# USING MS-DOS ON THE GRID COMPASS

NOVEMBER 7, 1983

Model No. 21040

2535 Garcia Avenue
Mountain View, CA 94043
(415) 961-4800

Manual Name: Using MS-DOS on the GRiD Compass
Model Number: 21040
Issue date: November 7, 1983

The information in this document is subject to change without notice.

NEITHER GRiD SYSTEMS CORPORATION NOR THIS DOCUMENT MAKES ANY EXPRESSED OR IMPLIED WARRANTY, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, OR FITNESS FOR A PARTICULAR PURPOSE. GRiD Systems Corporation makes no representation as to the accuracy or adequacy of this document. GRiD Systems Corporation has no obligation to update or keep current the information contained in this document.

GRiD System Corporation's software products are copyrighted by and shall remain the property of GRiD Systems Corporation.

UNDER NO CIRCUMSTANCES WILL GRiD SYSTEMS CORPORATION BE LIABLE FOR ANY LOSS OR OTHER DAMAGES ARISING OUT OF THE USE OF THIS MANUAL.

GRiD TM and GRiD Compass TM are trademarks of GRiD Systems Corporation. WordStar TM is a trademark of MicroPro International, Smartmodem 1200 TM is a trademark of Hayes Microcomputer Products, Inc., and Project Scheduler TM is a trademark of SCITOR Corporation.

IBM is a registered trademark of International Business Machines, Inc. Microsoft TM, Multiplan TM, and MS TM are trademarks of Microsoft Corporation. Lotus 1-2-3 TM is a trademark of Lotus Development Corporation. dBASEII TM is a trademark of Ashton-Tate, Inc.

## TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION

CHAPTER 2. GETTING STARTED WITH MS-DOS ON THE GRiD COMPASS

CHAPTER 3. THE GRiD COMPASS KEYBOARD

CHAPTER 4. DEVICE CONFIGURATIONS

## LIST OF FIGURES

## LIST OF TABLES

# CHAPTER 1: INTRODUCTION

This introduction provides an explanation of the fundamental terms and concepts you will be encountering throughout the user's guide. We assume that you have read the GRiD Compass Owner's Guide and that you know how to set up your GRiD Compass and attach the peripheral devices. If you are an experienced user well-versed in the terminology associated with computer equipment, you may skip this chapter and continue on to Chapter 2.

## STORING INFORMATION IN FILES

Most people begin the business day by getting out a collection of papers that they call a "file." You put the file folder on your desk, open it, and start "processing." Working with the GRiD Compass is very similar. You turn on the computer, search through a directory, and ask for a special file. One difference between using the GRiD Compass and working with paper is that the GRiD Compass finds your file more quickly and then displays the information on a screen instead of on your desk.

The files are stored in a systematic way on a number of different GRiD storage devices. A description of each device follows.

## GRiD STORAGE DEVICES

A Device is to computing what a file cabinet is to an office -- an organized storage area for all kinds of data. GRiD Compass files are stored on four different kinds of storage devices, shown below.

Bubble Memory is an electronic device located inside the GRiD Compass. This is a very special type of memory. Unlike most electronic memories in computers, Bubble Memory retains all the information stored in it even when you turn off the computer. So, you can use it for long-term storage of information.

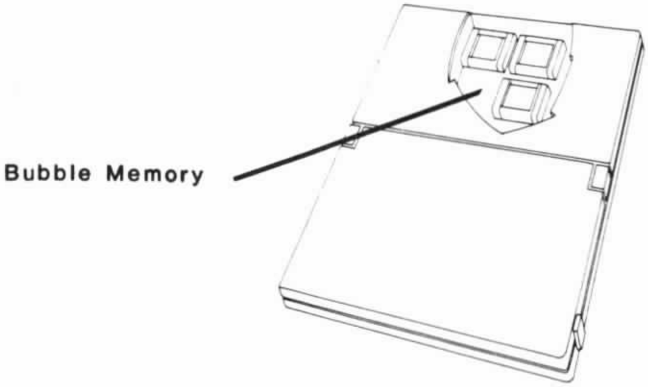


Figure 1-1. Bubble Memory

The Hard Disk Drive is located on the left side of a 2101 Disk System. The hard disk itself is permanently in place. It can store a lot of information (about 10 million characters) and move files or access file information very quickly. You'll use it when the information you need to store exceeds the capacity of Bubble Memory.

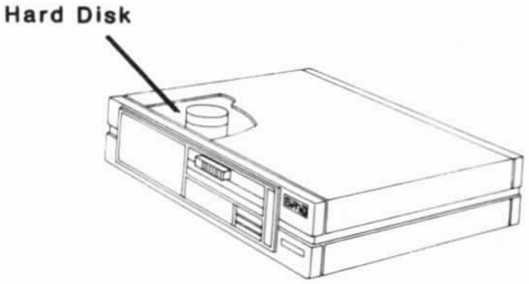


Figure 1-2. Hard Disk Drive

The Floppy Diskette Drive is located on the right side of the 2101 Disk System. The diskettes that you insert into and remove from the drive are known as floppies. Floppy diskette drives transfer data more slowly and hold less information than a hard disk drive. However, diskettes are handy for making information portable and for sharing data with co-workers. Each diskette has a maximum capacity of about 360 thousand characters.

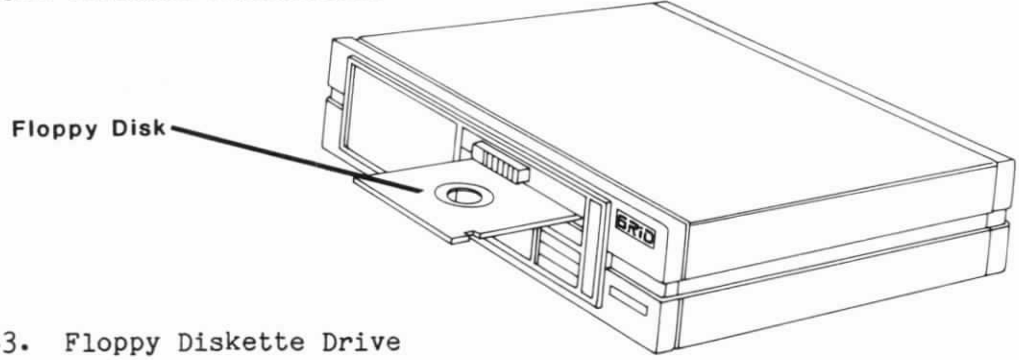


Figure 1-3. Floppy Diskette Drive

The Portable Floppy is the 2102 Portable Diskette drive. This device uses diskettes, just like the Floppy Diskette Drive. But, since it does not include a hard disk, it is much smaller and lighter in weight than the 2101 disk system.

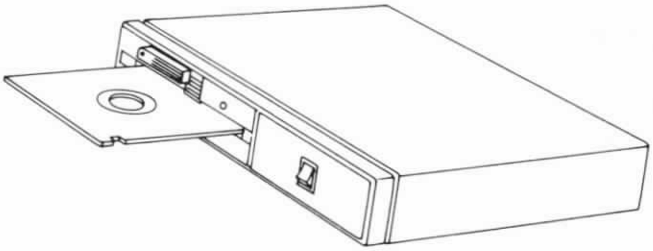


Figure 1-4. Portable Floppy Diskette Drive

## USING MAIN MEMORY

We've used the term "storage" to speak of devices that hold data permanently or on a long-term basis. When you ask for a file, the computer goes to a specific device and brings the file back to main memory. The following figure illustrates this.

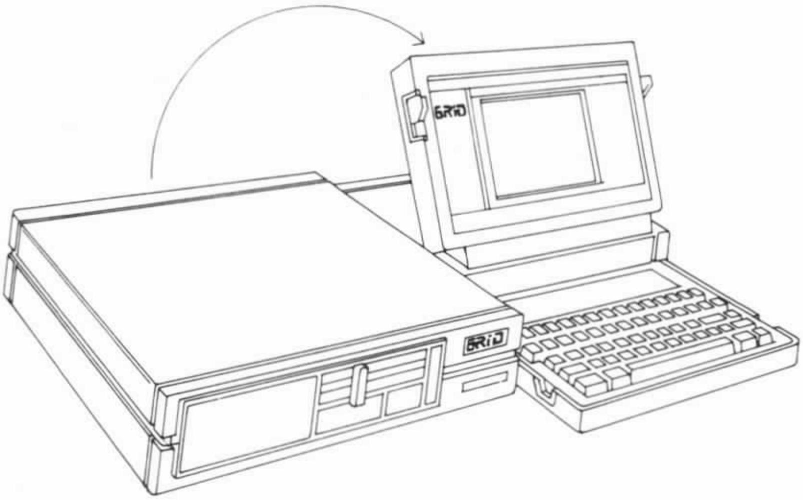


Figure 1-5. Moving Information from Storage to Main Memory

Main memory is also called working memory or RAM (Random Access Memory). When you're computing, main memory holds the program you're working with (such as WordStar) and the file or files you're working on (such as "MEMO.TXT"). You can move data in and out of main memory faster than you can move it to and from a GRiD storage device.

Unlike bubble memory, main memory is volatile. In other words, if you turn your GRiD Compass off or some other interruption of power occurs, everything in main memory is automatically reset. The most recent copy of the file in use remains on its storage device, however, and can be reloaded into main memory for further processing.

## THE OPERATING SYSTEM

The operating system provides a bridge between the physical components of the computer (the keyboard, disk drives, display screen) and the programs and files that you wish to work with. Files are put into categories, saved on storage devices or moved into main memory, and modified and sorted by the operating system. As the computer user, you need to learn how to fully utilize the power of the operating system. For example, you need to learn the rules for naming files, the commands to which the operating system will respond, the procedures for accessing storage devices, and techniques for running application programs.

The following chapter introduces the features of MS-DOS (Microsoft Disk Operating System) as customized for use on the GRiD Compass computer.

# CHAPTER 2. GETTING STARTED WITH MS-DOS ON THE GRiD COMPASS

The material within this manual supplements the Microsoft MS-DOS User's Guide. Special considerations and procedures that are specific to using MS-DOS on the GRiD Compass are described. For a more comprehensive description of MS-DOS, consult the Microsoft manual included in a separate section of your MS-DOS binder.

This chapter introduces the fundamental operations to start using MS-DOS, including booting from a floppy diskette, formatting a diskette, making backup copies, and running application programs. Each of these steps is covered in greater detail in subsequent chapters.

## MICROSOFT DOCUMENTATION ASSUMPTIONS

Microsoft's MS-DOS User's Guide assumes that MS-DOS will be run on a system that is floppy-disk based with two floppy disk drives available. That manual does provide an appendix describing system operation when only one floppy disk drive is available. However, since the GRiD Compass may include built-in bubble memory, as well as a variety of different disk drives, device designations will differ from those described in Microsoft's manual. You should refer to Chapter 4 of this document for a discussion of device configurations and designations for your GRiD Compass.

## GETTING STARTED WITH MS-DOS

MS-DOS is supplied on a floppy diskette. Insert this diskette into your diskette drive and turn on the drive. You can now start up or "boot" MS-DOS directly from this diskette by holding down the "F" key on the GRiD Compass while turning on the computer. Shortly, the message "Booting MS-DOS" will be displayed. When MS-DOS has been loaded into memory, information similar to the following is displayed:

```
GRiD Compass MS-DOS BIOS version A
Copyright c 1983 GRiD Systems Corporation

MS-DOS version 2.00
Copyright 1981, 82, 83 Microsoft Corp.

Command v.2.02
Current date is Tue 1-01-1980
Enter new date:_
```

NOTE: Because the GRiD Compass has a built-in real-time clock, a special command (SETTIME) has been added to the standard repertoire of MS-DOS commands. Consult Chapter 6 for a description of this command.

Since SETTIME is used to set the internal clock, the MS-DOS prompts for "new date" and "new time" are no longer needed; skip over them by pressing the RETURN key twice. The MS-DOS system prompt will then be displayed:

```
A>
```

The ">" prompt signifies that MS-DOS is now ready to process commands. The letter "A" tells you which device is the current default device; in this case, it is the floppy disk drive.

When typing MS-DOS commands, they may be in upper case or lower case.

NOTE: See Chapter 4 for a discussion of device configurations and identification.

The first MS-DOS command you will probably want to execute is the DIRectory command which will display the contents of the diskette containing MS-DOS. After the system prompt "A>" appears, type "dir" and press RETURN. Information similar to the following is displayed:

```
 Volume in drive A has no label
 Directory of  A:\

COMMAND  COM     15480   2-08-83   7:50p
CONFIG   SYS        58   8-31-83   6:50p
MODEM    SYS      6073  10-13-83   4:16p
SERIAL   SYS      3384  10-13-83   4:16p
THECLOCK SYS       561  10-18-83  11:33a
CHKDSK   COM      6330   2-01-83   9:16a
DEBUG    COM     11764   2-01-83  10:13a
DEVICES  EXE      1094   5-16-83  10:30p
DISKCOPY COM      1419   9-20-83   1:49p
EDLIN    COM      4389   2-01-83   9:31a
EXE2BIN  EXE      1649   2-01-83   9:19a
FC       EXE      2553   2-01-83   9:36a
FDISK    EXE     29568  10-30-83   3:56p
FIND     EXE      5796   1-14-83   6:35p
FORMAT   COM     12949   8-30-83   6:53p
LINK     EXE     41856  10-06-83  12:51p
MORE     COM      4364   1-14-83   6:42p
PRINT    COM      3808   2-01-83  12:39p
RECOVER  COM      2277   2-01-83   2:22p
SETTIME  EXE      4065   5-31-83   6:02p
SORT     EXE      1216   2-08-83   7:04p
SYS      COM       850   2-01-83   2:26p
        22 File(s)    147456 bytes free
```

NOTE: The SETTIME command requires two system files: THECLOCK and CONFIG. MODEM and SERIAL must be present for the internal modem and the serial (RS-232) interface to operate.

Next, you should make a copy of the diskette containing MS-DOS so that you can store the original away. Before you can copy the MS-DOS diskette, however, you must "format" a new diskette so that it can receive the copy.

## FORMATTING A DISKETTE

Microsoft's MS-DOS manual and Chapter 5 of this manual contain thorough descriptions of the FORMAT command. For convenience, a brief description is included here to help you get started.

NOTE: This procedure assumes that you have already booted into MS-DOS from the floppy diskette.

▶TO FORMAT A FLOPPY DISKETTE:

1. Issue the FORMAT command (with the MS-DOS diskette still in the disk drive) by typing the following after the MS-DOS prompt "A>" is displayed:

```
format a:
```

Then press RETURN.

2. MS-DOS then issues the following message:

```
Insert new diskette for drive A:
and strike any key when ready.
```

3. Remove the MS-DOS diskette and insert a blank unformatted diskette in the drive. Press any key on the keyboard. The message:

```
Hard format in progress.
```

is displayed. Shortly, the following message will be displayed:

```
# of 720 sectors formatted.
```

The number (#) will initially be 1 and will be incremented as the formatting proceeds. You can thus see how far the operation has progressed and gauge the time needed until completion.

4. When the formatting is complete, a message similar to the following one is displayed:

```
          362496 bytes total disk space
          362496 bytes available on disk

     Format another (Y/N)?
```

5. Type Y if you wish to format another disk. Type N to end the FORMAT operation.

## MAKING A WORKING COPY OF THE MS-DOS DISKETTE

Now that you have a blank, MS-DOS formatted diskette, you can make a working copy of the original diskette on which MS-DOS was delivered to you. Before making your copy, ensure that the original MS-DOS diskette has a write-protect sticker affixed to it.

▶TO MAKE AN MS-DOS WORKING COPY:

1. With the original MS-DOS diskette in the disk drive, issue the DISKCOPY command by typing the following after the MS-DOS prompt "A>" is displayed:

```
diskcopy
```

Then press RETURN.

During the disk copy process, the original MS-DOS diskette is the source diskette. The blank, formatted diskette is the target.

2. MS-DOS then issues the following message:

```
Insert formatted target diskette into drive A:
Press any key when ready.
```

   Insert the target diskette (the newly formatted diskette) in the drive, and press any key on the keyboard.

   MS-DOS checks to make sure that the target diskette is a formatted diskette and then issues the following message:

```
Insert source diskette into drive A:
Press any key when ready.
```

3. Re-insert the source diskette (the original MS-DOS diskette) in the drive and press any key on the keyboard. The DISKCOPY program begins reading data from the source diskette. After a few moments, the following message is displayed:

```
Insert target diskette into drive A:
Press any key when ready.
```

4. Remove the MS-DOS diskette from the drive and insert the newly formatted, blank diskette. Then press any keyboard key. The DISKCOPY program now begins writing data to the target. During the diskcopy operation, you will be prompted one or more times to "Insert source diskette ..." or "Insert target diskette ..." since memory limitations require that the diskcopy be perfomed in several steps. Each time one of these prompts is displayed, remove the diskette currently in the drive, insert the requested diskette, and then press any key. When the entire diskette has been copied, the following message is displayed:

```
Copy complete
Copy another (Y/N)?
```

5. Type N to exit and return to the the MS-DOS system prompt.

## RUNNING MS-DOS APPLICATIONS

Once MS-DOS is running on the GRiD Compass, you can run an MS-DOS-based application, such as Multiplan, WordStar, Lotus 1-2-3, or dBASE II. Simply insert the floppy diskette containing the application into the disk drive and enter the application program's name in response to the MS-DOS system prompt. For example, if the Multiplan application program is named MP.EXE and is on the floppy diskette in drive A, you can run Multiplan by typing

```
mp
```

and pressing RETURN in response to the "A>" prompt.

