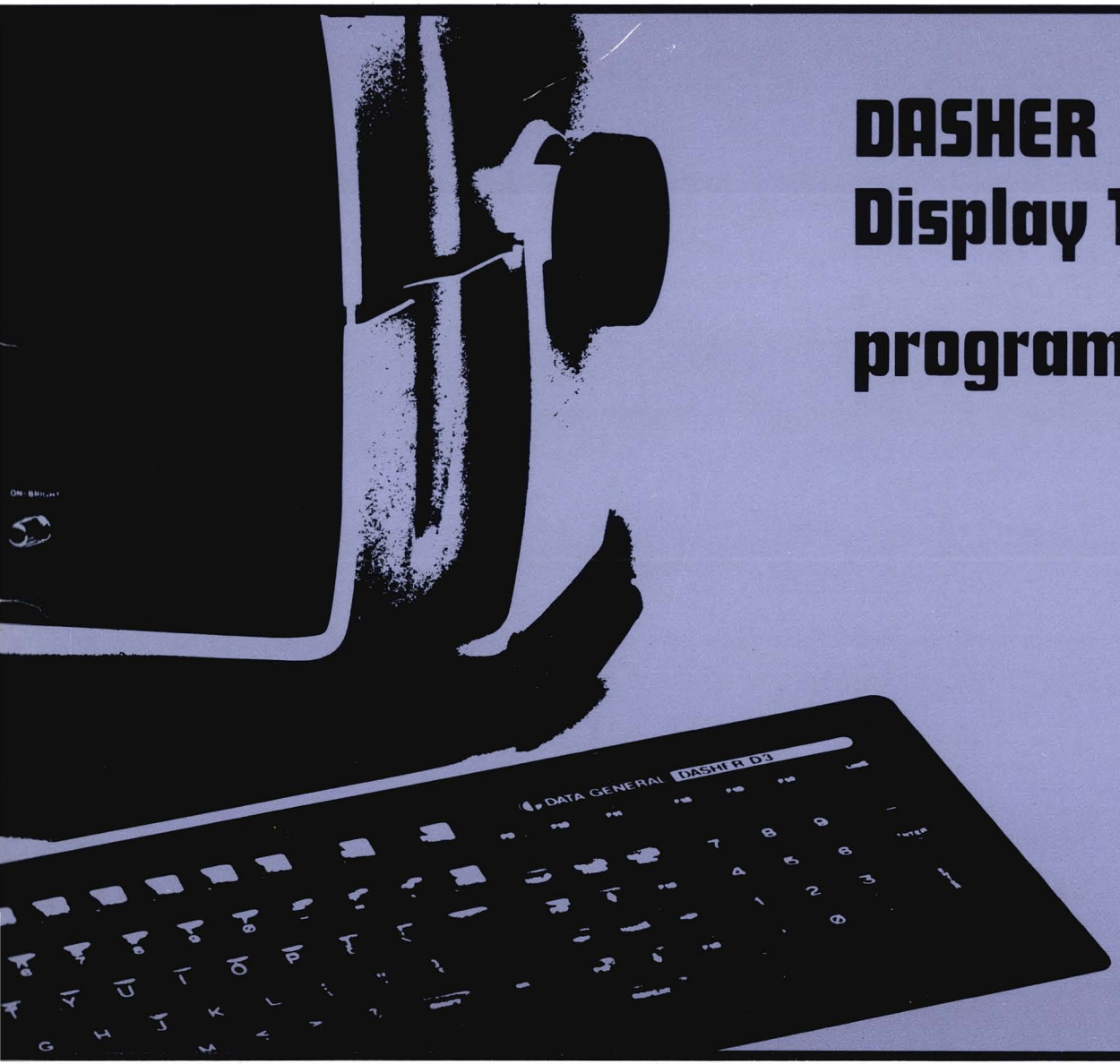


DASHER D3
Display Terminal
programmer's reference

Data General

© Copyright Data General Corporation, 1979. All Rights Reserved

Data General Corporation (DGC) has prepared this manual for use by customers, licensees, and DGC personnel. The information contained herein is the property of DGC and shall not be reproduced in whole or in part without prior written approval.

DGC reserves the right to make changes in specifications and materials contained herein without prior notice. DGC shall not be responsible for any damages, including consequential damages, caused by reliance on the information presented, or resulting from errors, including but not limited to typographical, arithmetic, or listing errors.

NOVA and ECLIPSE are registered trademarks of Data General Corporation. DASHER and microNOVA are trademarks of Data General Corporation.

Printed in U.S.A.

DASHER D3 Display Terminal programmer's reference

055-203-01

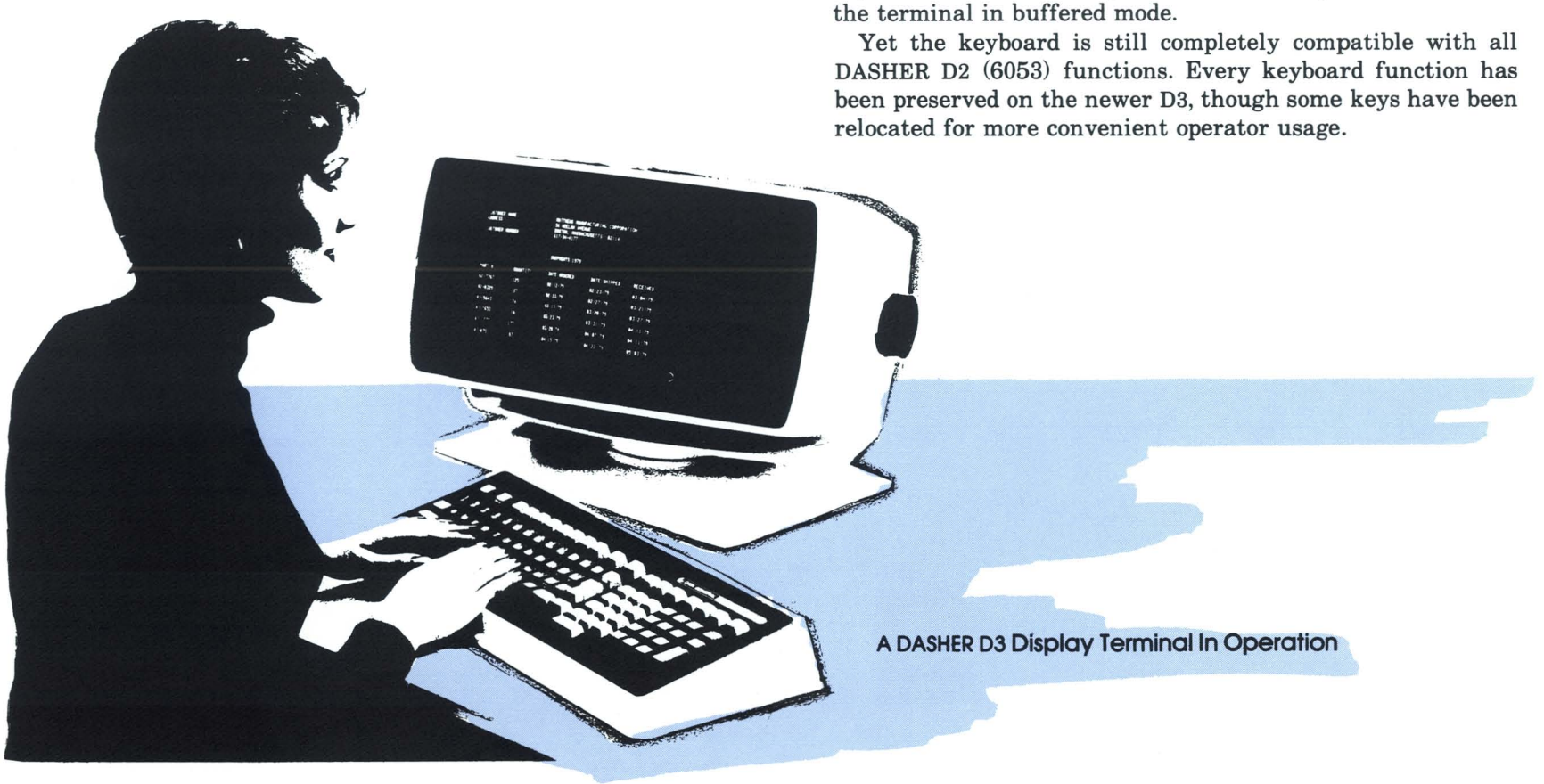
DATA GENERAL CORPORATION, WESTBORO, MASSACHUSETTS 01581

Table Of Contents

The DASHER D3 (Model 6093) Display Terminal preserves the basic layout of previous DASHER terminals with its tilting, swivelling screen, and separate keyboard.

The D3 keyboard incorporates significant improvements. Its keys are sculptured for better operator "feel". There are more user-defined function keys (18 keys for 72 functions). Dedicated keys for ENTER, NEGATIVE ENTER, and MINUS sign have been added next to the numeric pad. Legends for typical word-processing editing functions have been screened on the key fronts. These editing functions are implemented within the terminal in buffered mode.

Yet the keyboard is still completely compatible with all DASHER D2 (6053) functions. Every keyboard function has been preserved on the newer D3, though some keys have been relocated for more convenient operator usage.



A DASHER D3 Display Terminal In Operation

1. General Information

INTRODUCTION

The DASHER D3 (Model 6093) is a video display terminal that can be operated either interactively with a host computer system or as a buffered terminal of the host system.

The D3 features ease of use and operator convenience. The detached, sculptured, typewriter-style keyboard is designed for easy, simplified operation. On the display, screen management is facilitated with a set of programmable character attributes, pertaining to both display and terminal function. The special 32-character graphic set aids the creation of forms and other graphics.

The DASHER D3 functions are a superset of the DASHER D2 (Model 6053) Display Terminal functions. Thus, it can be used as a replacement for the DASHER D2 Display Terminal without modification of system software.

Programming for the D3 is easier and system overhead is reduced. Powerful interactive instructions initiate various screen operations which are now processed in the terminal. Even greater processing efficiencies are obtained in the buffered mode, where the host is completely freed of servicing the terminal's data input and display until the operator has entered and edited a full screen or record of data.

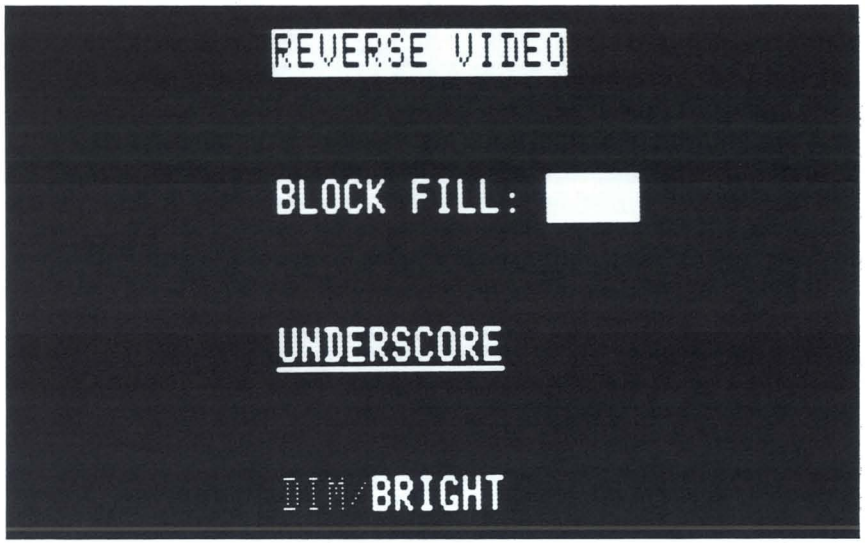
As an option, the D3 can have a direct-connected printer which can print data from either the terminal's screen or from the host as a pass-through function of the terminal.

FUNCTIONAL FEATURES

Display

The display has a storage capacity for 1920 characters which are presented on the screen in a 24-line, 80-characters-per-line format. Under control of the host program, data can be displayed with the following attributes: dim, blink, reverse video, underscore, data protected, and data modified (see Figure 1-1).

1-1 Sample Display of Character Attributes

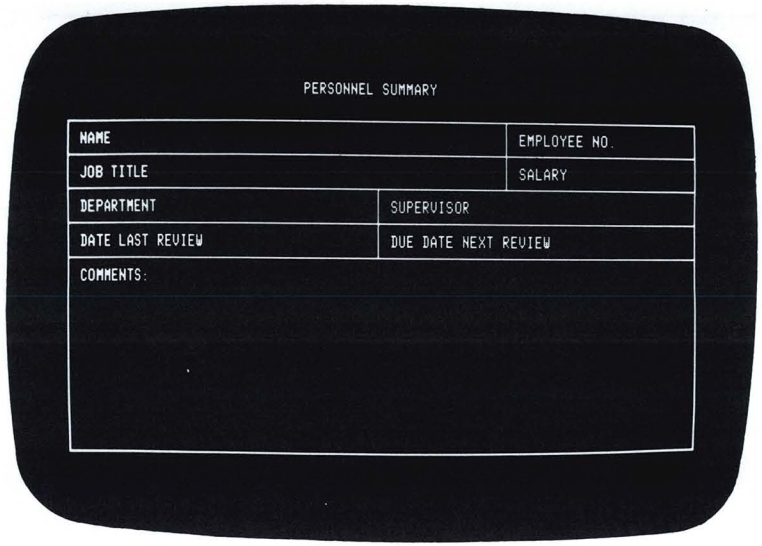


These attributes are not only a visual aid to the operator but also enable selective reading of screen data by the host, the defining of protected areas and data fields, and the printing of selected data from the screen. An example of such an application is shown in Figure 1-2. Here, a form has been created using protected fields for form areas not available for data entry. Areas where the operator should enter data are unprotected. Note the use of the special graphic characters in outlining form areas.

