC.COM in Chapter 4.

#### NOTE

Use of SETFDC's 2 or 3 mS step rate option requires that your computer contain a 1772 (rather than a 1770) floppy controller, and that your ROM-BIOS version is 3.1 or later (verify with CHKROM). Upgrades are available from AMPRO if required.

**Example:** A system with two 80-track drives (units 0 and 1). Include the following two command lines in your AUTOEXEC.BAT file, to specify a 3 mS step rate for floppy drive units 0 and 1 (first and second).

```
SETFDC /D0 S3  
SETFDC /D1 S3
```

### 3.7 HARD DISK DRIVES

Fixed and removable hard disk drives and controllers with SCSI interfaces may be easily added to your system. The Little Board/186 ROM-BIOS and AMPRO SCSI Utilities support drives of up to 128 megabytes on one or more SCSI disk controllers. Software utilities and drivers for SCSI device formatting, drive partitioning, tape backup, and other SCSI functions are provided in the AMPRO SCSI Utilities. Refer to the AMPRO SCSI Utilities User's Manual (P/N A74027), for additional information on the use of SCSI devices with the Little Board/186.

### 3.8 PC-DOS RAM DISK

PC-DOS Version 3.x provides a useful virtual disk device driver, called

VDISK.SYS, which allows you to configure one or more RAM disk drives based on memory on the Little Board/186. Activation of VDISK requires the addition of one or more lines referencing VDISK.SYS in the CONFIG.SYS file on your boot drive. Follow the instructions given in the PC-DOS reference manual.

### 3.9 CP/M FILE ACCESS

Another installation option available through the Little Board/186 Utilities software is the ability to access CP/M files on diskettes formatted a number of different 8-bit CP/M computers (including AMPRO), using the first or second floppy drives. This requires the use of the following additional command in your CONFIG.SYS file:

```
DEVICE = CPMDRVR.SYS
```

Refer to Chapter 4 for further information on the CP/M driver, and on the SETCPM.COM utility which is used to select a particular CP/M format.

### 3.10 EXAMPLE CONFIG.SYS

To summarize the driver installation information presented in this chapter, here is a complex system example. The hardware configuration represented is:

- 40-track floppy units 0 and 1 (1st and 2nd floppy drive)
- 80-track floppy unit 2 (3rd floppy drive)
- Drive quantity jumpers set for "3" (J7)
- Wyse WY-50 terminal, set for 38.4K baud
- CP/M file access on floppy units 0 and 1

Assuming you have done everything described in this chapter, you might end up with a CONFIG.SYS file that looks something like this:

```
DEVICE = SETCON.SYS B38400 D8 S1 PN
DEVICE = FUNKEY.SYS
DEVICE = AMPROKEY.SYS
DEVICE = TERM.SYS T\WY50.TRM P1 X0
DEVICE = VDISK.SYS 160 512 64
DEVICE = CPMDRVR.SYS
DRIVPARM = /D:2 /F:2
```

Here is the function of each CONFIG.SYS line:

(1) DEVICE = SETCON.SYS B38400 D8 S1 PN

This sets the terminal port baud rate to 38.4K baud.

(2) DEVICE = FUNKEY.SYS

This installs the preprocessor driver for function keys. This line is not really appropriate to the WY-50 terminal, since that terminal has programmable function keys.

(3) DEVICE = AMPROKEY.SYS

This installs the PC function key driver. Be sure to program the WY-50's function keys as required by AMPROKEY.SYS (see Chapter 4).

(4) DEVICE = TERM.SYS T\WY50.TRM P1 X0

This installs the terminal display driver for the WY-50 terminal, with 1 video page and no XON/XOFF handshaking.

(5) DEVICE = VDISK.SYS 160 512 64

This installs a 160K byte RAM disk.

(6) DEVICE = CPMDRVR.SYS

This allows CP/M diskette access on the first and second floppy drives.

(7) DRIVPARM = /D:2 /F:2

This informs PC-DOS that drive "C" is an 80-track (720K) type drive.

After booting your system with the above CONFIG.SYS file, your system drive letters will be arranged as follows:

A: 1st floppy drive (360K byte)  
B: 2nd floppy drive (360K byte)  
C: 3rd floppy drive (720K byte)  
D: 160K byte RAM disk  
E: First floppy drive, CP/M assignment  
F: Second floppy drive, CP/M assignment

Be sure to also include appropriate command lines in your BOOT diskette's AUTOEXEC.BAT file to:

- Set the step rate for the 80-track drive (SETFDC.COM)
- Select a desired format for the CP/M drives (SETCPM.COM)

## CHAPTER 4

### AMPRO DEVICE DRIVERS AND UTILITIES

#### 4.1 INTRODUCTION

This chapter contains detailed information on each of the driver and utility programs supplied on the AMPRO Little Board/186 Utilities software diskette. Each program's description explains what the program does and how it is used. Program descriptions are in alphabetical order, so this chapter can serve as a handy reference.

Each program is identified by a version and revision level. When the program is run, its version number (and a revision level) generally appear in a sign-on message such as the following:

```
        AMPRO Hard Disk Format Utility
        Copyright (C) 1986 AMPRO Computers, Inc.
        Version 2.1
```

In this case the program is version 2, revision 1. Versions of a program which have the same "version" number operate in the same manner. When a change is made to a program which necessitates a new description, its version number is changed, indicating that the old description is no longer accurate.

#### 4.2 PROGRAM DESCRIPTIONS

The following pages contain the program descriptions of the Little Board/186 Utilities software programs, alphabetically arranged.

#### NOTE

Read the contents of the ".DOC" files on your Little Board/186 Utilities diskette (filenames of the form "PROGRAM.DOC"). They contain information on recent program revisions, enhancements, or additions relative to the published user's manual.

Additional programs and features, offered on an on-going basis, will be available to you through software updates, at a nominal charge. You are encouraged to use the AMPRO customer support electronic bulletin board system which provides a 24-hour software update service, answers to user questions, public domain PC-DOS software, etc.:

**AMPRO Customer Support BBS — (415) 962-9023 — 300/1200 baud**

## AMPROCLK.EXE

(Version 1)

### Description

AMPROCLK.EXE supports both the AMPRO Expansion/186 and the AMPRO SCSI/IOP battery backed real time clock hardware options. The utility is a dual-purpose program, allowing you to:

- Set the PC-DOS system date and time values from the hardware values
- Set the hardware values from the PC-DOS system date and time values

### Operation

To use the AMPROCLK utility, enter the program's name from the PC-DOS prompt, followed by one or more options. For example:

```
A>AMPROCLK /(options)<RETURN>
```

If no options are included, a help screen such as the following is displayed:

```
A>AMPROCLK<RETURN>
```

```
AMPRO clock set utility.  Version 1.1  
Copyright (C) 1986 AMPRO Computers, Inc.  ALL RIGHTS RESERVED.
```

```
Usage:  AMPROCLK [/E] [/In] [/D]
```

```
/E      Select Expansion board clock  
/In     Select IOP clock (n = SCSI ID: 0 - 7)  
/D      Set selected clock from DOS clock
```

#### Examples:

```
AMPROCLK /E      Sets DOS clock from Expansion board clock  
AMPROCLK /I2     Sets DOS clock from IOP clock at SCSI ID 2  
AMPROCLK /E /D   Sets Expansion board clock from DOS clock  
AMPROCLK /I2 /D  Sets IOP clock at SCSI ID 2 from DOS clock
```

Before using the "D" option, be sure to first set the PC-DOS date and time using the standard PC-DOS commands (DATE, TIME). Thereafter, include an appropriate AMPROCLK command in your AUTOEXEC.BAT, to have the PC-DOS system clock automatically set to the battery backed hardware values on system powerup.