NOTE: Some applications which run under MS-DOS can be used directly on the GRiD Compass. Other MS-DOS application software may require adjustments or modifications to run correctly. Chapter 7 describes software products that have been customized for the GRiD Compass. See Chapter 8 for a discussion of the complete application software installation process.

# CHAPTER 3. THE GRiD COMPASS KEYBOARD

Every manufacturer has a different means of arranging keys on a keyboard. Some provide additional function keys or special purpose keys. Some alter the placement of keys or change the designations on the key tops.

This chapter explains how the GRiD Compass keyboard functions within the MS-DOS environment. This chapter also describes how specific combinations of GRiD Compass keys can be used to simulate the keyboard functions of the IBM Personal Computer. Thus, application packages that were written for use with an IBM PC keyboard can be run on the GRiD Compass.

## KEYBOARD USE

The keys on the GRiD Compass keyboard are color-coded to suggest their functions. The alpha-numeric keys, marked with white characters, provide the traditional typewriter-style keyboard.

The keys marked with yellow characters (TAB, ESC, SHIFT, RETURN, BACKSPACE, and the arrow keys) control cursor movement about the display screen and character entry. In addition, ESC (Escape) frequently offers a means to exit or cancel a command at any given point.

The keys marked with red characters (CTRL and CODE) modify other keys when used in combination with them. CODE may also be used in combination with the SHIFT key to provide additional functions. These functions and other details relevant to keyboard use are discussed in this chapter. If you are familiar with keyboard use while in GRID/OS, you will discover that many of the same functions apply. Others, as noted in the text, are specific to use with MS-DOS .

NOTE: Throughout this manual, when the CODE or CTRL key is shown in conjunction with other keys, it indicates that both keys are held down at the same time. For example, to enter CODE-2, the user should first press the CODE key and, while holding it down, press the "2" key.

Figure 3-1 illustrates the GRiD Compass keyboard.

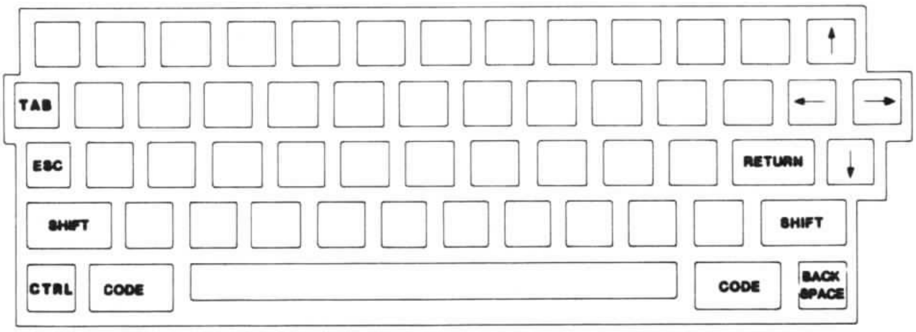


Figure 3-1. GRiD Compass Keyboard

## KEYBOARD SHIFT LOCK

The keyboard of the GRiD Compass can be shifted so that all of the alphabetic keys generate upper case characters. To activate this "Caps Lock" mode, simultaneously press the SHIFT and ESC keys. While in the Caps Lock mode, lower case letters can be generated by pressing the SHIFT key in conjunction with an alphabetic key. Pressing SHIFT-ESC a second time returns the keyboard to normal, lower-case operation.

## ADDITIONAL KEYBOARD CHARACTERS

In addition to the characters displayed on the GRiD Compass key tops, several more characters are available. These characters can be accessed by pressing the combinations of keys illustrated in Table 3-1.

Table 3-1. Additional Keyboard Characters

| GRiD Compass Keystrokes | Character |
|---|---|
| CODE-comma (,) | [ |
| CODE-period (.) | ] |
| CODE-SHIFT-comma (,) | { |
| CODE-SHIFT-period (.) | } |
| CODE-semicolon (;) | ~ |
| CODE-SHIFT-semicolon (;) | \| |
| CODE-apostrophe (') | ` |
| CODE-SHIFT-apostrophe (') | \ |

## KEYBOARD CODES

Every time a key or sequence of keys is depressed on the GRiD Compass keyboard, an ASCII character code (or sequence of character codes) is generated. Appendix D lists these keyboard codes as they are produced while running your computer under MS-DOS. Consult Table D-1 for this information.

## MS-DOS FUNCTION KEYS

The top row of keys on the GRiD Compass keyboard can be used in combination with the CODE key to provide a set of expanded functions when running MS-DOS.

Operation of the MS-DOS function keys (which can also be used within the line editor program known as EDLIN) are described in detail in Chapter 6 of the Microsoft MS-DOS User's Guide. The keystrokes used on the GRiD Compass for these functions are listed in Table 3-2.

Table 3-2. MS-DOS Editing and Function Keys

| MS-DOS Function Keys | GRiD Compass Keys | Description |
|---|---|---|
| <COPY1> | RightArrow | Copy one character from the template to the command line. |
| <COPYUP> | CODE-2 | Copy characters up to the character specified in the template and put these characters on the command line. |
| <COPYALL> | CODE-3 | Copy all remaining characters in the template to the command line. |
| <SKIP1> | CODE-(minus sign) | Skip over (do not copy) a character in the template. |
| <SKIPUP> | CODE-4 | Skip over (do not copy) the characters in the template up to the character specified. |
| <VOID> | CODE-BACKSPACE | Void the current input; leave the template unchanged. |
| <INSERT> | CODE-= | Enter/Exit insert mode. Alternate depressions turn the insert mode on/off. |
| <CTRL-Z> | CODE-6 | Put CTRL Z on command line. |
| <NEWLINE> | RETURN | Make the new line the new template. |

## COMPARING THE GRiD COMPASS AND IBM PERSONAL COMPUTER KEYBOARDS

With MS-DOS on the GRiD Compass, you can run programs also available on the IBM PC.

Table 3-3 lists various key combinations used with the IBM Personal Computer. The equivalent GRiD Compass key combinations are listed beside the IBM keystrokes.

NOTE: SHIFT-UpArrow,SHIFT-DownArrow, SHIFT-LeftArrow, and SHIFT-RightArrow conform to ANSI X3.64 for cursor positioning. There are no equivalents on the IBM PC.

Table 3-3. GRiD Compass and IBM PC Keystroke Comparisons

| IBM PC Keys | Equivalent GRiD Compass Keys | Output Code | |
|---|---|---|---|
| | | Hex | ASCII |
| cursor up | UpArrow | 00 48 | |
| cursor down | DownArrow | 00 50 | |
| cursor left | LeftArrow | 00 4B | |
| cursor right | RightArrow | 00 4D | |
| none | SHIFT-UpArrow | 1B 5B 41 | ESC [ A |
| none | SHIFT-DownArrow | 1B 5B 42 | ESC [ B |
| none | SHIFT-LeftArrow | 1B 5B 44 | ESC [ D |
| none | SHIFT-RightArrow | 1B 5B 43 | ESC [ C |
| page up | CODE-UpArrow | 00 49 | |
| page down | CODE-DownArrow | 00 51 | |
| home | CODE-SHIFT-UpArrow | 00 47 | |
| end | CODE-SHIFT-DownArrow | 00 4F | |
| ins | CODE-= | 00 52 | |
| del | CODE-- | 00 53 | |
| CTRL-cursor right | CTRL-SHIFT-RightArrow | 00 74 | |
| CTRL-cursor left | CTRL-SHIFT-LeftArrow | 00 73 | |
| CTRL-home | CTRL-CODE-SHIFT-UpArrow | 00 77 | |
| CTRL-end | CTRL-CODE-SHIFT-DownArrow | 00 75 | |
| CTRL-page up | CTRL-CODE-UpArrow | 00 84 | |
| CTRL-page down | CTRL-CODE-DownArrow | 00 76 | |
| SHIFT-tab | SHIFT-TAB | 00 0F | |
| null | null | 00 03 | |
| none | CODE-BACKSPACE | 00 85 | |
| F1 | CODE-1 | 00 3B | |
| F2 | CODE-2 | 00 3C | |
| F3 | CODE-3 | 00 3D | |
| F4 | CODE-4 | 00 3E | |
| F5 | CODE-5 | 00 3F | |

| IBM PC Keys | Equivalent GRiD | Output Code | |
|---|---|---|---|
| | Compass Keys | Hex | ASCII |
| F6 | CODE-6 | 00 40 | |
| F7 | CODE-7 | 00 41 | |
| F8 | CODE-8 | 00 42 | |
| F9 | CODE-9 | 00 43 | |
| F10 | CODE-0 | 00 44 | |
| F11 (SHIFT-F1) | CODE-SHIFT-1 | 00 54 | |
| F12 (SHIFT-F2) | CODE-SHIFT-2 | 00 55 | |
| F13 (SHIFT-F3) | CODE-SHIFT-3 | 00 56 | |
| F14 (SHIFT-F4) | CODE-SHIFT-4 | 00 57 | |
| F15 (SHIFT-F5) | CODE-SHIFT-5 | 00 58 | |
| F16 (SHIFT-F6) | CODE-SHIFT-6 | 00 59 | |
| F17 (SHIFT-F7) | CODE-SHIFT-7 | 00 5A | |
| F18 (SHIFT-F8) | CODE-SHIFT-8 | 00 5B | |
| F19 (SHIFT-F9) | CODE-SHIFT-9 | 00 5C | |
| F20 (SHIFT-F0) | CODE-SHIFT-0 | 00 5D | |
| alt A | CODE-A | 00 1E | |
| alt B | CODE-B | 00 30 | |
| alt C | CODE-C | 00 2E | |
| alt D | CODE-D | 00 20 | |
| alt E | CODE-E | 00 12 | |
| alt F | CODE-F | 00 21 | |
| alt G | CODE-G | 00 22 | |
| alt H | CODE-H | 00 23 | |
| alt I | CODE-I | 00 17 | |
| alt J | CODE-J | 00 24 | |
| alt K | CODE-K | 00 25 | |
| alt L | CODE-L | 00 26 | |
| alt M | CODE-M | 00 32 | |
| alt N | CODE-N | 00 31 | |
| alt O | CODE-O | 00 18 | |
| alt P | CODE-P | 00 19 | |
| alt Q | CODE-Q | 00 10 | |
| alt R | CODE-R | 00 13 | |
| alt S | CODE-S | 00 1F | |
| alt T | CODE-T | 00 14 | |
| alt U | CODE-U | 00 16 | |
| alt V | CODE-V | 00 2F | |
| alt W | CODE-W | 00 11 | |
| alt X | CODE-X | 00 2D | |
| alt Y | CODE-Y | 00 15 | |
| alt Z | CODE-Z | 00 2C | |

There are no direct GRiD Compass equivalents for the following IBM PC keys:

F21-30 (CTRL F1-10)
F31-F40 (alt F1-F10)
alt 1,2,3,4,5,6,7,8,9,0,-,=
CTRL PrtSc

# CHAPTER 4. DEVICE CONFIGURATIONS

Chapter 2 described how to start up MS-DOS from a diskette. It is possible to run MS-DOS and application programs solely from diskettes. However, if you want to use some of the other storage capabilities of the GRiD Compass system, the bubble memory and the hard disk drive can also be used by MS-DOS. This chapter describes the implications of using these other storage devices and the steps necessary to prepare these devices for use by MS-DOS.

## SYSTEM CONFIGURATIONS AND DEVICE ASSIGNMENTS

The most common GRiD Compass configuration utilizes a single disk drive: the 2102 Portable Diskette drive. In this configuration the Portable Diskette drive is device "A:" and the Bubble Memory is device "E:" (see Figure 4-1). Additional configurations are possible as well.

Table 4-1 lists the devicퟠ designation assignments made by MS-DOS when used on the GRiD Compass with various different system configurations. From this table, you will note that if there is only a single floppy diskette drive in the system, it is treated as a logical device: that is, the same physical disk drive is assigned two device designations -- A: and B:.

Table 4-1 also lists the General Purpose Interface Bus (GPIB) address for each of the devices. The GPIB address is set via a set of switches on the device itself and allows that device to communicate with the GRiD Compass. Check the GRiD Compass Owner's Guide and the owner's manual accompanying each peripheral device for the location of the GPIB switches on the device.

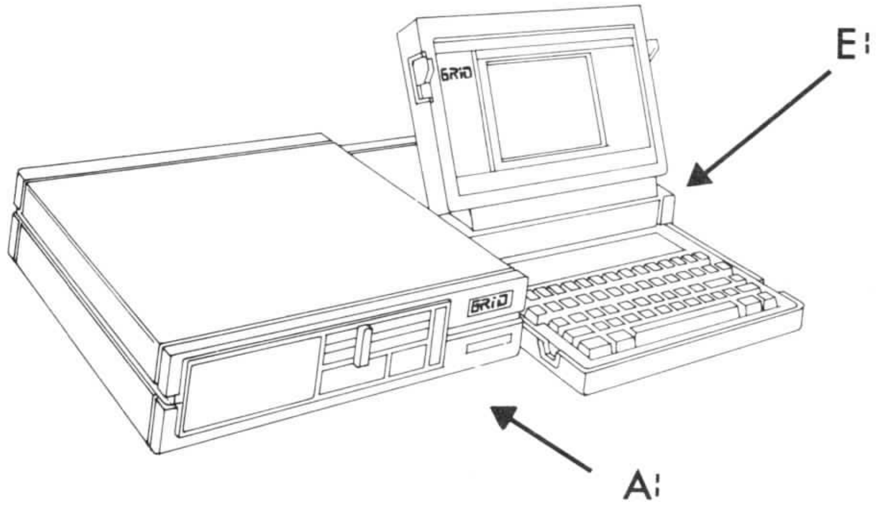


Figure 4-1. Letter Designations for the Floppy and Bubble

Notice that regardless of the configuration, the primary floppy diskette drive is always "A:" and the bubble memory is always "E:".

When the GRiD Compass is powered on, it examines the attached storage devices in a prescribed sequence to determine where to locate the bootstrap (or boot) routine (which allows the computer to load main memory with the necessary files for operation). The bubble memory is always checked first, followed by the hard disk drive, and then the floppy diskette drive.

If you wish to bypass this sequence when you turn your computer on, hold down the "F" key to boot from the floppy diskette drive, or hold down the "H" key to boot from the hard disk drive.

In order to use any storage device (the bubble memory, a floppy diskette drive, or the hard disk drive), the device must first be "formatted". An MS-DOS program named FORMAT is supplied on the diskette. The formatting procedure will be described later in Chapter 5.

If you have formatted the bubble memory as an MS-DOS device and copied MS-DOS to the bubble memory, you will automatically boot into MS-DOS when the GRiD Compass is turned on.

If you have partitioned and formatted the hard disk (partitioning is described later in Chapter 5) you can copy MS-DOS to the hard disk. You can boot directly into MS-DOS from the hard disk by holding down the "H" key while turning the computer on.

Table 4-1. DOS Device Designations and System Configurations

| Device Designation | SYSTEM CONFIGURATION | GPIB Address (Hexadecimal) |
|---|---|---|
| | Compass with one Portable Floppy (No Hard Disk) | |
| A: | Portable Floppy Drive (logical) | 6 |
| B: | Portable Floppy Drive (logical) | 6 |
| C: | Not available | |
| D: | Not available | |
| E: | Bubble Memory | |
| | Compass with Two Portable Floppies (No Hard Disk) | |
| A: | Floppy Drive (logical) | 5 |
| B: | Floppy Drive (logical) | 5 |
| C: | Not available | |
| D: | Not available | |
| E: | Bubble Memory | |
| F: | Portable Floppy #2 (logical) | 6 |
| G: | Portable Floppy #2 (logical) | 6 |
| | Compass with One 2101 Disk System (No Portable Floppy Drive) | |
| A: | Floppy Drive (logical) | 5 |
| B: | Floppy Drive (logical) | 5 |
| C: | Hard Disk | 4 |
| D: | Not available | |
| E: | Bubble Memory | |
| | Compass with One 2101 Disk System (Optional Portable Floppy) | |
| A: | Floppy Drive (logical) | 5 |
| B: | Floppy Drive (logical) | 5 |
| C: | Hard Disk | 4 |
| D: | Not available | |
| E: | Bubble Memory | |
| F: | Portable Floppy (logical) | 6 |
| G: | Portable Floppy (logical) | 6 |
| | Compass with Two 2101 Disk Systems (Optional Portable Floppy) | |
| A: | Floppy Drive (first 2101) | 4 |
| B: | Floppy Drive (second 2101) | D |
| C: | Hard Disk (first 2101) | 5 |
| D: | Hard Disk (second 2101) | C |
| E: | Bubble Memory | |
| F: | Portable Floppy (logical) | 6 |
| G: | Portable Floppy (logical) | 6 |

In addition to the storage devices available to the MS-DOS user, other peripheral devices can be accessed by signifying their appropriate designations. Some of these devices are normally addressable under MS-DOS; some of them have been added exclusively for the GRiD Compass version of MS-DOS. None of the following device names should ever be used as file names or extensions. The operating system reserves these names for the purposes listed in Tables 4-2 and 4-3.