Keyboard

The keyboard is separate from the display terminal, allowing it to be easily positioned for operator convenience. In addition, the keyboard has been organized for minimum movement in locating the most-used keys (see Figure 1-3) and the variation

1-2 Example Of A Prepared Form



of key characteristics (color, size, shape, and height) make searching simple. Sculpturing of the individual keys contributes an enhanced "feel" to the keyboard. Deeper depressions on the "F" and "J" keys aid touch typing.

An 18-key extended function array allows for the programming of up to 72 different functions. For buffered mode, a special set of editing functions is accessible using the CMD key (see description under heading "Editing Keys"). The extended numeric keypad parallels a standard calculator layout, further simplifying use.

The keyboard also has functional compatibility with the DASHER D2 keyboard, e.g., the same codes can be generated, even though some key locations are different.

Interactive Operation

In the interactive mode, the display portion of the terminal is, in effect, separated from the keyboard by the host computer. The input from the keyboard must go to the host and be echoed back to the display. This gives the host immediate control over the processing of the input data as well as its display on the screen.

In addition to the usual interactive instructions, the terminal has several powerful features that reduce host programming as well as processing overhead. A selective Read Screen command reads only those characters with specified attributes and within specified screen areas. This instruction allows selection of only that data which has been entered or modified by the operator. A Block Move/Fill instruction enables moving a block of characters from one screen area to another. If desired, the vacated area can be filled with a designated fill character.

Several other instructions are concerned with the control and use of the character attributes and the definition of fields on the screen.

The 32-character special graphic set is functional in the interactive mode and enables the creative development of a multitude of graphic designs.

Buffered Operation

In the buffered mode, the terminal functions independently of the host during data entry by the operator. As data is keyed, it is stored and immediately displayed on the screen. Upon completion of the data entry operation (up to 1920 characters can be stored), the operator advises the host to read the screen, thereby transferring the data to the host.

The buffered mode is also well suited for data entry applications where forms must be filled in by the operator. In a typical "fill in the form" application, the host supplies the screen format to the terminal, and then turns control over to the terminal in buffered mode for operator data entry.

Editing Keys

A powerful set of screen editing functions is available for the buffered mode. These are accessed using the CMD key in con-

junction with various centrally-located data keys (see Figure 1-3). Using these functions, the operator can easily insert or delete data, move the cursor from field to field, or prepare tabular columns.

Soft Sysgen

The setting of the terminal's operating parameters, (like baud rate, type of parity, tab interval, etc.) has also received special attention. The D3 incorporates a soft Sysgen feature, instead of hardware configuration methods using switches, jumpers, etc. A special non-volatile Sysgen memory stores the various operational and system configuration parameters. These are easily redefined from the terminal's keyboard.

One parameter has particular significance. A selectable "transmit character pacing rate" can be tailored to the host interrupt response, independently of the transmit baud rate. Therefore, higher baud rates can be used with attendant reduction in processing overhead.

Printer Support

As an option, a serial printer can be connected directly to the terminal. With this option, data from either the whole screen or selected portions, depending on cursor location and individual character attributes, can be printed. In addition, the host can use the terminal as an output channel (or pass-through device) to send data to the printer. In this manner, lines containing up to 132 characters can be printed. Importantly, screen data is not affected when the "pass-through" function is performed.

PROGRAMMING ARCHITECTURE

The DASHER D3 incorporates a number of features that can reduce the complexity of the host program and reduce the communications overhead on the system.

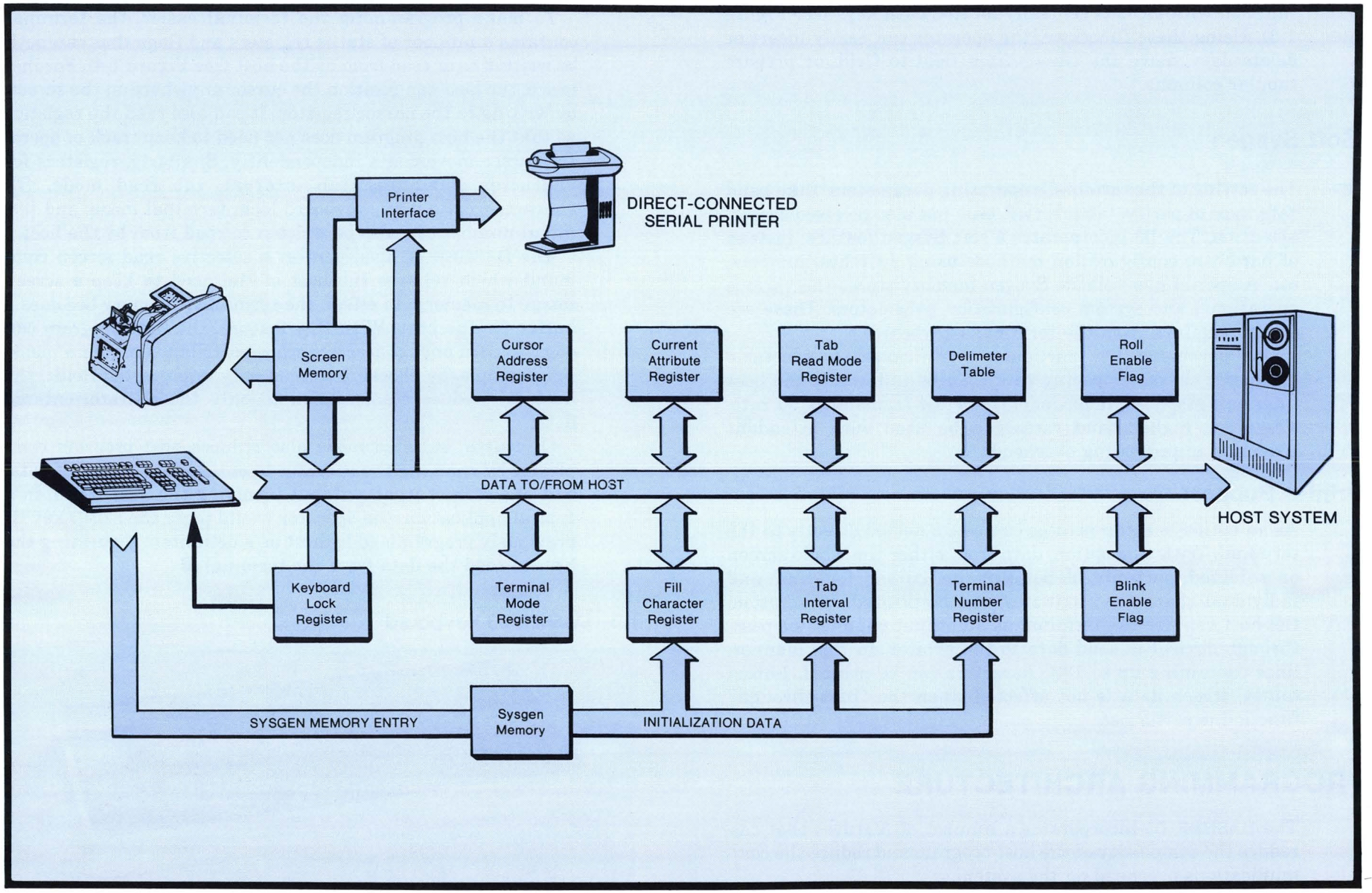
To make programming the terminal easier, the terminal contains a number of status registers and flags that can both be written to or read from by the host (see Figure 1-4). For instance, the host can position the cursor anywhere on the screen by writing to the cursor register. It can also read the register, so that the host program does not need to keep track of operator cursor movements independently. Similarly, registers for character attributes, tab interval, tab read mode, fill character, delimiters, keyboard lock, terminal mode, and terminal number can also be written to/read from by the host.

The DASHER D3 incorporates a selective read screen command which relieves the host of the need to keep a screen image in memory. In effect, the terminal's memory becomes a buffer for the host. With this feature, the host program can elect to read only characters whose attributes match a mask. For example, by choosing to read only unprotected fields, the host can reduce transmission to only the operator-entered data.

Of course, buffered mode also reduces host program complexity by enabling the operator to enter a screen full of data, edit and correct it, all without involving the host. Then, in a typical application, the operator would press the SEND key (if previously programmed by host as a delimiter) requesting the host to read the data from the terminal. ■

1-3 DASHER D3 Keyboard

1-4 Terminal As It Appears To The Host Program



2. Operator Controls And Procedures

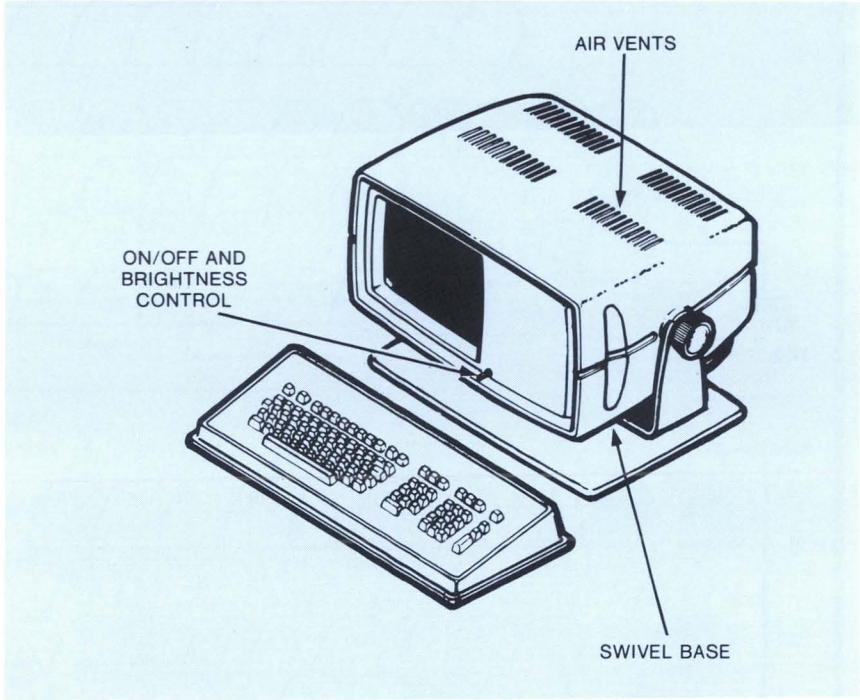
INTRODUCTION

The terminal serves as a communications link between the operator and the host computer. In this, it can function in two different modes, a direct full-duplex interactive mode and a buffered mode.

In the interactive mode, data entered from the keyboard goes directly to the host as each character is keyed. Data that is displayed is returned by the host system. A powerful instruction set allows the host to control terminal operation for such functions as display character attributes, screen/field management, data entry/editing, read screen, and data transfer to printer.

In the buffered mode, the terminal stores the individual characters as they are keyed. The data is immediately displayed for operator viewing, and the powerful screen editing functions available in the buffered mode facilitate correction of the data. Up to 1920 characters (referred to as a page or screen) can be stored and displayed before transfer to the host is required. The operator terminates a page and puts the host back on line to the terminal by pressing a delimiter key. The host can respond by giving a read screen command to initiate the data transfer.

