GRiD MS-DOS RELEASE 2.11 UPDATE

July 16th, 1984

Before using MS-DOS Release 2.11, be aware of the information in this document; it reflects changes in GRiD MS-DOS capabilities effective in GRiD MS-DOS Release 2.11, Version B; the Compass Computer MS-DOS User's Guide (shipped as part of this package) reflects GRiD MS-DOS Release 2.00, Version A.

Contents

***New edition. Discard old edition of Appendix E in the current manual.**

GRiD MS-DOS Release 2.11 Update

New capabilities available in GRiD MS-DOS Release 2.11 are summarized below.

- A new external command, CONFIG, makes all applications "device independent." You can now run serial printers and plotters, and other serial output devices with all MS-DOS applications. Formerly, for most applications, you could print or plot only on GPIB devices.

- The Serial and Modem drivers, which control input and output between application programs and external I/O devices, have been redesigned and rewritten. A replacement for Appendix C (attached) describes the interfaces and I/O functions of the new drivers. Existing programs performing Serial and Modem I/O operations will continue to run without modification.

- You now use the Time and Date commands provided by MS-DOS to change the internal clock and calendar on your computer; the SETTIME command, formerly used for these functions, is no longer supported.

- For programmers coding MS-DOS applications, new erase, delete, and insert code sequences have been added to the ANSI-compatible console driver.

Read the remainder of this notice for details.

Considerations For Release 2.00 Users Before Installing Release 2.11

The Release 2.11 system file (not visible in a directory) is 512 bytes larger than in Release 2.0; as a result, you cannot replace Release 2.00 MSDOS.SYS on certain storage media. These media include bubble memory, hard disks with an MS-DOS partition of one megabtye or less, or single-sided diskettes. Attempting to copy the system to these media generates the error message "Incompatible system sizes."

You must therefore take the following measures when installing MS-DOS Release 2.11 on the media just described:

- Boot your computer from the MS-DOS Release 2.11 distribution diskette.

- Back up all files in bubble memory you want to keep, reformat bubble memory using the FORMAT utility with the /S option, and then restore your backed-up files.

- Back up all files on a hard disk with an MS-DOS partition of one megabyte or less, reformat the hard disk using the FORMAT utility with the /S option, and then restore your backed-up files.

- Run the FORMAT utility using the /S option on new diskettes equal to the number of single-side system diskette you currently use. Then, copy all your data files and programs from the old diskettes onto the new.

Special Note to dBASE II Users

Only Version 2.41 or higher of dBASE II run under under MS-DOS Release 2.11. You must run earlier versions under MS-DOS Release 2.0 or earlier releases. Contact your software dealer for the latest version of dBASE II.

Publication Changes

The Compass Computer MS-DOS User's Guide was written to conform with MS-DOS Version 2.00. The following information gives changes to the first section of the manual (Using MS-DOS on the GRiD Compass) to reflect MS-DOS Version 2.11 (the latest version) as shipped to you; it also makes minor technical corrections to information in the manual.

Chapter 5 - Changes in FORMAT Utility

When used with the /S switch, the FORMAT utility now copies the system onto the diskette or hard disk immediately after the formatting process. Formerly, before copying the system, you had to reinsert the MS-DOS system diskette before the copying process. Thus, disregard Steps 4 and 5 in the formatting procedure described on pages 5-2 and 5-3.

Chapter 6 - Addition of CONFIG Command

The CONFIG command lets you run either serial or GPIB printers or plotters regardless of application. In addition, CONFIG permits the modification of certain values that control the exchange of data between the device and application, as described below.

To use the CONFIG Command

1. Enter CONFIG next to the standard MS-DOS prompt.

2. The Configuration form appears. (The items in the form are described in the sections that follow.) Use the UpArrow and DownArrow keys to select the items you wish to change; use

the LeftArrow or RightArrow key to select the new value or designator for each item.

3. Press CODE-RETURN to confirm your choices, or press ESC to exit leaving the original settings in effect.

The new configuration you specify is in effect after pressing CODE-RETURN; it remains in effect after each subsequent start-up. If you change from a serial device to a modem device, or vice versa, you will have to again use CONFIG to specify the new configuration.

The following sections describe the items in the Configuration form.