## AMPROKEY.SYS

(Version 1)

### Description

AMPROKEY.SYS is a device driver for the Little Board/186 which allows a serial ASCII keyboard connected to the console port (Serial Port A) to simulate the function keys of an IBM PC keyboard. Once AMPROKEY has been loaded, you type a specific keyboard sequence to produce the same result as would be obtained by pressing the desired function key on an IBM PC keyboard.

AMPROKEY is especially useful when combined with the AMPRO TERM.SYS video display driver. In addition, many terminals now have programmable function keys which can be programmed to provide the required keyboard sequences, thereby allowing single-key simulation of the IBM function keys.

### Operation

To use the AMPROKEY keyboard device driver, create a text file called CONFIG.SYS on your system boot diskette, containing at least one line, of the following form:

```
DEVICE = AMPROKEY.SYS
```

Be sure to reboot your system from a drive with the new CONFIG.SYS file containing the above line. AMPROKEY will indicate its presence by a message on your screen, when it loads. Once AMPROKEY is loaded, you can generate the IBM function keys from the key sequences shown in the table on the following page.

The required key sequences consist of a lead-in character followed by a second character indicating the desired function key. In all cases the lead-in character is <CTRL-^>, which is formed by pressing the <CTRL> key simultaneously with the ^ key. For example, to generate an F3 input, press the <CTRL> and ^ keys at the same time, then release them, then press the 3 key. On most terminals you will need to press the SHIFT along with the <CTRL> and ^ keys, because the ^ key is often the number 6 key, shifted. On many terminals, the <HOME> key generates a <CTRL-^>, so pressing <HOME>, then 3, would have the same result. With these terminals, when you actually need a "home" function, simply press <HOME> twice.

Naturally, the process can be greatly simplified if your terminal has programmable function keys! In the case of the WY-50 terminal, there are function keys which can be easily programmed to generate the appropriate sequences for F1-F10 and SHIFT F1-F10. Refer to your terminal's documentation for information on how to do this.

The key sequences used to simulate the various PC function keys are given in the table on the following page. Note that an additional AMPRO driver, FUNKEY.SYS, may allow your terminals' function keys to directly simulate the PC function keys. (Refer to the information on FUNKEY.SYS.)

## AMPROKEY Key Sequences

PC Function Key -----	Key Sequence Required -----	
F1	<CTRL-^>	1
F2	<CTRL-^>	2
F3	<CTRL-^>	3
F4	<CTRL-^>	4
F5	<CTRL-^>	5
F6	<CTRL-^>	6
F7	<CTRL-^>	7
F8	<CTRL-^>	8
F9	<CTRL-^>	9
F10	<CTRL-^>	0
Shift-F1	<CTRL-^>	!
Shift-F2	<CTRL-^>	@
Shift-F3	<CTRL-^>	#
Shift-F4	<CTRL-^>	\$
Shift-F5	<CTRL-^>	%
Shift-F6	<CTRL-^>	^
Shift-F7	<CTRL-^>	&
Shift-F8	<CTRL-^>	*
Shift-F9	<CTRL-^>	(
Shift-F10	<CTRL-^>	)
Home	<CTRL-^>	W (or w or <CTRL-w>)
Up Arrow	<CTRL-^>	E (or e or <CTRL-e>)
PgUp	<CTRL-^>	R (or r or <CTRL-r>)
Left Arrow	<CTRL-^>	S (or s or <CTRL-s>)
Right Arrow	<CTRL-^>	D (or d or <CTRL-d>)
End	<CTRL-^>	Z (or z or <CTRL-z>)
Down Arrow	<CTRL-^>	X (or x or <CTRL-x>)
PgDn	<CTRL-^>	C (or c or <CTRL-c>)

## CHKROM.EXE

(Version 1.1)

### Description

The Little Board/186 CHKROM utility is provided as a convenient means to verify the contents of your system's ROM-BIOS EPROM's. This can be helpful in cases of system difficulty or software updates.

### Operation

CHKROM is run by typing the following command at the PC-DOS prompt:

A><u>CHKROM</u><RETURN>

The program will then perform a checksum calculation on the ROM-BIOS EPROM's, and will display the checksums along with the version number of the ROM-BIOS.

## CPMDRVR.SYS

(Version 1)

### Description

The AMPRO CPMDRVR.SYS device driver allows your system to directly access files on diskettes which have been written by a variety of CP/M computers, including those manufactured by AMPRO, Kaypro, Morrow, Osborne, and others.

The desired CP/M format must be selected under control of the AMPRO SETCPM.COM utility program, described elsewhere in this chapter. The driver creates two CP/M access drive letters, corresponding to the first and second floppy drive units.

NOTE: Use of this utility requires that ROM-BIOS version 3.1 or later be installed. This can be verified using the CHKROM utility.

### Installation

To install CPMDRVR.SYS, add the following line to the CONFIG.SYS file on the drive from which your system boots:

```
DEVICE = CPMDRVR.SYS
```

### Operation

When the system boots with these lines included in the CONFIG.SYS file, the next two available PC-DOS disk drive letters are assigned by the system to floppy units 0 and 1 as CP/M access drive letters.

As with other PC-DOS disk device drivers, the two CP/M access drives will occupy two sequential drive letters, beginning with the next available drive letter beyond those defined by the computer board's drive quantity jumper settings. There is one exception: when the drive quantity jumpers are set to "1," PC-DOS assigns two drive letters to the first floppy drive. In this case, the single drive is accessed as both A and B, and additional drives begin with drive letter C. Also, the Little Board/186 ROM-BIOS (version 2 or later) recognizes the first hard disk drive automatically — without the use of HARD.SYS -- once it has been properly formatted, resulting in an additional occupied drive letter.

Example: Two floppy drives and a bootable hard disk; floppy drive quantity jumpers (J7) on the computer board set to "2." Using the two CONFIG.SYS lines given above, the system drive letters following boot will be:

- A: First floppy drive
- B: Second floppy drive
- C: Hard disk drive
- D: First floppy drive, CP/M assignment
- E: Second floppy drive, CP/M assignment

#### NOTE

Before you access the CP/M drives you must select a CP/M computer format with the AMPRO SETCPM.COM utility, described elsewhere in this chapter).

Once the driver is installed and a CP/M format is selected, you can use the CP/M diskettes as though they are PC-DOS diskettes, providing you access them under the specially assigned drive letters. This includes copying to and from the CP/M diskettes and use of PC-DOS commands and application programs.

When you want to access normal PC-DOS diskettes, simply use the ordinary floppy drive letters. In this manner, you can transfer files freely between PC-DOS (floppy or hard disk drives) and CP/M floppy diskettes.

## FUNKEY.SYS

(Version 1)

Written by Paul Bartholomew and released to AMPRO for distribution with the Little Board/186 Utilities Software.

### Description

FUNKEY.SYS allows the function keys of certain terminals to directly simulate the PC's function keys. As distributed, FUNKEY.SYS supports just one terminal: the Televideo TV925 terminal; however, source to the driver is included on the PC-DOS Support Software diskette (in the file FUNKEY.ASM) so you can modify FUNKEY.SYS for use with any terminal's function keys.

FUNKEY.SYS is used along with the AMPRO AMPROKEY.SYS keyboard device driver. FUNKEY translates the characters generated by terminal's function keys into the codes required by AMPROKEY. For example, when you press F1 on a Televideo 925 terminal, it transmits the following to the computer: "<CTRL-A> @ <CR>". In this case, FUNKEY intercepts these codes and translates them into "<CTRL-^> 1" -- which is the code sequence expected by AMPROKEY for <F1>.

FUNKEY can tell the difference between the <CTRL-A> at the beginning of a function key sequence and a <CTRL-A> that you would type. It does this by timing how quickly the next character is received from the terminal. So you can still use the <CTRL-A> key with programs like WordStar.

### NOTE

If your terminal has programmable function keys, you do not need to use FUNKEY.SYS. Instead, program the terminal's function keys according to the instructions in the terminal's user's manual.