2-1 DASHER D3 Display Terminal



OPERATOR FACILITIES

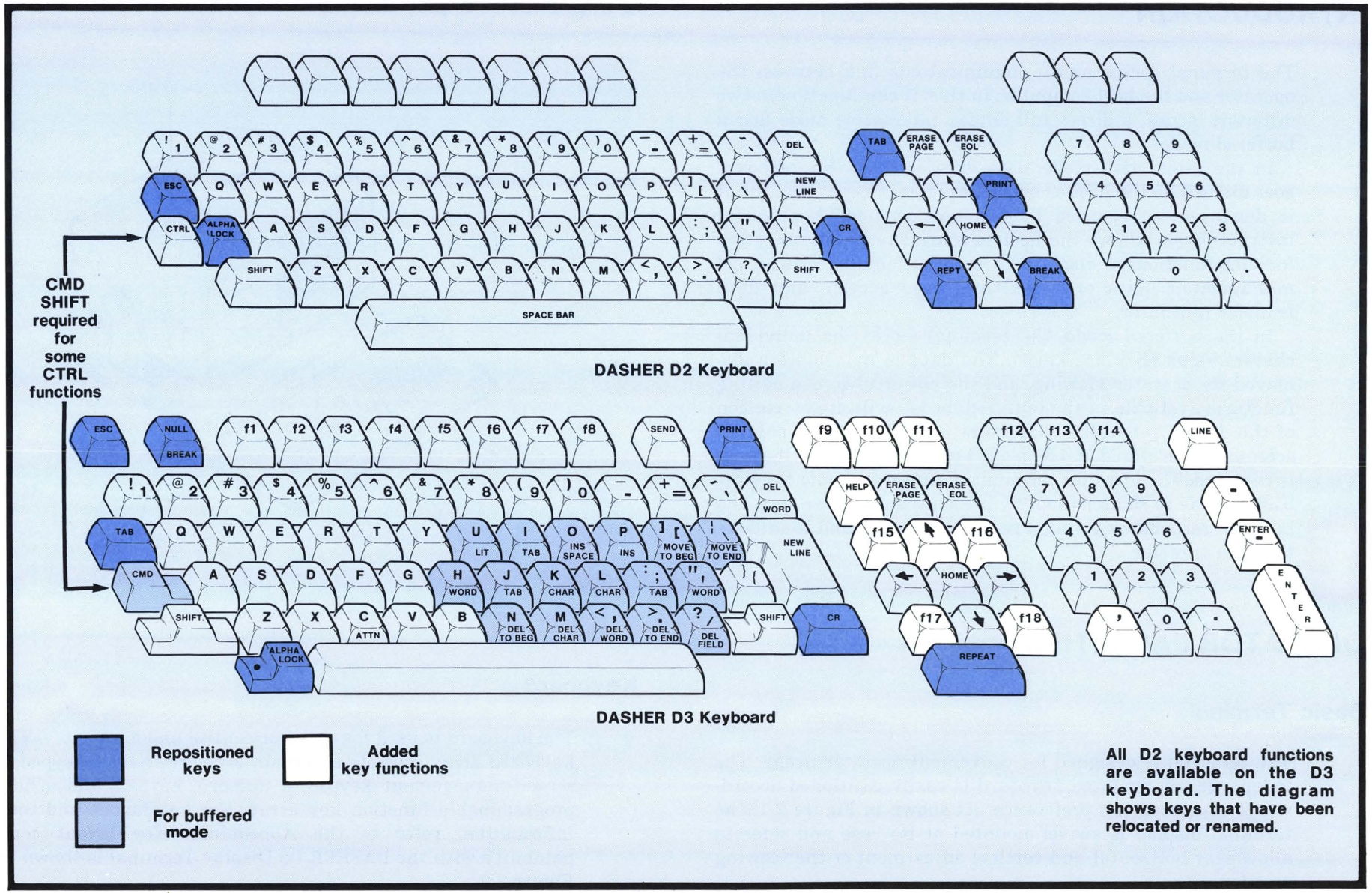
Basic Terminal

The terminal is designed for convenient operator usage. The independently-mounted keyboard is easily positioned according to the operator's preference. As shown in Figure 2-1, the terminal display is swivel-mounted at its base and sides to allow easy horizontal and vertical adjustment of the viewing position.

Keyboard

The keyboard is used for operator control and data entry. The keyboard array consists of a main typewriter-style keypad, a screen management keypad, a numeric keypad, and a host programmable function key array. For key layout and code information, refer to the Appendices. Key layout compatability with the DASHER D2 Display Terminal is shown in Figure 2-2.

2-2 Comparison of DASHER D2 and D3 Keyboards



Display

With exception of the audible alarm, the display is the only means for communication from the terminal and host system back to the operator. In general, the display provides an alphanumeric read-back of the operator-entered data, either directly from the keyboard, if in buffered mode; or as an echo back from the host system, if operating in the interactive mode. Other host-supplied data, resulting either from an operator request for previously stored data or from host initiated messages, is also displayed. Note that various screen messages can be programmed for the interactive mode to assist the operator. These can be either informative or interrogative in nature and can be constructed as desired to support system usage.

Audible Tone (Bell)

The terminal can generate an audible tone (called a bell) to alert the operator when an attempted function is not allowed. In interactive mode, the bell is activated by an instruction from the host. In the buffered mode, the bell sounds when an illegal function is keyed.

Power-On/Brightness Control

The power switch and CRT screen brightness control are combined in a single control knob. Pulling this knob out turns the power on; pushing it in turns the power off. Rotating the knob clockwise makes the screen image brighter; rotating it counter-clockwise makes it darker.

> CAUTION
> To allow proper cooling of the terminal, keep the top air vents uncovered when power is applied.

Printer Option

As an option, a DASHER Printer can be connected directly to the terminal. With this option, the host can supply data to the printer in two ways. One is to send screen data, either the whole screen or in part, directly from the terminal to the printer. The other is to send data from the host to printer via the terminal, but without effect upon the screen data.

GENERAL OPERATING PROCEDURES

Power-On/Initialization

With application of power, the terminal initializes itself, including the reading in of Sysgen parameters, and goes into the interactive mode for direct operation with the host system. For information on selecting Sysgen parameters, refer to Appendix A.

Selecting Operating Mode

Initially upon power-up, the terminal is in the interactive mode. Since the buffered mode can only be selected by the host system, operator procedures are dependent upon host programming. Once in the buffered mode, the only means the operator has to contact the host is by keying a delimiter. The host can then return the terminal to interactive mode for required servicing of the terminal, e.g. reading screen data, interactive communication with operator, etc.

Direct Interactive Operation

The actual operating procedures for interactive operation with a host system is inherently dependent upon the program run by the host.

Buffered Operation

The buffered mode is designed for the entry and editing of data by fields. These fields can be of two types: protected and unprotected.

A protected field is a portion of screen area that has all its data protected against operator entry (see discussion of Protected Data attribute under heading "Character Attributes" in Chapter 3).

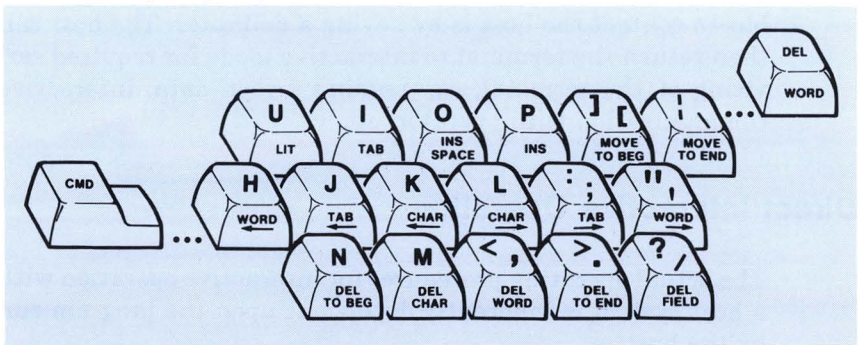
An unprotected field can be either a full screen line of unprotected data or a portion of a line between a protected field and another protected field. The beginning or ending of a line is also a field boundary.

> CAUTION
> When operating in the buffered mode, take care not to press the NULL key as this will lock up the terminal. To recover from such a situation, turn power off and then on again.

Editing Functions (Buffered Mode)

The buffered mode has a set of powerful screen editing functions such as delete and insert. These functions are accessed using the CMD key in conjunction with centrally-located data keys (see Figure 2-3). These functions are described below.

2-3 Editing Key Layout



Cursor Movement

MOVE CURSOR RIGHT ONE CHARACTER **CMD L**

Moves the cursor right one position; if already at end of line or unprotected field, then the bell sounds. Cursor cannot enter beyond first location of unfilled area.

MOVE CURSOR RIGHT TO NEXT TAB STOP **CMD ;**

Moves the cursor to next tab stop or end of entered data, whichever comes first. If cursor is at end of entered data in the field, then the bell sounds and no action occurs.

MOVE CURSOR RIGHT TO NEXT WORD **CMD '**

Moves cursor to beginning of next word or to the end of entered data in the field. If at end of entered data, field, or line, the bell sounds and no action occurs.

MOVE CURSOR TO END OF FIELD **CMD **

Moves cursor to last position in the field or to the first location of unfilled area in the field.

MOVE CURSOR LEFT ONE CHARACTER **CMD K**

Moves cursor one position to the left. If cursor is at beginning of the field, the bell sounds and cursor position remains unchanged.

MOVE CURSOR TO PREVIOUS TAB STOP **CMD J**

Moves cursor to previous tab stop or to beginning of the field. If cursor is at beginning of a field, the bell sounds.

MOVE CURSOR TO BEGINNING OF PREVIOUS WORD **CMD H**

If the cursor is in the middle of a word , it moves to beginning of the current word. If the cursor is at the beginning of a word or between words, it moves to the beginning of the previous word. If cursor is at the beginning of a field, the bell sounds.

MOVE CURSOR TO BEGINNING OF FIELD **CMD [**

Moves cursor to beginning of the current field.

MOVE CURSOR TO NEXT FIELD **CMD X**
(or → on Cursor Control Pad)

Moves cursor to first location in next unprotected field. If there are no protected fields, the cursor moves to beginning of the next line. If in last field of screen, cursor moves to the first unprotected field at top of screen.

MOVE CURSOR TO PREVIOUS FIELD **CMD Y**
(or ← on Cursor Control Pad)

Moves cursor to first character location of previous unprotected field.

Insert Functions

COMPLEMENT INSERT MODE **CMD P**

Alternately selects with each keying either the Insert Mode or the Overstrike Mode. The Insert mode is disabled (giving Overstrike Mode) upon entry to buffered mode or whenever a new field is entered. In Insert mode, as each key is struck (except for control keys), the characters starting at and to right of the cursor shift one position to the right. The character struck is inserted at the cursor location and the cursor moves one position to the right. Any character in the last position in the line or field is lost when the shift occurs. When in Overstrike Mode, each new data entry writes over and destroys existing data at the cursor location.

INSERT TAB **CMD I**
(or TAB key)

Characters at and to right of the cursor are shifted right to the next tab stop and the resulting space is filled with space codes.These spaces are flagged and are replaced by a single tab character before the data is transmitted to the host. However, if a Replace Tab instruction (from host) is stored in the Tab Read Mode Register, then the space codes are retained and transmitted when the Read Screen instruction occurs. The bell also sounds whenever a full field condition exists and additional Insert Tab functions are keyed. The cursor stays at end of tab field.

INSERT SPACE **CMD O**

Characters to the right of the cursor are shifted right one position and a space is inserted. If the field is full, the bell sounds and no action occurs.

Delete Functions

DELETE CURRENT CHARACTER **CMD M**

Deletes the character at cursor location and shifts subsequent characters left one position. The last position in the field is filled with the fill character. If the cursor is at end of the field, the last character is deleted and subsequent deletions will sound the bell and no other action occurs.

DELETE PREVIOUS CHARACTER **DEL**

The character previous to the cursor location is replaced with a space code. If at end of entered data, the current fill character is the replacement character. If the cursor is at last character in a full field, then the character at the cursor location is replaced with a fill character. If the cursor is at the beginning of the field, the bell sounds and no action occurs.

DELETE NEXT WORD **CMD ,**

Deletes characters from cursor location to beginning of next word. Subsequent characters in the field shift left to close up space and fill characters are inserted at the end to complete the field. If cursor is at end of the field, the bell sounds and no action occurs.

DELETE TO BEGINNING OF FIELD **CMD N**

Deletes characters between cursor and beginning of field. Subsequent characters in field and cursor are shifted left to beginning of field. Fill characters are inserted at end to complete field. If cursor is at beginning of the field, the bell sounds and no action occurs.

DELETE TO END OF FIELD **CMD .**

Inserts fill characters from cursor location to end of field. If cursor is at end of field or at end of entered data, the bell sounds and no action occurs.

DELETE ENTIRE FIELD **CMD /**

Entire field is filled with fill characters and cursor is repositioned to beginning of field.

DELETE PREVIOUS WORD **CMD DEL**

If cursor is pointing to a word separator (space code or any punctuation code) or to the first character of a word, the previous word is deleted; otherwise the same word is deleted. Subsequent data along with cursor are shifted left to close up space. Fill characters are inserted at end to fill the vacated locations. If cursor is at beginning of field, the bell sounds and no action occurs.

Delimiter Keys

By augmenting the standard buffered mode features of the terminal with delimiter-accessed interactive operations with the host, the function of the terminal can be as versatile and powerful as desired. In this manner, functions such as: moving a block of data, deleting a block of data, word search and replace, etc. can be implemented from buffered mode.

When the operator strikes a delimiter key, the keyboard is locked from further function until released by the host program. The terminal also sends the delimiter code to the host where the host program can discriminate which delimiter was keyed and, in turn, issue an appropriate response (e.g. a read screen operation).

The specific handling and functional dedication of the delimiters are as programmed into the host software. The host software defines to the terminal what key codes are delimiters (if none are defined, NEW LINE and CR keys are delimiters by default). Thus, up to 256 different codes may serve as delimiters.

Printer Option

The host can request the terminal to send data to an associated printer in three ways:

1. A PRINT command transfers all data starting from the leftmost character on the cursor line and ends with the last character on the screen.
2. A PRINT FORM command transfers all unprotected data to the printer; or if the Sysgen is configured for 6053 compatibility, the command transfers all the full intensity characters.
3. A PRINT LINE command transfers a host-prepared line of data via the terminal to the printer. Two advantages are obtained with this command. One, line lengths in excess of 80 characters and up to 132 can be printed. The other, data on the screen is not affected.

The specific operator procedures for initiating these print operations are dependent upon the host programming. Conceivably, the PRINT key, both with and without the SHIFT key, would be used. Of course, other keying could also be used. ■

3. Programming

INTRODUCTION

This chapter provides the system programmer with reference information to be used in preparing an interface program for the DASHER D3 Display Terminal. The chapter starts out with basic information, covering functional aspects of the terminal which are of concern to the programmer. Subsequent chapter areas describe the instructions that the host can use to control terminal operations. To wrap-up the chapter, two programming examples for using buffered mode are shown. Together, they contain the bulk of the kind of screen structures that will typically appear in business applications. The first example is of a form. The second is also a form, but with tab fields. Of course, these elements can be combined in all kinds of variations to accommodate specific applications.

For condensed code and instruction listings, refer to the back cover.