Printer

Initial setting is GPIB. Choices are: GPIB, Serial.

Specify GPIB or Serial, according to which connector (GPIB or the Serial) you plug in the cable to your printer. Thereafter, any data that an application writes to PRN goes to the device attached to the specified connector. (PRN is a symbolic name used as part of an I/O instruction within the application; it indicates, in an output operation, that data is to be written to a printer.)

Plotter

Initial setting is GPIB. Choices are: GPIB, Serial.

Specify GPIB or Serial, according to which connector (GPIB or the Serial) you plug in the cable to your plotter. Thereafter, any data that an application writes to PLOTTER goes to the device attached to the specified connector. (PLOTTER is a symbolic name used as part of an I/O instruction within the application; it indicates, in an output operation, that data is to be written to a plotter.)

Protocol

Initial setting is None. Choices are: None, XON/XOFF.

The XON/XOFF option is a standard data communication protocol used by some printers and plotters. When specified, XON/XOFF causes the application to stop transmitting when the device sends an XOFF character; the application resumes transmitting when the device sends an XON character.

Some devices respond to XON/XOFF sequences also. In such cases, if the application sends an XOFF character to the device, then the device stops transmitting until it receives an XON character.

Baud Rate

Initial setting is 300 bits per second. Choices are: 50, 150 300, 600, 1200, 2400, 4800, 9600, 19200.

The Baud Rate item is the speed at which data is transmitted and received. The modem or external device attached to the Serial connector determines which baud rate you choose.

Parity

A parity bit is a bit added to each transmitted character so that the receiving device can check the character for accuracy. Set Parity to match the parity expected by the external device.

Initial setting is Even. Choices are listed below.

None — No parity bit is set.

Odd — Parity is set to 1 to make the sum of the (binary) digits of the character into an odd number.

Even — Parity is set to 1 to make the sum of the (binary) digits of the character into an even number.

Stop Bits

Initial setting is 1. Choices are: 1, 1.5, 2.

In asynchronous transmission, one or two extra bits, called stop bits, are always placed on the end of each character. These bits let the receiver detect the beginning of the next character.

Data Bits (Bits Per Character)

Initial setting is 7. Choices are: 5, 6, 7, 8.

The Data Bits setting specifies the number of bits in each 8-bit byte to be considered as data. The standard ASCII code uses seven bits per byte.

Chapter 6 - Removal of SETTIME Command

MS-DOS Release 2.11 doesn't have the SETTIME command, used with Version 2.00 to set the time and date in the internal clock of your computer.

To set or change the time and date, use the TIME and DATE commands commands respectively. These commands are described in the second section (the Microsoft MS-DOS Operating Systems User's Guide) of the

Compass Computer MS-DOS User's Guide.

NOTE: The documentation shipped with Lotus 1-2-3 for the GRiD Compass and with certain other applications refers to the SETTIME command; the reference is no longer valid. The TIME and DATE commands perform the functions formerly associated with SETTIME.

Chapter 8 - Cursor and Screen Control Sequences

Add the following ASCII code sequences to Table 8-2 in Chapter 8:

ASCII Code Sequence	Action
ESC [0 J	Erase to end of display.
ESC [n M	Deletes one or more lines following the current cursor position, where n is the number of lines to be deleted. If you don't specify n, one line is deleted. After deletion, the cursor remains stationary and the lines below the deleted line(s) scroll up.
ESC [n L	Inserts one or more lines following the current cursor position, where n is the number of lines to be inserted. If you don't specify n, one line is inserted. After insertion, the cursor remains stationary and the lines below the inserted line(s) scroll down.

Chapter 9 -- BASIC and the GRiD Compass

Change the CLS statement on page 9-4 to read as follows:

```
CLS     PRINT CHR$(27%)+"[2J";
```

Double quotes (instead of single quotes) surround the ending characters in the statement.

Appendix B -- Graphics Using Bit-Mapped Video

Change the address locations shown in Figure B-1 on page B-2 as follows:

Now reads:	Should read:
406	404

Appendix C: Serial and Modem Driver Input/Output

This appendix provides information for MS-DOS programmers writing code to communicate with devices attached to the GRiD internal modem or serial port. The following topics are covered:

- Driver and application program interface
- Driver function requests
- Assembler language programming example

Driver Interface You request input/output operations in an applications program by issuing function requests to either the Serial or Modem driver. The function request instructs the driver to perform a certain action, such as read or write data, set baud rates, obtain status information, and the like. The driver interprets the requests and communicates them directly to the device.