Table 4-2. Normal MS-DOS Device Designations

| Device Designations | Definitions |
|---|---|
| AUX | Refers to an auxiliary device, a printer or disk drive, to which READ and WRITE operations are performed. |
| CON | Refers to a READ operation (of the keyboard) or a WRITE (to the terminal display screen). |
| NUL | Designates a surrogate file name for a READ or a WRITE when required by the command format (used if you do not wish to create a new file). |
| LST or PRN | References the printer. |

Table 4-3. GRiD Compass Additions to MS-DOS Devices

| Device Designations | Definitions |
|---|---|
| LPT1 | Equivalent to the MS-DOS 'PRN' designation. References the GPIB printer connector. This designation is the standard IBM name for the printer and has been added for IBM-PC compatibility. |
| LPT2 | Equivalent to the standard IBM name for a serial printer. The serial channel defaults that apply in this application are: 7 data bits, 1 stop bit, even parity, CTS handshaking during data exchange, and 1200 baud (identical to the IBM-PC default values). |
| COM1 | Refers to the asynchronous, serial data communication channel, labeled SERIAL or ACCESSORY on the GRiD Compass back panel. |
| COM1STAT | Provides control of the serial communication channel, and obtains status information from it. |
| COM2 | Refers to the asynchronous serial data communication channel routed through the GRiD Compass internal modem, labeled LINE on the back panel. |
| COM2STAT | Provides control of the internal modem default settings, and obtains status information from this data channel. |

NOTE: More information on the communication channels and their use is provided in the following section and in Appendix C.

## COMMUNICATION CHANNEL CONFIGURATIONS

The GRiD Compass provides two distinct channels for asynchronous, serial data communications under MS-DOS.

1. The serial connector, located on the back panel of the computer and labeled SERIAL (or ACCESSORY on earlier models), maintains RS-232-C and RS-422 conventions. A printer, modem, plotter or other asynchronous, serial device can be attached to this interface.

2. A telephone connector, labeled LINE, ties the GRiD Compass's internal modem to the telephone lines. Voice communication through the internal modem can also be accomplished via the AUDIO connector.

During system initialization, driver routines loaded by MS-DOS configure each of the communication channels in a specific way. The following values reflect the default condition for the SERIAL connector:

TABLE 4-4. SERIAL Default Values

| | |
|---|---|
| Data format: | 7 data bits, 1 stop bit, even parity |
| Baud rate: | 300 |
| Line feed insertion: | Off |
| Asynchronous mode: | Set on |

Appendix C lists the full specifications for the SERIAL port.

The described values correspond to the default settings on a IBM PC produced by an OPEN "COMn:" command within PC-BASIC. Serial devices, such as the DIABLO 630 printer or the HP 7470A plotter, can be driven with the communications channel configured in this manner. A modem, such as the Hayes Smartmodem 1200, can be attached only if the serial interface is deconditioned for DSR (Data Set Ready). Information on how to accomplish this is included in APPENDIX C of this manual.

The LINE connector supports data communications at 300 and 1200 baud using the GRiD Compass's internal modem. An emulator routine allows this internal modem to duplicate most of the functions of the Hayes Smartmodem 1200.

The following default values apply to this interface:

TABLE 4-5. LINE Default Values

| | |
|---|---|
| Data format: | 7 data bits, 1 stop bit, even parity |
| Baud rate: | 300 baud |
| Asynchronous mode: | Set on |

The values provided by the modem driver correspond to the following configuration switch settings on the Hayes Smartmodem 1200:

Switch 2...DOWN: Issue digital response codes

Switch 3.....UP: Issue no response codes at all

Switch 4...DOWN: Do not echo commands to user

Switch 5...DOWN: Must issue 'A' command to answer

The emulator operates using the normal default conditions that exist when a Hayes Smartmodem is initially powered on. However, not all of the functions of the Hayes Smartmodem are supported. For a detailed list of differences and exceptions, refer to Appendix C.

Programming either of these communication channels allows you to depart from the default values described to accommodate other data communications configurations. Consult Appendix C for a discussion of how to configure the modem port or the serial port for custom applications.

# CHAPTER 5. FORMATTING STORAGE DEVICES

The formatting process marks the storage media -- floppy diskette drive, bubble memory, or hard disk drive -- with the information that allows the operating system to locate specific storage areas. Since each operating system has a different system for marking and accessing a storage device, a hard disk drive formatted solely for MS-DOS is not usable by GRID/OS. However, the hard disk drive can be partitioned, dividing it into separate areas for use by more than one operating system. The FDISK process is explained in the second portion of this chapter.

The first portion of the chapter discusses the MS-DOS format command which can be used to format floppy diskettes, the bubble memory, or the hard disk drive.

## THE FORMAT COMMAND

You must format all new diskettes before they can be used with MS-DOS and you must also format bubble memory and the hard disk in order to use them as storage devices with MS-DOS. Before you format the hard disk drive, you must first run the FDISK utility.

NOTE: When you format a device, any data previously stored on the device is destroyed.

The syntax for the FORMAT command is as follows (parameters enclosed in brackets are optional):

```
     FORMAT [d:][/S][/1][/8][/V][/O][/C]
```

where...

d: is the designation of the device to be formatted. For example, "FORMAT c:" specifies that the hard disk (Device c:) is to be formatted. If no device designation is specified, the system assumes that the current default device is intended. Refer to Chapter 4 for an explanation of device designations.

/S is the optional parameter that causes the operating system files to be copied to the device so that you can then boot from the device. If this parameter is not specified, the entire device is devoted to data storage.

/1 is the optional parameter that specifies that a floppy diskette is to be formatted for single-sided use, regardless of the fact that the specified drive is a double-sided drive. If not specified, the diskette is formatted for double-sided use.

/8 is the optional parameter specifying that a diskette be formatted with 8 sectors per track. If not specified, the diskette is formatted with 9 sectors per track.

/V is the optional parameter that lets you include a volume label on the diskette. If you specify /V, the program will prompt you to enter a volume label allowing you to identify the diskette. The volume label, however, is not used by the system.

/O is the optional parameter that is used to create a diskette that is compatible with Version 1.X of MS-DOS or IBM PC-DOS.

/C is the optional parameter that can be specified to cause a "quick" reformatting of a previously formatted diskette. It removes all directory entries and, therefore, removes all data from the diskette. It does not, however, rewrite all of the sector headers. This option can only be used if the diskette is being reformatted in the same configuration (number of sides and sectors per track) as its existing format.

▶TO FORMAT A SYSTEM DISKETTE

1. The following command will format a new diskette in the floppy disk drive as a system diskette (one containing MS-DOS).

```
          FORMAT a:/s
```

2. MS-DOS then issues the following message:

```
Insert new diskette for drive A:
and strike any key when ready.
```

3. After you insert the new diskette in the drive, press any key on the keyboard. MS-DOS then issues the following message:

   ```
   Hard format in progress
   ```

   is displayed while MS-DOS is formatting the diskette and copying system files. When the formatting process has been completed, a message similar to the following will be displayed:

   ```
   # of 720 sectors formatted.
   ```

   The number (#) will initially be 1 and will be incremented as the formatting proceeds. You can thus see how far the operation has progressed and gauge the time needed until completion.

   When the formatting is complete, a message similar to the following one is displayed:

   ```
   Insert DOS diskette in drive A:
   and strike any key when ready.
   ```

   NOTE: this message is displayed only if you specified that the diskette is to be formatted as a system diskette (/s).

4. Insert the MS-DOS diskette and press any key. MS-DOS reads one last system file into memory and then (for the last time) issues the message:

   ```
   Insert new diskette for drive A:
   and strike any key when ready.
   ```

5. After you have inserted the diskette being formatted and pressed a key, the system files are written to the diskette and a message similar to the following one is displayed:

   ```
   System transferred
        362496 bytes total disk space
         58368 bytes used by system
        304120 bytes available on disk

   Format another (Y/N)?
   ```

6. Type Y to format another disk. Type N to end the FORMAT operation.

▶TO FORMAT BUBBLE MEMORY

The procedure for formatting bubble memory is essentially as described above. This command would format bubble memory and include MS-DOS on the bubble:

```
FORMAT e:/s
```

## HARD DISK FORMATTING AND PARTITIONING WITH FDISK

For a hard disk drive (also known as a fixed disk drive) to be accessed by the GRiD Compass, it must first have formatting information recorded upon the surface of the media. This formatting information, composed of sector addresses and links, makes available to the computer in a systematic way all data storage areas on the disk's surface. It also allows the computer to divide the disk up into areas, known as partitions, for specific uses.

The utility program FDISK allows the option of either maintaining the entire storage area of the hard disk drive under one operating system or creating a number of partitions, each related to a different operating system. For example, you may have one partition dedicated to MS-DOS and another to GRiD/OS. However, programs and files cannot be transferred back and forth between partitions without some type of utility program that converts the data formats.

You have the option of initially devoting all of your hard disk drive space to MS-DOS and then, if you decide that you wish to include another operating system on the same disk drive, to create additional partitions at a later time. However, to accomplish this you would first need to back up all files that you wanted to save since this procedure requires total reformatting of the disk drive. One additional operating system per partition may then be assigned up to a maximum of four partitions per disk drive.

The first procedure in this section describes how to prepare the hard disk drive for use with MS-DOS. The second procedure explains how to create partitions for using more than one operating system on the drive.

## PREPARING THE HARD DISK DRIVE FOR MS-DOS

NOTE: MS-DOS recognizes only one MS-DOS partition per hard disk drive.

▶TO CREATE AN MS-DOS PARTITION

1. With your unformatted hard disk drive properly cabled to the system, boot MS-DOS from the floppy disk drive. When the A-prompt (A>) appears, type: FDISK

2. Press the RETURN key and the FDISK set up options appear on the screen with option 2 automatically selected for you.

3. Press the RETURN key. The option to "Create DOS Partition" will be initiated. The screen responds:

   ```
   Current Fixed Disk Drive: 1

   Do you wish to use the entire fixed disk for DOS
   (Y/N).............................? (Y)
   ```

   NOTE: If your hard disk drive has already been set up, the screen will display a list of partition assignments. In such an instance, continue to the procedure 'ADDING PARTITIONS TO YOUR FIXED DISK' later in this section.

4. Otherwise, press the RETURN key to answer YES to the screen prompt. The message appears:

   ```
   Press any key to reboot...
   ```

5. Restarting MS-DOS in this manner allows the system to recognize the hard disk drive and assign a letter designation (C:) to it. Press any key to initiate this process.

   An MS-DOS partition has now been created, but the hard disk drive must still be formatted. Follow the described sequence only if your disk drive has not previously been formatted and used for data storage.

   **CAUTION:** Formatting destroys any data that is residing on the disk drive.

### ▶TO FORMAT THE HARD DISK DRIVE

To format your partition for MS-DOS (allowing the system to be booted from the hard disk drive) perform the following steps.

1. Enter the command:

   ```
   FORMAT C:/S (or FORMAT C:/S/V to add a
   volume label)
   ```

2. The screen prompts you:

   ```
   Press any key to begin formatting drive C:
   ```

   As soon as you press the next key, the red light on the disk drive illuminates and the message "# of # sectors formatted" appears on the screen, giving you a running count of the formatting process.

3. After several minutes the message "Format complete" appears, followed by the message "System transferred".

4. If you specify the /V option, MS-DOS will ask for the volume.

   ```
   Volume label(11 characters, RETURN for none)?
   ```

   Since this label affords a useful identification feature for certain programs, we recommend that you label the drive at this time. You cannot add the volume label later.

5. Once you complete this procedure, your hard disk drive is formatted and fully operational.

▶TO COPY PROGRAMS TO THE HARD DISK DRIVE

1. At this point, it is advisable to copy all the programs from your MS-DOS diskette (which should still be in the floppy disk drive) to your hard disk drive. To accomplish this, enter the command:

   ```
   COPY A:*.* C:
   ```

2. Other programs and files on separate diskettes that you may wish to use may be transferred to the hard disk drive in a similar manner.

▶TO INITIALIZE FROM HARD DISK

1. Use FDISK to activate the partition if you have not already done so.

2. To boot MS-DOS from the hard disk drive, simply hold down the "H" key when turning your GRiD Compass on.

   If your system has more than one hard disk drive, additional options are available. Enter option 6 from the first FDISK screen if you wish to create an active partition on your second hard disk drive. Note that it is only possible to boot MS-DOS from the first disk drive.

## ADDING PARTITIONS TO YOUR FIXED DISK

If you intend to use more than one operating system on your hard disk drive, the drive must be divided into partitions. Partitions may vary in size and occur in any order. Each operating system capable of interfacing to a hard disk drive will have its own program for creating a partition within the drive's data storage area. FDISK creates the partition for MS-DOS; you must use a separate program to create a partition for GRiD/OS.

Besides allowing you to create a partition for MS-DOS, FDISK allows you to change the designated active partition (the partition from which the system is initialized), delete any partition, display information about all partitions on the disk drive, and select a second hard disk drive within the system configuration. Each of these features is discussed in the following sections.

▶TO SELECT FROM THE OPTION LIST

1. To access any of these options, you must first type FDISK and then press the RETURN key. The screen displays your options.

```
Choose one of the following:

1.  Exit Fixed Disk Setup Program
2.  Create DOS Partition
3.  Change Active Partition
4.  Delete DOS Partition
5.  Display Partition Data
6.  Select Next Fixed Disk Drive

Enter choice: (2)
```

NOTE: Option 6 only appears if more than one disk drive is attached to the computer.

2. Select your option, type the appropriate number, and press RETURN. Turn to the description of your selected option for the correct procedure to follow.

### Create DOS Partition -- Option 2

CAUTION: Be sure to backup all data files currently on the hard disk drive before creating the DOS partition.

To use this option, you must first decide where on the disk drive you want your MS-DOS partition to be located, and how much disk space you wish to reserve for it.

The screen displays all the pertinent information: the total storage capacity of the drive, the number of cylinders available, and the number of bytes that may be contained on each cylinder. Using this information you can calculate what percentage of the data storage area you wish to devote to MS-DOS.

▶TO CREATE AN MS-DOS PARTITION

1. The first screen displayed when you select this option (if no other operating systems are resident on the drive) prompts:

```
Do you wish to use the entire fixed disk for DOS
(Y/N)...................? (Y)
```

2. Type N and press RETURN to create more than one partition. The screen displays the disk capacity in cylinders.

   ```
   Total fixed disk space is xxxx cylinders.
   Maximum available space is yyyy.
   Cylinders at cylinder zzzz.
   ```

3. If the disk drive has previously been initialized and other operating systems are present, the display will list the specifications for each. The same sequence applies as below, but you must take care not to attempt to overlap your MS-DOS memory assignments with areas that have already been assigned to other systems. An error message occurs under those circumstances.

4. The prompt appears:

   ```
   Enter partition size......:(xxxx)
   ```

   Type in the desired size of your MS-DOS partition in cylinders and press RETURN. If you press RETURN without entering a number, the program selects as a default value the largest area of memory available.

5. The next prompt reads:

   ```
   Enter starting cylinder number...:(yyyy)
   ```

   Type in the starting cylinder number you have selected and press RETURN. The cursor is placed at the bottom of the screen and the message displayed:

   ```
   Press ESC to return to the FDISK Options
   ```

6. The partition is now set up for MS-DOS but you must still format the partition before it can be used for data storage. Consult the previous section describing formatting procedure. If you wish this partition to be active, follow the procedure for Changing the Active Partition.

## Change the Active Partition -- Option 3

▶TO CHANGE THE ACTIVE PARTITION

1. When you select this option, the display lists the partitions describing their status, type, and sizes. Note that each partition is designated by a number. The prompt at the bottom of the screen reads:

```
Enter the number of the partition you want
to make active.
Enter 0 if no partition is to be made active
(0)
```

The partition number you select is automatically accessed whenever the system is initialized from the hard disk drive.

2. Type in your selection and press RETURN.

3. The selection is recorded. You may return to the FDISK options menu by pressing ESC. Press ESC twice to return to MS-DOS.

NOTE: If no active partition has been declared, the program issues a prompt when the system boot occurs asking which partition to boot from.

## Delete the MS-DOS Partition -- Option 4

**CAUTION:** Do not perform this function without backing up all files contained within the partition that you wish to save. All data is destroyed when you delete the partition.

▶TO DELETE A PARTITION

1. Reassign the active partition if you have been initializing from MS-DOS. Remember that after you delete the MS-DOS partition you will need a MS-DOS diskette if you wish to continue processing in MS-DOS.

2. When you select this option the screen displays the partition description and issues the following message.

```
Enter the number of the partition to be deleted:
```

3. When you make your selection, a final caution appears on the screen.

```
Warning!  All data in this partition will be
DESTROYED.   Do you wish to continue? (N)
```

Type Y and press RETURN only if you have backed up all valuable and necessary files. If you have not taken this precaution, you may exit the option by pressing RETURN or pressing the ESC key.

4. The partition has been deleted when you see the following message appear:

```
Press ESC to return to FDISK Options
```

## Display Partition Data -- Option 5

Invoking this option allows you to access useful information concerning the individual partitions. The display includes the following:

Partition: The identification number of each of the partitions currently assigned to the disk drive.

Status: An 'A' denotes the active partition (accessed immediately whenever the system is initialized from the disk drive). An 'N' denotes the remaining non-active partitions.

Type: Shows which partition is the MS-DOS partition. All others are not labelled.

Start and End: Defines the beginning and ending cylinders for each of the partitions.

Kilobytes: Declares the size of the partition in kilobytes.

% of Disk: Calculates the percentage of the disk drive utilized by the partition.