OVERVIEW OF TERMINAL OPERATIONS

The terminal functions as a combined data entry and display device for an associated host computer system. In this, it receives control and conditioning from the host system and responds to keyboard entry by generating appropriate ASCII codes. How these codes are displayed is dependent on which of two basic operating modes is being used. The two modes are: interactive mode and buffered mode.

In interactive mode, the codes are individually transferred to the host as each key is pressed. In turn, the host system echoes each data code back to the terminal for display.

In buffered mode, as each code is generated it is stored in the terminal and immediately displayed by the terminal for operator viewing. No host action is required. After the data has been corrected and edited, the operator can transfer control to the host. Generally, the host will read the accumulated data.

From the viewpoint of the host system, certain internal functions of the terminal are of concern to the programmer. Basically, these concerns relate to control of the terminal itself and also to the display of data on the screen. The following paragraphs discuss some of these concerns.

Control Registers and Flags

Several registers internal to the terminal are associated with functions controllable by the host. Among these registers are six status registers which can be written to or read out of by the host. The status registers are: Terminal Mode, Keyboard Lock, Tab Read Mode, Tab Interval, Fill Character, and Terminal Number.

Three other registers also available to the host are: the Delimiter Table, the Current Attribute Register, and the Cursor Address Register.

In addition, two status flags enable character blinking and Roll Mode (scrolling of the screen). Basic functions of the various registers and flags are as follows:

- Terminal Mode Register — Identifies the operating mode of the terminal. The codes are: 000 for interactive mode; 177 for buffered mode.
- Keyboard Lock Register — Identifies whether keyboard entry is permitted. The code 000 = keyboard unlocked; 177 = keyboard locked.
- Tab Read Mode Register — Indicates whether tab codes will be sent to host on a screen read operation or if the codes will be replaced with an appropriate number of space codes. The codes are: 001 for replace tab codes; 177 for send tab codes.
- Tab Interval Register — Stores the octal value (number of character positions) between tab stops.
- Fill Character Register — Stores the ASCII code for the fill character.

- Terminal Number Register — Stores the assigned device number (octal) of the terminal.
- Delimiter Table — Defines up to 256 delimiters which can be used by the operator in buffered mode operation to transfer control to the host. Any transmitting key can be defined as a delimiter.
- Current Attribute Register — Stores seven bits which define the screen attributes for current data entry. Function of the individual attributes is described under the heading: Character Attributes.
- Cursor Address Register — Stores the current screen address of the cursor. Any change in cursor location is obtained by writing a new address into the register.
- Blink Enable Flag — When set, enables any character with the blink attribute bit set to blink. Conversely, when cleared, all blinking (except for cursor) is disabled.
- Roll Enable Flag — When set, enables the screen to roll (scroll) a line each time the cursor is moved beyond the 24th line. When the flag is cleared, roll mode is disabled and page mode is enabled. In the page mode, when the last screen location is exceeded, the cursor moves to the top of the screen.

Display Concept

The screen display can consist of up to 24 lines of 80 characters each. In buffered mode the data is immediately displayed as it is keyed. In interactive mode, the host must echo the keyed data codes back to the terminal for display.

Cursor Addressing

As each data code is received, it is displayed at the current cursor address and then the cursor is advanced to the next screen location. The cursor address is stored in the two-byte Cursor Address Register.

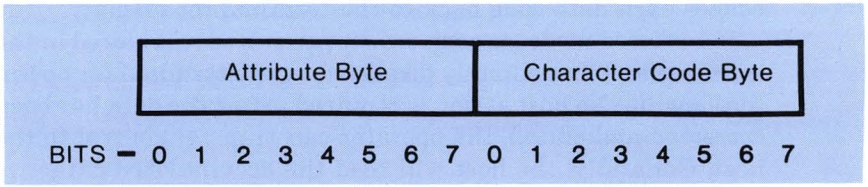
In interactive mode, the cursor can be relocated to any addressable screen location by a command from the host. The screen address is defined by a three-byte sequence; the first byte, octal 020 (037, if response from terminal), identifies that a cursor address follows. The second byte defines the column address (horizontal character position) on the screen and can consist of any number from 0 to 79 (octal 000 to 117). The third byte defines the row address (vertical line position) on the screen and can consist of any number from 0 to 23 (octal 000 to 027). If desired, the host can read the cursor address and from that calculate the repositioning.

In buffered mode, the cursor location is controlled directly from the keyboard.

Character Attributes

The individual characters on the display can have various attributes, such as blink, dim, underscore, etc. As each character is received, its attributes are defined according to the contents of the Current Attribute Register. To accomplish this, the display data is stored in the terminal in a two-byte format, with one byte containing the character code and the other byte containing the attribute information. The bit arrangement is shown in Figure 3-1 (also see Table 3-2).

3-1 Display Character Bit Arrangement



3-2 Attribute Byte Bit Assignment

BIT	NAME	FUNCTION
7	block fill	full character space is illuminated.
6	blink	displayed character blinks on and off.
5	reverse video	displayed character is black on white.
4	dim	displayed character has reduced intensity.
3	underscore	displayed character is underscored.
2	protected data	indicates character cannot be changed from keyboard.
1	modified data	indicates character code has changed. It is set by terminal when any data in field has changed.
0	tab stop	indicates a tab stop location (not a character attribute) which is set by the terminal according to the contents of the Tab Interval Register. Not accessible by the host.

Note that for the character byte, bits 1 thru 7 give the character code and bit 0, when set, defines a tab space which is sent to the host on a screen read operation.

Attribute bits 1 thru 7 are controllable from the host. Bit 0 of both the attribute byte and the character byte are controlled from within the terminal (see description of tabs under heading "Tab Stop/Read Functions"). In operation, a current attribute register in the terminal stores the current host-defined attributes. These are imparted to the new data as it is entered from the keyboard. In addition, the modified data attribute is set by the terminal (see description below). Also see the Field Control instructions for additional information on attributes.

In the set attribute commands, a combination of masking and XOR functions are used to achieve the desired bit configurations. Each bit of the current attributes is ANDed with the corresponding bit of the mask byte and then exclusive ORed with the XOR byte to produce the new current attributes. The results can be determined as follows:

Mask Bit	XOR Bit	Resulting Attribute Bit
0	0	0 (cleared)
0	1	1 (set on)
1	0	no change
1	1	complement

In addition to the capability of defining all the attributes at once with a direct write to the Current Attribute Register, they can be individually defined by specific commands from the host. Also, a Set Attribute String command can redefine the attributes of a specified screen area. Other commands provide for reading either current attributes or those of a specified area on the screen.

The various attributes are individually described in the following paragraphs.

Character Blinking — Blinking of displayed characters is defined on an individual character basis by bit 6 of the character's attribute byte. Whether the characters actually blink is determined by the Blink Enable Flag. Blinking is enabled when the flag is set; conversely, with the flag cleared, blinking is prohibited.

In addition to the appropriate blink commands, each time an ERASE PAGE command is generated the Blink Enable Flag is set and the display is blink enabled.

Character Dimming/Underscoring — Bits 3 and 4 of the individual character's attribute byte control underscoring and dimming, respectively. Thus, each character whose underscore

bit is set will be underscored on the display. Similarly, if the dim bit is set, the character will be displayed with reduced intensity.

An ERASE PAGE instruction clears both the dim and underscore bits in the Current Attribute Register.

Block Fill — The Block Fill attribute is controlled by bit 7 of the character's attribute byte. When the bit is set, the complete character field is illuminated with the result that the displayed image of the associated character is effectively masked out.

Reverse Video — The Reverse Video attribute is controlled by bit 5 of the character's attribute byte. When the bit is set, the character image is outlined in black on an illuminated background.

Protected Data — The Protected Data attribute is controlled by bit 2 of the character's attribute byte.

In buffered mode, the operator is not allowed to modify any character which has its Protected Data attribute bit set. This is accomplished primarily by not allowing the cursor to be positioned in any protected area of the screen.

In interactive mode, the cursor can be positioned any place regardless of the protected status of a character. This allows the screen to be formatted in preparation for entering buffered mode.

The correct procedure for modifying a protected character is to reset the Protected Data attribute bit to 0, modify the character and then set the Protected Data attribute bit to 1, if desired.

Any software manipulating screen data in interactive mode should also keep track of the cursor position relative to protected data, as output could be lost in these areas. Also, some character sequences could modify unwanted areas of the screen.

Modified Data — The Modified Data attribute (bit 1 of the character's attribute byte) is set for all characters in the field whenever any change is made to existing data in the field. This is essentially a flag to enable selective reading of screen data by the host. The Modified Data attribute is set by the terminal and can only be cleared by an instruction from the host.

Tab Stop/Read Functions

Tabs are treated in one of two ways, depending upon the host-set value of the Tab Read Mode Register. If the register is set to 001, space codes that fill from the end of data to the next tab stop are identified within the terminal by bit 0 of the character code byte. On a read screen operation they are replaced as a group by a single tab code.

If the register is set to 177, tabbing results in the storage of ASCII space codes. On a read screen, these space codes are transmitted rather than the single tab code.

Tab stop locations on the screen are flagged inside the terminal by the setting of bit 0 on the individual character attribute bytes. The interval between stops is defined by the Tab Interval Register. The register is initialized with the Sysgen value, but can be changed by a host instruction.

Roll Mode (Scrolling)

A Roll Enable flag in the terminal defines whether the screen will be in Roll Mode (permits scrolling) or Page Mode (scrolling prohibited). This flag is set or cleared by host-supplied instructions. When the display is roll enabled, the screen rolls up one line each time the cursor overflows the bottom line or a New Line code is received when the cursor is located on the bottom line.

When the display is not roll enabled, the cursor moves to the first character position on the top line each time the cursor overflows the bottom line or a New Line code is received when the cursor is on the bottom line.

The ERASE PAGE command does not affect this flag.

Field Control

In some data entry applications, a prepared form is sent to the screen by the host system. The form can contain two types of fields, protected and unprotected.

Data entry areas of the form are called unprotected fields. These fields can be identifed to the operator by fill characters which are overwritten as new data is entered.

Fixed areas of the form which the operator cannot change are called protected fields and are identified within the terminal by the protect attribute bit.

The modified data attribute can be used to select for transfer to the host system only that data which has undergone change. After such a reading by the host, the host would most likely want to reset the modified attribute bits on data which was read so that new or modified data can again be discriminated.

Since, in buffered mode, data entry is processed internally in the terminal, the host must be informed when data is ready for transfer to the host. For this purpose, the host system can establish delimiter codes which when generated at the terminal (by the operator pressing a key) will inform the host that data should be read.

The commands that the host can use to define and control fields within the screen are described under the heading "Screen Control". Also refer to the information under headings "Character Attributes" and "Attribute Control".

Data Printing

Data can be sent to an associated printer in two ways. In one, the data transfers directly from the screen buffer of the terminal to the printer. In the other, host-prepared data is transferred through the terminal to the printer without affecting the screen data.

CODING CONVENTIONS

> NOTE
> All data and instruction codes are shown as 7-bit stripped codes without parity.

Instruction format to terminal: 036 <instruction code> plus any required data or positioning bytes. May or may not have a terminator.

Terminal response format: 036 176 <function> plus any required data or cursor coordinates and terminator.

Data entry format, keyboard to host: 8-bit ASCII code consisting of 7 data bits and 1 parity bit. Code values greater than 177 octal are in a two-code format, 036 <___>. The 036 carries a value of 200 octal. Refer to the ASCII code listing in the Appendix.

Function code format, keyboard to host: 036 <function code> The various user function codes available to the programmer are listed in Appendix C.

Cursor position format: 020 <X> <Y> where:

X = column position, limits: 0 and 79 (117 octal)
Y = row position, limits: 0 and 23 (027 octal)

> NOTE
> Cursor position data coming from terminal to host is identified by a 037 code, instead of 020.

HOST/TERMINAL INSTRUCTIONS

The following instructions can be used by the host to control terminal operations. For a conceptual understanding of the

various functions, refer to the description given under heading "Overview Of Terminal Operations". Note that instructions which are compatible with the DASHER D2 (Model 6053) Display Terminal are so indicated.

Refer to the Appendices for ASCII code listings, key layout and code assignments (including user function keys), extended graphic character symbols and codes, and Sysgen information. Also see "Programmer's Instruction Summary" on back cover.

Program Timing

Some complex commands require that the terminal's internal processor execute extended routines (e.g., the ERASE PAGE command). At high baud rates, the execution time of these commands will exceed the minimum transmission periods from the host. All characters received from the host, both command and data, are internally buffered in the terminal. This buffering is sufficient to prevent loss of data or commands under any rational operating condition.

It is possible, however, to deliberately cause the loss of data by repeating complex commands at high baud rates. Two examples are:

- At 19.2 Kbaud, if more than six ERASE PAGE commands were transmitted successively followed by a long data stream, some of the data could be lost.
- At 19.2 Kbaud, in roll mode, if a sequence of NEW LINE followed by one or two data characters were repeated several hundred times, data could eventually be lost (in this example, the screen would be scrolling at about 500 lines per second).

If for some unusual reason, such a sequence of complex commands is required, the programmer must insert appropriate delays.

Terminal Management

MASTER RESET 036 025

Keying: SHIFT LINE (terminal reset from keyboard).

Re-initializes registers and flags in the terminal and loads Sysgen parameters. Unlocks keyboard. Refer to Table 3-3 for initialize conditions. An automatic reset also occurs on application of power to the terminal.