Before initiating a function request, you provide, in one or more registers, parameters needed by the driver. You initiate the request with the instruction INT 81H (for the Serial driver) or INT 82H (for the Modem driver).

See the section Driver Function Requests later in this chapter for a description of the function requests and their required parameters. An assembler language program containing examples of the requests follows in the last section.

Error Codes

After execution of a function request, your program must check the carry flag to determine if the operation was successful (the flag is set to 0) or if it failed (the flag is set to 1). If the operation failed, one of the following error codes is provided in the AX register:

- 1 - Either the driver isn't in memory or, when communicating with the Serial driver, the GRiDLink cable is attached to the Serial port.
- 2 - The remote device connected to the serial or modem port isn't responding.
- 3 - you specified an invalid command or parameter.

Modem Driver and Hayes Smartmodem Commands
The Modem driver emulates the Hayes Smartmode, allowing you to issue Hayes Smartmodem commands from your application. You issue the commands with Function 2 (Write Data) after providing a pointer to a buffer containing the command and any additional data required.

The Modem driver processes the following Smartmodem commands:

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.

For details on the operation of the Hayes Smartmodem, see the technical literature furnished by Hayes on the subject.

Differences Smartmodem functions as emulated by the Modem driver differ from those as implemented by the Hayes Smartmodem as follows:

- Auto-answer is not implemented. You must include code in your application to check the contents of the S1 register (which keeps a count of the number of rings of an incoming call) and then issue the Hayes Answer command (A).

- Registers 0 and Register 16 are not supported.

- The ranges of acceptable values for Register 6 is 2 through 65 seconds (for the Smartmodem, the range is 2 through 255 seconds).

- Register 9 is ignored; (for the Smartmodem, the value in this register determines the time a carrier must be present before the modem raises the condition "carrier detect active").

- A Baud rate of 110 isn't supported.

- You control the speaker volume for the internal modem using a function request (see Function Request 15 for the Modem driver); the user controls the volume on the Smartmodem by turning a control knob.