## Selecting Next Fixed Disk Drive -- Option 6

This option offers the ability to extend the use of the FDISK program to a second hard disk drive configured upon the system. Partitions can be created for this second drive in the same manner that has been described in this section.

This second hard disk drive cannot be used for system initialization. You can only boot from an operating system located on drive C.

# CHAPTER 6.  GRiD COMPASS MS-DOS COMMANDS

In addition to the MS-DOS commands described in Chapter 5 of Microsoft's MS-DOS User's Guide, two special MS-DOS commands -- DEVICES and SETTIME -- are provided for use with the GRiD Compass. These commands are external commands; that is, they are not always resident in memory when MS-DOS is loaded.

## THE DEVICES COMMAND

This command tells you what devices are active in your system and displays both the MS-DOS device designation (for example, A, B or D) and the GPIB address of each device. The GPIB address switches on disk devices are used by the GRiD Compass to route data to and from system devices for MS-DOS. When you issue the DEVICES command, a display similar to the following will appear:

```
Drive     Type                 GPIB Address (Hexadecimal)
=====     =========            ==========================

  A:      Logical Floppy          5
  B:      Logical Floppy          5
  C:      Hard Disk  1            4
  D:      Not available
  E:      Bubble
  F:      Not Available           6
  G:      Not Available           6
```

## THE SETTIME COMMAND

This command lets you take advantage of the real-time clock built into the GRiD Compass to supply the current date and time to MS-DOS. Because of the availability of this command and the real-time clock, the time and date prompts that you are given when you first boot into MS-DOS are now nonfunctional. Once you have run the SETTIME command, the "Current time" and "Current date" are displayed with the correct time and date every time you boot the system.

The prompts "Enter new time" and "Enter new date" are also displayed, but responses to these prompts are simply ignored by the system. To run the SETTIME program, enter the program name: SETTIME. You will be prompted with:

```
Current time is 0:00:00.00
Enter new time:_
```

Enter the current time in the 24-hour format (HH:MM:SS) and press RETURN. The following message is then displayed:

```
Current date is 1-01-1980
Enter new date:_
```

Enter the current date in the format MM-DD-YY, where

MM is a one- or two-digit number from 1-12 (representing month)

DD is a one- or two-digit number from 1-31 (representing day)

YY is a two-digit number from 80-99 (the 19 is assumed) or a four digit number from 1980-2099 (representing year).

Separators between the numbers can be hypens (-) or slashes (/).

# CHAPTER 7. RUNNING CUSTOMIZED APPLICATION SOFTWARE

This chapter discusses unique features of MS-DOS application programs which have been customized for the GRiD Compass. Examples are given for two popular MS-DOS programs, WordStar and Multiplan, to demonstrate how you can fill in the template provided with your GRiD Compass. These templates help to identify the correct key definitions for each application.

## CUSTOMIZED APPLICATION CONSIDERATIONS

Those programs that have been customized specifically for the GRiD Compass run more directly than software that must be installed. Typing in the program name generally is enough to access the program without the necessity of following a lengthy installation procedure.

The primary problem is recognizing which keys and key combinations on the GRiD Compass perform which functions within the context of the program. Each application package offers a different approach to keyboard use. In order to keep track of the key definitions for a specific program, such as WordStar, we recommend that you utilize the blank template that we provide. By filling in the template with the correct functions for each combination of keys, you will have a convenient reference while running the program.

Use the method described in the following WordStar section for any application program that requires the use of special purpose or function keys. By pressing each of the functions keys in sequence you can systematically record all the operations associated with each function key and notate the meanings on the appropriate template.

Examine the template cards provided with your MS-DOS program diskette. Note that the function keys F1 through F10 on the template are obtained on the Compass keyboard by holding down the CODE key while pressing the desired numerical key, 1 through 10 (0 on the GRiD Compass is used in place of 10). Holding down CODE and SHIFT while pressing a numerical key achieves the same results as function keys F11 through F20.

## DATA COMPATIBILITY

MS-DOS for the GRiD Compass is designed to format and create data diskettes which are 100% compatible with PC-DOS 2.0 data diskettes used by the IBM Personal Computer. This means that all GRiD Compass MS-DOS data diskettes can be interchanged between the GRiD Compass and any IBM compatible machines using PC-DOS 2.0

Data diskettes which have been created by MS-DOS version 1.1 are upwards compatible. To allow software running under MS-DOS version 1.1 to read data created while running under MS-DOS 2.0, you must write the data to diskettes which have been made format-compatible in one of two ways:

1. Write the data to be stored using diskettes formatted while running MS-DOS version 1.1.

2. Use the format command option -- /O -- when formatting a diskette while running MS-DOS version 2.0.

## WORDSTAR CUSTOMIZED FOR THE GRiD COMPASS

To access WordStar, type 'WSU' (or 'WS' if you have previously installed and run the program on your GRiD Compass).

If you load the WordStar program, you can see the result of pressing each of the GRiD Compass key combinations. Recording each of these actions results in the following table:

Table 7-1. WordStar Function Key Equivalents

| MS-DOS FUNCTION KEYS | GRiD COMPASS KEYSTROKES | WORDSTAR COMMAND | FUNCTION |
|---|---|---|---|
| F1 | CODE-1 | ^JH | Set Help Level |
| F2 | CODE-2 | ^OG | Paragraph Tab |
| F3 | CODE-3 | ^OL | Set Left Margin |
| F4 | CODE-4 | ^OR | Set Right Margin |
| F5 | CODE-5 | ^PS | Underscore Line |
| F6 | CODE-6 | ^PB | Boldface |
| F7 | CODE-7 | ^KB | Mark Block Beginning |
| F8 | CODE-8 | ^KK | Mark Block End |
| F9 | CODE-9 | ^KC | Block Copy |
| F10 | CODE-0 | ^KS | Save File and Reedit |

Note that you can enter the normal WordStar commands prefaced with the GRiD Compass's CTRL (Control) key. For example, rather than pressing CODE-2 on the Compass keyboard to generate the Paragraph Tab (^OG) function, you can hold down CTRL while pressing 'O' and then 'G'. Of course, simply pressing CODE-2 saves a keystroke.

Additionally, the combination of CODE/BACKSPACE deletes the word to the left of the cursor, a function normally not available with WordStar. BACKSPACE used alone produces the Delete function (character to the left).

You may enter these key conversions onto your template in any form that is convenient for you. The F7 block could contain the ^KB characters, or, if you wish to be more direct, you could enter MARK BLOCK BEGINNING.

As an example of how you might wish to mark your own template, Figure 7-1 illustrates the template corresponding to the WordStar key conversions with the blocks filled in.

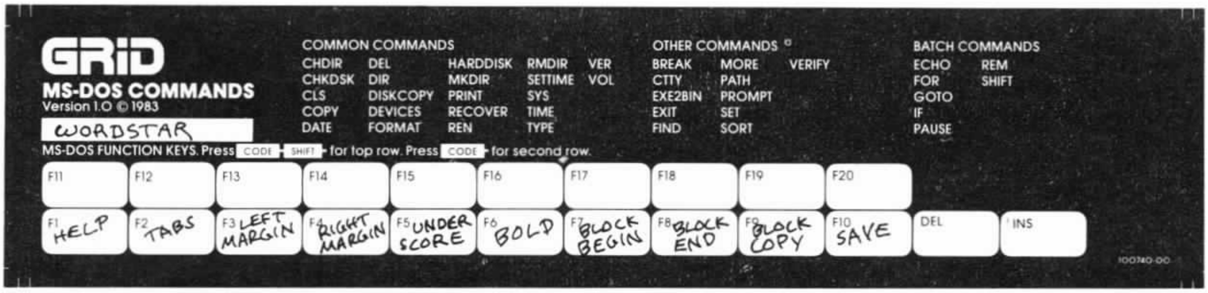


Figure 7-1. Example of Completed WordStar Template

The GRiD Compass arrow keys also can be used in combination with the CODE and SHIFT keys to perform specific WordStar functions. The following meanings apply:

Table 7-2. Arrow Key Definitions for WordStar

| GriD Compass Keys | WordStar Feature |
|---|---|
| LeftArrow | Control S (Move left one character) |
| RightArrow | Control D (Move right one character) |
| UpArrow | Control E (Move up one line) |
| DownArrow | Control X (Move down one line) |
| CODE-LeftArrow | Control A (Move left a word at a time) |
| CODE-RightArrow | Control F (Move right a word at a time) |
| CODE-UpArrow | Control R (Move up one page) |
| CODE-DownArrow | Control C (Move down one page) |
| CODE-SHIFT-LeftArrow | Control QS (Move to left margin) |
| CODE-SHIFT-RightArrow | Control QD (Move to right margin) |
| CODE-SHIFT-UpArrow | Control QR (Return to start of file) |
| CODE-SHIFT-DownArrow | Control QC (Move to end of file) |

You can alter the actual commands associated with each function key by using the WordStar installation program (WINSTALL) to define the desired functions. Refer to your WordStar documentation for an explanation of the procedure.

The GRiD Compass version of WordStar offers the capability of driving a letter-quality printer; the Diablo 630 with the GPIB interface is the recommended choice. You need to use the WSINSTALL program included on your program diskette (see the WordStar manual for instructions on running WSINSTALL). When you reach the option requesting your choice of printers, select the XEROX/DIABLO 1720/1620. This selection adapts your WordStar program for the Diablo 630, and will run this printer without any further modifications. Other printers may also be installed for use with the GRiD Compass.

## MULTIPLAN CUSTOMIZED FOR THE GRiD COMPASS

Multiplan has been customized to take advantage of the GRiD Compass keyboard. The keys that apply follow in Table 7-3.

Table 7-3. Multiplan Keystrokes

| Multiplan Function | GRiD Compass Keystrokes |
|---|---|
| Cancel | ESC |
| Delete | CODE-hyphen (-) |
| Character Left | CODE-9 or SHIFT-LeftArrow |
| Character Right | CODE-0 or SHIFT-RightArrow |
| Word Left | CODE-7 or CODE-SHIFT-LeftArrow |
| Word Right | CODE-8 or CODE-SHIFT-RightArrow |
| Help | CODE-? |
| Next Window | CODE-1 |
| Next Unlocked Cell | CODE-2 |
| Reference | CODE-3 |
| Recalculate | CODE-4 |
| Home | CODE-5 or CODE-SHIFT-UpArrow |
| End | CODE-6 or CODE-SHIFT-DownArrow |
| Page Up | CODE-UpArrow |
| Page Down | CODE-DownArrow |
| Page Right | CODE-RightArrow |
| Page Left | CODE-LeftArrow |

For an example of a template filled out for Multiplan, consult Figure 7-2.

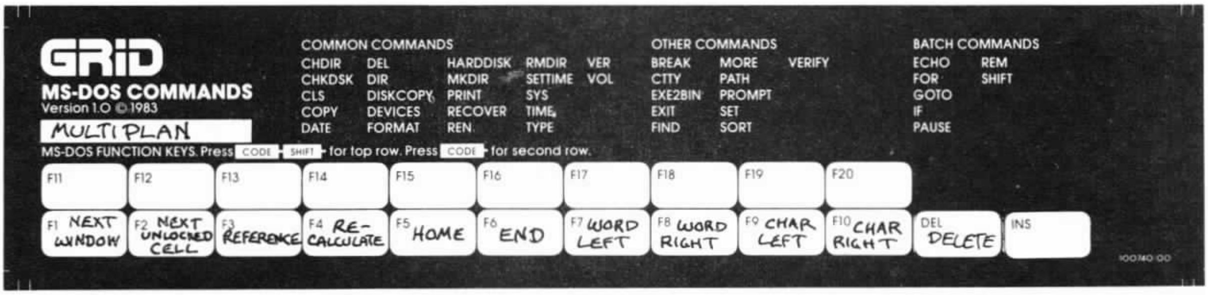


Figure 7-2. Example of Multiplan Template

## LOTUS 1-2-3 CUSTOMIZED FOR THE GRiD COMPASS

For users who are already familiar with the functions and the features of Lotus 1-2-3 on the IBM Personal Computer, this section highlights the unique features of 1-2-3 on the GRiD Compass.

### DATA COMPATIBILITY

Lotus 1-2-3 for the GRiD Compass is data compatible with the IBM Personal Computer version of Lotus 1-2-3. This means that you can create floppy diskettes containing 1-2-3 data on either system and use those same diskettes interchangeably on both the GRiD Compass and any IBM compatible machines.

Lotus 1-2-3 runs only under MS-DOS 2.0 (Version A) on the GRiD Compass. If you wish to interchange data diskettes with IBM-compatible machines running MS-DOS 1.1, you must follow the rules of MS-DOS compatibility.

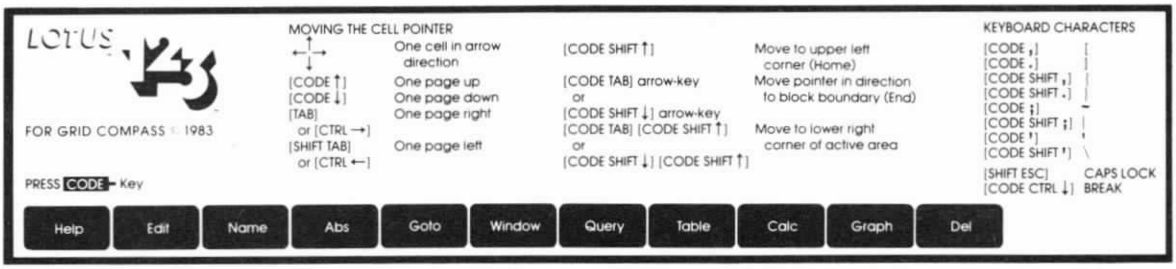
### KEYBOARD USE

The following table compares IBM keyboard use while running Lotus 1-2-3 to the equivalent GRiD Compass keystrokes.

Table 7-4. IBM PC to GRiD Compass Keystrokes

| IBM Keyboard | GRiD Compass keystrokes |
|---|---|
| F1 through F10 | CODE-1 through CODE-0 |
| CAPS Lock | SHIFT-ESC |
| Home | CODE-SHIFT-UpArrow |
| End | CODE-TAB or CODE-SHIFT-DownArrow |
| PgUp | CODE-UpArrow |
| PgDn | CODE-DownArrow |
| Del | CODE-hyphen (-) |
| Break | CODE-CTRL-DownArrow |
| Backslash (\) | CODE-SHIFT-apostrophe (') |
| Left Brace ({) | CODE-SHIFT-comma (,) |
| Right Brace (}) | CODE-SHIFT-period (.) |
| Tilde (~) | CODE-semicolon (;) |
| Alt-A through Alt-Z | CODE-A through CODE-Z |
| Alt-F1 | CODE-SHIFT-1 |

The 1-2-3 template card illustrates the GRiD Compass keystrokes used when running 1-2-3.



LOTUS 1-2-3

FOR GRID COMPASS © 1983

PRESS CODE - Key

MOVING THE CELL POINTER

| | |
|---|---|
| ←↑↓→ | One cell in arrow direction |
| [CODE ↑] | One page up |
| [CODE ↓] | One page down |
| [TAB] or [CTRL →] | One page right |
| [SHIFT TAB] or [CTRL ←] | One page left |
| [CODE SHIFT ↑] | Move to upper left corner (Home) |
| [CODE TAB] arrow-key or [CODE SHIFT ↓] arrow-key | Move pointer in direction to block boundary (End) |
| [CODE TAB] [CODE SHIFT ↑] or [CODE SHIFT ↓] [CODE SHIFT ↑] | Move to lower right corner of active area |

KEYBOARD CHARACTERS

| | |
|---|---|
| [CODE ,] | [ |
| [CODE .] | ] |
| [CODE SHIFT ,] | \| |
| [CODE SHIFT .] | \| |
| [CODE ;] | ~ |
| [CODE SHIFT ;] | \| |
| [CODE '] | ` |
| [CODE SHIFT '] | \ |
| [SHIFT ESC] | CAPS LOCK |
| [CODE CTRL ↓] | BREAK |

Figure 7-3. Lotus 1-2-3 Template

## SOFTWARE PROTECTION

The customized version of Lotus 1-2-3 utilizes the full power of the GRiD Compass. Portability is one advantage. Lotus 1-2-3 can be copied onto the bubble memory and run without any attached storage device. The capacity and performance of the GRiD 2101 hard disk drive is also utilized fully.

Before you can use Lotus 1-2-3 on the GRiD Compass, you must run a utility program entitled INITIAL. Consult your 1-2-3 documentation for instructions. INITIAL marks the 1-2-3 system diskette with a unique serial number, specific to your GRiD Compass. Once this process is completed, you can load 1-2-3 onto your bubble memory or hard disk drive.

NOTE: Lotus 1-2-3 works only on the bubble memory or hard disk drive identified during the INITIAL program. Do not run the INITIAL program on your back-up copy of the 1-2-3 system diskette at this time. If your original diskette or your GRiD Compass is lost, stolen, or damaged, you will then need to INITIAL your back-up diskette.

Once you have run the INITIAL utility, you will be able to run the 1-2-3 program from the bubble memory or hard disk drive of another GRiD Compass only by inserting the 1-2-3 system disk into a floppy drive connected to it.

# CHAPTER 8. INSTALLING MS-DOS APPLICATION SOFTWARE

A wide variety of software products, written by many software manufacturers, run under MS-DOS. These products are available in several levels of compatibility with the GRiD Compass, as explained below.