> NOTE
> Since the Master Reset instruction from the host has a relatively long execution time, the host program should wait at least one second before transmitting further commands.

3-3 Initialize (Master Reset) Conditions

REGISTER/FLAG	CONDITION
Terminal Mode Register	Interactive
Keyboard Lock Register	Unlocked
Tab Read Mode Register	Disabled (reads spaces)
Tab Interval Register	Current Sysgen (default is 4)
Fill Character Register	Current Sysgen (default is space)
Terminal Number Register	Current Sysgen (default is 0)
Delimiter Table	New Line, CR
Current Attribute Register	Cleared
Cursor Address Register	000 000 (Home)
Blink Enable Flag	Enabled
Roll Enable Flag	Enabled

ID REQUEST 036 012

Requests terminal for following identification data: terminal type (040=model 6093), revision level of terminal, and assigned terminal number. Terminal Response: 036 176 043 040 <Rev.#><term.#> 036 176 057

UNLOCK KEYBOARD 036 031

Releases keyboard from locked condition. The keyboard is locked while the terminal is responding to host supplied instructions, but unlocks itself after completion of the instruction. During buffered operation, the keyboard is locked whenever a delimiter key is pressed and must be unlocked by an instruction from the host. When the keyboard is locked, data entry from the keyboard is prohibited and the bell sounds with each keystroke. Terminal Response: 036 176 057

SET INTERACTIVE MODE 036 000

Removes terminal from buffered mode and places it in interactive mode with keyboard unlocked. Upon power-up or master reset, terminal automatically goes into interactive mode, roll enabled. If page mode (non-scrolling function) is desired, it must be set by the host. Terminal Response: 036 176 057

SET BUFFERED MODE 036 001

Puts terminal into buffered page mode. While in buffered mode, keyboard entry is blocked from the host except for the programmed delimiter codes which flag the host to service the terminal. The Roll Mode (screen scrolling) cannot be used in the buffered mode.

READ STATUS 036 024

Reads contents of the six status registers in the terminal back to the host in the following order:

- terminal mode
- keyboard lock
- tab mode
- tab interval
- fill character
- terminal number

Terminal Response: 036 176 045 <byte 1> . . . <byte 6> 036 176 057

BELL 007 (D2 compatible)

A short audible tone is produced on the speaker.

Screen Control

ROLL ENABLE 022 (D2 compatible)

The display is roll enabled; that is, the screen rolls up one line each time either the cursor overflows the bottom line or a New Line character is decoded when the cursor is on the bottom line. When this occurs, the cursor moves to the first character position on the bottom line, the bottom line becomes blank, and the data previously displayed on the top line is lost.

ROLL DISABLE 023 (D2 compatible)

The roll enable mode is terminated. When the display is not roll enabled, the cursor moves to the home position each time either the cursor overflows the bottom line or a New Line code is decoded when the cursor is on the bottom line. While the cursor position changes, the information displayed on the screen remains unchanged.

READ SCREEN 036 014 <mask> <word> < XE> <YE>

Causes the selected screen data to be read back to the host. The reading starts at the current cursor position and ends at the defined XE (column) and YE (row) screen coordinates.

For a character to be read, the attribute byte when ANDed with the "mask" byte must match the "word" byte. To read all data irrespective of attributes, set both the mask byte and word byte to zero. Then all the data on the screen from cursor location to ending coordinates will be sent.

If the ending coordinates are less than the cursor location, the terminal response will define the current cursor location but will not contain any data. If the ending coordinates exceed the screen ending, then the terminal response will show data only up to the screen ending.

When the read screen operation skips locations due to improper attributes, the starting address of the new location will be read before the subsequent data string. Terminal response: 036 176 040 <037 X Y> <data string > <037 X Y> <data string> 036 176 057

SET DELIMITER TABLE 036 020 <byte 1> <byte 2> . . . etc. 036 176 057

Clears the Delimiter Table in the terminal and writes in the new delimiters. These remain in effect until changed by a new command from the host, or a master reset is received, or a new power-up cycle occurs. In the case of the last two, the default delimiters are NEW LINE and CR. Up to 256 delimiter codes can be defined. For code values greater than 177 octal, the 036 <___> format must be used, with the 036 carrying a value of 200 octal. Terminal response: 036 176 057

READ DELIMITER TABLE 036 023

Causes the delimiter codes as defined by the Delimiter Table to be read back to the host. Terminal response: 036 176 044 <byte 1> <byte 2> . . . 036 176 057

SET FILL CHARACTER 036 022 <byte>

Loads the character code into the Fill Character Register in the terminal. This code is retained until changed by a new command from the host, a master reset, or a new power-up cycle. In the case of the last two, the register is set to the Sysgen defined character. Terminal response: 036 176 057

FILL UNPROTECTED FIELDS 036 003

Causes all unprotected fields to be filled with the fill character. If there are no protected fields, then the whole screen is filled. The cursor is positioned at the first unprotected location on the screen. Attributes of the fill characters are set according to contents of the current attribute register in the terminal.

CAUTION
Do not use this function with the protected data attribute set in the Current Attribute Register.

Terminal response: 036 176 057

FILL CHARACTER STRING 036 013 <no. of locations> <insert character>

Causes the specified number of locations starting at cursor to be replaced with the insert character regardless of the Protected Data attribute. If the specified number of locations exceed the line, then the fill operation terminates at line end. Cursor position remains unchanged. Attributes of the inserted characters are set to the current attributes. Terminal Response: 036 176 057

INSERT CHARACTER 036 010 <no. of characters> < insert Character>

Causes the specified number of characters to shift right one location (regardless of the Protected Data attribute) and the insert character to be inserted at cursor location. The last character of the string is deleted. If the number of characters specified exceeds the line, the operation terminates at line end. Attributes of the inserted character are set to the current attributes. Cursor remains at starting position. Terminal Response: 036 176 057

BLOCK MOVE/FILL 036 006 <XD> <YD> <XE><YE>

where: <XD> <YD> = <COL><ROW> for destination of move; <XE> <YE > = <COL> <ROW> for end location minus one of string to be moved.
Causes a block of data (characters and attribute bytes from cursor location to defined end location minus one) to be moved to defined destination. If the destination is after the cursor position, the block moves to the right and downward on the

screen. Conversely, if the destination is before the cursor location, the block moves left and upward on the screen. In either case, the vacated spaces are loaded with the fill character and the cursor is positioned at the first fill character location. Terminal response: 036 176 057

DELETE CHARACTER 036 011 <no. of characters>

Causes the specified number of characters starting at cursor location to shift left one position. In this operaton the character at cursor location is deleted and a fill character is inserted in the vacated location at end of string. If number of specified characters exceed the line end, the operation terminates at line end. Terminal Response: 036 176 057

ERASE TO END OF LINE 013 (D2 compatible)

Clears the screen from cursor location to end of line.

ERASE PAGE 014 (D2 compatible)

Clears the screen and moves the cursor to the home position. Additionally, this command sets the Blink Enable flag, and clears all the attributes in the Current Attribute Register. Also, see timing concerns described under heading "Program Timing".

Attribute Control

SET CURRENT ATTRIBUTE REGISTER 036 017 <mask byte> <XOR byte>

Sets the Current Attribute Register in the terminal to desired attribute configuration. See the masking/XOR description under heading "Character Attributes". Note that the blink attribute is not functional unless the Blink Enable flag is set in the terminal. Also, an Erase Page command from the host will clear the register, therefore reconfiguration will be required. This instruction can be used instead of the individual set and clear instructions (016, 017, 024, 025, 034, 035). Terminal response: 036 176 057

READ CURRENT ATTRIBUTE REGISTER 036 016

Reads contents of the Current Attribute Register in the terminal and transfers the information back to the host. Terminal response: 036 176 042 <register byte> 036 176 057

READ ATTRIBUTE STRING 036 015 <XE> <YE >

Reads the attributes of a string of characters starting at cursor location and ending at the specified XE (column) and YE (row) coordinates of the screen. The cursor stop position is at the XE and YE ending coordinates. If the XE and YE coordinates exceed the screen, then the function stops at last screen location. Terminal response: 036 176 041 <byte><byte > 036 176 057

SET ATTRIBUTE STRING 036 002 <#chars.> <mask byte> <XOR byte>

Redefines the attributes of a specified number of display characters starting at cursor location. If the number of characters specified is 0, then the whole screen from cursor on will be affected. Upon completion of the instruction, the cursor remains at ending location. See description of masking/XOR functions under heading "Character Attributes". Terminal response: 036 176 057

ENABLE BLINK 003 (D2 compatible)

Each character whose blink attribute bit is set is blinked on the screen.

DISABLE BLINK 004 (D2 compatible)

None of the characters displayed on the screen are blinked.

START BLINK 016 (D2 compatible)

Sets the Blink Attribute bit in the Current Attribute Register. This sets the blink attribute bit of each succeeding character as it is received. These characters blink if the screen is blink enabled (Blink Enable flag set).

END BLINK 017 (D2 compatible)

Clears the Blink Attribute bit in the Current Attribute Register. This clears the blink attribute bit on each succeeding character as it is received.

START DIM 034 (D2 compatible)

Sets the Dim attribute bit in the Current Attribute Register. This sets the Dim attribute bit of each succeeding character as it is received. These characters are displayed at reduced intensity (dimmed).

END DIM 035 (D2 compatible)

Clears the Dim attribute bit in the Current Attribute Register. This clears the Dim attribute bit of each succeeding character as it is received.

START UNDERSCORE 024 (D2 compatible)

Sets the Underscore attribute bit in the Current Attribute Register. This sets the Underscore attribute bit on each succeeding character as it is received. These characters are displayed with an underscore.

END UNDERSCORE 025 (D2 compatible)

Clears the Underscore attribute bit in the Current Attribute Register. This clears the Underscore attribute bit on each succeeding character as it is received.

Cursor Positioning

HOME 010 (D2 compatible)

The cursor moves to the first (leftmost) character position on the top line of the screen, which is the cursor home position.

NEW LINE 012 (D2 compatible)

The cursor moves to the first character position on the next line of the screen. If the cursor is on the bottom line, it moves to the home position, unless the terminal is roll enabled (refer to the ROLL ENABLE command described under heading "Screen Control").

CARRIAGE RETURN 015 (D2 compatible)

The cursor moves to the first character position on the line on which the cursor resides.

WRITE CURSOR ADDRESS 020 <X> <Y> (D2 compatible)

The display is forced to use the next two codes received as the cursor's new column and row (line) addresses. Both the columns and rows are numbered beginning with column 0, row 0. The columns are numbered from left to right across the screen; the rows are numbered from top to bottom. After the second character is received, the cursor moves to the location on the screen specified by the new coordinates.

READ CURSOR ADDRESS 005 (D2 compatible)

A sequence of three codes is sent from the display to the host computer. The first code is the ASCII control character 037 (Read Cursor Header); the second is the cursor's current column address; and the third is the cursor's current line address.
Terminal Response: 037 <X><Y>

CURSOR UP 027 (D2 compatible)

The cursor moves up one line while remaining in the same column position. If the cursor is on the top line, it moves to the bottom line of the screen.

CURSOR RIGHT 030 (D2 compatible)

The cursor moves one character (column) position to the right. If the cursor is at the end of the line, a New Line operation is performed.

CURSOR LEFT 031 (D2 compatible)

The cursor moves one character position to the left. If the cursor is in the leftmost position on the line, it moves to the rightmost position and then up one line.

CURSOR DOWN 032 (D2 compatible)

The cursor moves down one line while remaining in the same column position. If the cursor is on the bottom line, it moves to the top line.