Switch Settings Like the Smartmodem, the Modem driver assumes the following switch settings 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.
```

Initial Value in Registers The Modem driver uses 17 registers as described Table C-1; the driver sets the registers it supports to the same initial value as the Smartmodem uses.

Table C-1. Register Usage in Smartmodem Emulation

Register	Contents
S0:	Not supported. (On the Hayes Smartmodem, this register determines the number of rings after which the Smartmodem answers calls. Because the driver doesn't support auto-answer, this register has no effect on driver operations.)
S1:	Keeps a count of the number of rings on incoming calls. The default value is 0.
S2:	Contains the ASCII value of the Hayes escape code. The default value is 43 ('+').
S3:	Stores the value of the carriage return. The default value is 13.
S4:	Contains the ASCII value for the line feed character. The default value is 10.
S5:	Contains the ASCII value for the backspace character. The default value is 8.
S6:	Determines time between picking up the telephone and dialing the first number. The default value is 2 seconds.

Register	Contents

Table C-1 (continued). Register Usage in Smartmodem Emulation

S7:	Specifies the time in seconds to await a carrier signal from a distant modem. The default value is 30 seconds.
S8:	Specifies the time in seconds to pause when a comma is inserted in a dial command. The default value is 2 seconds.
S9:	Not supported. (For the Smartmodem, this register contains a value that specifies the time interval to wait for a carrier tone.)
S10:	Specifies the time in milliseconds the driver is to wait before disconnecting after a remote modem hangs up. The default value is 700 milliseconds.
S11:	Regulates the speed of the touch-tone dialer. The default value is 70 milliseconds.
S12:	Specifies the escape guard time. The default value is 1 second.
S13:	Not supported. (Used by the Smartmodem as the bit-mapped UART Status Register.)
S14:	Not supported. (Used by the Smartmodem as the bit-mapped Option Register.)
S15:	Not supported. (Used by the Smartmodem as the bit-mapped Flag Register.)
S16:	Not supported. (Used to place the Smartmodem in self-test mode.)

Driver Function Requests This section describes the function requests and their parameters. Table C-2 gives an alphabetical list of the the requests available. The sections that follow (in numerical order according to request numbers) describe the requests and gives their required parameters.

Table C-2. Driver Function Requests (ordered by function)

Function	Number (Decimal)	Default At Start-Up
Baud Rate	8	300[3]
Buffer Assign	12	Within driver.
Buffer Flush	6	Not applicable.
Character Timeout Value	13	700
Clear To Send (CTS)	19	1000[2]
Data Character Detect (DCD)	17	Enabled[2] (Ignored on Write)
Data Set Ready (DSR)[2]	20	1000
Data Terminal Ready (DTR)	16	Active[2]
Data Bits	10	7[3]
High Speed Delay	15	On[2]
Initialize	0	Not applicable.
Linefeed Insert	21	Off[2]
Parity	9	Even[3]
Read Data	1	Not applicable.
Read Query	3	Not applicable.
Read Status Information	5	Not applicable.
Request To Send	17	Active[2]
Return Settings	7	Not applicable.
Set Break On/Off	14	Off
Stop Bits	11	1[3]
Volume	15	75 (Soft)[1]
Write Command	4	Not applicable.
Write Data	2	Not applicable.

[1] Valid for the Modem Driver only.

[2] Valid for the Serial Driver only.

[3] The user can change this default value with the CONFIG command, described elsewhere in this document.

Function 0 — Initialize

Call: AH = 0 (function number)

Return: Carry flag = 0 (successful operation)

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)
82H (Modem Driver)

Function 0 resets the configuration to reflect the default values shown in Table C-2. These are the values provided within the driver as they have been modified (if applicable) with CONFIG. Function 0 also reassigns the interrupt buffer to the input area within the driver if you earlier assigned a different one using Function 12.

CAUTION: Always use Function 0 before exiting a program. Changing interrupt buffers or the default configuration values can adversely affect the operations of subsequent programs.

Function 1 -- Read Data

Call: AH = 1 (function number)
ES:DI = pointer to input buffer
CX = number of characters to read.

Return: Carry flag = 0 (successful operation)
AX = number of characters in the input buffer.

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)
82H (Modem Driver)

Function 1 reads data from the serial or modem port into an input buffer whose pointer you specify in ES:DI. The AX register contains either the number of characters read (if successful) or an error code (if unsuccessful) as indicated above.

To ensure you receve all of the characters you expect after a successful read, compare the number of characters actually read (the AX register) with the number of characters expected (the CS register). A timeout (see Function 13) may cut short the number of incoming characters.

Function 2 -- Write Data

Call: AH = 2 (function number)
ES:DI = pointer to output buffer
CX = number of characters to write.

Return: Carry flag = 0 (successful operation)
AX = number of characters written.

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)
82H (Modem Driver)

Function 3 write data to the serial or modem port from an output buffer whose pointer you specify in ES:DI.

Function 3 -- Read Query

Call: AH = 3 (function number)
ES:DI = pointer to status buffer
CX = number of characters to read (1 through 5).

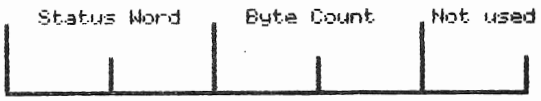
Return: Carry flag = 0 (successful operation)