### Operation

To install FUNKEY, add the following line to the CONFIG.SYS file on the drive from which your system boots, **before** the lines for AMPROKEY.SYS and TERM.SYS:

```
DEVICE = FUNKEY.SYS
```

Example:

```
DEVICE = SETCON.SYS B9600 D8 S1 PN
DEVICE = FUNKEY.SYS
DEVICE = AMPROKEY.SYS
DEVICE = TERM.SYS T\TV9XX.TRM P1 X0
```

## KT7KEY.SYS

(Version 1)

Written by Paul Bartholomew and released to AMPRO for distribution with the Little Board/186 Utilities.

### Description

KT7KEY.SYS is equivalent to the combination of the AMPROKEY.SYS and FUNKEY.SYS keyboard drivers, for the Kimtron KT7/PC terminal. It allows you to use the KT7/PC terminal in its "IBM Mode" to fully emulate the keyboard of a standard PC, including all function keys, print screen, CTRL-SCROLL\_LOCK (CTRL-BREAK), ALT + KEYPAD values, CTRL-ALT-DEL, etc.

KT7KEY.SYS is meant to be used in combination with the Kimtron KT7/PC display driver, KT7TERM.SYS (described elsewhere in this chapter).

### Operation

To install this driver, add a line to the CONFIG.SYS file on the drive from which your system boots **after** the line for KT7TERM.SYS:

```
DEVICE = KT7TERM.SYS T\KT7PC.TRM P1 X0  
DEVICE = KT7KEY.SYS
```

Be sure to delete lines for TERM.SYS, AMPROKEY.SYS, or FUNKEY.SYS from the CONFIG.SYS file, as they conflict with KT7KEY.SYS and KT7TERM.SYS.

After you have modified CONFIG.SYS, go into the "Setup Mode" of the Kimtron terminal using the combination keystroke: <ALT-CAPS\_LOCK>, and turn the IBM Mode ON. (Refer to your Kimtron terminal manual for instructions on the use of the Setup Mode.) Then exit Setup Mode and reset your computer, booting from the disk with the new CONFIG.SYS file. You should now have (almost) full PC keyboard compatibility.

## KT7TERM.SYS

(Version 3)

Written by Paul Bartholomew and released to AMPRO for distribution with the Little Board/186 Utilities.

### Description

KT7TERM.SYS is a special version of the Little Board/186 TERM.SYS terminal display driver, optimized for use with the Kimtron KT7/PC terminal. Like TERM.SYS, KT7TERM.SYS allows the use of some software which uses the IBM PC ROM-BIOS function calls.

The terminal control file which is used with KT7TERM.SYS is KT7.TRM. In addition, KT7TERM.SYS is intended to be used along with KT7KEY.SYS which provides Kimtron KT7/PC keyboard support. Refer to the description of KT7KEY.SYS elsewhere in this chapter.

Refer to the documentation on TERM.SYS elsewhere in this chapter for additional information on the ROM-BIOS functions provided by this driver, and on use of this driver along with the PC-DOS ANSI.SYS driver.

### Installation

KT7TERM.SYS is used in the same manner as TERM.SYS. Simply include the following two lines in the CONFIG.SYS file on your system boot drive:

```
DEVICE = KT7TERM.SYS T\KT7PC.TRM P1 X0  
DEVICE = KT7TERM.SYS
```

The first of these two lines loads the Kimtron display driver (KT7TERM.SYS) and terminal control file (KT7PC.TRM), while the second line adds the Kimtron keyboard driver. Be sure to delete lines for TERM.SYS, AMPROKEY.SYS, or FUNKEY.SYS from the CONFIG.SYS file, as they conflict with KT7TERM.SYS and KT7TERM.SYS.

After you have modified CONFIG.SYS, go into the "Setup Mode" of the Kimtron terminal using the combination keystroke: <ALT-CAPS LOCK>, and turn the IBM Mode ON. (Refer to your Kimtron terminal manual for instructions on the use of the Setup Mode.) Then exit Setup Mode and reset your computer, booting from the disk with the new CONFIG.SYS file. You should now have (almost) full PC keyboard compatibility.

You can add the option of software handshaking by using "X1" instead of "X0" in the first of the two CONFIG.SYS lines given above. Be sure to use the SETCON.SYS driver (described elsewhere in this chapter) if you wish to use a terminal baud rate other than 9600 baud, or if you want to enable hardware handshaking.

## LBCOMM.EXE

(Version 1.3)

Written by Jerry Haigwood and released to AMPRO for distribution with the Little Board/186 Utilities Software.

### Description

LBCOMM is a communications program written for the AMPRO Little Board/186 single board computer. It is one of the first programs written especially for this computer and takes advantage of the power and speed of the Intel 80186 processor on the Little Board/186.

LBCOMM has several features that make it very powerful but also easy to use. Some of these features are:

- Interrupt driven communications port
- Interrupt driven timers
- Buffered I/O
- XMODEM and XMODEM/CRC file transfer protocols
- X-ON/X-OFF ASCII file transfer protocol
- Run any PC-DOS command from within LBCOMM
- Run any other program from within LBCOMM
- Exit but leave LBCOMM resident
- Fast operation
- On line help screens
- Easily reconfigurable
- Automatic dialing
- Programmable function keys

These are described in detail in the following pages.

### Operation

LBCOMM requires other files for its operation. These are FUNCTION.KEY, MODEM.CFG, PHONE.LIB, and COMMAND.COM. These files need to be located in the same directory as LBCOMM.EXE. Although LBCOMM does not directly use the SETCOM1.COM utility, it is recommended that SETCOM1 also be placed in the same directory as LBCOMM.

If you have a hard disk, creating a subdirectory and copying these files into it will keep the root directory from becoming cluttered. If you are using a floppy only system, creating a new diskette with the above mentioned files on it is preferred. These files are all discussed in detail in later sections of this document.

Before running LBCOMM, it is necessary to set the required communications parameters using the AMPRO SETCOM1.EXE utility (described elsewhere).

LBCOMM can be executed two different ways. The first method involves simply entering LBCOMM at the DOS prompt. The second method involves entering LBCOMM

plus an additional command. Examples are:

Method 1:    A><u>LBCOMM</u><RETURN>           (load LBCOMM)

Method 2:    A><u>LBCOMM AD A</u><RETURN>       (load LBCOMM and auto-dial the  
  number associated with letter A  
  in the phone library)

Either method causes LBCOMM to attempt to load three configuration files: FUNCTION.KEY, MODEM.CFG, and PHONE.LIB. If any of these files is missing, LBCOMM will display an error message but will continue to load the others. However, some functions will not be available.

After loading the three configuration files, and if you started LBCOMM using Method 1, LBCOMM will ask if you want to initialize your modem. If you are connected to a modem and wish to initialize it, answer with a "y" or "Y". Any other answer will cause the initialization sequence to be skipped. (Note: the initialization parameters are located in MODEM.CFG.) Next, LBCOMM displays a signon message and enters the command mode, displaying LBCOMM's command mode prompt:

```
COMMAND>> _
```

If you started LBCOMM using Method 2, LBCOMM will execute the indicated command without pausing or displaying any signon message. Valid commands are discussed below.

You may wish to use a PC-DOS batch file to initialize your modem port baud rate and then start LBCOMM. For example, the file "MODEM.BAT" might contain the following text lines:

```
ECHO OFF
SETCOM1 B1200 D8 S1 PN
ECHO -----
ECHO Port B set to 1200 baud.
ECHO -----
LBCOMM
SETCOM1 B9600 D8 S1 PN H
ECHO -----
ECHO Port B restored to 9600 baud with handshaking.
ECHO -----
```

This short batch file first initializes Serial Port B to 1200 baud, 1 stop bit, 8 data bits, and no parity, then runs LBCOMM, and finally restores Serial Port B to 9600 baud after you exit from LBCOMM.

## LBCOMM Modes

While running, LBCOMM can be in one of several modes.