Tab Control

SET TAB INTERVAL 036 021 <byte>

Set tab interval in terminal to value of <byte>. Terminal response; 036 176 057

TAB TO NEXT TAB STOP 011

Moves cursor to next tab stop. To actually store a tab code, use the Insert Character instruction to insert the 011 code. Terminal response: 036 176 057

TAB TO NEXT UNPROTECTED FIELD 036 004

Moves cursor to first location in next unprotected field. If in page mode, the same starting location will result in a cursor move to the first unprotected field at top of screen. Terminal response: 036 176 057

TAB TO PREVIOUS UNPROTECTED FIELD 036 005

Moves cursor to first character location of previous unprotected field. Terminal response: 036 176 057

SEND TAB CHARACTERS 036 030

Used when tabbed data is to be read back to the host; causes space characters to be flagged (bit 0 of character byte set) and a tab character substituted in their place. This instruction must be programmed prior to the data entry. Terminal response: 036 176 057

REPLACE TAB CHARACTERS 036 027

The opposite effect of the SEND TAB CHARACTERS instruction. Causes the individual space codes to be sent back to the host instead of the substitute tab character. Terminal response: 036 176 057

Printer Control

PRINT 021 (D2 compatible)

Keying: PRINT key (generates 036 021 code to host)
Data displayed on the screen, beginning with the leftmost character on the cursor line and ending with the rightmost character on the bottom line, is printed. During the print operation, the keyboard is disabled. Terminal response: 006

PRINT FORM 001 (D2 compatible)

Keying: SHIFT PRINT (generates 036 001 to host)
If terminal Sysgen is configured for 6053 compatibility, then this command will execute a print-out of screen data as described for the PRINT command except only the full intensity characters will be printed. All dim characters will be treated as spaces. If Sysgen is not configured for 6053 compatibility, the unprotected characters are sent to the printer. Terminal response: 006

PRINT LINE 036 007 <data string> < terminator>

The data string is sent to the terminal for transfer to the printer. Each data string can contain up to 132 characters including the terminator. Character strings in excess of 132 will be truncated. The terminator can be any one of the following:

TERMINATOR	CODES TO PRINTER
015	015 (carr. ret.)
012	015 012 (carr. ret + new line)
014	015 014 (carr. ret. + form feed - selects new page)
000	000 (null - stops printing at current location)

Note that the PRINT LINE instruction does not affect data on the screen, but does lock the keyboard while the data is being transferred. Terminal response: 036 176 057

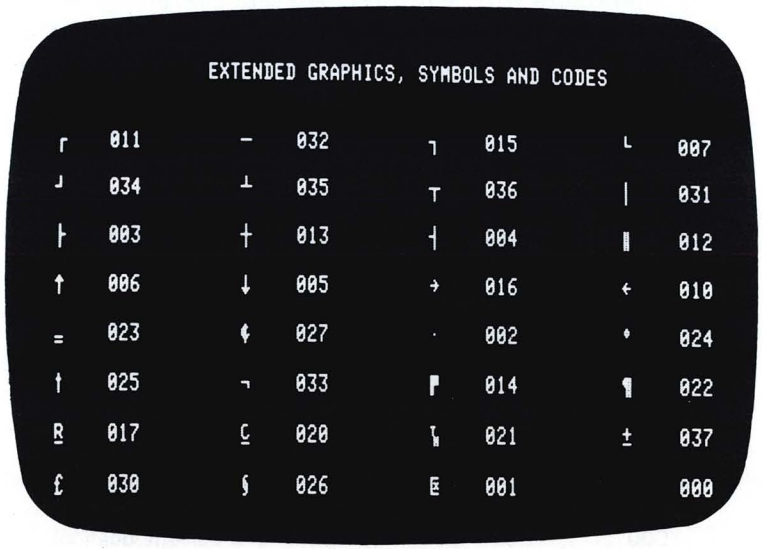
Extended Graphic Character Set

Individual characters of the 32-character Extended Graphic Character Set can be displayed by host instructions with the following two-byte format:

033 <character code>

Figure 3-4 shows the graphic characters and their codes (also see Appendix D). Since the codes include octal numbers 0-37, some care is required in using the Read Screen instruction when they are in use. To avoid confusion with the cursor position, use zero for both the XOR byte and word byte of the Read Screen instruction. Then the only cursor position that will be received is the first one of the terminal response. The host program can be structured to treat subsequent 037 codes as data.

3-4 Display of Extended Graphic Characters and their Codes



BUFFERED MODE PROGRAMMING

Basically, programming for buffered mode operation has two parts:

1. Preparation of the terminal for buffered operation
2. Host response to delimiter codes from the terminal.

In the first of these, preparing the terminal for buffered operation, the following operations are essential:

- Write any required screen format
- Define protected and unprotected fields
- Condition other character attributes as required
- Define delimiters
- Define Fill character, if required
- Define Tab interval, if required
- Position cursor at starting location
- Set buffered mode

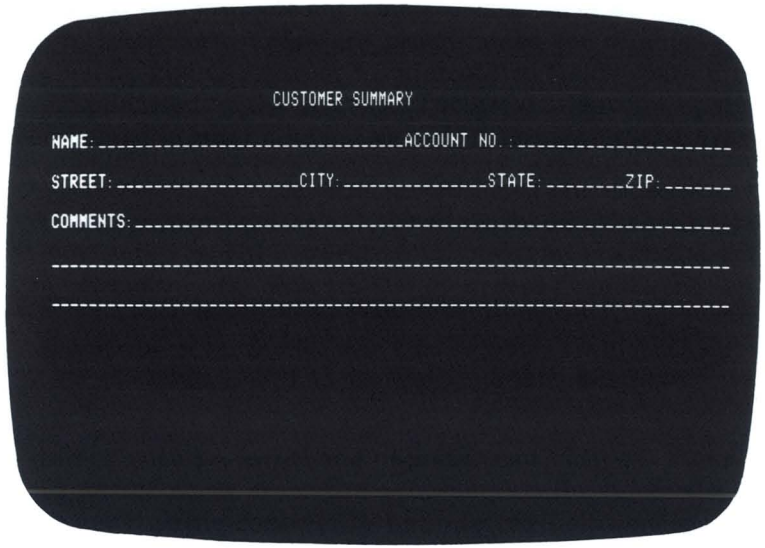
The host response to delimiter codes has the following operations:

- Set interactive mode to enable communication to terminal.
- Read screen data.
- Erase screen, either completely or by unprotected field.
- Write new screen format, if required.
- Set buffered mode.

Note that the response to different delimiter codes can be programmed for functions other than a Read Screen operation.

The following programming examples show how the terminal can be prepared for two types of buffered mode operations. In the first example, a sample format is written to the

3-5 Sample Screen For Protected/Unprotected Field Format



screen with protected and unprotected fields (see Figure 3-5). Protected form areas are programmed for dimmed display, thereby giving contrast to unprotected fields containing the data entered by the operator. The areas where the operator is expected to enter data are also identified by underscoring. The fill character is defined as a space code.

From viewpoint of the terminal operator, the program works in this manner:

1. The keyboard is locked while the form is written onto the screen.
2. The form consists of protected fields and data entry fields. The cursor is placed at the first position the operator is to enter data. The keyboard is unlocked and control goes to the operator.
3. The operator fills out the form and corrects any errors by using the buffered mode edit functions which enable deletions, insertions, and moving from field to field.
4. Upon completion of the data entry, the operator presses the SEND key (programmed delimiter), giving control to the host computer.
5. The host reads the unprotected fields (operator entered data) and prepares the screen for the next entry operation.

Table 3-6 gives a sample program for the screen format shown in Figure 3-5. The program is divided into functional catagories as follows:

1. Condition terminal for screen-write operation.
2. Write protected heads to screen.
3. Protect blank areas of the screen.
4. Set attributes for data entry areas (unprotected fields).
5. Condition terminal for buffered mode. Set buffered mode.
6. Service delimiter code from terminal (read screen).

3-6 Sample Program For Protected/Unprotected Field Format

CODING (octal)	FUNCTION
CONDITION TERMINAL FOR SCREEN WRITE	
014	Erase page
023	Disable Roll
WRITE PROTECTED HEADS	
020 030 004	Move cursor
036 017 000 040	Set Protect attribute
103 125 123 124 117 115 105 122 040 123 125 115 115 101 122 131	Write: CUSTOMER SUMMARY
020 000 006	Move cursor
036 017 040 010	Set Dim; maintain Protect
116 101 115 105 072	Write: NAME:
020 047 006	Move cursor
101 103 103 117 125 116 124 040 116 117 056 072	Write: ACCOUNT NO.:
020 000 010	Move cursor
123 124 122 105 105 124 072	Write: STREET:
020 034 010	Move cursor
103 111 124 131 072	Write: CITY:
020 060 010	Move cursor
123 124 101 124 105 072	Write: STATE:
020 075 010	Move cursor
132 111 120 072	Write: ZIP:
020 000 012	Move cursor
103 117 115 115 105 116 124 123 072	Write: COMMENTS:

3-6 (CONT.)

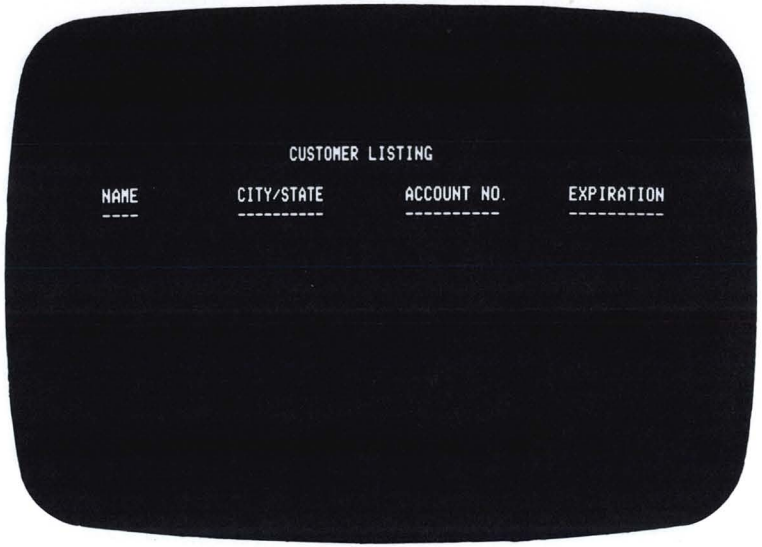
CODING (octal)	FUNCTION
PROTECT BLANK SCREEN AREAS	
020 000 017	Move cursor
036 002 000 000 040	Set Protect for rest of screen
036 002 177 000 040	Set Protect for blank areas
036 002 177 000 040	Set Protect for blank areas
036 002 132 000 040	Set Protect for blank areas
020 050 004	Move cursor
036 002 170 000 040	Set Protect for blank areas
020 000 007	Move cursor
036 002 120 000 040	Set Protect for blank areas
020 000 011	Move cursor
036 002 120 000 040	Set Protect for blank areas
020 000 013	Move cursor
036 002 120 000 040	Set Protect for blank areas
020 000 015	Move cursor
036 002 120 000 040	Set Protect for blank areas
PREPARE DATA ENTRY AREAS	
036 017 000 020	Set Underscore attribute; reset Protect
036 022 040	Set Fill Character = space
036 003	Fill unprotected fields
CONDITION TERMINAL FOR BUFFERED MODE	
020 005 006	Move cursor
036 020 036 022 036 176 057	Set delimiter = SEND key
036 001	Set Buffered Mode
SERVICE DELIMITER/READ SCREEN	
036 000	Set Interactive Mode
020 005 006	Move cursor
036 014 100 100 117 027	Read screen (modified data)
020 005 006	Move cursor
036 001	Set Buffered Mode

The second programming example shows how a tabular application could be implemented. In this example, a simple heading format is written to the screen (Figure 3-7) and is protected. The tab interval is defined to give four identical columns.

From viewpoint of the terminal operator, the program works in this manner:

1. The keyboard is locked while the tab format is written to the screen.

3-7 Sample Screen For Tabular Field Format



2. The operator keys the data into the screen tab columns. The cursor is moved from one tab stop to another by the TAB cursor positioning keys.
3. The operator presses the SEND key (programmed delimiter) to pass control to the host computer. Keyboard is locked.
4. The host clears the entered data from the screen and unlocks the keyboard.

Table 3-8 gives a sample program for the Tabular field format shown in Figure 3-7. The program is divided into functional catagories as follows:

1. Condition terminal for screen-write operation.
2. Write protected heads to screen.
3. Condition terminal for buffered mode. Set buffered mode.
4. Service delimiter code from terminal (read screen) and set buffered mode. ■

3-8 Sample Program For Tabular Field Format

CODING (octal)	FUNCTION
CONDITION TERMINAL FOR SCREEN WRITE	
014	Erase page
WRITE PROTECTED HEADS	
020 032 004	Move cursor
036 017 000 040	Set Protect attribute
103 125 123 124 117 115 105 122 040 114 111 123 124 111 116 107	Write: CUSTOMER LISTING
020 010 006	Move cursor
036 017 000 060	Set Protect/Underscore attributes
116 101 115 105	Write: NAME
020 027 006	Move cursor
103 111 124 131 057 123 124 101 124 105	Write: CITY/STATE
020 052 006	Move cursor
101 103 103 117 125 116 124 040 116 117 056	Write: ACCOUNT NO.
020 073 006	Move cursor
105 130 120 111 122 101 124 111 117 116	Write: EXPIRATION
CONDITION TERMINAL FOR BUFFERED MODE	
020 000 010	Move cursor
036 017 000 000	Clear attributes
036 021 024	Set Tab Interval = 20 (decimal)
036 020 036 022 036 176 057	Set delimiter = SEND key
036 022 040	Set Fill Character = space
036 001	Set Buffered Mode
SERVICE DELIMITER/READ SCREEN	
036 000	Set Interactive Mode
020 000 010	Move cursor
036 027	Set Replace Tab Mode
036 014 000 000 117 027	Read screen (all data)
020 000 010	Move cursor
036 001	Set Buffered Mode

APPENDIX A SYSGEN PARAMETERS

INTRODUCTION

Basic initialization parameters, such as baud rate, power line frequency, parity, etc., are set through the keyboard on the D3 rather than with back panel switches. This process of setting the parameters is called Sysgen. The parameters are stored in a separate non-volatile memory so that even when power is interrupted, the parameters last set are still in effect.

In general, it is expected that these parameters would be defined at the initial setup of the system and would only be changed if reconfiguration of system operation is desired. Conceivably, the responsibility for setting the Sysgen memory would be restricted to system supervisory personnel.

TO CHANGE OR EXAMINE PARAMETERS

Normally, the Sysgen parameters are not displayed unless called to the screen by the operator (see NOTE). The procedure to change or examine the Sysgen parameters is as follows:

- Press CMD and LINE keys simultaneously (selects local, off-line mode).
- Press CMD SHIFT LINE keys simultaneously (displays Sysgen screen format, see Figure A-1). The current parameter values are displayed in the CURRENT column and the cursor is initially in the top field of the NEW column.
- Key in the new parameter (if a change in parameter is desired). Only the symbols listed are accepted. If no entry is made, the current parameter stays in effect.
- Press NEW LINE key to advance to next parameter.
- To exit the Sysgen display, press the ESC key (works for any cursor location). Or, alternatively, if cursor is on the last parameter, press the NEW LINE key. With an exit, the screen is erased and the terminal re-initializes with the new contents (or old, if no changes were made) of the Sysgen.