AX = number of characters read.

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)
82H (Modem Driver)

NOTE: Function 3 is included for users who wish to modify existing applications written before MS-DOS Release 2.11. Use Function 5 (Read Status Information) for new applications; it provides the same information and is easier to code.

Function 3 returns the status of the serial or modem port at the time of the last Query command; no status information is returned unless a Query was previously issued. The information is returned in a buffer you define up to five byte long (depending on how many bytes are read). The following is the format of the buffer when five bytes are read:



The settings in the status word contains the pertinent information; see Function 5 for an explanation of the settings.

Function 4 — Write Command

Call: AH = 4 (function number)
ES:DI = pointer to command buffer
CX = number of characters to write

Return: Carry flag = 0 (successful operation)
AX = number of characters written.

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)
82H (Modem Driver)

NOTE: Function 4 is provided for users who wish to modify configuration commands in routines written before DOS Release 2.11. Use Functions 5 through Function 21 to configure the driver if you are writing a new program.

Function 4 writes configuration commands (such as baud rate, parity, data bits, etc.) placed as ASCII strings in the input buffer whose pointer you specify in ES:DI. The valid ASCII command are documented in Appendix C of the first edition of the Compass Computer MS-DOS User's Guide.

Function 5 — Read Status Information

Call: AH = 5 (function number)

Return: Carry flag = 0 (successful operation)
AX = status word (the layout is described below).
CX = number of characters in the input buffer.

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)
82H (Modem Driver)

Function 5 causes the modem status word, which contains device and input buffer information, to be written in the AX register in the following format:

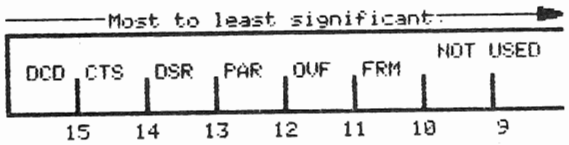


Table C-3 contains an explanation of the setting in the modem status word.

Table C-3. The Modem Status Word

Bit	Meaning When On
Bit 15	DCD (Data Character Detect) value specified in configuration command exceeded.
Bit 14	CTS (Clear To Send) time limit specified in configuration command exceeded.
Bit 13	DSR (Data Set Ready) time limit specified in configuration command exceeded.
Bit 12	Parity error
Bit 11	Overflow error
Bit 10	Framing error
Bits 0-9	Note used

Function 6 -- Buffer Flush

Call: AH = 6 (function number)

Return: Carry flag = 0 (successful operation)

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)
82H (Modem Driver)

Function 6 purges the input buffer of any data received which you don't want to process in your application.

Function 7 -- Return Settings

Call: AH = 7 (function number)

Return: Carry flag = 0 (successful operation)
AH = baud rate index (see Function 8)
AL = parity
0 = none
1 = odd
2 = even
CH = number of data bits (5, 6, 7, or 8)
CL = number of stop bits
1 = 1 stop bit
2 = 2 stop bits
3 = 1.5 stop bits (Serial driver only)

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)
82H (Modem Driver)

Function 7 returns the current baud rate and the values for parity, data bits, and stop bit.

Function 8 — Baud Rate

Call: AH = 8 (function number)
AL = baud rate index as follows:

AL	Baud	AL	Baud
0 =	50	9 =	2000
1 =	75	10 =	2400
2 =	110	11 =	3600
3 =	134.5	12 =	4800
4 =	150	13 =	7200
5 =	300	14 =	9600
6 =	600	15 =	19200
7 =	1200		
8 =	1800		

NOTE: Only the values 5 (for 300 baud) or 7 (for 1200 baud) are valid for the Modem driver.

Return: Carry flag = 0 (successful operation)

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)
82H (Modem Driver)

Default: 300 (except for the Serial driver, where the default can be modified by CONFIG)

Function 8 determines the rate for data transfer between two systems.

Function 9 -- Parity

Call: AH = 9 (function number)
AL = parity index
0 = none
1 = odd
2 = even
3 = mark (Modem Driver only)
4 = space (Modem Driver only)

Return: Carry flag = 0 (successful operation)

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)
82H (Modem Driver)

Default: Even (except for the Serial driver, where the default can be modified by CONFIG)

Function 9 determines the nature of the parity bit, if used.

Function 10 — Data Bits

Call: AH = 10 (function number)
AL = Number of data bits per character (5, 6, 7, or 8)

Return: Carry flag = 0 (successful operation)

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)
82H (Modem Driver)

Default: 7 (except for the Serial driver, where the default can be modified by CONFIG)

Function 10 specifies the number of bits that represent data in each 8-bit byte being transferred.

Function 11 -- Stop Bits

Call: AH = 11 (function number)
AL = stop bit index
1 = one stop bits
2 = two stop bits
3 = 1.5 stop bits (Serial Driver only)

Return: Carry flag = 0 (successful operation)

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)
82H (Modem Driver)

Default: 1 (except for the Serial driver, where the default can be modified by CONFIG)

Function 11 defines the number of stop bits that determine the extent of a data word.

Function 12 -- Buffer Assign

Call: AH = 12 (function number)
ES:DI = pointer to a new interrupt (FIFO) buffer
CX = number of bytes in the buffer

Return: Carry flag = 0 (successful operation)

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)
82H (Modem Driver)

Default: Internal buffer within driver.