- Many of the best-selling applications programs have been customized specifically for the GRiD Compass. In this case, you can simply use the product as received.

- The software product, without any customizing, will run without modifications on the GRiD Compass. Examples of this are Scitor's Project Scheduler, MS-BASIC, MS-PASCAL, and MS-FORTRAN.

- The software product manufacturer has included the characteristics of the GRiD Compass as one of the product options. You can simply select that option from the product's installation program. Microsoft's Multiplan fits this category.

- The software product may not have been intended for the GRiD Compass, but the manufacturer supplies one option in the product's installation program that is set for an ANSI terminal. Since the GRiD Compass keyboard and screen characteristics very closely conform to an ANSI terminal, selecting this option will probably let you run the program on the GRiD Compass.

- The software product has no options specifically tailored for the GRiD Compass, but a knowledgeable individual could modify it to run on the GRiD Compass.

In this last case, you will have to perform some sort of installation procedure to customize the product so that it can take full advantage of the capabilities of the GRiD Compass. (You may also want to further customize the take best advantage of the GRiD Compass.

Most installation programs concern themselves with the following characteristics of the computer to be used:

- Keyboard Characteristics: What keys are available on the keyboard, how are they used during program execution, and what codes do the keys generate?

- Cursor Control: What code sequences must be sent to the screen to control cursor movement and display characteristics?

- Graphic Capabilities: What built-in graphic characters and capabilities are provided by the terminal, and what codes are required to generate those graphics?

## DEFINING THE GRiD COMPASS KEYBOARD

In addition to the standard alpha-numeric keys, the GRiD Compass keyboard provides the following special purpose keys:

- Four arrow (or direction) keys for cursor movement

- RETURN, TAB, BACKSPACE and ESC (Escape) keys

- CTRL (Control) and CODE keys

As part of the installation procedure for software products running on MS-DOS, you will be asked to define what keys will be used for certain functions. You may need to supply the code or character sequence generated by depressing a key or combination of keys. The Keyboard Code Chart in Appendix D lists these key definitions.

Table 8-1 provides recommended key settings for common MS-DOS functions.

Table 8-1. Recommendations for GRiD Compass Keyboard Usage

| Function | Frequently Set to | Recommended GRiD Compass Key(s) | Keyboard Code HEX |
|---|---|---|---|
| Carriage Return | Control-M | RETURN | 0D |
| Tab | Control-I | TAB | 09 |
| Backspace | Control-H | BACKSPACE | 08 |
| Cancel or Escape | Control-C | ESC | 1B |
| Cursor | | | |
| Up 1 line | Control-E | UpArrow (↑) | 00 48 |
| Down 1 line | Control-X | DownArrow (↓) | 00 50 |
| Left 1 char. | Control-S | LeftArrow (←) | 00 4B |
| Right 1 char. | Control-D | RightArrow (→) | 00 4D |
| Left 1 word | Control-K | CODE-LeftArrow | D4 |
| Right 1 word | Control-L | CODE-RightArrow | D5 |
| Home | Control-Q | CODE-SHIFT-UpArrow | 00 47 |
| End | | CODE-SHIFT-DownArrow | 00 4F |
| Scroll | | | |
| Up by screen | | CODE-UpArrow | 00 49 |
| Down by screen | | CODE-DownArrow | 00 51 |
| Right by screen | | CODE-SHIFT-RightArrow | D9 |
| Left by screen | | CODE-SHIFT-LeftArrow | D8 |

## GRiD COMPASS CURSOR CONTROL

A software product being installed on MS-DOS to run on the GRiD Compass also needs to know how to manipulate the cursor on the screen. The GRiD Compass MS-DOS screen emulates what is known as an "ANSI terminal". This is Microsoft's recommended standard; many programs are set to these codes as a default.

Table 8-2 lists the cursor control code sequences to which the GRiD Compass responds and other screen-related functions that can be used by programs. Each action requires a sequence of codes that begins with the ASCII Escape code, followed by the ASCII code for a left bracket ([) and one or more additional characters.

The following notation is used in table 8-2:

ESC The ASCII escape character (hex 1B, decimal 27).
# A decimal number in ASCII.
col A decimal number in ASCII designating a column position on the screen. Valid values are 1 (leftmost character position) through 80 (rightmost character position.
row A decimal number in ASCII designating a row position on the screen. Valid values are 1 (top line of screen) through 26.

NOTE: If #, col, or row is not specified, then 0 (zero) is assumed.

With MS-DOS on the GRiD Compass, cursor movement is not permitted beyond the edge of the screen.

Table 8-2. GRiD Compass Cursor and Screen Control Sequences

| ASCII Code Sequence | Action |
|---|---|
| ESC [ # A | Cursor moved up # lines. |
| ESC [ # B | Cursor moved down # lines. |
| ESC [ # C | Cursor moved right # lines. |
| ESC [ # D | Cursor moved left # lines. |
| ESC [ row;col H | Cursor moved to location row,col. |
| ESC [ row;col f | Cursor moved to location row,col. |
| ESC [ 6 n | Device status returns information on the current cursor position. This is in the keyboard queue in the following format: ESC [ row; col R. |
| ESC [ s | Current cursor position is saved. |
| ESC [ u | Cursor is restored to previously saved position. |
| ESC [ 2 J | Erase screen and move cursor to home position. |
| ESC [ 0 K | Erase to end of line including cursor position. |
| ESC [ 0 m | Turn inverse video off. |
| ESC [ 7 m | Turn inverse video on. |
| ESC [ 3;3 z | Turn cursor on (steady underline). |
| ESC [ 3;4 z | Turn cursor off (no cursor displayed). |

## GRiD COMPASS GRAPHIC CAPABILITIES

The graphic capabilities available when running MS-DOS on the GRiD Compass include the standard ASCII alpha-numeric characters and some shapes that are useful for graphic effects. Figure 8-1 shows the complete set of characters and shapes in the font along with the ASCII hexadecimal code for each.

NOTE: Any hex code larger than A8 will produce the last shape in the font.

The shapes that are most frequently used by software packages, and the hexadecimal code for each, have been extracted from Figure 8-1 and are listed below for your convenience. These may be used by software packages to draw boxes, windows, and so on.

| SHAPE | Hexadecimal Code |
|---|---|
| Upper left corner | A0 |
| Upper right corner | A1 |
| Lower right corner | A2 |
| Lower left corner | A3 |
| Horizontal bar | A4 |
| Vertical bar | A5 |
| Top half of a plus sign (+) | A6 |
| Bottom half of a plus sign (+) | A7 |

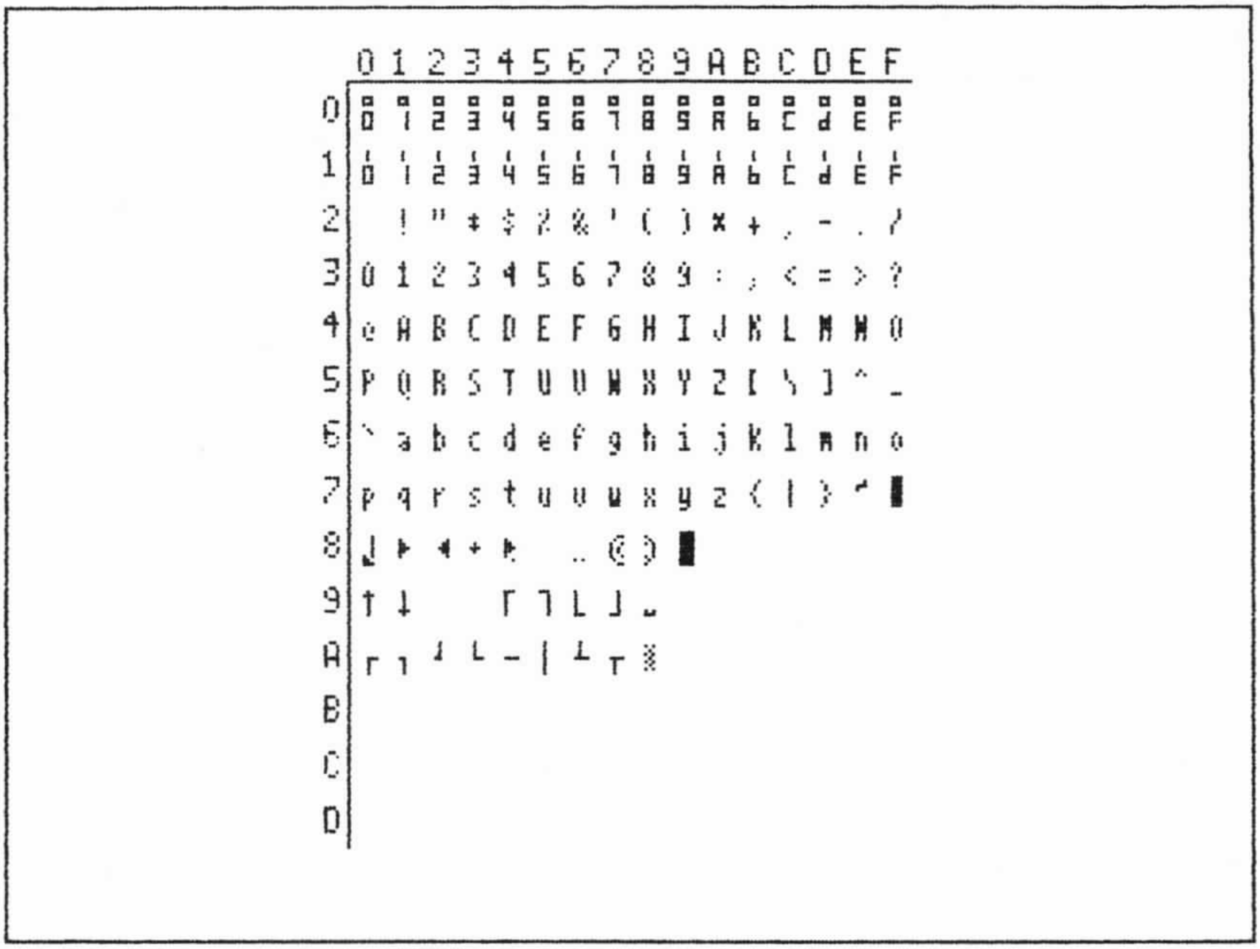

0 1 2 3 4 5 6 7 8 9 A B C D E F
0
1
2 ! " # $ % & ' ( ) * + , - . /
3 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
4 @ A B C D E F G H I J K L M N O
5 P Q R S T U V W X Y Z [ \ ] ^ _
6 ` a b c d e f g h i j k l m n o
7 p q r s t u v w x y z { | } ~
8
9
A
B
C
D

Figure 8-1. The GRiD Compass MS-DOS Display Font

## CHAPTER 9. BASIC AND THE GRiD COMPASS

This chapter reviews compatibility issues between various versions of the BASIC language. If you have both operating systems -- MS-DOS and GRiD/OS -- you have three different levels of BASIC available on your GRiD Compass.

| BASIC available on the GRiD Compass | Operating System Required |
|---|---|
| MS-BASIC | MS-DOS |
| Business BASIC | MS-DOS |
| GRiDBASIC | GRiD/OS |

The level of compatibility between BASIC programs written on other machines and the GRiD Compass depends on a number of considerations.

The following is a review of a number of different "BASICs" with some comments about their intercompatibility.

| | |
|---|---|
| MS-BASIC............ | available on Compass, IBM PC, and many other micros |
| Business BASIC...... | available on Compass, IBM PC, and other MS-DOS micros |
| IBM BASIC........... | available on the IBM PC only |
| IBM Advanced BASIC.. | available on the IBM PC only |
| GW-BASIC............ | available on an OEM basis only |
| C-BASIC............. | available on CPM machines |
| GRiDBASIC........... | available on the GRiD Compass only |

## BASIC COMPARISONS

The paragraphs that follow compare various versions of BASIC to the MS-BASIC Compiler and highlight obvious areas of incompatibility between MS-BASIC and other versions. We'll begin with an overview of MS-BASIC.

## MS-BASIC COMPILER AND INTERPRETER

This is the standard version of BASIC from Microsoft. There are over a million copies installed on various machines. A standard generic version will run on all MS-DOS machines and requires no changes to run on the different machines. It is also available as an interpreter.

MS-BASIC will run on the GRiD Compass (and IBM PC). Software can be ported between machines running the same MS-BASIC in its compiled form. The following commands of the interpreter version of MS-BASIC are not supported by the compiler: BLOAD, BSAVE, CHAIN (when using ALL, MERGE, DELETE, OR <Line#> options) and RUN (when using "R" option). Also, INKEY$ used with the compiler will return a null (0) followed by the appropriate key code (for example, for the function keys). The interpreter strips out the null characters.

MS-BASIC programs are 100% transportable if no machine-dependent features have been used. That is, if the "Peek" and "Poke" commands have not been used, it should run on the GRiD Compass and IBM PC (for example) with no changes.

WHAT THIS MEANS TO USERS:

> Using MS-BASIC, you can develop software to run unchanged on the GRiD Compass and the IBM PC (and many other MS-DOS machines) without any changes.

## BUSINESS BASIC COMPILER

This enhanced version of BASIC from Microsoft runs on all MS-DOS machines. It is a superset of MS-BASIC available as a compiler.

The following Business BASIC statements and compiler commands are not supported by the MS-BASIC compiler:

COMMAND$<br>
LCASE$<br>
UCASE$<br>
TRIM$<br>
LTRIM$<br>
RTRIM$<br>
SADD<br>
MATCH<br>
RENAME<br>
SIZE<br>
CREATE<br>
CONSOLE<br>
LPRINTER<br>
POS<br>
LINE READ<br>
ON END GOTO<br>
READ#<br>
REDIM<br>
STATIC<br>
SUB<br>
END SUB<br>
WRITE USING<br>
LBOUND<br>
UBOUND<br>
CONCHAR<br>
CONSTAT<br>
DELETE

Additionally, none of the Business BASIC compiler metacommands (e.g. $DYNAMIC, $MODULE) are supported by MS-BASIC.

Business BASIC is fully compatible with MS-BASIC at the text file (ASCII) level--that is, an MS-BASIC file, in text form, can be compiled in Business BASIC. It can then be enhanced, but it would no longer be compatible with MS-BASIC.

Business BASIC offers a C-BASIC conversion facility that will allow one to convert a C-BASIC source file into a Business BASIC source file.

Business BASIC programs are 100% transportable if no machine dependent features have been used. That is, if the "Peek" and "Poke" commands have not been used, it should run on the Compass and IBM PC (for example) with no changes.

WHAT THIS MEANS TO USERS:

> Using Business BASIC, you can develop software to run on the GRiD Compass and the IBM PC (and other MS-DOS machines) without any changes. Using Business BASIC, users can also convert C-BASIC to run on the Compass.

## IBM BASIC AND ADVANCED BASIC

The IBM Personal Computer comes with BASIC in two variations: standard BASIC and Advanced BASIC (BASICA). Both are versions of Microsoft's MS-BASIC that have been customized with additional commands and enhancements.

Both are compatible with MS-BASIC at the text file (ASCII) level. An IBM BASIC file unchanged on the GRiD Compass can be used with the MS-BASIC interpreter or compiler if none of the following commands and statements have been used:

```
CLS          LOAD          BLOAD        PSAVE         BEEP
CHDIR        CIRCLE        COLOR        COM           CSRLIN
DATE$        PSET          PRESET       RMDIR         SCREEN
SOUND        STICK         DRAW         FILES         KEY
LINE         LOCATE        LOF          MKDIR         MOTOR
ON COM       ON KEY        ON PEN       ON PLAY       ON STRIG
ON TIMER     OPEN "COM     PAINT        PEN           PLAY
PMAP         POINT         SYSTEM       TIME$         VARPTR$
VIEW         WINDOW
```

NOTE: The three most commonly used statements in the list above are CLS, BEEP and LOCATE. These statements can be duplicated in MS-BASIC as follows:

```
CLS       PRINT CHR$(27%)+'[2J';
BEEP      PRINT CHR$(7%)
LOCATE    PRINT CHR$(27%)+'['+R$+';'+C$+'H';
```

...where R$ stands for Row; C$, Column. These values must be entered without leading spaces.

MS-BASIC only recognizes the following standard notations for relational operators: <> (not equal), <= (less than or equal), and >= (equal or greater than).

The following PC BASIC relational operators are not supported by MS-BASIC: >< (not equal), =< (equal or less than), and => (equal or greater than).

Additionally, IBM BASIC tokenized version of a BASIC program may not be completely compatible with MS-BASIC in token form. Because of this, a BASIC program written on the IBM PC using IBM BASIC should be moved to the GRiD Compass as a text file.

WHAT THIS MEANS TO USERS:

> Programs written in IBM PC BASIC can be moved to the GRiD Compass as a text file. IBM PC BASIC programs in interpretable form may not run directly on the GRiD Compass.

## GW-BASIC

GW-BASIC is an enhanced version of Microsoft's MS-BASIC available only to OEM customers. It requires customization for each machine.

It is not currently available from GRiD for use on the GRiD Compass.

## C-BASIC

C-BASIC is a version of BASIC from Digital Research designed for use with the CP/M Operating System.

WHAT THIS MEANS TO USERS:

It is not currently available on the GRiD Compass.
It is compatible with MicroSoft's Business BASIC which is available on the GRiD Compass.

## GRiDBASIC

GRiDBASIC is a superset of MS-BASIC with some exceptions. Programs written in MS-BASIC will run on the GRiD Compass under GRiD/OS using GRiDBASIC, but the following commands must not be used.