A-1 Sysgen Screen Presentation

> NOTE
> In the event a STARTUP ERROR message is displayed on initialization, the number following the error message indicates to service personnel the type of malfunction. The terminal can usually be operated, even if an error is indicated; however, such operation requires that any modification to the default parameters be re-entered each time power is turned on.

DEFINITION OF PARAMETERS

Audible Tone Volume

Selects the audio level for the audible tone (Bell) alarm. Loud, Medium, or Soft can be selected. The default is Loud.

Cursor Presentation

Defines whether cursor will display as an Underscore or Reverse Video. Default is Underscore. In local mode, the cursor presentation is the compliment of the Sysgen selection.

AC Line Frequency

Configures terminal operation to either 50 hz or 60 hz input-power line frequency. Default is 60 hz.

Parity

Allows terminal to be configured to same parity mode as host for data transfer between terminal and host. Odd, Even, or Mark can be selected. Default is Mark.

Receive Baud Rate

Configures terminal for data bit transfer rate from host. Any one of following baud rates can be selected: 110, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200. The selected receive rate must be same as the host transmit rate. Default is 19200.

Transmit Baud Rate

Selects the data bit transfer rate to host. The selectable rates are: 75, 150, 300, or SAME (meaning same as the Receive Baud Rate).The selected transmit rate must be same as the host receive rate. Default is SAME.

Transmit Character Pacing Rate

Selects the rate that characters transfer to the host (characters per second). Since the rate that the individual data bits transfer are defined by the Transmit Baud Rate, this parameter in effect defines the time period between character transfers. The pacing rate should be selected to accomodate the interrupt response time of the host. As a result, a faster Transmit Baud Rate can be used to transfer the individual character bits, thereby reducing the overall data transfer time. The selectable pacing rates are as follows: 60, 120, 240, 480, or NONE (depends solely on Transmit Baud Rate). Default is NONE.

Keyboard Repeat Rate

Selects the rate which characters repeat when a key is held depressed in conjunction with the REPEAT key. Selectable repeat rates are: 10, 15, 30, or 60 Hz. Default is 10 Hz.

Printer Baud Rate

Selects the data bit transfer rate for transferring data from the terminal to a connected printer. The selectable rates are: 110, 150, 300, 600, 1200, 1800, or 2400. Default is 600 baud.

Modem Option

Since the modem option does not apply to the D3 terminal, NO should always be selected. Default is NO.

Fully 6053 Compatible

Defines whether the Print Form instruction functions the same as in the DASHER D2 Terminal. Either YES or NO can be selected. If YES is selected, the Print Form instruction causes the full intensity characters on the screen to transfer to the printer. Dim characters will have space codes sent in their stead. If NO is selected, unprotected data (characters without their protect attribute set) is sent. All other characters are treated as spaces. Default is NO.

Tab Interval

Defines the number of characters between Tab Stops (as initialized; host can redefine). Tab intervals from 0 to 80 (decimal) can be specified. Default is 4.

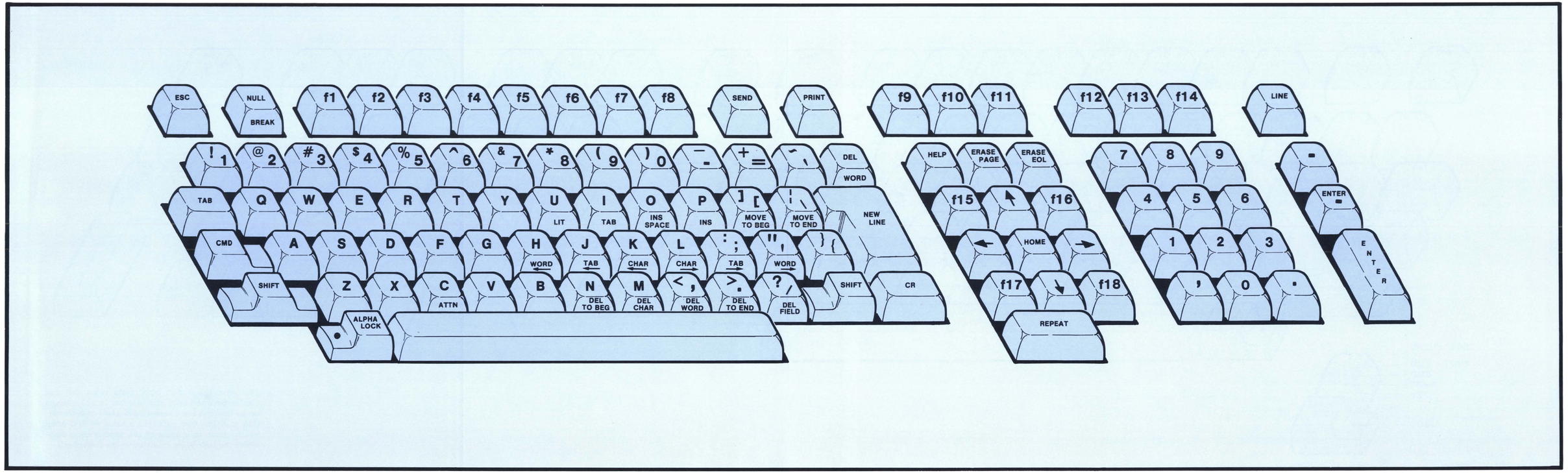
Fill Character

Defines the character that is used for fill operations (unless defined otherwise by the host). Any keyboard key can be pressed and will be displayed under NEW column as an alphanumeric character (its octal value will be displayed when in the CURRENT column). Default value is 40 (space).

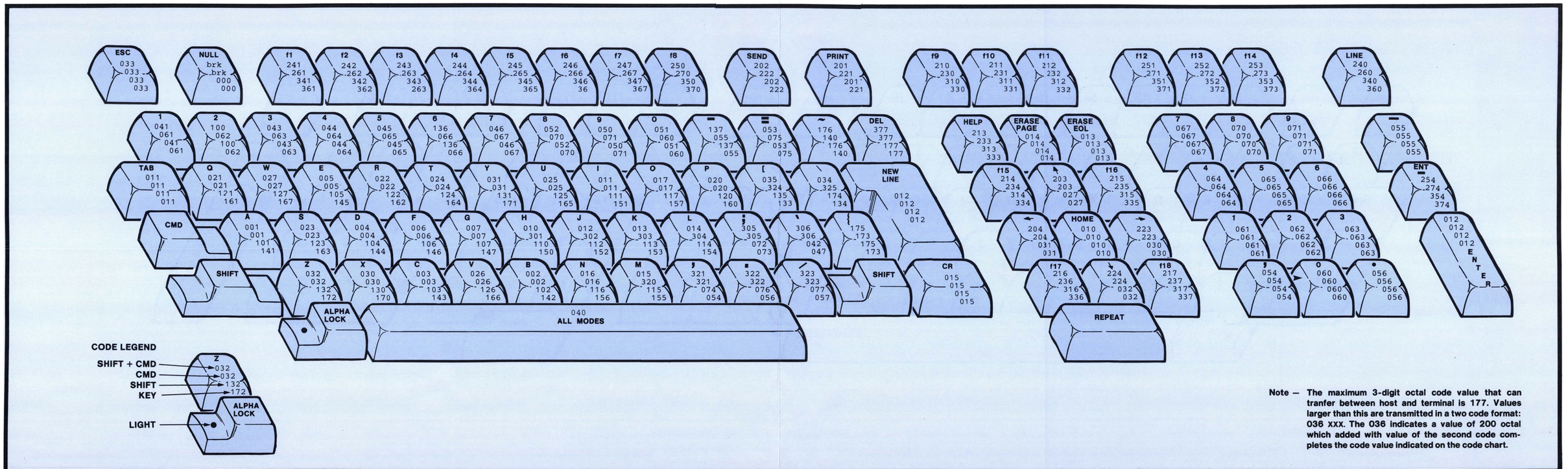
Terminal Number

Defines the assigned terminal number in a multi-terminal system. It is used as identification to host on an ID Request instruction. Requires a decimal numeric value for an entry. Default is 0.■

APPENDIX B KEYBOARD LAYOUT

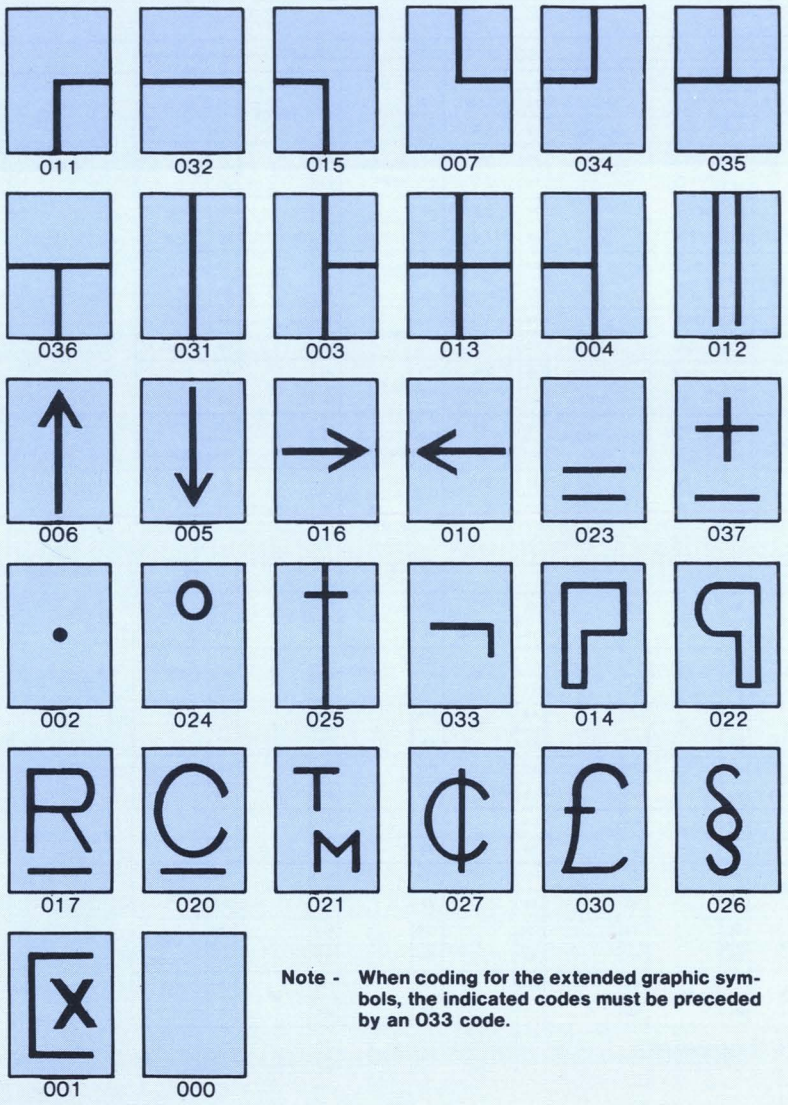


APPENDIX C KEYBOARD CODE ASSIGNMENTS



Note — The maximum 3-digit octal code value that can tranfer between host and terminal is 177. Values larger than this are transmitted in a two code format: 036 xxx. The 036 indicates a value of 200 octal which added with value of the second code completes the code value indicated on the code chart.

APPENDIX D EXTENDED GRAPHIC SYMBOLS AND CODES



Note - When coding for the extended graphic symbols, the indicated codes must be preceded by an 033 code.