Function 12 lets you define a input buffer to receive data ordinarily directed to the 256-byte interrupt buffer within the driver. The interrupt buffer is an intermediate buffer where data is stored en route to your application (when reading data) or en route to the external device (when writing data.

There may be instances when, due to extensive processing, your application cannot read characters as fast as they are received; this can cause an overflow conditions in the internal buffer and loss of data to your application. In such cases, use Function 12 to define a larger interrupt input for the driver.

NOTE: Even though the new interrupt buffer is defined within your application area, continue to read the data using Function 2 rather accessing the new buffer directly.

CAUTION: Always issue Function 0 (Initialize) to resume using the internal interrupt buffer before exiting an application. Failure to do so could corrupt the next program loaded after exiting your application.

Function 13 -- Character (CHAR) Timeout Value

Call: AH = 13 (function number)
DX = 1 through 65535, indicating the delay time in milliseconds.
DX = 0 indicates a limitless wait period.

Return: Carry flag = 0 (successful operation)

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Default: 700

Function 13 specifies a time limit to wait for the arrival of each character after issuing a READ request.

Function 14 -- Set Break On/Off

Call: AH = 14 (function number)
AL = 0 (turn on break), 0FFH (turn off break)

Return: Carry flag = 0 (successful operation)

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)
82H (Modem Driver)

Default: Off

Function 14 interrupts any ongoing communications with the serial port. To cause a break, first issue Function 14 turning on the break (specify 0FFH in the AL register), place your application in a wait state for an interval, then issue another Function 14 turning off the break (specify 0 in the AL register).

Function 15 -- High Speed Delay (Serial Driver Only)

Call: AH = 15 (function number)
AL = 0 (turn off delay) or 0FFH (turn on delay)

Return: Carry flag = 0 (successful operation)
AL = 0

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)

Default: On

Function 15 introduces a small delay interval between the each character as it is transmitted. This allows peripheral devices time to process data arriving a high speeds.

NOTE: Function 15 permits an HP7470A plotter to accept data transfers at relatively high speeds without producing communication errors.

Function 15 — Volume (Modem Driver Only)

Call: AH = 15 (function number)
DX = Volume index (0 through 255, from low to high)

Return: Carry flag = 0 (successful operation)

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 82H (Modem Driver)

Default: 75 ("soft")

Function 15 adjusts the volume of the modem speaker. Specify any value from 0 (the speaker is off) through 255 (the speaker is at its highest volume. At initialization, the volume a set to 75, a "soft" volume. A medium volume would be 125.

Function 16 -- Request To Send (Serial Driver Only)

Call: AH = 16 (function number)
AL = 0 (turn off signal), 0FFH (turn on signal)

Return: Carry flag = 0 (successful operation)

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)

Default: On

Function 16 allows an applications to control the Request to Send (RTS) signal)

Function 17 — Data Terminal Ready (Serial Driver Only)

Call: AH = 17 (function number)
AL = 0 (turn off signal), 0FFH (turn on signal)

Return: Carry flag = 0 (successful operation)

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)

Default: On

Function 17 allows an applications to control the Data Terminal Ready (DTR) signal)

Function 18 -- Data Character Detect (Serial Driver Only)

Call: AH = 18 (function number)
DX = 1 through 65535, indicating a timeout value in milliseconds.
DX = 0 disables the timeout

Return: Carry flag = 0 (successful operation)

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)

Default: Enabled (ignored on write)

Function 18 specifies the maximum amount of time, after issuing a READ, that the driver waits for the detection of a carrier.

Function 19 -- Clear To Send (Serial Driver Only)

Call: AH = 19 (function number)
DX = 1 through 65535, indicating a timeout value in milliseconds.
DX = 0 disables the timeout

Return: Carry flag = 0 (successful operation)

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)

Default: 1000 milliseconds

Function 19 specifies the maximum amount of time, after issuing a WRITE, that the driver waits for the clear to send (CTS) signal.

Function 20 -- Data Set Ready (Serial Driver Only)

Call: AH = 20 (function number)
DX = 1 through 65535, indicating a timeout value in milliseconds.
DX = 0 disables the timeout

Return: Carry flag = 0 (successful operation)

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)

Default: 1000 milliseconds

Function 20 specifies the maximum amount of time, after issuing a WRITE, that the driver waits for the data set read (DTR) signal.

Function 21 -- Linefeed Insert (Serial Driver Only)

Call: AH = 21 (function number)
AL = 0 (insert linefeed character after each carriage return character), 0FFH (don't insert linefeed character)

Return: Carry flag = 0 (successful operation)

Carry flag = 1 (an error was detected)
AX = error code (see the section Error Codes)

Interrupt: 81H (Serial Driver)

Default: Off

Function 21 command causes the driver to insert a line-feed character (hexadecimal 0A) after every carriage return (hexadecimal 0D) encountered in the data stream.

Assembler Language Example of Driver I/O