FRE HEX LPOS OCT SPC
CALL CHAIN CLEAR CLOAD TRON
TROFF WAIT WIDT WRITE WRITE#
ERR ERL ERROR BLOAD BSAVE
LPRINT NAME NULL LINE INPUT LINE INPUT#
POS USR VALPTR COMMON CSAVE
DEF FN DEFINT DEFSNG DEFDBL DEFSTR
DEFUSR DEFSEG ERASE RESUME RUN
SWAP

GRiDBASIC is 100% compatible with MS-BASIC at the text file (ASCII) level. Thus, an MS-BASIC file can be used in GRiDBASIC. If it is enhanced with GRiDBASIC functions, the file will no longer be compatible with MS-BASIC.

NOTE: GRiDBASIC arrays are numbered from 1 to N. MS-BASIC arrays are numbered from 0 to N-1 in the normal default state. To ensure compability, use OPTION BASE 1 in MS-BASIC.

Table 9-1. Comparison of BASIC Versions

| Runs on<br>Operating System: | GRiD<br>BASIC | MS-<br>BASIC | Bus.<br>BASIC | IBM<br>BASIC | IBM<br>BASICA | PC-<br>BASIC |
|---|---|---|---|---|---|---|
| CCOS | * | | | | | |
| MS-DOS | | * | * | | | |
| PC-DOS | | * | * | * | * | |
| CP/M | | | | | | * |
| **Compatible With:** | | | | | | |
| GRiD BASIC | - | * | | * | | |
| MS-BASIC | * | - | | * | | |
| Bus. BASIC | | * | - | * | | * |
| IBM BASIC | * | * | | - | | |
| IBM BASICA | | * | | * | - | |
| C BASIC | | | * | | | - |

NOTE: Compatibility between BASIC versions may be upwards only.

# APPENDIX A. DIRECT SCREEN WRITING

This appendix discusses technical details and tips that may be of value to system programmers using MS-DOS on the GRiD Compass.

## USING DIRECT CONSOLE OUPUT

Sometimes an application program requires the ability to send information to the screen faster than can be accomplished by going through MS-DOS. When more speed is needed, you can bypass MS-DOS and go straight to the subroutine in the BIOS that handles the screen. This can result in time savings of more than 50%. For example, when MS-DOS Function Call 2 is used to send one character at a time to the screen through the operating system, it takes about 40 seconds to send 16,000 characters. By going straight to the subroutine in the BIOS that handles the screen (bypassing the operating system), 16,000 characters can be sent in about 17 seconds.

Inside the BIOS there is a table of pointers to various subroutines. One of these pointers is to the subroutine that actually draws characters on the screen (this routine also handles the ANSI escape sequences). This character display subroutine requires that the character to be displayed be loaded in the AL register. The contents of all registers and flags are preserved by the subroutine.

NOTE: If you bypass the operating system, remember that the BIOS does not check the keyboard for depression of the CTRL-C key sequence.

The table of subroutine pointers in BIOS has the following structure:

```
     PublicData    STRUC
       reserved    DD       4 DUP (?)
       reserved    DD       ?
       reserved    DD       3 DUP (?)
       pCharOut    DD       ?
     PublicData    ENDS
```

Interrupt Vector 80H (located at absolute address 200H) contains the address of this PublicData table. Thus, to get the address of the CharOut subroutine, you could use code similar to the following:

```
     MOV   AX, 0
     MOV   ES, AX
     LDS   BX, ES:DWORD PTR 200H   ; get address of public table
     MOV   AX, DS:WORD PTR [BX].pCharOut  ; offset of CharOut in AX
     MOV   BX, DS:WORD PTR [BX].pCharOut+2  ; seg of CharOut in BX
```

```
; store AX, BX to a pointer that is used for indirect calls to
CharOut

     MOV  CS:WORD PTR CharOut, AX
     MOV  CS:WORD PTR CharOut+2, BX

; call the CharOut routine

     MOV  AL, 'A'
     CALL CS:CharOut
      .
      .
      .
```

# APPENDIX B. GRAPHICS USING BIT-MAPPED VIDEO

This appendix summarizes the technique of bit-mapping that is used to produce the display on the GRiD Compass. A collection of assembly language routines at the end of the appendix offers a programmer the fundamental tools to create graphic displays and patterns.

## VIDEO DISPLAY ARCHITECTURE

For each pixel that appears on the display screen a bit position is reserved within a portion of RAM that is dedicated to the video display. This one-to-one correspondence allows unlimited manipulation of the video screen graphic displays. Bytes written to the video memory area directly form the bit pattern that is shifted out to produce the display.

The screen dimensions extend 320 pixels horizontally and 240 pixels vertically. Although the bit patterns are written to the video memory 8 or 16 bits at a time by the CPU, they are always accessed 16 bits at a time by the video display circuitry. Twenty of these 16-bit words compose each row of pixels (slice) of the display screen. The full 240 rows of video display require 4800 words which are stored between the RAM memory addresses 00400 through 0297F hex.

Figure B-1 illustrates the manner in which the bit-mapping relates

RAM memory locations to the bit positions on the display screen.

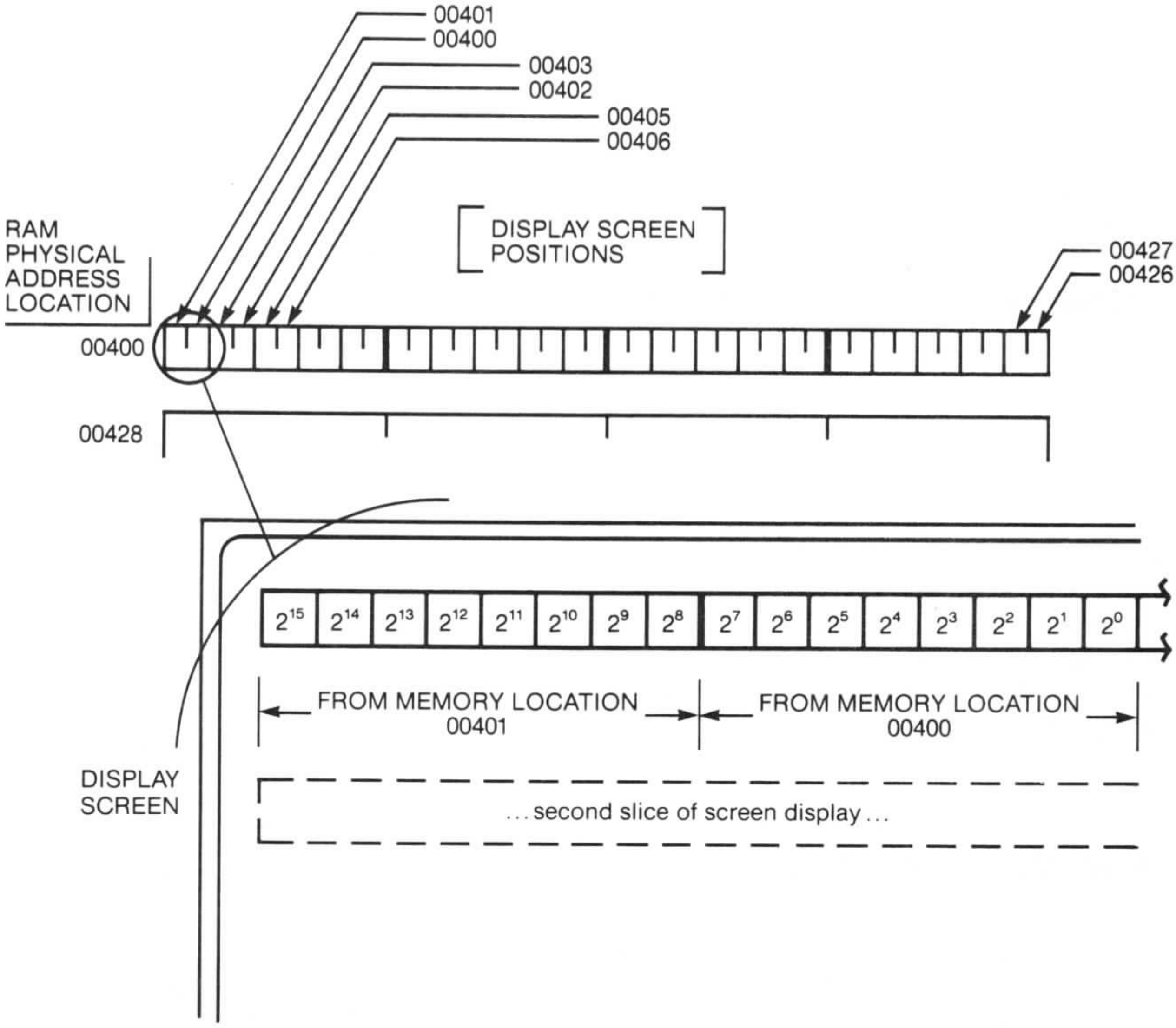


Figure B-1. Bit map for Video Display

Figure B-2 illustrates how a word stored at two consecutive memory locations appears when written to the video display.

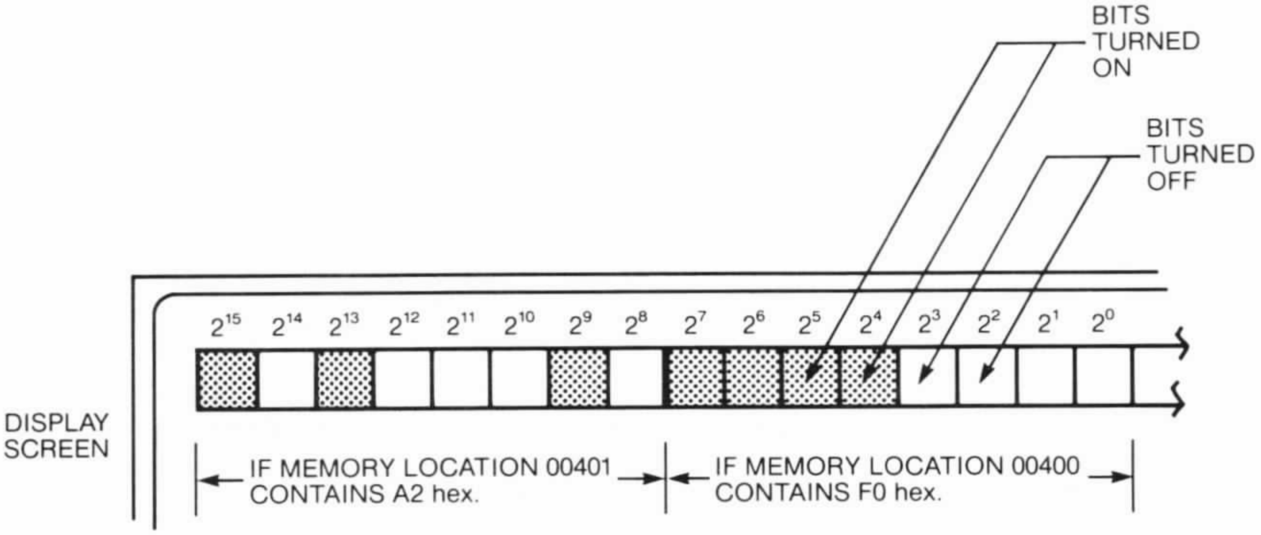


Figure B-2. Example of Bit-Mapping

Writing bytes to the video portion of memory requires no synchronization with the video timing circuitry; the computer's hardware ensures that actual timing matters are transparent to the programmer. By mastering a set of fundamental assembly language routines, the programmer can turn on and off the desired bits to create any possible pattern. Several of these routines follow, including basic descriptions of how to draw a line between two designated points, how to invert the bit pattern, how to erase a line, and how to move pixels about on the display screen.

Additional information concerning algorithms for graphic display can be obtained from the reference, Principles of Interactive Computer Graphics by Newman and Sproull, published by McGraw-Hill in 1979.

Figure B-3 shows addresses in the video memory corresponding with positions on the display screen.

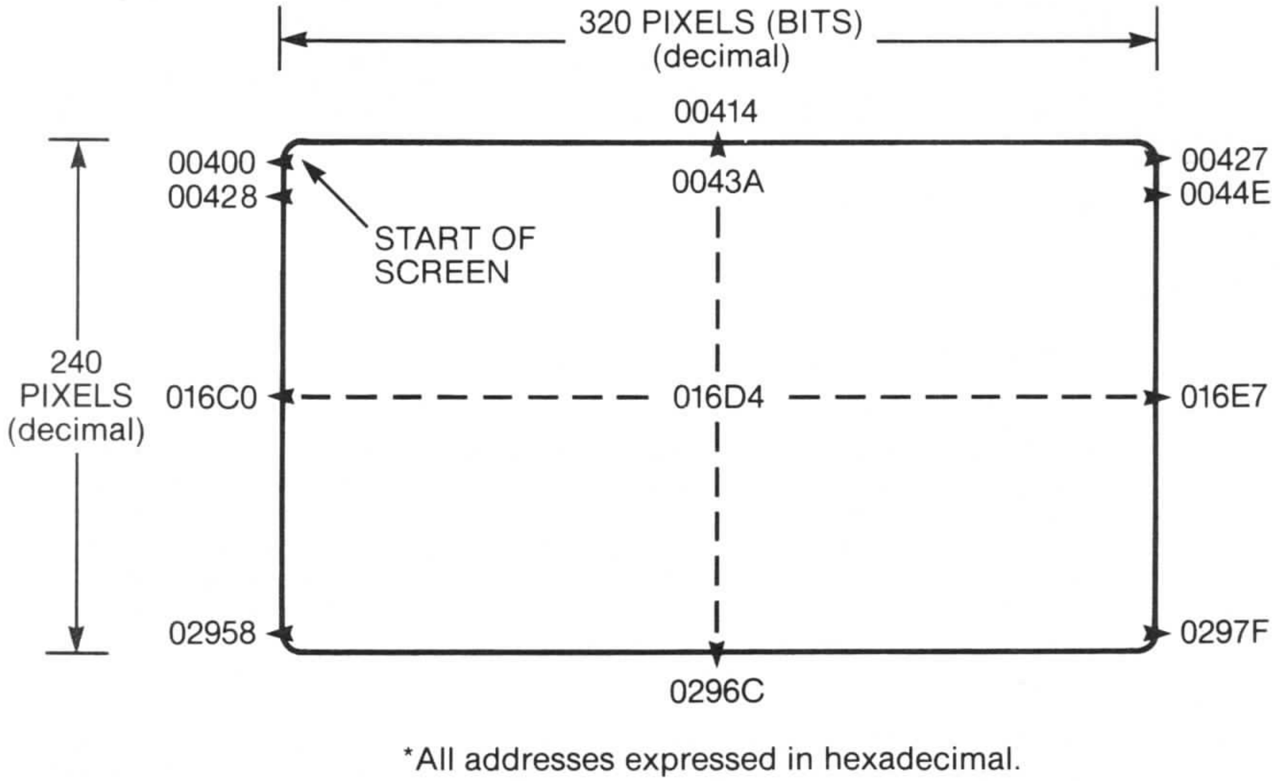


Figure B-3. Address Correspondence for Video Display.