**Command Mode.** You can tell LBCOMM is in Command Mode, from the

```
COMMAND>> _
```

prompt on the bottom line of the screen display. In Command Mode, LBCOMM responds to a number of two- or three-letter commands, followed by a <RETURN>. This is similar to PC-DOS's "A>" command prompt. The Command Mode Commands are described below. Command Mode help is obtained from the ?<RETURN> command.

**Remote Modes.** LBCOMM has three Remote Modes. These are used for remote connection, and allow you to communicate with an external device connected to the Little Board/186 Communications Port (Serial Port B). The three Remote Modes are similar in that they all are used for remote device access, but they differ in whether, and how, they echo input or output characters. Three Command Mode commands — TM, LM, and EM — are used to enter one of the three Remote Modes from the Command Mode prompt.

In all three of these Remote Modes, the Command Mode prompt and Command Mode commands are not available. Instead, a set of Remote Mode commands may be used to control data capture, etc. The Remote Mode commands are discussed below.

## COMMAND MODE COMMANDS

LBCOMM has a set of commands which can be executed from the Command Mode prompt. Some variation of these commands is also permitted. The syntax of the commands and variations are discussed in detail below. The following page contains a summary of the Command Mode commands. Detailed descriptions of each of these commands follow:

**COMMAND: AD**

**SYNTAX:** AD<RETURN>  
AD <a-z,0-9><RETURN>

**FUNCTION:** Auto-dial a number from the phone library. In order to use the AD function, PHONE.LIB and MODEM.CFG must have been present when LBCOMM was loaded and, of course, your modem must support auto-dialing. AD followed by a <RETURN>, will cause the phone number library to be displayed. You are given a chance to select a number to be dialed by entering a single letter or numeral. The phone number associated with the selected letter or numeral will be dialed. AD followed by a space then a letter or numeral terminated by a carriage return will cause the number associated with the letter or numeral to be dialed immediately -- the phone library will not be shown. After dialing the number you will be placed in either Terminal, Local, or Echo Mode -- whichever mode you were in last. On first entry to LBCOMM, Terminal Mode is entered following auto-dialing.

## Command Mode Command Summary

Command	Syntax	Function
AD	AD<RETURN>	Display phone library. Auto-dial number from phone library.
	AD <A-Z,0-9><RETURN>	Auto-dial number indicated
DL	DL FILENAME.TXT<RETURN>	Download an ASCII text file from remote system.
DM	DM<RETURN>	Disconnect the modem.
EM	EM<RETURN>	Enter Echo Mode.
EX	EX<RETURN>	Exit LBCOMM to DOS.
HE	HE<RETURN>	Display full screen help.
LM	LM<RETURN>	Enter Local Echo Mode.
MS	MS <DOS Command><RETURN>	Run MS-DOS Command.
	MS <Program><RETURN>	Run program and return to LBCOMM.
	MS COMMAND.COM<RETURN>	Load another shell. Type EXIT from DOS to return to LBCOMM.
SD	SD<RETURN>	Set delays used in UL command.
TB	TB<RETURN>	Toggle bell on/off.
TM	TM<RETURN>	Enter Terminal Mode.
UL	UL FILENAME.TXT<RETURN>	Upload ASCII text file to remote system.
XR	XR FILENAME.TYP<RETURN>	Receive file from remote system using XMODEM protocol.
XS	XS FILENAME.TYP<RETURN>	Send file to remote system using XMODEM protocol.
?	?<RETURN>	Show command summary.

**COMMAND:** DL

**SYNTAX:** DL FILENAME.TXT<RETURN>

**FUNCTION:** Download an ASCII file from a host or remote system. Before using the DL command, you must setup the remote or host system for a file transfer. Most host systems have a "TYPE" command. Use the type or similar command to display the file you wish to download. However, DO NOT enter the <RETURN>. The DL function will automatically issue a <RETURN> character before it initiates the file transfer. The following example of a transfer on a VAX using the VMS operating system assumes you are in Terminal Mode.

**EXAMPLE:** \$ type vaxfile.txt;1<CTRL-\\>x (Note: No <RETURN> is pressed!  
" <CTRL-\\>x" returns you to  
command mode.)

**COMMAND>>** dl b:vaxfile.txt<RETURN> (Note: the DL command supplies  
the required <RETURN> to the  
remote system.)

Creating b:vaxfile.txt

Downloading, press any key to close file

LBCOMM issues a <RETURN> causing the VAX to recognize the "TYPE" command and thus send the file. When the download is complete, you may press any key to close the file and return to command mode. If the file you are downloading is already on your disk, LBCOMM will ask if you want to erase the file or add to it. Answer with an "e" or "a".

**COMMAND:** DM

**SYNTAX:** DM<RETURN>

**FUNCTION:** Disconnect the modem. DM causes the de-initialization sequence (line 4 of the MODEM.CFG file) to be sent to the modem causing your modem to hangup.

**COMMAND:** EM

**SYNTAX:** EM<RETURN>

**FUNCTION:** Enter Echo Mode. This is one of the three Remote Modes. Echo Mode is very similar to Terminal Mode with one major exception — in Echo Mode, all characters received over the communications port (Serial Port B) are both displayed on your screen and echoed back out the communications port. Characters entered at the keyboard are also echoed to the screen. Echo Mode is also known as "Computer" or "Host Mode," because most hosts operate in Echo Mode. The difference between Echo Mode and Local Echo Mode is that in the former, your system echoes characters received **from** the remote system back to the remote system, while in the latter, your system echoes what you type to your own screen.

#### NOTE

During a communication session, only one system at a time may be in Echo Mode — either your system or the one you are communicating with. The other system MUST be in Terminal Mode. If both systems are in Echo Mode, the first character will cause an endless loop to begin. Use Echo Mode cautiously!

For a complete description of all the functions available under Echo Mode, see the section on Remote Mode Commands, below. (See also TM.)

#### COMMAND: EX

SYNTAX: EX<RETURN>

FUNCTION: Exit LBCOMM. If MODEM.CFG was present when LBCOMM was loaded, you will be asked if you wish to disconnect the modem from the phone line. A "y" or "Y" answer will cause the modem to be disconnected from the phone line; any other answer will cause the disconnect sequence to be skipped. If you plan to re-enter LBCOMM, do not answer "y" or "Y". If your communications is completed, it is best to use a "y" or "Y", to insure that your modem has been disconnected from the phone line.

#### COMMAND: HE

SYNTAX: HE<RETURN>

FUNCTION: Display full screen help. Several screens of Command Mode help are presented. After each screen is displayed, press any key to continue program or display operation.

#### COMMAND: LM

SYNTAX: LM<RETURN>

FUNCTION: Enter Local Echo Mode. This is one of the three Remote Modes. Local Echo Mode is very similar to Terminal Mode with one major exception -- in Local Echo Mode, all characters that you type are echoed to your screen, as well as passed out the communications port (Serial Port B). Local Echo Mode might therefore be called "Half Duplex Mode." You can use this mode when you communicate with remote systems which do not echo the characters you send, so that you can see what you type. The difference between Local Echo Mode and Echo Mode is that in the former, your system echoes what you type to your own screen, while in the latter, your system echoes characters received from the remote system back to the remote system.

For a complete description of all the functions available under Local Echo Mode, see the section on Remote Mode Commands, below. (See also TM.)

**COMMAND: MS**

**SYNTAX:** MS <DOS command><RETURN>  
MS <program><RETURN>  
MS COMMAND.COM<RETURN>

**FUNCTION:** The MS command allows you to run any program or DOS command from inside of LBCOMM. Use of the MS command requires that COMMAND.COM be present in the default directory. When the MS command is invoked, LBCOMM acts as a "shell." A typical use of the MS command is to change the COM1 port parameters by running the SETCOM1.COM utility from inside of LBCOMM. Other uses for the MS command are copying files, displaying a directory, etc.