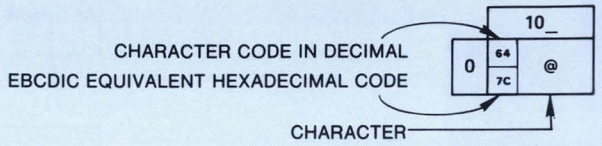
APPENDIX E ASCII CODES

OCTAL	00_		CONTROL FUNCTION	01_		CONTROL FUNCTION	02_		CONTROL FUNCTION	03_		CONTROL FUNCTION	04_		05_	
0	0 00	NUL	NULL	8 16	BS *(BACK-SPACE)*	HOME	16 10	DLE ↑P	WRITE CURSOR HEADER	24 18	CAN ↑X	CURSOR RIGHT	32 40	SPACE	40 4D	(
1	1 01	SOH ↑A	PRINT FORM	9 05	HT *(TAB)*	HORIZONTAL TAB	17 11	DC1 ↑Q	PRINT PAGE	25 19	EM ↑Y	CURSOR LEFT	33 5A	!	41 5D	)
2	2 02	STX ↑B	START OF TEXT	10 15	NL *(NEW LINE)*	NEW LINE	18 12	DC2 ↑R	ROLL ON	26 3F	SUB ↑Z	CURSOR DOWN	34 7F	" *(QUOTE)*	42 5C	*
3	3 03	ETX ↑C	ENABLE BLINK	11 0B	VT *(VERT. TAB)*	ERASE LINE	19 13	DC3 ↑S	ROLL OFF	27 27	ESC *(ESCAPE)*	ESCAPE	35 7B	#	43 4E	+
4	4 37	EOT ↑D	INHIBIT BLINK	12 06	FF *(FORM FEED)*	ERASE PAGE	20 3C	DC4 ↑T	UNDERSCORE ON	28 1C	FS ↑\	DIM ON	36 5B	$	44 6B	, *(COMMA)*
5	5 2D	ENQ ↑E	CURSOR ADDRESS READ	13 0D	RT *(RETURN)*	RETURN	21 3D	NAK ↑U	UNDERSCORE OFF	29 1D	CS ↑]	DIM OFF	37 6C	%	45 60	–
6	6 2E	ACK ↑F	PRINT DONE	14 0E	SO ↑N	BLINK ON	22 32	SYN ↑V	SYNCHRONOUS IDLE	30 1E	RS ↑↑	FUNCTION KEY HEADER	38 50	&	46 4B	. *(PERIOD)*
7	7 2F	BEL ↑G	BELL	15 0F	SI ↑O	BLINK OFF	23 26	ETB ↑W	CURSOR UP	31 1F	US ↑←	READ CURSOR HEADER	39 7D	' *(APOS)*	47 61	/

OCTAL	06_		07_		10_		11_		12_		13_		14_		15_		16_		17_	
0	48 F0	Ø	56 F8	8	64 7C	@	72 C8	H	80 D7	P	88 E7	X	96 79	` *(GRAVE)*	104 88	h	112 97	p	120 A7	x
1	49 F1	1	57 F9	9	65 C1	A	73 C9	I	81 D8	Q	89 E8	Y	97 81	a	105 89	i	113 98	q	121 A8	y
2	50 F2	2	58 7A	:	66 C2	B	74 D1	J	82 D9	R	90 E9	Z	98 82	b	106 91	j	114 99	r	122 A9	z
3	51 F3	3	59 5E	;	67 C3	C	75 D2	K	83 E2	S	91 8D	[	99 83	c	107 92	k	115 A2	s	123 C0	{
4	52 F4	4	60 4C	<	68 C4	D	76 D3	L	84 E3	T	92 E0	\	100 84	d	108 93	l	116 A3	t	124 4F	|
5	53 F5	5	61 7E	=	69 65	E	77 D4	M	85 E4	U	93 9D	]	101 85	e	109 94	m	117 A4	u	125 D0	}
6	54 F6	6	62 6E	>	70 C6	F	78 D5	N	86 E5	V	94 5F	↑ or ^	102 86	f	110 95	n	118 A5	v	126 A1	~ *(TILDE)*
7	55 F7	7	63 6F	?	71 C7	G	79 D6	O	87 E6	W	95 6D	← or _	103 87	g	111 96	o	119 A6	w	127 07	DEL *(RUBOUT)*

CHARACTER CODE IN OCTAL AT TOP AND LEFT OF CHARTS.

LEGEND



↑ MEANS CMD

reader comment form

Dasher D3 Display Terminal

Name _____ Title _____

Firm _____ Date _____

Street _____ Zip _____

City _____ State _____

Your comments will help us improve the quality of this publication. If your answer to a question is "NO" or requires qualification, please explain.

HOW DID YOU USE THIS PUBLICATION?

- ☐ As an introduction to the subject
- ☐ For information about operating procedures
- ☐ To instruct in a class
- ☐ As a student in a class
- ☐ As a reference manual
- ☐ Other *(please explain):*

DID YOU FIND THE MATERIAL:

	YES	NO
• Useful	☐	☐
• Complete	☐	☐
• Accurate	☐	☐
• Well organized	☐	☐
• Well written	☐	☐

	YES	NO
• Well illustrated	☐	☐
• Well indexed	☐	☐
• Easy to read	☐	☐
• Easy to understand	☐	☐

We would appreciate any other comments. Please refer to page numbers if appropriate. Your comments will be carefully reviewed by the writers of this document.

COMMENTS:

FOLD

FIRST CLASS
PERMIT NO. 26
Westboro, Mass. 01581

BUSINESS REPLY MAIL No Postage Necessary If Mailed In The United States

Postage will be paid by:

Data General Corporation

Westboro, Massachusetts 01581

Attn:

SMALL BUSINESS SYSTEMS DOCUMENTATION
MAIL STOP B 61

FOLD

PLEASE TAPE CLOSED - (NO STAPLES)

reader comment form

Dasher D3 Display Terminal

Name _____ Title _____

Firm _____ Date _____

Street _____ Zip _____

City _____ State _____

Your comments will help us improve the quality of this publication. If your answer to a question is "NO" or requires qualification, please explain.

HOW DID YOU USE THIS PUBLICATION?

- ☐ As an introduction to the subject
- ☐ For information about operating procedures
- ☐ To instruct in a class
- ☐ As a student in a class
- ☐ As a reference manual
- ☐ Other *(please explain):*

DID YOU FIND THE MATERIAL:

	YES	NO
• Useful	☐	☐
• Complete	☐	☐
• Accurate	☐	☐
• Well organized	☐	☐
• Well written	☐	☐

	YES	NO
• Well illustrated	☐	☐
• Well indexed	☐	☐
• Easy to read	☐	☐
• Easy to understand	☐	☐

We would appreciate any other comments. Please refer to page numbers if appropriate. Your comments will be carefully reviewed by the writers of this document.

COMMENTS:

FOLD

FIRST CLASS
PERMIT NO. 26
Westboro, Mass. 01581

BUSINESS REPLY MAIL No Postage Necessary If Mailed In The United States

Postage will be paid by:

Data General Corporation

Westboro, Massachusetts 01581

Attn:

SMALL BUSINESS SYSTEMS DOCUMENTATION
MAIL STOP B 61

FOLD

PLEASE TAPE CLOSED - (NO STAPLES)

Dasher D3 Instruction Summary

INSTRUCTION	CODING TO TERMINAL	TERMINAL RESPONSE TO HOST
TERMINAL MANAGEMENT		
master reset	036 025	none
ID request	036 012	036 176 043 040 <Rev #><term. #> 036 176 057
unlock keyboard	036 031	036 176 057
set interactive mode	036 000	036 176 057
set buffered mode	036 001	none
read status	036 024	036 176 045 <byte 1> . . . <byte 6> 036 176 057
*bell	007	none
SCREEN CONTROL		
*roll enable	022	none
*roll disable	023	none
read screen	036 014 <mask><word><XE> <YE>	036 176 040 <037 X Y><data string> 036 176 057
set delimiter table	036 020 <byte><byte> . . 036 176 057	036 176 057
read delimiter table	036 023	036 176 044 <byte 1><byte 2> . . .etc. 036 176 057
set fill character	036 022 <byte>	036 176 057
fill unprotected fields	036 003	036 176 057
fill character string	036 013 <# locations><ins char>	036 176 057
insert character	036 010 <# char><ins char>	036 176 057
block move/fill	036 006 <XD><YD><XE><YE>	036 176 057
delete character	036 011 <# char>	036 176 057
*erase to end of line	013	none
*erase page	014	none
ATTRIBUTE CONTROL		
set current attribute register	036 017 <mask><XOR>	036 176 057
read current attribute register	036 016	036 176 042 <register byte> 036 176 057
read attribute string	036 015 <XE><YE>	036 176 041 <byte><byte> . . .etc. 036 176 057
set attribute string	036 002 <# char><mask><XOR>	036 176 057
*enable blink	003	none
*disable blink	004	none
*start blink	016	none
*end blink	017	none
*start dim	034	none
*end dim	035	none
*start underscore	024	none
*end underscore	025	none
CURSOR POSITIONING		
*home	010	none
*new line	012	none
*carriage return	015	none
*write cursor address	020 <X><Y>	none
*read cursor address	005	037 <X><Y>
*cursor up	027	none
*cursor right	030	none
*cursor left	031	none
*cursor down	032	none
TAB CONTROL		
set tab interval	036 021 <byte>	036 176 057
tab to next tab stop	011	036 176 057
tab to next unprotected field	036 004	036 176 057
tab to previous unprotected field	036 005	036 176 057
send tab characters	036 030	036 176 057
replace tab characters	036 027	036 176 057
PRINTER CONTROL		
*print	021	none
*print form	001	none
print line	036 007 <data string><terminator>	036 176 057

LEGEND: * D2 compatible

NOTES:

1. Screen coordinates are:

   X = horizontal positioning (column); limits: 0 to 79 (117 octal)
   Y = vertical positioning (row, line); limits: 0 to 23 (027 octal)
   XE = ending column
   YE = ending row
   XD = destination column
   YD = destination row

2. Character attributes are:

BIT #	ATTRIBUTE
7	block fill
6	blink
5	reverse video
4	dim
3	underscore
2	protected data
1	modified data

3. Attribute selection using Mask and XOR functions:

MASK BIT	XOR BIT	RESULTING ATTRIBUTE BIT
0	0	0 (cleared)
0	1	1 (set ON)
1	0	no change
1	1	complement

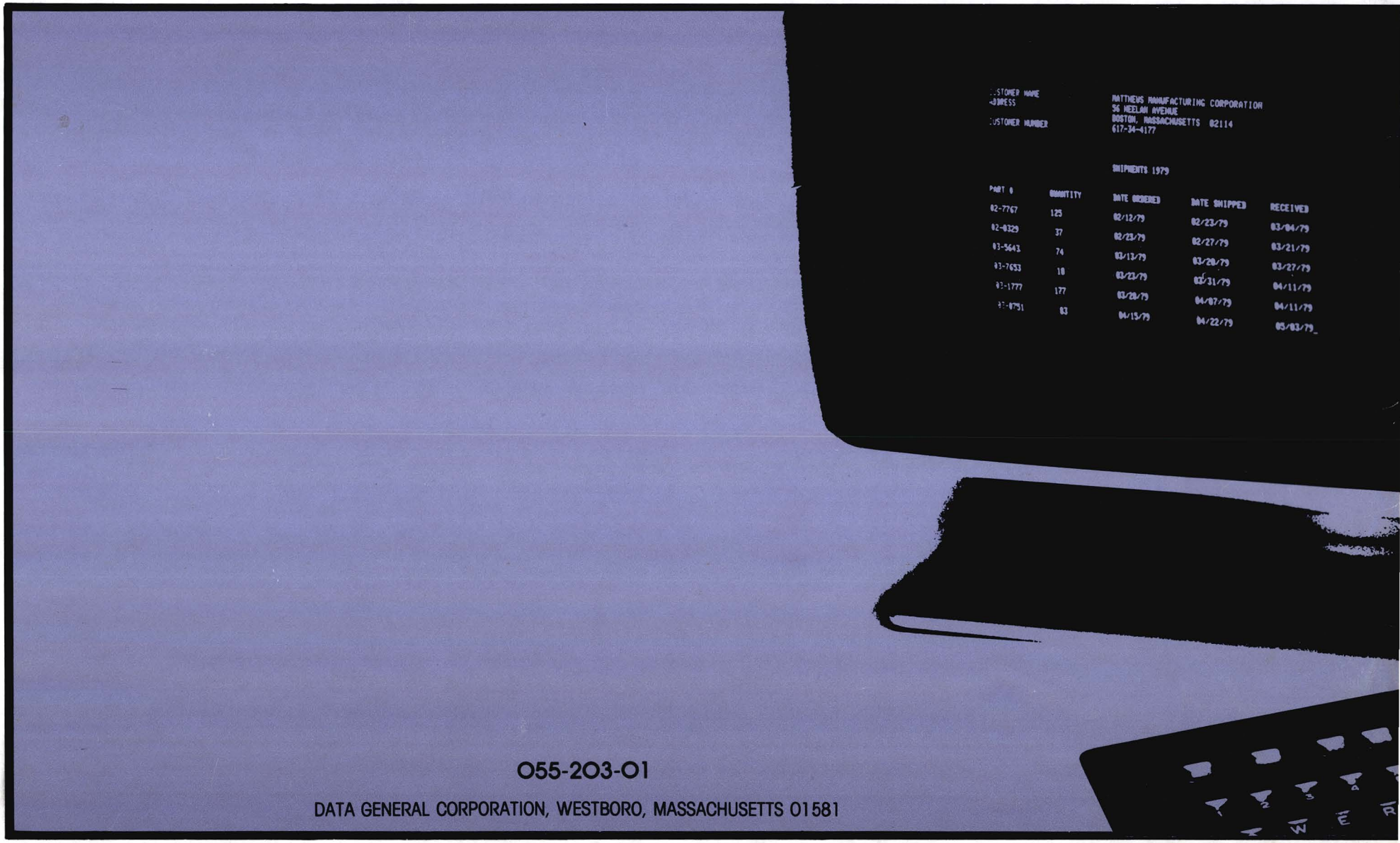
4. Print Line terminators are:

TERMINATOR	CODES TO PRINTER
015	015 (carr. ret.)
012	015 012 (carr. ret + new line)
014	015 014 (carr. ret. + form feed - selects new page)
000	000 (null - stops printing at current location)

5. Master Reset initializes terminal as follows:

REGISTER/FLAG	CONDITION
Terminal Mode Register	Interactive
Keyboard Lock Register	Unlocked
Tab Read Mode Register	Disabled (reads spaces)
Tab interval Register	Current Sysgen (default is 4)
Fill Character Register	Current Sysgen (default is space)
Terminal Number Register	Current Sysgen (default is 0)
Delimiter Table	New Line, CR
Current Attribute Register	Cleared
Cursor Address Register	000 000 (Home)
Blink Enable Flag	Enabled
Roll Enable Flag	Enabled

Plus the various other Sysgen functions according to current Sysgen configuration (see Appendix A).



055-203-01

DATA GENERAL CORPORATION, WESTBORO, MASSACHUSETTS 01581