The following assembly language routines can be used individually or combined to form intricate graphic patterns.

```
NAME  Grafix

CGROUP GROUP CODE

;**************************************************
;*   PUBLIC declaration of Graphics Procedures
;**************************************************

PUBLIC  GfxSetPixel, GfxClrPixel, GfxInvertPixel, GfxTestPixel
PUBLIC  GfxDrawLine, GfxEraseLine, GfxInvertLine

;**************************************************
;*  constant definitions for screen width, height,
;*  and masks for facilitating working with
;*  bit patterns
;**************************************************

zero        EQU  0
scrWidth    EQU  320
scrHeight   EQU  240
wordWidth   EQU  scrWidth/16     ; screen width in words
byteWidth   EQU  wordWidth*2
evenMask    EQU  0FFFEH          ; make a word even
true        EQU  0FFFFH

display  SEGMENT AT 40H
display  ENDS
```

```
CODE SEGMENT PUBLIC 'CODE'
ASSUME  CS:CGROUP, DS:display

;**************************************************
;*
;*    GfxSetpixel(x, y: Word)
;*
;* This will set the pixel at X = horizontal
;* and Y = vertical.
;* (0,0) is the upper lefthand corner.
;*
;* REGISTERS CHANGED: AX, BX, CX, DX, ES, DI, SI
;*
;**************************************************

GfxSetPixel PROC NEAR
  PUSH  BP
  PUSH  DS
  MOV   BP, SP

  MOV   CX, [BP+6H]           ; CX = y
  MOV   BX, [BP+8H]          ; BX = x
  MOV   SI, 8000H
  CALL  SetUpPixel

  OR    [BX], SI             ; set the pixel

  POP   DS
  POP   BP
  RET   4
GfxSetPixel ENDP
```

```
;***************************************************
;*
;*    GfxClrPixel(x, y : word)
;*
;* This is exactly the same as SetPixel except
;* for the last two statements
;*
;*****************************************************

GfxClrPixel PROC NEAR
  PUSH  BP
  PUSH  DS
  MOV   BP, SP

  MOV   CX, [BP+6H]
  MOV   BX, [BP+8H]
  MOV   SI, 7FFFH
  CALL  SetUpPixel

  AND   [BX], SI

  POP   DS
  POP   BP
  RET   4
GfxClrPixel ENDP
```

```
;***************************************************
;*
;*    GfxInvertPixel(x, y : word)
;*
;* This is exactly the same as SetPixel except
;* for the last two statements
;*
;*****************************************************

GfxInvertPixel PROC NEAR
  PUSH  BP
  PUSH  DS
  MOV   BP, SP

  MOV   CX, [BP+6H]
  MOV   BX, [BP+8H]
  MOV   SI, 8000H
  CALL  SetUpPixel

  XOR   [BX], SI

  POP   DS
  POP   BP
  RET   4
GfxInvertPixel ENDP
```

```
;***************************************************
;*
;*    GfxTestPixel(x, y : word)
;*
;* This is exactly the same as SetPixel except
;* for the last two statements
;*
;*****************************************************

GfxTestPixel PROC NEAR
  PUSH  BP
  PUSH  DS
  MOV   BP, SP

  MOV   CX, [BP+6H]
  MOV   BX, [BP+8H]
  MOV   SI, 8000H
  CALL  SetUpPixel

  MOV   AX, [BX]       ;Get the word
  AND   AX, SI         ;Mask out all other bits
  JZ    NotSet         ;do nothing if not set
  MOV   ASet to true if bit is set.
NotSet:

  POP   DS
  POP   BP
  RET   4
GfxTestPixel ENDP
```

```
;**************************************************
;*
;*     SetUpPixel
;*
;*   entry:  BX = x
;*           CX = y
;*           SI = original mask
;*
;*   Does the setup for Set- and Clr- Pixel.
;*
;*   exit:  BX ^ word in display
;*          ES ^ start of display buffer
;*          SI = rotated mask
;*
;*************************************************

SetUpPixel PROC  NEAR
  MOV  AX, display
  MOV  DS, AX                 ; DS ^ start of display

  MOV  AX, CX
  MOV  DX, byteWidth
  MUL  DX
  MOV  CX, AX                 ; CX = CX * byteWidth

  MOV  AX, BX                 ; AX = x
  SAR  BX, 1
  SAR  BX, 1
  SAR  BX, 1
  AND  BX, evenMask           ; BX = EVEN(x DIV 8)
  ADD  BX, CX                 ; BX ^ word in display

  AND  AX, 000FH              ; AX = x MOD 16
  MOV  CX, AX
  ROR  SI, CL                 ; rotate to position

  RET
SetUpPixel ENDP
```

```
;*********************************************************
;*
;*    GfxDrawLine(x1, y1, x2, y2)
;*
;* This will draw a line between the two points (x1, y1) and
;* (x2, y2).  It uses the DDA algorithm.  The act of
;* drawing a line is split into two cases for efficiency.
;*
;*  case AB: dx >= dy
;*  case CD: dx < dy
;*
;* During the inner loop of each case, the registers mean:
;*
;*  AX = dx              BX = dy
;*  CX = loop counter    DX = temp
;*  SI = mask            DI ^ display buffer
;*  BP = +- wordWidth
;*
;********************************************************

GfxDrawLine PROC NEAR
  PUSH  BP
  PUSH  DS
  MOV   BP, SP

  MOV   DX, [BP+06H]          ; DX = y2
  MOV   CX, [BP+08H]          ; CX = x2
  MOV   BX, [BP+0AH]          ; BX = y1
  MOV   AX, [BP+0CH]          ; AX = x1

  CMP   AX, CX               ; x1 <= x2 ?
  JLE   Line10
  XCHG  AX, CX
  XCHG  BX, DX               ; now x1 <= x2
Line10:

  SUB   CX, AX               ; CX = dx ( >= 0)
  SUB   DX, BX               ; DX = dy
  MOV   SI,8000H             ; SI = original mask
  CALL  AInitLine
  OR    [DI],SI
  CMP   AX, BX               ; dx < dy ?
  JL    CaseCD               ; yes -> case CD

CaseAB:            ; case AB
  MOV   CX, AX               ; loop 'dx' times
  MOV   DX, AX
  CMP   DX, 1
  JLE   NoShift
  SAR   DX, 1
NoShift:
  NEG   DX                    ; DX = -dx/2; DX > 0!!
  JCXZ  LoopDone

LoopOnAB:
  ADD   DX, BX                ; temp += dy
  JS    A10
  SUB   DX, AX                ; temp -= dx
  ADD   DI, BP                ; y += wordWidth
A10:

  ROR   SI, 1                 ; x += 1
  JNC   SameX
  ADD   DI, 2                 ; DI ^ next word
SameX:
  OR    [DI], SI              ; set the bit
  LOOP  LoopOnAB
  JMP   LoopDone

CaseCD:           ; case CD
  MOV   CX, BX                ; loop 'dy' times
  MOV   DX, BX
  CMP   DX, 1
  JLE   NoShift2
  SAR   DX, 1
NoShift2:
  NEG   DX                    ; DX = -dy / 2
  JCXZ  LoopDone

LoopOnCD:
  ADD   DX, AX                ; temp += dx
  JS    C10
  SUB   DX, BX                ; temp -= dy
  ROR   SI, 1                 ; x += 1
  JNC   C10
  ADD   DI, 2                 ; DI ^ next word

C10:
  ADD   DI, BP                ; next row of bytes
  OR    [DI], SI              ; set the bit
  LOOP  LoopOnCD

LoopDone:
  POP   DS
  POP   BP
  RET   8
GfxDrawLine ENDP
```

```
;**********************************
;*                                *
;*  GfxEraseLine(x1, y1, x2, y2)  *
;*                                *
;**********************************

GfxEraseLine PROC NEAR
  PUSH  BP
  PUSH  DS
  MOV   BP, SP

  MOV   DX, [BP+06H]          ; DX = y2
  MOV   CX, [BP+08H]          ; CX = x2
  MOV   BX, [BP+0AH]          ; BX = y1
  MOV   AX, [BP+0CH]          ; AX = x1

  CMP   AX, CX               ; x1 <= x2 ?
  JLE   ELine10
  XCHG  AX, CX
  XCHG  BX, DX               ; now x1 < x2

ELine10:
  SUB   CX, AX               ; CX = dx ( >= 0 )
  SUB   DX, BX               ; DX = dy
  MOV   SI, 7FFFH            ; SI = original mask
  CALL  AInitLine
  AND   [DI], SI
  CMP   AX, BX               ; dx < dy ?
  JL    ECaseCD              ; yes -> case CD

ECaseAB:           ; case AB
  MOV   CX, AX               ; loop 'dx' times
  MOV   DX, AX
  CMP   DX, 1
  JLE   ENoShift
  SHR   DX, 1
ENoShift:
  NEG   DX                   ; DX = - dx / 2
  JCXZ  ELoopDone

ELoopOnAB:
  ADD   DX, BX               ; temp += dy
  JS    ESameY               ; next y value ?
  SUB   DX, AX               ; temp -= dx
  ADD   DI, BP               ; y += wordWidth
ESameY:
  ROR   SI, 1                ; x += 1
  JC    ESameX               ; on to next word?
  ADD   DI, 2                ; yes
ESameX:
  AND   [DI], SI             ; erase the bit
  LOOP  ELoopOnAB
  JMP   ELoopDone            ; all finished

ECaseCD:           ; case CD
  MOV   CX, BX               ; loop 'dy' times
  MOV   DX, BX
  CMP   DX, 1
  JLE   ENoShift2
  SHR   DX, 1
ENoShift2:
  NEG   DX                   ; DX = -dy / 2
  JCXZ  ELoopDone
ELoopOnCD:
  ADD   DX, AX               ; temp += dx
  JS    EC10
  SUB   DX, BX               ; temp -= dy
  ROR   SI, 1                ; x += 1
  JC    EC10
  ADD   DI, 2                ; DI ^ next word
EC10:
  ADD   DI, BP               ; y += increment
  AND   [DI], SI             ; clear the bit
  LOOP  ELoopOnCD

ELoopDone:
  POP   DS
  POP   BP
  RET   8
GfxEraseLine ENDP
```

```
;****************************************
;*
;*    GfxInvertLine(X, Y, X1, Y1)
;*
;****************************************

GfxInvertLine PROC NEAR
  PUSH  BP
  PUSH  DS
  MOV  BP,SP
  MOV  AX,display
  MOV  DS,AX

  MOV   DX, [BP+06H]          ; DX = y2
  MOV   CX, [BP+08H]          ; CX = x2
  MOV   BX, [BP+0AH]          ; BX = y1
  MOV   AX, [BP+0CH]          ; AX = x1

  CMP  AX, CX    ; IS X <= X1
  JLE  XLine10    ; YES -> JUMP

  XCHG  AX, CX
  XCHG  BX, DX    ; NOW X < X1
XLine10:
  SUB  CX, AX    ; CX = dx (dx >= 0)
  SUB  DX, BX    ; DX = dy

  MOV   SI, 8000H           ; SI = original mask
  CALL  AInitLine
  XOR   [DI], SI

  CMP  AX, BX    ; IS dx < dy ?
  JL  XCaseCD    ; YES -> GOTO CASES C,D

XCaseAB:      ; MUST BE CASE A OR B
  MOV  CX, AX    ; LOOP dx TIMES
  MOV  DX, AX
  CMP   DX, 1
  JLE   XNoShift
  SHR   DX, 1
XNoShift:
  NEG  DX    ; DX = - dx /2
  JCXZ  XLoopDone
XLoopOnAB:
  ADD  DX, BX    ; temp += dy
  JS  XA10    ; JUMP IF RESULT < 0
  SUB  DX, AX    ; temp -= dx
  ADD  DI,BP    ; Y += INC Y
XA10:
  ROR  SI,1    ; X += 1
  JNC  XA20
  ADD  DI,2    ; DI ^ next word
XA20:
  XOR  [DI],SI    ; SET THE BIT
  LOOP  XLoopOnAB

  JMP   XLoopDone

XCaseCD:       ; MUST BE CASE C OR D
  MOV  CX, BX    ; LOOP dy TIMES
  MOV  DX, BX
  CMP   DX, 1
  JLE   XNoShift2
  SHR   DX, 1
XNoShift2:
  NEG  DX    ; DX = -dy / 2
  JCXZ  XLoopDone
XLoopOnCD:
  ADD  DX, AX    ; temp += dx
  JS  XC10    ; JUMP IF RESULT < 0
  SUB  DX, BX    ; temp -= dy
  ROR  SI,1    ; X += 1
  JNC  XC10
  ADD  DI,2    ; DI ^ next word
XC10:
  ADD  DI,BP    ; Y += INC Y
  XOR  [DI],SI    ; SET THE BIT
  LOOP  XLoopOnCD

XLoopDone:
  POP  DS
  POP  BP
  RET  8
GfxInvertLine ENDP
```

```
;*************************************************
;*
;*    AInitLine
;*
;*  entry:  AX = x
;*          BX = y
;*          CX = dx     (dx >= 0)
;*          DX = dy
;*          SI = original mask
;*
;*  exit:   SI = rotated mask
;*          DI ^ word in display
;*          AX = dx
;*          BX = dy
;*          BP = +- wordWidth
;*
;************************************************

AInitLine PROC   NEAR
  PUSH  CX
  PUSH  DX

  PUSH  AX                ; save during multiply

  MOV   AX, display       ; screen is data segment
  MOV   DS, AX

  MOV   AX, BX
  MOV   CX, byteWidth
  MUL   CX
  MOV   DI, AX            ; DI = y * byteWidth
  POP   AX

  MOV   CX, AX
  SAR   CX, 1
  SAR   CX, 1
  SAR   CX, 1
  AND   CX, evenMask      ; CX = EVEN(x DIV 8)
  ADD   DI, CX            ; DI ^ word in display

  AND   AX, 000FH         ; AX = x MOD 16
  MOV   CX, AX
  ROR   SI, CL            ; rotate to starting pos

  POP   BX                ; BX = dy
  POP   AX                ; AX = dx

  MOV   BP, byteWidth
  CMP   BX, zero
  JGE   DYGE0             ; dy >= 0
  NEG   BP
  NEG   BX              ; dy = |dy|
DYGE0:
  RET
AInitLine ENDP

CODE ENDS
  END
```

## APPENDIX C. MODEM AND SERIAL DRIVERS

The GRiD Compass is equipped with two asynchronous, serial communications channels. The COM1 path accesses the SERIAL connector and may be configured for a serial printer, an asynchronous modem, or any other asynchronous serial device which meets RS-232-C or RS-422 conventions. The COM2 path provides a direct interface with telephone lines through the GRiD Compass's internal modem.

Driver routines loaded during normal MS-DOS initialization set baseline values that may be altered by writing character strings to the COM1STAT path for the accessory (serial) port, and the COM2STAT path for the line (modem) port.

An emulator routine written for the internal modem mimics the functioning of the Hayes Smartmodem 1200 (in the COM2 path), duplicating most of the Smartmodem's command set. Those areas where significant differences exist are detailed in the following description.

This section describes the normal baseline values for each of the communication channels and the means by which a programmer can customize the port for a specific application.

### MS-DOS SERIAL DRIVER

Normal MS-DOS I/O operations can be performed to the COM1 path, accomplishing communications with whatever physical device is connected to the serial port.

At initialization time, the following values describe the baseline operation under MS-DOS:

7 data bits, 1 stop bit, even parity,
300 baud,
Line feed insertion turned off,
700 millisecond maximum wait for character during READ,
1000 millisecond maximum wait for Clear to Send on WRITE,
1000 millisecond maximum wait for Data Set Ready on WRITE,
Data Carrier Detect ignored on WRITE,
Interrupts enabled for detection of DCD, CTS, TXRDY, RXRDY,
Asynchronous mode,
DTR active (for answering phone from modem),
RTS active
HighSpeedDelay active (a brief delay at speeds greater than 1200 bps).

These values correspond to the conditions created by issuing the default OPEN "COMn:" command from PC-BASIC on an IBM PC.

NOTE: The activation of HighSpeedDelay permits an HP7470A plotter to accept data transfers at relatively high speeds without producing communication errors.

### Command Formats

The following structure applies to all commands written to the COM1STAT path in order to configure the COM1 path.

Lower case letters are optional.

The number of milliseconds used in timed operations must range between 1 and 65535. The letter 'n' represents this value. A '0' signifies that the signal should be ignored.

Any text enclosed in brackets represents commentary.

The equal sign -- '=' -- is necessary to the command structure. All commands must be strung together using commas.

Generally, the commands are recognized by the driver routine by their first letter, which must be capitalized. Those commands that require more identification are illustrated in the following list with all necessary letters capitalized.

Baud: This command sets the rates for data communication transfers.

```
B = 50
     75
    110
    134.5
    150
    300
    600
   1200
   1800
   2000
   2400
   3600
   4800
   7200
   9600
  19200
```

Parity:  This command determines the nature of the parity bit, if used.  The letter 'O' signifies Odd, 'E' signifies Even, and 'N' signifies None.

```
P = O
    E
    N
```

Stopbits: This command defines the structure of the stop bits used to determine the extent of a data word.

```
S = 1
    1.5
    2
```

Handshaking signals: The following commands determine factors controlling the data exchange operation on the serial line.  The value 'n' expresses a time in milliseconds.  The first three commands control the time to wait for the designated signal to come active after requesting a WRITE.  The driver routine evaluates the commands in the sequence listed.

```
CTS = n
DSR = n
```

The following values control the delay acceptable when servicing a READ request.

```
DCD  = n
CHAR = n
```

The following commands can be used to control an asynchronous data-exchange operation.

```
RTS = on
      off
DTR = on
      off
```

NOTE: If the RTS command is issued, then CTS will be set to be ignored during a WRITE unless the user issues a CTS command to specify the CTS timeout. This corresponds to the signal control logic of PC-BASIC.

Querystatus: This command causes a status buffer to be formatted reflecting current information on the state of the device attached to the communication channel and the state of the input buffer. Querystatus must be followed by a READ to the status register for 5 bytes in order to obtain the status information.

```
 Q
```

The returned modemstatus word has the following bit definitions:

```
Bit 0 = DCD
Bit 1 = CTS
Bit 2 = DSR
Bit 3 = Parity error
Bit 4 = Overflow error
Bit 5 = Framing error
```

All error condition flags are turned off when the driver is loaded. When an error occurs, the appropriate internal flag is set and latched. These flags become available whenever the status query takes place. All flags are then reset.

Flush: This command purges the input buffer of any data. This can be useful during diagnostic and test routines.

```
F
```

DAtabits:  This command controls the number of data bits composing each transmitted byte.

```
   DA = 5
        6
        7
        8
```

Linefeed stuff: This command enables line-feed stuffing causing the driver to insert a line-feed character (hex 0A) after every carriage return (hex 0D) encountered in the data stream.

```
   L = on
       off
```

Init: This command reestablishes the baseline conditions set during system initialization.   All internal error flags in the driver are reset.

Init should be issued before accessing the serial port if any other communication line activity has taken place since initialization.

```
   I
```

Assignfifo:  This command points the driver to a designated FIFO (first-in, first-out) to obtain input data.   Errors are not returned.  Rather, if any error is detected in the input data, the driver will automatically default to the internal 256-byte FIFO.

If an 'A','A=', or 'A=00h' is entered, the driver will also return to the internal FIFO.   Additionally, if the word length value entered is a null, the driver directs all port input to the internal FIFO.   When the INIT command is generated, the driver again points itself to the internal FIFO.