**EXAMPLES:** COMMAND>> MS dir<RETURN> (display directory)  
COMMAND>> MS ws<RETURN> (run WordStar)

Another use of the MS function is loading a second copy of COMMAND.COM:

COMMAND>> MS command.com<RETURN>  
A>\_\_

By running COMMAND.COM, you temporarily leave the LBCOMM environment, and exit to a DOS command environment. However, LBCOMM stays resident and continues to capture any received characters, up to a maximum of 4,096 characters before the buffer "wraps around." To re-enter LBCOMM, type

A>EXIT<RETURN>  
COMMAND>> \_\_

You have left the DOS environment and re-entered the LBCOMM Command Mode.

**COMMAND: SD**

**SYNTAX:** SD<RETURN>

**FUNCTION:** Set up the delays used during an ASCII file upload (UL command), and the duration of the break key which can be transmitted in Remote Mode. The SD command allows you to slow down LBCOMM for remote systems which cannot keep up. Delays up to 99 milliseconds for each character, and 999 milliseconds for each line, are allowed. The default delay times are 0. Note: LBCOMM responds to X-ON/X-OFF protocol from the remote system, so delays are unnecessary in most applications.

**COMMAND: TB**

**SYNTAX:** TB<RETURN>

**FUNCTION:** Toggles the ASCII bell character ON/OFF. The default mode is bell ON. IF you type TB<RETURN> at the command line, any bell characters (ASCII 7) will not be sent to your terminal in either Terminal, Local Echo, or Echo Mode. Command line mistakes will still cause the terminal to sound.

**COMMAND: TM**

**SYNTAX: TM<RETURN>**

**FUNCTION:** Enter Terminal Mode. This is one of the three Remote Modes. This Remote Mode might also be called "Full Duplex Mode" — no character input or output echoing is performed in this mode. Terminal Mode allows the Little Board/186 to emulate a smart terminal. All characters received at the COM1 port are sent to your terminal. All characters typed by you are sent out COM1. With a smart terminal attached to the Serial Port A, and Serial Port B connected to a host either through a modem or directly, the Little Board/186 will act as a "smart" buffer. No received characters are filtered except maybe the ASCII bell (see command TB above). Any valid ASCII sequence is transmitted by LBCOMM except <CTRL-\. The <CTRL-\. character is used as a lead-in character for commands to be executed while in the Terminal Mode.

The commands available in Terminal Mode are described in the section on Remote Mode Commands, below. (See also EM and LM.)

**COMMAND: UL**

**SYNTAX: UL FILENAME.TXT<RETURN>**

**FUNCTION:** Upload an ASCII file to a remote system. LBCOMM searches for the file named in the command line, and sends it out over the communications port (Serial Port B). You may abort the upload at any time by pressing any key. See the example below of uploading an ASCII file to a VAX host using the VMS operating system.

When you use the UL command, you will be asked if you want to send both carriage return and line feed characters, or only carriage return characters. Any answer other than a 'Y' will cause only <CR>'s to be sent. That is, a <CR><LF> sequence will be converted into a <CR> only. This is useful in uploading ASCII files to systems which double space text whose lines terminate in a <CR><LF> sequence. You will need to experiment to see which format the remote system prefers.

During the ASCII upload process, the remote system can stop and start file transmission at any time during the upload by sending an X-OFF character (<CTRL-S>) to stop, and an X-ON character (<CTRL-Q>) to resume, transmission. LBCOMM will also insert character and line delays, if the optional SD command prior to the UL command.

EXAMPLE: \$ create newfile.txt<RETURN> (tell the VAX to create a file  
named newfile.txt)  
  
<CTRL-\>x (exit to Command Mode.)  
  
COMMAND>> ul newfile.txt<RETURN> (start the upload)  
  
uploading newfile.txt  
press any key to close file

When the upload has been completed, close the file on the VAX by typing a  
<CTRL-Z>.

COMMAND: XR

SYNTAX: XR FILENAME.TYP<RETURN>

FUNCTION: Receive file using xmodem or xmodem/crc protocol. Xmodem file transfers are virtually error free. LBCOMM is compatible with either xmodem or xmodem/crc protocol. The xmodem/crc protocol transfers file with a better than 99.9% confidence factor. Xmodem/crc protocol is used as the default. However, since some remote systems don't use xmodem/crc, LBCOMM will automatically attempt xmodem protocol when xmodem/crc protocol is not accepted by the remote system. A typical xmodem file transfer is shown below.

EXAMPLE: BO>xmodem s newfile.exe<RETURN> (tell the remote system to send a  
file using an xmodem protocol)

XMODEM v104  
File Open: 12 records (2K)  
Send time: 0 mins, 15 secs at 1200 bps  
To cancel: use CTRL-X numerous times

<CTRL-\>x (exit to Command Mode)

COMMAND>> xr newfile.exe<RETURN> (tell LBCOMM to receive file  
using an xmodem protocol)

Creating newfile.exe  
Strike any key to abort transfer  
Rec'd Block ## Errors 0  
EOT Received  
File transfer complete

COMMAND: XS

SYNTAX: XS FILENAME.TYP<RETURN>

FUNCTION: Send a file using an xmodem protocol. The XS is very similar to the XR function except XS sends files via the xmodem, or xmodem/crc, protocols. As in the XR command, LBCOMM will automatically switch from xmodem/crc to standard xmodem as required by the remote system. (See also XR.)

**COMMAND:** ?

**SYNTAX:** ?<RETURN>

**FUNCTION:** Display a Command Mode command summary. Typing a ?<RETURN> at the Command Mode prompt causes a brief summary of Command Mode commands to be displayed. After using LBCOMM for a while, this command summary will probably be the only help needed.

## **REMOTE MODE COMMANDS**

Just as there are many LBCOMM commands in the Command Mode, there are also a number of Remote Mode commands. However unlike the DOS-like command prompt in Command Mode, the Remote Modes (Terminal Mode, Echo Mode, and Local Echo Mode) neither show a prompt nor accept commands directly. Since it is desirable that the Remote Modes pass all keystrokes you make directly to the remote system, Remote Mode commands must be given more subtly. This is done by means of a "lead-in" character, the <CTRL-\\> keys (that is, press the <CTRL> key and the "\\" simultaneously). After you press <CTRL-\\>, LBCOMM looks for the next key you press as a command indicator. The choices are listed in the table of Remote Mode commands, below. If you forget these commands, Remote Mode command help is available by pressing <CTRL-\\>, then the ? key.

The commands which can be executed from the three Remote Modes -- Terminal Mode, Local Echo Mode, and Echo Mode -- are:

## Remote Mode Command Summary

<CTRL-\<> Lead-in sequence followed by one of the following command characters.

- 0-9 Function keys. Causes LBCOMM to send the character string associated with the number. These strings are defined by the file FUNCTION.KEY.
- B Transmits a "break" key while in Remote Mode. This allows logging on/off of some mini's and mainframes (VAX, etc.). The length of the break signal is controllable by using the "SD" command from the COMMAND>> line. The default time is 350 ms.
- H Hangup the modem. Hangup sequenced defined by MODEM.CFG.
- S Start capturing received characters into the buffer. If the capture file has not be opened, LBCOMM will prompt you for a file name. If the file exists, you will be given a choice to erase the current file or add to it. LBCOMM will automatically write the captured characters to the file and close the file at the end of the session, or whenever you use the <CTRL-\<>s function (below).  
  
Capturing of characters into the buffer can continue even after you exit LBCOMM if you use the Command Mode command: MS COMMAND<RETURN>. In this case, LBCOMM stays resident and continues to capture received characters into its 4096 character buffer.  
  
Note that the capture buffer wraps around when it overflows; new characters will over-write old ones.
- Q Stop capturing received characters into the capture buffer.
- W Write the contents of the current capture file to disk. Allows multiple capture files during a single communication session. The <CTRL-\<>w function will flush the capture buffer and close the capture file. To continue capturing, use <CTRL-\<>s again. You will be prompted for the new file name.
- X Exit the Remote Mode and return to Command Mode.
- ? Display Remote Mode command help. A brief summary of the Remote Mode commands and their functions will be displayed on the terminal.