```
;*************************************************************;
;  This program illustrates using the MS-DOS modem driver.
;
;  Programming the serial driver is done in a similar fashion.
;
;  Program operation:
;
;         1.  Initialize the modem driver
;         2.  Set the baud rate to 1200
;         3.  Set parity to odd.
;         4.  Set seven data bits
;         5.  Set one stop bit
;         6.  Replace the internal receive FIFO buffer
;             with a larger buffer.
;         7.  Dial a phone number.
;         8.  Read five bytes from the modem.
;         9.  Write the five bytes back to the modem.
;        10.  Re-initialize the modem driver to put it
;             in a default state.
;        11.  Exit and return to MS-DOS.
;
;*************************************************************

NAME      MODEM_TEST

STACK     SEGMENT STACK 'STACK'

          DW   200 DUP (?)

STACK     ENDS

DATA      SEGMENT

          PHONENUMBER DB 'ATDT9,5551212',0DH
          ERRORMSG DB 'ERROR: ABORTING$'
          BUFFER  DB 100 DUP (?)
          RECEIVE_FIFO DB 512 DUP (?)

DATA      ENDS

CODE      SEGMENT
          ASSUME CS:CODE,DS:DATA

BEGIN:    MOV  AX,DATA
          MOV  DS,AX
          MOV  ES,AX

;INITIALIZE THE MODEM DRIVER

          MOV  AH,0
          INT  82H
          JNC  Q1
          JMP  ABORT
```

```
;SET BAUD RATE

Q1:     MOV  AL,7          ;1200 BAUD
        MOV  AH,8
        INT  82H
        JNC  Q2
        JMP  ABORT

;SET PARITY

Q2:     MOV  AL,1   ;ODD PARITY
        MOV  AH,9
        INT  82H
        JNC  Q3
        JMP  ABORT

;SET NUMBER OF DATA BITS

Q3:     MOV  AL,7   ;SEVEN DATA BITS
        MOV  AH,10
        INT  82H
        JNC  Q4
        JMP  ABORT

;SET NUMBER OF STOP BITS

Q4:     MOV  AL,1   ;ONE STOP BIT
        MOV  AH,11
        INT  82H
        JNC  Q5
        JMP  ABORT

;ASSIGN LARGER RECEIVE BUFFER

Q5:     MOV  AX,DS    ;SEGMENT OF NEW BUFFER
        MOV  ES,AX
        LEA  DI,RECEIVE_FIFO  ;OFFSET OF NEW BUFFER

        MOV  CX,LENGTH RECEIVE_FIFO

        MOV  AH,12
        INT  82H
        JNC  Q6
        JMP  ABORT

;DIAL PHONE NUMBER

Q6:     MOV  AX,DS    ;SEGMENT OF NUMBER
        MOV  ES,AX
        LEA  DI,PHONENUMBER  ;OFFSET OF NUMBER
        MOV  CX,14    ;LENGTH NUMBER

        MOV  AH,2
```

```
        INT  82H
        JNC  MORE
        JMP  ABORT

MORE:

;READ 5 CHARACTERS FROM MODEM

        MOV  AX,DS    ;SEGMENT OF READ BUFFER
        MOV  ES,AX
        LEA  DI,BUFFER  ;OFFSET OF READ BUFFER
        MOV  CX,5    ;NUMBER OF BYTES
        MOV  AH,1
        INT  82H
        JC   ABORT
        CMP  CX,AX    ;ABORT IF 5 NOT READ
        JC   ABORT