The command format is:

```
   A = <DWORD pointer to FIFO><word length value>.
```

NOTE: The DWORD -- double word pointer -- is expressed in the format 'offset/segment/', the same manner a pointer is handled in the 8086 microprocessor.

Highspeeddelay:  This command produces a readjustment of the driver transmission rate to allow peripheral devices extra time to respond to the transmitted data. At speeds greater than 1200 bps, a small delay is inserted by the driver whenever the HIGHSPEEDDELAY is activated. This delay is turned on during system initialization.

H = on
    off

## MS-DOS MODEM DRIVER

Communication through the COM2 path occurs by way of the internal modem, initialized under MS-DOS to emulate the Hayes Smartmodem 1200. The modem driver is also accessible through a secondary communication path, COM2STAT. Status and data format information is delivered via this route, and the modem may also be reconfigured to adapt to specialized communication needs.

The baseline values held by the driver routine following initialization are the same as those used by the Smartmodem as default values. The emulator assumes the following configuration switch settings for the Smartmodem are in effect:

Switch 2 -- Down: Issue digital response codes.
Switch 3 -- Up  : Issue no response codes at all.
Switch 4 -- Down: Do not echo commands to user.
Switch 5 -- Down: Must issue A command to answer.

A subset of the full Hayes command set is supported for this implementation. The commands are:

A: Answer phone now.
A/: Repeat last command string.
C: Control sending of the carrier.
D: Dial the phone number which follows.
E: Echo commands to the screen.
F: Duplex control.
H: Control ON/OFF hook.
M: Control speaker.
O: Return to on line state after processing command.
P: Signal for pulse dialing.
Q: Control presence of response codes.
R: Specify answer-mode handshake after dialing.
S: Read/Write modem control register.
T: Signal for touch-tone dialing.
V: Control verbal/numeric response codes.
X: Control character set of response codes.
Z: Software reset to default configuration.
; Specify reenter command mode after dialing.
, Specify pause for 2 seconds in dialing.

The default values adopted by the modem driver parallel the initialization state of the Smartmodem. Seventeen registers within the Smartmodem are initially set up in the following manner:

**Register S0:** Determines the number of rings on which the Smartmodem answers calls. Because the emulator does not support auto-answer, this register has no effect on modem operation.

**Register S1:** Increments to tally the number of rings that has occurred. The default value is 0.

**Register S2:** Contains the ASCII value of the Hayes escape code. The default value is 43 ('+').

**Register S3:** Stores the value of the carriage return. The default value is 13.

**Register S4:** Contains the ASCII value for the line feed character. The default value is 10.

**Register S5:** Contains the ASCII value for the backspace character. The default value is 8.

**Register S6:** Determines time between 'picking up the telephone' to dialing the first number. The default value is 2 seconds.

**Register S7:** Specifies the wait time to detect a carrier signal from a distant modem. The default value is 30 seconds.

**Register S8:** Sets the pause time for a comma in dial commands. The default value is 2 seconds.

**Register S9:** Specifies the time allotted for the Smartmodem to detect the presence of a carrier tone. This is not supported by the emulator.

**Register S10:** Specifies time between a distant modem hanging up and a disconnect by the Smartmodem. The default value is 700 milliseconds.

**Register S11:** Regulates the speed of the touch-tone dialer. The default value is 70 milliseconds.

**Register S12:** Specifies the escape guard time. The default value is 1 second.

**Register S13:** Bit-mapped UART Status Register. Not applicable to the emulator.

**Register S14:** Bit-mapped Option Register. Not applicable to the emulator.

**Register S15:** Bit-mapped Flag Register. Not applicable to the emulator.

**Register S16:** Produces self-test mode in the Smartmodem when set to 1. This feature has no equivalent in the emulator.

**Command Format**

In general, the same basic command structure applies to the use of the COM2STAT port as previously-described for the COM1STAT port. The commands that are handled identically are:

```
Flush
Init
Querystatus
DAtabits
Char
Assignfifo
```

Certain parameter differences apply to other previously-described commands. These commands (and the emulator limitations) are:

```
Baud = 300
       1200

Parity = Odd
         Even
         None
         Mark
         Space

Stopbits = 1
           2
```

Volume: This command adjusts the volume of the modem speaker. The range may be expressed between 0 (speaker off) and 255 (loudest volume). At initialization the volume level is set to Soft, a value of 75. A medium level would be 125.

```
V = n
```

## Notable Differences in the Emulator

The emulator cannot perform the auto-answer function. The application program must monitor the S1 register and issue the Hayes answer command. The emulator keeps track of the value stored in Register S1 (the number of rings that has occurred).

Several differences worth mentioning apply to the control registers -- equivalent to registers S0 through S16 in the Smartmodem.

Register 0 is not supported. Register 16 is also not supported. The ranges of acceptable values for Register 6 varies from the Smartmodem. The emulator will handle 2 through 65 seconds, whereas the Smartmodem accepts 2 through 255 seconds.

Register 9 is ignored and can have no effect on the time a carrier must be present before the modem raises carrier detect active.

Because the hardware of the GRiD Compass does not implement a Baud rate of 110, the emulator does not support this value.

As previously mentioned, the speaker volume for the internal modem is maintained through software control. The Smartmodem is equipped with an external volume control knob.

If you require additional details on the specifications and operation of the Hayes Smartmodem, consult the technical literature furnished by Hayes on the subject.

# APPENDIX D: KEYBOARD CODES

Table D-1 lists all of the hexadecimal codes produced by the GRiD Compass MS-DOS keyboard for different keys and key combinations. Most of the keys generate a one-byte ASCII code; for example, lower-case A generates a hexadecimal 61. The SHIFT-arrow keys (SHIFT-LeftArrow, -RightArrow, -UpArrow, and -DownArrow) generate a three-character sequence which begins with the Escape (ESC); for example, the DownArrow produces the sequence ESC [ B. These three-character sequences conform to the ANSI terminal standard for cursor movement. When the CODE key (or CODE-SHIFT keys) is pressed in combination with an alpha-numeric key (0-9, A-Z), a two byte sequence is generated with the first byte being a NUL (hexadecimal 00). Characters shown in parentheses in Table D-1 are the ASCII characters or control codes produced by keys or key combinations.

Table D-1. GRiD Compass MS-DOS Keyboard Codes

| Key | Unshifted | SHIFT | CODE | CODE-SHIFT | CTRL | SHIFT-CTRL | CODE-CTRL | CODE-SHIFT-CTRL |
|---|---|---|---|---|---|---|---|---|
| ' | 27 (') | 22 (") | 60 (`) | 5C (\) | 07 (BEL) | 02 (STX) | 00 03 | 1C (FS) |
| , | 2C (,) | 3C (<) | 5B ([) | 7B ({) | 0C (FF) | 1C (FS) | 1B (ESC) | 1B (ESC) |
| - | 2D (-) | 5F (_) | 00 53 | 7F DEL | 0D (CR) | 1F (US) | 8D | 1F (US) |
| . | 2E (.) | 3E (>) | 5D (]) | 7D (}) | 0E (SO) | 1E (RS) | 1D (GS) | 1D (GS) |
| / | 2F (/) | 3F (?) | BF (code ?) | BF (code ?) | 0F (SI) | 1F (US) | 9F | 9F |
| 0 | 30 (0) | 29 ()) | 00 44 | 00 5D | 10 (DLE) | 09 (HT) | 90 | 89 |
| 1 | 31 (1) | 21 (!) | 00 3B | 00 54 | 11 (DC1) | 01 (SOH) | 91 | 81 |
| 2 | 32 (2) | 40 (@) | 00 3C | 00 55 | 12 (DC2) | 00 03 | 92 | 80 |
| 3 | 33 (3) | 23 (#) | 00 3D | 00 56 | 13 (DC3) | 03 (ETX) | 93 | 83 |
| 4 | 34 (4) | 24 ($) | 00 3E | 00 57 | 14 (DC4) | 04 (EOT) | 94 | 84 |
| 5 | 35 (5) | 25 (%) | 00 3F | 00 58 | 15 (NAK) | 05 (ENQ) | 95 | 85 |
| 6 | 36 (6) | 5E (^) | 00 40 | 00 59 | 16 (SYN) | 1E (RS) | 96 | 9E |
| 7 | 37 (7) | 26 (&) | 00 41 | 00 5A | 17 (ETB) | 06 (ACK) | 97 | 86 |
| 8 | 38 (8) | 2A (*) | 00 42 | 00 5B | 18 (CAN) | 0A (LF) | 98 | 8A |
| 9 | 39 (9) | 28 (() | 00 43 | 00 5C | 19 (EM) | 08 (BS) | 99 | 88 |
| ; | 3B (;) | 3A (:) | 7E (~) | 7C (\|) | 1B (ESC) | 1A (SUB) | 1E (RS) | 1C (FS) |
| = | 3D (=) | 2B (+) | 00 52 | AB (code +) | 1D (GS) | 0B (VT) | 9D | 8B |
| A | 61 (a) | 41 (A) | 00 1E | 00 1E | 01 (SOH) | 01 (SOH) | 81 | 81 |
| B | 62 (b) | 42 (B) | 00 30 | 00 30 | 02 (STX) | 02 (STX) | 82 | 82 |
| C | 63 (c) | 43 (C) | 00 2E | 00 2E | 03 (ETX) | 03 (ETX) | 83 | 83 |
| D | 64 (d) | 44 (D) | 00 20 | 00 20 | 04 (EOT) | 04 (EOT) | 84 | 84 |
| E | 65 (e) | 45 (E) | 00 12 | 00 12 | 05 (ENQ) | 05 (ENQ) | 85 | 85 |
| F | 66 (f) | 46 (F) | 00 21 | 00 21 | 06 (ACK) | 06 (ACK) | 86 | 86 |
| G | 67 (g) | 47 (G) | 00 22 | 00 22 | 07 (BEL) | 07 (BEL) | 87 | 87 |
| H | 68 (h) | 48 (H) | 00 23 | 00 23 | 08 (BS) | 08 (BS) | 88 | 88 |
| I | 69 (i) | 49 (I) | 00 17 | 00 17 | 09 (HT) | 09 (HT) | 89 | 89 |
| J | 6A (j) | 4A (J) | 00 24 | 00 24 | 0A (LF) | 0A (LF) | 8A | 8A |
| K | 6B (k) | 4B (K) | 00 25 | 00 25 | 0B (VT) | 0B (VT) | 8B | 8B |
| L | 6C (l) | 4C (L) | 00 26 | 00 26 | 0C (FF) | 0C (FF) | 8C | 8C |
| M | 6D (m) | 4D (M) | 00 32 | 00 32 | 0D (CR) | 0D (CR) | 8D | 8D |
| N | 6E (n) | 4E (N) | 00 31 | 00 31 | 0E (SO) | 0E (SO) | 8E | 8E |
| O | 6F (o) | 4F (O) | 00 18 | 00 18 | 0F (SI) | 0F (SI) | 8F | 8F |
| P | 70 (p) | 50 (P) | 00 19 | 00 19 | 10 (DLE) | 10 (DLE) | 90 | 90 |
| Q | 71 (q) | 51 (Q) | 00 10 | 00 10 | 11 (DC1) | 11 (DC1) | 91 | 91 |
| R | 72 (r) | 52 (R) | 00 13 | 00 13 | 12 (DC2) | 12 (DC2) | 92 | 92 |
| S | 73 (s) | 53 (S) | 00 1F | 00 1F | 13 (DC3) | 13 (DC3) | 93 | 93 |
| T | 74 (t) | 54 (T) | 00 14 | 00 14 | 14 (DC4) | 14 (DC4) | 94 | 94 |
| U | 75 (u) | 55 (U) | 00 16 | 00 16 | 15 (NAK) | 15 (NAK) | 95 | 95 |
| V | 76 (v) | 56 (V) | 00 2F | 00 2F | 16 (SYN) | 16 (SYN) | 96 | 96 |
| W | 77 (w) | 57 (W) | 00 11 | 00 11 | 17 (ETB) | 17 (ETB) | 97 | 97 |
| X | 78 (x) | 58 (X) | 00 2D | 00 20 | 18 (CAN) | 18 (CAN) | 98 | 98 |
| Y | 79 (y) | 59 (Y) | 00 15 | 00 15 | 19 (EM) | 19 (EM) | 99 | 99 |
| Z | 7A (z) | 5A (Z) | 00 2C | 00 2C | 1A (SUB) | 1A (SUB) | 9A | 9A |
| BACKSPACE | 08 | C8 | 88 | 8A | 08 (BS) | 88 | 88 | 8A |
| RETURN | 0D | CD | 8D | 8C | 0D (CR) | 8D | 8D | 8C |
| DownArrow | 00 50 | ESC [ B | 00 51 | 00 4F | 84 | 8E | 00 76 | 00 75 |
| ESC | 1B | 1B | 9B | 9B | 1B (ESC) | 1B (ESC) | 9B | 9B |
| LeftArrow | 00 4B | ESC [ D | D4 | D8 | 86 | 00 73 | 94 | 98 |
| RightArrow | 00 4D | ESC [ C | D5 | D9 | 87 | 00 74 | 95 | 99 |
| Spacebar | 20 (SP) | 20 (SP) | 20 (SP) | 20 (SP) | 00 03(NUL) | 00 03 | 00 03 | 00 03 |
| TAB | 09 | 00 0F | 89 | 8B | 09 | 89 | 89 | 8B |
| UpArrow | 00 48 | ESC [ A | 00 49 | 00 47 | 85 | 8F | 00 84 | 00 77 |

# APPENDIX E: FABRICATING AN RS-232-C CABLE

For serial communication the GRiD Compass utilizes a 19-pin connector (DE-19) wired to maintain RS-232-C conventions. Some serial applications, particularly those used with MS-DOS, may require that the serial connector be equivalent to the IBM PC interface. Since the IBM PC uses a 25-pin serial connector (DB-25), you must fabricate a cable to provide the necessary signals to meet this interface standard. Table E-1 lists the pins from the GRiD Compass SERIAL connector that must be wired to the listed pins on the 25-pin connector.

Figure E-1 shows the configurations for each type of connector.

TABLE E-1. Necessary Signals for RS-232-C Interface

| DE-19 pin | Signal | DB-25 pin |
|---|---|---|
| 1 | Shield | 1 |
| 2 | Transmit Detect | 2 |
| 3 | Receive Detect | 3 |
| 4 | Ready to Send | 4 |
| 5 | Clear to Send | 5 |
| 6 | Data Set Ready | 6 |
| 7 | Ground | 7 |
| 8 | Data Carrier Detect | 8 |
| 10 | Ground | 7 |
| 12 | Data Terminal Ready | 20 |
| 17 | Ground | 7 |

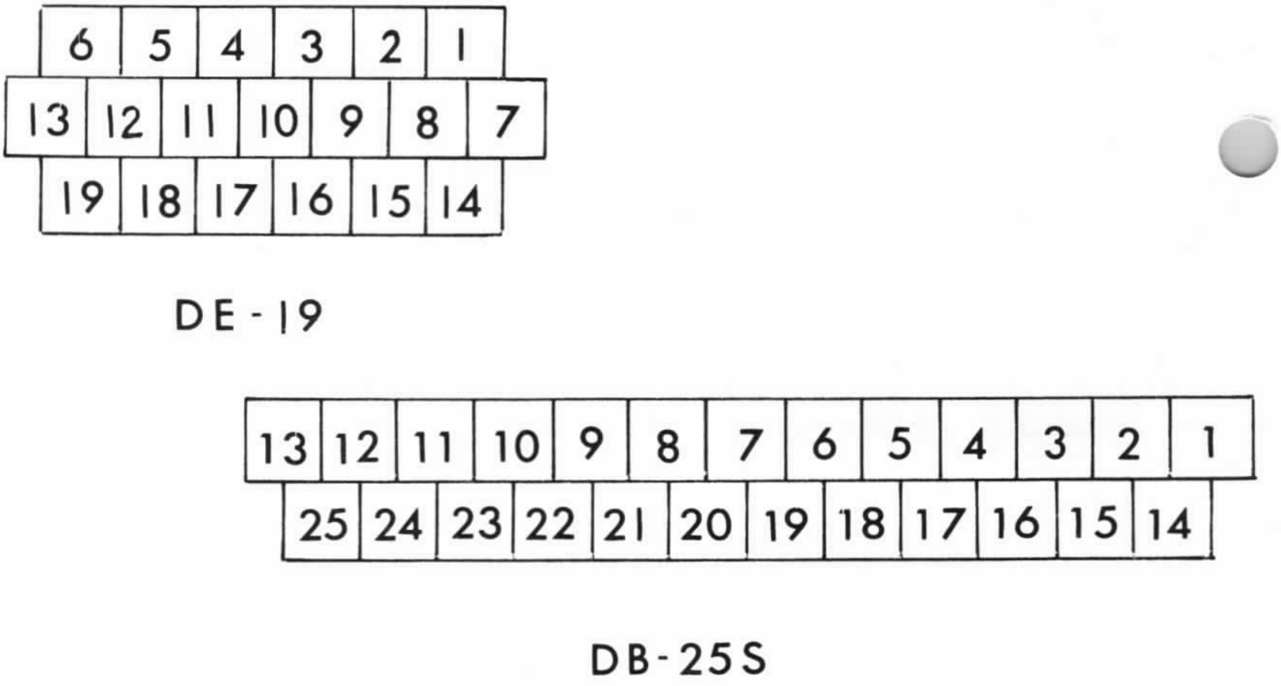


Figure E-1. DE-19 and DB-25 Connector Pin Configurations