Here are some examples of the use of Remote Mode Commands:

```
<CTRL-\<>x    ...exit to Command Mode
<CTRL-\<>s    ...start capture mode
<CTRL-\<>h    ...hangup the modem
<CTRL-\<>0    ...function key 0
```

## The FUNCTION.KEY File

FUNCTION.KEY is a text file created by the user. This file contains the definitions of the Remote Mode function key commands (0-9) and must be present in the same directory as LBCOMM.EXE.

The FUNCTION.KEY file must be very precisely constructed. The first line of the file contains the definition of function key 0, the second line defines function key 1, ..., and the 10th line of the file contains the definition for function key 9. The maximum length of each line is 80 characters.

FUNCTION.KEY can be created by any editor which works with standard ASCII text files. WordStar in the non-document mode (use "N" from the editing no file menu) will be acceptable. There are several special character sequences which may be used within a function key line that are interpreted as commands by LBCOMM. These are:

\r	interpreted as a <RETURN>
\n	interpreted as a line feed
^C	a ^ followed by a letter is interpreted as a control character
@	pause for 1 second

These commands can be used together with standard ASCII characters to create complex function key sequences. A typical FUNCTION.KEY file might look like the following:

```
Firstname\r
Lastname\r
password1\r
password2\r\n
First;Last;Password\r
DIR $UOAD\r
XMODEM S
XMODEM R
^C
@@+++@ATZ\r
```

In the above example, Firstname\r maps to function key 0. The \r will be interpreted as a <RETURN>. Invoking function key 0 (<CTRL-\\>0) would send "Firstname<RETURN>". Also in the above example, @@+++@ATZ\r is mapped to function key 9. The @@ symbols are interpreted as 1 second pauses each and the \r is interpreted as a <RETURN>. Invoking function key 9 (<CTRL-\\>9) would cause "<pause for 2 seconds>+++<pause for 1 second>ATZ<RETURN>" to be sent.

## The MODEM.CFG File

Like FUNCTION.KEY, MODEM.CFG is an ASCII text file. It is used to configure LBCOMM for almost any modem. It is up to the user to create this file. The same commands that apply to FUNCTION.KEY also apply to MODEM.CFG (i.e. \r,\n,^ and @). The maximum length of any line is 80 characters.

The first line of MODEM.CFG must contain the initialization sequence for your modem. The second line contains the prefix for auto-dialing. The third line contains the suffix for the auto-dial routine. If your modem does not auto-dial, leave the second and third lines blank. The last (fourth) line of MODEM.CFG contains the de-initialization sequence for your modem. A typical MODEM.CFG for a "Hayes compatible" modem is as follows:

```
+++@ATZS0=0T\r
ATDT
\r
@+++@ATHO\r
```

The first line tells the Hayes modem to reset and to not auto-answer. The second line is the Hayes prefix for auto-dialing, "ATDT". The third line is the Hayes suffix for auto-dialing, a <RETURN>. The last line is the hangup command.

## The PHONE.LIB File

PHONE.LIB contains the phone number library. It is also an ASCII file that is created by the user. Like FUNCTION.KEY and MODEM.CFG, PHONE.LIB is also sensitive to character positions within the text file.

The first line maps to phone number "A", the 26th line maps to phone number "Z", the 27th line maps to phone number "0" and the 36th line maps to phone number "9". Unlike the previous two ASCII files, PHONE.LIB has a MAXIMUM line length of 32 characters. Any characters past the 32nd will confuse LBCOMM.

When auto-dialing, LBCOMM will send all numbers after the first letter in the selected line. Therefore, do not include any numbers in the name field! Also do not use any of the MODEM.CFG or FUNCTION.KEY commands (\r,\n,@ etc). To allow programmed pauses in dialing, commas (,) in the number sequence are passed on to the modem (used by Hayes compatible modems to generate a delay). No other non-numeric characters are sent to the modem. This is especially useful when dialing on phone systems which require you to dial a 9 for access to an outside line. A comma following the 9 will provide a short delay, allowing the outside line to connect before the remainder of the number is dialed.

Here is an example of a PHONE.LIB file:

```
A=Joe Smith..... 325-2238
B=Dale Thatcher..... 823-1527
:
:
Z=Harry Sloan..... 387-9249
0=AMPRO bbs.....(415) 962-9023
1=AMPRO-ONE.....(408) 258-8128
:
:
9=AMPRO (voice)...(415) 962-0230
```

Even if you don't need 36 phone numbers on line, it is recommended you still fill in the lines with periods or some other appropriate character, to present a pleasant looking phone number library.

## MONITOR.EXE

(Version 1)

### Description

MONITOR.EXE provides a demonstration of the AMPRO Monitor EPROM option, which is intended for use by engineers in specialized applications which do not require the use of PC-DOS. The Monitor includes commands for memory, I/O port, and CPU register access, as well as floppy disk read and write functions.

MONITOR.EXE is not intended to be used as a PC-DOS debugging program, but is supplied for demonstration purposes only. Contact AMPRO for information on the optional Monitor EPROM set.

### Operation

The EPROM Monitor demonstration program is run by typing the following command at the PC-DOS command prompt:

A>MONITOR<RETURN>

The EPROM Monitor is a menu-based program. The various EPROM Monitor functions are listed in either the main menu, or the sub-menus. An EPROM Monitor User's Manual may be purchased from AMPRO.

### CAUTION

Use MONITOR.EXE's functions with extreme caution, as they may destroy valuable disk data, etc.

## SETCOM1.COM

(Version 2)

### Description

The AMPRO SETCOM1 utility program is used to initialize your system's Serial Port B (the DOS "COM1" device) for a variety of baud rates between 110 and 57.6K baud, data characteristics, and optional hardware handshaking. SETCOM1 is similar in function and usage to the PC-DOS MODE utility. Many of the initialization options available with SETCOM1 are also available using MODE; however, enabling or disabling of the port's automatic hardware handshaking feature requires the use of SETCOM1, rather than MODE. Refer to the technical note below, for further information on the use of the hardware handshaking option.

### Operation

To initialize Serial Port B (the DOS "COM1" device), type the program's name, followed by appropriate parameters, from the PC-DOS command prompt. The general form of the command is:

A>SETCOM1 <parameter,parameter,...><RETURN>

The parameters are defined as follows:

Baud rate: "B" followed by the desired baud rate; choose from:  
57600, 38400, 19200, 9600, 4800, 2400, 1200, 600, 300,  
150, or 110

Data width: "D" followed by: 7 or 8 (bits per character)

Stop bits: "S" followed by: 1 or 2

Parity: "P" followed by: O, E, or N (odd, even, or none)  
NOTE: parity is not allowed with 8 bit data width.

Handshake: "H" if automatic hardware handshaking is desired

The order of parameters on the setup line does not matter. Letters may be in either upper or lower case.

NOTE: Use of 150, 19200, 38400, and 57600 baud rates requires that ROM-BIOS version 3.1 or later be installed. This can be verified using the CHKROM utility.

### Examples

A SETCOM1 command can be placed in your AUTOEXEC.BAT file for automatic system initialization. You can also create a number of other batch files for easy port B modification. One batch file might be used to set the port for 1200 baud modem use, with handshaking disabled; another might set the port for 9600

baud serial printer use with handshaking enabled. They could be called MODEM.BAT and PRINTER.BAT, respectively. Here are two examples:

**Example #1: Using a 9600 baud serial printer, with automatic hardware handshaking.** Many serial printers can operate at 9600 baud, using automatic hardware handshaking. You can use the following SETCOM1 command to set the COM1 port for such a printer:

```
A><u>SETCOM1 B9600 D8 S1 PN H</u><RETURN>
```

This results in 9600 baud, 8 data bits, 1 stop bit, no parity, and automatic hardware handshaking. Be sure that the printer's READY handshake output signal (CTS, or equivalent) is connected to Serial Port B's Handshake In input signal.