;WRITE 5 CHARACTERS TO MODEM

        MOV  AX,DS
        MOV  ES,AX
        LEA  DI,BUFFER
        MOV  CX,5
        MOV  AH,2
        INT  82H
        JC   ABORT

        JMP  DONE

;DISPLAY ERROR MESSAGE IF NECESSARY

ABORT:  LEA  DX,ERRORMSG
        MOV  AH,9
        INT  21H

;RE-INITIALIZE THE MODEM DRIVER

DONE:   MOV  AH,0
        INT  82H

;RETURN TO MS-DOS

        MOV  AH,4CH    ;EXIT
        INT  21H


CODE    ENDS
        END  BEGIN
```

Appendix E: GRiD Compass Internal Keyboard Codes

NOTE: Discard Appendix E in the current edition of Using MS-DOS on the GRiD Compass and replace it with this updated edition; the old appendix contains technical inaccuracies.

The tables in this appendix lists the hexadecimal codes generated when a key or combination of keys is pressed on the GRiD Compass keyboard.

The column headings indicate the code generated when a key is pressed alone, or when it is pressed together with SHIFT, CODE, CODE-SHIFT, CTRL, SHIFT-CTRL, CODE-CTRL, or CODE-SHIFT-CTRL.

Table E-1 (Part 1 of 3). Internal Codes for the GRiD Compass Keyboard

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)	N/A
-	2D (-)	5F (_)	00 53	7F DEL	0D (CR)	1F (US)	0D	Soft Reset
.	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)	00 73	89
1	31 (1)	21 (!)	00 38	00 54	11 (DC1)	01 (SOH)	00 74	81
2	32 (2)	40 (@)	00 3C	00 55	12 (DC2)	00 03	00 76	80
3	33 (3)	23 (#)	00 39	00 56	13 (DC3)	03 (ETX)	00 84	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)	00 75	9E
7	37 (7)	26 (&)	00 41	00 5A	17 (ETB)	06 (ACK)	00 77	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	00 85
;	3B (;)	3A (:)	7E (~)	7C (|)	1B (ESC)	1A (SUB)	1E (RS)	1C (FS)
=	3D (=)	2B (+)	00 52	AB (code +)	1D (GS)	0B (VT)	9B	8B

Table E-1 (Part 2 of 3). Internal Codes for the GRiD Compass Keyboard

Key	Unshifted	SHIFT	CODE	CODE-SHIFT	CTRL	SHIFT-CTRL	CODE-CTRL	CODE-SHIFT-CTRL
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)	00 85	00 85
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)	00 73	00 73
Q	71 (q)	51 (Q)	00 10	00 10	11 (DC1)	11 (DC1)	00 74	00 74
R	72 (r)	52 (R)	00 13	00 13	12 (DC2)	12 (DC2)	00 76	00 76
S	73 (s)	53 (S)	00 1F	00 1F	13 (DC3)	13 (DC3)	00 84	00 84
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)	00 75	00 75
W	77 (w)	57 (W)	00 11	00 11	17 (ETB)	17 (ETB)	00 77	00 77
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

Table E-1 (Part 3 of 3). Internal Codes for the GRiD Compass Keyboard

Key	Unshifted	SHIFT	CODE	CODE-SHIFT	CTRL	SHIFT-CTRL	CODE-CTRL	CODE-SHIFT-CTRL
BACKSPACE	08	C8	00 85	8A	08	00 85	00 85	8A
RETURN	0D	CD	8D	8C	0D	8D	8D	8C
DownArrow	00 50	1B 5B 64	00 51	00 4F	84	8E	00 76	00 75
ESC	1B	1B	9B	9B	1B	N/A	9B	9B
LeftArrow	00 4B	1B 5B 44	D4	D8	86	00 73	94	98
RightArrow	00 4D	1B 5B 43	D5	D9	87	00 74	95	99
Spacebar	20	20	20	20	00 03	00 03	00 03	00 03
TAB	09	00 0F	89	8B	09	89	89	8B
UpArrow	00 48	1b 5b 41	00 49	00 47	85	8F	00 84	00 77