Assuming you wish to mainly use a serial printer for printed output, and that these parameters are the ones you need, a file called PRINTER.BAT -- or your AUTOEXEC.BAT file -- might contain the following two lines:

```
MODE LPT1:=COM1  
SETCOM1 B9600 D8 S1 PN H
```

If this file is called PRINTER.BAT, then you can easily initialize Serial Port B to these characteristics with the command:

```
A><u>PRINTER</u><RETURN>
```

**Example #2: Using a 1200 baud modem.** In this case, automatic hardware handshaking is generally not used. The following SETCOM1 command will initialize the COM1 port to 1200 baud, 8 data bits, 1 stop bit, no parity, and automatic hardware handshaking disabled:

```
A><u>SETCOM1 B1200 D8 S1 PN</u><RETURN>
```

Placing this command line in a file called MODEM.BAT allows easy preparation of Serial Port B for modem operation, with the command:

```
A><u>MODEM</u><RETURN>
```

#### **Technical Note: Hardware Handshaking**

When the handshaking feature is enabled, the system stops sending data to the external serial device whenever Serial Port B's HANDSHAKE IN signal is inactive. To make use of this feature, Serial Port B's input handshake signal must be connected to an appropriate handshake output signal from the external serial device. In addition, if the external device attempts to transmit data to Serial Port B faster than the CPU can read it, Serial Port B's HANDSHAKE OUT signal will be brought to an inactive level. For this to be useful, Serial Port B's output handshake signal must be connected to an appropriate handshake input signal on the external serial device.

When the handshaking feature is disabled, Serial Port B's HANDSHAKE IN signal

is ignored, and HANDSHAKE OUT is left permanently active.

#### NOTE

Regardless of whether the automatic hardware handshaking feature is enabled, software can utilize the ROM-BIOS COM1 port status call to sense the state of the HANDSHAKE IN status signal on Serial Port B. Therefore, automatic hardware handshaking may not be required, depending on the specific software application. On the other hand, in some cases you may need to tie Serial Port B's HANDSHAKE OUT and HANDSHAKE IN signals to each other.

Hardware handshaking is often required when using serial printers, but seldom used with modems.

Generally, modem communications software must be configured to use Serial Port B's HANDSHAKE IN signal as a Data Carrier Detect (DCD) input. ROM-BIOS calls can be used to transfer data and read the state of the HANDSHAKE IN status signal. "DCD" is not directly supported by the Little Board/186 ROM-BIOS, but must be sensed through the "CTS" status bit instead, which indicates the state of HANDSHAKE IN. Refer to the Little Board/186 Technical Manual for further details.

## SETCON.SYS

(Version 1)

### Description

The SETCON.SYS device driver, is used to initialize the serial data characteristics of the Little Board/186 console port, Serial Port A. This allows you to match the optimum characteristics of the console device to be used for keyboard and display. Without the use of the SETCON.SYS driver, the system defaults to the board's ROM-BIOS settings of 9600 baud, 8 data bits, 1 stop, no parity, and disabled hardware handshaking, on powerup or reset.

When SETCON.SYS is invoked from the CONFIG.SYS file on the system boot drive, it automatically initializes the system console interface to the values specified in the CONFIG.SYS line. SETCON.SYS allows use of terminals with any of the following baud rates: 110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400. In addition, SETCON.SYS permits: 7 or 8 bit data length; 1 or 2 stop bits; odd, even, or no parity; and the option of automatic hardware handshaking (RTS/CTS). Automatic hardware handshaking is often required for terminal baud rates of 19.2K and 38.4K baud.

When the automatic hardware handshaking feature is enabled, the system will stop sending data to the terminal when Serial Port A's HANDSHAKE IN signal is inactive. If required, this should be connected to the terminal's Clear to Send (CTS) output signal. In addition, if the terminal attempts to transmit data to Serial Port A faster than the Little Board/186 can handle it, the port's HANDSHAKE OUT signal will be brought to an inactive level. If desired, that signal can be connected to a terminal's Request to Send (RTS) input signal. When the automatic hardware handshaking feature is disabled, HANDSHAKE IN is ignored, and HANDSHAKE OUT is left permanently active.

### Operation

To use the SETCON.SYS console initialization device driver, create a text file called CONFIG.SYS on your system boot diskette, containing at least one line, of the following form:

```
DEVICE = SETCON.SYS <parameter,parameter,...>
```

Refer to the IBM PC-DOS reference documentation for further information on the use of the PC-DOS CONFIG.SYS function.

The parameters which follow "SETCON.SYS" determine the desired baud rate and other characteristics for the console (keyboard & display) port. An example is provided below. The parameter choices are:

Baud rate: "B" followed by the desired baud rate; choose from:  
38400, 19200, 9600, 4800, 2400, 1200, 600, 300, 110

Data width: "D" followed by: 7 or 8 (bits per character)

Stop bits: "S" followed by: 1 or 2

Parity: "P" followed by: O, E, or N (odd, even, or none)  
NOTE: parity is not allowed with 8 bit data width.

Handshake: "H" if automatic hardware handshaking is desired

The order of parameters on the setup line does not matter. Letters may be either upper or lower case.

The SETCON.SYS line should always be the first entry in your boot diskette's CONFIG.SYS file. This guarantees that the console serial port is initialized before PC-DOS attempts to send any characters to the display. Otherwise, "garbage" characters might be sent which could hang up the terminal, etc.

#### Example

The following CONFIG.SYS line is appropriate for a terminal set to 38.4K baud, 8 data bits, 1 stop bits, no parity, and automatic hardware handshaking:

```
DEVICE = SETCON.SYS B38400 D8 S1 PN H
```

Naturally, the terminal device must be set to match these characteristics before you attempt to boot from system disk whose CONFIG.SYS file contains this line.

## SETCPM.COM

### Description

The SETCPM.COM utility is used to select a CP/M format for file transfer under the control of the AMPRO CPMDRVR.SYS device driver (described elsewhere in this chapter).

### Operation

To use the program, simply type its name on the PC-DOS command line:

```
A>SETCPM<RETURN>
```

The program will display a menu of available CP/M formats. Press the letter of the format you wish to access. After you select the desired CP/M format, the program will exit to the PC-DOS command line.

You may prefer to use lines in your AUTOEXEC.BAT file to automatically select a particular CP/M format on system powerup. This is done by creating a tiny text file (SETCPM.TXT) containing a single letter -- the menu selection desired -- and then adding the following line to your AUTOEXEC.BAT file:

```
SETCPM <SETCPM.TXT
```

For example, if you want to pick menu selection "A", the file SETCPM.TXT consists of the letter "A", menu option "A" will be selected.

## SETFDC.COM

### Description

The SETFDC.COM utility is used to customize certain floppy disk drive parameters, including the floppy drive step rate, to values different than the board's ROM-BIOS defaults. At this time, only the four drive's step rate is under the control of SETFDC.COM. This is particularly valuable when 80-track mini (5-1/4") or micro (3-1/2") floppy drives are used, because they generally can step at 3 mS, rather than the ROM-BIOS default of 6 mS.

NOTE: Use of this utility requires that ROM-BIOS version 3.1 or later be installed. This can be verified using the CHKROM utility.

### Operation

To initialize a floppy drive's parameters, type the program's name, followed by appropriate parameters, from the PC-DOS command prompt. The general form of the command is:

```
A>SETFDC /Dd Ss<RETURN>
```

The parameters are defined as follows:

Drive number: "D" followed by 0, 1, 2, or 3

Step rate: "S" followed by 0, 1, 2, or 3 (see table)

The drive number code (D) corresponds to how the floppy drive unit number is jumpered on the drive.

The step rate code (S) depends on whether the system has a 1770 or 1772 floppy controller device installed, according to this table:

Step Rate Code	1770 Rate	1772 Rate
0	6 mS	6 mS
1	12 mS	12 mS
2	20 mS	2 mS
3	30 mS	3 mS

A separate command must be given to set or change the parameters for each floppy drive. Normally you would include a SETFDC command line in your AUTOEXEC.BAT file, if customized floppy parameters are required.

Example: A system with an 80-track microfloppy drive with 3 mS step rate, jumpered as the 3rd floppy unit. In this case, include the following line in the AUTOEXEC.BAT file on the system boot drive:

```
SETFDC /D2 S3
```

## TERM.SYS

(Version 3)

### Description

The purpose of the AMPRO TERM.SYS device driver is to allow the use of some software written to use the IBM PC ROM-BIOS video function calls. The function calls supported by TERM.SYS include cursor positioning, screen clearing, scrolling, and selection of video page. Calls to the supported ROM-BIOS functions are translated into control sequences for a choice of ASCII terminals, as defined by a terminal definition file. TERM.SYS does not support software which writes directly to Video RAM.

Refer to the "Technical Notes" section, below, for further details on the ROM-BIOS function calls supported by the current version of TERM.SYS.

### Installation

To use the AMPRO PC-DOS terminal device driver, create a text file called CONFIG.SYS on your system boot diskette, containing at least one line, of the following form:

```
DEVICE = TERM.SYS T\name.TRM Pn Xm
```

Refer to the IBM PC-DOS reference documentation for further information on the use of the PC-DOS CONFIG.SYS function.

These are the interpretations of the required parameters. Examples follow.

T\name.TRM	Terminal control filename. Must be DIRECTLY after the "T". Can include drive:path.
Pn	Number of screen RAM pages (1-4). If no "P" parameter is included, the default of 1 page is used.
Xm	XON/XOFF handshaking option (0 or 1). If X is followed by a 1, then this handshaking is used, otherwise not. Note that hardware handshaking is also available, through the use of the AMPRO SETCON.SYS driver. If no X parameter is included, the default of no handshaking is used.

At this time, the following terminals are supported, through the indicated terminal control files:

<b>File Name</b>	<b>Terminal Supported</b>
ADDSVP.TRM	Adds Viewpoint
VT101.TRM	DEC VT101
HAZ-ESC.TRM	Hazeltine 1500 (Escape mode)
HAZ-TILD.TRM	Hazeltine 1500 (Tilde mode)
HZ19.TRM	Heath/Zenith H19/Z19
TV9XX.TRM	Televideo 900 series
WY50.TRM	Wyse WY50

### **Examples**

Example #1: To use a Televideo 920 terminal with one page of screen RAM and no XON/XOFF handshaking:

```
DEVICE = TERM.SYS T\TV9XX.TRM
```

Example #2: A VT101 terminal with 2 pages of screen RAM and XON/OFF handshaking enabled. The terminal control file is in the UTILS directory:

```
DEVICE = TERM.SYS T\UTILS\VT101.TRM P2 X1
```

### **Using TERM.SYS with ANSI.SYS**

The IBM PC-DOS ANSI.SYS driver performs almost the opposite function to that of TERM.SYS. When the ANSI.SYS driver is installed (through a line in CONFIG.SYS), terminal control sequences intended for an "ANSI" compatible terminal are transformed into ROM-BIOS function calls compatible with an IBM PC video controller. TERM.SYS does the inverse: it transforms ROM-BIOS function calls into terminal control sequences.

When software requires the use of ANSI.SYS (on an IBM PC), use TERM.SYS in addition to ANSI.SYS. This will re-map the ANSI control sequences into sequences compatible with your specific terminal.

When you use ANSI.SYS in combination with TERM.SYS, be sure to observe the following order:

```
DEVICE = TERM.SYS (etc...)
DEVICE = ANSI.SYS
(etc...)
```

### **25 Line Screen Emulation**

When you are using the TERM.SYS driver with a 24 line terminal, you will be viewing 24 lines out of a 25 line "virtual PC screen." The top 24 lines of the virtual PC screen are what are normally displayed. Since many programs use the PC's 25th line as a status line, you might need to know what is on that line. To view the bottom 24 lines of the virtual PC screen, press the <CTRL-]> key

combination; that is: hold the CTRL key down while you press the ] key. Each time you press <CTRL-]>, the display will shift up or down by 1 line.

### Terminal Control File Customization

You can easily modify existing terminal control files (name.TRM), or create new ones. To assist you in doing this, a sample source file (TRM.ASM) has been included on your software diskette. Modify TRM.ASM as required, and then reassemble it using the file MAKETERM.BAT. A text editor, MASM, LINK, and EXE2BIN are the tools you will need.

### Technical Notes

TERM.SYS maintains four video display pages in memory, in a manner compatible with the IBM PC ROM-BIOS. Whenever a character is written by a program using the "Write Character at Cursor Position" functions, TERM.SYS saves a copy of the character sent to the terminal in the memory-resident screen data area. This allows TERM.SYS to support the ROM-BIOS "Read Character" function.

The memory-resident screen data also allows TERM.SYS to emulate the IBM PC's 25 line display on a 24 line terminal. Most ASCII terminal screens are limited to 24 addressable display lines, while the IBM PC screen display provides 25. Programs which use ROM-BIOS function calls for video display generally assume the availability of 25 display lines. To solve this problem, TERM.SYS allows you to view either the first 24 lines (0-23), or the last 24 lines (1-24), of the "current" video display page. The <CTRL-]> control character is used to toggle between the two display modes.

Here are the ROM-BIOS INT 10H functions supported by TERM.SYS:

```
AH = 0    SET MODE
AL = 0    40x25 BW (default on power-up)
AL = 2    80x25 BW
AL = 7    80x25 BW
```

NOTE: The video mode is saved, but the driver remains in 80x25 BW.

```
AH = 1    SET CURSOR TYPE
CH        Start line for cursor (0-31)
CL        End line for cursor (0-31)
```

NOTE: The cursor mode is saved, but the cursor remains the same.

```
AH = 2    SET CURSOR POSITION
BH        Page number (0-3)
DH        Row (top row is 0)
DL        Column (left column is 0)
```

```
AH = 3    READ CURSOR POSITION
BH        Page number (0-3)
DH        Row (top row is 0)
DL        Column (left column is 0)
CH        Start line for cursor (0-31)
```

CL End line for cursor (0-31)

AH = 4 READ LIGHT PEN POSITION  
Not implemented

AH = 5 SELECT ACTIVE DISPLAY PAGE  
AL New page value (0-3)

AH = 6 SCROLL ACTIVE PAGE UP

AH = 7 SCROLL ACTIVE PAGE DOWN  
AL Number of lines to scroll, 0 means clear entire window  
BH Attribute of new line(s)  
CH Upper row of scroll window  
CL Left column of scroll window  
DH Lower row of scroll window  
DL Right column of scroll window

AH = 8 READ ATTRIBUTE/CHARACTER AT CURSOR POSITION  
BH Page number (0-3)  
AL Character  
AH Attribute

AH = 9 WRITE ATTRIBUTE/CHARACTER AT CURRENT CURSOR POSITION  
BH Page number (0-3)  
AL Character  
BL Attribute  
CX Count of characters to write

AH = 10 WRITE CHARACTER AT CURRENT CURSOR POSITION  
BH Page number (0-3)  
AL Character  
CX Count of characters to write

AH = 11 SET COLOR PALETTE

AH = 12 WRITE DOT

AH = 13 READ DOT  
Not implemented

AH = 14 WRITE TTY  
AL Character

AH = 15 CURRENT VIDEO STATE  
AL Mode currently set (See AH=0 for explanation)  
AH Number of character columns on screen  
BH Current active display page

NOTE: This function always returns 80 (decimal) in AH

For all function calls:

CS, SS, DS, ES, BX, CX, DX preserved during call  
all other registers are destroyed